COMPUTER SCIENCE EDUCATION WITH ENGLISH LEARNERS

BY

SAMANTHA M. ARMENTI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE AND STATISTICS

UNIVERSITY OF RHODE ISLAND

2018

ProQuest Number: 10843130

ProQuest 10843130

www.manaraa.com

MASTER OF SCIENCE THESIS

OF

SAMANTHA M. ARMENTI

APPROVED:

Thesis Committee:

Major Professor   Victor Fay-Wolfe

Joan Peckham

Amy Correia

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2018

## ABSTRACT

The Computer Science For All initiative intends to teach computational thinking and programming skills to all levels of students with varying background and experience in computer science. This includes English Learner (EL) students. There should be a concentrated effort to narrow the achievement gap among EL students and non-EL students especially in computer science courses that can lead to economic prosperity and social mobility in the digital society. With the growing number of EL students joining the student population in public schools across the United States, it is critical that these students are provided with equitable access to computer science education through effective instructional strategies and meaningful resources. This research study intends to develop EL accessible curriculum and resources for an introductory computer science high school course and use research-based designs and strategies to evaluate the quality of the developed EL accessible materials, instructions, and assessments in computer science learning and achievement among EL students. As a result of this study, all of the developed EL accessible instructions, materials, and assessments are deemed satisfactory; however, there remain necessary revisions of these materials to be implemented in order to fully address the needs and concerns of EL students in computer science learning and achievement.

# ACKNOWLEDGMENTS

I would like to thank my major professor, Victor Fay-Wolfe, for his guidance, support, and encouragement throughout my graduate career. I am incredibly grateful for the amazing opportunities he has given me that allowed me to grow as an academic professional and discover my passion for computer science education. He has opened so many doors for me and I am excited to see where the next one leads. I am honored that I am able to continue my research under his direction and mentorship.

In addition to my major professor, I would also like to thank my committee members, Joan Peckham and Amy Correia. Throughout this process, they have both been incredibly helpful and have never once hesitated to assist me with any questions and concerns that I had. It was amazing to work these strong and intellectual women who share the same passion for equitable learning opportunities in computer science and had great interest in my thesis topic.

I would also like thank my K-12 staff co-workers including Jessie Barrett, Howard Dooley, Beth Very, Martha Bellows, Michael Conti, and Gabriel DePace. These are some of the best co-workers anyone could ask for. All of them are always willing to lend a helping hand and incredibly supportive. Their passion for computer science education made the job fun and exciting. Although we only met a couple years ago, it feels as if we are a small close-knit family.

Additionally, I would like to thank all the participating teachers that I had the chance to work with. These men and women are leading and shaping the next generation of youth into a new technological era. It was a great honor to work with them and share my enthusiasm for computer science with them and their students.

Throughout this journey in graduate school, I never expected to meet such a strong and independent woman who I would have develop an immediate connec-

tion with. I am not only walking away from this graduate experience with new knowledge and skills but also with a best friend. I would like to thank Martha Bellows for helping make graduate school enjoyable. We shared so many laughs on this journey and I cannot wait to share more in the future.

Lastly, but most importantly, I would like to thank my mother for always encouraging me to follow my dreams and supporting in so many ways that I cannot even begin to describe. She is my role model. She is my inspiration. She pushes me to overcome challenges and strive to be the best possible me that I can be. Without her, none of this would be possible.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

**Table**                                                                                    **Page**

# CHAPTER 1

## Introduction

### 1.1 Background Information

As our society continues to advance technologically, it is becoming increasingly apparent that computer science will be an essential skill for all citizens who desire social and economic growth. According to Smith, more than six hundred thousand high-income technical jobs in the United States were vacant in 2015 [1, 2]. As shown in the Figure 1, there were more than half a million vacant positions in information technology in comparison to other major occupations that pay significantly less in terms of the average salary [2, 3]. By 2018, Smith estimates that over fifty percent of STEM (Science Technology Engineering Mathematics) jobs will be in computer science related fields [1, 3]. Yet, computer and data science skills are necessary to all types of industries that may not be considered technological occupations such as healthcare, education, energy, transportation, local and federal government, financial, retail, manufacturing, and more [1, 2]. It is increasingly apparent that there is a significant demand to develop a technologically educated and inventive labor force within the United States to fill these vacant positions and lead the world in innovation.

The establishment of this technical labor force begins with educating the students within the United States public education system. Unfortunately, significant discrepancies have been observed in regards to the accessibility of computer science learning opportunities to all types of students. For instance, only forty-percent of United States principals in 2015 reported teaching a computer programming course [4, 3, 5]. Additionally, only sixteen states nationwide developed K-12 computer science standards while only thirty-five states nationwide allow computer science to count towards high school graduation credit [4, 3]. Without the opportunity

1
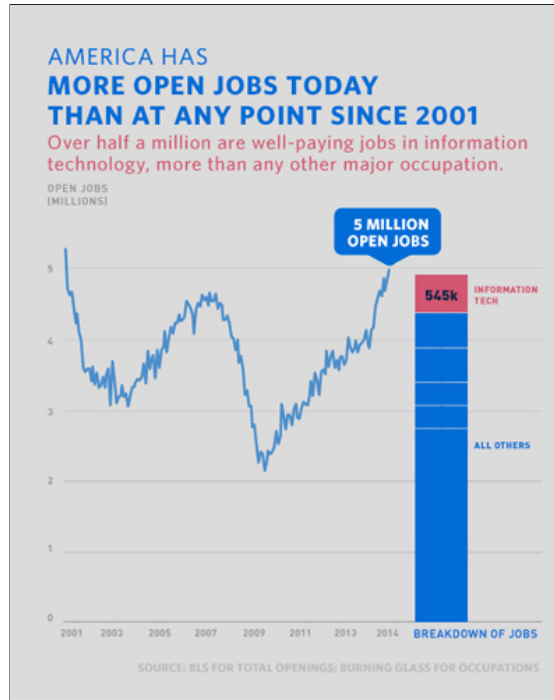
Figure 1: This graphic demonstrates that more than a half a million positions in information technology are left unfilled as of 2015. The data originates from reliable sources such as the U.S. Bureau of Labor Statistics (BLS) and labor analytics firm Burning Glass Technologies [2].

for exposure to computer science, students fail to even have a chance to learn and foster the fundamental skills in computer science such as computational and algorithmic thinking, aptitude in solving complex and challenging problems, and utilization of creativity and innovation.

There is also a severe lack of diversity in computer science courses that are currently being offered at all levels of education. In 2015, the College Board reported that of all the students that participated in their advanced placement exam for computer science (AP CS A exam) females and underrepresented minorities comprised only twenty-two and thirteen percent, respectively [4, 3]. The National Center for Education Statistics reported that of the students graduating with computer science bachelor degrees in 2015 females and underrepresented minorities both groups comprised only eighteen percent each [4, 3]. According to

the Bureau of Labor Statistics Current Population Survey, females and underrepresented minorities comprised twenty-four percent and fifteen percent of people employed in computing occupations in 2017 [4, 3]. The lack of diversity in the current technical workforce may largely be attributed to the lack of accessibility and exposure of computer science in K-12 schools. Since these students are not introduced to computer science in their primary and secondary education, they are not able to foster an interest in it and as a result of this they do not pursue further education and careers in computing fields. This not only affects the dwindling numbers to fill the vacant positions in technical fields but it can also have detrimental effects on the lack of diverse perspectives being incorporated into developments in technological devices, web and mobile applications, software, and more. Without the unique perspectives and experiences of diverse groups, there can be severe consequences as only one viewpoint dominates the approach to design and development of future technology.

Former President Barack Obama recognized the need for computer science in an effort to respond to the growing demands of a diverse and technical workforce in the United States. He introduced his federal initiative, Computer Science for All (CS4All), calling for computer science and computational thinking skills to be taught in all K-12 schools in the United States [1]. Through this initiative, states and school districts were provided with substantial funding to expand K-12 computer science through teacher trainings, expansion of accessibility for quality computer science instructional materials, and establishment of effective regional partnerships [1]. Additionally, local government representatives, educational leaders, and professional technical organizations were highly encouraged to become involved in the initiative to expand computer science opportunities for students in the K-12 schools.

With the enactment of the CS4All initiative, students at all levels in the United States public educational system should have equitable access to learning opportunities that expand their growth and achievement in computer science. Although this initiative has significantly improved the accessibility of computer science within schools especially to females and underrepresented minorities, there remains several obstacles to reaching a largely marginalized group of students—English Learners (EL). EL students comprise those who do not communicate fluently and/or learn effectively in English which can create difficult barriers for them to overcome and impede their progress in computer science learning. There is a significant lack of research and intervention in developing effective computer science materials for EL students causing a crucial impediment to the CS4All goals. It is important to merit investigative attention into this research as there is a need for integration of EL students into computer science classrooms as well as examine growth (learning) of computer science and computer science achievement.

## 1.2   Rationale for Research

El students require the proper resources and teaching instruction to learn fundamental computational and programming skills that will be essential in this advancing technological workforce. There should be more focus on preparing EL students to function in a technological and data-driven society [6]. The relevance of this becomes more apparent when analyzing the number of EL students entering the public the schools in the United States. According to a study conducted by the National Center for Education Statistics, there has been a steady increase of EL students in public schools throughout the United States since 2003 [7]. The number of EL students was higher in the 2013-2014 academic school year with "9.3 percent ... an estimated 4.5 million students" than in both 2003-2014 and 2012-2013 academic school years with "8.8 percent ... an estimated 4.2 million students

and 9.2 percent ... an estimated 4.4 million students", respectively [7]. Specifically in Rhode Island (RI), EL students made of 7% of total students (10,281) in the 2014-2015 academic school year with 87% enrolled in subsidized lunch programs and 75% living in Central Falls, Pawtucket, Providence and Woonsocket [8]. On a national and state level, there are significant achievement gaps among EL students and non-EL students who confront challenges that can affect their academic success such as language barriers, poverty, limited access to health care, and low parental education levels [7]. Additionally, EL students are more likely to attend schools that are "under-resourced, urban, large, serve high proportions of minority students, and are located in high-poverty communities" [7].

The CS For All initiative intends to teach computational thinking and programming skills to all levels of students with varying background and experience in computer science. This includes EL students. There should be a concentrated effort to narrow the achievement gap among EL students and non-EL students especially in computer science courses that can lead to economic prosperity and social mobility in the digital society. With the growing number of EL students joining the student population in public schools across the United States, it is critical that these students are provided with equitable access to computer science education through effective instructional strategies and meaningful resources.

### 1.3   Scope of Study

The scope of this study entails developing EL accessible curriculum and resources for an introductory computer science high school course and then utilizing a panel of experts to evaluate the quality of these materials. To define this scope of this study more thoroughly, there are several components of the study that have been identified including goals, tasks, and resources.

### 1.3.1 Goals

The goals of this study are to create quality and effective EL accessible materials to be incorporated into future enactments of the high school course. These quality materials will ideally address the needs and concerns of EL students as well as make a significant impact on their learning and achievement in computer science especially in regards to computational thinking and programming.

### 1.3.2 Tasks

This study consists of several tasks that need to be completed especially in terms of developing the EL accessible material as shown in Figure 2. There are five lessons that will be the subject of focus in the first major unit of the course. These five lessons require significant adaptations in terms of the lesson plan, student activity documents, and vocabulary.

| Introduction to Computing and Data Science: EL Accessible Materials | | | | |
|---|---|---|---|---|
| **UNIT** | **LESSONS TO ADAPT** | **ADAPTATIONS TO LESSON PLANS** | **DIFFERENTIATED VERSION OF STUDENT DOCUMENTS** | **VOCABULARY RESOURCES** |
| Programming and Simulation | (1) Introduction to Computing and Data Science | Develop content and language objectives | Introduction To NetLogo Student Activity Guide | Vocabulary Guide |
| | (2) Modeling and Simulation | Develop differentiated version of readings for students (less complex wording/vocabulary, dictionary links to vocabulary, visuals) | Modeling And Simulation Student Activity Guide | Vocabulary Cards |
| | (3) Agent Commands | Provide instructions on videos being presented (pausing and explaining, replacement video, vocabulary difficulty) | Party Simulation Student Activity Guide | Vocabulary Slideshow With Audio |
| | (4) Programming Procedures | Provide instructions on activity (explanation, demonstration/modeling, modifications) | Agent Commands Student Activity Guide | Vocabulary Slideshow |
| | (5) Program Your Own Simulation | | Programming Procedures Student Activity Guide | Vocabulary Wordsearch |
| | | | Variables Student Activity Guide | Vocabulary Crossword |
| | | | Monitors, Data, And Plotting Student Activity Guide | Vocabulary Word Wall |
| | | | Time And Parameters Student Activity Guide | Vocabulary Quizziz |
| | | | Program A Mouse Trap Simulation Student Activity Guide | |
| | | | Collecting Data In NetLogo Student Activity Guide | |

Figure 2: This graphic shows the list of tasks that need to be completed for each lesson in Unit One: Programming and Simulation of the Introduction to Computing and Data Science course.

The lesson plans require modifications in terms of developing language objectives to separate the content and language goals of the lesson, differentiated

versions of the required readings, and detailed instructions for the teacher when presenting videos to the students and explaining/demonstrating the student activities. Each lesson also contains student documents to be completed while participating in the activities. These student documents require differentiated versions to be used by EL students in which complex wording and sentences were reduced and more visualizations were used. Additionally, these differentiated student documents include links to an online dictionary for difficult vocabulary that some EL students may not be familiar with. The final materials that require development are the vocabulary resources to be used by the EL students including a vocabulary guide, cards, slideshow with audio pronunciation of the words, word search/crossword, word wall, and Quizziz (an online multiplayer quiz game) [9]. Further details on the adaptations to the curriculum will be provided in chapter 3.

After completing the tasks concerning the development of EL accessible materials for the course, the remaining tasks consisted of providing these materials to a panel of experts to evaluate. This panel of experts evaluated and provided feedback on the quality and level of effectiveness of the developed EL accessible materials, instructions, and assessments. Their responses were collected and analyzed to determine the caliber of the materials. Significant findings and trends were determined and noted after analysis of collected data to implement these revisions in the materials which were administered to high school teachers of the course in the 2017/2018 academic year.

### 1.3.3  Resources

The resources required for this study include the course curriculum, online tools, simulation web service, and participation of the experts. In terms of the course curriculum, the lesson plans, student activities and practical assessments

were used in the development of the EL accessible materials. Specifically, the content of these lessons and activities served as the foundation to developing differentiated versions of student documents and vocabulary resources. Additionally, the development of these materials relied on evidence-based research practices and strategies discussed in detail in chapter 2. The online tools consisted of several Google Products such as Documents, Spreadsheet, Slides, and YouTube which were used to create, adapt, and distribute the materials for the EL students. Another tool to be used in this study is the simulation web service used to introduce students to programming and simulation in the course. This tool will be discussed further in chapters 2 and 3.

## 1.4   Research Aims

The purpose of this study is to create EL accessible materials, instructions, and assessments for the Intro to Computing and Data Science (ICDS) curriculum to support computer science and evaluate the quality of these computer science materials for EL students. This study produced computer science materials for EL students with significant impact on their computer science learning and achievement and these materials will ideally be incorporated into the ICDS curriculum in the future.

Further details on the methodology of this research including research questions, hypotheses, and methods and procedures are discussed in chapter 3.

## 1.5   Outline of chapters

The remaining chapters discuss several significant components of this research study including the literature review, methodology, data collection findings, and conclusions. Specifically, chapter 2 will analyze the recent literature examining current policies and efforts to implement computer science into K-12 schools, com-

mon learning barriers that EL students confront, current issues in science, math, and content area instruction for EL students, and effective EL instructional strategies and practices. Chapter 3 will detail the methodology of the research including research methods, questions, hypotheses, participants, and procedures. In chapter 4, the collected data from the panel of experts will be analyzed for findings and trends as well as necessary revisions to the materials to meet quality and effectiveness standards defined by the experts. Chapter 5 is the final chapter discussing conclusions and the expansion of this research in future studies.

## List of References

[1] M. Smith. "Computer science for all." Jan. 2016. [Online]. Available: https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

[2] National Archives and Records Administration. "Tech hire." n.d. [Online]. Available: https://obamawhitehouse.archives.gov/issues/technology/techhire

[3] Code.org. "Code.org infographic source data." n.d. [Online]. Available: https://docs.google.com/document/d/1gySkItxiJn_vwb8HIIKNXqen184mRtzDX12cux0ZgZk/pub

[4] Code.org. "Promote computer science." n.d. [Online]. Available: https://code.org/promote

[5] Google and Gallup. "Searching for computer science: Access and barriers in u.s. k-12 education." 2015. [Online]. Available: https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf

[6] M. Warschauer and D. Healey, "Computers and language learning: An overview," *Language teaching*, vol. 31, no. 2, pp. 57–71, 1998.

[7] English Language Learners in Public Schools. "The condition of education." Mar. 2017. [Online]. Available: https://nces.ed.gov/programs/coe/indicator_cgf.aspf

[8] Rhode Island KIDS COUNT. "2016 rhode island kids count factbook." 2016. [Online]. Available: http://www.rikidscount.org/Portals/0/Uploads/Documents/Factbook%202016/2016%20Rhode%20Island%20KIDS%20COUNT%20Factbook.pdf

[9] Quizizz. "Quizizz: Fun multiplayer classroom quizzes." 2018. [Online]. Available: https://quizizz.com/

9

# CHAPTER 2

## Literature Review

There is currently limited literature available on supporting EL students in computer science especially concerning instructional strategies that can improve and/or impede their computer science learning and achievement. Several factors may be responsible for the lack of support for EL students in computer science including the lack of access to computer science in K-12 schools, lack of diversity in the computer science field, and unfavorable stereotypes associated with computer scientists.

This literature will first focus on the national and Rhode Island statewide K-12 computer science initiatives that led to the development of the introductory computer science high school course currently being implemented in more than twenty Rhode Island high schools. This course is the one that had a great need for EL accessible materials and it will be used in the development of EL accessible curriculum and resources. Additionally, this literature review will concentrate on the backgrounds and experiences of EL students as well as contemporary theories and evidence-based strategies to instruct EL students in STEM content areas.

### 2.1   National Computer Science Initiative

The development of K-12 computer science education began in 2006 with the National Science Foundation (NSF) Broadening Participation in Computing (BPC) program [1]. Its objective was to widen participation in computer science education with an emphasis on underrepresented groups [1]. It produced successful results by supporting National Center for Women and Information Technology (NCWIT), which leads in involving women in computing at all levels of education [2]. Further endeavors in computer science education were made with the Com-

puting Education for the 21st Century (CE21) program focusing on increasing the number and diversity of K-14 computer science students and teachers resulting in postsecondary students achieving success in computer science [3]. CE21 lead to the development of two high school computer science courses and NSF using CE21 success to improve computer science high school education [3]. NSF continued their computer science education initiative in 2015 with the STEM+C program by integrating computing into STEM subjects and continuing development of computer science education [4]. All these programs and developments in computer science education led to the CS4All initiative furthering the K-12 computer science development and scaling it to a national level.

## 2.2 Rhode Island Computer Science Initiative

The University of Rhode Island (URI) received a STEM-C award in 2014 to introduce computer science into K-12 schools of Rhode Island. This program produced an AP Computer Science Principles (AP CSP) course with free course materials and teacher professional development (PD) offered in twenty-five Rhode Island high schools in the 2017-2018 academic year [5]. In 2015, URI was awarded an NSF STEM+C grant allowing large-scale integration of computational thinking skills into Next Generation Science Standards (NGSS) course material [6]. This program led to the development of a computing skills class, Introduction to Computing and Data Science (ICDS), to teach students computer science skills necessary in NGSS science modules and STEM subjects [7].

## 2.3 The ICDS Course and Its Development

This introductory high school course was developed by faculty and staff of the Computer Science and Statistics Department at the University of Rhode Island to meet objectives of currently funded NSF grants. ICDS is a 9th/10th grade course

11

developed to teach all students computational thinking (algorithms, data science, programming, simulation) skills and concepts as a basis for use in other subjects [7]. ICDS teaches data collection through searching the Internet and collecting data in forms, performing data analysis, and creating data visualizations. It first teaches inference modeling and programming using the simulation environment NetLogo Web and then teaches the data science skills with spreadsheets [8, 9].

The ICDS course was piloted by a teacher cohort who received training in 2016 through in-person and online professional developments. In this initial pilot of the course, this teacher cohort enacted the materials and provided valuable feedback through observational site visits and follow-up meetings. In its conception, the ICDS course included limited content with simulations and concentrated more so on data and Python programming. Criticisms received from this teacher cohort included the time constraints, difficulty, and lack of engagement among students with the materials and structure of the ICDS course at that point in time. This feedback also presented issues concerning the lack of detailed teacher lesson plans, student activity guides, and EL accessible materials. As such, it was my responsibility to completely revamp the ICDS course to include more structured lessons plans, activities, and resources for the teachers to use before the 2017/2018 academic school year. I first redesigned the curriculum to include an additional unit on NetLogo web simulation programming and more creative activities such as graphic and web design. Additionally, I created detailed teacher lesson plans with student learning objectives and agendas to introduce concepts/activities and conclude with discussions emphasizing the main ideas. I also created scaffolded student activity guides for each lesson with step-by-step instructions and visual examples. Additionally, more focus was put on the simulation and model programming in the first unit to address concerns of engagement and difficulty while

12

the Python programming material became an optional resource for teachers to complete with their students. This updated ICDS curriculum was completed and implemented through the new teacher cohort receiving training in 2017 for the ICDS certification.

As of the 2017/2018 academic year, there are more than twenty schools in Rhode Island enacting the ICDS course by certified teachers. The teachers and students are much more receptive to this updated curriculum but continue to provide valuable feedback regarding improvements that can still be made to the course. Although this updated ICDS curriculum is well-received, there are significant issues concerning the lack of of EL accessible instruction, materials, and assessments in the course resulting in the development of this study.

## 2.4 Transitioning ELs into Mainstream Classrooms

The Every Student Succeeds Act (ESSA) mandates that states standardize statewide entrance and exit criteria for classification of EL students [10, 11, 12]. It defines reclassified EL students as those original EL students who no longer need language instruction due to achieving English proficiency. ESSA requires monitoring of reclassified students for four years after exiting EL services [10, 11, 12]. RI annually assesses English language proficiency among EL students using the World-Class Instructional Design and 123 Assessment Consortium (WIDA) standards of speaking, listening, reading, and writing as its proficiency standards [13]. An EL student in a RI school district can be exited from the specialized EL program and into mainstream classrooms after passing the state's annual English language proficiency assessment, called WIDA ACCESS [13]. All EL students are also allowed to participate in general education courses, which currently includes computer science, with the caveat that the teachers of the content area are EL certified or collaborating with EL certified teachers [13]. This proposed research

only addresses reclassified EL students.

## 2.5 Contemporary Theoretical Framework for EL Learning

This research uses multiple contemporary theories that provide the foundation to the WIDA standards framework for connecting language learning and state academic standards, and for supporting equitable education for EL students [14, 15]. The WIDA standards framework is based upon theories including functional model of language, second language acquisition, and sociocultural theories to promote effective strategies for language development [16]. A functional model of language allows EL students to engage in meaningful learning by concentrating on the types of language necessary to participate in instructional activities [16]. This functional approach to language can be useful in content areas that include unique language especially when clearly defining the language functions as this helps define the tasks that students are expected to perform within the content area and results in meaningful learning [16]. In addition to this approach, it is significant to include sociocultural theories in which meaningful learning can be applied through socially and culturally relatable context to the students as well as expand opportunities to develop language skills through oral and written activities [16].

## 2.6 Reading Level Performance of EL Students

Since most computer science curricular materials assume reading levels at their grade range, for example grades 6-10 for CS Discoveries, the fact that many EL students are not at the reading level for their grade can profoundly impact their ability to learn computer science from those materials. The National Assessment of Educational Progress (NAEP) provides data regarding reading level performance of students in grades 4, 8, and 12 across the nation, and has been doing so since 1998 [17]. Its data shows large achievement gaps in reading level performance

14

between non-EL and EL students where non-EL students score significantly higher than EL students [17]. Rhode Island demonstrates similar achievement gaps in reading performance between non-EL and EL students. In the 2015 Partnership for the Assessment of Readiness for College and Careers (PARCC), only 4% of RI EL students at the 7th grade level met proficiency in English Language Arts, compared to 40% of their non-EL peers [18]. Scores on the English Language Arts assessment demonstrate a significant drop with only 1% of RI EL students at the 7th grade level meeting proficiency compared to 41% of their non-EL peers on the 2016 PARCC test [19].

