Spring 5-29-2022

MIDDLE SCHOOL STUDENTS COMMUNICATING COMPUTATIONAL THINKING: A SYSTEMIC FUNCTIONAL LINGUISTICS-CASE STUDY OF BILINGUAL, COLLABORATIVE TEACHING/LEARNING OF COMPUTER PROGRAMMING IN PYTHON

Jose Antonio Lecea Yanguas

<u>José Antonio Lecea Yanguas</u>
Candidate

<u>Language, Literacy, and Sociocultural Studies</u>
Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Rebecca Blum-Martínez, Chairperson and advisor

Dr. Sylvia Celedón-Pattichis

Dr. Marios S. Pattichis

Dr. José Ignacio Navarro Guzmán, Universidad de Cádiz

Dr. José Fernando Calderero, Universidad Internacional de la Rioja

MIDDLE SCHOOL STUDENTS COMMUNICATING COMPUTATIONAL THINKING: A SYSTEMIC FUNCTIONAL LINGUISTICS-CASE STUDY OF BILINGUAL, COLLABORATIVE TEACHING/LEARNING OF COMPUTER PROGRAMMING IN PYTHON

BY

JOSÉ ANTONIO LECEA YANGUAS

B.A., Early Childhood Education, Universidad Autónoma de Madrid, 2011
M. A., Language, Literacy and Sociocultural Studies, University of New Mexico, 2015

DISSERTATION
Submitted in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

Language, Literacy and Sociocultural Studies

The University of New Mexico
Albuquerque, New Mexico

May, 2022

This study was supported by the National Science Foundation-innovative technology experiences for students and teachers (ITEST) award #1613637 and #1949230. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the funding agency.

DEDICATION

I dedicate this doctoral thesis to Reyes and the rest of my family. And to all the professors, friends and students who accompanied me in this journey. Muchas gracias.

ACKNOWLEDGEMENTS

Many people have supported me through the process that made this dissertation possible. A lot of professors, colleagues, friends, and family have been fundamental.

First, I thank Professor Emeritus Rebecca Blum-Martínez, my advisor and the chair of my dissertation committee. She has guided my work masterfully since its earliest stages. Professor Blum-Martinez always pushed my work, always exercised care and trust on my academic, professional, and personal endeavors. Her mentoring has been fundamental. I will always be grateful.

I also thank the rest of my dissertation committee: Professor Sylvia Celedón-Pattichis, Professor Marios Pattichis, Professor José Ignacio Navarro Guzmán and Professor José Fernando Calderero. They have provided invaluable support, shared their expertise with me and helped me improve my research.

In addition, I thank the AOLME team in its entirety. I had unforgettable times as part of it. AOLME has been an invaluable source of motivation. The AOLME principal investigators gave me the opportunity to conduct this study and be part of the team. I will always be grateful. Many other people have been important in this journey. My research has benefited greatly from the conversations I had with so many colleagues, friends, and family members who showed interest. I want to thank all of them too.

Last but not least, I appreciate the kindness and love I received from so many people in New Mexico and elsewhere during the years that I devoted to the dissertation process. Siempre llevaré en el corazón a mis herman@s Nuevomexicanos.

Above all, Reyes has been an unbelievable source of love and inspiration who has supported me in the difficult moments of this journey and marked it with joy.

MIDDLE SCHOOL STUDENTS COMMUNICATING COMPUTATIONAL THINKING: A SYSTEMIC FUNCTIONAL LINGUISTICS-CASE STUDY OF BILINGUAL, COLLABORATIVE TEACHING/LEARNING OF COMPUTER PROGRAMMING IN PYTHON

By

José Antonio Lecea Yanguas

B.A., EARLY CHILDHOOD EDUCATION

M. A., LANGUAGE, LITERACY AND SOCIOCULTURAL STUDIES

PH.D., LANGUAGE, LITERACY AND SOCIOCULTURAL STUDIES

ABSTRACT

This dissertation presents the first Systemic Functional Linguistics-based analysis of the teaching/learning of computational thinking through computer programming and comprehensive analysis of discourse of a whole computer programming course at any educational level. The current educational research raises questions about the nature of authentic computational thinking teaching/learning environments and how they happen moment-to-moment. In one such environment, I examined the discourse of a facilitator, three students, and their Language Arts teacher in an introductory middle school after-school course (approximately 30 hours) in spring 2017 as students created a video in Python.

Methodologically, I show how a Systemic Functional Linguistics-based analytical framework can operationalize the dimensions of an authentic bilingual (English-Spanish) computer programming environment, student positioning and indicators of computational thinking learning. I identify the following dimensions: complexity (abstraction included), pragmatism, procedurality, dependency, and flexibility. The facilitator positioned the students as capable computational thinkers and computer programmers whose prior world experience and linguistic identity mattered. She also positioned them to collaboratively model their prototypes with grade-level mathematics; create the algorithm; communicate algorithm thinking and computational thinking. I identify relevant teaching strategies; indicators of student learning were found. Strategies include (1) drawing on the students' languages and cultural resources, (2) capitalizing on student-known mathematical concepts, (3) using a soft focus on concepts, (4) adopting a motivational, pragmatic, mathematics-based heuristic procedure.

My findings illuminate the nature of authentic computational thinking environments and suggest teaching practices that prioritize student creation and communication of meaningful, simple algorithms and programs over complex conceptual explanations.

"Keywords": Bilingual, computational thinking, computer programming, discourse analysis, positioning, English learners, levels of abstraction, modeling, mathematics, Systemic Functional Linguistics.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

> *"The time is soon coming where basic computational thinking and the ability to develop software will be considered a basic skill necessary to every discipline, a requirement for many jobs and an essential skill akin to arithmetic."* (Bourke, 2018, p. 1)

This dissertation addresses the question of how the teaching of computational thinking through computer programming (CT-CP) can be delivered in effective and attractive ways to all students, especially to adolescent culturally and linguistically diverse (CLD) students. We are in the era of big data and automation. Computers pervade all spheres of human experience, be them personal, social, or professional. If we want children to have the chance to, rather than just consume technology, participate in the creation of the computation-based products, systems and devices that rule and facilitate today's world, we need to intensify efforts to seek the development of CT-CP in all students. There is much room for research on the matter of how best to teach CT-CP, what authentic CT-CP environments look like and how students are positioned in CT-CP practices where they can learn. The main aim of this dissertation has been to explore motivational educational gateways that enable students to communicate with computers and develop the specific ways of thinking that are needed to create computer programs. This kind of thinking is referred to as CT, a kind of thinking that makes possible the formulation of a task or problem in a way that a computer can effectively carry out (Grover & Pea, 2018). CT is also the kind of thinking that enables humans to program computers and think at multiple levels of abstraction (Wing, 2008). Wing (2006) advocated CT for all, not just for computer scientists.

However, CT-CP[1] and more generally CS have been reported to be powerful discourses (Dufva and Dufva, 2019; Santo, et al., 2020) which are difficult to teach and learn (NRC, 2011; Grover & Pea, 2013; Reppenning et al., 2016) Furthermore, participation in CT-CP discourses is very restricted in the United States, especially for CLD students (Goode, et al. 2020; Margolis et al., 2017; Santo et al., 2020). Despite the growing interest in CT and in CS in general, and of the efforts devoted to its education, these fields witness the lowest participation of students of color of any other STEM (Science, technology, mathematics, and engineering)-related area (Goode et al. 2020). This low participation is particularly relevant given the “high-status nature” of computer science and the “tremendous levels of power and influence that lie with those who have stature in this field” (Goode et al., 2020, p. 2). For students to use the language of computing, not just from the common user perspective, but from the standpoint of those that produce digital technology, CLD students need to develop CT-CP skills since CT-CP transfers across domains (Barr & Stephenson, 2011; Grover & Pea, 2018). Today’s CT-CP students will be tomorrow’s producers of the software and hardware of the digital devices that will be facilitating and controlling our lives. Society cannot afford to renounce CLD students’ participation (Noddings, 2018) in a demographic context it is expected that the Latino/a population will reach 132 million by 2050 (30% of US population) (Jacob et al, 2020). In a study conducted by Google it was found that Black and Latino/a students were less likely to pursue CS for reasons which included lack of opportunities to learn it and lack of role models (Google Inc. & Gallup Inc.,

[1] Computational thinking is a necessary computer programming skill in education (Grover & Pea, 2013; Wing, 2006) and computer programming (CP) is a part of computer science (CS) (Bourke, 2018) Therefore the terms CS and CP inherently include computational thinking. Additionally, CT is most taught and learned through CP. For these reasons, I use hereafter CT-CP when I refer to CT in CP environments and CT-CP/CS to include CS in my considerations or refer to CS educational research that concerns CT-CP. It’s also relevant to note that, according to Shutte et al., (2018), CT is broader than CS.

2016). Among the scientists and engineers working on science and engineering occupations in 2015, only 4 percent were Hispanic/Latino males and 2 percent Hispanic/Latino females (National Science Foundation, 2017). Although research has consistently denounced this situation, it remains unresolved (MacLean, 2017). Therefore, the development of CT-CP has become a paramount issue from ethical-equity and political perspectives since it determines who has and will be given opportunities to participate in CT-CP literacies and how (Papert, 1980; Vogel et al., 2020).

Although much work remains to be done in broadening the participation of CT-CP to all students (Vakil, 2018), leading institutions, associations and initiatives such as The National Research Council (NRC, 2010, 2011), the Computer Science Education Association (CSTA) (CSTA Standards Task Force, 2017) and *Computer Science for All* (K12 Computer Science Framework Steering Committee, 2016) have recommended to take in consideration equity, diversity and ethics both in research and practice to equip all students with the necessary skills to become, not just consumers, but creators of computation-based technology. Importantly for equity and ethical matters, there is a significant body of research that targets broadening participation of underrepresented groups in CT-CP/CS (eg., Goode et al., 2020; Mouza et al., 2020). This dissertation contributes to these efforts. My hope is that its findings contribute to the betterment of the CT-CP/CS education of CLD adolescent and younger students, and more generally of all students, both in and out of school.

This study centers more specifically on the teaching/learning of CT-CP at the middle school age, a period when students generally undergo a more evident ability to think in concepts and to be creative (Mahn, 2015; Vygotsky, 1978; Vygotsky, 1986). CT-CP/CS education curricula in the US derive, for the most part, from cognitive perspectives to learning such as constructivism (Kafai et al., 2019). However, according to Kafai et al., 2020), other frames that focus situational perspectives, student participation and social justice concerns are also part of the CT-CP educational landscape. While research has documented relevant pedagogical approaches to CT-CP (e.g., Grover et al. 2015; Kandemir et al. 2021; Waite, 2017), and specifically to second language learners (Jabob et al., 2018) there is much left to be understood about how CT-CP can be taught effectively in ways that encourage student participation and are inclusive of all students, and about how CT-CP happens during teaching/learning interactions.

This dissertation explores through linguistic perspectives and methods an authentic CT-CP teaching/learning environment where a small group of CLD students (English-Spanish bilinguals) produced after school a video in CP language Python with the guidance of a CT-CP facilitator (English-Spanish bilingual). More specifically it adopts Systemic Functional Linguistics (SFL) perspectives and methods, a powerful resource to explore educational environments discursively, including relevant educational aspects such as student agency and how students are positioned during teaching and learning. To extend SFL affordances, I have combined SFL with case study perspectives and methods, which afford additional resources to decipher what is achieved through discourse in product-oriented environments such as the one under focus while enabling triangulation of understandings.

The Logics of this Dissertation

The logic of this dissertation revolves around the power that Humans have to think, communicate, and achieve common goals by means of resources. Computers, computer programs and programming languages are resources, just like natural languages such as English and Spanish are resources. So are pencils and mathematics (Lenhard & Carrier, 2017). All these resources help us participate, communicate, and accomplish things that we would not be able to accomplish without them, or simply make our lives easier. Basically, two logics are involved in this dissertation. Logic 1 involves the logic of the educational and collaborative computational thinking through computer programming (CT-CP) practices that the participants of this study experienced, and I examined. Logic 2 involves the logic that I used in their examination. The former hinges fundamentally on the human logical thinking needed in the communication with powerful processors of information, i.e., computers, where mathematics plays a fundamental role in that it can provide the formalisms that are needed to make human-computer communication possible (Ben-Ari, 2012). The latter hinges on SFL, a theory of language use in context that provides powerful resources for the examination of human experience in consideration of the environment wherein it happens (Halliday & Matthiessen, 2014). In this study I strived to not lose attention to the alignment of both logics.

Logic 1

Cognition and natural languages, such as English and Spanish, are key resources used by *Humans* to participate in cultural activities, make meaning of them, pursue goals together, fulfil tasks and solve problems, and develop products. To work together with computers toward fulfilling complex tasks we rely on cognition and communication, which can be considered tantamount (Sfard, 2008). To communicate, we rely on conventions that were already there for us and on ever-new meanings that we establish interpersonally as we communicate with others (Halliday & Matthiessen, 2014). Effective human communication requires agreement on our intentions, common focal attention and clarity of the discursive focus that is used (i.e., that participants refer to the same things when using the same words) (Sfard, 2000). The effectiveness of natural language-mediated human communication has limitations because natural languages can only account for a part of our semiotic potential–although fundamental (Halliday & Matthiessen, 2014)–and because of the misexpressions, misunderstandings and disagreements that pervade our interactions (Sfard, 2000). Still, very often, we are able to, even through imperfect communication, accomplish our goals, fulfill tasks, and develop products in collaboration. However, communication with computers is unforgiving (Bourke, 2018). Computers are highly complex human resources that do not think, but process information and execute human instructions, provided instructions are formulated and communicated in formal and very precise ways. Two overarching questions of this dissertation are: How can CT-CP be presented to students in attractive ways that stimulate their thinking and communication with computers? How can CT-CP practices be delivered, especially to CLD students, in ways that foster their active participation?

Logic 2

To conduct this research, I adopted the logic of SFL, a theory of language and human experience which affords the examination of teaching/learning of CT-CP as it unfolds. Given that the CT-CP practices under focus aimed at the creation a video in CP language Python, I extended SFL perspectives and methods with case study perspectives and methods (Tracy, 2019) in consideration of the environment wherein this dissertation was situated where participation, practice and producing artifacts were intimately related (Wenger, 2010). Combining SFL affordances with case study methods has already been done in mathematics education research to study the discourse of facilitators (Liston, 2018). In this study, this combination enabled the synchronous examination of participant discourse (facilitator, students, and teacher discourse) with the Python program and digital image and video representations that were being developed by the participant students. SFL provides powerful resources to examine moment-to-moment teaching/learning interactions in consideration of the broader environments wherein it happens (e.g., Halliday & Matthiessen, 2014; Schleppegrell, 2004, 2012). In the last decades multilingual research has established relevant comparisons between the grammars of different languages (Caffarel et al., 2004) and shown evidence of fundamental similarities between English and Spanish (García, 2013; Lavid et al., 2010; Quiroz 2017). This has opened the ground for SFL-based educational studies in Spanish in Science (Mizuno et al., 2018; Rudolph et al., 2020) and mathematics (Fierro Lucero, 2020; López Acosta & Rodríguez Vergara, 2021).

Various studies have used SFL to investigate mathematics education through an equity lens (e.g. Alshwaikh & Morgan, 2018; Herbel-Eisenmman & Wagner, 2007; Scheleppegrell, 2012) More specifically SFL has been used to focus agency in mathematics (Morgan, 2005), to adopt an alternate focus between interactant exchanges of action and information (DeJarnette , 2014) and to explore science experiential domains (Mizuno, 2018). To examine student agency in CT-CP practices is relevant because agency has been directly correlated with learning (Farnsworth et al., 2016; Lave and Wenger, 1991; Sfard, 2001, 2008; Wenger, 1998, 2010; Wenger-Trayner & Wenger-Trayner, 2015). During the SFL-based analysis conducted, I identified signals of student learning in the discursive agency the participant CLD students of this study exerted during CT-CP practices.

The alignment of logics of CT-CP and SFL-based logic and perspectives (in combination with case study perspectives and methods and relevant educational theoretical perspectives) enabled the examinations, findings, explanations, and interpretations that follow. In this dissertation, while providing relevant CT-CP pedagogical strategies, I explain the nature of the actual CT-CP domain presented by the facilitator to her students (Halliday & Matthiessen; 2014; Wenger-Trayner & Wenger-Trayner, 2015). In addition, I explain how the students were positioned, the roles that the participants performed and the role that mathematics and CP concepts played. Exploration through the logics that I just described revealed student learning and communication of CT, which I also detail.

Following, I explain the philosophical approach and interdisciplinary nature of this dissertation. Later, I provide background on the relevance of investigating the teaching/learning CT-CP with special consideration to culturally and linguistically diverse (CLD) students. After that, I anticipate relevant outcomes of this study. Then, I discuss my positionality, describing my journey into computational thinking and educational research. Next, I describe the environment[2] where this dissertation was situated, the Advancing Out-of-school Learning in Mathematics and Engineering (AOLME) project (Celedón-Pattichis et al., 2013; Pattichis et al., 2017; LópezLeiva et al., 2019). AOLME (The Advancing Out-of-school Learning in Mathematics and Engineering). As a Community of Practice (CoP) (Farnsworth, 2016; Lave and Wenger, 1991; Wenger, 1998; Wenger, 2010; Wenger-Trayner & Wenger-Trayner, 2015), AOLME (See Appendix A) was located in the US Southwest with a research and educational agenda

[2] More details about AOLME are discussed in chapter 4.

implemented in a rural school after regular school time. Finally, I describe the remaining structure of this paper and its research questions (RQs)

Philosophical and Interdisciplinary Approach

This dissertation is informed by philosophical, historical, disciplinary, and ethical-equity considerations that situate it with respect to relevant issues such as current pedagogies, the actual nature of the CT-CP practices offered to students as it is taught and learned, who is invited to participate in CT-CP (CS) fields and the positions and roles that are offered to students.

Ontologically, in this dissertation I consider all the above as well as the existence of student agency to influence and change the world (Wenger, 2010) and the impact that social issues such as status, gender, race, ethnicity, have on participation (O'Connor & Michaels, 2019). Epistemologically, I held an overarching ethical-equity perspective based on Bernstein's (2000) pedagogical rights. Through this lens, I examined participant discourse. In so doing, while considering that humans realize experience, make meaning, interpret the world, establish relationships, and teach/learn through discourse, I held the view that examination of discourse through SFL-case study perspectives and methods in consideration of theoretical educational perspectives open up gateways of knowing about all the above. In this dissertation my interest was to examine student participation and agency in the powerful CS field (Santo et al., 2020) more specifically in CT-CP as a part of CS (Bourke, 2018). In my analysis (its micro perspective), I focused on facilitator/teacher-student interactions with respect to their relative status in their CT-CP educational practices within the Educational CT-CP CoP where they happened. In doing so, I maintained the view that during teaching/learning, the positions of facilitators, teachers and students are not immanent labels but positions that can change as activity unfolds (Cabral & Baldino, 2002). In other words, during teaching/ learning, all the interactants involved can, while participating with more and less agency in activity, call each other for action and provide and ask for relevant information. In the macro perspective, to frame understandings of the possible causes that shaped student participation and the potential implications of their performance, I considered the broader social and educational environments wherein the CT-CP teaching/learning practices I studied happened. While acknowledging the impact that issues of race, ethnicity and gender have on CT-CP teaching/learning, in this study I focus power as it relates to expertise and status in CT-CP teaching/learning practices

Consequently, my research approach aligns with critical realism (Fryer, 2020). My stance is that of a researcher whose understandings are informed by pre-existing theories and understandings but not determined by them. I approached this study in a bottom-up fashion. I did not follow a particular theory and contrast my developing understanding with the theory. Rather, I examined in depth the language used by my participants as it happened and what was achieved through it. I studied CT-CP-in-the-making. Importantly, my research design remained flexible throughout the study, and I incorporated new perspectives and constructs and even one of its RQs, i.e., RQ3, as it evolved (Tracy 2019). I kept a stronger focus on participation and the social aspects of learning, but without disregard of cognitive approaches (Sfard, 1998, 2001a; Skemp, 1987; Vygotsky, 1978, 1986), paying special consideration to mathematics-based abstraction (Hershkowitz et al., 2001; Lavie & Sfard, 2019; Skemp, 1987; Sfard, 1991a; Sfard & Lavie, 2005) and CT-CP//CS abstraction processes (e.g., Colburn, & Shute, 2007; Grover & Pea, 2018; Kong; 2019; Kramer, 2007; Wing 2008) and levels of abstraction (LOA) (Cutts et al, 2012;

Perrenet & Kaasenbrood, 2006; Perrenet et al., 2005; Statter & Armoni, 2016; Waite et al., 2018; Wing, 2008)

Addressing Kafai's (2020) call to put in communication a variety of theoretical frames and methods to study educational CT-CP/CS, this dissertation adopts a discursive, interdisciplinary, and social approach to the teaching/learning of CT-CP. Consequently, this study has been informed by various theoretical and methodological perspectives. This study incorporates views and constructs, mostly, from SFL theory and methods (Christie, 2005; Eggins, 1994; Eggins & Slade, 2005; Halliday, 1984, 1993, 2004; Halliday & Matthiessen, 2014; Halliday & Webster, 2004, Herbel Eisenmann & Wagner, 2007; Martin & Rose, 2003; Martin & White, 2005; Morgan, 2006; Morgan & Sfard, 2016; Schleppegrell, 2004, 2012), Communities of Practice (CoP) learning-as-participation theory (Farnsworth, 2016; Lave and Wenger, 1991; Wenger, 1998, 2010; Wenger Trayner & Wenger-Trayner, 2015), sociology of education (Bernstein 2003), equity in mathematics and CS education research (e.g., Goode et al, 2020; Morgan, 2012; Margolis et al. 2017; Moschkovich 2002; Schleppegrell, 2012) sociocultural theory (Mahn, 2015; Vygotsky, 1978, 1986; Wertsch, 2007), Sfard's (2001, 2008) commognitive theory, communication and cognition research (Sfard, 2000; Tomasello, 2003; Tomasello & Rakoczy, 2003), research on abstraction both in mathematics and computing (e.g., Colburn, & Shute, 2007; Dagienė et al. 2017; Hershkowitz et al., 2001; Perrenet & Kaasenbrood 2006; Statter & Armoni, 2020; Wing, 2008) and research on CS, and CP-CT (e.g., Ben-Ari 2012; Brennan & Resnick 2018; Grover & Pea, 2018; Lee et al., 2011; Reeves & Clarke, 1990).

The term "discourse" needs clarification. It is commonly used to refer to formal written or spoken monologues or to connected pieces of speech or writing. Yet in academic fields, "discourse" has various and distinct meanings. Most importantly, discourse has been intimately relatedly to literacy, which rather than being identified with coding and decoding written language, has modernly been identified with participation in a discourse such as the CT-CP/CS discourse (Vogel et al., 2020). Discourse has been understood as an operational tool inextricably embedded with practice and equity (Morgan, 2011). Evans et al. (2006) use "discourse" to refer to the "system of signs that organizes and regulates specific social and institutional practices" (p. 4). Regarding learning, Sfard (2001a) argues that learning mathematics may be "defined as an initiation to mathematical discourse, that is, initiation to a special form of communication known as mathematical" (p. 28). Halliday (1993) uses discourse to refer to 'operational text' be it spoken or written (See Chapter 3 for discussion), meaning cohesive language use that is functional in context. Schleppegrell (2004), a follower of Halliday's extensive work on functional linguistics wrote: "Halliday's functional linguistics offers a way of thinking about the relationship between the linguistic choices of speakers and writers in particular moments of interaction and the social contexts that the language helps realize" (p. x). Drawing on these perspectives and definitions, and on Grover and Pea's (2018) definition of CT (CT is a kind of thinking that makes possible the formulation of a task or problem in a way that a computer can effectively carry out), I define CT-CP discourse as follows: CT-CP discourse is a functional, social, and institutional form of communication that concerns both the participation in meaning-making processes in CT-CP environments and the realization of CT-CP contexts and practices. Thus, CT-CP discourse involves thinking and communicating towards the formulation of a task or problem in a way that a computer can effectively carry out.

Understanding CT-CP discourse from a social participationist and literacy perspective has evident ethical-equity implications both in the micro perspective of the classroom or after- school club and the macro perspectives, considering the power that computers have to transform the world that we live in. A focus on participation and literacy is fundamental since it helps investigate and discuss student access to relevant CT-CP practices and computational literacies in and out of school (Resnick et al., 2009; Kafai & Burke, 2014; Vogel et al., 2020)

Educational CT-CP: A Field Under Development

The field of CT-CP educational research has been experiencing significant growth since 2006 after Wing's (2006) seminal publication on the importance of developing CT for everyone, not just computer scientists (Tang et al., 2020). Ten years ago, the US National Research Council emphasized the importance of CT-CP and included the following among the reasons for promoting CT-CP in K-12 education: "succeeding in a technological society, increasing interest in the information technology professions, maintaining and enhancing U.S. economic competitiveness, supporting inquiry in other disciplines, and enabling personal empowerment" (NRC, 2011, p. 4). However, there is ongoing considerable debate on how to best teach CT-CP core ideas (Pollak & Ebner, 2019). In an increasingly digitized and automatized world, current debates on the relevance of computer CT-CP/CS have resulted in an increasing number of countries involved in the process of introducing CT-CP/CS in their K-12 curricula (Nouri et al. 2020), either integrated in other subjects or independently (Lee et al., 2020; Voogt et al., 2015). The importance of CT-CP is also reflected in its inclusion in the United States Next Generation Science Standards as one of the core science and engineering practices (NGSS Lead States 2013), in curricular recommendations made by US key national scope institutions, associations and initiatives (CSTA Standards Task Force, 2017; K12 Computer Science Framework Steering Committee, 2016; National Research Council, 2010, 2011, 2012), and in research that signal CT/CP as a key skill needed to function in the 21st century world regardless of the profession involved (Lodi & Martini, 2021).

Thus, the field of CT-CP is still under development which is reflected in the fact that there still is a lack of consensus on a definition of CT. Systematic literature reviews reveal a lack of consensus on the definition of CT (e.g., Taslibeyaz et al., 2020; Tang et al., 2020; Voogt et al., 2015). Research constantly reports renovated CT definitions and discussion of its key constructs (Bar & Stephenson, 2011; Brennan & Resnick, 2012; Grover & Pea, 2018; Weintrop et al., 2016). However, the referred research, extensively cited, has not explored CT discursively (as it is taught and learned) with fine-grained linguistic methods of analysis—such as the ones SFL provides—with special consideration to the participation of CLD students. In addition, CT has been reported to be domain dependent (Weintrop et al., 2016), varying from STEM to art to social disciplines. Consequently, further exploration of the domains where CT is enacted seems relevant.

This dissertation examines an authentic CT-CP domain as realized in English and Spanish during the teaching/learning of CT-CP in CP language Python, a language with significant presence in educational and professional fields (Barrozo do Amaral Villares & de Carvalho Moreira, 2017). It targets how student participation and agency in CT-CP discourses can be fostered, paying special focus to how the languages that were used during teaching/ learning, that is, English and Spanish mediated the CT-CP practices under focus. Given that abstraction processes and levels of abstraction (LOA) have been regarded as the 'nuts and bolts' of CT-CP

(Wing, 2008), I focused specifically on how discursive gateways can be opened up for all students to participate actively in these processes while they develop CT-CP. While CT-CP educational research centered on abstraction (e.g., Colburn, & Shute, 2007; Kramer, 2007; Wing, 2008) and LOA (Lee et al., 2011; Perrenet & Kaasenbrood, 2006; Statter & Armoni, 2016; Slatter & Armoni, 2020; Wing, 2008) has yielded relevant understandings concerning the role they play in the development of CT there is room for further understanding, especially on the role that students and more specifically CLD students can play in these processes.

Pedagogically, CT has been framed in four different kinds of experiences: unplugged (Bell, 2018), tinkering, remixing, and making (Kotsopoulos et al., 2017). Unplugged experiences distinguish from the rest in that they center on CT practices without the use of computers. In tinkering, students take devices apart to then modify and reconstruct them. In remixing, students mix computational components or objects appropriated from different objects and for different purposes. Finally, in making, the focus of this dissertation, the students create artifacts. More specifically, this dissertation centers on the creation of computational "logical artifacts" (Hoppe & Werneburg, 2019) for which students are engaged in developing logical prototypes that can be, after modelled with mathematics (Lesh & Doerr, 2003) computed to obtain videos through the CP language Python.

In exploring CT-CP teaching/learning, I specifically targeted agency, abstraction processes and LOA, examining the CT-CP discourse that was used—both in English and Spanish—by the participants of this study, that is, a facilitator of CP, her three middle school bilingual students and their bilingual language arts teacher. I was interested in studying how opportunities for students' agency and participation can be opened in CT-CP education because students learn through interpersonal gateways that open students meaning making processes in context (Halliday, 1993; Schleppegrell, 2017; Vygotsky, 1978, 1986). The problem is that student participation in meaning making processes is problematic in complex and abstract contexts, especially for second language learners (Colombi & Schleppegrell, 2002; Vogel et al., 2020). One important question is how CT-CP education and more broadly, CS is taught and learned with and through language (Vogel et al., 2020). This question has educational and political implications since the languages used in CT-CP/CS practices does not limit to CT-CP language and to the specific formalisms used in CT-CP but extends to the natural language used in the practices, be it English, Spanish, Navajo, or any other language. Thus, the natural language or languages used to facilitate CT-CP practices and is a key factor in the student participation in CT-CP/CS discourses, which concerns basic human rights, more specifically, linguistic rights (UNESCO, 2021).

The perspective I maintain in this dissertation regarding language and participation, far from setting the focus on comparisons between the linguistic competencies of monolingual English speakers and bilingual English-Spanish speakers (Grosjean,2012; Moschkovitch, 2010), situates it on the resourceful ways bilingual students navigate different cultural environments (Grosjean, 2013; Vogel, 2020; Zentella, 1997). More specifically to CT-CP education, I acknowledge both the potential and the struggles faced by English learners (and their families and communities), and by underrepresented students in general, to advance their education in the United States (and in so many countries), amid social injustice and powerful discourses such as CT-CP/CS (Santo et al., 2020).

Bernstein (2000) identified three interconnected student rights: enhancement, inclusion, and participation. Enhancement refers to experiencing intellectual and personal ‘new possibilities’, to ‘opening possible futures’ (p. xx). Enhancement cannot be understood without inclusion and participation. Inclusion refers to the right to be included, as opposed to absorbed into new practices, personally, socially, intellectually, and culturally. Finally, students have “the right to participate in procedures whereby order is constructed, maintained and changed” (Bernstein, 2000, xx). In line with the others that maintain ethical and equity lenses both in educational research and practices, the overarching concern of this dissertation is to contribute to efforts towards shifting CS education, and CT-CP in particular, “for some” to CS education “for all” (Goode et al., 2020). My hope is that my findings and methods shed light on how opportunities for all can be provided in the field of CS, more specifically in CT-CP, given the importance that these fields have worldwide in education and in general in human activity.

Anticipating Research Outcomes

This dissertation provides some relevant teaching strategies that can bridge the inherent complexity of CT-CP (NRC, 2010; 2011; Brocconi et al., 2016: Kong, 2019: Lyon & Magana, 2020). The CT-CP approach adopted by the participant facilitator enabled student CT-CP learning and the communication of the essence of CT, that is abstraction (Wing, 2008) and algorithm thinking. I contend that the student participants and rest of participants communicated CT itself and that CT can be learned without CT-CP conceptual explanations. In addition, I show in this dossertation how SFL perspectives and tools, in combination with case study perspectives and methods, constitute a powerful resource to investigate authentic CT-CP teaching/learning environments, how students are positioned to participate in them and the roles that are acted out, how CT-CP practices unfold and CT-CP teaching/learning actually happens moment-to-moment. In the practices examined, the facilitator engaged her students in a "low floor" or "easy to get started" approach to CT-CP (Papert, 1980; Resnick et al., 2009) that allowed them to participate and express themselves in a new literacy (Kafai & Burke, 2014; Resnick et al., 2009; Vogel et al., 2020) bilingually (Vogel et al., 2020).

Why I Used SFL as a Resource to Examine CT-CP Practices

To investigate this topic, I used use SFL, a powerful resource to examine discourse and what can be achieved through it within a particular environment (Halliday & Matthiessen, 2014: Herbel-Eisenmann et al., 2013; Morgan, 2012; Morgan & Sfard, 2016; Schleppegrell 2004). "Halliday's functional linguistics offers a way of thinking about the relationship between the linguistic choices of speakers and writers in particular moments of interaction and the social contexts that the language helps realize" (Schleppegrell, 2004, p. x) According to SFL, these linguistic choices create (contexts of) situation which account for the 'reality' created by participants as they interact in CT-CP teaching/learning practices and the CT-CP 'reality' that they construe (Morgan & Sfard, 2016). Importantly, the examination of the linguistic choices that were used and situations created by participants need to conduct in consideration of the broader environments wherein they happened (e.g., Alshwaikh & Morgan, 2018). To do so, I used Communities of Practice theory (CoP) (Farnsworth et al., 2016; Lave and Wenger, 1991; Wenger, 1998, 2010; Wenger-Trayner & Wenger-Trayner, 2015) a theory of learning based on apprentice participation which has been applied in multiple disciplinary practices (Chauraya & Brodie, 2018; Lave & Wenger, 1991) which affords relevant theoretical constructs to conceptualize CT-CP research-educational communities such as AOLME.

Importantly, as argued by Morgan (2012), discourse is inextricably embedded in the practices it mediates and is used by teachers, for instance, to position students with respect to disciplinary practices, which gives them more and less access to perform more and less active roles (Herbel-Eisenmann et al., 2013; Morgan, 2012). This, as argued by Morgan, has decisive equity and educational implications.

In addition to the relevant SFL-based studies mentioned above, SFL has been extensively used in the past 15 years to explore a variety of issues such as agency and mathematical definitions (Morgan, 2005); agency in negotiations of meaning during teacher-student interactions (González & DeJarnette, 2012), the roles of authors and readers in mathematics books (Herbel Eisenmann, 2007), semantic shifts in mathematical discourse (Herbel-Eisenmann & Otten 2011), student positioning (Herbel-Eisenmann & Wagner, 2007), classroom written and spoken discourse (Alshwaikh & Morgan, 2018) and the evolution of high-stakes examinations in England through the years (Morgan & Sfard, 2016).

I have used SFL because it affords the means to explore in depth mathematics and science teaching/learning as it happens moment-to-moment (Alshwaikh & Morgan, 2018; Christie, 2005) and the interpersonal facet of mathematics (e.g., Herbel-Eisenmann & Wagner, 2007; Morgan, 2006; Morgan & Sfard, 2016), including key aspects such as student agency (Morgan, 2005) which plays a key role in learning (Lavie & Sfard, 2019; Sfard, 2008; Wenger, 2010). Importantly for this study, it also affords the means to examine in-depth science educational domains (Mizuno, 2018)

To examine student experiential domains, agency, and participation exhaustively with the specific tools that SFL provides is relevant because agency and social participation have been correlated with the development of professional skills in CT-CP directly related fields such as professional software workplaces, (Vähäsantanen & Eteläpelto, 2018). Additionally, communication skills and collaborative work, which are also explored in this study by means of SFL and the support of other theoretical perspectives (e.g., Sfard, 2000, 2008), are among the most demanded skills in engineering higher education (Inti et al., 2018) and in international markets (Sarfraz et al., 2018).

SFL perspectives and methods have been instrumental in the analysis of discourse of disciplines such as science and mathematics. A significant body of research has determined that both are highly complex and alienated discourses where agency is typically granted to abstractions, rather than to humans (e.g., Halliday, 2004; Morgan & Sfard, 2016). Examining agency and participation is also important because learning has been conceptualized as agentive explorations in established practices or routines of disciplines such as mathematics (Lavie et al., 2018; Lavie & Sfard, 2019) and engineering (Martin & Betser, 2020), and conceptualized as shifts in student patterns of participation (e.g., Lave and Wenger, 1991; Wenger, 1998; 2010). As will be discussed, some established practices in CT-CP environments include algorithm creation and thinking, creating computational artifacts, problem decomposition, modularization, testing and debugging, collaboration and creativity (Grover & Pea, 2018).

Taking the above collectively, I opted to resort to SFL theoretical perspectives and methods in this dissertation which, for the reasons already explained (and will be further discussed in Chapter 3), I combined with case study affordances. Even though the CT-CP/CS education literature consulted reflects some presence of studies with a focus on discourse (e.g., Arastoopour et al., 2019; Grover & Pea; 2013; Wu et al., 2019;) and specifically in bilingual environments (e.g., Vogel et al., 2019; Vogel et al. 2020), this dissertation constitutes, to the best of my knowledge, the first extensive analysis of the discourse used, thus the first SFL analysis, by a facilitator in interaction with her students across all the lessons of a CT-CP/ CS course at any educational level. It is the first analysis of discourse in CT- CP/CS that explores CT-CP teaching/ learning as it happens moment to moment at the subtleties of discourse trough methods such as the ones that SFL affords (e.g., Alshwaikh & Morgan, 2018; Morgan & Sfard, 2016).

Relevance of CT-CP Out-of-School Education and AOLME

Currently, CT-CP is in the process of integration into K-12 education around the world (Bocconi et al., 2016; Nouri et al., 2020), especially in STEM education (Lee et al., 2020) However, integration has a long road to be transited ahead. While K-12 schools around the world gain ground in providing CT-CP educational practices to novice learners, popular organizations such as “codeacademy.com” and “code.org”, online platforms (OERS) (Moon et al., 2020) and out-of-school experiences (e.g., Afterschool Alliance 2021, Lee et al., 2011; Merino-Armero, González-Calero 2021) constitute good examples of extracurricular opportunities to develop this key 21st century skill. Digital learning environments and after- school programs have become significantly popular. CT education to young students is also provided at universities and other private entities (Hsu et al., 2019; Lee et al., 2011; National Research Council, 2011; Weintrop et al., 2021). Research on CT (CS-CP) teaching and learning experiences after school have provided invaluable insights on both teacher and student successes and challenges (Buss et al., 2017; Werner et al., 2012; Yadav et al., 2018).

Internationally, The Organization for Economic Co-operation and Development (OECD) (2021) has recognized the great relevance of non-formal education, stressing that, while voluntary, it typically features more flexible organization and learning goals. Internationally, based on reports prepared by more than 20 OECD countries, including countries like the U.S., Spain, Mexico, South Korea, and the U.K, there is increasing recognition of the competencies that people acquire in non-formal and informal environments Patrick (2010). Such gained competencies are viewed as stepping stones to further formal education and qualifications with value in the labor market. Importantly, Patrick emphasizes that the competencies gained in non-formal and informal education can facilitate structural adjustment through a process that will eventually achieve complete social recognition and delivery of fully equivalent qualifications to those obtained through formal learning.

In the United States, out-of-school programs with an educational orientation began at the beginning of the 20th Century to meet the needs of families during the rise in women's participation in the labor market (Mahoney et al., 2009). Federal, state and privately funded after-school programs have had a significant impact since then on the immersion of diverse students in STEM learning practices, increasing their participation in STEM fields and careers. Importantly, after-school STEM experiences allow for experimentation and failure and promote crucial relationships with mentors and peers granting CLD students with access to settings, expertise, and instruments often not otherwise available to them (Afterschool Alliance, 2021). Importantly, after-school programs offer safe environments and flexibility for incorporating to STEM learning emotional, ascetic, and social elements often not provided by schools (Bevan & Michalchik, 2013). Affordances of out-of-school programs such as the ones highlighted by Afterschool Alliance (2021) and Bevan & Michalchik (2013) are apparent in AOLME.

The AOLME team is an interdisciplinary engineering research-mathematics educational research team which has united the efforts of three principal investigators/university professors, an administrator, a few dozens of engineering and education graduate and undergraduate research assistants (including 32 facilitators), four schoolteachers and 135 middle school students (67 Boys and 44 girls). AOLME's work and success depends on the good and regular communication or *mutual engagement* (Wenger, 1998) of its members. AOLME's varied research-educational agenda or *joint enterprise* (Wenger, 1998) includes the development of a mathematics and image-video processing-based CT-CP curriculum that was implemented after school multiple times in the period 2013-2019 in two US Southwest middle schools (one urban, one rural). Fundamental AOLME's goals or joint enterprises are teaching and learning CT-CP and the creation of videos in CP language Python. In fulfilling such goals, AOLME facilitators and students such as the participants of this study, engage in CT-CP discourse and practices, which help constitute AOLME's shared repertoire of practices (Wenger, 1998) such as algorithm creation and thinking, creating computational artifacts, problem decomposition, modularization, testing and debugging, collaboration and creativity (Grover & Pea, 2018)

AOLME's CT-CP curricular implementations or introductory CT-CP courses, were divided in two levels of proficiency, i.e., Level I and Level II. This study centers on the implementation of Level 1 of the curriculum in the rural school in the Spring of 2017 and comprised 12 2-hour-long sessions. The curriculum of Level 1 (see Appendix B) was under constant development and revision until it was completed (it can be found at https://aolme.unm.edu/WebsiteModel/template/index.html)

Adapting the curriculum to teaching/learning needs and goals was a key AOLME concern. After gathering feedback from AOLME students, facilitators, schoolteachers, and research assistants, AOLME would dynamically and constantly revise it.

Researcher Positionality: My Journey towards this Dissertation and CT-CP

I have been an English learner since I was fourteen. As a researcher, I see myself as a combination of the experience I've gained over the years as a teacher, as a student at all levels, as a person who has enjoyed numerous rewarding experiences with all kinds of people in four continents and a research assistant in the period 2015-2019 in AOLME. I have a background in linguistics, mathematics, biology, and physics and in bilingual education which I gained in my undergraduate and graduate studies in the University Complutense de Madrid, the Universidad Autónoma de Madrid, and more recently at the University of New Mexico. I have taught mathematics, Spanish, and English to young students in Spain and Nicaragua for non-profit-ethics-oriented organizations and I have also taught at University of New Mexico how first and second languages are learned to undergraduate and graduate students. In addition, in New Mexico, I have been an elementary education bilingual teacher for two years, and a mathematics interventionist.

More specifically relevant to this dissertation, my roles as a research assistant in AOLME included my participation in both its research and educational agenda. I was an "observer as participant"(Merriam & Tisdell, 2015) in three implementations of the AOLME Level 1, including the one I focus on in this dissertation. My work included facilitating materials to facilitators and students during their CT-CP practices, addressing their questions when required, conducting interviews, and helping with multiple logistic and research-related issues. I worked hand in hand with many AOLME members onsite and establish good rapport with a significant number of facilitators and students, including the participant facilitator I selected for this study and her students, especially with one of them: Herminio (all participants and sites used hereafter are pseudonyms). Additionally, I h participated actively, under the supervision of AOLME principal investigators, with my colleague Gabino Noriega, in the design, implementation and follow-up of the ongoing professional development that was provided to AOLME facilitators in the period 2017-2018. This professional development centered, mostly on 'talk moves', discursive moves that promote student discussion (Chapin et al., 2009) and on strategies that foster equal opportunities of participation and collaboration (Cohen & Lottan, 2014).

Out of the schools my work centered mostly on video analysis and cataloging, focusing on advancing understanding on a variety of issues concerning the teaching/learning of mathematics-based CT-CP for the development of digital images and videos. I also (most often in close collaboration with my graduate research assistant colleague Gulnara Kussainova) explored and discussed different AOLME lines of research such as the exploration of teaching strategies that may promote student collaboration. My and our analysis of videos focused on a variety of facilitators, including the participant facilitator of this dissertation. This experience fostered my reflections upon different CT-CP teaching styles, different ways to promote or not student participation and collaboration in the same CT-CP environment and practices that I focus on in this study. This work, combined with my experience as "observer as participant" and the insights obtained from the AOLME principal investigators and rest of members in weekly meetings and informal conversations in the period 2015-2019, equipped me with contrasting views and a remarkable preparation to conduct this investigation.

The above-discussed experience positioned me to better understand many opportunities enjoyed and challenges faced in and out of the schools. My seven years' experience in the United States as an immigrant from Spain, second language learner, teacher, and research assistant in AOLME has conceded me the opportunity to further my understanding on the successes, innovations and struggles that have written the history of bilingual education in the US, more specifically in its Southwest (cf. Blum-Martínez & López, 2020). The efforts to further high-quality bilingual education for all continue and strongly need continuation, especially for CLD students as exemplified by AOLME. This dissertation constitutes an example of such efforts, in a field of paramount importance in the second decade of the 21st Century, which deserves to be uncovered.

Delimitations of this Dissertation

This dissertation is limited to the study of spoken discourse that was used by the participants of this study as they interacted during curricular CT-CP practices and the written discourse that was used by them, interviews, and questionnaires, leaving out of focus gestures and other semiotic means they used to communicate. However, as I will explain in the methodology section, I did rely on students' artifacts such as the prototypes that they designed and the Python code and digital images and video that they produced. But I only did it to decipher their oral interactions and to better understand what was achieved through oral discourse, not for the sake of semiotics.

My Journey Within this Dissertation and its Research Questions

I approached this dissertation with a wide-open mind. The preliminary understanding I had gained about CP throughout the experience explained above as an AOLME research assistant was a good start, but just that. In this dissertation, and generally in the world of programming computers, several disciplines intersect, boundaries are not clear and concepts transfer all the way from a well-established academic discipline such as computer science (cf., Ben-Ari, 2012; Bourke, 2018) to the popular term "coding" which can be found in all kinds of informal conversations (Lodi & Martini, 2021). Interestingly, the field notes that I had taken in AOLME meetings reflect conversations centered on defining the subject matter of its equity-oriented research-educational project. My notes taken in 2017 on the AOLME regular meetings reflect that the definition of the subject matter of the CT-CP practices that I focus on in this study was in evolution. My understanding before starting this study hinged on the idea that I was about to examine in depth equity-oriented, mathematics-based engineering and CP practices aimed at the development of images and video in Python. My notes also reflect that CT was necessarily involved since computer programming requires CT even if programmers are sometimes not aware that they think computationally when they program computers. In other words, when I started the work for this dissertation, its subject matter was not one hundred percent clear to me.

For this and other reasons related to my beliefs as a researcher, the design of this qualitative research remained "alive" and flexible from beginning to end (Maxwell, 2013; Tracy, 2019); its pieces were "loose". Its theoretical framework, methodology and research questions were open to reconsideration until my work was finished. My review of related literature remained open as well. During the exploratory phase of the analysis of this dissertation, which I detail in the methodology chapter (Chapter 4), I considered a variety of 'stories' that could be explored, justified, and explained in depth by means of the perspectives and methods chosen

(Tracy, 2019). The preliminary understandings I had gained through my work as an AOLME research assistant helped me consider contrasting lines of research. For instance, I had explored with my colleague, Gulnara Kussainova how the participant students of this study collaborated in their group in their CP practices and how the facilitator managed the learning environment created. We were able to observe, discuss and reflect upon how the facilitator was able to maintain control over the CP learning environment and how she distributed labor to make sure that the programming in Python was completed. We were also able to observe how the facilitator seemed to favor at times the student that was more responsive to her directions. Apparently, the facilitator favored one student at first, but then, as the sessions progressed, a different student who seemed more concentrated and skillful appeared to receive more attention. How the students collaborated to program in Python and how the facilitator managed collaborative programming constituted sound lines of research. However, as I examined the data of this dissertation on my own, another 'story' emerged. In other words, as my work in this dissertation progressed, I was able to focus my analysis by means of the theoretical perspectives and SFL-case study methods I was using: My findings emerged. I explain them thoroughly in Chapter 5. As findings emerged, I revised related literature to help me understand them in greater depth. I also adapted the methodology (Chapter 4) and theoretical framework (Chapter 3) accordingly. To gain understanding of the CT-CP practices at hand, I extended the affordances of SFL by combining its theoretical perspectives and methodological tools with case study methods and other frames and constructs that help understand communication, thinking and abstraction processes involved in CT-CP (e.g., Grover & Pea, 2018; Kramer 2007; Lavie et al., 2018; Lee, 2011; Sfard, 2000, 2008; Wing, 2006, 2008).

While eventually it became apparent that the participant facilitator of this dissertation engaged her middle school CLD students in CT-CP practices, it took careful analysis of my data to uncover it. CT had not been a construct that I had explored in depth in my previous work in AOLME. To be clear, I could not find the term CT in the video recordings of the participant teaching/learning practices, interviews, questionnaires, or field notes I used as data for this dissertation. However, CT, though hidden, was fundamental in the CP activity of the participants of this study as I demonstrate in the findings section (Chapter 5). In Chapter 6, I then discuss the significance of the findings for the teaching/learning of CT-CP and for CT-CP teaching/learning research and the limitations of this study. Finally, in Chapter 6, I conclude by elaborating on the significance of this dissertation and discussing some final thoughts.

Based on my research interests and flexible research design that I adopted in this dissertation, I have come to the following Research Questions (RQs):

(1) What is the nature of the computational thinking-computer programming (CT-CP) domain that a CT-CP facilitator of a CT-CP community of practice (CoP) offered her CLD students by means of oral discourse?

(2) How were these students positioned in terms of their participation and role in CT-CP practices?

(3) Can the CT-CP pedagogy adopted by the facilitator to teach CT-CP facilitate student learning of CT-CP?

Summary

In this chapter, I discussed how I became interested in the development of CT-CP, especially as it relates to CLD students. I discussed the current CT-CP educational research field and the relevance of exploring abstraction processes and the levels of abstraction involved in its development. Also, I explained the relevance of adopting discursive SFL-based perspectives and tools to study domains of human experience such as CT-CP and key constructs such as agency and participation, especially in what respects CLD students. Next in Chapter 2, I present the review of the literature that has informed this dissertation.

Chapter 2

Review of Related Literature

Overview

This chapter explores relevant literature that has informed this dissertation. It gathers research from a variety of related fields and includes discussion on the definition and essential components of CT-CP. It focuses on theoretical and methodological perspectives as well as on empirical findings that situate this study in the current CT-CP educational research landscape. As my analysis progressed, I identified a CT-CP teaching/learning progression and different ways of knowing taking place as well as aspects such as modeling with mathematics, abstraction, levels of abstraction (LOA) and communication playing a fundamental role in the CT-CP practices that my participants were engaged in. For this reason, I focused the review of literature on these dimensions.

Interdisciplinarity: Putting Different Frames in Communication

Extensive research advocates for a breath of vision in what respects putting in communication different frames to study teaching/learning in mathematics and CT-CP/CS fields. Some examples are Kafai et al., (2020), Lavie & Sfard (2019), Martin & Betser (2020), Morgan (2014a; 2014b), National Research Council (2011), Sfard (2001a) and Sfard (2008).

In mathematics education research, Sfard (2001a) argues for the consideration of both acquisition and participationist perspectives on learning as complementary rather than exclusive. The former sets the focus on individual capabilities, and the latter on more social aspects and with a clearer focus on social interaction. Sfard (2001a) argues that the individual/social dichotomy rests in the mechanisms of learning, rather than on the definition of learning, setting the foundations for a theoretical framework to examine the teaching and learning of mathematics based on the social communication of thinking—c–mmognition—which was further developed in Sfard (2008). Sfard reconciles individual and social perspectives on cognition by claiming that individual thinking is considered a particular type of thinking where the individual communicates with herself. Morgan (2014a; 2014b), from a social semiotics tradition (strongly based in Halliday's functional grammar), argues for the incorporation of perspectives and constructs of social theories to address equity concerns in mathematics education research (e.g., Bernstein, 2000). Morgan stresses the fundamental SFL tenet that understanding any spoken or written text is only possible through using knowledge of the immediate context of the practice and of the broader culture wherein these practices take place (e.g., Halliday & Matthiessen, 2014). On its part and in similar lines, CoP theory (Farnsworth, 2016; Lave and Wenger, 1991; Wenger, 1998, 2010; Wenger Trayner & Wenger-Trayner, 2015) regards "learning as situated in human activities, inseparable from the environments in which knowledge is used, and occurring through social processes that involve the negotiation of meanings with others" (Chauraya & Brodie, 2018; p. 2).

In CT-CP/CS research, Kafai et al.(2020) asserts that cognitive, situated, and critical framings are the most prevalently found perspectives within the landscape of CS education and advocates for a "theory dialogue" between these perspectives. Kafai et al.(2019) assert that cognitive framings that emphasize situated participation and sharing digital artifacts, and critical framings are the most prevalent perspectives within the landscape of CT-CP/CS education. Each of these framings draws from different learning perspectives. Kafai et al., (2020) argues for a dialogue between these framings and perspectives in an educational landscape where it has been argued that, on the one hand, CT-CP/CS is a powerful discourse to which there is restricted access, especially to CLD students (Goode et al., 2020; Santo et al., 2020) and, on the other, CT-CP/CS are challenging to teach and learn (Grover & Pea, 2013; NRC, 2010; Peng et al., 2019; Reppening, 2016).

The CoP theory notions discussed above (e.g., joint enterprise, mutual engagement, and shared repertoires) have been widely used to study participation and teaching/learning in varied environments such as mathematics in educational settings (eg. Moschkovich, 2007), business-oriented organizations and professional associations (Chauraya & Brodie, 2018; Wenger Trayner & Wenger-Trainer, 2015) and in informal environments in a variety of spheres of human knowledge and places such as craft apprenticeships in Liberia and meat cutting in the U.S. Navy.

Research in educational engineering shows that educational research that has used CoP theory is well complemented with frameworks such as the ones provided by Sfard's (2008) commognitive framework which provide constructs and mechanisms to help understand in-depth expert-novice interactions (Martin & Betser, 2020). Chauraya and Brodie (2018) focused on how a professional learning CoP yielded insightful teacher views on mathematics classroom practices in relevant issues such as students' reasoning in making errors while operating with ratio algorithms. Chauraya and Brodie's use of CoP theory constructs — joint enterprise, mutual engagement, and shared repertoires — helped them identify and analyze a number of key episodes of teacher conversations concerning their professional development and students' needs. Their examinations and findings helped them better understand teaching/learning practice. However, Chauraya and Brodie's analysis might have benefited from specialized discursive tools for analysis such as the ones used in Lavie and Sfard (2019) to study discourse and/or the fine grained SFL tools to study the subtlest aspects of teaching/learning interactions as they unfold moment-by-moment. More specifically, a deeper study of mutual engagement with specialized linguistic perspectives and tools for the analysis of teaching/learning interactions might have broadened their analysis scope and findings. In sum, studies like Chauraya and Brodie (2018), while insightful, lacked the affordances of complementary discursive analytic tools such as the ones discussed above (Lavie & Sfard's, (2019)), or the ones provided by SFL which might have helped to address their research concerns more subtly. *Therefore, CoP theory lacks the necessary constructs to look at interactions and participation* as they unfold over time and *SFL requires a social theory which provides understanding of the context where interactions happen.*

Therefore, relevant research points at the soundness of putting in communication several frames to help further understand the nature of CT-CP teaching/learning practices, how opportunities to learn can be opened and how the complexity of CT-CP can be bridged.

CT-CP: Definition and Essential Components

Systematic literature reviews on CT reveal a lack of consensus on a definition on CT (e.g., Tang et al. 2020; Taslibeyaz et al., 2020; Voogt et al., 2015). Haseski et al., (2018) found 59 definitions of CT. Research constantly reports on CT-CP main practices and concepts (e.g., Bar & Stephenson, 2011; Brennan & Resnick, 2012; Grover & Pea, 2018; Weintrop et al., 2016). In addition, CT-CP has been reported to be domain-dependent, varying from STEM (science, technology, engineering, and mathematics) to art to social disciplines (Weintrop et al., 2016). This dissertation reports in Chapter 5 on the characteristics of the specific CT-CP domain realized discursively by the participant facilitator. Let me recall the definition of CT-CP I use in this dissertation, which draws on Grover and Pea's (2018): CT describes the mental activity involved in formulating a task in a way that a computer can effectively carry out.

Fundamental CT-CP Concepts, Skills and Practices

Fundamental CT-CP Concepts

The main CT concepts are loops, conditionals, sequences, parallelism, data structures such as variables and lists, mathematics operators, functions and Boolean operators, event handling, procedures, and initialization (Kong, 2019).

Fundamental CT-CP Skills

Six fundamental CT-CP skills can be identified in the literature. Dagienė, et al. (2017) identified five and provided the explanations about them that can be observed in Table I. An extra skill identified by Weintrop et al. (2016) can also be observed.

Table 1

CT Skills and Identification

CT skill	How to spot use of that skill
Abstraction	Removing unnecessary details; Spotting key elements in problem; Choosing a representation of a system
Algorithmic thinking (to automate solutions)	Thinking in terms of sequences and rules; Executing an algorithm. Creating an algorithm
Decomposition	Breaking down tasks; Thinking about problems in terms of component parts; Making decisions about dividing into sub-tasks with integration in mind (e.g., deduction)

Evaluation	Finding best solution; Making decisions about good use of resources; Fitness for purpose.
Generalization	Identifying patterns as well as similarities and connections; Solving new problems based on already-solved problems; Utilizing the general solution (e.g., induction)
Data Management	Collecting data, creating data, manipulating data, analyzing data and visualizing data (Weintrop et al., 2016)

Adapted from Dagienė et al,. 2017, p. 37

Dagienė et al., assert that in solving a task, more than one skill could be involved. They recommend focusing on a maximum of three skills per task.

Fundamental CT-CP Practices

Table 2 gathers fundamental CT-CP practices and the number of studies where they have been identified.

Table 2

CT-CP Practices

Component	Study	Frequency
1. Abstraction/abstracting, modeling/abstracting, and modularising	Brennan & Resnick (2012), Denner et al. (2014), Gouws et al. (2013), Grover et al. (2015), Grover & Pea (2018), Mueller et al. (2017), Rodriguez et al. (2017), Seiter & Foreman (2013), Sherman & Martin (2015), Werner et al. (2012), Zhong et al. (2016)	11
2. Algorithmic thinking	Denner et al. (2014),	8

	Duncan & Bell (2015), Gouws et al. (2013), Grover & Pea (2018),Mueller et al. (2017), Rodriguez et al. (2017), Seiter & Foreman (2013), Werneret al. (2012)	
3. Testing and debugging	Brennan & Resnick (2012), Burke (2012), Fessakis et al. (2013), Grover et al. (2015), Grover & Pea (2018),Mueller et al. (2017), Román-González et al. (2017), Zhong et al. (2016)	8
4. Being incremental and iterative	Brennan & Resnick (2012), Grover et al. (2015), Grover & Pea (2018); Mueller et al. (2017), Zhong et al. (2016)	5
5. Problem decomposition	Grover et al. (2014; 2015), Grover & Pea (2018), Mueller et al. (2017), Seiter & Foreman (2013)	5
6. Planning and designing	Burke (2012), Zhong et al. (2016)	2
7. Reusing and remixing	Brennan & Resnick (2012), Mueller et al. (2017)	2
8. Data related activity	Weintrop et al. (2016)	1
9. Collaboration and creativity	Grover & Pea (2018)	1
10. Creating computational artifacts	Grover & Pea (2018)	1

Adapted from Kong (2019, p. 21)

Kong suggests that *abstraction/abstracting* should be merged with *modularization* to handle the inherent complexity of CT-CP and "build something large by putting together collections of smaller parts" (p. 129). Given the relevance of abstraction and modeling, I will return to them below. I have added to Table 2 data related activity, collaboration, creativity and creating computational artifacts, given their presence in the literature (Grover & Pea, 2018) and relevance in this dissertation. *Algorithm thinking* is a key practice (Wing, 2008) which describes the ability to think and communicate an *unambiguous* sequence of steps for processing information in a set of unambiguous instructions aimed at accomplishing a task or solving a problem (NRC, 2010). An algorithm expressed in a particular CP language constitutes a program that the computer can process and execute (Bourke, 2018). *Testing and debugging* are an integral part of CT-CP practices since, as programs are developed, they need to be evaluated in search of flaws and better results/solutions (Grover & Pea, 2018). In programming processes, *being incremental and iterative* is fundamental since a program is developed little by little until it is complete (Brennan & Resnick, 2012). To complete programs, planning and designing are indispensable practices too. Typically, tasks and problems are decomposed in smaller parts. This CT practice is referred to as *problem decomposition.* According to Kong (2019), problem decomposition is a high-level abstraction process which makes problems—and tasks—more tractable and manageable (Grover & Pea, 2018). Problem and task decomposition describes the CT practice of breaking down problems — and tasks — in parts. In this sense, it relates modularization, the practice of organizing problems, tasks, and programs in modules. Problem decomposition is intimately related to the *solution* of problems since "decomposition involves finding structure in problems and determining how the various components will fit together in the final solution" (Csizmadia et al., 2015, p. 5) According to Csizmadia et al., a good problem decomposition makes it easier to: (1) Modify solutions by changing individual components and

(2) Reuse components if and where needed. In STEM environments, CT has been associated with *data practices* (Weintrop, et al., 2016). Weintrop et al., identified five data related practices: Collecting data, creating data, manipulating data, analyzing data, and visualizing data. Finally, it is relevant to mention that *collaboration and creativity* have also been identified and evaluated as CT practices (Grover & Pea, 2018). Programming has been associated with collaboration which has been identified even as a precursor of CT itself (Chowdhury et al., 2018). Collaboration is reflected in the ideas of *sharing, remixing, and reusing* programs or parts of programs developed by others which have also been acknowledged as CT practices (Brennan & Resnick, 2012).

Abstraction: The Key CT-CP Practice.

Abstraction is highlighted as the most fundamental CT-CP practice in Wing's (2006) seminal article on CT-CP. Its importance is reflected on the fact that abstraction is the CT-CP practice that is most often discussed in CT-CP literature (Kong, 2019). Kong stresses that abstraction is often associated with modeling, the most important technique in engineering (Krammer, 2007). The use of computational models has extended across disciplines and learning their affordances and limitations is regarded with similar importance as developing CT-CP itself (Arastoopour, 2019).

Key Differences between Mathematical and CT-CP/CS Abstractions.

My exploration of CT-CP literature yielded salient discussion on abstraction and overlap between CT, CP, CS, and mathematics, which is not surprising given the multiple extant connections between these fields (Baldwin et al. 2013; Shute, 2017). The nature of mathematical abstractions and CT-CP/CS abstractions is quite different (Colburn & Shute, 2007). According to Colburn and Shute, mathematical and CT-CP/CS abstractions differ in two aspects. First, while mathematical abstractions are largely stable, CT-CP/CS abstractions are constantly changing. The "language" of mathematics is, for the most part, formed by formal abstractions such as numbers which do not change over time and space. In contrast, CT-CP/CS abstractions such as algorithms, programs, software, and computers constantly vary. One reason is the demands of users, another, the changes in the technological foundations of CS. Second, mathematics and CT-CP/CS abstractions differ in their objectives. While in mathematics, abstractions regard neglection of information, the objective of CT-CP/CS abstractions is not to neglect information, but to hide it. When students are engaged in a mathematics lesson about the Pythagorean theorem, information of the colors of triangles is inconsequential and can be neglected. By contrast, in computing, information is handled in different LOA, programmers and software designers may want to hide technical specificities to potential customers while providing precise details to constructors. Regarding CT-CP education and abstraction, the US National Research Council (NRC, 2010) highlighted two fundamental issues: (1) The representation of abstractions; (2) How should abstraction be structured to allow access to common operations. The first issue concerns mathematical representations and algorithms, that is "the notion of a precisely formulated unambiguous procedure that is repetitively applied" (NRC, 2010, p. 9) in a particular computer programming language and program that the computer can process and execute (Bourke, 2018). The abstractions that constitute algorithms need to be represented in forms that are stable and unambiguous so that they can be used in the 'truth statements' that computers require to process them (Ben-Ari, 2012). Mathematics abstract representations can provide these "true" formalisms to represent systems faithfully and fully so that computers can process them (Reeves & Clarke, 1990). The second issue concerns the use of LOA to hide unnecessary detail at the different levels where information is organized, which is key both in CS and software engineering (Alexandron et al., 2014). Information can be organized based on different observable features at different LOA (Floridi, 2008). Following Floridi's example, the information needed to describe the use of a traffic light in a city like Rome is not the same as the information needed for the purpose of constructing traffic lights. In the first case a simple mention of observable color type (red, amber, green) would suffice. However, for the purpose of construction, precise specificities of the traffic light colors such as their wavelength expressed in numerical values is required. Considering color information at both the level of use and the level of construction requires thinking at two LOA.

Therefore, both mathematical and CT-CP/CS abstractions are essential in CT-CP practices. The education of CT-CP requires consideration of both the mathematical abstract representations required to code information and the organization of information at different LOA.

Next, I center on relevant educational perspectives that have informed this dissertation. I examine both cognitive perspectives to learning and approaches that have focused student participation instead. I start what follows by framing discussion historically and ethically. Then I discuss relevant aspects of communication, some basics of image programming in Python and the methodologies that have been used to study CT-CP

A Historical Approach: Participation in Powerful, Complex and Objectified Discourses

The so-called human and natural scientists realized history together until the second half of the nineteenth century (Ordóñez et al., 2007). According to Ordóñez et. al, it was not until then that they took separate ways. The natural sciences specialized and fragmentized very significantly as can be observed today in the multitude of existing disciplines and subdisciplines. Importantly, all technology related fields become extremely pragmatic. The sciences lost the kind of global perspective they had in the times of da Vinci. Truth becomes equal to effectiveness: if something works, it is valid; otherwise, it must be discarded. The ends started to justify the means. Philosophy and ethics gave way; less and less room was left to human agency.

Halliday (2004) explains in linguistic terms the specialization processes that the sciences have experienced over the course of history which resulted in highly alienated discourses where agency is granted to abstractions, rather than to humans. Science and mathematics discourses (Scheleppegrell, 2004; 2007; Sfard, 2008) turned increasingly specialized and complex, presenting great difficulties for the participation of students, specifically of CLD students (Fang et al., 2006; Schleppegrell, 2007). Sfard (1991a) asserts that it took several centuries for mathematicians to arrive at structural versions of fundamental concepts such as number or function. According to Sfard, in mathematics, simple concepts lay the foundation of higher order concepts which are obtained through interplay between operational and structural processes. For instance, a natural number is a property of a set (structural) which can be obtained by counting (operational). Thus, mathematics as a discipline—and mathematical learning—develops in an interplay of structural and operational processes which enabled, for instance, the formation of the concept function from the concept number. In this way mathematics itself grew in complexity through history and so does student learning which can be conceptualized as a progression towards a more sophisticated mathematical discourse (Sfard, 2008). Both processes are necessary because in this progression "we must be able to deal with products of some processes without bothering about the processes themselves" (p. 10). And at the same time, we must be able to, complementarily, "see" counterintuitive entities (products) such as negative numbers. This process/product duality is inherent to mathematical activity and generally to the sciences which typically reify in 'self-contained' static constructs devoid of all human agency.

Objectified Discourses

The grammar of the language in STEM disciplines has maximized its resources through processes of condensation that while displacing the agency of humans have transformed processes into entities (Halliday, 2004; Halliday et al. 2014; Lavie et al., 2018; Morgan & Sfard, 2016; Scheleppegrell, 2004, 2007; Sfard, 1991). This idea was illustrated by Schleppegrell (2004) in the following example taken from a science lesson: "The three temperatures of acetone that were investigated produced calculated DAB values which increased with increasing temperature" (p. 120). This example shows the challenge that many students face with messages such as this, where no human agency seems to be involved. Abstract objects such as numerical values take the floor, values increase on their own and it is not clarified who participated in the investigation. Students might ask themselves: "What's going on?", "Who is the agent here?", "what is my role here? Lavie et al. (2019) argue that mathematical objects "are reifications of known procedures and can… be regarded as reifications of known procedures" (p. 171). When mathematical procedures of any kind are explained or recalled, concrete objects and agents no longer exist; the discourse has undergone a process of *objectification*. Stories become more and more abstract. Sfard (1991) argues that "advanced mathematical constructs are totally inaccessible to our senses" (p. 3). Objectification processes pervade the sciences, from mathematics through science and CT-CP/CS. Individuals and societies are becoming more and more dependent on highly abstract digital networks and artificial intelligence which are prominent in all fields of human activity, in a pervasive digitization of the world (Dufva & Dufva, 2019). Dufva and Dufva bring the question of participation in digital technology to the fore, asking "who sets the direction of the advance of society?' (p. 25) Their concerns revolve around agency resonating with Santo et al. (2020); they asked who has a "seat at the table" in determining K-12 CS experiences. Dufva and Dufva assert that there is an evident imbalance of interests between dominant players, mostly major corporations, and society at large. Dufva and Dufva argue that we are at the point where it is key to promote human-centric perspectives to approach digitalization and its impacts. These authors advocate for educational approaches that develop creativity-based and active relationships with an increasingly digitized society.

CT-CP: Difficult Discourses to Teach/Learn

Typically associated with computer programming (e.g., Brennan & Resnick, 2012; Grover & Pea, 2015; Nouri et al., 2020) and problem-solving (Barr & Stephenson, 2011; Brennan & Resnick, 2012; CSTA, 2011; Grover & Pea, 2013; Román-González et al., 2017; Shute et al., 2017; Wing, 2006), CT has become an integral part of all key professional fields and a foundational 21st Century educational goal which is mostly taught and learned through CP (Grover & Pea, 2013; Lee et al., 2020; Resnick, 2012; Sengupta et al., 2013; Voogt et al., 2015; Wing, 2006). However, teaching and learning CT-CP is no easy task (Grover & Pea, 2013; NRC, 2010; Peng et al. 2019; Reppening, 2016). The literature is replete with studies that highlight the complexity of the CT field (e.g. NRC, 2010, 2011; Broccoli et al., 2016, Lyon & Magana, 2020). Arastoopour et al. (2019) poses the paradox between the broadening participation in CT practices for all (Wing, 2006) and the fact that "computational thinking is continuously conflated with computer science and programming" (p. 3) which make developing CT-CP very challenging.

To exemplify the complexity of CT, I include below the reflection over CT published by the U.S. National Research Council [NRC] in 2010, which emanated from a workshop that was held the year before. This report gathered insights from recognized experts in computer science, information technologists, education researchers and cognitive scientists. It reads as follows:

> The elements of computational thinking are reasonably well known, given that they include the computational concepts, principles, methods, languages, models, and tools that are often found in the study of computer science. Thus, computational thinking might include reformulation of difficult problems by reduction and transformation; approximate solutions; parallel processing; type checking and model checking as generalizations of dimensional analysis; problem abstraction and decomposition; problem representation; modularization; error prevention, testing, debugging, recovery, and correction; damage containment; simulation; heuristic reasoning; planning, learning, and scheduling in the presence of uncertainty; search strategies; analysis of the computational complexity of algorithms and processes; and balancing computational costs against other design criteria. Concepts from computer science such as algorithm, process, state machine, task specification, formal correctness of solutions, machine learning, recursion, pipelining, and optimization also find broad applicability. (p. 3)

To the untrained schoolteacher, this account of the elements that comprise CT may sound considerably complex. With such a landscape, the ongoing debate on the definition of CT (e.g., So, 2020) may not seem striking, nor the ongoing debate on how to best teach CT core ideas (Pollak & Ebner, 2019). The challenge of teaching CT effectively was already posed in Wing (2008) in the shape of the following questions and dilemma: "What would be an effective ordering of concepts in teaching children as their learning ability progresses over the years? "At what point do we introduce each of the powerful capabilities of a computing machine? This was the dilemma posed by Wing: "We do not want the tool to get in the way of understanding the concepts. We also do not want people just to be able to use the tool but not have learned the concepts" (p. 3721). Seemingly, these questions and dilemmas remain unresolved; the debate continues. CT-CP concepts such as conditionals and variables have been reported to be challenging for novice students (Mouza et al., 2020; Grover et al., 2015; Maloney et al., 2008; Meerbaum-Salant et al., 2013). On the one hand, CT-CP educational research has pointed at the need to balance student engagement in programming and the learning of complex concepts (NRC, 2011). On the other hand, K-12 CT-CP education research reveals teacher misconceptions about what CT-CP entails and lack of knowledge of how to teach it (Sands et al., 2018). It also has been reported that teachers feel left on their own with the task of developing an understanding of the required learning content and how to teach it (Rich et al., 2019). Regarding CLD students, the literature also points at disconnections between formal learning environments and the home as a key hindering factor affecting the performance of English learners (Jacob et al., 2018).

Jacob et al. also claim that English learners have been reported to face additional challenges when CT teaching/learning relies on text-based languages (such as Python). In sum, seemingly, the educational CT-CP field is still under development and more research centered on teaching/learning is needed. In the past two decades several programming tools have been developed to assist CT-CP teaching/learning processes. However, these tools, while useful, also present some educational challenges.

The CT-CP and CS education landscape is replete with educational programming tools (Hooshyar et al., 2016; Weintrop & Wilensky, 2017). Scratch and Alice are among the most widely used (Shutte et al., 2018) Other popular tools include Code.org (Code.org, 2020), Tynker (Neuron Fuel, 2020) and Hopscotch (Hopscotch Technologies, 2020). Most educational programming tools support student independent use of key computing concepts such as variables, conditionals, and iteration in visually attractive ways, lowering the cognitive load and avoiding burdensome syntax errors (Bau et al., 2017). Resnick et al. (2009) describe Scratch as an appealing easy to use programming tool that can be used by everyone regardless of their age, background, or interest. Scratch was intended to nurture creative, systematic thinkers able to use programming to express their ideas. The authors presented Scratch as "a "low floor" (easy to get started) and a "high ceiling" (opportunities to create increasingly complex projects overtime) (p. 60) arguing that using Scratch was especially easy and productive. Scratch was continually revised to make it more meaningful and collaborative so that users could share their projects and build on each other's ideas, images, and programs. Nevertheless, some limitations of visual programming tools such as Scratch and Alice have also been highlighted in the literature (Rose et al., 2020). Rose et al. include financial access, teacher confusion as to which tools to use and bad programming habits among the challenges that visual programming tools present.Seemingly, visual educational tools are much easier to use than text-based professional languages such as Python, but then students face the need to transition from educational tools to the text-based computer programming. However, Fargan and Paine (2017) reported that their own experience in teaching programming in Python with no previous experience in CP was especially positive, stating that Python is a language of simple syntax which is very easy to read and write. Fargan and Paine assert that teaching computer programming "can be frightening if you have little to no experience, but it can be done, and is honestly easier than I first imagined" (p. 106).

Teaching/Learning as a Two-way Process

In this dissertation I use "teaching/learning" (as opposed to "teaching and learning") in view of reciprocity. Usanov (2020) and seminal educational research such as Vygotsky's (1978, 1986) consider teaching/learning as a two-way process, viewing that the interdependence of teaching and learning has an impact on the success of the endeavor. Under this view, both teachers and learners contribute to educational processes with previous experiences and ways of knowing (Wilensky & Papert, 2010). Importantly, "pedagogy is a term that means more than the practice and technique of teaching in the classroom, it also applies to theories that support educational practices for children" (Usanov 2020, p. 183).

Teaching/learning CT-CP: Relevant Cognitive Perspectives

CT-CP education curricula in the US derive, for the most part, from cognitive perspectives to learning such as constructivism (Kafai et al., 2019). According to Kafai et al. (2020), teaching CT-CP from a cognitive perspective focuses on the acquisition of skills and provides students with understanding of key CT-CP concepts and practices. Under this perspective, teaching can be more or less direct and learning is 'constructed' by the student, who draws on prior knowledge to make sense of new information, arranging it in previously built cognitive structures (Margulieux et al., 2019). In addition, conceptual understanding occupies a prominent place in US official educational recommendations in CT-CP/ CS (CSTA, 2016) and also in mathematics education (National Council of Teachers of Mathematics [NCTM], 2000).

In the following section I draw on relevant theoretical and empirical research to discuss a processual perspective on abstractions and concepts. I discuss how concepts originate in abstractions and how they can be viewed as process-product dualities. This perspective provides relevant insights that can help better understand the processes of concept formation. In addition, I discuss constructs such as procedural understanding, conceptual understanding, and instrumental understanding which I consider relevant in the interpretations of CT-CP teaching/learning processes from a cognitive perspective. I start by focusing on adolescence, a key developmental stage in the development of cognitive functions in what it relates to processes of abstraction, concept formation and creativity (Vygotsky, 1998).

Adolescence, Abstraction, Concepts and Creativity

To discuss creativity is relevant, since creativity has been identified as a CT-CP practice (Grover & Pea, 2018) and has been found to promote student development of CT (Israel-Fishelson et al., 2021). This section hinges on two fundamental considerations (1) CT development is a function of students' stage of cognitive development (NRC, 2011); (2) CT is domain and context dependent, varying from the humanities to STEM (Arastoopour et al., 2019)

The concept "concept" is a concept that a normal seven-year-old would not be able to use appropriately. However, seven-year-olds typically use the concept color appropriately. The reason behind these observations lies in the view that concepts such as "concept" have their origin in abstractions while other concepts such as "color" originate in our perceptions (Yen, 2019). As we develop, the way we interpret the world and interact with it varies significantly. As we leave the childhood years, we become progressively able to see beyond our concrete world and needs and gain control of our volition. Two fundamental characteristics of adolescents, as compared to younger students, are their ability to think in concepts and the ability to be creative (Vygotsky, 1998). Creativity is a cognitive and culturally mediated process originated in social practices where we perceive, think, imagine, and ultimately create (Glaveanu et al., 2020).

Vygotsky (1998) argues that, strictly speaking, the child cannot possibly fantasize since it is not until adolescence when they can move from a passive and imitative kind of fantasy to one that can be abstract, authentically creative, and voluntary. According to Vygotsky, adolescents can represent reality in abstract terms, transcending their concrete perceptual experiences. Vygotsky argues that, in adolescence, concept formation leads to a change in the manner adolescents think, in the whole content of thinking, and in the construction of their personality.

Basically, thinking in concepts allows the adolescent to move from a ‘stage of experiencing’ to a ‘stage of cognition’ where their ability to think abstractly allows them to recreate the world around them. For Vygotsky, fantasy is significantly richer and better developed at this stage of development than earlier in childhood on the grounds that the adolescent´s newly gained ability to think in concepts makes possible the "liberation from the concrete situation and the possibility of creatively reprocessing and changing its elements" (Vygotsky, 1998, p. 163).

Creativity allows people to reconstruct and act upon the world in more complex ways (Gruber & Vonèche, 1995) and has been identified as a key CT practice (Grover & Pea, 2018). Creativity is involved in the formulation of solutions in computer programming problem-solving situations (Snalune, 2015) and in the development of CT through storytelling (Kordaki &

Kakavas, 2017) Framing their research in efforts to connect computational literacy to existing curricular practices, Bourke & Kafai (2013) explored the use of Scratch as a means to use storytelling in the teaching of CT-CP. In a workshop where they engaged 19 middle school CLD students, they found that an appealing approach to CT-CP such as storytelling helped students learn the fundamentals of programming. Burque and Kafai argued that in developing their stories with Scratch, students learned concepts such as variables, loops, and conditional statements. The authors found that through storytelling, the students established meaningful connections between programming and the elements of a story. For instance, the students made connections between revising a story and debugging, drafting and design, story protagonists and sprites. In addition, strategies like brainstorming, drafting, feedback and revising the stories facilitated programming learning. Kordaki & Kakavas (2017), conducted a review of literature regarding CT and digital storytelling to develop a framework that highlights the CT abilities that can be learned at the K-12 level at the four stages of the development of a story, i.e setting the stage, design of the story, digital story development and assessment of the digital story. Kordaki & Kakavas found that the students can develop CT skills in the four stages. These skills included data analysis, logical thinking, critical thinking, problem composition and decomposition, data organization, modularization, parallelization, testing and debugging.

Conceptual Understanding in CT-CP/CS

Conceptual understanding is a dominant theme in CT-CP/CS educational research (e.g., Grover & Pea, 2018; Kong, 2019; Wing, 2006, 2008). Even definitions of CT, such as the one provided in Shutte (2017), are based on conceptual foundations: “We define CT as the conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts (p. 26). Not surprisingly, CT assessment in K-12 also focuses on the understanding of concepts such as conditionals, sequences, and loops (Roman-Gonzalez et al., 2017). Wing's (2006) seminal paper suggested that in order to exhibit the CT skills needed to "solve problems," "design systems" and "understand human behavior," one needs to draw on the concepts fundamental to computer science. In 2008, Wing (2008) raised two concerns: (1) how to use computers in ways that they do not get in the way of understanding CT concepts and (2) how to harmonize the use of computers and CT conceptual development. In this section, I draw on education research in different fields, (mostly in mathematics) to discuss concepts, conceptual understanding and how they develop from abstractions, and reifications in dual process-product activities.

