2018

Exploring how integrating art & animation in teaching text-based programming affects high school students' interest in computer science

Hadeel Mohammed Jawad

Exploring How Integrating Art & Animation in Teaching Text-Based Programming Affects High School Students' Interest in Computer Science

by

Hadeel Mohammed Jawad

Dissertation

Submitted to the College of Technology

Eastern Michigan University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Technology

Concentration in Information Technology

Dissertation Committee:

Yichun Xie, Ph.D., Co-Chair

Samir Tout, Ph.D., Co-Chair

Deborah de Laski-Smith, Ph.D.

Yaman Roumani, Ph.D.

Munther Abualkibash, Ph.D.

May 14, 2018

Ypsilanti, Michigan

Dedication

To my beloved parents, the cause of my existence

To my beloved husband, my friend and my other half

To my beloved son and daughter, the smile of my life

Acknowledgments

I would like to express my special appreciation and thanks to my two advisors, professors, and co-chairs, Dr. Yichun Xie and Dr. Samir Tout, for their help, support, and professional guidance throughout the duration of my study and during the different stages of writing this dissertation and related conference papers.

I would also like to thank my committee members, Dr. Deborah de Laski-Smith, Dr. Yaman Roumani, and Dr. Munther Abualkibash, for serving as my committee members and making my defense an enjoyable and memorable experience and for their crucial comments and suggestions.

I would especially like to thank Dr. Mohamad Qatu, the Dean of the College of Technology, for his valuable support and encouragement during my years of study at Eastern Michigan University.

A special thanks to Pioneer High School and MIA private school administration for their help in the data collection and for their hosting of three coding workshops. Thanks to Mr. Tracey Lowder, Dr. Nathaniel Powell, and Dr. Falak Ali. I also want to thank the writing center lecturers, Ms. Beth Sabo and Ms. Laura Kovick, for their time and help in writing this dissertation.

I would also like to thank Dr. Wade Tornquist, the Associate Provost of the graduate school for helping with the financial expenses during the data collection stage. Many thanks to Dr. Suleiman Ashur, the director of my department, for serving as a committee member in my proposal defense and for all the help in the final year of my study. I also like to thank Dr. Dorothy McAllen, the Ph.D. program coordinator, for her encouragement and guidance during my graduate study and in the dissertation defense process.

I cannot find enough words to express my gratitude and thanks to my husband, Mr. Haider AlRammahi, for his emotional and financial support, his encouragement, and his patience during my Ph.D. journey for four years. I also like to thank my daughter, Raneen AlRammahi, for her help in our house and for her suggestions in the developed tool that was used in my study. I also want to thank my son, Ramy AlRammahi, for his interest in my study and for his help during my dissertation defense. A lot of thanks to my sister Ms. Asseel Mohammed Jawad for always being there and keeping in touch despite living on a different continent for a long time.

I am grateful to my fellow classmates and friends, Dr. Negar Heidari Matin, Dr. Yassir Elrayah, and Dr. Essraa Fatani, for being such supportive friends and to my other friends who helped me in different ways throughout my study years. It was a great pleasure to meet all the wonderful people at Eastern Michigan University.

Abstract

As oil is the fuel of the industrial society, software is the fuel of our current information society. According to the Bureau of Labor Statistics, there will be more demand for computing jobs in the future. By 2024, more than one million computing jobs will be available. Statistics show that there is more demand for computing jobs than there is a supply of qualified graduates from universities. In this experimental study, three groups of high school students were targeted to explore how integrating art, animation, and code sharing into programming affects their interest in pursuing a degree in computer science (CS) after graduation. Moreover, the study also explored the effect of social factors and attitudes of the students toward programming and a CS degree. Pretest-posttest survey questionnaires were used to measure the study variables before and after taking a programming course. A new web-based learning environment was developed and used as a treatment in this study. The developed tool included the use of art, animation, and code sharing to increase students' motivation in learning computer programming. Three groups of students from Ann Arbor public and private schools participated in this study with different coding time. The demographic data were also collected and analyzed in this study. The field of CS is currently dominated by White and Asian males. This study also aimed to encourage and increase the motivation of female and underrepresented racial groups towards CS. The results of this study showed that the use of art, animation, and code sharing increased students' knowledge, enjoyment, and motivation in learning computer programming. It thereby increased their interest in pursuing a degree in CS after graduation.

Table of Contents

List of Tables

List of Figures

Chapter One: Introduction and Background

Learning computer programming or coding is tremendously empowering to students. It lets them go from just being a consumer of technology to being a producer of it. Computer science not only teaches them about technology, it also teaches them how to think differently about any problem, how to think logically, and how to be creative and productive. The founder of smartphones and the Apple Corporation, Steve Jobs, said "Everybody in this country should learn to program a computer because it teaches you how to think" (as cited in Moss, 2012). Similarly, the founder of Microsoft Corporation, Bill Gates, stated that learning computer programming language stretched the mind and created a way of thinking about things (AZ Quotes, n.d.). Computer science develops students' computational and critical thinking skills and shows them how to create new technologies and not simply use them. Students of the 21st century should have a chance to learn about algorithms, how to make an app, and how the Internet works just as they learn about photosynthesis and the digestive system or electricity (Promote Computer Science, 2016).

Coding is the new buzzword of today's tech world. No matter what the occupation is, it surely involves using technology, and those who know how to code are surely at an advantage. More than half of the projected Science, Technology, Engineering, and Mathematics (STEM) jobs are in computing occupations. According to the Bureau of Labor Statistics the computer occupations group is among the fastest growing major occupational groups (Bureau of Labor Statistics, 2017). There will be more than 500,000 new jobs between 2014 and 2024. This growth is due to the increased focus on the storage of big data and cloud computing in addition to the continued demand for mobile app development. As shown in Figure 1-1, by 2024 more than one million computing jobs will available (Code.org Infographic Source Data, 2015; Bureau of Labor Statistics, 2016).

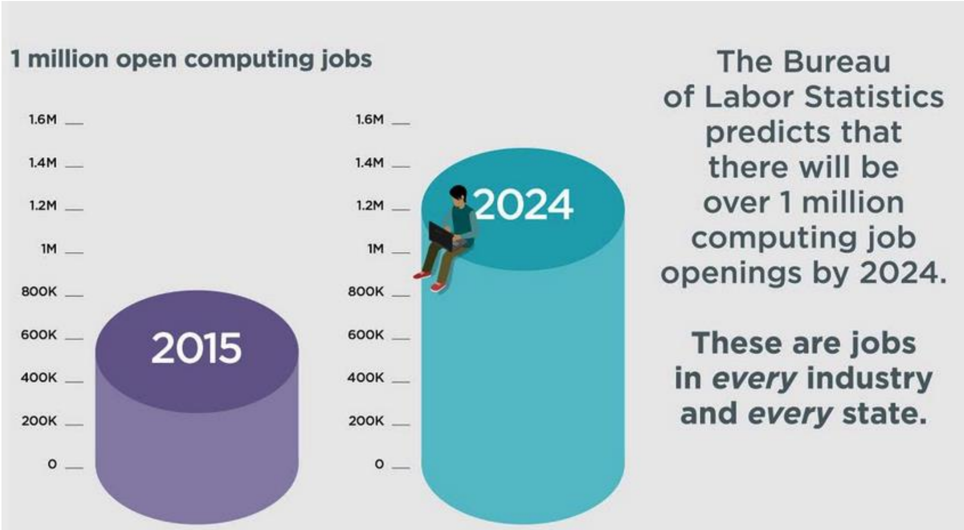


Figure 1-1. One million open computing jobs are expected by 2024. Source: Code.org (Code.org Infographic Source Data, 2015).

Figure 1-2 shows that the future demands for computing jobs will be 58% of the total STEM jobs. This means that the percentage of the future demand for all the other STEM fields combined will be around 40% (Code.org Infographic Source Data, 2016; Promote Computer Science, 2016). The same figure also shows that there is more demand for people who have computer programming skills than there is a supply of graduates from universities. According to the National Center for Education Statistics (NCES), there were 580,940 bachelor's degrees earned in STEM in 2015, and only 49,291 of those (i.e., 8.48%) were in computer science (Code.org Infographic Source Data, 2016).

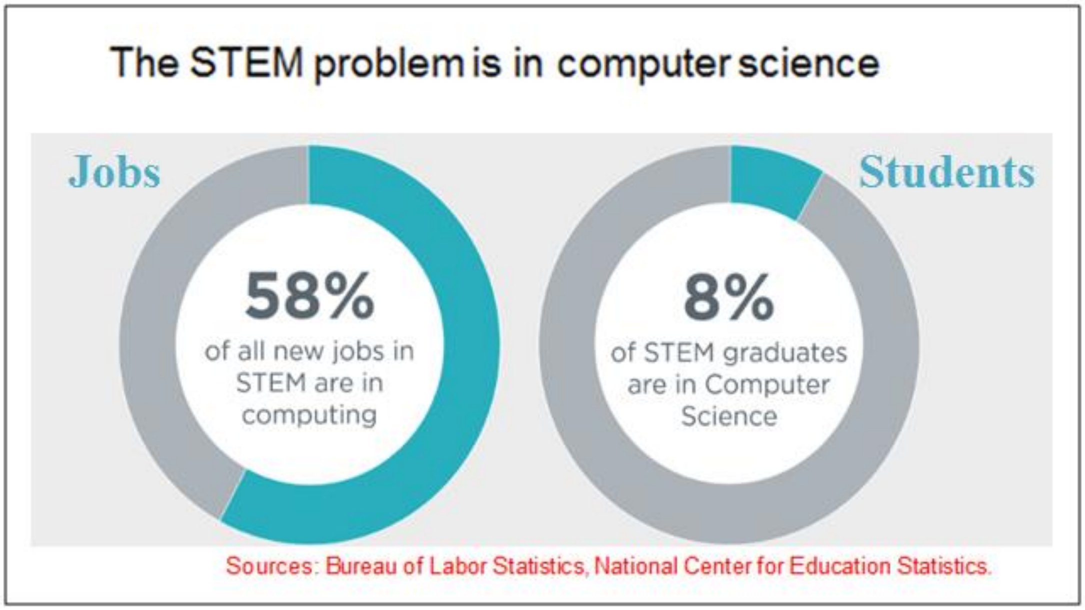


Figure 1-2. The jobs and student gap in computer science. . Source: Code.org (Promote Computer Science, 2016).

In many universities, the computer science departments have suffered from low enrollment for several years (Sloan & Troy, 2008). In 2010, there were seven job openings for every graduate with a computer major (Rothwell, 2016).The National Center for Education Statistics has collected data; about 60,000 students graduated from U.S. institutions with bachelor degrees in computer and information services (Kessler, 2017; Snyder, Brey & Dillow, 2016). According to Code.org and the Conference Board, there are about 530,000 computing jobs open (as cited in Kessler, 2017). This means almost 10 times more U.S. computing jobs are open than there were students who graduated with computer science degrees in 2015

Figure 1-3. 2015 College CS graduates vs. open computing jobs. Source: National Center for Education Statistics and Code.org (as cited in Kessler, 2017).

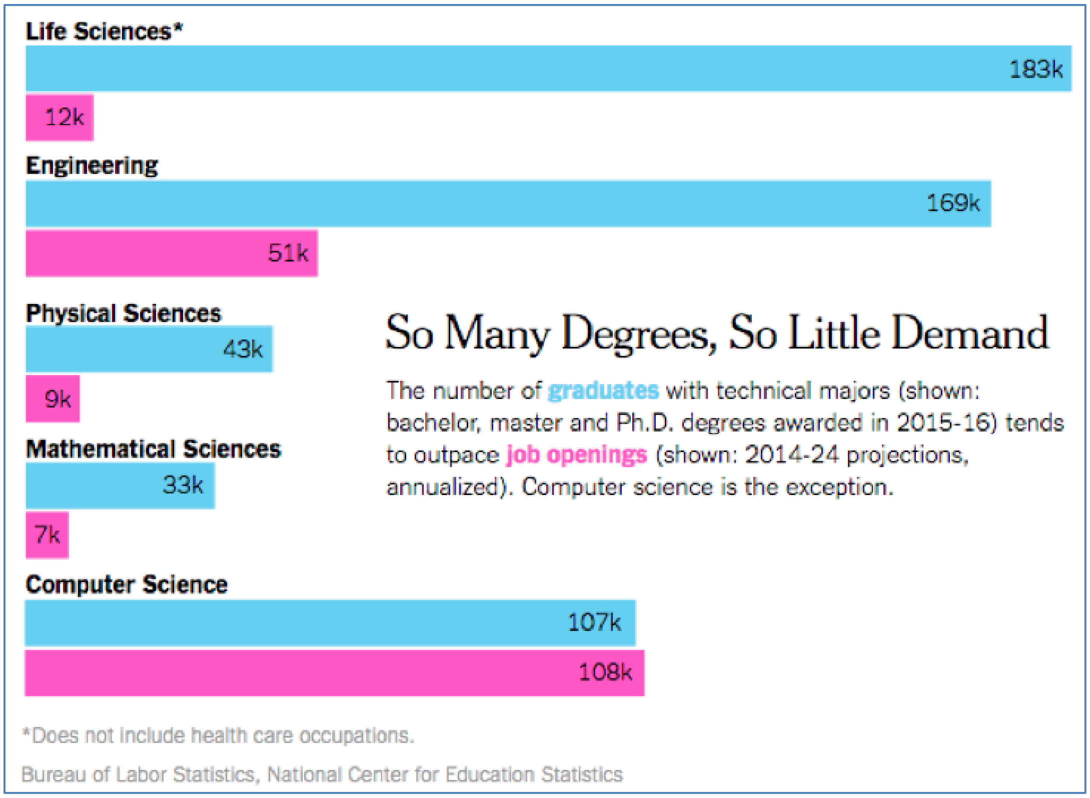


Figure 1-4. STEM degrees vs. demand Source: Bureu of Labor Statistic and NationalCenter for Education Statistics (as cited in Rincón, 2017).

A chart from the Bureau of Labor Statistics (Figure1-4) showed that the earned degrees in the STEM fields were more than the jobs available in the market, except for the computer science major (as cited in Rincón, 2017). The Information Technology and Innovation Foundation report indicated that there are not enough U.S. graduates to meet the demand for IT occupations (Information Technology and Innovation Foundation, 2013).

It is also important to consider what skills the job market really needs now and in the near future. For example, in the automotive industry, people want to connect cars to mobile apps to provide more features to the customer. One study showed that there is a shortage of people skilled in mobile development and the university courses should be updated to reflect the job market's needs (Xue & Larson, 2015). There are very few programs for mobile development in the universities and most of the current mobile developers are self-learners as the StackOverflow survey showed (StackOverflow, 2017).

