COMMUNAL LEARNING VERSUS INDIVIDUAL LEARNING: AN EXPLORATORY CONVERGENT PARALLEL MIXED-METHOD STUDY TO DESCRIBE HOW YOUNG AFRICAN AMERICAN NOVICE PROGRAMMERS LEARN COMPUTATIONAL THINKING SKILLS IN AN INFORMAL LEARNING ENVIRONMENT

by

Leshell April Denise Hatley
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Education

Committee:

_____________________________________ Chair

____

____

_____________________________________ Program Director

_____________________________________ Dean, College of Education and Human Development

Date: _______________________________ Spring Semester 2016
George Mason University
Fairfax, VA

Communal Learning versus Individual Learning: An Exploratory Convergent Parallel Mixed-Method Study to Describe How Young African American Novice Programmers Learn Computational Thinking Skills in an Informal Learning Environment

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Leshell April Denise Hatley
Master of Science
University of Maryland, 2011
Master of Science
Howard University, 1998
Bachelor of Science
Howard University, 1994

Director: Kevin Clark, Professor
College of Education and Human Development

Spring Semester 2016
George Mason University
Fairfax, VA

Dedication

This is dedicated to my amazing mother, Vernell Wilson, who always said I'd go to graduate school - way before I even knew what it was. Thank you for believing in me and being by my side, always.

Acknowledgements

Above all, I acknowledge and thank God for refilling and restoring my strength and perseverance throughout this journey – from start to finish! Without question, I would like to acknowledge the many family members, loved ones, friends, and colleagues who invested and shared their support, ideas, and encouragement throughout this intellectual and soul exploring journey. SPECIAL GRATITUDE goes to my mother, Vernell Wilson, and my grandmother, Clarine Stephenson, whose love, strength, and faith pushed me to keep going when nothing else could. To my brother, Marcel Hatley, III, who recently shared his belief that playing school all those days with me when we were younger, even after school and on weekends, probably planted the seed for my success – you are probably 100% correct, big brother! ☺ THANK YOU to my dear friend Crystal Swann, whose long standing support and generosity. You and my Aunt Margaret established the bookends for this amazing accomplishment - I could not have done this without you two. To my father, Marcel Hatley, II, thank you for thinking I'm a genius. May the baton of academic freedom be passed on to Evan and Ean Crouch-Pelham. The curiosity, wonder, and growth you shared with me throughout the last year helped me appreciate the joy of meeting the *new*, and certainly helped my outlook throughout the attempt to finish this dissertation. Thanks for continuously helping me to see the world afresh through your unique perspectives. I adore and love you both tremendously. Dr. Sheri Massey, thanks for your insight and support, it helped make the first step of this journey an easy one. To my circle of support and heartwarming cheerleaders (especially during the last stretch) –Thank you for everything; I truly appreciate you all: Tashona Beale, Kelly Gainer, Monique Lindsay, Alfreda McCray, Dr. Cynthia Winston-Proctor, Dr. Sonya Smith, Dr. Paula Whetsel-Ribeau, Dr. Rashunda Stittt, Dr. Kamilah Woodson, Jason Jones, Melissa Wynn, and all the HU ADVANCE-IT STEM Women Faculty!!! Without question, I am truly appreciative of the families and students that participated in this study. Your support made this possible! SPECIAL THANKS to Dr. Tony McEachern for coming through for me in a pinch; may the spirits of learning and exploration ever remain in those two computer labs to which you provided access. I appreciate your interest, support, and cooperation. Hats off to Dr. A. Wade Boykin, Dr. Jennifer Burrell, and Dr. Alicia Anderson for introducing me the culturally relevant theories used in this study (namely, the Black Cultural Ethos, Communal Learning, and Black Academic Identity). Thanks to Bailey Boo Puppy for her never ending company while writing, no matter the hour. And last but certainly not least, my sincerest thanks and gratitude go to the best advisory team any graduate student could ever have: Dr. Brenda Bannan, Dr. Len Annetta, Dr. Erin Peters-Burton, and Dr. Kevin Clark, my esteemed Chair.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abstract

COMMUNAL LEARNING VERSUS INDIVIDUAL LEARNING: AN EXPLORATORY CONVERGENT PARALLEL MIXED-METHOD STUDY TO DESCRIBE HOW YOUNG AFRICAN AMERICAN NOVICE PROGRAMMERS LEARN COMPUTATIONAL THINKING SKILLS IN AN INFORMAL LEARNING ENVIRONMENT

Leshell April Denise Hatley, Ph.D.

George Mason University, 2016

Dissertation Director: Kevin Clark

Today, most young people in the United States (U.S.) live technology-saturated lives. Their educational, entertainment, and career options originate from and demand incredible technological innovations. However, this extensive ownership of and access to technology does not indicate that today's youth know how technology works or how to control and use it to spawn innovation and create. The Computer Science Education (CSEd) research community recently made recommendations to help get young students more engaged in computer science, have longer exposure to the field’s concepts and practices, and thus use this longevity to persist through higher education and into computer science careers. However, low-income students and African American/Black students currently still have the least access to computer science learning opportunities when compared to that of all other student counterparts. More recommendations are

needed for targeting, reaching, and teaching computer science to this and all underrepresented populations. As such, the dissertation study presented here suggests and explores enhancements for the CSEd research community and CS educators to improve the teaching and learning of computational thinking and computer programming concepts for young African American students. These enhancements include: 1) using rigorous social science and education research methods, 2) focusing exclusively on underrepresented students (African American in this case), and 3) applying culturally relevant pedagogy.

In doing so, a convergent parallel mixed method research design is used to observe, describe, and compare how young African American novice programmers learn and use computational thinking and programming skills in two learning environments: 1) using culturally relevant pedagogy where students are assigned to a communal learning group where they work in pairs, and 2) an individual learning group where they work individually. Findings highlight participant performance outcomes, their strategies used while learning, as well as the resulting impact on their learning context preference and Black Academic Identity.

Chapter One: Introduction

"Today, there is more processing power and storage in the simplest mobile phone than in an entire university's computational arsenal of the 1970s. Children are immersed in a computational environment as users from a very early age and develop expectations of interaction which are often informed by interactive 3D games, among the most efficient, sophisticated, computationally complex, forms of interface yet developed. Powerful, capable machines are everywhere in their lives and it is a point of political will that '**computational thinking**' should be acquired by every student, in a way analogous to the '3R's' - reading, writing and arithmetic," (Wing, 2006).

"**African American children are the most likely consumers of digital technology but are rarely exposed to [learning] what it takes to create it**." – Leshell Hatley (Talbert, 2011)

"To teach, one must know the nature of those whom one is teaching." (Egyptian Proverb)

Today, most young people in the United States (U.S.) live technology-saturated lives. Their educational, entertainment, and career options originate from and demand incredible technological innovations. In April 2015, the Pew Research Center reported that 88% of U.S. teens (ages 13-17) personally have or have access to cell-phones, while 73% owned smartphones (Lenhart, 2015). This report also shared that 87% of teens have access to desktop/laptop computers, 81% have access to gaming consoles, and 58% have tablets (Lenhart, 2015). Regarding the usage of these mobile devices and consoles, Lenhart (2015) reports that 92% of the teens go online daily, with 56% going online several times a day; 72% play video games; 42% use video calling or chats; 52% use Instagram; and 71% use Facebook. Even more, it is predicted that the world in which they will work and engage in as adults will be filled with a plethora of computing devices, all having enormous amounts of computing power (Modi, Schoenberg, & Salmond, 2012; National Science Board, 2014; PCAST, 2010; Rising Above the Gathering Storm, 2007; Wilson, Sudol, Stephenson, & Stehlik, 2010; Wing, 2006). However, this extensive ownership of and access to technology does not indicate that today's youth know how technology works or how to control and use it to spawn innovation and create (Vaidhyanathan, 2008). In support of this, the National Academy of Sciences (2007) released a report generally warning "that Americans may not know enough about science, technology, or mathematics to contribute significantly to, or fully benefit from, the knowledge-based economy that is already taking shape around us," ("Rising Above the Gathering Storm," 2007, p. 94). Subsequently, a report released in 2010, entitled "*Running on Empty: The Failure to Teach K–12 Computer Science in the Digital Age,"* particularly highlighted the low numbers of women and people of color participating in computer science along with the few K-12 schools that had standards for teaching computer science, especially at the secondary school level (Computer Science Teachers Association, 2010). This news came on the heels of the creation of the Broadening Participation in Computing program by the National Science Foundation, between 2006 and 2009, to specifically increase engagement and retention of underrepresented populations (i.e. African-Americans, Latino-Americans, Native Americans, and Women) in computing disciplines across the country (NSF Broadening Participation Working Group, 2014). However, not much has changed since its formation. More explicitly, African-Americans, Latino-Americans, and Native

Americans are underrepresented in computer science across the United States (U.S.) within all levels of education and the workforce (Corney et al., 2010; Crutchfield et al., 2011; Freeman et al., 2014; Goode, 2010; Kolikant, 2012; Patil & Patil, 2002; Peckham et al., 2007; Scott et al., 2010; Google, 2015; Webb, Repenning, & Koh, 2012). Respresentatives of the U.S. government believe that [its citizens] not knowing how computers work, and more importantly, not having the technological skills needed to create using computing concepts and devices, threatens the future of innovation and problem-solving in the United States (PCAST, 2010; "Rising Above the Gathering Storm," 2007). Interest in finding a solution to these deficits, and thereby eliminating this threat, has caused educators, researchers, policy-makers, and government agencies to eagerly seek to learn, implement, and scale new and effective methods for teaching and learning these and other computing skills (e.g. computer programming) in hopes of revitalizing our economy and our global competitiveness (Weinberg, 2013; White House Office of Science and Technology Policy, 2014).

In 2011, the Computer Science Education (CSEd) research community responded to the 2007 K-12 school evaluation portion of the *Running on Empty* report, recommending a new idea: that elementary and middle school students be introduced to computer science concepts (Barr & Stephenson, 2011; Fessakis, Gouli, Mavroudi, 2013; Franklin, Conrad, Aldana, & Hough, 2011; Franklin et al., 2013; Seiter & Forman, 2013). The hope was that young students will specifically become more engaged, have longer exposure to the field's concepts and practices, and use this longevity to persist through higher education and into computer science careers. Consequently, many CSEd researchers and K-12 education practitioners followed this advice. As a result, the number of programs and approaches designed to teach computer science concepts to young people has continuously grown since 2011. In the midst of this growth, however, low-income students and African American/Black students currently still have the least access to computer science learning opportunities when compared to that of all other student counterparts (Google & Gallup, 2015). More recommendations are needed for targeting, reaching, and teaching computer science to this and all underrepresented populations. As such, the research study presented here suggests and explores enhancements for the CSEd research community and CS educators to improve the teaching and learning of computational thinking and computer programming concepts for young African American students. These enhancements include: 1) using rigorous social science and education research methods, 2) focusing exclusively on underrepresented students (African American in this case), and 3) applying culturally relevant pedagogy.

Background

The generation of youth born between 1980 and 1994 are described as ‘digital natives’ (Prensky, 2001), the ‘net generation, (Tapscott, 1999), and the millennials (Strauss & Howe, 2000) because they were born and have been surrounded by all forms of technology their entire lives – computers, video games, digital music players, video cameras, cell phones, technology-based toys, and more. This section shares more details about this generation, what they may and may not know about computing, along with what they do and do not do with technology. Statistics reflecting the lack of formal participation in computer science in academia and in the workplace are then provided and efforts put forth by the U.S. to increase participation of underrepresented populations in Computer Science are described. This section ends with an introduction to the computing concepts that, once learned, can potentially increase the number of young people successfully studying computer science.

The Technology-Saturated Lives of Tweens and Teens in the U.S.

Focused on tweens (ages 8-12) as well as teens (ages 13-17) of all races and ethnicities, the 2015 Common Sense Media Census reported that 67% of tweens and teens owned their own smartphones (Rideout, 2015). This census also stated that 56% of tweens and 63% of teens personally have or have access to desktop or laptop computers in the home, that 80% of tweens and 73% of teens have or have access to tablet mobile devices, and that 79% of tweens and 84% of teens have or have access to smartphone mobile devices (Rideout, 2015). Furthermore, Rideout (2015) concludes that tween and teen media consumption is highly mobile. The Pearson 2014 National Report on Student Mobile Device Survey for Grades 4-12 corroborates this data suggesting that 80% of all youth use mobile devices. Table 1 below summarizes these ownership data.

Table 1

Device Ownership (have or have access to) amongs Tweens and Teens

Population Source	Common Sense Media (Tweens)	Common Sense Media (Teens)	Pew Research Center (Teens)
Desktop/Laptop	56%	63%	87%
Smartphone	79%	84%	73%
Tablet	80%	73%	58%
Gaming Console	81%	83%	81%

In the classroom. Regarding classroom usage of mobile devices, The Pearson 2014 National Report on Student Mobile Device Survey states that 66% of elementary school students, 58% of middle school students, and 43% of high school students reported regularly using a tablet mobile device in school (Poll, 2014). Smartphone usage in schools reported by these same groups was 44%, 58%, and 75%, respectively (Poll, 2014). Of all the students surveyed for this 2014 Pearson report, 90% believe that tablets will change the way students learn in the future, 89% believe that tablets make learning more fun, and 79% believe that tablets help students do better in class. One-in-six students reported that their school provides them with dedicated computing devices for school work (Poll, 2014). This 2014 Pearson Report also states that 43% of tweens and 73% of teens report using computing devices for homework. Even more, most students want to use mobile devices in the classroom more than they currently did when they were surveyed, especially elementary school students at 71%. Overall, the data suggest that the personal usage of computing technologies described above extends well into the classroom.

For entertainment. When it comes to entertainment and game play, the Entertainment Software Association (2015) posits that 4 out of 5 American households own a device used to play video games, while 51% of all American households own a dedicated game console. The Common Sense Census (2015) reports that 81% of tweens and 83% of teens have or have access to game consoles in the home, and the Pew Research Center (2015) agrees, suggesting that 81% of teens have or have access to dedicated game consoles in the home. The Pew Research Center (Lenhart, 2015) also reports that 72% percent of American teens play video games. Additionally, digital games are used by 74% of K-8 teachers in the classroom, while 56% of parents say video games positively affect their children.

Tweens' and teens' access to and ownership of various technologies and computing devices is inescapably pervasive, with relatively low variance between White, Black, and Latino youth or household income levels (Lenhart, 2015). With these levels of access, assumptions and predictions suggest that the relationship between today's youth and technology is somehow enhanced when they communicate, learn, and create (Helsper & Eynon, 2010; Strauss & Howe, 2000). This is not always be the case however, as the next section describes.

Ownership of Technology and the Use of Technology to Create, Direly Unmatched

Amid this high ownership of and access to technology, young people's knowledge of exactly how computing technologies work and their ability to effectively use them to create, solve problems, and innovate, are direly unmatched. Vaidhyanathan (2008) provides a more specific description of this phenomenon by stating, "Every class has a handful of people with amazing skills and a large number who can't deal with computers at all. A few lack mobile phones. And almost none know how to program or even code text with Hypertext Markup Language (HTML). Only a handful come to college with a sense of …the Internet "Para. 5). Even though 95% of teens use the internet (Rainie, 2014) and more than half of teens (56%) go online several times a day (Lenhart, 2015), the majority of this time online is passive and based on interactive consumption, where tweens and teens alike are merely watching television shows, watching music videos, using social media, and/or playing games (Common Sense Census, 2015; Lenhart, 2015; Poll, 2014; Rainie, 2014). The 2015 Pew Report supports this observation affirming that tweens and teens only spend 3% of their time using technology to create. This lack of computing knowledge and the seemingly resulting low amount of time spent using technology to create and/or innovate significantly reflects the country's staggeringly low participation of U.S. citizens formally studying computer science (College Entrance Examination Board, 2015; Broadening Participation Working Group, 2014).

Low Participation in Computer Science

The U.S. has extremely low numbers of students formally studying computer science disciplines by citizens of all races and at every level of education. For example, on the high school level, the College Entrance Examination Board (2015) reports that 48,994 students took the Advanced Placement (AP) Computer Science exam. Although increasing by ~25% when compared to the number of students who took the AP Computer Science exam in 2014, this number is considerably low number when compared to the vast amounts of students taking AP exams in other topics. For instance, 469,689 students took the AP US History exam, 305,532 students took the AP Calculus AB exams, and 527,274 students took the AP English Literature and Composition exam (College Entrance Examination Board, 2015). These numbers are significantly larger than the total number of high school students taking the AP computer science exam. Additionally, in higher education, fewer students than ever are studying computer science despite the demand and projected growth of CS careers between 2008 and 2018 (Bureau of Labor Statistics, 2010; Corney et al., 2010; Fletcher & Lu, 2009; Piteira & Costa, 2012). Across the nation, there is an extremely low amount of participation in computer science careers as well, with the exception of white men (National Center for Science and Engineering Statistics, 2015).

Low Participation of Underrepresented Groups in Computer Science

There are extremely low rates of participation by underrepresented groups (e.g. African American, Latino American, and Native American) in Computer Science. According to the 2014 Taulbee Report (Computing Research Association, 2015), the principal source of information on the enrollment, production, and employment of those in Computer Science and Computer Engineering, 1.1% of Ph.Ds. in Computer Science and 1.5% of Ph.Ds. in Computer Engineering were awarded to African-Americans/Blacks, while 0.9% of Ph.Ds. in Computer Science and 1.5% of Ph.Ds. in Computer Engineering were awarded to Latino-Americans, and 0.1% Ph.Ds. in Computer Science and 0% Ph.Ds. in Computer Engineering were awarded to Native Americans (Computing Research Association, 2015). These statistics are additionally low at the Masters level, where 1.2% of the Masters degrees in Computer Science and 1.1% of the Masters degrees in Computer Engineering were awarded to African-Americans/Blacks, 1.8% of Masters degrees in Computer Science and 3.1% of Masters degrees in Computer Engineering were awarded to Latino- Americans, and 0.1% of Masters degrees in Computer Science and 0.2% of Masters degrees in Computer Engineering were awarded to Native Americans (Computing Research Association, 2015). The statistics regarding the number of Bachelor degrees in Computer Science and Computer Engineering are the same, with 3.2% and 3.3% being awarded to African-Americans/Blacks, 6.8% and 8.4%

to Latino-Americans, and 0.4% and 1.0% to Native Americans respectively (Computing Research Association, 2015). Furthermore, of the small number of total students who are majoring in Computer Science and Computer Engineering combined, 5.6% are African-American/Black, 9.6% are Latino-American, and 0.4% are Native-American (Computing Research Association, 2015). Across the nation in 2010, only 2%-3% of Black, Caucasian, *and* Latino freshmen combined indicated an intention to major in computer science while in college (National Science Board, 2014). In high schools across the country, the statistics of those taking the AP Computer Science exam by gender are 81.4% male and 18.6% female. In 2013, 0.89% of the number of female high school students who took the Advanced Placement (AP) Computer science exam was African-American. Even more, looking at the 2013 AP Computer Science exam by race and ethnicity alone, 46.39% *were Caucasian-American/White (non-Hispanic)*; 3.68% *were African-American/Black*, 8.14% were *Latino-American*, and 0.43% was *American Indian* (Exploring Computer Science, 2014). These statistics are displayed again in Table 2 below. Upon comparative review, there is an enormous mismatch between the statistics of technology ownership and access (Table 1 above) and those of academic pursuits in Computer Science and Computer Engineering (Table 2 below), especially as it relates to race and ethnicity.

Table 2

Percentage of Underreprented Populations in Computer Science (CS) and Computer Engineering (CE) in Academia

Population	AP Exam CS	College Majors CS & CE	Earned Bachelors CS	Earned Bachelors CE	Earned Masters CS	Earned Masters CE	Earned Ph.D. CS	Earned Ph.D. CE
African-Americans/Blacks	3.68%	5.6%	3.2%	3.3%	1.2%	1.1%	1.1%	1.5%
Latino-Americans	8.14%	9.6%	6.8%	8.4%	1.8%	3.1%	0.9%	1.5%
Native-Americans	0.43%	0.4%	0.4%	1.0%	0.1%	0.2%	0.1%	0.0%

These low participation statistics also reach beyond the classroom. The statistics released by the National Science Foundation early in 2015 show that the Computer Science Workforce consists of approximately 65% Caucasian American/White (non-Hispanic), nearly 6% African American/Black, 5% Latino American, and 0.2% American Indian workers (NSF, 2015). The 2013 of the U.S. Census Bureau reported similar statistics in its Disparities in STEM Employment by Sex, Race, and Hispanic Origin report (2013).

Recognizing these statistics forced the country into action towards attracting and retaining more citizens in computer science careers. Subsequently, a subset of the country's educators, policy makers, and CSEd researchers seek to find effective teaching and learning approaches to broaden the participation of African American, Latino American, and American Indian K-12th grade students in computer science, also known as Broadening Participating in Computing (BPC) (Cooper, Grover, Guzdial, & Simon, 2014); Doerschuk, Liu, & Mann, 2011; Fletcher & Lu, 2009; Grover, Cooper, & Pea, 2014; Grover & Pea, 2013; NSF Broadening Participation Working Group, 2014; Webb et al., 2012; White House Office of Science and Technology Policy, 2014; Wilson, Solo, Stephenson, & Stehlik, 2010; Wing, 2006).

In response, the private sector allocated $240 Million to the overall U.S. 2015 fiscal year science, technology, engineering, and mathematics (STEM) education budget, specifically targeting the educational and entertainment sectors of children from underrepresented groups (Wasserman, 2015). One notion supporting this funding, reminiscent of the recommendation put forth by CSEd research community mentioned above, is that if more diverse and underrepresented students are introduced to computer science and can effectively think computationally at an early age, then they would be more interested in majoring in computer science upon graduating from high school, be more successful in their introductory classes at the bachelor level once enrolled, and be interested in pursuing higher education degrees and/or advanced careers in computer science (Gilbert, 2006; National Science Foundation, 2013; NSF Broadening Participation Working Group, 2014; Rankin, Thomas, Brown, & Hatley, 2013; Repenning & Ioannidou, 2008; Seiter & Foreman, 2013).

Increasing Participation in Computer Science

The country's explicit efforts to increase participation in computer science fields overall began a little less than a decade ago, following two official simultaneous streams of evaluation. One evaluation was the result of a U.S. government funded examination of its competitiveness in the global marketplace, and the second evaluation came from members within the computing and computer science education community after it began evaluating the impact and use of computing technology by society (Cooper et al., 2014). The next section describes these simultaneous evaluations, which identified the need to increase participation in computer science, and why the U.S. government partnered with the CSEd community in hopes of finding strategic solutions to do so.

Evaluation from the U.S. Federal Government. In 2007, the National Science Foundation (NSF) examined the erosion of the U.S. advantages in the fields of science and technology in its "*Rising Above the Gathering Storm: Energizing and Employing America for a Brighter Economic Future*" report (2007). This report urgently called for a federally coordinated effort to dramatically enhance the country's competitiveness. Since this initial observation, funding and additional evaluation efforts in all branches of the federal government have been dedicated to bolster competitiveness in science, technology, engineering, and mathematics (STEM) fields overall (Federal Inventory of STEM Education, Fast-Track Action Committee, Committee on STEM Education, & National Science and Technology Council, 2011; PCAST, 2010). Typically, this equates to more and more funding for educational programs each year. In fact, in the fiscal year 2015 budget, the United States federal government allocated $2.9 Billion to STEM education efforts to prepare the country's youth for careers in the 21st Century (White House White House Office of Science and Technology Policy, 2014). As such, many federal agencies partnered with the CSEd research community in search for effective strategies to increase participation in computer science fields.

Evaluation from the CSEd research community. The Computer Science Education (CSEd) research community is a collective body of interdisciplinary researchers, mostly from CS & CE departments on college and university campuses and professional organizations, K-12 teachers, and professionals from technology companies of all sizes and types (Denning, 2007; Grover & Pea, 2013; Guzdial, 2008; Joy, Sinclair, Sun, Sitthiworachart, & López-González, 2009. It has been in existence since the late 1960s and is particularly concerned with curriculum issues; practical and at times theoretical pedagogy; systems and technologies used for instruction delivery, which is the most common focus; and sometimes social and psychological factors (Barr & Stephenson, 2011; Joy, Sinclair, Sun, Sitthiworachart, & López-González, 2008; Robins, 2015).

In 2011, the CSEd research community realized it was no longer sufficient to wait until students entered college to introduce them to in depth computational thinking and programming skills. As such, several federal, regional, and local policies were suggested (Barr & Stephenson, 2011). These policies included: 1) presenting a single message at all levels regarding the importance of computational thinking in K-12 education; 2) encouraging computer science professionals and teachers to create and advocate for K-12 standards; 3) incorporate computational thinking skills throughout the entire K-12 experience; 4) attach computational thinking to existing policies; 5) include computational thinking in all teacher pre-service preparation programs; 6) encourage school administrators to incentivize the adoption of these curricular and pedagogical changes; and 7) provide teachers with resources to support an increase in computer science instruction, such as materials, activities, websites, and professional development (Barr & Stephenson, 2011; Fessakis, Gouli, Mavroudi, 2013; Franklin, Conrad, Aldana, & Hough, 2011; Franklin et al., 2013; Seiter & Forman, 2013).

These conclusions by U.S. government and the CSEd research community made it clear that concerted efforts were needed to increase participation, of all U.S. citizens, in computer science. The U.S. government provided funding for programs and research to be done by those in the CSEd research community, and the CSEd research community made recommendations to educational institutions on all levels – K-12, undergraduate, and graduate (Grover & Pea, 2013; Joy, Sinclair, Sun, Sitthiworachart, & López-González, 2008; Robins, 2015; Weinberg, 2013). However, it is worth noting that the CSEd research community did not include improved policies and practices for itself in this 2011 evaluation.

Problem Statement

The statistics of low participation in computer science, both in academia and the workplace - particularly regarding those in underrepresented populations - represent and present major challenges regarding the future of American youth, their ability to innovate, and ultimately of the country's economy and global competitiveness. Effective strategies to combat these challenges are in order. Attempting to find solutions should be one of the primary concerns of the CSEd research community, as they are the most likely qualified community to do so.

Nonetheless, considering the current difficulties within the CSEd research community, finding a solution may not come easy (Grover & Pea, 2013; Joy, Sinclair, Sun, Sitthiworachart, & López-González, 2008; Robins, 2015; Weinberg, 2013). As a field of study, CSEd research is comparatively young and difficult to do well because of several reasons: computer science topics are inherently challenging to learn and to teach (Robins, 2015), computing technologies are ever evolving (Joy, Sinclair, Sun, Sitthiworachart, & López-González, 2008) and, as a result, the academic landscape intended to deliver instruction is constantly changing (Joy, Sinclair, Sun, Sitthiworachart, & López-González, 2008; Robins, 2015). Furthermore, after decades of research on core topics in computer programming, the CSEd research community still does not have a consensus of why so many novice programmers fail to learn, what best approaches to take as a result, or even the most optimal curriculum order with which to teach (Robins, 2015). Additionally, members of the CSEd community primarily deem themselves to be computer scientists and practitioners and may have most likely found their way to CSEd by accident, as a result of bad learning experiences (Robins, 2015). In fact, many CSEd researchers are not familiar with the landscape of instruction delivery or methods of education research (Grover & Pea, 2013; Joy, Sinclair, Sun, Sitthiworachart, & López-González, 2008; Robins, 2015; Weinberg, 2013). Furthermore, according to a report by the National Research Council (NRC, 2011c, p.4), there is a scarcity of research informing how to teach computational thinking in the early grades and computer science is often taught without consideration for age-appropriate learning. Taken together, these phenomenon are of particular interest to the study presented here.

More specifically, this study addresses the following three CSEd research problems described in the research literature:

1) the lack of studies using rigorous social science and education research methods (Grover & Pea, 2013; Joy, Sinclair, Sun, Sitthiworachart, & López-González, 2008; Mannila et al., 2014; Randolph et al., 2008; Robins, 2015; Sheard, Simon, Hamilton, & Lönnberg, 2009; Weinberg, 2013).

2) the lack of CSEd research that explicitly identify and/or focus on underrepresented populations in computer science as study participants (Grover & Pea, 2013; King, 2005); along with

3) the lack of using culture and culturally relevant pedagogy as a means to teach computational thinking (King, 2005; Scott, Sheridan, & Clark, 2014).

Purpose of Research Study

The purpose of this study results from combining the goals of the Broadening Participation in Computing (BPC) program - increasing and sustaining the number of underrepresented populations in computer science - with the recommendations made by the CSEd research community in 2011 to introduce computer science concepts to students early in elementary and middle school to the increase in their participation in computer science. It has four parts. The first purpose is to recommend three equally important research strategies to be used by the CSEd research community in an effort to directly counter the three problematic challenges described in the previous section:

1) to rigorously make use of the educational and social science research methods and statistics that are necessary and ideal for dealing with education research, comprised of real students in real classrooms;

2) to use these rigorous research methods in the deliberate design of empirical studies to understand, share, and contribute to how students in underrepresented groups (e.g. African American, Latino American, and Native American) learn, apply, and retain computing topics, and

3) to intentionally explore and apply culturally relevant pedagogy as a means for underrepresented groups to successfully learn computer science content.

The second purpose is to explore a specific implementation of these three recommendations within one study. As such, this study uses a mixed-methods education research approach to empirically study 42 young African American/Black elementary and middle students learning computational thinking and programming skills using culturally relevant pedagogy. The culturally relevant pedagogy used here is that of communalism. As a result, the third purpose is to extend the trajectory of educational research regarding communalism and communal learning to the teaching and learning of computational thinking and programming skills. Finally, the fourth purpose of this study is to explore the impact learning such skills has on the participants' Black Academic Identity (BAI).

As a result, Social Development Theory (Vygotsky, 1980), Communalism (Boykin, Jagers, Ellison, & Albury, 1997), Pair Programming, and the theory of Black Academic Identity (Anderson & Freeman, 2010) frame this study and are all described in the next section.

Theoretical Framework

Vygotsky's *Social Development Theory* (1980) describes learning as a social process based on three major themes: 1) social interaction – connections people make during shared experiences, 2) the more knowledgeable one (MKO) – the one interacting with the learner who knows and understands more, and 3) the zone of proximal development (ZPD) – the area between when a learner can solve a problem with the guidance of a teacher or assistant and when the learner can solve a problem his/her own. Vygotsky believed this is where learning occurs. Overall, Vygotsky believed that *social interaction*s play a fundamental role in the development of cognition.

"Every function in the child's cultural development appears twice: first, on the social level, and later, on the individual level; first, between people (interpsychological) and then inside the child (intrapsychological). This applies equally to voluntary attention, to logical memory, and to the formation of concepts. All the higher functions originate as actual relationships between individuals" (Vygotsky, 1980).

During the same time period, Vygotsky's Social Development Theory is mirrored by the description of *Communalism,* one of the nine elements of the Black Cultural Ethos (cultural characteristics), that Boykin (1977) posits as a distinctive cultural phenomenon that contributes to and enhances the academic performance of African American students.

These nine elements, including Communalism, are briefly described below:

- *Spirituality* – intuition, supreme force
- *Harmony* – versatility and wholeness
- *Movement* – rhythm of everyday life
- *Verve* – intense stimulation, action, colorfulness
- *Affect* – premium on feelings, expression
- **Communalism** – social orientation, group duty, sharing, identity***
- *Expressive individualism* – distinct, genuine, personal
- *Orality* – oral and aural modes of communication
- *Social time perspective* – time is marked by human interaction

More explicitly, Communalism has four dimensions (Boykin, 1986; Hurley, Boykin, & Allen, 2005):

1. Social Orientation - where the individual is oriented toward social relations and holds each social interaction as a valuable experience;
2. Group duty - where the person believes that the needs of the group supersede the needs of the individual;
3. Sharing - where exchange and mutual support are understood to be intrinsically rewarding in that they signify that participants contributed to the group; and
4. Identity - where the individual has a sense of belonging and selfhood based on group membership.

Communalism places a premium on the culture of the participants and places them within a learning context that aligns with their cultural inclinations. As such, this study follows a line of previous research investigating the impact that communalism has on African American student performance in the classroom and that has categorized these learning spaces as Communal Learning environments (Albury, 1991; Boykin, Coleman, Lilja, & Tyler, 2004; Coleman, 1996; Coleman, 2001; Dill & Boykin, 2000; Hurley, 1999).

In these studies, Communal Learning is compared to Individual Learning, where students work individually. With respect to learning computational thinking and programming skills in this study, communal learning was implemented using the concept of pair programming, where groups of two students were a pair of participants. Pair programming, first used by Frank Brooks – author of the Mythical Man Month, while he was in graduate school between 1953-1956 (Brooks, 1975), is at the root of a collaborative software development approach called 'extreme programming,' intended to improve quality and responsiveness to customer needs. Pair programming requires that teams of two programmers work simultaneously at the same computer on the same design, algorithm, code/program, or test (McDowell, Werner, Bullock, & Fernald, 2002; Nosek, 1998; Werner, Denner, & Campe, 2012; Williams, & Kessler, 2000).

Finally, recognizing that identity plays a role in the academic achievement of African American students (Oyserman, Harrison, & Bybee, 2001), the Theory of Black Academic Identity is also used here. This theory suggests that for some African American students, achievement is racialized, thereby combining academic identity with racial identity (Anderson & Freeman, 2010). Thus, African American students mix the meaning of race in their lives with the importance of performing well academically at varying degrees. Those with a high Black Academic Identity score do so more than those with a low Black Academic Identity score.

The objective of the study is to determine to what extent does using a communal learning environment enhance the computational thinking and programming skills of young African American novice programmers. In so doing, this study also examines the extent to which Black Academic Identity expresses itself. This theoretical framework is illustrated in Figure 1 below.

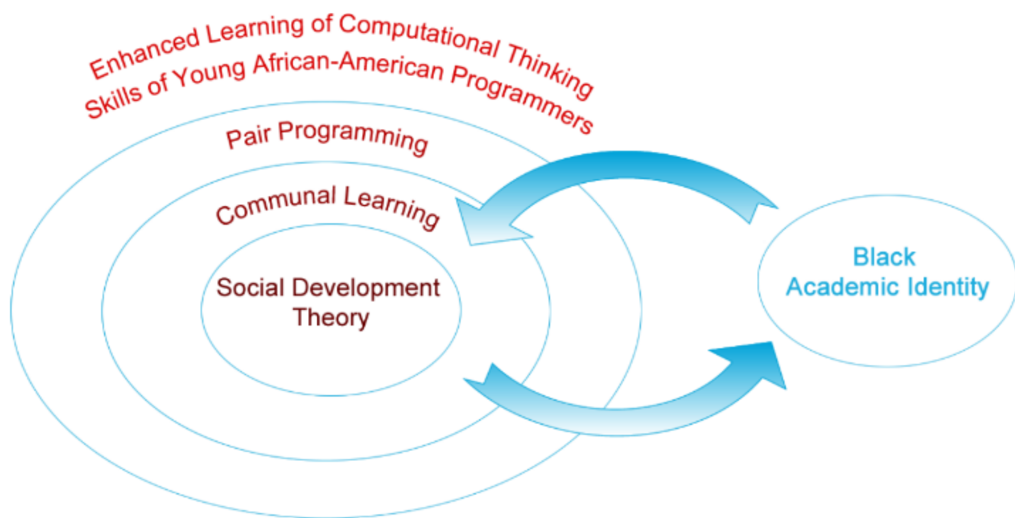


Figure 1. Theoretical Framework for this dissertation study.

Goals of the Study

The goals of this dissertation study were to capture and describe the learning processes of young African American novice programmers as they learn computational thinking skills using the Scratch programming platform. For a camp lasting 5 week days, 3 hours per day, study participants were placed in one of two learning contexts: 1) a Communal Learning group, where they were paired with a peer, and 2) an Individual Learning group, where participants worked independently of one another. This study captured the learning experience each week and allowed for a description and comparison of similarities and differences between grade-level and gender, within and across both learning groups. Finally, this study explored the impact learning computational thinking and programming skills had on participants' Black Academic Identity.

In the end, this study contributes to both the CSEd research and communal learning literature regarding the teaching and learning of computational thinking and programming skills for young Africa-American novice programmers.

Significance of This Study

This dissertation study is significant in many ways. Primarily, using Communalism in this study values the cultural assets of participants and is a strength-based approach that provides developmental and supportive mechanisms to promoting success. This is especially important because many learning environments have a deficit model approach, which assumes that cultural assets hinder success and therefore should be mitigated (Boykin, 1986). It supports the notions that "traditional classroom lecture methods are not preparing our youth for the challenges of the coming global change; [and that] we need to teach differently" (Johnson, Peters-Burton, & Moore, 2015). Secondly, the research on computational thinking and programming rarely includes or explicitly describes a population where 100% of the students are African-American. All participants of this study will be African-American.

Moreover, this study provides an instance of a rigorous mixed-method education research study to the computing education literature, an occurrence many researchers found does not happen very often. Ultimately, this dissertation study provides insight into the ways in which African American students learn computational thinking and programming skills and should help bring the computing education community closer to understanding how to broaden participation in computing and computer science.

Research Questions

The research questions driving this study are:

RQ1. During a summer camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context?

RQ2: Is there a change in the learning context preference of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?

RQ3: Is there a change in the Black Academic Identity of the young African American elementary and middle school novice programmers after participating in this Scratch programming camp?

Chapter Two: Literature Review

This chapter describes the literature concerning the two primary tenets that make up this proposed solution to increasing the participation of young African American novice programmers in computer science: 1) the use of communal learning contexts, a particular form of culturally relevant pedagogy for 2) teaching Computational Thinking and programming skills using Scratch, a visual programming language. As such, it describes the origins of Computational Thinking along with the challenges, research, and methods related to teaching young people how to program; the origins of Communal Learning; and the theory Black Academic Identity, a theory explaining one characteristic of African American student academic identity.

Human Computing and Computational Thinking

Understanding human computing, computational thinking, and how to effectively teach both are crucial to increasing participation in computer science.

Human computing. Computing is a natural human creative activity (Denning, 2007; Grover & Pea, 2013). Human computing, also known as human information-processing, involves the use of mental skills to analyze data, recognize patterns, create algorithms, and solve problems. Using computer software and hardware to apply these human information-processing skills to achieve a goal enables productivity, the design and implementation of a solution to a problem, and computing artifacts (Grover & Pea, 2013). Examples of these computational artifacts include computers themselves, computer programs (i.e. software), automated hardware and other devices (i.e. robotics and wearable technologies), augmented reality, and interactive websites, animations, simulations, games, data analysis, and more. Furthermore, these resulting computational artifacts can, in turn, be used for more computing, information processing, and to create computational artifacts that span all academic topics and activities in everyday life (Denning, 2007). The application and skills needed to create the resulting computing artifacts combined with the ability to work collaboratively with others and communicate using computing vocabulary describe what it means to *deal with computers* quoted above in chapter one. These skills are also known as *computational thinking skills* (Barr & Stephenson, 2011; Grover & Pea, 2013).

Computational thinking. As one of the opening quote of this document illustrates, Wing (2006) proposes that Computational Thinking be a way of thinking and problem-solving for everyone, not just for computer scientists in the March 2006 edition of the Communications of the ACM (Association for Computing Machinery), the leading professional organization for computer science researchers and practitioners. This 2006 article is regarded as the starting point for exploring computational thinking in the 21st century as well as how it can be taught to increase engagement and persistence in computer science fields.

Origins of Computational Thinking

College. During the 1960s, Allan Perlis, winner of the Association for Computing Machinery (ACM) 1st Turing Award (the association's most prestigious technical award), strongly suggested that college students of all disciplines learn the 'theory of computation' (Grover & Pea, 2013; Guzdial, 2008). Perlis "argued that programming was an exploration of process, a topic that concerned everyone, and that the automated execution of process by machine was going to change everything" (Guzdial, 2008). He proposed that understanding the theory of computing would lead to the understanding of many other topics such as economics and calculus (Guzdial, 2008).

Grades K-12. Within the K-12 context, the 1980s witnessed Seymour Papert's work with the LOGO programming language and its use in teaching youth how to program (Papert, 1980). His work lead to the emergence of computational technologies for learning, insisting that children's cognitive evolution flourish with rich toolkits and environments (Blikstein & Wilensky, 2006; Papert, 1971). This emphasis is the focus of many digital learning tools used to teach children computational thinking and programming skills. These tools are described in more detail in a later section.

In the 21st century. In 2006, Wing broadly describes Computational Thinking as a way of solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science Wing (2006). This description of Computational Thinking revitalized Papert's motives regarding teaching children to program and using programming as a mechanism towards cognitive development and problem-solving. Yet, although extremely influential, Wing (2006) did not precisely define the term Computational Thinking or state exactly what it meant for everyone (Selby & Woollard, 2013). Since 2006, its definition has been refined and refined again

(Pears et al., 2007; Selby & Woollard, 2013; Sheard, Simon, Hamilton, & Lönnberg, 2009; Weinberg, 2013).

Defining Computational Thinking

In 2010, researchers at Carnegie Mellon University and Microsoft Research partnered to create the Center for Computational Thinking and refined the definition of Computational Thinking. Accordingly, Computational Thinking was defined as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent."

Additional definitions of Computational Thinking are also posted on the Center's website:

- "Computational Thinking is a way of solving problems, designing systems, and understanding human behavior that draws on concepts fundamental to computer science. To flourish in today's world, computational thinking has to be a **fundamental part of the way people think and understand the world**.
- Computational Thinking means **creating and making use of different levels of abstraction**, to understand and solve problems more effectively.
- Computational Thinking means **thinking algorithmically** and with the ability to **apply mathematical concepts** such as **induction** to develop more efficient, fair, and secure solutions.

- Computational Thinking means **understanding the consequences of scale**, not only for reasons of efficiency but also **for economic and social reasons**."

Generally speaking, however, many CSEd researchers agree that the following concepts and practices make up computational thinking and form the basis of computer programming curricula that aim to support its learning as well as assess its development (Grover & Pea, 2013; Lee et al., 2011; Selby & Woollard, 2013):

- Human thought processes/creative activity,
- Abstractions and pattern generalizations (hiding complexity),
- Systematic processing of information (both by the human and a computing device),
- Structured problem decomposition (modularizing),
- Iterative, recursive, and parallel thinking (simultaneously executing several commands and processing information),
- Systematic error detection and resolution (i.e. debugging), which is done routinely as one builds upon a computer programming solution,
- Symbol systems and representations (data representation),
- Algorithmic notions of flow of control (a step-by-step process), and
- Conditional logic (e.g. if this, than do that).

CSEd Researchers also agree that these concepts can be taught at the K-12 grade levels using a variety of subjects, including but not limited to: computer programming, mobile application development, game design, robotics, e-textiles, and modeling and simulation, (Cheung, Ngai, Chan, & Lau, 2009; Fletcher & Lu, 2009; Kafai et al., 2013; Lee et al., 2011; Pears et al., 2007; Wilson, 2003).

Computational Thinking in the 21st Century in K-12

The report by the President's Council of Advisors on Science and Technology, *Prepare and Inspire: K-12 Science, Technology, Engineering, and Math (STEM) Education for America's Future (2010, p.46)* states that students need "a deeper understanding of the essential concepts, methods and wide-ranging applications of computer science. Students should gain hands-on exposure to the process of algorithmic thinking and its realization in the form of a computer program, to the use of computational techniques for real-world problem solving, and to such pervasive computational themes as modeling and abstraction, modularity and reusability, computational efficiency, testing and debugging, and the management of complexity. Where feasible, active learning, higher-level thinking, and creative design should be encouraged by situating new concepts and techniques within the context of applications of particular interest to a given student or project team."

The Computer Science Teachers Association (CSTA) distinguished the above Computational Thinking skills in grades K-12 as the general ability to manipulate data, regardless of school topic. This manipulation includes the ability to *collect data, analyze data, represent data, decompose* problems into smaller sub-problems, along with being able to understand and create abstractions (reducing information and detail to focus on concepts relevant to understanding and solving problems), algorithms & procedures, automation, parallelization, and simulations (CSTA, 2009). Currently, these and other entities in the CSEd research community, such as ACM's Special Interest Group in Computer Science Education, the International Society for Technology in Education, the College Board, the National Science Foundation; Google CS4HS (Computer Science for High School), and the Grace Hopper and International Computing Education Research Conference, are working on newer, more centralized standards for various grade levels, new curricula (pre-advanced placement and advanced placement), along with new advanced placement tests at the high school level.

As a result, more explicitly defined skills have been described to provide evidence of Computational Thinking (Barr & Stephenson, 2011; PACT, 2012; Wing, 2011).

These specific skills include:

1. the ability to use computers to collect, analyze, and represent data; analytically decompose problems;
2. to understand and manipulate information in such a way that masks its complexity (i.e. abstraction);
3. to create and comprehend algorithms, procedures, and automation;
4. to understand and use a computer programming language; along with
5. the ability to have a computer (program) run various functions simultaneously (in parallel) and in simulation.

Although these skills are rooted in computer science, their power can and should be expanded and applied to innovation in all other disciplines (Barr & Stephenson, 2011). Moreover, this multifaceted and multidisciplinary suggests that Computational Thinking skills are fundamentally for everyone, promoting computer science to a wider audience, regardless of their ultimate field of study (Selby & Woollard, 2013). As such, strengthening Computational Thinking skills is especially imperative for citizens, as they prepare for the 21st century. Although computational thinking skills can be applied to fields beyond computer science, mechanisms for teaching these skills are best aligned with computer programming (Lakanen & Isomöttönen, 2015; Lee et al., 2011; Zander et al., 2009). For this reason, the study presented here uses computer programming to teach computational thinking skills to novice learners.

Computer Programming and Novice Learners

"Computer programmers write code to create software programs. They turn the program designs created by software developers and engineers into instructions that a computer can follow" (Bureau of Labor Statistics, U.S. Department of Labor*, 2014-15 Edition*, para. 1). When learning how to program, a novice computer programmer focuses on both of these functions: 1) design the program and 2) write the code/program (computer instructions) needed to create the desired program to accomplish a desired outcome. Scratch*,* a programming platform designed specifically for teaching young people how to program, is the platform this study used as the mechanism for teaching computational thinking and computer programming skills to young African American novice programmers in an informal learning environment.

Why Youth Should Learn How to Program

Computer programming as a viable activity to build problem-solving skills for youth (Clements & Gullo, 1984; Feldman, 2004; Fessakis, Gouli, & Mavroudi, 2013). Those programming environments hosted on the internet and used with via a web browser, such as Scratch, allow for sharing, collaboration, and the increased ethical knowledge and behavior when consideration remixing (i.e. redesigning) another programmer's work (Kafai, Burke, & Resnick, 2014). Kafai, Burke, and Resnick (2014) suggests that learning to program and making programs using these web-based environments move students from computational thinking to computational participation, emphasizing the social aspects of programming. These benefits give young programmers an early advantage to understanding programming concepts. If successful, teaching computational thinking and programming skills at an early age will provide evidence for the theory that introducing computer science to students at an early age increases potentially increases their likelihood to persist in studying computing and computer science and establishing them as possible career paths (Barr & Stephenson, 2011; Fessakis, Gouli, Mavroudi, 2013; Franklin, Conrad, Aldana, & Hough, 2011; Franklin et al., 2013; Seiter & Forman, 2013).

The Challenge of Learning How to Program

For the past six decades, many computer science instructors and education researchers report that learning to program is challenging (Brooks, 1983; Corney, Teague, & Thomas, 2010; Fletcher & Lu, 2009; Kelleher & Pausch, 2005; Piteira & Costa, 2012; Sheard, Simon, Hamilton, & Lönnberg, 2009). This wide recognition and acknowledgment that students just "don't get it" (cite) led to a number of tools and approaches for teaching and learning computer programming (Corney, Teague, & Thomas, 2010; Lemos, 1979; Piteira & Costa, 2012).

The above cited researchers and others have identified some perceived barriers to entry when learning how to program. They include, but are not limited to:

1. the difficulty of understanding the purpose of programs and their relationship with the computer (Robins, Rountree, & Rountree, 2003);
2. difficulty in grasping the syntax and semantics of a particular programming language (Robins, Rountree, & Rountree, 2003);
3. misconceptions of programming constructs (Soloway & Spohrer 1989);
4. inability to problem-solve (McCracken, Almstrum et al. 2001);
5. inability to read and understand program code (Lister, Adams et al. 2004; Mannila & de Raadt, 2006); and
6. Motivation (Corney, Teague, & Thomas, 2010)

As such, introduction to programming courses have been continuously redeveloped with changes in language, paradigm, and swapping between breadth and depth of content (Corney, Teague, & Thomas, 2010).

Teaching Youth How to Program

If learning how to program is challenging, in what ways can it be successfully taught to youth, especially when younger students differ developmentally than older students (Hill, Dwyer, Martinez, Harlow, & Franklin, 2015)? To answer, visual, drag-and-drop, block-based programming languages are thought to be a common and successful approach held by many in the CSEd community (Adams & Webster, 2012; Brennan & Resnick, 2012; Davis, Kafai, Vasudevan, & Lee, 2013; Duncan, Bell, & Tanimoto, 2014; Freeman et al., 2014; Grover, Cooper, & Pea, 2014; Kelleher & Pausch, 2005; Kelleher, Pausch, & Kiesler, 2007; Medlock-Walton, Harms, Kraemer, Brennan, & Wendel, 2014; Mbogo, Blake, & Suleman, 2013; Papert, 1971; Salleh, Shukur, & Judi, 2013; Sorva, Karavirta, & Malmi, 2013; Werner, Campe, & Denner, 2012). Examples of these environments include but are not limited to (in no particular order): Scratch (by MIT); App Inventor (by MIT & Google); Blocky (by Google), Alice & Storytelling Alice (by Carnegie Mellon); EarSketch (by Georgia Tech);, Snap and its predecessor, BYOB - Bring Your Own Blocks (by Berkeley); Kodu Game Lab (by Microsoft), and many more. Besides the fact that these visual programming environments are interactive and entertaining (Wilson, 2003), many of these environments allow code/programs to be shared and remixed (modified). Harms, Cosgrove, Gray, and Kelleher (2013) claims the ability to view code/programs shared on the Internet enhances the ability to comprehend and enhance one's ability to program. Lee et al. (2011) suggests a "use-modify-create" framework for teaching youth how to program, based on observations of young programmer's cognitive and practical activity when learning computational thinking skills.