## 2.7    ELs in STEM Content Areas

There have been several studies conducted to examine EL learning in STEM education utilizing specialized instructional strategies and techniques. Fradd, Lee, Saxton, and Sutman [11] focused on several groups of EL students in the fourth grade who were taught using material that integrates science and literacy instruction. They concluded that teachers need more assistance with literacy and content area instruction to meet the learning needs of EL students, and that curriculum and materials in making science accessible to all students is the most important factor in student learning. They suggest utilizing the perspectives of teachers to promote science inquiry among their EL students, developing science curriculum that incorporates literacy and science learning, and reducing barriers in the inquiry process [11].

Lee [20] reviews research science curriculum and instruction and EL student learning. It summarizes key findings including that the diverse cultural background in EL students' home and communities should be incorporated into their science education, that there is a failure to provide sufficient instructional scaffolding for EL students in science classrooms, and that the reduction of equal science learn-

15

ing opportunities for EL students is increasing. Additionally, it describes findings in the literature that show that EL students can demonstrate academic achievement through "equitable learning opportunities" when science disciplines integrate EL students' linguistic and cultural experiences to further meaningful learning in science education [20]. The article concludes that the development of a research agenda in the future for EL students in science education should concentrate on the links between science outcomes and student learning, student diversity in science learning, diversity of student experiences in science curriculum and instruction, modifying high-stakes science assessment, and science teacher education [20].

Several articles emphasize that instructors in content areas should assist EL students in the development of their English literacy skills by providing meaningful learning with the language and content [20, 21]. Effective strategies for teaching EL students in content areas incorporates their prior linguistic and cultural knowledge in relation to science disciplines [20]. By integrating the prior knowledge and cultural experiences of EL students into science learning, these students can build upon their prior knowledge and relate the content to themselves in meaningful ways. It is also important to recognize that effective instructional strategies should incorporate a diverse variety of mundane experiences into the science learning process as students develop reasoning, inquiry, and discussion from such experiences [20].

Content area teachers should also take the necessary means of meeting with bilingual and/or EL certified teachers in an effort to discuss appropriate instructional strategies for EL students in their classes. Bilingual and EL certified teachers are "trained in the constructs of second language acquisition and related instructional methodology"; thus, they are able to help communicate the content material and instruction to EL students in effective and meaningful ways [22]. The bilingual

16

and EL certified teachers can assist content area teachers in making decisions on the best way to present and instruct the content area material to EL students. Additionally, content area teachers should consider all types of students in their classroom like EL students when creating standards for practical assessments to include differentiated instructional strategies for students with varied cognitive and linguistic abilities [23]. It is important that computer science instructors collaborate with EL instructors to provide the appropriate guidance required to teach computational thinking skills and concepts to EL students.

## 2.8    EL Instruction

Lee [20] shows that EL students confront the same level of academic rigor as their English-speaking peers; however, EL students are at a clear disadvantage as they are often taught by unprepared instructors lacking knowledge of effective EL strategies. Instructors may lower expectations for the level of learning and provide remedial tasks involving basic literacy and numeracy skills to EL students [20, 24]. EL students are denied equal learning opportunities as they find themselves focusing on basic literacy skills rather than higher-order thinking skills required in STEM courses. Additionally, the existing educational strategies do not successfully engage EL students in meaningful learning based upon their prior knowledge and experience within their unique cultures [20]. Therefore, EL students fail to connect science content in meaningful and relevant ways to themselves as well as fail to master competencies in English writing and speaking [20].

Instructors of EL students serve a significant role in their learning especially when utilizing effective practices in their classrooms. The instructors are responsible for integrating science content with English literacy in order to help their EL students develop new knowledge on the topic; thus, instructors should encourage student participation, assist and guide struggling students, review content, link

17

concepts, organize ideas, and summarize main ideas [11]. The implementation of these teaching practices by instructors may serve as invaluable tools for improving academic achievement of EL students in their classrooms as they continue to build upon and integrate new ideas into their prior knowledge. Lee [20] further concludes that instructors should use scaffolding in their classroom practices. There are also various forms of activities that can be incorporated into the classroom to assist in EL student learning. Hands-on activities can be helpful for EL students in classroom learning as these are not reliant on mastery of the English language via instruction. Collaborative learning is also very effective giving them the opportunity to develop their English language skills through communication about the STEM content as well as prepare them for real world models emphasizing collaborative projects in modern industries [20, 25]. Instructors need more detailed strategies and practices to prepare them to efficiently teach EL students. Within PDs, instructors should learn about integrating diverse languages into their content area curriculum, develop understanding of the cultural diversity, and clear their misconceptions about low expectations and underachievement of EL students [26].

## 2.9   ELs in Computer Science Education
### 2.9.1   Challenges

There are a number of challenges that educators confront when instructing EL students in computer science courses. The first challenge is the language used in most computer science curriculums is dense and complex with difficulty vocabulary [27]. EL students require necessary language skills and vocabulary to comprehend the content and utilization of the computer as a tool. Another challenge is the limited access EL students may have to computers and Internet within their home [27]. Without access to computers, EL students may not have exposure to technology or computing devices required to use in typical computer science courses.

As such, some EL students may lack necessary experience that can put them at a clear disadvantage than their non-EL peers. There is also the challenge of collaborating with schools that lack technical infrastructure and trained computer science instructors who can effectively serve all students including ELs [27]. Additionally, these trained computer science instructors may not be prepared to handle the needs and concerns of EL students in their classroom enacting ineffective techniques and strategies. With all these challenges, it is significant to implement effective practices commonly used in content areas to meet the needs of EL students in computer science classrooms.

### 2.9.2 Effective Instructional Methods

There are various methods that can be incorporated into computer science curriculums to effectively instruct EL students. These methods include utilizing hands-on activities, simple and step-by-step instructions, plethora of visualizations, condensed and piecemeal information, and real-world exercises [27, 20, 11]. Additional recommendations regarding the presentation of the materials consist of organizing exercises by tasks with concise and step-by-step instructions that uses visualizations to demonstrate the steps [27, 20, 11]. There should also be reinforcement opportunities in the EL student applies the knowledge and skills learned from previous tasks by practicing exercises that include less step-by-step instructions and visualizations [27, 20, 11].

The content in computer science curriculums can be made more accessible by incorporating important vocabulary. The vocabulary may range from basic (mouse, keyboard, button, computer) to advanced (computational thinking, data, simulation, programming) terms used in the content on the lesson [27]. The computer science instructor must determine the current vocabulary knowledge of EL students by speaking with them about the new words and allowing them to provide

19

multiple meanings of their definition of the terms [27]. Additionally, the instructor should review the technical terms that students will encounter as they progress through the exercises of the computer science materials [27]. It can even be beneficial to model the meaning of the vocabulary terms if possible [27]. Finally, the instructor should monitor the EL students for their understanding of the new vocabulary used in the material through informal or formative assessments [27].

Other effective instructional methods include providing handouts to EL students with visualizations and instructions. This can be beneficial especially if an instructor is demonstrating steps in an activity as the EL students can refer to this handout in the case that they fall behind or misunderstand the language used in instruction [27]. The creation of differentiated versions of exercises is also effective for EL students who may not be able to complete the same activities with complex reading and language that their non-EL peers may be able to [27, 23, 20, 11, 26]. These differentiated versions can contain less wording and more visualizations to emphasize steps to be completed in the exercises of the computer science curriculums. The instructors should also provide EL students with extended time to complete the exercises if the classroom time permits [27, 20, 11]. If not, it may be helpful for the instructor to offer an after-school program for EL students to participate in if they require additional time to complete the activities [27, 20, 11].

The final methods that can be incorporated into computer science education to effectively instruct EL students is meaningful learning and collaboration. The computer science curriculum material should provide the opportunity for meaningful learning and incorporate a variety of cultural knowledge and experiences into its curriculum [27, 23, 20, 11, 21, 26, 22]. EL students learning computer science should also be given the opportunity to work in collaboration with a partner while working on practical assignments. It may be best to pair EL students with

20

English-speaking students to provide them the opportunity to develop their language skills by communicating in English with their partner [27, 28]. It may also be helpful to consistently pair these students together throughout the semester so the EL student becomes more comfortable with the English-speaking peer allowing for more inquiry and discussion about the content [27, 28].

**List of References**

[1] National Science Foundation. "Broadening participation in computing." 2009. [Online]. Available: https://www.nsf.gov/pubs/2009/nsf09534/nsf09534.pdf

[2] National Center for Women In Technology. "Ncwit fact sheet." n.d. [Online]. Available: https://www.ncwit.org/ncwit-fact-sheet

[3] National Science Foundation. "Computing education for the 21rst century." 2012. [Online]. Available: fromhttp://www.nsf.gov/pubs/2012/nsf12527/nsf12527.htm

[4] National Science Foundation. "Stem + computing partnerships." 2015. [Online]. Available: https://www.nsf.gov/pubs/2015/nsf15537/nsf15537.htm

[5] University of Rhode Island Department of Computer Science. "Ap computer science principles course." 2017. [Online]. Available: http://k12.cs.uri.edu/apcsp.php

[6] "Next generation science standards." n.d. [Online]. Available: https://www.nextgenscience.org/2017

[7] University of Rhode Island Department of Computer Science. "Intro to computing and data science course." 2017. [Online]. Available: http://k12.cs.uri.edu/intro_computing.php

[8] Northwestern University. "Netlogo." 2017. [Online]. Available: https://ccl.northwestern.edu/netlogo/

[9] U. Wilensky. "Netlogo." 2016. [Online]. Available: https://ccl.northwestern.edu/netlogo/

[10] American Federation Of Teachers. "Every student succeeds act: A new day in public education - english language learners." 2016. [Online]. Available: http://www.aft.org/sites/default/files/essa_ells.pdf

21

[11] S. H. Fradd, O. Lee, F. X. Sutman, and M. K. Saxton, "Promoting science literacy with english language learners through instructional materials development: A case study," *Bilingual Research Journal*, vol. 25, no. 4, pp. 479–501, 2001.

[12] K. Lindahl. "The essa and esl teacher education." Dec. 2015. [Online]. Available: http://blog.tesol.org/the-essa-and-esl-teacher-education/

[13] Rhode Island Department Of Education. "Regulations governing the education of english learners in rhode island." Dec. 2016. [Online]. Available: http://www.ride.ri.gov/Portals/0/Uploads/Documents/Students-and-Families-Great-Schools/English-Language-Learners/EL_Regulations_Amendment_12_20_16.pdf

[14] A. C. Margo Gottlieb, M. Elizabeth Cranley. "Understanding the wida english language proficiency standards a resource guide prekindergarten through grade 12." 2009. [Online]. Available: https://www.wida.us/standards/Resource_Guide_web.pdf

[15] WIDA. "Why choose wida?" n.d. [Online]. Available: https://www.wida.us/membership/benefits.aspx

[16] WIDA. "The wida standards framework and its theoretical foundations." 2014. [Online]. Available: https://www.wida.us/get.aspx?id=731

[17] J. McFarland, B. Hussar, C. de Brey, T. Snyder, X. Wang, S. Wilkinson-Flicker, S. Gebrekristos, J. Zhang, A. Rathbun, A. Barmer, F. B. Mann, and S. Hinz. "The condition of education 2017." 2017. [Online]. Available: https://nces.ed.gov/pubs2017/2017144.pdf

[18] Rhode Island KIDS COUNT. "2016 rhode island kids count factbook." 2016. [Online]. Available: http://www.rikidscount.org/Portals/0/Uploads/Documents/Factbook%202016/2016%20Rhode%20Island%20KIDS%20COUNT%20Factbook.pdf

[19] Rhode Island KIDS COUNT. "2017 rhode island kids count factbook." 2017. [Online]. Available: http://www.rikidscount.org/Portals/0/Uploads/Documents/Factbook%202017/2017%20RI%20Kids%20Count%20Factbook%20for%20website.pdf

[20] O. Lee, "Science education with english language learners: Synthesis and research agenda," *Review of Educational Research*, vol. 75, no. 4, pp. 491–530, 2005.

[21] J. Meltzer and E. T. Hamann, "Meeting the literacy development needs of adolescent english language learners through content-area learning-part two: Focus on classroom teaching and learning strategies," 2005.

[22] K. Brooks, S. R. Adams, and T. Morita-Mullaney, "Creating inclusive learning communities for ell students: Transforming school principals' perspectives," *Theory Into Practice*, vol. 49, no. 2, pp. 145–151, 2010.

[23] A. Honigsfeld, "Ell programs: Not "one size fits all"," *Kappa Delta Pi Record*, vol. 45, no. 4, pp. 166–171, 2009.

[24] H. C. Waxman and K. Tellez, "Research synthesis on effective teaching practices for english language learners. publication series." 2002.

[25] M. Warschauer and D. Healey, "Computers and language learning: An overview," *Language teaching*, vol. 31, no. 2, pp. 57–71, 1998.

[26] J. Janzen, "Teaching english language learners in the content areas," *review of Educational research*, vol. 78, no. 4, pp. 1010–1038, 2008.

[27] K. Robertson. "Preparing ells to be 21st-century learners." 2017. [Online]. Available: http://www.colorincolorado.org/article/preparing-ells-be-21st-century-learners

[28] D. Barlow, "The sourcebook for teaching science: Strategies, activities, and instructional resources (grades 6-12)," *The Education Digest*, vol. 74, no. 6, p. 72, 2009.

23

# CHAPTER 3

## Methodology

This research study used research-based designs and strategies to evaluate the quality of the developed EL accessible materials, instructions, and assessments in computer science learning and achievement among EL students.

### 3.1 Programming and Simulation Unit

This study will focus on the first major unit, Programming and Simulation, of the ICDS curriculum comprised of two areas of focus: Inference/Models and Program Development. All of the lesson plans in both focus areas of the course are shown in Appendix A.

### 3.1.1 Inference/Models Area Of Course

The Inference/Models area of the course consists of the first two lessons in the Programming And Simulation unit of the ICDS course. Within this section of the ICDS course, students will be introduced to computational thinking, data science, and simulations which they will use to analyze real-world phenomenons, construct hypotheses regarding the outcomes of the simulation, and test through hypotheses by experimenting with the simulation to justify conclusions [1, 2, 3].

The first lesson is referred to as the Introduction and Computing and Data Science. In this lesson, students are introduced to concepts of computational thinking and data science through a series of brief classroom videos and instructor-led discussion [1, 2]. Students will then be separated into small groups to engage in an unplugged activity in which they use collaboration to solve problems requiring computational thinking skills [1, 2]. Once the instructor is satisfied that the students gained an understanding of computational thinking, students will complete a prac-

24

tical assessment introducing the interface of the simulation environment NetLogo Web simulating a virus spreading through a population [1, 2, 4]. In this practical assessment, the students will construct hypotheses regarding the outcomes of the virus spreading through the population by experimenting with parameters to modify the number of people in the population, infection and recovery rate, and duration of the virus [1, 2]. Using their results from conducted experiments in the simulation, students will justify their conclusions concerning their hypotheses.

The second lesson is referred to as Modeling and Simulation. Students begin this lesson by reading an article on computing and simulation modeling and watch a video introducing the interface and features of NetLogo Web to further introduce them to the concepts of simulations and their representation of real-world phenomenons [1, 3]. This lesson continues to use NetLogo Web in two practical assessments where students use several simulations to represent real world phenomenons and formulate hypotheses that can be tested through these simulations [1, 3, 4]. Specifically, students use a predation model in which they can experiment with parameters to expand or deplete populations of prey and predators. Additionally, the students will use a party simulation to experiment with the social interactions of different genders at a party [1, 3]. This lesson is concluded with a classroom discussion concerning the significance of simulations to represent real-world scenarios [1, 3].

### 3.1.2 Program Development Area Of Course

The Program Development area of the course consists of the remaining three lessons in the Programming And Simulation unit of the ICDS course. This section of the ICDS course introduces the concept that simulations are programmed to behave in certain ways when actions are instantiated [1, 5, 6, 7]. Within this section, students command elements in the simulations to perform actions, program

models to simulate a real-world phenomenon using the NetLogo Web programming language, modify existing code according to program specifications, and collect and analyze data to justify conclusions to hypotheses [1, 5, 6, 7].

The third lesson of the Programming and Simulation unit is referred to as Agent Commands. This lesson introduces the concept of agent based computer models used in NetLogo Web in which the agents are programmed to create the different features and behaviors of the model [1, 5]. Students will then complete one practical assessment where they read and modify existing code to create different behaviors in an existing traffic model [1, 5, 4]. This is the first exposure to programming in the ICDS course so this lesson eases students into program development by issuing simple commands such as changing the color of the cars and background [1, 5]. As students program elements in the traffic simulation, they detail the modifications and outputs being made to the model as they issue commands [1, 5].

The fourth lesson in this unit is referred to as Programming Procedures in which students further develop programming skills in NetLogo Web. Students begin this lesson by watching a motivational video detailing the significance of learning to code, which is the main focus in this lesson [1, 6]. Students complete four practical assessments where they develop their programming skills by building a simple ecosystem model [1, 6]. Students will program several procedures using the different agent types in NetLogo Web to implement behaviors in the model when clicking a button and adjusting parameter values in sliders [1, 6, 4]. This lesson introduces more advanced programming skills but it is significantly scaffolded to guide students in learning the various syntax and commands of the programming language [1, 6].

The fifth lesson is the final one in the unit. It is referred to as Program Your

26

Own Simulation. Within this lesson, students first watch a video reviewing advanced programming concepts and providing examples of these concepts in practice using NetLogo Web [1, 7]. Students then complete two practical assessments in which they build a mousetrap simulation by modifying existing code and collect data from it to answer an important question about the model [1, 7, 4]. This lesson encapsulates all previously learned skills and introduces more advanced programming concepts (logic, conditionals, loops) in NetLogo Web [1, 7, 4]. Additionally, it requires that students used previous knowledge and skills with limited scaffolding to program a unique simulation model that does not exist in the NetLogo Web library [1, 7, 4].

## 3.2 Purpose, Research Questions and Hypotheses

The purpose of this study is to create EL accessible materials, instructions, and assessments for the ICDS curriculum to support computer science and evaluate the quality of these computer science materials for EL students. This study will produce computer science materials for EL students resulting in significant impact on their computer science learning and achievement. These materials will ideally be incorporated into the ICDS curriculum in the upcoming academic school year.

The terms computer science learning and achievement can have varied meanings across different curriculums and studies. For the purposes of this research study, computer science learning and computer science achievement will be defined as the following:

> ***Computer Science Learning***: The student's growth over the semester measured through the practical assessments provided in each of the five lessons of the Programming and Simulation unit of the ICDS course. Each lesson uses one or more practical assessments to demonstrate understanding and skills regarding the learning objectives of the

respective lesson.

***Computer Science Achievement***: The student's performance at 9-10th grade level measured through a pretest and posttest emphasizing the key concepts of the entire Programming and Simulation unit.

There are several research questions and hypotheses pertaining to this area of inquiry. Both the research questions and their respective hypotheses are listed in the following.

***RQ1***: How do the instructional objectives meet the needs and concerns of the EL students?

***H1***: The instructional objectives will meet the needs and concerns of the EL students by using content and language objectives for each lesson, removing unnecessary information that does not pertain to the objectives, emphasizing content vocabulary at the beginning of each lesson, and developing assessments to evaluate the content.

***RQ2***: How is the quality of the materials in terms of computer science learning and achievement among the inference/models and program development areas of the course? In which area(s) will EL students demonstrate the highest level of computer science learning and achievement?

***H2***: The quality of the materials in the inference/models and program development of the course will produce significant impact in terms of computer science learning and achievement among EL students by achieving its instructional objectives, obtaining and maintaining interest, engaging prior knowledge and experience, and being reliable. The evaluation of the two areas of focus in the ICDS course will produce conjectures that EL students will demonstrate the highest level of learning in the inference/models area of the course due to its visual emphasis on the simulations and use of prior knowledge involving the scientific method.

28

**RQ3**: What effect, if any, does English language proficiency have on the ability to learn programming syntax?

**H3**: There will be significant effects on the ability to learn programming syntax due to the English language proficiency levels. The evaluation of the materials will produce conjectures that those with higher levels of English language proficiency will demonstrate a significantly higher level of learning involving programming syntax than those with lower levels of English language proficiency.

This proposed research study aims to satisfy these research questions with conclusive evidence that either proves or disproves the hypotheses.

### 3.3   Participants

The participants of this study consist of the four experts that will establish the panel to review and evaluate the quality of the developed EL accessible materials, instructions, and assessments. Each of these experts are extensively involved in instruction, curriculum and assessment design for EL students.

The first expert is Amy Correia, a URI Teaching English to Speakers of Other Languages who designed and leads the Teaching English to Speakers of Other Languages (TESOL)/Bilingual and Dual Language Education certificate program at the URI. This expert has extensive teaching experience with EL students at the high school level in which she taught Beginner English Language Learners (ELL) English, American Literature, and Public Speaking. Additionally, she served as the administrative head of the ELL Program in which she trained faculty, assessed and situated incoming EL students, aligned curriculum with national and state standards, and advocated for policy changes concerning EL education [8].

The second expert is Alison Murray who is an instructor at Central Falls High School (CFHS) with extensive background and experience in developing curriculum for EL students in science courses offered at CFHS. This CFHS instructor also

29

aided in the development of the EL accessible materials including the differentiated versions of readings and activities, additional instructions for the EL adaptations to lessons, and vocabulary resources. Additionally, Ms. Murray will pilot lessons and activities from the adapted ICDS course material with a mixture of 9-12th grade students in her engineering course. The specific lessons she will be enacting with her students include all five lessons in the Programming and Simulation unit.

The remaining participants who will provide feedback regarding the EL accessible materials, instructions, and assessments are two certified ICDS instructors, Suzanne Ross Walker and Nancy Koczera. These two participants will enact the ICDS course in its entirety (Programming and Simulation and Data units) at Woonsocket High School (WHS). Since WHS includes a large EL student population, both of these instructors agreed to use the developed EL materials for the ICDS course when teaching their EL students [9]. Additionally, these instructors will evaluate and provide feedback regarding the developed EL materials for the Programming and Simulation unit as they implement them in the ICDS course.

This panel of four qualified experts will be responsible for evaluating and providing valuable feedback on the quality and effectiveness of the EL accessible materials, instructions, and assessments developed for the ICDS course. Additionally, these experts will provide responses regarding necessary revisions to be made to the material in order to further meet the needs of EL students. Their evaluations will significantly aid in the production of reliable and quality materials that may be implemented in the ICDS curriculum for use by EL students in the upcoming academic school year.

## 3.4   Procedure

For this study, I worked with external experts to create EL adapted lesson plans, readings, videos, student activity guides, and vocabulary resources for the

Programming and Simulation unit. Each lesson plan included footnotes containing adaptations to the required readings, videos, and activity guides for EL students. Adaptations to readings and activity guides included less complex language, visualizations to make better connections with content driven vocabulary, and links to the Merriam-Webster online dictionary for difficult vocabulary [10]. The adaptations for the videos included notes regarding its vocabulary, timeframes to pause and reinforce concepts, text captioning with translation options, and questions to assess comprehension of concepts reviewed in the video. Each lesson includes various resources for vocabulary in the readings, videos, and activity guides. The vocabulary for each lesson include a vocabulary guide, cards, slideshow with audio, word search and crossword, word wall, and interactive quiz. Further details regarding the EL accessible materials to be developed for the ICDS course are presented in the subsections below.

The evaluation of the developed EL materials for the ICDS curriculum consisted of an expert appraisal system based upon Dr. Veniegas methods of evaluations [11]. This expert appraisal system included several systematic steps to evaluate the materials as well as validate the effectiveness and quality of them. First, the experts were provided a digital copy of all the materials and reviewed these materials by following all instructions and completing all student activities [11]. As the experts progressed through the material, they provided qualitative feedback on its relevance, learning objectives, reading and language level, difficulty, strategies associated with the experience and maturity of the learners, and effectiveness [11]. The experts also used a rating scale to evaluate the quality of all materials. The experts used a four-point rating scale developed by [11] which is shown in Table 1.

| Very Satisfactory (VS) | Satisfactory (S) | Unsatisfactory (U) | Very Unsatisfactory (VU) |
|---|---|---|---|
| 3 | 2 | 1 | 0 |

Table 1: This table shows four-point rating scale developed by [11] that will be used to evaluate the quality of the developed materials.