Moreover, the National Science Foundation (NSF) data and a survey from the Computing Research Association showed that 57% of bachelor's degrees were earned by women, but only 12% of the graduated women have a bachelor's degree in computer science (Code.org Infographic Source Data, 2015; Zweben, 2011; National Science Foundation, 2012). YouTube CEO Susan Wojcick mentioned that high school girls are using technology in their daily life but less than 1% of them are interested in a computer scince degree (Wojcick, 2014). Google's report of their workforce revealed that there is a gender and ethnicity gap in tech jobs and showed that tech jobs were dominated by White and Asian males (Naughton, 2017). Figure 1-5 shows the gender and racial representation at Google for tech jobs and all jobs. Similar reports published by other tech companies showed the same gender and ethnicity gap, as will be discussed in Chapter Two.

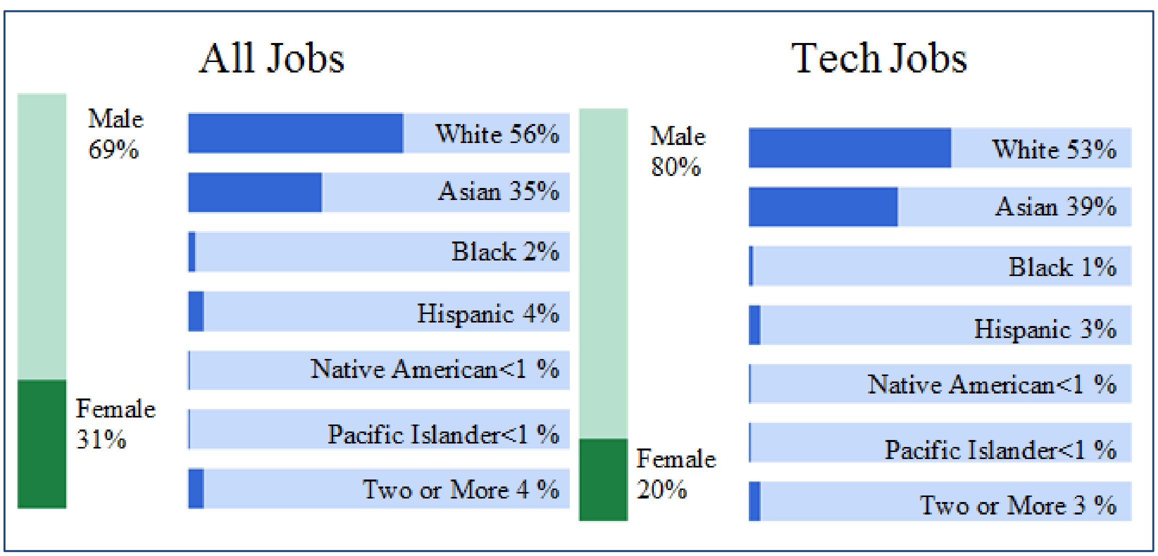


Figure 1-5. Google employee demographic, 2017. Source: Google blog (Naughton, 2017).

The gender imbalance problem was recognized by some countries like the United Kingdom, where primary and secondary school curricula are being revolutionized by the replacement of the Information and Communication Technology (ICT) curriculum in computer science (Mather, 2015). There are also other organizations in the United Kingdom that promote coding for kids like, Raspberry Pi foundation that creates a cheap, credit-card sized computer with an intention to promote programming and make it accessible for everybody including people in developing countries. The large community of Raspberry Pi makes it possible to try out many different applications from software programming to hardware development (Byrne, Fisher, & Tangney, 2015). Kano is another company that encourages kids to create technology rather than merely consume it. Kano operating system was built to be a kid-friendly operating system to help kids enjoy coding as much as they enjoy video gaming (Vincent, 2013). Apple Corporation recently joined the bandwagon in encouraging kids toward coding. Swift Playgrounds is a new app that was created to make the coding experience fun to learn. This app was released in mid-September, 2016 and is available now in the App Store. Tim Cook, the Apple CEO, announced that more than 100 schools in California will include Swift Playgrounds in their curriculum (Kolodny, 2016).

There are many attempts that encourage K-12 students to learn computer science in school. One of the well-known attempts was the Hour of Code by Code.org. The Code.org website was launched in 2013 with an intention to promote computer science education and make it accessible for everybody to increase the participation of women and underrepresented multiracial students (Code.org, 2013a). This website was founded by Hadi and Ali Partovi the twin brothers who think that computer science should be part of the core curriculum in education, along with other STEM courses: "Whether you're trying to make a lot of money or whether you just want to change the world, computer programming is an incredibly empowering skill to learn" (Code.org, 2013b).

Code.org's strategy is to run a four-year program that is focused on bringing computer science to K-12 schools nationwide. Hour of Code (HOC) is a nationwide initiative by Computer Science Education Week and Code.org to introduce millions of students to one hour of computer programming (Computer Science Education Week, n.d., HourOfCode.com, 2018). The HOC takes place during a week in December each year. Each school in the USA is encouraged to host HOC during this week by offering one hour of coding to each class in the school. One hour is not enough to teach coding. However, it is just enough to let students realize that computer science is fun and creative and that it is accessible at all ages regardless of background (Computer Science Education Week, n.d.). The HOC has gained wide acceptance and encouragement from many organizations and the government. Former President Obama himself appeared in the video promoting the HOC saying "Don't just play on your phone, program it" (Code.org, 2013c)

A measure of success for this campaign is reflected in the vast participation of females and underrepresented racial and socioeconomic groups. The goal of HOC is to increase student interest in computer programming so they would consider enrollment in a computer science degree after high school graduation. Hadi Partovi, the co-founder of Code.org, stated that the HOC event with the help of other organizations such as Microsoft were able to change the high schools graduation policy in 16 states. The number of states that allow computer science classes to satisfy high school graduation requirements increased to 26 where it was only 10 states prior to the HOC initiative (Partovi, 2015). Code.org (2013a) reported that after the HOC, teachers became more confident that they could teach computer science even though they may not have a computer science degree. Moreover, school administrators realized that their students want to learn computer programming (Code.org, 2013a). Many students enjoyed this hour of coding and wanted to do it for the whole day and many of them decided to enroll in a whole course of computer programming. The learning environment used in HOC event is not using a real programming language. It uses Blockly language, which is used to create visual block programming editors where students drag and drop blocks to develop a program. This will teach a student how to think logically like a programmer, but the student is not learning a real programming language. Code.org and similar environments, which are discussed more in Chapter Two, could be considered as a good start for elementary and middle school students in learning computer programming, but high school students need to know what programming really is so they can make informed decisions after their graduation.

In this study, a new educational software tool has been developed by the researcher in an attempt to teach programming to high school students with a real programming language that is currently used by software developers. At the same time and to keep the fun part of the other environments, the developed tool integrates art and design with programming. The developed environment allows the student to share the resulting code and art on social media or within the tool itself. The aim of this study was to explore the effect of using the developed tool on students' interest in learning computer programming and on their interest in pursuing a degree in computer science after graduation from high school.

1.1 Statement of the Problem

The number of graduating students with computing majors is less than the job market demand. In the future, this demand will increase, and more computing jobs will be available (Bureau of Labor Statistics, 2016). Computer science and other computing departments in many universities have suffered from low enrollment for several years (Sloan & Troy, 2008;

Xue & Larson, 2015; Kessler, 2017; Information Technology and Innovation Foundation, 2013). There is a need to increase high school students' interest in considering a degree in CS after graduation.

How integrating art, animation, and code sharing into teaching a text-based programming language affects high school students' interest and knowledge in programming and on their interest in pursuing a degree in CS after graduation has not been adequately explored.

Moreover, there is a need for more tools that integrate art and animation in teaching real programming language in a fun, simple, and interesting way that is suitable for high school students. There are several tools that use art and animation in teaching block-based programming language but very few tools that focus on the use of real programming language.

1.2 Nature and Significance of the Problem

In 2015, the number of smartphone users was 3.4 billion people. By 2021, those users are predicted to be 6.4 billion people. This is 80% of world’s population (Ericsson Mobility Report, 2016). Software to the information society is like oil to the industrial society. It is the fuel that keeps machines running. As stated earlier, the U. S. job market has a high demand for computer skilled people, and this demand will increase in the future. However, there aren't enough students enrolled in computer science departments in many universities.

Former President Obama was aware of this problem and addressed it in one of his weekly addresses in January 2016 (White House, 2016). "Computer Science for All" was the president's initiative to empower all American students from kindergarten through high school to learn computer science and be equipped with the computational thinking skills they need to be creators in the digital economy, not just consumers, and to be active citizens in the technology-driven world. The former president discussed his plan to give all students across the country the chance to learn computer science in school. He said, "In the new economy, computer science isn't an optional skill–it's a basic skill, right along with the three Rs. Nine out of ten parents want it taught at their children's schools" (White House, 2016). According to the White House, this initiative invested more than $135 million beginning in 2016 by the National Science Foundation (NSF) and the Corporation for National and Community Service to support and train computer science teachers (Smith, 2016). The initiative also provided $4 billion in state funding for a three-year plan to teach computer science in schools. One-hundred million was provided for districts in the former president's budget. Obama also called governors, mayors, education leaders, creative media, CEOs, and tech entrepreneurs and others to get involved in the efforts (Smith, 2016). In September 2017, President Donald Trump signed a memorandum directing $200 million a year for STEM and computer science education in schools. Private sector and big tech companies like Facebook, Google, and Microsoft will also invest $300 million to improve computer science education programs (White House, 2017). In October 2017, Amazon announced that they will donate $10 million to Code.org over the next five years to promote computer science in K-12 education (Nickelsburg, 2017).

All of the above makes this study significant. The result of such research will benefit educators, developers, and government leaders:

- For educators, to consider switching to more engaging learning environments to increase the students' interests in learning computer languages.

- For developers and entrepreneurs, to add the elements that make their learning environment effective enough to meet the students' needs in learning computer programming.
- For government or education leaders, to fund and support the firms or entrepreneurs who provide the best learning environment for computer programming and to provide the needed training for teachers.

Furthermore, this study was conducted through a five-day summer camp for high school students where a newly developed environment was used. The camp served as a curriculum model for high schools because the students' interest in learning programming increased after the study. The study also introduced a new development environment to high school students and teachers. The tool was available online and could be used in classrooms. The study also offered a new measurement instrument that could be used by researchers of similar studies.

While there are some other learning tools that also use art and animation for teaching computer programming logic and concepts, the tool that was developed for this dissertation study integrated art and animation in learning a real programming language that is used by professional developers. This study focused on increasing the students' programming interest in coding by using a real programming language and not a block-based programming language. The researcher thinks that high school students are ready to write a computer program with a real programming language while the block-based programming language is more suitable for middle and elementary school students as the main targeted population.

1.3 Objective of the Research

The objective of this experimental study was to encourage high school students to consider a CS degree after graduation. A new online development tool was built by the researcher of this study to be used as a treatment therein. It integrated art, animation, and code sharing in teaching computer programming. The tool was developed to encourage students to write a program with a real programming language rather than block-based programming language. The developed tool also encouraged students to share their artwork that was produced by code. The study also explored the effect of using art and animation on students' programming knowledge, their motivation to write and share code, their programming enjoyment and their interest in taking CS courses in high school.

Several factors that could increase the students' interest in a CS degree were also explored in this study. These factors include programming benefit and enjoyment, the support and encouragement from students' parents and their relatives, and the students' capabilities and confidence to overcome programming difficulties and to accept challenges.

The other objective was to test the usability of the developed tool, like its ease of use and usefulness, by observing the users' experience with the new tool and getting students' feedback on such development environment.

1.4 Limitations, Delimitations, Assumption

1.4.1 Limitation. In experimental studies, the sample size is usually not as large as in quantitative studies. In this study, the sample size was limited by the computer lab size. The study was conducted more than once through several coding workshops to collect as much data as possible. However, an experimental study provided real evidence to test the hypothesis. It included more interaction with the subjects, and it also provided field observation. The sample was dominated by two racial groups, Asian and Middle Eastern. The sample size of the underrepresented groups, such as African American, was not big enough to generalize the results.

Finding a high school that was willing to participate in a coding summer camp or coding workshop involved some challenges that the researcher faced. Finding high school students without any prior computer programming experience was another challenge. There is a probability that some students who already have an interest in programming had participated in coding camps, so the results might be affected by their previous experience, not by the treatment itself. However, the study tried to measure students' knowledge and interest in computer programming before and after the treatment to eliminate the effect of the previous experience factor. In addition, the study tried to consider many factors that may influence students' interest in CS as will be discussed in Chapter Three.

Pioneer High School in Ann Arbor, MI, agreed to host a summer coding camp and a fall workshop for one week each. Michigan Islamic Academy (MIA) gave permission for a one-day coding workshop. The offered time was not enough to teach all the programming language. However, the researcher tried her best to cover the most important programming concepts and the keywords needed to produce meaningful code.

1.4.2 Delimitations. In the future, there will be many computing occupations available in the job market, and not all of these jobs require a degree in computer science. There are many other university degrees that supply the market with graduates in the computing field such as computer engineering, electronic engineering, electrical engineering, biomedical engineering, human computer interaction, information assurance, computer information systems, or geographic information systems. All of those degrees provide the job market with qualified people who can fulfill computing job needs. This study was limited to measuring students' interest in pursuing a degree in computer science. Other studies or future work of this study could be conducted using the same methodology to measure students' interest in other computing majors. However, the study also focused on students' interest in programming and most of the mentioned degrees required programming skills.

There are many organizations that offer development programming environments like KhanAcademy.org, Codecademy.com, CodeCombat.com and Code.org but the researcher used a newly developed environment for this study to let students code with real programming language. Also, although there are many programming languages, the researcher used one language in this study which is the JavaScript programming language.

The research limited the study sample to high school students because they are going to graduate soon, and they are in a stage to make a decision about their major and university. The researcher chose Ann Arbor high schools since they are listed among the best schools in the nation (Knake, 2015). The other reason is that Ann Arbor is a diverse city, so students from different races can be found in the schools (DiversityData.org , 2011). One study showed that African Americans were less interested in CS (Margolis, 2010). Choosing a city with diversity was helpful to expose underrepresented students to computer programming and encourage them to consider a CS career in the future.

1.4.3 Assumptions. The researcher assumed that students answered the survey questionnaires honestly without external influence. Although one week may not be enough to teach a whole programming language, it was assumed that it was enough to expose the students to the main idea of using art with coding and to measure overall interest in programming. It was also assumed that the coding time was enough to introduce the students to the basic programming concepts that were measured in this study. Also, the researcher felt the three hours, which was the minimum coding time, was enough to explore the usefulness and usability of the tool.

1.5 Definition of Terms

- *CS*: Computer Science.
- *G1, G2, G3*: Group One, Two, and Three.
- *G9, G10, G11, G12*: Grades 9, 10, 11, and 12.
- *Gamification*: It is the buzzword for adding gaming elements such as points or badges to learning experiences to make them more engaging and to increase motivation (Morrison & DiSalvo, 2014).
- *HCI*: Human-Computer Interface.
- *HOC*: Hour of Code is a nationwide event to teach computer programming in one hour for each class in each school. This event usually held in the second week of December of each year since 2013 (HourOfCode.com, 2018).
- *IDE*: Integrated development Environment.
- *ILE*: Interactive Learning Environment.
- *MSLQ*: Motivated Strategies for Learning Questionnaire.
- *SBL*: Studio-based learning (or SBL) is an instructional technique that emphasizes collaborative, design-oriented learning (Hundhausen, Narayanan& Crosby, 2008).
- *STEM*: Science, Technology, Engineering and Math.
- *TAM*: Technology Acceptance Model.
- *Three Rs*: Reading, wRiting, and aRithmetic as the fundamentals of learning.
- *TPB*: Theory of Planned Behavior.