Characteristics of Research on Youth Learning How to Program

With the advent of the strong need to teach computational thinking (CT) and programming skills, research is growing concerning the use of visual programming languages to teach programming and CT skills to elementary, middle, and high school students across the globe. Many of these studies, several of them specifically using Scratch, often focus on learning gains in particular topics, most with pre- and post-test measures; self-report student engagement levels; and at times, resulting descriptive study characteristics and best practices (Burke & Kafai, 2012; Davis, Kafai, Vasudevan, & Lee, 2013; Franklin et al., 2013; Maloney, Resnick, Rusk, Peppler, & Kafai, 2008; Tekerek & Altan, 2014; Weinberg 2013). It is worth noting that very few studies seek to understand the learning processes students undergo while learning a visual programming language.

Additionally, Weinberg (2013) reports that the studies on computational thinking between 2006 - 2011 lacked research rigor. Eighteen percent or less of the ~164 computational thinking studies conducted and analyzed based on specified criteria of the studies described by Weinberg (2013) during this period were completed by computer scientists and or computer science professors with coauthors who had little to no experience in social science research or education. This suggests, alternatively, that a majority of these studies were completed with no real social science research procedures. Many did not describe their research methods and several of them only reported post-test scores (Weinberg, 2013).

Benefits of Visual Programming Platforms

Using visual programming platforms has been proven to be better absorbed by young students learning to program as they do not have to worry about the syntax (i.e. specific programming formats and rules in text-based programming languages) of a particular language or the challenge of debugging their code/programs (Chang, 2014; Cheung et al., 2009; Hu, Winikoff, & Cranefield, 2013; Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010). It has been reported that visual programming languages eliminate the need to memorize syntax, as they generally implement the analogy of puzzle pieces sticking together. Many of these visual environments are colorful and inviting, potentially increasing their appeal and thus the learner's engagement. The success of these environments and others like them has prompted them to be used in introductory computational thinking and programming curricula over the past several years. Exploring Computer Science (ECS) is one example of such curricula. ECS is being piloted in several cities around the country. Additionally, other platforms are designed specifically to engage one gender or the other. For example, Storytelling Alice was specifically designed to support storytelling. Kelleher, Pausch, and Kiesler (2007) found Storytelling Alice motivated girls to spend more time programming and increased their interest in the future of the Alice product line than the generic Alice version. Burke and Kafai (2012) also found that combining writing with programming in Scratch expanded girls' view of the necessity to learn how to program.

Assessing Computational Thinking and Programming Skills

As the need to learn computational thinking skills becomes more and more apparent and the CSEd research community improves the definition of computational thinking, its assessment becomes more and more essential (Grover & Pea, 2013). As such, in 2012, the NSF funded the Principled Assessment of Computational Thinking (PACT) project to advance the field of assessment of high school computer science and computational thinking skills. PACT (2012) expanded the domains of computational thinking beyond computer science and programming concepts (those listed above) to include two additional components: 1) inquiry and 2) communication & collaboration. Table 1 illustrates the skills that make up each of these three components of computational thinking, as defined by PACT (2012).

Table 3

PACT's Definition of Computational Thinking Skills

CS Concepts	Inquiry Skills	Communication & Collaboration Skills
Algorithms	Evaluate	Publish
Programming	Explore	Present
Recursion	Analyze	Build Consensus
Abstraction	Explain	Discuss
Debugging/Testing	Elaborate	Distribute Work
Variables	Model	Lead/Manage Teams

Although still a work-in-progress, PACT suggests the following guidelines for assessing computational thinking skills in each of these components:

1. Analyze One's Own Computational Work and the Work of Others
2. Apply Abstractions and Models
3. Design and Implement Creative Solutions and Artifacts
4. Analyze Effects of Development in Computing
5. Connect Computing with Other Disciplines
6. Communicate Thoughts, Processes, and Results in Simple Formats
7. Work Effectively in Teams

NOTE: As of this writing, PACT has released an updated website which illustrates the notion that the ideas in the table presented above are now considered computational thinking 'practices' instead of computational thinking skills.

Introduction to Scratch: A Programming Platform for Novice Programmers

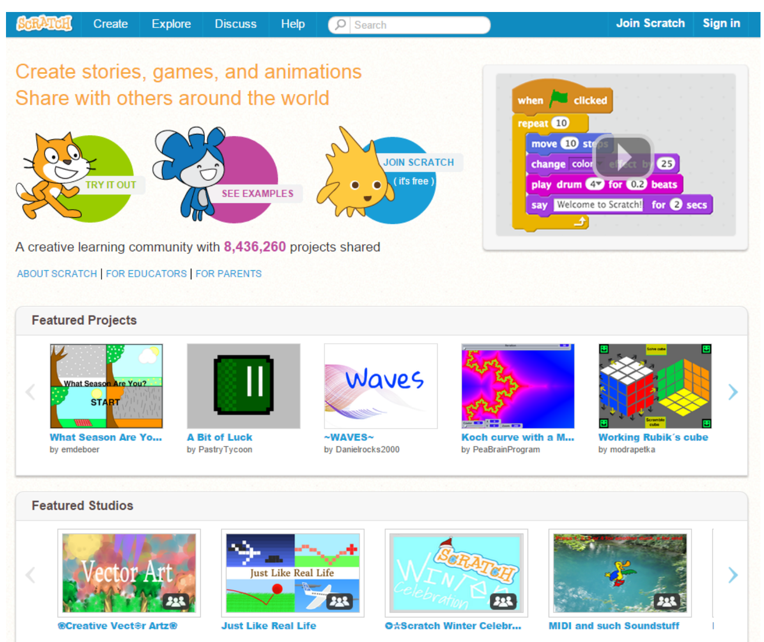


Figure 2. Screenshot of Scratch homepage. Retrieved from http://Scratch.mit.edu

Scratch was created in 2007 by the Lifelong Kindergarten Group at the MIT Media Lab. The project is led by Mitch Resnick, the group's director. Scratch is a visual programming environment that allows users (primarily ages 8 to 16) to learn computer programming while working on personally meaningful projects such as animated stories and games. Figure 2 above is a screenshot of the Scratch website homepage taken before this study began. A key design goal of Scratch is to support self-directed learning through tinkering and collaborating with peers (Maloney et al., 2010). Now at version 2.0, Scratch is an *online* visual programming language, user community, and learning environment used to teach computer programming concepts to students of all ages. Instead of typing text and using command line interfaces to create and run computer programs, learners drag and drop visual programmable bricks, which look like puzzle pieces on the computer screen, and snap them together, like Lego® bricks, in order to create a set or program block (multiple program puzzle pieces stacked together) of instructions to be executed. The only syntax required is that the puzzle pieces fit together. If the puzzle pieces do not logically fit together, they will not snap together either, giving instant feedback regarding a learner's programming logic. Figure 3 below illustrates the distinct difference between text and visual programming (text is on the left, visual programming block on the right).

Figure 3. Screenshot of Program Text and Scratch Puzzle Pieces (blocks) created to perform the exact same action.

Previous versions of Scratch (version 1.4) were downloaded and ran locally on a learner's computer, without the need for an Internet connection, except when a user wanted to upload and share the resulting programs, which are called Scratch projects. However, version 2.0 runs within a learner's web browser and requires an Internet connection. There is also a downloadable version of Scratch 2.0 in the event there is no Internet connection. At the time of this study, the Scratch website was home to over 7.1 million registered programmers (users) with Scratch accounts, who make up the Scratch Community and is where registered Scratch users upload and share their projects. Also, at the time of this writing, there are over 10.2 million Scratch projects. Scratch projects can be viewed, shared, executed and are designed to be 'remixed' and shared again by other Scratch community members.

Computational Thinking Skills Learned Using Scratch

The need to assess computational thinking skills has become apparent to the makers of Scratch as well. Brennan and Resnick (2012) suggested a framework for assessing computational thinking skills. These guidelines are very similar to the PACT (2012) domains of computational thinking skills (Table 3 above) and are referred to as concepts (sequences, loops, events, conditionals, operators, and working with data), practices (what Scratchers do and how they do them), and perspectives (how Scratchers view themselves and the world around them) – illustrated below in Table 4.

Table 4

MIT's Scratch *Team Definition of Computational Thinking Skills*

Concepts	Practices	Perspectives
Sequences Loops Events Conditionals Operators Working with Data	What Scratchers do and how they do them	How Scratchers view themselves and the world around them

The overlay of both PACT and Scratch computational thinking skills can be seen in Table 5 below.

Table 5

Overlay of PACT and Scratch *Computational Thinking Skills*

Source	Computational Thinking Knowledge	Computational Thinking Actions	Computational Thinking Soft Skills
PACT	Concepts	Inquiry	Communication & Collaboration
MIT/Scratch	Concepts	Practices	Perspectives

Scratch, Computational Thinking, and Computer Programming

Amidst the desire to assess the understanding and effective use of Scratch, Monroy-Hernández (2012) illustrates the frequency of use of each block within all projects in the Scratch community at the time at the time its of publication. Taken from this analysis, Figure 4 below shows a histogram illustrating the infrequent use of the very blocks that are associated with computational thinking – collecting, manipulating, and representing data. These blocks are represented in the histogram in figure as the List, Variables, and Numbers categories, to name a few. These particular blocks are used significantly less than many of the other more frequently used blocks (e.g. the Control, Looks, Motion blocks). This histogram shows that Scratch can be used to implement and assess computational thinking and programming skills, but Scratchers somehow lack the interest, knowledge, and/or efficacy needed to use *these* particular blocks. It should be noted that the analysis represented by this figure represents Scratch 1.4 usage and was conducted before the release of the online Scratch 2.0 version.

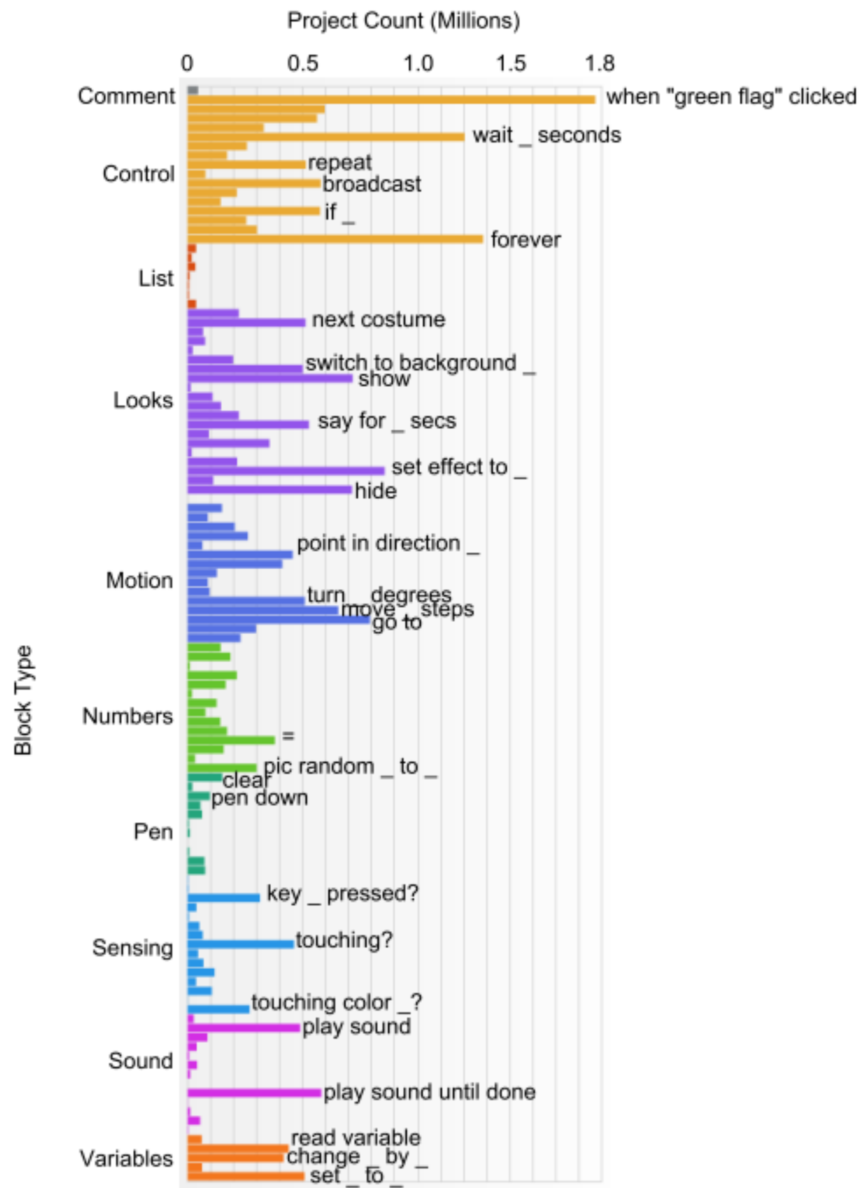


Figure 4. Histogram of frequency of block types used Monroy-Hernández (2012).

Assessing Computational Thinking Skills Learned Using Scratch

The Brennan and Resnick (2012) guidelines suggest the use of three facets for assessing computational thinking skills had by Scratcher. They include:

1. Analyzing the amount and complexity of the programming puzzle pieces used in a Scratchers projects (i.e. his/her project portfolio),

2. Introducing design scenarios and analyzing the Scratchers approach to solving or implementing them, and

3. Interviewing Scratchers about their various experiences using Scratch.

The study described here conducts all three of these suggested assessments to observe and compare how participants learn and use Scratch concepts as described above (i.e. sequences, loops, events, conditionals, operators, and working with data). The most recent illustration of these published guidelines used to assist Scratchers assessment of computational thinking skills is a new website found at http://Scratched.gse.harvard.edu/ct/index.html. This website defines what is meant by the computational thinking skills gained when using Scratch, namely concepts, practices, and perspectives. It also describes various approaches to assessing computational thinking skills using Scratch*,* described in the 2012 publication, along with curriculum guides and reflective techniques instructors can use to support the development of computational thinking skills. It should be noted that these guidelines and techniques are specific to the Scratch environment and may not be easily portable to other programming languages, visual or otherwise.

Summary of Computational Thinking and Teaching Programming Skills to Youth

Computing is a natural human activity. Using computer software and hardware to apply these human information-processing skills to achieve a goal enables productivity, the design and implementation of a solution to a problem, and computing artifacts. The

President's Council of Advisors on Science and Technology (PCAST) states that students need a deeper understanding of the wide applications of computer science. The CSEd research community agrees and pushes for introduction of computing and computer science principles as early as elementary and middle school. Considering the level of threat and non-competitiveness this country concludes from having low participation in computer science by all of its citizens, but particularly from those from extremely underrepresented populations (African American, Latino American, and Native American), it is imperative that policy makers, educators, researchers, and professional organizations work together to counter these low numbers. Teaching computer programming is challenging and researching ways how to do this effectively is rarely done well. Nonetheless, for many reasons, some CSEd researchers suggest that using visual programming languages, where students drag and drop code blocks to compose logical programs, has promise in introducing and familiarizing students to and with computational thinking and programming skills. This notion comes on the heels of Seymour Papert's work, as he insisted that children's cognition flourished with the use of rich toolkits and environments.

This study used Scratch, a visual programming language, to teach African American elementary and middle school students about computational thinking and programming skills. The specific computational thinking and programming skills focused on are: programming sequences, loops, events, conditionals, operators, and working with data – all identified as the computational thinking and programming concepts emphasized within Scratch. Using rigorous research methods, the goal of this study was to observe, describe, and compare how participants learn in two different learning contexts – Communal Learning and Individual Learning. Communal Learning is born out of culturally relevant pedagogy, which honors a student's culture and aligns instruction with it. The next several sections share more details about culturally relevant pedagogy, the origins of Communal Learning, and describes a trajectory of research studies which have observed improved performance in the academic performance of African American students when Communal Learning environments are used.

The Call for Improved Education in the United States

The threat to U.S. security and competitiveness regarding science and engineering described in chapter one is not the first of its kind. The launch of Sputnik 1, the first artificial earth satellite launched into the earth's orbit by the Soviet Union in 1957, also caused a period of public fear and anxiety known as the Sputnik Crisis (Brown, Kloser, & Henderson, 2010; Silva, Moses, Rivers, & Johnson, 1990). Not only did the Sputnik Crisis usher in country-wide anxiety about the nation's security, it also pushed the U.S. political and scientific communities to heighten emphasis on research and development in science and engineering productivity and improved education (Silva, et. al., 1990). The result of which were walls built between the general population and the scientific elite (Brown et al., 2010; Silva et al., 1990). The nation's poor people and minority children suffered the most. This was cause for alarm for members of these communities and further exacerbated the already unequal educational conditions and outcomes had by White mainstream American children and African American poor children.

The Academic Achievement Gap

African-Americans are indeed capable of learning and achieving academic success. There are a myriad of African American student honor roll members, college graduates, and successful doctors, lawyers, scientists, and entrepreneurs as a result. It is important to acknowledge, in spite of indicators of the perilous state of education in the African American and other underserved communities, that there are individual African American "successes" abound (King, 2005). However, there are some African American students who do struggle in school and find it difficult to get the support they need to accomplish high academic standing. Some education researchers suggest quite a few reasons for this struggle and difficulty, including low teacher expectation, lack of quality resources and qualified teachers, low levels of motivation and low socioeconomic status, peer-pressure, and poverty (Gregory, Skiba, & Noguera, 2010; King, 2005; Lee & Bowen 2006; Reardon, 2011). The combined result of these challenges and psychological barriers undoubtedly contribute to the actual low academic performance in school by some African American students along several other discouraging statistical outcomes. As a result, the [Black-White] "academic achievement gap" is a phrase and a phenomenon in the United States that describes the yearly statistical gap in scores on standardized academic achievement tests between the low scores obtained by some African American/Black students and the higher scores of their Caucasian-American (White) counterparts (Gregory, Skiba, Noguera, 2010; King, 2005; Lee & Bowen 2006; Reardon, 2011). The origins of the academic achievement gap can be traced back to 1969, when under guidance from the Department of Education and the National Center for Educational Statistics (NCES) within the Institute of Education Sciences (IES), the United States began the National Assessment and Educational Progress (NAEP) and its Nation's Report Card. NAEP collects and reports academic achievement at the national level, and for certain assessments, at the state and district levels and assessments have been conducted periodically in reading, mathematics, science, writing, U.S. history, civics, geography, and other subjects. The NEAP is the largest nationally representative and continuing assessment to inform the public about the academic achievement of elementary and secondary students (NCES, 2013).

The Call for Improved Education for African-Americans in the United States

Throughout the decades since the Sputnik Crisis, the political, scientific, and academic communities have called for an update in science, engineering, and mathematics education that appeals to and provides a pathway for success for all Americans (Brown et al., 2010; Silva et al., 1990). Coupled with the African American Civil Rights Movement (1954-1968), this focus on improved education in science and engineering also prompted scholarly investigation into the teaching and learning of African American and Latino American students, many of which were from families with low socioeconomic status (Brown et al., 2010; Silva et al., 1990). This focus uncovered the need for aligning curriculum and pedagogy with the cultural norms and values of these students. This became the birth of culturally responsive teaching and culturally relevant pedagogy.

Learning styles and culture of African American children. Research on learning styles and their relationship to culture have also been conducted for several decades. The literature defines learning styles as biological and developmental characteristics and preferences that affect how students learn (Hale, 1982; Serpell, 1997; Center for Research on the Education of Students Placed At Risk, 2000), similar to Boykin's cultural ethos described above. These preferences can help with classroom instruction and assessments, along with the design of classroom settings, and responses to a learner's individual need for quiet or sound, bright or soft light, warm or cold temperatures, seating arrangements, mobility, and/or grouping preferences (Dunn, Dunn, & Price, 1989).

In the years since the Sputnik Crisis, researchers have shared enormous amounts of data revealing that one's cultural patterns influence the way information is perceived, organized, processed, and used, resulting in what are called learning or cognitive styles (Adams, 1995; Barba, 1993; Boykin, 1977; Brown, Kloser, & Henderson, 2010; Edwards, 2010; Gay, 2002; Hale, 1982; Ladson-Billings, 1995; Lee, 1993, 1997; Pinkard, 1999; Silva, Moses, Rivers, & Johnson, 1990; Wilson, 1978; Wilson, 1992). From this, it is concluded that in order to maximize the learning potential of any learner, whether in or out of school, the learning environment and method of instruction *should match or be consistent with the cultural experiences of that learner*. This is a strategy called cultural scaffolding, cultural relevance, and/or culturally responsive teaching (CRT).

Valuing learning styles and culture in teaching and learning. The idea of valuing the knowledge and experience that youth bring with them to school was also emphasized in 1966 by John Dewey, one of the most significant educational thinkers of the 20th century. Dewey was a psychologist, philosopher, and educator whose social activism and writing fueled educational, philosophical, and social reform.

> "What we need is something which will enable us to interpret, to appraise, the elements in the child's present puttings forth and fallings away, his exhibitions of power and weakness, in the light of some larger growth-process in which they have their place." (Dewey, 1902, p.14)

Ten years later, Boykin (1977) frames what Dewey called a ‘child’s puttings forth and fallings away, his exhibitions of power and weakness’ as a child’s culture. More specifically, Boykin (1977) suggests nine afro-cultural ethos (cultural characteristics) that are distinctive to African-Americans.

The Black (Afro) cultural ethos. Boykin (1977) posits that these are characteristics naturally present and valued by African-Americans, their families, and their communities and are a vital part of their every day experiences, culture and thus, their values:

1. **Spirituality** – intuition, supreme force;
2. **Harmony** – versatility and wholeness
3. **Movement** – rhythm of everyday life
4. **Verve** – intense stimulation, action, colorfulness
5. **Affect** – premium on feelings, expression
6. **Communalism** – social orientation, group duty, identity, sharing
7. **Expressive individualism** – distinct, genuine, personal
8. **Orality** – oral and aural modes of communication

9. **Social time perspective** – time is marked by human interaction

Thus, culturally responsiveness in teaching and learning for African American students specifically can connote instructional relevance to these innate characteristics, every day experiences, and commonly held values. This seminal uncovering provided a foundation for decades of empirical research used to highlight effective methods of instruction for African American students, especially those that took advantage of their strengths: their cultures, learning styles, and modes of motivation and engagement. Boykin (1994a, 1994b) provide summaries of this work. Thus, integrating these afro-cultural characteristics into instructional design and delivery creates an innate connection between the instructional content and in African American students. In effect, pedagogically speaking, aligning instruction with students' culture is more than simply a source of motivation, it is natural and intrinsic way of being, knowing, and learning.

Matching or aligning classroom content with the culture of learners within the classroom has become a mantra for many educators and scholars concerned with the learning and academic achievement of culturally and linguistically diverse students (Adams, 1995; Barba, 1993; Brown, Kloser, & Henderson, 2010; Gay, 2002; Ladson-Billings, 1995; Lee, 1997; Leonard et. al, 2005). For instance, using the nine Black Cultural Ethos as a framework for describing culturally relevant and culturally responsive research, the *'Affect,'* '*Expressive Individualism,'* and '*Orality*' ethos appear in the works of reading researchers who have consistently found that African American and Caucasian American children differ in storytelling styles, knowledge of print conventions, oral language, and question asking style (Boykin, 1977; Edwards, 2010; Gay, 2002; Hale, 1982; Lee, 1993; Lee, 1997; Lee, 2003; Pinkard, 1999). Applying these same cultural ethos, '*Affect*,' '*Expressive Individualism,'* and '*Orality*' to culturally-responsive strategies to reading, science, and math instruction proves beneficial for students (Reis & Kay, 2007; Silva, et. al., 1990; Tharp, 1989). One specific investigation and application of the use of the '*Affect*,' '*Expressive Individualism'* and '*Orality,'* is apparent in Brown (2013) regarding African American science learners. This research points at the use language by African American students as an element of identity formation and explains the dichotomy between the 'cultureless' language of science and science classrooms and the expressed language of African American students. This dichotomy heavily impacts learning. For example, students may shy away from using science vocabulary as it may not fit or match the language used in forming their identity (i.e. "sounding Black"). Whether one agrees with this student perspective or not, a teacher knowledgeable about these identity sentiments can build bridges between the two and increase learning and engagement (Brown, et. al., 2010).

Additionally, the mathematics education system in the U.S., with its history of less than optimal performance by all students, not just African-American, has a prominent example of using culturally responsive teaching (Brown, Kloser, & Henderson, 2010). As a result of African American and Latino American students enrolling in advanced mathematics courses at rates significantly lower than their White counterparts, especially during and after the Sputnik Crisis, *The Algebra Project* was founded in 1982 by Dr. Robert P. Moses, a Harvard-educated Civil Rights Leader (The

Algebra Project). Using the cultural ethos of '*Verve*', '*Movement*,' '*Orality,*' '*Expressive Individualism*,' and '*Communalism*', *The Algebra Project* implements a curriculum where students are first introduced to mathematic concepts in the physical world around them, use their own language to describe the witnessed phenomenon, create symbols and use them to represent this phenomenon with their classmates, and are the introduced to mathematic concepts using their representations (Silva, et. al., 1990). With its success and teacher professional development, *The Algebra Project* reaches more than 200 middle schools across the country (The Algebra Project).

Valuing Culture and Learning Styles in Assessment in Science Education. The examples above bring attention to the need and benefit of incorporating culturally-responsive teaching strategies to a variety of academic topics, including science. Walls (2012) describes the benefit of assessing nature of science (NOS) views of young African American 3rd graders. While attempting to uncover the NOS views of its participants, students were assessed using multiple instruments that align with the '*Verve,*' '*Movement*,' '*Affect*,' *Expressive Individualism*,' and '*Orality*,' ethos presented by Boykin (1977) above. These instruments included an open-ended questionnaire, audio recorded semi-structured interviews, along with drawing and viewing images of scientists. According to Walls (2012), this combination of instruments appealed to the sociohistorical nature of science education as well as the underrepresentation of research for and with African American youth. This study contributes to NOS research in that it elicits the participants views as a result of tapping into their naïve concepts, emotions, and beliefs about science and themselves as scientists. This is in stark contrast to the traditional NOS research where participants are evaluated based on what extent their views contrast those of traditional scientists (Walls, 2012).

As a result, astounding insight resulted. Rich descriptions of specific views around what science is, how it is done (understanding natural science, experimentation), and what it is used for (invention and discovery) were elicited along with the general perception of who a scientist is and what he/she looks like (White & African-American, male, old, lab coat, glasses, intelligent, studious, happy), where they learn science (science textbooks and non-school locations), and what they think about it. This assessment also shed light on each participant's connection to science along with his/her confidence in being and doing science in the future. These and other findings contribute greatly to the creation of science education curricula along with what is known about the NOS views from a diverse (age and race) group of study participants. The study presented here seeks to accomplish similar contributions to CS Ed research.

The above examples specifically emphasize the alignment of innate cultural characteristics, learning styles, and the benefits their applications have on teaching, learning, engagement, and assessment. The next section shares fundamental requirements of and design principles for applying culturally-responsive.

Narrowing the Focus: Communalism and Communal Learning Research

Treisman (1985) and Fullilove and Treisman (1990) illustrate how African American college freshman students studying Calculus at UCLA Berkeley had significantly more pass rates they when studied together. Similar in methodology to the study presented here (i.e. ethnographic methods, audio and video recordings, observations, etc.), these studies uncovered and compared the study habits of African American calculus students to those of Asian-American calculus students. At a time when the gap in academic performance in science, engineering, and mathematics between ethnic groups was similarly described as that of the academic performance in computer science, these studies provide valuable insight into the conceptual frameworks that can be used to teach CT and programming skills to African American students. The hypothesis here follows that of Treisman (1985) in that when African American students spend more time studying together, they can help uncover errors, share knowledge, and perform better overall.

Communalism

Of the nine cultural ethos presented by Boykin (1977) above, this study specifically focus on the ethos of '*Communalism*' and applies it to the STEM subjects of computational thinking and programming. Operationalized, research on *Communalism (*social orientation, group duty, identity, sharing) in the classroom is called 'Communal Learning' and is often implemented by placing students in groups of two (pairs) or three, and their learning gains, efficacy, and engagement are usually compared to students receiving the same topics of instruction but who work individually (Hurley, Boykin, & Allen, 2005). Several studies have been conducted over the past two decades to explore the impact of Communal Learning on African American students from kindergarten through high school (Boykin, Coleman, Lilja, & Tyler, 2004; Boykin, Lilja, & Tyler, 2004; Burrell, 2012; Coleman, 2001; Dill & Boykin, 2000; Hurley, Boykin, & Allen, 2005). Topics of communal learning included text recall (Dill & Boykin, 2000), math estimation (Hurley, Boykin, & Allen, 2005), geography (Boykin, Lilja, & Tyler, 2004), and self and group efficacy in math estimation (Burrell, 2012). Many of these studies used scripted communal and individual learning prompts read to students before each unit of instruction. The communal prompt stressed the importance of working together as a group and sharing resources, while the individual prompt emphasized doing ones best while working alone.

Regarding study results, Watkins (2002) found that low-income African American kindergarteners from low-income backgrounds have more frequent displays of communal tendencies in their classroom environments than individual practices. Additionally, Coleman (1996) found that African American elementary students who were communally engaged in a creative learning task had more original and thoughtful responses than those students who performed the task individually. The performance of the communal learning group excelled the performance of the individual learning group in all of them. All studies followed a quantitative research design and data analysis involved analysis of variance (ANOVA). One study used analysis of covariance (Hurley, et. al, 2005). None of these studies found any significant gender difference. While no significant main or interaction effects have been found for the gender variable in previous studies examining the influence of communal and individualistic learning condition on academic performance, some grade-level effects did emerge, see Boykin and Bailey, (2000) for a review of the studies. Ultimately, taken together, these studies revealed that learning contexts that include familiar cultural themes are also more likely to sustain and enhance students' motivation to engage in required tasks than contexts characterized by unfamiliar themes.

Moreover, these studies inform the selection and use of a scale designed to measure students' actual learning context preference. The Learning Context Questionnaire (Johnson & Norem-Hebeisen, 1979) is a 22-item sentence-structure gender-neutral measure of cooperative, individualistic, and competitive orientation. Here and in the studies described above, it is used to measure a student's learning orientation preference. The competitive items are not used as this construct is outside the scope of what is being studied. Therefore, what is used is the LCQ-m, is a 14-item scale (Appendix F), where the m stands for modified. These 14 items are sentences with require a 4-point likert scale response ranging from 1 "Not at all like me" to 4 "Very much like me." to completely false. Examples sentences include: "I do better when I work alone" (individualistic orientation) and "It's a good idea for students to help each other learn" (cooperative orientation). Although students were assigned to either the communal learning or individual learning group in the studies described above, they each had their own authentic preference. This preference may or may not have matched their group assignment. The LCQ-m makes this authentic preference known.

Communal Learning and Pair Programming

The implications of the previous research on culturally responsive teaching, culturally relevant pedagogy, and communal learning environments led the researcher to a hypothesis that African American middle school students could learn computational thinking and programming skills using Scratch more effectively if done together, in pairs.. Pair programming, first used by Frank Brooks – author of the Mythical Man Month, while he was in graduate school between 1953-1956 (Brooks, 1975), is at the root of a collaborative software development approach called 'extreme programming,' intended to improve quality and responsiveness to customer needs. Pair programming requires that teams of two programmers work simultaneously at the same computer on the same design, algorithm, code, or test (McDowell, Werner, Bullock, & Fernald, 2002; Nosek, 1998; Williams and Kessler, 2000). One programmer is the *driver* and he/she types or illustrates and writes design ideas. The other is the *navigator* who actively choses best methods and approaches while observing the work of the driver looking for tactical or strategic defects. When used in industry, teams report a variety of benefits: improved product quality, fewer bugs, clearer code, improved knowledge sharing, motivation regarding coding, increased team morale, and a host of economic and other benefits (Denner, Werner, Campe, & Ortiz, 2012; Hanks, McDowell, Draper, & Krnjajic, 2004; McDowell, Hanks, & Werner, 2003; McDowell, Werner, Bullock, & Fernald, 2002; Sfetsos, Adamidis, Angelis, Stamelos, & Deligiannis, 2013).

As a result of the promise pair programming delivers, the use of pair programming techniques have been explored in various academic levels of computer programming courses showing great promise. When specifically implemented with college students, pair programming increased information technology fluency (adapting one's technology skills as technology changes) in middle school girls, (Campe, Werner, & Denner, 2005; Werner et al., 2005), confidence and satisfaction in the resulting program and the experience of programming (McDowell, Werner, Bullock, & Fernald, 2006), performance (McDowell et al., 2002), as well as socio-cultural differences when comparing two ethnic groups – Mexican and European (Ruvalcaba, Werner, and Campe, 2012). Thus, the notion of *pair programming* was introduced as a method of engaging young programmers.

Black Academic Identity

The theory of Black Academic Identity (BAI) for African American students comes from joining elements of racial identity with elements of academic identity, resulting in a connection between or overlap of being a Black student and doing well academically (Anderson & Freeman, 2010). One's Black Academic Identity can manifest in many ways. These are: 1) *Black Academic Identity*, where one's academic achievement is integrated with one's racial identity by thinking one's intellect is as a result of one's identity or that high academic achievement is crucial to being a successful Black student, thereby aligning one's behavior as such, 2) *Black Model Phenomenon,* where one is motivated to achieve and be successful to satisfy a desire to be a positive role model for other members of their race, 3) the *Proof of Black Ability*, where one reaches high academic achievement to dispel stereotypes, prove that African American students are intelligent, and can succeed, and 4) *Black Cultural Appreciation*, where an individual emphasizes the importance of knowing and appreciating their African American heritage and connects their achievement to those that came before them (Anderson & Freeman, 2010). The first three manifestations described above, *Black Academic Identify*, *Black Model Phenomenon*, and *Proof of*

Black Ability, are pertinent to and measured in this study. The theory and expected connection here is that the more a learning context is innately aligned with African American student culture, the more African American students will psychologically connect high achievement with their race and be highly motivated to achieve and align their behavior as such.

Summary of Culturally Relevant Pedagogy, Communal Learning, and Black Academic Identity

The U.S. concluded a threat to its security and competitiveness when the Soviet Union launched Sputnik 1 into the earth's orbit in 1957. This caused a period of fear and anxiety called the Sputnik Crisis and ushered in a heightened emphasis on research and development in science and engineering productivity and education. This unfortunately caused a divide between US citizens with the general population on one side and the scientific elite on the other. The nation's poor people and children of color suffered the most, as this Sputnik Crisis worsened the already existing unequal education conditions and outcomes between African American children and their White mainstream American counterparts. These unequal outcomes known as the academic achievement gap and was often explained by suggesting that African American students had an inherent deficit in cognition. Refuting these theoretical deficit models, African American education researchers called for pedagogy and classroom environments which honored the culture and learning styles of African American students and researched ways in which to value and align them. The result of this research is now called *culturally relevant pedagogy* and has a long history of empirical investigations resulting in improved and enhanced academic performance when implemented in classrooms educating African American students, in a variety of topics and at all levels of education. With culturally relevant pedagogy, success is rooted in the alignment of one's culture making it inherent, natural, and intrinsic as opposed to only a form of extrinsic motivation. One such cultural element is Communalism. Communalism is an element of the Black Cultural Ethos which places the social atmosphere of home and community culture within the classroom, emphasizing group duty and group identity, along with sharing. Classroom environments which align with Communalism are called Communal Learning environments. The success of Communal Learning environments mirrors Vygotsky's Social Development theory which posits that humans learn primarily by and through interacting with those around them. As such, with particular interest to the study presented here, Communal Learning aligns with the notion of pair programming, where two programmers work together on the same computer as they design, code (i.e. program), and test. The study presented here used culturally relevant pedagogy, specifically Communal Learning, by way of pair programming to teach African American elementary and middle school students computational thinking and programming skills. It explores the impact learning such skills has on a participants' Learning Context Preference and Black Academic Identity. Taken together, these concepts form the study's research questions, which will be implemented using a mixed-methods research approach.

Research Questions

The research questions driving this study are:

RQ1. During a summer Scratch programming camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context?

RQ2: Is there a change in the learning context preference of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?

RQ3: Is there a change in the Black Academic Identity of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?

The expected outcomes were that the all African American participants will demonstrate a preference for communal learning, that those in the communal learning group will score higher than the individual learning group on the computational thinking and programming post-test overall, and that their Black Academic Identity scores will increase. The next chapter describes the research design, research site, study participants, Scratch programming camp procedures, and data collection and analysis methods used to answer these research questions.

Chapter Three: Methods

This chapter shares an over of this study, its research design, and the overall methodology used. These details include an brief overview the research design, details about the research setting and site, the participants, the instructional materials, and data collection instruments, study procedures, as well as data analysis methods.

Overview of Research Design

A mixed method design was employed to gain more insight from the combination of quantitative and qualitative research than either form by itself (Creswell, 2009; Redmann, Lambrecht & Stitt-Gohdes, 2000). The quantitative portion involved a quasi experiment between two learning group contexts: Communal and Individual, where participants were randomly assigned to one or the other, to determine which learning group performed better. This study also used quantitative methods to collect pre- and post-camp scores for Scratch Content Knowledge, Learning Context Preference, and Black Academic Identity to measure participants' learning context preference and Black Academic Identity before camp began and to determine to what extent participating in this programming camp experience changed these measurements after camp ended. These quantitative data were analyzed using descriptive and inferential statistics. During camp, qualitative data collection was implemented through various questionnaires, Scratch programming assignments, audio and video recordings, interviews, participant written responses, and observations by the researcher, who was a nonparticipant observer. These qualitative data collection methods were used to capture the specific learning experiences of the participants in each learning context and examining that data to gain an in depth understanding of what happened each day with each participant. Data analysis was completed using the *Critical Incident Technique* (CIT) and the Cognitive Assessment of Participants' Problem-Solving and Program Development Skills. The CIT approach is defined as a set of procedures for systematically identifying behaviors that contribute to the success of failure of individuals in a specific situations (Redmann, Lambrecht & Stitt-Gohdes, 2000). The Cognitive Assessment of Participants' Problem-Solving and Program Development Skill measures the processes and products of a participants' problem-solving and programming skills (Deek, Starr, Kimmel, & Rotter, 1999). Overall descriptions and comparisons were made regarding how young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context as a result of combining these analyses. Additionally, these data collection and analysis methods were used to identify the change in participants' learning context preference and Black Academic Identity.

As such, this study followed Maxwell and Loomis' (2003) description of a convergent parallel mixed-method research design. The purpose of a convergent research design is to collect different but complimentary data on the same topic (Maxwell & Loomis, 2003). The quantitative and qualitative data collection methods were implemented in a pre-, during-, and post-intervention manner and are described in subsequent sections of this chapter. Quantitative data and qualitative data were collected concurrently and analyzed separately, while both methods were weighted equally. Creswell (2009) describes this as a concurrent triangulation mixed-method design. The resulting analysis merged to form an overall description and comparison of how participants learned computational thinking and programming skills in both learning groups.

Research Site and Setting

The research site contained two small computer labs on a college campus, with 15 Apple iMac desktop computers available in each, all connected to the Internet. Two one-week summer Scratch programming camps lasting 5 week days, three hours per day each, were administered in these computer labs (Monday-Friday, 10am-2pm).

Study Participants

Students. The student participants for this study were young African American novice programmers in elementary and middle school (rising 4th - 8th grade girls and boys), ranging in ages 9-13. Participation solicitation spanned seven days, during which the researcher sent an email to several listservs specifically soliciting African American families with children who had never programmed before to participate in a free summer programming camp, lasting 5 week days, 3 hours per day. One hundred and forty-one families responded with children of all ages. Forty-eight students were selected according to the age and grade criteria. Participants resided in the local areas around the research site and transportation to and from the site was provided by their parent/guardian. Of the 48 students selected, 42 participated. All 6 selected students who did not participate were girls, and they did not participate for varying reasons (lack of transportation, family travel, unexpected family emergencies, etc.). Table 6 below illustrates participant details and learning group assignments. Overall, there were 17 girls and 25 boys, 20 elementary school students (4^{th} and 5^{th} grade) and 22 middle school students (6^{th}, 7^{th}, and 8^{th} grade) who participated. Twenty-two participants were paired and assigned to the Communal Learning group. These assignments deliberately included two boy pairings, two girl pairings, and mixed pairings of one boy and one girl. These assignments resulted in 4 pairs of two boys, 4 pairs of two girls, and 3 pairs of a boy and a girl, totally 11 boys and 11 girls. Twenty participants were assigned to the Individual Learning group, where 6 were girls and 14 were boys. The 6 girls who did not participate were all assigned to the Individual Group. There absence resulted in a low amount of girl participants.

Table 6

The Distribution of Participants for this Dissertation Study

Grade-Level	Week One - Communal Learning Group	Week Two – Individual Learning Group	Total
Elementary School Students (4th & 5th grade)	10 (4 boys & 6 girls) 1 boy team, 2 girl teams, 2 boy and girl team	11 (7 boys & 4 girls)	21 (11 boys & 10 girls)
Middle School Students (6th, 7th, & 8th grade)	12 (7 boys & 5 girls) 3 boy teams, 2 girl teams, 1 boy and girl team	9 (7 boys & 2 girls)	21 (14 boys & 7 girls)
Total	22	20	42

Communal learning group description. Participants in the communal learning group were divided into pairs of participants by grade and gender, where there were eight pairs with two boys, eight pairs with two girls, and six pairs with a boy and a girl; totally 11 girls and 11 boys. Participants in this group totaled 22, 10 in elementary school and 12 in middle school. A detailed review of these assignments can be found in the Table 6 above. This group was introduced to one another once pair assignments were announced. The researcher encouraged introductory activities to facilitate interaction and to make it easier for those just meeting each other to be more open to working with one another. Once pairs met, middle school participants relocated to their assigned computer lab. It did not take long for the researcher to realize that pair assignments were near perfect as members of most of the pairs had a great deal in common. As a result, participants enjoyed meeting one another. So much so, that they were often distracted and spent a great deal of time in conversations related to getting to know each other. Additionally, analysis of the audio recordings revealed many conversations that were not relevant to the material of the study. This occurred throughout both weeks of camp with both learning groups, but was most prevalent in the communal learning group, where the structure of the study encouraged more interaction (i.e. working in pairs). This behavior presented productivity challenges which are described later in Chapter Four – Results and Findings. One boy from the communal learning group could not participate due to an emergency after the first day of camp. His partner is considered to be in the individual group for data analysis since he worked alone for the entire week of camp dedicated to the communal learning group, while everyone else in this group worked in pairs.

Figure 5 below features a series of snapshots of participants in Communal Learning group (i.e. pairs) in both grade levels. These snapshots provide illustrations of fidelity regarding the implementation of study, where the Communal Learning group participants worked in pairs, shared resources and materials, and helped one another.

Individual learning group description. There were 20 participants in the Individual Learning Group and they were also introduced to one another on the first day of week two of camp. There were 11 elementary school students and nine middle school students in this group. Participant in this learning group did not initially interact with one another as much on the first day of camp, as compared to the amount of interaction of the pairs in the Communal Learning group on their first day. However, they did get to know one another during lunch and other breaks throughout their week of camp. Many of these participants ended up removing their audio recording gear intuitively because they were quiet and not making any sounds. Figure 6 below features a series of snapshots of participants in Individual Learning group in both grade levels.

Fidelity. Figures 5 and 6 above show snapshots of participants in the Communal and Individual Learning groups working through the Scratch Booklet, with evidence of participants sharing resources are present within the Communal Learning groups along with evidence showing participants working alone. These provide fidelity regarding the implementation of the study.

Figure 5. A series of Communal Learning group pairs working through the Scratch Booklet.

Figure 6. A series of Individual Learning group participants working through the Scratch Booklet.

Researcher. There was one researcher present during camp. The role of the researcher was to perform silent observations, ensure recording equipment functioned and remained positioned properly, and classroom setup. The researcher also read the learning prompts, described below, during appropriate times throughout both weeks of camps. No instruction was given by the researcher with the exception of the first lesson about computers. The purpose of this short lecture was to ensure that everyone understood the purpose of the camp, how to use the computers, and important vocabulary (e.g. what is software, what is Scratch). Silent observations included taking field notes. The researcher also served as disciplinarian.

Instructional Materials

Scratch booklet. Both weeks of camp used the same instructional material. This material included a printed booklet with 15 units, a personal notebook, and each student was asked to create a Scratch account. While this booklet is literally a 233-page textbook, into which additional lessons were added by the researcher, a pdf version similar to it and all its supplemental material was created by (Armoni & Ben-Ari, 2013) and is made available under a Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported License (Appendix K) and can be downloaded from their website at http://stwww.weizmann.ac.il/g-cs/Scratch/Scratch_en.html. Participants received all instruction from reading this booklet throughout camp, which contained Scratch lessons, computational thinking concepts, and 175 small programming activities, and is heretofore referred to as the Scratch Booklet. Table 7 below contains a description of each Unit in this Scratch Booklet, specifically highlighting the computational thinking and programming skills targeted (sequences, events, loops, conditionals, operators, and working with data). There were a total of 230 pages and 175 programming projects.

Table 7

Description of Units within Scratch *Booklet*

Unit & Pages	CT Skill	Unit Concept & Programming Activities	Description
One (13)		Introduction to Scratch (5)	Scratch Interface, Functions (i.e. Saving), Programming Vocabulary (i.e. Bug)
Two (17)		Motion Blocks (3)	Motion Blocks, Stage Coordinates, Animation, Programming Vocabulary (i.e. Initialization)
Three (8.5)	Sequences	Multiple Sprites (6)	Scripts, Using more than one Sprite, Sprite Costumes
Four (12)	Loops & Conditionals	Short Scripts, Long Runs (9)	REPEAT LOOPS, CONDITIONALS, Changing Backgrounds, Concurrency
Five (9)	Conditionals	Communicating Between Sprits (20)	Sending & Receiving Messages, Conditional Wait
Six (16.5)	Conditional Loops	'On the Dance Floor' (14)	REPEAT LOOPS continued, Conditional Loops, Dance Animation Repeated Run Again
Seven (9.5)	Events	Realistic Animation (25)	Changing Costumes, Sound, Nested Instructions, Interaction with User
Eight (16)	Working with Data	Remembering Things – Variables (35)	Creating, Initializing, Storing, Reading, & Changing the values of Variables, Adding Buttons
Nine (28.5)	Complex Conditionals	It Depends – Conditional Runs (14)	CONDITIONALS: If/Then, If/Then/Else, OPERATORS, Changing Sprites (brightness, color, hue, etc.)
Ten (29)	Operators	Numbers (18)	Stamps, Accumulators, Comparing Numbers, Strings, Joining Strings, Compound Conditions
Eleven	Working	Lists (9)	Creating Lists, Entering Content,

Unit & Pages	CT Skill	Unit Concept & Programming Activities	Description
(27)	with Data		Reading a List, Deleting a List, Sprite Dialogue; Remembering Complex Information
Twelve (21)	Sequences	Concurrent Runs (Projects reused)	Simultaneous execution of blocks
Thirteen (9)	Working with Data	Abstraction & New Blocks (Projects reused)	Abstraction, creating Scratch blocks
Fourteen (11)	All	Game Design (7)	Step-by-step guide towards creating a game
Fifteen (3)		Practices	Creating and refining solutions, fixing errors, Scratch CT concepts, documentation, etc.

During camp, each pair within the communal learning group was given one set of instructional materials (one instructional booklet, one computer). Each student in the individual learning group was given his/her own set of instructional materials. All students (communal or individual) worked through these Scratch Booklets at their own pace.

Table 8 below illustrates the number of pages addressing each CT skill both specifically and generally, as a concept.

Table 8

The total number of pages dedicated to each of the six CT skills in the Scratch *Booklet*

CT Skill	Pages Dedicated Specifically	Overall Total Pages
Sequences	29.5	197
Events	9.5	20.5
Loops	28	55
Conditionals	66	77
Operators	29	40
Working with Data	52	141

The Scratch programming environment. The online Scratch Programming environment was used throughout both weeks of this summer camp. Participants created and recorded their account names and passwords and used the same account throughout the entire camp, saving each program as instructed in the Scratch Booklet.

Research Constructs and Measured Variables

Using the theoretical framework presented above in Chapter One – Introduction and illustrated in Figure 1, the research constructs of interest in this study include learning group (communal versus individual), learning context preference, and Black Academic Identity (BAI), and Scratch computational thinking and programming content knowledge. These constructs are described in detail in Chapter Two – Literature Review. These constructs inform the independent variables and their factor levels and the dependent variables for this study. All variables were measured before camp and after camp (i.e. pre- and post-tests). The independent variables were used to implement a between-subjects comparison of participants' Scratch computational thinking and programming content knowledge scores between the two learning groups (communal and individual), while the dependent variables were used to implement a within-subjects comparison of participants' learning context preference and BAI before and after camp.

Data Collection Sources

All students received four pre-intervention data collection instruments **before** the intervention (summer camp) began. All instruments were administered online via Google Forms. Table 9 below provides an overview of each. These instruments included:

1. **Prior Computer Experience Questionnaire –** This questionnaire (Appendix D) obtained information about each participant's previous computer experience. This questionnaire is a combination of questions taken questionnaires with permission (Appendix C) from Grover (2014), Clark, Brandt, Hopkins, & Wilhelm (2009), and Clark & Sheridan (2010) also used for the same purpose. They each have been used in previous research studied with middle and high school students.
2. **Scratch Computational Thinking and Programming Content Knowledge Pre- and Post-Tests**. This instrument (Appendix E) includes 25 questions relating to the six areas of computational thinking attributed to the Scratch programming language, namely sequences, loops, events, conditionals, operators, and working with data. This questionnaire is used with permission (Appendix C) from a similar study conducted previously with middle and high school students (Grover 2014).

3. **The Learning Context Questionnaire – modified (LCQ-m)** is a 22-item sentence-structure gender-neutral measure of cooperative, individualistic, and competitive orientation (Johnson & Norem-Hebeisen, 1979). Here, it is used to measure a student's learning orientation preference. The competitive items are not used for this study, thus making the scale a 14-item scale (Appendix F). Each sentence requires a 4-point likert scale response ranging from 1 "Not at all like me" to 4 "Very much like me." to completely false. Examples sentences include: "I do better when I work alone" (individualistic orientation) and "It's a good idea for students to help each other learn" (cooperative orientation). This questionnaire has been used in this way in all of the previously described communal learning studies. The LCQ-m yielded alpha reliabilities of $\alpha = .88$ and $\alpha = 0.80$ for cooperative orientation and individualistic orientation respectively.