Conceptual understanding can be viewed as processes that unfold in 4 stages: factual and procedural knowledge, making connections, knowledge transfer and metacognition (Mills, 2016). Mills's progression, based on a literature review conducted on mathematics, science, psychology, and nursing education, had factual or procedural knowledge as a starting point to the processes of conceptual understanding. According to Mills, pieces of factual information form the basis of conceptual understanding, but they need to be connected in order for deep learning to take place. When factual and procedural information is well connected, as opposed to isolated, knowledge can move back and forth between theory and practice and be transferred to a new topic where the learner can reinforce connections and think in new ways. Mills's arguments resonate with Kilpatrick et al.'s (2001); they argued that students who have developed mathematical conceptual understanding have "an integrated and functional grasp of mathematical ideas" and "know more than isolated facts and methods" (p. 118). Before connections between concepts and transfer happen, understanding is only at the surface level. After, when connections between factual and procedural knowledge are made, metacognition and conceptual understanding happen. Mills draws on the work of Giddens and Brady (2007) on nursing education to explain the view of conceptual learning as a process where students learn to organize information into mental structures, which enhances their conceptual understanding by strengthening their thinking processes. According to Mills, "as metacognition increases, so should conceptual understanding" (p. 551). As will be seen, metacognition has been found to be a condition for concept development (Vygotsky, 1986).

A Historical Perspective on Abstraction and Concept development

In mathematics, there is a long history of processual conceptions of abstraction and concept development share a look on abstractions and concepts as process-product dualities (e.g., Hershkowitz et al., 2001; Skemp, 1987; Sfard, 1991a). In this section I center on the genesis and evolution of concepts regarding learning and cognitive development. For Skemp

> An abstraction is some kind of lasting mental change, the result of abstracting, which enables us to recognize new experiences as having the similarities of an already formed class. Briefly, it is something learnt which enables us to classify; it is the defining property of a class. To distinguish between abstracting as an activity and an abstraction as its end product, we shall hereafter call the latter a concept. (1987, p. 11)

According to Skemp, an abstraction is both a product and a process through which one gets at such a product. Abstraction involves experience where we categorize objects with invariant properties hierarchically. For instance, we categorize stools and chairs into the concept furniture (Skemp's examples). In categorizing objects, Skemp argues, one increasingly bases classifications less on direct perception and more on abstract concepts. For skemp, mathematics includes concepts that are far more abstract than those of everyday life. According to Skemp, mathematics allows us to encapsulate meaning in symbols which enable precise communication without noise, a kind of communication which is increasingly effective based on the common understanding of the concepts that structure it. This kind of communication mediates learning in increasingly abstract directions, and builds mathematical disciplinary knowledge itself, i.e., arithmetics-algebra-calculus.

These processes of abstraction entail, according to Skemp (1987), some kind of *reconfiguration*. Reconfiguration is experienced by children, for instance when they learn that fractional numbers are a new kind of numbers with totally new laws for multiplication as compared to the laws they knew and applied with natural numbers. In the mathematics world of children, a reconfiguration happens when they become able to use and multiply fractions meaningfully for which they need to know that a number can be represented by an infinite number of fractions. Reconfiguration was experienced by Pythagoras when he did not have the mathematical tools to explain his theorem and ideas: "When Pythagoras discovered that the length of the hypotenuse of a right-angled triangle could not always be expressed as a rational number, he swore the members of his school to secrecy about this threat to their existing ways of thinking" (pp. 28-29). According to Skemp, mathematical knowledge is built through teaching/learning abstractions which can only be communicated through examples—not through definitions—which almost invariably are abstractions as well. Only if the latter abstractions are already formed in the mind of the learner can the former become new knowledge; only if the concept of number is formed in the mind of the learner can a learner build the knowledge associated with fractional numbers. It follows that mathematical knowledge is built through mathematical symbols which encapsulate abstract information which is the result of previous abstractions originated in processes that eliminate noise, i.e., unnecessary details to get at essential features.

Abstractions and Concepts Have a History

When students already know and recognize formal mathematical abstractions or constructs, they can apply them to achieve a goal (Hershkowitz et al., 2001). According to Hershkowitz et al. (2001) known mathematical constructs and abstraction-based structures from earlier activity can be used by students for further action. In other words, known mathematical constructs are generators of new abstractions when there is a need for a new structure to achieve a goal or create an artifact. In this manner, students capitalize on previous artifacts in successions of abstractions with a history that researchers can investigate and interview students about. In comparable terms, Sfard (1991) viewed mathematical abstract notions such as numbers both as objects and processes with a history. For Sfard, concept development follows a process which ends in a reification. Sfard tracks the development of the concept of number both from the perspective of the history of mathematics and from the perspective of the learning history of a student. Sfard writes:

> The new entity is soon detached from the process which produced it and begins to draw its meaning from the fact of its being a member of a certain category… Processes can be performed in which the new-born object is an input. New

mathematical objects may now be constructed out of the present one. (p. 22)

Sfard's words refer to how mathematical concepts originated in operational processes such as counting which reify in numbers. Sfard's views, seemingly in consonance with constructivist schema theory (Derry, 1996), viewed the processes of concept formation such as the evolution of concept of the number systems that are used today in modern mathematics in three stages. This progression has a historic evolution across centuries which accounts for the hierarchical progression that took place from the initial sets of concrete objects that were used in counting through to the definition of natural numbers, the positive rational, the positive real, the real numbers to the complex numbers. For Sfard (1991), this evolution of the formation of the concept of number went through three stages: 1) the preconceptual stage; 2) the operational stage; and 3) the structural phase. At the preconceptual stage, at the beginning of number use, mathematicians (and the rest of people) simply manipulated already known numbers, for instance natural numbers in the case of counting. Mathematical manipulations of objects were just processes and nothing else. And there was no need for new concepts, since all the operations were restricted to counting procedures. The operational stage was a long period of use of the numbers that were known, i.e., natural numbers, which saw the emergence of a new kind of number such as the rational numbers. Sfard claims that these new objects or abstract constructs would gain wide use while evoking strong objections and philosophical discussions. Finally, in the structural phase the new abstract concept, e.g., a rational number would be recognized by the mathematical community as a fully-fledged mathematical object. From this moment, different processes performed based on rational numbers would become common until new more advanced kinds of numbers would emerge. Sfard includes in her article an account of the birth of the irrational numbers which emerged in the era of the birth of the Pythagorean theorem. The theorem received serious objections at the time on the basis that no known to date numbers (i.e., natural and rational numbers) would be fit to carry out the mathematical operations that concerned the theorem. Hence, numbers can be conceived operationally at one level and structurally at a higher level. For Sfard, like for Skemp, a history of reifications has constructed mathematics as a structured and hierarchical discipline throughout the centuries. Similarly, students grow mathematical knowledge schema as they learn. "Such hierarchy emerges in a long sequence of reifications, each one of them starting where the former ends, each one of them adding a new layer to the complex system of abstract notions (p. 16). Under these views, knowledge is a construction in which building blocks are abstractions and concepts. Concepts can be conceived of as operational at one level and as structural at another.

Meyer and Land (2003), like the studies that follow in the next section, focused on the relationship between concepts and the new ways of thinking that their acquisition may allow and trigger. Meyer and Land focused on the preconceptual stage, that is, the stage where a concept has not been fully acquired. They used the term threshold concept to define a concept which, while nor fully developed, has the potential to transform our thinking and the way we perceive and experience phenomena in a particular discipline. A threshold concept "represents a transformed way of understanding, or interpreting, or viewing something… such a transformed way may represent how people 'think' in a particular discipline, or how they perceive, apprehend, or experience particular phenomena within that discipline" (p. 1)

Operational and Structural Thinking

Sfard (1991) distinguishes between two complementary ways of thinking: operational thinking and structural thinking. According to Sfard, numbers can be conceived both operationally and structurally. For instance, a natural number can be experienced by a child before they internalize the concept of number in counting operations. A number can be described operationally as “0 or any number obtained from another natural number by adding one ([the result of counting])” and structurally as “Property of a set or the class of all sets of the same cardinality (p. 5). Many mathematical operations are performed on familiar objects before the process reifies in an autonomous entity such as a number. However, for Sfard, other concepts such as the negative numbers are not experienced naturally by children directly and are learned only after multiple operational exercises such as abstractions combined with structural considerations and specific expert instruction.

Procedural and Conceptual Understanding

According to Sfard (1991), to learn a new concept, students need to be involved in both operational approaches and structural approaches. In other words, in developing mathematical knowledge, students need to both operate with concepts and develop an understanding of the concept in relation to other concepts. In Sfard’s words, “at certain stages of knowledge formation (or acquisition) the absence of structural conceptions may hinder further development” (p. 29). To put it in other Sfard’s terms, to acquire the concept of negative numbers, students need to operate with negative numbers while being provided with their corresponding concept as ‘way stations’ in their intellectual journeys. It can be noticed that Sfards’ views resonate with the Wills (2016) account of the progression of conceptual understanding discussed above. Sfard concludes her dissertations about the development of mathematical knowledge and how abstract notions become reifications of mathematical operations by concluding with the following thesis, which she describes as a vicious circle:

> A person must be quite skillful at performing algorithms in order to attain a good idea of the "objects “involved in these algorithms; on the other hand, to gain full technical mastery, one must already have these objects, since without them the processes would seem meaningless and thus difficult to perform and to remember. (p. 32)

To support her thesis Sfard resorts to complex numbers as an example, explaining that

> the concept of complex number cannot be reified until a person is able to make computations involving these numbers; at the same time, however, conceiving such constructs like i or 3 + 2i as fully-fledged numbers (and not just symbols for operations “without result") is a prerequisite for being proficient in manipulating them. (p. 32)

The operational/structural duality of mathematical concepts, as noted by Sfard, brings certain bearings with other conceptualizations such as procedural/conceptual understanding (Lesh and landau, 1983) which Sfard argues should be looked at dichotomies rather than dualities on the grounds that operational procedures are often a prerequisite of understanding concepts.

Instrumental and Relational Understanding

It is common in the mathematics classroom to carry out operations by just following rules, what skemp called ‘rules without reasons’ (Skemp, 1976). This is what Skemp refers to as instrumental understanding. This kind of understanding takes place when students follow procedures such as “‘borrowing in subtraction,’ ‘turn it upside down, and multiply for division by a fraction,' 'take it to the other side and change the sign' " (p. 10). However, rules can also be followed with understanding why they are used. This entails a different conceptualization of understanding, i.e., relational understanding, which functions on the grounds of 'rules with reasons. In Skem (1976) an advantage of instrumental mathematics is that students get immediate rewards when they carry out mathematical operations correctly which increases their self-confidence, while relational understanding results in understanding that can transfer to other environments and can be recalled for a longer time. Sfard, while viewing procedural/conceptual views of teaching/learning as dualities rather than dichotomies, proposes a third interesting possibility to Skemp's rules with and without reasons: "reasons without rules," which she describes as "purely intuitive understanding". Sfard (1991) suggests that, probably, intuitive understanding is the kind of understanding mathematicians explored at the early stages of mathematical concept development.

Teaching/learning within the Students' Zone of Proximal Development

The Zone of Proximal development (ZPD) describes a metaphoric space where teacher-student and student-student collaborative interactions can potentially be conducive to learning and cognitive development provided that the student stage of cognitive development is sufficient (Mahn, 2015; Vygotsky, 1978, 1986; Wertsch, 2007). According to these authors, intimately to sociocultural theory, discourse is a fundamental means to mediate these collaborative interactions which can result in learning, understood as conceptual development. Importantly for CLD students, the quality of the collaborative interactions can be enhanced by including bilingualism in the interactions (Moll & Whitmore, 1993). However, a fundamental distinction is made, that is, explicit and implicit mediation of concepts.

Explicit Mediation of Concepts

Vygotsky (1986) and Mahn (2015) argue that academic concepts develop through links to everyday concepts or previously learned concepts. In addition, they argue that academic concepts only be fully grasped if conscious awareness of the concept, volition in their use and systematicity in their organization in systems with other concepts are involved. Mahn considers key “the ability to voluntarily control the use of the concept” (p. 257). This fundamentally means that learning and development of CT-CP happens on the condition that concepts become part of students’ conscious thinking processes. Under this view, effective teaching must involve explicit discussions of concepts in processes where both teachers and students make available their thinking. In other words, In CT-CP teaching/learning environments both parties need to verbalize the concepts that structure their thinking about CT-CP activity in order for students to imitate teachers’ thinking processes, make meaning out of activity and learn (Mahn, 2015; Vygotsky, 1978, 1986).

In CT-CP research, similar conceptualizations of learning can be found. Lye and Koh’s (2014) in their systematic literature review on trends of empirical research in CT-CP, recommend fostering think aloud protocols during programming to foster CT-CP development. Lye and Koh advocate for the mediation of teachers who are recommended to ask the students to verbalize their thinking processes during programming. Indeed, Peng et al. (2019) found that making students aware of the stages of their programming projects from problem to solution with teacher *explicit mediation and conceptual feedback during the project and student articulation and reflection* on the task process helped students achieve a better performance in programming. Peng et al. found that implementing a progressive approach to programming in collaborative project-based programming and using scaffolds (Wood et al., 1976), specifically visual scaffolds, is beneficial to student learning as compared to overwhelming students with the complexity of programming. (Peng et al., 2019).

Implicit Mediation of Concepts: Learning 'Without Being Taught'

Edwards (2021) stresses the importance of the mediation role that teachers can potentially play during science lessons to promote higher order thinking. However, Edwards brings to the fore a different approach. While acknowledging the importance of explicit mediation, Edwards highlights the role of *implicit mediation* within the ZPD (Wertsch, 2007). Wertcsh argues that *implicit mediation* involves signs in the form of natural language which are less easily taken as objects of conscious reflection. In this line of thought, Shutte et al. (2018), drawing on Rowe et al. (2015) have used the notion of *implicit knowledge* to study CT-CP learning and assessment to refer to incipient knowledge that cannot still be articulated by the students but is evidenced in their actions. These ideas echo Papert's (1980) notions of 'learning without being taught.': "children learn to speak, learn the intuitive geometry needed to get around in space, and learn enough of logic and rhetoric to get around parents—all this without being "taught." (p. 7). Papert argues that the critical factor that explains the slow learning of CT-CP is not the complexity or formality of concepts but "the relative poverty of the culture in those materials that would make the concept simple and concrete" (1980, p. 7) To facilitate learning, Papert proposes to teach the child a method, "a heuristic procedure… [a method that can] "establish a firm connection between personal activity and the connection of formal knowledge (p. 58-59)

To support this view, Papert uses as an example the concept of quantity, arguing that our culture is rich in language for talking about one-to-one correspondences, pairs, and couples, which helps children learn about quantities without being taught. By contrast, Papert contends, our culture does not provide models of systematic procedures so frequently. In this respect, Papert, who referred to computers as 'objects-to-think with' uses as an example the concept of 'loop' to refer to concepts that are not typically found in the students' everyday. Loop is a fundamental CT-CP concept which refers to the capability that computers have for automation and describes the continuous repetition of a process while a condition is met (Bourke, 2018). Like Papert, Bourke considers procedures essential. Papert stresses the importance of procedurality and systematicity in programming, arguing that "as educators we can help by creating the conditions for children to use procedural thinking effectively and joyfully (p. 154-155). Procedurality is a key concept in CT-CP as reflected on the very nature of some CT-CP paradigms (cf., Dümmel et al., 2019). A concept related to procedural thinking is procedural abstraction, that is "the concept that a procedure or sequence of operations can be encapsulated into one logical unit (function, subroutine, etc.) so that a user need not concern themselves with the low-level details of how it operates" (Bourke, 2018, p. 594). Like Papert, Bourke argues that to help programmers work in their procedures, languages and CP educational approaches typically store libraries or "black boxes," which contain functions and procedures so that programs don't have to be written from scratch.

Therefore, according to Papert, given that children do not have easy access in their everyday to systematic procedures, and that they are the active constructors of their knowledge, what can be done is provide them with 'objects-to-think-with, i.e., computers, and learning environments and materials that afford "concrete ways to think about problems involving systematicity" (p. 22). Papert supported his views on problem solving tasks on Polya's (1945) ideas on heuristic learning (see table 3), an approach which focuses on learning as a discovery of the learner, with the teacher playing the role of providing the conditions for learning to happen (Newell, 1981; Polya, 2004). Rather than approaching teaching through a propositional approach where disciplinary content is completely specified, Papert advocates for an approach which can bring together student intuitions—gained by using computers, the appropriate materials, formal methods, and motivational learning environments. Heuristics places the emphasis on student experimentation and discovery and on their evaluation of possible solutions to complex real-world interdisciplinary problems which, based on the students' interests, are both challenging and accessible while relating to mathematics concepts (Polya, 2004). In contrast, experiments in physics, for instance, are typically designed "to prove, disprove, and "discover" already known propositions" (Papert, 1980, p. 139). Newelll (1981) highlights three characteristics which 'loom largest' in Polya's heuristic approach and are critically involved in problem solving tasks: Attention, memory, and motivation. "Attention in humans does not happen automatically but occurs only because the problem solver picks up this and that part of the problem, successively attending to the parts and then to their relations" (Newell, 1981, pp. 6-7). Memory is critical too. To get engaged in a problem-solving task students need to relate it to a previous problem-solving experience. Lastly, motivation is crucial as well. According to Newell, children are naturally curious and interested in exercising competence and searching for solutions to complex problems. The stages of problem-solving tasks are: (1) Understand the problem; (2) Design a plan; (3) Carry out the plan; and (4) Examine the solution. Newell stresses the importance of examining the solution in preparation for future problem-solving tasks where the same cycle would repeat. Following I include Polya's (2004) guide which was created to support teaching meaningful problem solving in middle grades:

Table 3

Guide to Meaningful Problem Solving Tasks

Guide to Meaningful Problem Solving Tasks

		Well-crafted tasks meet these strict criteria
1		♦ Engage and interest students ✓ Apply to the real world ✓ Connect to student interests ✓ Are equitable in that they appeal to *all* students ✓ Promote active involvement
2		♦ Contain important mathematical content ✓ Connect to other problems and mathematics concepts ✓ Align with current mathematics curriculum ✓ Integrate other subject areas
3		♦ Are open-ended and nonroutine ✓ Allow multiple approaches and solutions ✓ Are not readily solvable by using a previously taught algorithm
4		♦ Are challenging but accessible to students ✓ Require persistence ✓ Allow entry to the problem
5		♦ Are well-crafted ✓ Contain clear and unambiguous wording ✓ Describe expectations ✓ Elicit responses that can be scored

In the following section, I center on educational research, both theoretical and empirical, that has used participation as the fundamental unit of analysis. This fundamental research move entails theorizations of learning-as-participation which complement learning-as-cognitive development approaches as the ones discussed up to this point.

Teaching/Learning as Developing the Discourse of a Discipline

Sfard (2001a, 2001b, 2008) views mathematical discourse as tantamount to thinking and defined "learning as the process of defining one's discursive ways in a certain well-defined manner" Sfard (2001b, p. 3). Then, Sfard (2008) elaborated on Sfard's previous views on communication and thinking, developing a commognition framework with relevant tools to investigate and describe students' participation in mathematical discourse. More recently, Lavie and Sfard (2019) studied Milo's development of counting, a toddler who they followed for 18 months since he was 2 years and 8 months. Based on their investigation, Lavie and Sfard argue that when a child has learned to count, they have individualized the uniquely human routine of counting. "Collectively shaped over generations, these activities now enter the child's repertoire through *individualizing*, the process in which one gradually becomes capable of agentive participation" (p. 424). Activities related to counting such as producing statements about properties of objects, once individualized, become a part of the child's thinking, and can be communicated. Lavie and Sfard (2019), drawing on Sfard (2008) describe numerical discourse based on particular *routines, keywords*, *visual mediators*, and *endorsed narratives* that constitute it. In numerical discourse, counting is a typical routine; numbers and terms related to quantities such as bigger, more, etc. are some of its keywords; numerical symbols such as digits and numerals are its visual mediators and statements such as "seven is more than five" are an endorse narrative, "which mathematicians consider as useful and describe as true" (Lavie & Sfard, 2019, p. 424).

Martin and Betser (2020) used Sfard's commognitive framework to study youth learning and participation and development of engineering discourse in a youth out-of-school maker club.

The students were engaged in established practices of engineering such as reverse engineering (disassembling a printer) and design (brainstorming). The authors explored learning as shifts in newcomer's participation in engineering practices (routines), based on students' agency and development of engineering discourse. The development of student engineering discourse was based on student use of specialized engineering keywords such as energy and failure, visual mediators such as prototypes and sketches, and endorse narratives such as the accepted-as-fact idea that an electric motor converts electrical energy into mechanical energy. Martin and Betser's examination of spoken interactions and development of engineering discourse in selected oral events centered on established engineering practices and allowed the exploration of engineering learning as a newcomer's increased participation in engineering discourse.

Accordingly, in CT-CP environments some established practices include creating computational artifacts, problem decomposition, testing and debugging, refining products or their affordances, testing and debugging, collaboration and creativity (Grover & Pea, 2018; Kong, 2019). Some key words include specific CP-CP concepts and functions; some visual mediators include prototypes; and some endorsed narratives include the accepted-as-fact idea that computers need "truth statements" in order to function (Ben-Ari, 2012).

Teaching/Learning as Reification of Student Thinking into Artifacts

Like Wenger (2010), Bholah (in science education research) (2017) views learning as processes which revolve around teacher facilitation of activities that result in producing a product that displays students' thinking. These activities are generally conducted in collaborative learning groups that attempt to solve the same problem in different ways, and they often arrive at different answers" (p. 120). Therefore, Bholah advocates for teachers that, rather than "transmit" knowledge, facilitate socio cognitive practices that result in tangible products. A key aspect of CoP theory is that students' learning trajectories take place around engagement in social practices that result in reification of artifacts. According to Wenger, "meaningful learning in social contexts requires both participation and reification to be in interplay" (2010, p. 2). This interplay of social-based participation in practices and their resulting artifacts define history of learning and 'shared repertoire' of CoP like AOLME (The Advancing Out-of-school Learning in Mathematics and Engineering). The practices enacted and artifacts produced in a community of practice have a "life of their own," constituting a "response" to what is proposed to the students (Wenger, 2010, p. 2). Wenger claims that the artifacts produced because of participation in activity enrich the negotiations of meaning at hand. "So, you always have to look for both processes whenever you try to understand a moment of meaning making. Reification requires participation. And when reification and participation are separated, continuity of meaning is not guaranteed." (Fansworth et al., 2016, p. 9). Twenty-five years ago, Kafai and Resnick (1996) explored the interplay of learning CP and design by engaging students through their implication in their meaningful contexts to create their own video games. In their study, the educational focus was the relationship established between the creator (the student) and the computational artifact developed (i.e., the video game). Learning was viewed as generated through the students' identification and representation of their ideas in video games and stories.

Teaching/Learning as a Restructuration of Epistemologies in a Disciplinary Domain

Papert (1980) suggested that interacting with 'objects-to-think-with' , that is, computers, in digital computation-based learning environments can 'open intellectual doors' to students. According to Papert, in these environments students typically encounter a constellation of new ideas, "taking a step toward an epistemology of powerful ideas" (p. 137). They can learn, for instance, about being in command of their own learning processes, about new ways to understand errors, about systematic procedures to approach solving complex problems and tasks. Chalmers (2011) argues that if someone does not know anything about something, all scenarios are possible. When students take their first steps in a discipline the new knowledge they gain does not fall on vacuum. According to Chalmers, disciplinary scenarios constitute *epistemic space* where different *epistemic possibilities* can be explored. Chalmers suggests that during practice, *epistemological adaptations* typically happen. Chalmers argues that the notions of epistemic space and epistemic possibilities can be applied to the examination of a variety of domains.

To frame discussion on computer games epistemologies, Shaffer (2006), used Crowley and Jabob's (2002) notion of islands of expertise which are defined as general knowledge that students and other members of a community have already developed and can use as a baseline in different contexts. According to Shaffer (2006)

> islands of expertise include development of identity and adoption of practices associated with the ways of knowing of a particular community. That is, I argue that islands of expertise are organized around coherent epistemic frames, and that these frames – these ways of looking at the world associated with different communities of practice – are the ''abstract and general themes'' that students use to leverage experience in an island of expertise in new situations. (p. 232)

Students develop islands of expertise in and out of school, based on any topic which interests students to the point that they develop relatively deep knowledge. Video games are an example of student islands of expertise in which students have organized video games knowledge and practices in *epistemic frames* or ways of knowing video games. This stance was taken by Kafai (1995) in a study that uncovered how CT-CP practices transformed traditional school approaches to writing stories and, consequently, students transformed their ways of knowing about story writing and the medium to expressing them. Additionally, the computer became a new medium for the students' creative expression.

In similar lines, Wilensky and Papert (2010) use the construct of *domain restructuration* to frame discussion on the evolution of mathematics and science. They argue that human advancements in knowledge, such as the ones triggered by Euclides, Descartes, Galileo and Newton, have historically entailed a domain restructuration. Wilensky and Papert explain the concept of domain restructuration through a definition of structuration of a domain,

> By structuration we mean the encoding of the knowledge in a domain as a function of the representational infrastructure used to express the knowledge. A change from one structuration of a domain to another resulting from such a change in representational infrastructure we call a restructuration. (pp. 2-3)

A good example of domain restructuration provided by Wilensky & Papert happened in the shifting from Roman to Hindu-Arabic numerals in arithmetic around 2000 years ago. For Wilensky and Papert these shifts afforded progress in handling numerical relationships, thus epistemological progress in mathematics as a discipline. Shifts in representations and *domain restructuration* frame the disciplinary evolutions furthered by Euclides, Descartes, Galileo, and Newton. Drawing on Wilensky & Papert's (2010) construct of *domain restructuration*, Arastoopour et al. (2020) found that engaging students in computational activities resulted in a restructuration of their understandings of biological ecosystems thus a restructuration of the biology ecosystems domain. The student participants in their study connected micro-level actions such as eating and reproducing with macro-level concepts such as stability and extinction.

Edwards (2021) also focused on epistemological considerations. Edwards explains that modeling, which is broadly used in science and other disciplines, is a key *epistemic element* which students can transfer to varied contexts. For Edwards, developing expertise is developing expertise of a particular kind from the particular epistemic frame (ways of knowing) of a particular community of practice. Edwards posed the question: "How do the students come to know what they know in science?" (p. 146). Drawing on Shaffer's (2006) research on educational utilities of computer games, Edwards (an educator and researcher) focused on science education epistemic concerns. Edwards focused on whether teachers target in their teaching practice science-as-inquiry, science-as data driven, science-as-fact or science-as-practice. "These epistemic frames are also associated with particular communities of practice and serve as the organizing principle for practice" (Edwards, 2021, p. 152). Edwards argues that epistemic frames can transfer, conceptualizing learning as a transfer of epistemic frames. Edwards draws on the idea of "science-in-the-making" (Schuster et al., 2018) stressing the view that, during practice, teacher and student make shifts in their epistemic agency.

Therefore, it can be argued that when students take their first steps in a discipline such as CT-CP they have expertise in domains that relate to that discipline in some way. The new discipline offers them an epistemic space or ways of knowing where new epistemic possibilities or ways of knowing will be possible. In these new epistemic spaces students can make shifts in their epistemic agencies and go through *epistemological restructurations of their domains of* *expertise.* It follows that as CT-CP researchers we can explore learning around student shifts of their ways of knowing in a CT-CP CoP.

Teaching/learning with a Focus on Student Participation and Discourse

In this section I discuss several relevant approaches that offer complementary ways of looking at teaching/learning processes to the above discussed cognitive approach. The approaches to teaching/learning I explore can broadly be referred to as 'participationist' (Sfard, 1998) and constitute a complementary approach, as opposed to exclusionary, to cognitive perspectives. Participationist approaches foreground student collaboration and talk, placing emphasis on student development of disciplinary discourses. In mathematics education research, Österman and Bråting (2019) have argued for placing more emphasis on operational skills (without disregarding concepts) on the grounds that "students' numerical and computational skills have suffered over the years" (Billington & Gabrielsen, 2017, p. 467). This argument resonates with the above discussed views that place emphasis on initial focus on operational fluency as a building block towards subsequent conceptual development (Sfard's 1991). From a comparable perspective, Lavie et al., (2018) suggests "bypassing understanding" in educational research, on the grounds that understanding is not an operationalizable construct.

Although there is a growing body of literature on K-12 CT-CP educational research, most of the research has been conducted at the university level (Nouri et al., 2019). Grissom et al. (2017) found in their study on over 700 US CS faculty that only 20% rely on student interaction on a regular basis in class. They also found that about 38% use lectures for lesson content delivery. Grissom et al., call for a greater use of student-centered approaches to contribute to student learning and remaining in CS fields. Focusing on younger students, Waite (2017) conducted an extensive literature review centered on pedagogical approaches used in K-12 CS and CT-CP education which included reviews of studies and independent studies in different continents. Their review concluded that while teachers are starting to adopt approaches that aim at student active participation, further robust research is required to verify such adoption. Waite (2017) found that while teachers often count on a plethora of technology to teach computing, they often focus on content rather than on pedagogy (Kafai & Vasudevan, 2015; Rich et al., 2017), lacking the pedagogical guidance needed for effective teaching and learning. Waite argues for the development and testing of pedagogies in situ, advocating for motivational lessons to ensure the active participation of all students irrespective of their prior experience in computing.

Over the past 25 years significant research efforts have targeted student classroom discourse (O'Connor & Michaels, 2019). In CT-CP, Grover and Pea (2013) advocate for an intensive CT-CP discursive approach both for practice, claiming that talk can shape CT development. In engineering, Martin and Betser (2020) use Sfard's (2008) discursive commognitive framework to track student learning, based on the view that developing the discourse of engineers is part of learning to be an engineer. In mathematics education research, Sfard (2001b, 2008), identified thinking and communication and learning mathematics as developing mathematical thinking and mathematical discourse. To promote discourse in the classroom, talk tools or talk moves such as 'revoicing' and 'pressing to think' have been explored and used to position students as thinkers, arguers, and makers of meaning (O'Connor & Michaels, 2019). According to O'Connor and Michaels, teachers' use of talk tools such as revoicing open student discursive participation. These moves can be used by teachers to "encourage students to dig deeper into their own reasoning" i.e., "Why do you think that?", "Can you explain your thinking about that?", "Does it always work that way" (p. 23). O'Connor and Michaels argue that teachers can, if skillful in the use of talk moves, provide access to intellectual content while ameliorating the effects of racism, sexism, and prejudice of all kinds.

Student interaction lies at the heart of collaboration, and student discussions are collaborative in nature (Chapin et al., 2009). Collaboration and talking go hand in hand and are conducive to learning directly and indirectly. According to Chapin et al.(2009), classroom dialogue also supports student learning indirectly, through the building of a social environment—a community—that encourages learning. Chapin and colleagues discussed two different discourses which are crucial to learn mathematics: one is related to doing mathematics; the other to social skills such as getting along well. In similar lines, Bernstein (2003) studied teachers' pedagogical discourse, distinguishing a discourse of competence, i.e., instructional discourse, from a discourse of social order, i.e., regulative discourse. According to Bernstein, the instructional discourse is embedded in the regulative discourse and takes care of the 'content' of the discipline. On its part, the regulative discourse concerns pedagogical aspects such as the organization and sequencing of activity, collaborative work, and student behavior.

In line with Chapin and colleagues, a considerable body of research has focused on the promotion of classroom discourse. These studies have yielded findings and useful constructs to examine discourse. Sinclair and Coulthard's (1975) Initiation-Response-Feedback (IRF) patterns suppose limited invitations for student participation after a closed question (Initiation) that typically elicits a right/wrong answer and minimal feedback, as opposed to dialogic teaching, which promotes student participation, collaboration, thinking and learning (Alexander, 2018;

Cazden, 2001; Mortimer & Scott, 2003).The above empirical studies shed light on the relevance of promoting student talk and participation and exploring teaching discursive resources to do so. Early studies such as Mehan (1979), which explored the means to achieve student ‘effective participation’ had paved the way. In the following section I focus teaching/ learning from a ‘participationist’ perspective (e.g., Sfard, 1998) based fundamentally on CoP theory and a few notions of Sfard’s communicational approach to cognition or commognition approach (Sfard, 2001, 2008)

Broadening Participation to CLD Students in CT-CP/CS

Relevant research has pointed at United States governmental policies which aim at the promotion of equitable practices capable of reaching CS education for all students (Goode et al., 2020). Goode et al. advocate for inquiry-based pedagogies and a focus on inclusion and equity that ensure the diversification and democratization of CS. Importantly, in mathematics education research and practice, Celedón Pattichis et al. (2018) critiqued the United States system of tracking in schools which situates students from historically marginalized communities in classrooms with lower-level instruction. A common result is that not all students are challenged equally. Celedón Pattichis et al., following the US National Research Council of Teachers of Mathematics (NCTM, 2017), argue that high quality mathematics education for each and all students requires systemic change. To produce change, Celedón Pattichis et al. advocate for asset-based teaching approaches which draw on students’ language and culture as intellectual resources. Other relevant ideas in their study include collaboration between teachers, families, administrators and the students, and the idea of learning ‘with and from students’. In STEM teacher identity research, Keiler (2018) centered on ways to support teachers in meeting the needs of each and all students in high needs urban schools through implementation of student-centered pedagogies (eg. Darling-Hammond & Richardson, 2009; Emerling & Gallimore 2014). After collaborating closely with teachers and students, Keiler demonstrated that changes in the learning environment that include collaboration, discussion and peer-mediated instruction can lead to relevant shifts in teaching approaches. Teachers experienced shifts in their roles from deliverers of content to “developers of human potential,”“transitioning from changing from being the instructional star to being the director of learning” (p. 14). Papert (1980), arguably one of the fathers of CT-CP, rather than conceptualizing teachers as lecturers who feed content and concepts into the students, viewed them as facilitators who set up the conditions where students can learn. It seems relevant to consider teaching/learning conditions that adapt to the needs of CLD students. In a more recent study, Collins et al. (2020) argued for the relevance of targeting CS educational efforts at supporting the learning and development of outreach participants by implementing culturally sustaining pedagogical practices. Collins et al. described and emphasized the importance of supporting CLD needs by bringing in pop-culture into youth CS projects, developing positive relationships and communication between all the staff involved in CS educational projects, and supporting the teaching agency of undergraduate CS students.

By contrast, student-centered approaches have been questioned internationally in science teaching/learning in a study conducted on 54 countries (Cairns & Areepattamannil, 2019) and the effectiveness of traditional instructional approaches shown (Klahr & Nigam, 2004) More specifically, direct teaching has been found to be more successful in CS K-12 education to multilingual students (Jacob et al., 2020). Jacob et al., Jacob et al., found that inquiry-based CS teaching/learning without sufficient conceptual schema building, even when adapted linguistically and culturally to CLD students, can lead to lost opportunities and disaffection with the discipline. Jacob et al., argue that to broaden participation in computing, providing experiences is not enough; successful experiences are important. Balance could be the key ingredient for effective teaching/learning CT as claimed by the National Research Council (2011) who advocate for balancing the student engagement in programming with motivating them to learn complex concepts.

Teaching/learning as Shifts in Patterns of Participation in CoP

Without disregarding other approaches such as psychological or political approaches (Farnsworth, 2016), CoP theory (Farnsworth, 2016; Lave & Wenger, 1991; Wenger, 1998; 2010; Wenger-Trayner & Wenger-Trayner 2015) decenters individual cognitive processes in analyses of teaching/learning, placing the emphasis on practice and student agency and participation in social communities of learning (Wenger, 2010). With a strong socio-cultural foundation (Egeström, 1987; Vygotsky, 1978; Wertsch, 1995, 1998), CoP is a theory that views learning as participation in a shared domain of human endeavor (Wenger-Trayner & Wenger-Trayner, 2015). "Communities of practice are groups of people that share a concern or a position for something they do and learn how to do it better as they interact regularly" (Wenger-Trainer & Wenger-Trayner, 2015, p. 1). In other words, CoP share a 'domain of interest' around which people gather regularly with the goal of learning how to do it better.

CoP theory views learning fundamentally as a social process of enculturation in CoP which happens through expert-novice interactions (Lave & Wenger, 1991). Thus, Lave and Wenger identified learning with participation in the cultural practices of a community. They defined CoP as follows:

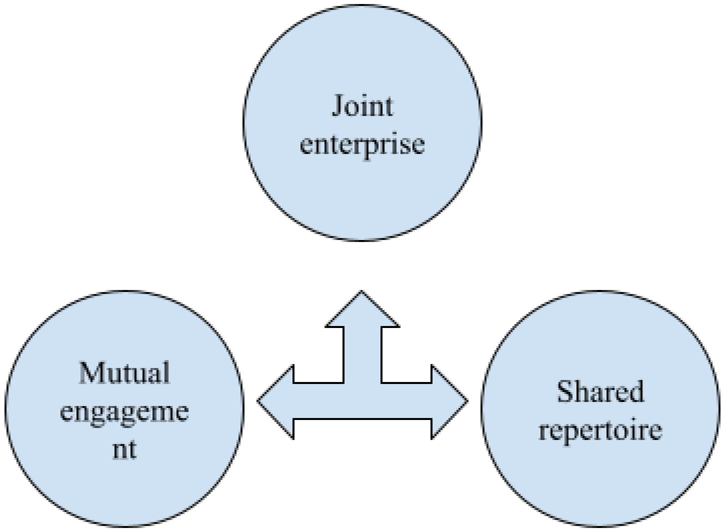
> A community of practice is an intrinsic condition for the existence of knowledge, not least because it provides the interpretive support necessary for making sense of its heritage. Thus, participation in its cultural practice in which any knowledge exists is an epistemological principle of learning. (p. 98)

Lave and Wenger's words suggest that communities of practice provide the means to make sense of cultural practices through participation. Indeed, CoP theory provides a view of *learning as shifts in patterns of participation overtime.* According to Wenger (1998), apprentices participate from the start. Wenger uses the term peripherality to support this idea. Peripherality is a form of participation that involves actual practice, that is, newcomers perform the same activities that full participants of the CoP perform, only with “lessened intensity, lessened risk, special assistance, lessened cost of error…” (p. 100). Student newcomers may or may not be legitimated (‘membership’) by their facilitators and the rest of the CoP to engage socially in the community defining practices (and may decide to stay or go). In their trajectories from peripheral participation to full membership and even central leading positions, apprentices learn through engagement in the practices of a CoP. The practices of a CoP have three interconnected dimensions which shape each other (Wenger, 1998):

1) Joint enterprise
2) Mutual engagement
3) Shared repertoire

Figure 1

The Three Dimensions of Practice (Wenger, 1998)



According to Wenger, the joint enterprise is the basis around which participation in the CoP happens. It agglutinates the goals of the CoP. A good example of a joint enterprise can be the videos developed by the students in Python code, i.e., one central AOLME goal. The mutual engagement refers to communication, search for agreement and collaboration in the practices of the CoP. Finally, the shared repertoire includes "*routines*, words, tools, ways of doing things, stories, gestures, symbols, genres, actions, or concepts that the community has produced or adopted in the course of its existence, and which have become part of its practice" (emphasis added) (p. 83).

Examination of the dimensions of practice allows identification of patterns of participation in particular CoP and consequently of opportunities to learn (Chauraya & Brodie, 2018). According to Wenger-Trayner and Wenger-Trayner, 2015, all CoP have an identity defined by a '*shared domain of interest'* where membership implies a commitment to the domain and consequently "a shared competence that distinguishes members from other people " (emphasis added) (p. 2). Thus a CoP such as AOLME engages their members socially, i.e., “mutual engagement” around a “shared repertoire” of elements (AOLME’s practice) to program videos in Python among the rest of AOLME’s “joint enterprise”, i.e., AOLME’s objectives.

Learning Originates in Student Agency

Wenger (2010) affirmed that *agency plays a key factor in learning* and in the modulation of the degree with which people, that is, students identify with the practices and people of a CoP. According to Wenger, getting involved with depth in any practice entails what he calls “modes of identification” with a particular CoP, claiming that “as we (and by extension communities) negotiate our participation in broader systems, we need to make sense of both the system and our position in it” (Wenger, 2010, p. 4). Wenger distinguishes three different modes of identification with practices: (1) Engagement, which includes “doing things,” “talking” and “producing artifacts”; (2) Imagination as a resource which “helps us understand how we belong or not” for which we can retort to “tools of imagination” such as pictures, stories and role models; and (3) Alignment with others and the CoP which involves coordination, communication and active participation “not merely compliance or passive acquiescence” (Wenger, 2010, p. 4). Wenger claims that alignment is a “two-way process” in which people and the CoP as a whole coordinate perspectives and actions so that “actions have the effects we expect” (p.5). According to the three parameters shown, students can evaluate whether they strive to get involved in a particular CoP. It is the students (novices) who imagine their futures, do the practices, and become full members with time if they choose to do so, but in a two-way process. Facilitators (experts) may open more or less the floor for them to participate.

To summarize, novices and experts of CoP align interests agentively in two-way processes of learning in which they actively participate through talking, doing, and producing artifacts that reflect the meanings realized in practice, which are coordinated around shared repertoires such as routines, tools and ways of talking and doing things that are constantly renewed in pursuance of a joint enterprise.

Learning as Agentive Explorations in Established Disciplinary Practices

Lavie et al. (2018) suggest to "bypass" the word *understanding* in educational research, arguing that the word understanding, while omnipresent and important in educational research, is not operationalizable. Lavie et al. assert:

> The word understanding, therefore, escapes operational definition, and when used, it blurs more than it clarifies. As such, it is practically useless for the researcher who wants her stories of learning to be truly helpful to practitioners. Indeed, what is the use of advice that can be interpreted by a person any way she wants? (p. 173)

As an alternative, Lavie et al. (2018), whose work draws on Sfard (2008), propose the use of the construct *routine* to conceptualize learning as a process of *routinization* of learners' actions. Lavie and Sfard (2019) define *routines* as "ways of doing things" that can be found in disciplinary discourses such as mathematics. Lavie et al. (2018) argue that "learning is a process of gradual routinization of our actions, investigating learning is tantamount to answering the question of how routines emerge and how they later evolve" (p. 162). According to Lavie et al., (2018), to identify routines, the observer must get at the history of participant actions. "The greater our access to the history of one's action, the more robust our claims on this person's evolving routines become” (p. 162). Lavie et al. (2018) distinguish between process-oriented rituals which participants follow simply to get social recognition from product-oriented routines. Discursive routines are called *explorations* and, although they entail following what others do, “bring novelty” and participant agency and creativity to activity. In contrast with the construct “understanding,” *exploration* is a “detectable” construct that can be used to track learning, both at the individual and collective level.

For instance, counting is a numerical routine. “Numerical routines are patterns of action that appear when people participate in numerical discourse” (p. 424). Learning has been conceptualized as agentive participation in *routines* which, if product-oriented, are called explorations (Lavie et al., 2018; Lavie & Sfard, 2019). According to Lavie et al. (2018), “learning is a process of gradual routinization of our actions, investigating learning is tantamount to answering the question of how routines emerge and how they later evolve” (p. 162). When students encounter a new discourse, they can only participate by following a preset procedure in search of social recognition, without contributing much to the discourse personally or intellectually. However, in further learning, students become capable of contributing in more agentive ways. Under this view, learning in a product-oriented CoP can be conceptualized as explorations in the routines of said CoP, and can be explored by examining the histories of the learners' discursive actions to obtain products that are meaningful to said CoP. Lavie et al.,(2018) suggest that learning can be identified in shifts in the characteristics of explorations or product-oriented routines. Lavie et al. define a routine as a task-procedure pair where a performer executes a procedure in order to execute a task. Some characteristics of explorations include flexibility (there is more than one way to perform a task), bondedness (the steps of a procedure feed into later steps towards obtaining a final product), applicability (lessons learned in a routine and environment can be applied in a different routine and environment), performers agentivity (participants make decision until they gain independence, in response to their own needs), *objectification* (abstract objects such as integers substitute concrete objects and procedures): “Objects are reifications of known procedures and can thus be regarded as condensed precedent sets” (p. 171), and, finally, *substantiation* (students can be called on to explain what they did). Their account would constitute a substantiation of the product-oriented task-procedure they adopted, which would provide relevant clues on their learning process.

A focus on Students’ Opportunities to Participate: Positioning

Herbel-Eisenmann et al. (2013) and Arkwash and Morgan (2018), among others, have studied teacher discourse in search of the opportunities that their discourse opens for student participation in mathematical practice. Herbel- Eisenmann et al. refers to positioning as a lens to study students’ opportunities to learn. These authors stress the importance of maintaining a dual focus on positioning that allows examination of both what constitutes doing mathematics and how interpersonal and dynamic issues such as authority, agency and power are negotiated. SFL allows examination of discourse in both respects, providing specific tools to analyze the nature of a field of experience and the interpersonal relations established between teachers and students (Halliday and Matthiessen, 2014). I will explain this in detail in the theoretical framework.

Entry Points to Participate in CT-CP Practices.

In this section I discuss relevant research that has pointed at specific CT-CP practices as entry points to students learning CT-CP.

Mathematics and Modeling in CT-CP Interdisciplinary Activity

Modeling with mathematics can be an entry point to CT-CP development and provide a window into students' CT-CP learning trajectories (Arastoopour et al., 2019; Hoppe & Werneburg, 2019). According to Hoppe and Werneburg (2019), CT-CP becomes evident in processes that involve abstraction-based entities such as the programmable models used across disciplines, both in the humanities and in the sciences. In CT-CP processes the importance of mathematics is evident, since mathematics can provide the formalisms needed to represent systems faithfully and fully in ways that computers can process (Reeves & Clarke, 1990). Mathematics affords the means to make sense of the world in a systematic way (Lehrer et al., 2001) thus playing a key role in modeling in disciplines such as biology (Arastoopour et al., 2019), physics (Farris & Sengupta, 2014), science (Farris et al., 2016) and the creation of videos in Python (LópezLeiva et al., 2019). Furthermore, using mathematics as a tool, as it is used in modeling processes, has an impact on preexisting mathematics knowledge; mathematics is in reciprocal dependence with practical challenges (Lenhard & Carrier, 2015).

A CT-CP model is a mathematics-based representation of an object or system that can be processed and executed by the computer (Arastoopour et al., 2019) In modeling processes a useful construct is mathematization, a technical term that describes the mathematical representation of characteristics and relationships of the world (National Research Council, 2012). Accordingly, to "mathematize" refers to the act of representing characteristics or relationships of the world with mathematical symbols. It follows that, in creating with a computer a digital image of, for instance, a tree, to mathematize refers to the act of representing the tree with the mathematical formalisms that make computer processing and execution possible.

For Hoppe and Werneburg (2019), “the essence of CT-CP lies in the creation of “logical artifacts” that externalize and reify human ideas in a form that can be interpreted and “run” on computers (p. 13). Logical artifacts, computational artifacts or computational models can be equivalent constructs in CT-CP environments that retort to modeling as a resource to represent objects and systems in computable ways. Drawing on Aho (2012) Hoppe and Werneburg make a key terminological distinction which can have fundamental research and pedagogical implications. One thing is a computational artifact or computational model, and a very different thing is a model of computation. Computational artifacts or models are the resulting artifacts of CT-CP activity as processed by computers, whereas models of computation are the prototypes that “model the world” by means of “abstractions as constructs”. This dissertation examines the digital images created by the students, that is, computational artifacts. To create digital images, the students modeled with abstractions pencil-crayon drawings to obtain prototypes or models of computation.

To “model the world” and create models of computation or prototypes of for instance the characters of a video, students need to retort to “abstraction as constructs” (e.g., numbers or other entities), with which prototypes can be mathematized and programmed. Mathematics abstractions provide the stable, precise, and unambiguous truth required in CT-CP/CS. The abstractions as constructs (e.g., mathematical representations and CP data structures) have no flexibility and often are “predetermined and not the focus of the learners’ own creative contributions” (Hoppe & Werneburg, 2019, p. 28). However, in addition to providing the students with these inflexible constructs, it is important to promote flexibility in student algorithmic strategies. It follows that in environments that focus on the creation of digital images the students can and should be given flexibility in developing the prototypes and algorithms that compose their programs. This room for maneuver allows for the inherent creativity needed in CT-CP (Kramer, 2007). Krammer suggests that in the process of developing a painting, map, or a computer program, one needs to be able to move from a concrete domain to a broader domain by abstracting common properties and extracting common features.

In this respect, CT-CP has been defined as an attitude that allows modeling problems (or tasks) with executable formalisms (mathematics abstract representations) while looking at them at different levels of abstraction to gain and refine insights on the temporal evolution of the problem/task at hand (Arastoopour et al., 2019; Priami, 2007). When programmers want to fulfill a task or solve a problem with a computer, their thinking moves from the higher level of the task, with all its constituents' details and characteristics to the lower level of the execution of the task. In doing so, programmers also move through intermediate levels that include the formulation of the task or problem and the program developed. As computable problems or tasks are programmed, their evolution can be checked based on the algorithm used (series of ordered instructions) and the automations achieved by the computer (ISTE & CSTA, 2011). Therefore, in CT-CP practices aimed at producing computational artifacts, computer programmers move between different LOA.

A Focus on LOA

As students develop their programs, for instance to create digital objects, the students need to move between the lower level of execution of programs and the higher level of the objects that they want to create. LOA are key in CT-CP/CS education because students need to interpret object-process dualities at different LOA (e.g., Perrenet et al., 2005; Perrenet &

Kaasenbrood, 2006). In computing, LOA has been conceptualized as a transition between the high level of what is wanted to be achieved through the CP process and the low level of how it is done (Taub et al., 2014). In classroom activities the transition of LOA has been conceptualized from the higher LOA of student discussion in English about a particular task or problem and what the computer does during run time (Waite et al., 2016). Perrenet et al. (2005) defined four levels of abstraction as a basis to explore whether Bachelor students of CS were beginning to think like computer scientists. The goal of their work was to support students thinking about algorithms and help them progress in programming. Based on the work of Skemp and his successors in mathematics (Cf., Tall & Thomas, 2002), they proposed three LOA for the algorithm concept, the *program*, *object,* and *problem* levels. After the study, which was based on written questionnaires filled by three Bachelor year groups, they added a fourth level of abstraction, the *execution* level or level where the algorithm is a specific run on a specific computer. Therefore, the LOA identified, from lower to higher level of abstraction were the following:

(1) The execution LOA: the algorithm is a specific run on a computer.

(2) The program LOA: the algorithm is a process formulated in a specific CP language.

(3) The object LOA: the algorithm can be viewed as an object that is not connected with a programming language yet.

(4) The problem LOA: At this LOA, the question is to find a suitable algorithm for a given Problem.

Following up on Perrenet et al. 's study (2005), Perrenet and Kaasenbrood (2006) strengthened the validity of the method used by Perrenet et al.,'s and confirmed the four LOA identified previously. Cutts et al. (2012) explored LOA in the student talk of 570 participants during peer instruction in an introductory CP course for undergraduate students provided at a "research-intensive" institution in the US. The students used computer programming tool Alice to program stories with the assistance of more advanced peers who supported them in creating their programs. While programming and later in lectures, experts and novices interacted, asking, and responding to formative assessment questions that aimed at supporting student cognitive apprenticeship in CP and transitioning between LOA. An example of a question asked to a student who was programming a story was: "How would you change the code to make her say hello while jumping up and down?" (p. 5). After analyzing all the responses categories were generated and LOA identified. Three LOA were identified in the expert novice discussions, which were refined to constitute what the authors called "Abstraction Transition Taxonomy". Cutts et al.'s abstraction taxonomy was formed by the following LOA

(1) The Code LOA: Discussion revolves mostly around the code.

(2) CS Speak LOA: Discussion involves specialized CS terminology.

(3) English Speak LOA: Discussion involves a scenario or goal with no use of CS specialized terminology.

At a later stage in their research, Cutts et al. used their taxonomy to identify in exams these three levels of abstraction. They found in the students' responses the three referred LOA variety of transitions between LOA which exemplified student progress in CT-CP. Cutts et al. suggest that asking students why and how questions around the three LOA can help them rationalize and describe what they did and thought in their CP process. Cutts et al. drew attention to the different vocabulary used at different LOA, ranging from everyday English through the problem

LOA to the more CS specialized vocabulary in discussions around the code that solves the problem. Cutts et al.'s findings suggest that students who are aware of the LOA they are working at and experience moving between LOA have a better chance to become competent programmers.

Armoni (2013) focused on grades 7-9 with novice students in CT-CP/ CS, suggesting a framework to teach abstraction and LOA based on previous literature and some anecdotal evidence. The framework suggested by Armoni modified the LOA proposed by Perrenet et al. (2005) and Perrenet & Kaasenbrood (2006) in one of the LOA. Armoni's LOA framework (2013) is the following:

(1) The execution LOA

(2) The program LOA

(3) The algorithm LOA

(4) The problem LOA

In my view, Armoni's main contribution revolved around language use. The framework suggests teachers be precise, use different vocabulary at different levels of abstraction, distinguish clearly between the different LOA and work with students from the problem level to the execution level. Slatter and Armoni (2016) implemented Armoni's (2013) framework to study abstraction development of 119 grade 7 students programming in Scratch. They divided the students in groups, finding that the students that produced oral and written descriptions of their algorithms performed better in abstraction assessments. Interestingly, in general, girls experienced greater development in abstraction than boys. Slatter and Armoni's (2016) findings align with Cutts et al.'s (2012), suggesting that moving between different LOA and awareness of LOA prepares students to become competent programmers. Waite et al. (2016) proposed a similar LOA framework for problem solving in programming projects in K-12 classroom activities that involved design in some way. Drawing on Taub et al.'s view (2014) that in CS learning, students move from the higher level of what simulations should do to the lower level of how it is done, they developed the following LOA:

(1) Problem: English, What it is wanted.

(2) Algorithm: CS talk, What it should do.

(3) Program: How it is done.

(4) Runtime.

Waite et al. (2016) suggested using three instruments to help students transition between LOA and thus progress in CP, i.e., labeled diagrams, concept maps and storyboards. Labeled diagrams would provide students with the opportunity to highlight the most important features of objects, processes, or systems; Concept maps would help visualize their abstractions and compare their ideas with those of other students; and Story boards would potentially impact the progression of abstraction by including in them what should be included and ignored. Waite et al. suggested the need for further research to determine the potential of such approaches. Waite et al. (2018), who focused on talk at different LOA, incorporated Waite et al. 's (2016) design component and previous research to their LOA framework for K-5 classroom activities. It is displayed in Table 4 below.

Table 4

LOAs Identified in Previous Research

LOA	Explanation of each LOA	Abstraction transition taxonomy (Cutts et al., 2012)	Abstraction truth simulation (Taub et al., 2012)
Problem	A short written or verbal description of a project	English	What is needed
Design	More detailed than the problem, but without referring to the code. It is a thought, written verbal or drawn depiction of the project.	CS Speak	What it should do
Code	The code itself or a description of the code using programming language specific vocabulary.	Code	How it is done
Running the code	Either the code running or any reference to the output of the program, "When I run the code, the variable score went from 0 to 1".	Results	What it does

Adapted from Waite et al. (2018)

In their study, Waite et al. concluded that a focus on LOA hierarchy may improve teaching/learning CP.

The above studies suggest the importance of focusing on LOA to support student development of abstraction and CT-CP especially when students are aware of the level they are working at and are able to move between LOA. In some studies, e.g, Cutts et al. (2012) and

Waite et al. (2018), as recommended by Grover and Pea (2013), special consideration was given to the promotion of student talk and reflection on the different LOA involved in CT-CP/CS. However, LOA has been typically used as a hierarchy to enable teachers and students' descriptions of the different levels they work at in CT-CP projects, rather than as a teaching methodology (Waite et al., 2018). In the following section, I focus on two relevant educational approaches that, while maintaining focus on LOA, have identified ways to tackle CT-CP complexity and proposed CT-CP teaching methodological strategies.

A Focus on Bridging CT-CP Complexity Progressively

We have seen that abstraction in mathematics can be viewd as a process and so can concepts and the development of creativity. We have also seen that CT-CP/CS involves the use of abstract representations, which are typically used in modeling objects and systems, and in hiding information at different LOA. CT-CP/CS learning has been explored as a process too, which enables its study and the design of relevant pedagogies. In this section I review two studies, i.e., Lee at al. (2011) and Alexandron et al. (2014) and more recent perspectives which have focused CT-CP/CS as a process and have helped grasp and break down CT-CP/CS teaching/learning complexity.

Testing and modifying computer programs engages students in cognitive cycles in increasingly complex scenarios (NRC, 2011). Based on observations conducted both in and out of school, Lee et al. (2011) proposed a framework for the development of CT-CP that helps design and explore CT-CP teaching/learning and describe cognitive practical activity in CT-CP. The framework structures CT-CP progressions in three stages: Use, modify and create. Lee et al. (2011) focused abstraction, automation, and analysis to explore how CT-CP develops across the three stages of the framework in different domains such as modeling and game design and development. The students first use and test a ready-made program to check its affordances by for instance playing a ready-made computer game. Over time, they may want to modify, for example, the color of a character. According to Lee et al., modifications require some understanding of the abstractions and automations contained in a program, model, or game. As the students develop confidence in modifying ready-made programs, they can be encouraged to create their own designs and programs. In the creation phase, the students will use and develop, according to Lee et al, the three key aspects of CT-CP, i.e., abstraction, automation, and analysis. Lee et al conclude by stating that their framework can be used by teachers and researchers alike stressing the importance of testing and debugging in analysis and of engaging the students in increasingly complex tasks to increase the ownership of their learning.

CT-CP researchers Panoff, Allan, Erickson and Denner suggested applying the use-modify-create continuum “over and over again to develop and examine student learning of CT... By iterating on this pattern, the student gains progressively more capabilities in the area of computational thinking” (NRC, 2011, p. 25). Hoppe and Vanderburg (2019) advocated for students creation of computational artifacts and CT development through use-modify-create progressions. Denner argues that storytelling aligns well with use-modify-create progressions where students can learn CT concepts such as iteration and conditionality (NRC, 2011).

Alexandron et al. (2014) investigated ways to reduce the complexity of CT-CP in education domains with 12th grade high school students and also with graduate students. Some of their findings go beyond the scope of this study but some others, in addition to the theoretical perspectives used, are relevant to this dissertation. Alexandron et al. view abstraction in terms of what and how. The what refers to the information encapsulated in a given symbol used in a particular procedure, the how refers to the different levels where information is stored, which range from the problem level to the solution level. In CT-CP practices, students need to move from the problem (or task) domain to the solution domain, for which they necessarily need to move between levels of abstraction. Debugging and testing CT-CP practices involve moving between LOA, since students compare the behavior of the program with what the students would want it to do.

Alexandron et al. pose the view that in teaching/learning CP it is better to *progress from the simple to the more complex*; start with simplicity in order to be able to see progress and leave complexity for later. Alexandron et al. suggest that using program functionalities without thinking much about them reduces the student cognitive load. The idea is consistent with Haberman & Ben-David Kolikant (2001) who proposed using the blackbox approach to introduce basic CP concepts to novices. Alexandron et al. defined a *Black box* as follows: "Working with the interface of a functional unit without dealing with its internal implementation" (p. 316). The rationale behind the use of functionalities and blackboxes is simply, use functions and concepts, explanations will happen later. This approach, according to Alexandron et al., reduces complexity and *students can learn about programming without having to understand functions and concepts in depth.* According to Alexandron et al., the content of the blackbox, an abstraction in itself, can be revealed before, during or after its use depending on the teaching approach used. Alexandron et al. used Green's (1989) concept of *abstraction gradient* to refer to the minimum and maximum LOA and to readiness or desire to deal with new abstractions. Another useful concept used by Alexandron et al. is the "closeness of mapping" which refers to the distance between the problem domain and the solution domain. The closer the distance, the easier the problem-solving ought to be (p. 311).

Collaboration

Collaboration is a theme that can be found across the CT-CP literature. It has been found that collaboration has a positive effect on the development of C. T (Chowdhury et al., 2018; Wu., 2019). Programming in pairs has been reported to be highly motivating and to foster programming activity and housekeeping tasks such as saving and testing code (NRC, 2011; Werner et al., 2015). Project-based learning is a pedagogy that centers on the view that students learn by developing products, typically in collaboration. This pedagogy has been used effectively in CP (Peng et al., 2019), allowing students to connect abstract knowledge with the real-life artifacts that they produce (Blumenfeld et al., 1991). Humans have a natural predisposition to collaborate (Tomasello, 2014) and to communicate (Tomasello, 2003; Tomasello & Razocky, 2003). As already mentioned, computers are unforgiving; communication with computers is based on “truth”, that is, absolute precision is a strict condition in order for computers to process information. Sfard (2000) argues that “Mathematical objects arise out of the needs of communication instead of being primary to communication” (emphasis in the original) … “the need to create new objects in order to communicate may not be specific to mathematics” (p. 323).

Communication with Computers

The object of study of computer scientists and computer programmers is information (NRC, 2010). According to the US National Research Council, computer scientists and computer programmers focus on the ways of representing and processing information. Basically, computers are powerful machines able to store and process structured data with incredible speed. Computers do not think, they process logical instructions organized in programs written in computer languages (Bourke, 2018). In comparing computer languages and human natural languages, Bourke explains that human and computer languages are very similar in that they both have syntax rules, arguing that computer variables are comparable to nouns and functions to verbs. Like English, Bourke argues, programming languages are written left to right and top to bottom, typically with one *executable* instruction or command per line, though. However, there is one fundamental difference between English and programming languages. While English is quite forgiving, allowing its speakers to understand 'broken' messages, computers cannot process information containing the smallest error or information that is not logical, unambiguous, and perfectly organized. According to Bourke, programmers need to represent and structure information in very specific ways if they want computers to process the information needed to fulfill tasks, solve problems, and create products. Programmers need to input the information in a particular computer programming language, such as Python, formulate it in the form of a repetitively applied unambiguous procedure, that is, an *algorithm,* and structure the information in specific ways (data structures); otherwise, the computer will not be able to process the information and the task will not be fulfilled the product created or the problem solved. Thus, algorithms organized in structures are called computer programs. The US National Research Council (NRC, 2010) provides the following two relevant definitions: 1) "A computer program expresses algorithms and structures information using a programming language" (p. 49); 2) "Algorithm embodies the notion of a precisely formulated unambiguous procedure that is repetitively applied" (p. 9). Mathematics and CT-CP/CS make a perfect fit in that mathematics can provide the kind of unambiguous representations that computer scientists and computer programmers require to communicate with computers (Reeves & Clarke, 1990)

Mathematics as Tool

Mathematics methods and affordances have been recognized as fundamental to CT-CP/CS (Baldwin et al., 2013; Hu, 2011; Perrenet & Kaasenbrood, 2005; Rich et al., 2019). Hardwares and softwares normally function in propositional logic, a kind of logic that basically functions in terms of the truth of statements (Ben-Ari, 2012). This kind of "truth," one that is precise and unambiguous, can be provided by mathematics symbolic abstractions. This is of critical importance because when a programmer inputs information without exquisite organization and precision, that is, with the smallest error in their program, a debugging process is required to identify the error and fix it, else the computer will not process the information and execute the instructions input in the program (Bourke, 2018). To fix the error, the programmer needs to check their program and/or the algorithm that it contains to make sure that they are flawless and executable, in which case, they have the possibility to check the output and see if it meets their expectations.

Basics of Programming Digital Images and Video in Python

A color digital image is data (information), it is nothing more than a collection of discrete picture "dots" called pixels (Feinberg, 2021). According to Feinberg, in order to program digital images, the programmer needs to organize with absolute precision the location and color information that will yield color pixels and images. In Python, this can be done by means of two-dimensional arrays which can organize information which, once processed by the computer, will yield corresponding pixels organized in two dimensional digital grids. Two dimensional arrays are the kind of data structure required to organize information, allowing to store color information at specific locations. To represent color information with precision, hexadecimal numbers are used, which allow coding colors between more than 16 million shades. To designate locations, Y and X values are used.

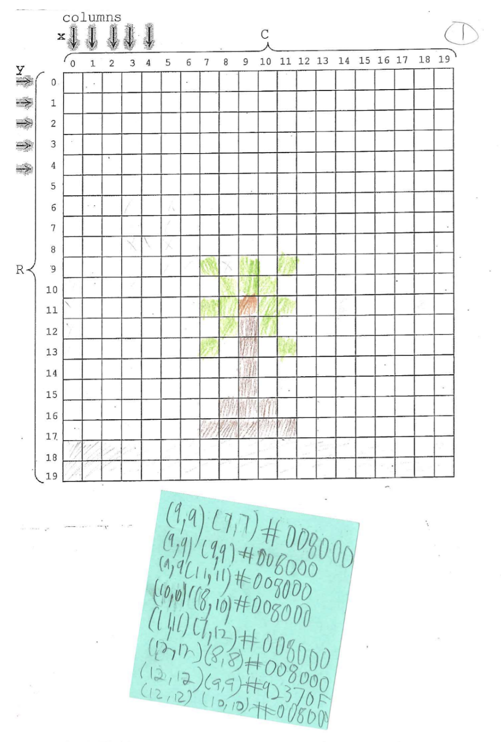
Thus, as explained by Feinberg, a Python digital image has a width and a height specified by rows and columns (20 by 20) that correspond to a Python matrix composed by Y and X values (from 0 to 19). Y and X values are organized in arrays which also support hexadecimal numbers that give colors to said specific locations. For instance, to create the Python program that yielded the the digital pixels of the tree of the background of the video, the participant students of this dissertation used instructions formulated in lines such as the following:

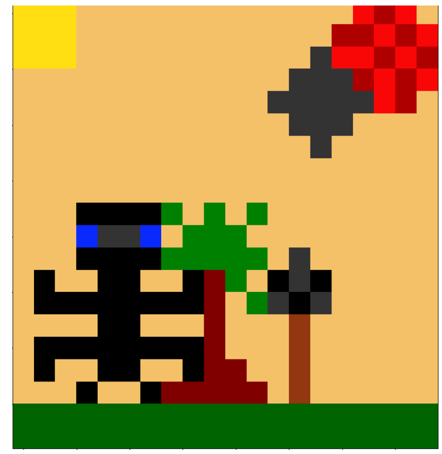
```
im_fill(frame1,[9,9],[7,7],"008000”).
```

This instruction told the computer by means of the Python function im_fill to work on frame1 of the video (video scene number 1) and fill the pixel located at [9,9], [7,7] with the specific color 008000. Figure 2 shows the tree prototype that was designed by the students (to the left) and the resulting image that was enabled by the Python program created with instructions such as the one shown above.

Figure 2

Tree Prototype, Mathematization, and Resulting Frame1 Digital Image





Mathematizing the tree with the help of a squared paper grid yielded the [9,9],[7,7] location of color 008000 (in hexadecimal notation), the green color of the tree crown that can be observed in the digital image (Note the little square drawn in green pencil at row 9 (Y value), column 7 (X value).

Python functions such as im_fill "call" or invoke the arguments inside the parenthesis that are placed in front of them (Bourke, 2018). Bourque explains that functions are available in libraries and used to perform input output operations. Other relevant Python functions are im_show and fps (frames per second) which, if called, enable respectively the display of a digital image and the speed at which images are displayed. In reality, according to Bourke, functions encapsulate sequences of instructions whose details are hidden which facilitates procedural abstraction. That is, the concept that "a procedure or sequence of operations can be encapsulated into one logical unit (function, subroutine, etc.) so that a user need not concern themselves with the low-level details of how it operates" (p. 594). In other words, functions comprise Python code which is hidden. The instruction line shown yielded one of the green pixels that composed the crown of the digital tree that the students created. When the students tested their Python program in progress, what they were actually doing was using the function im_fill to call specific pixels on the computer screen to reflect a specific shape and color for their tree. Each detail of the program the students were developing was key, the system is unforgiving. If one misses one parenthesis the instruction transmitted to the computer with that line of code just won't be processed. For instance, the parenthesis used in the line of the student Python program shown took care of executing the instruction that was given to the computer by means of that specific line, in this case, to fill with green color one specific pixel of a tree image in progress.

In developing their Python program, eleven similar instructions were input by the students for the computer to create the green and brown tree of the background of their video, which can be seen in Figure 2. The students proceeded similarly to create the whole Python program that yielded the rest of the digital elements of the background and all the video characters of each of the five video frames of their video. In order to develop their Python program for the video, the students were thinking in a specific way, they were thinking computationally as will be demonstrated in Chapter 5. This way of thinking allowed them to formulate their task in a way that the computer could effectively carry out (Grover & Pea, 2018) (the set of instructions included in their Python program such as the above shown

```
im_fill(frame1,[9,9],[7,7],"008000”)
```

Methodologies and Methods Previously Used to Explore Computational Thinking (CT)

As already noted, this is the first study to examine a whole course of CT- CP discursively (as it happens during teaching/learning practices) with fine-grained linguistic methods of analysis—such as the ones SFL provides (Morgan & Sfard, 2016). Grover &Pea (2013) advocated for intensive discursive approaches to CT-CP research. However, the CT-CP literature that has centered on discourse is very scarce. The reason might be that analyzing students' processes of design and program development requires considerable effort (Kong, 2019). Roig-Vila and Moreno-Isac (2020) did not identify any discourse analysis of a whole CT-CP course. Their systematic bibliometric research on computational thinking drew on the Web of Science database. More specifically, they explored "its principal collection: *Citation Index Expanded* (SCI- EXPANDED), *Social Science Citation Index* (SSCI), *Arts and Humanities Citation Index* (A-HCI), *Conference Proceedings Citation Index-Science* (CPIC-S), *Conference Proceedings Citation Index- Social Science and Humanities* (CPCI-SSH), *Book Citation Index-Science* (BKCI-S), *Book Citation Index-Social Science and Humanities* (BKCI-SSH)" (emphasis in the original) (p. 6) where, according to the authors, the principal and more impactful science and social science journals can be found. Roig-Vila and Moreno-Isac analyzed all the articles and proceeding papers regarding CT educational research until 2018. They found a first study dated in 1979 and CT educational research experiencing an upward trend which is more evident since 2013. Publications from more than 40 countries were identified. About 40% of the publications on CT education were theoretical and 60% empirical. Within the empirical investigations found that used a qualitative approach, most of them were based on questionnaires, interviews or observations. Roig-Vila and Moreno-Isac did not report on any CT-CP educational research that had carried out a discourse analysis of a whole CT-CP introductory course at any educational level. To date, the most common qualitative research methodology adopted to research CT is case study (Kalelioğlu, 2018). Kong conducted a comprehensive literature review aimed at identifying every study regarding CT evaluation and programming. The goal was to identify CT concepts, practices, and perspectives (Brennan & Resnick, 2012) and the methods of CT evaluation for young learners. The main methods used in CT research according to Kong are gathered in Table 5.

Table 5

Methods Used to Identify and Evaluate CT

Method	Study	Frequency
1. Task/project portfolio/rubrics	Denner et al. (2014), Rodriguez et al. (2017), Román-González et al. (2017), Seiter & Foreman (2013), Sherman & Martin (2015), Werner et al. (2012), Zhong et al. (2016)	7
2. Tests designed with task-based questions	Duncan & Bell (2015), Gouws et al. (2013), Grover et al. (2014, 2015)	4
3. Interviews	Brennan &Resnick (2012), Grover et al. (2014, 2015), Mueller et al. (2017)	4
4. Observations	Burke (2012), Fessakis et al. (2013), Grover & Pea (2018)	3
5. Reflection reports	Zhong et al. (2016)	1
5. Previous research	Grover & Pea (2018)	1

Adapted from Kong (2019)

Kong concluded that, while quantitative methods are best suited to conduct research on large numbers of CT-CP learners, qualitative methods such as task-based interviews, observations, and reflection reports for in-depth evaluations are more convenient with small groups of learners.

As for the methodology used in these studies to gather CT-CP concepts and practices, Brennan & Resnick's (2012) relied on project-portfolio analysis, artifact-based interviews and design scenarios. Grover & Pea (2018) assert that their concepts and practices were based on previous research, i.e., Wing (2006) (a commentary) and Grover & Pea (2013) (a literature review) and on observations they made on computer scientists, with no further explanation on the methodology used in these observations. In another extensively cited study, Barr & Stephenson (2011) relied on the opinions of a combined Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) committee to gather CT main concepts and capabilities.

Conclusion

The review of literature explored suggests further examination of real CT-CP educational environments with powerful resources such as SFL. The objective is twofold: to further understanding of the nature of CT-CP, and to inform CT-CP pedagogies in bridging CT-CP complexity and facilitating CT-CP practices to all students, especially CLD students. Herbel-Eisenmann et al. (2013) examined teacher discourse and student positioning in mathematics educational research arguing for the importance of maintaining a dual focus that allows examination of both what constitutes doing mathematics and how teachers and students are positioned to participate in mathematics education. In this study, I extend Herbel-Eisenmann et al.'s suggestion to CT-CP. SFL affords the tools for the examination of both matters.

Chapter 3

Theoretical Framework

To frame this study, I first used a sociocultural perspective (Lave & Wenger, 1991; Lavie et al., 2018; Lavie & Sfard, 2019; Mahn, 2015; Sfard, 2008; Vygotsky, 1978; 1986; Wertsch, 2007) to define CT-CP learning within an overarching ethical perspective based on students' pedagogical rights (Bernstein, 2000). I defined CT-CP learning as social and collaborative meaning-making processes within the students' Zone of Proximal Development (ZPD) in which students can experience a variety of epistemologies (Edwards, 2021) and develop a specific discourse (Lavie et al., 2018; Lavie & Sfard, 2019; Sfard 2008; Sfard, 2001b) such as CT-CP discourse through their participation in well-established CT-CP practices (Grover & Pea, 2018; Kong, 2019). I defined CT-CP discourse as a functional, social, and institutional form of communication that concerns participation in meaning-making processes in CT-CP environments and realization of CT-CP contexts and practices thus thinking and communicating towards the formulation of a task or problem in a way that a computer can effectively carry out. The CT-CP educational practices under examination were understood from a linguistic point of view, that is, as CT-CP educational oral discourse.

From a SFL perspective, educational discourse is described as a social practice aimed at negotiation of meaning in context (Christie, 2005). Accordingly, I interpreted CT-CP educational discourse as a social and specific kind of discourse that construes meanings aimed at the formulation of a task or problem in a way that a computer can effectively carry out while enacting roles during CT-CP curricular activity. Thus (second), I used SFL (Halliday & Matthiessen, 2014; Schleppegrell, 2012) to understand the discourse of a facilitator in what respects both the nature CT-CP practices (aimed the formulation of a task in a way that a computer can effectively carry out) that she offered the students and the positions or opportunities to learn that she opened for them (Herbel-Eisenmann et al., 2013). To enhance SFL affordances to study the nature of the CT-CP universe offered by the facilitator through her discourse, I relied on relevant views of pedagogical discourse (Bernstein 1973, 2003), abstraction processes (Hershkowitz et al., 2001; Kramer, 2007; Sfard 1991; Skemp, 1987; NRC, 2010) and effective communication (Ben-Ari, 2012; Bourke, 2018; Sfard 2000). To enhance SFL affordances to interpret the opportunities to learn offered to the students by the facilitator through her discourse, I relied on the facilitator's use of authentic/test questions (Gamoran & Nystrand, 1992; Nystrand, 1997) and inclusive commands (Herbel-Eisenmann & Wagner 2007; Morgan 2006; Rotman, 1998; Schelleppegrell, 2012). Third, to contextualize the CT-CP discourse used by the participants I used CoP theory (Farnsworth, 2016; Lave & Wenger, 1991; Wenger, 1998, 2010; Wenger-Trayner & Wenger-Trayner, 2015) and the reifications of student thinking (Bholah, 2017; Wenger, 2010) or prototypes and computational artifacts (Hoppe & Werneburg, 2019) produced by the students.

An Ethical and Sociocultural Perspective on Learning (RQ3)

I interpreted that CT-CP learning is a social and collaborative meaning-making activity mediated through discourse within the ZPD (Mahn, 2015; Vygotsky, 1978; 1986; Wertsch, 2007). In my interpretations, I held an overarching ethical perspective based on Bernstein's (2003) pedagogical rights and the belief that all humans can learn through the appropriate pedagogies and resources. Bernstein's pedagogical rights comprise the right of students to be introduced to possible futures, their right to be included, as opposed to absorbed, personally, socially, and intellectually, and their right to participate actively in relevant practices whereby "order is constructed, maintained and changed" (p. xx). This lens guided the theoretical and methodological decisions I adopted which revolved around exploring whether and how the facilitator opened opportunities to learn creating collaborative ZPDs. In other words, I kept an ethical perspective in exploring whether the facilitator opened opportunities for students' agency (Wenger, 2010), collaboration and bilingual talk in the practices under focus (Moll & Whitmore, 1993), student contribution of ideas and how their ideas were transformed through instruction (Gonzalez et al., 2001).

CT-CP Learning as development of CT-CP Discourse

I viewed CT-CP learning as the development of CT-CP discourse which I understood as an embodiment of culture (Halliday & Hasan, 1979) inextricably linked to the activity which it enables and to the social context (Halliday, 1978; Halliday & Matthiessen, 2014; Morgan, 2012). Accordingly, following Lavie et al. (2018), Lavie and Sfard (2019), and Sfard (2001b; 2008), I interpreted that student development of CT-CP discourse "in a certain well-defined manner" Sfard (2001b, p. 3) and agentive participation in product-oriented, well-established CT-CP practices and procedures signaled CT-CP learning. Thus, students' agentive participation in CT-CP modeling, modularization, algorithm thinking, testing, evaluating, and refining programs and creativity (Grover & Pea, 2018; Kong, 2019) signaled CT-CP learning. Therefore, I considered whether there were signs in the students' discourse of the following constructs:

- *flexibility* (was there more than one way to perform a task?), *bondedness* (did the steps of a procedure adopted by the students towards obtaining a final product feed into later steps?),

- *applicability* (did the students apply what they learned to different environments?),
- *agentivity,* which I identified in student use of commands and of the pronoun I (Halliday & Matthiessen, 2014) (did the students make decisions in response to their own needs),
- *objectification* (did the students substitute objects of the real-world with abstract concepts such as integers?) and,
- *substantiation* (were they able to explain what they did?)

Finally, signs in the students' discourse of CT-CP keywords, visual mediators and endorsed narratives such as CP functions, prototypes and mathematical "truth statements" respectively also signaled CT-CP learning.

CT-CP Learning as a Progression

I conceptualized CT-CP teaching/learning and the various constructs that it comprises as processes. Various research informed this perspective. CT-CP learning has been viewed as a staged progression: use-modify-create (Lee et al., 2011) where students use ready-made programs to then modify them and analyze changes, and finally create their own programs. CT-CP has also been deeply related to systematic heuristic procedures (Papert, 1980). Also, my study was informed by research that has conceptualized abstractions and concepts as processes (e.g., Hershkowitz et al., 2001; Skemp, 1987) and by research that has identified LOA as an entry point to CT-CP learning (e.g., Cutts et al., 2012). I viewed CT-CP constructs such as for loops, conditional statements, variables, modeling, creativity, algorithms, algorithm thinking, testing, debugging and computational artifacts as processes with a history. In interpreting these processes, I targeted CT-CP concepts specifically to explore how they were dealt with and whether conceptual, instrumental, operational, or structural discussions (Sfard, 2001a; Skemp, 1987) were at stake.

Explicit and Implicit Mediation

Additionally, I considered that CT-CP learning can be mediated by facilitators both explicitly and implicitly (Wertsch, 2007). Explicit mediation refers to teaching practices aimed at student conceptual development through conceptual discussions and conscious reflection of CT-CP concepts. Implicit mediation, rather than focusing on discussion of CT-CP concepts, involves everyday common discourse which is less easily taken as objects for conscious reflection.