1.6 Summary

In this experimental study, a newly developed programming environment was used. This development environment was created for this study to integrate art, animation, and code sharing into programming with an intention to increase high school students' interest in pursuing a degree in computer science. The study was conducted through one summer camp and two fall workshops in 2017. The students were selected from Ann Arbor public and private schools. Pretest-posttest was used, respectively, on the first and the last day of the camp using the developed tool to measure students' interest in learning programming language and their interest in pursuing a degree in CS after graduation.

A study like this is significant because there is currently low enrollment in the computer science departments of many universities (Kessler, 2017; Sloan & Troy, 2008; Xue & Larson, 2015). The Bureau of Labor Statistics (n.d.) stated that there will be around one million jobs that require computer skills in the future. There is a need to find a way to increase students' interest in pursuing a degree in computer science after high school graduation. One way is by finding the right environment that makes coding easy and fun. The HOC event is held each year in all participating U.S. schools with an intention to promote learning computer science in schools by making it fun and accessible for all ages in K-12 schools. The researcher of this study thinks that high school students should learn programming with a real programming language and not by visual block language, so a new development environment was built to measure student interest in programming using real programming language. The findings of this study could help educators, developers, and government leaders to consider the best way to teach computer programming languages in high schools in the U.S. Chapter Two of this dissertation reviews the related literature.

Chapter Two: Literature Review

This chapter reviews the literature to discuss what has been previously studied in this research area and what other development tools were used for high school students. First, the chapter starts in section 2.1 by emphasizing the importance of getting a computing degree and its influence on future jobs. In addition to the points mentioned in Chapter One, this chapter highlights more points about the importance of considering STEM careers and especially the computer science major for high school students. Section 2.2 illustrates the employees' demographic at big tech companies like Facebook, Twitter, and Apple. More statistics from developers' websites like StackOverflow and HackerRank are discussed in section 2.3.

Section 2.4 demonstrates the theoretical frameworks that were used as a guide to specify the variables and collect the data for the study. In section 2.5, block-based vs text-based programming languages are compared to show the differences between the two development environments that are used in teaching computer programming.

Section 2.6 illustrates the previous work and similar studies related to teaching computer programming in general and for high school students specifically. The following sections cover several studies that were done using different development environments or tools. The researcher gives responses to the existing tools and studies in several paragraphs of this chapter and in section 2.10. The Eight Golden Rules of the Human-Computer Interface design and the design process model are explained in sections 2.11and 2.12. Those sections provide guidance to the design of the development tool, which was used as the treatment in this study.

2.1 The Need for a CS Degree

Nowadays, people are using their mobile applications to do several things every day. They wake up in the morning, and the first thing they do is look for their smartphones, check their emails, social media and news, buy their coffee, get a ride through Uber, shop from Amazon, book their vacation through Expedia, check their bank accounts, and transfer money to others. All these activities and more are accomplished through mobile applications, which are nothing but software programming. This will keep the demand for software developers in the job market. In the future, many jobs will be automated except those that need creativity, empathy, judgment, or critical thinking (Wohlsen, 2016). Robert Cannon, the Internet expert, said "anything that can be automated will be automated" (as cited in Smith & Anderson, 2014, p.9). Smith and Anderson (2014) conducted a study with several technology experts. The study showed that 48% of the participants agreed that by 2025, advance technology like artificial intelligence applications, self-driving cars, and robotic devices will reduce human jobs. Most participants agreed that the current educational system is not adequately preparing people for skills that will be needed for future jobs. One participant stated that there will be more demand for software engineers and people who maintain and repair the future robots (Smith & Anderson, 2014). According to the U.S. Bureau of Labor Statistics (2018), software developer jobs are expected to grow 24% from 2016 until 2026 (Mazaika, 2017, January 20). This growth is considered much faster than the growth of other professions. The need for new mobile applications will help increase the demand for application software developers. Figure 2-1 shows the difference in growth of software developer jobs as compared to the other occupations.

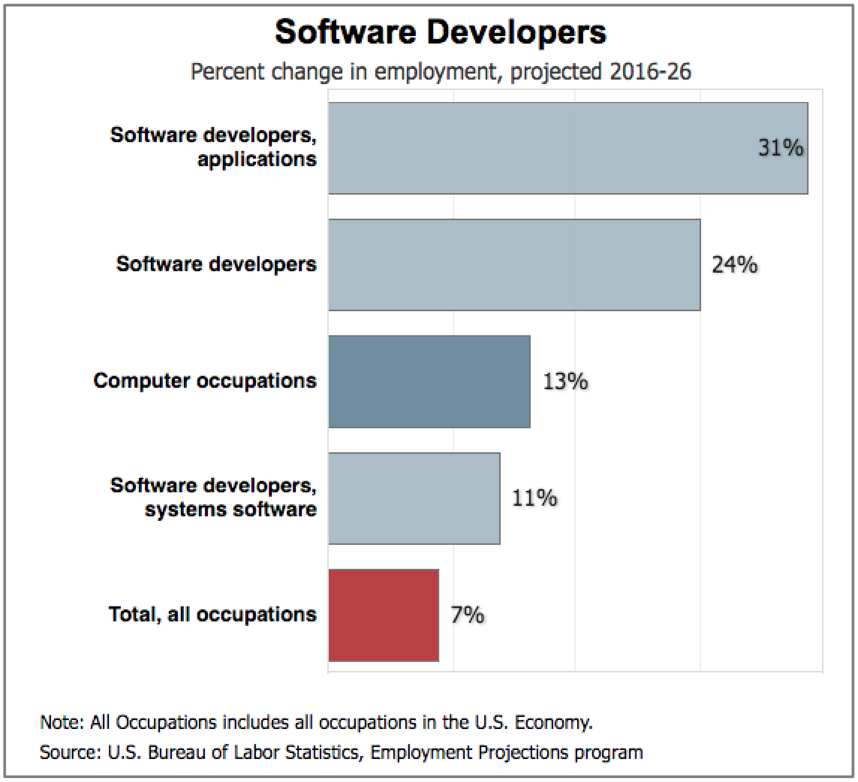


Figure 2-1. Software developers employment expected growth between 2016-2026. Source: The Bureau of Labor Statistics (Bureau of Labor Statistics, 2018).

One of the benefits of learning how to code is that students could earn money from home before the age of sixteen. They can develop their own websites and receive revenue from commercial advertisements or develop websites for others. Also, they can create new ideas and publish their mobile app in the app store. They can publish and express themselves and they can also be technology entrepreneurs and have their own companies like Robert Nay, a 14-year-old who published his mobile game "Bubble Ball" and got more than a million downloads in the first week. Nay now has his own company named "Nay Games" (Post, 2016). According to Tim Cook, the Apple CEO, learning how to code is more important than English for school students because it provides them with a tool to express themselves to the 7 billion people living on our planet (Hall, 2017).

2.2 Demographic Data at Big Tech Companies

The demographic reports published by big tech companies show that the tech jobs are dominated by White and Asian males. In addition to the Google employee demographic discussed in Chapter One, more data from other big companies are illustrated in this section. According to 2017 demographic report from Facebook, the percentage of females in tech jobs was 19%, while the percentage of males was 81% (Williams, 2017). Most of the tech employees are either Asian (49%) or White (45%), as shown in Figure 2-2.

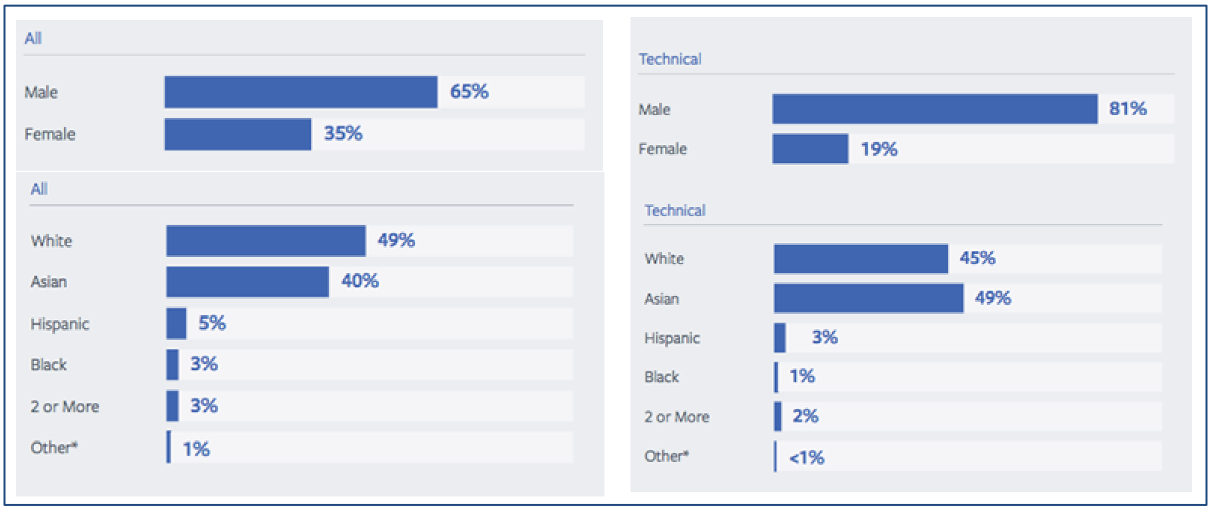


Figure 2-2. Facebook employee demographic, 2017. Source: Facebook newsroom (Williams, 2017).

Twitter demographic data showed that the female percentage in the tech jobs improved from 10% to 15% from 2014 to 2016 (Siminoff, 2017; Huysse, 2014). In 2017, tech jobs at Twitter were also dominated by White (52%) and Asian (39%) groups, as shown in Figure 2-3. The percentage of Black tech employees at Twitter rose from 1% in 2014 to 2% in 2016.

Similar to Facebook, the percentage ratio of the female to male employees in tech jobs, at Microsoft, was 19% to 81% (Microsoft, 2017). Most of the employees were either White (52%) or Asian (38%), as shown in Figure 2-4.

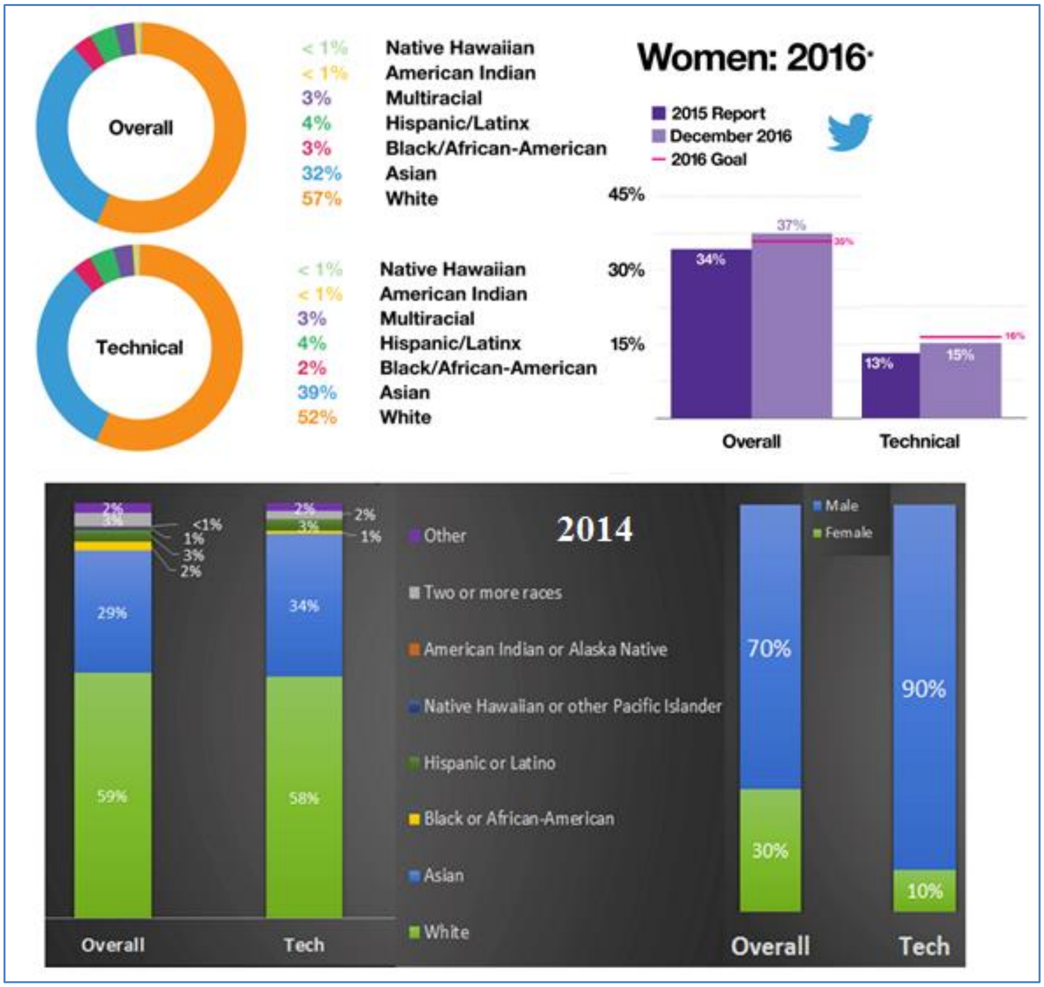


Figure 2-3. Twitter employee demographic, 2014 and 2016. Source: Twitter blog (Siminoff, 2017; Huysse, 2014).

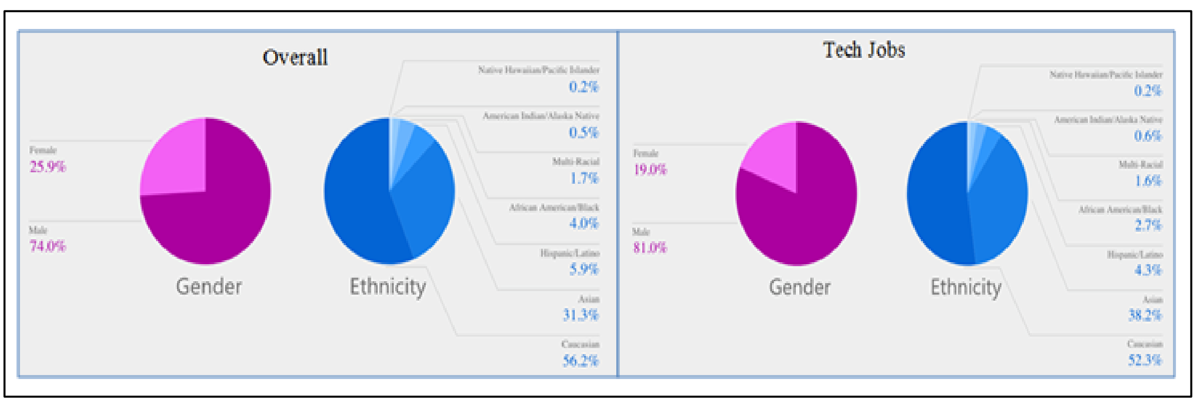


Figure 2-4. Microsoft employee demographic, 2017. Source: Microsoft (Microsoft, 2017).