4. **The Black Academic Identity Scale** is a 10-item measure (Appendix G) that seeks to explain the intersection of racial and academic identity for African American students (Anderson & Freeman, 2010). Each item requires a 5-point likert scale response ranging from 1 "Not at all true" to 5 "Completely true." Example statements include: "I think of myself as a Black student, not just a student" (Black Academic Identity) and " I want to show others that Black students are smart" (Black Model Phenomenon). This scale is used with permission and has previously been used in research studies with middle school and undergraduate college students.

5. **Opinion Prompts** after each unit of instruction. Participants were instructed to respond to these prompts, from Cain & Woodward (2013), when he/she finished a unit. (Appendix H)

a. Think about what you have learned in this unit, and reflect on what you think were key learning points or incidents. Answer questions such as: What did you learn? What do you think was important? What did you find interesting? What have you learned that will be valuable for you in the future? Which activities helped you most? Has this changed the way you think about [computer programming]? Did you learn what you wanted/expected to learn? Did you make effective use of your time? How could you improve your approach to learning in the future?

6. **Notebooks** – Throughout the camp, students will be asked to record their ideas as well as varying the planned sequence of instructions (i.e. algorithms) for each programming activity. These will be analyzed for evidence of computational thinking behaviors and thought processes.

7. **Video recording** of the entire classroom. These video cameras were positioned to view and record the entire classroom for the entire duration of each day of camp for the communal learning group. These recordings document interaction between pairs to help describe how they work (i.e. who types, when, and how often). Snapshots from these video clips provide still images of the participant activities.

8. **Audio recording** of each student with individual microphones recording throughout each camp day for the duration of the camp week for both learning groups. These audio recordings will be transcribed and will be used to describe students' overall learning process, the interactions between the pairs, spoken thought processes, and questions asked.

9. **Scratch Projects** – The actual computational artifacts created with Scratch by each student or student pair (in the communal learning group only). These projects were assigned within the units of the Scratch Booklet.

10. **Interview -** These questions are designed to gather opinions regarding a participant's experience using Scratch as well as their experience in their respective learning group. The questions for this interview (Appendix I)were modified from Brennan & Resnick (2012).

Table 9

The Matrix of Pre- and Post-Intervention Data Collection Instruments

Instrument	Pre	During	Post	Analysis	Research Question
Prior Computer Experience Survey	●			Descriptive Statistics, Correlation Analysis with Pre- and Post-Tests	None. Used to describe prior computer experience.

Instrument	Pre	During	Post	Analysis	Research Question
Scratch Computational Thinking and Programing Content Knowledge	●		●	(Qualitative and Quantitative) Score descriptions and comparisons by group, grade-level, pair-type, and gender. Independent Samples t-test, between pre- and post-test scores by group.	**RQ1**: During a summer Scratch programming camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context?
Learning Context Questionnaire-modified	●		●	(Qualitative and Quantitative) Score descriptions and comparisons by learning group via thick description and paired samples t-test.	**RQ2**: Is there a change in the learning context preference of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?
Black Academic Identity Scale	●		●	(Quantitative) Score Comparison by learning group and grade-level via thick description and paired samples t-test.	**RQ3**: Is there a change in the Black Academic Identity of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?

Instrument	Pre	During	Post	Analysis	Research Question
Opinion Prompts		●		(Quantitative & Qualitative) Thematic Analysis & Frequency Count; Description and comparison by learning group, grade-level, and gender and pair-type & Cognitive Assessment of Problem-Solving and Program Development	**RQ1**: During a summer Scratch programming camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context?
Notebooks		●		(Qualitative) Description and comparison of computational practices by grade-level, school-level, gender, and learning group & Cognitive Assessment of Problem-Solving and Program Development	**RQ1**: During a summer Scratch programming camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context?

Instrument	Pre	During	Post	Analysis	Research Question
Scratch Programming Projects		●		(Qualitative and Quantitative) Cognitive Assessment of Problem-Solving and Program Development	**RQ1**: During a summer Scratch programming camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context?

Instrument	Pre	During	Post	Analysis	Research Question
Interviews			●	(Qualitative) Thematic Analysis, Description and comparison of programming concepts, practices, and perspectives by gender, grade-level, school-level, and learning group.	**RQ1**: During a summer Scratch programming camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context? **RQ3**: Is there a change in the Black Academic Identity of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?

Quasi Experimental Context of Study and Context Learning Prompts

In this study, where participants were placed in one of two learning groups (communal or individual), each learning group was read learning prompts (Dill & Boykin, 2000) in the morning at the beginning of each camp day and then again just after lunch before when camp activities resumed. These prompts were also read just before pre- and post-tests. These prompts remind students of the importance of his/her learning context as it relates to learning computational thinking and programming skills (e.g. responsibility to group or responsibility to self).

Communal learning prompt.

> *Welcome to Day # of camp. I hope you are engaged in learning how to program together.* Scratch *is a great tool and a lot of fun to use. If you have a partner, please continue to rely on one another, share camp resources as well as what you know. Be helpful, considerate, and give your best to your team. Your goal as a team is to finish the booklet. Therefore encourage each other, stay on task, and do your best. How well you do as a team depends on how much you each take part in the learning process. Please enjoy this learning process but don't enjoy it so much that you are off task. Remember, you have the freedom to learn and explore as much about computer programming and computational thinking as possible. Use this freedom to do your best together as a team. Does everyone understand? I will remain in the room in case you have questions. You may begin.*

This prompt emphasis the nature of communalism and communal learning, namely its social orientation, group duty, group identity, and sharing.

Individual learning prompt.

Welcome to Day # of camp. I hope you are engaged in learning how to program. Scratch *is a great tool and a lot of fun to use. For this computational thinking and programming lesson, you should work individually. Each of you will receive your own materials to use. You will be learning <topic for current lesson>. You are to work by yourselves and may not help or be helped by others. It is important to learn and work on this lesson by yourselves because your performance will be based on what you can do on your own. If you have any questions, quietly raise your hand and ask me. You will have <time frame in minutes> to study the material. There will be a short programming activity after the <time frame>, so it is important that you work hard to do your best so you will do well. Please enjoy this learning process but don't enjoy it so much that you are off task. Remember, you have the freedom to learn and explore as much about computer programming and computational thinking as possible. Use this freedom to do your best together as a team. Does everyone understand? I will remain in the room in case you have questions. You may begin.*

This prompt emphasis the nature of individual determination and independence. The exact opposite of the communal learning prompt. These prompts have been modified from those in Center for Research on Evaluation Standards and Student Testing (2004) to suit the nature and topic of this current study. Images reflecting observations of students following this behavior are provided in Chapter Four – Results & Findings.

Intervention (Camp) and Data Collection Procedure

This study began on a Monday in the middle of the summer and took place Monday through Friday, for two consecutive weeks in the middle of the summer. Camp began at 10am and ended at 2pm each day, with a 30 minute lunch break, and small 5-10 min breaks (with snacks) given when and as often as needed throughout the morning and afternoon sessions. The total time participants spent in learning was approximately 3 hours a day (for 5 days), for a total of 15 hours for the week. Week one hosted the Communal Learning group, while the Individual Learning group was hosted during week two. The Apple computers, all with 24" screens, in the computer lab were newly purchased, relatively fast, and were connected to the institution's dedicated high speed Internet network.

Participants received all pre-intervention data collection instruments on the first day and were asked to respond to each question on all instruments. The first day of camp then began with the researcher providing instructions on how to use the audio recorders along with a brief introduction lecture on technology, hardware, software, what it means to write a computer program, and introduced the name of the Scratch platform. Participants then began reading the booklet through the 15 units booklet, either with their communal partner or individually, according to learning group context. Students read as much of the 15-unit instructional booklet consecutively throughout the 5 days of camp and worked at their own pace. This instructional booklet focused on various concepts of computational concepts, computational practices, and computational perspective using Scratch. The researcher was only available to answer a minimum number of questions, for instance if a participant was confused about booklet instructions.

The original implementation plan for this study featured all participants in one large computer lab as the research site for both weeks of camp. Unfortunately, the Friday before this study began, the computers in the intended computer lab were deemed unusable and that weekend, the study was reassigned to another building. This new building did not have one large classroom available that would accommodate all participants. Instead, the study was assigned to two separate smaller computer labs, next door to one another. As a result, the researcher assigned one room for elementary school participants and the other for middle school participants for both weeks.

Once camp was underway, data collection sources **during** the intervention included:

Participants followed the following learning procedure for each unit.

1. Introduction - Read about concept
2. Follow task instructions to implement concept (document ideas, problem-solving steps)
3. Creation Activity using that concept
    a. Program
    b. Test
    c. Debug
    d. Document
4. Submit response to Opinion Prompt

The last activity on the last day of the intervention (camp), the following post-intervention data collection instruments were used:

1. Post Content Knowledge
2. LCQ-m
3. Black Academic Identity
4. One the last day of each week of camp, the same instruments used as pre-test measures will be used as post-test measures. See Table 6 above.
5. Scratch Experience Interview Questions.

Data Analysis Procedure

Data analysis of this convergent parallel mixed-method design entailed the use both qualitative and quantitative data analysis software packages, namely SPSS and NVivo, respectively, and at times, hand coding was also implemented. Once data collection was completed for both weeks of the camp, all data was cleaned and prepared for analysis. The qualitative data analysis methods occurred first and the results were to describe the experiences, processes, and outcomes of this study for participants in both the communal learning and individual learning groups. The quantitative data analysis methods were preformed next and the results used to determine the statistical performance levels of both learning groups by grade-level and gender. These analysis methods are described next.

Qualitative data analysis. The qualitative data collected for this study were analyzed using a variety of tools and techniques. These included: 1) the Critical Incident Technique (CIT) was used to analyze data collected from the interviews, video and audio recordings, and 2) elements of a Cognitive Assessment of participants' problem-solving and program development skills were used to analyze participants' responses to the opinion prompts, notebooks, and Scratch project programming outcomes.

The critical incident technique. CIT is defined as a set of procedures for systematically identifying behaviors that contribute to the success of failure of individuals in a specific situations (Redmann, Lambrecht & Stitt-Gohdes, 2000). The specific situations of interest here were the two learning contexts to which study participants were randomly assigned, namely, the Communal Learning Group, where participants were assigned to work in pairs, and the Individual Learning Group, where participants worked alone.

Redmann, Lambrecht and Stitt-Gohdes (2000) highlights the fours steps involved in implementing this technique. They are:

1. Develop plans and specifications for collecting factual data about the situation,
2. Collect episodes or critical incidents
3. Identify themes and sort them into categories

Interpret and report.

The critical incidents/episodes of interest entailed participant behavior and self-reported strategies related to learning Scratch by reading the Scratch Booklet and completing the small programming activities (175 total) in each unit, the contents of participants' notebooks, and participant responses to opinion prompts at the end of each Scratch Booklet Unit. Data collection was done via in-person observations by the researcher, end of camp interviews, as well as audio and video recordings obtained during both weeks of camp. An audio recorder and a microphone were attached to each participant and a video recorder was positioned at the rear of each classroom with as much of the classroom in the viewfinder as possible. Relevant episodes were reviewed for initial patterns using data coding methods described by Corbin and Strauss (2008). "Open/Initial" and

"Descriptive" coding methods were conducted by hand to document basic topics data which emerged for each information source. One word or short phrases were used to describe what occurred in each critical incident (Saldaña, 2009). These were followed by "process" coding, which was used to make note of simple observable actions (Saldana, 2009). These word, short phrases, and actions were then combined in an attempt to describe one experience. Then, "axial coding" was used to refine specific points of comparison and were arranged to make descriptions and comparisons align with the independent variables involved in quantitative data analysis. These themes were then given to another researcher to review and a face-to-face meeting to develop consensus. These themes are discussed in Chapter Four – Results & Findings.

Cognitive assessment of participants' problem-solving and programming skills. This assessment is used to provide student process (e.g. skills), product (e.g. code and documentation), and self-reporting feedback regarding the use of the Dual Common Model for Problem-Solving and Program Development taught in beginner computer programming classes at the New Jersey Institute of Technology (Deek, Starr, Kimmel, & Rotter, 1999). The Dual Common Model includes 6 stages of problem-solving and program development. They are: 1) problem formulation, 2) solution planning, 3) solution design, 4) solution translation, 5) solution testing, and 6) solution delivery (e.g. quality and correctness). Successful completion of each stage consists of several tasks (Deek, Starr, Kimmel, & Rotter, 1999). Four-point and rubrics are used to score each stage, with a total of 20 possible points available. These rubrics can be found in

Appendix J (Deek, Starr, Kimmel, & Rotter, 1999). The opinion-prompts (Appendix H) were used to collect and analyze participant self-reporting feedback.

Scratch project analysis. The total number and types of sprites (i.e. characters) and puzzle pieces (i.e. blocks of code) used were counted. This frequency determines the amount of creativity and level of complexity present in each program. Comparisons between learning groups were made based on these attributes.

Quantitative data analysis. Scoring of the Computational Thinking and Programming Content Knowledge Questionnaire (Pre- & Post-Tests) was performed first. It featured a mixture of open ended and multiple-choice questions (Appendix E). Scoring of this pre- and post-tests involved tallying total points for fully correct, partially correct, and incorrect responses. A fully correct responds earned 2 points, while a partially correct response earned 1 point. Incorrect responses earned the participant 0 points.

Scores and responses for all dependent variable data collection sources (Scratch content knowledge, learning context preference, and Black Academic Identity) were entered into SPSS for descriptive and inferential statistic analysis. When preparing and entering the quantitative data in SPSS, all reverse coding was done on data that required it. Descriptive statistics methods were used to reveal and describe raw scores for Scratch content knowledge, learning context preference, and Black Academic Identity. These were followed by inferential methods, namely independent and paired samples t-tests, to determine mean differences between the pre- and post-test mean scores. Score reliability calculations were then performed, such as Cronbach's Alpha and Cohen's Kappa for interrater-reliability.

Fidelity, Reliability, and Validity

Several strategies were used to ensure fidelity and reliability and to avoid possible validity threats. These strategies include evaluation of study implementation fidelity, triangulation, using measurements and questionnaires that have been used previously in other studies by other researchers, several reliability statistics, as well as sharing a statement on researcher bias. These strategies are described below.

Fidelity. Fidelity is in indicator of how a research study was implemented as directed. To ensure the fidelity of this study's implementation during each week of camp, the researcher followed a script of daily activities and read instructions to participants twice each day, ensuring that they understood. After each week of camp, evidence of study fidelity was indicated with the use of video screenshots to provide visual evidence of what occurred. These images aligned with the intended research design and activities for each learning group context. Additionally, end of camp participant interviews provided indicators of fidelity as participants were asked to describe what they liked or disliked about the activities in their learning group. Participant in both learning groups described activities that aligned with the intended design and activities.

Triangulation. Triangulation is the use of several different methods and types of data sources to counterbalance and check one another in order to mitigate the potential bias of the result of one specific method of data collection instrument. Triangulation helps to support the formulation of a single conclusion along with the opportunity to analyze collected data from different perspectives (Kaplan & Maxwell, 2005). With these objectives in mind, the design of this study included 10 data collection sources to understand how participants in both groups learn and used Scratch computational thinking and programming skills and how this learning experience impacted their learning context preference and Black Academic Identity.

Data collection instruments. To help ensure validity of data collected in this study, all the data collection instruments have been used in dissertation and other research studies prior to this current study. The authors of these instruments have reported high reliability Cronbach's Alpha Reliability scores and have provided permission for the use of these instruments in this study.

Cronbach's alpha reliability. Additionally, the Cronbach's internal alpha (Cron's Alpha) reliability statistic was run and used to analyze the reliability of the resulting instances of the four pre- and post-Scratch camp scales and questionnaires in this study: 1) Pre-Intervention Computer Experience Questionnaire, 2) Scratch Content Knowledge (pre- and post-) Questionnaire, 3) the Learning Context Preference Scale, and 4) the Black Academic Identity scale.

Interrater reliability. Two third-party coders were assigned to score the open-ended questions on the Scratch Content Knowledge Questionnaire. Both were given explanations of the correct answers and with them, scored three participant questionnaires, each. This Cohen's Kappa, the statistical test for interrater reliability was calculated and compared to determine level of scoring agreement. Once calculated, each rater rated another three questionnaires and compared level of agreement until the level of agreement was high. The Cohen's Kappa statistical test (Cohen, 1988) was completed again to calculate the final level of agreement. This was done to avoid scoring bias.

Researcher bias. The researcher designed this study informed by her passion and 16+ year experience of teaching computer science and engineering concepts to PreK-12th grade students throughout several states. Most of these students have been African American. A desire to prepare students to become successful computer scientist combined with her membership in the CSEd Research Community, the researcher has searched for several practical best practices relating to instructional strategies for African American students (male and female) over the past several years and quickly noticed a gap. As a result, the researcher did not serve as instructor. Instead of the researcher's instruction influencing what, how, and how much participants learned, learning was allowed to occur according to the self-regulation practices of each participant. This allowed each participant to progress and his/her own pace and according to his/her level of understanding. In this way, the effect of being in a communal learning or individual learning group was authentic.

Regarding the combination of constructs and the design of this study, the researcher personally knows the cited researchers who conduct research on culturally relevant pedagogy, communal learning, and Black Academic Identity. They are or have been personal friends as well as professional collaborators. This made it an easy to combine these constructs and design this study after anecdotally noticing the potential impact of culturally relevant pedagogy on students and their level of performance for many years prior.

Author's use of Scratch***.*** The previous experience that creates the researcher bias described above has also positively influenced the design of this study because as a result, this multi-year experience has allowed for the obvious identification and selection of Scratch as the proven programming environment of choice for beginning programmers.

Summary of Methods

The overall aim of this research study was twofold. The first was to extend the body of education research investigating the impact of culturally informed pedagogy, or more specifically communal learning, to the teaching and learning of computational thinking and programming concepts in an informal learning environment (i.e. summer camp), and the second was to provide a rich description of how young novice African American programmers learn and apply these skills. The use of a mixed-methods design helped determine which learning context produced the best cognitive performance in these areas. The data collection and analysis methods used were both qualitative and quantitative. Qualitative data analysis techniques included the Critical Incidence Technique as well as a Cognitive Assessment of participants' problem-solving and computer program development skills. Quantitative analysis methods included descriptive and inferential statistics. Issues of fidelity, reliability, and validity were addressed throughout. As such, the study gleaned descriptions of how learning occurred, which specific topics were challenging, which were not, along with accounts of how this learning environment impacted students based on gender, age/grade level, and who they were paired with (i.e. two girls, two boys, a boy and a girl). The next chapter shares the outcomes of this dissertation study.

Chapter Four: Results and Findings

This study, implemented during two one-week summer camp, lasting five week days, 3 hours per day each, sessions designed to teach the Scratch programming language, used an exploratory convergent parallel mixed-method research approach towards the empirical investigation of how young African American elementary and middle school novice programmers learn computational thinking and programming skills. It included a quasi experimental design where 42 participants were conveniently sampled. Twenty-two participants were assigned to and worked in a culturally-responsive learning context (i.e. the Communal Learning group) and 20 participants were assigned to and worked in an individual learning context (i.e. the Individual Learning group). Each group participated in the programming camp for one week (i.e. 5 week days, 3 hours a day of learning time). Computational thinking and programming skills specifically related to the visual programming language Scratch were taught and measured via pre- and post- test scores. More specifically, these skills involved programming using: sequences, loops, events, conditionals, operators, and working with data – all of which are computational thinking and programming skills identified by the creators of Scratch (Brennan & Resnick, 2012). Additionally, this study explored whether or not and to what extent this programming camp experience changed participants' learning context preference and Black Academic Identity.

The following three research questions guided this study:

RQ1. During a summer Scratch programming camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context?

RQ2: Is there a change in the learning context preference of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?

RQ3: Is there a change in the Black Academic Identity of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?

Brief descriptions of the information sources and data analysis used to answer these questions follow.

Pre-Test Camp Data Sources

Before camp began and before pairs and pair-types were shared with participants, participants in both learning groups responded to four scales and one Computational Thinking & Scratch Content Knowledge (pre-) Test. These scales included: the Cooperative Learning Context Preference scale, the Individualistic Learning Context Preference scale, the Black Academic Identity scale, and the Black Model Phenomenon scale. This provided pre-camp scores and measurements for all dependent variables (CT knowledge, cooperative and individualistic learning context preference, black academic identity) in preparation for answering RQ1, RQ2 & RQ3.

During-Camp Data Sources

During camp for both learning groups, participants were individually audio recorded, wearing their own audio recorder, and a video camera was placed at the back of each classroom recording as much of the classroom as possible. Participants were asked to record their thoughts and any other information or drawings related to learning Scratch and completing their Scratch projects in their notebooks, and following the completion of each unit in the Scratch workbook, participants were asked to submit responses to Opinion Prompts about their experience of each unit. These qualitative information sources were used to obtain a rich portrayal of participants experiences in each learning to describe and compare the learning experiences of participants in each learning group, contributing to the answer of RQ1.

Post-Test Camp Data Sources

After having gone through camp, participants in each learning group submitted responses to the same scales and Computational Thinking & Scratch Content Knowledge (post-) Test. These data represented the change in all dependent variables (CT knowledge, cooperative and individualistic learning context preference, black academic identity, and black model phenomenon) in preparation for answering RQ1, RQ2, & RQ3.

Findings

This chapter shares descriptions and comparisons of those novice programmers in the Communal Learning group with those in the Individual Learning group using the primary independent variables: 1) learning group assignment (communal vs. individual); 2) grade level; 3) gender; and 4) pair-type (two girls, two boys, and a girl and a boy); and the study's dependent variables in order of the research questions featured above: 1) computational thinking and programming skills – overall, sequences, events, loops, conditionals, operators, and working with data; 2) learning context preference (scale); and 3) Black Academic Identity. The chapter begins with descriptions of research setting and learning group dynamics and shares descriptive statistics of the study's participants based on the Pre-Camp Computer Experience Questionnaire (Appendix D) asking about their academic and prior computer experiences along with their career goals.

Qualitative data analyses are then shared to help answer RQ1. These include rich descriptions and comparisons of communal and individual learning experiences through lenses of participant behavior, their experiences and thoughts about the Scratch Booklet, their strategies used for booklet completion, problem-solving and programming skills assessments using the contents of their notebooks, resulting Scratch projects, and responses to end-of-unit opinion prompts, audio and video recordings, and responses to end-of-camp interview questions. Subsequently, quantitative data results to help answer RQ1, RQ2, and RQ3 are shared in the form of descriptive statistics followed by inferential statistics. All three research questions are then explicitly answered and the chapter ends with a summary of the findings.

Resulting Sample Description

Sample description based on prior experience questionnaire. Of the 42 participants in this study (25 were boys and 17 were girls), most of the participants

(76.2%) were African-American/Black while, none were Caucasian-American/White. Race and ethnicity data was missing for 3 participants. The remaining 12% considered themselves to be Latino-American, African, Native-American, and Bi-racial. Detailed race/ethnicity data are shown in Table 10 below. There were 21 elementary school student (eleven 4th graders, ten 5th graders) and 21 middle school students (ten 6th graders, seven 7th graders, and five 8th graders). See Table 6 for more population details.

Participants met for the first time on the first day of camp each week, unless they were a part of the same family. Many participants in both groups were related, either siblings or cousins. All participants seemed excited to be in a computer programming camp and to help with the research study. Twenty-two (22) participants were assigned to the communal learning group and 20 were assigned to the individual learning group.

Table 10

Descriptive Statistics of the Population in this study

Independent Variables	N	Percentage
Male	25	40.5
Female	17	59.5
African-American/Black	32	76.2
African	1	2.4
Latino-American	1	2.4
Native-American	1	2.4
Bi-racial	2	4.8
Caucasian	0	0
Elementary School	21	50
Middle School	21	50
Communal	22	52.4
Individual	20	47.6
Pairs of Two Boys	8	19.0
Pairs of Two Girls	8	19.0
Pairs of a Boy & a Girl	6	14.3
Individuals	20	47.6

Academic and Computer-Related Experience Prior to Camp

A prior experience questionnaire was given to each participant on the first morning of camp for each group. This questionnaire asked questions regarding participant academic experience, prior computer-related experience, and future career goals. The results of this questionnaire are described below.

Academic experience. Self-reported responses to the Prior Computer Experience Questionnaire (Appendix indicate that 42.9% (18) of the participants had an A grade point average, 38.1% (16) had a B grade point average, and 11% (5) had a C grade point average or below. Eighty-three percent (83.3%) expressed an interest in attending college, while 59.5% reported that they would like to obtain an advanced (graduate) degree. Eighty-eight percent (88.1%) have a computer at home, but only 81% use them. A slightly lower amount (78.6%) use computers at school.

Table 11

Self-Reported Characteristics of the Population in this study (N = 42)

Participant Characteristics	N	Percentage	Missing
Grade point average (GPA)			3
3.5 – 4.0	18	46.2	
2.5 – 3.4	16	41.0	
Below 2.5	5	12.8	
Have a computer at home	37	94.9	3
Uses computer at home	34	87.2	3
Uses computer at school	33	84.6	3
Desires to go to college	35	89.7	
Desires to get an advanced degree	25	64.1	
Taken an online course	29	74.4	3
Uses computer to create, outside of school	28	71.8	3
Never programmed before this study	23	60.0	3
Level of Expertise in Scratch			3
I don't know what this is and have never used it	19	48.7	
I have no experience but I have heard of it	6	15.4	
I've played around with it	4	10.3	
I have used it to make something	8	20.5	
I am an expert and can teach others how to use it	2	05.1	
Desired (Technology-Based) Careers ("Interests Me A lot")			3
Engineer	10	25.7	
Game Designer	11	28.2	
Technology Specialist	10	25.7	
Web Developer	5	12.8	
Mathematician	6	15.4	
Computer Scientist	6	15.4	
Software Developer	7	17.9	
STEM-Related Favorite School Subject			3
Math	11	26.2	
Science	7	16.7	
Math & Science	2	4.8	

Computer-related experience. Sixty-nine percent (69%) of the participants had taken an online course and 66% used computers to create digital media projects outside of school. Fifty-four percent had never programmed and 45% had never heard of Scratch before this study. The remaining participants had at least heard of Scratch, while some reported having played around with it and actually creating something with it. Approximately one quarter reported that their favorite subject in school was either math or science. Of the five choices indicating interest in future career options ("I don't know what this job is," "It does not interest me," "It interests me just a little," and "It interests me a lot"), the top three careers that received a rating of "It interests me a lot" were Game Designer (28% of participants), Engineer (25% of participants), and Technology Specialist (25% of participants). The second highest ratings for career interests included Web Developer (12.8%), Mathematician, Computer Scientist, and Software Developer Table 11 above shares these self-reported participant characteristics and more in detail.

Camp officially began after students submitted their responses to the participant questionnaire about their academic and computer-related experience prior to camp, described above, along with the three pre-tests data collection instruments, namely for Scratch Content Knowledge, the modified Learning Context Questionnaire (LCQ-m) and the Black Academic Identity (BAI) scale. The researcher began both weeks of camp with explanations of how to use the audio recorders and a brief introductory lesson to ensure that all students began camp with a similar understanding of computer hardware, computer software, computer programming, programming/coding (writing computer code), and code, the final result of programming. This was the only time the researcher acted as instructor. After this introductory lesson, participants the video cameras and audio recorders were set to record and participants began reading their Scratch Booklet for all subsequent instructional lessons.

Qualitative Data Analysis: Camp Experience

The qualitative information sources (i.e. notebooks, audio and video recordings, end-of-unit opinion prompts, Scratch projects, and end-of-camp interviews) were used to collect data and used the resulting data analyses to describe and compare how young African American elementary and middle school novice programmers (i.e. participants) in the Communal Learning context learn and use computational thinking and programming the same to those in the Individual Learning context. As a result of the initial/open and process coding methods described in Chapter Three – Methods, perspective of analyses include: behavior, Scratch Booklet and Unit completion, Notebooks, Scratch Projects, and Cognitive Assessments of Problem-Solving and Program Development, and Resulting Scratch Projects. The following sections describe qualitative data analyses from used to answer RQ1:

> RQ1. During a summer Scratch programming camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context?

Behavior. Although not a planned measurement in the design of this study, it is worth noting behavior, as it often impacted each day of this study. As a result of study implementation occurring in two computer labs as opposed to one, as originally planned, managing behavior in both rooms simultaneously became an issue, especially since the building and computer lab reassignment came too late for the researcher to recruit and secure a inter to help manage one of the computer labs.

Most notably, the boys in the Communal Learning group experienced the largest amount of and the most profound behavioral challenges. The behavior exhibited by the boys in this learning group was a mixture of simply being unfocused or off-task along with what is consistent with the ways in which Black male adolescents are described in classroom environments by other researchers who use characterizations called *hegemonic masculinity,* an aggressive and competitive behavior, and at times includes oppositional and confrontational behavior (Connell & Messerschmidt, 2005). In fact, hegemonic masculinity behavior was exhibited and the researcher had to break up a physical altercation between a boy pair in the middle school during lunch on the third day of camp. This altercation disrupted focus for many participants in the middle school for the remainder of the day and resulted in pairs being reassigned for the remainder of the study. Each middle school boy was the paired with a middle school girl. This disrupted the middle school girl pairings. One girl pair in particular was having its own set of challenges as one girl claimed to be doing all the work, while the other girl did nothing but watch YouTube videos. The girl doing all the work made this complaint apparent through tears during lunch on the second day of camp. Another middle school boy participant in the communal learning group often changed the angle of the video camera, in an attempt to make the others in the room laugh. Unfortunately, these daily changes in camera angle, although often eventually corrected by the researcher, presented data collection challenges from this information source. Nonetheless, the daily angle changes ceased once the researcher spoke with this middle school boy participant about his actions and how important it was for him to stop.

Another behavioral issue which commonly occurred with all student was simply being off task. The temptation to play games on the relatively new, fast, computers connected to the institution's high speed internet network was the largest and most frequent behavioral distraction for all participants in both learning groups. These types of (mis)behaviors occurred most often after the lunch break, when all participants when outside in front of the building to each lunch and play. The researcher redirected participants back to focusing on the study material often and daily.

Figure 7. Middle School participants in the Communal Learning Group off-task.

Figure 7 above shows an example of middle school participants in the Communal Learning Group either watching *YouTube* videos or playing a game during a time when they should be working to complete the Scratch Booklets in pairs. Participants are not following instructions in this snapshot. One of these huddled participants has left the girl in the upper right corner to work alone. This experience could potentially lead her to believe that it is better to work alone than in a group. This proposition directly connects to the results of the learning context questionnaire described later in this chapter.

The elementary boys in the communal learning group especially attempted to play games several times throughout the study, and as a result were often off-task. Computer game playing was the largest disruption (relating and not relating to Scratch) in the computer lab designated for elementary school participants. Additionally, excitement caused by a discovery within Scratch *could be* described as the second largest classroom disruption. For instance, while learning about and exploring the different drum sounds, one elementary school boy participant stood up and began performing a popular dance. His dancing caused everyone in this computer lab to look, laugh, and even attempt to join him. On one hand, this can be considered a classroom distraction or disruption, while on the other, it can be viewed as this participant tapping into other cultural elements of the Black Cultural Ethos described in the literature review presented above. The cultural elements of *Movement* – rhythm of everyday life, *Verve* – intense stimulation, action, colorfulness, *Affect* – premium on feelings, expression, and *Expressive individualism* – distinct, genuine, and personal - all apply. These cultural elements may have also come into play when an elementary school boy participant stood up during the post-Scratch content knowledge test and used his own physical steps to help solve one of the post-test problems related to the 'move' block and referred to a series 'steps' taken by a Scratch sprite (see Figure 8 below). This movement and use of ones body is also often referred to as *embodied cognition,* the idea that the body influences the mind, and cognitions arise from bodily interactions with the environment (Jimenez, 1912) and can potentially be closely related to the use of various elements of the Black Cultural Ethos as a way of knowing.

Figure 8. Elementary school boy using embodied cognition to help respond to a post-test question.

Nonetheless, to help maintain ideal study implementation and when behavioral issues became profound, the researcher resorted to contacting parents if/when participant behavior reached an unbearable level. Parent support was strong. With the support of parents, the researcher also warned participants that if their misbehavior continued, they would no longer be allowed to participate. One day, a parent of an elementary boy participant reassured the researcher that after *appropriate communication* between she and her son the night before, there would be no more behavioral issues for the duration of the camp. This reassurance proved to be true.

The one boy & one girl pair type in the communal learning group experienced less severe behavioral issues, which often ceased after being told once or twice by the researcher and/or the researcher made contact with one of the pair participant's parents/guardians. The girls in the communal learning group were extremely excited to meet one another and find so much in common with their assigned partner. This often resulted in the girls talking about topics unrelated to camp and were often reminded to get back and/or stay on task. The elementary girl participants in the communal learning group and all girl participants in the individual learning group were engaged in each unit and presented no behavioral issues. As a matter of fact, on many days, several of the elementary girl participants stayed after camp to work on their Scratch projects more, while waiting to be picked up by a parent or guardian.

The boys in the individual learning group did not present many major behavioral challenges either. There were no instances of *cool pose* or *hegemonic masculinity* behavior. Nonetheless, however, the boys in this learning group were often off-task playing games, watching YouTube clips using their assigned headphones (after taking them out of their assigned audio recorders), or simply playing around and joking with one another. In contrast, the girls in the Individual Learning group did not seem to be as tempted to play computer games, or at least they did not get caught doing so.

Figure 9. A screenshot of a Communal Learning group pair celebration.

Subsequently, not all behavior provided evidence of distractions. Audio recordings reveal that many communal learning pairs were engaged in their work, used a variety of strategies to progress through the booklet, including: 1) taking turns reading, by page and sometimes by paragraph, 2) one partner reads aloud to the other, while the other partner is either listening or maneuvering the mouse working within the Scratch development environment, and 3) each partner reading the material for him/herself and then allowing his/her partner to read the same content. Additionally, many pairs in the communal learning group celebrated as they progressed through the Scratch Booklet. See figure 9 above. Many of the participants in the individual learning group did not read aloud, so it was difficult to determine their strategies for progressing through the Booklet and if they celebrated their accomplishments while at camp.

Scratch booklet and unit completion. The Scratch Booklet served as the source of instructional material for all computational thinking and programming content. Participants had to read the book to get computational thinking descriptions and work through programming examples and explanations. There were 15 units in the Scratch Booklet (see Table 7 for more Scratch Booklet details). After the booklet's introductory chapter, participants were instructed, by written instructions in the Booklet, to skip to Unit 15. There, participants read about computational thinking skills in general, programming, fixing errors, the importance of documenting and writing notes, and other behaviors associated with being a programmer. Once complete, participants were instructed to continue with Unit One and to try his/her best to progress to and finish Unit 14, where instructions were presented to create a game. The average number of units completed by participants in both learning groups was four (approximately 60 pages), although participants in the communal learning group progressed a bit farther in unit 4 than participants in the individual learning group. This equates to about four pages per hour for the duration of the camp.

Participants were instructed and reminded to respond to the 'Opinion Prompts' (Appendix G) after completing each unit. However, the data collected from this form indicates that not all participants remembered or chose to complete this form after each unit. As a result, it cannot be determined if participant completed each unit and decided against or forgot to respond to opinion prompts or if participants skipped some units. It could also be the case that some of the pairs in the Communal Learning Group responded to the 'Opinion Prompts' together. As such, this analysis is of the 29 of 42 participants who responded, 18 of 22 respondents in the Communal Learning Group and 11 of 20 respondents in the Individualistic Learning Group. It should also be noted not all participants who responded, submitted responses for each unit. Table 7 above lists and describes each unit in the Scratch Booklet.

Provided that no units were skipped, the highest average unit completed by participants in both learning groups was Unit Four (Loops). However, the distribution of units completed by each learning group vary, with 14 of the 18 participants who responded from the Communal Learning group completing between Units Four (Loops & Conditionals) through Unit Six (Conditional Loops), while only two of the participants in the Individualistic Learning Group completed Units Nine (Complex Conditionals) and Units 10 (Operators) and seven of the 11 respondents to the 'Opinion Prompts' in the Individualistic Learning Group only completing up to Unit Three (Sequences). The highest units completed in each learning group were Unit 10 (Operators) in the Individualistic Group and Unit Eight (Working with Data) in the Communal Group. The number of participants who responded within each group is evenly distributed for both boys and girls, pair-types, elementary, and middle school respondents.

Looking at grade-level, the middle school participants in both learning group completed more than half of Unit Five, with the middle school participants in the individual learning group completed slightly more units (4.8 units) than the middle school students in the communal learning group (4.5 units). Of those in both grade-levels in the communal learning group, the pair of boys and the girl & boy pair completed the most. The middle school two girls pair-type completed the least units of all (three units), while the elementary school 2 girls pair-type completed 3.5 units. Of those in the individual learning group the elementary school participants completed the least amount of unit (3.5 units) compared to the middle school participants (4.8 units).

Nonetheless, many participants were honest about how they spent their time. One opinion prompt question asked at the completion each unit was, "Did you make the best use of your time to learn during this unit?," one middle school girl in the communal learning group responded, "*no. we spent most of our time goofing.*"

A middle boy in the communal learning group responded, "*yes I sometime focus sometimes played around but yes*."

An elementary school girl responded, "*I think I could have gone faster if I paid attention more*."

It should be noted that for the most part, participants in both learning groups responded to this opinion-prompt question with, "yes."

When asked to describe strategies used to read through and complete each unit, the boys in the communal learning group could not provide a detailed description. The most common response provided by boys and the two boys pair-type in this learning group related to the speed at which they worked.

An elementary boy often replied, "*slow and steady*," for each unit while others remarked, "*work fast*" or "work hard" or "*we worked fast at first, but then we went back and did it slow. slow and steady wins the race! :-)*"

The most common response given by the one boy & one girl pair-type was, "*take turns*" or "*Our strategy was to take turn reading paragraphs.*"

The girls provided the most descriptive details about strategies used, "*we took turns reading page by page, and after we'd read all the material we did the stuff instructed. then we goofed till snack.*"

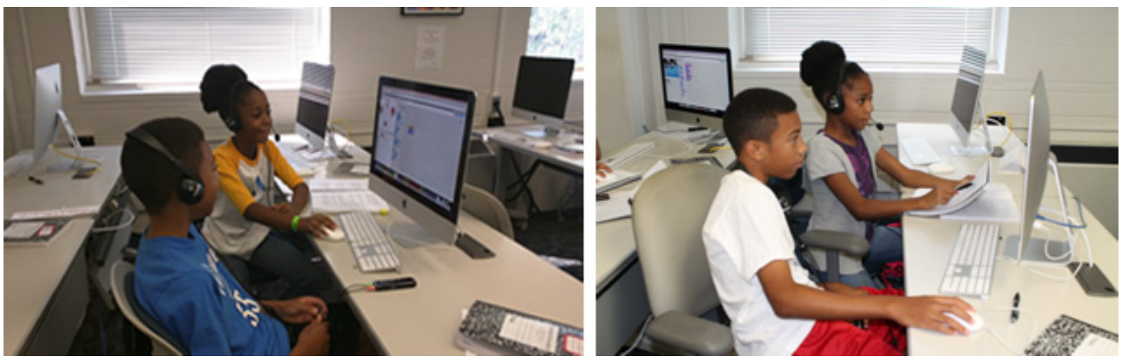


Figure 10. Elementary school Communal Learning Boy & Girl Pair taking turns controlling the computer from one day to the next.

Figure 10 above depicts an elementary school communal learning boy and girl pair taking turns controlling the mouse as they read through the Scratch Booklet. It is worth noting that the girl in Figure 10 has the book on both days and seems to have read the entire time. They verbally agreed to this strategy. Participants in the individual learning group simply described their strategy as reading and trying the activities as they progressed. This strategy was shared in the written responses as well as the interviews.

When asked what did the participant find hard or challenging after each unit, three response themes dominated for the Communal Learning group overall:

1. *Nothing*
2. *Getting the sprites to work interactively*
3. *My partner*

The Individualistic Learning group indicated these themes in their responses most often:

1. *Nothing*
2. *Getting the sprite to move/rotate*

The audio recordings from day to day provide insight into the reading abilities of each participant. Study participants in both learning groups sound as if they could have challenges reading. Some read extremely slow, while others seem to read words without full comprehension.

Notebooks and problem-solving and program development assessment. Individual notebooks were given to each participant on the first day of camp for each learning group. These were made available for use at the discretion of each participant. Participants were told they could scribble, draw, write notes related to what they were reading, topics they wanted to remember, ideas for Scratch projects, and/or problem solving information. Most participants in both learning groups took notes during the introduction to Scratch slideshow presentation and discussion and then wrote a few lines of notes from Unit One but did not continue to write in their notebooks. The keyword that appeared the most in notes was 'Initialization,' which was very often spelled incorrectly, along with the dimensions of the Scratch stage, which varied from participant to participant. Some indicated 180x180 pixels while others indicated 240x240 pixels.

The actual dimensions of the Scratch stage is 240x360, but it is referred to as a square with coordinates of the x & y positions of each point in the square, with the lower left coordinates of x = -240, y -180 and the upper right coordinate of x = 240, y = 180. Some participants used their notebooks to draw but did not indicate the purpose of the drawings, although many seem related to some type of Scratch project. Figure # features a series of photos of pages from participant notes. Ninety percent of all participants only wrote notes similar to the above description. The other 10% also included notes from the introductory session given by the researcher to ensure everyone started camp hearing the basic definitions hardware, software, programming/coding, and other related concepts previously described.

Thussday

Initilizatin:

Starting up

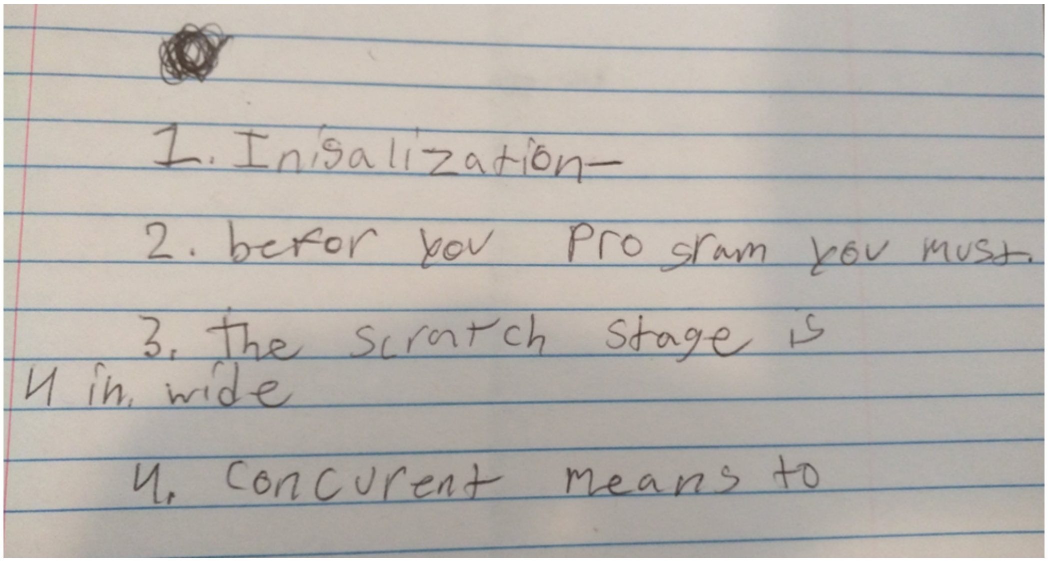
Sciatch stage is 240x - 240

concurrent mean things happening at the sametime.

1 inslisation:

~~right~~ right mebate

~~themida~~ the middle of ~~ab~~ ar v the s

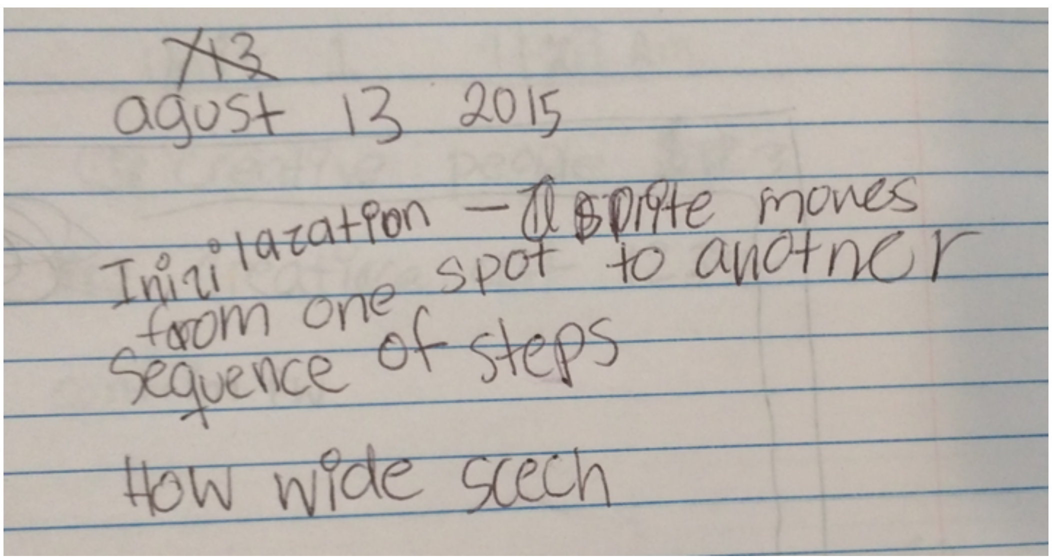


1. Inisalization-

2. befor you Pro gram you must.

3. the scratch stage is
4 in. wide

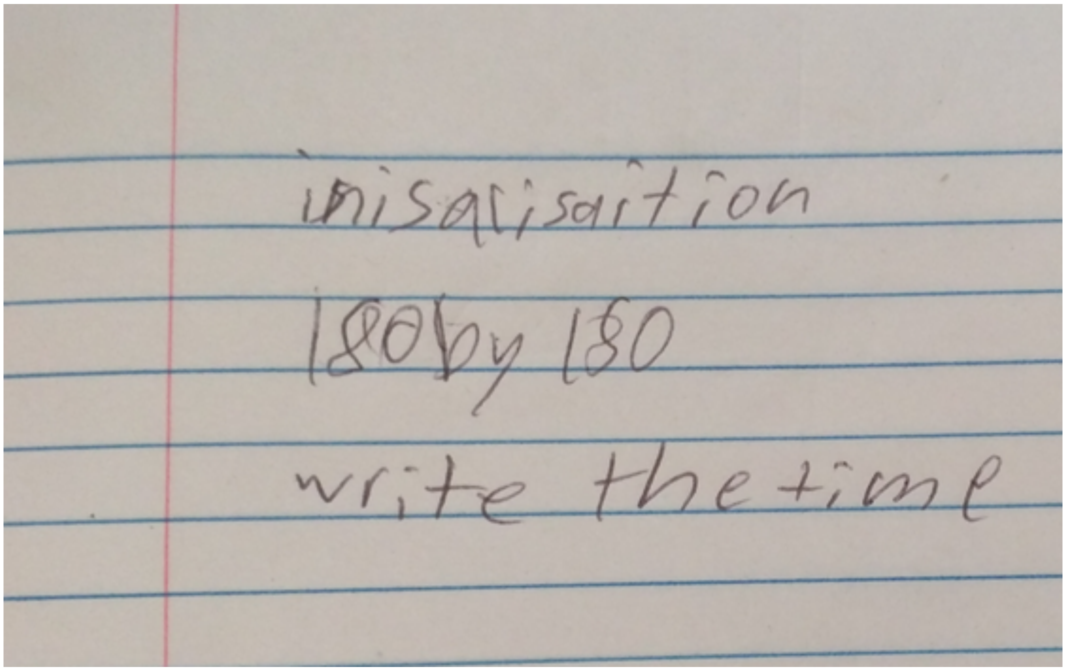
4. concurent means to

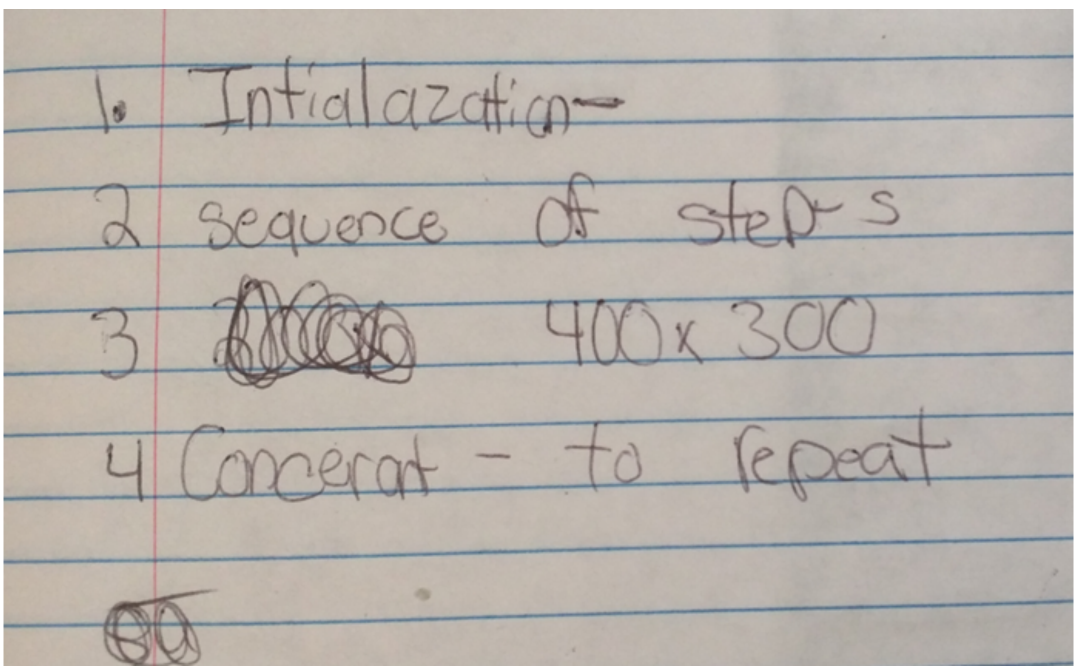


~~13~~
agust 13 2015

Inizilazation - ~~a~~ sprite mones
from one spot to another
sequence of steps

How wide scech


inisalisaition
180 by 180
write the time


1. Intialazation
2 sequence of steps
3 400 x 300
4 Concerant - to repeat

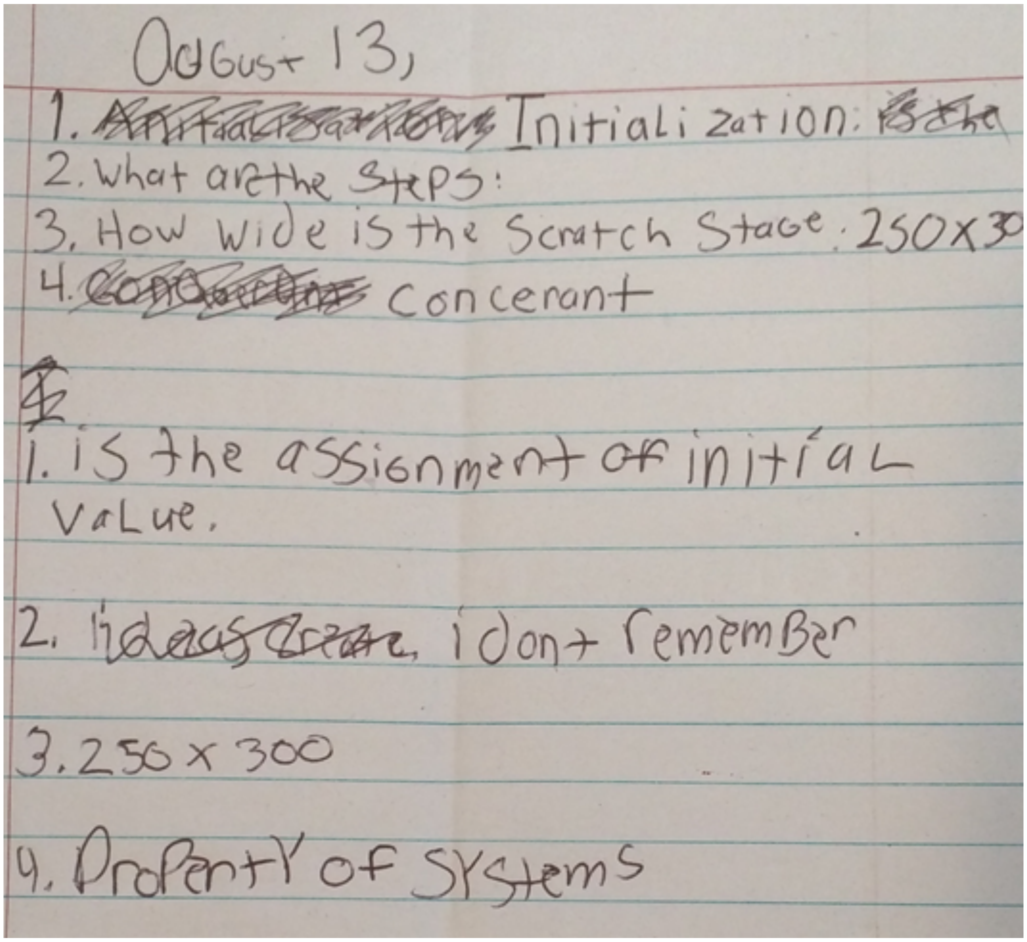


Figure 11. A Series of 7 notebook entries about Initialization and Scratch Stage size.