The experts also provided written feedback for all the materials by listing three of its strengths, highlighting three of its weaknesses that require improvement, and giving general commentary on it [11]. Additionally, the panel of experts were provided with the United Nations Educational, Scientific and Cultural Organization (UNESCO) quality curriculum criteria in which they respond to several questions to assess the quality of the curriculum [12]. The UNESCO quality curriculum criteria includes the following [12]:

(a) Are there clear aims for the curriculum?

(b) Is the curriculum up to date?

(c) Is it relevant to students' current and future lives, experiences, environments and aspirations?

(d) Is the curriculum equitable and inclusive?

(e) Is the curriculum learner-centered and learner-friendly?

(f) Is the curriculum open and flexible, so that it can address new challenges and opportunities by integrating new/emerging issues?

(g) Is the curriculum coherent and consistent across different education stages/grades/streams and learning areas/subjects?

After the experts completed their review of the materials, I collected their feedback and analyzed the qualitative and quantitive data. The qualitative data consisted of any remarks they made on the material as well as their responses to the UNESCO criteria. Using this qualitative data, I analyzed their comments for common suggestions that warrant revision especially if both experts concur on particular revisions. The quantitative data consisted of the ratings from experts. Using these ratings, the low scores indicated areas of the material that need to be reviewed and potentially revised. Overall, I used their valuable feedback to implement necessary revisions to the materials to ensure that the needs and concerns of the EL students will be addressed in the ICDS curriculum.

### 3.4.1 Lesson Plans

The lesson plans of the course provide instructors with several essential components to enhance scaffolding the material. Lesson plans are based on the Code.org outline [13]. They include an overview and purpose of the lesson, learning objectives, preparation and materials, and agenda. The agenda of each lesson consists of a warm-up, activity, and wrap-up section. The warm-up section includes introductory instructor remarks establishing the learning objectives and concepts to be covered in the lesson. Additionally, the warm-up section includes at least one required reading and/or video to further introduce students to the concepts they will be working with in the activities. The activity section includes additional instructor remarks regarding the tasks students will complete in the activities as well as copies of the student activity guides with notes on supporting students throughout the lesson. Finally, the wrap-up section includes several discussion questions to review key concepts and learning objectives of the lesson.

Each lesson plan includes footnotes containing adaptations to the required readings, videos, and activity guides for EL students as shown in Figure 3. Adap-
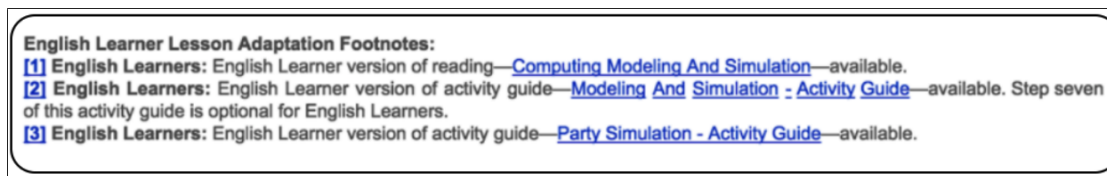
Figure 3: ICDS footnotes with the adaptations in this lesson for EL students.

tations to readings and activity guides include less complex language, visualizations to make better connections with content driven vocabulary, and links to the Learner's Dictionary (i.e., the Merriam-Webster online dictionary) for difficult vocabulary [10]. The adaptations for the videos include notes regarding its vocabulary, timeframes to pause and reinforce concepts, text captioning with translation options, and questions to assess comprehension of concepts reviewed in the video.

### 3.4.2 Content and Language Objectives

Each lesson contains content and language objectives that will measure the learning objectives for EL students. The content objectives will define the knowledge and skills that all students including ELs should learn by the end of the lesson. Students will be assessed on the content objectives of the lessons by completing the practical assessments that are referred to as activity guides in each lesson of the Programming and Simulation unit. The language objectives are used specifically for EL students in which it defines how ELs will use the four language domains (reading, writing, listening, and speaking) to show their understanding of the content objectives. An example of the content and language objectives used in a lesson of the Programming and Simulation unit is shown in Figure 4.

34

**Content Objectives**

*Students will be able to:*

- Explain that models are programmed.
- Command elements of NetLogo to perform actions.
- Read and modify NetLogo code to create different behavior in an existing model.
- Operate the Command Center of NetLogo.
- Issue commands to different agents in NetLogo.

(a) Content Objectives

**Language Objectives**

*Students will be able to:*

- *Orally explain* and *summarize in writing* that models are programmed.
- *Write commands* and *summarize in writing* the elements they command in NetLogo to perform actions.
- *Read* and *write* NetLogo code to create different behavior in an existing model.
- *Read* and *write* code in the Command Center of NetLogo.
- *Read* and *write* commands to different agents in NetLogo.

(b) Language Objectives

Figure 4: The content and language objectives for Lesson 3 - Agent Commands

### 3.4.3 Readings

The lessons include at least one required reading containing information on concepts that students should be introduced to before beginning the activities of the lesson. There are differentiated versions of the required readings where one is for average high school reading levels and the other is for EL students and struggling readers. Focusing on the EL version of the readings, these will encompass more simplified wording, imagery to represent concepts and vocabulary, links to the Learner's Dictionary for simplified definitions of difficult vocabulary, and comprehension questions that can be used in oral or written assessments. Additionally, the readings include a vocabulary guide sheet in which difficult vocabulary in the reading will be given with definitions and visualizations which is further discussed in the vocabulary section below.

In Figure 5, the left side shows the current reading on simulations assessed at the average reading level for a 9th/10th grade student. On the right side of Figure 5 shows the differentiated reading on simulations for EL students. As shown, the differentiated version includes simplified language, images to represent real world simulations, and links to the online Learner's Dictionary for difficult vocabulary such as simulation, abstract, and hypothesis.

35

Figure 5: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Comparison of required reading versions on simulations.



Figure 6: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation reading vocabulary guide.

Additionally, students will be provided with a vocabulary guide with vocabulary specific to the required reading as shown in Figure 6. This guide includes all difficult vocabulary with the word hyperlinked to the online Learner's Dictionary, a simple definition, and a visual to represent the written word. Further information on the vocabulary guide is discussed in the Vocabulary Resources section below.

### 3.4.4 Videos

The lessons include at least one required video containing information on concepts that students should be introduced to before beginning the activities of

36

the lesson. The videos are significant tools in learning as it adds a visual component to the concepts being taught in the lesson. The videos will serve as motivational messages to students encouraging them to pursue computer science and STEM careers as well as informational in regards to concepts in the lessons that can be visualized and include more details. The instructor will receive lecture notes on the videos of each lesson with details regarding complex vocabulary, instructions and timeframes on where to pause videos to reinforce concepts, vocabulary guides, and comprehension questions that can be used in oral or written assessments. Additionally, all videos include features with options for closed captioning and translation into various languages.



Figure 7: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation video vocabulary guide.

As shown in Figure 7, the vocabulary guide also includes a section for the vocabulary in the required video in this lesson. It includes all difficult vocabulary with the word hyperlinked to the online Learner's Dictionary, a simple definition, and a visual to represent the written word.

In Figure 8, the top left side shows the closed captioning option turned on and automatically translating the captions to English. Selecting the Auto-translate feature allows the option to translate the closed captioning of the video to various languages as shown in the top right image. The bottom image shows selecting Spanish as the translation option and translating the closed captioning of the video to Spanish rather than English. The purpose of translation for EL students
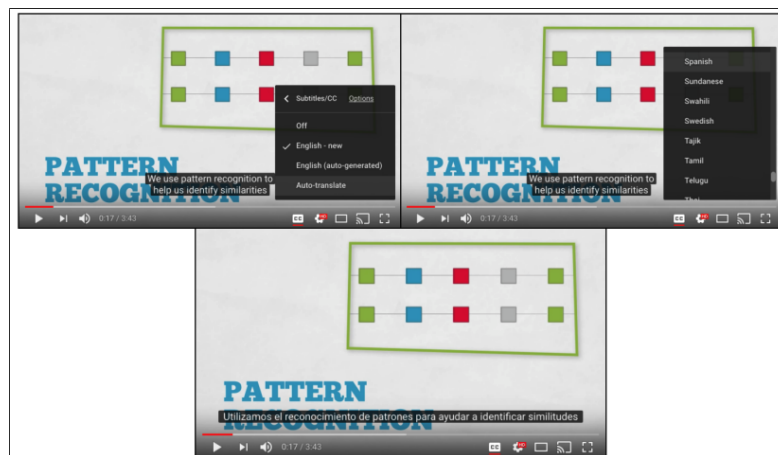
Figure 8: ICDS UNIT ONE: Lesson 1 - YouTube video, Solving Problems at Google Using Computational Thinking

is to ease students into the introduction of concepts in the lesson without difficulty of processing and translating the spoken and unfamiliar words.

### 3.4.5 Activities

Each lesson includes at least one activity guide for students to complete. Activity guides include detailed procedural steps to apply practical skills to the concepts reviewed in readings, videos, and discussions as well as necessary scaffolding for the students. Each activity guide will guide students through steps to help students answer questions and produce a final product (i.e., a completed program, spreadsheet, slideshow, etc.). These activity guides also include visual examples for students to refer to ensure they are correcting following each step in the activity. Additionally, all activity guides include symbolic icons to represent actions to be performed by the students. These icons are significant to enhance learning of EL students as they can more easily interpret pictures with minimum English language skills [14].

Figure 9 shows the procedural steps that students will see in every activity guide. Students will note that the green keywords *Do The Following* serves as an
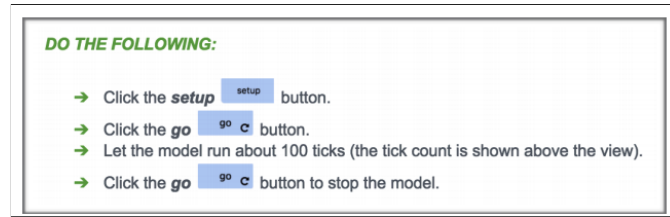
Figure 9: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Activity Guide
Do The Following icon indicates performing a set of tasks.

indicator in which they will perform a certain set of tasks such as clicking a button
as shown above.
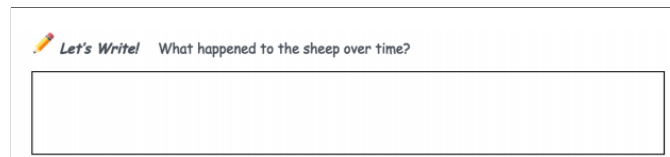


Figure 10: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Activity Guide
Pencil icon indicates written response for the question.

Figure 10 shows a pencil icon with the phrase *Let's Write!*. Students will note
that this icon and keyword serve as an indicator in which they will write a few
sentences to answer the question within the box provided for them.
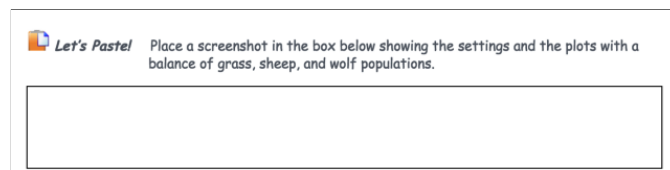


Figure 11: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Activity Guide
Paste icon indicates pasting a screenshot or image for the question.

Figure 11 shows a paste icon with the phrase Let's Paste! Students will
note that this icon and keyword serve as an indicator in which they will paste a
screenshot or image to answer the question within the box provided for them.

There are differentiated versions of the activity guides where one is for non-
EL students and the other is for EL students. All versions of the activity guides
include all the components described above; however, the EL version of activity

guides will use more simplified language with additional hyperlinks on difficult vocabulary connecting them to the online Learner's Dictionary definitions.
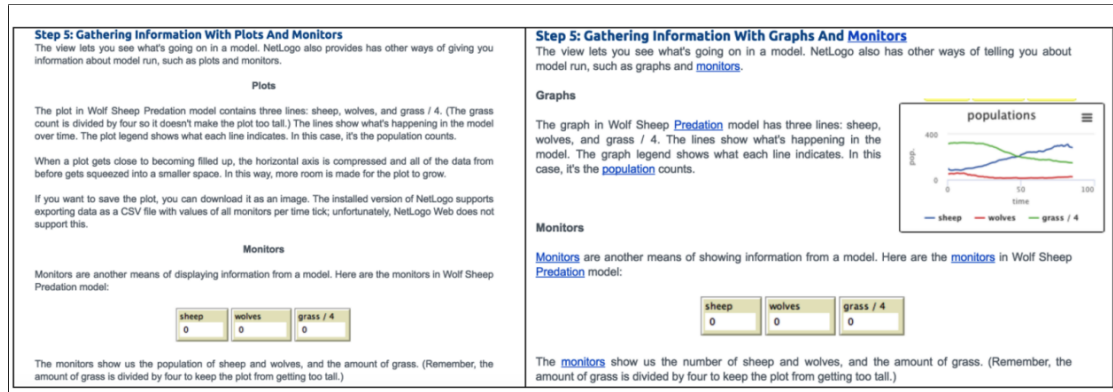


Figure 12: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Activity Guide Comparison of activity guide versions in Step 5.

Figure 12 shows the different versions of activity guides available for non-EL and EL students. The image on the left side shows the steps of the activity guide that non-EL students would see. The image on the right side shows the differentiated activity guide to be used for EL students which include simplified language (i.e., replacing plots with graphs), links for difficult vocabulary to the Learner's Dictionary definitions and vocabulary guides, and additional visuals (i.e., the plot of the populations).

### 3.4.6 Vocabulary Resources

Each lesson includes various resources for vocabulary in the readings, videos, and activity guides. The vocabulary for each lesson includes a vocabulary guide, cards, slideshow with audio, word search and crossword, word wall, and an interactive quiz tool called quizizz as shown in Figure 13.

The vocabulary guide contains vocabulary from the readings, videos, and activity guides of the lesson. It includes a table with the vocabulary word hyperlinked to the Learner's Dictionary, the definition of the word, and a visual for the word.

www.manaraa.com
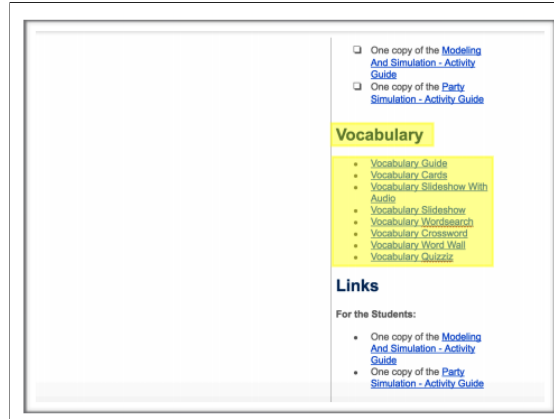
Figure 13: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Lesson Plan Each lesson plan contains a section dedicated to vocabulary resources.

Examples of the vocabulary guide can be seen in Figures 6 and 7.



Figure 14: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Vocabulary Cards.

The vocabulary cards contain vocabulary from the readings, videos, and activity guides of the lesson. It includes a printout of boxes with different vocabulary words and definitions that can be cut and separated. This allows the students to arrange the vocabulary word with its respective definition. The vocabulary cards serve as a fun and interactive method for students to gain a stronger understanding of the vocabulary in the lesson. An example of these vocabulary cards are shown in Figure 14.

41
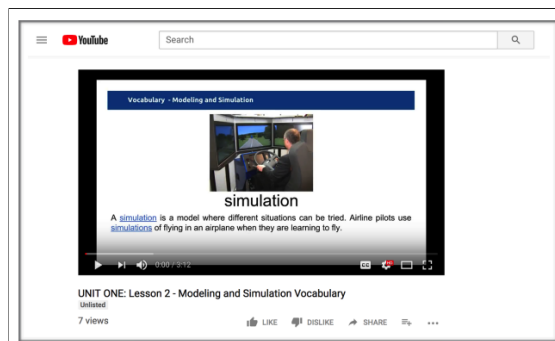
Figure 15: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Vocabulary Video.

The vocabulary slideshow with audio contains vocabulary from the readings, videos, and activity guides of the lesson. It includes a link to a YouTube video where the vocabulary words and definitions are pronounced and read aloud. These videos can be especially beneficial to EL students and struggling readers in the classroom as they are able to make a connection with the written word and auditory pronunciation of it. An example of the vocabulary slideshow with audio can be seen in Figure 15.
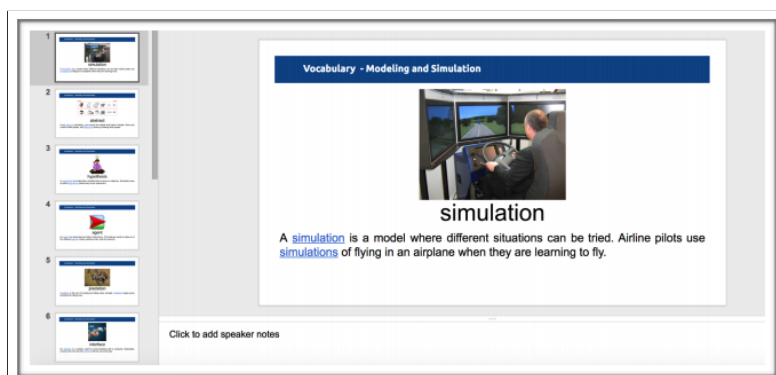


Figure 16: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Vocabulary Slideshow.

The vocabulary slideshow contains vocabulary from the readings, videos, and activity guides of the lesson. It includes the same slideshow presentation used in the YouTube video for the slideshow with audio. Additionally, the slideshow includes the vocabulary word hyperlinked to the Learner's Dictionary, the definition of the

word, the word used in a sentence, and a visual for the word. An example of the vocabulary slideshow can be seen in Figure 16.
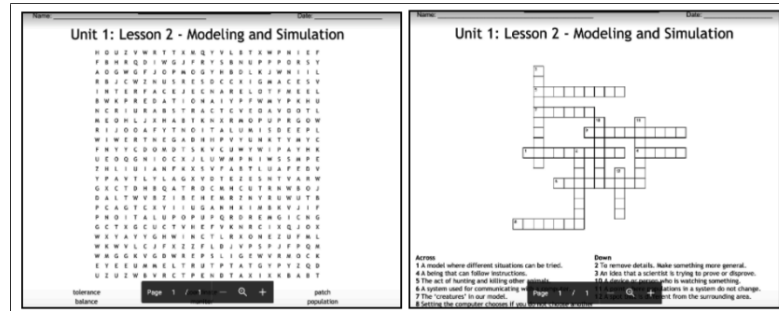


Figure 17: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Vocabulary Word Search/Crossword.

The vocabulary word search and crossword contains vocabulary from the readings, videos, and activity guides of the lesson as shown in Figure 17. It is yet another fun and interactive method for students to gain a stronger understanding of the vocabulary in the lesson.
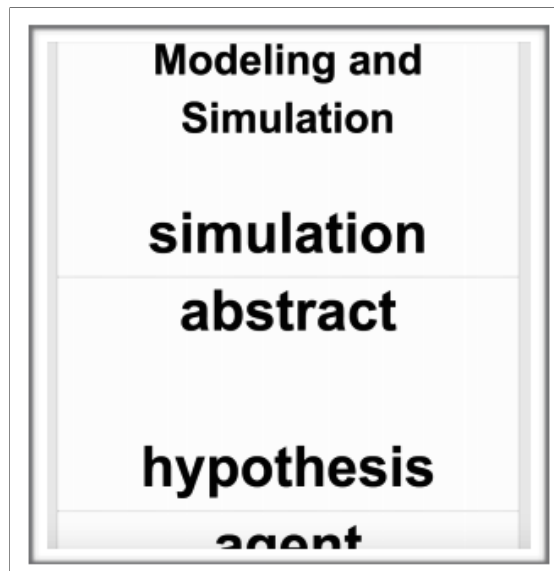


Figure 18: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Vocabulary Word Wall.

The vocabulary word wall contains vocabulary from the readings, videos, and activity guides of the lesson as shown in Figure 18. The word wall can be printed

and hung on a wall of the classroom to serve as yet another interactive and visual tool for the students.
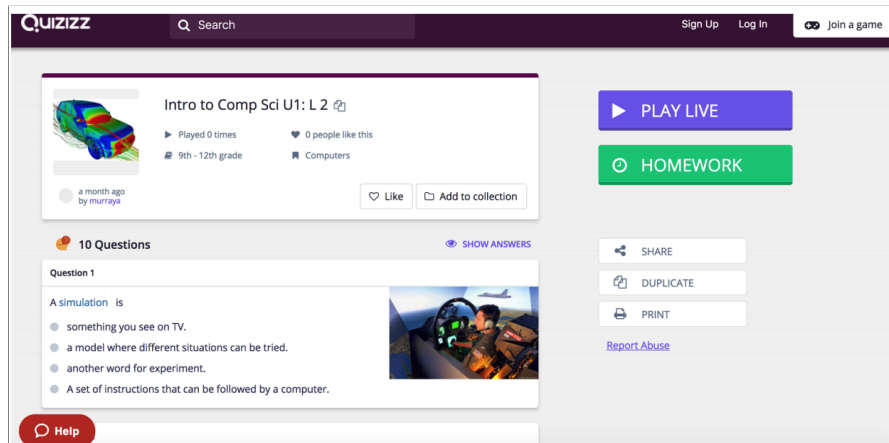


Figure 19: ICDS UNIT ONE: Lesson 2 - Modeling And Simulation Vocabulary Quizizz.

The vocabulary Quizizz contains vocabulary from the readings, videos, and activity guides of the lesson as shown in Figure 19. Quizizz provides a student-paced formative assessment in a fun and engaging way in which questions appear on each student's screen, so they can answer questions at their own pace, and review their answers at the end. More importantly, Quizizz gives detailed information regarding student answers on each quiz.

### 3.4.7   Interpersonal Strategies

A majority of the activities involve group work in which students will work together for multiple lessons to produce a final product (i.e., program, spreadsheet, data story presentation). The group projects allow both non-EL and EL students to participate in collaborative learning in which they exchange, write, develop, and share ideas [14]. Instructors will also arrange student groups by pairing EL students with either bilingual students who can easily translate the steps in the activities or with non-EL students with strong English language communication skills who can benefit in learning the material by teaching it to EL students [14].

44

Additionally, instructors will implement the think/pair/share strategy allowing EL students to communicate their ideas in English through interaction with other students, written responses to questions, and classroom discussion lead by the instructor [14].

## List of References

[1] University of Rhode Island. "Intro to computing and data science curriculum." May 2017. [Online]. Available: http://k12.cs.uri.edu/intro_computing_curriculum_2017-18.php

[2] University of Rhode Island. "Lesson 1: Introduction to computing and data science." 2017. [Online]. Available: https://docs.google.com/document/d/13SZN2rEnNK_OWEOjYEXTU70DE7LfBgjV9MgELSDUj_k/edit?usp=sharing

[3] University of Rhode Island. "Lesson 2: Modeling and simulation." 2017. [Online]. Available: https://docs.google.com/document/d/1d3ys4k8w-MXrXzPUaVir6vb4j72SqQvSMhhJHiQLsyw/edit?usp=sharing

[4] U. Wilensky. "Netlogo." 2016. [Online]. Available: https://ccl.northwestern.edu/netlogo/

[5] University of Rhode Island. "Lesson 3: Agent commands." 2017. [Online]. Available: https://docs.google.com/document/d/10Tsv7JQ2eSPeDWItagxUI7AcYWbuQ6_xzfetJNFaiVQ/edit?usp=sharing

[6] University of Rhode Island. "Lesson 4: Programming procedures." 2017. [Online]. Available: https://docs.google.com/document/d/1W2apJadkWqsJ_a_jMRUij-vU2FkGwPDvuLl28fFO0xk/edit?usp=sharing

[7] University of Rhode Island. "Lesson 5: Program your own simulation." 2017. [Online]. Available: https://docs.google.com/document/d/1Nq4swYf9DfjTx4F6GjG81RFz0xRYwzgPUiUHlsXwZsY/edit?usp=sharing

[8] University of Rhode Island School of Education. "Amy correia." 2018. [Online]. Available: https://web.uri.edu/education/meet/amy-correia/

[9] Rhode Island KIDS COUNT. "2016 rhode island kids count factbook." 2016. [Online]. Available: http://www.rikidscount.org/Portals/0/Uploads/Documents/Factbook%202016/2016%20Rhode%20Island%20KIDS%20COUNT%20Factbook.pdf

[10] Merriam-Webster. "Merriam-webster's learner's dictionary." 2018. [Online]. Available: http://www.learnersdictionary.com/

[11] O. P. Veniegas. "Ensuring the effectiveness and quality of learning materials: How to use and evaluate literacy/ ce materials." n.d. [Online]. Available: http://www.accu.or.jp/litdbase/pub/dlperson/pdf0106/rpp24_1.pdf

[12] P. Stabback. "What makes a quality curriculum?" 2016. [Online]. Available: http://unesdoc.unesco.org/images/0024/002439/243975e.pdf

[13] Code.org. "Cs discoveries 2018." 2018. [Online]. Available: https://curriculum.code.org/csd-18/

[14] D. Barlow, "The sourcebook for teaching science: Strategies, activities, and instructional resources (grades 6-12)," *The Education Digest*, vol. 74, no. 6, p. 72, 2009.