CT-CP Learning as Reconfigurations of Student Epistemic Frames

Finally, I viewed learning as a reconfiguration of student epistemic frames or ways of knowing (Willensky & Papert, 2010) which I assumed were significant, given that they were novices in CT-CP practices. I interpreted that producing their own computer technology (i.e., digital images in Python from the pixel level) meant a reconfiguration of the students' ways of knowing digital images and video games.

An SFL Perspective on CT-CP Educational Discourse (RQs 1 and 2)

Following SFL, I interpreted that CT-CP educational discourse originates in the whole language system which is organized by its grammar, which, far from being conceptualized as a system of rules, is considered a functional system of choice (Halliday & Matthiessen, 2014). In SFL, language is a resource that participants use to make meaning in context, be it in English (Eggins, 1994; Halliday & Matthiessen, 2014; Schleppegrell, 2012) or in Spanish (García, 2013; Lavid et al., 2010). To examine the discourse realized by the participants of this study I used

SFL's main units of analysis, i.e., clause and text. A clause is a group of words that are organized to construe activity around people, things, and optionally, circumstances such as time, place, manner, and purpose (Martin & Rose, 2007). All clauses have one verb (only one) with the function to realize a process, one or more nouns or pronouns to involve *participants* in said processes, and optional adjectives, adverbs, and other elements to express characteristics of participants and to express the attendant circumstances of processes (Halliday & Matthiessen, 2014). The clauses and functional meanings that participants realize in context as they talk are intertwined to produce discourse or 'operational text' (Halliday, 1993) which is both a process and a product that can be analyzed (Halliday & Matthiessen, 2014). Following Martin and Rose (2003), I was able to interpret spoken texts as follows:

1) Instances of participant CT-CP interactions

2) Instances of a specific culture, that is AOLME CT-CP CoP

In practice, to get at the findings of this dissertation, I selected 15 significant texts which I analyzed as will be explained in the following chapter. In close correlation with my RQs, I adopted the fundamental SFL view that the meanings encoded in the clauses and discourse used by the participants of this study *at once represented a particular CT-CP experiential domain or universe and positioned interactants interpersonally either for action or exchanges of information* (Halliday & Matthiessen, 2014). Importantly, in what respects the facilitator, positioning the students for information exchanges and for action meant differentiated opportunities to learn (Herbel-Eisenmann et al., 2013). I will discuss these opportunities later, in the theoretical views that relate to RQ2. First, I will focus on how SFL allowed me to interpret the CT-CP domain represented in the discourse of the facilitator (RQ1).

The CT-CP Experiential Universe Represented in the Discourse of the Facilitator (RQ1)

I used the SFL system of transitivity (Halliday & Matthiessen, 2014) to interpret the experiential CT-CP domain represented in the discourse of the facilitator thus the nature of the experiential CT-CP domain or universe which she offered her students. The transitivity system organizes experience in processes (verbs) unfolding through time, the *participants* involved in processes and the circumstances associated with the processes. The participant facilitator used in each clause a specific *process* and involved one or several *participants* (e.g., you, asteroid, integer, Michael), associating with the process (not with all processes) *circumstances* such as time, space, manner, purpose, cause, and conditionality. Patterns of these three variables can be used to understand and describe experiential domains such as the CT-CP domain that I examined in this dissertation.

Participants **and agencies**

In the examination of the CT-CP universe offered to the participant CLD students by the facilitator, I considered who and what the facilitator construed as agentive (Halliday & Matthiessen, 2014; Shleppegrell, 2012) and if she obscured agencies by means of the passive voice (Morgan & Sfard, 2016). To further understand the nature of the *participants* involved in activity consider their qualifiers (typically adjectives) and adapted Bernstein's (1973) categorizations of pedagogical discourse to come up with four categories: imaginative & personal worlds, the world of mathematics, the world of CP and the world of image & video processing. This view helped me explore and understand students' contributions to the discourse of the facilitator. Given that the students were novices in computer programming and image and video processing practices, I expected their contributions to proceed mostly from their imaginative & personal worlds and mathematics.

Processes

To examine the CT-CP universe offered to the participant CLD students by the facilitator, I used the system or model of process type which allows categorization of human experience based on the processes realized discursively (Halliday & Matthiessen, 2014). The model organizes experience in three domains:

1) The physical world (doing),

2) The world of consciousness (sensing),

3) The world of abstract relations (being).

The main processes of these three worlds are *material, mental* and *relational* (see Table 6 below). The model helped me explore whether the processes used by the facilitator involved the outer (e.g., doing) or inner worlds (e.g., wanting). I also could explore the domain of abstract relationships, which pertains to clauses devoted to relational processes used to attribute properties to things (and people), to identify and symbolize, which are so often used in Mathematics (Veel, 1999).

Table 6

Main Process Types Available in the Grammar

	The physical world (doing)	The world of consciousness (sensing)	The world of abstract relations (being)
Main process types	Material processes	Mental processes	Relational processes

Prototypical members of process type	Doing, creating, changing, happening (being created)	Thinking, wanting, seing, feeling	Having attribute, having identity, symbolyzing (using symbols)

Material processes can be subdivided into processes of doing and processes of happening (e.g., the pressure formed rocks vs rocks formed). Similarly, mental processes are subdivided in cognitive (eg., think), desiderative (e.g., want) perceptive (eg. see) and emotive (e.g., enjoy). Relational processes are subdivided in identifying (eg. Sarah is the leader), attributive (eg. she is certain) and symbolizing (e.g., every fourth African is a Nigerian). There are three more categories: *behavioral*, which represent typical human behaviors such as laughing and sleeping, *verbal,* i.e, processes such as saying and explaining; and *existential,* i.e, processes that express "to be" such as exist (Halliday & Matthiessen, 2014). Analysis of patterns of processes allowed my interpretations of the Ct-CP experiential domain offered by the facilitator.

Circumstances

The circumstances associated with processes helped me understand and describe activity. Prepositions, conjunctions, and other resources are used to mean, for instance, location (e.g., "in"), means (eg., "with"), cause—which includes reason (e.g. "because"), and purpose (e.g., "to"), contingency—which includes condition (e.g., "if"), and negative condition (e.g., "unless) (Halliday & Matthiessen, 2014).

To organize the discursive flow, the grammar of the language provides resources such as ellipsis (the omission of elements that were already used by interactants), substitution (e.g., pronouns) and conjunctions to make logical connections between clauses. Clauses connect to form clause complexes with connectors such as "if" and "in order to". A repeated use of the conjunction “if” gives “flavor” of conditionality; a repeated use of conjunction “to” or connector “in order to” represents purpose.

Modal verbs

Another relevant resource that the grammar provided is modal verbs. According to Halliday &Matthiessen (2014), modal verbs allow speakers to modulate their assessment of the likelihood of some information to be certain and the desirability of an action. Facilitators‘ use of modal verbs (e.g., have to, going to, need to, can, should, could…) affect how a discipline is positioned with respect to the students; whether as something fixed and authoritative with no space for student agency or as something flexible in which they can participate agentively (Schleppegrell, 2012). I scrutinized forms such as “need to” and “have to” to identify in what respects the facilitators depicted CT-CP as an authoritative discipline, reducing the space for student agency; and forms such as “can” and “could” to identify flexibility, thus options for students to act. Table 7 gathers modal verbs according to their value. A higher value depicts an authoritative discipline with conditions that must be followed whereas a lower value depicts flexibility.

Table 7

Modal Verbs and Their Value

Modal verb value	Verbal form
High value	must, ought to, need to, have to,
Median value	will, would, should, going to
Low value	may, might, can, could

In sum, to interpret the nature of the specific CT-CP domain that the facilitator offered her small group of CLD students within the AOLME CT-CP CoP I explored configurations of *participants*, process types, *and* circumstances represented in the discourse of the facilitator and her use of modal verbs to represent conditionality and flexibility. The verbs chosen by a CT-CP facilitator in each clause represents a particular CT-CP domain, for instance, as fixed and as a flexible domain; as place where things happen and as a CT-CP domain where her students act (Mizuno et al., 2018). Similarly, a facilitator can involve students in her clauses, and she can involve concepts while she can choose who or what to construe as agentive in her clauses (Schleppegrell, 2012), and even obscure agencies by means of the passive voice (Morgan & Sfard, 2016). In addition, processes can be linked logically by means of conjunctions which concede particular 'flavor' to discourse such as purpose and condition. Close examination of sets of clauses (text) can yield patterns of all these things and conclusions can be drawn about the nature of a particular domain of experience such as the CT-CP domain of a given CoP.

***Opportunities for CT-CP Learning in the Discourse of the Facilitator: Positioning Students* (RQ2)**

Following SFL, I interpreted that the discourse of the facilitator positioned the students either for exchanges of information or goods and services, that is, action (Halliday, 1984; Halliday & Matthiessen, 2014). I adopted the view that the facilitator's discourse can open for students opportunities to learn, or not so much (Herbel-Eisenmann et al., 2013). Accordingly, I used the notion of inclusive commands (Rotman, 1998) which has been used in mathematics educational research (e.g., Herbel-Eisenmann & Wagner 2007; Morgan 2006; Rotman, 1998; Schleppegrell, 2012) to refer to teaching moves that position students as thinkers and capable members of a mathematics teaching/learning community, thus opening for them opportunities to learn. I extended this view to this dissertation, interpreting that positioning students through commands to be active in CT-CP practices positioned them as thinkers, thus, capable members of the CT-CP educational community. In addition, I considered the importance for learning of instruction in the language and culture of the students (Celedón Pattichis et al. 2018), focusing on the use of Spanish during CT-CP practices. Finally, I took into consideration in my analysis the question raised by Herbel-Eisenmann and Wagner (2007) of how written texts (I extended their question to facilitator's oral discourse) can position students in relation to their experience of the world. I interpreted that the experience of the world of the students mattered if they could cope with the academic level of the mathematics operations needed in the CT-CP practices and if their world experience was included in curricular activity.

Participants' Dialogic interactions **(RQs 2 and 3)**

I understood that in teaching/learning interactions, the positions and roles adopted by facilitators and students are not fixed but change during curricular activity (Cabral & Baldino, 2002). This perspective strengthened my view of teaching-learning as a two-way process, urging me to search for what both the facilitator and the students contributed to curricular activity. Following SFL (e.g., Halliday & Matthiessen, 2014), I interpreted that by means of the four primary speech functions, i.e., statements, questions, offers and commands[3], the participants took on two fundamental roles: giving and demanding. *The participants realized either exchanges of information called <u>propositions</u> or exchanges of action, called <u>proposals</u>.* Propositions are

[3] Eggins and Slade (1997; 2004) added a fifth speech function: the *check*, typically a yes or an interjection such as "mhm", "aha", which speakers use to clarify or confirm communication.

typically realized by means of questions[4], when interactants take on the role of demanding information, and by means of statements when they take on the role of giving information. In exchanges of 'action' (proposals) typically offers or commands are used. Table 8 gathers propositions and proposals, the fundamental roles that participants can take, i.e., giving and demanding, and the four primary speech functions which enable them.

Table 8

Primary Speech Functions, Propositions, and Proposals

	Commodity exchanged	
Role in the exchange	5) Action (Proposal)	b) Information (Proposition)
1. Giving	*Offer* "Let me help him"	*Statement* "I helped him"
2. Demanding	*Command* "Help him"	*Question* "Did you help him"

Adapted from Halliday & Matthiesen (2014, p. 136) (Examples added)

Hence, during dialogue, participants continuously take on roles in information and action exchanges and in so doing they position each other. In Halliday's (1984) words:

> When the speaker takes on a role of giving or demanding, by the same token he assigns a complementary role to the person he is addressing. If I am giving, you are called on to accept; if I am demanding, you are called on to give. (p. 12).

[4] The speech functions formulated are and were the most typical. Interactants can also, for instance, ask for something indirectly by means of a question or a statement.

Halliday and Matthiessen (2014) stress that proposals and propositions are *invitations* used by speakers of the language in their negotiations of meanings during dialogue. In the act of speaking, offers can be accepted or rejected and commands carried out or rejected; and statements acknowledged or not and questions answered or not.

Dividing the discourse of the participants of this dissertation in proposals and propositions, under the view of teaching-learning as a two-way process, illuminated several relevant explorations and interpretations. It helped me explore and interpret the importance that each participant conceded to information and to action in specific moments of the CT-CP practices at hand. As I will further explain in the next chapter, I was able to explore who led activity with commands and offers at particular moments of curricular activity, who provided information through statements and asked the questions, and what was the content of the interactions.

Authentic/Test Questions and IRE Patterns

To enhance SFL affordances in interpreting the questions used by the facilitator I relied on the facilitator use of authentic/test questions (Gamoran & Nystrand, 1992; Nystrand, 1997). Following SFL, I interpreted that in asking authentic questions, the facilitator positioned the students as providers of information, considering that their information mattered if it was incorporated to the CT-CP practices at hand. I interpreted that also Sinclair and Coulthard's (1975) teaching IRF patterns (Initiation-Response-Feedback) meant restricted invitations for student participation since they typically limit responses to right/wrong results and trigger minimal feedback, as opposed to dialogic teaching, which promotes student participation, collaboration, thinking and learning (Alexander, 2018; Cazden, 2001; Mortimer and Scott, 2003).

A Finer Look on Dialogue: Effective Collaboration, Communication and Thinking

I followed the view that humans have a natural predisposition to collaborate (Tomasello, 2014) and to communicate (Tomasello, 2003; Tomasello & Razocky, 2003). Not surprisingly, it has been found that collaborative work in CT-CP projects fosters much more advanced reasoning and CT development (Chowdhury et al. 2018; NRC, 2011; Wu et al., 2019). To explore CT-CP teaching/ learning, I viewed thinking and communication as tantamount (Sfard, 2001b, 2008) and mathematics as a resource to cope with practical challenges (Lenhard & Carrier, 2015), to communicate with others effectively (Sfard, 2000) and with a computer in a precise and logical way (Ben-Ari, 2012; Reeves & Clarke, 1990). I interpreted that the objects of communication and thinking are the same (Sfard, 2008) and that to collaborate, humans coordinate communication and thinking to achieve common goals (Tomasello, 2003). I held Sfard's (2000) perspective that collaborative activity is made possible through common will and verbalization of public intentions around a shared attended focus. Given a common intention, the effectiveness of communication is a factor of the quality of the discursive focus exchanged between interactants. In Sfard's words:

> Effectiveness may be presented as dependent on the degree of clarity of the discursive focus: The communication will not be regarded as effective unless, at any given moment, all the participants seem to know what they are talking about and feel confident that all the parties involved refer to the same things when using the same words. (p. 303).

According to Sfard, ambiguity is a serious weakness of communications since it compromises the quality of focus. Sfard described discursive focus as a triple comprising what is pursued in the communication act, what is pronounced and the attended focus which mediates between the two. About communication effectiveness, Sfard argues that it arises out of the need of operative precision. Sfard writes,

> Effectiveness of verbal communication has been presented as a function of the quality of its focus, among other factors. The discursive focus was described as a triple: The pronounced element is public; the intended component is mainly private; and the attended focus mediates between the two. Because of the attended
>
> focus, translating private into public and vice versa, the activity that is the essence of communication becomes possible. (p. 320)

In other words, for communication to be effective, interactants need to have agreed on a common intention, attend to a common focus, and pronounce it unambiguously. The quality of the focus can be made possible with mathematical representations e.g., numbers, which can provide the necessary formalisms both to communicate effectively between participants and with computers. Mathematics formalisms thus, inasmuch as precise and unambiguous are fundamental in CT-CP/CS communications.

Thus, I maintained the perspective that communication with computers is unforgiving since they require input information to be formulated logically, unambiguously and with absolute precision, the kind that mathematics representations can provide. Similarly, I viewed that mathematics can provide precision and unambiguity for effective human-to-human communication.

CoP Theory

SFL provides the necessary theoretical perspectives and methodological tools to study discourse in context which entails a strong consideration of the environment wherein teaching/learning takes place (Alshwaikh & Morgan, 2018). CoP theory provides the theoretical tools to examine teaching/learning environments (e.g., Wenger, 1998, 2010).

SFL-CoP Complementarity: A Focus on the Disciplinary Domain

Wenger (2010) stressed the *relevance of the domain of CoP* as sites of shared interest and development of competence via participation. The domain of a CoP is based on a "shared competence that distinguishes members from other people" (Wenger-Trayner & Wenger-Trayner, 2015).

In this dissertation, I conceptualize a CT-CP CoP domain discursively, as it is realized orally. As explained above, being part of practices includes both 'talking' and 'producing artifacts'. For this reason, as I detail in Chapter 4, I complemented SFL affordances with case study tools. I proceeded this way mostly in order to be able to decipher the discourse used by the participants in light of the artifacts, i.e., the Python code and resulting digital images that the participant students were creating as they interacted.

SFL-CoP Complementarity: Agency

Wenger-Trayner & Wenger-Trayner (2015) affirm that *agency plays a key role in learning* and in the degree with which people identify with practices and the communities where these happen. Close attention to how mutual engagement happens to participate in the joint enterprise and the shared repertoire of a CoP are key in understanding opportunities to learn. These dimensions of practice can be studied in depth with the affordances of SFL, as will be demonstrated.

Following I discuss SFL perspectives and affordances to explore the domain of a CT-CP CoP as presented by a CT-CP facilitator. I also explain SFL affordances to examine agency and interactions closely, thus the resources that it provides to examine facilitators and teachers' opportunities for student participation. SFL allows examination of how facilitators position students (and students self-position themselves) and the roles they adopt during teaching/learning interactions.

Theoretical Perspectives and Constructs Used to Interpret AOLME CT-CP CoP

I conceptualized AOLME, the environment that the participants of this study contributed to constitute and wherein they engaged in CT-CP practices to create videos in Python as a CoP. CT-CP communities of practice are groups of people that get together to practice CT-CP, as obvious as it may seem. CoP theory helps understand practice through its three dimensions and constructs: Joint enterprise, shared repertoire and mutual engagement (Wenger, 1998). The AOLME CT-CP CoP pursued a composite joint enterprise which revolved around teaching/learning CT-CP through the creation of videos in Python to any student that wanted to join, keeping a special focus on serving CLD students. AOLME's shared repertoire included the discourse that was used, the stories that were told, the concepts that were used, the practices that were adopted, the ways of knowing that were enacted, in short, all that AOLME produced as a result of AOLME's CT-CP practice. AOLME's mutual engagement refers to the agreements, disagreements and collaborations that happened, and to the communication that made possible AOLME's CT-CP practice. Below, I will return to a number of views and constructs that helped further interpret AOLME's CT-CP CoP dimensions of practice in what respects the participants inasmuch as AOLME members.

Epistemic Considerations: Interpreting Ways of Knowing in CT-CP Domains

To interpret the participant facilitator invitations for students' ways of knowing during their CT-CP problem solving tasks in the specific CT-CP domain offered by the CT-CP participant facilitator, I drew on several views and constructs that I discuss following. When students enter a new disciplinary domain for the first time, they typically encounter new epistemologies or ways of knowing in such domains (Papert, 1980) which provide what Chalmers (2011) calls epistemic space. I interpreted that the specific CT-CP domain offered by the CT-CP participant facilitator to her small group of CLD students had its own ways of knowing or e*pistemic frames* where *epistemic possibilities* were available to for students to develop practice and their epistemic agencies (Schuster et al., 2018) As a result students' experienced *epistemic restructurations* of the ways of knowing they were used to in similar domains to the one they encountered for the first time (Wilensky & Papert, 2010), e.g., video games. Under these views, I interpreted learning as an expansion of the students' meaning potential (Halliday, 1993) *and as a transfer of epistemic frames* (Edwards, 2021). I viewed the students as experts in their own epistemic spaces, such as the video game Minecraft, and modeling as an *epistemic element* that put different epistemic spaces and frames in communication, i.e ways of knowing in mathematics, art and video games.

To investigate student CT-CP learning, I explored these six characteristics in the history of the CT-CP product-oriented practices that resulted in the student creation of a video in Python code. Lavie et al. (2018) argue that whatever the observer's interpretive method, looking at just one isolated episode is not enough.

> The greater our access to the history of one's action, the more robust our claims on this person's evolving routines become… investigating learning is tantamount to answering the question of how routines emerge and how they later evolve. (pp. 162-163).

Following this idea, in the analysis process, I focused fundamentally on teaching/learning interactions that targeted the procedure that was adopted to produce a video in Python, in which the production of a computational artifact (Arastoopour, 2019; Hoppe & Werneburg, 2019) was critical. In doing so, I focused mostly on the processes that involved modeling with mathematics, abstracting, algorithm thinking, testing and debugging (Grover & Pea, 2018; Kong, 2019) within the use-modify-create framework for CT-CP development (Lee et al., 2011).

Chapter 4

Methodology

Overview

The dynamic relationship between theoretical perspectives and methodological tools has been of the utmost importance in all phases of this research study, and they derive from, but are not limited to, Systemic Functional Linguistics (SFL), as discussed in detail in Chapter 3. In order to better understand (a) the nature of the bilingual CT-CP practices offered by a STEM facilitator to a small group of CLD middle school students (b) how the facilitator positioned the students to learn CT-CP, I combined SFL and case study perspectives and methods. It is important to note that the interdisciplinary nature of the teaching and learning practices of this study required a 'loose analysis' process (Tracy, 2019) open to perspectives, strategies, and tools from various relevant fields, which I will explain next. I present below the RQs, participants[5]

[5] For the sake of simplification, I have opted to use the word participant to refer to the human partakers focus of this investigation (i.e., facilitator, students and the students' language arts teacher) and *participant* in italics to refer, in grammatical terms, to the element of the clause (e.g., asteroid, you) without which it is not possible to realize whatever process (e.g., gets bigger, calculate). It is beyond the scope of this study to discuss the variety of terms used in SFL to refer to the people or things involved in processes, which include, actor, goal, senser, sayer etc. For instance, in the clause "The asteroid gets bigger", asteroid is the agent as opposed to the participants (without italicization) of this study, that is, Teresa, Michael, Herminio, Juan and their language arts teacher. The participants of this study were construed as agentive by the facilitator activity very often, as will be explained, in which cases they were

(and selection criteria), and description of the research sites and data collection protocol, along with a brief review of key SFL perspectives and tools that includes special emphasis on the meaning of "context". In addition, I review the essence of the AOLME Program because its characteristics are key to understanding the results and the implications for education and research (see Chapter 6). I also discuss the analysis process, which I carried out in close consideration of ethical perspectives and broader sociocultural environments. I conclude this chapter with a brief summary. Chapter 4 is organized as follows:

1) RQs
2) Participants
3) Meaning of "Context" in this Study and AOLME
4) An Example of Implementation of the Analytical Framework Adopted
5) Rationale for Using and Coordinating SFL and Case Study Qualitative Methods
6) Data Sources:
    - Primary Data Sources: Transcriptions of video recordings of the practices (i.e., the *whole text*) and the students' prototypes, Python code and resulting digital images and video.
    - Secondary Data Sources: Field notes, student, and facilitator questionnaires, and interviews.
7) Analysis method
    - Units of Analysis and SFL "Toolkits"

---

participants, linguistically speaking. Note that in clauses like "Let us look at the code", "us" is the *participant* in the linguistic sense and also represents participant Teresa and the participant students.

- Exploratory Phase
- Application Phase
- Summary of Chapter Four

RQs

1) What is the nature of the computational thinking-computer programming (CT-CP) domain that a CT-CP facilitator of a CT-CP community of practice (CoP) offered her CLD students by means of oral discourse?
2) How were these students positioned in terms of their participation and role in CT-CP practices?
3) Can the CT-CP pedagogy adopted by the facilitator to teach CT-CP facilitate student learning of CT-CP?

Participants and Participant Selection Criteria

In this section I discuss key details about the participants, all members of the "AOLME team," which I found out primarily through the secondary data sources. The description provided in the present chapter includes key empirical and theoretical information to set the stage for subsequent sections and chapters. The description of the participants is based on the three main theoretical and methodological constructs that informed the selection criteria. All names presented here are pseudonyms. Teresa, the facilitator; Juan, Herminio and Michael, the middle school CLD students; and Yanet, their language arts teacher, are the main participants of this study. In what follows, I present the criteria by which they were selected and explain relevant aspects of their academic, personal, and linguistic profiles.

Equity-oriented, SFL, and case study perspectives were the combined, driving forces that motivated the participant selection decisions. There are also three fundamental constructs to keep in mind: *Bilingual talk, collaboration,* and *agency*. The perspective I maintain in this dissertation, far from setting the focus on comparisons between the linguistic competencies of monolingual English speakers and bilingual English-Spanish speakers (Grosjean, 2012; Moschkovitch, 2010), situates it on the resourceful ways bilingual students navigate different cultural environments (Grosjean, 2013; Zentella, 1997). In addition, this study contributes to research that centers on broadening participation of bilingual CLD students in CT-CP educational and professional fields (Goode et al., 2020). Both considerations about bilingual students and bilingual talk are signaled by Moschkovich (2012) as in intimate relation to equity. Systemic Functional Linguistics originates mostly in the work of Halliday, whose early work (e.g., Halliday, 1974) was presented in Nairobi for the United Nations raising concerns on key issues related to equity and this dissertation such as 'the respective status of the different languages of the community' (p. 15), and all students' 'learning to exchange meanings in social contexts' (p. 10). As already explained, SFL provides tools to study moment-by moment participation and agency of students and facilitators (Morgan 2006; Schleppegrell, 2012). One important reason for me to have used case study perspectives is their affordances to get at the perspectives of the participants with respect to bilingual talk and collaboration. These constructs formed part of the facilitator and students interviews and questionnaires, given that they were constitutive of AOLME foundational principles (See the section on the context of this study). Accordingly, the above-mentioned perspectives and constructs guided the participant selection process from the beginning and were regularly reviewed throughout the research process to ensure their relevance and applicability.

As noted previously, the discourse used by the facilitator, Teresa, will be examined. Teresa's realization and facilitation of *Bilingual talk, collaboration,* and *agency* was already apparent from the very start of the AOLME program. Later, as the analysis progressed, these constructs became apparent both in the primary data and the secondary data.

Participant Selection Criteria #1: Bilingual English-Spanish Talk

The Facilitator: Teresa

Given the bilingual-bicultural nature and goals of AOLME, I decided that it was essential for the facilitator to be a bilingual Spanish-English speaker, given the importance of fostering talk among the small group of students (Chapin et al., 2009; O'Connor & Michaels, 2019). This decision was informed by existing research that shows that multilingual individuals use their languages resourcefully depending on specific domain purposes (e.g., Grosjean & Li, 2013; Moschkovich, 2002) and on research in various STEM fields, which correlates talking and learning (Alexander, 2005; 2008; 2018; Cazden, 2001; Chapin et al., 2009; Mercer & Wegerif, 2004; Pimentel & McNeill, 2013). While she self-identified herself as Latina and bilingual, she also declared feeling more comfortable in English. Teresa emigrated to the United States during childhood. My observations and evaluations coincided: She was fluent both in English and Spanish. From the interview we had in October, 2017, I also gathered that Teresa understood the need for facilitators to adapt to the students' linguistic strengths and challenges. In other words, I wanted a facilitator whose linguistic beliefs and pedagogical approach promoted the CLD students' resourceful use of their two languages, including code switching (Grosjean, 2012).

Doing so would allow the students to communicate more freely and strategically in different contexts (Moschkovich, 2012; Zentella, 1997). Right before the start of the program, on Feb 2, 2017, in one AOLME pre-participation questionnaire, Teresa wrote that using Spanish and/or English helped the students “to learn better” and “to talk,” which further confirmed for me the soundness of the decision to bring her onboard as the facilitator. In this study I refer to her either as Teresa or “the facilitator”.

I proceed now to describe the bilingual English-Spanish language ability of the other participants: the CLD students (Juan, Herminio, and Michael) and their language arts teacher (Yanet). I identified a range of language abilities among these bilingual individuals, basing this conclusion on participant declarations during practice and on my observations of their language use. My professional experience prepared me well for this task: I have had 10 years of experience as a bilingual teacher, including six years teaching in language attrition environments such as New Mexico and two years as a certified Instituto Cervantes Language Proficiency Examiner of Spanish.

The students

Herminio, Juan (of Mexican heritage) and Michael (of Guatemalan heritage) constituted Teresa’s small group of CLD students. Juan is US-born, and Michael and Herminio emigrated to the US during childhood (Michael when he was only one). Herminio (7th grade) English well and with overall ease, although showing slight difficulty at times. He was fully proficient in Spanish (constituting his first language and the language spoken at home), and code-switched with ease, which allowed him to participate in interactions in both languages.

Based on his oral interactions in the videos and my face-to-face observations, I rate Juan (7th grade) as a fully proficient adolescent speaker of English; he showed no apparent difficulty in any context. As for Spanish, he declared that he did not use this language. However, I noticed on many occasions during my observations and viewings of the video recordings that he was able to participate quite naturally when Spanish was used for a given activity. Grosjean (2012) reported that this phenomenon (in which speakers understand Spanish but claim to not be able to speak it or lack confidence in their abilities) is very common in US classrooms. Although proficient orally in both languages, Michael (8th grade) displayed greater fluency in English and seemed a little more comfortable interacting in English than in Spanish. Typically, he initiated conversation in English, but, if addressed in Spanish, he would respond naturally in Spanish.

The Students' Language Arts Teacher: Yanet.

In this section I present the linguistic profiles of the participants whose use of English and Spanish changed as a function of who they interacted with. Yanet was the participant students' language arts teacher. Her oral contributions, fundamentally initiated in Spanish, fostered the use of Spanish and code-switching in the group. Prior to the start of the AOLME program she declared in a questionnaire that she had no previous experience in CP. Cuban born; she is a fully proficient Spanish Speaker. At the time of this study, she was a teacher of Spanish and examiner of Spanish proficiency at the Instituto Cervantes. Yanet often spent time in the laboratory where the AOLME practices took place every Thursday, offering to assist the AOLME students and the rest of the "AOLME team" with anything within her reach, while also attempting to learn some basics about CP. Although fluent in English, I noticed that she rarely used English to interact with those whose first language was Spanish. Her interactive approach, willingness to help and to learn, and her dominant use of Spanish added to the interactions and enriched this investigation.

Table 9

Bilingual English-Spanish Proficiency Profiles

Participant	Speaking	Listening
1. Facilitator (Teresa)	Both	Both
2. Language Arts teacher (Yanet)	Both	Both
3. Hermino	Both	Both
4. Juan	Only English	Unknown
5. Michael	Both	Both

Participant Selection Criteria # 2. Collaboration

I chose collaboration as a selection criterion on the grounds that language can be used by teachers and facilitators to open "interpersonal gateways" to foster student agentive participation and collaboration in STEM practices (Shreyar et al., 2010).

Teresa's data revealed her awareness of the importance of student collaboration in practices such as those in the AOLME program experience. Initially, my field observations (consistent with those of other AOLME researchers) had pointed to Teresa as one of the best spring 2017 facilitators in that she was effective in getting students to be engaged and collaborate in a variety of activities. In our interview in October 2017, Teresa talked about the key role that collaboration played in learning in her group. She declared that many informal group conversations that happened off-camera, such as video game-centered *talk*, helped her students to bond as a group, which in her opinion possibly translated into enhanced *collaboration* during practice. Following Christie (2005), from here on I will use *whole text* to refer to the transcript of all the spoken interactions that the main participants of this study realized during their bilingual computer programming-mathematics-image and video processing practices. The whole text that my participants co-constructed is significantly rich in Teresa's words that index collaboration, such as "together," "we," "us," "help," "group".

The student quotes below were recorded during the last month of their participation in the AOLME program and highlight the role that collaboration played during their computer programming-mathematics-image and video processing practices.

(a) "You work together to understand the computer and you get a stronger understanding." (Michael)

(b) "My small group helped me stay on track." (Juan)

(c) "How we worked in groups." (was positive and productive) (Herminio).

Participant Selection Criteria # 3. Agency

Agency is the third fundamental theoretical and methodological construct that guided the selection of the participants. In this section, I focus the discussion on the facilitator because I had noted salient assertiveness in the personality of the facilitator in my interactions with her. *Agency* is associated with various stakeholders and elements in teaching and learning environments of various fields and has been extensively addressed for instance by Turner et al. (2013), Pinnow & Chval (2015), Morgan (2016) in mathematics education research; and Mejia et al. (2018) in engineering and Sentance et al. (2019) in CT-CP. The present study has strong theoretical and methodological foundations on agency, and I ground its definition in SFL (Halliday & Matthiessen, 2014) and social psychology (Bandura, 2000, 2001), which correlate agency, confidence, and desired outcomes. Agency is a goal-oriented human drive mediated through language and aimed at a *desired outcome* which can be pursued, mostly by means of tools (e.g., a computer or the mathematization of a drawing in this study) individually, in collaboration with another person, or collectively. Halliday and Matthiessen (2014) also correlate agency and desire. These SFL researchers distinguished between exchanges of information (propositions) and actions (proposals[6])—exchanges of goods & services to be more accurate (See Chapter 3). Halliday and Matthiessen further specify that "The difference in the mental processes is that propositions are projected by cognitive processes, whereas proposals are projected by desiderative ones" (p. 548). For Halliday and Matthiessen, speakers' proposals originate from the desires of the agent that produced the proposal. Bandura's (2001) theory about agency adds to our understanding of this construct by connecting confidence with agency and desired outcomes. Teresa herself in a questionnaire response included confidence as a key outcome of STEM education. As one indication of this, I present her response to a question concerning her opinion on what needs to be learned by young students to get college-ready: "I think students need to learn and be aware of STEM fields. Motivation. Confidence" (Feb. 4, about two weeks before meeting the students).

Summary of Participant Selection Criteria and key features

[6] Most propositions and proposals of the whole text were realized with lexicogrammatical congruency. In other words, propositions for the most part were realized through statements in the declarative mood and commands in the imperative mood and proposals through commands in the imperative mood. (See Chapter 3 for discussion on this topic).

The bilingual-bicultural facilitator Teresa, along with her three CLD bilingual-bicultural students and their Bilingual-bicultural language arts teacher, Yanet are the main participants of this study. *Bilingual talk, collaboration,* and *agency* constitute the three key theoretical and methodological constructs, which constituted the criteria for the selection of participants and were observed in the analyses of the secondary data (interviews and questionnaires) and primary data (the whole text of transcribed interactions during practice). The participants (with varying degrees of bilingual proficiency) interacted during practice in both English and Spanish. Even though English was the dominant language, Teresa, seemingly aware of the important relationship between student talk, language choice, and learning processes, encouraged the students to use the language they felt most comfortable with (English, Spanish, or code-switching). In addition, she also emphasized collaboration and confidence as important dimensions of teaching/learning. Her apparent confidence in herself and experience with children are additional traits that, apparently, contributed positively to the teaching/learning dynamics she established. Finally (and unexpectedly), Yanet, the students' language arts teacher became a key participant since she induced sustained use of Spanish and Spanish-English code switching.

The Meaning of "Context" in this Study and AOLME

Following SFL, the participants of this study created functional linguistic contexts by means of the clauses that they used—in SFL called *contexts of situation* (hereafter, I will use situations). However, to refer to AOLME, I use the term *environment or ecosystem.* The situations created by the participants by means of the clauses they used include three dimensions or types of meaning[7]: (a) The experience (and logical meanings) construed by the participants,

[7] In SFL (e.g., Halliday & Matthiessen, 2014), these three dimensions of meaning or metafunctions of the language are called respectively the ideational or experiential (which includes logical meanings), interpersonal, and textual metafunctions of the language. All metafunctions occur in every clause. In this

(i.e., field), (b) the interpersonal relationships and positions they enacted (i.e., tenor), and (c) the cohesion and flow of curricular activity and the role played by language in the total event (i.e., mode) (Halliday & Matthiessen, 2014). In this study, I have focused on the experience and logical meanings construed and the interpersonal relationships enacted by the participants as they relate to the environment where the communication occurred, that is, AOLME. Following CoP Theory, the participants of this study and the rest of AOLME members communicated and collaborated (mutual engagement) in the joint enterprise of CT-CP teaching/learning CT-CP through the creation of videos in Python by means of a specific shared repertoire of practices which include specific CT-CP practices, concepts and ways of knowing.

The physical location where the CT-CP practices took place was one of the science laboratories of a US Southwest rural middle school. After school, every Thursday from February 16th to May 11th of 2017. AOLME implemented an introductory mathematics-based CT-CP curriculum aimed at the creation of videos in Python. I was a Research Assistant for AOLME during 2016-2019 and present there every Thursday, and then in the follow-up meetings that the AOLME team shared every Friday in the university. I gained invaluable insider data for the present study through these observations and the field notes taken over this period, especially during the spring 2017 sessions. This privileged position allowed me to learn about AOLME's work and perspectives, which paved the way for this study and prepared me for the 'observer as participant' role (Merriam & Tisdell, 2015) I performed during the AOLME's Spring 2017 implementation.

The SFL Units of Analysis Used: Clause, Text, and Whole Text

---

dissertation I focused mostly on ideational (which include logical meanings) (RQ 1 and 3) and interpersonal meanings (RQ 2). However, to examine language one necessarily considers the resources used by participants to organize the flow of events (e.g. deixis and ellipsis), which allowed me to examine abstraction.

Clause[8], text, and whole text are the fundamental discursive units of analysis of this dissertation. The meanings that participants realize in each clause as they talk produce text; and text is what interactants engage with and interpret (Halliday & Matthiessen, 2014, p. 3). In this study, I examined the *text* (or discourse) produced by the participants of this study which, according to Halliday and Matthiessen, is at once a process and a product. Martin and Rose (2003) argue the following:

> Since each text is produced interactively between speakers, and between writers and (potential) readers, we can use it to interpret the interaction it manifests. And since each interaction is an instance of the speaker's culture, we can also use the text to interpret aspects of the culture it manifests. (p. 1)

Thus, researchers examine spoken texts as

1) Instances of participant interactions, for instance, CT-CP interactions.

2) Instances of a given culture, for example, instances of a CT-CP CoP such as AOLME.

Texts are processes of meaning-making in context and at the same time, products that researchers can analyze based on the linguistic resources that were used. "The linguistic analysis of a text is not an interpretation of a text; it is an explanation" (Halliday & Hassan, 1976, p. 327). Ultimately, my analysis method aimed at deciphering how a cohesive selection of texts meant, which would allow me to explain what was meant (Eggins, 2004).

Drawing on Halliday, mathematics education researchers Shreyar et al. (2010) argued the following:

---

[8] As already mentioned, a clause is a group of words that are organized to construe activity around people, things, and optionally, circumstances such as time, place, manner and purpose (Martin & Rose, 2007).

> A text can be as short as a single utterance and as long as an entire novel. Texts can be spoken, written, diagrammed, or all of the above. We can think of a text either as process (e.g., a whole-class conversation unfolding in real time) or as product (e.g., the written transcription of that conversation (p. 26)

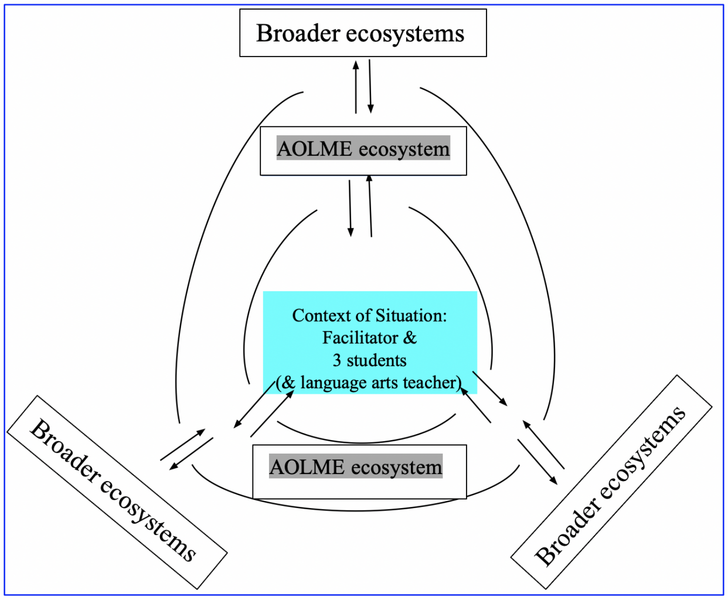
Thus, the spoken utterances used by the participants of this dissertation produced texts which can be analyzed. To refer to all the interactions that they produced, I use the "whole text" (Christie, 2005). To refer to shorter texts which I selected for focused analysis I use "text", "significant text" or "significant event". As products, texts mean clause by clause and SFL provides the tools to examine and interpret them in consideration of the specific environment where they happen.

The Connection between Clause, Meaning and Environment

Following SFL, there is an intimate dialectical relationship between the situations (meanings) realized discursively by the participants of this study and the AOLME environment wherein they happened—and between AOLME and broader environments.

Figure 3

(Context of) Situation-AOLME Environment and Broader Environments



Following SFL perspectives (Eggins, 1994; Halliday & Matthiessen, 2014; Martin & Rose, 2003; Martin & White, 2005) the situations that the participants realized linguistically—and thus the meanings realized—were conditioned by the interdisciplinary and sociocultural aspects that define the AOLME environment, and, in turn, the situations that my participants realized helped constitute and shape AOLME. The interconnection and flow of meanings illustrated in the figure (signaled by the arrows) occurred through the functional correspondence between the discursive situations realized in AOLME, and the broader environments. In other words, the meanings realized at different levels were in dialectical relationship.

Research Design

To compose the research design used in this study I took careful consideration of the characteristics of the participant spoken interactions and the AOLME environment. The participant interactions that happened during curricular activity in the introductory CT-CP course under focus featured great fragmentation and constant overlap of simultaneous conversations. During the creation of the video, the students were often working simultaneously on two computers, even three sometimes. Side conversations were normal; it was rare for the facilitator to maintain uninterrupted attention of all three students for sustained periods of time. Repeatedly, several things were happening at once. For instance, while two students discussed the video story, the facilitator guided the third student in programming something on the computer. In addition, sometimes spoken interactions did not happen at all. Halliday and Matthiessen (2014) note that exchanges of information are typically dependent on discourse, whereas in exchanges of action, discourse is ancillary to the task at hand. This happened in the practices under focus, especially as the students gained independence in the procedure adopted to program their characters. In my opinion, SFL methods constitute a phenomenal resource to examine CT-CP discourse as will be demonstrated, however, in fundamentally hands-on environments such as AOLME, resort to other data sources such as the code in progress produced and what was achieved through it as participant interactions happened is key. Often, in this study, based only on the discourse used by the participants in the video recordings, it was difficult to decipher what was going on exactly. In these instances, the recordings of the Python program and digital images were indispensable to decipher discursive meanings. In sum, it took careful thought to design the methodology that I eventually adopted and explain below.

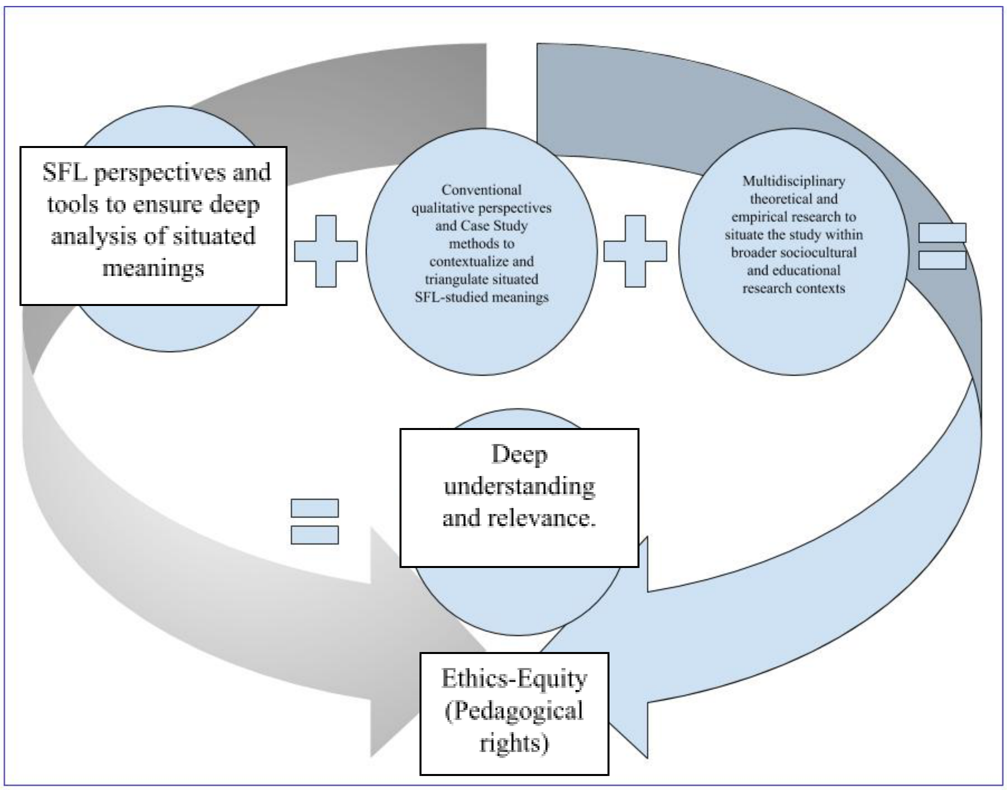
Research of a multidisciplinary nature such as this one, in complex environments such as AOLME hinge on (and calls for) what Bloome et al. (2004) refer to as "research imagination," that is, explorations of new understandings and links between research perspectives. Accordingly, this study combines approaches and methods from multiple perspectives and disciplinary fields. The concept of a "loose analysis outline" (Tracy, 2019) stresses that "the most promising analysis directions are inductively poignant and at the same time offer new or underexplored insight, connect up with research priorities, make use of past expertise, and meaningfully interact with existing research" (p. 181). This idea, in consonance with flexible research designs (Maxwell, 2013), adapted to the specific research environment, goals and analysis process that I detail below, lies at the heart of this study

Case study methods can be "a bridge across the paradigms" (Luck et al., , 2006, p. 105). Case studies enable inquiry into the complexities of unusual multi-relational systems like AOLME, which can be studied in depth and holistically and with paradigmatic flexibility (Anaf Drummond & Sheppard, 2007). The present study can be considered an ethics-equity oriented SFL qualitative case study because, through an ethical-equity lens, it coordinates (a) SFL perspectives and tools to analyze in depth the situations realized orally by the main participants and (b) case study qualitative perspectives and methods to understand them within the environment where they occurred (i.e., AOLME) and which they helped to shape.

The arrows in Figure 4 portray the ethical stance, based on student pedagogical rights (Bernstein, 1973), and responsibility of researcher and research of this study (Aguirre et al., 2017). I have used SFL because it affords the means to explore in depth the nature of situated CT-CP discourse (e.g., Kelly et al., 2012; Morgan & Sfard, 2016) and the interpersonal aspect of mathematics (e.g., Alshwaikh & Morgan, 2018; Herbel-Eisenmann & Wagner, 2007; Morgan, 2005, 2006; Morgan & Sfard, 2016). SFL provides the theoretical framework and linguistic tools to establish systematic relationships between the language used by facilitators and teachers in their interactions with students, and the *situations* they create through language within the broader multisemiotic environments—material, virtual and human—they are part of and constitute (Christie, 2005; Halliday & Matthiessen, 2014, Alshwaikh & Morgan, 2018).

Figure 4

An Ethics-Equity-Oriented Combined SFL-Case Study Research Design



I have combined SFL with case study qualitative methods because SFL perspectives require deep understanding of the 'context of culture' (i.e., environments) where they happen (e.g., Alshwaikh & Morgan, 2018; Martin & Rose, 2003; Morgan 2006; Morgan & Sfard, 2016). To address this need, a case study methodology is a perfect fit, since case studies are ideal for studying bounded systems such as AOLME, providing insight, discovery, and interpretation of phenomena (Merriam & Tisdell, 2015). In research terms, the overarching idea is to broaden research lenses. Also, I opted for case study methods on the grounds of the uniqueness of the study environment chosen (Merriam & Tisdell, 2015). The AOLME program, the environment wherein the CLD participants taught, learned and interacted, can be considered rather unique in that it combines research and educational practices where young university students of engineering, under the training and supervision of professors with strong interest in social justice, were the facilitators of the after-school mathematics-based CT-CP practices for the creation of videos I studied. Consequently, and with the ultimate objective to frame this study pertinently and in relevant ways within historical and socioeconomic realities where the study takes place, I have coordinated the SFL perspectives and tools with qualitative concepts such as strong consideration of the environment, researcher self-reflexivity, systematic analysis, thick description, and an orientation to real world concerns (Tracy, 2019).

A Contextualized Flexible and Adaptive Research Attitude and Design

I have approached the entire research design process of this dissertation with flexibility (e.g., theoretical perspectives, methods, participants, RQs) and careful and iterative examination (Cresswell, 2017). Tracy (2019) highlights the importance of staying open to multiple meanings and recommends initially 'throwing a wide net' (p. 188) and progressively approaching the question pointed out by Weck (2001): "what is a story here" as opposed to "what is the story here." Accordingly, I took into consideration contrasting perspectives and interpretations of the case I studied, mostly from a colleague AOLME Research Assistant with whom I conducted preliminary analysis of the practices and interactions of my participants. Also, I maintained a reflective iterative analysis (Tracy, 2019) which "alternates between emic, or emergent, readings of the data and an etic use of existing models, explanations, and theories" (p. 184). As introduced above, I iteratively conducted the analysis of the primary data source in triangulation with the secondary data sources. That is, I triangulated the study of the transcription of all the spoken interactions (captured on video) among the main participants (and other members of the program who approached the table where my participants worked) and the prototypes digital images and code that the students programmed to create their video with secondary data sources such as participant interviews, questionnaires and field notes.

This triangulation yielded conclusions that were recurrently illuminated by relevant theoretical and empirical research in various related fields. Depending on the instance, I searched for or reviewed key literature to enhance my interpretation of the data (e.g., Christie, 2005, Halliday & Matthiessen, 2014 in SFL; Bernstein 1971; 2003; Christie, 2005 in pedagogy; Bandura, 2000; 2001 in social psychology; Herbel-Eisenmann et al., 2012; Sfard, 2000 in mathematics; Bourke, 2018; Brennan & Resnick, 2012 in CT-CP).

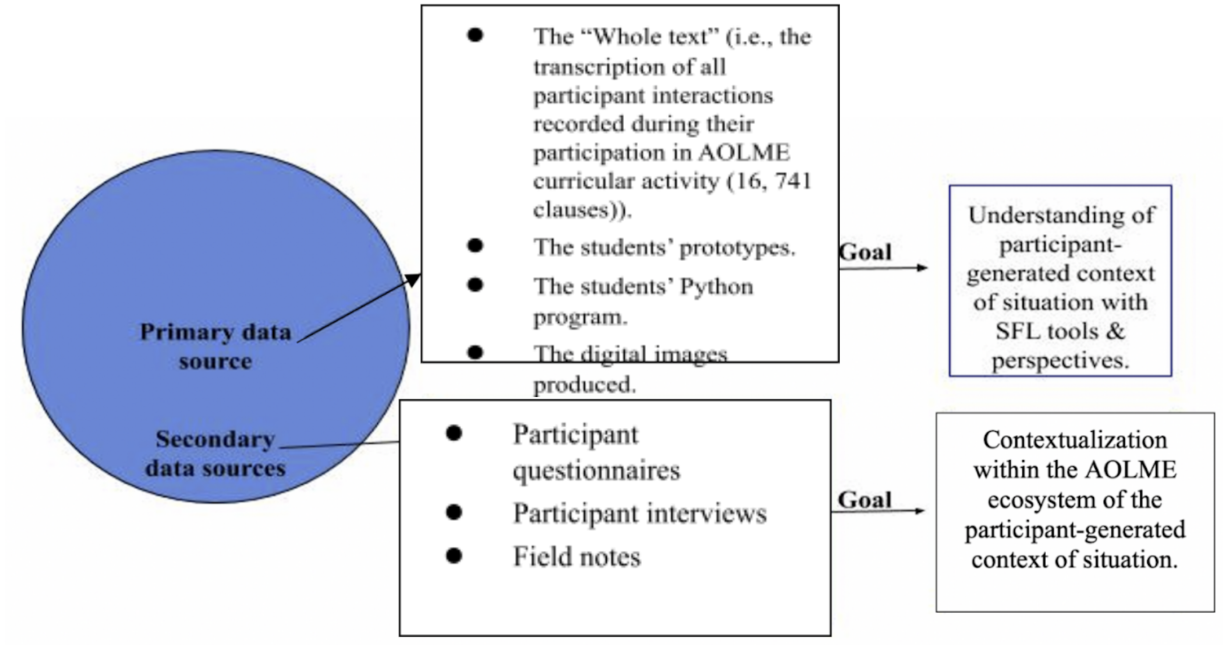
Consequently, the adoption of a flexible approach to the research design helped me in adopting a broad research lens, the reexamination of recurrent patterns and reevaluation of methodological decisions. These included refining my RQs, considering current STEM educational research equity-oriented directions in STEM fields (e.g., Herbel-Eisenmann et al., 2016) and applying the criteria I adopted to select *texts* for deep SFL analysis.

Primary and Secondary Data Sources

All data were gathered through AOLME. The primary and secondary data sources of this study pursued two differentiated goals. The primary sources, that is, the whole text and the recordings of the Python program, prototypes and images produced by the students, targeted an exploration of the *situations* (Halliday & Matthiessen, 2014) generated orally by the main participants of this study: A CT-CP facilitator, three students and their language arts teacher. The secondary data sources were targeted at contextualizing the knowledge derived from the SFL analysis of the primary data (Yin & Davis, 2007) (See Figure 5).

Figure 5

Primary and Secondary Sources, and Their Context-Environment-Related Goals



The primary data sources, therefore, have been the video recordings of the 11 sessions (about 2 hours per session) and the prototypes, Python program and digital products that were produced in the after-school computer programming practices. A camera focused the table where the participants used to work with the computer (s) (at times two or three computers were used at the same time) and another the monitor of the main computer that the participants used (The sessions were manually transcribed verbatim by myself in collaboration with two AOLME undergraduate students of engineering for subsequent linguistic analysis with SFL perspectives and tools) The video images of the participant interactions and rest of primary sources were used only to decipher the spoken language. Additionally, a second camera recorded the monitor screen of the main computer that was used by the students. This camera kept moment-to-moment records of what the students were developing, that is, the Python code. Recurrently, the students and, at times, the facilitator, ran the code that they were developing to check if it worked and if the digital images that it yielded were what they wanted for their video. These images were recorded too (several of them will be shown below). To be clear, this camera focused constantly on the main computer monitor, recording all that could be seen in it by the participants, be it the Python program or digital images under development or any other thing. I used this data to decipher the spoken language that the participants produced over the course of the 11 sessions. All total, the whole text was constituted by 16,741 clauses.

The secondary data was used to contextualize and triangulate the understandings gained through the analysis of the primary sources. These sources included: (a) Written questionnaires aimed at students perceptions about mathematics and engineering before and after participating in the AOLME program; (d) Facilitator questionnaires before and after participating in the AOLME program; (e) Participant language arts teacher questionnaires before and after participating in the AOLME program (Concerning their previous experiences and expectations (pre) and evaluations of the importance of the experience (post); (f) Student interviews concerning their attitudes and their perceptions on what they had learned (toward the end of their participation); (g) Facilitator individual interviews and focus group interview (with all spring 2017 facilitators) concerning the overall experience as facilitators in the Program (post-program); and, (h) Field notes at two sites: i) Middle school laboratory where the participants practices were experienced and ii) University premises where AOLME team meetings were held.

Coordination of primary and secondary sources yielded the findings of this study. Next, I describe the analysis method I adopted and the analysis scheme I developed and used to analyze the primary data source or *whole text.*

The Analysis Method

Informed by the methods used by Alshwaikh and Morgan (2018) and Morgan and Sfard (2016) in mathematics educational research, the analysis method that I adopted included two consecutive and differentiated phases aimed at progressively attaining deep and relevant understandings of the data: (a) The *Exploratory phase,* which was devoted to attaining initial and tentative understandings and the design of an analytic scheme and (b) The *Application phase,* which provided systematicity and depth in relevant understandings. The analysis method fulfilled several functions detailed in Table 10.

Table 10

Analysis Method

I. Phase	II. Function
1. Exploratory Phase.	a) The identification of significant linguistic choices and situations. b) The compilation of a CT-CP specific analysis toolkit. c) The division of the whole text in pedagogy-based manageable parts and select

	relevant text exemplars for focused SFL analysis. d) The design of a relevant Analytic Scheme based on the toolkits compiled.
2. Application Phase.	e) The application of the Analytic Scheme created to each text exemplar selected. f) The implementation of focused analysis in search for patterns, categories, and deep understanding.

The fact that the Exploratory phase of the Analysis method started the analysis and yielded the Analytic Scheme that was applied in the subsequent Application Phase, does not mean that the former ended once I later started. In fact, the Exploratory phase remained open during the whole analysis process. This was so, mostly because of the recursive and flexible nature of this dissertation (Tracy, 2019). Each function displayed in table 10 built on the previous one. Following, I explain both phases framing the discussion on the functions that were fulfilled and providing examples. Eventually, I focused analysis on 15 relevant text exemplars which were at once representative of the CT-CP experience of the participants in AOLME and relevant to the RQs of this study. The Analytic scheme designed is based on the SFL twofold view that discourse is at once a representation of human experience and an enactment of social relations and positions. It is informed by the previously discussed theoretical and empirical basis. I draw mostly on Halliday and Matthiessen (2014), and on applications of Halliday's extensive theoretical and empirical research. I am referring more specifically to research on the representation of science domains (Mizuno et al, 2018) and positioning in mathematics (Alshwaikh & Morgan, 2018; Herbel-Eisenmann & Wagner, 2007; Herbel-Eisenmann et al., 2013; Schleppegrell, 2012; Wagner & Herbel-Eisenmann, 2014). I also relied on research on mathematics discourse that focused discourse, thinking and communication (Lavie et al., 2018; Lavie & Sfard, 2019; Sfard, 2001b; 2008), effective communication (Sfard, 2000), and on SFL research that focused classroom discourse analysis and the structure curricular activity and pedagogical discourse in science and literacy development (Christie, 2005).

Exploratory Phase.

The explorative phase of the analysis yielded relevant initial understandings concerning my RQs. I started the *Exploratory phase* by dividing the *whole text* that the participants produced orally during curricular activity into clauses and transferring them to an *Excel sheet*, for the most part, in the order in which they took place. The whole text comprised 16,741 clauses. I inserted each clause in a cell of the *Excel sheet*; thus, the whole text was distributed in 16,741 cells and lines. Some participant utterances were not included in the transcript in the exact order that they happened. I had to make this decision when conversation overlaps happened, which was often the case. In these cases, preference was given to the facilitator-student interactions, rather than to student-student interactions. In other words, when two or more conversations were happening at the same time, I typically included in the *Excel sheet* first the facilitator-student conversation and right after, the student-student or student-language arts teacher conversation.

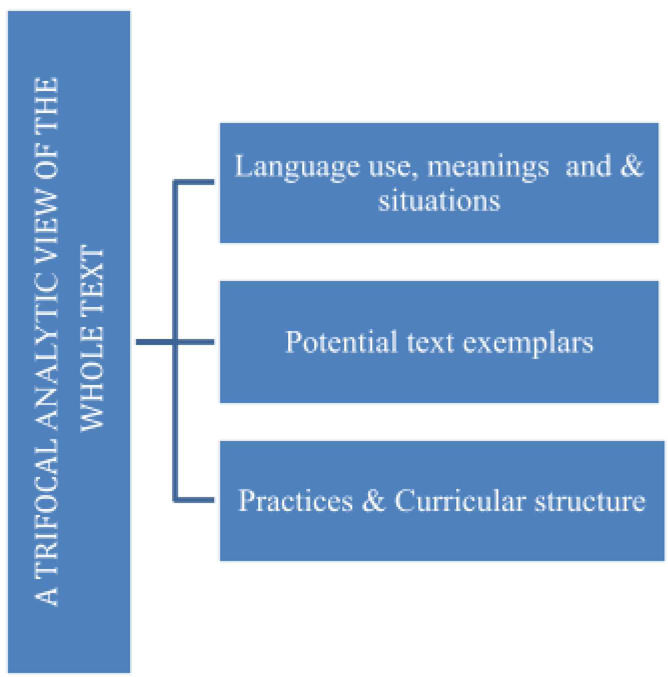
With text selection and the design of an appropriate analytic scheme as my ultimate goals, I examined the whole text carefully and on countless occasions. In so doing, I used the Excel affordances to highlight significant language uses and relevant situations, and to navigate the whole text with word searches. The word searches were instrumental. For instance, I used the word search function to identify when in curricular CT-CP key concepts such as “loop” or “variable” were used by the participants, finding that they could not be found in the excel sheet once the creation of the video started. This led me to conclude later that conceptual discussions on for loops and variables were not the focus of instruction during the video creation.

The Exploratory phase was by no means a straightforward process. It was a thorough inquiry which involved both scrutiny of the language used, the pedagogical situations created by the participants, and the ongoing selection of relevant “text candidates” for focused SFL-based analysis. The goal was to understand language use and what was being achieved through it in the *whole text*, for which I often resorted to the video recordings and the code and digital images that the students were generating. My greatest concern was to select a cohesive collection of texts of the participants’ spoken curricular activity that had the potential to “tell a story”, as opposed to a “cherry-picking of interesting samples” (Tracy, 2019). I opted to not lose sight of the joint enterprise shared by the participants and the rest of AOLME, that is, mathematic based-CT-CP teaching/learning and the video, and the processes that led to its creation. While I carefully explored the whole text from start to end, I kept in mind that curricular activity was aimed at such final project.

In the Exploratory phase, I was at once in search of significant meanings at the clause level[9] and a curricular structure that helped me locate text exemplars that were relevant to my RQs. Given that meaning is situational, that is, dependent on the environments where experience happens, I needed to compile from said SFL comprehensive toolkit, the specific toolkits that my study demanded. I did so as I explored the whole text. Thus, I kept a trifocal view: I maintained focus on (a) meanings at the clause level and the situations created (b) potential text exemplars, and (c) the practices and actual curricular structure that was being realized linguistically.

Figure 6

A Trifocal Explorative Analysis of the Whole Text



[9] In using "clause level", I refer to individual clauses and clause complexes, that is, two clauses, linked with a connector (Halliday & Matthiessen 2014)

Initially, to make sense of the whole text while examining it, I followed the basic structure which I had identified in the written lessons that the participants followed. The curriculum (see Appendix B) comprised two parts: (1) Six collaborative hands-on lessons centered on several computer programming, mathematics, and image & video processing concepts; (2) A final seventh lesson which centered on the application of the previous lessons to the creation of a video in Python. The students were to create their own-envisioned animation video, through a process in which they were urged to discuss what they wanted to happen in the video, draw and mathematize[10] on paper its scenes or frames, and program it in Python.

Identification of Significant Linguistic Choices and Situations.

Following the fundamental SFL idea that participant linguistic choices realize meaning and create situations, I highlighted on the whole text excel sheet (a) significant language use, both in English and Spanish, and resulting situations that concerned my research interests and questions. These concerned fundamentally (a) the nature of the experience offered to the CLD students by the facilitator (i.e., *participants*, processes and circumstances involved in curricular activity) (RQ1); (b) the roles and positions offered and adopted through exchanges of propositions (information) and proposals (action) (RQ2); and (c) student learning as signaled in their own discourse (RQ3). Student learning had not been anticipated and was not the focus of this study at its start. However, the finding of evident student linguistic choices that signaled their agency at key moments of curricular activity stimulated my interest in the analysis of student learning and the addition of a new RQ, in congruence with the flexible research design adopted (Tracy, 2019) Thus, my focus on *significant linguistic choices* uncovered relevant

[10] In this study "mathematization is a technical term representing relationships in the natural world using mathematics" (National Research Council, 2012, p. 16) and mathematize concerns the processes that lead to mathematization.

meanings and situations regarding the content of the instruction, the interpersonal relationships and positions enacted, and student learning.

Following are several noteworthy examples of linguistic choices and the situations which were identified in the discourse of the facilitator in the Exploratory phase. They relate to the nature of the CT-CP activity offered to the CLD students (Based on Halliday & Matthiessen, 2014):

1) A salient facilitator's use of *participants* such as "we" and "you" (the participants) which were continuously construed as agentive by the facilitator in CT-CP *material processes* such as "add," "put" (meaning "code in the computer") and run (execute what was coded). These linguistic choices helped create CT-CP hands-on situations such as testing.

2) Non-academic *participants* such as "milk" "spoon", asteroid and "fin" shared the instructional field (Christie, 2005) with academic ones such as "integer," "algorithm," "im_fill" "pixel" and "image". These linguistic choices helped create CT-CP meaningful situations, that is, meaningful creation of algorithms.

3) A salient facilitator's use of Spanish *participants* such as "color" ("colour") and processes such as "cambia" ("change") helped realize CT-CP situations such as debugging situations.

4) A salient facilitator's use of abstract, objectified *participants* such as "8 to 8" and "9, 9" pervaded discourse specially as curricular activity progressed during the creation of the video in Python. Such objectifications helped realize CT-CP algorithm creation and communication situations.

5) A salient facilitator's use of modal verb forms such as "have to" and "can" signaled high and low degree of obligation, thus, authority of disciplinary discourse alternated with latitude (Herbel-Eisenmann & Wagner, 2014). Thus, such discourse alternated CT-CP situations where conditions had to be met with situations where flexibility was possible.
6) A pervasive facilitator's use of logical meanings (connectors) of causal nature such as "to," "in order to," "so," and "so that" helped create situations that signaled CT-CP practices of a pragmatic nature.

On the other hand, informed mostly by Halliday & Matthiessen (2014), Herbel-Eisenmann and Wagner (2007), Herbel-Eisenmann et al. (2013), and Schleppegrell (2012), I identified in the discourse of the facilitator relevant interpersonal meanings and positions. My foci were relevant propositions and proposals and the relevant pedagogical situations that they helped create. Relevant instances when the students were positioned as providers of key information and as doers of computer programming activities were identified. Thus, I highlighted instances when the facilitator made relevant statements and requests of information (authentic and test questions), and relevant suggestions and commands. A few IRE patterns (Sinclair & Coulthard, 1975) were identified, but they were rare. While special attention on the nature of the statements and the questions that the facilitator formulated was maintained, the instances when it was the students who asked the questions, which were very rare, were also in focus. Similarly, who made the proposals was under scrutiny. Importantly, for this investigation, the focus on propositions and proposals helped uncover that the students made proposals (commands) with increasing frequency as curricular activity progressed. This was key in the identification of student agency in key moments of curricular activity such as in the creation and communication of the algorithm of the video, which signaled student CT-CP learning. This preliminary finding informed the research design to the extent that I created RQ to address student learning. The interpersonal linguistic choices used by the facilitator (propositions and proposals) helped create relevant pedagogical situations. Two salient examples are:

(1) A salient facilitator's use of authentic questions aimed at eliciting students' ideas for the video project. These questions framed ongoing discussion centered on the students interests and knowledge which helped create situations that signaled that the students' experience in the world mattered (Herbel-Eisenmann & Wagner, 2007).

(2) A salient facilitator's use of proposals during fundamental CT-CP practices. "Let' look at the code" and "9, 9" (a step in the algorithm of the video as will be explained) are examples of two facilitator proposals that created situations where the students were positioned as capable computational thinkers and programmers as will be explained (as doers and thinkers in testing and coding CT-CP practices).

Last, informed by Halliday and Matthiessen (2014), Lavie et al. (2018), Lavie and Sfard (2019), Sfard (2001b, 2008), I identified instances that signaled student learning in the discourse of the students. Examples will be illustrated in Chapter 5. Three examples are as follows:

(1) The students' pervasive use, towards the end of curricular activity, of highly abstract and objectified commands constituted with *participants* such as "5,5" and "5,17", which were mathematical representations of an asteroid and constituted the algorithm of the students' video in Python, helped create CT-CP situations that signal student CT-CP learning.

(2) The students' salient use of highly abstract and objectified commands in Spanish, which included mathematization representations of the background of the video, helped create bilingual collaborative CT-CP modeling situations where one student communicated mathematization codes to his language arts teacher. The student abstract language used helped create CT-CP situations that signal student CT-CP learning.

(3) The students' use of the pronoun *I* helped create situations that signal student CT-CP learning. A salient situation was the modification by a student of the Python code the students produced to improve the symmetry of a digital image (the student use of the pronoun *I* signaled his agentivity, thus learning in this situation).

Thus, in the Explorative phase of the analysis I identified in the whole text significant language use, and situations. In doing so I examined the meanings that were realized, gaining insight about the nature of the curricular activity at hand, about agency, about facilitator and student roles and positions and about signs of student learning. My explorations revealed initial understanding of how the facilitator engaged the students in situations such as the collaborative construction of a video narrative for the students' video, the collaborative mathematization and drawing of characters and background elements that the students wanted in the video story, the collaborative creation and communication of the algorithm that enabled it, the explanation of CP concepts, and the coding in Python of the video. Also, I gained initial understanding about how the facilitator was managing activity (regulative discourse) (Bernstein, 2003). For instance, I noticed how the facilitator was distributing labor and its pace concerning activities such as mathematizations, when the main computer was turned on and off, who was using it, when she decided that more than one computer be used, who the facilitator was guiding one on one when this happened, etc.

The compilation of an appropriate analysis toolkit.

The SFL "toolkit" provides an exhaustive and comprehensive compilation of textual indicators that explain the whole lexicogrammar[11] of human experience. This toolkit allows focus on what is achieved by clauses and texts (Morgan & Sfard, 2016). Like Morgan and Sfard, I used the SFL "toolkit" (Halliday & Matthiessen, 2014) to analyze the functioning of my whole text with respect to my research interests and questions. To select from the whole toolkit of the English language the specific toolkit which concerned this study, as I explored the whole text, I compiled the textual indicators that related its RQs. I developed three toolkits, one for each of my three RQs. These toolkits were updated as I explored and re-explored the whole text. Then I used them to create the Analytic scheme that I used in the Application phase of the analysis.

The toolkits, basically a compilation of *textual indicators,* constitute the fundamental analytic instrument I used to examine the meanings realized by the participants of this study. Following I have displayed in Tables 11, 12, 13 and 14 the four toolkits that I used for the analysis that concerned RQ1 and RQ2. Tables 11 and 12 concern RQ1 (i.e., the nature of the CT-CP experience), Table 13, RQ2 (i.e., positioning) and Table 14 RQ3 (i.e., student learning). To address RQ3, I used a combination of SFL indicators and non-SFL discursive indicators based on Lavie et al. (2018).

[11] Following Halliday and Matthiessen (2014)' functional perspectives on language use, I use the term "lexicogrammar" to refer both to the lexis (i.e., vocabulary) and (functional) grammar of the whole language. Halliday and Matthiessen clarify that "[…grammar and vocabulary are not two separate components of a language – they are just the two ends of a single continuum" (p. 7). I use lexicogrammar to refer to the resources that the language system affords. Put in simple terms for the purposes of this study, lexicogrammar allows speakers of the language both a system of words (mostly nouns and verbs) that relate specifically to the specialized practices at hand and a system

Table 11 displays Toolkit #1. It allowed the exploration of the nature of the CT-CP universe offered by the facilitator. Therefore, Toolkit #1 focuses (a) the process types used by the facilitator, (b) the different *participants* involved in activity by the facilitator (e.g., "computer," "asteroid," "We," "I," "Herminio") and (c) the circumstances that the facilitator associated with processes. The table includes the textual indicators (grammatical categories) used to identify processes, *participants,* and circumstances as well as some examples (e.g., textual indicators for purpose ("to"), condition ("if"), etc.).

Table 11

Toolkit #1: To Inquire into the Nature of CT-CP Experience (Field), Including Logico-Semantic Links Between Clauses (RQ1)
(Note: To keep this table simple,I have opted to use footnotes for key information)

The *participants*[12] involved in activity (Textual indicators: nominal groups[13] and pronouns)		The processes involved in activity (Textual indicators: verbs)		The circumstances involved in activity (Textual indicators: prepositions, adverbs and modal[14] verbs)	
Participant type	*Participant* appellatives used	Process type	Example	Circumstance type	Example
-Human	-"We" -"You" -"You guys" -"I" -"Herminio" -"Juan" -"Michael" -Any non-present human agent (typically "they")	-Material -Mental-cognitive -Mental-desiderative -Mental-perceptive -Mental- emotional -Relational-attributive -Relational-identifying -Relational-Symbolizing -Verbal -Behavioral -Existential	- "run" - "multiply" - "know" - "want" - "see" - "There you go!" - "have legs" - "is bigger" - "it is your variable" -"represent" - "tell" - "kidding"	-Space -Time -Manner-means (instrumentality) -Manner-quality -Manner-degree -Cause-reason -Cause-purpose -Contingency-condition -Contingency-default -Flexibility	- "here" - "later" - "with" - "fast" - "too" - "thanks to" - "to" - "if,""have to" (high degree of obligation) - "unless"

[12] Although the whole text is not at all rich in descriptions, adjectives and pronouns were commonly used to qualify and substitute *participants*, both human and non-human. Possessive adjectives such as ""your" were frequently used by the facilitator to signal student possession of objects and concepts (e.g., "your code", "your frame", "our video"). Deixis was also very common and typically realized by pronouns such as "this"
[13] Nominal group refers to a qualifier + noun (Halliday & Matthiessen, 2014) or to a noun alone. For example: "This number", "red asteroid", "Computers".
[14] I drew from Alshwaiikh and Morgan (2018) who identified mathematics rules and conventions in mathematics classrooms using indicators such as "necessary" and "always". I have used strong modal verbs such as "have to" to proceed similarly. By contrast, I used modal verbs such as "can" and "could" as indicators of low degree of obligation, thus flexibility.

-Non-human	- “equation,” “variable,” “asteroid,” “tree,” “character,” “computer”		- “Here is our formula”		“can,” “could” (low degree of obligation)

Toolkit #1 was thus instrumental in examining in the facilitator's discourse who or what was construed as agentive and their characteristics, as well as the nature of the processes that the facilitator involved in the CT-CP practices under focus and their characteristics (e.g., purpose and conditional circumstances).

Toolkit #2, displayed next, allowed further exploration of the characteristics of activity through scrutiny of the use of links that connect clauses (Halliday & Matthiessen, 2014) as well as other resources that organize the flow of activity, such as ellipsis and in-text referencing. Table 12, below, includes the textual indicators used to examine the meanings used by the facilitator to construe the CT-CP experience offered to the CLD students of this study.

Table 12

Toolkit #2: To Inquire into the Nature of Experience at the Level of the Logical Meanings Created Between Clauses[15] *(Mode) (RQ1)*

Textual indicators	Examples
-Conjunctions[16]	"and" (addition), "so that" (cause-effect), "if," "and then" (condition), "to" (purpose)
-Ellipsis[17]	(code) "9, 9" (i.e., when Juan was using a command to invite the facilitator to code "9, 9", then he omitted the word "code".)
- Referencing[18] the text	"the" (article "the" references a code which is known by the participants) (e.g., "Let's look at the code") (anaphoric)

[15] To focus on what happens in a text [… "it is important to be able to think of text dynamically, as an ongoing process of meaning" (Halliday & Mathiessen, 2014, p. 593). According to these researchers, "There are four ways by which cohesion is created in English: by (i) conjunction, (ii) reference, (iii) ellipsis, and (iv) lexical organization" (p. 603).

[16] Conjunctions are used to link clauses (Halliday & Matthiessen, 2014).

[17] "Ellipsis is usually confined to closely contiguous passages, and is particularly characteristic of question + answer or similar 'adjacency pairs' in dialogue" "Ellipsis makes it possible to leave out parts of a structure when they can be presumed from what has gone before. Ellipsis indicates continuity, allowing the speaker and addressee to focus on what is contrastive…] (Halliday & Matthiessen, 2014, p. 606)

[18] Halliday and Matthiessen (2014) distinguish between referencing inside and outside the text (i.e., anaphoric & exophoric reference)

These conjunctions, for instance, realized sequences of action verbs, typical in procedures (Derewianka & Jones, 2016). For example, the facilitator was giving directions to Herminio to start programming the first frame of the video: … "**and then** put || the rows is equal to 20, || the columns is equal to 20, || **and then** your name begins, || just frame one, right? || **and then** frame is equal to, || **and** here you are gonna put… (Lines 12178-12185).

Toolkit #3 (See Table 13 below) includes the textual indicators that allowed the examination of (a) whether and when the facilitator positioned the CLD students as computational thinkers and computer programmers (through **proposals**); (b) whether and when the facilitator positioned the students as receivers of information (when she used statements) or providers of information (through **propositions**) (exemplified on Table 18 below)

Table 13

Toolkit #3: To Inquire into the Way the Facilitator Positioned the CLD Students (Tenor) (RQ2)

Textual indicators	Function	Position
- *Proposal*	-Command	Computational programmer and thinker
-*Proposition*	-Statement	Students whose prior experience in the world

		matters (Grade-level mathematics)
	-Authentic question	Students whose prior experience in the world matters (providers of key, meaningful information kneaded for the creation of algorithms)
	-Test question	Students whose prior experience in the world matters (Grade-level mathematics)

The examination of the use of propositions and proposals at key moments of activity yielded key preliminary understandings at the Exploratory phase which were further explored in the Application Phase by means of the analytic scheme that I illustrate below. However, before doing so, let me provide the last toolkit developed, which I used after reviewing the literature in search for indicators that signaled student learning.

Informed by Halliday and Matthiessen (2014), Lavie et al.,(2018), Lavie and Sfard (2019), Sfard (2008; 2001b;), I used the indicators displayed in Table 14 to search for instances that signaled student learning in the discourse of the students. The table includes examples that were identified in the Exploratory phase.

Table 14

Toolkit #4. To Inquire Signals of Student Learning (RQ3)

Indicators	Examples
-Signals of student active participation in CT-CP *established practices.*	-modeling, modularization, algorithm thinking, testing, evaluating & refining programs, and creativity.
-Student use of CT-CP *keywords.*	- "im_show", "parenthesis"
-Students' use of *visual mediators.*	-Student creation of pencil-crayon drawn prototypes.
-Students use of *endorsed narratives or mathematical "truth statements".*	-Students' use of algorithms constituted by *participants* such as "5, 5" and "5, 17".
Flexibility: Signals that the students had more than one way to perform a task.	-Flexibility in the design of the prototypes, in the creation of the algorithm and the organization of the Python code.
Bondedness: Signals that the steps of a procedure adopted by the students towards obtaining a final product fed into later steps.	-The students modeled with mathematics each of their pencil-crayon drawn prototypes to then program each of them in Python.
Applicability: Signals that the students applied what they learned to different environments.	-The students presented their Python code and digital video in the school library

Agentivity: Signals that the students were agentive in the CT-CP practices they experienced.	-The students used the pronoun *I* and commands to lead their participation in CT-CP practices such as testing-evaluating-modifying a program and algorithm communication.
Objectification: Signals that the students substituted objects of the real-world with abstract concepts.	-The students used two-dimensional mathematical arrays to define their pencil-crayon drawn prototypes which had originated in the ideas that framed the story that they wanted to tell in their video in Python.
Substantiation: Signals that the students were able to explain what they did.	-The students explained their video to their families and the rest of the AOLME team.

The development of the 4 toolkits displayed above was key since it allowed the design of the Analytic Scheme that I used to further the analysis on a selection of carefully chosen text exemplars. Additionally, I divided the whole text in four stages that helped organize curricular activity on a pedagogical basis. To do so, I drew on Christie's (2005) SFL discourse analysis research on literacy development and science classrooms. Following, I discuss first the four curricular stages I used, then the text selection criteria I adopted and, right after, the methodological and theoretical considerations that guided my selection.

The Division of the Whole text in Pedagogy-based Curricular Stages and Relevant Text Exemplars.