It is unclear why the participants who wrote this set and only this set of notes in their notebooks. Of those who wrote these notes, none wrote anything else. It is worth noting that many had difficulties spelling 'initialization' even though the word is written in the Scratch Booklet. See figure 11 above.

Cognitive assessment of participants' problem-solving and program development. The content of the figures above illustrate the only content written in the notebook. As a result, the scores on the Cognitive Assessment scores for Formulating the Problem (4), Planning the Solution (4), and Designing the Solution (4) (e.g. the Process) were zero out of 12 points for all participants in both the Communal Learning and the Individual Learning Groups. These scores illustrate that all participants learned little to no problem-solving and program development skills. These skills are closely related to computational thinking skills. The assessment scores for the Product (i.e. Solution Efficiency (2), Solution Reliability (2), Solution Readability (2), and Solution Correctness(2)) were no more than four out of a total possible eight points and are low assessment scores. Evidence of these scores is featured the next section – Resulting Scratch Projects.

Resulting Scratch projects. Throughout the Scratch Booklet, all participants were encouraged to create various Scratch projects. These projects included: a) worked examples in the Scratch Booklet that can be used and remixed by participants (these were created ahead of time by the researcher and stored in a Scratch studio located at https://Scratch.mit.edu/studios/1428272/ and made available to each participant on the camp's website), b) original Scratch projects created by study participants at will, and c) projects throughout the Scratch website shared by other Scratchers available for remixing by participants. Participants in both learning groups were encouraged to share their projects, as viewing and analysis of projects that are not shared is not feasible. Unfortunately, many participants in both learning groups did not share their projects, despite several daily reminders and encouragement from the researcher. It is unclear if the reason for not sharing was because participants forgot to share or did not want to share their projects. The screenshots featured here were selected from those participants in both learning groups who shared their projects on Scratch.

Previous projects created and shared by the researcher. Of the Scratch projects created previously by the researcher and made available for participants in either learning group to remix, none were remixed more than once. The ‘cat-meets-dog’ project was viewed the most (15 times) and remixed twice. This project is featured in Unit Three and illustrates how to move two sprites (characters) around the stage autonomously when the green flag is clicked, using ‘glide’ ‘point in direction’ and ‘go to’ command blocks. Figure 12 below is a screenshot of ‘cat-meets-dog.’

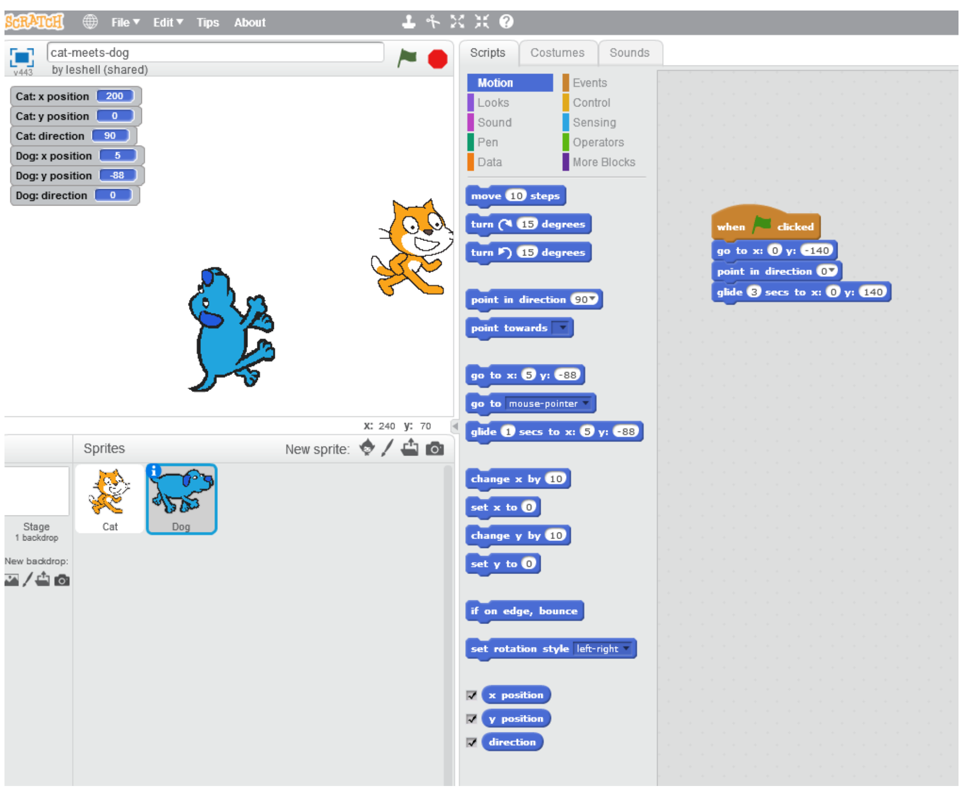


Figure 12. Screenshot of Scratch project ‘cat-meets-dog’ (sprite motion instruction).

The least viewed available Scratch project created by the researcher was 'get-bone2.' It was viewed twice and never remixed and is featured in Unit Five. The shared Scratch project created by the researcher called 'get-bone2' illustrates how to communicate between sprites using messages, features a forever loop, sprite movement, and sensing, which is used to determine when a sprite is touched by another sprite. In this case, the dogs race to get the bone. Figure 13 below is a screenshot of 'get-bone2.'

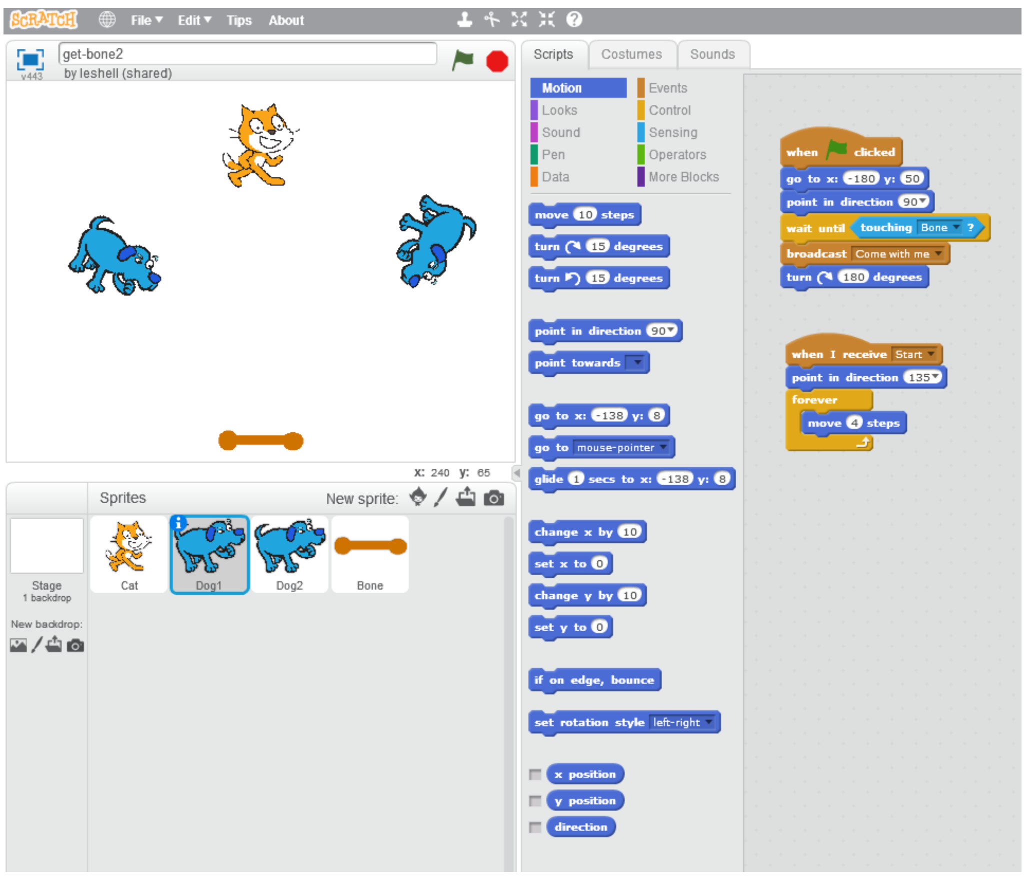


Figure 13. Screenshot of Scratch project 'get-bone2' (communication, movement, sensing).

Since these projects were the most and least viewed and are featured in Unit Three and Unit Five respectively, this totals supports unit completion by at least 15 participants was up to Unit Three and at most Unit Five. This is consistent with the average Scratch Booklet unit completion reported by all study participants.

Further analysis of end-of-camp interview questions reveal that many participants in both learning groups were not yet aware of the remixing feature and the ability to remix these projects previously created by the researcher. Instead, they simply showed the researcher's project on one screen and recreated it on another. One example of this is 'BowRace' created by a one boy & one girl pair in the communal learning group. This pair copied the 'get-bone2' project with their own sprites. Doing so took more time than what may have been needed to simply remix 'get-bone2,' especially since this pair added more dialogue via sprite communication and messaging. These additions can be seen in figure 14.below, a screenshot of 'BowRace.'

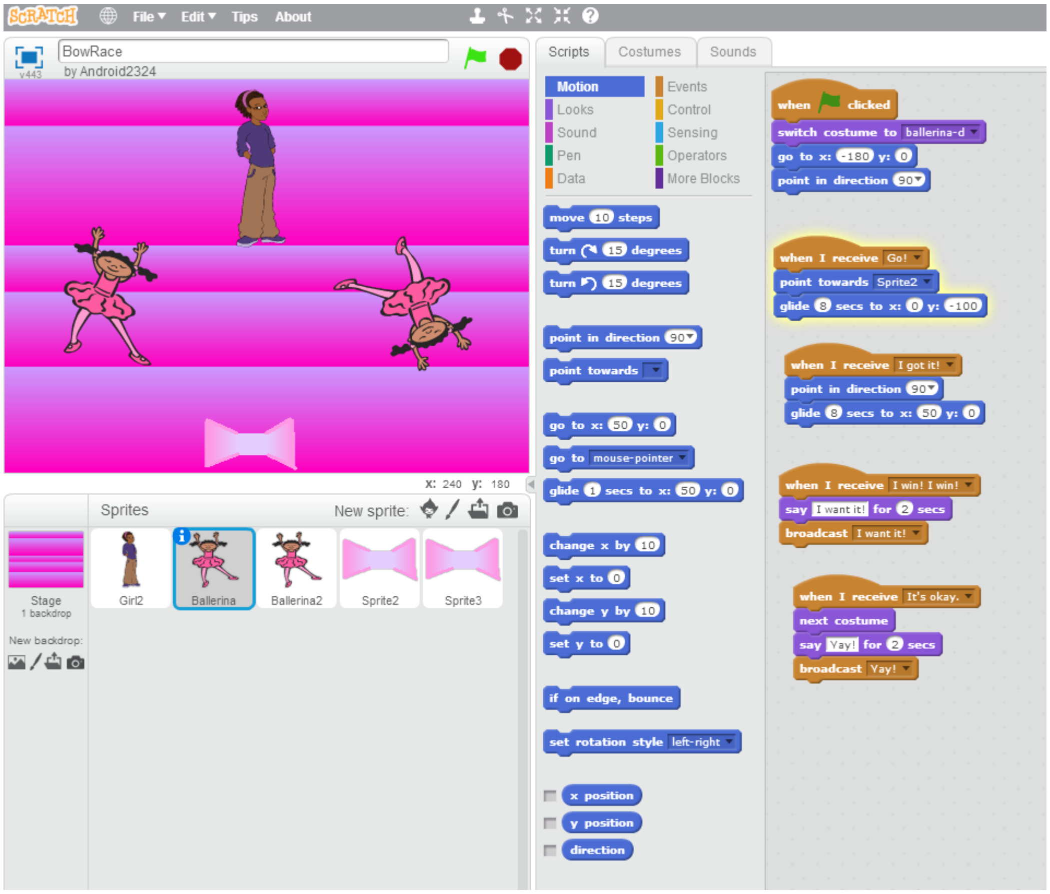
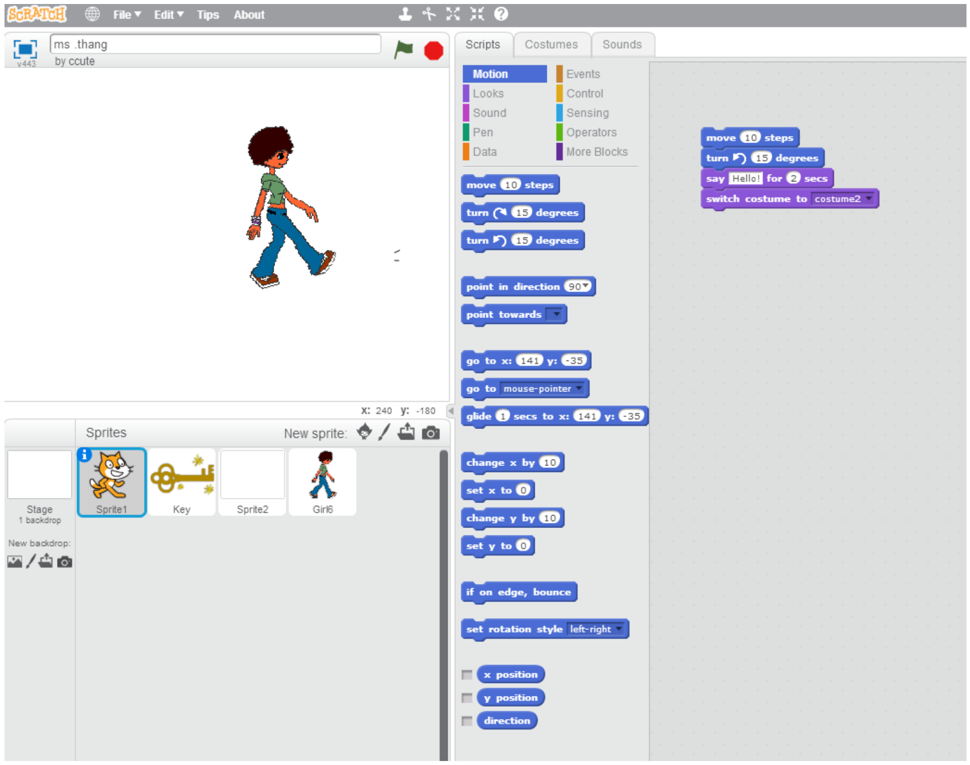
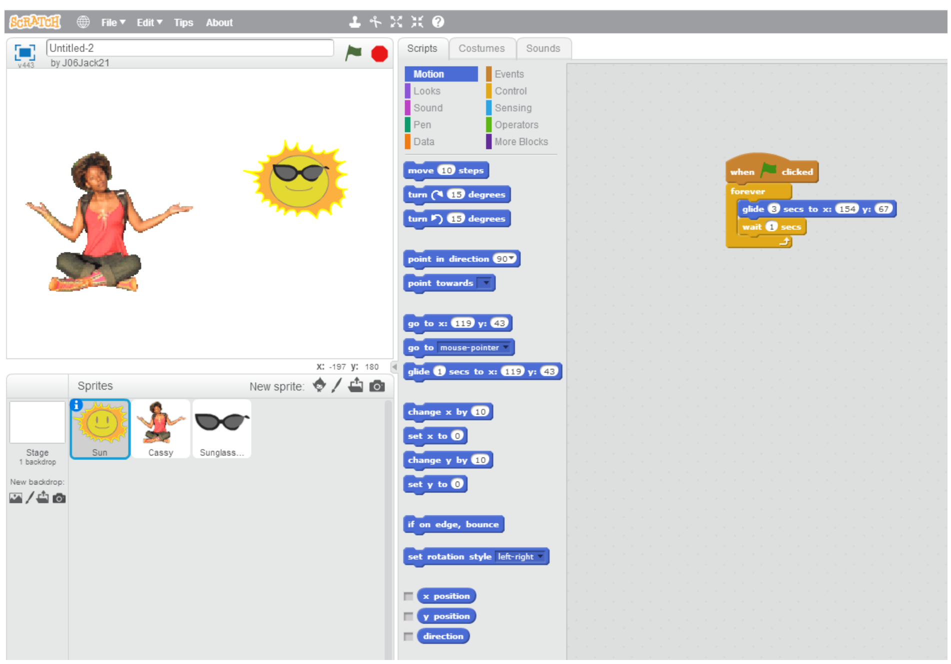
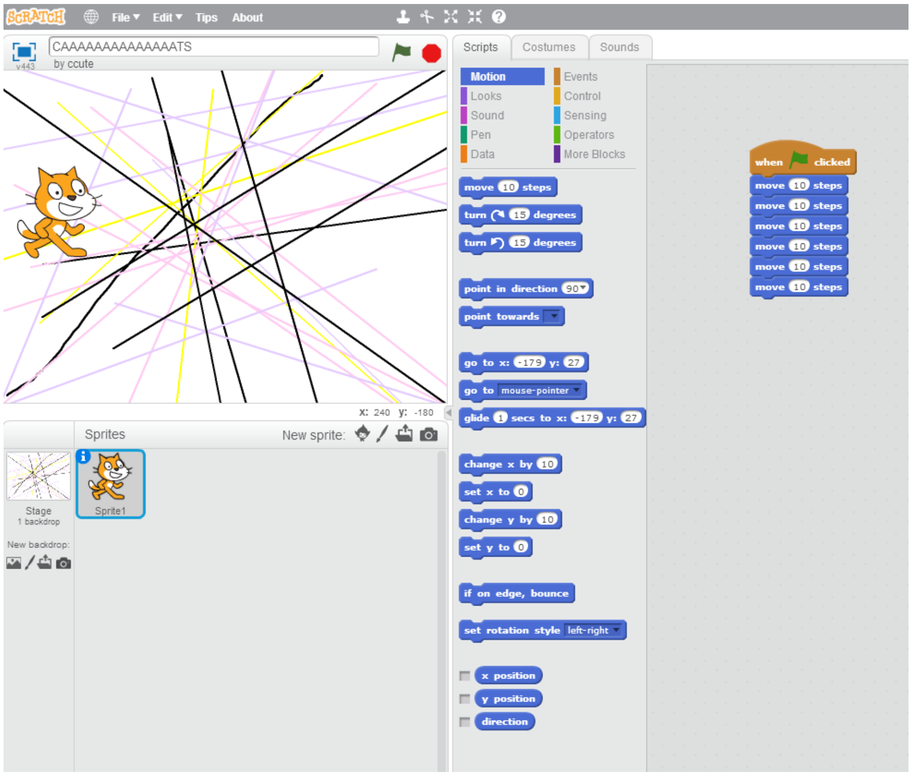


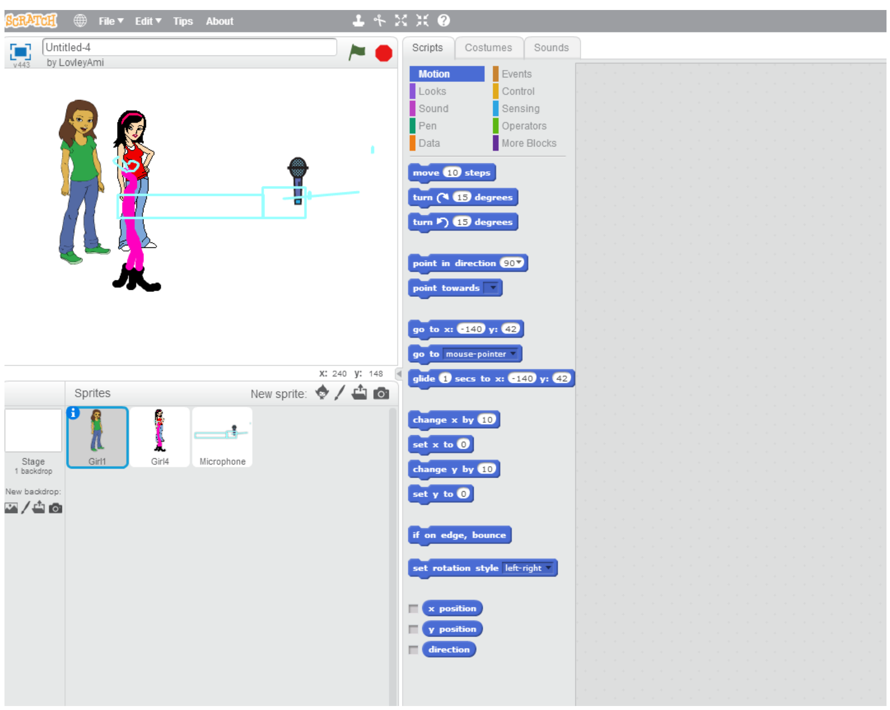
Figure 14. Screenshot of 'BowRace' a unique Remix of Scratch Booklet project.

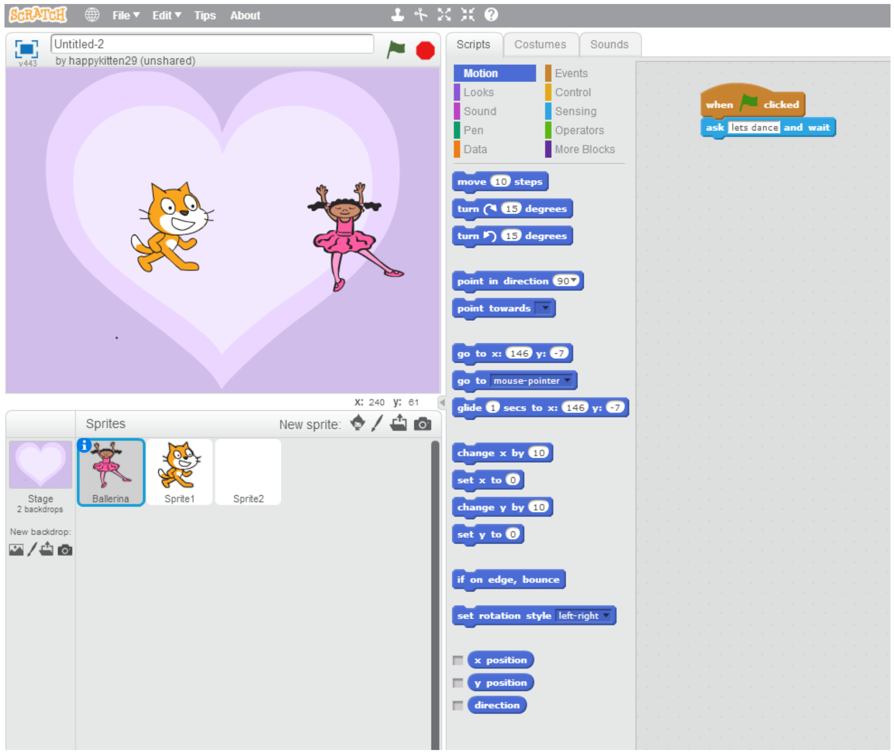
Original* Scratch *creations. Scratch projects that were original creations by study participants most often contained one or two sprites and less than five program blocks. These were the most common levels of complexity (not at all) of all resulting Scratch projects. Many of these program blocks were from the motion category and/or the looks category and were used in an attempt to move at least one sprite and/or make it appear to say or think something. Figure 15 below illustrates several examples of these types of projects from participants in both learning groups.


SCRATCH
File
Edit
Tips
About
ms .thang
by ccute
Scripts
Costumes
Sounds
Motion
Looks
Sound
Pen
Data
Events
Control
Sensing
Operators
More Blocks
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: 141 y: -35
go to mouse-pointer
glide 1 secs to x: 141 y: -35
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
set rotation style left-right
x position
y position
direction
move 10 steps
turn 15 degrees
say Hello! for 2 secs
switch costume to costume2
x: 240 y: -180
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
Sprite1
Key
Sprite2
Girl6


SCRATCH
File
Edit
Tips
About
Untitled-2
by J06Jack21
Scripts
Costumes
Sounds
Motion
Looks
Sound
Pen
Data
Events
Control
Sensing
Operators
More Blocks
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: 119 y: 43
go to mouse-pointer
glide 1 secs to x: 119 y: 43
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
set rotation style left-right
x position
y position
direction
when clicked
forever
glide 3 secs to x: 154 y: 67
wait 1 secs
x: -197 y: 180
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
Sun
Cassy
Sunglass...


File
Edit
Tips
About
CAAAAAAAAAAAAAAATS
by ccute
Scripts
Costumes
Sounds
Motion
Looks
Sound
Pen
Data
Events
Control
Sensing
Operators
More Blocks
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: -179 y: 27
go to mouse-pointer
glide 1 secs to x: -179 y: 27
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
set rotation style left-right
x position
y position
direction
when clicked
move 10 steps
move 10 steps
move 10 steps
move 10 steps
move 10 steps
move 10 steps
x: 240 y: -180
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
Sprite1


File
Edit
Tips
About
Untitled-4
by LovleyAmi
Scripts
Costumes
Sounds
Motion
Looks
Sound
Pen
Data
Events
Control
Sensing
Operators
More Blocks
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: -140 y: 42
go to mouse-pointer
glide 1 secs to x: -140 y: 42
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
set rotation style left-right
x position
y position
direction
x: 240 y: 148
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
Girl1
Girl4
Microphone



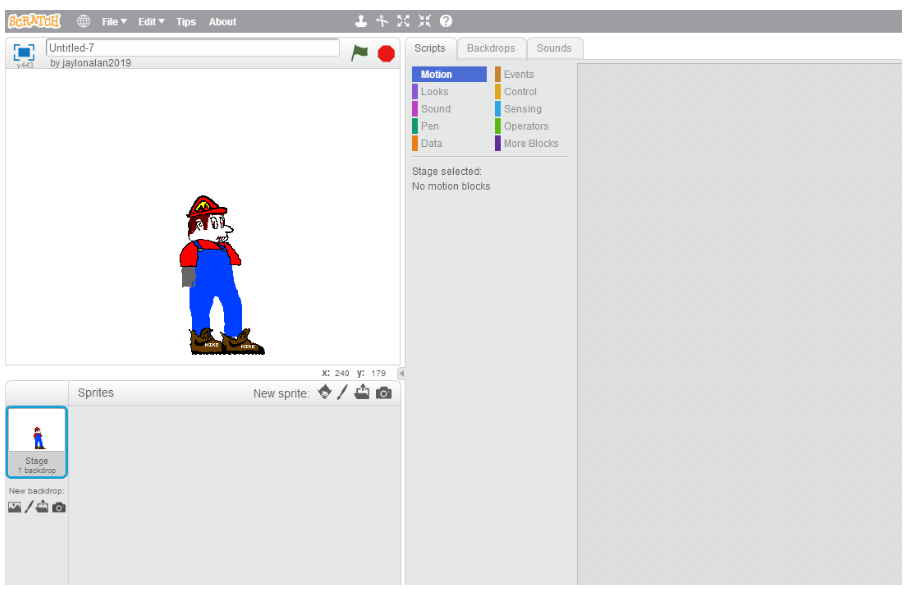


Figure 15. A series of 6 Scratch projects with minimal sprites and program blocks.

This occurrence supports “getting the sprite to move (interactively)” as one of the top most challenging features of Scratch reported by participants in both learning groups, as described above.

Complex* Scratch *projects. Participants specifically in the one boy & one girl pair type and the two girls pair type created the most complex original Scratch projects of all study participants. One example of a comparatively complex Scratch project is *‘relay race 1’* by an elementary one boy & one girl pair-type. This project features four sprites and a background with interactive events, movement, sensing, conditionals, and looks (dialogue) and can be seen in figure 16 below. Another relatively complex Scratch project created by a middle school communal learning pair of two girls is *‘shark meets ghost fish.’* This project features seven sprites and a background with events, movement, sound, and looks (dialogue). It also seems to be a creative extension of ‘cat-meets-dog’ created by the researcher for demonstration purposes and described above. Figure 17 below is a screenshot ‘shark meets ghost fish.’

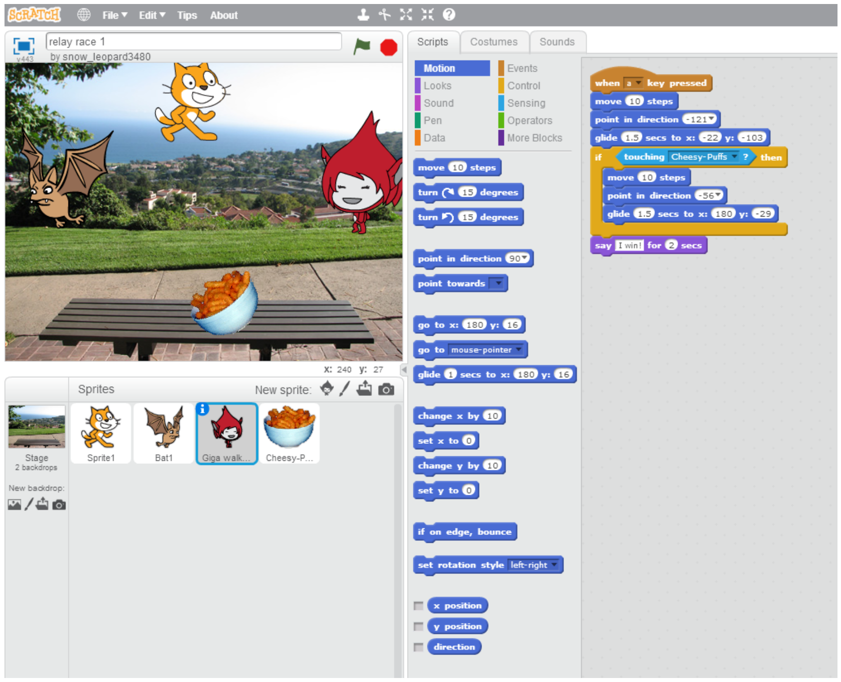


Figure 16. Example of a complex Scratch project called *'relay race 1.'*

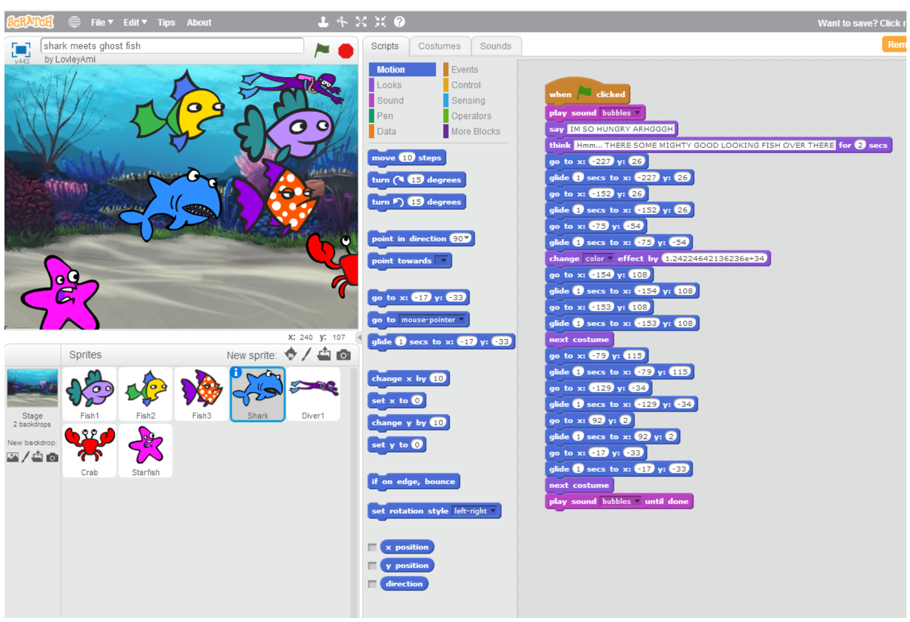


Figure 17. Example of a complex Scratch project called *'shark meets ghost fish.'*

Remixes of other Scratchers' projects. Another common occurrence regarding the type of Scratch projects created is those remixed from shared projects created by other Scratchers in the Scratch community. This was done by all participants in the study, regardless of learning group.

Creativity and art. Another category of Scratch projects originally created by study participants includes those which were high in artistic ability and/or creative expression. These projects demonstrate other features of the Scratch programming language and illustrate that some participants spent a great deal of time drawing within Scratch. Figure 18 below shows a middle school boy participant who was assigned to a middle school boy pair in the communal learning group. He wanted to make a Super Mario game in Scratch and chose to spend a considerable amount of time drawing Mario. The researcher postulates the reason his partner is not seen in the screenshot is because he may have been disengaged due to the amount of time this participant spent drawing and creating. These phenomenon are connected to the performance of the middle school two boy pair-type in the communal learning group, described above and referred to again below (i.e. Unit completion and Scratch content knowledge pre- and post-test scores).

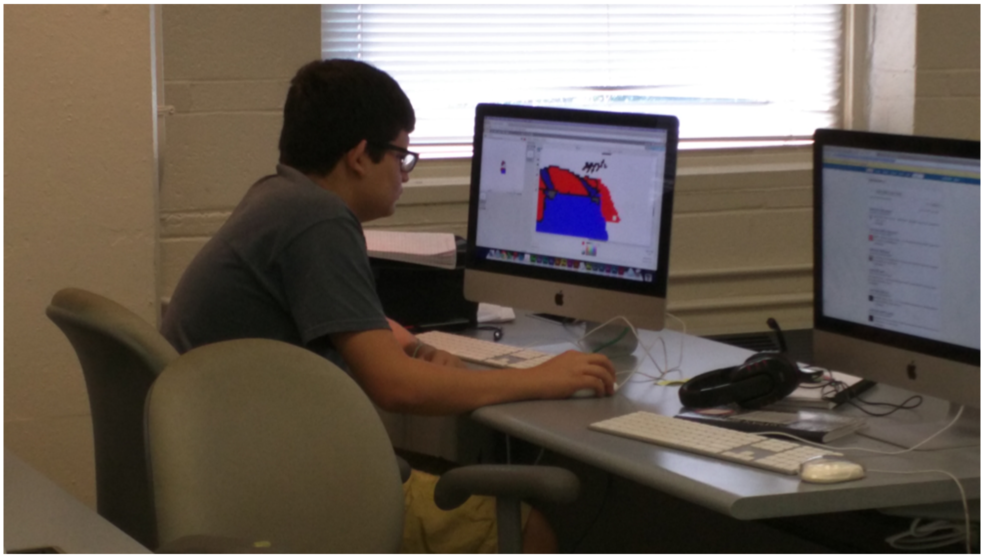


Figure 18. Participant drawing a Super Mario character for his Scratch project.

End-of-camp interviews. One the last day of each week of camp, the researcher selected a few participants to interview. The interview questions can be found in Appendix H. One participant from each of the independent variables in this study were selected for interviewing (one elementary school girl participant from each pair-type containing girls in learning group, one elementary boy participant from each pair-type containing boys in learning group, one middle school girl participant from each pair-type containing girls in the learning group, and one middle school boy participant from each pair-type containing boys in the learning group, one elementary girl and one middle school from the individual learning group and one elementary and one middle school boy from the individual learning group. This totaled 12 interviewees. The levels of engagement and behavior issues experienced by these participants vary. Most of the participants interviewed said they would go back and read any sections pertaining to

Scratch concepts on which they were stuck in an attempt to solve a problem and get 'unstuck.' The Scratch community feature used most often by these participants was the remixing function. Only a few had actually commented on Scratch project. Participants really enjoyed learning Scratch and seemed to list several likeable features of Scratch. These include "the ability to create anything you want," "the ability to look at other projects to see what is possible," "the fact that you can start programming even if you have don't know how," and *"the ability to remix other people's projects."* When asked what was disliked about Scratch, two very interesting responses were given, *"that you can't tell which is a good project that works and which is a bad project that doesn't until you spend time looking or remixing it,"* and *"that* Scratch *doesn't have one path for beginners and another for more advanced users once you login."* The researcher views these responses as thoughtful and engaging responses regarding the Scratch development environment overall. These dislikes resulted into suggestions for changes to Scratch. The conclusion at the end of camp for those in the communal learning group was that they like working with their partner overall, although it should be noted that one of the top three challenges experienced by participants in the communal learning group was "*my partner*." Overall students enjoyed reading the Scratch Booklet, regardless of learning group assignment, while some suggested a preference of a mixture of Booklet and teacher/facilitator for optimal teaching and learning of computational thinking skills. However, audio recordings suggest that many participants had trouble reading. Perhaps those participants who enjoyed reading the Scratch Booklet are not aware of any reading or comprehension challenges they may have. Nonetheless, all participants interviewed found value in learning to program, while only some of the older participants found and described a connection between being a young black <girl/boy> and doing well while learning to program in Scratch. Additionally, while none of those interviewed logged into Scratch during the week of camp, all of them said they would login now that camp was over.

Summary of Qualitative Data Regarding Camp Experience

Overall student engagement in camp and in learning to program with Scratch was relatively high for all participants despite the fact that this study took place in the middle of their summer vacation. Behavioral issues by elementary and middle school boys in both learning groups caused major distractions throughout camp, however. Additionally, all participants were tempted to play games on a daily basis, which cause the second largest amount of distractions and behavioral challenges. Girls who made up the two girls pair-type in the communal learning group and girls in the individual learning group were the most engaged and on-task of all participant pair-types. This was the most profound distinction between all pair types and learning group participants. However, despite this seemingly high level of engagement, Scratch Booklet Unit completion was relatively low, with an average of four Units completed by everyone, regardless of learning group. This suggested a page reading speed of four pages per hour. As such, most participants were only able to attempt programming projects that involved creating sprites, moving them, and attempting to make them interact with one another (e.g. dialogue).

While attempting to complete the Units in the Scratch Booklet and their accompanying Scratch projects, no participant managed to show evidence of performing and problem-solving and program development skills. No participant recorded evidence of going through the process stages for the Dual Common Model for problem-solving and program development (Deek, Starr, Kimmel, & Rotter, 1999) of Formulating the Problem, Planning the Solution, and Designing the Solution. Attempting to make sprites interact with one another was expressed as one of the top three Scratch challenges by participants in both learning groups and Scratch projects with only a few sprites and little to no programming blocks were the most common resulting project type. Other Scratch projects types included: 1) the attempt to make more complex projects using program blocks that specifically speak to the computational thinking and programming skills targeted in this study (sequences, events, loops, conditionals, operators, and working with data), 2) projects that were remixed by other Scratchers in the Scratch community that were not related to this study, and 3) extremely creative and art-based projects where participants spent a considerable amount of time drawing and creating their own sprites. One interesting project type that emerged was created by those participants who did not understand from the reading that example projects in the Scratch Booklet were already created, stored in a Scratch studio, and available for their use as they work through examples in the book. As a result, instead of remixing these available projects, participants opened them on one computer screen and recreated them originally using another computer. These resulting projects allowed for students to select their own sprites and add more complexity to the project. While these projects can still be considered remixes, they are a new type of remix not often discussed in the literature.

Learning group comparisons. Overall, comparisons between the communal learning group and the individual learning group can be made in two areas: 1) Scratch projects and 2) learning process/behavior. These comparison details are share below.

Regarding the completion of Scratch projects, the two learning groups were similar in that both groups completed up to a little more than Unit four (Loops) of the Scratch booklet and no one wrote in the notebooks in an effort to problem-solve, brainstorm, plan, or design. Differences between the two learning groups specifically focused on liking to read, preference for method of learning Scratch should there be a next time, what they found challenging in Scratch, and creativity and complexity of the resulting projects. The communal learning groups liked to read, by generally read slow, where as the individual learning group did not like to read. If there was a next session of Scratch camp, those in the communal learning group preferred to have both a teacher and use of the Scratch booklet, whereas the those in the individual learning group generally preferred a teacher and no book. Participants in the communal learning group indicated that making sprites interact and working with their partners as the two most challenging aspects of camp. On the other hand, those in the individual learning group indicated that making sprites move was the most challenging aspect of camp. With this, the pairs in the communal learning group created more complex and more creative Scratch projects than the booklet prescribe, while those made by the participants in the individual learning group where exactly like the booklet prescribed and nothing more.

When considering the learning processes and/or behaviors of the two learning groups, comparison can be made related to strategies for reading the Scratch booklet, other forms of expression while learning, causes of distraction, and levels of encouragement and engagement. Details about these comparisons are shared below.

When it came to using time wisely, participants in both groups reported that they did not use their time as wisely as they should or could have. Participants in the communal learning group used a variety of strategies for reaching the Scratch booklet. These strategies changes from day-to-day and sometimes throughout the day. See figure 10 above as an example Pairs would agree to take turns reading pages, paragraphs, and even reading sections to themselves and then testing each other after both members of the pair finished reading. Participants in the individual learning group seemed to read silently to themselves each day. Even more, as each week of camp progressed, participants in the communal learning group expressed other elements of the Black Cultural Ethos. Many used movement, especially when learning the concepts related to loops and music was used, expressive individualism, affect, and verve. Participants in the individual learning group did not express as many BCE elements and were generally quiet. They did however, use movement as many used embodied cognition to help them think through a challenge. See figure 8 above. When it came to sources of distraction, the communal learning group had 'good' and 'bad' sources. They were often distracted by computer games and YouTube videos, while participants in the individual learning group did not succumb to these distractions. Additionally, participants in the communal learning group had 'good' distractions involving celebration, encouragement, and sharing of accomplishments with their neighbors when a project went well. Participants in the individual learning group did not exhibit noticeable moments of celebration or self-encouragement. Finally, regarding levels of engagement, participants in the communal learning group seemed more engaged overall, especially the two girl pair-types. They often asked to stay after camp and requested that their parents wait a while as they worked more on their projects. Participants in the individual learning group did not seem as engaged, as they often deleted their Scratch projects and some even deleted their Scratch ID immediately after camp.

While the previous section provides a rich description and comparison of participant experiences in both learning groups, the next section provides quantitative analyses and comparisons the performance of each study participant according to responses on the pre- and post- computational thinking and programming skills content-knowledge questionnaire.

Quantitative Data Analysis: Camp Experience

This section illustrates quantitative findings related to the camp experience for participants in both the communal learning group and the individual learning group. These data are used to help answer RQ1, RQ2, and RQ3 and appear in that order. For each research question, descriptive statistics and graphs are illustrated with explanations, followed by indications of whether or not assumptions for each respective statistical tests were met. These are followed by tests to determine statistical significance, effect size, and power. The next section starts this format and features the quantitative data for RQ1.

RQ1: Scratch Computational Thinking and Programming Scoring and Analysis

This section features the quantitative data findings used to help answer RQ1: RQ1. During a summer Scratch programming camp lasting five week days, three hours per day, how do young African American elementary and middle school novice programmers in a Communal Learning (CL) context learn and use computational thinking concepts and programming skills compared to those in an Individual learning (IL) context? The Pre- and Post-Test Scratch Content Knowledge Questionnaire provided the data to be analyzed here. It featured 25 questions, each asking about one or all of the six Scratch CT and programming skills (sequences, events, loops, conditionals, operators, and working with data). Table 12 below details the CT and programming category, questionnaire questions that relate to that category, total number of questions in that category, and the total possible points earned for that category.

Table 12

CT and Programming Skills Categories on Scratch *Content Knowledge Questionnaire*

Category	Pages in Booklet	Test Questions in Category	Number of Questions	Total Points
Sequences	29.5	2, 3, 10, 19, 20, 22, 24	7	14
Events	9.5	13, 20	2	4
Loops	28	7, 11, 12, 16, 17, 20, 22, 23	8	16
Conditionals	66	8, 9, 14, 18, 20, 24, 25	7	14
Operators	29	20, 22, 24	3	6
Working with Data	52	1, 4, 5, 6, 15, 20, 21, 22, 24	9	18

Preparation and interrater reliability (Cohen's Kappa). To prepare for scoring overall, each content knowledge questionnaire item was labeled as a variable in SPSS with one or more of these six skill groups (e.g. a content knowledge question containing the concepts of loop and one with a concept of conditionals would have received a L and C as labels, respectively). Participants responses to each item received a score of two if it was correct, one if it was partially correct, and zero if it was in correct. Cohen's κ (crosstabs) was analyzed in SPSS to determine the level of agreement between two third-party raters scoring of open-ended questions. There was very good agreement level between the two raters, $\kappa = .882$, $p < .0005$. Using the same scale, the researcher graded the multiple-choice questions. Points were tallied for each participant and these data were used to make comparisons. A total of 50 points were possible on this computational thinking and programming performance instrument.

The next sections features these quantitative results regarding the overall Scratch Content Knowledge and each individual CT and programming skills taught using Scratch (sequences, events, loops, conditional, operators, and working with data). For each CT concept, pre- and post-test descriptive statistics are illustrations and their accompanying descriptions include:

1. A colored 2-bar graph – This first bar graph compares learning group mean scores (the communal learning uses the color green, while the individual learning group uses blue);

2. A colored line graph – The first bar graph is followed by a colored line graph representing pre- and post-test mean scores by learning group (i.e. communal vs. individual), where the time (pre- and post-) is represented along the x-axis and the mean scores for each time is along the y-axis. A colored line is drawn from the pre-test mean to the post-test mean for each learning group(the communal learning uses the color green, while the individual learning group uses blue);
3. A colored 4-bar graph - These colored bar graphs illustrate the post-test mean scores by pair-type (boy & girl pair-type is red, two girl pair-type is yellow, two boy pair-type is green, and individual is green). Please note that for the sake of comparison in these bar graphs, the individual group is a pair-type even though that bar represents all of the individual participants.
4. A colored line graph - The bar graphs are followed by another set of line graphs illustrating the mean gender differences within the individual learning group.

Descriptive statistics for overall Scratch content knowledge scores. There were 20 participants in the Individual Learning Group and 22 participants in the Communal Learning Group. Regarding, the overall The participants in the Communal Learning Group ear than the participants in the Individual Learning Group on the CT and programming post-test. The Communal Learning Group's mean Scratch Content Knowledge post-test score was higher than the Individual Learning Group's mean score.