# CHAPTER 4

## Findings

After completing the development of the EL accessible instructions, materials, and assessments for the Programming and Simulation unit, these materials were made accessible to the panel of experts for their review. The panel reviewed the materials and provided feedback to an online survey with questions pertaining to its strengths, weaknesses, commentary, the rating scale to evaluate the quality of all materials, and the UNESCO quality curriculum criteria [1, 2]. The following sections are separated by their responses relating to either the Inference/Models or Program Development area of the course.

## 4.1 Inference/Models Area of the ICDS Course
### 4.1.1 Instructional Objectives

The instructional objectives are designed to be understandable to EL students regardless of English level proficiency. These instructional objectives focus on delivering content to the students and eliminating difficult language as well as providing proper scaffolding that can allow students to independently work through activities on their own. Additionally, the instructional objectives allow EL students numerous opportunities to practice these objectives.

The instructional objectives for the inference/models area of the course do not explicitly define the methods in which the language objectives will be measured. There are also some activities within this section that are longer than others and require more explicit clarification to EL students.

### 4.1.2 Strengths of the Language

In terms of the strengths of the language used in the content of this section, there are numerous opportunities for EL students to discover definitions of words

47

used within the readings, videos, and activities. EL students are often supported with various visualizations as they progress through the exercises for the lessons with the same icons used to represent language throughout the exercises. This allows EL students to become comfortable with the icon matched with the language so they can more easily navigate exercises in future lessons. Additionally, all the language domains are addressed in the Inference/Models area of the course including speaking, listening, reading, and writing.

Other strengths of the language used in the content demonstrated in this section are that it is clear, concise, understandable, less wordy and complex, and more direct. The language used in the exercises also allow EL students to focus on the content rather than becoming burdened with interpreting complex vocabulary and lengthy sentences.

### 4.1.3  Weaknesses of the Language

There are several weakness of the language used in the content of these lessons that require improvement. Firstly, the language objectives for these lessons are written more as tasks to complete by the students rather than measurable outcomes. There is also a need to explicitly recommend that teachers use the closed captioning feature whenever playing a video in the class. All the videos in these lessons use YouTube as the platform to deliver the videos meaning that the closed captioning feature is a built-in function that can easily be incorporated. Additionally, some of the content is still difficult for EL students to complete especially if they have limited experience with computer science concepts and simulations.

### 4.1.4  General Commentary

Overall, the concentration of the perspectives of EL students is a compelling idea especially as more of these students populate the computer science educa-

tion courses within the public education system. The use of leveled supports as footnotes for the teachers to incorporate differentiated learning for EL students is helpful to both the teachers and students. It will need to be made explicitly clear to incoming teachers of the course that these leveled supports for EL students exist as well as the best method to deliver these materials to students. Additionally, these materials are not only useful for EL students but also all types of students who require simplification of the language used in the exercises and vocabulary resources to better understand the presented content. The supplementary vocabulary materials can also be used as formative assessments to demonstrate the understanding of the vocabulary and its relation to the content.

### 4.1.5 Quality of EL Accessible Materials

The quality of the objectives of the lessons in the Inference/Models area of the course is rated as very satisfactory by three experts (75%) and satisfactory by one expert (25%) as shown in Figure 20. As stated previously, there are some improvements that can be made to the objectives in terms of designing them as measurable outcomes rather than tasks for students to complete.



**Use the four-point rating scale to evaluate the quality of the OBJECTIVES of the lesson.**

4 responses

Figure 20: Responses to using the four-point rating scale to evaluate the quality of the objectives of the lessons in the Inference/Models section of the ICDS course.

The quality of the vocabulary resources of the lessons in the Inference/Models area of the course is rated as very satisfactory by all four of the experts (100%), which is shown in Figure 21. The qualitative feedback regarding the vocabulary

49

resources seems to be positive and highly recommended by the teachers enacting the course. It is helpful for the teachers to have these vocabulary resources so they can concentrate on delivering the content rather than consuming valuable classroom time finding and creating these materials. A recommendation from the qualitative feedback included alphabetizing the vocabulary in the resource guide containing the words, definitions, and visualizations.
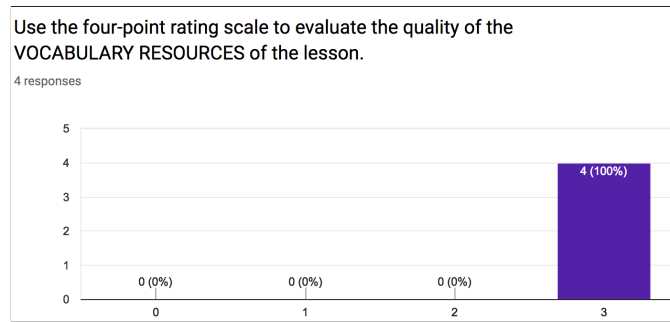


Figure 21: Responses to using the four-point rating scale to evaluate the quality of the vocabulary resources of the lessons in the Inference/Models section of the ICDS course.

The quality of the EL lesson adaptation footnotes in the Inference/Models area of the course is rated as very satisfactory by all four of the experts (100%) as shown in Figure 22. As noted in the qualitative feedback, the use of leveled supports as footnotes in the lesson plan are valuable to teacher who have EL students requiring differentiated versions of exercises and activities. Although these leveled supports are rated as very satisfactory, there needs to be explicit instruction on the existence of these materials and the best method to digitally deliver these materials to EL students.

The quality of the warm-up section in the Inference/Models area of the course is rated as very satisfactory by three experts (75%) and satisfactory by one expert (25%), which is shown below in Figure 23. The qualitative feedback regarding the warm-up section consisted of recommendations on the readings and videos to be
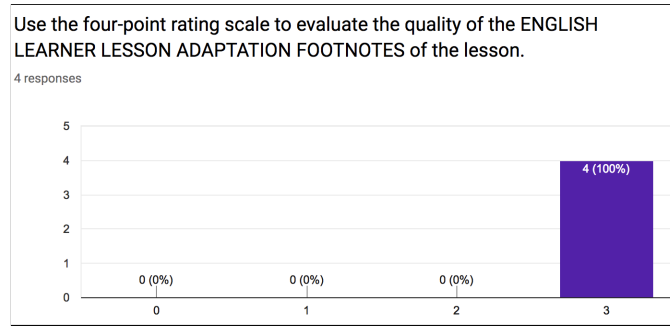
Figure 22: Responses to using the four-point rating scale to evaluate the quality of the EL lesson adaptation footnotes of the lessons in the Inference/Models section of the ICDS course.

used to introduce the concepts in the lesson. As noted, there are some readings that have not been differentiated for EL students. These readings are dense and contain complex wording and sentence structure that may be difficult for an EL student to interpret the content. As such these readings will need to be adapted to include less complicated wording, more visualizations, and links to an online dictionary for content vocabulary. Other recommendations for the videos include advising teachers to use the close captioning feature whenever presenting these videos to the classroom.
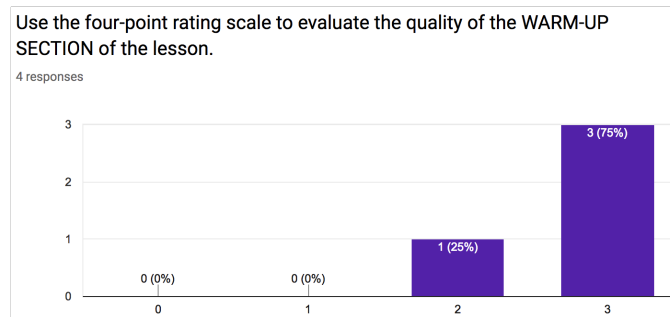


Figure 23: Responses to using the four-point rating scale to evaluate the quality of the warm-up section of the lessons in the Inference/Models section of the ICDS course.

The quality of the activities of the lessons in the Inference/Models area of the course is rated as very satisfactory by all four the experts (100%) shown below in

51

Figure 24. According to the qualitative feedback, the activities provide opportunities for the students to better understand and relate to the content as well as practice the instructional objectives of the lessons. Although some of the content may still be challenging to EL students, the activities are designed to be less wordy and more direct with the steps to complete. Additionally, the activities in these lessons use various visualizations and links to an online dictionary for the content vocabulary.
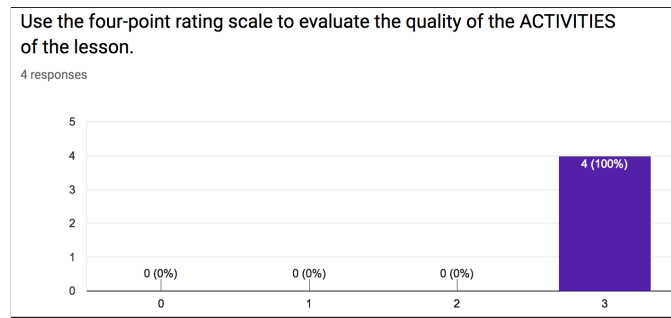


Figure 24: Responses to using the four-point rating scale to evaluate the quality of the activities of the lessons in the Inference/Models section of the ICDS course.

The quality of the wrap-up section in the Inference/Models area of the course is rated as very satisfactory by three experts (75%) and satisfactory by one expert (25%), which is shown in Figure 25. Additional feedback on the wrap-up sections of these lessons noted that it relied heavily on student engagement via classroom discussion. It is recommended to devise and incorporate additional methods to ensure that all students participate in the reflections of the overall lessons.

The quality of the reading and language level in the Inference/Models area of the course is rated as very satisfactory by two experts (50%) and satisfactory by two experts (50%) shown in Figure 26. Although the reading and language level are satisfactory, there are still some improvements that can be made especially to the readings that are not currently differentiated for EL students.

The quality of the visualizations in the Inference/Models area of the course is
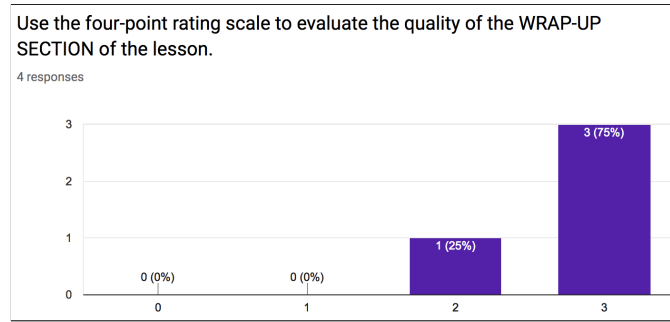
Figure 25: Responses to using the four-point rating scale to evaluate the quality of the wrap-up section of the lessons in the Inference/Models section of the ICDS course.
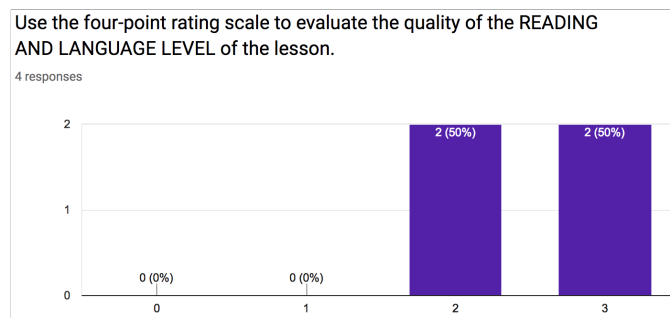


Figure 26: Responses to using the four-point rating scale to evaluate the quality of the reading and language level of the lessons in the Inference/Models section of the ICDS course.

rated as very satisfactory by three experts (75%) and satisfactory by one expert (25%), which is shown below in Figure 27. The qualitative feedback regarding the visualization is fairly positive as these visualizations often serve to support EL students in learning and practicing the content and icons are used consistently throughout all exercises allowing EL students to navigate the material more easily.

### 4.1.6 UNESCO Quality Curriculum Criteria

The panel of experts provided responses to the UNESCO quality curriculum criteria regarding the lessons in the Inference/Model area of the course.

A majority of the experts (75%) strongly agreed that there are clear aims for
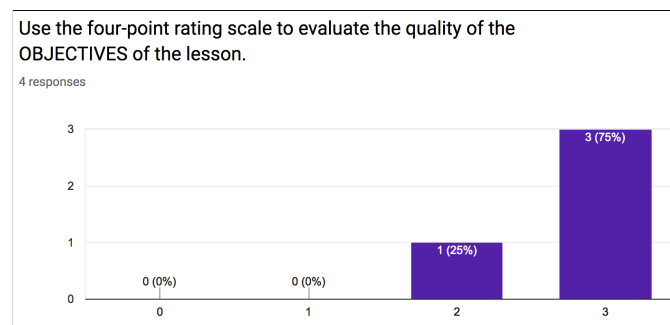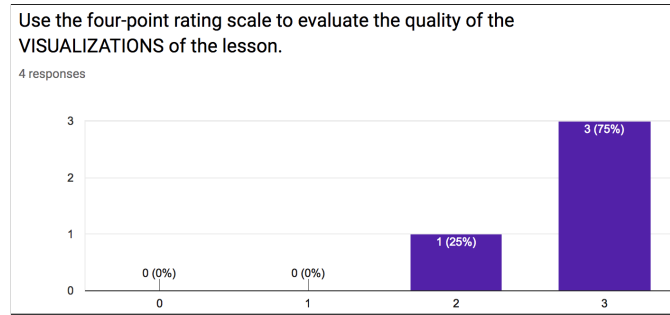
Figure 27: Responses to using the four-point rating scale to evaluate the quality of the visualizations of the lessons in the Inference/Models section of the ICDS course.

the lessons in the Inference/Model area of the course as shown below in Figure 28. There are necessary improvements to be made to the instructional objectives in terms of designing them as measurable outcomes instead of tasks to complete. These modifications will further improve and clarify the aims of the lessons in this section.
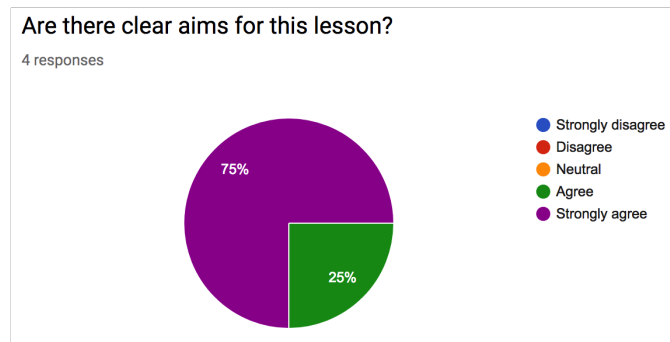

Figure 28: Responses evaluating if there are clear aims for the lessons in the Inference/Models section of the ICDS course.

A majority of the experts (75%) agreed that the content in the Inference/Model area of the course are relevant to the current and future lives, experiences, environments, and aspirations of the students, which is shown in Figure 29. Students are engaging in the utilization of simulations and the scientific method within these lessons. Additionally, students are learning to hypothesize, experiment, analyze their results, and prove/disprove their hypothesis using the data

they obtained from running the simulations. These critical thinking and computational thinking skills are essential to any course and career in which students will be presented with a problem that they will need to solve through experimentation and data analysis to justify their solution. The ability to think critically and justify solutions with evidence is a highly necessary skill that leads to innovative ideas in various industries. Additionally, students are learning how to think computationally in which they learn to use the computer as an effective learning tool, determine the appropriateness to use this tool, and participate in collaborative work that uses this tool. The students achieve this by completing the exercises, learning the skills and producing the artifacts, reflecting and articulating the importance and usefulness of such skills, and applying these skills to other appropriate situations. This may need to be emphasized more to the students so they can clearly understand the purpose of the course and how it connects to the overall computing progressions and outcomes.
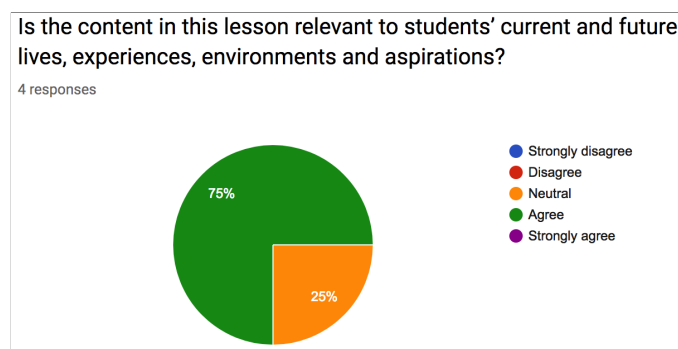


Figure 29: Responses evaluating if the content in these lessons are relevant to students' current and future lives, experiences, environments and aspirations in the Inference/Models section of the ICDS course.

A majority of the experts (75%) strongly agreed that the content in the Inference/Model area of the course are equitable and inclusive as shown in Figure 30. The content in these lessons are designed to introduce computer science to students with varying experiences ranging from none to advanced. As such, the

55

development of EL accessible material in this course allows more equitable learning opportunities to an underrepresented group of students.
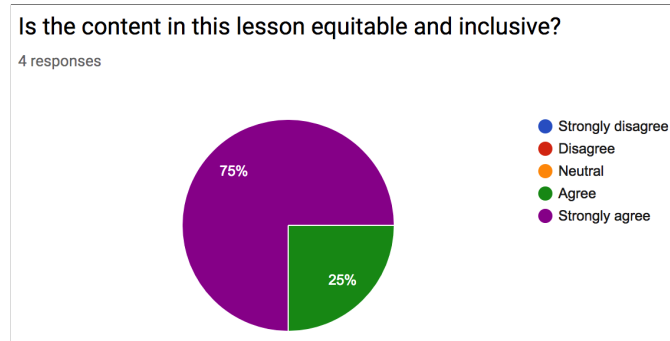


Figure 30: Responses evaluating if the content in these lessons are equitable and inclusive in the Inference/Models section of the ICDS course.

All of the experts (100%) strongly agreed that the content in the Inference/Model area of the course are learner-centered and learner-friendly, and open and flexible allowing it to address new challenges shown in Figures 31 and 32. As noted in the qualitative feedback, the exercises are designed to allow the students to independently work at their own pace. Although students may require clarification on particular tasks in the activities, the students are able to successfully complete these exercises without much dependence on the teacher guiding them through it. This further aids in developing essential computational thinking skills. Additionally, the qualitative feedback from one expert noted that the lessons are designed to be relevant to the needs of EL students through differentiated versions of readings and activities and supplementary vocabulary resources.

A majority of the experts (75%) strongly agreed that the content in the Inference/Model area of the course is coherent and consistent, which is shown in Figure 33. The structure of each lesson is frequently consistent as it begins with a warm-up section introducing the concepts of the lessons through readings and videos, continues with at least one activity where students can apply the concepts and skills they were introduced to, and ends with a wrap-up section engaging students

www.manaraa.com
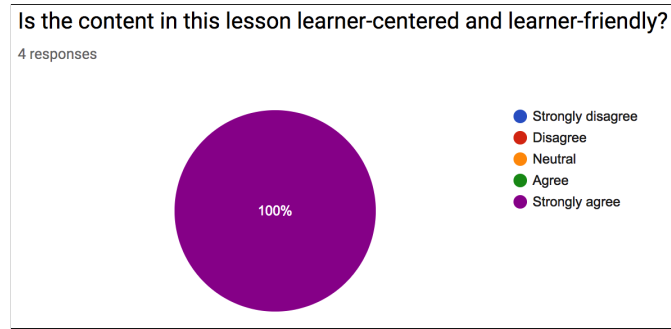
Figure 31: Responses evaluating if the content in these lessons are learner-centered and learner-friendly in the Inference/Models section of the ICDS course.
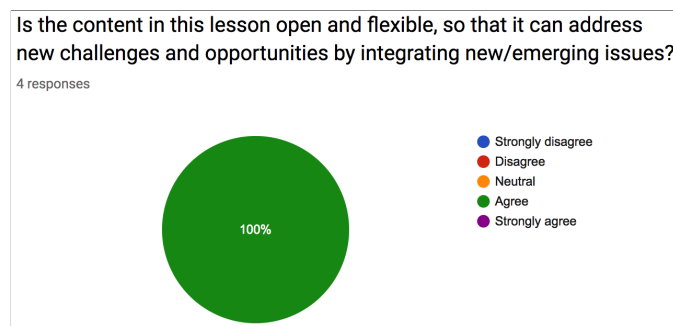


Figure 32: Responses evaluating if the content in these lessons are open and flexible, so that it can address new challenges and opportunities by integrating new/emerging issue in the Inference/Models section of the ICDS course.

in discussion reviewing the concepts and skills they learned. Yet, the number of activities and their length can vary depending on the exercises that are expected to be completed in each lesson. As such, this can affect the consistency of the content in the lessons and students may become overwhelmed with the more lengthy activities after completing briefer ones. This can be improved by compartmentalizing the lengthy activities into smaller and separate activities that build off one another.

## 4.2 Program Development Area of the ICDS Course
### 4.2.1 Instructional Objectives

The instructional objectives in these lessons use scaffolding as well as access prior knowledge and previous learning in the Program Development area of the
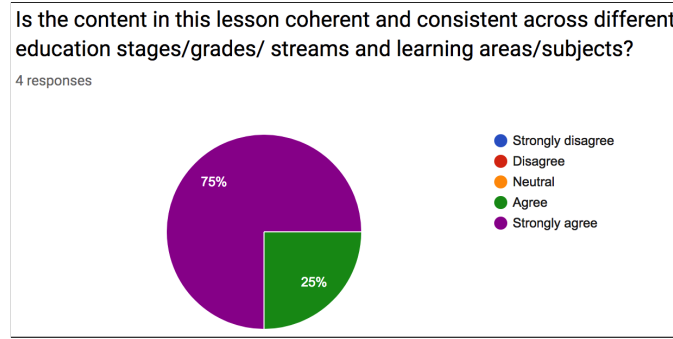
Figure 33: Responses evaluating if the content in these lessons are coherent and consistent across different education stages/grades/ streams and learning areas/subjects in the Inference/Models section of the ICDS course.

course. Additionally, the third and fourth lessons of the Program Development area of the course demonstrate similar qualities in the previous section of the course that allow it to meet the needs and concerns of EL students. Such qualities include clear and understandable objectives, concentration on delivering content, elimination of difficult language, scaffolding, and numerous opportunities to apply and practice the objectives.

The fifth lesson of the Program Development area of the course seems to be the most difficult of the course as it provides less scaffolding and expects students to apply previously learned programming skills to a new program without much guidance. One expert specifically noted that this lesson does not meet the needs and concerns of EL students. This lesson will need to be evaluated and potentially redesigned according to the provided feedback of the expert panel.

### 4.2.2   Strengths of the Language

The strengths of the language emphasized in the content of the lessons in the Program Development area of the course include strong objectives for the third and fifth lessons, adapted readings to support access to the content for EL students, and consistent language used throughout the instructions of the activities. Additionally, the language used in the content of all these lessons is concise, simplistic,

simplified, continuous, and direct for EL students.