The percentage of the female employees in tech jobs at Apple and LinkedIn was better than that at Facebook and Twitter. It was 23% at Apple (Apple, 2017) and 21% at LinkedIn

(Durruthy, 2017). Again, most of the tech employees were White and Asian in both companies. Figure 2-5 and Figure 2-6 shows the 2017 demographic data at Apple and LinkedIn, respectively.

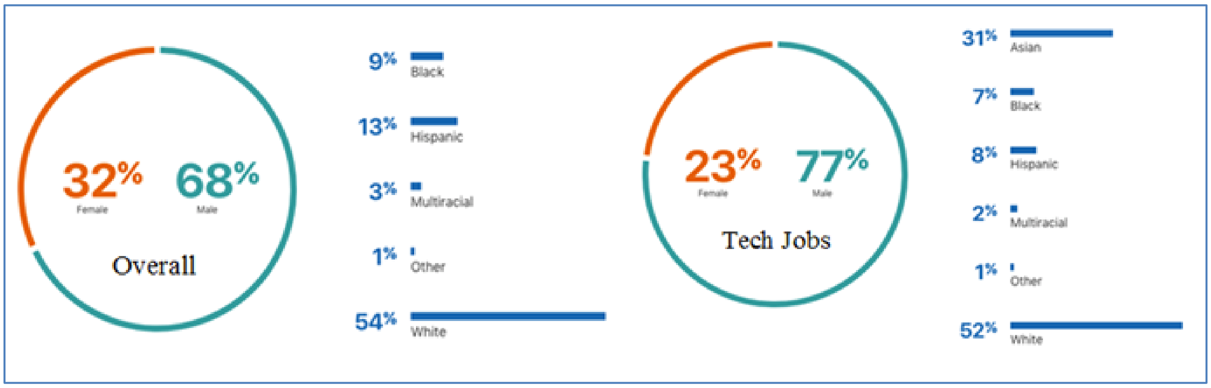


Figure 2-5. Apple employee demographic, 2017. Source: Apple diversity report (Apple, 2017).

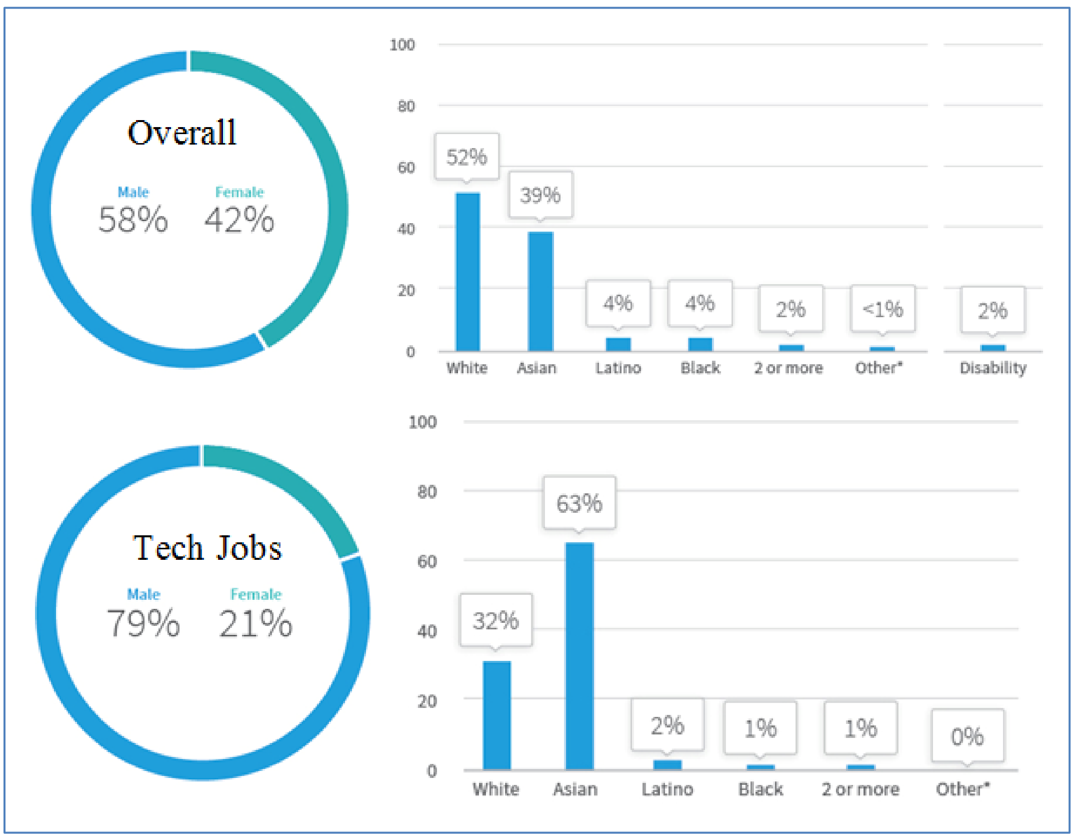


Figure 2-6. LinkedIn employee demographic, 2017. Source: LinkedIn workforce diversity report (Durruthy, 2017).

2.3 More Statistics

One of the findings that was useful for this study was from the StackOverflow (2016) developer annual survey. StackOverflow is a well-known website that is heavily used by software developers for sharing knowledge and finding jobs. It is a computer programming question-and-answer website founded in 2008 by programmers to serve programmers. The website has a large developer community with more than 40 million visitors each month to learn, share, and level-up their profile. It is estimated that 16.8 million of the visitors are professional developers and university-level students. This estimation comes from the visitors' activities that can only be done by developers, such as asking by writing a code or answering with a code. It is like a social media for developers where they share their code and get more points if others like it (StackOverflow, 2008).

Each year since 2011, StackOverflow has conducted a survey asking developers several questions and providing valuable information to the developers and industry. In 2017, more than 64,000 developers responded to the survey questionnaire. In the following paragraphs, useful information from this survey is highlighted and was used in this study. This included dominant gender and ethnicity among developers, the most popular programming language used by them, and other findings.

StackOverflow's developer survey results (StackOverflow, 2017) showed that only 7.6% of the participants were female and 88.6% were male developers as shown in Figure 2-7. A study by Alvarado (2010) confirmed the survey findings that females were less interested than men in computer programming and engineering in general. In this dissertation, gender was one of the moderating variables that had an effect on students' interests in learning computer programming and considering a computing career.

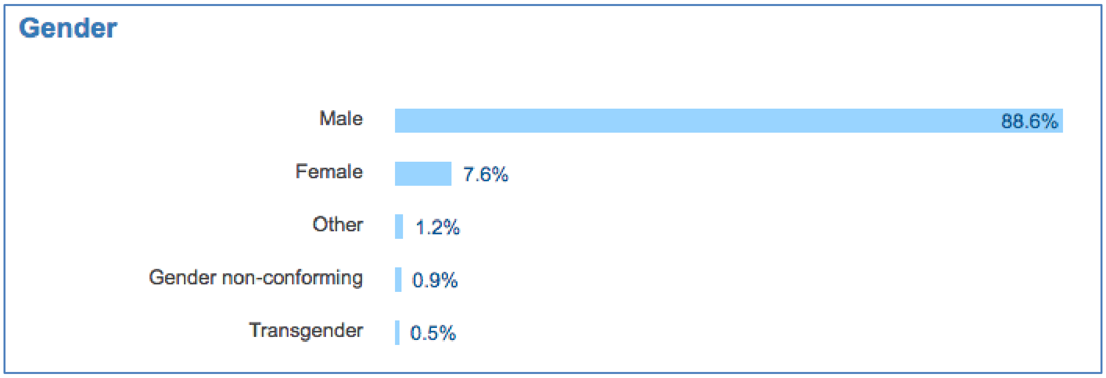


Figure 2-7. Stackoverflow 2017 survey demographic data. Source: StackOverflow developer survey results (StackOverflow, 2017).

Regarding the developer roles, the survey found that most of the developers who used StackOverflow were web developers. Figure 2-8 shows that 72% of the participants were web developers. The survey also showed that women were more likely to be represented in some developer roles than others. They were more represented among data scientists, mobile and web developers, quality assurance engineers, and graphic designers. In the previous year's survey, StackOverflow (2016) indicated that the survey underrepresented women in Asian countries where the probability of women developers may have increased (StackOverflow, 2016).

The StackOverflow Developer Survey showed that most of the developers were White or of European descent and only 2.5% were Black or of African descent, as shown in Figure 2-9. A study by Margolis (2010) supported the survey findings about race differences in CS. The study found that the number of African Americans and Latinos receiving undergraduate and advanced degrees in computer science was low (Margolis, 2010). Race could be another moderating variable affecting the student performance and students' interest in pursuing a degree in computer science.

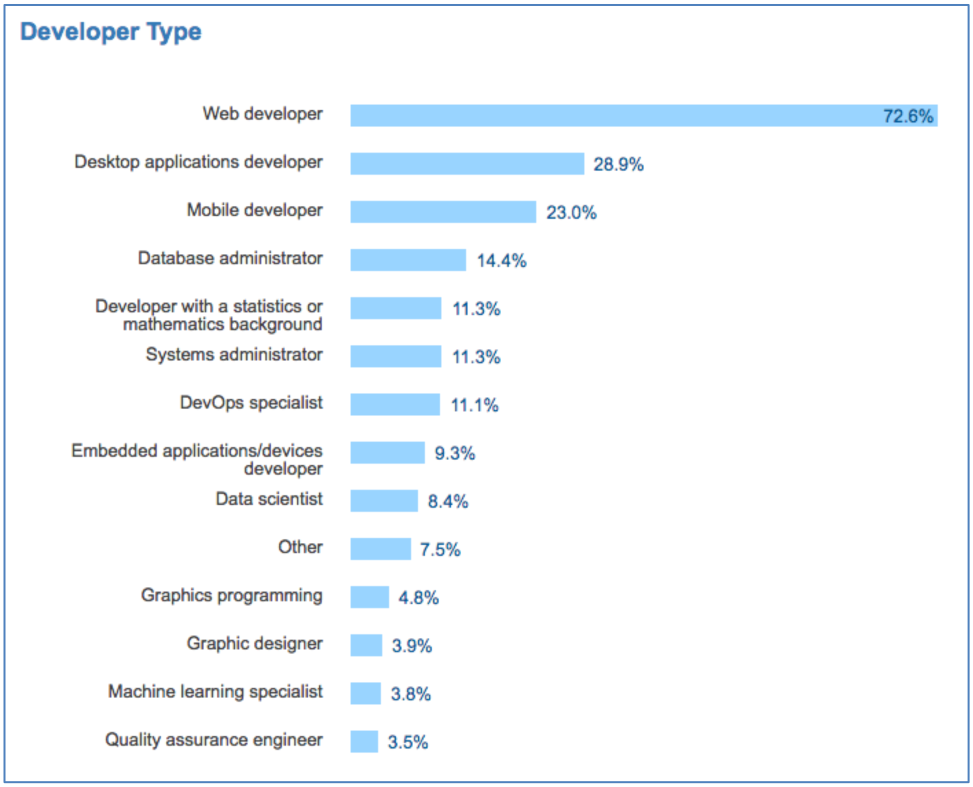


Figure 2-8. Stackoverflow 2017 survey, developer roles. Source: StackOverflow developer survey results (StackOverflow, 2017).

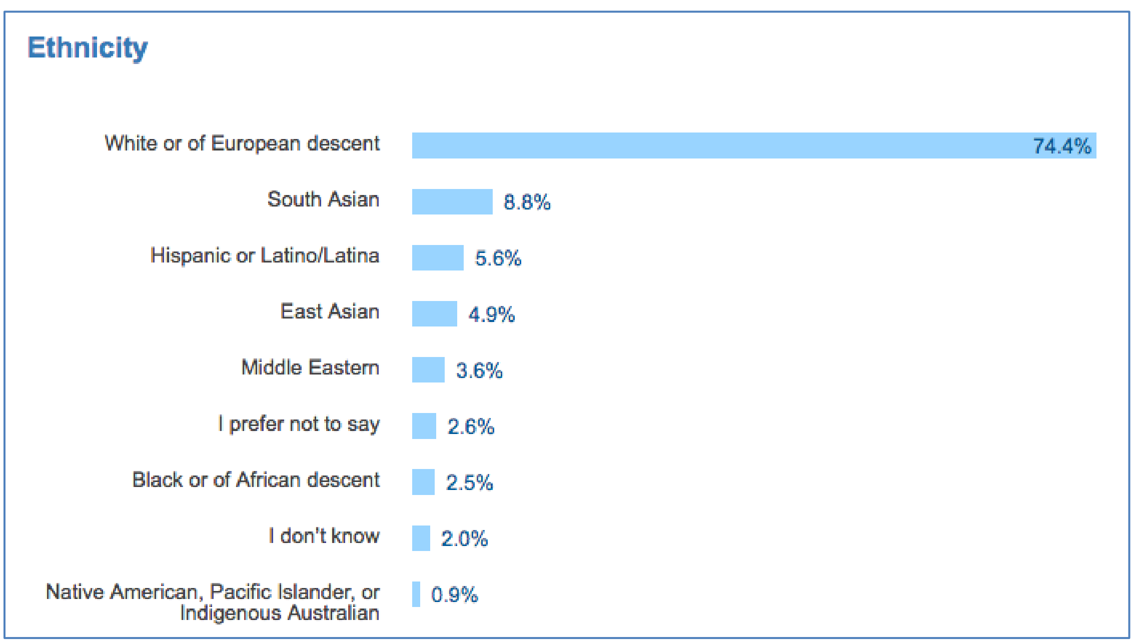


Figure 2-9. StackOverflow 2017 survey, ethnicity of the participants. Source: StackOverflow developer survey results (StackOverflow, 2017).

The other survey finding that could be interesting and useful for this study was that coding was not just a career or a job that needs be done. For developers, it can also be a passion. Programmers like to program even when they do not have to. The survey showed that 48% of the developers wrote a program as a hobby and 26.8% plus 5.9%, which is 32.7% of developers said they contributed to open source projects as shown in Figure 2-10. "Programming Enjoyment" was one of the variables in this study.