Table 13

Learning Group Scratch *Content Knowledge Post-Test Mean Scores*

Learning Group	n	M	SD
Individualistic	20	16.80	7.32
Communal	22	18.95	9.30

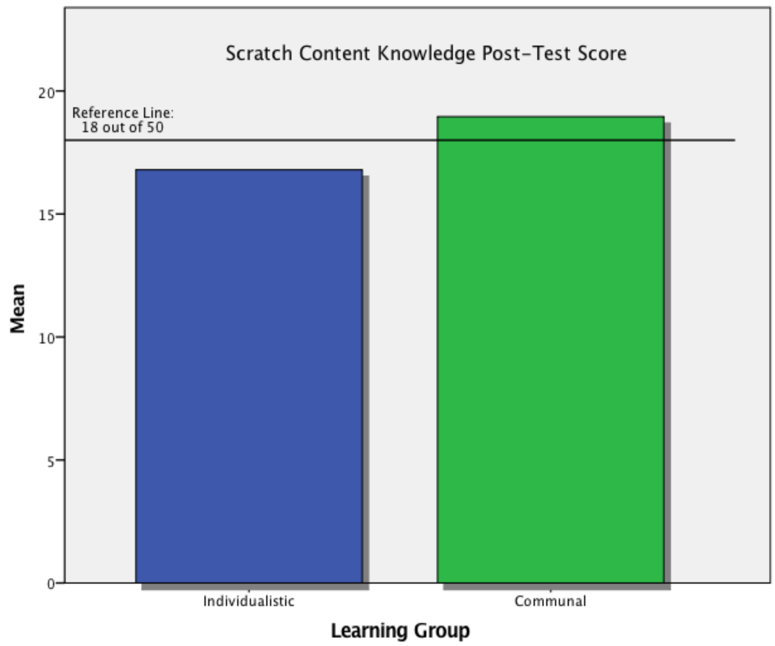


Figure 19a. 'Scratch' Content Knowledge Post-Test Mean Scores by Learning Group.

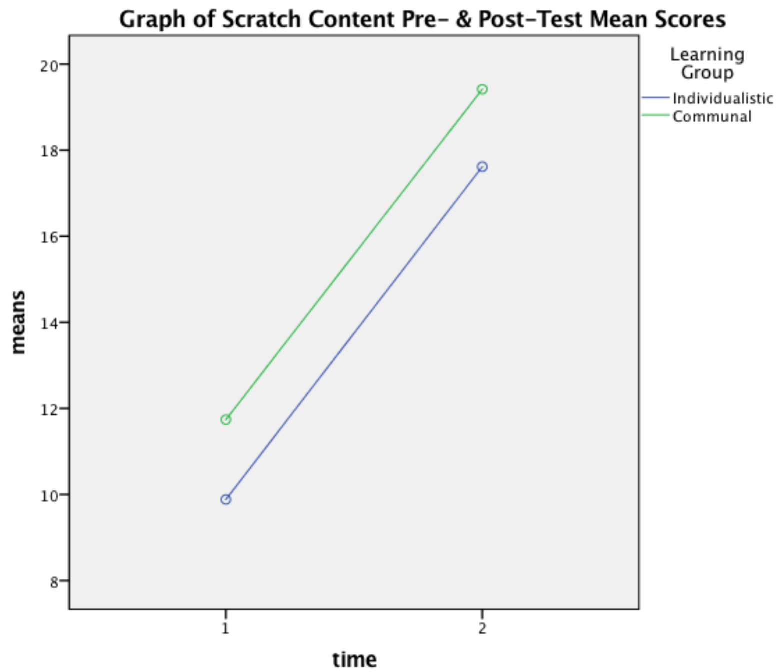


Figure 19b. 'Scratch' Content Knowledge – Change in Mean Scores by Learning Group.

Grade-level, pair-type, and gender comparisons: overall.

Table 14

Pre- and Post-Test Scores for Computational Thinking and Programming Content Knowledge

CT & Programming Skills			Communal Learning			Individual Learning		
(50 pts)	n	Test	M	SD	N	M	SD	N
All	42	Post	18.95	9.30	22	16.80	7.32	20
		Pre	12.23	5.96		9.95	3.53	
Gender:	17 Girls	Post	20.82	11.50	11	17.83	9.24	6
		Pre	11.45	5.77		8.33	3.08	
	25 Boys	Post	17.09	6.46	11	16.36	6.70	14
		Pre	13.00	6.33		12.00	2.94	
Pair Type:	4 Pairs of Girls	Post	19.25	11.99	8	-	-	-
		Pre	9.88	5.99		-	-	-
	4 Pair of Boys	Post	17.00	7.56	8	-	-	-
		Pre	14.63	6.65		-	-	-
	3 Pairs of Girl & Boy	Post	21.17	8.33	6	-	-	-
		Pre	12.17	4.36		-	-	-

Also of note, as it has also been discovered as a pattern for every CT and programming skill discussed next, is that the two girls pair-type and the one boy and one girl pair-type typically had the highest post-test scores and the largest learning gains of all pair types. Additionally, as can be seen in Figure 20a and 20b below that the one boy and one girl pair-type scored the highest. Another pattern which revealed itself in almost every test is that the two boys pair-type earned the lowest post-test mean score of all pair-types. These patterns are described separately for the performance in each of the six Scratch CT and programming skills below. There is an occasion or two when the post-test mean score of the two boys pair-type is not the lowest, but unfortunately, this occurrence is rare in this study.

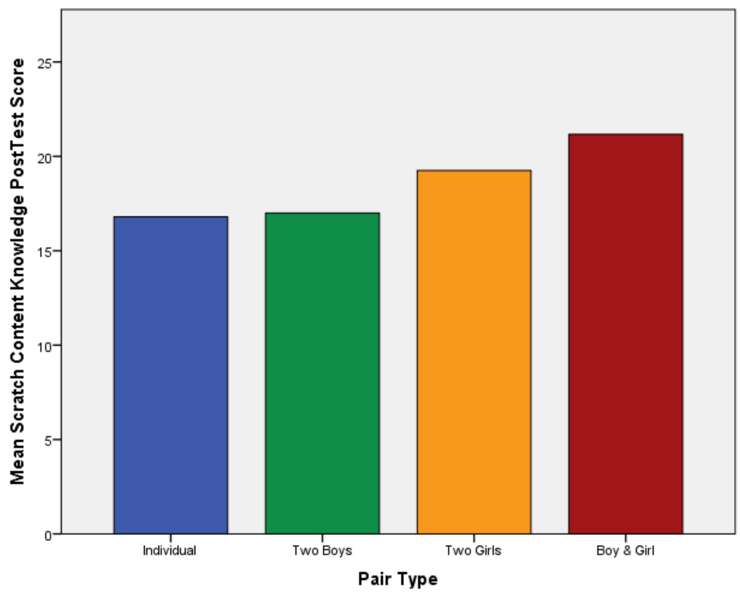


Figure 20a. 'Scratch' Content Knowledge Post-Test Mean Scores by Pair-Type.

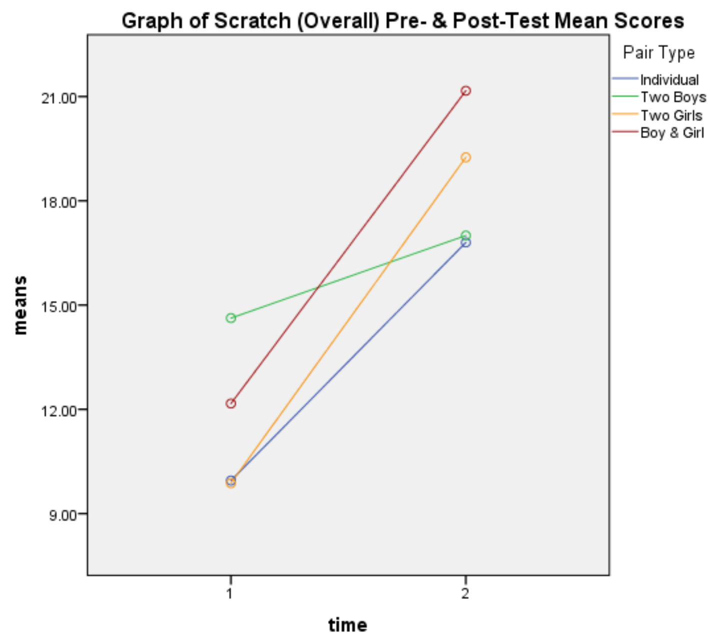


Figure 20b. 'Scratch' Content Knowledge Change in Mean Scores by Pair-Type.

An investigation of the post-test Scratch content knowledge by gender in the Individual Learning group revealed that the girl participants mean scores (M = 17.83) are also higher and they seem to have had more learning gains than the boy participants (M = 16.36) in the same learning group. This can be seen graphically in figure 20c below.

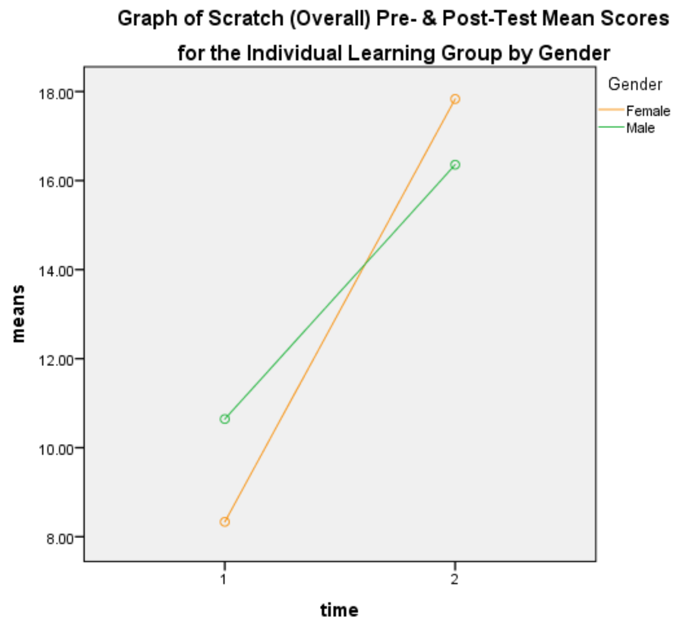


Figure 20c. 'Scratch' Content Knowledge Change in Mean Scores for Individual Group by Gender.

The next six sections describe and compare the performance of pre- and post-test mean scores of the six Scratch CT and programming skills (sequences, events, loops, conditionals, operators, and working with data).

Descriptive statistics for 'Scratch: Sequences' content knowledge scores. The CT and programming skill of 'sequences' relates to the participant's ability to identify a series of steps for a task. There were a total of 29.5 pages in the Scratch Booklet specifically about Sequences, starting with Unit Three, and seven questions about Sequences on the pre- and post-test. The amount of total possible points earned on these tests for this category was 14.

Table 15

Learning Group Scratch*: Sequences Content Knowledge Post-Test Mean Scores*

Learning Group	n	M	SD
Individualistic	20	4.80	2.79
Communal	22	5.86	3.68

Learning group comparison: Sequences. Pre- and post-test descriptive statistics revealed that although neither group scored very high in this category, the Communal Learning group had a higher mean post-test score (*M* = 5.86, *SD* = 3.68) than the Individualistic Learning group (*M* = 4.80, *SD* = 2.74). See table 15 above and figures 21a and 21b below for illustrations of these data.

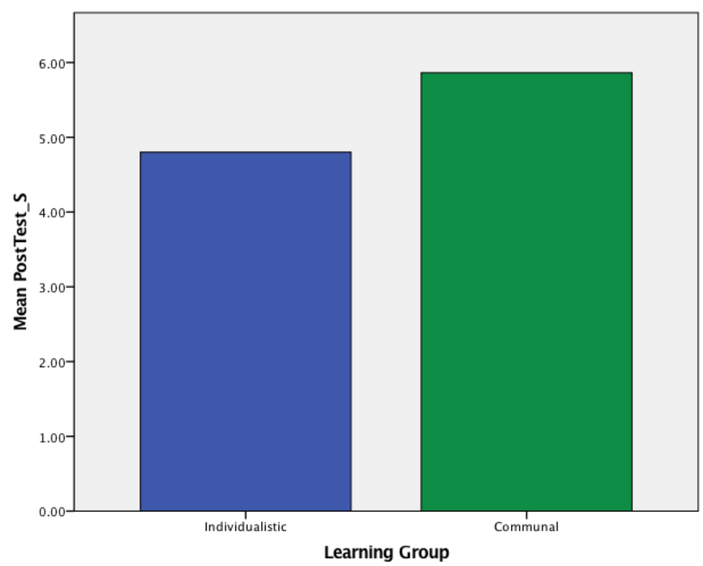


Figure 21a. 'Scratch: Sequence' Content Knowledge Post-Test Mean Scores by Learning Group.

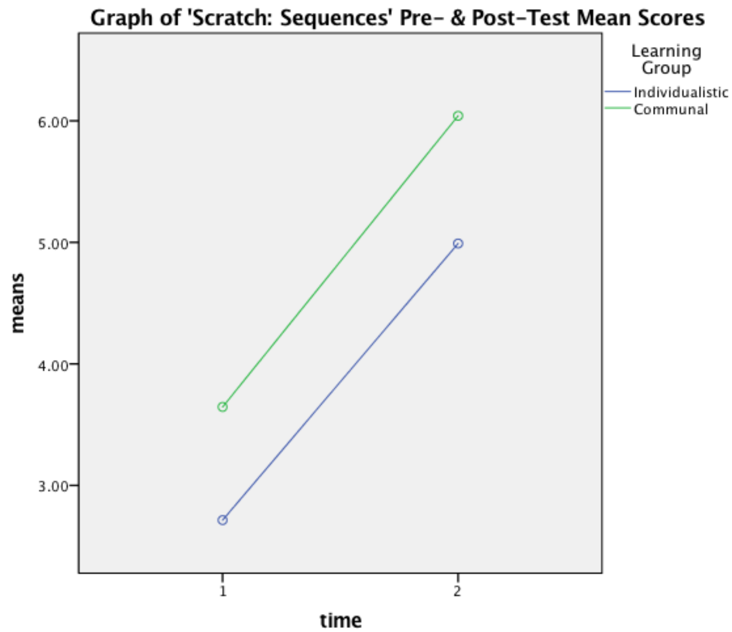


Figure 21b. 'Scratch: Sequences' Content Knowledge Change in Mean Pre- and Post-Test Scores by Learning Group.

Table 16

Pre- and Post-Test Scores Programming Content Knowledge of Sequences (14 pts)

CT & Programming Skills		Test	Communal Learning			Individual Learning		
"SEQUENCES"			M	SD	N	M	SD	N
All	n=42	Post	5.86	3.68	22	4.80	2.79	20
		Pre	3.50	2.39		2.75	1.25	
Gender:	17 Girls	Post	6.27	4.17	11	4.67	3.08	6
		Pre	3.09	2.47		2.33	1.38	
	25 Boys	Post	5.45	3.28	11	4.86	2.77	14
		Pre	3.91	2.34		2.93	1.21	
Pair Type:	4 Pairs of	Post	6.25	4.17	8	-	-	-
	Girls	Pre	2.50	2.56		-	-	-
	4 Pair of	Post	5.13	3.18	8	-	-	-
	Boys	Pre	4.00	2.67		-	-	-
	3 Pairs of	Post	6.33	4.13	6	-	-	-
	Girl & Boy	Pre	4.17	1.14		-	-	-

Pair-type, and gender comparisons: Sequences. Table 16 above shows detailed pre- and post-test descriptive mean score statistics regarding Scratch: Sequences content knowledge for all participants by grade level and communal learning pair-type. Again, post-test mean scores revealed that neither group scored very high in this category. By pair-type, the one boy & one girl pair-type obtained the highest post-test mean score (M = 6.33, SD = 4.13). This was followed by the two girl pair-type, then the two boy pair-type, then and the individual group. The girls in the communal learning group scored highest (M = 6.27, SD = 4.17) of all girls and boys in both communal and individual learning groups. Figures 22a and 22b below graphically depict post-test mean scores.

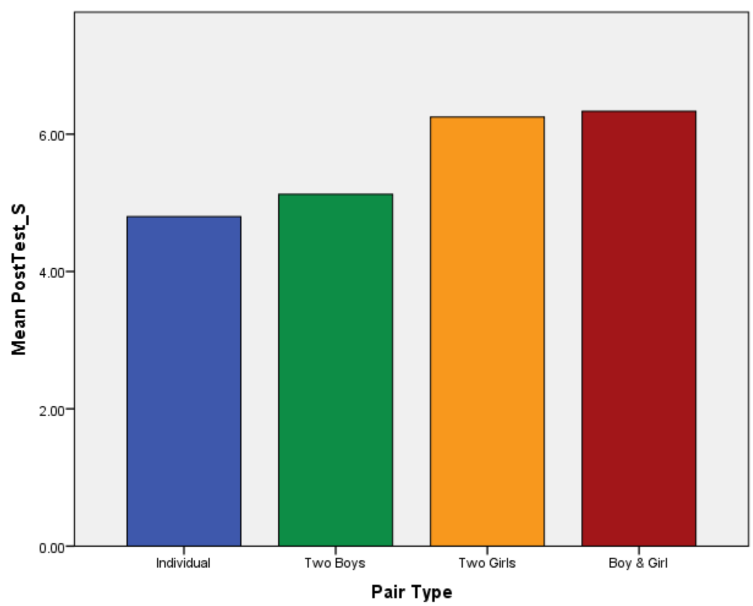


Figure 22a. 'Scratch: Sequences' Content Knowledge Post-Test Mean Scores by Pair-Type.

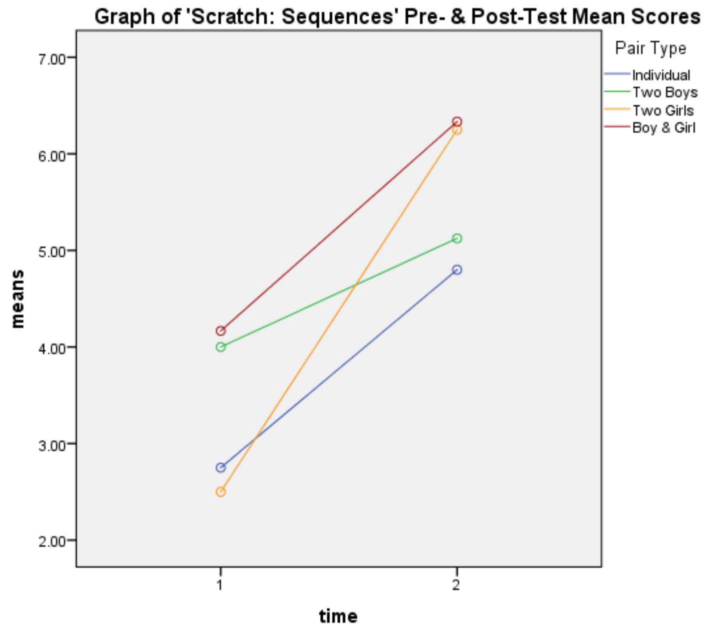


Figure 22b. 'Scratch: Sequences' Content Knowledge Change in Mean Pre- and Post-Test Scores by Pair-Type.

The Individual Learning group had the lowest mean post-test score, with the boys' mean score (*M* = 4.86, *SD* = 2.77) slightly better than the girls *(M* = 4.67, *SD* = 3.08). This is shown in figure 22c below. These data are the opposite when compared to the pair-type of two boys and two girls in the Communal Learning group.

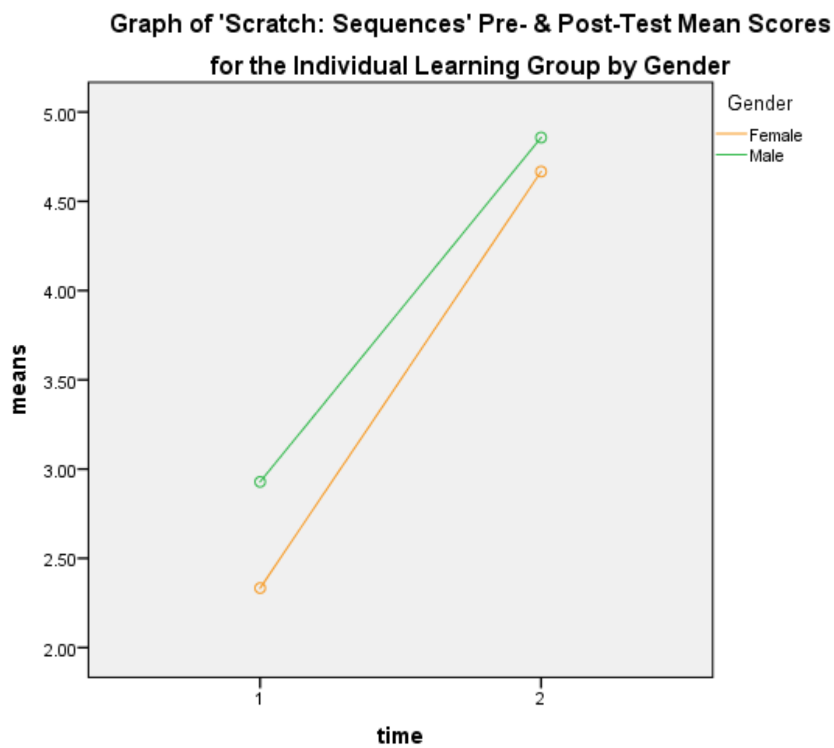


Figure 22c. 'Scratch*: Sequences*' Content Knowledge Change in Mean Pre- and Post-Test Scores for Individual Group by Gender.

Descriptive statistics for 'Scratch: Events' content knowledge scores. In Scratch, the CT and programming skills categorized as events is the ability to recognize when and program one thing causing another thing to happen (e.g. when the green flag is clicked, the cat sprite moves five steps). There were 9.5 pages about Events in the Scratch Booklet. This category started in Unit Seven, which most study participants did not reach. It had two questions on the pre- & post-tests. The total amount of points earned on the Scratch: Events portion of the Content Knowledge questionnaire is four. The overall mean post-test scores for both learning groups can be seen below in table 17.

Table 17

Learning Group Scratch*: Events Content Knowledge Post-Test Mean Scores*

Learning Group	n	M	SD
Individualistic	20	0.70	1.08
Communal	22	1.00	1.48

Learning group comparison: Events. Post-test descriptive statistics for Scratch: Events Content Knowledge revealed that the Communal Learning Group had mean post-test mean scores just a bit higher n the Scratch: Events Content Knowledge category (M=1.00, SD=1.48) than the Individual Learning Group (*M* = 0.70, *SD* = 1.08). Graphs of these data appear in figures 23a and 23b below.

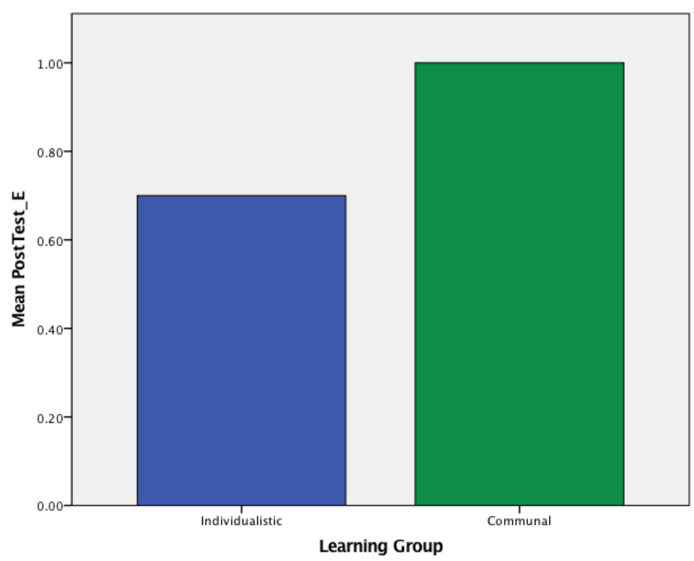


Figure 23a. 'Scratch: Events' Content Knowledge Post-Test Mean Scores by Learning Group.

Figure 23b. 'Scratch: Events' Content Knowledge Change in Mean Scores by Learning Group.

Table 18

Pre- and Post-Test Scores Programming Content Knowledge of Scratch*: Events*

CT & Programming Skills		Test	Communal Learning			Individual Learning		
"EVENTS"		(4 pts)	M	SD	N	M	SD	N
All	n=42	Post	1.00	1.48	22	0.70	1.08	20
		Pre	0.55	0.96		0.40	0.99	
Gender:	17 Girls	Post	1.64	1.75	11	0.33	0.82	6
		Pre	0.64	1.12		0.00	0.00	
	25 Boys	Post	0.36	0.81	11	0.86	1.17	14
		Pre	0.45	0.82		0.57	1.16	
Pair Type:	4 Pairs of Girls	Post	1.50	1.77	8	-	-	-
		Pre	0.25	0.71		-	-	-
	4 Pair of Boys	Post	0.25	1.63	8	-	-	-
		Pre	0.38	0.74		-	-	-
	3 Pairs of Girl & Boy	Post	1.33	1.63	6	-	-	-
		Pre	1.17	1.33		-	-	-

Pair-type, and gender comparisons: Events. Neither group had high post-test scores in the Scratch: Events category. Mean pre- and post-test score details are listed in table 18 above. The two girls pair-type (M = 1.50, SD = 1.77) had the highest mean score on the post-test and also showed the largest learning gain. The post-test mean score of two boys pair-type decreased, from (M = 0.38, SD = 0.77) to (M = 0.25, SD = 1.63). However, again, for gender overall, regardless of pair-type, the girls in the communal learning have a higher post-test means score (M = 1.64, SD = 1.75) than the boys in the communal learning (M = 0.36, SD = 0.81) and the girls (M = 0.33, SD = 0.82) and boys (M = 0.86, SD = 1.17) in the individual learning group. See figures 24a and 24b below, respectively.

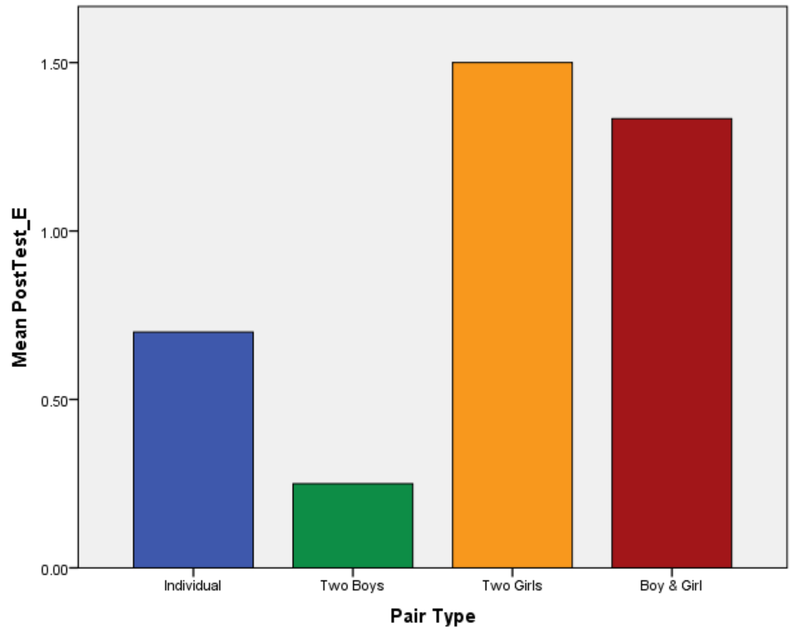


Figure 24a. ‘Scratch*:* Events’ Content Knowledge Post-Test Mean Scores by Pair-Type.

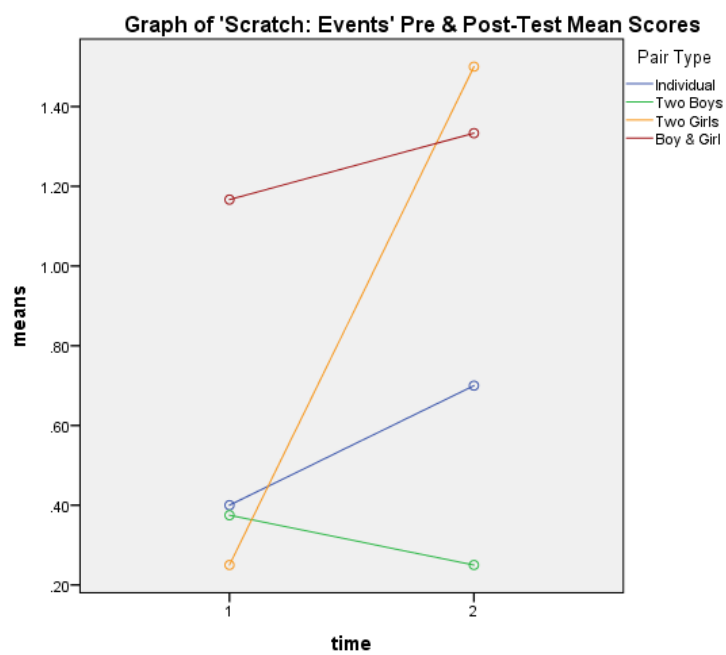


Figure 24b. ‘Scratch: Events’ Content Knowledge Change in Mean Pre- and Post-Test Scores by Pair-Type.

Regarding the Individual Learning group only, the boys (*M* = 0.86, *SD* = 1.17) had a slightly higher post-test mean score than the girls (*M* = 0.33, *SD* = 0.82). Figure 24c illustrates this below.

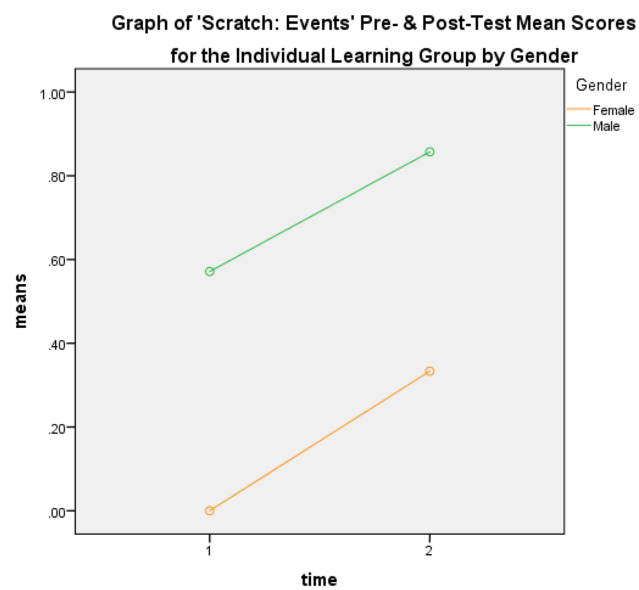


Figure 24c. 'Scratch*:* Events' Content Knowledge Change in Pre- and Post-Test Mean Scores for the Individual Learning Group by Gender.

Descriptive statistics for 'Scratch: Loops' content knowledge scores. The Scratch CT & programming skill of Loops indicates the ability to recognize and program the running the same sequence multiple times. There were 28 pages in the Scratch Booklet totally dedicated to Loops, starting in Unit Six. The total possible points to be earned for this category is 16, coming from eight questions.

Table 19

Learning Group Scratch*: Loops Content Knowledge Post-Test Mean Scores*

Learning Group	n	M	SD
Individualistic	20	7.40	3.79
Communal	22	7.55	3.85

Learning group comparison: Loops. Overall, study participants in both learning groups did fairly well on the Loops portion of the post-test, considering many of them did not reach Unit Six in the Scratch Booklet. This could be due to a familiarity with repeating music, also called loops. Nonetheless, descriptive statistics indicated the Communal Learning Group had mean scores just slightly higher on the Scratch: Loops Content Knowledge post-test (M = 7.55, SD = 3.85) than the Individualistic Learning Group (M = 7.40, SD = 3.79). See table 19 above and figures 25a and 5b below.

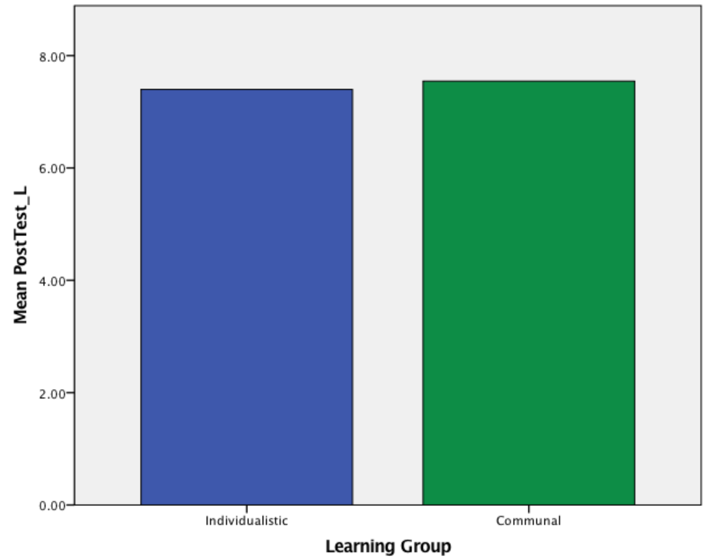


Figure 25a. 'Scratch*:* Loops' Content Knowledge Post-Test Mean Scores by Learning Group.

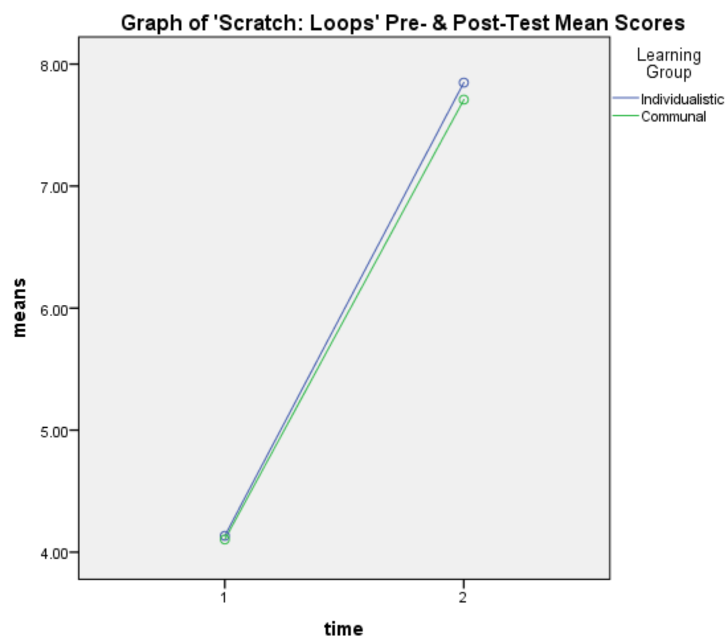


Figure 25b. 'Scratch*:* Loops' Content Knowledge Change in Mean Scores by Learning Group.

Table 20

Pre- and Post-Test Scores Programming Content Knowledge of Scratch*: Loops* (16 pts)

CT & Programming Skills LOOPS		Test	Communal Learning M	SD	N	Individual Learning M	SD	N
All	n=42	Post	7.55	3.85	22	7.40	3.79	20
		Pre	4.41	2.42		4.20	2.67	
Gender:	17 Girls	Post	8.36	4.88	11	8.17	4.58	6
		Pre	4.27	1.85		3.17	1.85	
	25 Boys	Post	6.73	2.41	11	7.07	3.54	14
		Pre	4.55	2.98		4.64	2.85	
Pair Type:	4 Pairs of Girls	Post	7.50	5.10	8	-	-	-
		Pre	4.00	1.60		-	-	-
	4 Pair of Boys	Post	7.00	2.83	8	-	-	-
		Pre	5.63	2.77		-	-	-
	3 Pairs of Girl & Boy	Post	8.33	3.67	6	-	-	-
		Pre	3.33	2.50		-	-	-

Pair-type, and gender comparisons: Loops. Table 20 above and figures 26a, 26b, & 26c below graphically show post-test mean score details by pair-type and gender. Again, the girl and boy pair-type scored the highest (*M* = 8.33, *SD* = 3.67). The girls in the individual learning group closely followed (*M* = 8.17, *SD* = 4.58), while the two girl pair-type followed with the third highest score (*M* = 7.50, *SD* = 5.10). They were followed by the boys in the individual learning group and the two boy pair-type. This ordering can be seen in figures 26a and 26b. By gender overall, the communal learning group girls had the highest mean post-test score (*M* = 8.36, *SD* = 4.88) followed by the individual learning group girls (*M* = 8.17, *SD* = 4.58), while the communal learning group boys had the lowest mean post-test score (*M* = 6.73, *SD* = 2.41). This occurred in the Individual Learning group as well, where the girls (*M* = 8.17, *SD* = 4.58) performed better than the boys (*M* = 7.07, *SD* = 3.54). See the graphical representation of this in figure 26c below.

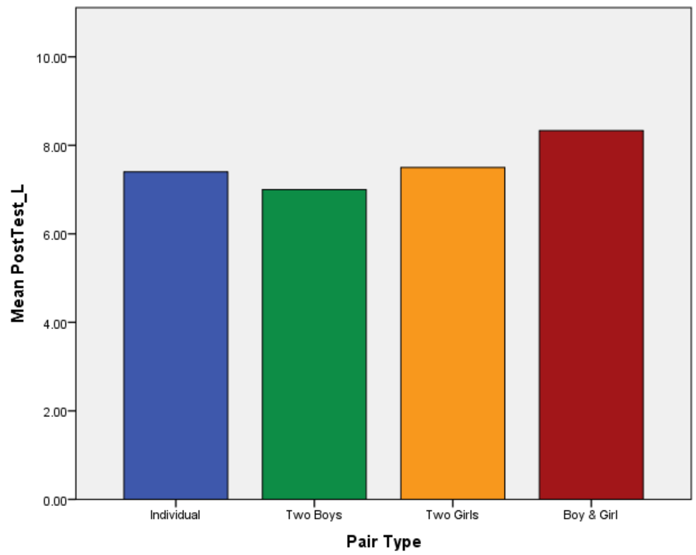


Figure 26a. ‘Scratch: Loops’ Content Knowledge Post-Test Mean Scores by Pair-Type.

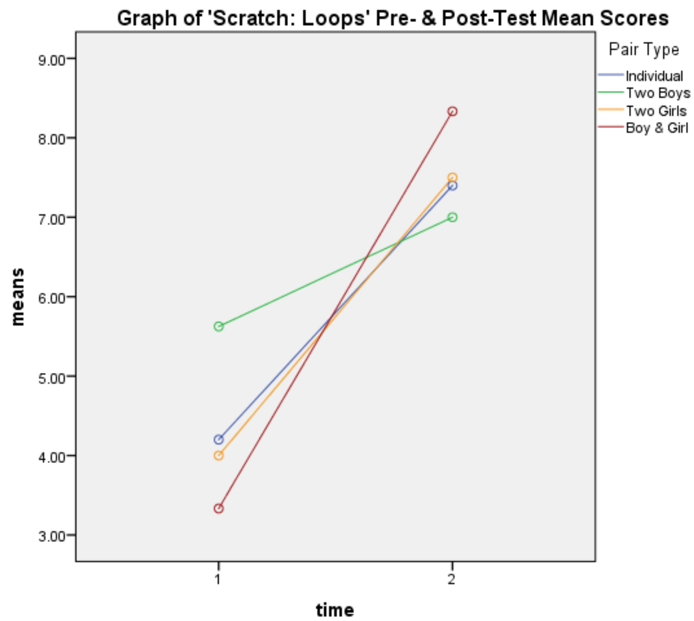


Figure 26b. 'Scratch: Loops' Content Knowledge Change in Mean Scores by Pair-Type.

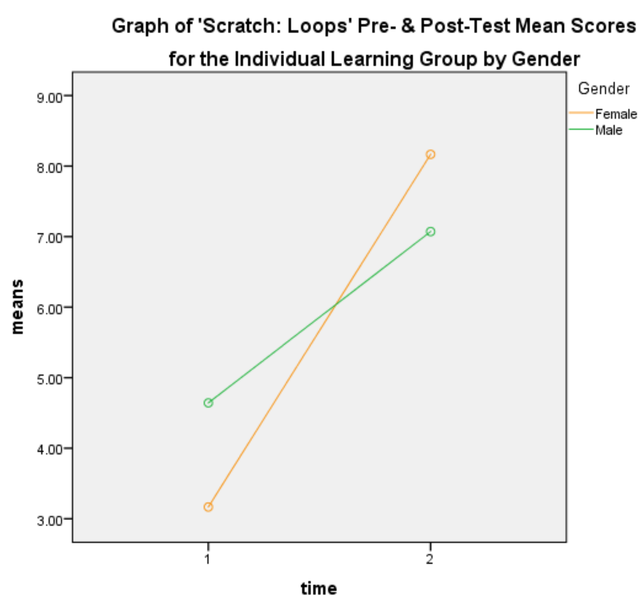


Figure 26c. 'Scratch*:* Loops' Content Knowledge Change in Mean for the Individual Learning Group by Gender.

Descriptive statistics for 'Scratch: Conditionals' content knowledge scores. The Scratch CT and programming skill category of conditionals basically is the ability to recognize and code/program the notion of making decisions based on conditions. There were 66 pages focused on conditionals and how to use them, starting in Unit Four, with seven questions on the Scratch Content Knowledge pre- and post-tests. The total number of possible points earned for this category is 14.

Table 21

Learning Group Scratch*: Conditionals Content Knowledge Post-Test Mean Scores*

Learning Group	n	M	SD
Individualistic	20	3.20	2.38
Communal	22	3.91	2.86

Learning group comparison: Conditionals. Again, neither group reached particularly high post-test mean scores in this category. It is worth noting that, on average, study participants only completed up to and including a portion of Unit Four, the unit introducing conditionals in the Scratch Booklet. This could explain the low mean post-test scores. Post-test mean statistics indicated the Communal Learning Group scored just a bit higher on the Scratch: Conditionals Content Knowledge post-test (M = 3.91, SD = 2.86) than the Individual Learning Group (M = 3.20, SD = 2.38). Table 21 above and Figures 27a and 27b below show mean post-test score details.

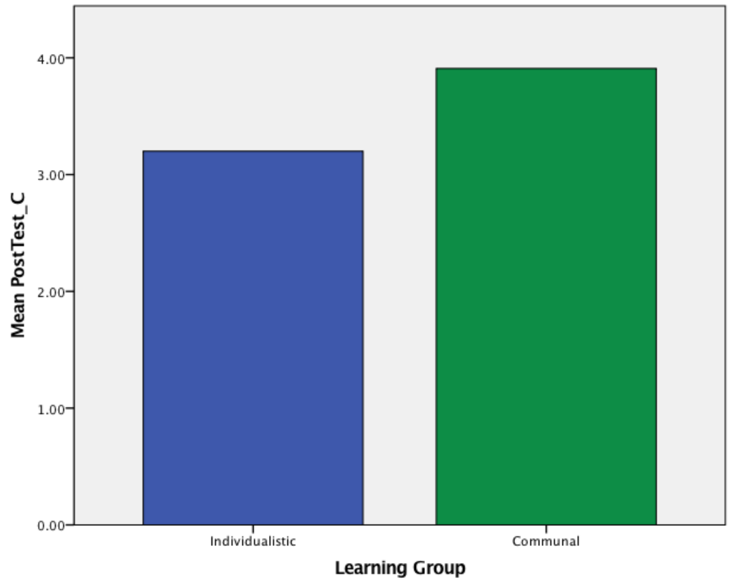


Figure 27a. 'Scratch*:* Conditionals' Content Knowledge Post-Test Mean Scores by Learning Group.

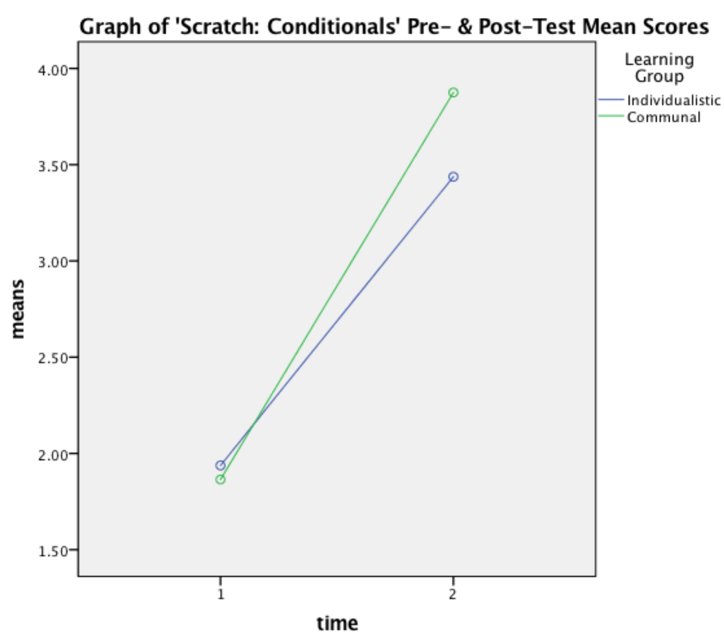


Figure 27b. 'Scratch*:* Conditionals' Content Knowledge Change in Mean Scores by Learning Group.

Table 22

Pre- and Post-Test Scores Programming Content Knowledge of Scratch*: Conditionals*

CT & Programming Skills (14 pts)		Test	Communal Learning			Individual Learning		
"CONDITIONALS"			M	SD	n	M	SD	n
All	n=42	Post	3.91	2.86	22	3.20	2.38	20
		Pre	2.05	2.10		1.90	1.45	
Gender:	17 Girls	Post	3.82	2.60	11	3.67	3.14	6
		Pre	1.91	2.12		1.67	1.03	
	25 Boys	Post	4.00	3.23	11	3.00	2.08	14
		Pre	2.18	2.18		2.00	1.62	
Pair-Type:	4 Pairs of Girls	Post	3.50	2.73	8	-	-	-
		Pre	1.13	1.64		-	-	-
	4 Pair of Boys	Post	4.13	3.68	8	-	-	-
		Pre	2.50	2.45		-	-	-
	3 Boy & Girl Pairs	Post	4.17	2.14	6	-	-	-
		Pre	2.67	2.07		-	-	-

Pair-type, and gender comparisons: Conditionals. Regarding pair-type overall, regardless of grade-level, the one boy & one girl pair-type post-test mean score (M = 4.17, SD = 2.14) was the highest overall, this time with the two boys pair-type (M = 4.13, SD = 3.68) close behind. The two girls pair-type mean post-test score (M = 3.50, SD = 2.73) was the lowest. Figures 28a and 28b below illustrate these data. The two boys pair-type post-test means score (M = 4.13, SD = 3.68) was the highest and the one boy & one girl pair-type (M = 3.67, SD = 2.08) followed, with the two girls pair-type post-test mean score (M = 3.00, SD = 2.00) at the lowest. This is the first time this occurred. It could be concluded that the participants in the two boys pair-type were no longer hindered by challenging behavior and could therefore perform at optimal levels learning and using this category of program blocks within Scratch. Regarding gender, the boys in the communal learning group overall had the highest post-test score in this category (M = 4.00, SD = 3.23). All other scores by gender were fairly close, this includes the girls in the communal learning group as well as the boys and girls in the individual learning group. These data are represented below in figures 28a, 28b, and 28c.

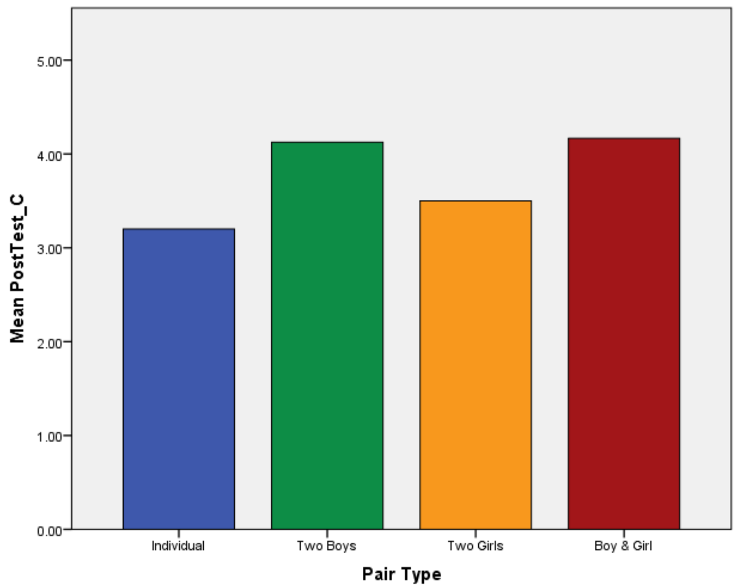


Figure 28a. ‘Scratch: Conditionals’ Content Knowledge Post-Test Mean Scores by Pair-Type.

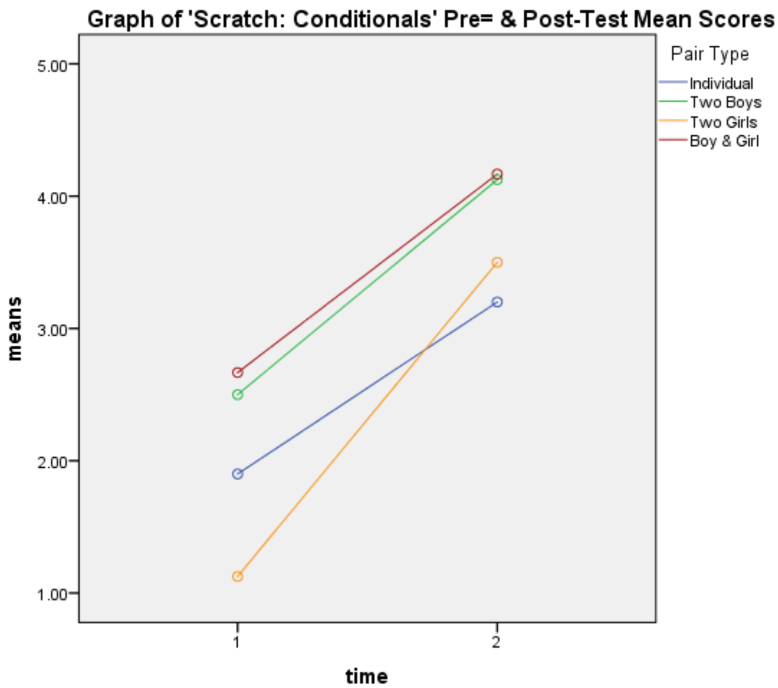


Figure 28b. ‘Scratch*:* Conditionals’ Content Knowledge Change in Mean Scores by Pair Type.

Figure 28c. ‘Scratch*:* Conditionals’ Content Knowledge Change in Mean Scores of the Individual Learning Group by Gender.

Descriptive statistics for 'Scratch: Operators' content knowledge scores. The Operators Scratch CT and programming skill category refers to support for mathematical calculations and logical expressions. In the Scratch Booklet, material on how to program operators began in Unit 10, with 29 pages dedicated to its explanation. The total number of possible points earned on the post-test for this category was six. Not many study participants in either learning group reported reaching Unit 10. This is a possible contribution to the low mean scores in this category.