### 4.2.3   Weaknesses of the Language

There are several weaknesses in the lessons of the Program Development area of the course that need to be addressed. One weakness is that the fourth lesson of this section only includes reading objectives; therefore, it does not address all language domains (speaking, listening, and writing). This relates to another weakness in which the only opportunities for EL students to engage in interaction is during the brief warm-up and wrap-up sections of the otherwise lengthy instructional lessons. Significant revisions must be made to address these concerns and allow EL students to address all language domains. Specifically, pair programming should be implemented in these lessons where students can be paired with an English-speaking partner to program the simulations in these lessons as well as communicate and collaborate with their partners to reach solutions.

Additional weaknesses include the means of incorporating the vocabulary resources into the content of these lessons. Since these are lengthy activities, there may be time constraints and difficulties implementing the supplementary vocabulary resources within these lessons. There need to be explicit instructions provided to teachers detailing how to incorporate the vocabulary into the lessons. It will be helpful to the students, especially ELs, if the vocabulary is introduced and reviewed before beginning the activities. This will help the students familiarize themselves with the technical jargon and syntax of the NetLogo programming language as well as gain a better understanding of the different types of agents that they will be utilizing in their programming.

The final weakness of the Program Development area of the course is that difficult language still exists in the differentiated versions of activities in the fifth lesson. Specifically, the language in these activities can be confusing in context

and much of the language will need to be deciphered with the assistance of the teacher. The activities in the fifth lesson are lengthy and dense even for non-EL students. As such, these activities should be improved by compartmentalizing the concepts and parts of the activity into smaller and separate activities that build off one another. Additionally, more scaffolding may need to be provided in the differentiated version of the activities to assist EL students in completing the most difficult activities of the unit.

### 4.2.4    General Commentary

Overall, the third and fourth lessons of the Program Development area of the course require some improvements in terms of addressing all language domains, incorporating more interaction in the lengthy programming exercises, and reviewing the vocabulary before progressing through the activities. The fifth lesson of the Program Development area of the course requires significant modifications as it currently does not address the needs and concerns of EL students. As noted, this lesson must be adapted to include more compartmentalized concepts and exercises as well as more scaffolding to guide EL students through difficult programming aspects of the activity.

### 4.2.5    Quality of EL Accessible Materials

The quality of the objectives of the lessons in the Program Development area of the course is rated as very satisfactory by two experts (50%) and satisfactory by two experts (50%) as shown in Figure 34. The qualitative feedback received on the objectives includes favorable responses regarding how these objectives use scaffolding and build upon prior knowledge and skills. Additionally, it is noted in the feedback that the third and fifth lessons of the Program Development area of the course demonstrate strong objectives for the students. Although the objectives

are rated as satisfactory, there are still necessary improvements to be implemented especially concerning the second lesson of this section as it does not address all language domains.



Figure 34: Responses to using the four-point rating scale to evaluate the quality of the objectives of the lessons in the Program Development section of the ICDS course.

The quality of the vocabulary resources in the Program Development area of the course is rated as very satisfactory by all four experts (100%), which is shown in Figure 35. As noted, the vocabulary resources are significant tools that support EL students in learning the content. Additional qualitative feedback stated that explicit instructions incorporating the vocabulary into these lengthy lessons are necessary in order to ensure that these resources are used to support EL students. These vocabulary resources should be used before beginning the exercises in the lessons as it is important to familiarize the students with content and programming syntax they will be using in the activities.

The quality of the EL lesson adaptation footnotes in the Program Development area of the course is rated as very satisfactory by all four experts (100%) as shown in Figure 36. Similar to the Inference/Models area of the course, these adapted footnotes provide leveled supports for EL students through the differentiated versions of the readings and activities as well as explicit instructions on presenting the videos to the students. It is important to make teachers aware of the existence of these adapted materials and their locations. This can be done
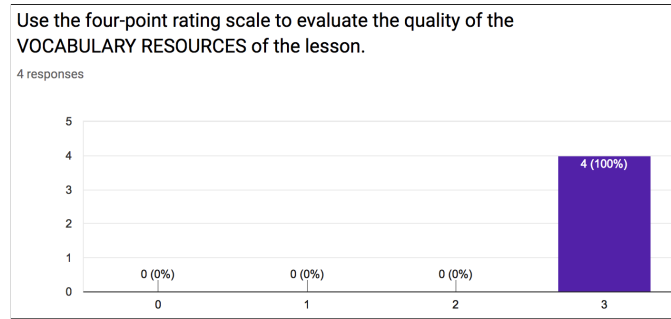
Figure 35: Responses to using the four-point rating scale to evaluate the quality of the vocabulary resources of the lessons in the Program Development section of the ICDS course.

through in-person professional developments for the ICDS course and/or through online communication via email and the community of practice.
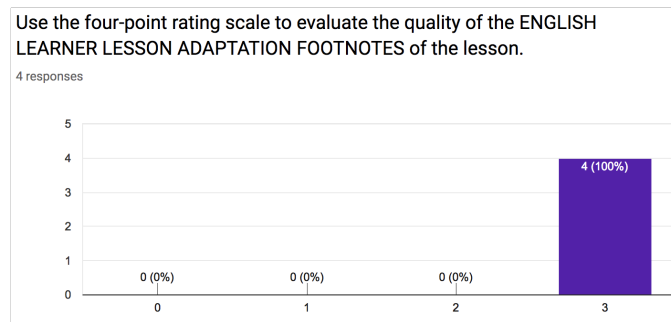


Figure 36: Responses to using the four-point rating scale to evaluate the quality of the EL lesson adaptation footnotes of the lessons in the Program Development section of the ICDS course.

The quality of the warm-up section in the Program Development area of the course is rated as very satisfactory by three experts (75%) and satisfactory by one expert (25%) , which is shown below in Figure 37. The qualitative feedback noted that there are more differentiated versions of the readings in this area of the course allowing EL students further access to the content. Additionally, the warm-up sections provide brief introductions to concepts through readings and videos offering other forms of engagement that do not depend on independent work on the activities.

62
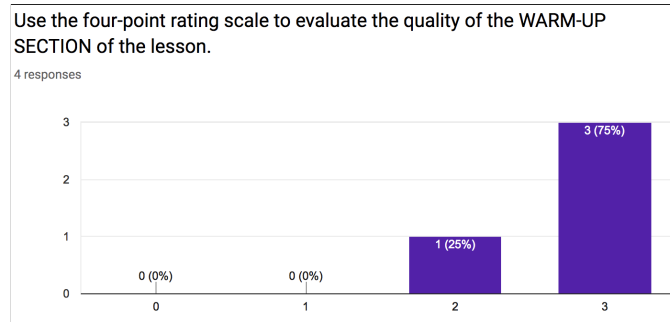
Figure 37: Responses to using the four-point rating scale to evaluate the quality of the warm-up section of the lessons in the Program Development section of the ICDS course.

The quality of the activities in the Program Development area of the course is rated as very satisfactory by three experts (75%) and satisfactory by one expert (25%) shown below in Figure 38. As mentioned in the qualitative feedback, the activities are dense and lengthy in comparison to previous ones and these activities do not offer opportunities for interaction with peers. Rather, students must work independently to complete the activities and do not engage in collaborative work. This requires improvement as EL students are not addressing all language domains and they are mostly engaging in reading. In order to address this issue, these activities should be modified to include more interaction through pair programming in which EL students can participate in the activities with another student (preferably an English speaking student) to practice their oral language skills in addition to their reading and writing abilities.

The quality of the wrap-up section in the Program Development area of the course is rated as as very satisfactory by three experts (75%) and satisfactory by one expert (25%), which is shown in Figure 39. The wrap-up section consists of discussion prompts to reflect on the concepts and skills learned in the lesson but it does not ensure participation of all students. Additionally, this currently serves as the only brief interaction students will engage in after completing the lengthy

63
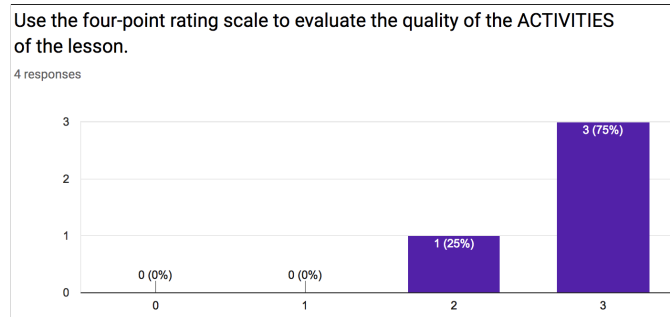
Figure 38: Responses to using the four-point rating scale to evaluate the quality of the activities of the lessons in the Program Development section of the ICDS course.

independent activities. Currently, the wrap-up sections are significant to these lessons due to their interactive discussions but this should be improved upon.
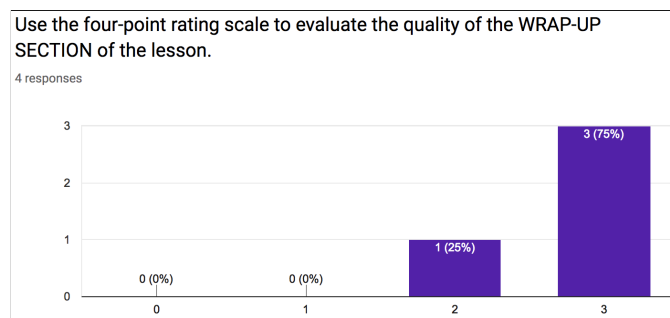


Figure 39: Responses to using the four-point rating scale to evaluate the quality of the wrap-up section of the lessons in the Program Development section of the ICDS course.

The quality of the reading and language level in the Program Development area of the course is rated as very satisfactory by one expert (25%) and satisfactory by three experts (75%) shown in Figure 40. As noted within the qualitative feedback, there is still difficult language present in the activities to be completed by the EL students. Although the language remains consistent throughout the instructions, EL students may still struggle with the language as it is complex and confusing and it requires clarification from the teachers. Additionally, these exercises rely heavily on interpreting and writing programming language syntax based in the English language presenting a potential issue when scaffolding is not

64

provided as demonstrated in the fifth lesson of this section where students program a mouse trap simulation from scratch.
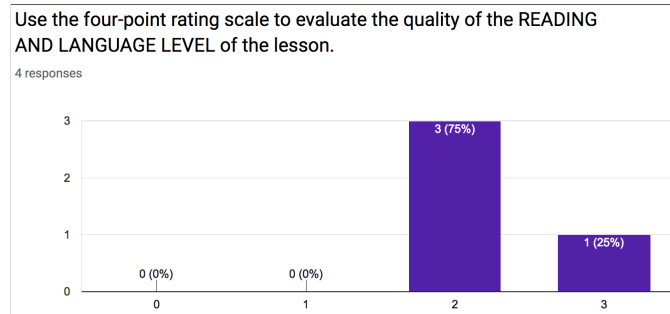


Figure 40: Responses to using the four-point rating scale to evaluate the quality of the reading and language level of the lessons in the Program Development section of the ICDS course.

The quality of the visualization in the Program Development area of the course is rated as very satisfactory by all four experts (100%), which is shown below in Figure 41. The visualizations remain consistent with the lessons in the Inference/Model area of the course as the same icons are used to represent language. Once students reach the lessons of the Program Development area of the course, the EL students should already be familiar with these icons and their language representation to help them more easily navigate these activities.
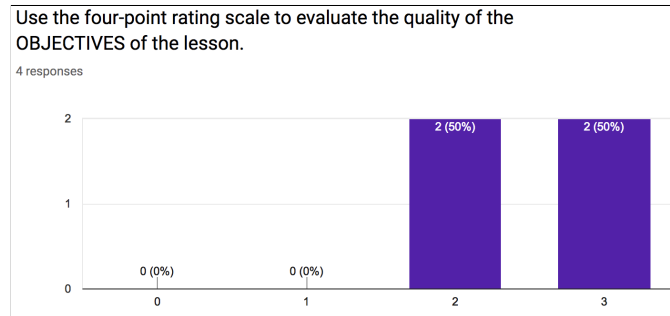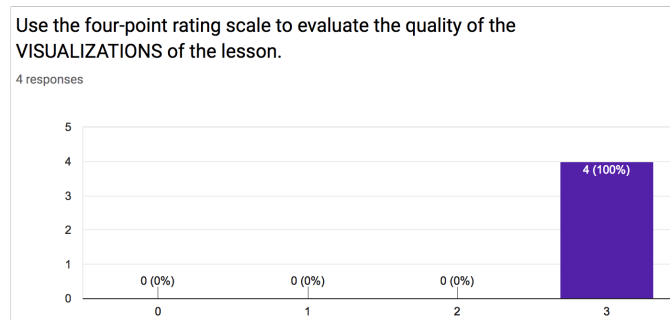


Figure 41: Responses to using the four-point rating scale to evaluate the quality of the visualizations of the lessons in the Program Development section of the ICDS course.

### 4.2.6   UNESCO Quality Curriculum Criteria

The panel of experts provided responses to the UNESCO quality curriculum criteria regarding the lessons in the Program Development area of the course.

There is a split decision among the experts regarding the clear aims for the lessons in the Program Development area of the course as shown below in Figure 42. Half of the experts strongly agreed (50%) that there are clear aims for the lessons while the remaining half of the experts agreed (50%). Although the experts agreed that there are clear aims for the lessons, there are significant improvements that need to be made as noted in the qualitative feedback.



**Are there clear aims for this lesson?**
4 responses

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree
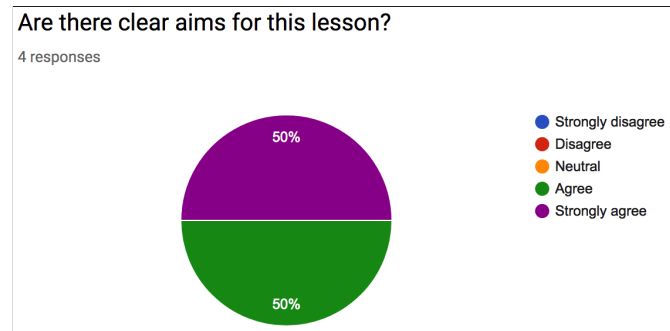
Figure 42: Responses evaluating if there are clear aims for the lessons in the Program Development section of the ICDS course.

All of the experts (100%) agreed that the content in these lessons in the Program Development area of the course are relevant to the current and future lives, experiences, environments, and aspirations of the students, which is shown in Figure 43. These programming lessons establish the essential skills of a computer scientist who learns to use the computer and programming languages as tools to solve simple and complex problems.

A majority of the experts (75%) strongly agreed that the content in the Program Development area of the course are equitable and inclusive as shown in Figure 44. The content in these lessons introduce students to essential programming skills by designing simulations based upon real-world phenomenons such as an ecosys-
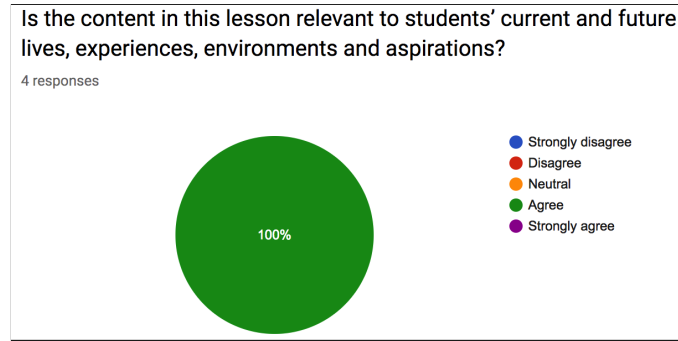
Figure 43: Responses evaluating if the content in these lessons are relevant to students' current and future lives, experiences, environments and aspirations in the Program Development section of the ICDS course.

tem where turtles live, eat, and die. These foundational programming skills can be applied in almost any programming language with the exception of different syntax. Additionally, students can learn to program their own simulations to test and experiment with their theories while collecting data as evidence to justify their conclusions.
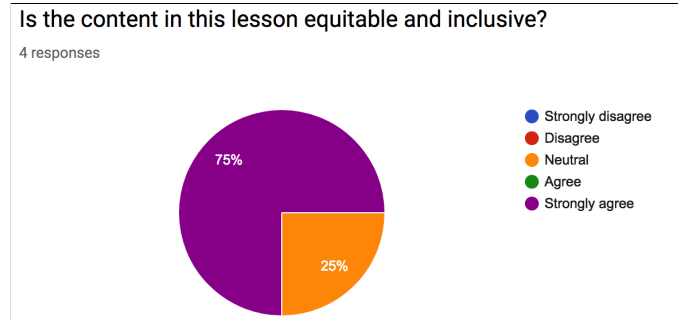


Figure 44: Responses evaluating if the content in these lessons are equitable and inclusive in the Program Development section of the ICDS course.

All of the experts (100%) strongly agreed that the content in the Program Development area of the course are learner-centered and learner-friendly while a majority of the experts (75%) agreed that the content is open and flexible allowing it to address new challenges shown in Figures 45 and 46. The third and fourth lessons provide plenty of scaffolding when teaching the students how to program simulations for the first time. The fifth lesson, however, provides the least amount

of scaffolding and it is more difficult as it expects students to apply previously learned skills to program an original simulation. As reported in the qualitative feedback, this is the most difficult lesson that is not well received by most students as it is intimidating and overwhelming due to its density, length, complexity, and difficulty. This lesson require significant improvements in terms of compartmentalizing the concepts and exercises into smaller and separate activities. Additionally, the simulation programming language does not currently have a simplified debugger system with clear error messages detailing the line number and error. This makes it even more difficult for EL students to determine the errors within their code and how to resolve these errors. As such, a simplified debugger detailing the errors and utilizing visualizations should be developed for EL students to use when they encounter an error.
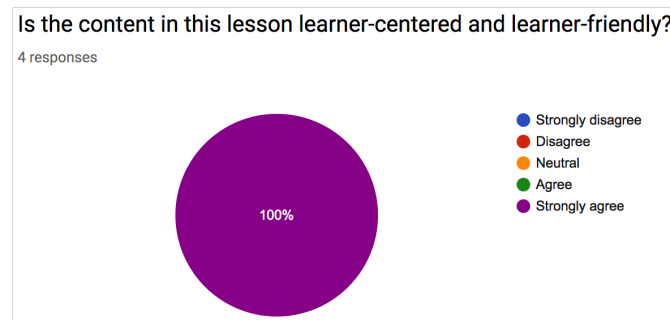


Figure 45: Responses evaluating if the content in these lessons are learner-centered and learner-friendly in the Program Development section of the ICDS course.

Half of the experts agree (50%) that the content in the Program Development area of the course is coherent and consistent, which is shown in Figure 47. The remaining experts either strongly agreed or chose to remain neutral regarding the coherency and consistency of the content. The content within these lessons are consistent when examining the language and visualizations used in the activities as the same language in the instructions and icons to represent language are used throughout all the lessons. Yet, the scaffolding provided throughout the first two
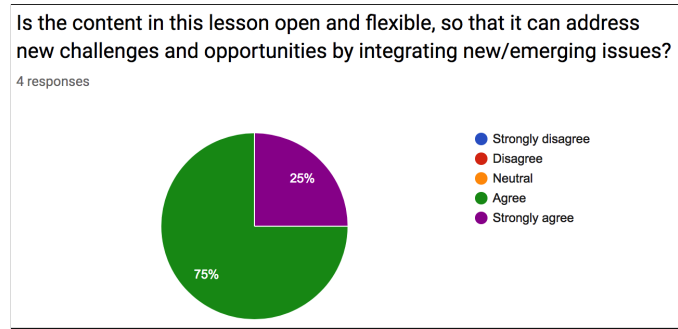
Figure 46: Responses evaluating if the content in these lessons are open and flexible, so that it can address new challenges and opportunities by integrating new/emerging issue in the Program Development section of the ICDS course.

lessons is drastically reduced in the third lesson of the Program Development area of the course. Students are expected to apply prior knowledge and skills to code their own mouse trap simulation but less guidance is provided as students should begin to get comfortable writing their own code. This unexpected shift from plenty to limited scaffolding when learning to program disrupts the pacing and progress towards programming the final simulation and it makes the final lesson difficult and overwhelming for EL students to complete.
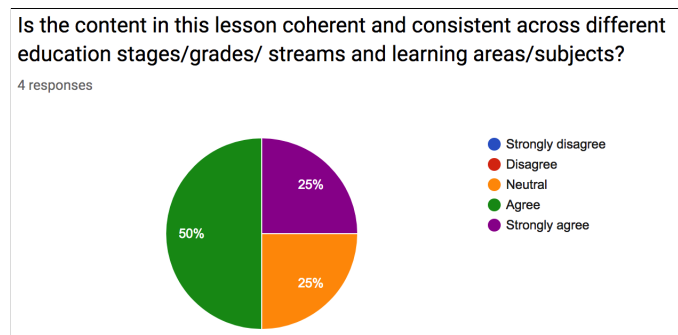


Figure 47: Responses evaluating if the content in these lessons are coherent and consistent across different education stages/grades/ streams and learning areas/subjects in the Program Development section of the ICDS course.

69

## 4.3 Revisions to Programming and Simulation Unit
## 4.3.1 Program Your Own Simulation

A majority of the experts stated in the qualitative feedback that the fifth lesson, Program Your Own Simulation, in the Program Development area of the course is the most difficult for all students; however, EL students struggle significantly in this lesson as the scaffolding in this lesson is drastically reduced from previous lessons. The initial justification for minimizing the scaffolding was the expectation that students would build their simulation program independently using prior knowledge and skills from the previous programming lesson. Yet, this lesson is extremely lengthy, complicated, and difficult for students to complete independently. The lack of scaffolding in this lesson overwhelms and intimidates most students especially EL students who are struggling with the wordiness, complex content vocabulary, and complicated programming syntax. Additionally, EL students are not provided opportunities to interact and communicate with other students throughout this lesson due to its heavy reliance of independent programming. As such, EL students are not able to address all language domains, specifically listening and speaking, and do not have the opportunity to practice their oral skills using the current content and materials.

Significant revisions have been made in the Program Your Own Simulation lesson in the Program Development area of the course. In terms of density and length, the main activity has been compartmentalized into smaller and briefer activities that build off one another to produce the final mouse trap simulation program. More scaffolding is provided to students within these activities to address the difficulty, inconsistency and challenge of completing the programming exercises in this lesson. Additionally, the EL lesson adaptation footnotes contain more detailed and explicit instruction regarding these activities to guide these students through and explain step-by-step each exercise. If an instructor does not have available

70

classroom time to guide students through the exercises, then the instructor can pair EL students with bilingual students who can translate the steps or non-EL students with strong English language communication skills as recommended by [3]. These footnotes also include instructions on pairing EL students with a non-EL partner in this lesson to engage in collaborative learning as they progress through the exercises as well as a provide an opportunity to practice listening and speaking language domains. These revisions address all the qualitative feedback received on the Program Your Own Simulation lesson in terms of compartmentalizing, scaffolding, incorporating collaboration and interaction, and addressing all the language domains in the content/language objectives and activities.

### 4.3.2    NetLogo Debugger

The NetLogo programming language causes more challenges when a student develops an error within their code. Unlike a typical debugger, NetLogo will not report the line the number the error appears on nor will it provide a clear message describing the issue with the program such as if its a syntactic, semantic, or logical error as shown in Figure 48. It will produce a vague and complex error message that does not direct students on how to clearly resolve their errors. Additionally, these errors contain complicated wording and vocabulary that can be especially difficult for an EL student to interpret and resolve by examining each line of code in their program since a line number is not given. The current structure of this final lesson makes it challenging for EL students to meet their content and language objectives as well as succeed in developing the foundation of their programming skills.

In response to the feedback regarding the difficulty of the programming in NetLogo, I developed an online interactive debugger program to assist students with resolving the errors in their code. This online debugger requests that stu-

71
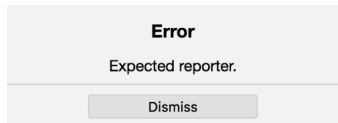
(a) A NetLogo error resolved by replacing parentheses with square brackets within code.



(b) A NetLogo error resolved by replacing square brackets with parentheses within code.