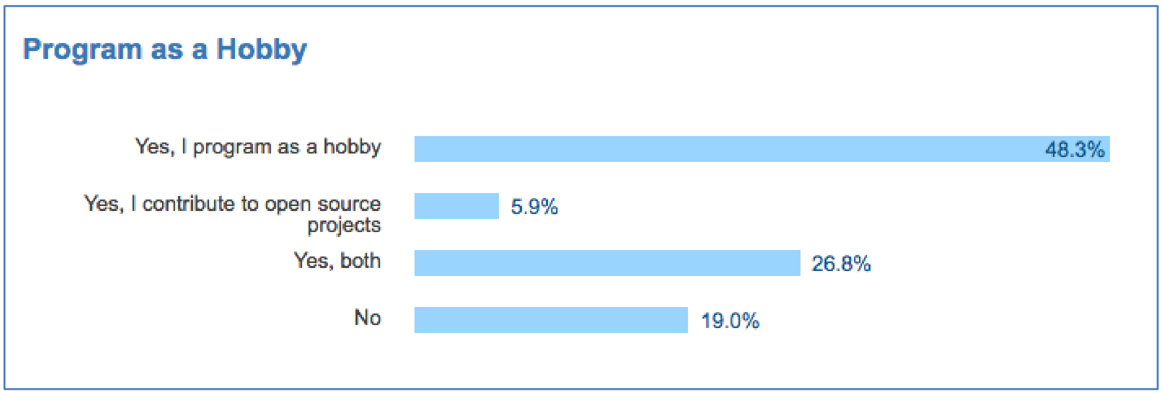


Figure 2-10. StackOverflow 2017 survey, program as a hobby. Source: StackOverflow developer survey results (StackOverflow, 2017).

The survey also showed that the most popular programming language that is used among developers was the JavaScript language. JavaScript language was used in this experimental study for its simplicity and popularity. It is also the language that is used in web development and, as shown in Figure 2-11, most of the developers are web developers. As defined by Study.com (n.d.), JavaScript is a programming language that is run by most modern browsers like Chrome, Firefox, and Safari. It supports object-oriented programming and procedural programming. In combination with HTML and CSS, JavaScript language is used in web development to control web pages on the client side of the browser. It can also be used on the server-side programs and in mobile applications. (Study.com, n.d.).

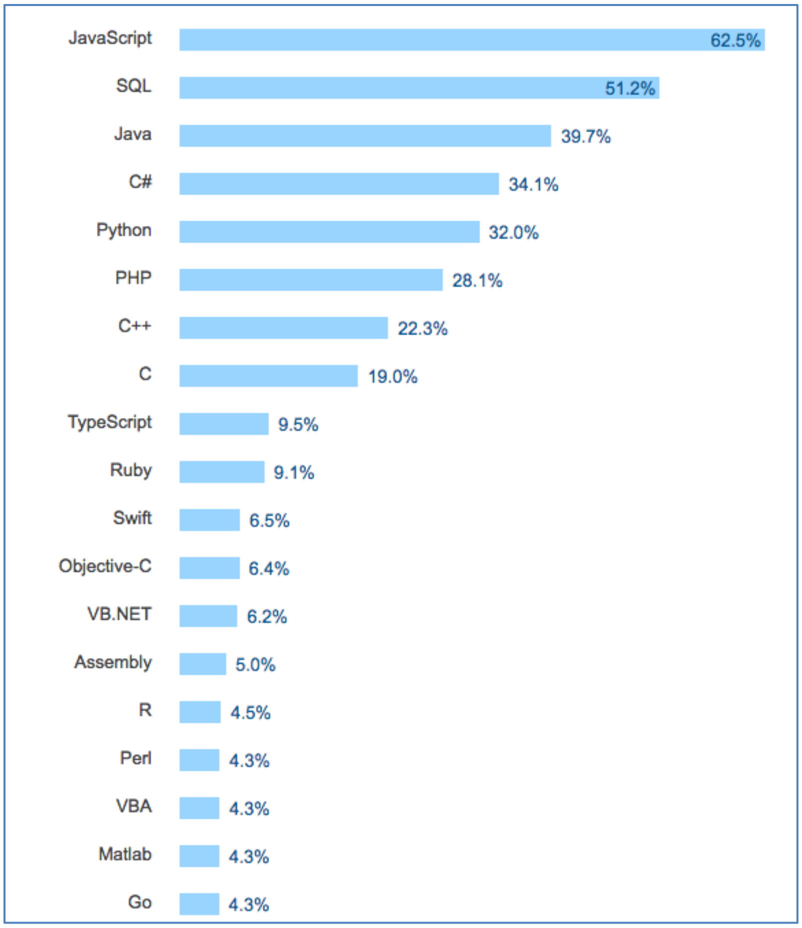


Figure 2-11. StackOverflow 2017 survey, popular programming language. Source: StackOverflow developer survey results (StackOverflow, 2017).

HackerRank is another gamification website where developers from all over the world compete to solve programming challenges to learn, to get badges and points, to find a good job, to share knowledge, or just for fun. (HackerRank, 2008, n.d.). HackerRank conducted a study with over 14,000 professional software developers (McDowell, 2018). Only 2,000 (14%) of the participants were women. The study showed that young women were 33% more likely to study computer science than women who were born before 1983. The study also stated that the gender gap in age of learning to code is slowly shrinking. Figure 2-12 shows this study finding.

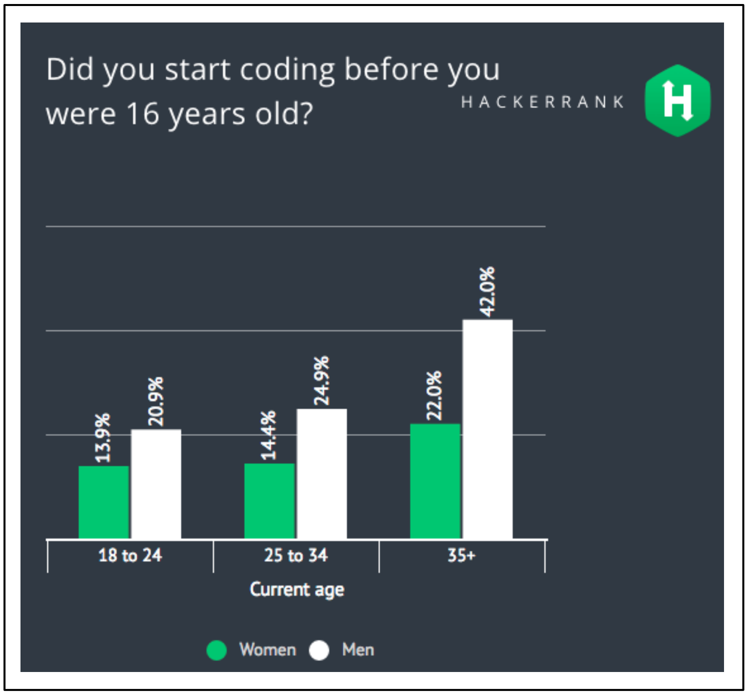


Figure 2-12. Age to learn coding, HackerRank study. Source: women in tech report (McDowell, 2018).

Another study was conducted to explore university students' intention to major in CS in the University of California Los Angeles or UCLA (Lehman, Sax, & Zimmerman, 2016). The study was conducted by analyzing the surveys completed by 187,717 freshmen at UCLA. The results showed that only 1,636 (0.87%) female students indicated an intention to major in CS, less than one percent of the 2015 freshman at UCLA. This number was small as compared to the number of female students who intended to major in biological science, which was 17,553 or 9.3%, while the percentage of the male students with the CS major intention was 2.3%, which was also small.

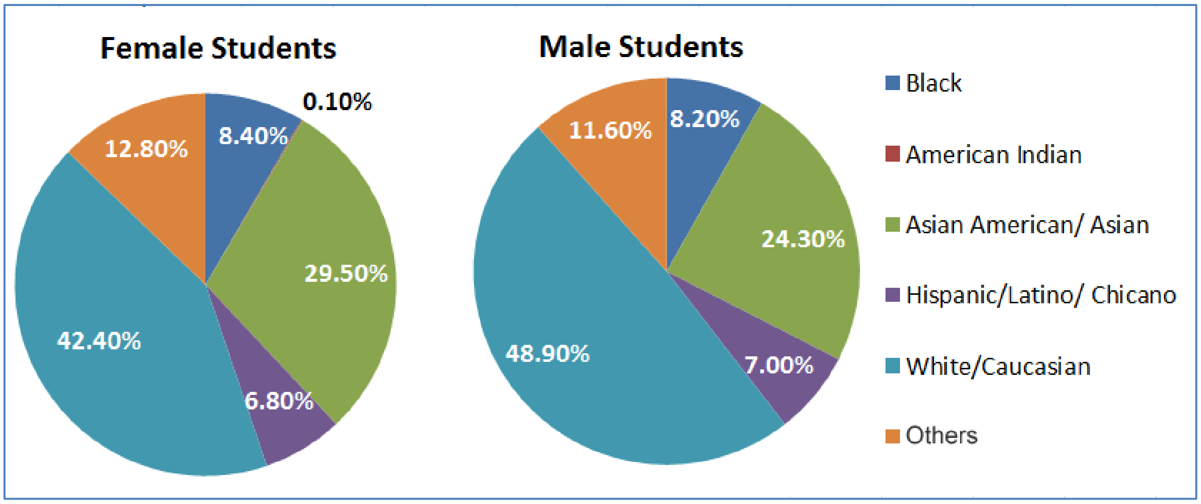


Figure 2-13. Freshman who intended to study CS at UCLA (Lehman, Sax, & Zimmerman, 2016).

Figure 2-13 shows the demographic data of freshman who intended to study CS at UCLA in the Lehman et al.'s (2016) study. The percentages of the male and female students were close for all racial groups. The Asian female percentage was a little more than the Asian male, and there was a small percentage of American Indian (0.10%) in the female students. The Lehman study was based on freshman's intention to pursue a degree in computer science, but they may have studied a different major. A similar study that analyzed the demographic data of the students who were actually in the CS major was conducted by a student in that department at Harvard University (Wu, 2015). The results showed that more than half of the students in the CS department at Harvard were Asian (53%), about 40% White, and only 3% were Black students. The female students' percentage was 27%, and most of those students were Asian. Another study by a CS student at Stanford showed similar results (Cueto, 2015). About half of the CS students at Stanford were Asian (46%) followed by White (38%). The percentage of the Black students at Stanford was double that at Harvard (6%). The female students' percentage (30%) was similar but a little more than that at Harvard.

2.4 Theoretical Framework

Three theoretical frameworks were used to help specify variables in the study and develop the survey questionnaire. These were motivation theories, the technology acceptance model, and the theory of planned behavior.

2.4.1 Motivation theories. Some items from the Motivated Strategies for Learning Questionnaire (MSLQ) (Ahmad, 2012; Pintrich, 1991) guided the survey items development that were used to measure motivation for this study. This included some scales such as *Intrinsic* Motivation and *Extrinsic* Motivation. The Intrinsic motivation scale measured the student's engagement in the learning process by internal reasons such as challenge, curiosity, mastery, or enjoyment (Ryan & Deci, 2000). A high rank on this scale showed an interest in learning a programming language. On the other hand, the extrinsic motivation scale measured the student's engagement in the learning process by external reasons such as grades, rewards, performance, and evaluation by competition with others. See Figure 2-14.

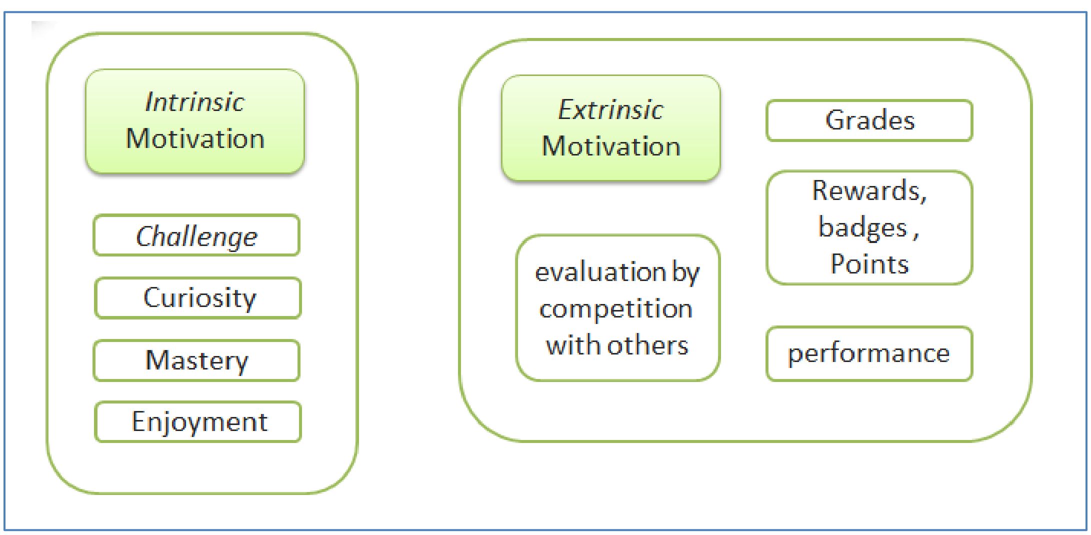


Figure 2-14. The motivated strategies for learning questionnaire.

2.4.2 Technology acceptance model. The other theoretical framework that guided the survey development was the Technology Acceptance Model (TAM). As shown in Figure 2-15, TAM is among a few models that include psychological factors that affect technology acceptance such as ease of use and usefulness of the technology (Davis, Bagozzi, & Warshaw, 1989). According to Toe (2010), some variables can be used from TAM to measure teacher's acceptance to an interactive learning environment. Similarly, this model could be used to measure student's acceptance to the same environment.

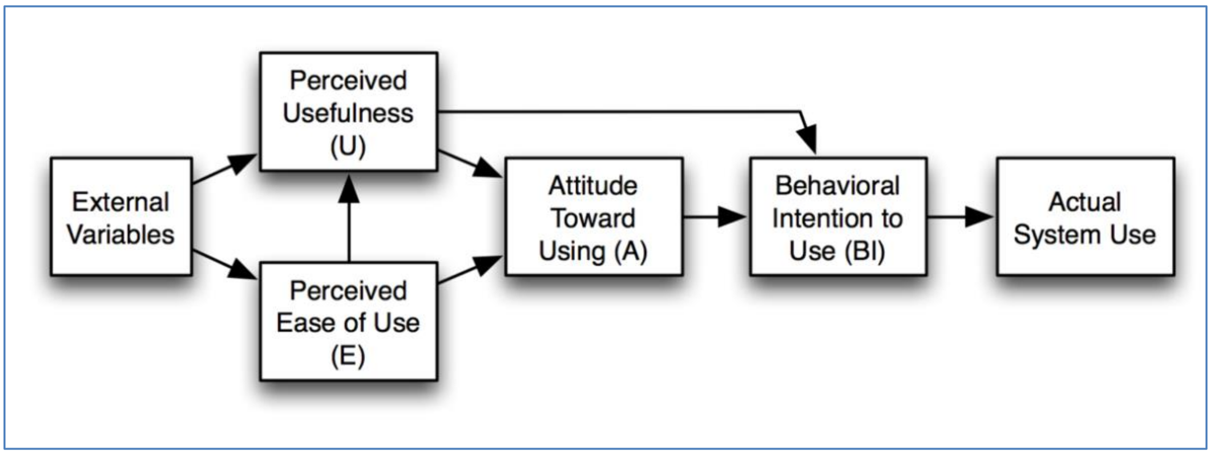


Figure 2-15.The technology acceptance model or TAM (Davis, Bagozzi, & Warshaw, 1989).