Following are the curricular stages experienced by my participants

1. *Task Preparation Stage*. This stage comprised a set of hands-on activities framed by various mathematics-based CT-CP and image and video processing concepts. It had the final goal of preparing the students with the necessary knowledge and skills to create their animation video in Python. Delimiting this stage was quite straightforward. It amounted to checking that the participants covered the lessons of the curriculum. To do so I checked that the discourse used by the facilitator contained the concepts—signaled by specialized nouns—in the sequence of activities detailed in the written lessons. This stage was defined linguistically by a combination of facilitator monologic and dialogic modes. However, the dialogic mode was far more frequent.
2. *Task Orientation Stage.* In this stage the facilitator established the overall pedagogic directions and expectations concerning the students' collaborative video project. Propositions pervaded, that is, the facilitator fostered exchanges of information. On the one hand, the facilitator used statements with modals and other authority structures to set the pedagogic expectations, and the conditions and affordances of Python to create the video. On the other hand, she used authentic questions to inquire about the story the students wanted to tell with their video. Pedagogic directions were reviewed during the next stages but progressively disappeared as the whole text progressed.
3. *Task Specification-Realization Stage.* In this stage the facilitator's proposals gained evident ground. In other words, action concerning the creation of the video started. The facilitator provided one-on-one guidance of student coding and started to

distribute labor among the students since design and modeling processes were at stake too. The facilitator foregrounded expositions of the conditions and affordances of Python. The explanations of the facilitator were typically concise and pragmatic; student questions were rare.

4. *Task Realization stage.* This stage was marked by student independent activity, agency, collaboration, and the use of Spanish. Facilitator proposals were pervasive, which she occasionally coordinated with propositions with the circumstances associated with the processes realized. Student use of proposals was extensive. Student use of pronoun *I* and commands signaled self-initiated agencies in fundamental CT-CP practices. Collaboration among all the participants pervaded (including the facilitator), now with the participation of the language arts teacher who continuously fostered the use of Spanish by initiating discussion in Spanish.

Text Selection Criteria and Process.

The exploration phase process resulted in 15 significant *text exemplars*, which I interchangeably refer to as *significant text*, *significant event*, or simply *text* associated with a number that helps in its identification. The 15 texts selected comprise 1,169 clauses out of the 16,741 total clauses that comprise the whole text. Of these 1,169 clauses 486 clauses were pronounced by the facilitator and the rest by the students (most) and by the language arts teacher.

These fifteen texts average around 80 clauses each, ranging from 22 to 181clauses. My target was to articulate with strong foundations 'a story' based on the curricular activity of my facilitator. To select them required an "ongoing process of digging sorting, coding and reflecting" (Tracy, 2019, p. 217) and hinged crucially on the systematic relationship between the language used by the facilitator and rest of participants, and the *situations* they created, as well as on etic (macro) understandings. I explain below the text selection criteria that I used and the process of text selection I followed, for which I drew on knowledge gained from Bloome (2005), Christie (2005), and Tracy (2019).

1) The texts selected concern the research interests and questions of this study.
2) The texts selected comprise a cohesive set which I consider representative of the AOLME mathematics-based CT-CP curriculum for the creation of videos in Python that my participants experienced.
3) The texts selected comprise 'a story' based on the discourse of the facilitator and the rest of participants which is relevant to CT-CP teaching/learning and previous research in various fields of knowledge, from pedagogy and sociology through mathematics to CT-CP.
4) Although fragmented, the texts selected constitute sustained conversational segments with thematic coherence and are limited before and after by a sustained topic change.
5) The final selection of texts remained open until the end of the analysis process (in congruence with the loose analysis design of this study)

As Martin & Rose (2003) note, some texts seemed to stand out with clarity because they included situations that were significant and salient. Other times it was the lexicogrammar used by the participants which illuminated the situations. For instance, I selected Text # 4: Lines 10,862-10,930, which I titled "Brainstorm*:* Favorites for the Video" because the situation Teresa was creating with Herminio, Juan, and Michael called my attention. It seemed clear that the students took the floor, talking in excitement about what they wanted to include in their video It was subsequent close examination of the discourse used which revealed Teresa's continuous use of authentic[19] questions to get at the students' favorites for the video. In contrast, I selected Text #13 (Lines 14,267-14,300), which I titled "Student explains Python code to facilitator: Don't delete it," because the use of grammar stood out (instead of the situation created). In this case, Michael's salient use of commands in the imperative mood was striking (as well as his helping Teresa navigate the code, who seemed lost at that moment), given that in educational environments it is more common for teachers and facilitators to use commands to address students rather than the other way around. In my search for significant texts and "a story," I also relied on etic considerations centered on pertinent educational research that I deemed significant for CLD students. For instance, I selected Text #11 (Lines 13,109-13,317), which I titled "Community Debugging with a School Teacher" which involved a schoolteacher, 3 CLD students, and one university student of engineering (the facilitator) because of the relevance of CT-CP collaborative practices for schools in general and underrepresented communities and students. Last, but not least, I selected Text #9 and Text#10 because they included significant use of Spanish. A key condition for text selection was that together they represented a cohesive and representative set of the AOLME curriculum experienced. To achieve this end, I relied on (a) Christie's (2005) curricular structures, which assisted my organization of significant texts across the whole curricular activity experienced, (b) the knowledge I had gained of the whole text in the Exploratory phase by means of the toolkits (Tables 11, 12, 13, and 14).

The SFL Analytic Scheme Adopted.

[19] [1] I use *Authentic questions* to refer to questions for which the facilitator (presumably) does not know the answer (e.g., the favorite place of a student). In contrast I use *test questions* to refer to questions used by facilitators to *test* student knowledge (e.g., What's 253 in binary notation?)

To systematize and increase the analytic focus implemented in the Exploratory phase, I created an SFL-based Analytic scheme based on the toolkits shown above. To do so, I drew on various SFL-based research, most importantly, Alshwaikh & Morgan (2018), Halliday and Matthiessen (2014), Herbel-Eisenmann & Wagner (2014), Morgan & Sfard (2016) and Schleppegrell (2012). The SFL-based Analytic scheme comprises four different analytic tables. I implemented each of the four on the 15 significant texts selected. The Analytic scheme was designed initially to address RQs 1 and 2. Yet since significant signs of student learning were found, as was noted, I also used the analytic tables to address RQ3. Below, I explain the purpose of each of the four analytical tables. .

1) *Analytical Table 1*: Facilitator-process links: Focus on processes (Schleppegrell, 2012) This table enabled focus on the p**rocesses** used by the facilitator in the 15 significant texts selected. This aspect of the analysis is reflected in Table 15 below.
2) *Analytical Table 2*: Facilitator-process links: Focus on *participants* (Schleppegrell, 2012) (Recall that throughout this dissertation, *participant* (in italics) is a grammatical category—any person or thing construed as agentive by the facilitator, as opposed to any study participant (no italics)) I used Analytical Table 2 to get at who or what the facilitator construed as agentive in her discourse in the 15 significant texts selected. This aspect of analysis is reflected in Table 16 below.
3) *Analytical Table 3*: Facilitator use of modals. This table helped me examine CT-CP disciplinary authority and flexibility (Alshwaikh & Morgan; 2018) in the discourse of the facilitator in the 15 significant texts selected (Table 17 in this document, below)

4) *Analytical Table 4*. Participant-speech function. I used this table to examine separately the proposals (commands and offers) and propositions[20] (Halliday and Matthiessen , 2014) (statements, authentic and test questions) used by each of the 5 participants of this dissertation in the 15 significant texts selected. Please let me recall that participant (not italized) refers to the participants of this study, that is, the facilitator, the students, and their language arts teacher. In order to get increased focus on the content that was being exchanged in propositions and proposals, I separated them into instructional and regulative discourse (Bernstein, 2003) (Table 18 in this document, below)

In the section below (the one devoted to the application phase of the analysis), I illustrate how I used the four analytic tables by including excerpts of Text #4: “Brainstorming “Student Favorites” for the Video (Lines 10,862-10,930 of the whole text). Yet first let me summarize the ideas discussed in the Exploratory phase of the analysis.

Summary of the Exploratory Phase.

[20] A key modification with respect to the other analytic tables was introduced in the analysis of propositions and proposals in Analytical Table 4. The modification concerns the arrangement and exploration of clauses in the table. Previously, I had displayed and explored each clause separately. Now, I displayed and explored clauses in connection with other clauses, i.e., ‘clause complexes. This decision refined my scrutiny of ‘logico semantic’ relations between processes (Halliday and Matthiessen, 2014), which afforded examination of the characteristics of the CT-CP at hand. It also allowed categorization of complete proposals and propositions (including questions).

The Exploratory phase was a thorough immersion in the oral interactions produced by the participants during their CT-CP practices, that is, a scrutiny of the whole text. It allowed initial understanding of the lexicogrammar resources used and what was being achieved during their curricular activity. I resorted to the SFL comprehensive toolkit of the language (Halliday & Matthiessen, 2014) to progressively compile the CT-CP specific toolkits that the CT-CP environment and research interests involved demanded. This phase targeted the selection of "text candidates" for focused SFL analysis. To organize text selection and complete a cohesive representative set of text exemplars, I used curricular structures identified in Christie's (2005) SFL studies of classroom discourse. Eventually, the Exploratory Phase yielded fifteen text exemplars with the potential to comprise a relevant "story" of the CT-CP practices experienced by the facilitator and the rest of my participants in AOLME. To further the understanding gained in the Exploratory phase of the analysis, I created an SFL-based analytic scheme based on the toolkits compiled. The analytic scheme comprised 4 analytic tables which were specifically designed to address the RQs by getting at the meanings realized by the participants. The objective pursued with the analytic scheme created was the systematization and deepening of the analysis conducted in the Exploratory phase.

Application Phase

The Application phase of the analysis comprised two distinct implementations. First, I implemented the Analytic Scheme created in the Exploratory phase to each of the 15 text exemplars selected. Second, I implemented focused analysis in search for patterns, categories, and deep understanding.

Implementation of the Analytic Scheme.

I recursively applied the four different analytic tables designed in the Exploratory phase to the 15 significant texts selected. Importantly, the implementation of the analytic scheme was assisted by synchronous examination of the in-process prototypes, Python code and digital images developed by the students. As was noted, these programs and digital image and video representations.

products were used to decipher the discourse used by the participants of this study. In this section I display Excerpts of the tables as they were implemented to analyze Text #4 with the toolkits developed in the Exploratory phase of the analysis.

- Table 15 (an excerpt of Analytical Table 1) focuses on the nature of the processes (verbs) that the facilitator used in Text #4;
- Table 16 (an excerpt of Analytical Table 2) focuses on the *participants* that she construed as agentive;
- Table 17 (an excerpt of Analytical Table 3) focuses on the facilitator use of modals and the authority/flexibility of CT-CP disciplinary discourse;
- Table 18 (an excerpt of Analytical Table 4) focuses on the exchanges of action (proposals) and information (propositions) realized by the participants of this study.

Table 15, below, is an excerpt of the actual Analytical Table 1 as implemented on Text#4. It illustrates the processes (in bold, for emphasis) that the facilitator used to construe the CT-CP experiential universe in relation to the *participants* that she involved in CT-CP curricular activity. I organized the *participants* in the categories *We, You, You guys, They* (human agents other than the participants*), Any student name* (i.e., “Herminio,” “Juan,” or “Michael”), *Facilitator/Teresa/Miss, The student's language arts teacher/Janet, Concept* (i.e., any concept with grammatical agency such as "asteroid") and *Computer*. I added an extra category for *passive voice* to explore obscuration of agency (Morgan & Sfard, 2016). For reasons of space, I only included the *participants We, You, You guys, I*, *Concept* and *Computer.* As can be observed, only the *We* row is filled with clauses. All process type cells (top of the Table) contain the number of times the process was used by Teresa and the ratio of process to total number of clauses used by Teresa in Text #4.

Table 15

Facilitator-Process Links. Focus on Processes (Text #4)

Process: *Participant:*	Mater[21] 3 (12%)	Ment. (cogn.) 1 (4%)	Ment. (desid.) 15 (62%)	Ment. (percep.) 2 (8%)	Ment. (emot.) 0 (0%)	Relat. (attrib.) 1 (4%)	Relat. (ident.) 1 (4%)	Relat. (symb.) 0 (0%)	Verb. 2 (8%)	Behav. 0 (0%)	Exist. 0 (0%)	Clause Total 25 (100%)
We/Us	(What do we wanna) ***do?*** let's *go* ahead (we could *make* it like)[22] the planets	and ***brains torm,***	What do we ***wanna*** (do?) okay, So, we *like* sports,			so, here we ***got*** these three in the sports section, right?						

[21] Mater. (material); ment. (mental); cogn (cognitive); desd. (desiderative); emot. (emotional); relat. (relational); attrib (attributive); ident. (Identifying); symb. (symbolizing); verb (verbal); behave. (behavioral); exist (existential).
[22] In parenthesis I typically include clauses that were omitted through processes of ellipsis.

You												
You guys												
I												
Concept[23]												
Computer												

[23] I used this category to categorize any *participant* construed as agentive by the facilitator, be it academic, e.g., *integer* ("an *integer* can represent positive or negative values" or nonacademic, e.g., *asteroid* ("The *asteroid* gets bigger").

This excerpt illustrates that the material processes used by the facilitator constituted 12% of the processes used by her in Text#4, the mental processes 4% and so on. The excerpt also illustrates the facilitator's construal of a *collective* experience (use of "We") where she used the *material processes* "do" and "make," the *cognitive process* "brainstorm" and *desiderative process* (processes which concern desires). "want". These processes and participants contributed to the facilitator construal of the CT-CP universe that she offered the CLD students of this study.

Table 16, below, is an excerpt of Analytical Table 2 as implemented on Text#4. The focus is now on *participants*[24] in relation to process types. Now the *participants* are bolded for emphasis and calculations are shown in the *participant* cells only with respect to *participants We and You,* in this excerpt.

[24] Let me insist on the fundamental difference between a *participant,* and the participants of this study. *Participant*, in italics, refers to a essential grammatical element of experience which or who is fundamental in the realization of a process (e.g., in clause "the asteroid is red" the asteroid is the *participant*) background and that , *and the* participants of this study (e.g., Michael and Teresa) who may or may not be *participants* in a given clause. Full discussion of the different types of *participants* is beyond the *scope* of this study; however, a few considerations concerning this topic are included in Chapter 4.

Table 16

Facilitator-Process Links. Focus on Participants (Text #4)

Process: *Participant:*	**Mater.**	**Ment. (cogn.)**	**Ment. (desid.)**	**Ment. (percep.)**	**Ment. (emot.)**	**Relat. (attrib.)**	**Relat. (ident.)**	**Relat. (symb.)**	**Verb.**	**Behav.**	**Exist.**	**Clause Total** **25**
We (8/25=32%)	(What do we wanna) *do*? -let's *go* ahead -(we could *make* it like) the planets	-and *brainstorm,*	-What do we *wanna* (do?) -okay, So, we *like* sports,			-so, here we *got* these three in the sports section, right?						
You (9/25= 36%)	(you)								-(you) give me your			

									favorites			
You guys												
I												
Concept												
Computer												

The excerpt shown illustrates that in Text#4, the facilitator construed the participant "we" as agentive in 32 % of the processes, which pointed at the collaborative nature of the CT-CP universe that the facilitator offered the CLD students of this study. It also shows that she construed the *participant* "you" as agentive in 36 % of the processes, which indicates that their participation was important.

Table 17, below, is an excerpt of Analytical Table 3 as implemented on the same Text #4. This analytical table allowed examination of the modals used by the facilitator to associate degrees of obligation/flexibility to the CT-CP universe that she offered discursively. The table includes all the modal verbs that were used by the facilitator in Text#4.

Table 17

Facilitator Use of Modals in Both Instructional and Regulative Discourse (Text #4)

	Instructional discourse high modals	Instructional discourse median modals	Instructional discourse low modals	Regulative discourse high modals	Regulative discourse median modals	Regulative discourse low modals
Facilitator use of modal verbs	(Nothing found)	(Nothing found)	-Okay, it **could** be, person, place, or thing. -Alright, so, look, we **could** do,	(Nothing found)	(Nothing found)	(Nothing found)

			with all the ideas, here is my idea, -and you **could** tell me more.			

The excerpt shown illustrates that in Text#4, the facilitator, by means of the low value modal verb “could”, offered the students a high degree of flexibility in choosing the persons, places, and things that they wanted to include in the video. Also, the facilitator used “could” to express flexibility with respect to the creation of the video (“do”) and to leave the door open for more ideas.

Finally, Table 18 is an excerpt of Analytical Table 4 as implemented on Text#4. As explained, this analytical table allowed examination of (1) the exchanges of propositions and proposals between the participants, (2) the positions offered to the students by the facilitator, and (3) their own self-positioning in instances where they construed themselves as agentive and in instances where they used commands to urge other participants to “do” CT-CP.

Table 18

Participant-*Speech Function in Both Instructional and Regulative Discourse: Proposals & Propositions, Authentic & Text Questions (Text #4)*

Interaction type Participant	Instructional discourse proposal (exchanges of action)	Instructional discourse proposition (statement) (exchanges of information)	Instructional discourse proposition (authentic question) (exchanges of information)	Instructional discourse proposition (test question) (exchanges of information)	Regulative discourse proposal (exchanges of information)	Regulative discourse proposition (exchanges of information)
The facilitator	-Give me your favorites.	-Okay, it could be, person, place or thing, -like the planets. -So, here we got these three. -Okay, a place. -A tropical island for Michael.	-What do we wanna do? -What's your favorite thing? -What's your favorite thing or things? -Alright, Michael, what's your favorite thing?	(Nothing was found)	-Let's go ahead and brainstorm. -Alright, so, look, we could do, with all the ideas, here is my idea. -And you could tell me more.	(Nothing was found)
Herminio	-Do something of outer space, like a planet or					

	-something, that would be nice. -We could make it small and then like white dots					
Juan	-(Let's do)[25] Something like a wither skeleton with an iron armor on it. -(Let's do it in) Winter -(Let's do) Ice -Like sports, gaming, - (Let's do) Minecraft.	-That's cool, but then like I wanted a skeleton like regular, though. -That's too complicated.	-The thing? -What's your favorite thing?			

[25] I have included within parenthesis meanings that apparently were omitted by the participants to facilitate the smooth flow of their interactions. In SFL terms they were using the textual resource called ellipsis (Halliday & Matthiessen, 2014). As discussed above, "Ellipsis makes it possible to leave out parts of a structure when they can be presumed from what has gone before. Ellipsis indicates continuity, allowing speaker and addressee to focus on what is contrastive…], (p. 606)

	- (Let’s do a) Wither skeleton.					
Michael	-How about this guy? -How about this guy? [Shows phone to J] -Minecraft -Snow. -How about this guy? -A place, a place, a place, a place, somewhere where it's warm. -Tropical island					

The language arts teacher	Nonpresent (not with the students at that moment)	Nonpresent	Nonpresent	Nonpresent	Nonpresent	Nonpresent

The excerpt reflects exchanges of proposals and propositions which are relevant to this dissertation. For instance, in using the command “Give me your favorites”, the facilitator positioned the students as providers of key information for the creation of their video. In doing so, she positioned Michael, Herminio and Juan as students whose prior experience mattered. That the students’ prior experience of the world mattered for the facilitator was also shown in her consecutive use of authentic questions: “What do we wanna do?” “What's your favorite thing?”, “What's your favorite thing or things?” and “Alright, Michael, what's your favorite thing?” where she elicited the places and things that the students wanted to see in their video. Her validation of the students’ proposals: “Tropical island” (Michael’s), “Do something of outer space,”\ like a planet or something (Herminio’s) in her propositions “A tropical island for Michael” and “like the planets” further illustrate this point. The final demonstration can be seen in the digital images created in Python, which will be displayed in Chapter 5 where the wither skeleton proposed by Juan can also be observed. Finally, the excerpt illustrates how the facilitator managed the teaching/learning activity, which is foregrounded in her regulative discourse activity. “Let's go ahead and brainstorm” “Alright, so, look, we could do, with all the ideas, here is my idea”, “And you could tell me more” are examples of clause complexes which were used by the facilitator to regulate the organization and pace of curricular activity rather than to focus on content (instructional register).

In sum, in this section I discussed and exemplified with Tables 15, 16, 17, and 18, the Analytic Scheme I used to advance the analysis of the fifteen significant texts selected with respect to RQs 1 and 2. The SFL-based toolkits which I compiled during the Exploratory Phase of the whole text afforded the creation of the analytic scheme and subsequent implementation.

The implementation

of the analytic scheme yielded qualitative and quantitative understanding that helped explain the nature of the CT-CP universe offered by the facilitator based on the nature of the processes and the modal verbs that she used, the *participants* that she involved and the circumstances she associated to the processes. However, the excerpts shown did not include relevant examples of the circumstances associated to the processes as realized in the logico-semantical connectors used between processes by the facilitator. These connectors are key since they realize meanings that concern, for instance time, cause, purpose, and condition. I will discuss these in the next section along with the *indicators of student learning.*

The recursive and flexible nature of this dissertation afforded a progressive understanding of the data. As I implemented the analytical scheme shown to the 15 text exemplars with the support of toolkits developed, I got deeper into the meanings that related to the nature of the experience, basically its content (field), and the exchanges of action and information and the way the students were positioned and self-positioned themselves (tenor) and ellipsis, which encoded *abstraction* (tenor). To complete the Analytic scheme adopted, I further focused the analysis.

Implementation of Focused Analysis.

In this final stage of the analysis, I implemented two cycles of coding to the fifteen significant texts selected. In doing so, I relied on the toolkits I had compiled in the Exploratory phase of the analysis, which concerned the three types of meanings (i.e., ideational or experiential, interpersonal, and textual) and the situations they created. I applied the coding on the Participant-speech function analytical tables as they allowed scrutiny of the logical meanings that link clauses as well as what and who propositions and proposals concern. In other words, the table facilitated scrutiny of who was saying what and whether it was a call for action (proposals) or an exchange of information (propositions). Also, conjunctive elements that realize meanings of time, cause, purpose, and condition were illuminated because of the organization of discourse in the table in clause complexes. In addition, the Participant-speech function analytical table illuminated when the students used proposals and when they used propositions, and what they concerned, as well as what they responded to questions and when they asked them (and the nature of the questions). Furthermore, the separation in propositions and proposals by participants illuminated student agency (use of the pronoun *I* and of commands) in relevant moments of CT-CP activity, an indicator of CT-CP learning.

In the first cycle of coding, I color coded first level codes with descriptive words that showed the situations that were realized. As recommended by Tracy (2019), I avoided any interpretation in this primary cycle coding, assigning words to them that 'captured their essence'. Some examples of codes that I used include "ownership of concepts" (realized with possessive pronouns (e.g., "your code," "your im_show"), "instrumentality" (realized with cause-result conjunctives such as "in order to") and "abstraction" (realized through processes of ellipsis and transformation of concrete entities into mathematical representations). An especially relevant example of abstraction, since it indicates student CT-CP learning, is "5,5; 12,17" which was realized as a command by one of the students during a modeling and coding activity

Then, in a second cycle of coding, I proceeded to "organize, synthesize, and categorize them [codes] into interpretive concepts" (Tracy, 2019, p. 214). However, prior to categorizing the codes, I aligned them to my RQs. To address RQ1 (the nature of the CT-CP universe offered), I used the codes that concerned *participants*, processes and circumstances and the flow of discourse. To address RQ2 (student positioning by the facilitator), I focused on the codes that concerned the use of commands and questions, whether Spanish was used, and the concepts used were accessible to the students. And to address RQ3, I focused (initially) on the use by the students of pronoun I and commands in the 15 significant Texts.

Once I had aligned the codes with the RQs, I proceeded to their categorization. I categorized the codes that related to the nature of the experience offered by the facilitator (RQ1) in the 5 dimensions that will be discussed in the findings chapter. The categorization process included the classification of processes and participants into the worlds (i.e., the students' personal and imaginative worlds, CP, mathematics, and image & video processing) where they, I assumed, proceeded from before they entered the CT-CP universe offered by the facilitator.

As for RQ3, drawing on Lavie et al. (2018), I focused on identifying instances when the students participated agentively in recognized practices which featured bondedness (the steps of a procedure feed into later steps that result in a final product), objectification (abstract objects substitute concrete objects and procedures), flexibility (there is more than one way to perform a task), substantiation (student explanations of what they did), and applicability (students application to different environments of what they learned). Once I had identified indicators of student learning, I strengthened my focus on abstraction. I researched the 15 texts and, occasionally, the whole text, to trace the history of objectifications of discourse such as Juan's "5,5; 12,17" (i.e., a two-dimensional array or piece of the asteroid of the video ready to be programmed in Python) to identify the situations that had led to such abstractions from their origin in spoken English. Necessarily, in their way from their discussion about asteroids and the objectification of an asteroid in two dimensional arrays, the students had had to consider different LOA. I intensified the focus on the Python program and digital image and video representations that were being produced by the students as discourse was used. Students' abstract utterances such as "5,5; 12,17" were proposals (commands) which certainly codified concrete objects and procedures. I reviewed the literature in search for relevant CT-CP practices and pedagogical strategies and learning progressions that could illuminate my data. I found that the history of the abstraction processes that my data revealed run parallel to recognized CT-CP recognized practices such as modeling, algorithm thinking and testing (Kong, 2019; Grover & Pea, 2018). Furthermore, I found that "5,5; 12,17" was a little step of the algorithm of the video which the students were communicating. In reviewing the data, I discovered that the facilitator and students of this study adopted a similar teaching/learning progression to the Lee et al's (2011) use-modify-create progression, where students use and modify ready-made programs as a preparation to create their own. Finally, I examined the students' explanations in their presentation of the video project in search of further indicators of CT-CP learning.

I had found commands such as "5,5; 12,17" through the SFL lens inherent to toolkit #2 and the separation of propositions and proposals which I had specially designed to address RQ2. I categorized the codes that related to these questions in three fundamental dimensions, which are also discussed in the findings chapter. They directly concern the facilitator use of commands and authentic questions to position the students as doers of computer programming whose prior experiences matter. The findings concerning RQ2 are intimately associated to the grammatical resources used by the facilitator in her interactions with the students (i.e., commands and authentic questions) during key CT-CP practices.

Tracy (2019) argues that “Of course, any group of codes, when combined in different ways, could answer any number of questions” (p. 193). I agree. While categorizing the codes that related to RQ1 was a relatively straight forward process closely based on the lexicogrammar used by the facilitator, complexity of interpersonal relationships and roles enacted allowed for different interpretations. I considered contrasting perspectives that I had gathered in my analytical and reflective notes, especially those that reflected my discussions around student positioning with my colleague Research Assistant Gulnara Kussainova. In turn, I reviewed my analysis critically, triangulating the understandings gained through primary sources with the secondary sources, paying special consideration to the views expressed by the facilitator. I reviewed the lexicogrammar, situations and Python program and digital image and video representations realized by my participants and confirmed the soundness of the story that I had identified. In Tracy’s terms, I had identified a story that deserved to be uncovered. As will extensively be discussed, it fundamentally concerned computational thinking and its communication.

Data reduction of secondary sources

I used SFL perspectives to identify in questionnaires interviews and facilitator field notes the key nouns and verbs which signaled the basic constructs of this study: Bilingual talk, collaboration, and agency. I collected all the participant clauses that included these constructs and essentialized them to assist the contextualization of the knowledge I gathered from the analysis of the 15 ignificant texts and whole text. Also, as mentioned above, constant retort to the student in-process code and digital images gathered by the minicameras set up in front of the monitors was invaluable to decipher understandings. The student prototypes were also invaluable.

To conclude this chapter, I display Table 19 with descriptors which assisted my examination of LOA in my data, then discuss an example that illustrates the SFL-case study perspectives adopted. I then finish with a discussion on validity and a summary.

Table 19

LOA Hierarchy with Discursive Descriptors

LOA	Descriptors to identify each LOA	Abstraction transition taxonomy (Cutts et al., 2012)	Abstraction through modeling with mathematics (Taub et al., 2012)
Problem	A short written or verbal description of a project	English	What is needed
Design	More detailed than the problem, but without referring to the code. It is a thought, written verbal or drawn depiction of the project.	CS Speak	What it should do
Code	The code itself or a description of the code using programming language specific vocabulary.	Code	How it is done
Running the code	Reference to the	Results	What it does

	output of the program.		

Adapted from Waite et al. (2018).

Based on the descriptors specified in the table, I was able to explore in the discourse of my participants the LOA they were working at and what was at stake (e.g., what is needed or the task is wanted at the problem stage) at specific times that I considered important. I considered that during the creation of the video, at the problem LOA, the participants discussed what they needed in order to achieve what they wanted to do; at the design LOA, they discussed specific technical aspects of the project, and designed prototypes; at the code LOA they discussed on the code itself; and at the running the code LOA, they discussed the output of the program under development.

An Example

Following, as an opener for the discussions that the following chapters comprise, I illustrate with an example the affordances of the SFL-based analysis implemented in this dissertation to respond to RQ1, RQ2 and RQ3. The example that I discuss below is an excerpt of Text#12 and was selected on the basis that it shows student agency in the CT-CP practices at hand. At this point, Michael had developed significant independence and agency in the CT-CP procedure adopted and used commands to address the facilitator, who positioned Michael as the provider of some key information that she needed.

I focus on a succession of interactions that happened between the facilitator and Michael during CT-CP curricular activity. My explanations are supported on the above-discussed three ideas: (1) each clause used by the facilitator contributed to her discursive representation of a specific CT-CP domain in AOLME, thus defining CT-CP (RQ1); (2) each clause used by the facilitator was an exchange of either information (a proposition) or action (a proposal) through which she positioned Michael (RQ2); (3) In turn, Michael also self-positioned himself (RQ3), positioned the facilitator in interaction and represented a CT-CP experience. Thus, the clauses used by the facilitator in the excerpt below, encoded, on the one hand, how she defined CT-CP as a discipline through the processes that she represented in her discourse (RQ1), and on the other, how she positioned Michael with respect to CT-CP as a receiver/provider of information or as a doer of CT-CP processes (commands and offers) (RQ2). And Michael, through the clauses that he used positioned himself (RQ3)

Participant student Michael (M) was programming the background of the video, more specifically its sky and dirt (its ground). The facilitator (F) (in black) approached him, took control of the laptop keyboard, and addressed consecutive questions to him to inquire about the Python program that he was developing. The following excerpt[26], illustrates the dialogue that they had (Lines 15, 340-15,373 of the whole text):

F: What are you trying to **do**, buddy?

M: **I'**m adding the land.

F: Which **number is** the **sky**?

isn't that **this one**?

M: The sky is down here.

F: **It's this one**, right?

M: No,

[26] Note that each line, be it produced by the facilitator or by Michael, has just one clause, as suggested by Halliday and Matthiessen (2014). Please, also note that in each clause one process (or verb) and one *participant* (noun or pronoun) was involved (circumstances not in all the clauses). In SFL, compound verb forms such as “trying to”, “be gonna” “ain’t gonna” , “wanna” count as just one process. In these examples “trying”, and “gonna” add to the process a meaning of intention, and “wanna”, a meaning of desire.

that's dirt

M: The sky is down here to here

F: Alright.

M: **Don’t delete it**

F: **I** ain't gonna **delete it**

M: **Don’t delete it**

F: **I** just wanna **check something**.

M Oh no, gonna kill it.

F: **I**'m not gonna **kill it.**

M She is gonna kill it.

She is gonna kill it.

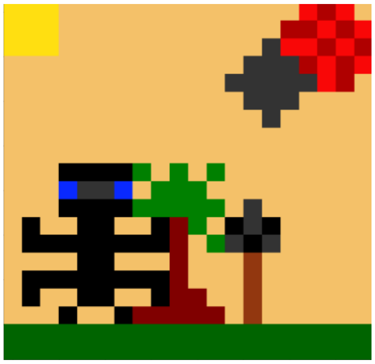
To address RQ3 (student learning) in excerpts such as this one, I coded signs of student discursive agency which indicate student development of CT-CP discourse and learning (Lavie et al., 2018; Lavie & Sfard, 2019). In this excerpt, I identified Michael’s agency in his use of the personal pronoun **I** and of the two commands “**Don’t delete it”**. In using the personal pronoun *I,* Michael involved himself in the process of adding the land to the program under development, a sign that he was involved in the modularization of the program that the students were creating. The land was a part of the background of the video and a part of each of the 4 of the 5 frames that composed the Python program and video developed by the students. Thus, Michael involved himself agentively in two CT-CP recognized practices, i.e., modularization and problem decomposition. I also coded student agency in Michaels’ use of the command “Don’t delete it” through which he positioned himself agentively and in authority with respect to the facilitator and the process “delete”. In insisting that the facilitator did not delete “it”, i.e., the land of the background, more specifically the part of the Python program that encoded the land of the background of the video Michael was also displaying his agency in CT-CP modularization and problem decomposition practices. Following are the lines of Python code at stake in the excerpt. After a hashtag, the word "sky" marked the start in the program developed by the students of the sand-like sky of the background of the video (See in Appendix E the whole Python code developed by the students).

```
#sky
im_fill(frame1,[7,8],[0,19],"f4c169")
#land
im_fill(frame1,[18,19],[0,19],"006400")
```

Figure 7 shows the image (frame1 or scene 1 of the video) on which Michael was working at the moment captured in the excerpt, which was produced when the program was completed and run. The Python line of code `im_fill(frame1,[7,8],[0,19],"f4c169")` including the hexadecimal number `f4c169`, which coded the sand-like color of the sky of the background, was the object of the controversy in which Michael realized his agency, thus, showed signals of CT-CP learning.

Figure 7

Digital Image of the Background of the Video and Its Characters



To address RQ1 (nature of CT-CP) in excerpts such as "I'm adding the land, don't delete it," I identified, coded and categorized the nature of the processes used by the facilitator based on the system of process type (Halliday & Matthiessen, 2014) and who or what she construed as agentive based on the categories personal & imaginative, mathematics, CP and image and video processing. In the excerpt just shown the facilitator used the relational process **is** and the material processes **do**, **delete**, **kill** (which was used to mean "delete") and **check.** Also, she involved in said processes the *participants* **it** and **one** (pronouns that signaled a part of the Python program under development), **number** (the hexadecimal number that coded the color of the sky of the background of the video) and **sky** (the expected digital result of the program once completed). The imaginative world of the students (the sky was part of a tropical land that Michael had envisioned as the setting where the video story would take place), mathematics (hexadecimal numbers) and CP (the Python code) were represented in the discourse of the facilitator. In doing so in this example, the facilitator (and Michael) was using four LOAS: (1) the execution level (i.e., the digital image of the sky), (2) the Python code that programmed the sky, (3) the hexadecimal number that coded the color of the sky and (4) the English language (i.e., the word "sky"). The LOAS and processes, *participants,* and circumstances used by the facilitator in this excerpt, in combination with many others in other significant excerpts selected (see below), yielded patterns which signaled the nature of the CT-CP universe that the facilitator offered the CLD students of this study.

The separation made into propositions (statements and questions) and proposals (commands and offers) and the use of toolkit #3 to address RQ2 (positioning students) in excerpts such as the one shown above was instrumental. In this excerpt, the facilitator only used propositions, thus inviting Michael to participate in exchanges of information. The facilitator's consecutive authentic questions "Which number is the sky?", "Isn't that this one?" and "It's this one, right?" positioned Michael as the provider of key information that the facilitator needed at that moment of CT-CP curricular activity, which I coded. Again, analysis of propositions and proposals at significant moments of curricular activity yielded patterns which signaled how the students were positioned in terms of their participation and role in CT-CP practices.

Excerpts such as "I'm adding the land, don't delete it" illustrate the power of SFL methods to explore CT-CP teaching/learning practices in the AOLME CoP environment, where they happened. In the excerpt, the facilitator and Michael participated in the three dimensions that define practice in the AOLME as a CoP, that is, mutual engagement, joint enterprise, and shared repertoire (e.g., Wenger, 1998, 2010). The mutual engagement shown by the facilitator and Michael in the practice of creating the video was apparent; their use of a shared repertoire, that is a specific CT-CP discourse which involved different LOAs and specific processes and elements of very different nature too, and the joint enterprise of creating a video for CT-CP educational purposes define the nature of AOLME's practices as well. In turn, the dimensions of the practice enacted by the participants was determined by the AOLME environment.

Validity

Gee (2014) argues that the validity in the analysis of any piece of language in use resides in the extent to which researchers are able to broaden their knowledge about 'context' (I have used here the word "context" in its most typical sense, as Gee does). He states, "We cannot really argue an analysis is valid unless we keep widening the context in which we consider a piece of language until the widening appears to make no difference to our interpretation" (p. 75). I think that the combination of the following has contributed to widening my knowledge of the AOLME environment and of the broader environment around it in light of SFL and case study methods: (1) My privileged standpoint as an AOLME Research Assistant in the period 2016-2019 which allowed me to learn about AOLME's work and perspectives, and my participant observant role during the implementation that concerns this study and (2) My thorough review of current trends in CT-CP and mathematics educational research and related literature that concerns communication, thinking and abstraction processes.

Summary of Chapter Four

In this chapter I described the combined SFL-case study methodology I used. I showed the power of SFL perspectives and tools to analyze spoken interactions and the soundness of combining said perspectives with case study perspectives and methods to broaden perspectives, help decipher discourse and contextualize understanding through an ethical-equity lens. I also discussed the coherent selection of participants based on key theoretical constructs and motifs of this study: Bilingual talk, collaboration, and agency. I illustrated the interrelationship between the lexicogrammar used by my participants in their clauses, the meanings that they realized and the situations they created to construe experience, enact relationships and positionings and co-construct a cohesive text that I could explain as a researcher. I also discussed the ‘loose analysis’ approach I adopted and the importance of consideration of macro perspectives to contextualize and broaden understanding. This approach helped in the selection of relevant texts (events) for deep SFL analysis in close consideration of the Python program and digital image and video representations that were developed in triangulation with secondary sources. The chapter included detailed discussion on the Analysis method adopted, which comprised two differentiated phases, relied on and SFL-based Analytic tables and toolkits tailored to the specificities of this study.

Chapter 5

Findings

In this chapter I present the findings of this dissertation. I have organized the chapter according to its RQs, which I list following:

RQ1) What is the nature of the computer computational thinking-computer programming (CT-CP) universe that a CT-CP facilitator of a CT-CP community of practice (CoP) offered her CLD students by means of oral discourse?

RQ2) How were these students positioned in terms of their participation and role in CT-CP practices?

RQ3) Can the CT-CP procedure adopted to teach CT-CP facilitate student learning of CT-CP?

This SFL-based dissertation draws fundamentally on the spoken discourse used by its participants. It was motivated by the pressing need to better understand how teachers can deliver computational thinking practices through computer programming (CT-CP) in ways that welcome (or not) the active participation of cultural and linguistically diverse (CLD) students and all students to CP-CT practices. To further understanding on this matter, I focused on the language used by a CT-CP facilitator in her interactions with a small group of three middle school CLD students and their bilingual language arts teacher (All Spanish-English bilinguals). Since, during the analysis, I identified indicators of student CT-CP learning, I extended my focus to the discourse used by the students as they interacted to create their Python program and digital image and video representations with the guidance of the facilitator. What follows is an explanation of what was achieved by my participants based on the oral discourse that they used—i.e., the lexicogrammatical resources they used—and the Python program and digital image and video representations that they created while engaging in a curriculum of computer programming-mathematics-image and video processing integrated activities. The curriculum targeted the eventual goal of programming in CP language Python an animation video from the pixel level. The study was situated in a rural US Southwest middle school in which more than twenty CLD middle school students participated voluntarily after their regular school classes in a CT-CP education-oriented CoP called AOLME. One fundamental AOLME objective, among an extensive research agenda, is to support interactive learning in engineering and mathematics related activities of middle school students, especially from underrepresented groups. The students, Herminio, Juan and Michael (pseudonyms) were given the chance to choose their preferred facilitator, Teresa (pseudonym), a student of engineering in her third year and CLD student herself. A fifth participant, the students' bilingual language arts teacher, with no previous experience in CT-CP, became key in the group since her participation fostered the use of Spanish. All five participants were bilingual in the sense that they were able to participate actively in bilingual social activities. The CP facilitator, a university student of engineering in her early twenties, declared having only some experience in teaching, though not professional. AOLME provided her and the rest of AOLME facilitators with a few sessions of professional development and follow-ups, which basically centered on the promotion of communication and collaboration-based pedagogies (Chapin et al., 2009; O'Connor & Michaels, 2019) that support learning and student equal opportunities of participation (Cohen & Lottan, 2014). For this reason, I always refer to her as the facilitator (or as Teresa) as opposed to referring to her as a teacher. The facilitator declared the importance of promoting student collaboration and participation, and of allowing the students to communicate in the language they felt most comfortable. Teresa introduced her small group of CLD students to a bilingual CT-CP experience in a universe which was new for them. Only Michael reported some experience in programming, although minimal.

This study hinges on the processes that the facilitator and the students used through time, the *participants*[27] that they involved in the processes and the circumstances that they associated with the processes. And, on the way that the facilitator positioned the students and the agency that they displayed in the CT-CP practices provided. Accordingly, the explanations that follow are based on the lexicogrammatical resources used by the facilitator in the 15 selected text exemplars examined[28] as revealed in the *participant*-process analysis. To contextualize and frame my explanations, I use the Python program and digital image and video representations created by the students and secondary sources.

What is the nature of the computer computational thinking-computer programming (CT-CP) domain that a CT-CP facilitator of a CT-CP community of practice (CoP) offered her CLD students by means of oral discourse?

[27] Let me recall that I use *participant* in italics to refer, in grammatical terms, to the element of the clause (e.g., asteroid, you) without which it is not possible to realize whatever process (e.g., gets bigger, calculate).
[28] Occasionally, to add evidence, clauses or short excerpts were used from outside the significant 15 Texts that constitute the basis of the analysis.

In this section I describe the nature of the CT-CP universe offered by the participant facilitator based on the processes that the facilitator used to deliver curricular activity, the *participants* she involved (their qualities included) and the circumstances which she associated with the processes. The characteristics of the bilingual CT-CP domain or universe offered by the CT-CP facilitator to her small group of CLD students comprised 5 dimensions: (1) Complexity; (2) Pragmatism; (3) Strategy; (4) Dependency; and (5) Flexibility. The main *participants* (agents) involved in CT-CP activity were the participant CLD students who shared prominence with abstract entities such as two-dimensional mathematics arrays and Python functions. The students were involved by the facilitator mostly in material processes of doing, in other words, in hands-on CT-CP practices.

Importantly, throughout the CT-CP curricular activity the facilitator allowed free use of Spanish or English, typically responding in Spanish when she was addressed in Spanish and in English when she was addressed in English. While English was far more common, Spanish had a significant presence which was recurrently triggered by students' language arts teacher use of Spanish. In broad terms, following the categorizations used by Dagienė et al. (2017), the facilitator offered the CLD students a bilingual CT-CP universe which centered mostly on "Algorithms and programming" (e.g, algorithm, loop), and to some extent on "Data, data structures and representations" (e.g., array, string) and "Computer processes and hardware" (eg., memory, CPU) rather than on "Communication and networking" or "Interaction systems and society". The facilitator, therefore, did not focus on aspects of CT-CP concerning client/servers and computer networks or ethics and social issues, but on CT-CP that aimed at the creation of an algorithm which was eventually written in Python to enable a digital video.

While it was apparent that the facilitator engaged the CLD students in CT-CP practices, it took careful analysis of the data to understand so. *Discursively speaking, computational thinking remained hidden.* Many fundamental CT-CP keywords that define CT-CP concepts and practices, which were discussed in Chapter 2 (Cf., Dagienė et al., 2017; Kong, 2019), were never used by the participant facilitator or the students. Still, not only did the facilitator offer her CLD students a rich CT-CP universe, but she positioned them as competent computational thinkers and programmers (RQ2) who developed CT-CP (RQ3) as will be shown in subsequent sections. While *participants* such as "computational thinking," "abstraction,"\ "encapsulation," "procedure," "modeling," "computational artifact," "iteration," "modularization," "execution," "recursion," "evaluation," or "automation" were never pronounced, they all were there. As for *participants* which are specific to computer programming such as "conditional statement", "for loop" and "variable" they practically disappeared from discourse once the creation of the video started.

Main agents: The CLD students

The CLD students of this study were construed by the facilitator as the main agents of the CT-CP universe that they were offered. The prototypes that they designed and the Python code and the digital images and video that they produced constitute tangible evidence of their agency. In addition, the facilitator involved them in about 65% of the processes that she used. The significance of this finding is supported by Vähäsantanen & Eteläpelto (2017) who found that learning and agency are closely intertwined in software development in the workplace. The facilitator construed a CT-CP universe where the most common actors where human agents rather than abstract objects (non-human agents) and their relations as was found in science

(Halliday & Webster, 2004) and in mathematics (Morgan & Sfard, 2016; Sfard, 1991, 2008; Veel, 1999). As can be observed in Figure 8, the facilitator-process (focus on *participants*) analysis revealed that the facilitator continuously involved the students in activity, mostly by means of the pronouns *you* (about 35%) and *we* (about 27%).

Figure 8

Distribution of Participants Involved in CT-CP Processes in the Whole Curricular Activity

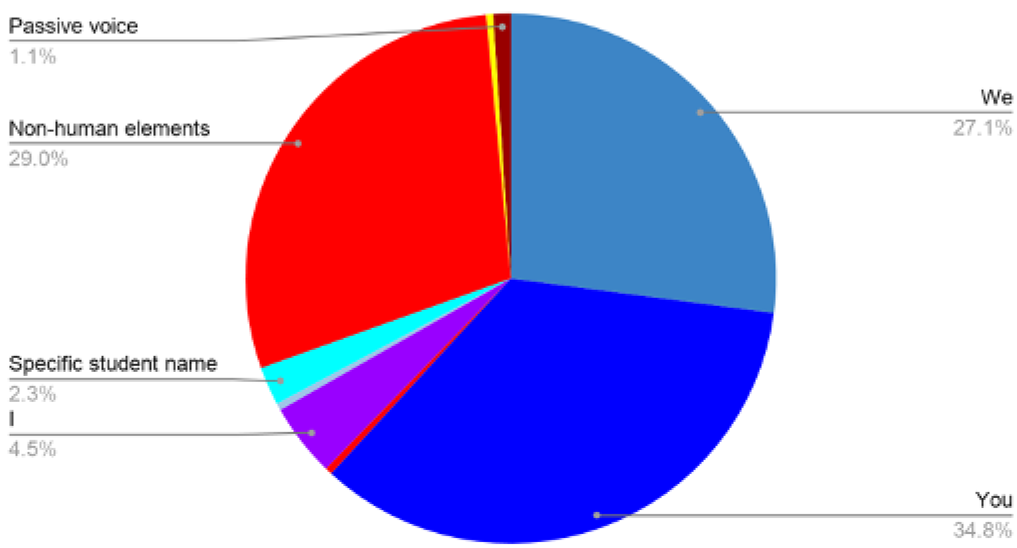


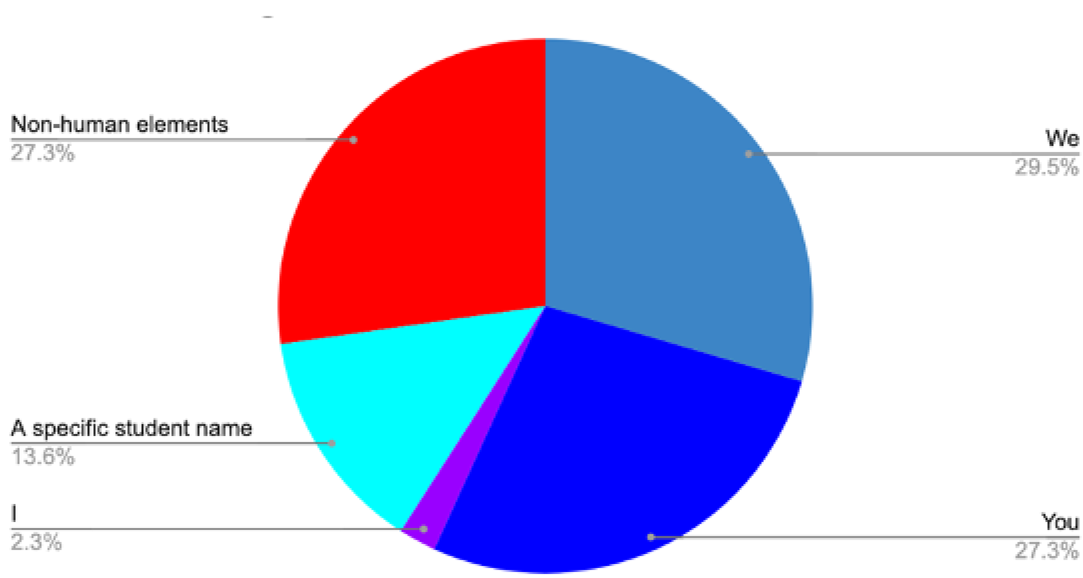
Figure 8 shows the distribution of *participants* involved in CT-CP processes by the facilitator. The *participants* most used by the facilitator to involve the students in CT-CP activity were "you" (around 27%) and"we[29]" (around 29.5%). These pronouns typically referred to the students ("you"), and to the facilitator plus the students ("we"). Her prominent use of the

[29] Even though the pronoun *you* can also be used in a general sense (Rowland, 2000) and the pronoun *we* to involve people outside the group (Pimm, 1987), for the most part, the *participants* "you" and " we" were used to involve the students in activity.

pronoun *we* to include the whole group in activity constitutes evidence that the facilitator promoted collaboration (Pimm, 1987). The significance of this finding is supported with Vähäsantanen and Eteläpelto (2017) who showed that individual and collective learning are typically embedded in professional organizational aims and strategies. To a much lesser extent (about 2.5 %), the facilitator used the students' first names to summon up a specific student involvement in activity when inclusion seemed particularly important. This percentage was significantly increased by the facilitator in the task orientation stage as reflected in Figure 9.

Figure 9

Participants Involved in CT-CP Activity by the Facilitator in the Task Orientation Stage



The facilitator typically made sure to include the ideas of all the students in CT-CP activity. She did so in key instances of CT-CP activity, mostly in the task orientation stage, where the decisions concerning the narrative of the video were made. To do so, the facilitator used *vocatives*, i.e., the names of the students, to involve specifically one of them in activity. Teresa did so in about 14% of the clauses she used in the task orientation stage as compared to the approximately 2% of the clauses where she used vocatives in all 15 text exemplars. Following are two examples which happened during the brainstorm of ideas about what story to tell in the video: “What do you wanna do, **Herminio**?” (line 10,781) and “**Michael**, what's your favorite thing?” (line 10,891),

seemingly, to make sure that each of them had a say as to what to include in the students’ collaborative video. In the analysis, I took into consideration who and what were construed as agentive by the facilitator, and to whether agency was alienated. That is, agency “was free from human presence” (Morgan & Sfard, 2016, p. 107). Figure 10 reveals that the facilitator involved human agents in activity in about 70.3 percent of the processes that she used[30]. The facilitator obscured human agency by means of the use of the passive voice, where she did not specify any human agent, and by construing objects[31] such as numbers or variables as agentive. Figure 10 captures the distribution of agency and alienation of agency realized by Teresa in the 15 text exemplars selected. Interestingly, the percentage when the computer was construed as agentive by the facilitator amounted to only 0.3% as opposed to an overwhelming presence of human agency (70.3%). In sum, *the facilitator construed the students as the main agents of the CT-CP*

---

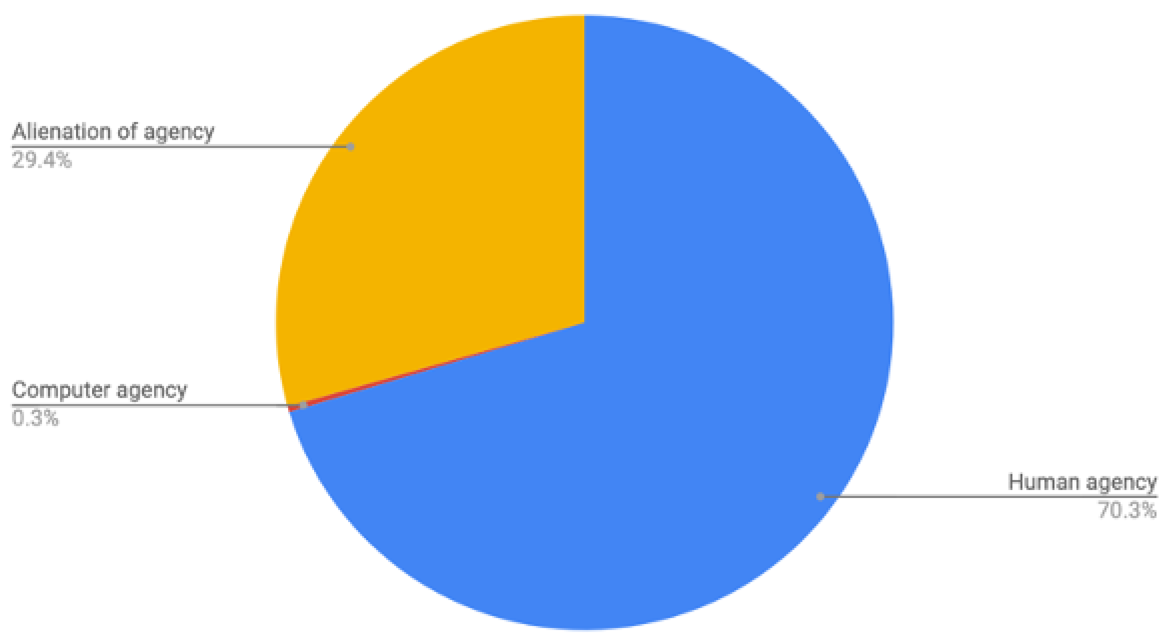
[30] Only a few times did the facilitator involve other human agents outside the study participants (in around 0.4% of the clauses she used)

[31] The following clauses, which happened when the facilitator was introducing the CP variable integer to the students, illustrate this point: “You got the first one || ‘which is an integer and || it’s referred to as an int and || it can represent positive or negative values”. In the clause “it’s referred to as an int” the facilitator used the passive voice, not specifying who refers to integers as “int”, obscuring this way the agency of the process “refer”. In the clause “it can represent positive or negative values” the facilitator construed “it”, i.e., “integer” as the agent of the process “represent”, obscuring this way human agency too.

universe that they were offered, which may contribute to the students seeing themselves as potential computational thinkers and computer programmers.

Figure 10

Human and Computer Agency and Alienation of Agency



Herminio, Juan and Michael's video was inspired by the popular video game Minecraft (cf. Karsenti & Bugmann, 2017) and in elements from the outer space, namely an asteroid. The video constitutes the students' fundamental contribution to one key AOLME's CoP joint enterprise, i.e., that the students produce computer programs and digital image and video representations . In Minecraft, "the second most popular video game of all time, with over 100,000,000 copies sold" (p. 1) players create virtual environments with building blocks. The referred virtual environments feature a variety of characters and elements which include hostile wither skeletons and tools such as shovels, which provide players with extra powers.

The facilitator CT-CP universe offered to the students comprised conceptualizing, designing, modeling, programming, testing, and debugging practices that resulted in the creation of a digital video in Python which featured the referred wither skeleton ("the spider") and shovel who are on a tropical island and are destroyed by the asteroid. Following I present the five dimensions of the bilingual universe offered to the students.

Complexity

The facilitator presented to the students an essentially complex CT-CP universe in that it was interdisciplinary and abstract. Teresa intertwined in her discourse processes and *participants e.g.,* video characters, concepts, and tools from four different worlds i.e., the student's personal & imaginative world, the world of computer programming, the world of mathematics and the world of image/video-processing (Adapted from Bernstein (1973)). Significant examples of processes used by the facilitator are "draw," "add," and "code," which continuously defined relevant participant interactions in fundamental CT-CP practices such as designing, modeling, testing, and debugging (Kong, 2019). Table 20 includes some of the most representative processes used by the facilitator.

Table 20

Examples of Bilingual Processes in Their Original Worlds

	The students' personal and imaginative world	The world of computer programming	The world of mathematics	The world of image/ video-processing
Processes (verbs)	squish, digest, disappear, get bigger, make bigger	code, test, *probar* (test) run, *correr* (run), *cambiar* (change), change, debug	add, multiply, convert, calculate	draw, make bigger, play, *moverse* (move), move

Table 21 further illustrates the interdisciplinarity of the bilingual CT-CP universe offered by the facilitator to the CLD students in a representative collection of the *participants*[32] she involved in activity.

Table 21

[32] Many key *participants* were Spanish (e.g., "nosotros" (we)). Also, many *participants* such as "color" and "variable", which presumably would have been classified by the students in just one world ("color" in their personal & imaginative and "variable in the world of mathematics"), were used by the facilitator (and as activity progressed by themselves too) in more than one world. I included in all the worlds of the table the *participants* "you", "we", "tú" (you), "you", "we" and "nosotros" (we, us) because they were used in all four worlds from the start of curricular activity.

Examples of Bilingual and Non-Human Participants *in Their Original Worlds*

	The students' personal and imaginative world	The world of computer programming	The world of mathematics	The world of image/ video-processing
Participants (nouns and pronouns)	you, we, tú (you), nosotros (we, us) asteroid, dust, land, sun, wither skeleton, shovel, fin (end), end, cereal, mom, *cabeza* (head), head, *árboles* (trees), trees, *color* (color), color, step, "13 to 15", "11, 11"	you, we, tú (you), nosotros (we, us), ayuda (help), help, variable, function, algorithm, integer, float, string *paréntesis* (parenthesis), parenthesis, the code, computadora (computer), computer, im_show, im_fill, frame_list, fps *color* (color), color	you, we, tú (you), nosotros (we, us) variable, integer, algorithm, function, parenthesis, coordinate, axis, number, number, *color* (color), color, step	you, we, tú (you), nosotros (we, us) crayon, pencil pixel, frame, *color* (color), color

The *participants* gathered in Table 21 were key in the bilingual construction of CT-CP curricular activity that led to the student creation of their Python program and digital image and video representations. The table includes mathematical abstractions such as "13 to 15", "11,11", which were fundamental since they represented and coded segments of elements of the video (i.e., segments of the asteroid) in 20 by 20 squared grids and constituted the algorithm that the students wrote in Python. Table 21 reflects that the discourse of the facilitator featured an evident presence of non-academic participants from the students' imaginative and personal worlds (“non-academic”) such as “asteroid” and “wither skeleton”. Non-academic *participants* shared instructional field with academic *participants* from CP, mathematics, and video & image processing world such as “im_fill”, “coordinate” and “pixel” respectively. In other words, non-academic *participants* such as “asteroid” and its abstract academic version “13 to 15”, “11,11” turned out to be key since the latter became constitutive of the algorithm and Python program that enabled the students’ video.

Table 22 includes excerpts of the facilitator discourse where she used key *participants* such as the ones just mentioned and reference to the significant text where they were used, classifying them in human and non-human types in four different worlds.

Table 22

Examples of Human and Non-Human Participants *Involved in Activity by the Facilitator*

Participant type	Participant	Excerpt	Text #
Human *participant*	You (Michael)	“Okay So here, Michael, this is where ***you*** are gonna add the framelist.”	9
Human *participant*	We (The facilitator, Herminio, Juan and Michael)	“Maybe ***we***'ll do the background and program it today.”	5
Human *participant*	I	“So, you know how ***I*** had told you guys about the inverse?”	2

Non-human personal-imaginative world *participant*	Asteroid	"Your ***asteroid*** gets bigger."	12
Non-human computer programming world *participant*	Im_show	"Put ***im_show***...so it's telling it to show, right"	15
Non-human mathematics world *participant*	13 to 15; 11, 11	**"13 to 15; 11,11"**	14
Non-human image-video processing world *participant*	Frame (Digital image)	"And that is only the first ***frame***"	12
The computer	It	"Then, you run it and see what ***it*** does." (It references the computer)	3

The table further exemplifies the abstract and interdisciplinary complexity delivered by the facilitator in that a segment of the *asteroid* such as the one represented by the mathematical array *"13 to 15; 11,11"* composed a *frame* (the first frame) that could be shown provided that the Python function *im_show* told it (the computer) to show it.

The facilitator consistently conceded ownership of CT-CP concepts such as "im_show", "code" and "frame" to the students through possessive qualifiers such as "your" and "our" (e.g., "your im_show," "our code"). This use of possessive qualifiers adds to the fact that the students were the architects of their own Python code and video which they created based on their own ideas as will be demonstrated. This is important because ownership has extensively been argued to be an essential aspect of college readiness (Conley & French, 2014) and distinguished as key in CT-CP processes such as abstraction, automation, and analysis (Lee et al., 2011). Along with these qualifiers and other that account for abstraction (e.g., hexadecimal number f90707), Table 23 displays relevant qualifiers such as "this" and "that" which were used by the facilitator pervasively and situated CT-CP activity in the here and now (Halliday & Matthiessen, 2014). Methodologically speaking, their use justifies by itself the use of video recordings, and student computational artifacts to decipher what these qualifiers referred to, thus resort to case study methods.

Table 23

Examples of Bilingual Participant *Qualifiers Used by the Facilitator*

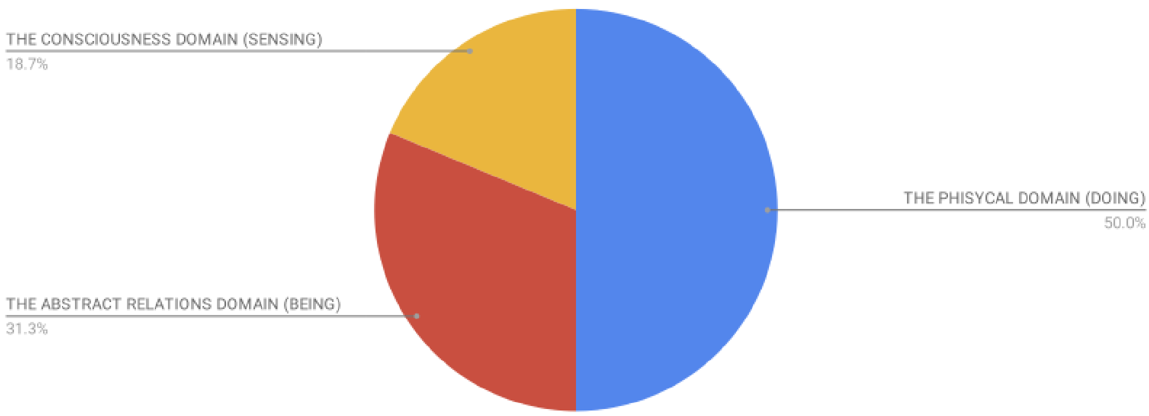
	The students' personal and imaginative world	The world of computer programming	The world of mathematics	The world of image/ video-processing
Participant qualifiers (Adjectives, determiners)	Your, our, this, that, red, favorite, *verde* (green),	Your, our, this, that, red, f90707, *verde* (green), green, 008000, hexadecimal	Your, our, this, that hexadecimal, red, f90707, *verde* (green),	Your, our, this, that, red hexadecimal, red, f90707, *verde* (green),

Pragmatism

The facilitator-process (focus on processes) analysis conducted on the 486 processes used by Teresa in the 15 text exemplars selected revealed that *the CT-CP universe offered by the facilitator to the CLD students was essentially pragmatic* in that it was *action-oriented (mostly student-driven), product oriented; tool-mediated; accuracy-based, and efficiency based.* While the discourse of the facilitator revolved around aesthetics elements such as the video background and character shape and color, it emphasized practical means and results. The students, who as was discussed were construed as the main agents of CT-CP activity, were offered an action-oriented CT-CP universe dominated by material processes of doing (about 50 percent of the processes) over processes of sensing or processes centered on abstract relationships. A dominance (about 50 percent) of material processes of doing (e.g., “draw,” “put,”, “run,” “add”). as opposed to processes where things happen (as in “the rocks formed” (Halliday & Matthiessen, 2014)) was found. This demonstrates discursively the hands-on nature of the CT-CP universe offered to the CLD participant students. This finding is consistent with researchers’ call to provide students with hands-on experience in preparation to CT related professional fields (Marquardson & Gomillion, 2018). Figure 11 displays the distribution of processes offered by the facilitator in the three domains of human experience (Halliday & Matthiessen, 2014, p., 216).

Figure 11

Distribution of Processes by Nature in the Universe Offered to the CLD Students

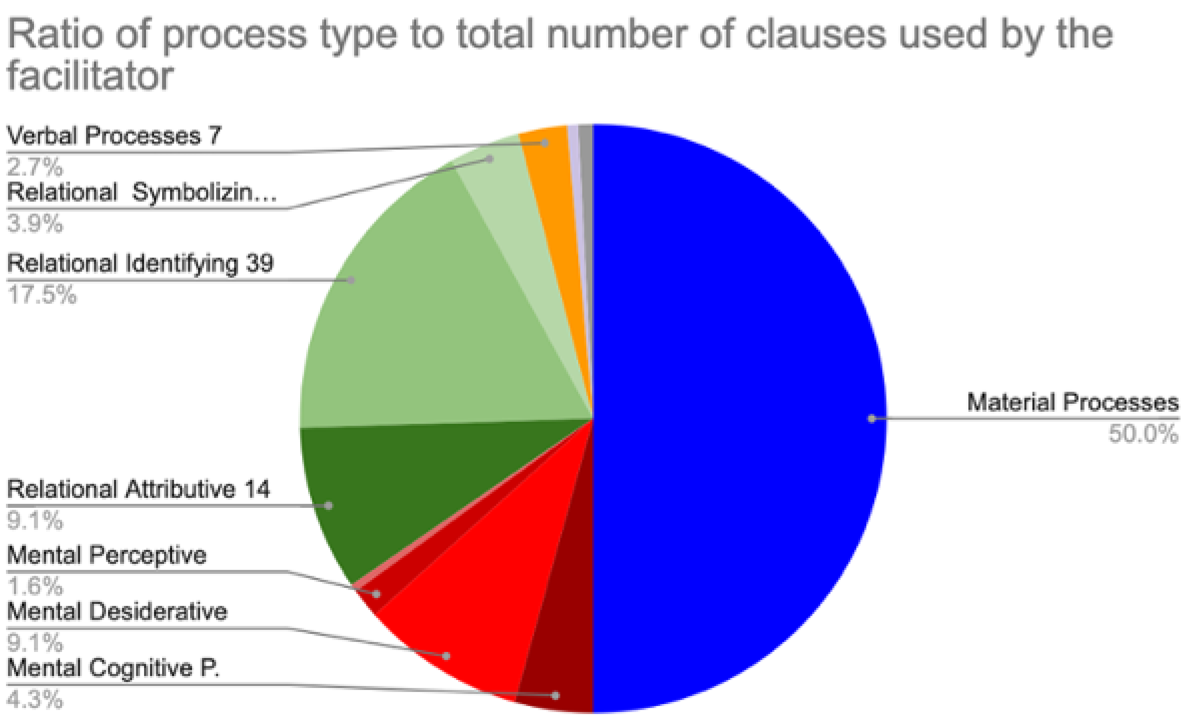


To a lesser extent, the universe offered by the facilitator included abstract relationships between *participants* (about 30 percent) in relations involving entities such as numbers, numerical operations, and variables. "L is your friend's number" (line 1,341) would be a good example, which was used when the facilitator was using a letter as a variable in an equation that helped constitute the first program made by the students. Thus, *participants* of the 4 above-referred worlds were involved in relational processes, more specifically in processes of "being" and "having" used to describe abstract entities. The world of consciousness (about 19 percent) was also represented in the facilitator's discourse with a stronger presence of desiderative (i.e., processes that concern desires) over cognitive processes.

A more delicate *participant*-process analysis (Halliday & Matthiessen, 2014; Schleppegrell, 2012) yielded a finer picture of the type of processes in which the facilitator involved the *participants,* including participants Herminio, Juan and Michael. Figure 12 shows that the facilitator devoted to mental processes about 15 percent of the processes she used in her discourse. However, the most prominent mental processes she used concerned the sphere of desires (about 9 percent). In other words, the facilitator used processes which concerned what the students wanted or liked more often than their thinking (about 4 percent of the total process), perceiving, or feeling (almost nonexistent). Process “want” was used by the facilitator consistently to elicit students’ desires in questions regarding key aspects of curricular activity such as the story that the students wanted to tell, and the comments they wanted to include in the program developed in Python.

Figure 12

Ratio of Process Type to Total Number of Clauses Used by the Facilitator



"What do we wanna do" (line 10, 962) is an illustrative example of the use of desiderative process 'want' used by the facilitator in an instance when she was inquiring the students' desires concerning the video project. It is important to recall that the fact that the facilitator resorted to cognitive processes in a 4 percent of the clauses does not mean that the students engaged in cognitive activity only in a 4 percent of the CT-CP activity that they were offered. It just means that the CT-CP universe offered by the facilitator discursively included cognitive verbs such as "think" and "see" in 4 percent of her clauses, centering more often on what the students wanted to do and on action-oriented activity, as will be shown following. More specifically, in response to RQ3, I will demonstrate that the students developed and used computational thinking constantly in the CT-CP practices that they were engaged in by their facilitator, especially in the creation of the video in Python. Regarding relational processes, the facilitator used them mostly to identify *participants* (About 17 percent) in instances such as "these are the most important variables" (line 696) when she was introducing the computer programming variables that would be used by the students. As for verbal processes, typically realized by verbs such as "say," "tell," or "explain," they happened in about 3 percent of the clauses used. A salient example would be "Do one of you want to *explain* that one to him?" when the facilitator was encouraging Herminio and Juan to explain to Michael a decimal to binary number conversion, in the stage devoted to preparing the students for their video project (task preparation stage). This example evidences the use of "talk moves" which the facilitator used to encourage the flow of communication between students (Chapin et al., 2009; O'Connor & Michaels, 2019).

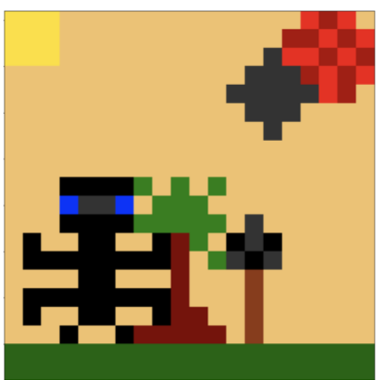
The facilitator's prominent use of material processes (about 50 percent; see Figure 12) typically engaged the students in CT-CP activity that resulted in the *reification of student ideas of their imaginative world into* Python program and digital image and video representations, significantly, Python code and the resulting digital images. *The pragmatic CT-CP universe offered by the facilitator features a prominent use of tools.* Function *fps* constitutes an example of the tools that mediated the development of products in this tool-mediated universe. Teresa facilitated the use of this tool in Text # 9 when Michael was on the computer keyboard. The students had just finished coding frame # 2 of the video (See Figure 13 below) and wanted to see the above-mentioned flipbook-like motion effect as the asteroid approached (Note the asteroid's bigger size as compared to the asteroid in frame1). Next is the dialogue that happened between the facilitator (F) and Juan (J) as the facilitator was guiding Michael's coding (Michael made no comments) (Text #9):

F: There you go,
now enter,
and you are gonna **put fps,**
what that means is frames per second.
J: Mhm
F: Just put a 1 for now,
that way we **see it**
run,
and then after that we can
J: like keep going over and over and over,
speed it
F: Yeah

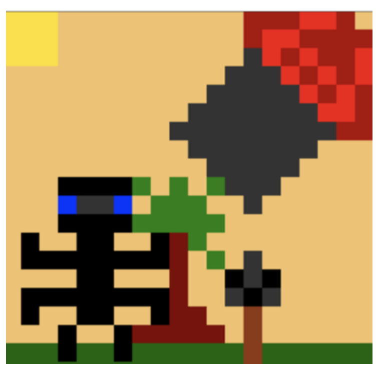
This short dialogue shows lexicogrammar in action, specifically, the conjunctive binder "that way" signaling the purpose associated with the processes "put", "see" and "run". The binder "that way" in association with noun *fps* and the mentioned processes exemplifies the pragmatic and tool-mediated nature of the CT-CP universe offered by the facilitator. Other binders that realized a meaning of purpose such as "to," "in order to," and "so that" were pervasive in discourse of the facilitator. Tools in this universe included the Raspberry Pi, laptops, cellular telephones, sticky notes, pencils, crayons, square paper grids, several Python computer programming functions and a color-picker application which was used to identify color codes in hexadecimal number notation. Figure 13 displays the 5 frames developed by the students to produce their digital video.

Figure 13

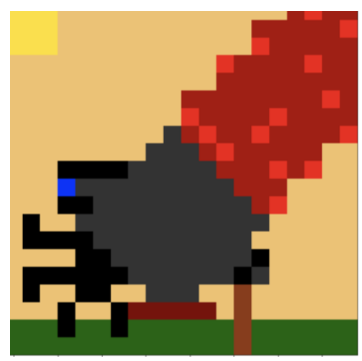
The Five Video Frames of the Students' Video



FRAME 1



FRAME 2



FRAME 3

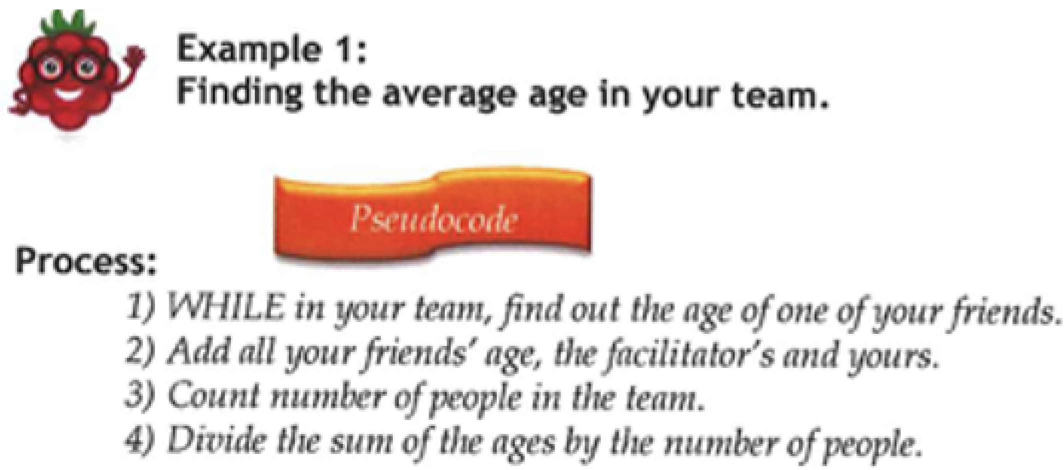
 FRAME 4	 FRAME 5	

As can be observed, the asteroid approaches (frames 2 and 3) the other characters of the video i.e., the blue-eyed wither skeleton and the shovel that can be seen on the right of the green tree. As a result of the collision, the characters and the tree disappear (frame 4). The video concludes with FIN (The end) in big red letters (frame 5).

In preparation for the creation of the video, the facilitator engaged the students in CT-CP practices through a discourse that includes adjectives such as “precise,” “right,” and “wrong,” which signal accuracy. In the preparatory stage, Teresa introduced the key computer programming concepts “algorithm, “flowchart,” loop,” control statements,” and “conditional control statements” (Kong, 2019) in exercises centered on the students’ everyday personal experiences. For instance, the facilitator engaged the students in creating their own algorithm (Figure 14) based on the steps needed to calculate their team’s average age.

Figure 14

Average-age Pseudocode Provided to the Students


Example 1:
Finding the average age in your team.

Pseudocode

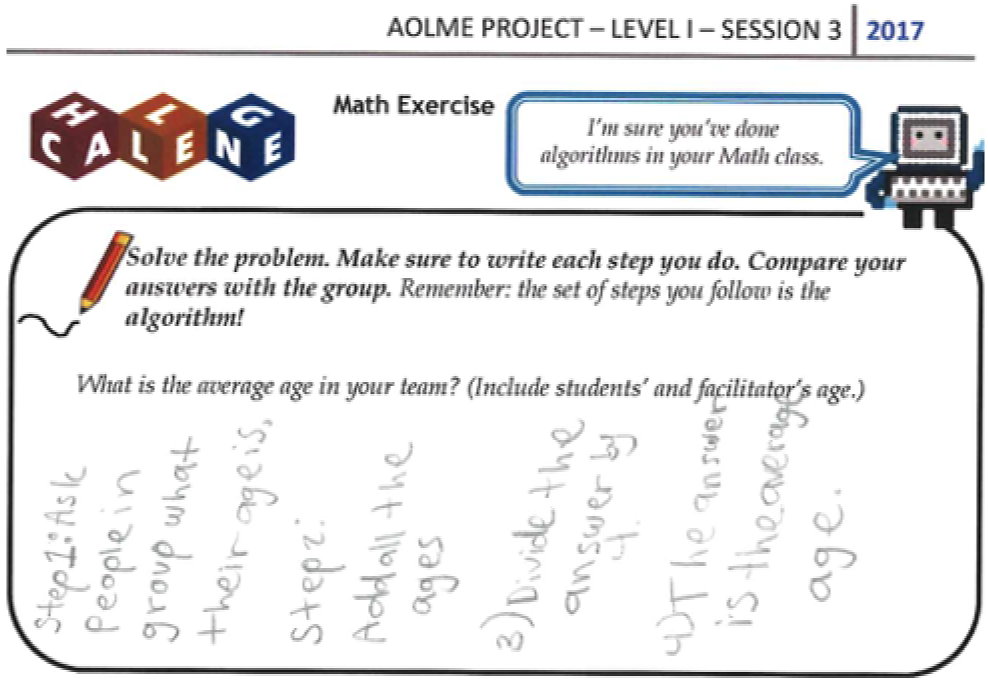
Process:

1) *WHILE in your team, find out the age of one of your friends.*
2) *Add all your friends' age, the facilitator's and yours.*
3) *Count number of people in the team.*
4) *Divide the sum of the ages by the number of people.*

Figure 15, below, shows a picture of Herminio's binder which illustrates in his handwriting the steps his group agreed were necessary to calculate said average age.

Figure 15

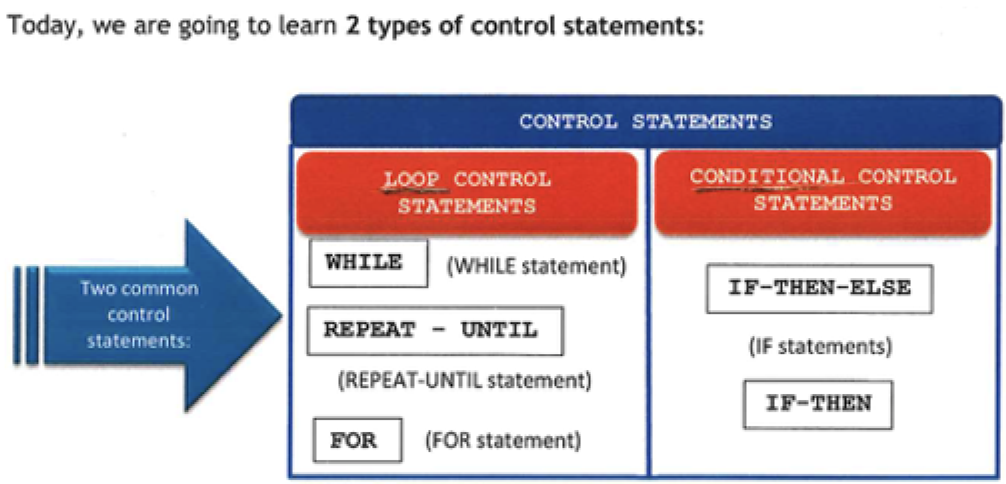
Student Algorithm of the Group Average Age



The goal pursued with this exercise was that the students established connections between pseudocode, i.e., code that is expressed by means of sequentially organized everyday language and the typical control statements used in CP (Figure 16).

Figure 16

Control Statement Chart Provided to the Students



After the students completed their average team age algorithm, Teresa invited the team through several questions to compare the students' algorithm with the pseudocode presented to them in their binder (Figure 14 above). To facilitate comparisons, Teresa used the following set of questions: (1) "Did you notice the changes between the way ||they wrote it? and || the way we wrote it (lines 2272-2273); and (2) "Right, so, what was the difference?" (line 2275); and 3) "Did you guys get that?" (line 2280). Michael responded "Yeah" (line 2281). Then, Juan pronounced one key word that from that moment was repeatedly associated with *accuracy* and computer programming, that is, "*precise*" (line 2,283). The facilitator confirmed right away, "*Precise* about it, right?" (line 2,284).