Table 23

Learning Group Scratch*: Operators Content Knowledge Post-Test Mean Scores*

Learning Group	N	M	SD
Individualistic	20	1.55	1.468
Communal	22	2.27	2.051

Learning group comparison: Operators. Post-test descriptive statistics indicated the Communal Learning Group scored higher on the Scratch: Operators Content Knowledge post-test (M = 2.27, SD = 2.051) than the Individualistic Learning Group (M = 1.55, SD = 1.47). These data are represented graphically in table 23 above and figures 29a and 29b below.

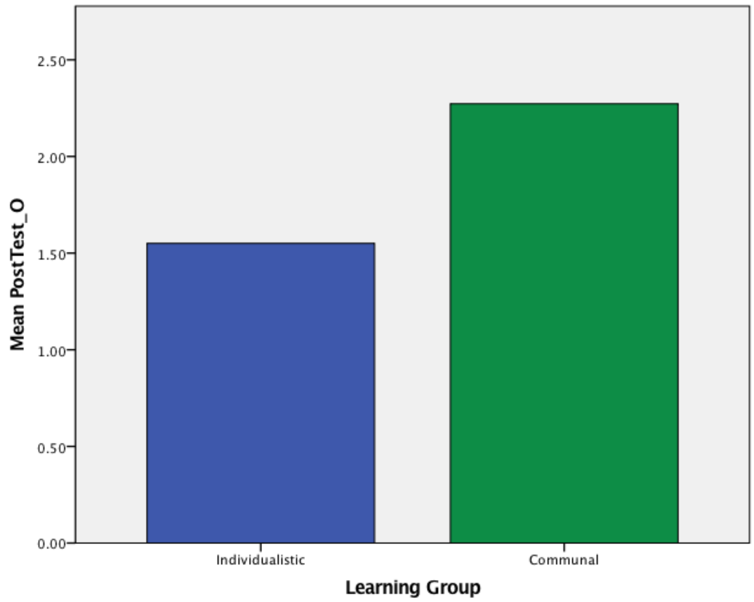


Figure 29a. 'Scratch*:* Operators' Content Knowledge Post-Test Mean Scores by Learning Group.

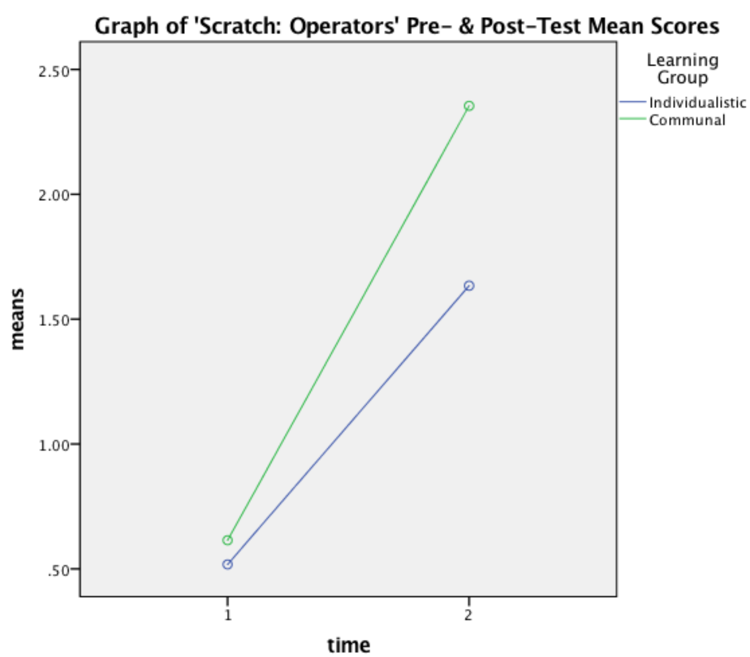


Figure 29b. 'Scratch*:* Operators' Content Knowledge Change in Mean Scores by Learning Group.

Table 24

Pre- and Post-Test Scores Programming Content Knowledge of Scratch*: Operators*

CT & Programming Skills "OPERATORS"		Test (6 pts)	Communal Learning M	SD	n	Individual Learning M	SD	n
All	n=42	Post	2.27	2.051	22	1.55	1.468	20
		Pre	0.64	0.953		0.65	0.745	
Gender:	17 Girls	Post	2.36	2.111	11	1.33	1.506	6
		Pre	0.55	0.820		0.33	0.516	
	25 Boys	Post	2.18	2.089	11	1.64	1.499	14
		Pre	0.73	1.104		0.79	0.802	
Pair Type:	4 Pairs of	Post	2.25	2.121	8	-	-	-
	Girls	Pre	0.38	0.744		-	-	-
	4 Pair of	Post	2.13	2.100	8	-	-	-
	Boys	Pre	0.88	1.246		-	-	-
	3 Pairs of	Post	2.50	2.258	6	-	-	-
	Girl & Boy	Pre	0.67	0.817		-	-	-

Pair-type, gender comparisons: Operators. Again, comparing all pair types, the one boy & one girl pair-type post-test mean score (M = 2.50, SD = 2.258) was the highest in the Scratch: Operators category, this is followed by the two girls pair-type (M = 2.25, SD = 2.121), then the two boys pair-type (M = 2.13, SD = 2.100), followed last by the individual learning group (M = 1.55, SD = 1.468). Within the Individual Learning group, the boys post-test mean score (M = 1.64, SD = 1.499) was higher than the girls (M = 1.33, SD = 1.506). See table 23 above for more mean score details and figures 30a, 30b, and 30c below, which visually represent these data.

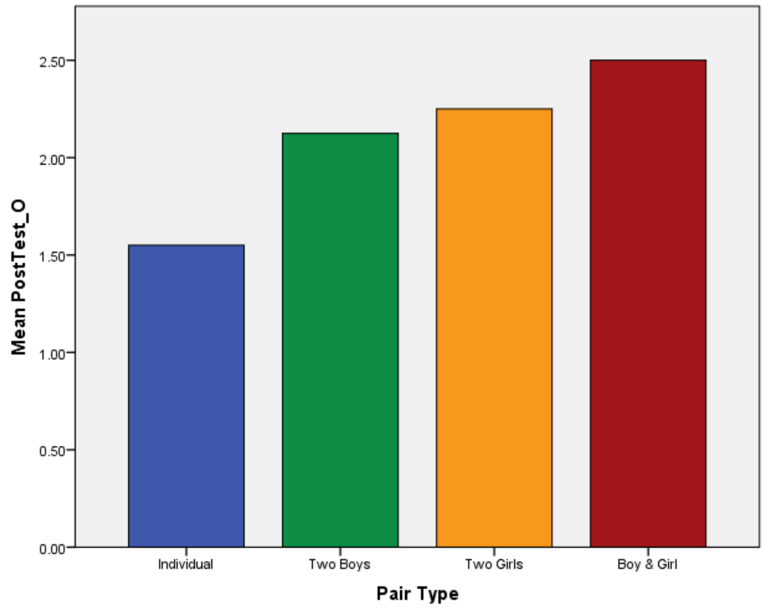


Figure 30a. 'Scratch: Operators' Content Knowledge Post-Test Mean Scores by Pair-Type

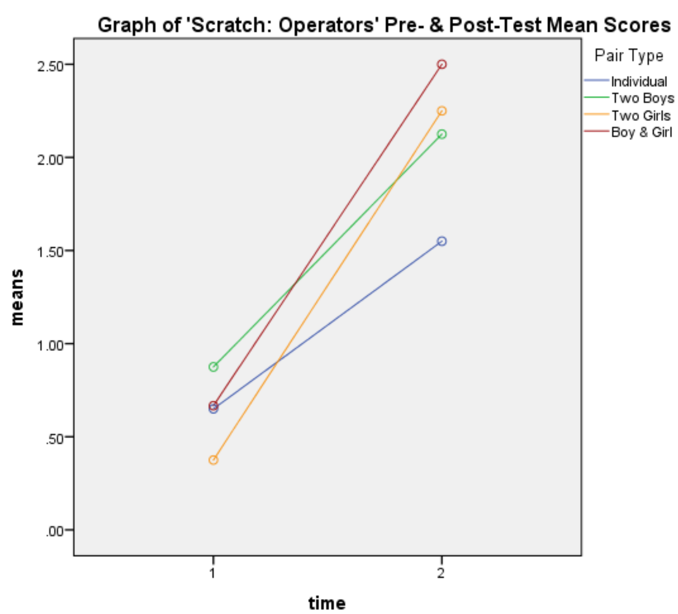


Figure 30b. 'Scratch*:* Operators' Content Knowledge Change in Mean Scores by Pair-Type.

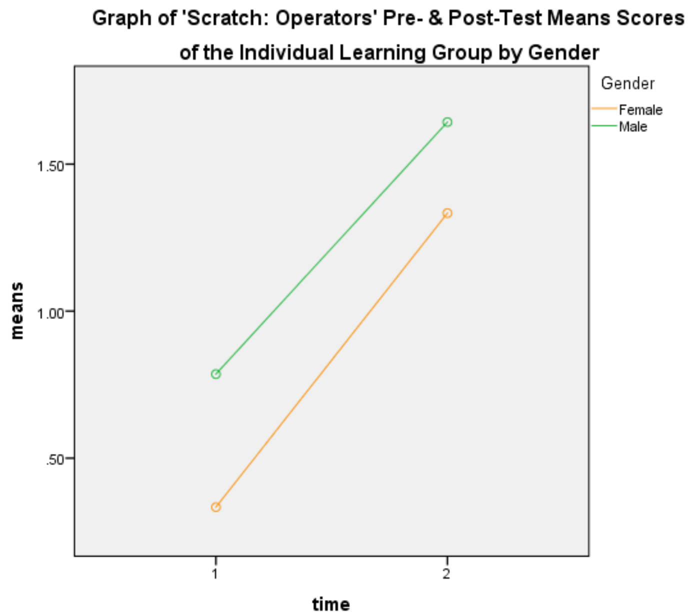


Figure 30c. 'Scratch: Operators' Content Knowledge Change in Mean Scores by Pair Type of Individual Learning Group by Gender.

Descriptive statistics for 'Scratch: Working with Data' content knowledge scores. The Working with Data category in Scratch means the ability to recognize and implement the storing, retrieving, and updating values. This notion was featured throughout the Scratch Booklet. It had 52 pages specifically dedicated to it, which began in Unit Eight. The total amount of possible points for this category of CT and programming skill is 18 on the pre- and post-tests, the highest amount of points possible for any one category. Again, not many study participants in either learning group reported reaching Unit Eight. This possibly contributes to the low mean scores in this category.

Table 25

Learning Group Scratch*: Working with Data Post-Test Mean Scores*

Learning Group	N	M	SD
Individualistic	20	4.20	2.44
Communal	22	6.09	4.09

Learning group comparison: Working with data. Neither group performed well in this category. However, post-test mean score descriptive statistics indicated the Communal Learning Group scored higher on the Scratch: Working with Data Content Knowledge post-test (M = 6.09, SD = 4.09) than the Individualistic Learning Group (M = 4.20, SD = 2.441). These data are in table 25 above and in figures 31a and 31b below.

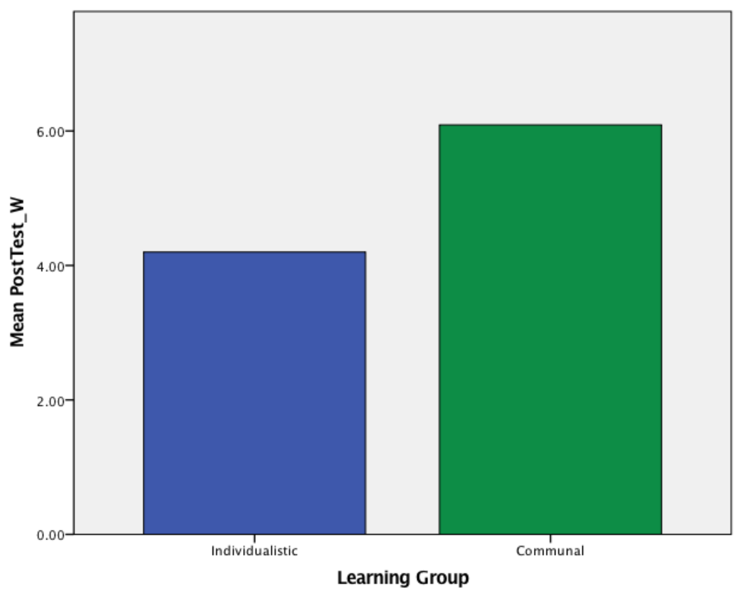


Figure 31a. 'Scratch*:* Working with Data' Content Knowledge Post-Test Mean Scores by Learning Group.

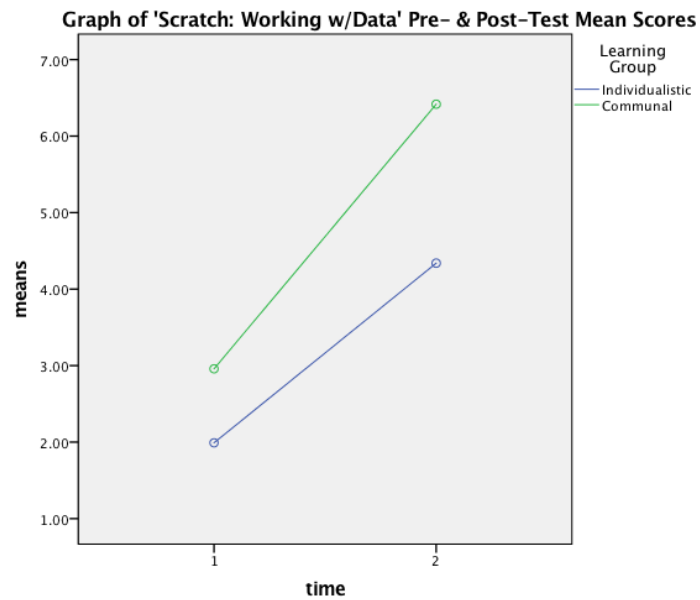


Figure 31b. 'Scratch*:* Working with Data' Content Knowledge Change in Mean Scores by Learning Group.

Table 26

Pre- and Post-Test Scores Programming Content Knowledge of Scratch*: Working with Data*

CT & Programming Skills			Communal Learning			Individual Learning		
"WORKING WITH DATA"		Test	M	SD	n	M	SD	n
All	n=42	Post	6.09	4.09	22	4.20	2.44	20
		Pre	3.18	2.1		2.20	1.11	
Gender:	17 Girls	Post	6.55	4.61	11	3.67	2.66	6
		Pre	2.82	2.14		1.83	0.98	
	25 Boys	Post	5.64	3.67	11	4.43	2.41	14
		Pre	3.55	2.12		2.36	1.15	
Pair Type:	4 Pairs of Girls	Post	6.00	4.81	8	-	-	-
		Pre	2.75	2.32		-	-	-
	4 Pair of Boys	Post	2.25	3.73	8	-	-	-
		Pre	4.13	2.17		-	-	-
	3 Pairs of Girl & Boy	Post	7.33	3.93	6	-	-	-
		Pre	2.50	1.52		-	-	-

Pair-type, and gender comparisons: Working with Data. Table 26 above shows pre- and post-test mean score details for all and pair-types in both learning groups. When looking at pair-type comparisons, the elementary school two girls pair-type post-test mean score (M = 8.25, SD = 5.50) was the highest of all elementary school participants and the one boy & one girl pair-type in the middle school communal learning group scored highest of all (M = 9.00, SD = 2.83). These pair-type having the highest mean scores follows the communal learning pair-type pattern of performance data described above. Figures 29a and 29b below show the mean scores by pair-type, as well as their change in scores. By gender, the girls in the communal learning group post-test mean score (M = 6.55, SD = 4.61) was the highest of all. This score was followed by the communal learning group boys (M = 5.64, SD = 3.6), the individual learning group boys (M = 4.43, SD = 2.4), with the individual learning group girls post-test mean score (M = 3.67, SD = 2.66) the lowest. The one boy & one girl pair-type performed highest, as described above with the largest change in mean score (i.e. learning gain), while the two boys pair-type scored the lowest of all pair-types. The Individual Learning group scored the lowest. See figures 32a, 32b, and 32c.

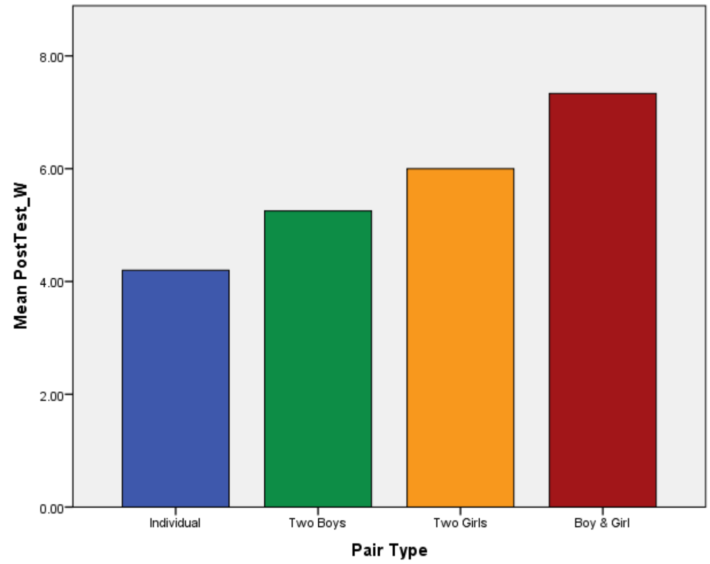


Figure 32a. 'Scratch*:* Working with Data' Content Knowledge Post-Test Mean Scores by Pair-Type.

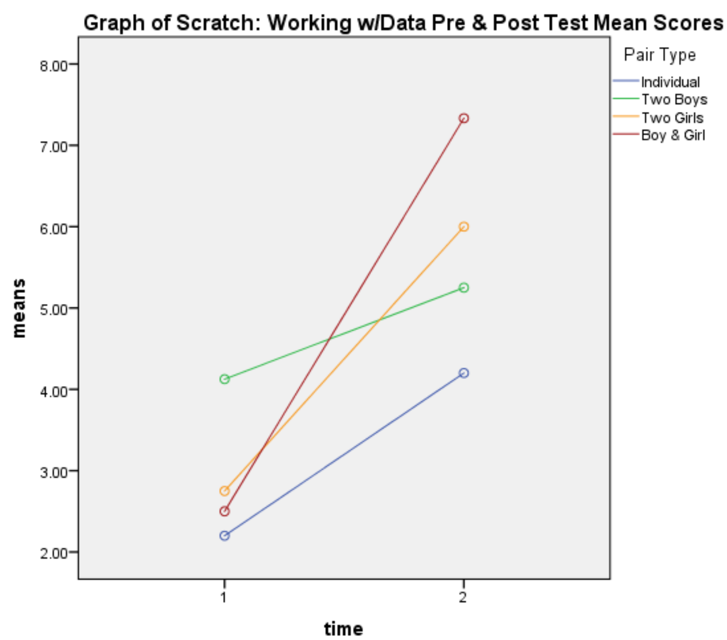


Figure 32b. 'Scratch*:* 'Working with Data' Content Knowledge Change in Mean Scores by Pair-Type.

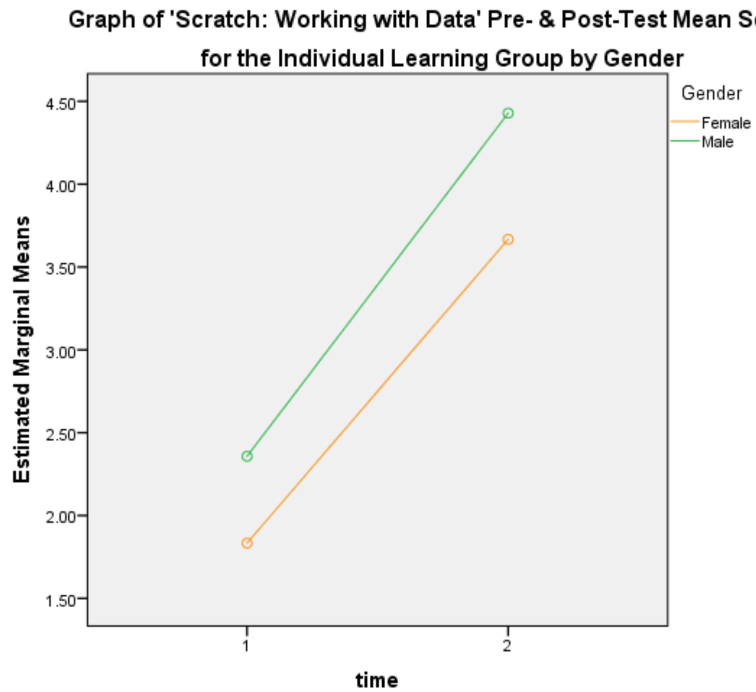


Figure 32c. 'Scratch*:* Working with Data' Content Knowledge Change in Mean Scores of Individual Learning Group by Gender.

Inferential statistics of overall Scratch scores by learning group. There were 22 participants in the Communal Learning group and 20 in the Individual Learning group. An independent-samples t-test was run to determine if there were differences in post-test mean scores between learning groups. All assumptions were met. There were two outliers in the post-test Scratch Content Knowledge Questionnaire scores within the Individual Learning group and no outliers in the post-test scores in the Communal Learning group, as assessed by inspection of resulting boxplots. The two outliners were kept in the analysis since the skewness and kurtosis assumptions were met and because the sample size for the Individual Learning group is small. Pre-test and post-test mean scores were normally distributed for both learning groups, as assessed by Shapiro-Wilk's test ($p > .05$). There was homogeneity of variances for post-test scores for both the Communal Learning and Individual Learning groups, as assessed by Levene's test of equality of variances ($p = .463$). The post-test man Scratch Content Knowledge score for the Communal Learning group ($M = 18.95$, $SD = 9.297$) was higher than that of the Individual Learning group ($M = 16.80$, $SD = 7.324$). However, there was no statistically significant difference between the two post-test mean scores, $t(41) = -0.828$, $p =.412$. Therefore, it can be concluded that putting students in groups does not change their cognition towards Scratch. A post-hoc power analysis was run using GPower and it revealed a power value of 0.39. This is likely a result of a small sample.

Summary of Performance on Scratch Computational Thinking and Programming

Overall summary. The number of participants in this mixed method study was 42, with 22 in the Communal Learning group and 20 in the Individual Learning group. With 50 total points allowed on the Scratch computational thinking and programming skills content knowledge pre- and post-test, the mean pre- and post-test scores were high for neither learning group.

Pair-type and gender performance summary. Here pair-types include two girls, two boys, one boy & one girl, and individual. The one boy & one girl and two girls pair-types performed the highest in six of the seven post-test performance categories analyzed (Scratch overall, sequences, loops, conditionals, operators, and working with data. The one time they did not (events), the two girls pair-type performed the best, with the one boy & one girl pair-type performing second best. The two boys pair type usually had the lowest mean scores, but not always. There was no particular pattern of performance in the Individual Learning group. However, the boys did perform higher than the girls in four of the seven categories. These were sequences, events, operators and working with data. The degree of higher performance, i.e. post-test mean scores, varied throughout. Moreover, the Individual Learning group had the lowest means scores in all categories except two, events and loops. When this occurred, the Individual Learning group usually had the second lowest mean score.

Grade-level performance summary. Overall, the one boy & one girl pair-type consistently performed best of all pair-types in both grade levels. When this did not occur, the pattern of the two girls pair-type performing the best consistently held true, while the two boys pair-type almost always performed the lowest. Qualitative data analysis revealed, however, that the boys in the middle school communal learning group exhibited the greatest amount of and most profound behavioral issues, consistent with have what education researchers call a cool pose and expressing hegemonic masculinity – to the point of having a physical altercation (fight).

Mean difference summary. Using SPSS, a independent samples t-test was performed on the pre- and post-test scores and determined that there was no statistically significance difference between the Communal Learning (M = 18.95, SD = 9.297) and Individual Learning group (M = 16.80, SD = 7.324) regarding Scratch overall content knowledge. Therefore, it can be concluded that group assignment does not impact cognition towards Scratch. The power value is 0.39.

Given the above variety of strategies and outcomes regarding learning and using Scratch computational thinking and programming skills and that assigning potential students to either work in groups or work alone, does not have an impact on the cognition when learning Scratch (i.e. students can work in group or by themselves their cognition stays the same), however, learning group assignments may change a student's attitude towards working in a group or individually. This is explored in the second research question in this study – what impact does this experience have on participants' learning context preference. The results of this exploration are described in the next section.

RQ2: Learning Context Preference

Although participants were randomly assigned to one of the two target learning groups, the modified Learning Context Questionnaire was used to measure their preferences for working cooperatively and individualistically. This questionnaire was taken before and after camp to determine a baseline for learning preferences for all participants as well as to observe any potential changes to these preferences after having experienced camp working in the Communal Learning group or Individual Learning group. The highest possible score for both learning preferences is four. Thus, a midpoint (i.e. cut-off) score of 2.5 or more endorses a measured learning preference. A score lower than 2.5 does not endorse a measured learning preference.

RQ2: Is there a change in the learning context preference of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?

Cooperative learning preference affirmed. As can be seen in Table 12 below, before camp started, the pre-test mean scores for a Cooperative learning context preference suggests that all 42 participants endorse cooperative learning environments (M = 3.23, SD = 0.649) and the pre-test mean scores for an Individualistic learning context preference show that all participants dislike individual learning environments (M = 2.04, SD = 0.589).

Overall, these same preferences remained after camp, as can also be seen in Table 12, where the post-test mean scores for Cooperative learning contexts suggest participants endorse cooperative learning environments (M = 3.34, SD = 0.654) and a post-test mean score for Individualist learning contexts suggest participants do not endorse individualist learning environments (M = 2.01, SD = 0.75).

At first glance, these pre- and post-test scores provide an initial affirmation of the theory and research regarding African American students innate preference for learning contexts that align with their culture, in this case communalism/communal learning, as described in the literature review above.

Table 27

Means and Standard Deviations for Learning Context Preferences (n = 42)

Learning Context Preference (LCQ-m)	Before (Pre-)		After (Post-)	
(1-4)	M	SD	M	SD
Cooperative	3.23	0.65	3.34	0.65
Individualistic	2.04	0.59	2.01	0.76

Cronbach's alpha reliability. The Cron's Alpha measure of internal consistency or reliability was calculated for the LCQ-m scale before and after camp. These measurements indicated high levels of internal consistency for Cooperative Learning Context Preference BEFORE and AFTER, 0.88 and 0.91, respectively, and Individualistic Learning Context Preference, BEFORE and AFTER, 0.756 and 0.789, respectively. These statistics are displayed in Table 28 below.

Table 28

Cronbach's Alpha Reliability for LCQ-m (BEFORE and AFTER Camp)

Learning Context Preference	Before	After
LCQ-m Cooperative	0.881	0.912
LCQ-m Individualistic	0.759	0.789

The next two sections describe descriptive and inferential statistics for study data regarding the Cooperative Learning Context Preference scale.

Descriptive Statistics for Cooperative Learning Context Preference

The resulting pre- and post-test mean Cooperative Learning Context Preference scores appear in Table 29 below. It illustrates that all participants in both learning groups saw an increase in preferences for cooperative learning, with a slight higher increase among those in the individual learning group, who had a resulting mean score of 3.50 out of 4.0, *SD* = 0.47. This is considerably more than the resulting cooperative learning preference means score by the communal learning group, which was a mean score of 3.18, *SD* = 0.767. The boys in both learning groups started and ended with having a higher preference for working cooperatively than the girls in both learning groups. Looking more closely at gender within each learning group, everyone experienced an increase in cooperative learning preference except the girls in the individual learning group. They experienced a decrease in mean score for cooperative learning preference, from 3.33 to 3.26. Of the pair-types in the communal learning group, the elementary school two boys pair type was the only pair-type to experience a decrease in the preference to learning cooperatively. They started with strong mean score of 3.92. However, the mean score of this pair type dropped to 3.71 after camp. The elementary and middle school two boys pair-type all experienced behavioral challenges at various times throughout the study, which caused distractions and slowed progress at one point or another. This could be a potential reason for the drop.

Table 29

Pre- and Post-Test Cooperative Learning Context Preference Scores

Learning Context Preference		Test	Communal Learning			Individual Learning		
"COOPERATIVE"		(1-4)	M	SD	N	M	SD	N
All	42	Post	3.18	0.77	22	3.50	0.47	20
		Pre	3.11	0.76		3.37	0.48	
Gender:	17 Girls	Post	2.97	0.95	11	3.26	0.55	6
		Pre	2.86	0.85		3.33	0.58	
	25 Boys	Post	3.40	0.50	11	3.60	0.41	14
		Pre	3.36	0.60		3.38	0.46	
Pair Type:	4 Pairs of Girls	Post	3.16	0.77	8	-	-	-
		Pre	3.04	0.70		-	-	-
	4 Pair of Boys	Post	3.39	0.56	8	-	-	-
		Pre	3.44	0.67		-	-	-
	3 Pairs of Girl & Boy	Post	2.95	1.04		-	-	-
		Pre	2.78	0.90	6	-	-	-

Inferential Tests for Statistical Significance of Cooperative Learning Context Preference

Paired-samples t-tests were run on the differences between pre- and post-test mean Cooperative Learning Context Preference scores to determine statistical significance. Three outliers were present in the difference scores for pre- and post-test mean Cooperative Learning Context Preference score, as assessed by inspection of resulting boxplots. However, these outliers were not extreme and were kept in the analysis. All other assumptions were met. Differences between the pre- and post-test mean Cooperative Learning Context Preference scores were normally distributed, as assessed by Shapiro-Wilk's test ($p > .05$). There was homogeneity of variances for the difference in pre- and post-test mean Cooperative Learning Context Preference scores, as assessed by Levene's test of equality of variances (p = .215).

Statistical significance of cooperative learning context preference. This camp experience had no effect on Cooperative Learning Context Preference according to pre- and post-test scores. The result of the paired-samples t-test showed no statistically significant mean difference, $t(41) = 1.70$, $p = .098$. Thus, this camp experience did not have a significant impact in altering a participants Cooperative Learning Context Preference.

Moreover, this is supported by reviewing this data on a case-by-case bases, revealed that as a result of this camp, two participants in the Communal Learning group moved from not endorsing a cooperative learning environment to endorsing a cooperative learning environment, from for a raw score of 2.17 to 3.29 and a raw score of 2.17 to 2.71, respectively. There was no endorsement change in the Individual Learning group, as they all preferred cooperative learning environments to start, as discussed above.

The next few sections share resulting pre- and post-test descriptive statistics followed by inferential statistics for the Individualistic Context Learning Preference scale.

Table 30

Pre- and Post-Test Individualistic Learning Context Preference Scores

Learning Context Preference		Test	Communal Learning			Individual Learning		
"INDIVIDUALISTIC"		(1-4)	M	SD	N	M	SD	N
All	42	Post	2.24	0.83	22	1.75	0.57	20
		Pre	2.16	0.68		1.91	0.46	
Gender:	17 Girls	Post	2.40	1.00	11	1.62	0.43	6
		Pre	2.25	0.87		1.90	0.37	
	25 Boys	Post	2.09	0.64	11	1.82	0.62	14
		Pre	2.07	0.44		1.92	0.50	
Pair Type:	4 Pairs of Girls	Post	2.18	0.97	8	-	-	-
		Pre	2.18	0.87		-	-	-
	4 Pair of Boys	Post	2.02	0.71	8	-	-	-
		Pre	2.18	0.47	8	-	-	-
	3 Pairs of Girl & Boy	Post	2.64	0.79	6	-	-	-
		Pre	2.10	0.75		-	-	-

Descriptive Statistics for Individualistic Learning Context Preference

All individual learning context preference mean scores were near or below the 2.50 (of 4.0) mark, suggesting that everyone had a relatively low preference for working individualistically. See Table 30 above. Moreover, according to these descriptive pre- and post-test scores, the preference to work individualistically increased for the communal learning group and decreased for the individual learning group. This holds true for both genders in both learning groups as well. This suggests that participants in both learning groups experienced an adverse reaction to the learning context of their assigned learning group. Perhaps after experiencing a few behavioral issues, the communal learning group determined it was better to work alone. Additionally, a conjecture can be made that the individual learning group potentially thought it would be better to work cooperatively, after having experienced an entire week of camp working alone. Of all the girls, those in the elementary school two girls-pair type were the only girls to experience a decrease in their preference to work individualistically. During camp, the elementary school two girls pair-type worked well together, often visibly celebrating milestones (e.g. giving each other high-fives). Thus, this decrease makes sense. Table 30 above shows more details.

Inferential Statistical Tests for Significance Individualistic Learning Context Preference

Paired-samples *t*-tests were run on the differences between pre- and post-test mean Individualistic Learning Context Preference scores to determine statistical significance. No outlier were present in the difference scores for pre- and post-test mean Cooperative Learning Context Preference score, as assessed by inspection of resulting boxplots. All assumptions were met. Differences between the pre- and post-test mean Individualistic Learning Context Preference scores were normally distributed, as assessed by Shapiro-Wilk's test ($p > .05$). There was homogeneity of variances for the difference in pre- and post-test mean Individualistic Learning Context Preference scores, as assessed by Levene's test of equality of variances ($p = .88$).

This camp experience resulted in with a mean difference of -.030 in Cooperative Learning Context Preference between pre- and post-test scores. However, this difference is not statistically significant, $t(41) = -0.30$, $p = 0.76$. Thus, this camp experience did not have a significant impact in altering a participants Individualistic Learning Context Preference.

Moreover, this is supported by reviewing the raw scores on a case-by-case, revealed that as a result of this camp, one participant in the Individual Learning group who moved from just barely endorsing an individualistic learning environment to not endorsing an individualistic learning environment. This participant's raw score decreased from 2.57 to 2.00. There was endorsement change in the Individual Learning group, as they all preferred cooperative learning environments to start, as discussed above.

The next few sections share resulting pre- and post-test descriptive statistics followed by inferential statistics for the Individualistic Context Learning Preference scale.

RQ3: Black Academic Identity

This section describes data analysis to respond to the third research question in this study:

> RQ3: Is there a change in the Black Academic Identity of the young African American elementary and middle school novice programmers after participating in this summer Scratch programming camp?

Here, measures of Black Academic Identity focused on three of the four possible manifestations of this construct, as described in Chapter 2 – Literature Review. These are: a) *Black Academic Identity*, where one's academic achievement is integrated with one's racial identity by thinking one's intellect is as a result of one's identity or that high academic achievement is crucial to being a successful Black student, thereby aligning one's behavior as such, b) *Black Model Phenomenon,* where one is motivated to achieve and be successful to satisfy a desire to be a positive role model for other members of their race, which is combined c) the *Proof of Black Ability*, where one reaches high academic achievement to dispel stereotypes, prove that African American students are intelligent and can succeed. The goal here was to determine participants' initial BAI and measure the change in BAI as a result of participating in their respective learning contexts – essentially exploring to what extent does learning computational thinking and programming skills impact BAI on African American elementary middle school novice programmers after learning in a communal learning context compared to those learning in an individual learning context. The Black Academic Identity (BAI) and Black Model Phenomenon (BMP) pre- and post-test scale scores (Appendix G) were analyzed to answer this research question and the results are described below, with descriptive statistics explained first followed by inferential statistics.

Cronbach's alpha reliability. The Cron's Alpha measure of internal consistency or reliability was consulted for the Black Academic Identity (BAI) and Black Model Phenomenon (BMP) scales before and after camp. These measurements indicated above average levels of internal consistency for BAI BEFORE and AFTER, 0.632 and 0.649, respectively. The internal consistency or reliability for BMP BEFORE and AFTER indicates relatively high levels, with scores of 0.825 and 0.872, respectively. These statistics are displayed in Table 31 below.

Table 31

Cronbach's Alpha Reliability for BAI & BMP (BEFORE and AFTER Camp)

	Before	After
Black Academic Identity (BAI)	0.632	0.649
Black Model Phenomenon (BMP)	0.825	0.872

Reliability and Descriptive Statistics for Black Academic Identity

The Black Academic Identity (BAI) and Black Model Phenomenon (BMP) pre- and post-test scale scores (Appendix G) were analyzed to answer RQ3 and the results are described below, with reliability and descriptive statistics explained first, followed by inferential statistics.

Table 32

All BAI & BMP Means and Standard Deviations BEFORE and AFTER Scratch *Camp*

Component	Pre	POST
Black Academic Identity (BAI)	2.86	2.88
Black Model Phenomenon (BMP)	3.76	3.62

Pre-camp BAI & BMP overall. Initial analysis revealed that the pre-camp (before anyone was given their assigned learning group) mean score for Black Academic Identity (BAI) was a little over the midpoint (M = 2.86 of 5.0, SD = 0.90, suggesting either a relatively average integration of racial identity and academic identity or one that BAI is not yet realized (neither high or low). At the end of camp, the experience of a programming camp revealed a slight increase in associating racial identity with academic identity overall, suggesting that perhaps the content of camp (i.e. learning how think computationally and program a computer) presented a slightly clearer focus on and motivation towards high academic achievement or still no realization of identity (neither high nor low). Additionally, the overall predisposition to be a model Black student (BMP) was fairly high ($M = 3.76$ of 5.0, $SD = 0.74$) for all participants before camp started and learning group assignments were shared. Of note, is that these mean scores did not drastically change at the end of camp either way, showing that participants' responses seem to be relatively consistent, in this case there was little to no change. See table 32 above. More analyses of the impact of these two identity categories on participants in each learning context presented a clearer account of the impact of this programming camp. These are analyses are described in detail below.

Post-camp BAI & BMP overall. Overall, this summer camp experience resulted in the BAI mean scores slightly increasing from participants in the Individual Learning group (from $M = 3.00$, $SD = 0.77$ to $M = 3.01$, $SD = 0.74$) and for those in the Communal Learning group (from $M = 2.71$, $SD = 0.99$ to $M = 2.76$, $SD = 0.86$). Additionally, the after summer camp experience mean BMP scores decreased for both learning group, going from $M = 3.69$, $SD = 0.80$ to $M = 3.62$, $SD = 0.75$ for those in the Communal Learning group and from $M = 3.83$, $SD = 0.68$ to $M = 3.62$, $SD = 0.71$ for those in the Individual Learning group, perhaps suggesting realization that the content and/or learning context of camp presented more challenging content (i.e. a new experience of learning how to program a computer) making it harder to succeed. It is interesting to note, however, that all these changes are extremely minimal and can almost be considered as no change. These data are represented below in table 33 figures 33 and 34 below, with more descriptive details about each following.

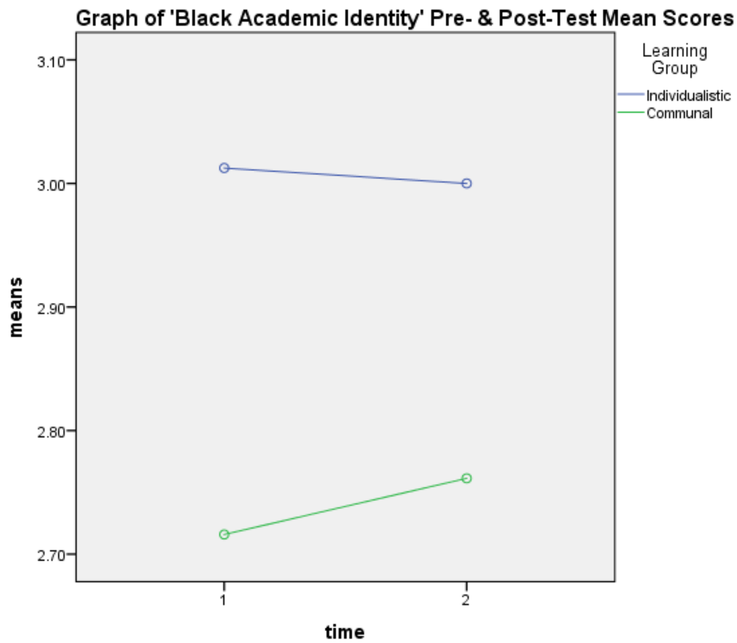


Figure 33. Graph of BAI Pre- & Post-Test Mean Scores for both Learning Groups.

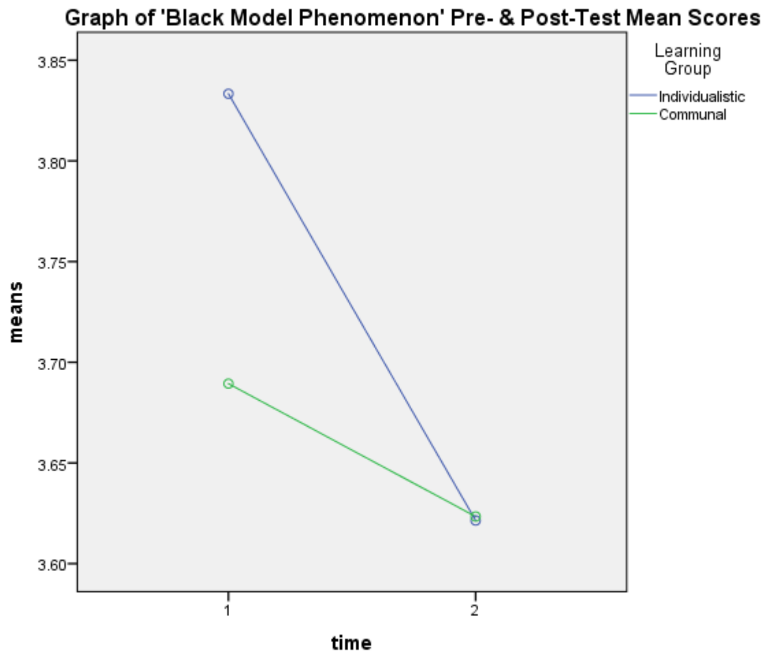


Figure 34. Graph of BMP Pre- & Post-Test Mean Scores for both Learning Groups.

Pre-camp BAI by pair-type, gender, and grade-level. Analysis by learning context (i.e. learning group) revealed more details about the impact of learning to program on BAI & BMP. The overall pre-camp (before Scratch lessons began) mean BAI scores were $M = 2.71$, $SD = 0.99$ for the communal learning group and $M = 3.00$, $SD = 0.77$ for the individual learning group. See table 33 below. Likewise, all mean scores by learning group and gender before camp started were above the midpoint, possibly suggesting an awareness of and agreement with the integration of racial identity and academic identity.

Table 33

Pre- and Post-Test Means Scores for Black Academic Identity (BAI) (n=42)

CT & Programming Skills		Test	Communal Learning			Individual Learning		
"BAI"		(5 pts)	M	SD	n	M	SD	n
All	n=42	Post	2.76	0.86	22	3.01	0.74	20
		Pre	2.71	0.99		3.00	0.77	
Gender:	17 Girls	Post	2.75	0.74	11	2.71	0.64	6
		Pre	2.75	1.10		2.63	0.57	
	25 Boys	Post	2.77	1.01	11	3.13	0.76	14
		Pre	2.68	0.93		3.18	0.81	
Pair Type:	4 Pairs of Girls	Post	2.84	0.50	8	-	-	-
		Pre	2.84	1.20		-	-	-
	4 Pair of Boys	Post	2.53	1.01	8	-	-	-
		Pre	2.34	0.78		-	-	-
	3 Pairs of Girl & Boy	Post	2.96	1.10	6	-	-	-
		Pre	3.04	0.94		-	-	-
Elementary:	All (21)	Post	3.15	0.57	10	3.11	0.74	11
		Pre	3.35	0.99		3.14	0.74	
Gender:	10 Girls	Post	3.15	0.49	6	2.50	0.71	4
		Pre	3.21	1.22		2.75	0.61	
	11 Boys	Post	3.50	0.54	4	3.46	0.53	7
		Pre	3.56	0.59		3.36	0.75	
Pair Type:	2 Pairs of Girls	Post	2.75	0.50	4	-	-	-
		Pre	3.31	1.55		-	-	-
	1 Pair of Boys	Post	3.13	0.53	2	-	-	-
		Pre	3.13	0.53		-	-	-
	2 Pairs of Girl & Boy	Post	2.56	0.43	4	-	-	-
		Pre	3.50	0.61		-	-	-
Middle:	All (21)	Post	2.44	0.95	12	2.86	0.75	9
		Pre	2.18	0.64		2.86	0.83	
Gender:	7 Girls	Post	2.55	0.99	5	3.13	0.18	
		Pre	2.20	0.67		2.38	0.53	2
	14 Boys	Post	2.36	1.00	7	2.79	0.85	
		Pre	2.18	0.67		3.00	0.88	7
Pair Type:	2 Pairs of Girls	Post	2.94	0.55	4	-	-	-
		Pre	2.38	0.63		-	-	-
	3 Pairs of Boys	Post	2.33	1.09	6	-	-	-
		Pre	2.08	0.68		-	-	-
	1 Pair of Girl & Boy	Post	1.75	1.06	2	-	-	-
		Pre	2.13	0.88		-	-	-

Grade-Level and learning group. Before camp started, analysis by grade level and learning group shown that the elementary school participants had higher mean BAI scores than their middle school counterparts, also indicated in table 33 above. More specifically, the elementary school communal learning group had a mean BAI score of 3.35, *SD* = 0.99 and the middle school communal learning group's Mean BAI score was 2.18, *SD* = 0.64, while the elementary school individual learning group had a mean BAI score of 3.14, *SD* = 0.74 and the middle school individual learning group's mean BAI score was 2.86, *SD* = 0.83. Thus, the elementary school participants consistently had a higher BAI than the middle school participants

Grade-level and gender. Analysis before camp by grade-level, gender, and learning group depicted the boys in each grade level for both learning groups had higher BAI score than their female counterparts, except the middle school communal learning group boys, who scored marginally less than their female counterparts. The elementary school communal learning group boys mean BAI score was 3.56, *SD* = 0.59, while that of their female counter parts pre-test mean BAI score was lower at 3.21, *SD* = 1.22, the elementary individual learning group boys mean BAI score was 3.36, *SD* = 0.75, while their female counterparts was also lower at *M* = 2.75, *SD* = 0.61. The middle school individual learning group boys had a pre-test mean BAI score of 3.00, *SD* = 0.88 and their female counterparts had a lower pre-test mean BAI score of 2.38, *SD* = 0.53. The slight exception to this pattern was with the middle school communal learning group boys who had a mean BAI score of 2.18, *SD* = 0.67 while their female counterparts mean BAI score was higher at *M* = 2.20, *SD* = 0.67. Furthermore, of all participants, the elementary school boys in the individual learning group (*M* = 3.56, *SD* = 0.59) expressed the highest pre-test mean BAI score, while the elementary girls in the individual learning group (*M* = 2.75, *SD* = 0.61) expressed the lowest of all participants. As a result, with the boys largely had a higher mean BAI score before camp than their female counterparts. These data are present in Table 32 above.

End of camp analysis sharing the overall impact of learning computational thinking and programming skills on BAI and are described after Table 32, below.

Post-camp BAI by gender, pair-type, and grade-level. By gender, the girls in the Communal Learning group basically no change in mean BAI score. They went from *M* = 2.75, *SD* = 1.10 to *M* = 2.75, *SD* = 0742, while the girls in the Individual Learning group had a slight increase in mean BAI scores (from *M* = 2.63, *SD* = 0.57 to *M* = 2.71, *SD* = 0.64). Additionally, the boys in the Communal Learning group had a slight increase (from *M* = 2.68, *SD* = 0.93 to *M* = 2.77, *SD* = 1.01), while the boys in the Individual Learning group had a slight decrease in mean BAI score (from *M* = 3.18, *SD* = 0.81 to *M* = 3.13, *SD* = 0.76).

By grade and gender. The elementary school communal learning group experienced a slight decrease in their mean BAI score (from *M* = 3.35, *SD* = 0.99 to *M* = 3.15, *SD* = 0.57) as well as those in the individual learning group (from *M* = 3.14, *SD* = 0.74 to *M* = 3.11, *SD* = 0.74). All elementary school individual learning boys were the only group to experience an increase in BAI (from *M* = 3.36, *SD* = 0.75 to *M* = 3.46, *SD* = 0.53), while everyone else experienced a decrease. Conversely, the mean BAI scores for the middle school increased for everyone (from *M* = 2.20, *SD* = 0.67 to *M* = 2.55,

$SD = 0.99$ for the communal learning girls, $M = 2.18$, $SD = 0.67$ to $M = 2.36$, $SD = 1.00$ for the communal learning boys, and $M = 2.38$, $SD = 0.53$ to $M = 3.13$, $SD = 0.18$ for the individual girls), except the individual learning group boys, whose mean BAI score when from $M = 3.00$, $SD = 0.88$ to $M = 2.79$, $SD = 0.85$.

By pair-type and gender. By pair-type, regardless of grade-level, the two girls pair-type experienced almost no change in mean BAI scores (from $M = 2.84$, $SD = 1.20$ to $M = 2.84$, $SD = 0.50$). The two boys pair-type experienced an increase in BAI, from $M = 2.34$, $SD = 0.78$ to $M = 2.53$, $SD = 1.03$, while the one boy & one girl pair-type had a decrease from $M = 3.04$, $SD = 0.94$ to $M = 2.96$, $SD = 1.10$. When grade-level was considered the elementary school two boys pair-type had no change in mean BAI score (from $M = 3.13$, $SD = 0.53$ pre and post-test), while the two girls and the one boy & one girl pair-types experienced decreases, from $M = 3.31$, $SD = 1.55$ to $M = 2.75$, $SD = 0.50$ and from $M = 3.50$, $SD = 0.61$ to $M = 2.56$, $SD = 0.43$ respectively. At the middle school level, the two boys pair-type experienced an increase (from $M = 2.08$, $SD = 0.68$ to $M = 2.33$, $SD = 1.09$), while the one boy & one girl pair type experienced a decrease (from $M = 2.13$, $SD = 0.88$ to $M = 1.75$, $SD = 1.06$). Ultimately, the middle school two girls pair-type had highest increase in mean BAI scores at the end of camp (from $M = 2.38$, $SD = 0.63$ to $M = 2.94$, $SD = 0.55$). The mean BAI scores decreased for the one boy & one girls pair-type and seemed to stay consistent with the two girls pair-type and with the Individual Learning group. See figure 35 below.