2.4.3 The theory of planned behavior. The third theory is the theory of planned behavior (Ajzen, 1991). This theory states that the intention of the individual to do something, for example pursuing a degree in computer science, is affected by three variables or predictors. These are attitudes toward the behavior, subjected norm, and perceived behavioral control, as shown in Figure 2-16. The attitude toward the behavior is how the individual thinks and feels about the behavior, which could be effective attitude and instrumental attitude. Effective attitude implies how an individual feels about the behavior. For example, is it enjoyable? Instrumental attitude is about how beneficial or harmful the behaviors are. Subjective norms are about the support from others that the individual might receive for the behavior. Subjective norms could be injective or descriptive. The injective norm is about encouragement and support from others. While the descriptive norm appears when others do the behavior, not only supporting it. The third predictor is the perceived behavioral control, which questions whether or not the individual is capable and confident to do the behavior and to what extent can he/she accepts challenges and overcomes barriers.

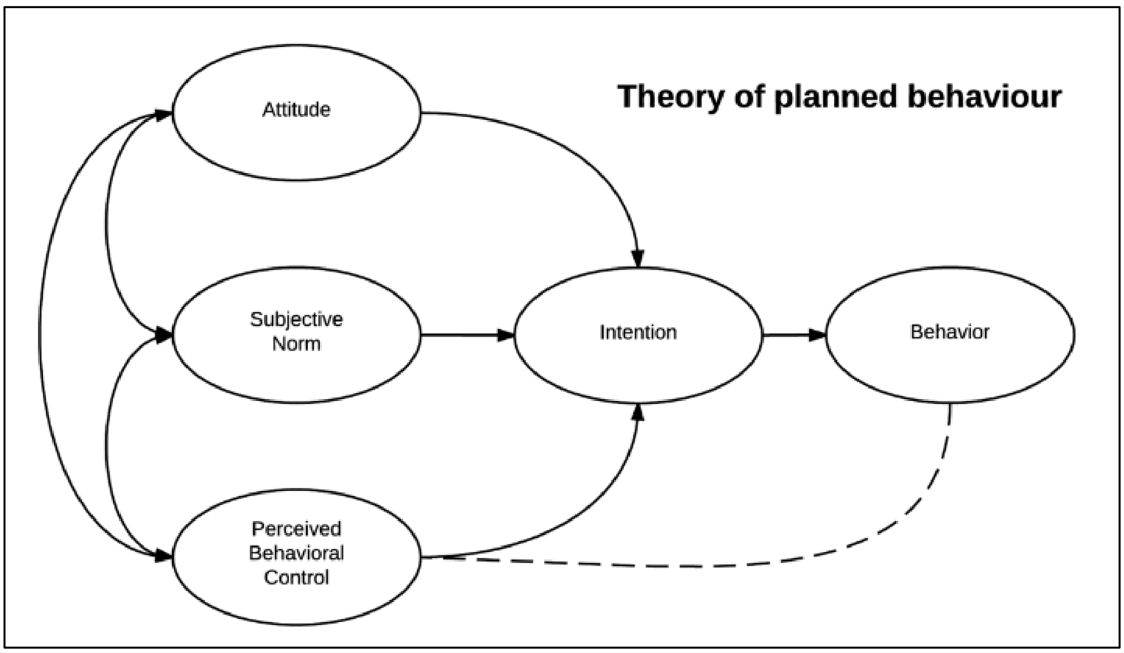


Figure 2-16. The theory of planned behavior (Ajzen, 1991).

2.4.4 Sharing as a motivation model. According to the theory of planned behavior, the intention to the action is affected by the social norms. This means if people around a person did some action, there is a probability that the person will do the same action. Using this concept, a model was developed by the researcher of this study to emphasize the effect of sharing on social media as a motivation for writing a program. In Figure 2-17, the cycle starts when User 1 writes a code using the developed tool then shares his code on the development tool itself or on the social media website so other users can see it. This user will act as a motivation transmitter to other users. When User 2, who is a friend of User 1, finds the shared code with a link to the online development tool, he/she will act as a motivation receiver. If the shared code or the art produced by the code was good enough to attract User 2, he/she will click the link and go to the development tool. If User 2 was able to write a good code and share it on social media, he/she will act as a motivation transmitter to motivate other users to write the code and share it, and the cycle will continue.

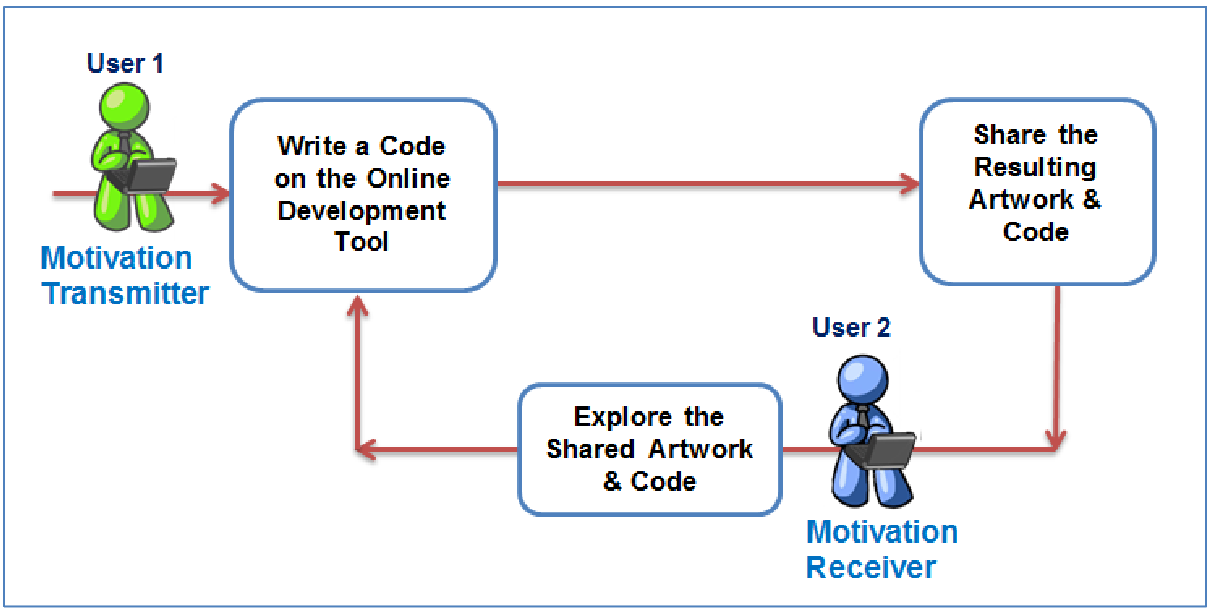


Figure 2-17. Sharing as a motivation model.

2.5 Block-Based vs. Text-Based Learning Environment

Block-based programming technique is an instructional strategy used in many applications and websites that teach programming like Scratch from MIT (Scratch, n.d.) and Code Studio from Code.org (Code.org, 2018). Many educators and designers support block-based programming as the best and easiest way to teach computer science. In the following paragraphs, an experimental study that was highly related to this dissertation study is discussed in detail. Weintrop and Wilensky (2015a) compared interactive block-based programming and text-based programming. The researchers followed the students' learning process in a selective CS course in a public high school in a Midwestern city. Three classes were followed for 10 weeks. Each class spent the first five weeks of the course using a block-based programming environment. The students then switched to Java text-based programming for the next five weeks and continued with Java for the rest of the school year. During the first five weeks, the teachers' role was limited. They followed a workshop style course where students work on assignments and ask questions if needed. Snap!, a block programming environment that is similar to Scratch but with more advanced features, was used. The three classes were defined as: read-only, read-write, and graphical. In the first class, a hybrid of block/text read-only environment was used where a student could right-click on the block to see the code, but he/she could not edit it. The second class used a hybrid block/text read-write environment where a student could define the behaviors of new blocks. The students in this class could read the text and define new blocks or copy and paste from existing block to a new block, but they could not write a code from scratch. In the third class, students could not see any text; they just used block. Survey and content assessments were administrated three times: by the beginning of the school year, by the end of the 5 weeks, and by the end of the 10 weeks. In the same three times, 27 student interviews were conducted to collect students' perspectives. The survey showed that most of the students found that block-based programming was easier, and the interviews showed that there were some reasons behind the students' preference to the block programming. The shape and layout of the blocks made it useful and easy for the students to differentiate the block usage and avoid mistakes. "It is like a puzzle," one student said (Weintrop & Wilensky, 2015a, p. 203). In addition, block-based programming was easier to read because it used more human language than computer language. Blocks were easier to compose with the drag and drop feature instead of writing the code and getting syntax errors. Furthermore, the graphical feature of blocks made it easy to remember how to do some tasks. In the block programming environment, all the blocks were available and organized in categories. This made it easy for the student to find the keyword that he/she wanted to use, while in the text-based programming the student had to learn the keywords before being able to use it. The researchers in this study mentioned that the block programming supported various cognitive aspects of programming activity. From the shape, color, and the category of the block, students could tell how and where it could be used. Researchers also mentioned that more topics were covered with Snap! than with Java within the same five-week period, which supported the notion that Snap! was easier to teach. Quantitative statistics of the same study (Weintrop & Wilensky, 2015a) showed that most students thought that block-based programming was easier and more enjoyable than the traditional text-based programming. However, interviews showed that there were three drawbacks of block-based. Students found block programming was less powerful, and they said that with Java, one could do a lot more. There was not a block for everything, and there could be somethings that were too complex to be in a block. The second drawback was that the block-based programming was slow to author and required more blocks. This means many blocks were needed to compose a program compared to fewer statements that did the same thing in a text-based program. One student found text-based more creative and quicker than the block. Another student stated that when there are many blocks to compose a program, reading an existing code became more confusing and difficult to manage. The third drawback was that the block-based programming was inauthentic. One student stated that Java was a real programming language while Snap! was just used to learn programming and nobody used it to develop a real program. Some students thought that block programming was just for beginners. Researchers of this study described block-based as a top-bottom language, while the real programming language was a left-to-right language. They considered all of the mentioned drawbacks as guidelines to improve the tools that should be used to teach programming for the older learner who wanted to develop skills to be used beyond the classroom, which was the case with high school students.

On the other hand, text-based programming languages like Java or ASP.NET are real programming languages that are used by professional developers to create software products. They are not as easy to learn as block-based programming, and students always have to deal with syntax and semantic errors that need to be fixed. Some students may find text-based programming not interesting enough or even frustrating and difficult to learn. The third type of learning environment is the hybrid environment, which combines the features of both block-based and text-based environments. Pencil Code is an example of the hybrid environment. It is an online open source tool that was developed to be a bridge between the two programming learning environments (Bau, Bau, Dawson, & Pickens, 2015). This tool will be discussed in detail in the Related Programming Environment section.

2.6 Other Related Studies

This section illustrates several studies that have been done on teaching programming using different development tools. The researcher of this dissertation study used some of the variables that were used by Al-bow et al. (2009) to measure some programming concepts. The second study that is discussed in this section was performed by DiSalvo (2014), where African American high school students were targeted. The third one is a comparison between visual and hybrid environments (Koitz & Slany, 2014). Another study that used the hybrid environment is also discussed (Bau et al., 2015). In addition to the Weintrop and Wilensky

(2015a) study that was explained in the previous section, another study by the same researchers that provides assessment tools is discussed in this section (Weintrop & Wilensky 2015b).

In a study funded by the National Science Foundation grant, Al-bow et al. (2009) found that the use of art and design in a project-based learning model increased student interest and knowledge in computer programming. A two-week summer camp was held in Denver, Colorado, to teach computer programming to 26 high school students. In this 10-day camp, students started with playing with a pre-made game to become comfortable with the Greenfoot development environment. Then they had to create their own game projects. Pre-post survey results indicated a strong increase in students' knowledge and confidence in writing a computer program. In a previous study done by the same researchers (Al-bow et al., 2008), their summer camp included four weeks training for eight teachers in addition to the students. Teachers were trained for two hours every day on how to use the Greenfoot open source development environment. The last week of the camp focused on the technology and information literacy, how to be educated for the 21st century, creativity, innovation and intellectual property.

DiSalvo (2014) conducted a study with African American high school students in Georgia. The study compared the use of drag-and-drop and the text-based programming environment. In this study, 12 Black male teenagers were hired as game testers by Glitch Game Testers. Glitch started as a research project conducted by the Georgia Institute of Technology and Morehouse College and is funded by the National Science Foundation grant. Students in this program work in summer in quality assurance for various companies. Students spent most of the day as video game testers and one hour a day in a CS workshop.

For the first four weeks, Alice visual drag-and-drop programming language was used to teach coding. While Jython, a text-based version of Python, was used the next four weeks. Interviewing the students showed that three students preferred Alice, six preferred Jython and three had no preference. One student, who preferred Jython, mentioned that Alice was too easy for people of his age and any kid can do the same thing. Another student said Alice was time consuming and if a user made a mistake, he would have to change everything to fix it. DiSalvo (2014) mentioned that the challenge of Jython text-based programming motivated learners of this age, and the difficulty that they faced made them proud of their accomplishment. One student stated that with Jython he could develop a game, and this was a dream come true. On the other hand, one student preferred Alice because he could break down problems into smaller problems, and the top-down design could be applied to anything in life. This showed that students were not just learning CS; they were also getting some problem-solving skills. With Alice, participants were able to explain the basic operation with confidence while they were less confident explaining functions or algorithms. In contrast, participants were more confident to describe examples of algorithms with Jython. Upon completion of the three-year Glitch program, 65% of the participants enrolled in a computer related field after graduating from high school.

Another empirical study that compared the visual and hybrid environment in educational programming languages for teenagers was conducted in Austria (Koitz & Slany, 2014). The researchers in this study conducted an experimental usability study to compare the formula manipulation in both a Scratch visual environment and a Pocket Code app, which was considered a hybrid environment that teaches programming. Their participants were 13 teenagers with an average age of 15.5 years. After training sessions, participants were asked to do four different tasks on both Scratch and Pocket Code. Screen and facial expression recording software was used to collect the data. The study found that the hybrid environment of the Pocket Code was easier to use and less time consuming and participants were able to complete the given tasks successfully (Koitz & Slany, 2014).

From the researcher's point of view, it is not very accurate to compare software on a computer device with software on a touch portable device. Koitz and Slany's (2014) study could be more valid if the two environments were tested using the same hardware device, so the difference would be with the software exclusively, and the confounding variable of the device difference would be eliminated. The touch devices could be more interesting to participants. In addition, the use of a mouse affects the timing needed to perform the tasks. However, since it was a usability test, both software and hardware could be considered in the comparison.