Figure 48: The error messages provided by the NetLogo programming language to help debug code.

dents submit their code into a numbered text box on the left side of the page labeled *NetLogo Program*. Students can copy and paste their code into this text box and then click the debug icon which is located next to the word *Debug!* to have the debugger program cross-check their coded solution against the correctly programmed solution that is provided to teachers in the answer keys. If an error is found, it will produce their debugged program in the box on the right side highlighting in red and crossing out the error in their code and producing the correct line of code in green. Additionally, it will state the line number that the error occurs on at the bottom of the page in their code. Using this debugger program, students can locate the error in their code as well as implement the solution to resolve this error within their code. This online debugger is shown in Figure 49 demonstrating the error in the code in red followed by the correction in green.

This online debugger also incorporates several techniques and tools that are useful to EL students. There is significant emphasis on the visualizations used in the interactive NetLogo debugger including the icons to represent actions and the highlighted colors to represent errors and solutions. In terms of icons, there are several large and universal icons that are used to represent actions in the debugger program. These icons also include a hover element in which hovering over the icon will result in a pop-up appearing that shows the English word describing the action that will occur if clicked as shown in Figure 50.

72
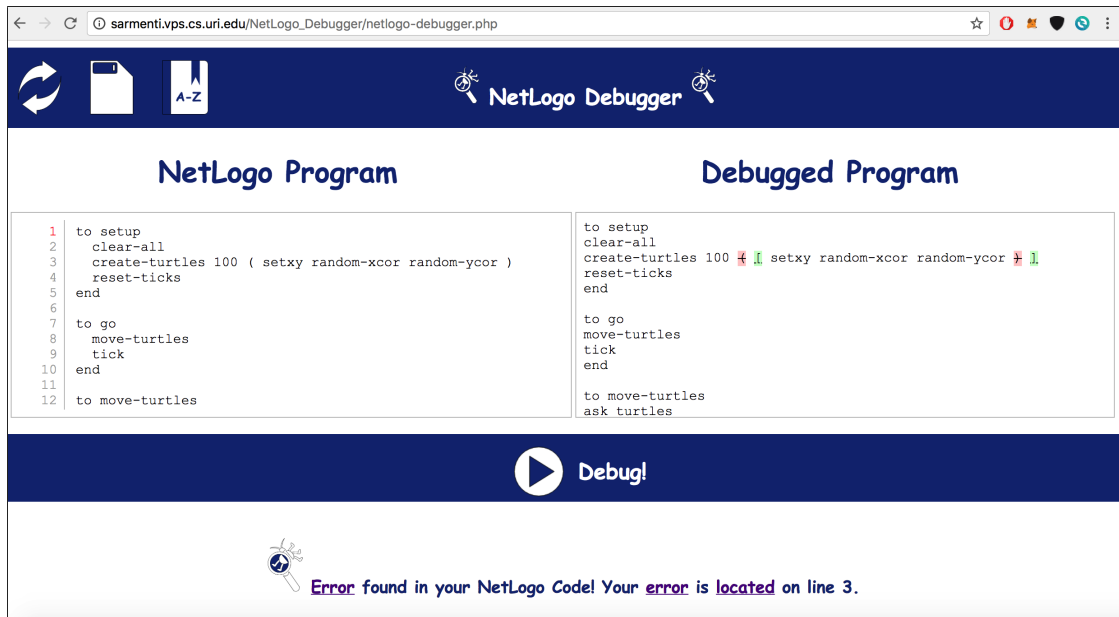
Figure 49: The online NetLogo debugger showing the error on line three in which parentheses need to be replaced with square brackets.

The debug icon shown in Figure 50a will run the program that debugs their code and produce the debugged program in the box on the right side. It is important to note that the debugger will stop checking their submitted code once an error is located. This means students should correct their code using the solution and line number provided by the debugger and then run the debugger program again to locate the next error in their code until the debugger no longer finds any errors reporting this at the bottom of the page.

The refresh icon as shown in Figure 50b will refresh the NetLogo debugger page and clear the existing submitted code from the text box as well as the debugged program. Additionally, it will show a message at the bottom of the page that no errors have been found because no code has been submitted yet. Figure 51 shows the output of the NetLogo debugger program after clicking the refresh icon.

When the copy icon shown in Figure 50c is clicked, the NetLogo debugger will copy the current code in the editable text box on the left to the clipboard of their

(a) Debug Code icon


(b) Refresh debugger program icon


(c) Copy Code icon


(d) Open NetLogo Dictionary icon
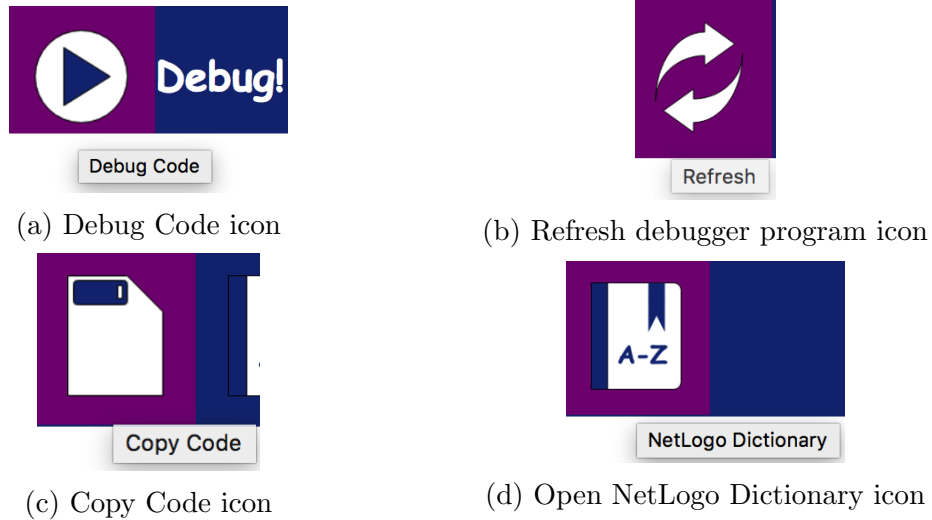
Figure 50: The icons used to represent various actions such as debug, refresh, copy, and open dictionary in the NetLogo debugger program. It also shows the text that appears when the cursor hovers over the icon.

computer. As stated previously, students are expected to modify and correct their code in this text box until the debugger program returns that there are no errors in their code in its output. Once students have corrected all of their code, they can then copy their code to the clipboard of their computer and paste it into NetLogo as shown in Figure 52.

The final icon shown in Figure 50d represents a web link to the NetLogo dictionary. If this icon is clicked, it will redirect the user to the NetLogo Dictionary webpage which contains all the built-in keywords and functions of the NetLogo programming language. This is useful as students can find common keywords and functions that they may use when programming their simulations in NetLogo as well as gain a better understanding of their features through their definitions. Additionally, students can look up the proper spelling of these built-in features to ensure that they do not have typos in within their code. Additional links to an online dictionary are used for EL students in defining difficult vocabulary used in the debugger program such as error which will contain a hyperlink.
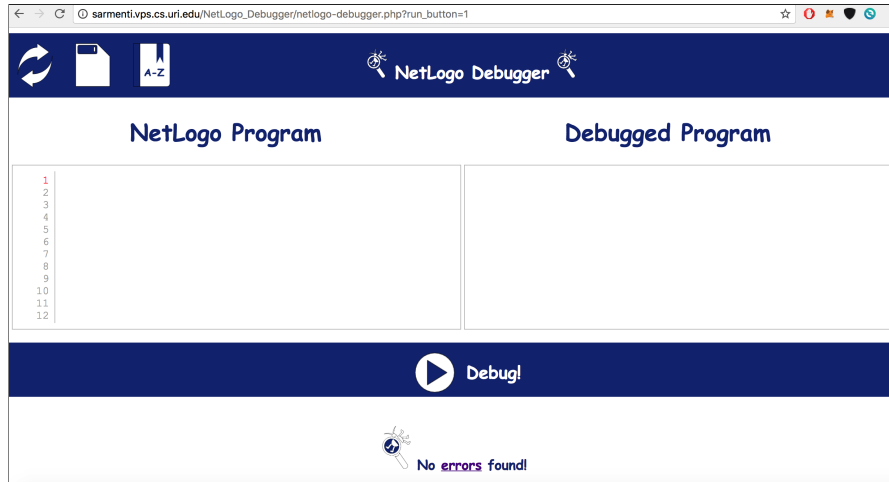
Figure 51: The online NetLogo debugger after clicking the refresh icon.

The highlighting of the errors and corrections is the final visualization technique used to assist EL students in locating and resolving their errors. The examples shown in Figure 53 demonstrate the various cases in which errors can be found and resolved using the NetLogo debugger program.

In addition to the NetLogo debugger program, I developed a NetLogo debugger guide with common errors to address the error messages that the original NetLogo debugger provides. Within this guide, students are given a hyperlinked table of content of common errors that they can click on to direct them to the section in the guide that the particular error is discussed. This is shown in Figure 54.

The guide will show an image of the exact error message produced in the original NetLogo debugger as well as example code cases that may have caused the error and how to correct them. Additionally, it includes written justification pertaining to how and why the error occurred. Examples of these are shown in Figure 55. Note that these cases use visualizations similar to the online NetLogo debugger to remain consistent as well as assist EL students in recognizing the visual cues to resolve their errors within their code.
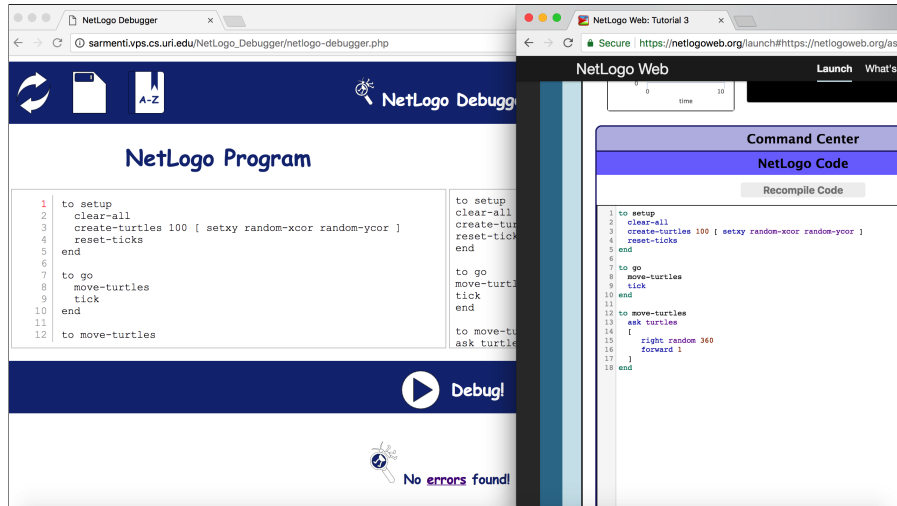
Figure 52: The online NetLogo debugger after clicking the copy icon and pasting copied code into NetLogo.

## List of References

[1] O. P. Veniegas. "Ensuring the effectiveness and quality of learning materials: How to use and evaluate literacy/ ce materials." n.d. [Online]. Available: http://www.accu.or.jp/litdbase/pub/dlperson/pdf0106/rpp24_1.pdf

[2] P. Stabback. "What makes a quality curriculum?" 2016. [Online]. Available: http://unesdoc.unesco.org/images/0024/002439/243975e.pdf

[3] D. Barlow, "The sourcebook for teaching science: Strategies, activities, and instructional resources (grades 6-12)," *The Education Digest*, vol. 74, no. 6, p. 72, 2009.

(a) Incorrect procedure name.    (b) Missing hyphen for predefined function.



(c) Missing bracket to close command.    (d) Missing hyphen for predefined function.

Figure 53: Examples showing the use of the NetLogo debugger program.



**NetLogo: Debugging Common Errors**

| | |
|---|---|
| SET expected 2 inputs. | 1 |
| Nothing named ENERGY-1 has been defined. | 1 |
| Expected Command. | 2 |
| Keyword Expected | 2 |
| END Expected | 3 |
| Opening Bracket Expected | 4 |
| Closing Bracket Expected | 4 |
| Expected Reporter | 5 |
| Expected a literal value. | 5 |
| IF expected 2 inputs, a TRUE/FALSE and a command block. | 6 |
| Nothing named == has been defined. | 6 |
| There is already a procedure called SETUP | 7 |
| Nothing named SETUP-TURTLES has been defined. | 8 |
| Nothing named SETUP-PATCHES has been defined. | 9 |

Figure 54: The NetLogo debugging guide with a list of common errors and links to their section.

77
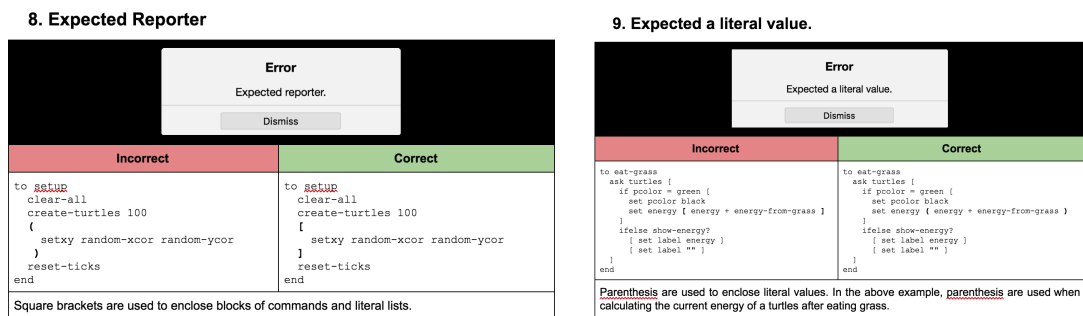
(a) A NetLogo expected reporter error shown in the debugger guide.



(b) A NetLogo literal expected error shown in the debugger guide.

Figure 55: The error messages provided by the NetLogo programming language to help debug code.

# CHAPTER 5

## Conclusions and Future Work

### 5.1 Conclusions

The first research question of this study is concerned with the ways in which the instructional objectives of the Inference/Models and Program Development areas of the ICDS course are meeting the needs and concerns of EL students. According to the responses gathered from the panel of experts, a majority of the lessons in both focus areas of the Programming and Simulation unit include instructional objectives that meet the needs and concerns of EL students especially through the development of content and language objectives. In the Inference/Models area of the ICDS course, the instructional objectives of the two lessons are designed to be understandable to EL students regardless of English level proficiency and focus on delivering content to these students. Additionally, the instructional objectives in the Inference/Models area of the course provide EL students with numerous opportunities to practice and apply these objectives. Yet, there are issues with the current instructional objectives for the Inference/Models area of the course pertaining to the lack of explicit definition of the methods to be used to measure the language objectives. The language objectives for these lessons are not defined as measurable outcomes but rather tasks to be completed. This will make it difficult for instructors to measure computer science learning and achievement among EL students as well as if these students are understanding the content. As for the Program Development area of the ICDS course, there are strong instructional objectives for the third and fifth lessons but all the lessons in this section effectively use scaffolding and access prior knowledge; however, the fifth lesson in this unit is the most difficult and required reconstruction to fully address the needs and concerns of EL students.

79

Within this study, the second research question pertains to the evaluating the quality of the materials in among the Inference/Models and Program Development area of the ICDS course as well as which of these areas EL students would demonstrate the highest level of computer science learning and achievement. As noted by the feedback, both areas of the Programming and Simulation unit included materials and resources adapted for EL accessibility that are deemed at least satisfactory in terms of quality. Additionally, both areas of this unit met the UNESCO quality curriculum criteria as the panel of experts agreed that these areas had clear aims for each lesson and the content included qualities of relevance, inclusion, learner-centered education, flexibility, and coherency. The quality of the materials in the Inference/Models and Program Development of the course produce significant impact in terms of computer science learning and achievement among EL students by achieving its instructional objectives, obtaining and maintaining interest, engaging prior knowledge and experience, and being reliable. According to the feedback of instructors enacting these materials, EL students demonstrated the highest level of learning in the Inference/Models area of the course due to its visual emphasis on the simulations and use of prior knowledge involving the scientific method. The EL students struggled in the Program Development unit especially in the fifth lesson as scaffolding was reduced and it contained lengthy and complicated programming exercise. As a result of this, the fifth lesson was redesigned and compartmentalized to include more scaffolding as well as a visual debugger to help EL students resolve common errors they would encounter in their code.

The final research question of this study consists of determining the effect English language proficiency may have on the ability to learn programming syntax. Although EL students struggled with the fifth lesson of the Program Development area of the course, it is unclear if these students had difficulty due to the program-

ming content or the prior ineffective and dense construction of the activities lacking scaffolding. Additionally, feedback from instructors enacting the fourth lesson of the Programming Development, which includes extensive programming but also increased scaffolding, did not note any significance of struggles with programming among their EL students and determined that this lesson did meet the needs of EL students. As such, it is inconclusive if English language proficiency has any effect on the ability of EL students to learn programming syntax.

## 5.2 Future Work

Although significant revisions have been made to the Programming and Simulation unit of the ICDS course, there are still necessary improvements to implement to enhance the learning experience of computer science education among EL students. Such revisions include updating the instructional objectives (content and language) to reflect measurable learning outcomes rather than tasks to be completed, updating the language in these activities to be more simplified and condensed, redesigning lessons to include more interaction and collaboration rather than independent work, and modifying the wrap-up sections to incorporate other methods of participation in reflections. Additionally, the tools used in the Programming and Simulation unit to assist in EL computer science learning and achievement must be implemented in the other two units (Data and Python) of the ICDS course.

An expansion of this study would include enacting a pretest and posttest of the material to measure computer science learning and achievement among EL students using the adapted curriculum and materials. This would help determine if the materials are making an impact on their learning and achievement in the ICDS course. Additionally, if the materials are effective with EL students then these strategies and techniques could be implemented in other computer science

81

courses offered in the K-12 public education system.

A final note concerning future work for this study includes developing a NetLogo debugger tool that does not rely on cross-checking submitted code against the teacher solution. Rather, this debugger would verify that the submitted code uses proper syntax in the NetLogo programming language as well as check for semantic and logical errors. This debugger could also use unit testing to account for infinite variations of coded solutions and ensure that submitted code is working properly by verifying the expected output against the output of the solution.

# APPENDIX

## Appendix A



# Lesson 1: Introduction To Computing And Data Science

## Overview

Students will be introduced to the Intro To Computing course by watching a series of videos relating to computational thinking and data science. Additionally, students will participate in an unplugged activity released by Code.org in which they will need to use computational thinking and collaboration in their groups to draw a specific monster.

## Purpose

The purpose of this lesson is to introduce students to core concepts of the course in a fun and engaging way. More specifically, students will be given an introduction to computational thinking and data science through a series of videos that explain these concepts in concise and meaningful way. Students will then participate in an unplugged activity in small groups where students will draw a specific monster based upon the instructions of other students. At the end of this lesson, students will have gained an understanding of the importance of computational thinking and data science that will be utilized throughout the course.

## Agenda

**Warm Up (25 min)**
> **Let's Watch! Computational Thinking**
> **Let's Watch! Data Science**

**Activity (30 min)**
> **Let's Play! Unplugged Computational Thinking Activity**

**Wrap Up (5 min)**
> **Let's Talk! Why Do We Care?**

## Objectives

*Content Objectives*

*Students will be able to:*

- Define computational thinking and identify its real-world applications.
- Define data science and identify its real-world applications.
- Use collaboration in a group to solve complex problems.
- Use modeling and simulations to represent phenomenons.
- Control NetLogo models with buttons, speed slider, sliders and switches.

*Language Objectives*

*Students will be able to:*

- *Listen* to and *orally discuss* the definition of computational thinking and its real-world applications.
- *Listen* to and *orally discuss* the definition of data science and its real-world applications.
- *Listen* and *orally discuss* solutions to complex problems in a group.
- *Read descriptions*, *explore*, *ask* and *answer questions in writing* using modeling and simulations to represent phenomenons.

83

- *Read descriptions* to control NetLogo models with buttons, speed slider, sliders and switches.

## Preparation

**For the student:**

- ❏ Monster Catalog (1 per group)
- ❏ Blank pieces of paper (3 per person)
- ❏ Markers, pens, or pencils (1 pack per group)
- ❏ Scissors

**For the teacher:**

- ❏ A computer with working Internet access to various websites such as Google Products (Google Docs, YouTube, etc.)
- ❏ A Gmail account, either personal or school account
- ❏ Print monster images and cut monster parts out for each group of students
- ❏ Bring in tracing paper, markers and scissors for each group of students

## Vocabulary

- Vocabulary Guide
- Vocabulary Cards
- Vocabulary Slideshow With Audio
- Vocabulary Slideshow
- Vocabulary Wordsearch
- Vocabulary Crossword
- Vocabulary Word Wall
- Vocabulary Quizziz

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** Replace above video with Computer Science Changes Everything [5:34 min].
[2] **English Learners:** Video contains complex vocabulary. If used, you will need to run it slow with many stops.
[3] **English Learners:** Take students through an example.

## Links

**For the Students:**

- One copy of [Introduction To NetLogo Activity Guide](#)
- [NetLogo Controls Layout](#)

**For the Teacher:**

- [Unplugged Activity: Computational Thinking](#)
- [Computational Thinking Lesson Plan](#)
- One copy of [Introduction To NetLogo Activity Guide](#)
- [NetLogo Controls Layout](#)

## Support

**Report a Bug**

- Are the videos engaging? Do the videos convey the concepts well to your students?
- Is a reasonable amount of time given to students to complete the activity?

**[Give us feedback!](#)**

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** Replace above video with [Computer Science Changes Everything](#) [5:34 min].
[2] **English Learners:** Video contains complex vocabulary. If used, you will need to run it slow with many stops.
[3] **English Learners:** Take students through an example.

# Teaching Guide

## Warm Up (25 min)

### Let's Watch! Computational Thinking

🎙️ *Remarks*

Welcome to the Intro To Computing course! In this course, you will learn various skills and concepts relating to computational thinking and data science. Now, I realize that many of you may be unfamiliar with computational thinking and data science. That is why we will watch a couple videos that can help you understand and visualize these concepts. As we watch these videos, I would like you to keep the following questions in mind:

💬 *Let's Think!* What is computational thinking? Why is it important?

[1] **Watch:** As a class, watch <u>Kids in a Tech World: A Million Opportunities</u> [2:46 min].

[2] **Watch:** As a class, watch <u>Solving Problems at Google Using Computational Thinking</u> [3:43 min].

🎙️ *Remarks*

After watching the last two videos, we realize that computational thinking is a problem-solving method that computer scientists use for their solutions to digital problems. Computational thinking is important because it helps computer scientists break larger complex problems into smaller pieces that can be more easily solved. The most important thing is that computational thinking can be applied to anything! Think some ways that you might already use computational thinking in your classes without even knowing it. Do you think you use it in your science classes? Maybe even in english classes? I bet we could even think of ways to use computational thinking in your art classes!

### Let's Watch! Data Science

🎙️ *Remarks*

Now that we have a better understanding of computational thinking, we can take a look at data science. We will watch a couple videos that can help you understand and visualize data science. As we watch these videos, I would like you to keep the following questions in mind:

💬 *Let's Think!* What is data science? What is a data story? Why are they important?

[2] **Watch:** As a class, watch <u>The Data Scientist - 60 Second Data Science</u> [1:48 min].

**Watch:** As a class, watch <u>Jackson: An IBM Data Story</u> [2:30 min].

🎙️ *Remarks*

After watching the last two videos, we realize that data science is using data to solve problems, discover patterns, and gain insight. Data science is important because we have so much data generated by a lot of people across the world that we need to make sense of it and figure out what we can do with it all.

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** Replace above video with <u>Computer Science Changes Everything</u> [5:34 min].
[2] **English Learners:** Video contains complex vocabulary. If used, you will need to run it slow with many stops.
[3] **English Learners:** Take students through an example.

Let's think about the video with Jackson for instance. For over thirty years, Guiding Eyes for the Blind has thoroughly collected data sets of all manner guide-dogs consisting of complex genetic mapping, medical records, and journals and logs from thousands of puppy raisers, fosters parents and trainers to improve the success rate of guide dog breeding. So they had thirty years worth of data! With all this data, they applied data science to it in order to find trends to help improve the success of guide dog breeding. As a result of their findings, they created and published a data story about one of their guide dogs, Jackson. Pretty cool, right?

In this course, you will be building your computational thinking and data science skills through the various activities we complete. Additionally, you will create your own data story about a topic you are interested and use a real data set to support it. My hope is that you walk away from this course with positive feelings towards computer science as well as essential skills that you can utilize in not only any class but also any career you choose in the future.