Then, the following short dialogue happened (lines 2,289-2,294):

F: So, why don't we write down, **precise.**

That's our observation.

M: They were *precise*

F: Yeah.

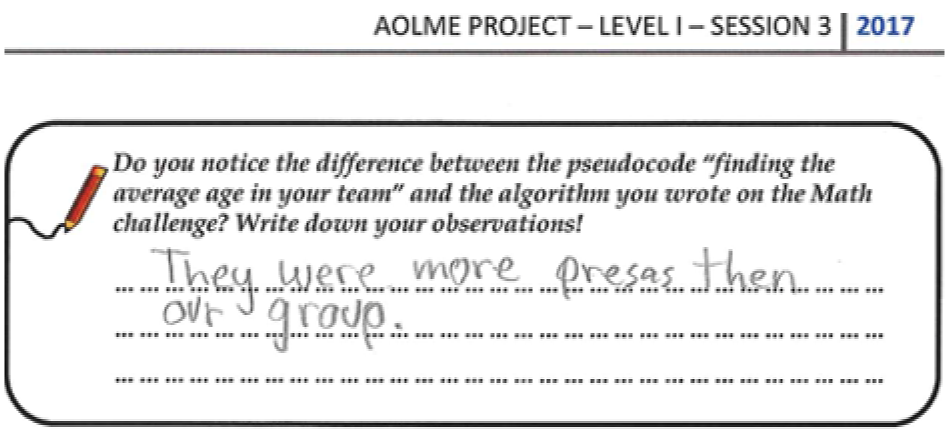
H: So, just it was *precise*?

F: Yeah, they were more **precise.**

Repetition of the adverb “precise” indicates the facilitator’s promotion of accuracy. It seems that, unknowingly for the students, the facilitator was preparing them for the kind of operative precision (Sfard, 2000) required for effective CT-CP communication (Bourke, 2018) (Bourke, 2018). The students took their pencils and filled out the box as can be observed in Figure 17.,Note in Herminio’s handwriting the words “presas” and “then”, which indicate his progress in the English language, typical at the middle school age (Lightbown & Spada, 2013). Herminio was probably trying to match spelling with the pronunciation that he was hearing.

Figure 17

Student Reflection Concerning the Precision Associated with Algorithms


AOLME PROJECT – LEVEL I – SESSION 3 | 2017

Do you notice the difference between the pseudocode “finding the average age in your team” and the algorithm you wrote on the Math challenge? Write down your observations!

They were more presas then our group.

Accuracy is a key CT-CP prerequisite (Grover & Pea, 2018) was typically signaled by means of the *adverbs/adjectives wrong and right* and commonly involved by the facilitator in collaborative coding, testing, and debugging practices. In short, *the facilitator promoted accuracy in multiple ways, also, explicitly*, as she did in one of the multiple student co-coding episodes when she asked Juan and Michael to check on Herminio’s Python codes on the computer screen while he inserted them to make sure they were correct: “You two make sure he is writing it correctly” (line 5,156).

Finally, the pragmatic CT-CP universe offered to the students had an *efficiency-based nature.* The facilitator recommended the students use the function *im_fill* to provide color to a whole segment of pixels at once, as opposed to using the *im_fill* function to provide color to pixels one by one. The dialogue that follows (Text #8) illustrates Teresa’s facilitation of *efficiency-based* activity to Herminio (H). Right after the short excerpt, I include three lines of code that Herminio developed after Teresa’s recommendation.

H: So, like for this one it's gonna be a **longer distance,**

F: Yeah,
So, like for this one you can **do im_fill** for every single **segment** like this.
That way you **get more** out of it **instead of**
doing *each one* **individually.**

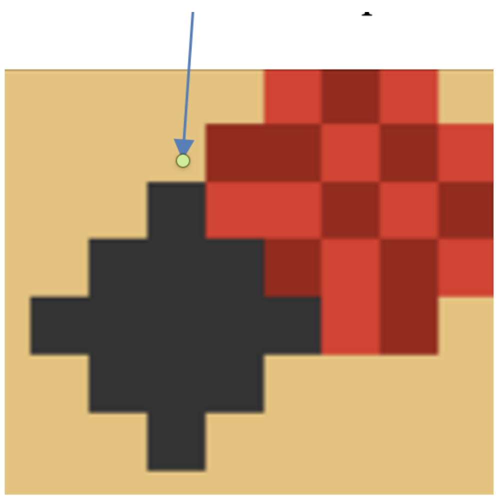
H: Mhm

Herminio was suggesting the use of function *im_fill* to give color to a longer distance of pixels at once. Teresa accepted Herminio’s suggestion, as can be read in the excerpt. Teresa’s recommendation by means of the verbal operator “get more out of” associated with process “do im_fill”, indicates *efficiency-based activity.* The line of code that follows constitutes evidence that this kind of efficiency-based activity was actually implemented.

`im_fill(frame1,[2,6],[14,14],"333333")` is the line of the Python program created by the students that coded the dark grey area of the asteroid of frame1. It shows that the Python function *im_fill* was used to at once call and give color to a vertical segment of 5 pixels, i.e., the “longer distance” or vertical segment `[2,6]` Herminio was talking about, which covered at once the 5 pixels that go from 2 to 6 in the Python digital grid used, instead of just one at the time, in which case five lines of code would have been needed (one for each pixel). Figure 18 clarifies this point. closer look at the center of the grey part of the asteroid (coded with hexadecimal color 333333) permits noticing that its vertical dimension is constituted by 5 dark grey pixels which exactly correspond to the range `[2,6]` Herminio was referring to (Note that in Python, columns start at 0).

Figure 18

Detail of Frame1's Asteroid



To conclude this section on pragmatism, I include Table 24, which gathers excerpts and textual indicators (lexicogrammatical resources) that further illustrate the pragmatic nature of the

CT-CP universe offered by the facilitator. Note that clauses were typically connected with conjunctive prepositions of purpose and result ("to," "in order to") which signal pragmatism.

Table 24

Characteristics of CT-CP Universe Offered by the Facilitator in Her Oral Discourse. Dimension I: Pragmatism

Pragmatism Dimension of the CT-CP universe offered	Text excerpt	Textual indicators
Action-oriented	The facilitator distributes jobs among the students (Text #6). F: He is ***doing*** the fin and you ***edit*** it.	*Use of verbs* that indicate *material processes of doing,* that is, **action,** as opposed to other process types and material processes of happening. A universe in which action dominates can be associated with ***pragmatism.***
Goal-product oriented	The facilitator brainstorms students about the video project (Text# 6). F: What do we wanna ***do***? J: Something like a wither skeleton with an iron armor on it.	The *verb* **"do"** indicates a creative material process. Here it references the creation of a ***tangible product***: a Python program which can display a video featuring a wither skeleton. Activity aimed at producing tangible products is a characteristic that can be associated with ***pragmatism.***
Tool- mediated (Both purpose & result)	During the co-construction of the video narrative, the facilitator provides the students with sticky notes to help them order each of the video frames or images (seemingly to	

	help them make sense of how the video story evolves in each of the frames) (Text#7). F: Yeah, so, you can ***use*** sticky notes ***to*** kinda' *figure out* the order that *you* want it to… During the coding of frame one, the facilitator provides Michael with a piece of paper with a line of code which includes computer programming function *im_show* (i.e., a command which makes possible to screen the video) (Text #12). F: ***In order to*** *play* the video, you have to *add* this line of code	 The *conjunctive preposition* ***to*** associated with material process *figure out,* indicates *purpose, that is,* ***<u>pragmatism.</u>*** The *conjunctive preposition in order to* associated with the verb *play,* indicates *result*, that is ***<u>pragmatism.</u>***
Accuracy-based	While reflecting on algorithms the facilitator encourages students to write down that one characteristic of algorithms is that they are more precise than pseudocode. F: So, why don't we write down precise That's our observation. M: They were *precise* F: Yeah. H: So, just it was *precise*? F: Yeah, They were more **precise.**	The adjective "precise" reflects accuracy, a characteristic that can be associated with ***<u>pragmatism.</u>***

Efficiency-based	The facilitator recommended the students to use the function *im_fill* to provide color to a whole segment of pixels at once and in a single line of code, as opposed to using the *im_fill* function to provide color to pixels one by one. H: So, like for this one it's gonna be a **longer distance,** F: Yeah, So, like for this one you can **do im_fill** for every single **segment** like this. That way you **get more** out of it **instead of doing** *each one* **individually.** H: Mhm	The process "get more out of" precise" reflects efficiency, a characteristic that can be associated with ***pragmatism.***

Procedurality

The CT-CP universe the facilitator offered the CLD students of this study was essentially procedural (Bourke, 2018; Papert, 1980). More specifically, the facilitator offered a CT-CP universe where activity was systematic, design and modeling-based, iterative, sequential, and modular. Also, in the CT-CP universe offered by the facilitator the program produced by the students could be reused.

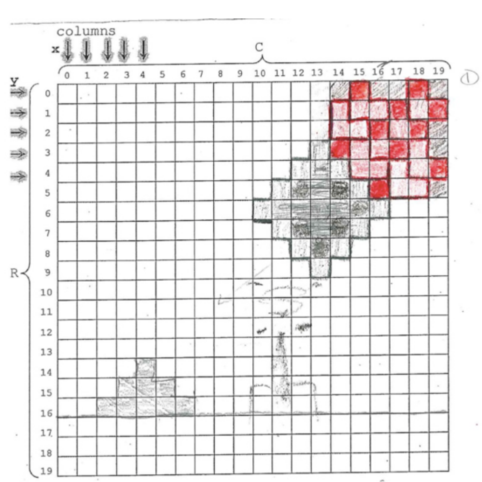
The students were offered a CT-CP universe of procedural nature which enabled the creation of digital images and an animation color video from the pixel level through modeling with mathematics the student pencil and crayon drawn prototypes (LópezLeiva et al., 2019). This procedure was carried out iteratively and involved brainstorming for ideas concerning what characters and background to include in the video and the composition of its frames, drawing prototypes of the video characters and background elements; modeling them with mathematics; programming them in Python and testing the program for the evaluation of the results; and, debugging errors if necessary.

For the computer to possibly process the pencil and drawn prototypes they were essentialized by the students in mathematical formalisms, that is, they were mathematized (NRC, 2011). These formalisms accounted for the shape and color of the students' prototypes and constituted the algorithm on the video which was coded by the students in Python data structures to create the Python program that enabled the video. In other words, the mathematization of the students' prototypes mediated the creation of computable 'logical artifacts' (Hoppe & Werneburg, 2019) which were programmed in Python with the support of an intermediate code that had been prepared for the students at the university (See Appendix F). To facilitate the mathematization of prototypes, the students were provided with 20 rows by 20 column paper grids wherein they drew them in pencil and/or color crayons with the shape and color that they chose. Once drawn, the students took note of the coordinates of the two-dimensional arrays that defined their shape and of the exact shade of color in hexadecimal number notation that defined their color, which they selected in a color picker app. Then, the students used these mathematizations to program the prototypes in Python. The students followed this procedure systematically and iteratively to create each of the elements that featured each of the five frames of their animation video. Thus, the procedure included: (1) a design phase where the video elements and characters were conceptualized (through brainstorm ongoing discussion), drawn and mathematized; and (2) a subsequent computer programming phase where the students coded each of their prototypes in Python in order to digitize them and provide them with motion.

For instance, Teresa facilitated the design and programming of the asteroid of frame1 of the video, which prototype can be seen next (Figure 19).

Figure 19

Prototype of Frame1's Asteroid of the Student Video



Following are the first two lines of the dust part of the asteroid (the red part) which were programmed, in this case by Herminio, where AF0000 is the hexadecimal number that he used for red.

```
#dust
```

```
im_fill(frame1,[0,0],[17,17],"AF0000")
im_fill(frame1,[1,1],[15,15],"AF0000")
```

(The Python function *im_fill* was used by Herminio to fill the two-dimensional arrays of the asteroid specified with Y and X values in squared brackets) with the color specified between quotation marks (AF0000, i.e, a specific shade of red).

Prototypes such as the ones shown in Figure 19 constituted the basis of the digital frames or scenes of the video. The first frame developed by the students took a significant part of the work since it included most of the video objects thus constituting the basis of the next two frames of the video program. Figure 13 (above) shows in order the five frames that were developed by the students *iteratively* by repeating the design-mathematization-programming processes I described. The procedure adopted involved developing the Python program and video in an incremental way where the program builds on previous work as this develops. Being incremental and iterative has been identified as core CT-CP practices (e.g., Brennan and Resnick, 2012; Grover & Pea, 2018). As can be observed in Figure 13, frame2 and frame3 look very similar to frame1. Consequently, the actual Python code of frames 2 and 3 are very similar to the Python code of frame1 as evidenced in the final whole Python code of the video (Appendix E). The facilitator discourse evidences that the CT-CP universe that she offered the students included reusing code (Brennan & Resnick, 2012). The students reused the Python code of frame1 to create frame2 and frame3. In other words, once the students had programmed frame1, they modified it by changing the size and location of the asteroid, the feet of the blue-eyed wither skeleton or spider and the brown handle of the shovel (See Figure 13). The background remained constant, with the sun and the land appearing the same, only the tree had almost disappeared in frame3 because of the asteroid impact. In frame4, the asteroid had made the wither skeleton, the shovel and the tree disappear completely. Accordingly, they were deleted in the Python program.

The following Text # 12 excerpt illustrates the **code reuse-prone** and **sequential** nature of the procedural CT-CP universe that Teresa facilitated, and the students adopted iteratively The group had just finished coding the characters of frame1 of the video when she explained to the group that programming the following frame would just be a matter of copying said frame1, and pasting it for frame2, thus, ***reusing*** it, and then modifying it as needed. Reusing code has been identified by Brennan & Resnick (2012) as a key CT-CP practice. Right the moment when the students finished programming frame1 of the video Herminio and the facilitator said (Text # 12):

H: It looks nice
Too much work for that little one.
F: Well, guess what though,
this was the hardest part,
you wanna know why?,
cause now, all you have to do is ***copy and***
paste for the ***next frame***
and then change a few **coordinates** *because*
your **asteroid gets bigger.**

Teresa's language use in this short dialogue reflects the procedural nature of the activity that she facilitated. Conjunction "and" and conjunction group "and then" associated with the processes "copy," "paste," and "change" reflect the additive and ***sequential*** relationships typically found in procedures in academic settings (Derewianka & Jones, 2016), which in this case involved the students' first and second frame of the video and its character's shape defining coordinates. Additionally, the facilitator explained by means of *the conjunction* "*because"* the reason for changing some *coordinates*. Below is an analogue example from Text #9 when Teresa was facilitating Michael's coding of frame2. A similar use of conjunctive binders can be observed which provided a ***sequential*** flavor to the CT-CP systematic procedure that was represented in the facilitator discourse and adopted by the students (Text # 9).

F: ***And then*** you are gonna put
   <u>*frame 1*</u> and <u>*frame 2*</u>
   frame one
M: ***And now***, <u>*underscore*</u>?
J: 1
F: Just 1
   There you go,
   and then <u>*frame coma 2*</u>
   and then end it.
   Alright, ***and***
   There you go,
   Now enter

Thus, the facilitator was engaging the students in programming in Python the video frames or scenes they wanted to see in their video. The procedure she facilitated was systematic and sequential and included the construction of the Python program, which featured a modular structure.

As just illustrated in the above excerpt of Text #9, the facilitator offered the students a CT-CP universe where programs are organized in modules, e.g., "frame1" and "frame2". The modules of the program started with the name of the frame, i.e., frame1, frame2, frame3 and so on, and then in each frame elements such as `#tree`, `#sun` and `#Michael character` mark the start of a module of the program. The following excerpt shows the facilitator explaining

Michael the modification affordances of the universe that she facilitated. (Lines 15, 036-15, 045). It shows the facilitator encouraging Hermino to use *participant* "smoke" to differentiate in the Python program the part that corresponded to the trail of the asteroid which they eventually called "dust" as can be seen in the Python program developed, shown in Appendix E.

F: So that it **doesn't look**
so clustered,
put like a hashtag and
say smoke.
H: What hashtag smoke?
F: **Hashtag smoke.**
H: What does that do?
F: That way, it's still in your character file,"
But that way you don't get confused with the rest of the code just in case
you have to **change** it later.

Developing programs in modules, that is, modularizing has been signaled as a core CT-CP practice (Kong, 2019). The procedure facilitated by Teresa resembles many of the characteristics of iteration-based workforce methodologies commonly adopted for the development of software which allows modifications as it is developed (See Chandra, 2015).

To encourage code analysis, the facilitator fostered testing typically through the *process* "run" or by simply saying "F5" (the key in the computer keyboard that would execute the Python program). Following is an example that took place when Michael was coding the tree for frame1 (Lines 14, 809- 14, 814).

F: Alright,
you wanna **run** it, right?
M: Run it
F: You do an im_show

‘cause that's the command.
and then you
that should **run** it

Additionally, the facilitator reminded that the student needed to include in the program the Python function im_show. The universe offered by the facilitator allowed easy and instant analysis or evaluation of the Python program under development. As the students developed the Python program, they tested it to see if it produced the digital results that they wanted. If the program they coded yielded a solution that satisfied them, they resumed their programming activity; if it didn’t, they would modify what was needed to be satisfied with the result. For instance, the students continuously tested, evaluated, and modified where needed the Python program that coded the asteroid and the shovel of frame1 to create an effect of asteroid shape change to make it look bigger in frame2 and frame3 (See Figure 13). Analyzing programs, that is, testing and evaluating them has been correlated with automation and specifically with CT-CP development (Lee et al., 2011). The modularization discussed above allowed access to parts of the Python program for their modifications or reuse when it was convenient. Thus, the “frame1” and “frame2” mentioned in the excerpt of Text # 9 shown above, and the tree and other elements characters that composed them could be and were modified after being tested for satisfaction (testing was continuous, specially at the early stages of programming until the students seemingly gained confidence in programming).

When an error was made, the computer did not provide solutions at all. The computer system would call the error on its screen, typically pointing it out as “syntax error” next to the line in the program where it had been produced. On these occasions, the program was debugged in search for the error that interrupted its execution. Testing and debugging have been identified as key CT-CP practices (Brennan & Resnick, 2012; Grover & Pea, 2018; Kong, 2019).

Dependency

The CT-CP universe offered by the facilitator to the CLD students was dependent on both social and CT-CP disciplinary conditions. On the one hand, words which are typically associated with collaboration and agreement pervade the facilitator's discourse. Throughout curricular activity pronouns with an inherent social meaning such as "we" and "us" nouns such as "group", adverbs such as "together" and "each other" and processes such as "agree", "help," "work" associated with high modals (e.g, "have to") and said nouns, pronouns and adverbs indicate that collaboration and agreement were pervasive in the facilitator discourse. Collaboration has been identified in Grover and Pea (2018) as a key CT-CP practice. In mathematics education research Cobb & Yackel (1996) identified socio-mathematical rules established in mathematics classrooms. The collaboration and agreement found in the discourse in the participant facilitator can be said to be two socio CT-CP norms in the CT-CP universe that she offered to the CLD participants. The analysis of modal verb uses in the 15 texts selected revealed that Teresa's use of the high value modal verb *have to* was salient at strategic moments to highlight collaboration and agreement as social requirements of the CT-CP universe offered:

1) *The need to collaborate*. For instance, in the task orientation stage, when the facilitator was providing the general expectations for the making of the video she said: "Come up with some idea together, okay? ||'cause you guys have to work together" (Lines 10,562-10,563). On this occasion, the presence of one of

AOLME's principal investigators, who happened to be there at that moment, seemed to add further importance to the requirement to collaborate.

2) *The need to agree*. For instance, in the task orientation stage, a condition associated to activity was established by the facilitator, that is, the need to agree on the story that the group would develop in Python: "What do you guys think || we should do? (Lines 10,742-10,743); and "Okay, see ||'cause we will have to agree on this" (Lines 10, 785-10,786).

On the other hand, CT-CP disciplinary conditions were salient in the discourse of the facilitator. For instance, some CT-CP rules for the creation of videos had to be followed. The need to create a background for the video was established as a must: "We are gonna have to do a background (line 11,998). While the background was needed to situate the students' story, it was a Python technical requirement. In addition, the story needed to be designed in scenes or frames: "cause remember, || you have to do it in frames" (Lines 3812-3813). By means of the high value modal "have to", the facilitator set the conditions that the background of the video was a requirement and that the story had to be told in frames.

Also, the recursive use of Python functions such as the Python function `im_show` was a must. This function was needed for the frames of the video "to be shown" on the computer screen. The facilitator indicated this need through clauses such as the following: "And then im_show this one, || 'cause you have to do the im_show for all of them. (Lines 15, 654- 15,655). By means of the high value modal "have to," the facilitator set the condition that the Python function im_show had to be used in each frame of the video for the computer to display all the frames. Additionally, syntax rules needed to be followed. The facilitator consistently reminded the need to follow syntax rules such as the quotes used in strings, a Python variable recurrently used in the CT-CP practices offered. The facilitator introduced strings in the task preparation stage, referring to them as “information that we want to show” (line 720) stating that “we’re always gonna show it in quotes” (line 721). In this example, the use of median modal structure *gonna* enhanced by the adverb *always* indicated the need to use quotes whenever strings were used.

Finally, some mathematical formalisms needed to be adopted. For instance, the two-dimensional arrays that defined the shapes of the prototypes had to include the Y value first and then the X value. The facilitator stated this recurrently with apparent clarity without making use of modal verbs, leaving no space for different options in this respect: “you wanna do your Y coordinates first” (ine 14, 155); “the Y goes first” (line 15, 158). The need to program first the Y value of the coordinates that composed the arrays that defined the students’ prototypes was a must too since it is also a Python requirement**.** This created some confusion in the students since it was apparent that they were used to using X values first as typically is done in mathematics.

Flexibility

As suggested by Hoppe and Werneburg (2019), the CT-CP universe the facilitator offered to the CLD students was flexible in the creation of algorithms. The analysis of modal verbs used in the 15 selected texts helped uncover this flexibility. The facilitator offered the students to design and model their prototypes with the shapes and colors that they chose. To do so, the facilitator frequently used low modal verbs such as *can* and *could* which opened space for students’ free action (Halliday & Matthiessen, 2014). The students were offered flexibility in the creation of algorithms. The facilitator offered flexibility in several algorithm-creation related activities by means of low value modal verbs (e.g., "can" and "could").

For instance, the ideas and characters used in the video were up to the students. In the task orientation stage, the facilitator used low modal verbs to brainstorm ideas concerning the elements that they wanted to include in their video and the story they wanted to tell. The facilitator asked the students for their favorites. "Okay, it could be, person, place or thing" "So look, ||we could do||, with all the ideas, here is my idea||, and you could tell me more. || (lines, 10, 927-10,931). The setting of the story was also up to the students: "We can have the tropical island where it snows" (line 10, 937). The facilitator offered multiple options. Another one was: "we could do a tropical island that has different animals" (line 10,979).

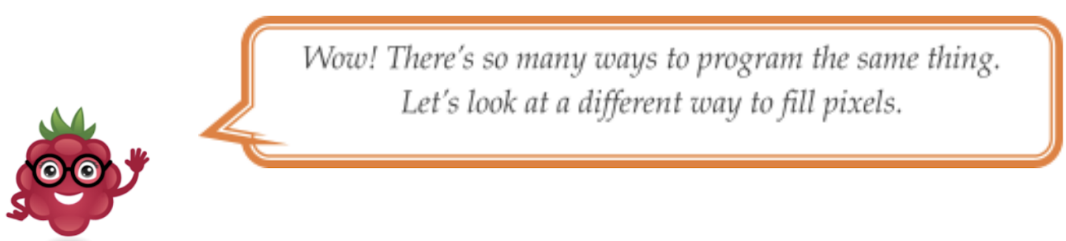
The facilitator also left up to the students the actual design of their prototypes. Their shape and color were left up to them. However, the facilitator at times provided suggestions. In the task orientation stage, Juan was engaged in making the asteroid of frame4 bigger. The facilitator suggested: "You *can also do it* like to where like || it goes extra, here like an extra ||you know what I mean ||that way it just looks bigger in general (lines, 16, 569-16,573) (Text #14). By adding extra squares to the asteroid Juan was designing in the paper he used for frame4's asteroid the resulting digital asteroid in frame4 had more pixels and looked bigger in the digital image and video than already designed by Juan. The idea of making the asteroid bigger had been his and Herminio's "I'm gonna only make it bigger, "I'll make it bigger now" (lines 16,456-16,453) In fact the facilitator in the above-mentioned lines 16,569-16,573 was only validating Juan's desire to make the asteroid bigger. As for the colors that were used, the facilitator was also totally flexible, only intervening if the color of an element of a frame interfered with the colors used in another frame. As was explained, when the students ran the code, the frames sort of flipped like pages of flipbooks do. The frames programmed by the students in Python were superposed in order. For the computer to display the frames so that all the characters and elements coded in their digital grids could be seen, the colors at analogue pixel locations in the digital grids could not be the same. Aside from this circumstance, the students were free to use the colors that they wanted. The flexibility to use colors for the students designs was first offered by the facilitator in the following way (Task orientation stage): "If you go to Google||, and you pick the html color picker||, you can actually look through the pic, through the colors||, and change them||, and right there it tells it to you in hex||, so, you know, if you know|| that that's what you want (lines 11, 076-11, 082) Also, the facilitator's discourse shows flexibility in the use of tools to be used in design. The facilitator suggested to the students the (flexible) possibility of using sticky notes to make sense of the sequence of the frames or scenes of the story (and Python modular code) the students wanted to use in telling what would happen in their video. "You can use sticky notes || to kinda' figure out the order that || you want. (Lines 11,990-11,992 from Text #5).

Furthermore, the universe offered by the facilitator included flexibility in the use of Python functions to program the students video. The facilitator stressed, as the following excerpts illustrate, that the Python function `im_fill` could be used flexibly, allowing to digitize and fill in with color longer or shorter distances of a prototype: So, like if you wanted to ||you can do, so like for this one you can do im_fill for every single segment like this". (Lines, 12, 878- 12, 881)… "Using this code is that || you can, you can do like long distances" (12,886-12, 887) "Well, this, the in–fill allows us ||to pick where we want ||to fill in (lines 12, 890-12,892)

(Text #8).Thus, the students were offered flexibility in the use of `im_fill` to add color to a single square of a particular character or video element or use the function `im_fill` to digitize and add color to longer segments. As explained above in the section devoted to procedurality, the shapes of the prototypes were determined by the Y-X values that defined the two-dimensional arrays that composed them. Eventually, the whole set of two-dimensional arrays that defined the shape of each of the elements of the video constituted the algorithm that they created. The students, as was explained, inserted all the two-dimensional arrays of the shapes of all the prototypes that they designed, along with their color in hexadecimal numbers, in the structures provided by Python for the computer to process all this information and produce their digital images and video. Their algorithm was a step-by-step sequence of shapes and colors which they formulated with the precision that mathematics affords, an "unambiguous procedure that is repetitively applied" (NRC, 2010, p. 9). The facilitator and the students were adopting a recommendation included in the written AOLME curriculum as is evidenced in Figure 20:

Figure 20

Promotion of Flexibility in the CT-CP Written Curriculum



Finally, the students were offered flexibility in the comments which they used to organize their Python program based on a modular structure. The facilitator discussed this with the students several times. In the task realization stage (Text #12), Herminio suggested the use of "dust" as the name to tag the asteroid: He asked: "can I put dust?", the facilitator responded yeah (you can put) dust (line 15, 199).

Summary

The SFL-based analysis of the discourse of the facilitator revealed a significant number of aspects that describe the nature of the CT-CP universe that she offered to her small group of CLD students. The facilitator construed the CLD students as the main agents of hands-on relevant CT-CP practices where she also involved herself. Therefore, material processes dominated activity. A relevant finding concerns a low presence of cognitive processes in the discourse of the facilitator, which pervades with material processes of doing. Hence the facilitator discourse presented a CT-CP universe where action and desires dominate over thinking processes and conceptual explanations.

The CT-CP practices offered were situated on the here and now, centering initially on computer hardware to then focus on algorithms, programming, data structures, modeling, and representations. CT-CP practices such as abstraction, designing, modeling, data related activity, computational artifact creation, algorithm creation, being incremental and iterative, modularizing, testing, debugging, reusing, collaboration and creativity were identified. However, for the most part, with the relevant exception of collaboration and algorithms[33], they were not

[33] The word algorithm was used by the facilitator only in the task preparation stage where the students were engaged in algorithm-oriented activities such as creating a bowl of cereal and calculating the group's average age. However, the term algorithm was not used by the facilitator or the students again once the creation of the main student project started, that is, the creation of an animation video in Python.

made explicit by the facilitator during the curricular activity offered. As for the CT-CP concepts and syntax-related elements pervasively used by the facilitator, such as Python function im_show and parenthesis, they were typically qualified with possessive qualifiers, which may have developed student ownership of CT-CP concepts and practices.

Fundamental in the CT-CP universe offered to the CLD students was the inclusion of their own non-academic personal & imaginative worlds, which were incorporated by the facilitator to the practices at hand, most importantly to the creation in Python of an animation video. Thus, the facilitator involved elements such as wither skeletons, trees and asteroids in processes that mediated their design and modeling with mathematics, which provided the necessary "truth" for these elements to become computational artifacts that could be represented in their computer programs. In so doing, an essentially interdisciplinary universe was presented to the students in the form of an abstraction and creativity based systematic procedure. The processes of abstraction that enabled the CT-CP activity offered involved several LOAs to which the students were exposed so that they could navigate from their original ideas for the video expressed in English and Spanish through the mathematized crayon and pencil drawn designs to the mathematization codes, to the Python code and digital images and video that were produced and displayed by the computer. These images and video were recurrently tested and refined by the students for correctness and satisfaction in a universe where accuracy, efficiency, and flexibility were also important characteristics. Finally, the universe offered combined flexibility and conditions. It allowed for flexibility, for instance, in the student creation of the algorithm that originated in their interests and creativity formed part of the Python program that they developed for their digital animation video. A few disciplinary conditions were presented as non-negotiable such as several rules inherent to computer systems and to CP language Python. A social rule was also specifically demanded by the facilitator: collaboration.

How were these CLD students positioned in terms of their participation and role in CT-CP practices?

The participant facilitator positioned the participant CLD students as capable collaborative producers of their own creative computer technology. The facilitator engaged the students in a systematic procedure that led them to produce in Python a digital video which narrative, setting, characters originated in their own creative ideas. The ideas used in the video were elicited by the facilitator by means of authentic questions, which positioned students in contexts that related the CT-CP practices performed to their knowledge and experience about the world (Herbel-Eisenmann & Wagner, 2007). In doing so, two things seemed to matter:

- *The students' knowledge about the world:*
    - Their non-academic knowledge
    - Their academic knowledge
- *The students' linguistic identity*

For the students' ideas to become a digital reality, the students transformed them into mathematical formalisms that the computer could process (Ben-Ari 2012; Colburn & Shute, 2007; Reeves & Clarke, 1990). By means of inclusive commands the facilitator constantly positioned them as thinkers (Herbel-Eisenmann & Wagner 2007; Morgan, 2006; Rotman, 1998; Schelleppegrell, 2012) thus as computational thinkers and capable members of the CT-CP community and producers of <u>their own</u> creative computer technology.

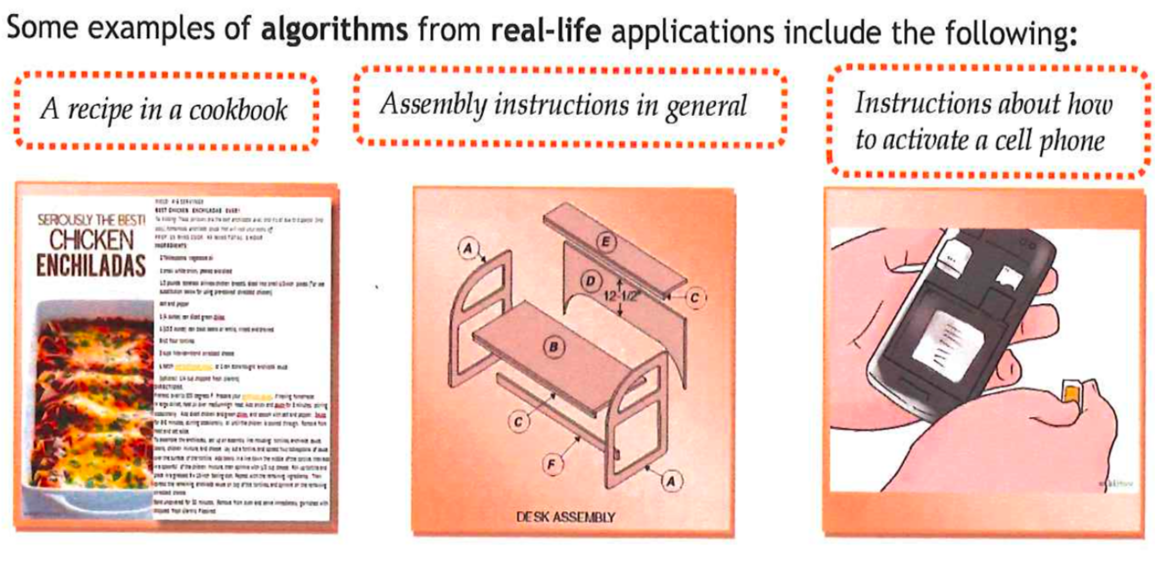
In what follows, I first show how the facilitator by means of authentic questions positioned the students as providers of the information that the students' knowledge about the world mattered. Then, I illustrate that their linguistic identity mattered in a variety of ways. Last, I center on how the facilitator, using inclusive commands, positioned the students as capable computer thinkers and programmers in several key CT-CP practices involved in producing their own creative computer technology.

The Non-academic Knowledge of Each Student about the World Mattered

The facilitator's salient use of authentic questions revealed from the start of her teaching that the students' non-academic knowledge of the world mattered. Both the students' personal & imaginative worlds and their academic knowledge were considered. In the preparatory stage, the facilitator engaged the students in hands-on CT-CP practices which prepared the students for their final project, their Python video. Seemingly, a fundamental goal at this stage was to get the students familiarized with the tools, practices and CP-mathematics and image-processing skills and concepts that would be needed. A fundamental CT-CP practice concerns the creation of algorithms (Kong, 2019), which the facilitator focused at this stage. Teresa engaged the students in creating an algorithm, seemingly to prepare them for the algorithm that would later be included in the Python program that they developed for their Video in Python. The algorithm the students created at this stage was based on a few models provided to them and can be seen in Figure 21.

Figure 21

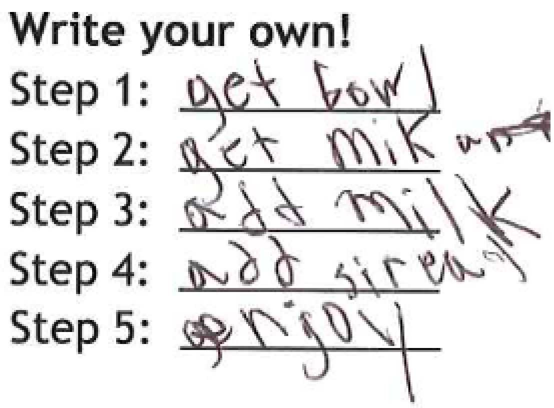
Examples of Algorithms Provided to the Students in their Written Curriculum



The facilitator discussed with the students that an algorithm is basically a set of instructions or steps to perform a task or solve a problem. Then she asked them several consecutive authentic questions to identify the algorithm they wanted to create, positioning the students as providers of the key information that was needed in said creation. To get at the task that would be focused, she asked: “What kind of task?” (line 1880). The students responded that the algorithm should regard ‘how to play’. The facilitator asked: How to play what? (line 1884), then another student mentioned shoes: How to tie your shoes? (line 1990) the facilitator asked. Discussion continued: they considered “playing outside” “tying shoes,”, ‘eat nachos”… Eventually the students agreed on creating the algorithm for the task of preparing a bowl of cereal and on the 5 steps that were required to accomplish the task (See Figure 22) .

Figure 22

Student's Bowl of Cereal Algorithm Steps as Written by Michael



The authentic questions used by the facilitator in this exercise are illustrative of the facilitator positioning the students in CT-CP practices that related to their non-academic experience in the world (Herbel-Eisenmann & Wagner, 2007). While the students were provided several models of algorithms (Figure 18), they were urged to do their own. As they created the algorithm shown, they were taken from the personal non-academic world to abstract CT-CP practices such as the creation of algorithms and algorithm thinking. Apparently, the facilitator was preparing the students for the complex abstraction processes that she would have to engage them in to program their video in Python.

In the task orientation stage, the facilitator positioned the students very similarly. She continued focusing the students' experience on the world. This time the focus was their knowledge and interest in video games (Michael's and Juan's) and about outer space (Herminio's). At the very start of this stage the facilitator brainstormed the students about the video story, setting and characters. She asked the students: "So, what do you guys wanna do? (line 10,775), positioning them as providers of information and decision makers of the computer technological creation that they were going to produce. This information was crucial since later it would be transformed into mathematical abstract formalisms that the computer could process At this stage, in her authentic questions, the facilitator used vocatives (the students' names) to address each of the students specifically, seemingly, to make sure that all of them had a say in this crucial stage where important decisions about the video project were discussed. Also, when it was the students who brought forward ideas without having been asked, the facilitator consistently validated them, provided that disciplinary or time constraints did not interfere, showing them that their interests, creativity, and experience in the world mattered. Following, I include some excerpts which illustrate this point (Text # 4).

Line 10,777 **F:** **So, what do you guys wanna do?**

Line 10,778: J: Wither skeleton.

Line 10,779 M: **Wither skeleton.**

This time, Herminio did not respond to the facilitator's question, only Juan and Michael did**.** Then, the facilitator asked about the setting where the students wanted their story to happen (Lines 10,912-10,919):

F: **What do you like, Michael?**

M: What?

F: A place.

M: A place,

A place, a place, a place,

M: Somewhere where it's warm

Tropical island

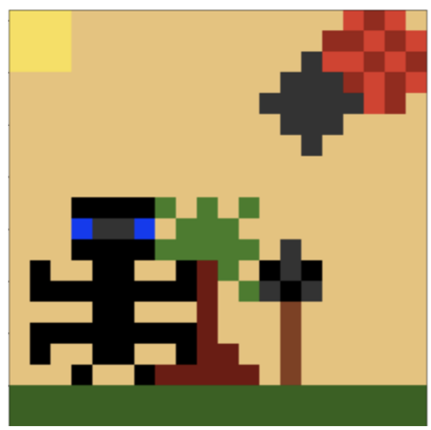
F: A **tropical island** for Michael.

Soon afterwards, the facilitator asked “**What do you wanna do, Herminio?”** (line 11,003); Herminio responded: “Maybe an **asteroid** crashing into the island, the tropical island with an alien” (line 11,004). Evidence that the students’ knowledge of the world mattered is the reification of the students’ creative ideas that the facilitator elicited in the Python code and digital images that were produced. They can be observed in Figure 23.

Figure 23

Frame1 Comments and Digital Image

```
#FRAME1
#tree
#land
#sun
#michaels character
#Juan character
#HERMINIO character
```



As can be observed, the tropical island, which was agreed by the students to be integrated by a tree (brown and green), a land (green), and a sun, made it to the Python code and to the video images. Similarly happened with the other elements included in the Python code and frame1 image: “michaels character”,i.e, the blue eyed “wither skeleton,” “Juan character,” (i.e., the object that can be seen next to the green crowned tree—a shovel with magical powers), and "HERMINIO character," (i.e., the asteroid that is approaching from the digital image's upper right corner).

For such computer technological products to become a digital reality, the facilitator demonstrations that the students' interests and knowledge of the world mattered continued in her facilitation of the abstraction processes that were required to create the video in Python. The following excerpt captures the facilitator's discourse when Herminio showed his steroid prototype to her (task orientation stage, lines 3060-3071)

F: That looks awesome!
yeah,
now if you want to
borrow my phone,
and look at the html
and you can see
what colors you want?
so, like say you want this one color,
you want it red,
that way, you should go the other way,
do you want it bright?

By means of the authentic questions I bolded, the facilitator was urging Herminio to decide the colors in hexadecimal numbers that he wanted for his asteroid. Soon afterwards, Herminio showed to Teresa written on a piece of paper the colors that he had chosen (Lines 3089-3093)

F: That's your asteroid, right?
H: Number 89, 89797A
F: Then on the bottom put asteroid,
F: that way you'll remember what it is.

The hexadecimal number 89797 actually encoded the grey color that can be seen in the asteroid image of frame1 shown above in figure 20.

Therefore, when the work on the Python video started, the exchanges of information centered on ideas that concerned the video background elements and characters. However, this information had different forms at different moments of curricular activity. It took careful analysis to identify that data representations such as “89797A” and “7,7; 13,17” referred to the same object, that is, “asteroid”, which was also a pencil drawn prototype and a digital image. As will be demonstrated, the set of numbers “7,7; 13,17” used at the very end of curricular activity (line 16, 680, Text# 15) was an abstract representation of a piece of the asteroid. Just one of the dozens of abstract representations of segments of video elements that were communicated between participants and coded in the computer to create the digital images that constituted the video in Python code. It took tracing the history of abstractions (Hershkowitz et al., 2001) such as these to identify the data encoded in abstract representations such as “89797A" and “8 to 8; 13 to 17” which could be traced all the way back to the brainstorm that the facilitator had had with the students when she had asked” “W**hat do you wanna do, Herminio?”** (line 11,003) and Herminio had responded: “Maybe an **asteroid** crashing into the island” (line 11,004). The two-dimensional array “8 to 8; 13, to 17” was just a little “step” of the algorithm that integrated the Python program that the students created. It encoded 5 horizontal little squares of the trail of the asteroid pencil drawn prototype which were transformed by the computer through the Python program developed into 5 digital pixels of frame3’s asteroid. Coordinates such as “8 to 8; 13 to 17” encoded CT-CP, they represented the characters of the digital video and the sequential programming procedure that was adopted to create it. Below I include the interactions that happened about half an hour before the end of curricular activity. They capture the exact moment when the facilitator substituted Michael and Juan in a final collaborative event which was basically led by the students where all participants contributed (see whole Text# 15 in Appendix D). The group was pressed for time and the facilitator intervened, seemingly to accelerate their work. “Hey, M, we don't have time” she said at line 16, 679. The facilitator continued Michael and Juan’s work who had been until that moment mathematizing and communicating the abstractions of the asteroid to Herminio for him to code them in the Python program. Here is the above referred excerpt of Text #15:

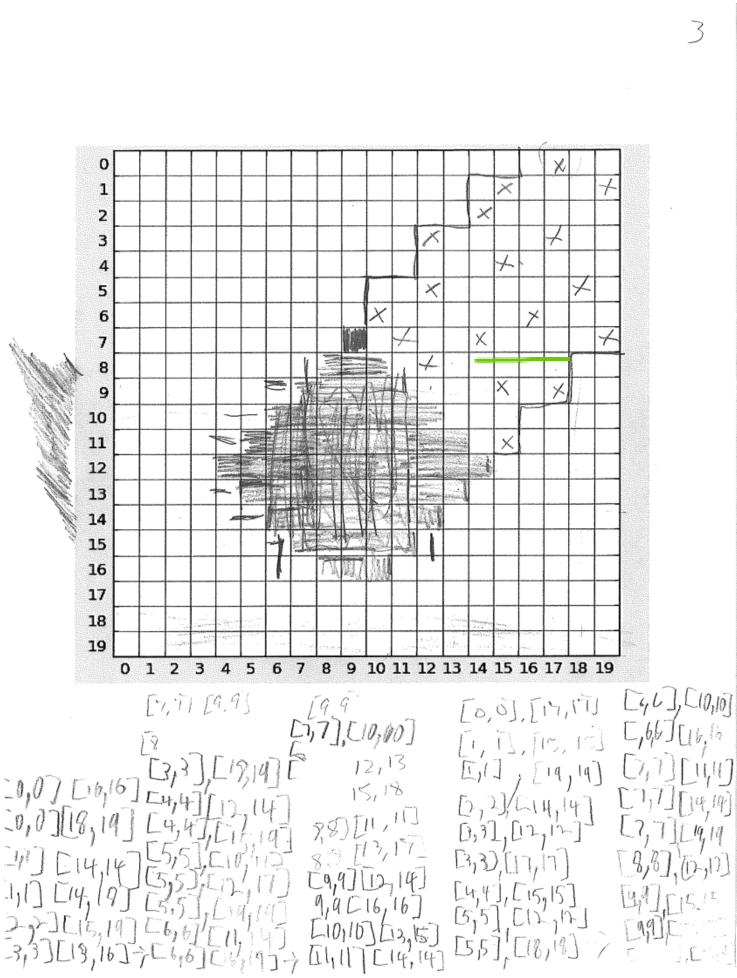
Line 16,679	**J:**	**8 to 8**
Line 16,680	**F:**	**13 to 17**
Line 16,681	**H:**	**13 to 17?**
Line 16,682	F:	yeah
Line 16,683		**9, 9**
Line 16,684		**12 to 14**
Line 16,685		**9, 9**
Line 16,686	M:	So much work for one meteorite trail
Line 16,687	F:	**15, 16**

Close attention to the asteroid prototype and digital image shown below in Figure 24 allows observing that the abstractions communicated by the facilitator (bolded) correspond to the segments (marked in green) of the prototype and digital image of the frame4 of the video.

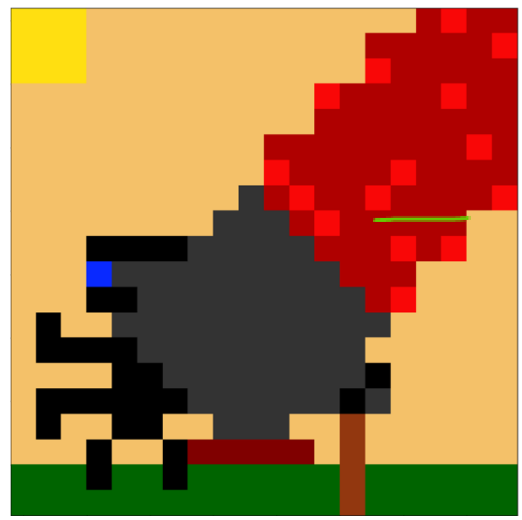
Figure 24

Frame3 Prototype, Mathematization Codes, Digital Image, and Python Code

Frame4 asteroid prototype



Frame4 digital image



Corresponding line of the Python Program

```
im_fill(frame4,[8,8],[13,17],"AF0000")
```

As was demonstrated, two-dimensional arrays such as “8 to 8; 13 to 17” (and the ones that followed) were a little part of the asteroid that Herminio wanted to see in the video. The spaces left blank in the squared 20 by 20 paper grid (marked with the green line) signaled where to fill with by means of function `im_fill` with dark red AF0000 instead of the above-mentioned lighter red 89797A (marked with green on the digital image shown). Note the line of code that was used (See Appendix D), which includes the coordinates that were used, i.e., `[8,8]`, `[13,17]` and the hexadecimal number `AF0000` for dark red

By means of brainstorming authentic questions the facilitator had positioned Herminio as a provider of key information that was needed in the creation of the computer code and digital images and video that the students created. It was demonstrated that the students’ knowledge of the world and interests mattered. Abstract data such as “8, 8 to 12, 17” was just a little segment of the asteroid, a little piece of the algorithm that defined the task that the computer would process to make the students’ story digital with the Python functions and data structures that the students programmed. As had happened in the task preparation stage, the facilitator had used the students’ non-academic knowledge as a basis to construct abstract algorithms and data representations.

The same procedure was followed with the rest of the elements of the video. The video games characters (Minecraft characters) discussed by the facilitator with Michael and Juan, that is the wither skeleton and the shovel also made it to the computer code and digital images and video that the students created. Hence the students’ non-academic knowledge of the world mattered. Following I show that the facilitator also considered the students’ academic knowledge

The Students’ Academic Knowledge Level Mattered

A Functional Use of Accessible Mathematics

Teresa facilitated a creativity-based CT-CP activity that depended on being precise in the use of mathematics affordances that seemed accessible to the participant students, who were at that moment in 7th or 8th grade. By means of coordinate plane mathematics[34], which apparently were not new for them, and the use of hexadecimal numbers[35], the facilitator engaged the students in transforming their non-academic knowledge into the abstractions needed for their initiation in CT-CP practices. The academic mathematics knowledge and skill that they needed to produce their own creative computer image data structure limited to knowing how to locate shapes in the coordinate plane (on the squared 20 by 20 paper grids provided), thus, indexing and basic geometry. By using cartesian coordinates and two-dimensional arrays of columns and rows the students were able to mathematize[36] the original pencil-and-crayon-drawn prototypes that they wanted to feature their video in Python. There was one thing to pay attention to, though. In Python, and virtually in all programming languages, the Y values are used first as opposed to mathematics where X values precede Y values. The facilitator was very explicit about this and reminded the students several times For instance, when Michael was engaged

[34] Prior to the start of the video, the facilitator engaged the students in activities aimed at locating coordinates and arrays on the two-dimensional plane and in communicating them to each other. Beyond some mistakes and hesitations and the fact that in programming the mathematics cartesian y and x coordinates were reversed (Y axis coordinates first), the students did not encounter major difficulties, as per the dialogues produced.

[35] Prior to the start of the video project, the students were engaged in number conversions between the decimal, binary and hexadecimal systems and in activities aiming at developing the typical binary logic of computer programming. However, in the making of the video, the use of hexadecimal numbers was straightforward, and no more calculations were made. The students would use a color picker application to select the exact color in hexadecimal number notation that they wanted to use to give color to their images.

[36] "Mathematization is a technical term representing relationships in the natural world using mathematics" (National Research Council, 2012, p. 16)

in programming the wither skeleton or spider of the video (Lines 12,814-12,817), the facilitator said: “Now this is where *you get* your coordinates||this is where *you look* at your picture and ||*you are gonna* do your y axis first (arrays starting from row = 0 amd column = 0). By picture, the facilitator was referring to the crayon-pencil drawn prototype.

CT-CP Academic knowledge: A Low Presence and Functional use of CT-CP Concepts

The facilitator engaged the students in a hands-on “easy to get started (low floor)” approach to computer programming (Resnick et al. 2009, p. 63). Analysis of participant-speech function across the 15 Text exemplars and the separation of proposals and propositions revealed that the facilitator CT-CP explanations were typically brief and functional and decreased as time passed and the students gained independence. A simple search in the excel spreadsheet used to support and analyze the whole text revealed that all the CP concepts that had been used in the task preparation stage, such as loops and conditionals, practically disappeared once the orientation stage started. Only the concept variable was used at the start of the programming of the video. As for CT-CP concepts/practices, they never formed part of the instructional field, except for algorithm and testing, as already explained.

When the programming of the video started, work and explanations were pragmatic, very straightforward and used just to explain affordances when they were needed. As was explained in the response to RQ1, the universe offered to the students was action-oriented and clause links such as “to,” “in order to,” “so that,” and “that way“ pervaded. Once these functional explanations were provided, the students would not ask questions, they would simply use the function/affordance that was needed. In other words, explanations were instrumental;

explanations were never conceptual (concepts were not even mentioned, except for "variable" that was mentioned only once).

The Students' Linguistic Identity Mattered in Key CT-CP Practices

Although, generally, English dominated activity in terms of frequency of use, Teresa's teaching conceded bilingualism a prominent role. On May 15, 2017, right after the closure of the AOLME 2017 Spring implementation, Teresa had expressed her opinion concerning the students' use of Spanish and or English during the program. To the question "How did the use of Spanish and or English facilitate the process of learning or doing the tasks?" , she responded: "Depending on each student, using the language they are most comfortable with helps them learn the material better". Thus, for the facilitator the students' linguistic identity, seemingly, mattered as is shown below.

Next, I include and discuss excerpts of design, modeling, testing, and debugging which happened in Spanish. I also show the final code, which included Spanish as well.

Spanish was used in CT-CP Design Practices

The following excerpt of Text #5 illustrates the facilitator constant validation of the student design-related proposals, most importantly Juan's, who suggested including Spanish in the video. Interestingly, Juan was the only participant student who, while understanding Spanish, could not speak it. The excerpt starts with Herminio's explanation of his idea for the story and for its organization in frames. I have bolded the interactions where the facilitator validated students' design related ideas. Below the excerpt, Figure 25 illustrates the last frame of the video with "FIN" (The end) in big red letters.

Line 12,004 H: Miss, it's gonna probably be like this [pointing at his designs drawn on the sticky notes the students used to organize the frames or scenes of the story]

Line 12,005 And then when it falls

Line 12,006 It’s gonna get bigger, so.

Line 12,007 F: Yeah.

Line 12,008 H: Here is smaller

Line 12,009 and then here a little bit bigger,

Line 12,010 **H: and then here bigger.**

Line 12,011 F: Yeah,

Line 12,012 Yeah, Yeah

Line 12,013 H: so, when it hits Juan's character

Line 12,014 it should already **be pretty big.**

Line 12,015 H: Mhm

Line 12,016 F: Should be big.

Line 12,017 H: That's how big it's gonna end, [Draws the asteroid on the stickynote]

Line 12,018 J: Really?

Line 12,019 It’s just gonna kill all of them,

Line 12,020 It’s gonna be like, oh my God!

Line 12,021 H: It doesn’t explode in them so,

Line 12,022 J: Just squishes them?

Line 12,023 H: Yeah,

Line 12,024 F: Squishes them

Line 12,025 H: Yeah,

Line 12,026 H: And then, nothing.

Line 12,027 J: Then it should be like dust

Line 12,028 H: Yeah

Line 12,029 F: So, the last one is gonna be just black.

Line 12,030 it’s gonna be pitch-black

Line 12,031 J: And it's gonna say

Line 12,032 H: The end.

Line 12,033 M: The end

Line 12,034 J: Fin.

Line 12,035 F: Fin

Line 12,036 It's gonna say fin,

Line 12,037 okay

Line 12,038 M: We should take fin into the show.

Figure 25

Fin's Video Frame: Tangible Evidence that the Students' Language Identity Mattered

As can be observed, all students' statements were validated by the facilitator who either used "yeah" or repeated students' suggestions. In Line 12,029, the facilitator contributed to the story with the pitch-black color in response to Herminio's "And then, nothing" (Line 12,026). Then, both Herminio and Michael suggested "the end" (Lines 12,032-12,033). Unlike the facilitator who did not respond, Juan contested their proposal by proposing "Fin," which was later programmed by Michael as the video frame5 demonstrates tangibly.

Spanish was used in CT-CP modeling Practices and Communication of CT-CP Abstractions

In Text# 10, Michael used Spanish to engage the language arts teacher in helping him model with mathematics the tree of the video background. In doing so, he communicated to her the mathematization codes he had used to model the tree of the video background. This kind of abstraction-based and highly objectified communications were common ground, mediating a variety of collaborative configurations in a variety of CT-CP practices, as will be shown in the response to RQ3.

The following excerpt illustrates Michael's agency both in convincing the language arts teacher to help him and in its mathematization, initiating her in CT-CP modeling practices. Michael led the interactions that took place. Note how he convinces the teacher to help him in lines 14,348-14,362 and how he leads the communication of abstract mathematized codes by commanding the teacher in the imperative mood (Lines 14,370-14,385). In the part that centers on communication, the teacher wrote on a piece of paper the tree mathematization codes that Michael told her, which he used later to program the tree.

Line 14,339	F:	Did you write the codes? [to M]
Line 14,340	**T:**	**¿Ya tú acabaste de**
Line 14,341		**escribir todos tus códigos?** (Did you already finish writing all your codes?)
Line 14,342	M:	For the tree?
Line 14,343		Ya (Okay)
Line 14,344		I'm doing it
Line 14,345	T:	Okay,
Line 14,346		¡Pues vamos! (¡Come on, then!)
Line 14,347		Ay (Oh, dear)
Line 14,348	**M:**	**¡Ven a**

Line 14,349 **hacerlo!** (Come and do it!)

Line 14,350 T: ¡Vamos!

Line 14,351 M: ¿Por qué no, eh? (Why not, eh?)

Line 14,352 T: No, si eres tú. (No, it is you)

Line 14,353 M: ¿Eh? (What?)

Line 14,354 T: ¿Quieres

Line 14,355 que vaya? (Do you want me to go)?

Line 14,356 M: A ver. (Let's see)

Line 14,357 **Y los voy a hacer más rápido.** (And I will do them faster)

Line 14,358 T: ¿Si? (Yes?)

Line 14,359 M: ¡Vamos! (Come on!)

Line 14,360 T: Bien, pero me tienes que

Line 14,361 ir diciendo tú. (Alright, but you gotta be telling me)

Line 14,362 M: Muévete acá (Move over here)

Line 14,363 T: Okay.

Line 14,364 Entonces yo voy escribiendo y

Line 14,365 tú me vas diciendo, ¿sí? (Then, I will be writing, and you will be telling me, okay?)

Line 14,366 Tú vas mirando y

Line 14,367 me vas diciendo (you keep looking and telling me)

Here started the communication of abstract CT-CP mathematization codes:

Line 14,370 M: Doce por doce (twelve by twelve)

Line 14,371 T: ¿Aquí? (Here?) [F starts writing on a sticky note the coordinates of the tree as M dictates them to her]

Line 14,372 M: Siete por siete (Seven by seven)

Line 14,373 T: ¿Aquí, al lado? (Here, on the side?

Line 14,374 **¿Son dos códigos o uno solo?** (is it two codes or just one?)

Line 14,375 M: Solo son dos, (it's just two of them)

Line 14,376		**Ocho, ocho** (Eight, eight)
Line 14,377	T:	¿Aquí? (Here?)
Line 14,378	M:	Son dos aquí. (It's two here)
Line 14,379	T:	Son dos (It's two)
Line 14,380	T:	¿Ocho? (Eight?)
Line 14,381	**M:**	**Ocho** (Eight)
Line 14,382	T:	Okay
Line 14,383	**M:**	**Trece, trece** (Thirteen, thirteen)
Line 14,384	T:	Trece, trece (Thirteen, thirteen)
Line 14,385	M:	no, no, no, digo doce, doce (No no, no, I mean twelve, twelve)
Line 14,386	T:	¿ Doce, doce? (Twelve, twelve?)
Line 14,387	M:	ajá (aha)

Note that clauses such as "Doce por doce (twelve by twelve)" were commands which had undergone ellipsis. With "Twelve by twelve" Michael was actually saying: "Write down 12 by 12" (seemingly, Michael was mistaking prepositions "by" and "to"). Therefore, Michael was using commands to urge the language arts teacher to take note of pieces of the tree with specific locations on the paper grid he used to design and model it. Seemingly, to support the facilitator and encourage Michael to keep on working the teacher said: "Ya tú acabaste de escribir todos tus códigos? (Did you already finish writing all your codes?)" (Lines 14,340-14,341). Michael's response in lines 14,348-14,349 "¡Ven a hacerlo! (¡Come and do it!)" and his subsequent "¿Por qué no, eh? (¿Why not, eh?)" (Line 14,351) and "Muévete acá (Move over here)" (Line 14,362) show his invitations to help him in modeling. His two commands in the imperative mood and the question addressed to the teacher indicate his agency. Then, Michael's "Doce por doce (twelve by twelve)" (Line 14,370) and his subsequent ellipted proposals of abstractions communicated with the language arts teacher exemplify his initiating the teacher in computational thinking communications in Spanish. The excerpt just shown exemplifies that Spanish and the students' language identity mattered in modeling CT-CP practices. Michael could have contested the use of Spanish initiated by the teacher, and turn to English, but he didn't. The facilitator could also have intervened in English, but she didn't. Importantly, it was a student move in Spanish which engaged the teacher in helping mathematize the tree, getting her on board in the systematic CT-CP procedure facilitated by Teresa. The excerpt just shown exemplifies that Spanish and the students' language identity mattered in modeling CT-CP practices.

Spanish was used in CT-CP Testing Practices

The students could test the result of the Python program they were developing with great immediacy and ease. In the following excerpt, Michael used Spanish to explain to the language arts teacher what he was programming, the head of the wither skeleton, and that he could simply run it to test the result of the program (Lines 13, 063-13,077)**.**

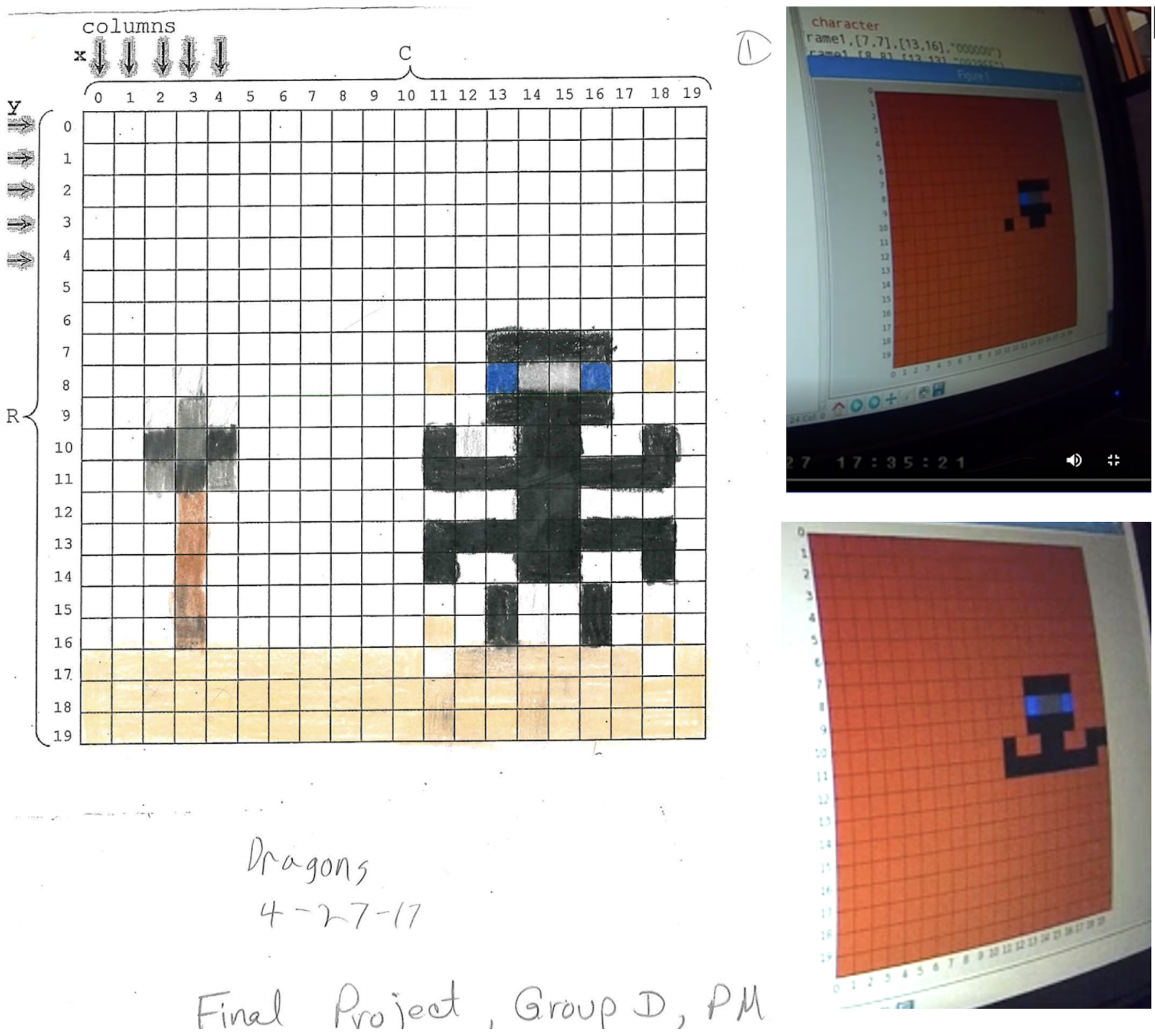
M: Ahorita esto es solo el código de mi,
mira solo esto
es el código de esto [grabs the grid ands point at it to show T]
T: ¿Solamente de la cabeza? (just of the head?)
M: Mira de esta de acá, de la cabeza (Look at this one, of the head.)
[pointing to the head on his paper grid]
T: Wow
M: De la cabeza (Of the head)
T: Entonces, ¿tienes que poner todo? (Then you have to put it all?)
M: Mhm,
eso es apenitas el pequeño código, (that is only the little code)
no estamos ni llegar al cuerpo todavía (we have not started the body yet)
T: Oh, God,
M: Y si quieres ver si está bien (And if you want to test it it's okay,)

le das a run (You click run).

Importantly, the systematic procedure to CT-CP that Teresa facilitated allowed instant testing (See Figure 26). As the students programmed their prototypes in Python code, they could—and they did in uncountable occasions—execute the program to test it and evaluate its result. Consistently, the students would call their tests out whenever they performed them. They or the facilitator would produce short proposals such as: “run it,”” F 5,” “watch,” “look,” or “See the magic happen” as Michael used to say. Clicking F5 would trigger participants’ cheering comments if the results were satisfactory—for instance, the facilitator’s pervasive, “there you go”—or assistance if needed. Below, Figure 26 shows the prototype of the wither skeleton and how far Michael was in its digitalization at two testing moments.

Figure 26

Wither Skeleton Prototype and Two Results Obtained After Testing the Python Program Under Development



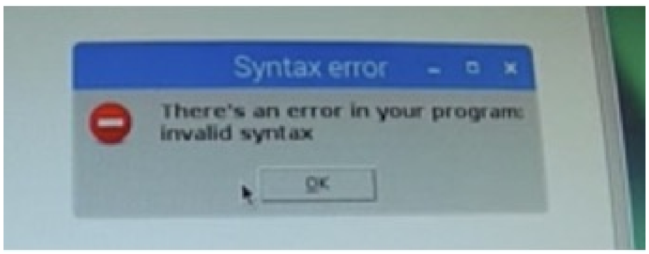
Testing the code involved the evaluation of the outcome produced, that is, evaluation of the effect that the Python code that the students were developing had on the images under development, and consequently immediate moves between LOAs. At a minimum, the students experimented with the LOAs involved at the prototype, the execution level, and the Python program level. The facilitator showed the students how to do the tests (just with a click of F5), and allowed them to do them as many times as they wanted and never questioned them, often without intervening.

When the facilitator intervened, she would say “click on the code” or “go back to your code”, which typically triggered comparison between the code and the image it had produced or the analysis of the error that had impeded its production. As time passed, seemingly, the students gained confidence, they used the F5 key with less and less frequency. Figure 26 shows the wither skeleton under development as Michael was coding it at the start of the creation of frame1. When Michael pressed F5 to “see the magic” displayed at the lower part of Figure 26, the computer system returned a crooked arm (note the left arm of the character with an extra pixel). The students were not satisfied with the outcome, thus, they accessed the code and fixed it. “You messed up right there” Juan said at Line 13, 494; to which Michael responded: “I’m curing him” (Line 13,500). The students were developing CT-CP. They were learning to evaluate the results executed by the computer, analyze the Python program they were developing and as will be discussed in response to RQ3. Hence, the students could see the effect that, for instance, a misplaced coordinate could have in the resulting digital image. As Michael had explained in Spanish to the teacher: “Y si quieres ver si está bien (And if you want to test it it’s okay) || le das a run (You click on run).

A problem of a different sort happened when the Python rules were not met. A missed parenthesis or too many digits in a hexadecimal number would impede the computer to process the Python program. The system would return an error (See Figure 27). Typically, syntax errors would be signaled on the screen with a notice, along with the line where the error happened. Then, debugging would start, and the group of CLD students would search for, identify, and fix the error—with the help of the facilitator and occasionally of other facilitators and even of students from other groups. After a particular error was fixed the students would test their program again and resume the programming of the video. Following I show one of these instances, in a debugging event mediated in Spanish.

Figure 27

A Syntax Error as Signaled to the Students by the Computer System

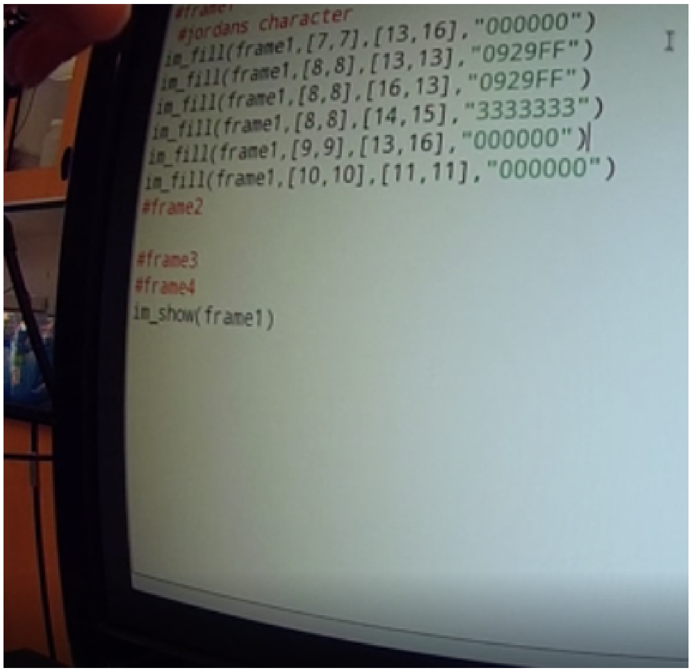


Spanish was used in CT-CP Debugging

Text # 11 showcases a long debugging event (See Appendix D) which was resolved in Spanish. Michael himself signaled the error that was impeding the execution of the program. More specifically, there was an error in the part of the Python program that coded the grey area (the nose) between the blue eyes of the wither skeleton. Note how in the fourth line of the code shown on the computer screen in figure 28 there is a 7-digit number in green, i.e., 3333333. This number, a hexadecimal number, was supposed to be constituted by just 6 digits. This syntax error, that is, the extra digit 3 that can be observed in Figure 28, was causing the failure and had been indicated by the computer system with the notice shown in Figure 27. The students, one of Teresa's colleague facilitators and the language arts teacher had tried to fix the error, but for some reason, they had not been successful.

Figure 28

One More Digit in a Hexadecimal Number



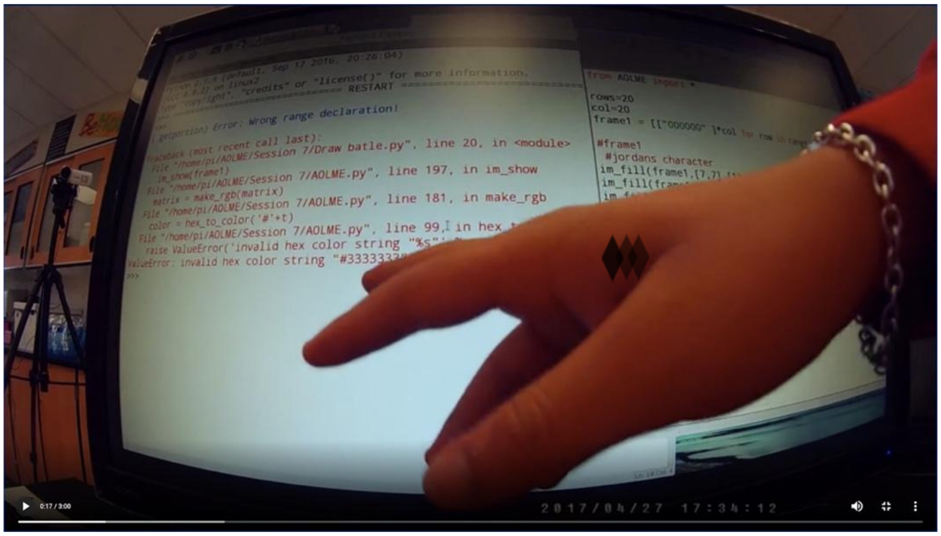
Following is the dialogue that happened and a picture of Michael's hand pointing to the error

(Figure 29)

Line 13,283	**M:**	**Right there**
Line 13,284		I am following the instructions like that and then
Line 13,285		**there's a color that is wrong**
Line 13,286		because, look,
Line 13,287		**this color is 3, 3 [pointing where on the screen],**

Figure 29

Michael's Finger Pointing to an Error on the Screen During CT-CP Debugging in Spanish



Line 13,288	M:	this one color is named 3,3,
Line 13,289		which is grey
Line 13,290	F:	So,
Line 13,291	**M:**	**are you sure** **it was 6 threes?** [To J]
Line 13,292	J:	Nods
Line 13,293	M:	It's like literally down here specifically,
Line 13,294		telling 3 numbers and there the names
Line 13,295	**T:**	**Los números esos son los colores, ¿verdad? (Those numbers are the colors, right?**
Line 13,296	F:	Sí. (Yes)
Line 13,297		A lo mejor, (May be)
Line 13,298	T:	¿Y qué está mal, el 3,3,3? (And what is wrong, the 333?
Line 13,299	F:	Está mal el 3,3 (The 3,3 is wrong)
Line 13,300	**M:**	**Lo estoy diciendo, que el 3,3,3,3,3,3 (I am saying it, the 3,3,3,3,3,3)**

Line 13,301	**F:**	**Oh, sí, ¿hay seis? (Oh, yes, are there six of them?**
Line 13,302		**¿Hay seis o siete? (Are there six or seven?**
Line 13,303	M:	Entonces digo que sí,
Line 13,304		hay 6 (Then I say the yes, there are six)
Line 13,305		una, dos, tres (One, dos, tres)
Line 13,306	T:	Siete (Siete)
Line 13,307	M:	Cuatro, cinco, seis (four, five, six)
Line 13,308		Oh (Oh)
Line 13,309		Chuckles [M corrects error with keyboard]
Line 13,310	F:	Siete (seven)
Line 13,311	T:	Chuckles
Line 13,312	**F:**	**Tenemos uno extra (We got one extra)**
Line 13,313	M:	Yeah, we have.

In line 13,283 Michael identified the error on the monitor screen, pointing with his finger the string of 3s. Apparently, he knew that the error was in the hexadecimal number and in the color that it coded, however, according to his proposition (Line 13,300) he was not totally clear on what was happening specifically. The error had been made on the line of frame which eventually, once corrected, appeared as `im_fill(frame1,[10,10],[4,5],"333333")` in the final Python program developed by the students (See Appendix E). Michael had said "Right there is a color that is wrong". Apparently, he knew that the string of 3s coded the grey color of the nose of the wither skeleton but was not sure what had happened. In Line 13,291 he asked Juan, seemingly trying to find a definitive answer to the problem: "are you sure it was 6 threes?" he had said. The language arts teacher also tried to help in this "community" debugging event, but, apparently, she did not know what was going on, which is shown in her comment in line 13,295 (Los números esos son los colores, ¿verdad? (Those numbers are the colors, right?). Seemingly, as typically happened when she spoke (typically in Spanish), her comment provoked a language code change. From that moment on, the interactions happened in Spanish, except for one turn. Michael insisted (Line 13,300) that the error was in the string of threes: "Lo estoy diciendo, que el 3,3,3,3,3,3 (I am saying it, the 3,3,3,3,3,3)". Finally, it is the facilitator who in lines 13,301-13,302 specifies what, specifically, was causing the error: "Oh, sí, ¿hay seis? (Oh, yes, are there six of them?)".

The above discussed "community" debugging event was not the only debugging event which featured other facilitators or students. On one occasion, up to nine people were involved in a "community debugging" CT-CP practice, but this one happened in Spanish.

Spanish was Included in the Students' Python Program

Eventually, as could be observed above in the digital image shown in Figure 25, the Spanish letters F, I, and N made it to the Python code developed by the students, giving closure to the students' animation video. Following is the Python code that Michel programmed to produce the video's last frame in Spanish. (Note letters I, and N, and that letter F was not included in the comments written after the hashtag.)

```
#frame5
im_fill(frame5,[6,6],[4,7],"a80a0a")
im_fill(frame5,[6,13],[4,4],"a80a0a")
im_fill(frame5,[9,9],[5,6],"a80a0a")
#letter I
im_fill(frame5,[7,8],[9,10],"a80a0a")
im_fill(frame5,[10,13],[9,9],"a80a0a")
im_fill(frame5,[10,13],[10,10],"a80a0a")
#latter N
im_fill(frame5,[7,13],[12,12],"a80a0a")
im_fill(frame5,[8,8],[13,15],"a80a0a")
im_fill(frame5,[9,13],[15,15],"a80a0a")
im_show(frame5)
```

The Facilitator Positioned the CLD Students as Capable Computational Thinkers in Key CT-CP Practices

Based on mathematics education research (Herbel-Eisenman & Wagner 2007; Morgan 2006; Rotman, 1998; Schlleppegrell, 2012)**, t**he following findings are supported on the assumption that the facilitator commands that targeted student CT-CP processes inasmuch as thinking-related (i.e., inclusive commands) positioned the students as capable computational thinkers, members of the CT-CP community and producers of their own creative computer technology.

The Participant-Speech function analysis revealed a constant use by the facilitator of inclusive commands, i.e, commands that urge students to think (Rotman, 1988). Interestingly, in the final part of the task realization stage (Text # 15), more specifically in the last half an hour of curricular activity, it was the students' commands which gained prominence, as will be shown. Across curricular activity, the facilitator positioned the students as capable computational thinkers and members of the CT-CP community in several key CT-CP practices. The students were positioned this way during (1) designing; (2) modeling prototypes with mathematics; (3) programming abstract CT-CP concepts such as Python functions; (4) testing and debugging, thus evaluating programming and its results, for which the facilitator positioned them in different LOAs; and, (5) algorithm communication and thinking, which was identified in the interpersonal communication of (a) the creation of a simple digital black and white image in Python after using and modifying a ready-made Python program and digital image (b) the algorithm that was eventually integrated in the Python code developed by the students.

Designing-Planning CT-CP Practices

In the following excerpt (lines 6009- 6017) of the task realization stage, the facilitator positioned Michael as capable in CT-CP design practices by means of inclusive commands. As can be observed, the command used by the facilitator in the excerpt (bolded) is inclusive since it provided the students with latitude in thinking how they wanted to design characters.

F: Michael, go ahead and **start planning** what your character is gonna do in the next frame.

M: I don't know,

M: We haven't talked about it.

F: agh, **you and Herminio talk about it.**

M: We gotta discuss this again,

M: what the shovel is gonna do.

H: Oh, maybe it's gonna go,

H: Oh, I got a good idea,

H: It's gonna go hiding under the sand, disappearing.

The students eventually programmed what was discussed in this excerpt. The shovel actually moved down in frame2 getting into the ground (See Figure 13)

Modeling Prototypes-Creating Computational Artifacts with Mathematics

The facilitator used inclusive commands to position students as capable computational thinkers and programmers in modeling practices where the students removed detail from "real-life" objects, selecting the 'common core or essence' (Kramer, 2007) regarding shape and color information. To position the students in CT-CP modeling practices to create computational artifacts, the facilitator used several processes such as "write down" and "figure out", typically involving the two-dimensional (2D) arrays that defined the students' drawn prototypes. Following are two examples:

"Go ahead and **write** your coordinates **down**. (Line 13,951) (Asteroid).

"I think we are gonna add the background, **|| Figure out** the coordinates" (Lines, 14,172-14,073).

When it came to color, the facilitator proceeded similarly. In the following example, the facilitator used the word "picker" to refer to the color picker app typically used to find the hexadecimal numbers the students used to define and program color. In this case, the facilitator was positioning Michael as capable in CT-CP modeling practices, inviting him to decide which colors in hexadecimal numbers he wanted to use for the background of the Python video: "**Use** the picker, **|| Figure out** the colors (Lines 15,061-15,062).

Programming Python Encapsulations

The facilitator positioned the students as capable computational thinkers and programmers, facilitating their programming of abstract concepts such as Python functions which encapsulated information such as Python's im-show. The Python function im_show made possible the display of the 2D arrays and, eventually, the video frames that the students programmed. Arguably, the bolded commands below "do an im_show" and "im_show" (Text #9) were inclusive because they concerned Python affordances that were needed for the students' program to work. Using the im_show function would allow the students to see the digital scenes that the students wanted to see, which they had designed as they chose. In this case, the facilitator was putting Michael in the position to use the Python function im_show in frames 1 and 2 for the students to be able to see the result of their programming work.