Furthermore, Bau et al. (2015) conducted a study using the Pencil Code hybrid environment. The study was conducted on a group of eight middle school students with four after school lessons. None of the students had any prior experience in any programming language or similar blocking environment. Researchers found that students used the text mode during 95% of the class time. They stated that students preferred the text mode over the block mode. In pretest, one student said that both modes were equally good, while in the posttest all students said they were both equally good. In another paper published by the same researcher (Bau, 2015), he found that from observation of the Pencil Code usage during two months, most of the students preferred to use the Block mode over the text mode, and 26% of the students used both modes. Bau (2015) conducted his study with 14 high school students with prior coding experience; 13 of them said they used text mostly or both text and blocks equally. While conducting the study again with five high school students who did not have prior experience, Bau (2015) found that three students preferred the Block mode and two students preferred the text mode.

Another study that helped to specify some variables of this dissertation study was conducted by Weintrop and Wilensky (2015b). These researchers published a set of commutative assessments that they developed to measure the students' fundamental programming concepts' understanding in two different learning environments: block-based, like Scratch and Blockly, and text-based. This paper was based on a study that was done by the same researchers (Weintrop & Wilensky, 2015a), which was discussed in the previous section. Weintrop and Wilensky (2015b) illustrated the assessment tool in detail that includes questions, such as the iterative logic questions, conditional logic questions, variable questions, function questions and comprehension questions. They found that students understood the conditional questions with blocks better than with text because the students thought that both branches of if-else statement would be run with text. For the iterative questions, students understood the word "repeat" of the block statement better than "for" which they interpreted as "can" rather than iterative keyword. Block for the variable question was also better; students understand (set__to__) better than (var=__). Similarly, with the function question, the different shape of block made students perform better in this question. In comprehension, students were given a piece on code in both block and text, and they were asked to find the output of the program. Only with the comprehension question, students' responses were the same for both block and text-based modalities. Researchers attempted to use these findings to design a new hybrid environment (Weintrop & Wilensky, 2015b).

2.7 Related Programming Environment

Programming is not an easy subject, and some students either drop-out of the computer science course or perform poorly (Lahtinen, Ala-Mutka, & Järvinen, 2005; Williams, Wiebe, Yang, Ferzli, & Miller, 2002). There have been many attempts to promote programming for high school students in order to make it easier and fun by building interesting learning environments. In this section, some of these attempts are discussed. General Purpose (GP) is a programming language developed by the same people who developed Scratch at MIT (Monig, Ohshima, & Maloney, 2015). It was developed as the next stage after Scratch block-based programming (Figure 2-18).

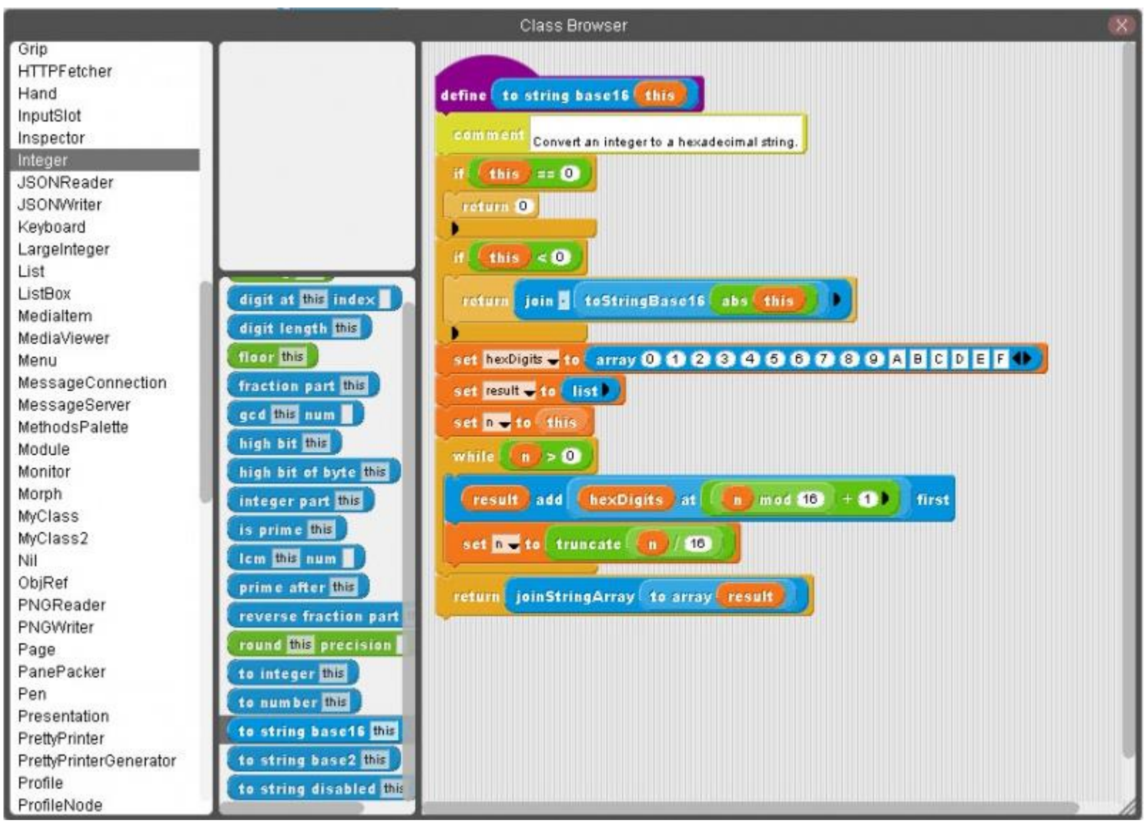


Figure 2-18. GP development environment from MIT (Monig, Ohshima, & Maloney, 2015).

With GP, novice programmers who started with block programming do not need to switch to the text-based environment as their ambition and capabilities grow. Block programming has two drawbacks: it takes more screen than text-based and modifying a long program with dragging and dropping is slower than modifying the text-based program. Monig el al. (2015) mentioned that taking more screen area makes it harder to see an overview of the code without scrolling. Additionally, colors and the graphical elements of blocks can be visually distracting as the program gets longer. Furthermore, browsing many categories and scrolling to find the desired block is time consuming for non-beginner block programmers. Developers of GP aimed to combine the benefits of blocks with the time efficiency and screen optimization of the text-based programming.

Pencil Code is another programming environment that was developed by software developers working at Google (Bau et al., 2015; PencilCode, n.d.). Pencilcode.net is an online open source tool that was developed to be a bridge between block and text-based programming learning environments. Developers stated that there are two ways to teach programming, either by using a simple and fun environment that helps beginners achieve results and avoid frustration or by teaching the language that is used by professionals. Pencil Code was designed to combine these two ways of learning so that learners can write code outside this environment after gaining confidence and experience using it. Pencil Code allows students to program using web languages like HTML, JavaScript, CoffeeScript, and CSS. It motivates beginners with turtle graphics, music, storytelling, tutorials, and networking. The block view can be toggled on and off so that the block will disappear and the text remains. This is done with smooth animation transition so that toggling between the two modes is fast and easy. The block editor component is called Droplet. With the Droplet data model, any programming language can have block interface using a language parser (Bau et al., 2015). The authors stated that Pencil Code can be used in classrooms ranging from Grade 6 to college. From the perspective of the researcher of this dissertation study, the tool is useful, and it is a hybrid environment except that the default language is CoffeeScript, which is not as common and popular as JavaScript. Figure 2-19 shows the sample code of this tool. However, the developer of Pencil Code mentioned that it also supported JavaScript, but the user had to choose it through the setting button.

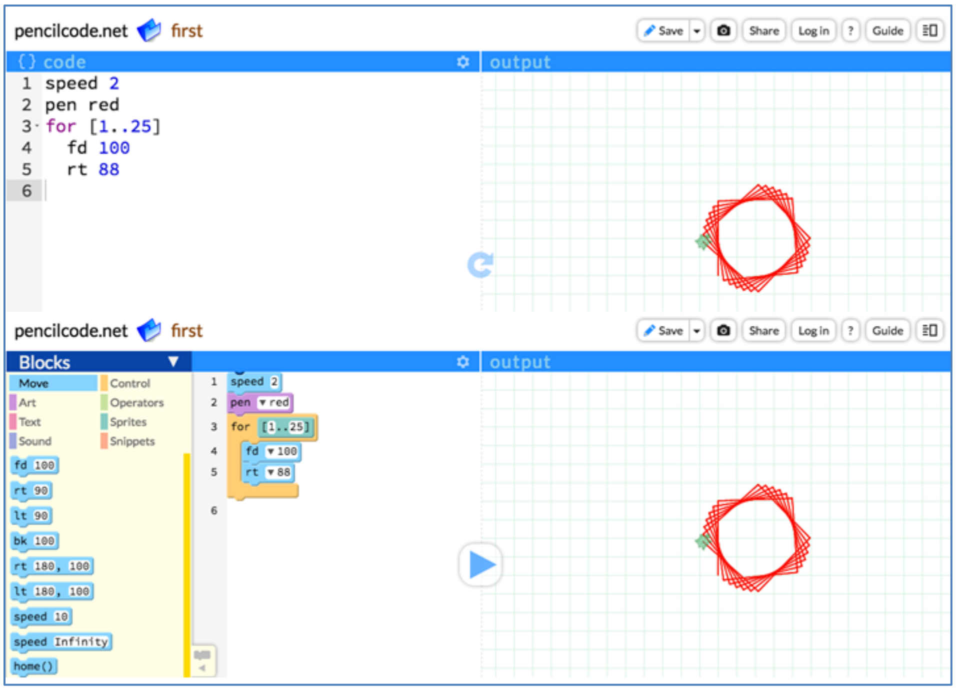


Figure 2-19. PencilCode.net online development environment (PencilCode, n.d.).

Droplet is a programming editor that was developed to close the gap between blocks and text-based programming (Bau, 2015). Bau (2015) mentioned the gap of confidence as a disadvantage of block programming. By using blocks, users cannot say confidently "I can write a computer program" and cannot communicate with the larger community of C, Java, Python, or JavaScript community. To close this gap, Bau (2015) created Droplet editor for Pencil Code. It was designed to load a text program that can be edited as blocks and then saved as text. In Pencil Code, a user can switch between text and block mode when writing a program. Bau (2015) illustrated the mechanism of converting text to block by using a language parser to insert block tags like HTML tags; for example “OK” will be converted to <block>if(<socket> OK</socket>)</block>.

Greenfoot.org is another development environment that was developed by researchers at the University of Kent in the UK. This environment was developed to combine the best between text and block programming (Brown, Altadmri, & Kölling, 2016). This study presented the design and implementation of a novel way to edit a program using a frame-based editor in the Greenfoot programming learning environment. Figure 2-20 shows a screenshot of this environment (Kölling, 2012). The editor supported Stride programming language, which was similar to Java. However, researchers said that this way of editing could be applied to different programming languages (Brown et al., 2016). The frame-based editor provided a hybrid learning environment combining the advantages of text-based with the structuring features of blocks. Brown et al. (2016) stated that this environment was tested in a study by McKay and Kölling (2013), where it was compared with other block environments like Scratch, Alice, and StarLogo TNG. Greenfoot frame-based environment showed better performance in terms of faster entry as compared to block-based environment. Brown et al. (2016) believed that this environment satisfied the needs of learners of different levels of proficiency. Frame-based editing supported navigation features that were not supported by block programming (Brown et al., 2016). The navigation feature allows the user move between the methods usage and the definition of programming objects. This activity is usually used by intermediate and professional programmers more than novice programmers. With block-programming, the syntax error can be avoided. Frame-based editing supports this feature by entering the correct text in the frame and setting the cursor for user’s editing. This minimizes the user’s syntax errors. Brown et al. (2016) aimed to improve learning programming environments. The suggested features of their frame-based editor could also be used to improve the professional IDEs.

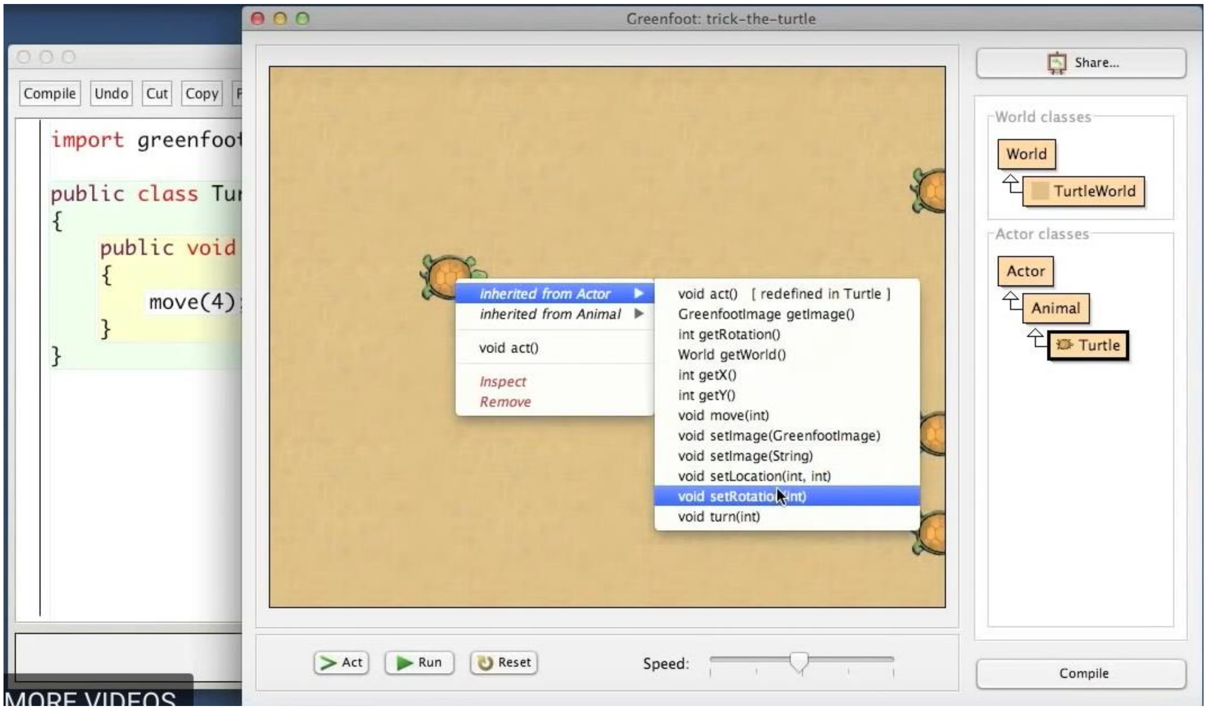
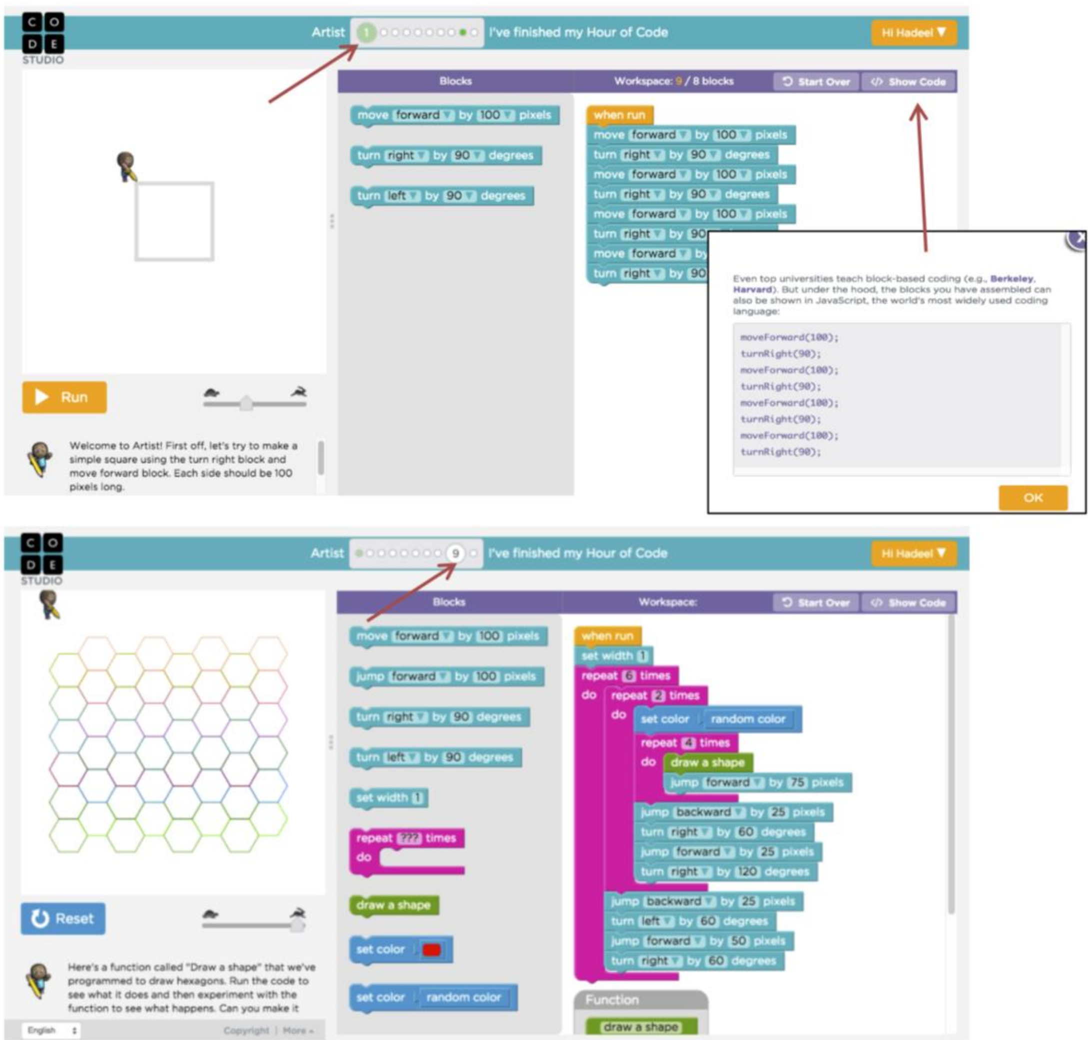