## Activity (30 min)

### Let's Play! Unplugged Computational Thinking Activity

🎙 *Remarks*

We will start our course off with a fun activity that can help us build our computational thinking skills. I know it may seem silly at first but I want you guys to have fun with this activity! I will be breaking you all into small groups and you will need to work together to help solve the problem. I will review more of the instructions for this activity as I pass out the materials you will need.

**Distribute:** Each group of students should receive the following materials for this activity:

- Monster Catalog (1 per group) on page 7 of Computational Thinking Lesson Plan
- Blank pieces of paper (3 per person)
- Markers, pens, or pencils (1 pack per group)
- Scissors

[3] **Support:** Allow students time to work on the assignment in their groups. As you circulate the classroom, assist any students that may have questions or issues that need to be resolved as they work through the activity.

**Transition:** As the class time comes to an end, try to bring students attention back to you to have a final discussion about computational thinking and data science.

## Wrap Up (5 min)

### Let's Talk! Why Do We Care?

🎙 *Remarks*

I hope you all had fun working on the first activity of this course. I can see that many of you demonstrated good computational thinking skills in this activity. This only a preview of what is to come in this course. I just want to quickly reiterate that computational thinking and data science will be used in every activity of this course as we look at simulations and data sets throughout the semester. I will see you all in the next class and be ready to examine some simulations!

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** Replace above video with Computer Science Changes Everything [5:34 min].
[2] **English Learners:** Video contains complex vocabulary. If used, you will need to run it slow with many stops.
[3] **English Learners:** Take students through an example.

# Lesson 2: Modeling And Simulation

## Overview

This unit presents basic concepts in modeling complex systems through computer programming of agent-based simulations using NetLogo. It uses NetLogo Web, which is a free web-based programming environment for simulations. The site comes with hundreds of pre-defined models of things like the spread of disease, the effect of clouds on global warming, etc. The student looks at the existing simulations, tests hypothesis by experimenting with parameters (e.g. the number of sheep) to come to make observations.

## Purpose

The purpose of this lesson is to introduce students to the NetLogo interface, which they will be using throughout the semester. Student will learn that the simulations serve as representations of phenomenons within our world. With these simulations, students will learn to analyze these phenomenons and construct hypotheses about the outcomes of the simulation. Students will test their hypotheses by adjusting the various parameters of the simulation.

## Agenda

**Warm Up (10 min)**
> **Let's Read! Computing Modeling and Simulation**
> **Let's Watch! Intro To NetLogo**

**Activity (80 min)**
> **Let's Play! Modeling And Simulation Activity**

**Wrap Up (10 min)**
> **Let's Talk! Why Do We Care?**

## Objectives

### Content Objectives

*Students will be able to:*

- Use modeling and simulations to represent phenomenons.
- Use models to formulate, refine, and test hypothesis.
- Explain that models are programmed.
- Control NetLogo models with buttons, speed slider, sliders and switches.
- Gather information from NetLogo models with plots and monitors.

### Language Objectives

*Students will be able to:*

- *Read descriptions*, *explore*, *ask* and *answer questions in writing* using modeling and simulations to represent phenomenons.
- *Ask* and *answer questions in writing* using models to formulate, refine, and test hypothesis.
- *Orally explain* and *summarize in writing* that models are programmed.
- *Read descriptions* to control NetLogo models with buttons, speed slider, sliders and switches.
- *Read* and *explain in writing* information gathered from NetLogo models with plots

and monitors.

## Preparation

**For the student:**

- ❏ A computer with working Internet access to various websites such as Google Products (Google Drive, Google Docs, YouTube, etc.), and NetLogo
- ❏ A Gmail account, either personal or school account
- ❏ One copy of the Modeling And Simulation - Activity Guide
- ❏ One copy of the Party Simulation - Activity Guide

**For the teacher:**

- ❏ A computer with working Internet access to various websites such as Google Products (Google Docs, YouTube, etc.)
- ❏ A Gmail account, either personal or school account
- ❏ One copy of the Modeling And Simulation - Activity Guide
- ❏ One copy of the Party Simulation - Activity Guide

## Vocabulary

- Vocabulary Guide
- Vocabulary Cards
- Vocabulary Slideshow With Audio
- Vocabulary Slideshow
- Vocabulary Wordsearch
- Vocabulary Crossword
- Vocabulary Word Wall
- Vocabulary Quizziz

**English Learner Lesson Adaptation Footnotes:**
**[1] English Learners:** English Learner version of reading—Computing Modeling And Simulation—available.
**[2] English Learners:** English Learner version of activity guide—Modeling And Simulation - Activity Guide—available. Step seven of this activity guide is optional for English Learners.
**[3] English Learners:** English Learner version of activity guide—Party Simulation - Activity Guide—available.

## Links

**For the Students:**

- One copy of the [Modeling And Simulation - Activity Guide](#)
- One copy of the [Party Simulation - Activity Guide](#)

**For the Teacher:**

- [Modeling And Simulation Conceptual Quiz](#)

- [Modeling and Simulation - Activity Guide Rubric](#)
- [Party Simulation - Activity Guide Rubric](#)

- [Answer Key: Modeling And Simulation - Activity Guide](#)
- [Answer Key: Party Simulation - Activity Guide](#)

## Support

**Report a Bug**

- Are the simulations engaging? Is enough guidance provided for students to complete the activity?
- Is a reasonable amount of time given to students to complete the activity?

**[Give us feedback!](#)**

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** English Learner version of reading—[Computing Modeling And Simulation](#)—available.
[2] **English Learners:** English Learner version of activity guide—[Modeling And Simulation - Activity Guide](#)—available. Step seven of this activity guide is optional for English Learners.
[3] **English Learners:** English Learner version of activity guide—[Party Simulation - Activity Guide](#)—available.

# Teaching Guide

## Warm Up (10 min)

Let's Read! Computing Modeling and Simulation

🎤 *Remarks*

> In the last class, we learned a little bit about computational thinking and data science. Recall that computational thinking is breaking larger complex problems down into smaller pieces that can be more easily solved. Data science is using data to help us solve problems problems and find patterns. We will now learn how scientists and programmers alike use models and simulations to help them test their predictions and to see how things behave.
>
> Think about if we wanted to know how people interacted and socialized at a party. We could construct a simulation that shows us how people would behave at a party. We will work with a similar simulation later in our activity. For now, let's do a quick reading about modeling and simulation. As we read, I want you to keep the following question in mind.

💬 *Let's Think!* Why would a scientist use a model or simulation? Why are they important?

[1] **Read:** As a class, read Computing Modeling and Simulation [5 min].

🎤 *Remarks*

> After reading that article, we understand that models and simulations are very important to scientists. Scientists use models and simulations to help them study problems that would be impossible to carry out in the real world. For example, astronauts practice in simulations of the moon before they actually go there and land to get better accustomed to the way that real spaceships move. Scientists can use these models and simulations to test their hypotheses about specific outcomes of the models and simulations.

Let's Watch! Intro To NetLogo

🎤 *Remarks*

> Now that we have a better understanding of models and simulations, we can take a look at NetLogo. NetLogo Web is a free web-based programming environment for simulations. The site comes with hundreds of pre-defined models of things like the spread of disease, the effect of clouds on global warming, etc. We will watch a video that talks about what NetLogo is and shows us some of the simulations it has. As we watch this video, I would like you to keep the following questions in mind:

💬 *Let's Think!* What is NetLogo?

**Watch:** As a class, watch Intro To NetLogo [1:59 min].

## Activity (80 min)

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** English Learner version of reading—Computing Modeling And Simulation—available.
[2] **English Learners:** English Learner version of activity guide—Modeling And Simulation - Activity Guide—available. Step seven of this activity guide is optional for English Learners.
[3] **English Learners:** English Learner version of activity guide—Party Simulation - Activity Guide—available.

### Let's Play! Modeling And Simulation Activity

🎙️ *Remarks*

We can now begin working on activities with NetLogo Web. In this activity, you will complete a tutorial using the Wolf Sheep Predation model in NetLogo. This tutorial will familiarize you with the interface as well as teach you basic functionalities in NetLogo such as setting up and running models, adjusting settings of models, gathering information with plots and monitors, and more. After you complete the tutorial, you will then experiment with the Party simulation in NetLogo that will guide you in forming your own hypotheses about the social interactions of different genders at a party. At the end of the activity, you will submit your completed activity guide to me. Please call me over if you need any help as you work through this activity.

**Distribute:** One copy of the following to each student:

- [2]Modeling And Simulation - Activity Guide         [40 min]
- [3]Party Simulation - Activity Guide                      [40 min]

As a class, review the overview of the activity as well as the expectations and necessary submissions for the activity as discussed in the *Remarks* above.

**Support:** Allow students time to work on the activities. As you circulate the classroom, assist any students that may have questions or issues that need to be resolved as they work through the activity.

**Transition:** As the class time comes to an end, try to bring students attention back to you to have a final discussion about modeling and simulation.

## Wrap Up (10 min)

### Let's Talk! Why Do We Care?

🎙️ *Remarks*

I hope you all had fun playing with the different simulations in this activity. I would like to have a quick discussion on models and simulations.

**Discussion Questions:** The following can be used as potential discussion questions during your wrap-up with students:

- How are models and simulations useful?
- Why are models and simulations important to scientists?
- What is NetLogo?
- What other simulations in the NetLogo library looked interesting to you? Why?
- What are some real world examples (i.e., astronaut) that could be modeled?
- What did you learn from using the Party simulation? Was the outcome what you expected?

💬 **Discuss:** Allow students an opportunity to share their responses with the class.

Great responses! I think we learned a lot about modeling and simulation today. We will continue to work with NetLogo in future activities where you will learn how the models are programmed and eventually program your own simulation! Make sure you have submitted your completed activity guide to me before you leave class. I will see you all in the next class where we will examine the NetLogo programming language!

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** English Learner version of reading—Computing Modeling And Simulation—available.
[2] **English Learners:** English Learner version of activity guide—Modeling And Simulation - Activity Guide—available. Step seven of this activity guide is optional for English Learners.
[3] **English Learners:** English Learner version of activity guide—Party Simulation - Activity Guide—available.

# Lesson 3: Agent Commands

## Overview

This unit presents basic concepts in modeling complex systems through computer programming of agent-based simulations using NetLogo. It uses NetLogo Web, which is a free web-based programming environment for simulation. This lesson shows the students that NetLogo simulations are programmed in the NetLogo programming language. It has them do a tutorial on commanding elements (such as turtles and patches) of the NetLogo world to learn that these elements respond to program control. It then has them do a tutorial that introduces NetLogo programming.

## Purpose

The purpose of this lesson is to introduce students to the concept that models are programmed. Students will learn to explain that models are programmed. Specifically, students will use the NetLogo Command Center to command various agents in the models to perform actions. Students will also explore the NetLogo code and experiment with code to produce different behaviors in an existing model.

## Agenda

**Warm Up (10 min)**
  **Let's Watch! Agent Based Computer Models**
**Activity (40 min)**
  **Let's Play! Agent Commands Activity**
**Wrap Up (10 min)**
  **Let's Talk! Why Do We Care?**

## Objectives

### Content Objectives

*Students will be able to:*

- Explain that models are programmed.
- Command elements of NetLogo to perform actions.
- Read and modify NetLogo code to create different behavior in an existing model.
- Operate the Command Center of NetLogo.
- Issue commands to different agents in NetLogo.

### Language Objectives

*Students will be able to:*

- *Orally explain* and *summarize in writing* that models are programmed.
- *Write commands* and *summarize in writing* the elements they command in NetLogo to perform actions.
- *Read* and *write* NetLogo code to create different behavior in an existing model.
- *Read* and *write* code in the Command Center of NetLogo.
- *Read* and *write* commands to different agents in NetLogo.

## Preparation

**For the student:**

- ❏ A computer with working Internet access to various websites such as Google Products (Google Drive, Google Docs, YouTube, etc.), and NetLogo
- ❏ A Gmail account, either personal or school account
- ❏ One copy of the Agent Commands - Activity Guide

**For the teacher:**

- ❏ A computer with working Internet access to various websites such as Google Products (Google Docs, YouTube, etc.)
- ❏ A Gmail account, either personal or school account
- ❏ One copy of the Agent Commands - Activity Guide

## Vocabulary

- Vocabulary Guide
- Vocabulary Cards
- Vocabulary Slideshow With Audio
- Vocabulary Slideshow
- Vocabulary Wordsearch
- Vocabulary Crossword
- Vocabulary Word Wall
- Vocabulary Quizziz

## Links

**For the Students:**

- One copy of the Agent Commands - Activity Guide

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** Very difficult and fast moving vocabulary in the first minute but it gets much better as the video goes on.
[2] **English Learners:** English Learner version of activity guide—Agent Commands - Activity Guide—available.

**For the Teacher:**

- UNIT ONE: Lesson 3 - Agent Commands Quiz
- Agent Commands - Activity Guide Rubric
- Answer Key: Agent Commands - Activity Guide

## Support

**Report a Bug**

- Are the simulations engaging? Is enough guidance provided for students to complete the activity?
- Is a reasonable amount of time given to students to complete the activity?

**Give us feedback!**

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** Very difficult and fast moving vocabulary in the first minute but it gets much better as the video goes on.
[2] **English Learners:** English Learner version of activity guide—Agent Commands - Activity Guide—available.

# Teaching Guide

## Warm Up (10 min)

Let's Watch! Agent Based Computer Models

🎤 *Remarks*

In the last class, we started to learn about modeling and simulation using NetLogo. We will now get into the programming behind the models that make them run. We will now take a look at a specific modeling type known as agent based computer modeling. We will watch a video that defines and describes an agent based computer model. As we watch this video, I would like you to keep the following questions in mind:

💬 *Let's Think!* What is an agent based computer model? What is an agent? How can this

type of model be useful to us?

[1] **Watch:** As a class, watch Agent Based Computer Models [5:04 min].

🎤 *Remarks*

After watching the this video, we have learned that in agent based modeling you create the model by describing the agents, providing the agents with rules to follow, and allowing them to interact with other agents and the environment. We can then use this agent based computer model to run various experiments to help us understand the real world. Remember in the video she described a model based on an epidemic? In this epidemic model, you would have different agents—healthy ones and sick ones—represent a population to observe how a disease might spread when these agents interact (when a healthy agent touches a sick agent). We can use agent based computer models to model and understand complex adaptive systems with the rules we provide the agents with.

NetLogo is an agent based programming language. This means that the models in NetLogo use agents and give them rules to follow in order for us to study how they interact with other agents and the environment. In the next activity, we will be working with an agent based computer models about traffic. In this traffic model, the agents are the observer, turtles, patches and links. The observer is the person overseeing everything that is going on in the simulation. The turtles in this model would be the cars. The patches are the ground on which the turtles (or cars) move. Finally, the links are the different connections between the turtles. This will make more sense as we do the activity so you can actually see what I am describing. So let's get started!

## Activity (40 min)

Let's Play! Agent Commands Activity

🎤 *Remarks*

We can now begin working on activities with NetLogo Web. In this activity, you will complete another tutorial to teach you about how simulations are programmed in NetLogo. You will use a traffic model to issue commands to the model using the NetLogo programming language. At the end of the activity,

English Learner Lesson Adaptation Footnotes:
[1] **English Learners:** Very difficult and fast moving vocabulary in the first minute but it gets much better as the video goes on.
[2] **English Learners:** English Learner version of activity guide—Agent Commands - Activity Guide—available.

you will submit your completed activity guide to me. Please call me over if you need any help as you work through this activity.

**Distribute:** One copy of the following to each student:

- [2]Agent Commands - Activity Guide [40 min]

As a class, review the overview of the activity as well as the expectations and necessary submissions for the activity as discussed in the *Remarks* above.

**Support:** Allow students time to work on the activities. As you circulate the classroom, assist any students that may have questions or issues that need to be resolved as they work through the activity.

**Transition:** As the class time comes to an end, try to bring students attention back to you to have a final discussion about modeling and simulation.

## Wrap Up (10 min)

### Let's Talk! Why Do We Care?

🎙 *Remarks*

I hope you all had fun working the traffic simulation in this activity. I would like to have a quick discussion on agent based computer models.

**Discussion Questions:** The following can be used as potential discussion questions during your wrap-up with students:

- What is an agent based computer model? How is this type of model useful?
- How could a scientist use an agent based computer model?
- What is an agent? How are agents used in models?
- In the traffic model, what are the agents? How can manipulate each agent?

💬 **Discuss:** Allow students an opportunity to share their responses with the class.

Great responses! We learned a lot about agent commands in NetLogo today. We will continue to work with NetLogo in future activities where we will begin programming some simulations. Make sure you have submitted your completed activity guide to me before you leave class. I will see you all in the next class where we will examine the NetLogo programming procedures!

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** Very difficult and fast moving vocabulary in the first minute but it gets much better as the video goes on.
[2] **English Learners:** English Learner version of activity guide—Agent Commands - Activity Guide—available.

# Lesson 4: Programming Procedures

## Overview

This unit presents basic concepts in modeling complex systems through computer programming of agent-based simulations using NetLogo. It uses NetLogo Web, which is a free web-based programming environment for simulation. This lesson will provide students with the essential skills and tools for constructing simulations in NetLogo. Students will build a simple ecosystem model in NetLogo to simulate turtles' behaviors in real life such as moving, eating, reproducing, and dying. Students will be provided with a basic interface of their model in which they will program the functionality of each button and slider. Additionally, students will write several procedures using the four types of agents to define behaviors for the turtles, monitors, sliders, and plots.

## Purpose

The purpose of this lesson is to introduce students to programming procedures in NetLogo. Students will write and run several procedures in the model that define the behaviors of each button and slider. At the end of this lesson, students will have programmed their first model in NetLogo Web simulating real world phenomenon. Students will learn the essential skills and tools to define procedures in NetLogo that they will utilize in the next lesson where they construct their own simulation.

## Agenda

**Warm Up (10 min)**
> **Let's Watch! What Most Schools Don't Teach**

**Activity (160 min)**
> **Let's Play! Programming Procedures Activities**

**Wrap Up (10 min)**
> **Let's Talk! Why Do We Care?**

## Objectives

*Content Objectives*

*Students will be able to:*

- Build a simple ecosystem model in NetLogo to simulate real life phenomenon.
- Use four types of agents--patches, turtles, links, and observer--to run procedures.
- Define and program several procedures in NetLogo to control behaviors of buttons and sliders on the model interface.
- Construct procedures in NetLogo that make turtles move, eat, reproduce and die.
- Make monitors, sliders, and plots.
- Analyze the effect of some parameters to the model.
- Program with variables, procedures with conditional statements (i.e., if, ifelse), and primitives (i.e., die, hatch) in the NetLogo programming language.
- Generate and plot data produced from running experiments with the model.
- Interpret data produced from running experiments with models.

*Language Objectives*

*Students will be able to:*

- *Read instructions, read* and *write code* to build a simple ecosystem model in NetLogo to simulate real life phenomenon.
- *Read instructions, read* and *write code* to use four types of agents--patches, turtles, links, and observer--to run procedures.
- *Read instructions, read* and *write code* to define and program several procedures in NetLogo to control behaviors of buttons and sliders on the model interface.
- *Read instructions, read* and *write code* to construct procedures in NetLogo that make turtles move, eat, reproduce and die.
- *Read instructions, read* and *write code* to make monitors, sliders, and plots.
- *Read* and *interpret* data from the model and *summarize in writing* their findings of the effect of some parameters to the model.
- *Read instructions, read* and *write code* to program with variables, procedures with conditional statements (i.e., if, ifelse), and primitives (i.e., die, hatch) in the NetLogo programming language.
- *Read instructions, read* and *write code* to generate and plot data produced from running experiments with the model.
- *Read* and *interpret* data from the model and *summarize in writing* their findings to interpret data produced from running experiments with models.

**English Learner Lesson Adaptation Footnotes:**
    **Make sure to go over the saving the code procedure very carefully and check that all students understand it.**
**[1] English Learners:** English Learner version of activity guide—Programming Procedures - Activity Guide—available.
**[2] English Learners:** English Learner version of activity guide— Variables- Activity Guide—available.
**[3] English Learners:** English Learner version of activity guide—Monitors, Data, And Plotting - Activity Guide—available.
**[4] English Learners:** English Learner version of activity guide—Time And Parameters - Activity Guide—available.

## Preparation

**For the student:**

- ❏ A computer with working Internet access to various websites such as Google Products (Google Drive, Google Docs, YouTube, etc.), and NetLogo
- ❏ A Gmail account, either personal or school account
- ❏ One copy of the Programming Procedures - Activity Guide
- ❏ Once copy of the Variables - Activity Guide
- ❏ One copy of the Monitors, Data, And Plotting - Activity Guide
- ❏ One copy of Time And Parameters - Activity Guide

**For the teacher:**

- ❏ A computer with working Internet access to various websites such as Google Products (Google Docs, YouTube, etc.)
- ❏ A Gmail account, either personal or school account
- ❏ One copy of the Programming Procedures - Activity Guide
- ❏ Once copy of the Variables - Activity Guide
- ❏ One copy of the Monitors, Data, And Plotting - Activity Guide
- ❏ One copy of Time And Parameters - Activity Guide
- ❏ One copy of NetLogo FAQ: Debugging Common Errors Guide

## Vocabulary

- Vocabulary Guide
- Vocabulary Cards

**English Learner Lesson Adaptation Footnotes:**
**Make sure to go over the saving the code procedure very carefully and check that all students understand it.**
[1] **English Learners:** English Learner version of activity guide—Programming Procedures - Activity Guide—available.
[2] **English Learners:** English Learner version of activity guide— Variables- Activity Guide—available.
[3] **English Learners:** English Learner version of activity guide—Monitors, Data, And Plotting - Activity Guide—available.
[4] **English Learners:** English Learner version of activity guide—Time And Parameters - Activity Guide—available.

- [Vocabulary Slideshow With Audio](#)
- [Vocabulary Slideshow](#)
- [Vocabulary Wordsearch](#)
- [Vocabulary Crossword](#)
- [Vocabulary Word Wall](#)
- [Vocabulary Quizziz](#)

## Links

**For the Students:**

- One copy of the [Programming Procedures - Activity Guide](#)
- Once copy of the [Variables - Activity Guide](#)
- One copy of the [Monitors, Data, And Plotting - Activity Guide](#)
- One copy of [Time And Parameters - Activity Guide](#)

**For the Teacher:**

- [UNIT ONE: Lesson 4 - Programming Procedures Quiz](#)

- [Programming Procedures - Activity Guide Rubric](#)
- [Variables - Activity Guide Rubric](#)
- [Monitors, Data, And Plotting - Activity Guide Rubric](#)
- [Time And Parameters - Activity Guide Rubric](#)

- [Answer Key: Programming Procedures - Activity Guide](#)
- [Answer Key: Variables - Activity Guide](#)
- [Answer Key: Monitors, Data, And Plotting - Activity Guide](#)
- [Answer Key: Time And Parameters - Activity Guide](#)

- [NetLogo FAQ: Debugging Common Errors Guide](#)

**English Learner Lesson Adaptation Footnotes:**
   Make sure to go over the saving the code procedure very carefully and check that all students understand it.
[1] **English Learners:** English Learner version of activity guide—[Programming Procedures - Activity Guide](#)—available.
[2] **English Learners:** English Learner version of activity guide— [Variables- Activity Guide](#)—available.
[3] **English Learners:** English Learner version of activity guide—[Monitors, Data, And Plotting - Activity Guide](#)—available.
[4] **English Learners:** English Learner version of activity guide—[Time And Parameters - Activity Guide](#)—available.

## Support

**Report a Bug**

- Is enough guidance provided for students to complete the activity?
- Is a reasonable amount of time given to students to complete the activity?

**Give us feedback!**

**English Learner Lesson Adaptation Footnotes:**
        **Make sure to go over the saving the code procedure very carefully and check that all students understand it.**
[1] **English Learners:** English Learner version of activity guide—Programming Procedures - Activity Guide—available.
[2] **English Learners:** English Learner version of activity guide— Variables- Activity Guide—available.
[3] **English Learners:** English Learner version of activity guide—Monitors, Data, And Plotting - Activity Guide—available.
[4] **English Learners:** English Learner version of activity guide—Time And Parameters - Activity Guide—available.