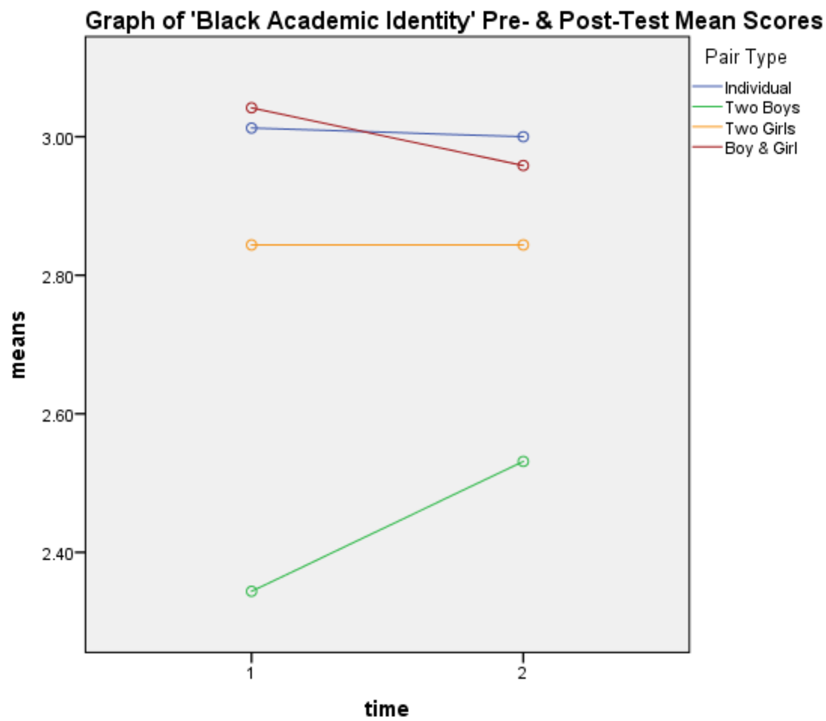


Figure 35. Graph of BAI Pre- & Post-Test Mean Scores by Pair Type.

However, closer analysis of the Individual Learning group revealed that the BAI decreased slightly for the boys (from $M = 3.18$, $SD = 0.81$ to $M = 3.13$, $SD = 0.76$) and increased for the girls (from $M = 2.63$, $SD = 0.57$ to $M = 2.71$, $SD = 0.64$). Figure 36 below represents this.

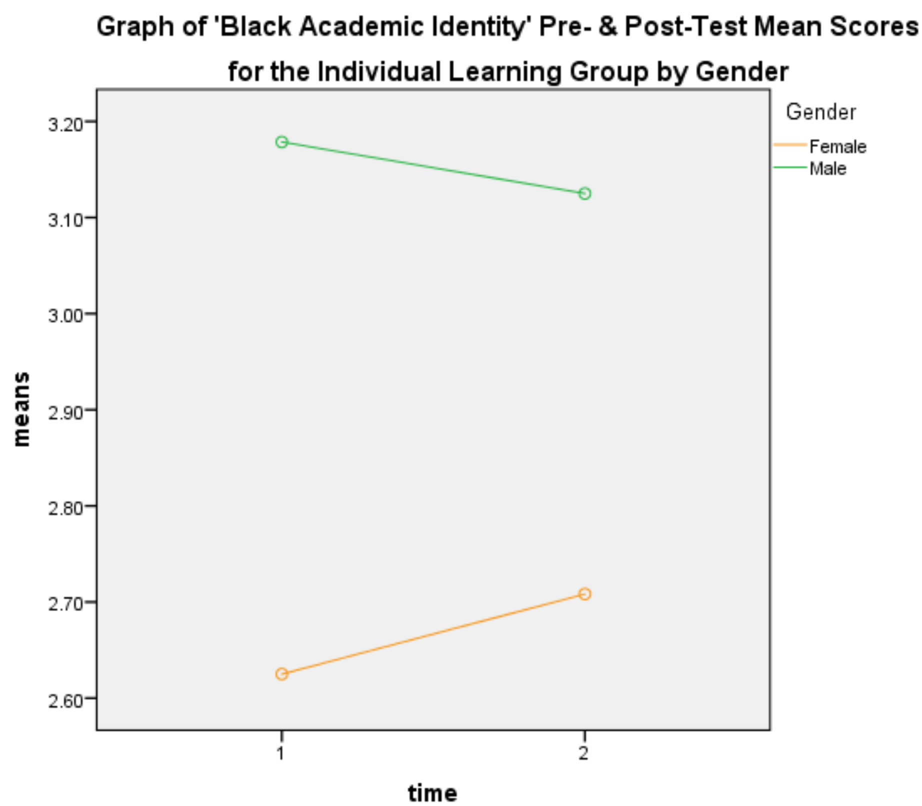


Figure 36. Graph of BAI Pre- & Post-Test Mean Scores for the Individual Learning group by Gender.

Of note is that mean BAI scores increased for boys working together communally and decreased for boys working individually, while also increasing for girls working individually. The researcher probed further to examine the specific responses to each component of the BAI scale. The results of which are described below.

Black Academic Identity components. The "**Being a good Black student is an important part of who I am.**" BAI component resulted with the highest mean BAI score for all participants, with a mean of 3.79. Conversely, the component with the lowest mean score was for "**I get good grades because I am Black**." This suggests that participants value getting good grades as part of their Black identity, but they do not believe their good grades simply come from them being Black. This suggests an interesting relationship between the race and academics. Table 34 below shares more of the mean scores and standard deviations of all the separate BAI components.

Table 34

Specific Means and Standard Deviations for the Components of Black Academic Identity

Black Academic Identity Components (1-5)	Communal Learning Group		Individual Learning Group	
	Pre	POST	Pre	POST
I think of myself as a Black student, not just a student.	2.82 (1.27)	2.95 (1.43)	3.15 (1.27)	3.45 (1.43)
Being a good Black student is an important part of who I am.	.68 (1.42)	3.68 (1.46)	3.90 (1.17)	4.00 (1.43)
I get good grades because I am Black.	1.68 (1.17)	1.68 (1.13)	1.80 (1.28)	1.70 (1.30)
I seldom think of myself as a Black student (Reverse coded).	3.32 (1.29)	2.72 (1.43)	2.80 (1.28)	2.85 (1.13)

The next sections share details about the mean BMP scores from the pre- and post-tests.

Table 35

Pre- and Post-Test Means Scores for Black Model Phenomenon (BMP)

CT & Programming Skills		Test	Communal Learning			Individual Learning		
"BMP"		(5 pts)	M	SD	n	M	SD	n
All	n=42	Post	3.62	0.75	22	3.62	0.71	20
		Pre	3.69	0.80		3.83	0.68	
Gender:	17 Girls	Post	3.53	0.77	11	3.57	0.78	6
		Pre	3.78	0.79		3.94	0.66	
	25 Boys	Post	3.71	0.76	11	3.64	0.71	14
		Pre	3.59	0.85		3.79	0.71	
Pair Type:	4 Pairs of Girls	Post	3.25	0.70	8	-	-	-
		Pre	3.54	0.79		-	-	-
	4 Pair of Boys	Post	3.63	0.83	8	-	-	-
		Pre	3.50	0.97		-	-	-
	3 Pairs of Girl & Boy	Post	4.12	0.45	6	-	-	-
		Pre	4.14	0.45		-	-	-
Elementary:	All (21)	Post	3.86	0.79	10	3.87	0.70	11
		Pre	3.83	0.89		4.09	0.63	
Gender:	10 Girls	Post	3.69	0.95	6	3.29	0.80	4
		Pre	3.75	0.94		3.88	0.84	
Pair Type:	11 Boys	Post	4.11	0.47	4	4.20	0.39	7
		Pre	3.96	0.93		4.21	0.51	
	2 Pairs of Girls	Post	3.32	0.98	4	-	-	-
		Pre	3.45	1.07		-	-	-
	1 Pair of Boys	Post	3.93	0.71	2	-	-	-
		Pre	3.92	1.53		-	-	-
Middle:	2 Pairs of Girl & Boy	Post	4.36	0.14	4	-	-	-
		Pre	4.17	0.33		-	-	-
	All (21)	Post	3.42	0.69	12	3.32	0.63	9
		Pre	3.57	0.74		3.52	0.64	
Gender:	7 Girls	Post	3.34	0.52	5	4.14	0.40	2
		Pre	3.83	0.66		4.08	0.12	-
	14 Boys	Post	3.49	0.82	7	3.08	0.45	7
		Pre	3.38	0.79		3.36	0.63	-
Pair Type:	2 Pairs of Girls	Post	3.18	0.43	4	-	-	-
		Pre	3.63	0.53		-	-	-
	3 Pairs of Boys	Post	3.52	0.90	6	-	-	-
		Pre	3.36	0.87		-	-	-
	1 Pair of Girl & Boy	Post	3.64	0.51	2	-	-	-
		Pre	4.08	0.83		-	-	-

Pre-camp BMP by learning group, gender and pair-type, and grade-level. Before camp started, the mean BMP score for everyone (M = 3.76 of 5.0, SD = 0.74) was relatively high overall, suggesting these participants have a higher disposition to think that high academic achievement is crucial for them being model Black students along with a high desire to dispel stereotypes about the low academic ability and intellect of African American people. The pre-test mean BMP score for the communal learning group before Scratch lessons began was M = 3.69, SD = 0.80 and the mean BMP score for the individual learning group was M = 3.83, SD = 0.68.

By gender. Overall the girls in both learning groups had higher mean BMP scores than their male counterparts, but not much higher. The pre-test mean BMP score for the girls in the communal learning group was M = 3.78, SD = 0.79, while the pre-test mean BMP score for their male counterparts was M = 3.59, SD = 0.85. Within the individual learning group, the pre-test mean BMP score for the girls was M = 3.94, SD = 0.66, while the pre-test mean BMP score for their male counterparts was M = 3.79, SD = 0.71. The girls having a larger pre-test mean BMP score than the boys is a pattern only repeated by the middle school participants, where the girls in the middle school communal learning group had a pre-test mean BMP score of M = 3.83, SD = 0.66, while their male counterparts had a pre-test mean BMP score of M = 3.38, SD = 0.79. Additionally, the girls in the middle school individual learning group had a pre-test mean BMP score of M = 4.08, SD = 0.12, which was much higher than their male counterparts with a pre-test mean BMP score of M = 3.36, SD = 0.63. However, the elementary school participants broke away from this pattern because the elementary school boys scored higher than their female counter parts, as the boys in the elementary school communal learning group had a pre-test mean BMP score of M = 3.96, SD = 0.93, while the pre-test mean BMP score of their female counterparts was M = 3.75, SD = 0.94. Likewise, the mean BMP score for the boys in the elementary school individual learning group was M = 4.21, SD = 0.51, while the pre-test mean BMP score for their female counterparts was M = 3.88, SD = 0.84. Furthermore, the boys in the elementary school individual learning group had the highest pre-test mean BMP score of M = 4.21, SD = 0.51. Here, however, it should also be noted that the lowest pre-test mean BMP score before Scratch lessons belonged to both the boys in the middle school individual learning group, M = 3.36, SD = 0.63 and the two boy pair-type in the communal learning group, M = 3.36, SD = 0.87.

Grade-level and learning group. Analysis by grade level and learning group revealed that the elementary school participants also had a slightly higher mean BMP score than the middle school participants. Before Scratch lessons began, the pre-test mean BMP scores for the elementary school communal learning group was $M = 3.83$, $SD = 0.89$ and $M = 4.09$, $SD = 0.63$ for the elementary school individual learning group. The pre-test mean BMP scores for the middle school communal learning group before Scratch lessons started was $M = 3.57$, $SD = 0.74$ and $M = 3.52$, $SD = 0.64$ for the middle school individual learning group.

By gender and pair-type. The one boy & one girl pair-type, regardless of grade, started with the highest mean BMP score ($M = 4.14$, $SD = 0.45$), followed by the two girls pair- type ($M = 3.54$, $SD = 0.79$) and two boys pair-type ($M = 3.50$, $SD = 0.97$). The one boy & one girl pair-type started with the highest pre-test mean BMP score at the elementary and middle school levels as well, *M* = 4.17, *SD* = 0.33 and *M* = 4.08, *SD* = 0.83, respectively. These and more detailed mean BMP scores can be found in Table 35 below.

Post-camp BMP by learning group, gender and pair-type, and grade-level. The resulting post-test mean BMP score for both groups was 3.62, *SD* = 0.72. As a result, both learning groups saw a decrease, with the mean BMP score of the communal learning group from *M* = 3.69, *SD* = 0.80 to 3.62, *SD* = 0.75 and the post-test mean BMP score of the individual learning group from *M* = 3.83, *SD* = 0.68 to *M* = 3.62, *SD* = 0.71. The overall trend for the Black Model Phenomenon (BMP), the desire to be a positive example of a well performing Black student, decreased for all participants in both learning groups except for the two boys pair- type in the Communal Learning context.

By gender. The girls in both learning groups experienced a decrease in mean BMP scores, with those in the communal learning group going from 3.78 to 3.53 and those in the individual learning group going from 3.94 to 3.57. The boys in the communal learning group were the only group to experience an increase in mean BMP scores, going from 3.59 to 3.71, while the individual learning group boys experienced a decrease from 3.79 to 3.64.

By grade-level and gender. Both learning group experienced a decrease in mean BMP scores with those in the elementary school communal learning group experiencing the smallest increase, going from *M* = 3.83, *SD* = 0.89 to *M* =3.86, *SD* = 0.79, while those in the elementary school individual learning group experienced a larger decrease from *M*

= 4.09, *SD* = 0.63 to *M* = 3.87, *SD* = 0.70. The steepest decrease in post-test mean BMP scores was experienced by the two girls pair-type, followed by those in the Individualistic Learning context. Even more, within the Individual Learning group, both the boys and the girls experienced decreases in mean BMP scores with the boys going from *M* = 3.79, *SD* = 0.71 to *M* = 3.64, *SD* = 0.71 and the girls going from *M* = 3.94, *SD* = 0.66 to *M* = 3.57, *SD* = 0.78. This is illustrated in Figure 38 below.

By pair-type and grade-level. Overall, regardless of grade-level, the only pair-type that experienced an increase in mean BMP scores was the two boys pair type (from *M* = 3.50, *SD* = 0.97 to *M* = 3.63, *SD* = 0.83), while the two girls and one boy & one girl pair-type experienced slight decreases, from *M* = 3.54, *SD* = 0.79 to *M* = 3.25, *SD* = 0.70 and from *M* = 4.14, *SD* = 0.45 to *M* = 4.12, *SD* = 0.45, respectively. At the elementary school level, only the two girls pair-type experienced a decrease, from *M* = 3.45, *SD* = 1.07 to *M* = 3.32, *SD* = 0.98. Conversely, the two boys and one boy & one pair-type had increases in mean BMP scores. These were a slight increase with the two boys pair type from *M* = 3.92, *SD* = 1.53 to *M* = 3.93, *SD* = 0.71 along with an increase from *M* = 4.17, *SD* = 0.33 to *M* = 4.36, *SD* = 0.14 for the one boy & one girl pair-type. With the middle school participants, only the two boys pair type had an increase in mean BMP scores, from *M* = 3.36, *SD* = 0.87 to *M* = 3.52, *SD* = 0.90. The two girls and one boy & one girl pair-type experienced decreases in post-test mean BMP scores, from *M* = 3.63, *SD* = 0.53 to *M* = 3.18, *SD* = 0.43 and *M* = 4.08, *SD* = 0.83 to *M* = 3.64, *SD* = 0.51, respectively. These data are present in table 35 above and figures 37 and 38 below.

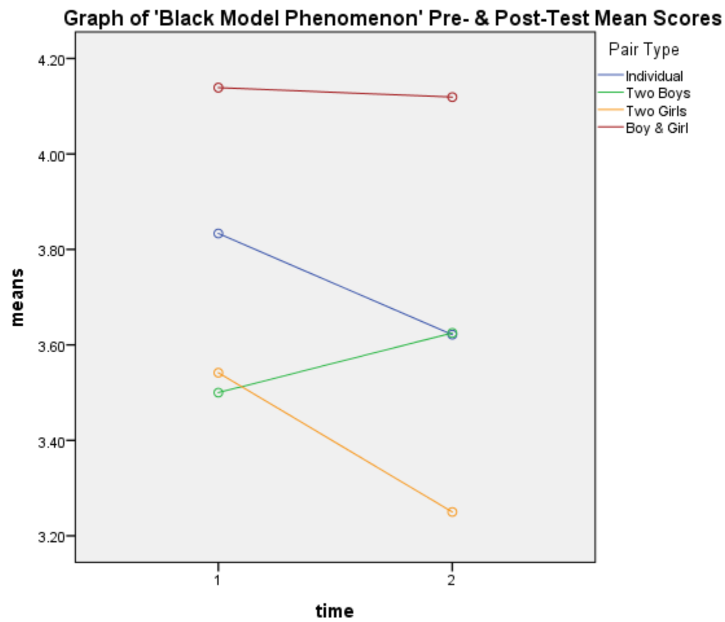


Figure 37. Graph of BMP Pre- & Post-Test Mean Scores by Pair-Type.

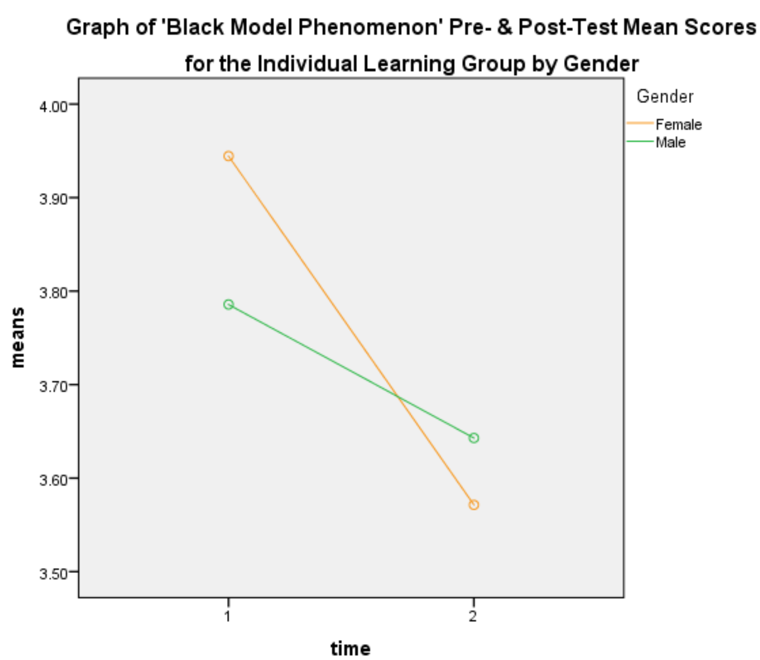


Figure 38. Graph of BAI Pre- & Post-Test Mean Scores for the Individual Learning group by Gender.

To continue the investigation into the potential impact this study may have had on its participants, the researcher explored each component question of the BMP scale. The details of this exploration are described below.

Black Model Phenomenon components. The lowest of all BMP components, the component "It is a burden to prove to others that I am a smart Black," suggests that students do not find it a burden to prove to others that they are smart Black students. This is illustrated in table 36 below. The remaining components of the BMP scale received relatively high scores individually and these relatively high scores remained fairly consistent before and after this study. This suggests that participants are aware of the connection between their racial identity and academic identity and that although some means scores increased while other decreased, these changes were small. This suggests that this summer programming experience may have only had a small impact on participant BMP.

Table 36

Specific Means and Standard Deviations for the Components of Black Model Phenomenon

Black Model Phenomenon Components (1-5)	Communal Learning Group Pre	Communal Learning Group POST	Individual Learning Group Pre	Individual Learning Group POST
I want to be an example of Black success.	3.73 (1.42)	3.77 (1.23)	4.05 (1.10)	4.10 (0.97)
I want to represent Black students in a positive way.	4.14 (0.99)	3.86 (1.04)	4.30 (0.87)	4.10 (0.91)
I want to be Black role model.	3.91 (1.27)	3.77 (1.19)	3.65 (1.29)	3.60 (1.35)
I want to show others that Black students are smart.	4.36 (0.95)	4.14 (1.13)	4.05 (1.23)	3.85 (1.31)
Getting good grades in school is the best way to prove society wrong about Black people.	3.73 (1.42)	3.82 (1.33)	4.10 (1.12)	3.70 (1.30)
It is a burden to prove to others that I am a smart Black student.	2.27 (1.55)	2.41 (1.47)	2.85 (1.66)	3.45 (1.54)

Interviews

To discover more about what participants thought about the connection between being Black and their academic performance, the research asked two questions at the end of the study to connect notions of academic performance to performance during camp and notions of performing well during camp to being a young black male or female. These questions were:

1. Is there a connection between your academic performance and how you'd like to perform in Scratch?
2. Is there a connection between being a young black <male/female> to doing well in Scratch?

Twenty-two participants were interviewed, 12 from the Communal Learning group and 12 from the Individual Learning group; six girls and six boys. The first question was asked to determine if participants made a connection between school and using Scratch in this study, particularly regarding level of effort and level of performance. Most participants saw no connection between these two. The only connection made was with the use of computers. Two middle school boys, in the individual learning group expressed in separate interviews, that there are deadlines and grades in school and neither of these existed during camp. One elementary school girl in the individual learning group recognized that doing well in both school and in Scratch required time and effort by saying, *"there is no connection, but you have to work hard and spend a lot of time with both."*

When asked if there was a connection between participants being Black and doing well in Scratch, some saw no connection. This was expressed most clearly by one elementary school boy in the communal learning group expressed it this way, *"*Scratch *is a game, being black is a whole nothing thing."* However, many participants in both groups were indicated an awareness of a connection but could not explain it. These participants responded with words such as *"kind of,"* and *"sort of,"* and "*yes, but it's confusing."* One middle school boy in the communal learning group did no see a connection but was aware that other people made a connection. This was expressed when he responded, *"I don't see a connection, but I know other people make the connection a lot."*

Inferential Statistical Tests for Significance Black Academic Identity

Overall. Paired-samples *t*-tests were run on the differences between pre- and post-test mean Black Academic Identity scores to determine statistical significance. No outliers were present in the difference scores for pre- and post-test Black Academic Identity score, as assessed by inspection of resulting boxplots. All assumptions were met. Differences between the pre- and post-test mean Black Academic Identity scores were normally distributed, as assessed by Shapiro-Wilk's test ($p > .05$, $p =.391$).

This camp experience caused a mean difference of -0.0179 in Black Academic Identity between pre- and post-test scores. However, this change is not statistically significant, $t(41) = -0.17$, $p =.86$ Thus, this camp experience did not have a significant impact in altering a participants Black Academic Identity. This is supported by the findings in the interviews and in the slight changes between pre- and post-test mean BAI descriptive statistics mentioned above.

By grade-level. The descriptive statistics illustrated and annotated above consistently revealed that the BAI scores for the elementary school students were consistently higher than their middle school student counterparts. As such, an independent t-test was performed on BAI by grade-level (i.e. elementary school vs. middle school) to determine if the difference in pre- and post-test BAI scores was statistically significant.

BAI pre-test grade level significance. The pre-test mean Black Academic Identity score for the elementary school participants ($M = 3.09$, $SD = 0.63$) was higher than that of the middle school participants ($M = 2.69$, $SD = 0.564$). The highest BAI score possible is 5, which means that a mean score of 2.5 or higher indicates a high degree of Black Academic Identity. In this case the elementary school students have a higher sense of Black Academic Identity than the middle school students. The results of the independent t-test shows that this difference is statistically significant, $t(40) = 2.194$, $p = 0.034$.

BAI post-test grade level significance. The post-test mean Black Academic Identity score for the elementary school participants ($M = 3.13$, $SD = 0.65$) was higher than that of the middle school participants ($M = 2.61$, $SD = 0.88$). The results of the independent t-test shows that this difference is statistically significant, $t(40) = 2.146$, $p = 0.038$.

Inferential Statistical Tests for Significance Black Model Phenomenon

Overall. Paired-samples t-tests were run on the differences between pre- and post-test mean Black Model Phenomenon scores to determine statistical significance. Two outlier were present in the difference scores for pre- and post-test Black Model Phenomenon score, as assessed by inspection of resulting boxplots. However, because these were outliers were not extreme, they were kept in the analysis. All other assumptions were met. Differences between the pre- and post-test mean Black Academic Identity scores were normally distributed, as assessed by Shapiro-Wilk's test ($p > .05$, $p = .476$).

This camp experience caused a mean change of -0.135in Black Academic Identity between pre- and post-test scores. However, this change is not statistically significant, $t(41) = -1.497$, $p = .142$. Thus, this camp experience did not have a significant impact in altering a participants Black Model Phenomenon. This is supported by the findings in the slight changes between pre- and post-test mean BMP descriptive statistics mentioned above.

By grade-level. The descriptive statistics illustrated and annotated above consistently revealed that the BMP scores for the elementary school students were consistently higher than their middle school student counterparts. As such, an independent t-test was performed on BMP by grade-level (i.e. elementary school vs. middle school) to determine if the difference in pre- and post-test BMP scores was statistically significant.

BMP pre-test grade level significance. The pre-test mean Black Academic Identity score for the elementary school participants (M = 3.97, SD = 0.76) was higher than that of the middle school participants (M = 3.55, SD = 0.68). The highest BMP score possible is 5, which means that a mean score of 2.5 or higher indicates a high degree of Black Model Phenomenon. In this case the elementary school students have a higher sense of Black Model Phenomenon than the middle school students. The results of the independent t-test shows that this difference is not statistically significant, $t(40)$ = 1.892, p = 0.066.

BMP post-test grade level significance. The post-test mean Black Model Phenomenon score for the elementary school participants (M = 3.86, SD = 0.72) was higher than that of the middle school participants (M = 3.38, SD = 0.65). The results of the independent t-test shows that this difference is significant, $t(40)$ = 2.273, p = 0.029.

Summary of the Impact on Black Academic Identity

Although there seems to be small patterns in the change in mean BAI and BMP scores in the pre- and post-tests as a result of this study, these changes are very minimal. One interesting pattern exists with the two boys pair-type. When most other participant mean BAI scores slightly decreased or remained the same, the mean BAI score of the participants in the two boys pair-type increased. Regarding mean BMP scores, participants in the two boys pair- type experienced an increase when all other pair types experienced a decrease. These contrasts are illustrated above in figures 35 and 37 respectively.

However, there was a statistically significant difference in pre and post-test mean BAI scores, where the elementary school participants had higher BAI score than their middle school counterparts. This statistical significant held for the difference in post-test mean BMP score, but not for the pre-test mean BMP score.

When asked specifically about the connection between being Black and academic achievement, most participants saw no connection. Likewise, when asked to describe if there was a connection between being Black, most participants did not express one. Interestingly, however, one participant shared his awareness that other people make this connection.

Although participants did not express a definite connection between being Black and academic performance, their pre- and post-camp responses were higher than the midpoint for both BAI and BMP and especially high for BMP. More analysis revealed this to be true for each of the BAI and BMP component questions, where responses were consistent on the pre- and post-camp tests, with little variability. Regarding BAI, the rating for the "**Being a good Black student is an important part of who I am"** statement received the highest mean score for participants in both learning groups. The same is true for the "**It is a burden to prove to others that I am a smart Black student**" BMP component statement. This suggests some awareness of the connection between racial identity and academic identity by participants in both learning groups, even though they could not seem to verbally express it.

Chapter Five: Discussion and Conclusion

Discussion

The 42 participants in this study reported various levels of technology use in their daily lives as responses to the pre-camp previous computer experience questionnaire (Appendix D). Eighty-nine percent reported using computers to play games. Eight-four percent enjoy playing games on mobile devices. Seventy-two percent reported watching movies or online videos at least once a week. Sixty-six percent reported that they can explain how to use a computer to someone who needs help. Forty-six percent reported that they participate in online multi-player video games at least once a day, while 23% participate in the games several times a day. Thirty percent reported using the Internet to read comics at lease once a week. While 68% said they take an online class at least once a month, only 30% reported that they consider themselves a part of the computational thinking (computer science, engineering, scientist) community. Twenty-five percent reported working on digital media projects outside of school on a daily basis while fifty-one percent said they have either never done it or have done it less than once a month. Fifty-six percent reported that they have never used a computer to program, and nearly 40% said they have never used a computer to create a multimedia presentation. Seven percent said they have made used Scratch six or more times prior to this study. With these statistics, the statistics describing the population of this study closely matches the statistics used to describe the tweens and teens described in Chapter One – Introduction; they consume technology in large amounts but many do not use computing technology to create.

Even more, the population of this study also matches the underrepresented communities in computer science and computer engineering also described in Chapter One. Without strategic interventions, it will be extremely challenging to change these statistics.

Brief description of study. Therefore, the purpose of the study presented here was to offer a way towards determining and exploring strategies that will contribute to the successful teaching and learning of computational thinking and computer programming skills. Counter to the typical studies designed to teach these skills, this study solicited and accepted participants who are all members of the underrepresented populations described above (African American and Latino American) and thus contributes to the CSEd literature as such. More specifically, these participants are African American elementary and middle school novice programmers. Additionally, it experiments with and explores the benefits of using culturally relevant pedagogy to teach computational thinking and programming skills. The specific culturally relevant pedagogical instructional strategy explored here focused on Communalism. Communalism is one of the nine elements of the Black Cultural Ethos, distinctive cultural phenomenon that contributes to and enhances the academic performance of African American students. Here, communalism was specifically instantiated in the context of a communal learning environment, where participants were placed in pairs, expected to share resources, and rely on one another to learn and use computational thinking and programming skills using the Scratch visual programming language. Placing communal learning group participants in pairs was also done as a result of the benefits of pair programming in the computer science literature. More explicitly, during a summer camp, lasting 5 week days, 3 hours per day, a mixed methods research design was used, where 22 participants were randomly assigned to the group where they work in pairs, i.e. the communal learning group, and 20 participants were randomly assigned to work alone, i.e. the individual learning group, to observe, describe, and compare how participant in each learning group learned computational thinking and programming skills. Furthermore, the pairs in the communal learning group were specifically designed with three pair-types: two girls, two boys, and one boy & one girl. Data collection and data analysis methods followed a convergent parallel mixed-method research design, where quantitative and qualitative research methods were used to understand the learning processes which occurred in both learning groups, i.e. communal and individual learning. These methods included independent t-tests, paired samples t-tests, the critical incident analysis technique, and a cognitive assessment of participants' problem-solving and program development skills. Data sources included pre- and post-study Scratch content knowledge tests, participant notebooks and resulting Scratch projects, in addition to audio and video recordings, opinion prompts, end-of-camp interviews, and a multi-item scale used to measure participant learning preferences and Black Academic Identity.

Results: learning and using Scratch computational thinking and programming skills. All participants expressed a relatively high preference for working cooperatively and a relatively low reference for working individualistically. This affirms the notion of communalism, one of nine elements highlighted here as a cultural phenomenon which contributes to enhanced academic performance of African American students. Furthermore, post-test mean scores of content knowledge related to Scratch were measure and compared. Although neither learning group ended the camp experience with particularly high post-test mean scores, overall, the participants in the communal learning group performed better than participants in the individual learning group regarding Scratch content knowledge as well as the core computational thinking and programming skills identified by the creators of Scratch, namely sequences, events, loops, conditionals, operators, and working with data. These differences in scores were statistically significant for all skills except events, which had just two questions on the pre- and post-test dedicated to it. More specifically, a pattern emerged where the one boy & one girl pair-type within the communal learning group consistently scored the highest in all but one category. These included Scratch overall, sequences, loops, conditionals, operators, and working with data. The one category where the one boy & one girl pair-type did not score the highest was that of events. Here, the one boy and one girl pair-type scored the second highest to the two girls pair-type. Another pattern also emerged; participants in the individual learning group scored the lowest in all but two programming categories, when compared to all three pair-types. These categories included Scratch content overall, sequences, loops, conditionals, operators, and working with data. When the individual learning group did not score the lowest, the two boys pair-type scored the lowest. This occurred twice with the events and loops categories. It is worth noting that the middle school boys in the communal learning group had severe behavioral challenges. These challenges are in line with what some researchers call hegemonic masculinity. These behavioral challenges may have contributed to the low performance by participants in the two boys pair-type.

Learning and using Scratch. The learning process revealed various characteristics of learning and using Scratch as well. Participants in the communal learning group had to share learning resources. These included a Scratch Booklet and one computer. Each participant had his or her own notebook to use at his or her discretion. Participants in the communal learning group shared a variety of reading strategies including taking turns reading pages and paragraphs along with discussing what was read after each member of the pair read. Participants in the individual learning group simply shared that they read and attempted programming activities as they progressed linearly through the book. On average, all participants in both learning groups completed 4.5 units of 15. No one expressed skipping around the Scratch Booklet. Also, regardless of learning group membership, to the researcher, many participants seemed to have reading challenges, as evidenced by the audio recordings. Participants were not aware of these challenges and reported enjoying in reading the Scratch Booklet. Resulting Scratch projects had a variety of characteristics, from being simple to relatively complex (for novices), to remixed, and those artfully creative. It is worth noting that the relatively complex Scratch projects were created by participants in the one boy & one girl and two girls pair-types within the communal learning group – the group which most often scored the highest in most categories of the Scratch post-test.

Even though participants ended the study with relatively low mean post-test scores and there was a lack of many complex Scratch projects, the resulting scores, Scratch projects, and interviews suggest that the African American elementary and middle school novice programmers in this study learned and were able to use some of the computational thinking and Scratch programming skills presented during this summer camp. The notebooks were not used often by either learning group. However, a pattern emerged where many participants wrote the exact same information in their notebooks. This information included the definition of 'initialization' and the size in pixels of the Scratch stage. This is the only information recorded by participants and it is not clear why this information was recorded and only this information. Nonetheless, the commonality suggests the potential for teaching young novice programmers how and when to document their thoughts, ideas, problem-solving steps, and information about their resulting programs.

Although a majority of the participants in this study never used Scratch prior to this summer camp, their experiences throughout the duration of the camp enabled them to ponder intelligently about the Scratch environment and offer some suggested changes for improvement. Some of these suggestions included: 1) creating a path for beginner, intermediate, and 2) advanced students upon login to help Scratchers sift through the millions of projects currently online, to provide more readily available guide and tutorials that also match these levels of expertise. Additionally, suggestions were given to improve the structure and format of the Scratch Booklet, many of which may even help learning with reading challenges. The design and functional suggestions provided a glimpse into participant's ability to design, think computationally, and problem solve in the context of programming and learning to program.

Black Academic Identity (BAI). An additional purpose of this study was to determine participants' Black Academic Identity and to measure the extent to which this summer programming camp experience impacted it, if at all. The theory and expected connection here is that the more a learning context is innately aligned with African American student culture, the more African American students will psychologically connect high achievement with their race and be highly motivated to achieve and align their behavior as such. Most often, participants in both learning groups reported recognizing no connection between their academic identity and level of performance, their performance using Scratch, and their identity as Black youth. However, their mean BAI scores revealed that they make a connection between academic performance and Black identity to some degree. Pre- and post-camp mean scores fluctuated within learning group, gender and pair-type. Some experienced increases where others experienced decreases. Overall, however, these changes were extremely minimal, suggesting this summer camp experience had no overall impact on participants' Black Academic Identity. It is important to note that one existed, nonetheless.

Computer science education research methods. Exploring the use of a mixed-methods research approach was provided another purpose for this study. The CSEd research literature posits that not many studies on the teaching and learning of computer science and more specifically on computer programming use rigorous social science/educational research methodology. For this reason, pedagogical theories and educational research methods were explored to determine utility and feasibility, namely culturally relevant pedagogy (Communalism) and critical incident technique using a convergent parallel mixed method approach. As a result, there were many types of data collected, which produced descriptions of not only the resulting programming performance but also regarding the processes undertaken to learn how to program throughout each day of camp.

Limitations

Although patterns emerged from and contributions to the field of culturally relevant and CSEd research were identified, this section shares a few limitations of this study recognized by the researcher. These limitations mainly involve the structure of study, the selection of participants, and the research design.

Structure and timing. Regarding structure of the study, one week (five week days, three hours per day) is not long enough to effectively teach computational thinking and programming skills using the Scratch programming language. Although Scratch was originally designed to be learned at the pace of the student, the growing importance of the teaching, learning, and use of computational thinking skills call for more structured learning approaches. As a result, the expected outcome is that students master as much of the material as possible. With this, a camp lasting 5 week days, 3 hours per day is not sufficient. Additionally, the last minute computer lab change and the inability to find a research assistant to consistently help for two weeks presented a behavior management challenge for the study. As a result, the researcher believes the misbehavior of some of the participants impacted the results of the study, especially since some misbehavior resulted in change pair assignments. Even more, structuring the study under the guise of a summer camp instead of during school or during the school year in an afterschool program could have deterred students from focusing on learning the material. Instead the notion of summer camp, fun, and game playing was the expected norm. The researcher believes this my have limited participants' focus on learning and retaining the material even more.

Participant selection. Many communal learning research studies are conducted in formal learning environments where students already know one another. Here, in this study, students met for the first time. The researcher believes that students just meeting one another for the first time and enjoying the process of getting to know one another provided a small distraction to learning for many students, especially those paired in the communal learning group. Even more, since communal learning have typically been conducted in formal learning environments, most if not all participants are already familiar with and are accustomed to traveling to these environments. This study was conducted on a college campus, where the lack of familiarity with the environment initially caused anxiety with the parents and the lack of easy transportation hindered some students from participating.

Sample size. Additionally, limited statistical power because of the modest sample size in the present study (n = 42) may have played a role in limiting the significance of some of the statistical comparisons conducted. A post hoc power analysis revealed that to get a medium effect size (d = .5), an n of approximately 176 participants (88 in each group) would be needed to obtain statistical power at the recommended 0.80 level (Cohen, 1988).

Research design. Another potential limitation regarding the research design was studying elementary school student participants simultaneously with middle school student participants, especially since both groups used the same Scratch Booklet. As such, developmentally appropriate material (i.e. reading level) was not used for each group and could have caused a lack of engagement for some as a result. Moreover, the Scratch Booklet and the pre- and post- Scratch Content Knowledge questionnaire was unbalanced in the amount of pages and questions dedicated to each concept. This unbalance potentially provided a limitation regarding the amount of material presented and tested on any of the core Scratch computational thinking and programming skills. These limitations will be reconsidered in future work.

Future Work

The researcher intends to continue this line of inquiry regarding culturally relevant pedagogy with those underrepresented in computer science, mainly African American students, the teaching and learning of computational thinking and programming skills, the creation and evaluation of curriculum and instructional materials and activities, along with the resulting impact of learning these skills on behavior and identity. As such, I wish to embark on the following projects in the near future.

Study improvements. For this reason, the researcher intends to explore longer studies, a balance in the amount of instructional material and assessment questions per core Scratch computational thinking and programming skill, acquiring more research assistants to help with study implementation and facilitation. Regarding identity exploration, the research would also like to more formally explore instruments that measure attitudes about computer science.

Scratch CT scale creation and validation. This study helped to highlight the need for an authentic Scratch CT and programming assessment scale. In the future, the researcher plans to explore the creation of such a scale such that there is a balance of questions between all six CT elements and that each element can stand alone as its own construct in addition to all elements serving as one overall validated assessment.

Culturally relevant instructional Scratch book. The researcher would also like to write and publish a Scratch book containing instructional strategies, learner activities, and assessments focused on culturally relevant computing and culturally relevant pedagogy.

Computational thinking community identity scale. The researcher would like to collaborate with others interested in creating and validating a CT scale that measures identity, efficacy, sense of belonging to the computing sciences community from a culturally relevant computing perspective.

Expand dual common model of problem-solving and program development. Another near future endeavor is to create a framework for problem-solving and programming that includes all of the CT assessment categories created by the Scratch Team and PACT regarding computing, computational thinking, and communicating, along with behavior, motivation, and persistence.

Conclusion

Taken together, this work presented here contributes to culturally relevant educational research that highlights the enhanced performance and academic benefits of African American youth working in learning environments which support the nine elements of the Black Cultural Ethos. More specifically, implementing communal learning groups as an instantiation of Communalism aligns the learning environment of an African American child with his or her culture and therefore innately enables enhanced performance. Here, even though not statistically significant in this particular study, the Scratch content knowledge mean score of the communal learning group perform was higher than the individual learning group on all programming tasks. This study also uncovered a pattern where the one boy & one girl pair-type within a communal learning environment performed best of all pair types and participants working individually.

This study also integrates the culturally relevant research, namely Communalism and Black Academic Identity and integrated them with research into the teaching and learning of computational thinking and programming skills.

Additionally, the results of this study contribute to the CSEd research literature in three ways: a) it provides an example of a CSEd research study where 100% of the participants identify as being African-American or Latino-American and who are underrepresented in computer science fields of work and study, b) it uses culturally relevant pedagogy to teach computational thinking and programming skills to underrepresented populations in computer science, and c) it uses strategic educational and social science research methods when exploring the teaching and learning of computer programming. Thereby providing CSEd Researchers with examples of more potential tools to use, especially when teaching and working with students who are underrepresented in computer science.

Appendix A

IRB Approval Letter

Office of Research Integrity and Assurance

Research Hall, 4400 University Drive, MS 6D5, Fairfax, Virginia 22030
Phone: 703-993-5445; Fax: 703-993-9590

DATE:	April 17, 2015
TO:	Kevin Clark
FROM:	George Mason University IRB
Project Title:	[605607-2] Understanding How Students Learn Programming and Computational Thinking Skills
SUBMISSION TYPE:	Continuing Review/Progress Report
ACTION:	APPROVED
APPROVAL DATE:	April 17, 2015
EXPIRATION DATE:	April 16, 2016
REVIEW TYPE:	Expedited Review
REVIEW TYPE:	Expedited review categories 6, 7

Thank you for your submission of Continuing Review/Progress Report materials for this project. The George Mason University IRB has APPROVED your submission. This submission has received Expedited Review based on applicable federal regulations.

Please remember that all research must be conducted as described in the submitted materials.

Please remember that informed consent is a process beginning with a description of the project and insurance of participant understanding followed by a signed consent form. Informed consent must continue throughout the project via a dialogue between the researcher and research participant. Federal regulations require that each participant receives a copy of the consent document.

Please note that any revision to previously approved materials must be approved by the IRB prior to initiation. Please use the appropriate revision forms for this procedure.

All UNANTICIPATED PROBLEMS involving risks to subjects or others and SERIOUS and UNEXPECTED adverse events must be reported promptly to the Office of Research Integrity & Assurance (ORIA). Please use the appropriate reporting forms for this procedure. All FDA and sponsor reporting requirements should also be followed (if applicable).

All NON-COMPLIANCE issues or COMPLAINTS regarding this project must be reported promptly to the ORIA.

The anniversary date of this study is April 16, 2016. This project requires continuing review by this committee on an annual basis. You may not collect date beyond this date without prior IRB approval. A continuing review form must be completed and submitted to the ORIA at least 30 days prior to the

anniversary date or upon completion of this project. Prior to the anniversary date, the ORIA will send you a reminder regarding continuing review procedures.

Please note that all research records must be retained for a minimum of five years, or as described in your submission, after the completion of the project.

If you have any questions, please contact Bess Dieffenbach at 703-993-5593 or edieffen@gmu.edu. Please include your project title and reference number in all correspondence with this committee.

This letter has been electronically signed in accordance with all applicable regulations, and a copy is retained within George Mason University IRB's records.

Appendix B

Consent & Assent Forms

Understanding How Students Learn Programming & Computational Thinking Skills

PARENTAL INFORMED CONSENT

RESEARCH PROCEDURES

This research is being done to study how (the process of) young people learn computer programming and computational thinking skills. If you agree with having your child participate, he/she will be observed throughout the course in order to help me understand how children learn to program. Students will also be observed to see how they interact with other students and teachers while they learn how to program and create their own programming projects. The complexity of designs changes over time will also be observed. Observations will occur in real-time by the researcher in the room along with the use of video recording equipment (microphones and cameras), where students will be recorded for later playback, viewing, and analysis. In addition to observation, students will be asked to participate in some interviews (15 minutes each), complete brief questionnaires (5-10 minutes each), and complete a series of pre- and post- surveys and assessments (totally no more than 120 minutes, depending on speed of child). Taken together, these activities along with the actual lessons and class activities should not last longer than 25 hours total.

RISKS
There are no foreseeable risks.

BENEFITS

There are no direct benefits. However, your child's participation may help to further research in learning and technology.

CONFIDENTIALITY

The data in this study will be confidential. Data and video & audio recordings will be collected through online questionnaires, observations, and interviews. Your child's names and other matching identifiers will not be used in the data analysis. All data will be stored in 3 locations – the student researcher's computer, an external hard drive for backup, and a computer located on George Mason University's campus. Data storage will be password protected and only Leshell Hatley, the student researcher, and Dr. Kevin Clark, the principal investigator, will have access to the data and the passwords used. This data will be stored forever for future longitudinal research studies and comparisons. While it is understood that no computer transmission can be perfectly secure, reasonable efforts will be made to protect the confidentiality of your transmission. Unique identification numbers will be used throughout data collection, analysis, and reporting.

PARTICIPATION
Your child's participation is voluntary, and you or your child may withdraw from the study at any time and for any reason. If your child decides not to participate or if your child withdraws from the study, there is no penalty or loss of benefits to which you are otherwise entitled. If this is the case, your child can still complete all course activities without being recorded and his/her data will not be used in the study. There are no costs to you or any other party.

CONTACT

This research is being conducted by Kevin Clark or Leshell Hatley of the Instructional Technology program at George Mason University. Kevin may be reached at (703) 993-3669 and Leshell may be reached at 202.758.2005 for questions or to report a research-related problem. You may contact the George Mason University Office of Research Integrity & Assurance at 703-993-4121 if you have questions or comments regarding your rights as a participant in the research.

This research has been reviewed according to George Mason University procedures governing your participation in this research.

CONSENT

___I have read this form and agree to my child's participation in this study.

___I agree to have my child video recorded.

___I agree to have my child audio recorded.

____
Name of Child

____
Parent's Signature   Date of Signature

Understanding How Students Learn Programming & Computational Thinking Skills

YOUTH INFORMED ASSENT

RESEARCH PROCEDURES

This purpose of this project is to study how young people learn computer programming and computational thinking skills. If you agree to help, you will wear a microphone and will be video and audio recorded as you learning how to program.

These recordings will help me see how you interact with other students and teachers while you learn how to program and create your own projects. You will be asked several questions throughout the project. You will answer some of them on the computer and you will simply say other answer into a microphone. Your answers will help us measure how much you learn. It should take no more than 25 hours for you to learn how to program and to answer all the questions.

RISKS
There are no risks to you.

BENEFITS

If you help with this project, you may assist me in understanding how people learn:

1. About computer programming and computational thinking skills,
2. What makes people want to use computers,
3. and what makes people not want to use computers.

CONFIDENTIALITY

The information and recordings in this project will be kept secret. Your names and other information about you will not be shared with others. This information will be kept locked with passwords and keys in 3 places – the student researcher's computer, an external hard drive for backup, and a computer located on George Mason University's campus. Only Leshell Hatley, the student researcher, and Dr. Kevin Clark, the principal investigator, will have access to the information, the keys, and the passwords used. This information will be stored forever for future research projects.

PARTICIPATION

You may stop being recorded at any time and for any reason. If you decide to stop that is okay, you can still complete all class activities, but you will not be recorded and your data will not be used in the study. You don't have to pay anything to be part of this project.

CONTACT

My name is Leshell Hatley, and I am studying Learning Technologies Design Research at George Mason University. You can call me at this phone number (301) 736-2379 if you have any questions about this study. You can also call my teacher, Dr. Kevin Clark at George Mason University, at this phone number (703) 993-3669. The George Mason University Office of Research Integrity & Assurance knows all about my research and said that it was OK for me to do it. You can call them at 703-993-4121 if you have any questions about being a part of this research.

ASSENT

___I have read this form and agree to participate in this study.

___I agree to be video recorded.

___I agree to be audio recorded.

____
Name of Youth

____
Youth's Signature

____
Date of Signature

Appendix C

SG

Shuchi Grover <shuchig@cs.stanford.edu>

To: Leshell Hatley, GMU;

Reply all

Thu 8/13/2015 12:13 AM

Inbox

You replied on 8/13/2015 12:37 AM.

Thanks for your patience, Leshell.

Sure, you may use the pre- and post- tests and questions from the computer experience survey and I'd appreciate appropriate citations to my dissertation and/or the journal article (Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, *25*(2), 199-237.)

My concerns and questions were mainly around the use of the video materials and content that you'd mentioned earlier. My advisors suggested I chat with you about that as permission for the use of that cannot be given at this point.

Best of luck with your dissertation, and I'm happy to continue to find a time to chat although this is a busy month, unfortunately.

Regards,
Shuchi

Appendix D

Pre-Camp Academic and Computer Experience Questionnaire

BASIC INFORMATION

Q1 Name

Q2 Age

Q3 Gender

Q4 Grade

Q5. Have you ever written a computer program?

Q7 How did you get into this summer camp?

“FUTURE SELF” QUESTIONS

Q8. In the future, can you see yourself . . .

	Definitely No	Probably No	Probably Yes	Definitely Yes
Taking more classes about computers or computer science?				
Becoming a computer programmer or engineer of some sort?	mm	mm	mm	mm
Becoming a graphic designer or Web designer?	mm	mm	mm	mm
Becoming a computer or network specialist?	mm	mm	mm	mm
Becoming a computer or technology teacher?	mm	mm	mm	mm
Becoming a computer game designer?	mm	mm	mm	mm
Becoming an app developer?	mm	mm	mm	mm
Becoming a computer scientist?	mm	mm	mm	mm
Becoming a scientist?	mm	mm	mm	mm
Becoming a teacher?	mm	mm	mm	mm

Becoming a doctor or nurse?	mm	mm	mm	mm
Becoming an artist?	mm	mm	mm	mm

Becoming a designer?	mm	mm	mm	mm
Starting a business?	mm	mm	mm	mm

Q9 Please describe your ideal job for the future (in one sentence):

Q10 What are currently your TOP 3 favorite subjects in school?