15, 643	F:	Oh, **put the im_show**
15, 644		hold on,
15, 645		hold on.
15, 646	M:	Oh, **im_show 2**, 2, right?
15, 647	F:	Yeah
15, 648		Do we have the imshow for the first frame?
15, 649	M:	I'm pretty sure, yeah
15, 650	F:	we don't
15, 651		Okay, so,
15, 652		after this let's put im
15, 653	M:	im_show
15, 654	F:	show, frame 1.
15, 655		and then **im_show** this one

Modularizing and Task Decomposition CT-CP Practices

Similarly, the facilitator positioned the students as capable in CT-CP modularizing and task decomposition practices by means of inclusive commands: And then after that, this is when **you do** another hashtag **||** So, what this does is just commenting *your* work (lines 12,193-12,194, Text # 7). Teresa was facilitating the very start of the video program urging Herminio to use hashtags to introduce comments. In another example, she positioned him as capable with the following clauses: "So that it doesn't look so clustered, **||** **put** like a hashtag **||** and **say** smoke[37]". The comments and modularization were used as can be observed in the video Python code that was developed by the students, which is shown in Appendix E. The comments were chosen by the students (most of them). Also at the start of the programming of the video the facilitator positioned the students as capable in CT-CP modularizing and task decomposition practices by

[37] Eventually, the word "smoke" was not used in the comments. "Dust" was used instead.

means of the command “put” used to involve frames: “and then **put** || the rows is equal to 20, || the columns is equal to 20, || and then your name begins, || just frame one, right? || and then, **frame is equal to**…(Note that the process “put” was ellipted to avoid repetition) (lines 12,178-12,183, Text # 7).

Bourke (2018) notes that “good code is organized, easy to read and well documented” (p. 14). Thus, the facilitator encouraged the students to organize their code and to document it by using hashtags for frames and video characters and background elements. In the AOLME graduation ceremony (May, 11, 2017), which was held right after the CT-CP practices under focus in this dissertation were finished, the facilitator stressed the importance of using hashtags:

> So, a big part that I tried to push on the boys was putting comments in their code, ‘cause I want them to be able to go back if they had missed something without getting lost or if they decided to share their code later on, the other person could see what they did as well.

As stated by Teresa and was discussed in the response to RQ 1, the facilitator invited the students to develop their Python code in a block structure. Each of the five frames developed by the students and the elements included in them was a block, that is “a section of code that has logically grouped together” (Bourke, 2018, p. 13). It has already been discussed that modularization facilitated code analysis and evaluation. In addition, it facilitated the decomposition of the students’ video project into smaller subtasks.

In positioning the students as capable in organizing their Python program in blocks comprising frames, and elements such as asteroids, spiders and trees the facilitator was also positioning them to *decompose* their computational task, i.e, programming a video in Python. In other words, the facilitator was positioning the students so that they could carry out a finite number of subtasks that would compose the whole Python code they eventually developed. As was discussed, the facilitator encouraged the use of sticky notes for this purpose, which helped the students make sense of the frames that would organize their story and the Python code that would enable it

Testing and Debugging (Analysis-Evaluation) CT-Practices

The facilitator positioned the students in CT-CP testing and debugging (analysis evaluation practices) constantly, and they self-positioned themselves constantly too. As already mentioned, the systematic procedure facilitated by Teresa afforded instant and easy testing which allowed the students to check the results of the Python program that they were developing or other short Python programs that had been developed for them. With a F5 click the students could run the program. Then, they could access the Python code and analyze what was in the code/was missing that produced expected/unexpected results. When the facilitator intervened, she typically used commands such as "**click** on the code" (line 13, 259) and "**let's go** to your code" (Line 13, 352). The former she used to position the students in debugging activity to identify and fix an error and the latter to position the students in analyzing-evaluating CT-CP activity, to analyze the code and evaluate what needed to be done to change the color of a part of the background of the video. Thus, this kind of facilitator discursive moves positioned the CLD students as capable CT-CP analysts-evaluators.

The Facilitator Positioned the CLD Students as Capable Communicators of CT-CP Abstractions to Use, Modify and Create a Python Program and Digital Image

The use of processes such as “talk,” “read off” (codes),” and “communicate” was salient in the facilitator discourse. Teresa used these processes in the form of commands, positioning the students as capable communicators of CT-CP abstractions. Positioning the students as communicators of CT-CP abstractions was important because these communications mediated the CT-CP collaborative modeling and collaborative coding among all participants that happened at different stages of curricular activity. For instance, Michael communicated to the language arts teacher the Y-X value coordinates that defined the shape of the tree of the video background and she wrote them down on a sticky note for subsequent programming; Juan communicated the abstractions of the dust of the asteroid of frame3 to the facilitator for her to program it; and Michael and Juan communicated coordinates for Herminio to program the asteroid of frame4 of the video. Unknowingly, I assume, the facilitator was engaging the students in a collaborative use-modify-create CT-CP learning progression (Lee et al. 2011) based on the communication of CT-CP abstractions. In facilitating this CT-CP learning progression, the facilitator positioned the students as communicators of CT-CP abstractions.

Use and Modify

In the task preparation stage, in an exercise which the facilitator identified as key in the students’ participation in AOLME (Interview held on May 17, 2017), Teresa engaged her students in *using* a simple Python program that had been created for them *adoc*. The program included a ten by ten “0” and “1” matrix (Figure 30). Each 0 in the Python matrix produced a black square (off) on the computer screen; each 1 produced a white one. When the matrix was composed only by ones, as in the figure shown following (Figure 30), the result was a white ten by ten-pixel digital image that could be seen on the computer screen. An all zero matrix would yield a black ten by ten-pixel image. The system allowed the students to turn any number of 1s into 0s in the matrix to turn them off and see them black on the resulting image. Equally, they could turn 0s into 1s to obtain a white pixel. Also, the students could turn pixels on and off directly.

Figure 30

Python Matrix Used to Create Black and White Pixels

```
[['1', '1', '1', '1', '1', '1', '1', '1', '1', '1'],
['1', '1', '1', '1', '1', '1', '1', '1', '1', '1'],
['1', '1', '1', '1', '1', '1', '1', '1', '1', '1'],
['1', '1', '1', '1', '1', '1', '1', '1', '1', '1'],
['1', '1', '1', '1', '1', '1', '1', '1', '1', '1'],
['1', '1', '1', '1', '1', '1', '1', '1', '1', '1'],
['1', '1', '1', '1', '1', '1', '1', '1', '1', '1'],
['1', '1', '1', '1', '1', '1', '1', '1', '1', '1'],
['1', '1', '1', '1', '1', '1', '1', '1', '1', '1'],
['1', '1', '1', '1', '1', '1', '1', '1', '1', '1']]
```

With a click on their mouse (by changing zeroes and ones), they produced changes on the digital image and, consequently, on the Python matrix that created it. Following are the words that the facilitator used to describe in our interview (October 10, 2017) the activity in this exercise and what she was doing to facilitate it:

> It gives you the coordinates in the activity, right, and you put ones and zeros, and then at the bottom it gives you some code that is supposed to be the code that you have already programmed, but like they programmed it for you and so what I wanted, when I came down, ‘cause previously they had asked about it, so I came down and I showed them the, the code and I told them that **the same way that**

you can go up and change the zeros to ones, **you can also do that in the code**, if you just change either like the coordinate or the number.

Originally, the Python matrix had been programmed for the students to yield a digital smiley face made up by black and white pixels, which the students modified. Following the suggestions of the facilitator, the students turned on and off pixels of the smiley face and then updated the image, observing how it changed. The facilitator was teaching them how to create digital patterns in Python.

Communicate CT-CP Abstractions and Create Collaboratively

After the students used and modified the ready-made Python code and black and white pixels in the smiley face activity, the facilitator engaged them in creating their own digital pattern in Python. The students *created* collaboratively through communicating CT-CP abstractions their own digital image, a letter D. The CT-CP learning progression facilitated by Teresa resembles the ones discussed in Lee et al (2011) but with a new ingredient: the participant communication of CT-CP abstractions. At first, in the task preparations stage, the students communicated abstractions between themselves to create collaboratively a black and white digital image. Later, in the task realization stage, all participants were engaged in communicating CT-CP abstractions as will be shown.

Following the facilitator commands to 'communicate' and 'read off codes' the students *communicated CT-CP abstractions*, reading off to each other and coding into the computer in Python the specific locations (coordinates) where they wanted to locate the white and black pixels that formed the D (See Figure 31) for the name of their group (the Dragons). The facilitator declared (May 17, 2017):

One of them had the keyboard and mouse and the other one had the paper, and they were reading off the coordinates and then the other two that weren't doing really anything there, they were still doing something in the sense that they were making sure that they had like the right numbers and that they didn't skip a number on the screen.

Figure 31

The Initial of the Name of the Group in a 10 by 10 Squared Paper Grid

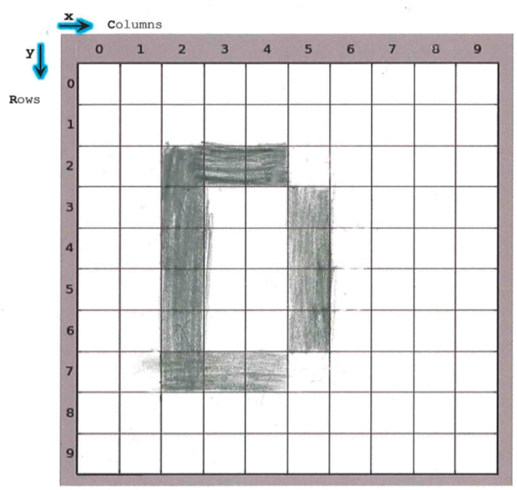


Figure 31 displays Letter D (D of Dragons, i.e., the groups' name), as it was drawn in pencil by Michael. As explained by the facilitator, the students were coding collaboratively, keeping *joint attention,* and communicating CT-CP abstractions, that is, specific coordinates and zeroes and ones that coded their digital creation in Python. Each pencil-drawn square on the 10 by 10 paper grid shown in the figure became a digital pixel (black or white) after being coded by the students in Python. While urging the students to communicate the codes, Teresa guided their work between the pencil-drawn letter, the Python code, and its resulting digital image. The facilitator indicated in the interview of October 10, 2017, that this activity had triggered student collaboration in AOLME, bonding them as a group while preparing them to program their video in Python. The following interview declarations together with the discursive moves used by the facilitator evidence that the facilitator promoted student collaboration and communication of CT-CP abstractions. The above-mentioned social condition to collaborate, strongly promoted by the facilitator through clauses such as “you have to work together” (RQ1), in combination with inclusive commands which positioned the students as collaborators are evidence that the facilitator seemed to associate student communication and collaboration. Only two weeks after the end of the program, when asked in an interview (May 17, 2017) about what she would highlight about the students’ work, the facilitator declared: “I think one of the big highlights was them sharing the keyboard and mouse and kind of take turns when they programmed ‘cause it influenced not only like collaboration but communication and how important communication is.”

The students rotated a wireless keyboard which, in combination with taking turns in programming, facilitated student communication and collaboration. This was corroborated by the students’ mathematics teacher who, on June 5, declared:

> What I have noticed, ehm, just with them being collaborative is them being able to hand off that keyboard, you know, and then be supporting the person who is actually in control of the keyboard, like you need to do this, you need to do that and, and making them accountable for rotating the role of who is actually entering the information (indecipherable word) into the computer.

“Don’t hesitate to || communicate with each other || if you feel like you are stuck” (Line 3950-3953) said the facilitator during the students’ collaboration in the student creation of their digital D for the Dragons in Python. This activity was pointed by Teresa as key in the student’s learning trajectory in AOLME: “This was kind of like the turning point for the group” she stated. The facilitator appeared convinced that the students were able to connect the Python code and what it enabled.

> They were so excited to like finally connect the coding to one of the activities, ‘cause this was in the beginning of AOLME and, you know, the reason why most of them joined was for the coding aspect, so to finally see that connection, they were all very excited.

According to the facilitator, the students had experienced an “aha moment” in session four where they had connected how the Python code enabled the digital images. The students had learned, as she said in the interview, how to transform each pencil-drawn square in the paper grids into a pixel by means of Python. The student communication of coordinates, 0’s and ‘1’s, which had mediated their collaborative creation of the digital D that stood for their name. It had “clicked for them,” as Teresa stated.

According to the facilitator, in their collaborative creation and modifications of digital images, the students had made meaningful connections between the Python code and the creation of digital images, that is, between different LOAs. Apparently, the use and modification of the *adoc*-programmed smiley face and their abstraction-communication mediated creation of a digital D had prepared the students for upcoming greater Python challenges (the video). To conclude this section, before I center on how the facilitator positioned the CLD students at different LOAs, I include below a brief excerpt that shows the facilitator positioning the students as communications of CT-CP abstractions at the very end of their creation of the video, about half an hour before they finished it (Text #15, lines 16,524-16,527).

F: Here, Michael

Come help,

come help Herminio,

tell him the coordinates

Again, the facilitator, by using inclusive commands (e.g., "tell him the coordinates") positioned the students as communicators of CT-CP abstractions and capable computational thinkers and programmers. As I will demonstrate in the response to RQ3 Michael actually communicated to Herminio the abstract CT-CP mathematization codes which Herminio needed to finish programming the students' video in Python code.

The Facilitator Positioned the Students at different LOAs

Following is an excerpt of the dialogue that mediated Teresa's facilitation of the above discussed activity (task preparation stage) where the students eventually created a digital D in Python after having modified a ready-made image programmed in Python. The facilitator introduced the students to LOAs by means of the command-suggestion starting with "Let" (Halliday and Matthiessen (2014) argue that the let form is in between commands and suggestions) (Lines 4023-4046; Task preparation stage). She then used commands.

F: Let me show you something.

Okay, you see this

This is all of your **coordinates.**

So, later on, later on, later in this session, we’re gonna learn

to start

doing this with our **images.**

M: Will it be easier or harder?

F: Well, see, you want, you know,

you can actually, you can change your **numbers** on here and

it would change the code too.

So, like this is what you put in **Python.**

This is the **code.**

F: So, say

you wanted it extra like smiley

and you wanted that one

to be a one and

then **put** the **zero** right here instead

and **put** another **zero** here.

M: You could go through there?

F: Yeah, you could go through here

So, let’s see. Where…

H: Row 6.

F: Yeah.

So, this is based off of **the eyes**

So, it’s all of these.

These are all the **rows** and then

it switches everything else for you

So, where in this one eye, which eye is this one?

By means of the inclusive command “put” in the clauses “put the zero right here instead” and “put another zero here”, and by telling them that changing numbers “switched everything” (the digital pattern of the face), the facilitator positioned the students at different LOAs. The facilitator was drawing the students’ attention alternatively to the eyes of the smiley face and to the coordinates, rows, ‘1’s,‘0’s, and the Python code that created its eyes. The facilitator was moving student attention from the "concreteness” of the digital eye of the smiley face that they could see on the image to the “abstractness” of the zeros and ones of the matrix in the Python code which enabled its shape. As explained by the facilitator, with a simple click, the students could change any zero and one in the digital image or they could change the Python code directly. In Teresa’s words, “they’d learned about zeros and ones and how each box was a pixel, or each, yeah, each square was a pixel… and how zeros and ones were black and white”.

Later, when the students were engaged in programming their video (task realization stage), the facilitator continued positioning the students at different LOAs. As was mentioned above, in the testing and debugging events that happened during the realization of the video, the facilitator used commands such as “**click** on the code” and “**let’s go** to your code”. These commands also positioned the students in different LOAs. They positioned the students between up to six LOA, from the execution level to the English/ Spanish/code switching they used in their discussions. (See section on LOA in the response to RQ1 above).

Summary

In this section, I have demonstrated for the participant CT-CP facilitator the students' knowledge of the world, both non-academic and academic mattered. The facilitator guided curricular activity taking into consideration and integrating the students' non-academic experience into the CT-CP practices that she delivered. Also, the prior academic level of the students seemed to matter since the CT-CP practices delivered relied on accessible mathematics and a heuristic systematic action-oriented procedure that focused on student action rather than on explanations of difficult CT-CP concepts.

In addition, design, modeling, testing, and debugging CT-CP practices happened in Spanish, and, importantly, the Python code and digital video produced by the students included Spanish. The fact that all these happened in Spanish proves that in the CT-CP curricular activity facilitated by Teresa the linguistic identity of the students mattered. In addition, I provided evidence that the facilitator positioned the students as capable in several essential CT-CP practices. These included designing-planning, creating computational artifacts by modeling prototypes with mathematics, programming Python abstract encapsulations, modularizing and task decomposition, testing and debugging (Analysis-Evaluation) CT-Practices. Finally, the Facilitator positioned the CLD students as capable computational thinkers and communicators of CT-CP abstractions which mediated the student collaborative creation of a simple black and white digital image after using and modifying a ready-made Python program and digital image. Positioning the students as explained, while preparing them for the collaborative creation of their own creative computer technology, a digital animation video in Python code also positioned them at different LOAs (up to 6 different LOAs).

Can the CT-CP universe offered by the facilitator to the CLD students and the way that she positioned them facilitate student development of CT-CP?

In this section, I demonstrate that the CT-CP universe offered by the facilitator to teach CT-CP to create a video in CP language Python and the way she positioned them facilitated student development of CT-CP. I demonstrate CT-CP learning in 5 CT-CP practices: Evaluation, abstraction, algorithm thinking, communication of abstractions and CT itself, creativity, and modularization. First the students (Text #12), after evaluating the result of the Python program they were developing, refined it to obtain a symmetrical image. Second, I demonstrate that the students communicated algorithm thinking and CT itself (Text # 15). Third, I show creativity, abstraction, and modularization in the discourse used by the students at the AOLME Spring 2017 graduation ceremony. In doing so, I show oral excerpts and the artifacts that they concern. I also resort to some secondary sources. I support my claims on theoretical grounds already discussed, which I drew from Lavie et al. (2018) and I recall next:

1) Learning can be identified in shifts in student discursive participation in product-oriented practices (Learning can be identified in student oral discourse).

2) Learning is a function of shifts in student participation in the following 6 characteristics of product-oriented practices.

Agentivity[38]: Participants make decisions in response to their own needs until they gain independence.

[38] Agentivity or agency can be identified when someone involves themselves in a process by means of the pronoun "I" and in commands, where they urge their addressee (s) to do something (Halliday & Matthiessen, 2014)

Objectification: Abstract objects such as integers substitute concrete objects and procedures.

Bondedness: The steps of a procedure feed into later steps towards obtaining a final product.

Flexibility: There is more than one way to perform a task.

Substantiation: Student explanations of what they did provide relevant clues on their learning processes.

Applicability: Refers to the idea that the lessons learned in a particular product-oriented practice and environment can be applied in a different practice and environment.

Given that the participant students were novices in CT-CP practices, it is assumed that the existence in the student participants' discourse of the above-listed characteristics implies that CT-CP development happened. In my explanations, I pay special consideration to the different LOAs at stake. To identify in the students' discourse whether the students exhibited CT-CP skills I used the following indicators:

Abstraction: Abstraction can be identified in student discourse if there is evidence of removal of unnecessary details and use of abstract representations.

Algorithm thinking: Algorithm thinking can be identified in student discourse if there is evidence that the students created/executed an algorithm, i.e., a "precise step-by-step plan or procedure to meet an end goal or to solve a problem" (Grover & Pea, 2018, p. 24).

Evaluation: Evaluation can be identified in student discourse if there is evidence that the students were engaged in finding an executable solution that fit their own purposes.

Creativity: Creativity can be identified in student discourse if there is evidence of their own ideas and imagination in their program and video.

Modularization can be identified in the students discourse if there is evidence that their program and video is divided in parts that make a whole.

The discussion on the findings that follow is supported with the artifacts produced by the students. I discuss excerpts from Text #12 and Text #15 which demonstrate that CT-CP development happened discursively as the students were engaged in curricular activity on the grounds that agentivity, objectification, bondedness and flexibility were constitutive of the discourse of the students. Later, at the end of this section, to demonstrate substantiation and applicability in student discourse I used the students' explanations of their own computer technology, i.e., their Python program and video, in their presentation to the families and AOLME team in AOLME's Spring 2017 graduation ceremony.

The CLD Students Refined their Python Program after their own Evaluations

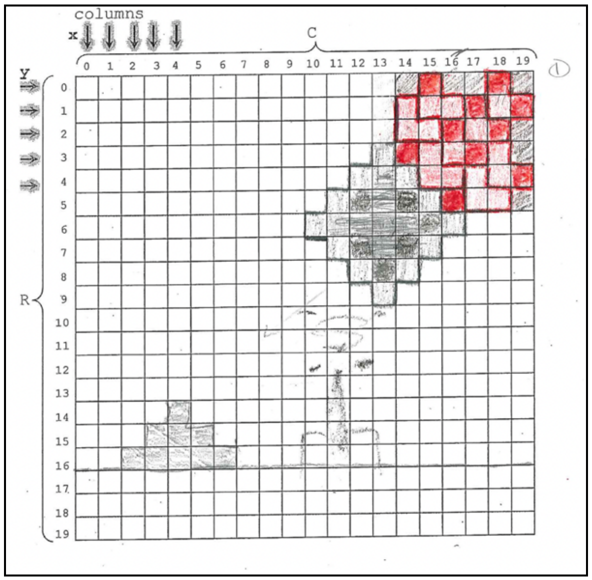
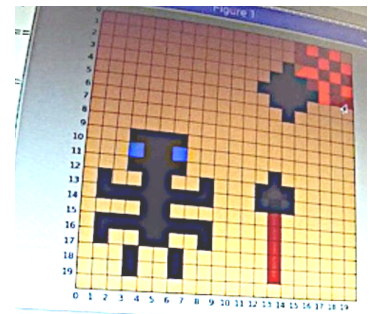
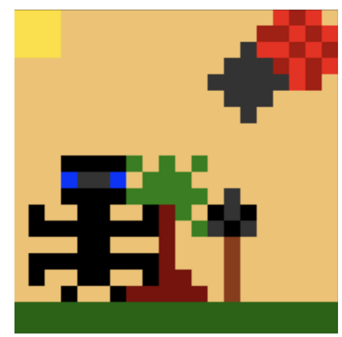
In what follows I demonstrate student development of CT-CP based on the *agentivity, objectification, bondedness* and *flexibility* shown in student discourse. What follows has its origins in the students' conceptualization, design and modeling with mathematics and programming the asteroid of frame1 or scene 1 of the video that the students created in Python. As discussed, the facilitator had offered the students a CT-CP universe that featured flexibility and positioned the students to periodically test whether the Python program they developed was correct and its result satisfactory. Errors triggered debugging and lack of satisfaction, modifications of the program to attain student desired shapes and color. Testing positioned the students in different LOAs.

The excerpt shown next (Text #12) illustrates a test run on frame1 of the Python program developed by the students. The discussion that the execution of the program triggered illustrates student CT-CP development. The students' discourse shows that after the work put in frame1, the result displayed by the computer was not satisfactory, at least to Herminio. Only a few seconds after testing, Herminio involved himself in the abstraction processes concerning the modifications that needed to be made. The processes Herminio (and Juan) involved himself in demonstrate student use of abstraction, algorithm thinking and evaluation.

I observed the group's reaction to the first run of the students' programming of frame1. Herminio had just pressed the key F5 to test the program. He, Michael, Juan, and the facilitator seemed to look at the screen with great attention. It seems that the group enjoyed the result (the digital image) displayed on the monitor but, apparently, Herminio was not 100% satisfied. As per Herminio's discourse, after *evaluating* the result displayed on the monitor, Herminio's sense of symmetry had not been satisfied: he wanted to take off a couple of pixels. The excerpt starts with Herminio's calling for attention (Lines 15, 162- 15, 179).

Figure 32

Original Symmetrical Frame 1's Asteroid Prototype, Asymmetrical Result, and Symmetrical Result after Correction

Frame1’s Asteroid Prototype	What was said	What could be seen seen on the computer monitor: Asymmetrical and Symmetrical Results
	**H:** **Look,** **watch it.** J: That's our biggest dun (son) M: Holy! F: And that's only the first frame. **H:** **We could have colored,** **take those 2 off.** J: No, it's okay, Herminio, It's fine. It hits like a rocket. H: It looks nice Too much work for that little one.	 

	H: **I wanna take those two off.** F: these two, okay, so 00 **H: 19 to 4, 19 to 5** **J: That's 19** F: Okay, So, let's go look at the code.	

Close attention to the upper right corner of the image shown first in Figure 32 allows noticing the asteroid's asymmetry that the participants could see at the very moment when Herminio said "Look, || watch it.". Note the arrow cursor, seemingly left there by Herminio, pointing to one of the two specific pixels that were causing the asymmetry (I have marked the other one with a red arrow). Herminio's statements "We could have colored || take those 2 off" (Lines 15,167- 15,168) anticipate, apparently, his desire to *refine* the asteroid red trail. A few lines later he insisted:*" I* wanna take those two off "(Line 15,182) which evidences his ***agentivity*** in this key moment of curricular activity. The Figure illustrates the original prototype of the asteroid as it was drawn by Herminio himself. As can be observed, his original drawing did feature the symmetry that he seemingly wanted to achieve. Frame1's definitive image (Also shown in Figure 32) allows noticing the final symmetry of the asteroid after the Python code was refined by Herminio. To Herminio's agentive statement ("I wanna take those two off"), the facilitator responded: "these two, okay, so 00" (row 0,0, i.e., the first row of the digital grid that frames the image, at its top), validating Herminio's words. Then, Herminio's "19 to 4, 19 to 5?" and Juan's "That's 19" (the rightest column of the image) exemplify the ***objectification*** of their discourse, where the abstract numeric representations they used defined the concrete shape of the asteroid that they had designed and wanted to digitalize. Also, the numbers they used are evidence of the *abstraction* processes that the students had experienced from the conceptualization of the asteroid through the prototype design drawn to final refined digital image. Unnecessary details of the asteroid had been removed by the students in the design and mathematization process. Now, Herminio wanted to remove, rather modify some detail. In addition, the students' discourse shows evidence of ***bondedness.*** Herminio's "19 to 4, 19 to 5?", and Juan's "That's 19" referred to one little segment, specifically one pixel of the trail part of the asteroid of frame1, which was one of the 5 frames that composed the Python program that the students developed.

Herminio and Juan hit the nail on the head with “19”, they were right on the column where one of the pixels that broke the asteroid symmetry was located. Herminio hesitated on the row where it was located though, “19 to 4 || 19 to 5”, he said—is it 4, is it 5? he seemingly meant to say. A few lines afterwards, the facilitator stated with accuracy the two pixels that Herminio apparently wanted to take off. The facilitator said: “And then we had, 4,4, 19,19” (Line 15,213[39]) (row 4, column 19). She also had said “Yeah, okay, so, || it was 0,0; 15,15” (Line 15,204-15,205[40]) (row 0, column, 15). Meanwhile, Herminio was still on the computer keyboard, apparently looking at the code, as the facilitator had suggested (Line 15,187). The facilitator's highly abstract and objectified statement communicated with accuracy to the group the two pixels that Herminio wanted to take off, which he was about to do by removing the lines of the Python program that encoded them. In CT-CP terms, *algorithm thinking* was involved in the students' and in Teresa's abstract numeric representations. Herminio and Juan's “19 to 4”, and “That's 19” encoded one of the two steps that needed to be removed. The other step was encoded in the pixel specified by Teresa with “0,0; 15,15”. The students had created these steps, i.e., they had created the algorithm, but when they executed them (the Python program that “contained” the steps), they were not satisfied. In other words, two steps of the systematic procedure that they had adopted to complete their task needed to be undone. The two steps (of the algorithm of their task) were inserted in the structure of the two lines of the Python program that Herminio deleted to remove the two pixels he did not want. Algorithm thinking was necessarily involved.

---

[39] Not included in the excerpt shown here but in Text #12 in Appendix D.

[40] Not included in the excerpt shown here but in Text #12 in Appendix D

To observe this removal of the pixels in the Python program, I display next the definitive version of the Python program of the asteroid of frame1 which was labeled as `#Herminio character` (`#dust` corresponded to its grey part). I have bolded and underlined the arrays that encode the pixel of the trail of the asteroid located at column 19 of the digital asteroid. Note that there are only three (rows 1, 2 and 3) The fourth one, the one which was originally at row 4, column 15 (See figure 32), is not there: it was deleted by Herminio as can be observed in Figure 32 (Symmetrical image). I have also bolded (no underlining) the arrays that encode the pixels located at row 0 of the digital asteroid. Note that there are only three (columns 16, 17 and 18). Likewise, the fourth one, the one which was originally at row 0, column 15 (See Figure 32) cannot be found since it was also deleted. In other words, Herminio deleted the following lines of code:

```
im_fill(frame1,[4,4],[19,19],"F90707")
```

and

```
im_fill(frame1,[0,0],[15,15],"F90707")
```

Here is the definitive version of the Python program of the asteroid of frame 1 (see Appendix E to find the whole Python program of the video)

```
#HERMINIO character
im_fill(frame1,[0,0],[16,16],"F90707")
im_fill(frame1,[0,0],[18,18],"F90707")
im_fill(frame1,[1,1],[17,17],"F90707")
im_fill(frame1,[1,1],[19,19],"F90707")
im_fill(frame1,[2,2],[15,15],"F90707")
im_fill(frame1,[2,2],[16,16],"F90707")
im_fill(frame1,[2,2],[18,18],"F90707")
im_fill(frame1,[3,3],[17,17],"F90707")
im_fill(frame1,[3,3],[19,19],"F90707")
im_fill(frame1,[4,4],[17,17],"F90707")
#dust
im_fill(frame1,[0,0],[17,17],"AF0000")
```

```
im_fill(frame1,[1,1],[15,15],"AF0000")
im_fill(frame1,[1,1],[16,16],"AF0000")
im_fill(frame1,[1,1],[18,18],"AF0000")
im_fill(frame1,[2,2],[17,17],"AF0000")
im_fill(frame1,[2,2],[17,17],"AF0000")
im_fill(frame1,[3,3],[16,16],"AF0000")
im_fill(frame1,[3,3],[18,18],"AF0000")
im_fill(frame1,[4,4],[18,18],"AF0000")
im_fill(frame1,[2,2],[19,19],"AF0000")
#dust
im_fill(frame1,[2,6],[14,14],"333333")
im_fill(frame1,[3,5],[13,15],"333333")
im_fill(frame1,[4,4],[12,16],"333333")

im_show(frame1)
```

Finally, Herminio's "We could have colored" combined with the agentivity of *" I* wanna take those two off" are signs of the **flexibility** that the students had been offered (see response to RQ1), which congruently was validated by the facilitator and executed after her "Okay, || So, let's go look at the code". In programming the asteroid and the rest of the video elements the students enjoyed the flexibility to program pixels that they wanted. The systematic procedure the students adopted allowed them to flexibly choose the shape and color of their characters. It was a matter of choosing the abstract representation that encoded the shape and color of characters such as the asteroid.

The students Led Communications of Algorithm Thinking and CT Itself

In this section, I also provide evidence of CT-CP learning. Text # 15, which happened at the very end of the AOLME program was decisive in the finalization of the video and illustrates the students taking the lead in the *communications of abstractions* and *algorithm thinking.* Interestingly, all the participants were involved in these communications. The separation of proposals and propositions that the agent-participant analysis afforded highlighted the agentive role that the students performed in this event. The constant commands used by the students shown below illustrate the *agentivity* of their discourse and its *objectification*. To lead in the *communications of abstractions* and *algorithm thinking* the students used commands, which proves their agentivity; and the numeric representations (arrays) of their discourse, its objectification. Also, as in the previous section, it was evident that the content of the communications indicated *bondedness*, that is, the excerpt shown features and evident incremental nature where each most participant turns contributed towards the consecution of their final product, the video. Finally, as in previous occasions, the student discourse evidenced *flexibility*. In other words, the students could have decided to design frame4's asteroid differently, with a different shape and color. The shape that they chose determined the two-dimensional arrays that defined it and thus where communicated. Accordingly, a different design would have yielded other mathematization codes, thus a different algorithm, Python program and, consequently, different pixels, images, and video.

Text # 15, which happened at the very end of the AOLME program, was decisive in the finalization of the video and illustrates the students taking the lead in the *communications of abstractions* and *algorithm thinking.* Interestingly, all the participants were involved in these communications. In the excerpt shown, while Michael mathematized the trail of the asteroid of frame4, he and Juan communicated its shape, that is, its defining two dimensional arrays (row and column coordinates) to Herminio, who coded it in the Python program the students had been developing since the start of the task specification-realization stage. The language arts teacher helped with the effectiveness of the communications by repeating the numeric representations conveyed by Juan and Michael to Herminio. The facilitator also helped.

What follows happened right before the facilitator and the students presented their video in the Spring 2017 AOLME Project Graduation Ceremony in the students' school library. Only a few days after, on May 15, 2017, the facilitator declared in a written questionnaire that it was at the end of the AOLME project when the students had made the connection between mathematics, CT and CP. The excerpt starts after the moment (already shown) when the facilitator positioned Michael to communicate to Herminio the arrays of coordinates of the asteroid that needed to be programmed in Python in order to finalize frame4 and the video. Soon after she positioned Michael this way, all participants were engaged in a collaborative event based on said communications. It happened as follows:

Here, Michael, || Come help, come help Herminio, || tell him the coordinates Lines 16,524-16,525). Not long afterwards, the language arts teacher got involved. She said: "Herminio, te puedo ayudar? || (Herminio,. Can I help you?). Herminio responded: sí, venga aquí (Yes, come here) (Lines 16536-16,537). Juan joined too. While Michael (M) and Juan (J) were mathematizing the drawing of the dust (trail) of the asteroid of frame4, Herminio was coding in Python the arrays that Juan and Michael were telling him. The facilitator (F) was standing in the back supervising the activity without intervening. The language arts teacher (T) made sure that there was no failure in the objectified oral communications that were going on (Lines 16, 605-16, 647). The center column of Table 25 (below) shows the highly abstract and objectified discourse that was produced to facilitate the mathematizations and programming. Its left column illustrates the prototype that the students were mathematizing. The right column of the table, the resulting Python code developed and the resulting digital image of frame4.

Table 25

The Students Take the Lead in the Communications of CT-CP Abstractions

Frame4 asteroid prototype modeled	Communication of abstractions	Resulting Python code developed and the resulting digital image of frame4

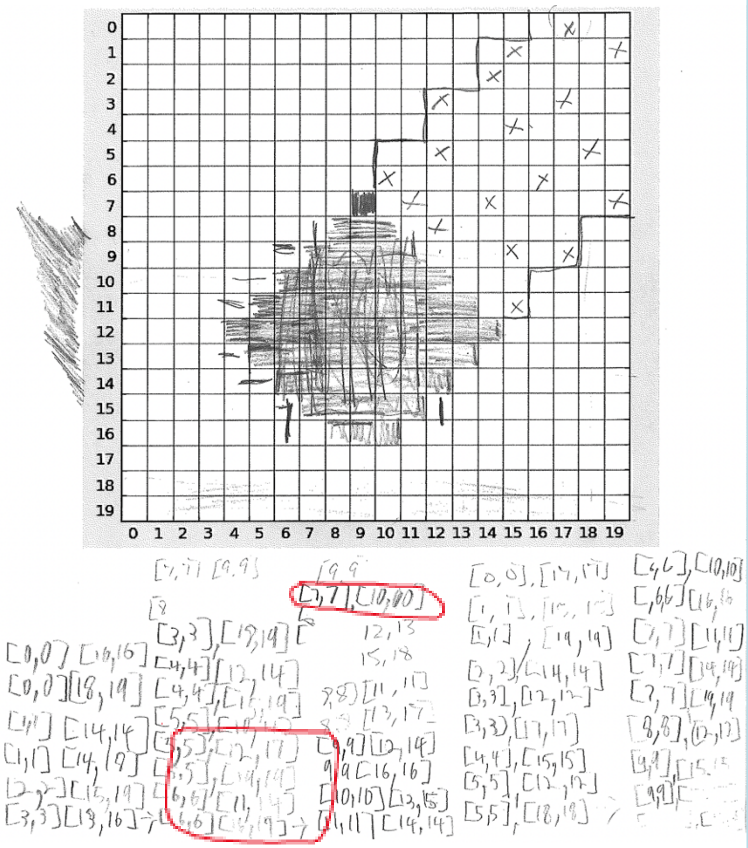
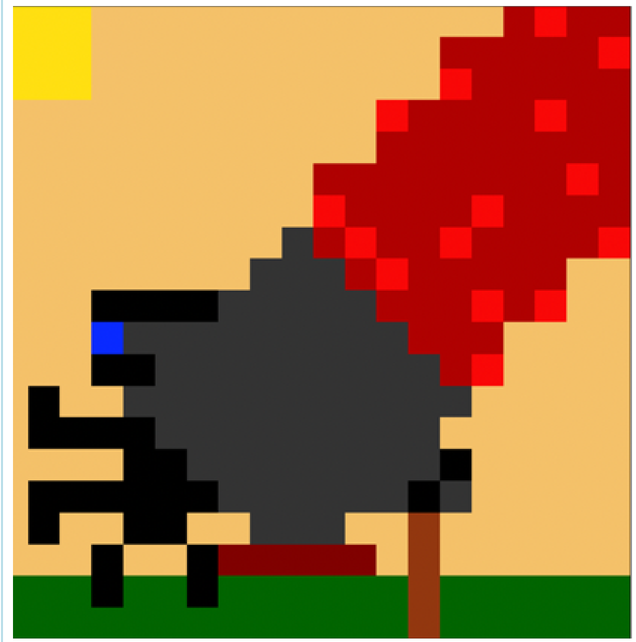
	**J: 5,5; 12, 17**
	T: 5,5,
	5,5;12,17
	J: Okay
	5,5; 19,19
	T: 5,5
	5,5 what?
	H: 19
	J: 19, 19
	T: 19, 19
	H: 19, 18 or?
	J: 19,19
	H: What else?
	J: 6,6; 11,14
	H: 11
	M: What the
	H: What else?
	J: 6, 6; 16, 19
	M: This is like so advanced
	T: Next one
	H: Next,
	M: Yeah
	7,7
	T: 7,7
	H: oh,
	7, 7?
	T: J, 7,7?
	J: **10**
		#dust
		`im_fill(frame4,[5,5],[12,17],"AF0000")`
		`im_fill(frame4,[5,5],[19,19],"AF0000")`
		`im_fill(frame4,[6,6],[11,14],"AF0000")`
		`im_fill(frame4,[6,6],[16,19],"AF0000")`
		`im_fill(frame4,[7,7],[10,10],"AF0000")`
		

Figure 33. *Frame4's asteroid prototype and the numeric abstractions that defined its shape. The ones that were communicated and programmed in this excerpt are inside the red line*	F: **10** M: Yeah, 10 H: 10,10? J: Yeah	**Figure 34.** *Asteroid look in Frame4*

Table 25 illustrates the highly abstract nature of the CT-CP practices that the CLD students of this study were engaged in by their facilitator. It shows the 5 LOAs that the students navigated in Text 15: (1) The Execution level or digital images, (2) the Python program developed by the students, (3) the student mathematization codes (in pencil), (4) the student prototype of frame4's asteroid, and (5) the natural language that was used (in this case, English). The asteroid prototype and the arrays shown were obtained by Michael in the mathematization process he performed. He noted them down[41] in pencil right below the prototype (Figure 33). The Python code shown was developed by Herminio, who was in command of the computer, while the participants' oral highly abstract, objectified communications displayed in the middle of the table happened. The reified result of these communications was the digitized image of the red part of the asteroid of frame4, which can also be observed (Figure 34).

Simultaneous attention to the excerpt provided and to both the pencil-written coordinates and the lines of Python code provided in Table 25 allow checking that the asteroid arrays communicated orally by Juan, such as "5,5 ; 12,17" originated in the coordinates written by Michael (the ones within the red line) and ended in the Python program that Herminio was developing. The two-dimensional arrays (sets of coordinates) or segments of the asteroid communicated in this excerpt (center column of Table 25 above) originated in abstraction processes realized by Michael as he modeled with mathematics the prototype of the asteroid.

Basically, Juan and, occasionally, Michael and the facilitator were transmitting the arrays, Herminio was receiving and programming them on the laptop and the students' language arts

---

[41] Only a few lines later, Michael was substituted by the facilitator in the mathematizations and in the transmissions of coordinates to Herminio. When this happened, the facilitator used the pencil too.

teacher[42] was ensuring the effective transmission and reception of the communications. The facilitator was standing at the back, apparently, supervising the correct mathematization and communication of arrays. The excerpt of Text # 15 shown starts with Juan's command "5,5; 12,17" (Line 16,605) urging Herminio to code in Python the segment of the prototype defined in his command (Row 5, columns 12 to 17). Juan's command is inherently **agentive** and could be rewritten as: "Herminio, program the segment of the trail part of the asteroid (its red part) codified by the two-dimensional array defined by coordinates "5,5; 12,17". By means of ellipsis, his discourse was reduced to the essential. Juan's "5,5; 12,17" is an abstraction (and representation) of the asteroid that Juan and Michael (who is writing down the coordinates that define it) want Herminio to program in Python code. The ***abstractions*** used in his **objectified discourse** represented concrete and essential parts of the asteroid the students had conceptualized and designed. Unessential details had been discarded. To Juan's "5,5; 12,17", the language arts teacher (T) responded "5,5[43]" (Line 16, 606) and "5,5; 12,17" (Line 16,607), repeating Juan's codes, confirming, this way, the coordinates he had just communicated. Then, Juan re-confirmed by saying: "Okay" (Line 16,608). In like manner, the communication of CT-CP abstractions continued with the contribution of all participants, with more numeric representations transmitted, checks, adjustments, and requests of more. Three more objectified commands evidence Juan's agentivity:"5,5; 19,19", "6,6; 11,14" and "6, 6; 16,19". Each of them represented a concrete segment of the asteroid design of frame4 and each of them was programmed in Python by Herminio, as can be observed in the resulting Python code (Table 25 above). The last

[42] Interestingly, the language arts teacher had been engaged previously by Michael in CT-CP modeling practices.
[43] Note that many of the arrays communicated in the excerpt shown above were split, seemingly to facilitate their communication.

line of code programmed in this excerpt, that is `im_fill(frame4,[7,7],[10,10],"AF0000")` was mediated by Michael's command when he said "7,7" which was coded as `[7,7]`. Before this happened, though, apparently, he responded with a request by the teacher and by Herminio for confirmation. Both Juan and Michael responded "Yeah". Finally, Juan and the facilitator completed the second part of the array when both commanded "10", for Herminio to code `[10,10]`. Apparently, the facilitator was supervising the communications (see Table 25 above), only intervening once[44] in the excerpt when she said "10" to confirm Juan's command. The participants' mutual engagement was evident.

It is apparent that the facilitator was giving the students leverage to succeed in AOLME's shared enterprise. Not only had she positioned them as capable computational thinkers and programmers as was shown, but they had taken up the challenge agentively. It seems that the facilitator knew what she was doing in her management of the last minutes of the students' participation in AOLME. It was with good reason when she had declared right before the start of her AOLME experience on Feb 4, 2017: "I want the students to learn how to program and enjoy it. I want them to be confident in their skills" (in response to the question "What experiences do you want to provide to middle school students as a facilitator?"). On October 10, 2017, I interviewed Teresa and a few days afterwards the students' mathematics regular time school teacher[45], who was also part of the AOLME team. I discussed with each of them Teresa's

[44] The whole transcription of the interactions that constituted Text#15 can be found in Appendix D, including an interruption that took place within the excerpt shown which was not included in Table 25.
[45] The students' school mathematics teacher was typically present at the AOLME sessions where the students of this study learned about CT-CP. However, according to the video recordings and to my notes, while she provided valuable support, she did not participate actively in the specific CT-CP educational practices that the students received in AOLME.

teaching during her facilitation of CT-CP activity to the CLD students of this study. Teresa declared “I was trying to give them leverage, do it by themselves. I was still standing there, but I was trying like not to comment too much on things”. The mathematics teacher declared about the facilitator the following: “La clave es cuando ella les da tiempo para hacer sentido y otras veces intercede (the key is when she gives them time to make meaning, while other times intervenes). Teresa’s and the students mathemathics school teacher declaration echo the *talk move* “wait time”, typically used in teaching to let students think, not interrupting their talk (Chapin et al., 2009; Michaels & O’Connor, 2019).

In Lines 16,626-16, 627, Juan’s “Hold up, || he’s still writing it.” show that Michael was still mathematizing and noting down the arrays of the part of the asteroid that Juan was communicating to Herminio. It is also apparent that the students' CT-CP discourse was developing, that is, that the students were learning CT-CP . Each of the arrays mathematized and communicated contributed to the students’ task, that is, the video. Therefore, the excerpt shown is evidence of student discourse **bondedness**. Also, Juan’s “5,5;12,17” (and the whole excerpt) is evidence of student discourse **flexibility** because the students could have chosen to design, for instance, a wider asteroid. Instead of “5,5;12,17” (row 5; columns from 12 to 17), the students could have used “5,5;12,18” (row 5; columns from 12 to 18) which would have resulted in a one-pixel wider asteroid. Thus, the students were creating (Michael, mostly), communicating (Michael, Juan, and Herminio mostly) and executing their own algorithm (Herminio on the computer), that is, their “precise step-by-step plan or procedure to meet an end goal or to solve a problem.” (Grover & Pea, 2018, p. 24). Therefore, *the students were communicating algorithm thinking* to create the video in Python that they had conceptualized and designed.

The programming of the trail of the asteroid continued. They were almost done. The colors were already in; the Python program almost completed. A few more abstract communications, executing the program to check the animation video for correctness and student satisfaction before the graduation ceremony, and that was it. Here is what happened next (Text #15):

Line 16,686 **M:** **So much work for one meteorite trail**
Line 16,687 F: 15, 16
Line 16,688 M: Oh, God
Line 16,689 F: 16, 19
Line 16,690 13 to 15
Line 16,691 M: [undecipherable, as he erases what may have been a mistake on the asteroid coordinates he was writing on the paper grid]
Line 16,692 **H:** **15**
Line 16,693 F: 13 to 15
Line 16,694 and then 11, 11
Line 16,695 **M:** **and 14, 14**
Line 16,696 **H:** **11, 11, 14,14**
Line 16,697 M: I think we should take this just
to show
how much work it takes.
Line 16,698 **H:** We've probably done the most
Line 16,699 **Do we put im_show or?**
Line 16,700 F: Yeah,
Line 16,701 Put im_show
Line 16,702 or do you have it already?
Line 16,703 Do you have the the im_show?
Line 16,704 H: No
Line 16,705 F: Then, now, put im_show

Line 16,706	**H:**	**Like what is it now?**
Line 16,707		**parenthesis?**
Line 16,708	F:	yeah,
Line 16,709		Parenthesis, frame 4.
Line 16,710		show is gonna be
Line 16,711		So, it's telling it
Line 16,712		to show frame, show frame 4.
Line 16,713		Alright.

Each new array of the asteroid communicated supposed a small increment that got them closer to their long awaited and desired outcome. Michael's propositions in lines 16,680 ("So much work for one meteorite trail") and 16,697 show that the students were seemingly aware of the load of work that they had put in the video project; Herminio's response in 16,698 (We've probably done the most) expresses both that a lot had been done and that they were almost done. The students persisted in the communications of arrays and programming until the end (eg., Lines 16,695-16,696), as they had been persistent in previous practices such as debugging their developing Python code until fixing troublesome errors. Importantly, Herminio's "Do we put im_show or?" (Line 16,699) and "Like what is it now?", || parenthesis? (Lines 16,706-16,707) prove student knowledge of Python abstract affordances and rules such as *im_show* and the parenthesis, and the need to use them timely. Indeed, they both were needed for the program to work and for the video to display, avoiding thus more debugging events.

Michael's and Herminio's statements "so much work for on meteorite trail" and "We've probably done the most" in the middle of their communications of mathematical arrays show, apparently, their awareness of the fact that a lot of arrays had been already mathematized, communicated, and coded during and before Text # 15. And, that a few more were still to be communicated before the asteroid or meteorite of frame4 was completed. These small communicative signs probably show that the students were aware of the iterative and incremental nature of CT-CP they were facilitated, that is, array by array, logical and unambiguous step by logical and unambiguous step (like algorithms). In other words, pixel by pixel.

It is apparent that the students were thinking and communicating (tantamount processes, according to Sfard (2008)). The students had individualized the flexible mathematics-based systematic procedure adopted (routine) and could thus communicate it (Lavie & Sfard, 2019). It also was evident that the students were developing CT-CP discourse. Thus, in Sfard's (2008) terms, the students learned CT-CP. In other words, the students participated agentively in established CT-CP practices such as creating algorithms, algorithm thinking, creating computational artifacts and refining them after testing (Grover & Pea, 2018); they used keywords such Python function im_show; the students used visual mediators such as their prototypes; and used CT-CP endorsed narratives such as the accepted-as-fact "truth statements" (Ben-Ari, 2012). They used (their algorithm) and communicated for the computer to function and digitize their prototypes into images.

The CLD Students Communicated Computational Thinking Itself

The students had refined the communication system that had started in the exercise where, relying on reading off codes, they had collaboratively created their black and white D for "The Dragons". Now that in the last moments of their participation in AOLME, the group was pressed for time, communication effectiveness (Sfard, 2000) seemed more necessary than ever. Following Sfard, communication effectiveness was needed because a need for operational precision was at stake. The process of communicating the arrays met two needs. On the one hand, the communication was founded on agreement of intention, i.e, all students wanted the same, to finalize their video. To do so, they had to finalize programming frame4, the last frame they programmed. More specifically, when the communications of Text # 15 happened, the students had already agreed on the shape of the asteroid that they wanted to obtain. They had the common intention of programming it so that it was big, to destroy the other characters of the video. On the other hand, the students needed operative precision because communication with a computer is unforgiving (Burke, 2018), with “truth” (mathematical) as a fundamental premise (Reeves & Clarke, 1990). There was an evident need for effective communication, which was even more so, because the group was pressed for time (The graduation ceremony was approaching fast). Effective communication was mediated by the “truth” that the mathematical arrays that they were communicating afforded. The participants were attending the same focus (i.e, the specific arrays that were being mathematized and communicated), which defined with absolute precision the shape of the asteroid. In doing so the participants shared focus on the mathematizations and communications of arrays such as “5,5;12,17”, for Herminio to code them in Python. Thus, coding in Python arrays such as “5,5;12,17” was necessary in order for the students to obtain the pixels that were needed to finalize their desired image of the asteroid of their video’s frame4. Doing so meant finishing their own computer technology so that they could present it to the families and the rest of the AOLME team in the graduation ceremony. It was going to be their contribution to the CT-CP community.

In the communications just shown in the two excerpts of Text # 15 displayed it is apparent that Michael, Juan and Herminio (and also the facilitator and teacher) targeted effective communication that both humans and computers can process (i.e., computational thinking). ***I***

contend that the students, in communicating the algorithm of their video, i.e., each of the mathematical two-dimensional arrays that constituted it, were communicating CT itself.

The Students Presented their own Python Code and Video at the School Library

In the school library, on a digital screen, the students explained to their families and the AOLME team their own creative computer technology, i.e., the Python program and video that they developed. *The students presented their reified response to the CT-CP practices that Teresa and AOLME had offered them (Wenger, 2010), their learning as represented in the images that they produced* (Kafai & Resnick, 1996). The student bilingual explanations of what they did provide relevant clues on what they learned. After the students presented the video and explained the story that it told they centered on the Python program that enabled it. **Substantiation** and **applicability** of student CT-CP learning is evident in the discourse that they used in the presentation. The students were able to explain what they learned in a different context to where learning happened (applicability). In their explanations, several CT-CP practices can be identified (substantiation). These are discussed next.

Creativity

Herminio started the student presentation by explaining the story that the video told. As previously discussed the story included the students' personal and imaginative worlds.

> So, lo que está pasando aquí es que so, eh, so, la araña y el, la pala están peleando. Entonces empiezan a ver que el asteroid está cayendo y se empiezan a esconder y ya cuando cae pues ya no se dieron cuenta y pues les cae arriba. (So, what is going on here is that, so, ehm, the spider and the, the shovel are fighting. Then, they start seeing that the asteroid is

falling and they start hiding and when the asteroid approaches them they had lost attention and it falls on them."

The spider, shovel and asteroid Herminio involved in his speech and the processes he used to explain what happened are evidence of CT-CP student creativity.

Modularization

Herminio started the students' presentation of the Python program by saying "So éste es es el código que usamos para hacer cada parte de las fotos" (So, this is the code that we used to make each part of the pictures). The fact that the student explained that pictures had parts and that the Python code was used to make each of their parts signals modularization in student discourse thus substantiation of student learning of CT-CP modularization.

Michael's discourse also reflected CT-CP modularization. His very first words to present the Python program developed were the following:

"This is the code to like the different images, they are like the background. And then we got like we made the characters". He also said*, "this is the code for the tree".*

Michael's explanations in this short excerpt show how while showing the Python program to the audience, he involved in his speech the Python program and the different images or frames and characters. Specifically, his use of the prepositions "to" and "for" signal his association of the Python code to different parts of the video. His use of the identifying process "is" putting in relation the code and the tree also signals modularization

Abstraction and Precision.

Herminio centered on abstraction and precision, associating them. He involved in his discourse the mathematical abstraction "333333" identifying it with a real-world color. i.e., grey. He explained that the Python function "fps", meant the speed at which the video frames moved.

Then he centered on another abstraction, that is, hexadecimal numbers. He explained that in using them, precision was a must by means of the modal verb "can" in its negative form and the qualifier "specific".

> So, el color, so hay 3, so por eso son 6 números. Primero, el primero rojo, verde y azul. So, como 333333 es el gris. So, no más puedes poner cualquier número o cualquier letra, tiene que ser uno específico, so un específico número. (So, color, so, there are 3, so, that's why they are 6 numbers. First, the first red, green and blue. So, like, 333333 is the grey. So, you can't just put any number or any letter, it has to be a specific one, so a specific number).

It's worth noticing that, while talking about these kinds of numbers, Herminio did not involve the name typically used to refer to them (i.e., "hexadecimal"), he just referred to the alphanumeric abstractions that were used. Michael proceeded similarly. His discourse reflects CT-CP abstractions too: *"These are the coordinates which you decide like which block you want to go to… the coordinates for where you want the, the little ehm pixel to appear or like squares."*

Michael involved the coordinates in his explanation of the Python program and related them to the blocks or squares of the 20 by 20 paper grids that the students used in the *modeling* of their prototypes and to the resulting pixels. Importantly, his use of the processes "want" and "decide" indicates *flexibility*. As was explained in response to RQ1, the facilitator offered them significant freedom to decide the shape of their characters (in fact, as was explained, they were offered significant latitude in everything that concerned design).

LOAs

In the excerpts shown of the student presentation of their Python program and video, Michael's and Herminio's discourse evidence the following LOA:

(1) The Execution level or digital images: the students referred to the pixels of the video and to the story that was shown in the video.

(2) The Python program developed by the students. The students referred to parts of the program and to Python functions and mathematical abstractions used in it.

(3) The Design level: The students referred to the blocks and squares that made possible the mathematization of their prototypes.

(4) The natural languages that were used: Spanish and English.

Chapter 6

Discussion

This dissertation explored CT-CP in an authenticCT-CP environment to better understand its nature and teaching and learning. Using a SFL-Case study methodology, I examined the discourse of a CT-CP facilitator and that of a small group of CLD middle school students and their Language Arts teacher over a whole introductory course while they were engaged in CT-CP practices. These practices aimed at students creating a digital video in Python from the pixel level. It was demonstrated that the novice CLD students learned CT-CP.

This study was driven by two problems: the limited participation of CLD students in CT-CP/CS practices (Goode, et al. 2020; Santo, 2020) and the challenges that the teaching and learning of CT-CP/CS present (NRC, 2011; Grover & Pea, 2013; Peng et al. 2019; Reppenning, 2016). Both challenges are especially relevant for CLD students in abstract contexts (Colombi & Schleppegrell, 2002; Vogel et al. 2020). In what follows, I first summarizethe findings chapter: I explain the CT-CP teaching/learning that happened, a CT-CP practice discovered that was not identified in the literature, that is, the communication of algorithm thinking and CT, and the teaching/learning experience. I then discuss a CT-CP teaching model aimed at bridging complexity based on salient strategies used by the facilitator. Next I discuss the relevance of the findings. Finally, I discuss study limitations and suggest areas for future research.

Summary of Study Findings

Discourse is both a resource to represent the universe of our experience (I relied on this idea to address RQ1 and RQ3) and a resource to establish interpersonal relationships, used by interactants to position each other (I relied on this idea to address RQ2). Thus, based on the indicators detailed in the methodology chapter which signaled what was realized linguistically, I explored the meanings that were realized by the participants. Following the flexible methodology design, I triangulated findings through varied data sources, theoretical and analytical perspectives (Tracy, 2019). Yet the analysis presented in this study constitutes just one way of understanding the nature of the CT-CP practices offered to the students, the ways in which they were positioned, and whether CT-CP learning happened.

This dissertation investigated a real and successful mathematically-based CT-CP educational environment where a small group of middle school CLD students learned CT-CP through text-based computer language Python. I examined the spoken discourse produced by their facilitator, the students, and their Language Arts teacher during an introductory CT-CP course, with respect to the Python program and images students produced.

The SFL-case study perspectives and methods used yielded the following seven main findings:

- *Discursive illustrations that indicate CLD students' CT-CP learning.* This finding is theoretically grounded on Lavie et al.'s (2018) theorizations on student learning. These are based on their development of discourse, and I extended them to student development of CT-CP discourse.
- *A previously unreported (to the best of my knowledge) CT-CP educational practice*: *student algorithm-CT-CP communication.* Algorithm-CT-CP communication is the simultaneous communication of algorithm thinking and computational thinking. The students established algorithm-CT-CP

communication both with the computer and between themselves, the facilitator, and the Language Arts teacher.

- *Discursive illustrations that indicate (in the facilitator's discourse) the characteristics, main agents (the students), processes and CT-CP practices of a real CT-CP educational environment.* The practices identified were abstracting, algorithm-CT-CP communication, creative designing-planning, modeling prototypes-creating computational artifacts with mathematics, using Python abstract encapsulations, modularizing, task decomposition, being iterative and incremental testing and debugging (analysis-evaluation).
- *A unique use-modify-create communicative learning progression (Lee et al., 2011) which prepared CLD students for algorithm-CT-CP communications* (to the best of my knowledge, communicative learning progressions have not been previously reported).
- *Six levels of abstraction (LOA):*
    - Execution level: the digital images produced by the students.
    - Program level: the student-created Python program.
    - Mathematical representation level: the mathematization codes that constituted the algorithm created by the students.
    - Design level, pencil/crayon student prototypes of video elements and characters.
    - Image level, thought-provoking images researched in telephones.

- Natural language level, English/ Spanish/code switching used in brainstorm and ongoing discussion.

- *Thirteen CT-CP ways of knowing*: CT-CP-as-practice, CT-CP-as-communicative, CT-CP-as-collaborative practice- CT-CP-as-procedural, CT-CP-as-creative, CT-CP-as-dependent on Mathematics, CT-CP-as-data driven, CT-CP-as-able to modify conceptions of mathematics disciplinary rules, CT-CP as-constant-move-from-problem to-solution, CT-CP-as-experimentation, CT-CP as-constant-move-between-LOA, CT-CP-as-fact, CT-CP-as-a flexible-experience.
- *An example of the power of SFL-based perspectives and methods to explore CT-CP educational environments* and yield relevant findings concerning the definition of CT-CP educational domains, the way facilitators position students, and student CT-CP learning.

Importantly, the facilitator safeguarded students' pedagogical rights (Bernstein, 2003) and students exercised those rights: (1) The students were invited to participate in CT-CP practices which relate to current and possible futures that they could imagine (Wenger, 2010) in a plethora of disciplines and fields where technology is produced (Dufva & Dufva, 2019), (2) The students were included in the CT-CP community personally (the facilitator considered students' linguistic identity, prior experience and interests), socially invited students as a group to form part of a CoP), and intellectually (the facilitator positioned students as thinkers, and (3) The students were encouraged to participate actively in the powerful CT-CP discourse (Goode et al, 2020; Santo et al, 2020) and to create their own computer technology, which the did, thus contributing to the CT-CP community joint enterprise (Wenger, 1998). The study adds to successful educational experiences with text-based programming languages (Fargan & Paine, 2017) which is relevant based on the limitations that visual programming tools such as Scratch reveal (Rose et al., 2020).

On the grounds that the students' discourse developed towards an evident objectification, agentivity, bondedness, flexibility, substantiation, and applicability (Lavie et al., 2018), it was found that the participant CLD students learned CT-CP. Student learning was mediated both in English and Spanish. The students discourse underwent an evident transformation from the one they used to communicate the concrete ideas that expressed what they wanted to see happen in the video that they programmed to the highly abstract and objectified discourse they used while modeling the prototypes of the video characters with mathematics and communicating the algorithm to each other (and to the rest of participants) during collaborative coding so that the computer could process them. The students' discourse was agentive as demonstrated by their continuous use of the pronoun I and of commands at key moments of curricular activity which they addressed to other students and also to the facilitator and the language arts teacher. In addition, bondedness and flexibility were identified, that is, the students' creations constituted parts of the whole product that they produced which they could have chosen to produce in very different ways. Finally, the students substantiated their discourse to explain their computer creation with relevant details in an environment which was very different to the one where they had created it. In other words, the CLD students applied what they had learned to present their computer program and video in the school library to their families and CT-CP community.

The learning processes experienced by the participant CLD students were facilitated by a female English-Spanish bilingual undergraduate student of engineering without professional experience in teaching. The discourse of the facilitator represented a CT-CP universe where the CLD students were the main agents of processes which were pervasively material, a universe whose main characteristics were complexity, pragmatism, procedurality, dependency and flexibility. To engage (Wenger, 2010) the CLD bilingual students in the CT-CP community practices that enabled CT-CP learning, the facilitator positioned them as capable producers of computer technology whose linguistic identity, i.e., English and Spanish, and experience in the world mattered. The students experienced 13 different ways of knowing (See Chapter 3). The facilitator discourse revealed a teaching practice which focused on student action and practicality rather than on conceptual explanations and discussions. Once the facilitator finished teaching the lessons of the written curriculum and the development of the video started, the facilitator left behind CP concepts, i.e., they practically disappeared from her discourse. Similarly, with the exception of testing, CT practices were practically not mentioned by the facilitator. Let me clarify that the practices happened but were not specifically mentioned.

To facilitate the student creation of their computer technology and CT-CP learning, the facilitator involved the students in a motivational, mathematics-based, heuristic, and systematic, procedure which featured some social conditions such as collaboration and agreement and some disciplinary conditions such CP rules and mathematical formalisms that had to be followed. However, the CT-CP universe that the facilitator offered the CLD students featured flexibility too. The students were invited to tell the story that they wanted to tell and to include in the video the characters and background elements that they wanted, with the creative shape, size, and color that they chose, as long as they fit in 20 by 20 square grids. Recommendations for efficient coding were discussed too. In facilitating this systematic procedure which led to the student production of a technological creation, the facilitator positioned the students as capable computational thinkers and computer programmers.

The students drew the prototypes of said video characters and elements with pencil and crayons to then model them with mathematics. Thus the students used mathematical formalisms such as two-dimensional arrays and hexadecimal numbers to flexibly obtain the mathematical representations that encoded the shapes and colors that they wanted to program. These mathematical representations defined each of their characters and background elements unambiguously and could, therefore, be programmed. These representations constituted the algorithm of their task, a series of logical and unambiguous steps that the computer processed to digitize their pencil and crayon prototypes, executing the lines of code programmed by the students in Python.

The systematic procedure adopted mediated CT-CP learning and was based on human communication and communication with the computer. The mathematics-and-creativity-based algorithm created by the students as a result of modeling featured the kind of logic, abstract representation, and unambiguity that computers can process. The students' algorithm could at once be understood by the students and the facilitator—and to some extent by the language arts teacher—and processed by the computer. This particularity made possible the highly abstract communications that happened and the kind of collaborative programming that took place.

Collaboratively, the students programmed the video, at times engaging their language arts teacher and the facilitator in its sequential, iterative, and modularized creation processes. Several salient collaborative and communicative configurations took place which illustrate the mutual engagement (Wenger, 1998) experienced by the students in the CT-CP community. For instance, two students communicated mathematical representations of shapes to another student for him to program them in Python; one student communicated mathematical representations to the language arts teacher for her to take note of them so the student would later program them in Python; one student mathematized one character and communicated the facilitator its defining two-dimensional arrays for her to program them; finally, two students modeled with mathematics the shape of the prototype of one character and communicated its arrays to a third student for him to program the arrays in Python while the language arts teacher and the facilitator checked that the communication between students had the necessary operative precision (Sfard, 2000). In these communications, alignment between CT-CP CoP members was evident (Wenger, 2010). What was being communicated was the algorithm of the video, which could be at once understood by the participants and processed and executed by the computer. Therefore, in doing so, the students were communicating to each other, to the rest of participants and to the computer the thinking processes involved in formulating a task in ways that a computer can carry out, thus CT (Grover & Pea, 2018). In other words, *algorithm thinking and CT itself were at once communicated.* This finding had not been anticipated at the start of this study. It was uncovered through the deep SFL-mediated scrutiny of the highly abstract discourse used by the participants at the end of curricular activity. Juan had said: “5,5; 12,17”, The language arts teacher responded “5,5, 5,5;12,17,” then Juan said, “Okay 5,5; 19,19” to which the teacher responded “5,5, 5,5 what?”. Then Herminio said: “19” and Juan confirmed “19, 19”. I highlighted these interactions for future deeper SFL-case study analysis. I traced back the meaning encoded in the numeric representations pronounced by the participants. They concerned the asteroid of frame4 (Text #15). I understood that Juan’s turns were actually commands, that the teacher was checking the accuracy of his communications and so was Herminio who was coding what Juan was requesting, as was explained in detail in Chapter 4. I traced the history of the abstraction processes (Hershkowitz et al., 2001) that had resulted in the participant's numeric utterances until I discovered what Juan's "5,5; 12, 17" encoded. I compared the Python code with the asteroid prototype and the discussions that had taken place about the asteroid before it was abstracted into numeric arrays. Everything matched. I discovered that "5,5; 12,17" was important. Furthermore, it was fundamental, just as the rest of the two-dimensional arrays that were communicated. They were all fundamental to the video that the students programmed in Python. They constituted the algorithm of the program that the students created to produce their computer technology. Not only this, but "5,5; 12,17" was also important because it was a command through which Juan was leading the algorithm thinking-CT-CP communication at that moment, which signaled the agentivity, objectification, bondedness and flexibility of his discourse thus learning (Lavie et al., 2018).

The communications of algorithm thinking and CT were explicitly encouraged by the facilitator. Not only this, but the facilitator also integrated the communication of algorithm thinking in a use-modify-create learning progression (Lee et al. 2011), which prepared the students to create digital images and video while allowing them to make connections between different LOA. In other words, to scaffold the learning of CT-CP the facilitator engaged the students in using and modifying a ready-made black and white digital image in which pixels could be turned on and off. The facilitator encouraged the students to observe that changes in the pixels that composed the image produced changes in the Python code that enabled its shape. In this way the facilitator positioned the students in two LOA (i.e, the digital image and the Python code). Then, the facilitator engaged the students in programming their own black and white image by having them communicate to each other the Y and X values of its pixels (While one student coded in Python the values, the rest communicated them to the student who coded them on the computer. This communicative use-modify-create progression got the students ready for the collaborative programming in Python of their video. In doing so, the students and rest of participants implemented the same kind of communications that they had practiced, that is, one student programmed in Python the algorithm called out by another student or by the facilitator. The students tested the program under development as often as they wanted with a simple F5 click. This way, they evaluated, analyzed, and refined the images that they created by modifying the Python code or debugged errors when called out by the computer system. Typically, testing and debugging triggered discussion centered on the look of the resulting image or what had caused the error, involving several LOAs.

Over the course of curricular activity, including the creation of the video, the facilitator engaged the students in six LOAs: (1) Execution level, i.e., the digital images and video displayed by the computer; (2) The student Python program; (3) Mathematical representations, i.e., the student two-dimensional arrays and hexadecimal numbers obtained in their modeling practices; (4) Prototype designs, i.e., pencil/crayon student prototypes of video elements and characters; (5) Images researched in telephones to stimulate the student creativity; (6) English/ Spanish/code switching in brainstorming and ongoing discussions.

The findings of this study suggest a bilingual, product-oriented, heuristic, and procedural model for teaching/learning CT-CP that, while aiming at bridging the inherent complexity of CT-CP, is supported theoretically on relevant perspectives and constructs discussed in Chapter 3.

Fundamentally, the model and discussion that follows is supported with the idea that CT-CP learning may happen as a result of collaborative and implicit mediation within the students' ZPD (Wertsch, 2007). Accordingly, the CT-CP teaching/learning model suggested below is based on several strategies used by the facilitator which may have created zones of proximal development (ZPD) by bridging CT-CP complexity, decentering complex explanations and discussions of CT and CP concepts at the expense of meaningful, fluent, and effective communication.

Relevance of Findings

The relevance of this study is not in demonstrating that CT-CP complexity can be bridged in the ways discussed, mediating learning implicitly through adopting a systematic procedure like the one uncovered or that doing so constitute effective strategies that result in CT-CP learning. Rather, I view that its relevance lies in that it identifies some strategies that can be used to help facilitators, teachers and students walk together the complex CT-CP learning processes. While the strategies identified are relevant to the current CT-CP educational field and apparent in the data, other interpretations are also possible. This dissertation uncovered algorithm-CT-CP communication, a CT-CP practice which has not been found in the literature and was adopted by the CLD students of this study to produce their computer technology. It is also particularly relevant that the facilitator engaged the students in a *communicative* use-modify-create learning progression (Lee et al., 2011) which indicates CT-CP learning and was unique in that it prepared the students for the above-mentioned communication of algorithm-CT-CP communications. In addition, it is important that the CLD students were engaged in creative designing-planning practices, modeling prototypes-creating computational artifacts with mathematics, using Python abstract encapsulations, modularizing the Python program in parts such as characters and frames (video scenes), task decomposition, being iterative and incremental, testing and debugging (analysis-evaluation). Finally, the CT-CP ways of knowing and LOAs experienced by the students are relevant because they open gates for future learning experiences and the SFL-case study perspectives and methods used because they, hopefully, open ground for future research.

Teresa's Bilingual Teaching Model to Bridge CT-CP Complexity

Teresa's model is fundamentally based on the students' active participation in the creation of their own computer technology. The focus is pragmatism and student operational fluency in the CT-CP practices experienced as opposed to engagement in CT-CP conceptual discussions. The model includes the following components:

(1) The students' intellectual resources and interests: Their languages, culture, and prior experiences.

(2) A motivational, pragmatic, mathematics-based, heuristic, and systematic procedure.

- Preparing the students for CT-CP abstractions and LOAS.
- Algorithm-CT-CP communication and collaboration.
- Conditions, Flexibility and Efficiency.

Bridging Complexity with the Students' Intellectual Resources: Their Languages, Culture and Prior Experiences

The findings of this dissertation suggest a CT-CP bilingual pedagogy that bridges CT-CP complexity by promoting students' agentive participation (Wenger, 2010) in the production of computer technology (Peng et al, 2019) that is truly meaningful to them, such as the video produced by the students. They suggest that CT-CP practices that capitalize on the languages, culture, ideas, interests, and expertise both academic and non-academic of each student can bridge CT-CP complexity. This applies more specifically to CLD students, whose lack of motivation has been reported to originate in a disconnection between formal and informal knowledge (Jacob et al., 2018). However, in this study, the CLD students were positioned as capable computational thinkers and computer programmers of digital technology significantly based on their prior non-academic experiences. Therefore, it highlights the importance of the role that facilitators and teachers play isn positioning students for learning (Chval et al., 2021). This study, where the facilitator promoted the use of the language the students felt more comfortable with and bridged academic and student non-academic knowledge, corroborates the findings of Collins et al. (2021) which emphasized the importance of culturally sustaining pedagogies that draw on students' language and culture, incorporating pop culture into CLD CT-CP/CS educational projects. Contrary to Jacob et al. (2020) who found traditional direct teaching methods to be more successful in CS K-12 education to multilingual students, this dissertation shows that student-centered pedagogies that position the students as the main agents of CT-CP activity can be successful.

Gathering Information about the Interests and Expertise of all Students

The dissertation findings suggest the adoption of CT-CP teaching methods that while fulfilling complex tasks/solve complex problems are based on students interests, and real-world experience (Polya, 2004) focusing on their 'islands of expertise' (Crowley & Jabob, 2002), that is, their cultural and linguistic assets (Celedón-Pattichis et al., 2018). As was done by Teresa, brainstorming and ongoing discussion can help teachers and facilitators in gathering this information to then incorporate into CT-CP curricular activity. The pedagogy suggested aims at teacher and facilitator's use of constructive dialogue and authentic questions, as opposed to test questions, to get at the ideas and expertise of all the students, ensuring their identification with the practices (Wenger, 2010) and inclusion of ideas from each of the students and their subsequent abstraction and transformation in the tangible computer technology produced. Some examples of topics in which the students may be the experts and were used by the participants of this study include the students' breakfast diet, their favorite video games and characters, outer space, and their favorite places and colors. Computing concerns processing data (Bourke, 2018). The information gathered by the facilitator was processed by the students in many hands-on activities in which they were invited to participate.

Capitalizing on the Students' Developmental Intellectual Stage

The findings suggest the soundness of taking into consideration the students' developmental stage (NRC, 2010; 2011) capitalizing on the abilities that characterize it. In particular, the facilitator of this study capitalized on the adolescent's ability to think abstractly and be creative (Vygotsky, 1998). The facilitator engaged the students in constant discussion (typically also allowing side conversations) about multiple creative ideas for the video project at hand. "Creativity is the interaction among aptitude, process, and environment by which an individual or group produces a perceptible product that is both novel and useful as defined within a social context" (Plucker et al., 2004, p. 90). The data of this dissertation points at pedagogies that, drawing on the students' culture and potential, stimulates their minds with the resources needed to perceive, think, imagine and ultimately create (Glaveanu et al., 2020). This study, like Israel-Fishelson et al.'s (2020) uncovered the importance of creativity to promote student development of CT thus points at drawing on it as a resource to bridge the complexity of CT-CP. It was shown, in the original story discussed and programmed by the students, that creativity mediated the CT-CP processes of abstraction experimented by the students. Thus, the findings of this study suggest the soundness of promoting the interplay of creativity and abstraction to solve complex problems and tasks, capitalizing at the same time on children's natural interest in exercising competence and searching for solutions to complex problems (Newell, 1981). As was illustrated in Chapter 5, the students were engaged in discussions that stimulated their creativity and then in abstracting their characters into mathematical representations through modeling processes which eventually resulted in their computation. In other words, the adolescent participants of this study were able to "express themselves and their ideas in computational terms" (NRC, 2011, p. 8).

Capitalizing on Student-known Mathematical Concepts and Implicit Mediation

The findings of this dissertation suggest that, rather than on CT and/or CP conceptual explanations to mediate CT-CP learning, the teaching of novices can rely on implicit mediation and on mathematical concepts already known by the students. The facilitator implicit mediation (Wertsch 2007) of task/problem solving and CT-CP learning included the communication of algorithms constituted by meaningful mathematical abstract representations (Polya, 2004) of the characters designed by the students that coded elementary concepts such as shape, size and location in the coordinate plane. Rather than focusing conceptual CT-CP reflection by explicitly engaging the students in CT-CP conceptual discussion (explicit mediation), the facilitator introduced every-day discourse which 'evolved in the service of communication' and became 'integrated with other forms of goal-oriented behaviour' (implicit mediation) (Wertsch, 2007, p. 185). The everyday discourse used by the facilitator evolved, in interaction with the students, into a co-constructed instructional discourse which underwent processes of abstraction and objectification in the service of communication (Sfard, 2000; Wertsch, 2007) and the logical and unambiguous "truth" that the computers can process (Ben-Ari, 2012; Bourke, 2018). Thus, the discourse used by the facilitator to help reorganize student cognitive activity, therefore, learning as curricular activity progressed, while often abstract and objectified, kept a stronger focus on action rather than on CP and CT concepts (except for testing).

Therefore, the findings suggest a pedagogy that relies on elementary mathematical concepts as a foundation for an initial student reflection on the difficult-to-learn CT-CP concepts and future solid development of said concepts. In particular, the facilitator of this study relied on elementary mathematical concepts associated with the coordinate plane, shape and color, which although conceptualized differently, the students already knew. For the most part, the facilitator did not target CT-CP conceptual explanations. Python affordances such as the Python function im_fill, for instance, were used as resources as they were needed, and the explanations about it were purposeful as was shown. Teresa's teaching practice points at pedagogies that focus on the use of CT-CP affordances such as Python functions as they are needed (Meerbaum-Salant et al., 2013) through pragmatic explanations, leaving conceptual discussions for a later stage. The findings also suggest that CT-CP learning can happen with practically no explicit mention of CT concepts (NRC, 2011), only testing was specifically mentioned and promoted by Teresa. Further research is needed to explore whether approaches without a strong focus on conceptual schema building, like the one presented here, motivate students to further their education in the CT-CP/CS field or lead to their disaffection with the discipline (Jabob et al. 2020). I will explain in the section devoted to future research that the solid formation of CT-CP concepts in student cognitive structures can wait until a later stage, in lessons and exercises that target student verbalization of what they actually did during their CT-CP practices (Lye & Koh, 2014; Waite et al., 2017). In Teresa's practice the focus was operational fluency, a necessary step towards subsequent deeper conceptual development (Österman & Bråting, 2019; Sfard, 1991b).

Thus, the findings suggest pedagogies where CT-CP concepts do not get in the way of CT-CP action. It is important to mention that CT-CP practices inherently involve thinking, having abstraction in their essence (Wing, 2008). The participant CLD students were thinking abstractly and at different LOA while modeling their characters with mathematics to obtain the mathematical representations of their size, location and color, while communicating the simple algorithm that they created (size, location and color), and while testing, debugging and analyzing their program and evaluating its results. Fundamentally, the students were engaged in abstract thinking processes to eliminate unnecessary details of their characters and get at their essence, which they programmed so that said essence could be processed by the computer (Kramer, 2007). Consequently, pedagogies that position students as the main agents of CT-CP practices position them to think and reflect about these practices, like Teresa did. Thus the findings suggest that, in a first stage, the focus be student CT-CP active participation and communication so that they can think without "interruptions" with difficult-to-learn CT-CP concepts (Mouza et al., 2020; Grover et al., 2015; 2016; Mahoney et al., 2008; Meerbaum-Salant et al., 2013). Therefore, as suggested by Arastoopour et al. (2019), CT-CP education should not be conflated with CP concepts. CT-CP education can rely on implicit mediation (Wertsch, 2007).

Bridging Complexity with A Motivational, Pragmatic, Mathematics-based, Heuristic and Systematic Procedure

The findings of this dissertation point at pedagogies that focus CT-CP teaching/learning through the students active participation in a method, "a heuristic procedure… [a method that can] "establish a firm connection between personal activity and the connection of formal knowledge (Papert, 1980, p. 58-59), specifically a motivational, mathematics-based, pragmatic, and systematic procedure where computational action is foregrounded (Tissenbaum et al, 2019). The focus should be on practices that aim at the fulfillment of a task or product and, while challenging, are accessible and interesting to the students, relate to mathematics concepts and other subject areas and place weight on experimentation, discovery, and evaluation of possible solutions (Newell, 1981; Papert, 1980; Polya, 1945, 2004, Lee et al., 2020). As was shown, Teresa's teaching facilitated the novice CLD students the challenging and complex task of creating a video in Python, placing weight on their testing the program under development as much as wanted by the students to ensure that it worked, and its digital results (images) satisfied the students' standards. Also, as suggested in heuristics, the systematic procedure facilitated by Teresa depended fundamentally on mathematics, more specifically on abstract concepts associated with the coordinate plane, shape, and color. The complexity of the task at hand was recognized also by Michael when he said "this is so advanced" in text 15 during the collaborative and communicative coding of the last frame of the video. Yet, the practices were accessible to the students as shown by their active participation and agentivity, for instance, in their planning and designing practices, their modeling with mathematics of their prototypes, and their communication of the algorithm that constituted their Python program.