# Teaching Guide

## Warm Up (10 min)

Let's Watch! What Most Schools Don't Teach

🎙 *Remarks*

> In the last class, we started to learn about agent based computer models and issuing commands to models in NetLogo. We will now learn about the real heart of NetLogo models in which we will code the functionality of the model. So what do I mean when I say this? I mean that we will be telling the model exactly what to do like when a button is clicked or when a slider is pulled. Before we get into the actual activity, I want to share a video with everyone about coding. As we watch this video, I would like you to keep the following questions in mind:

> 💬 *Let's Think!* What is coding? Why is it important to learn how to code?

**Watch:** As a class, watch [What Most Schools Don't Teach](#) [5:43 min].

🎙 *Remarks*

> After watching the this video, we can recognize that learning how to code is very important! Although coding may seem intimidating at first, we know that it will become easier with more practice over time. The best part about coding is anyone can do it!

> Coding requires computational thinking. We learned about computational thinking in our very first class where we passed instructions to each group on how to create their monsters. Remember that computational thinking is breaking down a larger and more complex problem into smaller ones that we can more easily solve. When we have these smaller problems, we can define a solution for how to solve them using coding.

> Over the next few classes, we will be coding our first model in NetLogo Web. Our large and complex problem will be building a basic ecosystem model that simulates turtles moving, eating, reproducing, and dying. We will use computational thinking skills to break this problem into smaller pieces—like defining a solution for how the turtle will move, then how the turtle will eat, etc. We can define these solutions using the NetLogo Code tab where we can write procedures that combine a series of NetLogo commands into a single new command that you define. This will make more sense as we actually work through the tutorials in the activity so let's get started!

## Activity (160 min)

Let's Play! Programming Procedures Activities

🎙 *Remarks*

> We can now begin working on activities with NetLogo Web. For this lesson, you will have four activities to complete but you will be given more time than usual to complete them. Each activity will provide guided steps on how to code the model in NetLogo Web. In the beginning, you will learn to code the procedures that define behaviors for the setup and go buttons in the model. You will also learn to code your own procedures that help define behavior for the turtles. Additionally, you will learn core programming concepts in NetLogo such as defining procedures, variables, monitors, parameters, and

**English Learner Lesson Adaptation Footnotes:**
      Make sure to go over the saving the code procedure very carefully and check that all students understand it.
[1] **English Learners:** English Learner version of activity guide—[Programming Procedures - Activity Guide](#)—available.
[2] **English Learners:** English Learner version of activity guide— [Variables- Activity Guide](#)—available.
[3] **English Learners:** English Learner version of activity guide—[Monitors, Data, And Plotting - Activity Guide](#)—available.
[4] **English Learners:** English Learner version of activity guide—[Time And Parameters - Activity Guide](#)—available.

more. As you continue to work through the activities, you will gradually construct a working model that simulates a simple real life ecosystem. At the end of the activity, you will submit your completed activity guide to me. Please call me over if you need any help as you work through these activities.

**Distribute:** One copy of the following to each student:

- [1]Programming Procedures - Activity Guide          [40 min]
- [2]Variables - Activity Guide          [40 min]
- [3]Monitors, Data, And Plotting - Activity Guide          [40 min]
- [4]Time And Parameters - Activity Guide          [40 min]

As a class, review the overview of the activity as well as the expectations and necessary submissions for the activity.

**Support:** Allow students time to work on the activities. As you circulate the classroom, assist any students that may have questions or issues that need to be resolved as they work through the activity.

**Transition:** As the class time comes to an end, try to bring students attention back to you to have a final discussion about programming procedures.

## Wrap Up (10 min)

### Let's Talk! Why Do We Care?

🎤 *Remarks*

I hope you all had fun coding your first model in this activity. I would like to have a quick discussion on coding in NetLogo.

**Discussion Questions:** The following can be used as potential discussion questions during your wrap-up with students:

- What is coding? Why is it so important to learn how to code?
- Why is computational thinking important to coding? How did you use computational thinking in your activities?
- What are procedures in NetLogo? How did you use procedures in your activities?
- How is the Wolf Sheep Predation model different from the one you just made? How do you think the Wolf Sheep Predation model was made?
- What ideas/topics do you have for building your own simulations? What are some of the skills that you learned in these activities that you can apply when building your own simulation?

💬 **Discuss:** Allow students an opportunity to share their responses with the class.

Great responses! We learned a lot about coding in NetLogo today. We will continue to work with NetLogo in one final activity where you all will program your own simulations! Make sure you have submitted your completed activity guides to me before you leave class. I will see you all in the next class where we will begin programming on our own simulations!

English Learner Lesson Adaptation Footnotes:
    Make sure to go over the saving the code procedure very carefully and check that all students understand it.
[1] **English Learners:** English Learner version of activity guide—Programming Procedures - Activity Guide—available.
[2] **English Learners:** English Learner version of activity guide— Variables- Activity Guide—available.
[3] **English Learners:** English Learner version of activity guide—Monitors, Data, And Plotting - Activity Guide—available.
[4] **English Learners:** English Learner version of activity guide—Time And Parameters - Activity Guide—available.

# Lesson 5: Program Your Own Simulation

## Overview

This unit presents basic concepts in modeling complex systems through computer programming of agent-based simulations using NetLogo. It uses NetLogo Web, which is a free web-based programming environment for simulation. This lesson will provide students with some guidance on constructing their own program in NetLogo. Students will create a NetLogo program to help determine how many mouse traps to lay to catch a pesky mouse. Students will utilize prior skills and knowledge of programming procedures in NetLogo to write this program. Within this program, students will modify existing code of a simulation in NetLogo to adjust the color and position of patches, the number of turtles, and the behavior of turtles when moving around the patches (i.e., turtle moves around black patches but stops moving when it touches a yellow patch). Additionally, students will demonstrate their understanding of the program as they explain each line of code that they either reuse or modify. After students complete the program, they will then use their simulation to run various trials of an experiment and record their data using spreadsheet software. Students will use Google Sheets as their spreadsheet software in which they will record and analyze their collected data through summary tables and visualizations. At the end of this lesson, students will be able to answer a big question using the analysis of their collected data.

## Purpose

The purpose of this lesson is to utilize prior knowledge and skills in NetLogo to build an original simulation. Students will gain exposure to program specifications that outline the functionality of the mousetrap simulation they will write. Students will then establish a big question based upon the outline of the simulation. The big question allows students to realize the value of the simulation as this simulation they write will provide them with the data needed to answer this big question. Students will then begin to program their simulation by reusing existing code they have written in the previous lesson. Using this existing code, students will modify it

## Objectives

### Content Objectives

*Students will be able to:*

- Establish a big question for the mousetrap model.
- Reuse and modify existing code.
- Explain each line of code and its effect on the program.
- Program an original mousetrap model in NetLogo.
- Collect data from the NetLogo model.
- Use Google Sheets to record and analyze data with summary tables and visualizations.
- Support their answer to a big question with analysis of data.

### Language Objectives

*Students will be able to:*

- *Read* the established big question for the mousetrap model.
- *Read* and *write* to reuse and modify existing code.
- *Read* and *write* to explain each line of code and its effect on the program.
- *Write* code to program an original mousetrap model in NetLogo.
- *Read*, *interpret*, and *summarize in writing* their findings of collected data from the NetLogo model.

according to the program specifications. Additionally, students will be expected to explain each line of code they reuse and modify to show their understanding of how the program works. Students will then use their completed program to run trials and collect data to help answer the big question. Using a spreadsheet program, students will record their data from the trials and analyze it using summary tables and visualizations. Students will then utilize their analysis of the collected data as evidence when answering the big question. Students will gain exposure to collecting and working with data which they will be utilizing extensively in the next unit.

## Agenda

**Warm Up (20 min)**
> **Let's Watch! Introduction to Loops in NetLogo**

**Activity (120 min)**
> **Let's Play! Program Your Own Simulation**

**Wrap Up (10 min)**
> **Let's Talk! Why Do We Care?**

- *Read*, *interpret*, and *summarize in writing* using Google Sheets to record and analyze data with summary tables and visualizations.
- *State* and *justify in writing* their answer to a big question with analysis of data.

## Preparation

**For the student:**

- ❑ A computer with working Internet access to various websites such as Google Products and NetLogo
- ❑ A Gmail account, either personal or school account
- ❑ One copy of Program A Mouse Trap Simulation - Activity Guide
- ❑ One copy of Collecting Data In NetLogo - Activity Guide

**For the teacher:**

- ❑ A computer with working Internet access to various websites such as Google Products and NetLogo
- ❑ A Gmail account, either personal or school account
- ❑ One copy of Program A Mouse Trap Simulation - Activity Guide
- ❑ One copy of Collecting Data In NetLogo - Activity Guide
- ❑ One copy of NetLogo FAQ: Debugging Common Errors Guide

## Vocabulary

- Vocabulary Guide
- Vocabulary Cards
- Vocabulary Slideshow With Audio
- Vocabulary Slideshow
- Vocabulary Wordsearch

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** Video contains complex vocabulary. If used, you will need to run it slow with many stops.
[2] **English Learners:** English Learner version of activity guide—Program A Mouse Trap Simulation - Activity Guide—available.
[3] **English Learners:** English Learner version of activity guide—Collecting Data In NetLogo - Activity Guide—available.

- Vocabulary Crossword
- Vocabulary Word Wall
- Vocabulary Quizziz

## Links

**For the Students:**

- One copy of Program A Mouse Trap Simulation - Activity Guide
- One copy of Collecting Data In NetLogo - Activity Guide

**For the Teacher:**

- UNIT ONE: Lesson 5 - Program Your Own Simulation Quiz

- Program A Mouse Trap Simulation - Activity Guide Rubric
- Collecting Data In NetLogo - Activity Guide Rubric

- Answer Key: Program A Mouse Trap Simulation - Activity Guide
- Answer Key: Collecting Data In NetLogo - Activity Guide
- Answer Key: Collecting Data In NetLogo Spreadsheet

- NetLogo FAQ: Debugging Common Errors Guide

## Support

**Report a Bug**

- Is enough guidance provided for students to complete the activity?

**Give us feedback!**

English Learner Lesson Adaptation Footnotes:
[1] **English Learners:** Video contains complex vocabulary. If used, you will need to run it slow with many stops.
[2] **English Learners:** English Learner version of activity guide—Program A Mouse Trap Simulation - Activity Guide—available.
[3] **English Learners:** English Learner version of activity guide—Collecting Data In NetLogo - Activity Guide—available.

# Teaching Guide

## Warm Up (20 min)

Let's Watch! Let's Talk! Review Skills

### 🎤 *Remarks*

In the previous class, we worked extensively with programming procedures in NetLogo to create a simple real life ecosystem. We used the four types of agents to run our procedures including patches, turtles, links, and the observer. Using these agents, we were able to rewrite programming procedures to control the behavior of buttons and sliders on the model interface. Additionally, you learned core programming concepts in NetLogo such as defining procedures, variables, monitors, parameters, and more.

We will now begin programming an original simulation using NetLogo. The idea for this simulation involves a mouse running around the floor that is covered in mouse traps. It seems like a simple idea but we will need to outline the exact specifications of this program. When we click the setup button, our View for the mousetrap simulation should have a black floor with various yellow traps laid randomly on it and a mouse randomly positioned on the floor. When we click the go button, this mouse should move randomly around the floor until it gets caught in one of the yellow traps. We understand what the simulation should do when the setup and go buttons are clicked but how can we relate these actions to the NetLogo agents? Let's consider the following questions:

### ☁ *Let's Think!*  What agent(s) should represent the floor in this simulation?

What agent(s) should represent the yellow traps in this simulation?

What agent(s) should represent the mouse in this simulation?

We now understand the different agents that our mousetrap simulation will be using including the patches for the floor and traps and one turtle for the mouse. The tricky aspect of this simulation will be making some of the patches yellow and randomly positioned on the floor. We will need to use a program loop to accomplish this in our mousetrap simulation. I would like us to watch a video that helps define and explain program loops in NetLogo. As we watch this video, I would like you to keep the following questions in mind:

### ☁ *Let's Think!*  What is a loop? Why would we use a loop? How can we use a loop in our simulation

for the patches?

[1] **Watch:** As a class, watch Introduction to Loops in NetLogo [5:05].

### 🎤 *Remarks*

As the video stated, program loops are used to repeat commands over and over again. The number of times a command is repeated is known as the number of iterations in a loop. We use loops to create more efficient programs that may result in far fewer lines of codes. Additionally, the video mentioned that loops should be used when performing tedious tasks, analyzing data, performing repeated calculations, and reusing code. Although several types of loops were mentioned in this video, we will be using a conditional loop in our simulation in which our program loop will run until the number of

---

traps is not equal to number on the slider of the simulation. With our understanding of loops, we can now begin to construct our mousetrap simulation in the NetLogo programming language.

## Activity (120 min)

### Let's Play! Program Your Own Simulation

🎙 *Remarks*

We can now begin working on the activities for creating and experimenting with our original mousetrap simulation in NetLogo. For this lesson, you will complete two activities but you will be given more time than usual to complete them. In the first activity, you follow steps to code the procedures for the mousetrap simulation with the program specifications that we outlined earlier in class. You will be reusing previously written code in the last lesson to help you construct this simulation. Additionally, you will modify this reused code to fit the behavior of the patches and turtle agents of this simulation. At the end of the first activity, you will have successfully programmed the mousetrap simulation in NetLogo.

The second activity for this lesson will utilize your mousetrap simulation to collect data to help answer the big question—*How does the average time (in ticks) to catch a mouse depend on the number of traps used?*. You will run several experiments in your mousetrap simulation and collect data from your experiments. The collected data will be stored in a spreadsheet using Google Sheets. You will use this spreadsheet software to analyze your data and look for significant trends to support your answer to the big question. This activity serves to give you exposure to data which we will be working with for the remainder of the course.

Once you have finished these activities, you will submit your completed activity guides to me. Please call me over if you need any help as you work through these activities. Let's get started!

**Distribute:** One copy of the following to each student:

- [2]Program A Mouse Trap Simulation - Activity Guide    [80 min]
- [3]Collecting Data In NetLogo - Activity Guide    [40 min]

As a class, review the overview of the activity as well as the expectations and necessary submissions for the activity.

**Support:** Allow students time to work on the activities. As you circulate the classroom, assist any students that may have questions or issues that need to be resolved as they work through the activity.

**Transition:** As the class time comes to an end, try to bring students attention back to you to have a final discussion about modeling and simulation.

## Wrap Up (10 min)

### Let's Talk! Why Do We Care?

🎙 *Remarks*

I hope you all had fun writing and collecting data from this mousetrap simulation in these activities. I would like to have a final discussion on NetLogo.

**Discussion Questions:** The following can be used as potential discussion questions during your wrap-up with students:

**English Learner Lesson Adaptation Footnotes:**
[1] **English Learners:** Video contains complex vocabulary. If used, you will need to run it slow with many stops.
[2] **English Learners:** English Learner version of activity guide—Program A Mouse Trap Simulation - Activity Guide—available.
[3] **English Learners:** English Learner version of activity guide—Collecting Data In NetLogo - Activity Guide—available.

- What program specifications did we come up with for our mousetrap simulation? Why is it important to outline the program specifications first before actually programming the simulation?
- What is a loop? Why would we use a loop? How did we use a loop in our simulation?
- What code (i.e., programming procedures) did you reuse from previous lessons? How did you reuse this code in your mousetrap simulation? How did you modify this code in your mousetrap simulation?
- What is your answer to the big question? How did you support your answer to the big question with evidence? How did summary tables support your answer to the big question? How did data visualizations support your answer to the big question?
- Did you expect this answer to your big question? Why? Why not?

**Discuss:** Allow students an opportunity to share their responses with the class.

Great responses! We learned a lot about coding in NetLogo today. We will continue to work with NetLogo in one final activity where you all will program your own simulations! Make sure you have submitted your completed activity guides to me before you leave class. I will see you all in the next class where we will begin programming on our own simulations!

**English Learner Lesson Adaptation Footnotes:**
**[1] English Learners:** Video contains complex vocabulary. If used, you will need to run it slow with many stops.
**[2] English Learners:** English Learner version of activity guide—Program A Mouse Trap Simulation - Activity Guide—available.
**[3] English Learners:** English Learner version of activity guide—Collecting Data In NetLogo - Activity Guide—available.

# BIBLIOGRAPHY

"Next generation science standards." n.d. [Online]. Available: https://www.nextgenscience.org/2017

American Federation Of Teachers. "Every student succeeds act: A new day in public education - english language learners." 2016. [Online]. Available: http://www.aft.org/sites/default/files/essa_ells.pdf

Barlow, D., "The sourcebook for teaching science: Strategies, activities, and instructional resources (grades 6-12)," *The Education Digest*, vol. 74, no. 6, p. 72, 2009.

Brooks, K., Adams, S. R., and Morita-Mullaney, T., "Creating inclusive learning communities for ell students: Transforming school principals' perspectives," *Theory Into Practice*, vol. 49, no. 2, pp. 145–151, 2010.

Code.org. "Cs discoveries 2018." 2018. [Online]. Available: https://curriculum.code.org/csd-18/

Code.org. "Code.org infographic source data." n.d. [Online]. Available: https://docs.google.com/document/d/1gySkItxiJn_vwb8HIIKNXqen184mRtzDX12cux0ZgZk/pub

Code.org. "Promote computer science." n.d. [Online]. Available: https://code.org/promote

English Language Learners in Public Schools. "The condition of education." Mar. 2017. [Online]. Available: https://nces.ed.gov/programs/coe/indicator_cgf.aspf

Fradd, S. H., Lee, O., Sutman, F. X., and Saxton, M. K., "Promoting science literacy with english language learners through instructional materials development: A case study," *Bilingual Research Journal*, vol. 25, no. 4, pp. 479–501, 2001.

Google and Gallup. "Searching for computer science: Access and barriers in u.s. k-12 education." 2015. [Online]. Available: https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf

Honigsfeld, A., "Ell programs: Not "one size fits all"," *Kappa Delta Pi Record*, vol. 45, no. 4, pp. 166–171, 2009.

Janzen, J., "Teaching english language learners in the content areas," *review of Educational research*, vol. 78, no. 4, pp. 1010–1038, 2008.

Lee, O., "Science education with english language learners: Synthesis and research agenda," *Review of Educational Research*, vol. 75, no. 4, pp. 491–530, 2005.

Lindahl, K. "The essa and esl teacher education." Dec. 2015. [Online]. Available: http://blog.tesol.org/the-essa-and-esl-teacher-education/

Margo Gottlieb, M. Elizabeth Cranley, A. C. "Understanding the wida english language proficiency standards a resource guide prekindergarten through grade 12." 2009. [Online]. Available: https://www.wida.us/standards/Resource_Guide_web.pdf

McFarland, J., Hussar, B., de Brey, C., Snyder, T., Wang, X., Wilkinson-Flicker, S., Gebrekristos, S., Zhang, J., Rathbun, A., Barmer, A., Mann, F. B., and Hinz, S. "The condition of education 2017." 2017. [Online]. Available: https://nces.ed.gov/pubs2017/2017144.pdf

Meltzer, J. and Hamann, E. T., "Meeting the literacy development needs of adolescent english language learners through content-area learning-part two: Focus on classroom teaching and learning strategies," 2005.

Merriam-Webster. "Merriam-webster's learner's dictionary." 2018. [Online]. Available: http://www.learnersdictionary.com/

National Archives and Records Administration. "Tech hire." n.d. [Online]. Available: https://obamawhitehouse.archives.gov/issues/technology/techhire

National Center for Women In Technology. "Ncwit fact sheet." n.d. [Online]. Available: https://www.ncwit.org/ncwit-fact-sheet

National Science Foundation. "Broadening participation in computing." 2009. [Online]. Available: https://www.nsf.gov/pubs/2009/nsf09534/nsf09534.pdf

National Science Foundation. "Computing education for the 21rst century." 2012. [Online]. Available: fromhttp://www.nsf.gov/pubs/2012/nsf12527/nsf12527.htm

National Science Foundation. "Stem + computing partnerships." 2015. [Online]. Available: https://www.nsf.gov/pubs/2015/nsf15537/nsf15537.htm

Northwestern University. "Netlogo." 2017. [Online]. Available: https://ccl.northwestern.edu/netlogo/

Quizizz. "Quizizz: Fun multiplayer classroom quizzes." 2018. [Online]. Available: https://quizizz.com/

Rhode Island Department Of Education. "Regulations governing the education of english learners in rhode island." Dec. 2016. [Online]. Available: http://www.ride.ri.gov/Portals/0/Uploads/Documents/

Students-and-Families-Great-Schools/English-Language-Learners/
EL_Regulations_Amendment_12_20_16.pdf

Rhode Island KIDS COUNT. "2016 rhode island kids count fact-book." 2016. [Online]. Available: http://www.rikidscount.org/Portals/0/Uploads/Documents/Factbook%202016/2016%20Rhode%20Island%20KIDS%20COUNT%20Factbook.pdf

Rhode Island KIDS COUNT. "2017 rhode island kids count fact-book." 2017. [Online]. Available: http://www.rikidscount.org/Portals/0/Uploads/Documents/Factbook%202017/2017%20RI%20Kids%20Count%20Factbook%20for%20website.pdf

Robertson, K. "Preparing ells to be 21st-century learners." 2017. [Online]. Available: http://www.colorincolorado.org/article/preparing-ells-be-21st-century-learners

Smith, M. "Computer science for all." Jan. 2016. [Online]. Available: https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Stabback, P. "What makes a quality curriculum?" 2016. [Online]. Available: http://unesdoc.unesco.org/images/0024/002439/243975e.pdf

University of Rhode Island. "Intro to computing and data science curriculum." May 2017. [Online]. Available: http://k12.cs.uri.edu/intro_computing_curriculum_2017-18.php

University of Rhode Island. "Lesson 1: Introduction to computing and data science." 2017. [Online]. Available: https://docs.google.com/document/d/13SZN2rEnNK_OWEOjYEXTU70DE7LfBgjV9MgELSDUj_k/edit?usp=sharing

University of Rhode Island. "Lesson 2: Modeling and simulation." 2017. [Online]. Available: https://docs.google.com/document/d/1d3ys4k8w-MXrXzPUaVir6vb4j72SqQvSMhhJHiQLsyw/edit?usp=sharing

University of Rhode Island. "Lesson 3: Agent commands." 2017. [Online]. Available: https://docs.google.com/document/d/10Tsv7JQ2eSPeDWItagxUI7AcYWbuQ6_xzfetJNFaiVQ/edit?usp=sharing

University of Rhode Island. "Lesson 4: Programming procedures." 2017. [Online]. Available: https://docs.google.com/document/d/1W2apJadkWqsJ_a_jMRUij-vU2FkGwPDvuLl28fFO0xk/edit?usp=sharing

University of Rhode Island. "Lesson 5: Program your own simulation." 2017. [Online]. Available: https://docs.google.com/document/d/1Nq4swYf9DfjTx4F6GjG81RFz0xRYwzgPUiUHlsXwZsY/edit?usp=sharing

113

University of Rhode Island Department of Computer Science. "Ap computer science principles course." 2017. [Online]. Available: http://k12.cs.uri.edu/apcsp.php

University of Rhode Island Department of Computer Science. "Intro to computing and data science course." 2017. [Online]. Available: http://k12.cs.uri.edu/intro_computing.php

University of Rhode Island School of Education. "Amy correia." 2018. [Online]. Available: https://web.uri.edu/education/meet/amy-correia/

Veniegas, O. P. "Ensuring the effectiveness and quality of learning materials: How to use and evaluate literacy/ ce materials." n.d. [Online]. Available: http://www.accu.or.jp/litdbase/pub/dlperson/pdf0106/rpp24_1.pdf

Warschauer, M. and Healey, D., "Computers and language learning: An overview," *Language teaching*, vol. 31, no. 2, pp. 57–71, 1998.

Waxman, H. C. and Tellez, K., "Research synthesis on effective teaching practices for english language learners. publication series." 2002.

WIDA. "The wida standards framework and its theoretical foundations." 2014. [Online]. Available: https://www.wida.us/get.aspx?id=731

WIDA. "Why choose wida?" n.d. [Online]. Available: https://www.wida.us/membership/benefits.aspx

Wilensky, U. "Netlogo." 2016. [Online]. Available: https://ccl.northwestern.edu/netlogo/

114