Q11 How MANY TIMES do you use a computer (anywhere) to do each of the following:

	Never	Less than once a mont h	Once a mont h	A few times a mont	Once a week	A few times a week	Dail y	Sever al times a day
Play games (on the computer, online or on a game console)	mm	mm	mm	mm	mm	mm	mm	mm
Participate in multi-user online games	mm	mm	mm	mm	mm	mm	mm	mm
Work on your own digital media projects outside of school assignments?	mm	mm	mm	mm	mm	mm	mm	mm
Conduct research on the Internet for school	mm	mm	mm	mm	mm	mm	mm	mm
Collect/view/organize e images or music (e.g. put your photos, images, or sounds	mm	mm	mm	mm	mm	mm	mm	mm

from the Web into folders).								
Write for fun	mm	mm	mm	mm	mm	mm	mm	mm
Read or send email	mm	mm	mm	mm	mm	mm	mm	mm
Read comics (e.g. Manga)	mm	mm	mm	mm	mm	mm	mm	mm
Do some artwork	mm	mm	mm	mm	mm	mm	mm	mm
Doing homework, checking grades	mm	mm	mm	mm	mm	mm	mm	mm
Watching movies and online music videos	mm	mm	mm	mm	mm	mm	mm	mm
Take online courses in science/math/other	mm	mm	mm	mm	mm	mm	mm	mm
Watch online academic videos and lectures (e.g. Khan Academy)	mm	mm	mm	mm	mm	mm	mm	mm
Social networking (e.g. facebook)	mm	mm	mm	mm	mm	mm	mm	mm
Do computer programming	mm	mm	mm	mm	mm	mm	mm	mm

Q12 How MANY TIMES have you EVER CREATED the following using some software on the computer?

	0 times	1-2 times	3-6 times	6+ times
Created a multimedia presentation (e.g. PowerPoint)				
Written computer program (code) using a computer language (e.g. LOGO, Java, Python, C++)			mm	mm

Made computer creations using Scratch or Alice or Tynker (block-based programming)	mm	mm	mm	mm
Created a Web site using HTML	mm	mm	mm	mm
Created an app for iPhone or Android	mm	mm	mm	mm
Created a piece of art using a software application (e.g. PhotoShop, Illustrator)	mm	mm	mm	mm
Built a robot or other invention of any kind using electronics and technology	mm	mm	mm	mm
Created a digital movie (e.g. iMovie or MovieMaker)	mm	mm	mm	mm
Created an animation (e.g. Flash, Alice, Scratch)				
Created a computer or video game (e.g. Stagecast, GameStar, Scratch, Kodu)	mm	mm	mm	mm
Created a piece				

of music (e.g. GarageBand, FruityLoops)	mm	mm	mm	mm
Created a spreadsheet, graph, or chart (e.g. Excel)	mm	mm	mm	mm

Q13 How would you describe your level of experience with the following computer applications/equipment?

	I don't know w what this is	I have no experience but I have heard of it	I've played around with it	I have used it to make somethin g	I'm an expert and can teach someon
Flash	mm	mm	mm	mm	mm
Photoshop/Fireworks/Illustrato	mm	mm	mm	mm	mm
Scratch/Tynker	mm	mm	mm	mm	mm
Alice	mm	mm	mm	mm	mm
LOGO	mm	mm	mm	mm	mm
MIT App Inventor	mm	mm	mm	mm	mm
Java programming	mm	mm	mm	mm	mm
Python programming	mm	mm	mm	mm	mm
Javascript programming	mm	mm	mm	mm	mm
HTML/XML	mm	mm	mm	mm	mm
iPhone SDK/Objective C	mm	mm	mm	mm	mm
GameStar Mechanic	mm	mm	mm	mm	mm
Kodu	mm	mm	mm	mm	mm
FruityLoops/Audacity/GarageBan d	mm	mm	mm	mm	mm
iMovie/MS MovieMaker	mm	mm	mm	mm	mm
Arduino	mm	mm	mm	mm	mm
Lego Mindstorms	mm	mm	mm	mm	mm
Microsoft Word/Powerpoint	mm	mm	mm	mm	mm

C or C++ programming	mm	mm	mm	mm	mm

Q14 How often do you use a computer in the following places :

	Never	Less than Once a	Once a Month	2-3 Times a	Once a Week	2-3 Times a Week	Daily	Several Times a Day
At home	mm	mm	mm	mm	mm	mm	mm	mm
At school during class	mm	mm	mm	mm	mm	mm	mm	mm
At school on your own time	mm	mm	mm	mm	mm	mm	mm	mm
At a relative's house	mm	mm	mm	mm	mm	mm	mm	mm
In an after school program / club	mm	mm	mm	mm	mm	mm	mm	mm
At a friend's house	mm	mm	mm	mm	mm	mm	mm	mm
At the library	mm	mm	mm	mm	mm	mm	mm	mm

Q15 How many classes/workshops/camps have you participated in before this class for the following-- Scratch, Snap, Alice, Tynker, LOGO, Robotics? [Write "learned at home" or "learned on my own" if you have never learned these formally, but have picked
nni90ooolllplplppllll;them up at home or on your own]

Scratch

Snap

Tynker
Alice
LOGO
Robotics

Appendix E

Computational Thinking and Programming Pre- & Post-Test

Camp ID:

computational thinking VOCABULARY

Here is a list of concepts. Write a short explanation of each one. If the concept is not familiar, write X.

algorithm
Initialization
variable
input/output
loop
conditional
boolean variable

Sequence of Instructions-2

Here is a sequence of instructions:

1. Stand at the origin
2. Turn left
3. Carry out 10 times:
    3.1 Move 5 steps
4. Turn right
5. Carry out 10 times:
    5.1 Move 5 steps
6. Turn right
7. Carry out 10 times:
    7.1 Move 5 steps

(a) If you carry out these instructions, you will follow a path that is the form of some letter in the English alphabet. What is it?

Scratch-1

What is the above an example of?

Conditional execution
Handling an event
Sending a message
Loop
Variable assignment

Scratch-2

How many times will 

be executed above?

mm25

mm20

mm100

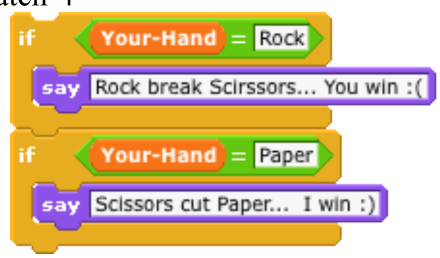
mm10

mmI don't know

Scratch-3

What is the above an example of?
mmConditional execution
mmHandling
an event mmSending a
message mmLoop
mmVariable assignment

Scratch-4



What is the above an example of?
mmConditional execution
mmHandling
an event mmSending a
message mmLoop
mmVariable assignment

Scratch-5

What is this an example of?

mmConditional execution

mmHandling
an event mmSending a
message mmLoop
mmVariable assignment

Scratch-6

What does the following code do?

What does the following code do?

mmRepeat a simple animation mmDraw a square using a pen mmMake a ball fall

mmIncrement the score

mmStamp the current costume at the current mouse location

Scratch-7

What does the following code do?

What does the following code do?

mmRepeat a simple animation

mmDraw a square

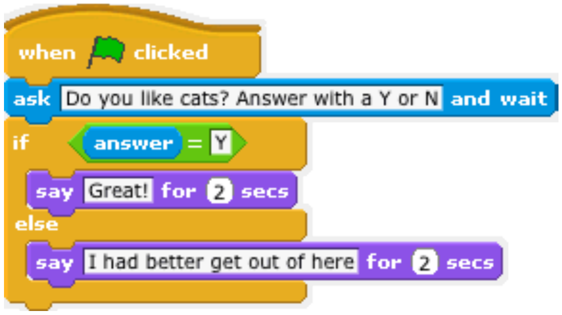
mmMake a ball fall

mmIncrement the score

mmStamp the current costume at the current mouse location

Scratch-8

What will be said when the following executes and the user answers with No?



mmGreat!

mmI had better get out of here

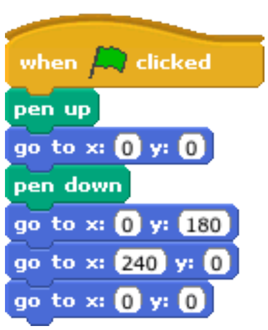
mmIt won't say anything

mmI don't know

mmYou will get an error message

Scratch-9

What will be the result of executing the following script? The coordinates of the center of the stage are x=0; y=0.



Scratch-10

(1) Describe in plain English what does the program above does. What is the goal of the program?

(2) What is the term for what the 3 orange blocks in the beginning are doing?

(3) There are 3 yellow blocks in this image above. The FIRST is the Run block (with the green flag). What is the SECOND yellow block doing? What is the term used to describe such a block?

(4) What is the THIRD (last) yellow block in the code doing? What is the term used to describe such a block?

Appendix F

Learner Context Questionnaire-modified (LCQ-m)

This questionnaire is designed to assess attitudes towards different learning situations. The items on this questionnaire each consist of a single statement and a four-point scale is provided for each statement. Using this scale, please respond to each statement by circling the number that best represents the degree of truth or falseness that statement has for you.

(Cooperative Portion)

(1) I like to help other students learn.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(2) I like to share my ideas and materials with other students.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(3) I like to cooperate with other students.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(4) I can learn important things from other students.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(5) I try to share my ideas and materials with other students when I think it will help them.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(6) Students learn lots of important things from each other.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(7) It's a good idea for students to help each other learn.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(Individualistic portion)

(8) I don't like working with other students.
I like to work with other students. (reverse)

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(9) It bothers me when I have to work with other students.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(10) I do better work when I work alone.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(11) I like work better when I do it all myself.

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

(12) I would rather work on school work alone than with other students.
Working in small groups is better than working alone. (reverse)

Not at all (1) like me	Not much (2) like me	Somewhat (3) like me	Very much (4) like me

Appendix G

Black Academic Identity Scale

Directions: Here are some questions about how you see yourself as a student. Please read each question carefully and select the answer that is most true for you. Items are measured on a 5-point likert scale ranging from 1= "not at all true" to 5 = "very true."

Black Academic Identity

1. I think of myself as a Black student, not just a student.
2. Being a good Black student is an important part of who I am
3. I get good grades because I am Black.
4. I seldom think of myself as a Black student.

Black Model Phenomenon

1. I want to be an example of Black success.
2. I want to represent Black students in a positive way.
3. I want to be a Black role model.
4. I want to show others that Black students are smart.
5. Getting good grades in school is the best way to prove society wrong about Black people.
6. It is a burden to prove to other that I am a smart Black student. (reverse)

Appendix H

Opinon Prompts

1. Enter your camp number.
2. Scratch username
3. First and Last Name
4. Which unit did you just complete?
5. What did you learn in this unit?
6. Did you ever read or learn anything about the topic(s) in this unit before coming to this camp?
7. If you answered 'yes' or 'a little' to the previous question, can you describe how much and where you read or learned something about the topic(s) in this unit?
8. What did you think was important in this unit and why?
9. What did you find interesting in this unit?
10. What did you find challenging or hard in this unit?
11. Describe which booklet exercises in this unit helped you the most and tell why?
12. How has this unit changed the way you think about computer programming?
13. Did you make the best use of your time to learn during this unit?
14. Describe your team's strategy/approach to learn the material in this unit?
15. How could you improve this strategy/approach to learning for future units?

Appendix I

Scratch Interview Protocol

1. So, what is Scratch? Describe it in your own words.
2. What do you do with it? [Only if this part is not in the answer above.]
3. How did you get the idea for the project you created today? Scratch or remix?
4. Have you thought of any of your own ideas for Scratch projects? If so, what are they? How did you get started with your Scratch projects?
5. What did you do when you got stuck in Scratch?
7. Have you used any of the Scratch community features? If so, which ones?
8. What do you like about Scratch?
9. What do you dislike about Scratch?
10. What would you change or add to Scratch?
11. How did you like working with a partner? (For Communal Learning group only)
12. How did you like reading and learning from of the booklet? What do you dislike about reading and learning from the booklet?
13. You've worked alone learning Scratch this week. How do you like it? (For Individual Learning Group only)
14. Would you prefer to work alone or work with a partner?
13. Is there a connection between your academic performance and how you'd like to perform in Scratch? Is there a connection between being a young black <male/female> to doing well in Scratch?
15. Overall, what do you think about learning to program?
16. This week have you logged into Scratch?

Appendix J

Cognitive Assessment of Students' Problem-Solving and Program Development Skills

The Process - Formulating the Problem		
Outcome	**Indicator**	**Scoring Scale**
Excellent representation of problem and complete identification of relevant facts, indicating full understanding, required to solve the problem.	Problem is clearly and correctly stated. All goals, givens, and unknowns are identified.	4
Reasonable representation of problem and identification of almost all relevant facts, indicating adequate understanding, required to solve the problem.	Problem is correctly stated. Most goals, givens, and unknowns are identified.	3
Incomplete representation of problem and/or identification of facts, indicating some understanding, but not enough to solve the problem.	Problem is partially stated and/or some facts are identified.	2
Inappropriate representation of problem and inability to identify relevant facts, indicating complete misunderstanding, required to solve the problem.	Problem statement is incorrect and meaningless facts are identified.	1
Lack of problem representation and identification of relevant facts, indicating complete misunderstanding, required to solve the problem.	No problem representation/fact identification attempted or completely irrelevant work.	0

Table 1. Instrument for assessing students' problem formulation skills.

The Process - Planning the Solution		
Outcome	**Indicator**	**Scoring Scale**
Excellent planning strategy and refinement of goals that will lead to a correct solution for the problem.	Detailed and clear planning. Complete goal refinement , task identification, and data representation.	4
Reasonable planning strategy and refinement of goals that could lead to a correct solution for the problem.	Adequate planning. Sufficient goal refinement, task identification, and data representation.	3
Incomplete planning strategy and/or some evidence of goal refinement, but not enough to solve the problem.	Partially correct planning and/or some goal refinement, task identification, and data representation.	2
Inappropriate planning strategy and complete lack of adequate goal refinement necessary to solve the problem.	Incorrect planning and meaningless goal refinement.	1
Lack of planning and refinement necessary to solve the problem.	No planning/refinement attempted or completely irrelevant work.	0

Table 2. Instrument for assessing students' planning skills.

The Process - Designing the Solution		
Outcome	**Indicator**	**Scoring Scale**
Excellent design strategy and module specifications that will lead to a good quality solution for the problem.	Complete module decomposition, organization, and detailed specifications.	4
Reasonable design strategy and module specifications that could lead to a solution for the problem.	Sufficient module decomposition, organization, and sufficient specifications.	3
Incomplete design strategy and/or some evidence of module specifications, but not enough to solve the problem.	Partial design and/or some module specifications.	2
Inappropriate design strategy and specifications that will not lead lead to a solution for the problem.	Improper module decomposition, organization, and specifications.	1
Lack of design and specifications necessary to solve the problem.	No design/specifications attempted or completely irrelevant work.	0

Table 3. Instrument for assessing students' design skills.

The Product - Solution Quality: Efficiency		
Outcome	**Indicator**	**Scoring Scale**
Well suited solution is produced.	Most appropriate algorithms, data structures, control structures, and language constructs for this problem situation are chosen.	2
Minimally acceptable solution is produced.	Program accomplishes its task, but lacks coherence in choice of either data and/or control structures.	1
Unacceptable solution quality is produced.	Program solution lacks coherence in choice of both data and control structures.	0

Table 4. Instrument for assessing solution efficiency.

The Product - Solution Quality: Reliability		
Outcome	**Indicator**	**Scoring Scale**
Robust solution is produced.	Program functions properly under all test cases. Works for all valid input, and responds to all invalid input.	2
Minimum requirement solution is produced.	Program functions under limited test cases or works only for valid input and fails to respond to invalid input.	1
Unacceptable solution quality is produced.	Program fails under most test cases.	0

Table 5. Instrument for assessing solution reliability.

The Product - Solution Quality: Readability		
Outcome	**Indicator**	**Scoring Scale**
Clear and understandable solution is produced.	Program includes commented code, meaningful identifiers, indentation to clarify logical structure, and user instructions.	2
Minimally documented solution is produced.	Program lacks clear documentation and/or user instructions.	1
Unacceptable solution quality is produced.	Program is totally incoherent.	0

Table 6. Instrument for assessing solution readability.

The Product - Solution Correctness		
Outcome	**Indicator**	**Scoring Scale**
Appropriate solution is produced.	Correct solution specifications, program code and results consistent with problem requirements.	2
Incomplete solution is produced.	Partial solution specifications/program code and/or some results.	1
No solution or totally inappropriate solution is produced.	No solution specifications/ program code, or results inconsistent with problem requirement.	0

Table 7. Instrument for assessing solution correctness.

Appendix K

Attribution-NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0)
Notice and License

Computer Science Concepts in Scratch

Michal Armoni and Moti Ben-Ari

This book will familiarize you with the Scratch visual programming environment, focusing on using Scratch to learn *computer science*. The book is structured as a collection of *tasks*. Each chapter teaches a new concept, but the concept is introduced in order to solve a specific task such as animating dancing images or building a game. Each chapter starts with a simple task, but as soon as we solve one task, we add additional tasks to extend the existing task. The sequence of tasks will require a new construct of Scratch or the use of constructs you know in new ways.

The textbook was written for Scratch 1.4. We have also written a supplement that explains the changes and additional features in Scratch 2.0.

Download

The textbook is available in three formats: (a) with equal margins for screen display and one-sided printing; (b) with margins for two-sided printing and binding in the left margin; (c) with a large font on a small text area that enables better accessibility by magnifying the pdf. The layout was carefully done for the full-size pages and will be suboptimal in the small format; we have no plans at this time to improve this format or to produce other formats for ereaders.

- Textbook for Scratch 1.4—version 1.0, 5 May 2013 (one-sided, two-sided, small format).
- Supplement for Scratch 2.0—version 1.0, 23 May 2013 (one-sided, two-sided, small format).
- Scratch projects accompanying the textbook.
- Costumes for Scratch projects accompanying the textbook.
  This is to enable the instructor to give the students the costumes without revealing the source code of the projects.

Our other learning materials for Scratch

Moti has created many Scratch projects that can be used as learning materials:

- Implementations of the activities of Computer Science Unplugged.
- Projects on robotics: implementations of Karel the Robot and Braintenberg's vehicles, projects for the LEDO WeDO kit, and simulations of the Thymio education robot.
- A game for learning mathematics.

You can find the Scratch projects on the MIT Scratch website or on his local Scratch page.

Research publications

- O. Meerbaum-Salant, M. Armoni, M. Ben-Ari. Habits of programming in Scratch. *Sixteenth Conference on Innovation and Technology in Computer Science Education*, Darmstadt, Germany, 2011, 168--172.
- O. Meerbaum-Salant, M. Armoni, M. Ben-Ari. Learning computer science concepts with Scratch. *Computer Science Education*, 23(3), 2013, 239--264.
- M. Armoni, O. Meerbaum-Salant, M. Ben-Ari. From Scratch to "Real" Programming. *ACM Transactions on Computing Education*, 14(4), article 25, 2015.

The questionnaires used in the research can be downloaded from here.

Hebrew language website: http://stwww.weizmann.ac.il/g-cs/scratch/index.html.

Contact: Please send comments and suggestions to: michal.armoni@weizmann.ac.il.

References

Adams, J., (1995). *Risk*. London, UK; UCL Press.

Adams, J. C., & Webster, A. R. (2012, February). What do students learn about programming from game, music video, and storytelling projects?. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 643-648). ACM.

Anderson, A. S. & Freeman, K. E., (2010). *The black academic identity scale*. Washington, DC: Howard University.

Armoni, M., Ben-Ari, M. (2013). *Computer science concepts in scratch.* Rehovot, Israel. Weizmann Institute of Science. Retrieved from http://stwww.weizmann.ac.il/g-cs/Scratch/Scratch_en.html

Albury, A. (1991). *Social orientations, learning conditions and learning outcomes among low-income Black and White grade school children*. Unpublished doctoral dissertation, Howard University, Washington, DC.

Barba, R. H. (1993). A study of culturally syntonic variables in the bilingual/bicultural science classroom. *Journal of Research in Science Teaching, 30(*9*),* 1053-1071.

Barr, V. & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2(*1*),* 48–54. http://doi.org/10.1145/1929887.1929905

Blikstein, P., & Wilensky, U. (2006). "The missing link: a case study of sensing-and-modeling toolkits for constructionist scientific investigation*," In Sixth International Conference on Advanced Learning Technologies*, *(ICALT'06)*, Kerkrade, 2006, pp. 980-982. http://doi.org/10.1109/ICALT.2006.1652608

Boykin, A. W. (1977). Experimental psychology from a Black perspective: Issues and examples. *Journal of Black Psychology*, *3*(2), 29-49.

Boykin, A. W. (1986). The triple quandary and the schooling of Afro-American children. In U. Neisser (Ed.), *The school achievement of minority children: New perspectives* (pp. 57-92). New York, NY: Routledge.

Boykin, A.W. (1994a). Afro-cultural expressions and its implications for schooling. In E. Hollins, J. King, & W. Hayman (Eds.), *Teaching diverse populations: Formulating a knowledge base* (pp. 243-256). New York, NY: Suny Press.

Boykin, A.W. (1994b). Harvesting culture and talent: African American children and school reform. In R. Rossi (Ed.), *Schools and students at risk: Context and framework for positive change* (pp. 116-138). New York: Teachers College Press.

Boykin, A. W., Jagers, R. J., Ellison, C. M., & Albury, A. (1997). Communalism: conceptualization and measurement of an afrocultural social orientation. *Journal of Black Studies*, *27*(3), 409-418.

Brennan, K., & Resnick, M. (2012, April). *Using artifact-based interviews to study the development of computational thinking in interactive media design*. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.

Brooks, F. P. (1975). *The mythical man-month*. Reading, MA: Addison-Wesley.

Brooks, R. (1983). Towards a theory of the comprehension of computer programs. *International Journal of Man-Machine Studies, 18(6),* 543–554. http://doi.org/10.1016/S0020-7373(83)80031-5

Brown, B. A., Kloser, M., & Henderson, J. B. (2010, April). *Building bridges towards cognition: Cultural Continuity and the Language-Identity Dilemma.* Paper presented at annual American Educational Research Association meeting, Denver, CO.

Burrell, J. O. (2012). *Cultural learning context as it relates to efficacy and the mathematics performance of African-American middle school students* (Order No. 3591937). Available from ProQuest Dissertations & Theses Global. (1435630814). Retrieved from http://search.proquest.com/docview/1435630814?accountid=14541

Burke, Q., & Kafai, Y. B. (2012). The writers' workshop for youth programmers: digital storytelling with scratch in middle school classrooms. *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 433-438). Raleigh, NC: Association for Computing Machinery.

Campe, S., Werner, L., & Denner, J. (2005). Information technology fluency for middle school girls. *Proceedings from Eighth International Conference on Computers and Advanced Technology for Education (CATE),* Oranjestad, Aruba: Internal Association of Science and Technology for Development.

Center for Research on Evaluation Standards and Student Testing (CRESST). (2004). *Building on children's cultural assets in simulated classroom performance environments. Research Vistas in the Communal Learning Paradigm. Report No. 68*. Baltimore, MD: Boykin, A. W., Coleman, S. T., Lilja, A. J., & Tyler, K.

Center for Research on the Education of Students Placed At Risk (CRESPAR). (2000). *Classroom cultural ecology. The dynamics of classroom life in schools serving low-income African-American children*. Baltimore, MD: Ellison, C. M., Boykin A. W., Towns, D. P., & Stokes, A.

Chang, C.-K. (2014). Effects of using alice and scratch in an introductory programming course for corrective instruction. *Journal of Educational Computing Research, 51*(2), 185–204. http://doi.org/10.2190/EC.51.2.c

Cheung, J. C. Y., Ngai, G., Chan, S. C. F., & Lau, W. W. Y. (2009). Filling the gap in programming instruction: a text-enhanced graphical programming environment for junior high students. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education* (pp. 276–280). New York, NY, USA: ACM. http://doi.org/10.1145/1508865.1508968

Clark, K., Brandt, J., Hopkins, R., & Wilhelm, J. (2009). Making games after-school: participatory game design in non-formal learning environments. *Educational Technology, 49*(6), 40-44.

Clark, K. & Sheridan, K. (2010). Game design through mentoring and collaboration. *Journal of Educational Multimedia and Hypermedia, 19*(2*),* 125-145.

Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, *76*(6), 1051-1058.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). New Jersey, NJ: Lawrence Erlbaum.

Coleman, K. (1996). *The influence of communal learning contexts on African American elementary students' creative problem solving.* Master's thesis, Howard University, Washington, DC.

Coleman, S. (2001). *Communal versus individualistic learning contexts as they relate to mathematical task performance under simulated classroom conditions*. Paper presented at the annual American Educational Research Association meeting, Seattle, WA.

Computer Science Teachers Association (2010). *Running on empty: The failure to teach k-12 computer science in the digital age.* New York, NY: Wilson, C., Sudol, L. A., Stephenson, C., & Stehlik, M.

Computing Research Association. (2015). *2015 Taulbee Survey.* Washington, DC: Zweben, S. & Bizot, B. Retrieved from http://cra.org/wp-content/uploads/2015/06/2014-Taulbee-Survey.pdf

Connell, R. W., & Messerschmidt, J. W. (2005). Hegemonic masculinity rethinking the concept. *Gender & society*, *19*(6), 829-859.

Cooper, S., Grover, S., Guzdial, M., & Simon, B. (2014). A future for computing education research. *Communications of the ACM, 57*(11), 34–36. http://doi.org/10.1145/2668899

Corbin, J., & Strauss, A. (2008). *Basics of qualitative research: techniques and Procedures for developing grounded theory* (3rd ed.). Los Angeles, CA: Sage Publications. doi:10.4135/9781452230153

Corney, M., Teague, D., & Thomas, R. N. (2010). Engaging students in programming. *Proceedings of the Twelfth Australasian Conference on Computing Education* (pp. 63–72). Darlinghurst, Australia: Australian Computer Society, Inc.

Creswell, J. (2009). *Research design: Qualitative, quantitative, and mixed methods approaches* (3rd ed.). Los Angeles, CA: Sage.

Crutchfield, O. S., Harrison, C. D., Haas, G., Garcia, D. D., Humphreys, S. M., Lewis, C. M., & Khooshabeh, P. (2011). Berkeley Foundation for Opportunities in Information Technology: A decade of broadening participation. *ACM Transactions on Computing Education (ToCE)*, *11*(3), 1-24.

Davis, R., Kafai, Y., Vasudevan, V., & Lee, E. (2013, June). The education arcade: crafting, remixing, and playing with controllers for Scratch games. In *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 439-442). ACM.

Dill, E. M., & Boykin, A. W. (2000). The comparative influence of individual, peer tutoring, and communal learning contexts on the text recall of African American children. *Journal of Black Psychology*, *26*(1), 65-78.

Deek, F. P., Starr, R. H., Kimmel, H., & Rotter, N. (1999). Cognitive assessment of students' problem solving and program development skills. *Journal of Engineering Education, 88*(3), 317-326.

Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM, 50*(7), 13–18. http://doi.org/10.1145/1272516.1272529

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?. *Computers & Education*, *58*(1), 240-249.

Dewey, J. (1902). *The child and the curriculum.* Chicago, IL: The University of Chicago Press.

Doerschuk, P., Liu, J., & Mann, J. (2011). INSPIRED high school computing academies. ACM *Transactions on Computing Education (ToCE), 11*(2), 1–18. http://doi.org/10.1145/1993069.1993071

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding?. *Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (pp. 60-69). Berlin, Germany: ACM.

Dunn, R. S., Dunn, K. J., & Price, G. E. (1989). Learning style inventory (LSI). Price Systems, Incorporated.

Edwards, D. (2010). *The lab: Creativity and culture*. Cambridge, MA: Harvard University Press.

Entertainment Software Association (ESA). (2015). "2012 essential facts about the computer and video game industry." The Entertainment Software Association. Available at http://www.theesa.com/wp-content/uploads/2015/04/ESA-Essential-Facts-2015.pdf

Federal Inventory of STEM Education, Fast-Track Action Committee, Committee on STEM Education, & National Science and Technology Council. (2011). The federal science, technology, engineering, and mathematics (stem) education portfolio. Retrieved from https://www.whitehouse.gov/sites/default/files/microsites/ostp/costem__federal_stem_education_portfolio_report.pdf

Feldman, S. (2004). A Conversation with Alan Kay. *Queue*, *2*(9), 20–30. http://doi.org/10.1145/1039511.1039523

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87–97. http://doi.org/10.1016/j.compedu.2012.11.016

Fletcher, G. H. L., & Lu, J. J. (2009). Education: Human computing skills: rethinking the K-12 experience. *Communications of the ACM, 52*(2), 23–25. http://doi.org/10.1145/1461928.1461938

Franklin, D., Conrad, P., Aldana, G., & Hough, S. (2011). Animal tlatoque: attracting middle school students to computing through culturally-relevant themes. *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education* (pp. 453–458). New York, NY: ACM. http://doi.org/10.1145/1953163.1953295

Franklin, D., Conrad, P., Boe, B., Nilsen, K., Hill, C., Len, M., … Waite, R. (2013). Assessment of computer science learning in a scratch-based outreach program. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (pp. 371–376). New York, NY,: ACM. http://doi.org/10.1145/2445196.2445304

Freeman, J., Magerko, B., McKlin, T., Reilly, M., Permar, J., Summers, C., & Fruchter, E. (2014). Engaging underrepresented groups in high school introductory computing through computational remixing with earsketch. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (pp. 85–90). New York, NY,: ACM. http://doi.org/10.1145/2538862.2538906

Fullilove, R. E., & Treisman, P. U. (1990). Mathematics achievement among African American undergraduates at the University of California, Berkeley: An evaluation of the mathematics workshop program. *The Journal of Negro Education*, *59*(3), 463-478.

Gay, G. (2002). Preparing for culturally responsive teaching. *Journal of Teacher Education, 53*(2), 106–16.

Gilbert, J. E. (2006). Making a case for BPC [Broadening Participation in Computing]. *Computer, 39*(3), 83–86. http://doi.org/10.1109/MC.2006.96

Goode, J. (2010). Connecting k-16 curriculum & policy: making computer science engaging, accessible, and hospitable for underrepresented students. *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 22–26). New York, NY: ACM. http://doi.org/10.1145/1734263.1734272

Google. (2015). Searching for computer science: access and barriers in U.S. K-12 education. Mountainview, CA. Retrieved from https://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf.

Gregory, A., Skiba, R. J., & Noguera, P. A. (2010). The achievement gap and the discipline gap two sides of the same coin? *Educational Researcher, 39*(1), 59-68.

Grover, S., Cooper, S., & Pea, R. (2014). Assessing computational learning in k-12. *Proceedings of the 2014 conference on innovation & technology in computer science education* (pp. 57–62). New York, NY: ACM. http://doi.org/10.1145/2591708.2591713

Grover, S., & Pea, R. (2013). Computational thinking in k–12: a review of the state of the field. *Educational Researcher, 42*(1), 38–43. http://doi.org/10.3102/0013189X12463051

Guzdial, M. (2008). Education: paving the way for computational thinking. *Communications of the ACM, 51*(8), 25–27. http://doi.org/10.1145/1378704.1378713

Hale, J. E. (1982). *Black children: Their roots, culture, and learning styles*. Baltimore, MD: JHU Press.

Hanks, B., McDowell, C., Draper, D., & Krnjajic, M. (2004, June). Program quality with pair programming in CS1. *ACM SIGCSE Bulletin 36*(3), 176-180.

Harms, K. J., Cosgrove, D., Gray, S., & Kelleher, C. (2013). Automatically generating tutorials to enable middle school children to learn programming independently. *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 11-19). New York, NY: ACM.

Helsper, E. J., & Eynon, R. (2010). Digital natives: where is the evidence? *British Educational Research Journal, 36*(3), 503-520.

Hill, C., Dwyer, H. A., Martinez, T., Harlow, D., & Franklin, D. (2015). Floors and flexibility: Designing a programming environment for 4th-6th grade classrooms. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 546-551). ACM.

Hu, M., Winikoff, M., & Cranefield, S. (2013). A process for novice programming using goals and plans. *Proceedings of the Fifteenth Australasian Computing Education Conference* (pp. 3–12). Darlinghurst, Australia: Australian Computer Society, Inc.

Hurley, E. (1999). *An ethnographic analysis of culture in group functioning Manifestations of communalism in African American students' cooperative learning group behavior*. Unpublished doctoral dissertation, Howard University, Washington DC.

Hurley, E. A., Boykin, A. W., & Allen, B. A. (2005). Communal versus individual learning of a math-estimation task: African American children and the culture of learning contexts. *The Journal of Psychology*, *139*(6), 513-527.

Jimenez, P. (1912). *Dance research within the context of the work of maud robart* (Doctoral dissertation, University of Hawaii at Manoa).

Johnson, C. C., Peters-Burton, E. E., & Moore, T. J. (2015). *Stem road map: a framework for integrated stem education*. New York, NY: Routledge.

Joy, M., Sinclair, J., Sun, S., Sitthiworachart, J., & López-González, J. (2009). Categorising computer science education research. *Education and Information Technologies*, *14*(2), 105-126.

Kafai, Y. B., Burke, Q., & Resnick, M. (2014). *Connected code: Why children need to learn programming*. Cambridge, MA: MIT Press.

Kafai, Y. B., Searle, K., Kaplan, E., Fields, D., Lee, E., & Lui, D. (2013). Cupcake cushions, scooby doo shirts, and soft boomboxes: e-textiles in high school to promote computational concepts, practices, and perceptions. *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 311–316). New York, NY: ACM. http://doi.org/10.1145/2445196.2445291

Kaplan, B., & Maxwell, J. A. (2005). Qualitative research methods for evaluating computer information systems. In J.G. Anderson, C.E. Aydin, S.J. Jay (Eds.), *Evaluating the organizational impact of healthcare information systems* (pp. 30-55). New York, NY: Springer.

Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, *37*(2), 83-137.

Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling alice motivates middle school girls to learn computer programming. *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1455-1464). San Jose, CA: ACM.

Kolikant, Y. (2012). (Some) grand challenges of computer science education in the digital age: a socio-cultural perspective. *Proceedings of the 7th Workshop in Primary and Secondary Computing Education* (pp. 86–89). New York, NY: ACM. http://doi.org/10.1145/2481449.2481471

Ladson-Billings, G. (1995). But that's just good teaching! The case for culturally relevant pedagogy. *Theory into practice, 34*(3), 159-165.

Lakanen, A.-J., & Isomöttönen, V. (2015). What does it take to do computer programming?: surveying the k-12 students' conceptions. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 458–463). New York, NY: ACM. http://doi.org/10.1145/2676723.2677229

Lee, C. D. (1993). Signifying as a scaffold for literary interpretation: The pedagogical implications of an African American discourse genre (Research Report Series). Urbana, IL: National Council of Teachers of English.

Lee, C. D. (1997). Bridging home and school literacies: A model of culturally responsive teaching. In J. Flood, S. B. Heath & D. Lapp (Eds.), *A handbook for literacy educators, Research on teaching the communicative and visual arts* (pp. 330-341). New York, NY: Macmillan.

Lee, C. D. (2003). Toward a framework for culturally responsive design in multimedia computer environments: Cultural modeling as a case. *Mind, Culture, and Activity, 10*(1), 42–61.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads, 2*(1), 32–37. http://doi.org/10.1145/1929887.1929902

Lee, J. S., & Bowen, N. K. (2006). Parent involvement, cultural capital, and the achievement gap among elementary school children. *American Educational Research Journal, 43*(2), 193-218.

Lemos, R. S. (1979, January). Teaching programming languages: A survey of approaches. In *ACM SIGCSE Bulletin* (Vol. 11, No. 1, pp. 174-181). ACM.

Lenhart, A. (2015). Teens, social media and technology overview 2015. Retrieved from http://www.pewinternet.org/2015/04/09/teens-social-media-technology-2015/

Leonard, J., Davis, J. E., & Sidler, J. L. (2005). Cultural relevance and computer-assisted instruction. *Journal of Research on Technology in Education, 37*(3), 263-284.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., ... & Simon, B. (2004, June). A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin 36*(4), 119-150.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). *Programming by choice: urban youth learning programming with* Scratch. *Proceedings of the 39th SIGCSE technical symposium on Computer science education* (pp. 367-371). Portland, OR: ACM.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions of Computing Education (ToCE), 10*(4), 1–15. http://doi.org/10.1145/1868358.1868363

Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in k-9 education. *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference* (pp. 1–29). New York, NY: ACM. http://doi.org/10.1145/2713609.2713610

Mannila, L., & de Raadt, M. (2006). An objective comparison of languages for teaching introductory programming. *Proceedings of the 6th Baltic Sea conference on Computing education research* (pp. 32-37). New York, NY: ACM.

Maxwell, J. A., & Loomis, D. M. (2003). Mixed methods design: An alternative approach. In A. Tashakkori & C. Teddlie (Eds.) *Handbook of mixed methods in social and behavioral research*, (pp. 241-272). Thousand Oaks, CA: Sage.

Mbogo, C., Blake, E., & Suleman, H. (2013). A mobile scaffolding application to support novice learners of computer programming. In *Proceedings of the Sixth International Conference on Information and Communications Technologies and Development: Notes-Volume 2* (pp. 84-87). ACM.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B. D., ... & Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, *33*(4), 125-180.

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair programming improves student retention, confidence, and program quality. *Communications of the ACM, 49(*8), 90–95. http://doi.org/10.1145/1145287.1145293

McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). The effects of pair-programming on performance in an introductory programming course. In Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education (pp. 38–42). New York, NY, USA: ACM. http://doi.org/10.1145/563340.563353

Medlock-Walton, P., Harms, K. J., Kraemer, E. T., Brennan, K., & Wendel, D. (2014, March). Blocks-based programming languages: simplifying programming for different audiences with different goals. *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 545-546). Atlanta, GA: ACM.

Modi, K., Schoenberg, J., & Salmond, K. (2012). Generation STEM: what girls say about science, technology, engineering, and math. *The International Tertiary Education and Research Database*. Retrieved from http://www.voced.edu.au/content/ngv:70545

Monroy-Hernández, A. (2012). *Designing for remixing: supporting an online community of amateur creators* (Doctoral dissertation, Massachusetts Institute of Technology).

National Science Board. (2014). *Science and engineering indicators 2014 (No. NSB 14-01).* Arlington, VA: National Science Foundation. Retrieved from http://www.nsf.gov/statistics/seind14/

National Science Foundation. (2013). Women, Minorities, and Persons with Disabilities in Science and Engineering (National Center for Science and Engineering Statistics No. NSF 13-304). Retrieved from http://www.nsf.gov/statistics/2015/nsf15311/start.cfm

Nosek, J. T. (1998). The case for collaborative programming. *Communications of the ACM, 41*(3), 105–108. http://doi.org/10.1145/272287.272333

NSF Broadening Participation Working Group. (2014). Pathways to broadening participation in response to the ceose 2011-2012 recommendation (No. nsf15037). The National Science Foundation (NSF). Retrieved from http://nsf.gov/pubs/2015/nsf15037/nsf15037.pdf

Oyserman, D., Harrison, K., & Bybee, D. (2001). Can racial identity be promotive of academic efficacy?. *International Journal of Behavioral Development*, *25*(4), 379-385.

Papert, S. (1971). Teaching children thinking. Retrieved from http://dspace.mit.edu/handle/1721.1/5835

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas.* New York, NY, USA: Basic Books, Inc.

Patil, B., & Patil, B. (2002). *Equal: novice programming language and computing environment for native users.* Paper presented at the EdMedia: World Conference on Educational Media and Technology (pp. 1554–1560). Denver, CO: Association for the Advancement of Computing in Education (AACE).

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., … Paterson, J. (2007). A survey of literature on the teaching of introductory programming. *In Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education* (pp. 204–223). New York, NY, USA: ACM. http://doi.org/10.1145/1345443.1345441

Peckham, J., Harlow, L. L., Stuart, D. A., Silver, B., Mederer, H., & Stephenson, P. D. (2007). Broadening participation in computing: issues and challenges. *In Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 9–13). New York, NY, USA: ACM. http://doi.org/10.1145/1268784.1268790

Pinkard, N. D. (1999). Lyric Reader: An architecture for creating intrinsically motivating and culturally responsive reading environments. *Interactive Learning Environments, 7*(1), 1-30.

Piteira, M., & Costa, C. (2012). Computer programming and novice programmers. In Proceedings of the Workshop on Information Systems and Design of Communication (pp. 51–53). New York, NY, USA: ACM. http://doi.org/10.1145/2311917.2311927

Prensky, M. (2001). Digital natives, digital immigrants part 1. *On the Horizon, 9*(5), 1-6.

Principled Assessment of Computational Thinking (PACT). Retrieved August 31, 2014 from http://pact.sri.com

Poll, H. (2014). Pearson student mobile device survey 2014 national report: students in grades 4-12. Pearson. Retrieved from http://www.pearsoned.com/wp-content/uploads/Pearson-K12-Student-Mobile-Device-Survey-050914-PUBLIC-Report.pdf

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), 137-172.

Rainie, L. (2014, July 23). 13 things to know about teens and technology. Retrieved from http://www.pewinternet.org/2014/07/23/13-things-to-know-about-teens-and-technology/

Randolph, J., Julnes, G., Sutinen, E., & Lehman, S. (2008). A methodological review of computer science education research. *Journal of Information Technology Education*, *7*, 135–162.

Rankin, Y., Thomas, J., Brown, Q., & Hatley, L. (2013). Shifting the paradigm of african-american students from consumers of computer science to producers of computer science. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 11–12). New York, NY, USA: ACM. http://doi.org/10.1145/2445196.2445204

Reardon, S. F. (2011). The widening academic achievement gap between the rich and the poor: New evidence and possible explanations. In R. Murnane & G. Duncan (Eds.), *Whither opportunity? Rising Inequality and the Uncertain Life Chances of Low-Income Children* (pp. 91-116). New York, NY: Russell Sage.

Redmann, D. H., Lambrecht, J. J., & Stitt-Gohdes, W. L. (2000). The critical incident technique: A tool for qualitative research. *Delta Pi Epsilon Journal, 42*(3), 132.

Reis, N. M., & Kay, S. (2007). Incorporating culturally relevant pedagogy into the teaching of science: The role of the principal. *Electronic Journal of Literacy Through Science*, *6*(1), 54-57.

Repenning, A., & Ioannidou, A. (2008). Broadening participation through scalable game design. *SIGCSE Bull., 40*(1), 305–309. http://doi.org/10.1145/1352322.1352242

Common Sense Media. (2015). *The common sense census: media use by tweens and teens*, Common Sense Media. Washington, DC: Rideout, V.

Rising Above the Gathering Storm: Energizing and Employing America for a Brighter Economic Future. (2007). Retrieved July 27, 2015, from http://www.nap.edu/openbook.php?record_id=11463

Robins, A. (2015). The ongoing challenges of computer science education research. *Computer Science Education, 25*(2), 115-119.

Saldaña, J. (2009). First cycle coding methods. In J. Seaman (Ed.), *The coding manual for qualitative researchers*. Thousand Oaks, CA: Sage Publications

Salleh, S. M., Shukur, Z., & Judi, H. M. (2013). Analysis of research in programming teaching tools: An initial review. *Procedia-Social and Behavioral Sciences*, *103*, 127-135.

Scott, K.A., Clark, K., Sheridan, K., Hayes, E. & Mruczek, C. (2010). Engaging More Students from Underrepresented Groups In Technology: What Happens if We Don't? *Proceedings of Society for Information Technology & Teacher Education International Conference* 2010 (pp. 4097-4104). Chesapeake, VA: Association for the Advancement of Computing in Education (AACE).

Scott, K., Sheridan, K., & Clark, K. (2014). Culturally responsive computing: a theory revisited. *Learning, Media, and Technology, 40*(4), 412-536.

Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research* (pp. 59–66). New York, NY, USA: ACM. http://doi.org/10.1145/2493394.2493403

Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition. Presented at the Special Interest Group on Computer Science Education (SIGCSE) 2014. Retrieved from http://eprints.soton.ac.uk/356481/

Serpell, R. (1997). Critical issues literacy connections between school and home: how should we evaluate them? *Journal of Literacy Research*, *29*(4), 587-616.

Sheard, J., Simon, S., Hamilton, M., & Lönnberg, J. (2009). Analysis of research into the teaching and learning of programming. *In Proceedings of the Fifth International Workshop on Computing Education Research Workshop* (pp. 93–104). New York, NY, USA: ACM. http://doi.org/10.1145/1584322.1584334

Silva, C. M., Moses, R. P., Rivers, J., & Johnson, P. (1990). The algebra project: making middle school mathematics count. *The Journal of Negro Education, 59*(3), 375–391. http://doi.org/10.2307/2295571

Sorva, J., Karavirta, V., & Malmi, L. (2013). A review of generic program visualization systems for introductory programming education. *ACM Transactions on Computing Education (ToCE)*, *13*(4), 49-54.

Spohrer, J. C., & Soloway, E. (1989). Simulating student programmers. *Ann Arbor*, *1001*, 48109.

Sfetsos, P., Adamidis, P., Angelis, L., Stamelos, I., & Deligiannis, I. (2013). Heterogeneous personalities perform better in pair programming: The results of a replication study. *Software Quality Professional*, *15*(4), 4-15.

Strauss, W., & Howe, N. (2000). *Millennials rising: The next great generation*. New York, NY: Vintage.

Talbert, M. W. (2011, February). Game time: What will it take for black students to excel in math and science? *Black Enterprise, 41*(7), 43-46.

Tapscott, D. (1999). Educating the net generation. *Educational Leadership, 56*(5), 6-11.

Tekerek, M., & Altan, T. (2014). The effect of Scratch environment on students achievement in teaching algorithm. *World Journal on Educational Technology*, *6*(2), 132-138.

Tharp, R. G. (1989). Psychocultural variables and constants: Effects on teaching and learning in schools. *American Psychological Association* 44(2), p. 349.

The Common Sense Census: Media Use by Tweens and Teens | Common Sense Media. (2015). Retrieved November 11, 2015, from https://www.commonsensemedia.org/research/the-common-sense-census-media-use-by-tweens-and-teens

Treisman, P. U. (1985). *A study of the mathematics achievement of Black students at the University of California, Berkeley*. (Unpublished doctoral dissertation), University of California, Berkeley, Professional Development Program.

Vaidhyanathan, S. (September 19, 2008). Generational myth. *Chronicle of Higher Education*, 55(4). Retrieved from: http://chronicle.com/article/Generational-Myth/32491

Vygotsky, L. S. (1980). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Walls, L. (2012). Third grade African American students' views of the nature of science. *Journal of research in Science Teaching*, *49*(1), 1-37.

Wasserman, S. (March 27, 2015). White house gets $240 million from private sector for stem education. *Engineering.com*. Retrieved from http://www.engineering.com/Education/EducationArticles/ArticleID/9853/White-House-gets-240-Million-from-Private-Sector-for-STEM-Education.aspx

Watkins, A. F. (2002). Learning styles of African American children: A developmental consideration. *Journal of Black Psychology*, *28*(1), 3-17.

Wilson, A. N. (1978). *The developmental psychology of the Black child*. New York, NY: Africana Research Publications.

Wilson, A. N. (1992). *Awakening the natural genius of Black children*. New York, NY: Afrikan World InfoSystems.

Wilson, L. (2003). Breaking into the universe: computer science is interactive entertainment. *Computers in Entertainment (CIE)*, *1*(1), 1–7. http://doi.org/10.1145/950566.950586

Webb, D. C., Repenning, A., & Koh, K. H. (2012). Toward an emergent theory of broadening participation in computer science education. *In Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 173–178). New York, NY, USA: ACM. http://doi.org/10.1145/2157136.2157191

Weinberg, A. E. (2013). *Computational thinking: An investigation of the existing scholarship and research* (Ph.D.). Colorado State University, Ann Arbor. Retrieved from ProQuest Dissertations & Theses Global. (1413309206)

Werner, L. L., Campe, S., & Denner, J. (2005). Middle school girls + games programming = information technology fluency. *Proceedings of the 6th Conference on Information Technology Education* (pp. 301–305). New York, NY: ACM. http://doi.org/10.1145/1095714.1095784

Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: measuring computational thinking in middle school. *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 215-220). Raleigh, NC: ACM.

Werner, L., Denner, J., & Campe, S. (2014). Children programming games: a strategy for measuring computational learning. ACM *Transactions on Computing Education (ToCE), 14*(4), 1–22. http://doi.org/10.1145/2677091

White House Office of Science and Technology Policy. (2014). The 2015 Budget: science, technology, and innovation for opportunity and growth. United States Government. Retrieved from https://www.whitehouse.gov/sites/default/files/microsites/ostp/Fy%202015%20STEM%20ed.pdf

Williams, L. A., & Kessler, R. R. (2000). All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM, 43*(5), 108–114. http://doi.org/10.1145/332833.332848

Wilson, C., & Guzdial, M. (2010). How to make progress in computing education. *Communications of the ACM*, *53*(5), 35–37. http://doi.org/10.1145/1735223.1735235

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49*(3), 33–35. http://doi.org/10.1145/1118178.1118215

Zander, C., Boustedt, J., McCartney, R., Moström, J. E., Sanders, K., & Thomas, L. (2009). Student transformations: are they computer scientists yet? *Proceedings of the Fifth International Workshop on Computing Education Research Workshop* (pp. 129–140). New York, NY: ACM. http://doi.org/10.1145/1584322.1584337

Biography

Leshell Hatley is a passionate computer engineer, educator, and researcher who continuously combines these three attributes to create innovative approaches to teaching STEM concepts to students between that ages of 3 and 73. She has logged over 15,000 hours teaching African American and Latino American elementary, middle, and high school students in formal and informal educational environments throughout NY, NJ, VA and DC. As the Founder and Executive Director of Uplift, Inc., a nonprofit organization whose mission embodies her passion, Leshell leads teams of enthusiastic students, dedicated volunteer instructors, and teams of engineering to achieve award winning success, national news coverage, and innovative tech product designs. Over the past several years, Uplift's trailblazing work with students in the areas of robotics, mobile app development, and culturally relevant learning technology has provided inspiration, insight, and guidance to similar organization's across the globe. Leshell has a Bachelors in Computer Engineering and a Masters in Computer Science from Howard University, a Masters in Library Science with a focus on Human-Computer Interaction from the University of Maryland, and is currently a PhD student at George Mason University studying Learning Technologies Design Research. In total, Leshell sees past experiences, current successes, and future partners as tools to create awe-inspiring educational experiences for the next generation of life-long learners and STEM pioneers.