Teresa's teaching suggests a pedagogy which is framed in procedures that hinge on student motivation, memory, and attention (Newell, 1981; Polya 1945). Indeed, the students were motivated to create their video. Not only were they recruited based on this criterion, but the facilitator made sure that the students told the story that they wanted to tell, with the characters and background that they planned, designed, and created. As the students implemented the procedure, they used their memory when they repeated the sequence that they had done previously, for instance when they successively modeled with mathematics each on the characters of the video to then program them. Finally, the procedure was based on attention, more specifically on collective attention, which was displayed in the algorithm communications that happened, which were based on an attended focus, that is, the mathematization codes that represented the prototypes, and the need for operative precision (Sfard, 2000). The systematic procedure suggested includes following a sequence and is inherently iterative.

A Sequential Task or Problem-solving Procedure

The findings suggest that CT-CP pedagogies include systematic procedures that focus on sequentiality and include the problem-solving steps identified by Newell (1981) and Polya (1945): Understand the problem, design a plan, carry out the plan and examine the solution. The analysis of the data of this dissertation revealed that the facilitator followed a similar sequence in facilitating the video creation task, which can be described as follows: (1) Student familiarization with CT-CP affordances and practices; (2) Brainstorm-ongoing discussion to plan and discuss intermediate goals such as the appearance of characters and background and the video frames constituents. Final goals such as the story that would be told, and the conditions and affordances available to produce the desired computer technology were also discussed; (3) Drawing of prototypes of the characters and background elements; (4) modeling with mathematics to obtain mathematical representations of prototypes shapes, sizes, locations and colors that the computer can process (these representations constituted the algorithm of the video); (5) Program the mathematized prototypes; (6) Test to analyze program results and refine program or debug if needed.

The procedure suggested is essentially divisible in parts. In other words, the procedure suggested promotes task or problem decomposition (Grover & Pea, 2018; Kong, 2019), that is to program each character, the students typically repeated the sequence 2-6 (see paragraph above). Hence, the Python program developed by the students included modularization (Grover & Pea, 2018; Kong, 2019) In addition, as was shown, the procedure facilitated by Teresa included collaboration, creativity, planning and designing, abstracting, modeling, algorithmic thinking, creating computational artifacts, reusing, and testing and debugging (Grover & Pea, 2018; Kong, 2019) and the communication of algorithm thinking and CT. In participating actively in these CT-CP practices, the CLD students explored the ways of knowing detailed above. Therefore, the findings of this study suggest pedagogies that, while centered on the facilitation of task-problem solving procedures that follow a sequence and divide tasks and programs in smaller parts, focusing also on fundamental CT-CP practices and ways of knowing, that is, targeting the shared repertoire (Wenger, 1998) of CT-CP CoP and disciplinary fields.

A Focus on Simple Algorithms that Center on the Students' Expertise and Interests

The study findings point at pedagogies that capitalize on the students' expertise and interests to engage them in practice (Wenger, 2010) and motivate them to create simple algorithms and programs. Information gathered from the students can be the one which they abstract to create their algorithms and programs in the computer language chosen. Processing this information can be key in establishing the firm connection between personal activity and formal knowledge that Papert (1980) wrote about. The algorithm uncovered in this dissertation was simple. It was constituted by the shape and color of the characters and background elements that the students wanted to see in their video, basically a logical and unambiguous succession of two-dimensional arrays and hexadecimal numbers. The algorithm that the students coded in Python to create their video had originated in their expertise, that is, Minecraft characters (A wither skeleton and a shovel) and an asteroid.

An Iterative and Incremental Procedure

The findings point at the adoption of procedures that are inherently iterative and incremental. Programs are typically developed iteratively and incrementally until they are complete (Brennan & Resnick, 2012). The collaborative and communicative coding practices that were shown in Chapter 5 (response to RQ2, Text # 15), which resulted in the completion of the meteorite or asteroid of the video were iterative and incremental. In each turn, the initiator transmitted a new piece of information (a two-dimensional array that coded a piece of a pencil-and-crayon drawn prototype) that the receiver coded in Python for its digitalization into pixels. Each array constituted a portion of the algorithm that constituted the video, a new step in its completion. Each step was iterative, steps extending to dozens of turns where each new step added an additional increment to what had already been coded. The students were aware of this, as reflected in Michael's statement "so much work for a meteorite trail" (Text # 15).

Preparing the students for CT-CP Abstraction and LOAs Communicatively

The study findings point at teaching methods that prepare the students communicatively to the inherent abstract nature of CT-CP and different LOAs (Wing, 2008) through collaborative experimentation with ready-made programs. The goal is to help develop thinkers "who can rapidly change levels of abstraction, simultaneously seeing things "in the large" and "in the small" (NRC, 2010, p. 48). Ready-made programs that the students can use and modify can help the students experiment at different LOAs and get ready to create their own programs (Lee et al., 2011). As was shown, the facilitator of this dissertation prepared the students for CT-CP abstraction and LOAs. Importantly, she added an ingredient not reported by Lee et al.: communication. Teresa prepared the students for the CT-CP abstractions and LOAs they would encounter in programming their video in two separate activities. First, she engaged the students in modifying the location of the pixels of a black and white digital image to observe the changes that it caused on the Python code that produced it. And then, she urged the students to communicate to each other the Y and X values that located the squares that they had shaded in pencil in a squared paper grid (a letter D) and wanted to digitize into a black and white image. The facilitator of this dissertation reported that creating the D digital image by communicating to each other the mathematical abstractions that created in Python the pixels of a digital image (the black and white letter D) constituted an eye-opener for the students, while bonding them as a group. One student coded in Python the abstractions that coded the "D" as were communicated by the rest. Then the students tested the program to check the result. In doing so, the students were able to observe the direct cause-effect relation between an algorithm, programming, and the resulting digitalization, thus, different LOAs. Then, the iterative communications of the two-dimensional arrays that composed the algorithm that the students created for their video (Text, 15) constitute a good example of the communication of algorithm thinking and CT-CP itself, which I contend can mediate CT-CP learning implicitly. These two activities and the subsequent communicative creation of algorithm and Python program that created the video constitute a *relevant example of a communicative use-modify-create learning progression.* The effectiveness of these practices in preparing the students for creating their own programs suggest the soundness of pedagogies that adopt communicative use-modify-create learning progressions.

Algorithm-CT-CP Communication and Collaboration

The collaboration identified in the systematic procedure adopted by the participants of this dissertation, which corroborates previous work that have associated collaboration with CT-CP reasoning and development (Chowdhury et al. 2018; NRC, 2011; Wu., 2019), suggest a pedagogy that fosters collaboration specifically. In this dissertation, explicit encouragement of collaboration and communication was found to be important at strategic moments of CT-CP activity, including the start of the students' video project and during key CT-CP practices. In addition, collaborative practices that blur the power differentials between teachers, facilitators and students were effective CT-CP practices adopted. This was illustrated in the response to RQ2 (coding of asteroid of frame4 (Text # 15)) where the participants' mutual engagement (Wenger, 1998) and effectiveness of CT-CP communicative practices were apparent. This kind of collaborative communication was also adopted by the students, for instance, to agentively lead (Michael) in Spanish the modeling of the prototype of the tree with the language arts teacher (Text #10), and to agentively lead (Juan) the coding of frame3 with the facilitator (Text # 14). As was explained, what was communicated in the collaborative coding practices of the video was its algorithm and CT itself.

On the grounds that communication and thinking can be considered in intimate interrelationship (Sfard, 2008), the facilitator-student and student-student algorithm-CT-CP communication uncovered in this study may be a fundamental practice conducive to CT-CP learning. The findings suggest the relevance of encouraging the students to communicate with others the constituents of the algorithms that they create. *Fostering student communication of the abstractions that constitute the algorithms of simple programs may induce CT-CP learning (through implicit mediation) while instilling in the students a collaborative attitude to programming.* It is important to mention here that, as was shown, as the students programmed their characters, they constantly tested the result for satisfaction. At any moment of the programming process the students could, and actually did, check the effect that each array (inserted in the structure of its corresponding line of code) had in the evolving digital image that they were creating. In this way, the students could establish cause-effect connections between their algorithm (which included the two-dimensional arrays that defined their shape), the Python program that they were typing and the resulting images produced by the computer system when they pressed F5 to test it.

Following Sfard (2008), each array of the algorithm communicated between the facilitator and the students or between students was a "piece" of thinking transmitted between them. As already explained, each array was a little segment of a character (e.g., the asteroid), each character a part of a frame and each frame a part of the whole program developed. Consequently, the iterative communication of arrays that happened was an iterative communication of the thinking involved in the decomposition of the task at hand, i.e., the creation of the video in logical and unambiguous steps that the computer processed.

Thus, the communication of arrays involved the thinking used in the modeling with mathematics that the students carried out on their pencil drawn prototypes, the thinking processes used in modularizing the Python program, and in the testing (evaluation and analysis), debugging and refining practices that the students carried out. The fact that the students tested their program continuously both for effectiveness and satisfaction, as was shown, and that the facilitator was with the students, often validating verbally their communications of arrays as illustrated in the excerpts of Text # 15 shown in the response to RQ2 indicate that the students "knew what they were doing". In other words, the students while adopting a procedure with rules had their reasons (Skemp, 1976). The findings suggest that through the systematic procedure facilitated by Teresa, the students had learned CT-CP, they could communicate it effectively, the computer and Teresa validated it. It was apparent that the students developed CT-CP knowledge as was shown.

Thus, facilitator-student, student-student and student-facilitator or student-teacher *communication of meaningful algorithms may result in CT-CP learning within the ZPD.* I argue that while apparent, the knowledge developed by the students was implicit. "Implicit knowledge refers to knowledge that may not yet be formalized or expressed by the learner but may be evident through actions and behaviours" (Shutte et al., 2018, p. 29). Given that CT-CP learning was mediated without conceptual discussion of CT-CP concepts, but through communication of algorithms and everyday language *I contend that CT-CP can be developed through implicit mediation* (Werstch, 2007)

Conditions, Flexibility and Efficiency

The findings of this dissertation indicate the soundness of pedagogies that, within unavoidable computing conditions, allow for flexibility in the CT-CP practices (Hoppe & Werneburg, 2019) involved in the systematic procedure adopted. As was shown, the facilitator engaged the students in fulfilling an open-ended complex task/solving a problem (Polya, 2004) Indeed, the facilitator gave the students flexibility in deciding the story they wanted to tell, the characters they wanted to include and their shapes, location, sizes, and color they wanted to program with the Python functions and mathematical affordances that they counted on. Consequently, the facilitator provided flexible conditions for the students to create their algorithm, thus their Python program and resulting computer technology. In programming their characters as they wanted by using the mathematical arrays that they used to define the shape, size, and color of their characters as they chose to, the students must have experienced what Sfard (1991) calls intuitive understanding, a kind of understanding where the students had reasons to proceed as they did while still lacking profound conceptual understanding of the rules that regulate CT-CP activity. In doing so the students developed implicit CT-CP knowledge (Shutte et al., 2018) and CT-CP threshold concepts (Meyer & Land, 2003).

Importantly, the findings also suggest the relevance of promoting efficiency in coding, as encouraged by the facilitator in the efficient use of the Python function im_fill (see response to RQ1) and enacted by Herminio in the event where he deleted two pixels that broke the symmetry of the asteroid of frame1 (see response to RQ3). Effective coding and discriminating key pixels are key in image processing and compression, which are fundamental in nowadays image compression and transmission demands (Gangwar et al., 2014, Nosratian et al., 2021).

Summary

This dissertation is important for different reasons that relate to the relevance of its findings, the methodology that was used and the new areas of research that it opens to continue bridging the complexity of CT-CP discourses and making them more accessible, specially to CLD students. The underrepresentation of CLD students in CT-CP/CS gives additional relevance to this dissertation since the CLD participants were able to produce their own computer technology and present it publicly, learning CT-CP in the process. Additionally, the way they were guided by the facilitator to their achievements is significant, and the CT-CP practices that they experienced, especially the communication of algorithm thinking and CT-CP, and the gateways that were opened for them. The facilitator invited the students to a highly complex universe based on simplicity (creating a simple mathematics-based algorithm and program based on size, location, and color) and on the basis of the students' languages, culture and expertise. The findings of this study suggest pedagogies that aim at bridging CT-CP complexity by creating ZPD in a variety of ways, while mediating CT-CP learning implicitly, without deep conceptual discussions on CT-CP concepts. The findings suggest that the essentials of CT-CP can be taught based on the essence of students' ideas and participation, placing them as the main agents of activity. The students' ideas can be transformed by them into algorithms and programs that the computer can process thus creating technology that is truly meaningful to the students. The findings of this study point at pedagogies that maximize students' prior experience in the world, both non academic and academic. In doing so, teachers and facilitators can draw on one essential human characteristic, that is, collaboration (Tomasello, 2010), and the essential resource humans can resort to collaborate, that is, communication (Tomasello, 2014). The facilitator, language arts teacher and the students of this study had a joint enterprise (to create the video) and mutually engaged communicatively to produce it following the shared repertoire of the CT-CP community (Wenger, 1998), which included its ways of knowing, discourse, and practices. The students engaged in the CT-CP community practices and learned to use the tools of the community (e.g., the computer) to produce relevant artifacts (the video) and to adopt its ways of doing things (Wenger 2010), that is they adopted CT-CP practices and experienced multiple CT-CP ways of knowing. Importantly, in doing so the facilitator gave them voice and flexibility of action. The facilitator engaged the students in two-way communicative teaching/learning processes where their voices were heard, their previous knowledge and experience mattered (Usanov, 2020), including their languages and culture. It was discovered, at the middle school level, that human communications can be fundamental in communicating with computers. The facilitator positioned the students as capable computational thinkers and programmers who could communicate effectively both with humans and with computers. Thus, the findings of this study suggest pedagogies that focus on the students' assets and potential to create computer technology communicatively. The CLD students of this study communicated effectively between themselves and with the computer, and also with their language arts teacher and the facilitator to achieve the goal that had identified them with the CT-CP community in the first place, i.e, the video in Python. Communication with computers is unforgiving, and the thinking involved in developing programs is complex. However, the facilitator guided the students into communicating with computers effectively and to bridge CT-CP complexity effectively. Drawing on the students' motivations and ideas for the video, she guided them into developing a logical, and unambiguous mathematics-based algorithm and Python program that the computer could process. In doing so, she guided the students into adopting a pragmatic, mathematics-based, heuristic systematic procedure in which they were the main agents while promoting the exploration of the results of their interactions with the computer and testing the program they were developing as many times as they wanted. The procedure adopted resulted in student CT-CP learning and involved student active engagement in CT-CP practices such as collaboration, creativity, planning and designing, abstracting, modeling, algorithmic thinking, creating computational artifacts, reusing, and testing and debugging (Grover & Pea, 2018; Kong, 2019) and the communication of algorithm thinking and CT. Therefore, the findings suggest that pedagogies include the adoption of pragmatic, mathematics-based, heuristic systematic procedures in their methods. Also, the findings point at the relevance of bridging CT-CP complexity by preparing the students for CT-CP abstractions and LOAS and the creation of programs through CT-CP communicative use-modify-create learning progressions. Finally, the findings suggest that pedagogies include flexibility and efficiency in their methods so that, while adopting a procedure and disciplinary and social conditions such as collaboration, the students can effectively create efficient algorithms, and programs as they want to produce the technology that they want.

Limitations and Future Research

The limitations of this dissertation relate to its design, population and environment and the findings apply to these factors. While acknowledging that the findings apply more specifically to similar populations and environments (after-school practices), I also think that its results can inform CT-CP pedagogies both in and out of school, especially in the current educational landscape where CT-CP definition, practices and curricula are still under development. The SFL-case study methodology adopted in this study allowed scrutiny of the CT-CP universe presented by the facilitator through her discourse and the separation into action and information of what was said by each participant during CT-CP curricular activity. Consideration with the Python program that was being developed and the resulting digital images was instrumental in broadening my view of what was achieved linguistically which resulted in the identifying patterns and eventually in the study findings. The findings were triangulated through varied data sources, theoretical and analytical perspectives, being consistent with the facilitator descriptions of her participation in the CT-CP practices explored. However, spoken discourse only covers part of the communicative resources that the participants used. Exchanges of information depended on language mostly, however, exchanges of action could be realized silently. While associating participant exchanges of information and action to what was being produced strengthened understanding and the explanations provided, what was not said remained out of focus. Thus, the fact that facilitator-student and student-student exchanges of action can be realized without words leaves areas out of focus which would be worth exploring and, similarly, methods that get at their investigation. Regarding student discourse, the methodology used is more applicable to environments that involve novice students at any level who probably comment more on the things that they are doing while they are doing them and to students who are rather communicative. Also, this study was particularly rich in the discourse that concerned both exchanges of information and action because of the collaborative dynamics that flourished which were promoted by the facilitator insistently and in turn triggered abundant spoken communication.

The fact that this dissertation kept a stronger focus on the facilitation of CT-CP practices rather than on whether learning was happening in combination with the fact that I had not anticipated to find indications that student CT-CP learning happened may have informed my interpretations. I view that my positionality as an educator may have informed the study of the discourse that mediated the teaching/learning practices I explored. While my focus, in coherence with the SFL-based methodology adopted, was the discourse that was used to explore the CT-CP universe offered by the facilitator and how she positioned them, I could not isolate my analysis from ideas concerning whether learning was actually happening. Furthermore, I strongly believe that it happened and that the students thought computationally while communicating their algorithm and programming it in Python, especially at the end of curricular activity. However, by no means did I intend in this study to demonstrate that learning happened, my intention was just to provide indications that point at its happening.

Discussing conceptual understanding is relevant since it is a dominant theme in CT-CP/CS educational research (e.g., Grover & Pea, 2018, Kong, 2019, Wing, 2006, 2008). Sfard claims that "understanding" is an ungraspable concept, not operationalizable. I agree. And so is thinking. This research centered on thinking, more specifically on CT, the thinking processes involved in formulating tasks and problems in ways that computers can carry out (Grover & Pea, 2018). In my view, research on CT is complex because of the complexity of getting at human thinking. Sfard's (2008) equivalences between thinking and communication were apparent in this dissertation. Computers can be thought of as sieves that separate unambiguous truth from confusing and misunderstanding-provoking noise. The participants of this dissertation communicated ideas expressed in the kind of truth that mathematics can code without noise (Skemp, 1987) and computers carry out. I believe that the communications discovered in this dissertation where all the participants were aligned among themselves and with the computer as an unforgiving siever of truth illustrate a perfect equation of computational thinking and communication. My view is that this equation between communication and computational thinking can explain CT-CP learning based on Werscht's (2007) ideas on implicit mediation within the ZPD. However, I think that further research is needed to strengthen this perspective. Another interpretation to explain how learning happened (based on the view of the evolution of student discourse in CT-CP practices) can be framed around explicit mediation

(Mahn, 2015; Wertscht, 2007). The facilitator did mention and explain CP concepts such as variables, for loops and conditionals. However, as was shown, she focused on pragmatic explanations and discussion (implicit mediation, thus) of said concepts as opposed to conceptual explanations. If we base explanations of learning on the idea that student conceptual development must be founded on student intentional use of concepts and ability to voluntarily control their use (Mahn, 2015; Vygotsky, 1986). CT-CP learning could not have happened because the students did not get engaged in conceptual conscious discussions. The explanations of CP concepts provided by the facilitator were brief and pragmatic as was explained (Also, as already said, CT concepts were practically not even mentioned, except for testing). Several researchers have argued for promoting students to reflect on and articulate what they are learning in computing (e.g., Lye & Koh, 2014; Sentance & Csizmadia, 2015). The facilitator could have explained in depth, for instance, that each line of the students' program encapsulated in the im_fill functions that they used for loops which took care of digitizing in color the segments of the prototypes of their video that they coded in mathematical arrays. However, she didn't. Instead, she chose to let the students use the im_fill Python function and experience by themselves whether it worked. In other words, she promoted that they experienced by themselves through testing the digital results that the lines of code and function im_field yielded. Instead, she chose to promote that the students communicated the arrays that coded the segments that constituted the video character prototypes with the goal that they collaboratively programmed them in Python. Doing so along with the function im_fill would digitize said segments as long as the students coded them between parenthesis and followed the rest of the Python disciplinary conditions and syntax.

Consequently, based on the discursive indicators (Lavie et al., 2018; Sfard, 2008) illustrated in Chapter 5, my view on this matter is that CLD student CT-CP learning did happen and that it was mediated implicitly, directly and "noise free" through the precise mathematics-based computational thinking which was iteratively communicated between the students and validated as unambiguous by the computer. From my perspective, the students developed CT-CP threshold concepts, "a transformed way of understanding, or interpreting, or viewing something" ..." such a transformed way may represent how people "think" in a particular discipline, or how they perceive, apprehend, or experience particular phenomena within that discipline" (Meyer & Land, 2003, p. 1).

The question is when and how to focus on concepts (Wing, 2008). Grover et al. (2015) argue that discovery approaches with insufficient conceptual guidance favor student agency and engagement but often miss out on helping students develop mental models of concepts. Rather than a strong focus on concepts, I think that student agency in motivational accessible and heuristic systematic procedures should be prioritized; conceptual models can wait. I think that it is important to keep things as simple and meaningful as possible to then build on the implicit knowledge and threshold concepts that were learned as a basis to engage the students in conceptual discussions that help develop increasingly complex mental models of concepts. In any case, further research that focuses both on conceptual development and student agency is needed.

Another hypothesis is that learning may be facilitated by the students' constant testing, analysis and evaluation of the program and images they were developing and the different LOAs that they experienced in doing so. This opens lines of research that focus on the exploration of hypotheses built around the ideas that CT-CP learning can be mediated through computational thinking communication. Also, it would be worth exploring whether facilitator or teacher-student and student-student communication of algorithms can mediate learning.

Finally, I agree with researchers Panoff, Allan, Erickson and Denner who argue for successive applications of the use-modify-create continuum (NRC, 2011) "to develop and examine student learning of CT" (p. 25). I add to their suggestion the inclusion of the communication of CT-CP algorithms and/or CT-CP thinking and promotion of LOAs. Accordingly, I think that research methods that focus on algorithm communication and/or CT-CP communication that include interviews that center on asking students what they were doing in each LOA in relation to other LOA (Cutts et al., 2012) would be especially illuminating. Getting at students' perspectives on what they did in their CT-CP operations while thinking computationally can provide relevant information as to how guide the reconfigurations of meanings that they need to develop conceptual understanding and development (Skemp, 1987). Once the students have developed procedural fluency in CT-CP practices, teachers can guide their understanding and development of CT-CP concepts helping them make connections between what they were thinking and doing and the concepts. In other words, one they have exercised agency and are fluent in meaningful, flexible, heuristic, and systematic CT-CP procedures they will be ready for explicit knowledge transfer discussions where they will be able to make conceptual connections and get involved in metacognition (Mahn, 2015; Vygotsky, 1986; Mills, 2016).

In conclusion, while much is yet to be learned and researched to continue broadening the participation in CT-CP, making it accessible to all and easier to teach and learn, this dissertation has suggested relevant teaching strategies and insights concerning how learning happens and can be mediated.

Conclusion

This dissertation has shown that the promotion of CLD student agency and algorithm communication in bilingual, meaningful, mathematics-based, and heuristic CT-CP procedures can bridge CT-CP complexity and result in CT-CP learning. Rather than CT-CP conceptual discussion, student communication of own-created algorithms obtained through prototype modeling with simple mathematics may help mediate CT-CP learning. It is important to motivate students so that they participate in CT-CP practices actively. In doing so, they will have to think and will do so computationally. In a first stage, teaching should not target conscious conceptual development, but operational fluency.

Positioning CLD students as capable computational thinkers and programmers of their own digital creations whose prior experiences and linguistic identity matter, and as algorithm communicators who can test, evaluate, and refine their own programs can help create the necessary conditions for students to learn. In diSessa's words, "it is important to highlight that abstraction has to connect with their concerns, whether they are menial or whether they are grand. It has to be grounded in people's beliefs and feelings some way or other." (NRC, 2010, p 17). Indeed, the abstraction processes involved in CT-CP processes at different LOAs require that they be grounded in the production of tangible products that originate in the students' prior experiences, interests, and desires. Setting up collaborative and communicative environments that combine CT-CP disciplinary conditions and flexibility for student active participation can help bridge CT-CP complexity while promoting effective synergies between students, facilitators and teachers that help CT-CP educational endeavors move forward. Similarly, the combined affordances of SFL and case study perspectives can be instrumental in defining the nature of CT-CP environments and explore the roles that students, teachers and facilitators can play in the road towards student meaningful participation in the complex CT-CP digital worlds.

Key Terminology

- Abstraction. Abstraction is “ a technique for managing complexity whereby levels of complexity are established so that higher levels do not see or have to worry about details at lower levels” (Bourke, p. 2018, p. 587).
- *Abstraction (Procedural)*. Procedural abstraction is the concept that a procedure or sequence of operations can be encapsulated into one logical unit (function, subroutine, etc.) so that a user need not concern themselves with the low-level details of how it operates. (Bourke, 2018, p. 594)
- *Abstract thinking*. In CT-CP/ CS to think abstractly involves the ability to generalize to identify common properties of instances, getting at their essence by removing unnecessary detail. (Krammer, 2007)
- *Agency*. Agency involves causing a process to unfold which extends beyond the agent to some other entity.
- *Algorithm.* “Algorithm embodies the notion of a precisely formulated unambiguous procedure that is repetitively applied” (NRC, 2010, p. 9).
- *Algorithm thinking.* Algorithm thinking is a CT concept that refers to the cognitive processes involved in developing “precise step-by-step plans or procedures to meet an end goal or to solve a problem.” Grover & Pea, 2018, p. 24)
- *Bilingualism.“Bilingualism* is the use of two languages in everyday life” (Grosjean, 2012, p. 6).
- *Boolean*. Boolean is a data type that represents the truth value of a logical statement. Booleans typically have only two values: true or false (Bourke, 2018).

- *Computational logical artifact.* A computational logical artifact is a creative logical artifact that can be run, tested against the original intentions, and can be refined accordingly (Hoppe & Werneburg, 2019).
- *Computational thinking (CT)*. Describes the mental activity involved in formulating a task in a way that a computer can effectively carry out (Grover & Pea, 2018).
- *Computer program*. "A computer program expresses algorithms and structures information using a programming language. Such languages provide a way to represent an algorithm precisely enough that a "high-level" description (i.e., one that is easily understood by humans) can be mechanically translated ("compiled") into a "low-level" version that the computer can carry out ("execute"); the execution of a program by a computer is what allows the algorithm to come alive, instructing the computer to perform the tasks the person has requested." (NRC, 2010, p. 49)
- *Collaboration*. Collaboration is collective participation in pursuance of a common goal.
- *Computer*. A computer is a device that stores, receives, processes and outputs information (Bourke, 2018).
- *Computer Engineering*. Computer engineering is a discipline integrating electrical engineering and computer science that tends to focus on the development of hardware and its interaction with software (Bourke, 2018).
- *Computer science*. Computer science is the scientific discipline encompassing principles such as algorithms, data structures, programming, sys- tems architecture, design, problem solving, etc. In addition to principles and a stable set of concepts, Computer Science

incorporates rigorous techniques, methods and ways of thinking including "computational thinking (the Royal Society, 2012, p. 5).

- *Computer programming.* At its core, programming is about taking a problem defined in the problem domain, and building a solution using the tools of the solution domain (Alexandron et al., p. 2).
- *Debug.* To debug is the process of analyzing a program to find a fault or error (Bourke, 2018).
- *Facilitator*. I use CT-CP *facilitator* (as opposed to a professional teacher) to refer to a person who, while facilitating CT-CP practices, does not have a degree in the field of education.
- *Flowchart*. A flowchart is a diagram that represents an algorithm or process, showing steps as boxes connected by arrows which establish an order or flow (Bourke, 2018).
- *Function.* A function is a sequence of program instructions that perform a specific task, packaged as a unit, also known as a subroutine (Bourke, 2018).
- *Hexadecimal.* Hexadecimal base-16 number system using the symbols 0, 1, . . . , 9, A, B, C, D, E, F; usually denoted with a prefix 0x such as 0xff1321ab01 (Bourke, 2018).
- *Mathematization*. "Mathematization is a technical term representing relationships in the natural world using mathematics" (NRC, 2012, p. 16).
- *Model.* A CT-CP model is a mathematics-based representation of an object that can be executed by the computer and this way tested to check its evolution (Arastoopour, 2019).
- Output. Output data is information that is produced as the result of the execution of a program (Bourke, 2018).

- *Participation* is the "act or fact of sharing or partaking in common with another or others; act or state of receiving or having a part of something," (Online etymology dictionary)...." literally "to make common," (Online etymology dictionary). Participation, practice and producing computational artifacts are intimately related and constitute " a way of learning of both absorbing and being absorbed in—the culture of practice" (Lave & Wenger 1991, p. 95).
- *Position, positionings* and *roles*. According to Halliday (1984) and (Halliday & Matthiessen 2014), in dialogues interactants continuously self-position themselves and, reflectively, others in each exchange. As explained below, these positions change continuously over the course of activity. For this reason I often use the gerund form, i.e., positioning to confer a meaning of change. There are two variables involved in each exchange: a commodity to be exchanged (either information or goods and services), and two roles that can be taken on (either giving or demanding). Accordingly, there are four basic speech functions or positions which can vary in each exchange: interactants can *provide information* (typically through a statement), *ask for information* (typically through a question) , offer goods and services (typically truth a suggestion) or *demand goods and services* (typically through a command). These four speech functions and roles constitute the interpersonal system of the language which allows people to communicate, enacting in each exchange a particular role by positioning oneself and others as providers or seekers of information. In Halliday's (1984) words: "When the speaker takes on a role of giving or demanding, by the same token he assigns a complementary role to the person he is addressing. If I am giving, you are called on to accept; if I am demanding, you are

called on to give" (Halliday, 1984, p. 12). Eggins and Slade (1997; 2004) added a fifth one: the *check*, typically an interjection such as "mhm," "aha," which is used to clarify or confirm communication.

- *Software.* Software engineering is the study and application of engineering principles to the design, development, and maintenance of complex software systems. (Bourke, 2018)
- *Program.* A program is a sequence of instructions enabling a computer to perform a task; piece of software (Wordreference Dictionary, 2021).
- *Variable*. A variable is a memory location which stores a value that may be set using an assignment operator. Typically a variable is referred to using a name or identifier (Bourke, 2018).
- *The zone of proximal development (ZPD)*. The ZPD is the *link* between everyday and scientific concepts. This zone represents a metaphoric developmental space which is based on what the students' can potentially accomplish with the assistance of a teacher or more capable peer (Vygotsky, 1986; Mahn, 2015).

List of Appendices

Appendix A

AOLME at a Glance

1. Definition STEM interdisciplinary project with research-educational agenda (Comp. Program.-Mathematics)

2. Locations US Southwest University & Middle School (After school)

3. Core Objectives Educational Equity: Provide access to Latino/a to genuine STEM practices and a window to STEM careers.

   Research: Target best teaching & learning practices of interdisciplinary STEM.

4. Rationale -Student implication in a composite genuine STEM practices in an informal environment result in learning outcomes (Cole et al. 2006).

5. Guiding principles -Culturally-linguistically congruent STEM identities

   -Integrated curriculum

   -Collaborative learning

   -College-based and workforce practices

-Formative continuous assessment

---

6. Curricular Key Components -Design: Bilingual-Bicultural undergraduate students of engineering facilitate collaborative practices to small groups (3-5 students). Their envisioned stories grow from pencil & paper to digital color video in Python.

- Model with Mathematics: Two-dimension coordinate systems, decimal, binary and hexadecimal numbers, and algebraic equations to mathematize (define) their drawn models with color and variables, and understand how pixels define video frames and sequential programming.
- Computer Programming Implementation: Student Program their mathematized models in Python using an intermediate code that facilitates the process and display of the video.
- A welcoming, genuine playful engineering environment: An interactive playful collaborative environment based on coding, testing and refining models, and where errors and fixing errors are natural practices.

Appendix B

AOLME Written Curriculum Followed by the Study Participants

Lesson 1. Basics of Raspberry Pi and Linux (MSP)

- Explore and discuss how computers work inside and how information flows in a computer system.
- Utilize the components and functions of a Raspberry Pi.
- Practice assembling components and cables of a computer system.
- Access and navigate the filesystem using Linux.

Lesson 2. Introduction to Python Programming (MSP)

- Apply basics of Python Programming.
- Program basic operations and variables in Python.
- Solve and create own operations, and arithmetic-algebraic expressions.
- Program a number guessing game using Python.

Lesson 3 Algorithms (SCP)

- Introduce students to the notion of algorithm and its relevance.
- Understand the link of algorithms and mathematics.
- Become familiar with pseudocodes, flowcharts, loop control statements, and conditional control statements in Python.

Lesson 4 The Coordinate Plane and Black & White Images in Python (CLL)

- Identify the connections between x-y coordinate plane and use of binary numbers to represent black and white images.

- Design basic black and white images.

Program binary images using Python

Lesson 5 Binary and Hexadecimal number systems (CLL)

- Develop connections and number sense across decimal, binary, and hexadecimal systems.
- Identify real-world applications of binary numbers.
- Convert number values across systems

Lesson 6 Images and Their Components (CLL)

- Represent grayscale and color images using RGB.
- Manipulate real images (taken by a digital camera).
- Open an image file in Python and get familiar with the AOLME Python Library.
- Link the development of images with binary and hexadecimal numbers.

Lesson 7 Creation of Images and Video (SCP)

- Create, code, and display black and white, grayscale, and color images using Python.
- Learn how to create a digital color video using own images.

Appendix C

Significant Texts in Curricular Stages

Task preparation stage

-Text #1 - S. Event #1: *F instructs Students on Variables* (Lines 691-751)

-Text #2 - S. Event #2: *The Student's First Program*: *Coding the Guessing Number.* (Lines 1244-1366)

-Text #3 - S. Event #3: *The Birth of a Facilitator: A student explains* binary number conversions to another (Lines 8222-8280).

Task orientation stage

-Text #4 - S. Event #4. *Brainstorming the Favorites for the Video: Background and Characters* (Lines 10,862-10,930)

-Text #5 - S. Event #5. *Free design with conditions* (Lines 11,988-12,035)

Task specification-realization stage

-Text #6 - S. Event #6. *Negotiation of Fin: The last frame of the Video* (Lines 16,068-16,089)

-Text #7 - S. Event #7. *Facilitator one-on-one guidance of student coding* (Lines 12114-12197)

-Text #8 - S. Event #8. *Some codes are useful and optional* (12, 783-12,882)

-Text #9 - S. Event #9. *Facilitator one on one guidance of student coding* (15,636-15,698).

Task realization stage

Text #10-S. Event # 10. *Community debugging (persistence) with a schoolteacher* (Lines 13109-13,317)

Text #11-S.Event # 11. *Student dictates mathematization codes to School teacher in Spanish* (Lines 14,339-14,444)

Text #12-S. Event # 12. *F explains the coding procedure of how to code a frame from an existing frame* (15,078-15283).

Text #13-S. Event # 13. *Student explains code to the facilitator: "Don't delete it"* (Lines 15,340-15,375)

Text #14-Significant Event # 14. *Student dictates facilitator what to code* (Lines 16551-16596)

Text #15-S. Event 315.*Community coding: Coding frame 4 asteroid* (Lines 16601-16741)

Appendix D

Significant Texts or Events Transcripts

-Text #1 - S. Event #1: *F instructs Students on Variables* (Lines 691-751)

H: Let’s get started then.
F: Yeah.
M: Yeah.
F: Alright.
So, we’ve got 3 variables,
these are the most important variables
'cause you’re gonna be using them.
You got the first one,
which is an integer
and it’s referred to as an int
and it can represent positive or negative values
So, 1, negative 2, negative 5... You know?
And the function is gonna be put as int.
And then you’re gonna have the parenthesis
and that’s gonna convert it to an integer.
Then you have the next one,
which is a float,
and in programming you can refer to it as a float.
M: Wow.
F: What it does is that it shows you real values.
So, when we want decimals to show and fractions
H: Oh, we use the float?
F: We use the float.
M: Wow, this is literally (indecipherable) Float, float.
F: So, you can remember it.
It’s just float.
M: Yeah, Float.
F: Now, the string is represented by quotes.
So, you know how
we did the single quotes
M: Yeah.
F: We formed strings,
meaning is information
that we want to show
So we are always gonna show it in quotes.
H: Okay
F: And it can be

Now the cool thing about the quotes is you can put numbers, symbols, letters,
you can put it all.
So, you know how we had to add earlier,
I'm sure if we had to put like
H: add
F: yeah, add in quotes it would've worked.
So, now that we kind of know a little bit about what they do,
let's go to the beginning.
H: Oh, this one?
F: Yeah,
and we're gonna take turns and
type all those little codes
and then, we're gonna write the answer in that little space, Okay?
M: Oh my god. He's like, ready to type it.
F: Right. You gotta be serious.
Just kidding.
M: What if he messes it up?
H: We'll do it again.
F: Yeah.
Then, you run it
and see what it does.
M: Can't you delete it and put back space?
F: If you already pressed enter,
no.
M: Can you, you can alter it?
F: You can if you're in a file.
M: Oh.
F: But, we're not in a file yet,
we're in a little shell.
M: Why can't we go into a file?
H: mumbles something
F: We will,
that's gonna be our next task.

-Text #2 - S. Event #2: ***The Student's First Program*****:** ***Coding the Guessing Number.*** **(Lines 1244-1366)**
F: We're gonna do L.
H: okay
M: What?
F: Yeah!
H: Where do we put it, Miss?
F: Yes.

So, just go ahead and put an L there because
that's our variable, right?
See?
Because we're gonna do is here on the left
is what we want it to do
and here on the right is our equation.
So, it's what's gonna, it's what's gonna make sure that
we have the number that
our friend wanted.
H: Okay.
F: So, what it is is L is going to be our friend's
number
and so, what we want to do is we want to
make an equation that
allows us to have L as our final answer. Right?
J: Yeah.
F: So here we go.
We have an L
F: So, now the next thing, do we want to multiply or add?
M: Add
H: We could multiply
M: Add by 5?
no, no, no
H: If we multiply by 5
it's gonna be 5L
F: By 5?
M: If you multiply it by 5
it's gonna…
J: Bhy 20
M: Add by 20.
F: By point 8?
M: 20.
J: 20.
H: 20
F: Do you like 20, U?
U: smiles
M: We agree,
everybody agrees by 20.
F: You want 20?
M: 3 of us said 20 so
it's overruled.
U: Yeah, sure.
F: Alright.
So, now, we have L times 20
Let's go ahead and put that on the next one.

M: So, just right
there?
F: Yeah.
So, L times 20. Right?
Okay. So, on the next part do we want to
add?
Do we want to add this time?
J: Yeah.
add by 5
M: a hundred
H: So, we wanna get to L
F: Yeah
M: Are you trying to let them
win?
F: Yeah.
J: By a hundred
F: So, you know how I had told you guys about the inverse, [J and M continue
suggesting math
operations.]
So, what we're gonna do is pretty much
M: How about 200?
F: We're gonna have it this way, right?
We are gonna do multiplication and then
addition.
And then, after our addition,
we're gonna go backwards.
M: How about 200?
H: Oh, okay
F: Does that make sense?
H: Yeah
J: Add by 200
M: 200.
F: Okay. So, L times 20,
plus…
M: 200.
F: 200?
200, okay.
So, now we're at
M: Divide it by
2.
F: L times 20 plus (F writes on the little board)
Plus 200?
H: Yes
F: Alright, 200
So, now,

M: Divide it by 2
F: L times 20 plus 200
Alright, so now, now since we want to go back to L,
we have to work backwards.
M: Divide it by 2
H: So now… it's gonna be 220 so divide it by
M: 2
H: 2.
M: 2 Yeah, that's what I just said.
F: Okay
So, here's the thing
Do we want to divide it
or subtract first?
J: Divide
M: Divide, divide
F: That's gonna be the problem, right?
Because if you think about it,
we can divide by 20.
H: No, subtract
and then divide, so
you can get to…
F: Right.
So, that you can get back.
H: Yeah
F: Because think about it
If we, if we divide now,
we're gonna be off by a certain number.
M: Yeah
J: Okay, let's
M: Oh, subtract.
J: Substract
F: So, now we subtract.
So, L times 20, subtract. (F writes the equation on the little board and the students on their binder)
M: Subtract…
J: By 10.
M: By a 100?
H: So…
M: 50!
H: What happened to the 200?
F: That's right
So, we have a 200,
so we just subtract by 200?
M, H: Yeah.
J: Nods

F: Okay. So…
M: See, I was right.
F: Subtract by 200, like that.
M: Divide by 2
F: Alright.
Now, we do division.
So, now, that we subtracted the 200 hundred,
we're back at L times 20, right?
M: Divide by 2
F: So…
H: No, because you're not gonna get to L
F: So, you're not get
J: What's L?
F: L is your friend's number.
M: We're not gonna be able to get there.
J: Our friend's number
H: We are not gonna be able to get there
J: What's our friend's number?
F: We don't know yet
So, since we multiplied by 20 up here,
we should divide by 20 down here, right?
Do you like 20, U?
U: Yes, sure
H: Yeah, but that's gonna be on 1.
F: So, look.
You divide.
Bam! We get out.
J: oh, okay
M: Divide by 20?
F: Yeah,
and I'm sorry guys
I should probably write it this way so
It's 200 - 200 so they get canceled.
H: Yeah.
F: Yeah.
Alright, so,
this… here it is,
here is our formula,
J: Okay, who wants to go first? [M reaches out for the keyboard.]
[TO M] No, don't touch it [F talks but it is unintelligible.]
M: So, on the final one when we write all the steps.
F: So that's 200
and then minus 200

and then divide it by 20.
H: So, on the final one do we write all the steps.
F: Yeah, So, here on this last part go ahead and write out all your steps.

-Text #3 - S. Event #3: *The Birth of a Facilitator: A student explains* binary number conversions to another (Lines 8222-8280)

F: Now, let's go to the next one
M: 1010011
F: No sir
M: what?!
F: for 253?
M: yeah, 10100011
F: At the end you would only have one 0
So, why don't we work that one together?
Do one of you want to
explain that one to him?
J: H!
H: So, so ah, 253?
So, 111, three ones first
And then ehm, and then another 1
M: Next to the 64?
Oh, wait, wait
H: Well, there's four ones in the beginning
and then two ones, two
So, one, one
I'll go [H stands up, gets his binder with him and goes by M's side to assist him] [J is writing something on his board]
So, this one? [pointing at the exercise on the M's paper] [F remains on her seat watching]
So, 1,1, another 1, and then two ones, two ones, this one and erase this one [pointing at the exercise on the M's paper. M corrects with his pencil.]
So, what would it be?
We are grouping by fours
So, 1111, so what would it be?
Imagine it
So, let's start over here
So, this is one,
and then, you add
This one is turned off,
So, it doesn't count
And then you add 4

What does it equal?
M: 13?
H: Oh, you are doing,
 yeah, 13
 What is 13?
F: 13?
Yeah, you got that
Now what's the other one?
H: Now the other.
M: F?
 I got 15
F: mhm
 Yeah, so, here is the reason for that
 So, you turned the 128 on, right?
 and then you add the 64
 which gets you 192
 and then that's why you add 32
 which gets you to 224
M: mhm
F: and then if you add 16
it gets you to 240, right?
 So, those are your first 4 and they are all on
 And you have 240
 And you have to get to 253 [H returns to his seat]
so, the easy way to do that one
would just be 253-240,
which is 13
so, which ones have to be on to get to 13?
M: okay

-Text #4 - S. Event #4. ***Brainstorming the Favorites for the Video: Background and Characters***

(Lines 10,862-10,930)

F: What do we wanna do?
J: Something like a wither skeleton with an iron armor on it.
M: How about this?
F: Let's go ahead and brainstorm,
M: This should be the background
J: That's the background, really?

That's too complicated
F: What's your favorite thing?
M: No,
because you gotta copy and paste it.
J: Winter.
M: Minecraft
H: Like
J: Ice
M: Snow.
H: Like how?
F: Like do you
okay, it could be, person, place or thing
M: How about this guy? [showing image on his phone to J]
F: Give me your favorites.
J: That’s cool, but then like
I wanted a skeleton like regular, though
H: Do something of outer space, like a planet or something
That would be nice
M: How about this guy? [Shows phone to J]
H: We could make it small and then like white dots
F: Like the planets
H: Yeah
F: Alright,
so, here is the thing
J: The thing?
F: Yeah,
What's your favorite thing or things?
J: football.
F: football, Alright
M: We should do this one (F shows the phone to J)
J: No
F: Alright, M, what's your favorite thing?
M: like what?
J: what's your favorite thing?
M: Like what!?
J: Like sports, gaming,
F: What's your favorite sport?
J: Minecraft
M: Playing games and basketball
F: Basketball
J: Wither skeleton
F: Okay, I am seeing you guys a list of all the sports, right?
M: I like games too
F: so, here we got these three.
M: I like games too.

F: in the sports section, right?
M: Oh, yeah, we should do this (shows phone to J)
F: so, what we could is we can mhm
J: That's too complicated.
M: How about this then?
F: okay,
So, we like sports,
What about places?
J: winter stuff, like
F: You like winter?,
Okay
M: You are gonna hate me
if I show you this one
F: What do you like, M?
M: What?
F: A place
M: a place,
a place, a place, a place,
somewhere where it's warm.
Tropical island
F: A tropical island for M.
H: Yeah, warm.
F: warm?
    Alright, so
    you want a tropical island too?
H: Nods.
F: Alright,
So look,
we could do,
with all the ideas, here is my idea,
M: How about this guy?
F: and you could tell me more.

-Text #5 - S. Event #5. ***Free design with conditions*** **(Lines 11,988-12,035)**

(PIC: Si quieren que ayude en algo, a pintar me dicen,)
F: Okay,
Gracias)

F: Yeah, so,
        you can use sticky notes to
        kinda' figure out the order that

you want it to
M: I would've done it but as I said, my entire thing was,
F: That way what it is, that way what it is, those are going to be like your frames, right?
yeah,
And we gotta figure out the background.
Maybe we'll do the background
and program it today [as she grabs the mouse]
M: Are we gonna have to do a background?
F: We are gonna have to do a background.
M: Why!?
J: Because we have to do the background.
M: But I don't want to, I want to [indescipherable word]
J: You'll have to.
M: but I don't want to!
H: Miss, it's gonna probably be like this [pointing at his work on the sticky notes]
and then when it falls
it's gonna get bigger, so.
F: Yeah.
H: Here is smaller
and then here a little bit bigger,
and then here bigger.
F: Yeah.
yeah,
so, when it hits J's character
it should already be pretty big.
H: Mhm
F: Should be big.
H: [Draws the asteroid on the stickynote] That's how big it's gonna end,
J: Really?
It's just gonna kill all of them,
It's gonna be like, oh my God
H: It doesn't explode in them so,
J: Just squishes them?
H: Yeah,
F: Squishes them?
H: Yeah,
And then, nothing.
J: then it should be like dust
H: Yeah
F: So, the last one is gonna be just black.
it's gonna be pitch black
J: And it's gonna say
H: The end.
M: The end
J: Fin.

F: Fin
It’s gonna say fin,
okay
M: We should take fin into the show.

-Text #6 - S. Event #6. ***Negotiation of Fin: The last frame of the Video*** **(Lines 16,068-16,089)**

J: Okay
I will make the fin
F: I think he is making the fin [Looking at M]
J: No, you are making the asteroid.
M: No, I'm not doing nothing.
J: Yes you are.
F: What do you mean
you are not doing nothing?
M: I'm doing the end.
F: He is doing the fin and
you edit it [To J]
H: He is picking the easiest thing.
M: I don’t wanna.
J: The easiest thing is fin.
H: I know.
F: You have the easiest thing [to J]
all you have to do is
change the coordinates
M: I have to get fin,
it took me forever to make the
F: Everything is already programmed.
J: Oh, find out where the coordinates are?
F: Yeah,
J: So how was, where is H's at?
M: I did like my artist thing and you deleted it so,
F: Give me a second and I will [as she focuses on the laptop]
H: fin is also easy, you just put 0 to 19 to 0, 19 and you just put where the little fin is.
F: Sorry, buddy, I’m gonna save this real quickly.

-Text #7 - S. Event #7. *Facilitator one-on-one guidance of student coding* (Lines 12114-12197)

F: Yeah, so, [H gets hold of the keyboard]
Let me kind of explain.
So, you are gonna make a frame, right?
H: Mhm
F: And you are gonna label it.
T: ¿Qué es lo que van a hacer, M, éste? [points at a design on the table]
¿Todos?
¿Son diferentes todos?
M: Mhm
It's gonna be a real challenge, though.
T: ¿qué tal?, [to H]
¿todo bien?
H: Mhm
F: So, like when you make a frame, right?
Like in this case, checkerboard,
H: So?
F: It's your variable.
Name it, right? [F shows the code written on paper to H]
So, I don't know, if you guys just wanna do frame one, frame two, frame three, and frame four?
H: Frame one, frame 2,
(M and J discuss on location of characters and storyline while they draw)
F: See, you are gonna actually have to have [indescipherable few words]
H: Capitalized or lower?
F: No,
it actually can be lower case.
M: How many frames are we gonna have, too?
J: How many frames can we have?
F: Oh, we can have
M: five?
F: We can have five.
J: we are gonna need five
H: But why five?
J: 'Cause he wants to push me, my character back,
and then, that's when H's thing strikes.
M: Yeah, so,
like first he is gonna slam me and then
I'm gonna like push him back.
F: Okay, so let me Let me see if I, one, two, three, four, five [counts the sticky notes they will need]

Exact moment when F counts the sticky notes and J is “pushing” with his hand, imitating the character move.
[as she counts sticky notes to deliver to students so that they draw the frames on them]
M: Cause in the middle he is gonna get extremely huge
and then is when he smashes
H: Both, so one thing is, he is gonna be saved really ‘cause
he doesn't have time to save himself.
J: Alright, I get it H
F: Alright
H: So, frame 1? (meaning should we start?)
M: The meteorite dies anyway, too.
F: Yeah
H: Copy all this? [looking at his binder]
And put just put frame one, frame two, frame three frame four.
F: So, here is the thing,
You’re gonna have, you have to put your rows and columns in
H: So it’s like 20 by 20?
20 by 20?
F: Yes
So, you have to put rows
is equal to 20,
and columns is equal to 20.
H: Do I have to put?
Oh, no.
F: Put, you should put though, in a hashtag, put frame one.
that way you can,
(M and J continue talking about the storyline)
It’s the comments of your work.
H: So?
F: So, like here, you are gonna put hashtag, frame one, right?
H: and then just put the rows and columns? [M & J keep on discussing the story line. F let's them be]
F: Yeah,
and then put the
rows is equal to 20 [F writes all this on paper],
the columns is equal to 20,
and then your name begins, just frame one, right?
and then frame is equal to and
here you are gonna put
What are we gonna have?
This is probably gonna be our background, right?
H: Yeah,
F: And this, this is just gonna be
until we find the color for it,
we'll just do 000000

So, we'll look for a color here in a bit.
Star, columns for row in range, rows,
So, that's like the first line of code
and then after that, this is when you do another hashtag,
so, what this does is just commenting your work,
That way if you have to go back,
you know where to
go to fix what
you need to fix,
H: Mhm,
    mhm.
F: And then here it's where you actually start
your coding for, for our thing.
H: Okay.
F: This, all this does is it just opens a 20 by 20
    and it's gonna make it black for now
till we figure out the colors.
H: Okay, so frame one and then [H starts coding here]
F: Mhm

-Text #8 - S. Event #8. *Some codes are useful and optional* (12, 783-12,882)

F: Go to page 6 on session 7
So, I was showing M this code right here [to H]
[M begins a conversation with J about color]
Because using this code
is that you can, you can do like long distances.
So like if you wanted to you can do
H: So like for this one it's gonna be a longer distance.
F: Yeah,
So like for this one you can do in_fill for every single segment like this
That way you get more out of it instead of
doing each one individually.
H: Mhm
F: 'Cause that was the problem with,
you know when we looked through session 7
and we saw the checkerboard
you know how they did each, each number.
H: Oh, yeah
F: And it just took way too long.
    Well this, the in–fill allows us to pick where we want to fill in.

-Text #9 - S. Event #9. *Facilitator one on one guidance of student coding* (15,636-15,698)

F: Okay,
so here, in order to play the video,
you have to add this line of code [passes him a piece of paper with the code]
M: Where?
F: At the bottom.
J: the bottom
we put the (indescipherable) thing first
F: Oh, put the imshow
hold on,
hold on.
M: Oh, imshow 2, 2, right?
F: Yeah
do we have the im_show for the first frame?
M: I'm pretty sure, yeah
F: We don't
after this let's put im
M: im_show
F: show, frame 1.
and then im_show this one
'cause you have to do the im_show for all of them.
(At the same time there is an interaction between H and J on the mathematization of a character for the video).
Okay
So here, M
this is where you are gonna add the framelist
here, buddy [passes the keyboard to him]
so, you are gonna add the framelist
what this does is it makes it [indescipherable word]
so framelist is equal to.. and then brackets [F dictating to M for him to code]
M: like this?
this one?
F: No
M: I mean, this one?
F: That one
and then you are gonna put frame 1 and frame 2
cause those are the names of the arrows, right?
frame one
M: and now underscore?
J: 1
F: just 1

There you go
and then frame
coma
2
and then end it
alright
and
There you go
now enter
and you are gonna put fps
what that means is frames per second
J: mhm
F: Just put a 1 for now,
that way we see it run.

Text #10-S. Event # 10. ***Community debugging (persistence) with a schoolteacher*** **(Lines 13109-13,317)**

F: Did you write the codes?
T: Ya tu acabaste de escribir todos tus códigos?
M: For the tree?
¿ya?
I'm doing it
T: okay,
¡pues vamos!
ayy
M: Ven a hacerlo.
T: ¡Vamos!
M: ¿Por qué no eh?
T: No, si eres tú.
M: ¿Eh?
T: ¿Quieres que vaya?
M: A ver.
Y los voy a hacer más rápido.
T: ¿si?
M: Vamos.
T: Bien, pero me tienes que ir diciendo tú
M: Muévete acá
T: Okay.
Entonces yo voy escribiendo y tú me vas diciendo, ¿sí?
Tú vas mirando y me vas diciendo.

H: Where is the?
J, can I borrow it?
M: 12 x 12
T: ¿Aquí? [F starts writing on a sticky note the coordinates of the tree as M dictates them to her]
M: 7 x 7
T: ¿aquí al lado?
¿son dos códigos o uno solo?
M: Solo son dos,
8,8
T: ¿Aquí?
M: Son dos aquí.
T: Son 2
T: ¿8?
M: 8
T: Okay
M: 13,13
T: 13,13,
M: no, no, no, digo 12,12
T: ¿12, 12?
M: aha
es que se me olvidó que no puedo brincar asi de repente.
So, es que 12,12.
T: Okay, entonces, 12, 12
M: 9, 9
T: 9, 9
Okay
M: 12,12
10, 10,
T: 10, 10,
M: 13, 13,
T: Ehm
M: 7,7
T: Okay
M: 13,13,
T: 9, 9, okay.
M: 12,12
T: 12, 12,
M: 10, 10
T: 10, 10,
M: 13, 13
T: Ehm
M: 7, 7
T: Okay
M: 13,13.

T: Ehm.
M: 9, 9.
13, 13.
T: Ehm.
M: 11,11
T: Ehm.
M: 14, 14
T: Ehm.
M: 9,9
T: Ehm.
M: 15, 15
T: Ehm.
M: 9, 9
T: Ehm.
M: 16, 16.
T: Ehm.
M: 8 a 10
8 y 10.
T: ¿Entonces pongo 8, 10?
M: Ehm.
T: Okay.
M: 17, 17
T: ¿17, 17?
M: Ehm, 17, 17
7 a 11.
T: Okay.
Okay, viste qué rápido entre los dos?
Muy bien, ya terminaste, ahora qué esperamos por Juan y después tienen que meter ustedes dos no?
M: ehm
ahora voy a ahora ya solo entonces de una vez ya voy a meter los colores (Now, I am going to just once for all then put the colors in)
T: Los colores, okay.
¿Me los vas diciendo tú o los quieres hacer tú solo ya?
¿Sí? [M starts using his pencil]
¿Seguro?
M: Sí

Text #11-S. Event # 11. ***Student dictates mathematization codes to schoolteacher in Spanish***

(Lines 14,339-14,444)

T: M, explícame qué pasa, por qué no va?
No sabes?
M: No
Y tiene asistencia, pero tampoco.
T: Ya
PIS: So, what didn't?
Did it run?
M: nop
I tried to run it and then
it said an error
Oh, now she is back
T: Hay algo mal.
PIS: laughs.
J: It didn't work
F: It didn't work, it's okay
M: This is mean, now she is back
F: We'll figure it out.
T: Hay que descubrir qué es lo que está mal, ¿no?
F: Sí. (Yes, we have to find out what's wrong, right?)
M: I am telling you, now she is back,
she left us and now she returns.
F: Hey you should be able to run it
try it again
M: F5, go click,
and There you go [the system returns an error]
F: Oh, line 3, okay,
that's okay
Now, click on the code.
Okay, line three
One
M: two, three
F: two, three.
No, buddy, I think it's
M: I think I added backwards, 13 by 16
J: No, it would show, but then it would
F: Take this off, the coma and
put an enter[pointing to the screen]
M: This coma just take it off?
F: Yeah, take it off and then enter
Alright
Try it, try like that
A lo mejor no más es no más eso [M operates keyboard]
Okay, a ver
wrong way oh, okay [reading the screen]

ahora no más no está esto.
Open this, right next to it
and then like move it over
like where we have the errors right next to it
Yeah, so move it that way
M: Oh, I get it,
F: There you go, okay
M: Right there
I am following the instructions like that and then
there's a color that is wrong.
because, look,
this color is 3, 3 [pointing to the screen],
this one color is named 3,3,
which is grey
F: So,
M: [To J]are you sure it was 6 threes?
J: Nods [F looking at the screen silently]
M: It's like literally down here specifically,
telling 3 numbers and there the names
T: Los números esos son los colores, ¿verdad?
F: Sí.
A lo mejor,
T: ¿y qué está mal, el 3,3,3?
F: está mal el 3,3 (not a statement, not a question, thinking to herself, limitations of SFL)
M: Lo estoy diciendo, que el 3,3,3, 3,3,3
F: Oh, sí, ¿hay seis?,
¿hay 6 ó 7?
M: Entonces digo que sí hay 6
una, dos, tres
T: Siete
M: Cuatro, cinco, seis
Oh
Chuckles [M corrects error with keyboard]
F: Siete
T: Chuckles
F: Tenemos uno extra.
M: Yeah we have
Ya yo puedo decir que va a decir otro error
T: No importa
F: Eso no sabes,
T: Tú tienes que probar
No! ningún error,
Oh, oh
F: Ah, no, es porque esto,

Es porque está en negro,
Cambia el color de la que
es como café que no?

Text #12-S. Event # 12. ***F explains the coding procedure of how to code a frame from an existing frame*** **(15,078-15283)**

F: There you go, buddy.
H: look [taps j on his arm],
watch it.
J: That's our biggest, dun
M: Holy!
F: And that's only the first frame.
H: We could have colored,
take those 2 off.
J: No, it's okay, H, it's fine.
It hits like a rocket
H: It looks nice
Too much work for that little one.
J: Okay, How am I gonna put his?
M: i just wanna watch,
i just wanna watch it when
you gotta make it bigger,
imagine that
F: Well, guess what though,
this was the hardest part,
you wanna know why?,
cause now, all you have to do is
copy and paste for the next frame
and then change a few coordinates because
your asteroid gets bigger.
J: Add a couple, add a couple
to make it bigger
M: does this look good for sand? [shows celular phone screen to j]
J: Nop
Just kidding, yes it does,
H: i wanna take those two off.
F: Those two, okay,
0,0,
H :19 TO 4, 19 TO 5
J: That's 19

F: Okay,
    So, let's go look at the code,
J: I did his, I did the characters,
    but I need to put H's [grabbing the paper
    grid with character and shovel and pointing
    where it would fit]
F: Asteroid right there in the middle?
    YOU WANNA FIGURE IT OUT? [as she passes J
    her percil]
J: Is there fire on it?
    Yes, there has to be fire on it,
    so like [as J begins drawing with the pencil F has just given
    him]
H: Was it the dark one or?
F: It was the dark one, so
H: The dark one starts right here
H: Can i put it slash dust?
F: Oh, that's the
Yeah, dust
H: So, I don't forget
which one is
F: Right
H: was it 2,2?
F: Yeah, okay, so, it was 00
J: so, is it gonna be that h,
or bigger?,
F: 15,15 [pointing at screen]
So, that first one,
take that one off
J: H, H, you want it bigger?
H: mhm, yeah
J: Okay,
Like from right here
H: Or, maybe that’s okay
J: Let me see
H: What was the other one, Miss [to F]
J: How about that big?
    wait
F: What was the other one?
F: And then we had, 4, 4; 19, 19
  There you go.
J: That big then? [to H as he shows what he just drew on the grid]
H: yeah
J: Okay
  or bigger

I can make it bigger
Just put xs so that I know where it's at [H grabs the pencil and starts drawing on J's grid]
H: No, it's , watch
so, what is it?
J: The asteroid
H: 1,2,3,,4,5,,6,7,8
J: 9?
H: 9
mumbles something
that's correct
the bottom is correct
this one is wrong
J: add one, add one, add one [as H adds xs to J’s drawing of the asteroid for frame 2]
T: Oh my God! [as she takes a picture of finished frame 1]
Ya solo falta tu árbol M
J: got it
F: So, That's all with grey, right?
Is that all the grey?
H: yeah.
F: okay
and then
J: I'm gonna put the fire
F: Alrighty, we got it on this one too!
woo!
H: So, that's frame one
F: So that's frame one
H: So frame 2 now?
F: Yes

Text #13-S. Event # 13. *Student explains code to the facilitator: “Don’t delete it”* (Lines 15,340-15,375)

F: So, what are you trying to do, buddy? [M working on the background on the red laptop. F looking at the screen, T too]
M: I'm adding the land
F: What color is the, the, the sky?
which one is that one, here?[pointing to the laptop screen]
M: Well, that's a major difficulty putting in the color here so I'll just put it into this so
F: But which one is the sky?
like[begins typing] here, is this, is this the sky?
M: that's the dirt

F: That's the dirt
F: What's the?
M: Those are all water
F: So, is this the sky?
M: No,
That's shov, that's tree.
That's tree.
F: So, where is your sky?
M: The sky needs a separate?
F: No,
but which number is the sky?
isn't that this one?
M: The sky is down here.
F: It's this one right?
M: No,
That's dirt
The sky is down here to here
F: Alright
M: Don't delete it
F: I ain't gonna delete it
M: Don't delete it
F: I just wanna check something
M: Oh no, gonna kill it.
F: I'm not gonna kill it.
M: She is gonna kill it.
She is gonna kill it.
2,4,6
F: That's how blue, that blue is
[smiles]

Text #14-Significant Event # 14. *Student dictates facilitator what to code* (Lines 16551-16596)

F: You want it bigger, right?
J: Right,
Cause I made, I just made
aghh [it seems J notices a mistake and begins erasing design on
paper grid]
H: Haya los x (/ex/) [To M]
J: That one wasn't like grading?
cause I added 2 to this, but I want to make it [as he completes his drawing]
H: Miss,
eh, do you remember, which ones is for the dark reds, these ones or these ones? [pointing

to the screen]
H: I think these ones.
F: I wanna say those ones
give me my paper [as she finds a sheet of paper]
H: yeah, it’s those
this one has more than the top one
F: Okay.
yes,
Copy and paste it [To H]
J: That's bigger now [as he ends fixing his drawing of his asteroid design [frame 3]
F: You think that's bigger?
J: Nods
F: You can also do it like to where like it goes extra, here [points at where on the paper grid]
like an extra you know what I mean?, line
That way it just looks bigger in general.
J: ‘Cause this is big enough.
F: Yeah
J: What code, cause 0 to 3 [as he points to the grid]
F: Here it is
J: Oh, I got it
F: Let's go ahead and [as she gets hold of laptop keyboard]
J: 0 to, that side [Points to the screen] 0 to 3
F: 0 to 3?
J: Yeah,
F: and then 8 to 19?
J: Mhm
No, it's 0 to 2
F: Oh,
0 to 2.
J: 0 to 2 and then 3 to 6
no, 3 to 4.
wait [as he draws again on the asteroid grid]
3 to 5
F: 3 to 5?
J: Yeah.
F: 3 to 5
J: 6 to 8
F: 12 to 19?
J: Yeah
and then it’s gonna be 12 [J is drawing]
F: Okay, so, 12 to 19, this one [checks by pointing with her finger to the background
J is drawing on on the paper grid. While doing so, F codes the coordinates J is telling her]
6 to 8

GS: [to H] ¿Cómo va eso amigo?
H: Bien
J: It'll be 19 to 15
And 19 to 14

Text #15-S. Event 315.*Community coding: Coding frame 4 asteroid* (Lines 16601-16741)

M: 5, 5
J:10, 11 [H operating laptop]
T: 5,5
H: 10,11
M : Thank you, a breath break.
T: next one, J
J: 5,5 12, 17
T: 5,5,
5,5,12,17
J: Okay
5,5,19,19
T: 5,5
5,5 what?
H: 19
J: 19, 19
T: 19, 19
H: 19, 18 or?
J: 19,19
H: what else?
J: 6,6: 11,14
H: 11
M: What the? [as he erases something seemingly a wrong coordinate on the grid]
H: What else?
J: 6, 6; 16, 19
M: This is like so advanced [as he keeps on writing coordinates on the grid]
T: Next one
H: Next
J: Hold up
he still writing it
T: up
M: Yeah
7,7
GS: [cmes to table to briefly comment with H something about the presentation]
H: PIC
todos nosotros

GS: Okay
T: 7,7 (In English)
H: oh,
7, 7?
T: J, 7,7?
J: 10
F: 10
M: Yeah,
10
H: 10,10?
J: Yeah [as F gently takes his pencil] from M's hand]
M: What?
F: I'll help you out from here on, okay?
Okay, ready?
M: I wanted to make sure they are exactly there because like if I get it wrong we all
F: 4, 14
H: 4,14?
F: Yeah
M: I am trying my best not to exactly what I was doing but like I was making sure
H: Plus
F: Kay
7,7
15
H: 15
F: 18
M: I like computers like this
F: 8, 8
Hey, M
we don't have time
M: Where are they at?
F: 8, 8 [grabs pencil again]
M: 8, 8
11,11
F: ‘Cause this one is 8, 8
T: Falta más?
H: si
T: Oh, My God,
M: What 9, 9? (M gets pencil again)
F: Yeah
9,9
Yeah, you finish copying
M: 11, 11?
F: Mhm
Hold on, buddy, I think that's the 8 [grabs M's pencil again]
T: [To H] Mejor que sobre ¿no?

H: Mhm
F: 9, 9 is this one [drawing a horizontal line on the grid]
M: okay
F: Okay
You are ready?
T: aha
F: 8, 8
13 to 17
T: 8,8
H: 13 to 17?
F: Yeah
9, 9
12 to 14
9, 9
M: So much work for one meteorite trail
F: 15, 16
M: oh, God
F: 13 to 15 [M erases what probably was a mistake]
15
13 to 15
and then 11, 11
M: And 14, 14
H: 11, 11, 14,14
M: I think we should take this just to show how much work it takes [As he gets up and leaves the table]
H: We've probably done the most
Do we put imshow or?
F: Yeah,
put imshow
or do you have it already?
do you have the the imshow? [as she operates a couple of keys while H is still sitting in front of laptop]
H: no
F: Then, now
put imshow
H: like what is it now ?, parenthesis?
F: yeah,
Parenthesis, frame 4.
H: show is gonna be?
F: so, it's telling it to show frame, right?
Yeah, you had it right
show frame 4 [as she types]
alright
Let's run it.
H: Oh you,

H: Who missed that one? [J points at M. everybody chuckles. A student from another group shows up to watch screen]
H: somebody missed one
F: Okay, sir, this is the dark one, right? [F gets control of laptop]
M: 7, 7
F: 7, 7 where is it?
M: I didn't see it [F is operating H's laptop]
F: I'm sorry,
let me
We'll just do it like this [points to the screen]
yeah
7,7
M: 7,7
H: 15,15
M: yeah,
F: 15, 15?
M: 7,7, 15, 15
7,7, 15, 15
7,7, 15, 15 [J standing by M's side all the time]
F: Okay
this is the dark, I mean, the one that's missing right?
H: 7,7
F: 15, 15
[F continues operating laptop; students talk about graduation party which would be happening in only a few minutes. The student that joined the table is told that the video is finished. M tells him that she is just adding the dust and fire]

Appendix E

The Program in Python Developed by the CLD Students

```
rows=20
col=20
frame1 = [["f4c169"]*col for j in range(rows)]
frame2 = [["f4c169"]*col for j in range(rows)]
frame3 = [["f4c169"]*col for j in range(rows)]
frame4 = [["f4c169"]*col for j in range(rows)]
frame5 = [["000000"]*col for j in range(rows)]

#FRAME1
#tree
im_fill(frame1,[9,9],[7,7],"008000")
im_fill(frame1,[9,9],[9,9],"008000")
im_fill(frame1,[9,9],[11,11],"008000")
im_fill(frame1,[10,10],[8,10],"008000")
im_fill(frame1,[11,11],[7,11],"008000")
im_fill(frame1,[12,12],[8,8],"008000")
im_fill(frame1,[12,12],[10,10],"008000")
im_fill(frame1,[13,13],[7,7],"008000")
im_fill(frame1,[13,13],[11,11],"008000")
im_fill(frame1,[12,15],[9,9],"800000")
im_fill(frame1,[16,16],[8,10],"800000")
im_fill(frame1,[17,17],[7,11],"800000")

#land
#im_fill(frame1,[7,8],[0,19],"f4c169")
im_fill(frame1,[18,19],[0,19],"006400")

#sun
im_fill(frame1,[0,2],[0,2],"ffdf11")
 #Michael character
im_fill(frame1,[9,9],[3,6],"000000")
im_fill(frame1,[10,10],[3,6],"0929FF")
im_fill(frame1,[10,10],[6,6],"0929FF")
im_fill(frame1,[10,10],[4,5],"333333")
im_fill(frame1,[11,11],[3,6],"000000")
im_fill(frame1,[12,12],[1,1],"000000")
im_fill(frame1,[12,12],[4,5],"000000")
im_fill(frame1,[12,12],[8,8],"000000")
im_fill(frame1,[13,13],[1,8],"000000")
im_fill(frame1,[14,14],[4,5],"000000")
```

```
im_fill(frame1,[15,15],[1,8],"000000")
im_fill(frame1,[16,16],[1,1],"000000")
im_fill(frame1,[16,16],[4,5],"000000")
im_fill(frame1,[16, 16],[8,8],"000000")
im_fill(frame1,[17,17],[3,3],"000000")
im_fill(frame1,[17,17],[6,6],"000000")
im_fill(frame1,[18,18],[3,3],"006400")
im_fill(frame1,[18,18],[6,6],"006400")

 #JUAN character
im_fill(frame1,[11,11],[13,13],"333333")
im_fill(frame1,[12,12],[12,12],"000000")
im_fill(frame1,[12,12],[13,13],"333333")
im_fill(frame1,[13,13],[12,12],"333333")
im_fill(frame1,[13,13],[14,14],"333333")
im_fill(frame1,[12,12],[14,14],"000000")
im_fill(frame1,[13,13],[13,13],"000000")
im_fill(frame1,[15,17],[13,13],"92370F")
im_fill(frame1,[14,14],[13,13],"92370F")

 #HERMINIO character
im_fill(frame1,[0,0],[16,16],"F90707")
im_fill(frame1,[0,0],[18,18],"F90707")
im_fill(frame1,[1,1],[17,17],"F90707")
im_fill(frame1,[1,1],[19,19],"F90707")
im_fill(frame1,[2,2],[15,15],"F90707")
im_fill(frame1,[2,2],[16,16],"F90707")
im_fill(frame1,[2,2],[18,18],"F90707")
im_fill(frame1,[3,3],[17,17],"F90707")
im_fill(frame1,[3,3],[19,19],"F90707")
im_fill(frame1,[4,4],[17,17],"F90707")
#dust
im_fill(frame1,[0,0],[17,17],"AF0000")
im_fill(frame1,[1,1],[15,15],"AF0000")
im_fill(frame1,[1,1],[16,16],"AF0000")
im_fill(frame1,[1,1],[18,18],"AF0000")
im_fill(frame1,[2,2],[17,17],"AF0000")
im_fill(frame1,[2,2],[17,17],"AF0000")
im_fill(frame1,[3,3],[16,16],"AF0000")
im_fill(frame1,[3,3],[18,18],"AF0000")
im_fill(frame1,[4,4],[18,18],"AF0000")
im_fill(frame1,[2,2],[19,19],"AF0000")
#dust
im_fill(frame1,[2,6],[14,14],"333333")
im_fill(frame1,[3,5],[13,15],"333333")
```

```
im_fill(frame1,[4,4],[12,16],"333333")

im_show(frame1)

#FRAME2


#tree
im_fill(frame2,[9,9],[7,7],"008000")
im_fill(frame2,[9,9],[9,9],"008000")
im_fill(frame2,[9,9],[11,11],"008000")
im_fill(frame2,[10,10],[8,10],"008000")
im_fill(frame2,[11,11],[7,11],"008000")
im_fill(frame2,[12,12],[8,8],"008000")
im_fill(frame2,[12,12],[10,10],"008000")
im_fill(frame2,[13,13],[7,7],"008000")
im_fill(frame2,[13,13],[11,11],"008000")
im_fill(frame2,[12,15],[9,9],"800000")
im_fill(frame2,[16,16],[8,10],"800000")
im_fill(frame2,[17,17],[7,11],"800000")

#land
#im_fill(frame1,[7,8],[0,19],"f4c169")
im_fill(frame2,[18,19],[0,19],"006400")

#sun
im_fill(frame2,[0,2],[0,2],"ffdf11")
 #michaels character
im_fill(frame2,[9,9],[3,6],"000000")
im_fill(frame2,[10,10],[3,6],"0929FF")
im_fill(frame2,[10,10],[6,6],"0929FF")
im_fill(frame2,[10,10],[4,5],"333333")
im_fill(frame2,[11,11],[3,6],"000000")
im_fill(frame2,[12,12],[1,1],"000000")
im_fill(frame2,[12,12],[4,5],"000000")
im_fill(frame2,[12,12],[8,8],"000000")
im_fill(frame2,[13,13],[1,8],"000000")
im_fill(frame2,[14,14],[4,5],"000000")
im_fill(frame2,[15,15],[1,8],"000000")
im_fill(frame2,[16,16],[1,1],"000000")
im_fill(frame2,[16,16],[4,5],"000000")
im_fill(frame2,[16, 16],[8,8],"000000")
im_fill(frame2,[17,17],[3,3],"000000")
im_fill(frame2,[17,17],[6,6],"000000")
im_fill(frame2,[18,18],[3,3],"000000")
```

```
im_fill(frame2,[18,18],[6,6],"000000")

 #JUAN character
im_fill(frame2,[13,13],[13,13],"333333")
im_fill(frame2,[14,14],[13,13],"333333")
im_fill(frame2,[15,15],[13,13],"000000")
im_fill(frame2,[15,15],[12,12],"333333")
im_fill(frame2,[15,15],[14,14],"333333")
im_fill(frame2,[14,14],[12,12],"000000")
im_fill(frame2,[14,14],[14,14],"000000")
im_fill(frame2,[16,19],[13,13],"92370F")

 #HERMINIO character
im_fill(frame2,[0,0],[16,16],"F90707")
im_fill(frame2,[0,0],[17,17],"F90707")
im_fill(frame2,[1,1],[14,14],"F90707")
im_fill(frame2,[1,1],[15,15],"F90707")
im_fill(frame2,[2,2],[14,14],"F90707")
im_fill(frame2,[2,2],[16,16],"F90707")
im_fill(frame2,[2,2],[19,19],"F90707")
im_fill(frame2,[3,3],[15,15],"F90707")
im_fill(frame2,[3,3],[17,17],"F90707")
im_fill(frame2,[3,3],[19,19],"F90707")
im_fill(frame2,[4,4],[16,16],"F90707")
im_fill(frame2,[4,4],[18,18],"F90707")
im_fill(frame2,[5,5],[17,17],"F90707")
im_fill(frame2,[5,5],[18,18],"F90707")

#dust
im_fill(frame2,[0,0],[13,13],"AF0000")
im_fill(frame2,[0,0],[14,14],"AF0000")
im_fill(frame2,[0,0],[15,15],"AF0000")
im_fill(frame2,[0,0],[18,18],"AF0000")
im_fill(frame2,[1,1],[13,13],"AF0000")
im_fill(frame2,[1,1],[16,16],"AF0000")
im_fill(frame2,[1,1],[17,17],"AF0000")
im_fill(frame2,[1,1],[18,18],"AF0000")
im_fill(frame2,[1,1],[19,19],"AF0000")
im_fill(frame2,[2,2],[15,15],"AF0000")
im_fill(frame2,[2,2],[17,17],"AF0000")
im_fill(frame2,[2,2],[18,18],"AF0000")
im_fill(frame2,[3,3],[16,16],"AF0000")
im_fill(frame2,[3,3],[18,18],"AF0000")
im_fill(frame2,[4,4],[17,17],"AF0000")
im_fill(frame2,[4,4],[19,19],"AF0000")
```

```
im_fill(frame2,[5,5],[19,19],"AF0000")
im_fill(frame2,[6,6],[18,18],"AF0000")
im_fill(frame2,[6,6],[19,19],"AF0000")
#dust1
im_fill(frame2,[2,10],[13,13],"333333")
im_fill(frame2,[3,9],[12,14],"333333")
im_fill(frame2,[4,8],[11,15],"333333")
im_fill(frame2,[5,7],[10,16],"333333")
im_fill(frame2,[6,6],[9,17],"333333")
im_show(frame2)


#frame3
#land
#im_fill(frame1,[7,8],[0,19],"f4c169")
im_fill(frame3,[18,19],[0,19],"006400")

#sun
im_fill(frame3,[0,2],[0,2],"ffdf11")
#HERMINIO character
im_fill(frame3,[8,19],[0,2],"333333")
im_fill(frame3,[10,19],[3,5],"333333")
im_fill(frame3,[12,19],[6,8],"333333")
im_fill(frame3,[14,19],[9,11],"333333")
im_fill(frame3,[16,19],[0,14],"333333")
im_fill(frame3,[18,19],[15,17],"333333")
im_show(frame3)

#frame4
#tree
im_fill(frame4,[9,9],[7,7],"008000")
im_fill(frame4,[9,9],[9,9],"008000")
im_fill(frame4,[9,9],[11,11],"008000")
im_fill(frame4,[10,10],[8,10],"008000")
im_fill(frame4,[11,11],[7,11],"008000")
im_fill(frame4,[12,12],[8,8],"008000")
im_fill(frame4,[12,12],[10,10],"008000")
im_fill(frame4,[13,13],[7,7],"008000")
im_fill(frame4,[13,13],[11,11],"008000")
im_fill(frame4,[12,15],[9,9],"800000")
im_fill(frame4,[16,16],[8,10],"800000")
im_fill(frame4,[17,17],[7,11],"800000")

#land
#im_fill(frame1,[7,8],[0,19],"f4c169")
```

```
im_fill(frame4,[18,19],[0,19],"006400")

#sun
im_fill(frame4,[0,2],[0,2],"ffdf11")
 #Michaels character
im_fill(frame4,[9,9],[3,6],"000000")
im_fill(frame4,[10,10],[3,6],"0929FF")
im_fill(frame4,[10,10],[6,6],"0929FF")
im_fill(frame4,[10,10],[4,5],"333333")
im_fill(frame4,[11,11],[3,6],"000000")
im_fill(frame4,[12,12],[1,1],"000000")
im_fill(frame4,[12,12],[4,5],"000000")
im_fill(frame4,[12,12],[8,8],"000000")
im_fill(frame4,[13,13],[1,8],"000000")
im_fill(frame4,[14,14],[4,5],"000000")
im_fill(frame4,[15,15],[1,8],"000000")
im_fill(frame4,[16,16],[1,1],"000000")
im_fill(frame4,[16,16],[4,5],"000000")
im_fill(frame4,[16, 16],[8,8],"000000")
im_fill(frame4,[17,17],[3,3],"000000")
im_fill(frame4,[17,17],[6,6],"000000")
im_fill(frame4,[18,18],[3,3],"000000")
im_fill(frame4,[18,18],[6,6],"000000")

 #JUAN character
im_fill(frame4,[13,13],[13,13],"333333")
im_fill(frame4,[14,14],[13,13],"333333")
im_fill(frame4,[15,15],[13,13],"000000")
im_fill(frame4,[15,15],[12,12],"333333")
im_fill(frame4,[15,15],[14,14],"333333")
im_fill(frame4,[14,14],[12,12],"000000")
im_fill(frame4,[14,14],[14,14],"000000")
im_fill(frame4,[16,19],[13,13],"92370F")

#HERMINIO character
im_fill(frame4,[7,16],[9,9],"333333")
im_fill(frame4,[8,16],[8,10],"333333")
im_fill(frame4,[9,15],[7,11],"333333")
im_fill(frame4,[10,14],[6,12],"333333")
im_fill(frame4,[11,13],[5,13],"333333")
im_fill(frame4,[12,12],[4,14],"333333")
#dust

im_fill(frame4,[0,0],[17,17],"F90707")
im_fill(frame4,[1,1],[15,15],"F90707")
```

```
im_fill(frame4,[1,1],[19,19],"F90707")
im_fill(frame4,[2,2],[14,14],"F90707")
im_fill(frame4,[3,3],[12,12],"F90707")
im_fill(frame4,[3,3],[17,17],"F90707")
im_fill(frame4,[4,4],[15,15],"F90707")
im_fill(frame4,[5,5],[12,12],"F90707")
im_fill(frame4,[5,5],[18,18],"F90707")
im_fill(frame4,[6,6],[10,10],"F90707")
im_fill(frame4,[6,6],[16,16],"F90707")
im_fill(frame4,[7,7],[11,11],"F90707")
im_fill(frame4,[7,7],[14,14],"F90707")
im_fill(frame4,[6,6],[15,15],"F90707")
im_fill(frame4,[7,7],[19,19],"F90707")
im_fill(frame4,[8,8],[12,12],"F90707")
im_fill(frame4,[9,9],[15,15],"F90707")
im_fill(frame4,[9,9],[17,17],"F90707")
im_fill(frame4,[11,11],[15,15],"F90707")
#dust
im_fill(frame4,[0,0],[16,16],"AF0000")
im_fill(frame4,[0,0],[18,19],"AF0000")
im_fill(frame4,[1,1],[14,14],"AF0000")
im_fill(frame4,[1,1],[14,18],"AF0000")
im_fill(frame4,[2,2],[15,19],"AF0000")
im_fill(frame4,[3,3],[13,16],"AF0000")
im_fill(frame4,[3,3],[18,19],"AF0000")
im_fill(frame4,[4,4],[12,14],"AF0000")
im_fill(frame4,[4,4],[15,19],"AF0000")
im_fill(frame4,[5,5],[10,11],"AF0000")
im_fill(frame4,[5,5],[12,17],"AF0000")
im_fill(frame4,[5,5],[19,19],"AF0000")
im_fill(frame4,[6,6],[11,14],"AF0000")
im_fill(frame4,[6,6],[16,19],"AF0000")
im_fill(frame4,[7,7],[10,10],"AF0000")
im_fill(frame4,[7,7],[12,13],"AF0000")
im_fill(frame4,[7,7],[15,18],"AF0000")
im_fill(frame4,[8,8],[11,11],"AF0000")
im_fill(frame4,[8,8],[13,17],"AF0000")
im_fill(frame4,[9,9],[12,14],"AF0000")
im_fill(frame4,[9,9],[16,16],"AF0000")
im_fill(frame4,[10,10],[13,15],"AF0000")
im_fill(frame4,[11,11],[14,14],"AF0000")
im_show(frame4)

#frame5
im_fill(frame5,[6,6],[4,7],"a80a0a")
```

```
im_fill(frame5,[6,13],[4,4],"a80a0a")
im_fill(frame5,[9,9],[5,6],"a80a0a")
#letter I
im_fill(frame5,[7,8],[9,10],"a80a0a")
im_fill(frame5,[10,13],[9,9],"a80a0a")
im_fill(frame5,[10,13],[10,10],"a80a0a")
#latter N
im_fill(frame5,[7,13],[12,12],"a80a0a")
im_fill(frame5,[8,8],[13,15],"a80a0a")
im_fill(frame5,[9,13],[15,15],"a80a0a")
im_show(frame5)


frame_list = [frame1,frame2,frame4,frame3,frame5]
fps = 10
play_video = vid_show(frame_list,fps)
```

Appendix F

The Intermediate Python Code Developed for the Students at the University

```
====import cv2
import os
import sys
import matplotlib.pyplot as pyplot
from matplotlib import animation as animation
import numpy as np
import re

grid_lines = False
SAFE = True
easy_messages = None #if you don't want traceback turn this on

if easy_messages:
    sys.tracebacklimit = 0

def grid_lines_on(width, height):
    fig1, ax = pyplot.subplots() # make figure
    ax.grid(linestyle='-',linewidth=0.5)

    xticks = np.arange(-0.5,height -0.5,1)
    yticks = np.arange(-0.5, width-0.5,1)

    ax.set_xticks(xticks)
    ax.set_xticklabels([int(y+0.5) for y in xticks])
    ax.set_yticks(yticks )
    ax.set_yticklabels([int(x+0.5) for x in yticks ])
    return fig1,ax

def check_input(img,which_lib):
    if img is None:
        #only need to do this once for the entire thing
        raise TypeError('You input an image with '+str(type(img))+', check your file path,
you may have typed it incorrectly.')

    if which_lib == 'cv':
        if isinstance(img,np.ndarray):
            if SAFE:
                if img.dtype!=np.uint8:
                    print ("Type mismatch...converting for you...")
                    #im_show(img)
                    img = matrix_to_img(img)
```

```
                return img
            else:
                return img
        else:
            if img.dtype!=np.uint8:
                raise TypeError("Type mismatch...check your input! You may need to use
matrix_to_img().")
            else:
                return img
    else: #this case is a list
        if SAFE:
                print ("Type mismatch...converting for you...")
                #im_show(img)
                img = matrix_to_img(img)
                return img
        else:
                raise TypeError("Type mismatch...check your input! You may need to use
matrix_to_img().")

if which_lib == 'custom':
    if isinstance(img,np.ndarray):
            if img.dtype==np.uint8:
                raise TypeError ("Type mismatch...Check your input! These functions do not
work with OpenCV images.")
if which_lib == 'save_list':
    if isinstance(img,np.ndarray):
        if img.dtype!=np.uint8:
            img = matrix_to_img(img)
            return img
        else:
            return img
    else:
        raise TypeError("Type mismatch...Check your input!")

def check_format(input):
    '''
    Function to check that student input is in string format, will correct for them in safe
mode.
    '''
    if type(input) != np.string_:
        #print type(input)
        if SAFE:
                #print "Input was not in string format, be sure to use quotes, correcting for
you..."
                return str(input)
```

```
        else:
            raise TypeError ("Input is not in string format, please correct this by using
quotes.")
    else:
        return input

def hex_to_color(s):
    ''' Helper function for translating hex input to RGB color strings, used in makergb.

    Inputs:
    s: hex color string without # prefix

    Outputs:
    RGB tuple as (r,g,b) in decimal format
    '''
    hexColorPattern = re.compile("\A#[a-fA-F0-9]{6}\Z")
    if not isinstance(s, str):
        raise TypeError('hex2color requires a string argument')
    if hexColorPattern.match(s) is None:
        raise ValueError('invalid hex color string "%s"' % s)
    return tuple([int(n, 16)/255.0 for n in (s[1:3], s[3:5], s[5:7])])

def bnw_to_hex(bnw):
    '''
    Converts '0' digit to correct '000000' pattern.

    Inputs:
    bnw: A user created matrix containing '0' or '1' values for black or white.

    Outputs:
    Returns the same matrix but in hex format.
    '''
    bnw=np.array(bnw)
    rows = bnw.shape[0]
    columns = bnw.shape[1]
    hex_bnw=[[]]*rows
    for i in range(rows):
        hex_bnw[i]=[[]]*columns
        for j in range(columns):
            if bnw[i][j]=='0':
                hex_bnw[i][j]='000000'
            elif bnw[i][j]=='1':
                hex_bnw[i][j]='FFFFFF'
    return np.array(hex_bnw)
```

```
def rgb_to_gray(rgb):
    '''
    Helper function for making images grayscale in vidfill.

    Inputs:
    rgb: An nxn matrix filled with RGB tuples.

    Outputs:
    A nxn matrix with gray value tuples.
    '''
    if rgb.shape[0]*rgb.shape[1]>400:
        print ("Image too large!! Shrinking...")
        rgb = rgb[0:20,0:20]
    rows = rgb.shape[0]
    columns = rgb.shape[1]
    gray=[[]]*rows
    for i in range(rows):
        gray[i]=[[]]*columns
        for j in range(columns):
            r,g,b= rgb[i][j][0], rgb[i][j][1], rgb[i][j][2]
            gray[i][j]=0.2125 *r + 0.7154 *g + 0.0721 *b
    return gray

def make_rgb(matrix): #helper function for vidfill
    '''
    Helper function used in vidfill to convert hex code to rgb.

    Inputs:
    matrix: A nxn matrix filled with hex values.

    Outputs:
    A nxn numpy array filled with rgb tuples.

    '''
    matrix = np.array(matrix)
    if len(matrix[0][0]) == 1:
        matrix = bnw_to_hex(matrix)
    if matrix.shape[0]*matrix.shape[1] > 400:
        print ("Image too large!! Shrinking...")
        matrix = matrix[0:20,0:20]
    rows = matrix.shape[0]
    columns = matrix.shape[1]
    matrix2 = [[]]*rows
    for i in range(rows):
        matrix2[i] = [[]]*columns
```

```
            for j in range(columns):
                if len(matrix[i][j]) == 2:
                    t = matrix[i][j]+matrix[i][j]+matrix[i][j]
                    t = check_format(t)
                    color = hex_to_color('#'+t)
                    matrix2[i][j] = (color[0],color[1],color[2])
                elif len(matrix[i][j]) == 1:
                    print('invalid matrix')
#                     print ("Invalid length for matrix["+str(i)+"]["+str(j)+"]'s element, please use
hexadecimal format.")
                else:
                    t = matrix[i][j]
                    t = check_format(t)
                    color = hex_to_color('#'+t)
                    matrix2[i][j] = (color[0],color[1],color[2])
        return np.array(matrix2)

    def im_show(matrix): #previously aolme_imshow
        '''
        A function that shows a single nxn matrix frame on the screen.

        Inputs:
        matrix: A nxn matrix filled with hex values (without leading #) or 0's and 1's.

        Outputs:
        A figure containing the designed image frame in color, grayscale or black and white.

        '''
        check_input(matrix,'custom')
        matrix = make_rgb(matrix)
        if matrix.shape[0]*matrix.shape[1]>400:
            print ("Image too large!! Shrinking...")
            matrix = matrix[0:20,0:20]
        if (len(matrix[0][0])>1):
            if not grid_lines:
                pyplot.figure()
                pyplot.tick_params(axis='both', which='both', bottom='off', top='off',
labelbottom='off', right='off', left='off', labelleft='off')
                pyplot.imshow(matrix, interpolation='none')
                pyplot.tight_layout()
            else:
                fig,x =grid_lines_on(matrix.shape[0],matrix.shape[1])
                pyplot.grid(linestyle='-', linewidth=0.5)
                pyplot.imshow(matrix, interpolation='none')
                pyplot.tight_layout()
```

```
        else:
            if not grid_lines:
                pyplot.figure()
                pyplot.tick_params(axis='both', which='both', bottom='off', top='off',
labelbottom='off', right='off', left='off', labelleft='off')
                pyplot.imshow(matrix, interpolation='none')
                pyplot.tight_layout()
            else:
                grid_lines_on(matrix.shape[0],matrix.shape[1])
                pyplot.grid(linestyle='-', linewidth=0.5)
                pyplot.imshow(matrix, interpolation='none')
                pyplot.tight_layout()
        pyplot.show()

    def im_fill(matrix, rng_rows, rng_cols, val): #previously aolme_imfill
        '''
       A function that fills a range of rows and columns with a single color value.

       Inputs:
       matrix: A nxn sized matrix. Can be empty or have been previously filled.
       rng_rows: A range of rows input as [from,to].
       rng_cols: A range of columns input as [from,to].
       val: A hex color value or 0 or 1 which will fill the requested ranges of rows and
columns.

       Outputs:
       The same nxn matrix but with range of rows and columns filled with val.

       '''
        check_input(matrix,'custom')
        col_0 = [row[0] for row in matrix] # Getting column zero
        ncols = len(matrix[0])
        nrows = len(col_0)

        nrows_portion = rng_rows[1] - rng_rows[0]
        ncols_portion = rng_cols[1] - rng_cols[0]

        if (nrows_portion < 0) or (ncols_portion < 0):
            print ('( getportion) Error: Wrong range declaration!');
            return None;

        if (rng_rows[1] > nrows) or (rng_cols[1] > ncols):
            print ('( getportion) Error: Index out of range!');
            return None;
```

```
    for i in range(rng_rows[0], rng_rows[1] + 1):
        for j in range(rng_cols[0], rng_cols[1] + 1):
            matrix[i][j] = val;

    return matrix;

def  im_print(matrix): #previously aolme_imprint
    '''
    A function that will print the contents of a matrix.

    Inputs:
    matrix: A nxn matrix.

    Outputs:
    Text printout of the entire matrix's contents.

    '''
    check_input(matrix,'custom')
    matrix = np.array(matrix)
    print ("img = ",matrix)

    return None;

def vid_show(vid,fps):    #previously aolme_vidshow
    '''
   A function that 'plays' a list of frame, creating a 2d video. Note, this must be set equal to
some value to work!!!

   Inputs:
   vid: A list of frames, set as [frame0,frame1,...,framen], where each frame is a nxn matrix
of the same size.
   fps: A number which represents the number of frames that should be played per second.

   Outputs:
   A visual animation containing each frame in the order listed. Returns the animation.

   '''
    matrixf = make_rgb(vid[0])
    if not grid_lines:
        fig = pyplot.figure(2)
        pyplot.tick_params(axis='both', which='both', bottom='off', top='off',
labelbottom='off', right='off', left='off', labelleft='off')
    else:
        fig1,ax = grid_lines_on(matrixf.shape[0],matrixf.shape[1])
    fps = 1000./fps
```

```
    if len(vid) < 1:
        print ("Incorrect input, make sure you give function a video to play!")

    if matrixf.shape[0]*matrixf.shape[1]>400:
            print ("Image too large!! Shrinking...")
            i = 0
            for frame in vid:
                frame = frame[0:20,0:20]
                vid[i]=frame
                i+=1
    im = pyplot.imshow(matrixf, interpolation='none')

    # function to update figure
    def update_fig(j):
        # set the data in the axesimage object
        frame = make_rgb(vid[j])
        im.set_array(frame)
        pyplot.draw()
        return im,
    # kick off the animation
    if (grid_lines):
        ani = animation.FuncAnimation(fig1, update_fig, frames=range(len(vid)),
                            interval=fps, blit=False, repeat=True)
    else:
        ani = animation.FuncAnimation(fig, update_fig, frames=range(len(vid)),
                            interval = fps, blit=True, repeat=True)
    pyplot.tight_layout()
    pyplot.show()

    return ani

def save_vid(vid,fps,name):
    if os.name != 'nt':
        vid.save(name,fps = fps, writer='imagemagick')

def matrix_to_img(matrix): #previously me_matrix2img
    '''
    Takes a nxn image frame and converts it to jpg format, saves it and shows the image.

    Inputs:
    matrix: A nxn matrix filled with hex colors.

    Outputs:
    A .jpg file saved to disc as picture.jpg, and the image is also displayed on screen.
    Returns the matrix as an opencv image.
```

```
    '''
    check_input(matrix,'custom')
    try:
        os.remove("picture.jpg")
    except:
        pass
    matrix = make_rgb(matrix)
    pyplot.axis('off')
    pyplot.imshow(matrix, interpolation='none')
    pyplot.savefig('picture.jpg',format='jpg',bbox_inches='tight', pad_inches=0)
    pyplot.show()
    pyplot.axis('on')
    c = cv2.imread('picture.jpg', 1)
    c = cv2.resize(c, (600, 400))
    #cv2.imshow('picture',c)
    try:
        os.remove("picture.jpg")
    except:
        pass

    return c

def make_img_gray(img):#previously me_rgb2gray
    '''
    Convert an open image to grayscale.

    Inputs:
    img: An open image file.

    Outputs:
    Returns the same image except converted to grayscale.

    '''
    img = check_input(img,'cv')

    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    show_img(gray_img)
    return gray_img

def show_img(img):#previously me_imshow
    '''
    Displays an open image on screen.

    Inputs:
```

```
        img: An open image file.

        Outputs:
        Displays the open image file on screen.
        '''
        check_input(img,'cv')
        cv2.imshow('picture',img)
        #this part looks ridiculous but it's a bug in cv2 with live interpreters.
        cv2.waitKey(0)
        cv2.destroyAllWindows()
        cv2.waitKey(0)
        cv2.destroyAllWindows()
        cv2.waitKey(0)
        cv2.destroyAllWindows()
        cv2.waitKey(0)

    def read_img(img): #previously me_imread
        '''
        Reads an image from disc.

        Inputs:
        img: A string containing the name of the image to be read on disc, with the file
extension.

        Outputs:
        Returns the read image as a numpy array.

        '''
        c = cv2.imread(img,1)
        return c

    def save_img(img,name):#previously me_imsave
        '''
        Saves an open image from variable to disc.

        Inputs:
        img: An open image file in a variable in numpy array format.
        name: A string containing the name the image should be saved as, including the file
extension.

        Outputs:
        A saved image file on disc inside the same folder as the python script.

        '''
        img = check_input(img,'save_list')
```

```
        cv2.imwrite(name,img)

    def get_pixel(img,loc): #previously me_impix
        '''
        Gets a single pixel from an open image.

        Inputs:
        img: The image from which to get a pixel, stored as a numpy array in a variable.
        loc: The x,y location of the desired pixel, input as [x,y]

        Outputs:
        Returns the color of the pixel at the desired location as (r,g,b).

        '''
        check_input(img,'cv')

        pixel = img[loc[0],loc[1]]

        return [pixel[2],pixel[1],pixel[0]]

    def img_size(img):#previously me_imsize
        '''
        Returns the size of the image.

        Inputs:
        img: The image from which to get a pixel, stored as a numpy array in a variable.

        Outputs:
        Returns the number of rows and columns in the array in (numberofrows,
numberofcolums).

        '''
        check_input(img,'cv')

        print ('# of rows: ' + str(img.shape[1])) # of rows
        print ('# of cols: ' + str(img.shape[0])) # of columns
        size = (img.shape[1], img.shape[0]) # (nrows, ncols)
        return size;

    def show_comps(img): #previously me_showcomps
        '''
        Displays the red, blue, and green components of an image on screen.

        Inputs:
        img: An image stored as a numpy array in a variable.
```

```
        Outputs:
        Displays on screen the red, green and blue components of the given image. Returns
nothing.

        '''
        check_input(img,'cv')
        #image = cv2.imread(img)
        #zero = np.zeros(image.shape)

        #zero = np.zeros((image.shape[0],image.shape[1],3), np.float64)
        zeros = np.zeros((img.shape[0], img.shape[1]), np.uint8)
        B,G,R = cv2.split(img)

        blue_component = cv2.merge((B, zeros, zeros))
        green_component = cv2.merge((zeros, G, zeros))
        red_component = cv2.merge((zeros, zeros, R))

        cv2.imshow('blue',blue_component)
        cv2.waitKey(0)
        cv2.imshow('red',red_component)
        cv2.waitKey(0)
        cv2.imshow('green',green_component)
        cv2.waitKey(0)
        #more of this due to cv2 bug...maybe put in a loop to look 'nicer'
        cv2.destroyWindow('green')
        cv2.waitKey(-1)
        cv2.destroyWindow('green')
        cv2.waitKey(-1)
        cv2.destroyWindow('green')
        cv2.waitKey(-1)
        cv2.destroyWindow('green')
        cv2.waitKey(-1)
        return [red_component,green_component,blue_component]

    def get_comps(img):#previously me_getcomps
        '''
        Returns a list of the different combinations of rgb components of an image in a list.

        Inputs:
        img: An image stored as a numpy array in a variable.

        Outputs:
        Returns a list of images stored as numpy arrays for each combination of components
RGB.
```

```
    Val[0] is an image of only red component.
    Val[1] is an image of only green component.
    Val[2] is an image of only blue component.
    Val[3] is an image of only green and red components creating yellow.
    Val[4] is an image of only blue and green components creating cyan.
    Val[5] is an image of only blue and red components creating magenta.

    '''
    check_input(img,'cv')
    #image = cv2.imread(img)
    #zero = np.zeros(image.shape)

    #zero = np.zeros((image.shape[0],image.shape[1],3), np.float64)
    zeros = np.zeros((img.shape[0], img.shape[1]), np.uint8)
    B,G,R = cv2.split(img)

    Ir = cv2.merge((B, zeros, zeros))
    Ig = cv2.merge((zeros, G, zeros))
    Ib = cv2.merge((zeros, zeros, R))
    Iy = cv2.merge((zeros, G, R))
    Ic = cv2.merge((B, G, zeros))
    Im = cv2.merge((B, zeros, R))
    return (Ir, Ig, Ib, Iy, Ic, Im) # these are  RGB images

def rotate_img(img,degrees): #previously  me_imrotate
    '''
    Rotates an image.

    Inputs:
    img: An image file stored in a variable as a numpy array.
    degrees: The amount of degrees the image should be rotated by.

    Outputs:
    Displays the rotated image on screen.

    '''
    check_input(img,'cv')
    (h, w) = img.shape[:2]
    center = (w / 2, h / 2)

    M = cv2.getRotationMatrix2D(center, degrees, 1.0)
    rotated = cv2.warpAffine(img, M, (w, h))
    show_img(rotated)
    return rotated
```

```
def crop_img(img,ranges):#previously me_imcrop
    '''
    Trims edges off of an image.

    Inputs:
    img: An image file saved as a numpy array in a variable.
    ranges: An array filled with pixel values saved as int.
    ranges[0]: the x1 value from where to start cropping as part of (x1,y1) coordinates.
    ranges[1]: the x2 value from where to end cropping as part of (x2,y2) coordinates.
    ranges[2]: the y1 value from where to start the cropping as part of (x1,y1) coordinates.
    ranges[3]: the y2 value from where to end the cropping as part of (x2,y2) coordinates.

    Outputs:
    Displays the cropped image on screen.

    '''
    check_input(img,'cv')
    cropped = img[ranges[0]:ranges[1],ranges[2]:ranges[3]]
    show_img(cropped)
    return cropped

def put_pixel(img, position, val):#previously me_putpixel
    '''
    Places a pixel on an image at a chosen location.

    Inputs:
    img: An image file saved as a numpy array in a variable.
    position: the position at which to place the pixel, given in (x,y) coordinates as
position[0] for x and position [1] for y.
    val: the rbg or black and white value or color of the pixel to be placed on the image,
with val = 0 or 1 or val=[r,g,b] where r, g, and b are float values which define a color.

    Outputs:
    No outputs, the pixel is saved on the original image and must be displayed using
showimg(img).

    '''
    check_input(img,'cv')
    if len(img.shape) < 3: # grayscale
        img[position[1],position[0]] = val
    else: # RGB
        img[position[1],position[0]] = (val[2], val[1], val[0])
    return None

def put_pixel_group(img, ranges, val): #previously me_putpixelgroup
```

```
        '''
        Places a group of pixels onto an image at a chosen location.

        Inputs:
        img: An image file saved as a numpy array in a variable.
        range: An array filled with pixel values saved as int.
        range[0]: the x1 value from where to start paste as part of (x1,y1) coordinates.
        range[1]: the x2 value from where to end paste as part of (x2,y2) coordinates.
        range[2]: the y1 value from where to start the paste as part of (x1,y1) coordinates.
        range[3]: the y2 value from where to end the paste as part of (x2,y2) coordinates.
        val: the rbg or black and white value or color of the pixel to be placed on the image,
with val = 0 or 1 or val=[r,g,b] where r, g, and b are float values which define a color.

        Outputs:
        No outputs, the pixel range is saved on the original image and must be displayed using
showimg(img).

        '''
        check_input(img,'cv')
        nra = ranges[0]
        nrb = ranges[1]
        nca = ranges[2]
        ncb = ranges[3]
        for i in range(nra, nrb+1):
            for j in range(nca, ncb+1):
                if len(img.shape) < 3: # grayscale
                    img[i,j] = val
                else: # RGB
                    img[i,j] = (val[2], val[1], val[0])
        return None

    def print_img_info(img):
        '''
        Prints information about a user-created 2d image.

        Inputs:
        img: A user created 2d matrix filled with color values.

        Outputs:
        Prints on screen the number of pixes (rows*columns), image type (grayscale, color or
black and white), height (number of rows), and width (number of columns)
        '''
        #check_input(img,'cv')
        if isinstance(img,list):
          im_show(img)
```

```
     img = np.array(img)
    elif img.dtype != np.uint8:
        im_show(img)
    else:
        show_img(img)
    print ("Num of pixels: ", img.shape[0]*img.shape[1])
    print ("Height: ", img.shape[1])
    print ("Width: ", img.shape[0])
    if len(img[0][0]) == 3:
        print ("RGB color")
    else:
        print ("black and white")

    return None

def print_vid_info(vid):
    '''
    Prints information about a user created video.

    Inputs:
    vid: A list of 2d matrices filled with color values, created by the user in format
[frame0,frame1,frame2...]

    Outputs:
    Prints the number of pixels on each frame (height*width), the height(number of rows),
width (number of columns), number of frames, and whether the video is color, graysale or black
and white.
    '''
    for i in range(len(vid)):
        vid[i] = np.array(vid[i])
    print ("Num of pixels: ", vid[0].shape[0]*vid[0].shape[1])
    print ("Height: ", vid[0].shape[1])
    print ("Width: ", vid[0].shape[0])
    print ("Num of frames: ", len(vid))
    if len(vid[0][0][0]) == 3:
        print ("RGB color")
    else:
        print ("black and white")
    return None

def print_img_segment(img,ranges):
    '''
    Prints a portion of a user created image.

    Inputs:
```

```
        img: A user defined 2d matrix filled with color values.
        ranges: A list of ranges which define the portion of the matrix to be printed, defined as
a list of numbers with [x1,x2,y1,y2] coordinates.

        Outputs:
        Displays an image on screen containing only the portion of the original image
requested by the user.

        '''
        check_input(img,'custom')
        img = np.array(img)
        im_seg = img[ranges[0]:ranges[1],ranges[2]:ranges[3]]
        im_show(im_seg)
        return None

    def print_vid_segment(vid,ranges,frames,fps):
        '''
        Prints a portion of a user created video.

        Inputs:
        vid: A list of 2d matrices filled with color values, created by the user in format
[frame0,frame1,frame2...]
        ranges: A list of ranges which define the portion of the matrix to be printed, defined as
a list of numbers with [x1,x2,y1,y2] coordinates.
        frames: A range of frames to play on the video, must be continuous, input as
[startframe,endframe]...need to add all as an option.
        fps: The rate at which the video should be played.

        Outputs:
        Displays an image on screen containing only the portion of the original image
requested by the user.

        '''
        if len(vid) < 2:
            print ("Please pass a video to the function.")
        vid_seg=[]
        for vids in vid:
            vids = np.array(vids)
            vid_seg.append(vids[ranges[0]:ranges[1],ranges[2]:ranges[3]])
        vid_seg = vid_seg[frames[0]:frames[1]]
        return vid_show(vid_seg,fps)
```

References

Aguirre, J., Herbel-Eisenmann, B., Celedón-Pattichis, S., Civil, M., Wilkerson, T., Stephan, M., & Clements, D. H. (2017). Equity within mathematics education research as a political act: Moving from choice to intentional collective professional responsibility. Journal for Research in Mathematics Education, 48(2), 124–147

Alexander, R.J. (2005). Culture, dialogue and learning: Notes on an emerging pedagogy. *Paper presented at the Conference of the International Association for Cognitive Education and Psychology*. University of Durham, UK.

Alexander, R. J. (2008). *Towards dialogic teaching: Rethinking classroom talk*. Dialogos.

Alexander, R. (2018). Developing dialogic teaching: Genesis, process, trial. *Research Papers in Education*, *33*(5), 561-598.

Alexandron, G., Armoni, M., Gordon, M., & Harel, D. (2014, May). Scenario-based programming: Reducing the cognitive load, fostering abstract thinking. *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 311-320).

Alshwaikh, J., & Morgan, C. (2018). A framework for the study of written and spoken discourse: school mathematics in Palestine. *ZDM*, *50*(6), 1041-1051.

Anaf, S., Drummond, C., & Sheppard, L. A. (2007). Combining case study research and systems theory as a heuristic model. *17*(10), 1309-1315.

Arastoopour Irgens, G., Dabolkar, S., Bain, C., Woods, P., Hall, K., Swanson, H., Horn, M. ... & Wilensky, U. (2019). Modeling and measuring students' computational thinking practices in science. *Journal of Science Education and Technology*. *29*(1), 137-161.

Armoni, M. (2013). On teaching abstraction in computer science to novices. *Journal of Computers in Mathematics and Science Teaching*, *32*(3), 265–284.

Baldwin, D., Walker, H. M., & Henderson, P. B. (2013). The roles of mathematics in computer science. Association for Computing Machinery *(ACM) Acm Inroads*, *4(*4), 74-80.

Bandura, A. (2000). Exercise of human agency through collective efficacy. *Current Directions in Psychological Science*, *9*(3), 75-78.

Bandura, A. (2001). Social cognitive theory: An agentic perspective. *Annual review of psychology*, *52*(1), 1-26.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *Acm Inroads*, *2*(1), 48-54.

Barrozo do Amaral Villares, A., & de Carvalho Moreira, D. (2017). *Python on the Landscape of Programming Tools for Design and Architectural Education*. SIGRADI, 2017. Concepción, Chile.

Bau, D., Gray, J., Kelleher, C., Sheldon, J., & Turbak, F. (2017). Learnable programming: Blocks and beyond. *Communications of the ACM*, *60*(6), 72–80. https://doi.org/10.1145/3015455.

Bell, T., & Vahrenhold, J. (2018). CS unplugged—how is it used, and does it work?. In *Adventures Between Lower Bounds and Higher Altitudes* (pp. 497-521). Springer, International Publishing.

Ben-Ari, M. (2012). *Mathematical logic for computer science*. Springer Science & Business Media.

Bernstein, B (1973). *Class, codes, and control, Vol. 2*. Routledge & Kegan Paul.

Bernstein, B (2000). *Pedagogy, symbolic control, and identity*. Rowman and Littlefield.

Bernstein, B. (2003). *Class, codes, and control: Applied studies towards a sociology of language* (Vol. 2). Psychology Press.

Bevan, B., & Michalchik, V. (2013). Where it gets interesting: Competing models of STEM learning after school. Afterschool Matters, 17, 1–8. Retrieved from https://files.eric.ed.gov/ fulltext/EJ1003837.pdf

Bholah, R. (2017). Developing problem-based learning approaches to water education in Mauritius. In *Schooling for Sustainable Development in Africa* (pp.119-127). Switzerland, Springer. International Publishing.

Billington, M. G., & Gabrielsen, E. (2017). The older the better? Are younger Norwegian adults losing ground on basic numeracy skills. *Nordic Studies in Mathematics Education*, *22*(1), 45-64.

Bloome, D., Carter, S. P., Christian, B. M., Otto, S., & Shuart-Faris, N. (2005). *Discourse analysis and the study of classroom language and literacy events: A microethnographic perspective*. Lawrence Erlbaum.

Blum-Martínez, R., & López, M. J. H. (Eds.). (2020). *The Shoulders we stand on: A history of bilingual education in New Mexico*. University of New Mexico Press.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing computational thinking in compulsory education-Implications for policy and practice* (No. JRC104188). Joint Research Centre (Seville site).

Bourke, C. (2018). *Computer Science I*. Department of Computer Science & Engineering, University of Nebraska–Lincoln, NE.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, *1* (25). Vancouver, Canada.

Buss, A., & Gamboa, R. (2017). Teacher transformations in developing computational thinking: Gaming and robotics use in after-school settings. In *Emerging research,* *practice, and policy on computational thinking* (pp. 189-203). Springer. International Publishing.

Cabral, T. & Baldino, R. (2002). Lacanian psychoanalysis and pedagogical transfer: Affect and cognition. In A. Cockburn & E. Nardi (Eds.), *Proceedings of the 26th conference of the International Group for the Psychology of Mathematics Education*, *2,* 169-176.

Caffarel, A., Martin, J. R., & Matthiessen, C. M. (Eds.). (2004). Language typology: A functional perspective (Vol. 253). John Benjamins Publishing.

Cairns, D., & Areepattamannil, S. (2019). Exploring the relations of inquiry-based teaching to science achievement and dispositions in 54 countries. *Research in Science Education*, *49*(1), 1-23.

Cazden, C. (2001). *Classroom discourse: The language of teaching and learning*. Portsmouth, NH: Heinemann.

Celedón-Pattichis, S., Borden, L., Pape, S., Clements, D., Peters, S., Males, J. Chapman, O., & Leonard, J. (2018). Asset based approaches to equitable education research and practice. Journal for Research in Mathematics Education, 49(4), 373-389.

Celedón-Pattichis, S., LópezLeiva, C. A., Pattichis, M. S., & Llamocca, D. (2013). An interdisciplinary collaboration between computer engineering and mathematics/bilingual education to develop a curriculum for underrepresented middle school students. *Cultural Studies of Science Education*, *8*(4), 873-887.

Chalmers, D. J. (2011). The Nature of epistemic space. In A. Egan & B. Weatherson's (Eds.) *Epistemic Modality* (pp. 60-107). Oxford University Press.

Chandra, V. (2015). Comparison between various software development methodologies. *International Journal of Computer Applications*, *131*(9), 7-10.

Chapin, S. H., O'Connor, C., O'Connor, M. C., & Anderson, N. C. (2009). *Classroom discussions: Using math talk to help students learn, Grades K-6*. Math Solutions.

Chauraya, M., & Brodie, K. (2018). Conversations in a professional learning community: An analysis of teacher learning opportunities in mathematics. *Pythagoras*, *39*(1), 1-9.

Chowdhury, B., Bart, A. C., & Kafura, D. (2018, February). Analysis of collaborative learning in a computational thinking class. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 143-148).

Christie, F. (2005). *Classroom discourse analysis: A functional perspective*. Bloomsbury Publishing.

Chval, K. B., Smith, E., Trigos-Carrillo, L., & Pinnow, R. J. (2021). *Teaching Math to Multilingual Students, Grades K-8: Positioning English Learners for Success*. Corwin.

Cobb, P., & Yackel, E. (1996). Constructivist, emergent, and sociocultural perspectives in the context of developmental research. *Educational Psychologist*, *31*(3-4), 175-190.

Code.org. (2020). Code.org. https://code.org/.

Cohen, E. G., & Lotan, R. A. (2014). *Designing Groupwork: Strategies for the Heterogeneous Classroom Third Edition*. Teachers College Press.

Colburn, T., & Shute, G. (2007). Abstraction in computer science. *Minds and Machines*, *17*(2), 169-184.

Cole, M., & the Distributed Literacy Consortium. (2006). *The fifth dimension: An after-school program built on diversity*. Russell Sage Foundation.

Collins, J., De Owens, K. O., Cabardo, V., Hug, S., Cota, R., & Pontelli, E. (2021, May). Culturally Sustaining Pedagogies in CS: Modern K12 Outreach and its Adaptability in a Global Crisis. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)* (pp. 1-5). IEEE.

Colombi, M. C., & Schleppegrell, M. J. (2002). Theory and practice in the development of advanced literacy. In M. J. Schleppegrell & M. C. Colombi (Eds.), Developing advanced literacy in first and second languages: Meaning with power (pp. 1–19). Lawrence Erlbaum.

Computer Science Framework Steering Committee. (2016). K-12 computer science framework. http:// www.k12cs.org.

Computer Science Teachers Association (CSTA) (2017). *CSTA K- 12 computer science standards, revised 2017*. http://www.csteachers.org/standards.

Conley, D. T., & French, E. M. (2014). Student ownership of learning as a key component of college readiness. *American Behavioral Scientist*, *58*(8), 1018-1034.

Crowley, K., & Jacobs, M. (2002). Islands of expertise and the development of family scientific literacy. In G. Leinhardt, K. Crowley, & K. Knutson (Eds.), Learning conversations in museums. Lawrence Erlbaum

Csizmadia, A., Curzon, P., Dorling M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking: A guide for teachers. (Available at: http://community.Computingatschool.org.uk/resources/2324). Unpublished report.

CSTA & ISTE. (2011). Operational definition of computational thinking for K-12 education. https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf.

Cutts, Q. , Esper, S., Fecho, M., Foster, S. R. & Simon, B. (2012, Sept, 12). The abstraction transition taxonomy: Developing desired learning outcomes through the lens of situated cognition. In: Clear, A., Sanders, K. and Simon, B. (Eds.) *Proceedings of the ninth annual international conference on international computing education research* (pp. 63-70). ACM. doi:10.1145/2361276.2361290.

Dagienė, V., Sentance, S., & Stupurienė, G. (2017). Developing a two-dimensional categorization system for educational tasks in informatics. *Informatica*, *28*(1), 23-44.

Darling-Hammond, L., & Richardson, N. (2009). Research review/teacher learning: What matters? *Educational Leadership*, *66*(5), 46–53.

DeJarnette, A. F. (2014). *Students' conceptions of trigonometric functions and positioning practices during pair work with Etoys*. University of Illinois at Urbana-Champaign.

Derewianka, B., & Jones, P. (2016). *Teaching language in context*. Oxford University Press.

Derry, S. J. (1996). Cognitive schema theory in the constructivist debate. *Educational Psychologist*, *31*(3-4), 163-174.

Dufva, T., & Dufva, M. (2019). Grasping the future of the digital society. *Futures*, *107*, 17-28.

Dümmel, N., Westfechtel, B., & Ehmann, M. (2019, April). Work in progress: Gathering requirements and developing an educational programming language. In *2019 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1-4). IEEE.

Edwards, N. (2021). Exploring pre-service science teachers' epistemic agency to develop their pedagogy for science teaching. In America, C., Edwards, N., Robinson, M.

(Eds) (pp. 145-165). *Teacher Education for Transformative Agency: Critical perspectives on design, content, and pedagogy*.

Eggins, S. and Slade, D. (2004). *Analysing casual conversation*. Equinox.

Evans, J., Morgan, C., & Tsatsaroni, A. (2006). Discursive positioning and emotion in school mathematics practices. *Educational Studies in Mathematics*, *63*(2), 209-226.

Fagan, B. J., & Payne, B. (2017). Learning to program in Python—by teaching it! In Proceedings. *Interdisciplinary STEM Teaching and Learning Conference 1*, 99-107.

Farnsworth, V., Kleanthous, I., & Wenger-Trayner, E. (2016). Communities of practice as a social theory of learning: A conversation with Etienne Wenger. *British journal of educational studies*, *64*(2), 139-160.

Fang, Z., Schleppegrell, M. J., & Cox, B. E. (2006). Understanding the language demands of schooling: Nouns in academic registers. *Journal of Literacy Research*, *38*(3), 247-273.

Farris, A.V., & Sengupta, P. (2014). Perspectival Computational Thinking for Learning Physics: A Case Study of Collaborative Agent-based Modeling. In: Proceedings of the 12th International Conference of the Learning Sciences. (ICLS 2014), pp 1102 - 1107

Farris, A. V., Dickes, A. C., & Sengupta, P. (2016). Development of Disciplined Interpretation Using Computational Modeling in the Elementary Science Classroom. Proceedings of the 13th International Conference of the Learning Sciences. (ICLS 2016), pp 282 - 290.

Fierro Lucero, J. (2020). *La explicación de ejercicios de matemáticas: un análisis de género y metafunciones* (Publication No. #) [Master's thesis, Benemérita Universidad Autónoma de Puebla]. Repository institucional Buap.

Floridi, L. (2008). The method of levels of abstraction. *Minds and machines*, *18*(3), 303-329.

Fryer, T. (2020). *A short guide to ontology and epistemology: Why everyone should be a critical realist*. https://tfryer.com/ontology-guide/

Gamoran, A., & Nystrand, M. (1992). Taking students seriously. In F. M. Newmann (Ed.), Student engagement and achievement in American secondary schools (pp. 40–61). Teachers College Press.

Gangwar, R. K., Kumar, M., Jaiswal, A. K., & Saxena, R. (2014). Performance analysis of image compression using fuzzy logic algorithm. *Signal & Image Processing*, *5*(2), 73.

Garcia, A. (2013). Construing experience in Spanish: Revisiting a systemic functional description of Spanish nuclear transitivity. *Signs Magazine* , *46* (81), 29-55.

Gee, P. (2014). *An introduction to discourse analysis: Theory and method* (4th Ed.). Routledge.

Giddens, J. F., & Brady, D. P. (2007). Rescuing nursing education from content saturation: The case for a concept-based curriculum. *Journal of Nursing Education*, *46* (2), 65-69.

Glaveanu, V. P., Hanchett Hanson, M., Baer, J., Barbot, B., Clapp, E. P., Corazza, G. E., ... & Sternberg, R. J. (2020). Advancing creativity theory and research: A socio−cultural manifesto. *The Journal of Creative Behavior*, *54*(3), 741-745.

González, G., & DeJarnette, A. F. (2012). Agency in a geometry review lesson: A linguistic view on teacher and student division of labor. *Linguistics and Education*, *23*(2), 182-199.

González, N., Andrade, R., Civil, M., & Moll, L. (2001). Bridging funds of distributed knowledge: Creating zones of practices in mathematics. *Journal of Education for Students Placed at Risk*, *6(*1 & 2), 115-132.

Goode, J., Skorodinsky, M., Hubbard, J., & Hook, J. (2020). Computer science for equity: Teacher education, agency, and statewide reform. *Frontiers in Education*, *3*(2), 47-53.

Google Inc. & Gallup Inc. (2016). Diversity gaps in computer science: Exploring the underrepresentation of girls, Blacks ,and Hispanics 2016. Retrieved from https://ser

vices.google.com/fh/files/misc/diversity-gaps-in-computer-science-report.pdf.

Green, T. R. G. (1989). Cognitive dimensions of notations. In Proceedings of the fifth conference of the British Computer Society, Human-Computer Interaction Specialist Group on People and computers V (pp. 443–460). Cambridge University Press.

Grissom, S., Mccauley, R., & Murphy, L. (2017). How student centered is the computer science classroom? A survey of college faculty. *ACM Transactions on Computing Education (TOCE)*, *18*(1), 1-27.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, *42*(1), 38-43.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer science education*, *25*(2), 199-237.

Grover, S., & Pea, R. (2018). Computational Thinking: A competency whose time has come. In S.Sentance, E. Barendsen, & S. Carsten (Eds.) Computer Science Education: Perspectives on teaching and learning in school (pp. 19-37). London: Bloomsbury Academic.

Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology*, *29*(1), 1-8.

Grosjean, F. (2012). *Bilingual: Life and reality* (Reprint edition). Harvard University Press.

Grosjean, F. & Li, P (2013). *The psycholinguistics of bilingualism.* Wiley-Blackwell.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38-43.

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer science education*, *25*(2), 199-237.

Grover, S., & Pea, R. (2018). Computational Thinking: A competency whose time has come. In S.Sentance, E. Barendsen, & S. Carsten (Eds.) Computer Science Education: Perspectives on teaching and learning in school (pp. 19-37). London: Bloomsbury Academic.

Grover, S., Basu, S., & Schank, P. (2018, February). What we can learn about student learning from open-ended programming projects in middle school computer science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*(pp. 999-1004).

Grover, S., Pea, R., & Cooper, S. (2016, March 2–5). Factors influencing computer science learning in middle school. *Proceedings of the 47th ACM technical symposium on computer science education* (pp. 552–557).

Gruber, H., & Vonèche, J. J. (Eds.) (1995). The essential Piaget. An Interpretive Reference and Guide. New Jersey: Aronson.

Haberman, B., & Kolikant, Y. B. D. (2001, June). Activating "black boxes" instead of opening "zipper"-a method of teaching novices basic CS concepts. In *Proceedings of the 6th annual conference on Innovation and technology in computer science education* (pp. 41-44).

Halliday, M.A.K. (1978). *Language as social semiotic: The social interpretation of language and meaning*. Arnold.

Halliday, M. A. K. (1984). Language as code and language as behaviour: A systemic-functional interpretation of the nature and ontogenesis of dialogue. In R. Fawcett, M. A. K. Halliday, S. M. Lamb, & A. Makkai (Eds.), *The semiotics of language and culture, Vol 1: Language as social semiotic* (pp. 3-35). Pinter.

Halliday, M. A. K., & Hasan, R. (1989). *Language, context, and text: Aspects of language in a social-semiotic perspective.* Oxford: Oxford University Press.

Halliday, M. A. K. (1993). Towards a language-based theory of learning. *Linguistics and Education*, *5*(2), 93-116.

Halliday, M. A. K. (2004). *The Language of Science, Vol 5*. Continuum.

Halliday, M. A. K., & Matthiessen, C. M. I. M. (2014). *Halliday's introduction to functional grammar* (4th ed.). Routledge.

Haseski, H. İ., Ilic, U., & Tugtekin, U. (2018). Defining a New 21st Century Skill-Computational Thinking: Concepts and Trends. *International Education Studies*, *11*(4), 29-42.

Herbel-Eisenmann, B. A. (2007). From intended curriculum to written curriculum: Examining the voice of a mathematics textbook. *Journal for Research in Mathematics Education*, *38*(4), 344-369.

Herbel-Eisenmann, B., & Wagner, D. (2007). A framework for uncovering the way a textbook may position the mathematics learner. *For the Learning of Mathematics*, *27*(2), 8-14.

Herbel-Eisenmann, B. A., & Otten, S. (2011). Mapping mathematics in classroom discourse. *Journal for Research in Mathematics Education*, *42*(5), 451-485.

Herbel-Eisenmann, B., Choppin, J., Wagner, D., & Pimm, D. (Eds.). (2012). *Equity in discourse for mathematics education*. Springer.

Herbel-Eisenmann, B. A., Steele, M. D., & Cirillo, M. (2013). (Developing) teacher discourse moves: A framework for professional development. *Mathematics Teacher Educator*, *1*(2), 181-196.

Herbel-Eisenmann, B., Sinclair, N., Chval, K. B., Clements, D. H., Civil, M., Pape, S. J., ... & Wilkerson, T. L. (2016). Research committee: Positioning mathematics education researchers to influence storylines. *Journal for Research in Mathematics Education*, *47*(2), 102-117.

Hershkowitz, R., Schwarz, B. B., & Dreyfus, T. (2001). Abstraction in context: Epistemic actions. *Journal for Research in Mathematics Education*, *32*(2), 195-222.

Hooshyar, D., Pedaste, M., Yang, Y., Malva, L., Hwang, G. J., Wang, M., & Delev, D. (2021). From gaming to computational thinking: An adaptive educational computer game-based learning approach. *Journal of Educational Computing Research*, *59*(3), 383-409.

Hoppe, H. U., & Werneburg, S. (2019). Computational Thinking—More Than a Variant of Scientific Inquiry!. In: S.-C. Kong, H. A. (Ed.), Computational Thinking Education. Springer.

Hopscotch Technologies. (2020). Hopscotch. https://www.gethopscotch.com/.

Hu, C. (2011, June). Computational thinking: what it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (pp. 223-227).

Hsu, Y., Irie, N. R., & Ching, Y. (2019). Computational thinking educational policy initiatives (CTEPI) across the globe. *TechTrends,63*, 260–270.

Inti, M. M., Latib, A. A., Bakare, F. S., Kamin, Y., Amin, N. F., & Saud, M. S. (2018). Pedagogical approaches required for fostering interpersonal skills among mechanical engineering technology students. *Advanced Science Letters*. https://doi.org/10.1166/asl.2018.11077.

Israel-Fishelson, R., Hershkovitz, A., Eguíluz, A., Garaizar, P., & Guenaga, M. (2021). The associations between computational thinking and creativity: The role of personal characteristics. *Journal of Educational Computing Research*, *58*(8), 1415-1447.

ISTE, & CSTA. (2011). *Computational thinking in K–12 education leadership toolkit*.

ISTE. (2016). *CT leadership toolkit.* http://www.iste.org/ docs/ct-documents/ct-leadershipt-toolkit.pdf?sfvrsn=4.

Jacob, S., Nguyen, H., Tofel-Grehl, C., Richardson, D., & Warschauer, M. (2018). Teaching computational thinking to English learners. *NYS TESOL Journal*, *5*(2), 12-24.

Kafai, Y. B. (1995). Minds in play: Computer game design as a context for children's learning. Hillsdale: Lawrence Erlbaum Associates

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. Mit Press.

Kafai, Y. B., & Vasudevan, V. (2015). Constructionist gaming beyond the screen: Middle school students' crafting and computing of touchpads, board games, and controllers. *Proceedings of the workshop in P\primary and secondary computing education* (pp. 49–54).

Kafai, Y. B., Proctor, C., & Lui, D. (2019). Framing computational thinking for computational literacies in K-12 education. In Proceedings of the Weizenbaum conference 2019 “Challenges of Digital Inequality – Digital Education, Digital Work, Digital Life” (pp. 1–6). https://doi.org/10.34669/wi.cp/2.21

Kafai, Y., Proctor, C., & Lui, D. (2020). From theory bias to theory dialogue: embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education. *ACM Inroads*, *11*(1), 44-53.

Kandemir, C. M., Kalelioğlu, F., & Gülbahar, Y. (2021). Pedagogy of teaching introductory text−based programming in terms of computational thinking concepts and practices. *Computer Applications in Engineering Education*, *29*(1), 29-45.

Karsenti, T. and Bugmann, J. (2017). “Exploring the Educational Potential of Minecraft: The Case of 118 Elementary-School Students”. , Proceedings of the International Association for Development of the Information Society (IADIS) International Conference on Educational Technologies.

Kelly, J. E., Krause, S. J., & Baker, D. R. (2012, June). Using systemic functional linguistics to analyze engineering speak in an introductory materials science and engineering course. In *2012 ASEE Annual Conference & Exposition*.

Keiler, L. S. (2018). Teachers’ roles and identities in student-centered classrooms. *International journal of STEM education*, *5*(1), 1-20.

Kilpatrick, J., Swafford, J., & Findell, B. (Eds.). (2001). *Adding + it up: Helping children learn mathematics*. National Academy Press.

Kong, S. C. (2019). Components and methods of evaluating computational thinking for fostering creative problem-solvers in senior primary school education. In Kong SC., & Abelson H. (Eds.), Computational Thinking Education (pp.119-141). Springer.

Kordaki, M., & Kakavas, P. (2017). Digital Storytelling As an Effective Framework for the Development of Computational Thinking Skills. *Edulearn17 Proceedings*, *1*(May), 6325– 6335.

Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, *50*(4), 36-42.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge university press.

Lavid, J., Arús, J., & Zamorano-Mansilla, J. R. (2010). *Systemic functional grammar of Spanish: A contrastive study with English* (Vol. 1). A&C Black.

Lavie, I., & Sfard, A. (2019). How children individualize numerical routines: Elements of a discursive theory in making. *Journal of the Learning Sciences, 28*(4-5), 419-461.

Lavie, I., Steiner, A., & Sfard, A. (2019). Routines we live by: From ritual to exploration. *Educational Studies in Mathematics*, *101*(2), 153-176.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Marlyin-Smith, J,. & Werner, L. (2011). Computational thinking for youth in practice. *Acm Inroads*, *2*(1), 32-37.

Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology*, *29*(1), 1-8.

Lehrer, R., Schauble, L., Strom, D., & Pligge, M. (2001). Similarity of form and substance: Modeling material kind. In S. Carver & D. Klahr (Eds.), Cognition and instruction: Twenty-five years of progress (pp. 39-74). Mahwah, NJ: Lawrence Erlbaum

Lenhard, J., & Carrier, M. (2017). *Mathematics as a Tool: Tracing New Roles of Mathematics in the Sciences* (Vol. 327): Springer.

Lesh, R. and Landau, M. (Eds.) (1983). A*cquisition of mathematics concepts and processes*. Academic Press.

Lesh, R., & Doerr, H. M. (2003). *Beyond constructivism: Models and modelling perspectives on mathematics problem solving, learning, and teaching.* Lawrence Erlbaum.

Lightbown, P. M., & Spada, N. (2013). *How languages are learned 4th edition-Oxford Handbooks for Language Teachers*. Oxford university press.

Liston, J. M. S. (2018). *A Linguistic Perspective: The Meanings Student Teachers of Secondary Mathematics Make in Roles as Discourse Facilitators* (Doctoral dissertation, The University of Arizona).

Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, *30*(4), 883-908.

López-Acosta, L. A., & Rodríguez-Vergara, D. (2021). Análisis sistémico-funcional de textos algebraicos: hacia un entendimiento de su naturaleza discursiva en la historia y algunas implicaciones en su enseñanza. *IE revista de investigación educativa de la REDIECH*, *12*, e1150-e1150.

LópezLeiva, C. A., Pattichis, M. S., & Celedón-Pattichis, S. (2019). Modelling and programming of digital video: A Source for the integration of mathematics, engineering, and technology. In Doig, B., Williams, J., Swason, D., Borromeo Ferri, R., Drake, P., *Interdisciplinary Mathematics Education* (pp. 135-153). Springer International Publishing.

Luck, L., Jackson, D., & Usher, K. (2006). Case study: A bridge across the paradigms. *Nursing Inquiry, 13*(2), 103-109.

Lyon, J. A., & J. Magana, A. (2020). Computational thinking in higher education: A review of the literature. *Computer Applications in Engineering Education*, *28*(5), 1174-1189.

MacLean, L. M. (2017). *Cracking the code: how to get women and minorities into STEM disciplines and why we must*. Momentum Press.

Mahoney, J. L., Parente, M. E., & Zigler, E. F. (2009). Afterschool programs in America: Origins, growth, popularity, and politics. *Journal of Youth Development*, *4*(3), 23-42.

Mahn, H. (2015). Classroom discourse and interaction in the zone of proximal development. In Markee, N., (Ed.). The handbook of classroom discourse and interaction (pp, 250-264). John Wiley & Sons, Inc.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2017). *Stuck in the shallow end: Education, race, and computing*. MIT Press.

Margulieux, L., Dorn, B., Searle, K. (2019). Learning Sciences for Computing Education. In: Fincher, S. and Robins, A. (Eds.), The Cambridge Handbook of Computing Education Research. Cambridge University Press, 208–230.

Marquardson, J., & Gomillion, D. (2018). Cyber security curriculum development: protecting students and institutions while providing hands-on experience. *Information Systems Education Journal*, *16*(5), 12.

Martin, J.R. & Rose, D. (2003). W*orking with discourse: Meaning beyond the clause*. Continuum.

Martin, J.R. and Rose, D. (2007). *Working with discourse: meaning beyond the clause* (2nd edition). Continuum.

Martin, L., & Betser, S. (2020). Learning through making: The development of engineering discourse in an out−of−school maker club. *Journal of Engineering Education*, *109*(2), 194-212.

Maxwell, J. (2013). *Qualitative research design: an interactive approach*. Sage.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with Scratch. *Computer Science Education*, *23*(3), 239–264. doi:10.1080/ 08993408.2013.832022.

Mejia, J. A., Revelo, R. A., Villanueva, I., & Mejia, J. (2018). Critical theoretical frameworks in engineering education: An anti-deficit and liberative approach. *Education Sciences*, *8*(4), 158.

Merino-Armero, J. M., González-Calero, J. A., & Cózar-Gutiérrez, R. (2021). The effect of after-school extracurricular robotic classes on elementary students' computational thinking. *Interactive Learning Environments*, 1-12.

Merriam, S. B., & Tisdell, E. J. (2015). *Qualitative research: A guide to design and implementation*. John Wiley & Sons.

Meyer, J.H.F. and Land, R. (2003). 'Threshold concepts and troublesome knowledge: Linkages to ways of thinking and practising within the disciplines', in Rust, C. (ed.), Improving Student Learning: Improving Student Learning Theory and Practice – Ten Years On. Oxford: Oxford Centre for Staff and Learning Development.

Mills, S. (2016). Conceptual understanding: A concept analysis. *The Qualitative Report*, *21*(3), 546.

Mizuno, J., Barletta, N., & Chamorro, D. (2018). La representación del mundo mediante los procesos del hacer y suceder. En Sara Regina Scotta Cabral & Leila Barbara (Eds.), *Estudos sistêmico-funcionais no âm- bito do projeto* SAL (pp. 92–109).

Moll, L. C, & Whitmore, K. (1993). Vygotsky in educational practice. In E. Forman, N. Minick, & C. A. Stone (Eds.), *Contexts for learning: Sociocultural dynamics in children's development* (pp. 19-42). Oxford.

Moon, J., Do, J., Lee, D., & Choi, G. W. (2020). A conceptual framework for teaching computational thinking in personalized OERs. *Smart Learning Environments*, *7*(1), 1-19.

Morgan, C. (1996). The language of mathematics: Towards a critical analysis of mathematics texts. *For the Learning of Mathematics*, *16*(3), 2-10.

Morgan, C. (2005). Word, definitions, and concepts in discourses of mathematics, teaching and learning. *Language and Education*, *19*(2), 102-116.

Morgan, C. (2006). What does social semiotics have to offer mathematics education research? *Educational Studies in Mathematics*. https://doi.org/10.1007/s10649-006-5477-x.

Morgan, C. (2012). Studying discourse implies studying equity. In *Equity in Discourse for Mathematics Education* (pp. 181-192). Springer.

Morgan, C. (2014a). Social theory in mathematics education: Guest editorial. *Educational Studies in Mathematics*, *87*(2), 123-128.

Morgan, C. (2014b). Understanding practices in mathematics education: Structure and text. *Educational Studies in Mathematics*, *87*(2), 129-143.

Morgan, C. (2016). Studying the role of human agency in school mathematics. *Research in Mathematics Education*, *18*(2), 120-141.

Morgan, C., & Sfard, A. (2016). Investigating changes in high-stakes mathematics examinations: A discursive approach. *Research in Mathematics Education*, *18*(2), 92-119.

Moschkovich, J. (2002). A situated and sociocultural perspective on bilingual mathematics learners. *Mathematical thinking and learning*, *4*(2-3), 189-212.

Mortimer, E., & Scott, P. (2003). *Meaning Making In Secondary Science Classroomsaa*. McGraw-Hill Education.

Moschkovich, J. (2002). A situated and sociocultural perspective on bilingual mathematics learners. *Mathematical thinking and learning*, *4*(2-3), 189-212.

Moschkovich, J. (2007). Examining mathematical discourse practices. *For the Learning of Mathematics*, *27*(1), 24-30.

Moschkovich, J. N. (2012). How equity concerns lead to attention to mathematical discourse. In *Equity in discourse for mathematics education* (pp. 89-105). Springer.

Mouza, C., Pan, Y. C., Yang, H., & Pollock, L. (2020). A multiyear investigation of student computational thinking concepts, practices, and perspectives in an after-school computing program. *Journal of Educational Computing Research*, *58*(5), 1029-1056.

National Council of Teachers of Mathematics (2000). Principles and Standards for School Mathematics Reston Virginia: NCTM.

National Council of Teachers of Mathematics (2017). *Positioning NCTM for a second century*. Retrieved from https://www.nctm.org/secondcentury/.

National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. National Academies Press.

National Research Council. (2011). *Report of a workshop of pedagogical aspects of computational thinking committee for the workshops on computational thinking*. National Academies Press.

Neuron Fuel. (2020). Tynker. https://www.tynker.com/.

Newell, A. (1983) “The heuristic of George Pólya and its relation to artificial intelligence. In R. Groner, M. Groner, & W. Bischof (Eds.), Methods of heuristics (pp. 195–243). Hillsdale, NJ: Lawrence Erlbaum.

NGSS Lead States. (2013). Next Generation Science Standards: for states by states. The National Academies Press.

Noddings, N. (2018). *Philosophy of education*. Routledge.

Nosratian, S., Moradkhani, M., & Tavakoli, M. B. (2021). Hybrid data compression using fuzzy logic and Huffman coding in secure IOT. *Iranian Journal of Fuzzy Systems*, *18*(1), 101-116.

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, *11*(1), 1-17.

Nystrand, M. (1997). *Opening dialogue: Understanding the dynamics of language and learning in the English classroom.* Teachers College Press.

OCDE. (2020). Job Creation and Local Economic Development 2018. Retrieved from: http://www.oecd.org/cfe/job-creation-and-local-economic-development-26174979.htm.

O'Connor, C., & Michaels, S. (2019). Supporting teachers in taking up productive talk moves: The long road to professional learning at scale. *International Journal of Educational Research*, *97*, 166-175.

Ordóñez, J., Brotóns, V. N., & Ron, J. M. S. (2007). *Historia de la ciencia*. Espasa Calpe Mexicana, S.A.

Österman, T., & Bråting, K. (2019). Dewey and mathematical practice: revisiting the distinction between procedural and conceptual knowledge. *Journal of Curriculum Studies*, *51*(4), 457-470.

P. Rose, S., Habgood, M. J., & Jay, T. (2020). Designing a programming game to improve children's procedural abstraction skills in scratch. *Journal of Educational Computing Research*, *58*(7), 1372-1411.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Patrick, W. (2010). *Recognising non-formal and informal learning outcomes, policies and practices: Outcomes, policies and practices* (Vol. 2009, No. 35). OECD publishing.

Pattichis, M. S., Celedón-Pattichis, S., & LópezLeiva, C. A. (2017). Teaching image and video processing using middle school mathematics and the Raspberry Pi. In Special Session on Advances in Signal Processing Education (pp. 6349- 6353). International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

Peng, J., Wang, M., Sampson, D., & van Merriënboer, J. J. (2019). Using a isualization-based and progressive learning environment as a cognitive tool for learning computer programming. *Australasian Journal of Educational Technology*, *35*(2), 52-68.

Perrenet, J., Groote, J. F., & Kaasenbrood, E. (2005). Exploring students' understanding of the concept of algorithm: levels of abstraction. *ACM SIGCSE Bulletin*, *37*(3), 64-68.

Perrenet, J. C., & Kaasenbrood, E. J. S. (2006). Levels of abstraction in students' understanding of the concept of algorithm: the qualitative perspective. *ITiCSE06-Proceedings of the 11th annual SIGCSE conference on innovation and technology in computer science education* (pp. 270-274). ISO 690.

Pimentel, D. S., & McNeill, K. L. (2013). Conducting talk in secondary science classrooms: Investigating instructional moves and teachers' beliefs. *Science Education*, *97*(3), 367-394.

Pimm, D. (1987). *Speaking mathematically: communication in mathematics classrooms.* Routledge and Kegan Paul.

Pinnow, R. J., & Chval, K. B. (2015). "How much You wanna bet?": Examining the role of positioning in the development of L2 learner interactional competencies in the content classroom. *Linguistics and Education*, *30*, 1-11.

Plucker, J. A., Beghetto, R. A., & Dow, G. T. (2004). Why isn't creativity more important to educational psychologists? Potentials, pitfalls, and future directions in creativity research. *Educational Psychologist*, *39*(2), 83-96.

Pollak, M., & Ebner, M. (2019). The missing link to computational thinking. *Future Internet*, *11*(12), 263.

Polya, G. (2004). *How to solve it: A new aspect of mathematical method* (No. 246). Princeton university press.

Priami, C. (2007). Computational thinking in biology. In In: Priami C. (eds) *Transactions on Computational Systems Biology VIII* (pp. 63-76). Springer.

Quiroz, B. (2017). Gramática interpersonal básica del español: una caracterización sistémico-funcional del sistema de MODO. *Lenguas Modernas*, (49), ág-157.

Reeves, S., & Clarke, M. (1990). *Logic for computer science*. Addison-Wesley.

Repenning, A., Basawapatna, A., & Escherle, N. (2016, September). Computational thinking tools. In 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) (pp. 218-222). IEEE.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67.

Rich, K., Strickland, C., & Franklin, D. (2017). A Literature review through the lens of computer science learning goals theorized and explored in research. *Proceedings of the 2017 ACM SIGCSE*.

Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education, 27(2), 165-205.*

Roig-Vila, R., & Moreno-Isac, V. (2020). El pensamiento computacional en Educación. Análisis bibliométrico y temático. *Revista De Educación a Distancia (RED)*, *20*(63), 1-24.

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior*, *72*, 678-691.

Román-González, M., Moreno-León, J., y Robles, G. (2019). Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions. In Computational Thinking Education (79-98). Springer, Singapore. doi: https://doi.org/10.1007/978-981-13- 6528-7_6

Rotman, Brian. (1988). Towards a Semiotics of Mathematics. Semiotica, 72(1-2): 1-35.

Computing Research. 58 (7), *1372–1411.*

Rose, D. (2018). Languages of schooling: Embedding literacy learning with genre-based pedagogy. *European Journal of Applied Linguistics*, *6*(1), 59-89.

Rose, S.P. Habgood, M.P.J., & Jay, T. (2020). Designing a programming game to improve children's procedural abstraction skills in Scratch. *Journal of Educational* Ross, A., & Wilson, V. (2012). The effects of representations, constructivist approaches, and engagement on middle school students' algebraic procedure and conceptual understanding. *School Science and Mathematics*, *12*(2), 117-128.

Rowe, E., Asbell-Clarke, J., & Baker, R. S. (2015). Serious games analytics to measure implicit science learning. In C. S. Loh, Y. Sheng, & D. Ifenthaler (Eds.), *Serious games analytics: Methodologies for performance measurement, assessment, and improvement* (pp. 343-360). Springer International Publishing.

Rowland, T. (2000). *The pragmatics of mathematics education: vagueness in mathematical discourse*. Falmer Press.

Rudolph, J. L. (2020). The lost moral purpose of science education. *Science Education*, *104*(5), 895-906.

Runestone Interactive (2021). Retrieved from https://csawesome.runestone.academy/runestone/books/published/fopp/Iteration/NestedIterationImageProcessing.html. Runestone Academy LTD.

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. *Computational thinking in the STEM disciplines* (pp. 151-164). Springer, Cham.

Santo, R., Vogel, S., Ryoo, J., Denner, J., Belgrave, C., Moriss, A., & Tirado, A. (2020, February). Who has a seat at the table in CSed? Rethinking equity through the lens of decision-making and power in computer science education initiatives. *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 329-330).

Sarfraz, I., Rajendran, D., Hewege, C., & Mohan, M. D. (2018). An exploration of global employability skills: A systematic research review. *International Journal of Work Organisation and Emotion*, *9*(1), 63-88.

Schleppegrell, M. (2004). *The language of schooling*. Lawrence Erlbaum.

Schleppegrell, M. J. (2012). Linguistic tools for exploring issues of equity. In HerbelEisenmann, B., Choppin, J., Wagner D., & Pimm D., (Eds.), Equity in discourse for Mathematics Education: Theories, practices, and policies (pp. 109-124). Springer.

Schleppegrell, M.J. (2017). Systemic Functional Grammar in the K-12 classroom. In Eli Hinkel (Ed.), *Handbook of research in second language teaching and learning V. 3* (pp. 384- 396). Routledge.

Schuster, D., Cobern, W.W., Adams, B.A., Undreiu, A. & Pleasants, B. (2018). Learning of core disciplinary ideas: Efficacy comparison of two contrasting modes of science instruction. *Research in Science Education, 48*(2):389-435. https://doi.org/10.1007/s11165- 016-9573-3.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, *18*(2), 351-380.

Sarfraz, I., Rajendran, D., Hewege, C., & Mohan, M. D. (2018). An exploration of global employability skills: a systematic research review. *International Journal of Work Organisation and Emotion*, *9*(1), 63-88.

Sentance, S., & Csizmadia, A. (2015, July). Teachers' perspectives on successful strategies for teaching Computing in school. In *IFIP TC3 Working Conference 2015: A New Culture of Learning: Computing and Next Generations*.

Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, *29*(2-3), 136-176.

Sfard, A. (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin. *Educational Studies in Mathematics*, *22*(1), 1-36.

Sfard, A. (1998). On two metaphors for learning and the dangers of choosing just one. *Educational Researcher*, *27*(2), 4-13.

Sfard, A. (2000). Steering (dis) course between metaphors and rigor: Using focal analysis to investigate an emergence of mathematical objects. *Journal for Research in Mathematics Education*, *31*(3), 296-327.

Sfard, A. (2001a). There is more to discourse than meets the ears: Looking at thinking as communicating to learn more about mathematical learning. *Educational Studies in Mathematics*, *46*(1), 13-57.

Sfard, A. (2001b). Learning mathematics as developing a discourse. *Proceedings of 21st Conference of PME-NA* (pp. 23-44). https://www.researchgate.net/profile/Anna-Sfard/publication/252619091_Learning_mathematics_as_developing_a_discourse/links/0f31753c8e5cc02dbd000000/Learning-mathematics-as-developing-a-discourse.pdf

Sfard, A., & Lavie, I. (2005). Why cannot children see as the same what grown-ups cannot see as different?—Early numerical thinking revisited. *Cognition and Instruction*, *23*(2), 237-309.

Sfard, A. (2008). *Thinking as communicating: Human development, the growth of discourses, and mathematizing*. Cambridge University Press.

Shaffer, D.W. (2006). Epistemic frames for epistemic games. *Computers & Education*, *46*(3):223-234. https://doi.org/10.1016/j.compedu.2005.11.003.

Shier H. 2001. Pathways to participation: openings, opportunities and obligations. *Children and Society, 15*(2), 107–117.

Shreyar, S., Zolkower, B., & Pérez, S. (2010). Thinking aloud together: A teacher's semiotic mediation of a whole-class conversation about percents. *Educational Studies in Mathematics*, *73*(1), 21-53.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142-158.

Sinclair, J., & Coulthard, M. (1975). *Towards an analysis of discourse: The English used by teachers and pupils*. Oxford University Press.

Sjöberg, C., Risberg, T., Nouri, J., Norén, E., & Zhang, L. (2019< March 11-13). A lesson study on programming as an instrument to learn mathematics and social science in primary school. *13th International Technology, Education and Development Conference* (pp. 2230-2235). Valencia, Spain. The International Academy of Technology, Education and Development.

Skemp, R. R.: (1976). Relational understanding and instrumental understanding. *Mathematics Teacher*, *77*(1), 20-26.

Skemp, R. R. (1987). The Psychology of Learning mathematics (Expanded American Ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Snalune, P. (2015). The benefits of computational thinking. *ITNOW*, *57*(4), 58-59.

So, H. J., Jong, M. S. Y., & Liu, C. C. (2020). Computational thinking education in the Asian Pacific region. *The Asia-Pacific Education Researcher*, *29*(1), 1-8.

Statter, M. (2016). Teaching abstract thinking in introduction to computer science for 7th graders. *Proceedings of the 11th workshop in primary and secondary computing education—WiPSCE '16* (pp. 80–83).

Statter, D., & Armoni, M. (2020). Teaching abstraction in computer science to 7th grade students. *ACM Transactions on Computing Education (TOCE)*, *20*(1), 1-37.

Tall, E. & Thomas, T. (Ed.) (2002). Intelligence, learning and understanding in mathematics: A Tribute to Richard Skemp. Post Pressed.

Tang, K.S., Degado, C. & Moje, E. (2014). An integrative framework for the analysis of multiple and multimodal representations for meaning-making in science education. *Science Education*, *98,* 305-326. https://doi.org/10.1002/sce.21099.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers and Education, 148*, 1-22. https://doi.org/10.1016/j.compedu.2019.103798

Taslibeyaz, E., Kursun, E., & Karaman, S. (2020). How to develop computational thinking: A Systematic review of empirical studies. *Informatics in Education*, *19*(4), 701-719.).

Taub, R., Armoni, M. & Ben-Ari, M. (2014). Abstraction as a bridging concept between computer science and physics. *Proceedings of the 9th workshop in primary and secondary computing Education* (pp. 16–19). ACM.

Royal Society. 2012. Shutdown or restart? The way forward for computing in UK schools. http://royalsociety.org/education/policy/computing-in-schools/report/.

Tissenbaum, M., Sheldon, J., & Abelson, H. (2019). From computational thinking to computational action. *Communications of the ACM*, *62*(3), 34-36.

Tolk, A., Koehler, M. T., & Norman, M. D. (2018, July). Epistemological constraints when evaluating ontological emergence with computational complex adaptive systems. *International Conference on Complex Systems* (pp. 1-10). Springer, Cham.

Tomasello, M. (2003). The key is social cognition. In D. Gentner & S. Kuczaj (Eds.), Language and thought (pp. 47– 58). Cambridge, MA: MIT Press.

Tomasello, M., & Rakoczy, H. (2003). What makes human cognition unique? From individual to shared to collective intentionality. *Mind & Language*, *18*(2), 121-147.

Tomasello, M. (2010). *Origins of human communication*. MIT Press.

Tomasello, M. (2014). The ultra−social animal. *European journal of social psychology*, *44*(3), 187-194.

Tracy, S. J. (2019). *Qualitative research methods: Collecting evidence, crafting analysis, communicating impact*. John Wiley & Sons.

Turner, E., Dominguez, H., Maldonado, L., & Empson, S. (2013). English learners' participation in mathematical discussion: Shifting positionings and dynamic identities. *Journal for Research in Mathematics Education*, *44*(1), 199-234.

UNESCO (2021). Multilingualism: a key to inclusive education. Retrieved on 08/01/2021 from http://www.unesco.org/new/en/education/themes/strengthening-education-systems/languages-in-education/single-view/news/multilingualism_a_key_to_inclusive_education/.

Usanov, F. (2020). The eight ways to advance pedagogy to the next level. *Mental Enlightenment Scientific-Methodological Journal*, *2020*(1), 181-190.

Vähäsantanen, K., & Eteläpelto, A. (2018). Agency and Learning in the Work of Software Professionals. In *The Impact of Digitalization in the Workplace* (pp. 161-179). Springer International Publishing.

Vakil, S. (2018). Ethics, identity, and political vision: Toward a justice-centered approach to equity in computer science education. *Harvard Educational Review*, *88*(1), 26-52.

Veel, R. (1999). Language, knowledge and authority in school mathematics. In F. Christie (Ed.), Pedagogy and the shaping of consciousness: Linguistic and social process (pp. 185–216). London: Continuum.

Vogel, S., Hoadley, C., Ascenzi-Moreno, L., & Menken, K. (2019, February). The Role of translanguaging in computational literacies: Documenting middle school bilinguals' practices in computer science integrated units. *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 1164-1170).

Vogel, S., Hoadley, C., Castillo, A. R., & Ascenzi-Moreno, L. (2020). Languages, literacies, and literate programming: Can we use the latest theories on how bilingual people learn to help us teach computational literacies? *Computer Science Education*, *30*(4), 420-443.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715-728.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.

Vygotsky, L. S. (1986). *Thought and language*. MIT Press.

Vygotsky, L. S. (1994). Imagination and creativity of the adolescent. In R. Van Der Veer, & J. Valsiner (Eds.), *The Vygotsky Reader*.Blackwell.37.

Vygotsky, L. S. (1996). Psicología infantil. Obras escogidas, vol. 4. Visor.

Vygotsky, L.S. (1998) Child Psychology. The Collected Works of L.S. Vygotsky: Vol. 5. Problems of the Theory and History of Psychology. Plenum.

Wagner, D., & Herbel-Eisenmann, B. (2014). Identifying authority structures in mathematics classroom discourse: A case of a teacher's early experience in a new context. *International Journal on Mathematics Education* 46(6), 871-882.

Waite, J. (2017). *Pedagogy in teaching computer science in schools: A literature review.* Royal Society.

Waite, J. L., Curzon, P., Marsh, W., Sentance, S., & Hadwen-Bennett, A. (2018). Abstraction in action: K-5 teachers'nuses of levels of abstraction, particularly the design level, in teaching programming. *International Journal of Computer Science Education in Schools*, *2*(1), 14-40.

Waite, J., Curzon, P., Marsh, W., & Sentance, S. (2016, October). Abstraction and common classroom activities. *Proceedings of the 11th workshop in primary and secondary computing education* (pp. 112-113).

Weintrop, D., & Wilensky, U. (2013). RoboBuilder: A Computational thinking game. *Sigcse*, *736*. https://doi.org/10.1145/2445196.2445430.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127-147.

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. *ACM transactions on computing education* (TOCE), *18*(1), 1-25.

Weintrop, D., Wise Rutstein, D., Bienkowski, M., & McGee, S. (2021). Assessing computational thinking: an overview of the field. *Computer Science Education*, *31*(2), 113-116.

Wen, W., Kuroki, Y., & Asama, H. (2019). The Sense of agency in driving automation. *Frontiers in Psychology*, *10(2691), 1-12.* https://www.researchgate.net/publication/337709684_The_Sense_of_Agency_in_Driving_Automation

Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity.* Cambridge University Press.

Wenger, E. (2010). Communities of practice and social learning systems: The career of a concept. In C. Blackmore (Ed.), *Social learning systems and communities of practice* (pp. 179–198). London: Springer Verlag and the Open University.

Wenger-Trayner, E. & Wenger-Trayner, B. (2015). Communities of practice: A brief introduction. Retrieved from https://goo.gl/y1g6HU

Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The Fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings* *of the 43rd ACM Technical Symposium on Computer Science Education* (SIGCSE '12), 215-220. ACM.

Werner, L., Denner, J., Campe, S., Ortiz, E., DeLay, D., Hartl, A., & Laursen, B. (2013). Pair programming for middle school students: Does friendship influence academic outcomes? *Proceeding of the 44th ACM technical symposium on computer science education* (pp. 421–426). ACM.

Werner, L., Denner, J., & Campe, S. (2015). Children programming games: a strategy for measuring computational learning. *ACM Transactions on Computing Education (TOCE)*, *14*(4), 24.

Wertsch (1985). *Vygotsky and the social formation of mind*. Harvard University Press.

Wertsch J. (2007). Mediation. In: H. Daniels, M. Cole & J. Wertsch (Eds.), *The Cambridge companion to Vygotsky*. Cambridge University Press, pp. 178-192. https://doi.org/10.1017/CCOL0521831040.008.

Wertsch, J. V. (1998). *Mind as action.* Oxford University Press.

Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. In J. Clayson & I. Kallas (Eds.), *Proceedings of the constructionism 2010 conference.*

Wing, J. M. (2006). Computational thinking. *Communications of The ACM*, *49*(3), 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical transactions of the Royal Society A. *Mathematical, Physical and Engineering Sciences*, *366(*1881), 3717-3725.

Wood, D. J., Bruner, J. S. and Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychiatry and Psychology,17*(2), 89-100.

Wu, B., Hu, Y., Ruis, A. R., & Wang, M. (2019). Analysing computational thinking in collaborative programming: A quantitative ethnography approach. *Journal of Computer Assisted Learning*, *35*(3), 421-434.

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, *28*(4), 371-400.

Yin, R. K., & Davis, D. (2007). Adding new dimensions to case study evaluations: The case of evaluating comprehensive reforms. *New Directions for Evaluation*, 2007(113), 75–93.

Zentella, A. C. (1997). Latino youth at home, in their communities, and in school: The language link. *Education and Urban Society*, *30*(1), 122-130.