Figure 2-20. Greenfoot development environment (Kölling, 2012).

The researcher of this dissertation study did not find the environment easy to use for the novice user without training. In addition, simplicity is one of the usability guidelines that needs to be followed more adequately. Also, Greenfoot is not a web-based tool, and some tutoring is needed to install the tool and start using it.

One of the famous learning environments in STEM and in CS for high schools is the Hour of Code (HOC) environment provided by the Code.org organization. This kind of environment is also known as an interactive learning environment. The following figure shows an example of a task that is available on the Code.org website (Code.org, n.d.-b). Artist application is one of the activities provided by the HOC (Code.org, n.d.-a). In the first level of the artist application, a student has to draw the square using three block statements as shown. Figure 2-21 shows the first and the last levels in this application, which include several levels or activities that get more difficult through moving from one to the other. The student has to finish all the activities shown in the top of the screen, and, after finishing them, another task will appear and so on.



Source: https://studio.code.org/s/artist/stage/1/puzzle/1

Figure 2-21. An example used in the HOC event (Code.org, n.d.-a).

The "Run" button shows the implementation of the code step-by-step so that the student can debug his program and know the error location. Additionally, there is the "Show Code" button on the left of the screen that the student can always use to see his/her text code.

The artist application provided by Code.org is still a block-based environment where students are only viewing the code, but not writing a program with a real programming code.

2.8 The Role of the Keyboard in Programming

As block programming mainly uses drag-and-drop to build a program, it is considered a mouse-centric interface. This might be desirable for beginners but not for intermediate and expert users. Researchers in the computer science field (Brown, Kolling, & Altadmri, 2015) emphasized the role of the keyboard in programming and stated that students should get to use the keyboard when coding to prevent students from getting bored when they move to the intermediate and professional levels. They suggested activating the keyboard's role in the blocking environment. Drag-and-drop is time consuming. For example, eight blocks are needed to calculate the hypotenuse of a triangle Sqrt(x^2*y^2), and each block requires some settings, while in text-based it only requires 13 keypresses. In addition, a user might select the wrong blocks and should detach and re-attach the correct blocks. With formulas and mathematical calculations, the use of the keyboard is more efficient. Brown et al. (2015) stated that in block programming "the ease is outweighed by the lack of speed" from the intermediate and professional programmers' perspectives. If computing some mathematical operation would take time, some users may find it easier to calculate by hand or mentally without the use of a computer.

2.9 Discussion

The researcher gave responses to some of the literature illustrated above. In this section, the researcher will shed more light on some points that need to be considered in this study. From the literature review, different studies recommended using a hybrid environment for high school students. The tool developed for this study could be considered a hybrid type environment. Adding the features of categories, browsability, and ease of composing that exist in the block environment to the text-based environment these make the latter easy to use. At the same time, students still have to use the real programming language rather than just moving blocks to develop a program. In this way, students will not be surprised when they have to take a real programming course at a university. In addition to that, with block-based programming, the teachers' role is more limited, and the class will become a workshop. This could put teachers’ jobs at risk because anybody can monitor such a workshop while not anybody can teach coding.

Activating the keyboard feature in block programming is an important feature to be considered for those who prefer using the blocking environments. This is not only important for the high school students, but also for younger students who are used to pressing keys on a keyboard or buttons on a joystick when they play video games. Moving blocks with a keyboard aligns with the usability guidelines of the human-computer interface rules. In addition, all the block categories could be accessed with keys; for example, "Alt + R" could open the red block categories, or the learner could create his own shortcut for the frequently used block.

The Pencil Code discussed earlier is a hybrid environment that encourages students to program in both block-based and text-based programming language (Bau et al., 2015). However, it still focuses on the block programming more than the text programming. The default mode in the Pencil Code is the block mode. A user who uses the text mode would switch back to the block mode to find and add new programming keywords. The block mode has all the block categories while the text mode is still like any other text editor. The text editor mode is mostly used to view the code after building the program with the blocks. The author and the developer of Pencil Code states that the observation of the Pencil Code usage during two months revealed that most of the students preferred to use the block mode over the text mode and 26% of the students used both modes (Bau, 2015).

Moreover, CoffeeScript is the default language in the Pencil Code environment. One of the things that a user might face by changing the language to the JavaScript option is an error due to the existing default CoffeeScript example available when a user first accesses the tool, as shown in Figure 2-22.

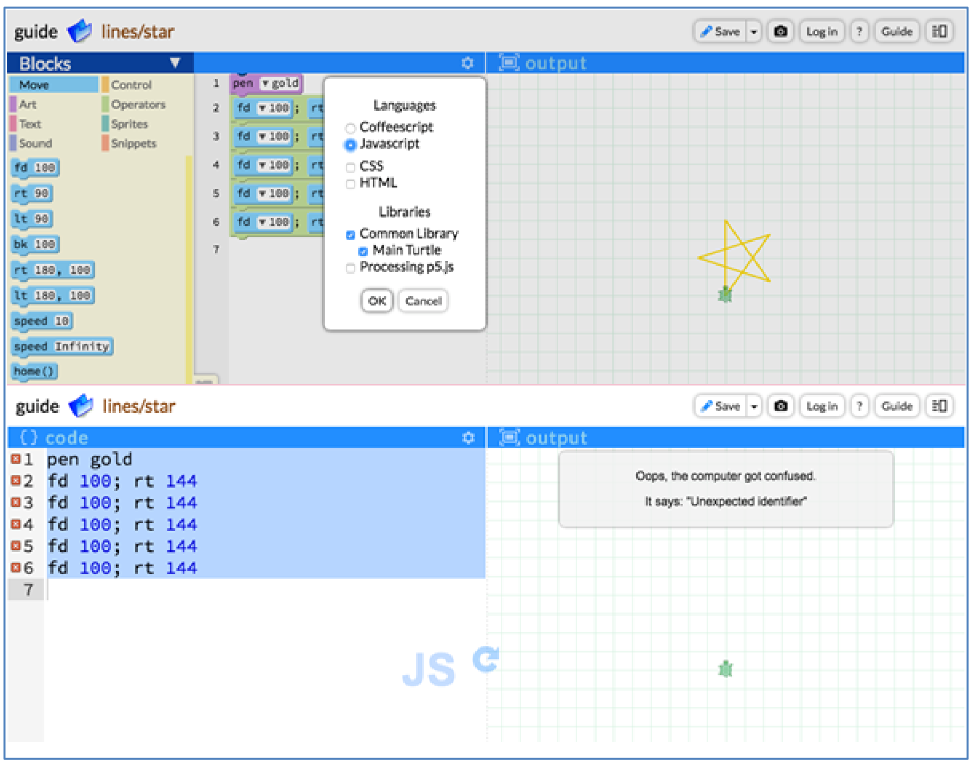


Figure 2-22. Pencil Code an error when switching to Javascript (PencilCode, n.d.).

In this case, the user has to delete the existing code and rewrite a new JavaScript code. It would be better if the code either switched to JavaScript or deleted with a warning message. Also, a new user cannot easily find the shared items. More examples or templates are needed to familiarize the user with the environment; however, the developers do provide an online guide. The other feature that is missing in Pencil Code is the use of meaningful function names. This feature, with the "Syntax Highlighting" feature, is used to improve the code readability and make programming easier for beginners. For example, Pencil Code uses "fd 100" which means move the cursor forward 100 pixels. The use of a more meaningful function name, like "MoveForward 100," would be more readable especially for beginners.

The Code Genie development environment focuses on encouraging students to use text-based programming, which is also a real programming language. The environment provides pre-written sample codes that the student can modify to learn before writing his own code. Code Genie emphasizes the role of the keyboard by encouraging students to write their own program with the keyboard as the major input device with the help of the mouse as an assistant input device. As discussed earlier in block-based environments, the mouse is the major device and the keyboard plays a minor role. This is not the case for professional developers who use the keyboard as the main device. Code Genie also uses meaningful function names and the "Syntax Highlighting" feature.

2.10 Characteristics of an Interactive Learning Environment

There are many characteristics to be considered in designing an interactive learning environment including human-computer interactive guidelines in designing the colors, shapes, and patterns of the environment's elements. One of the techniques that is considered effective in teaching computer courses is Studio-Based Learning (SBL). SBL is an instructional technique that emphasizes collaborative, design-oriented learning (Hundhausen, Narayanan, & Crosby, 2008). This pedagogy focuses on a learning-by-doing approach, and its high degree of interaction, collaboration, and feedback offers many advantages to the student (Boud & Feletti, 1997). Studies showed that the use of SBL increased students' enjoyment in problem-solving and raised their motivation levels and interest in computer programming. This approach engaged and excited students, and, at the same time, it effectively facilitated learning (Hundhausen, Narayanan, & Crosby, 2008). The other buzzword in the area of an interactive learning environment is "gamification." Gamification is used for adding gaming elements to the learning environment such as points or badges to make students more engaged and to increase their motivation (Morrison & DiSalvo, 2014). The term is also used when adding some gaming features for learning purposes to gamify the environment and make it more interesting. The environment that was offered by Code.org in the HOC events, for example, followed gamification and the studio-based learning instructional technique.

The following two sections discuss the HCI rules that guided the design of the Code Genie tool and the ADDIE Model that guided the design process of the tool.

2.10.1 Eight Golden Rules of Interface Design. To design any software tool, human-computer interface guidelines should be taken into account. The developed tool for this study was built with consideration for these guidelines and the Eight Golden Rules of Interface Design to improve the usability of an application (Shneiderman, Plaisant, Cohen, Jacobs, & Elmqvist, 2016). These guidelines include the following:

1. strive for consistency,
2. seek universal usability,
3. offer informative feedback,
4. design dialogs to yield closure,
5. prevent errors,
6. permit easy reversal of actions,
7. keep users in control,

8. Reduce Short-Term Memory Load.

2.10.2 The ADDIE Model. The ADDIE instructional design process model stands for analyze, design, develop, implement, and evaluate (Peterson, 2003). This model is used by many designers including software designers. As shown in figure 2-23, the first four phases are repeated whenever a new feature is added to a software product, and the evaluation phase works in the middle as it is typically considered in each of the other phases.This model was followed to build the Code Genie learning environment that was used as the treatment in this experimental study.

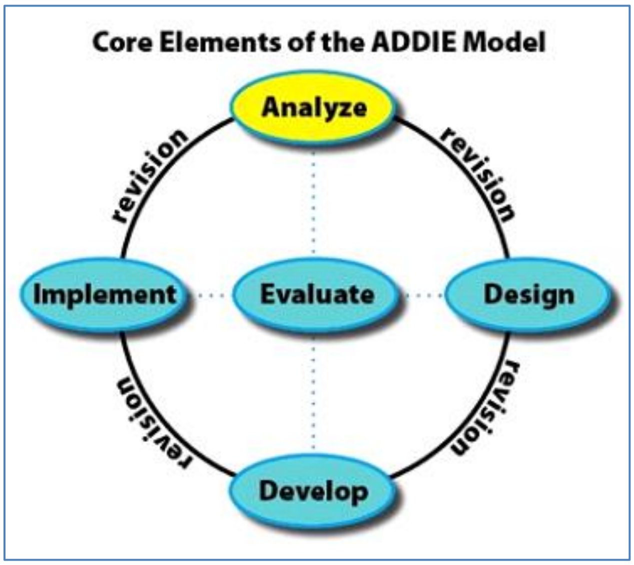


Figure 2-23. ADDIE design process model (Peterson, 2003).

Analyze. In the analysis phase, a problem should be clarified, the goals should be established, and the learning environment should be identified. In this phase, the targeted audience and their characteristics should also be specified along with the timeline of the project. Additionally, some questions could be addressed such as the following: What is the timeline for project completion? Is there a similar tool? Why this tool is needed? What technology is needed to implement this tool?

Design. The design phase could start with paper and pencil first; then a detailed wireframe or rich prototype could be built with the design or illustration tools. The prototype should include controls, navigation mechanisms, interface display, colors, fonts, style structure, and overall workflow. In this phase it would be useful to show the design to an expert for evaluation.

Develop. In this phase the needed technology, required skills, and cost should be determined. The material needed like the text, photos, and videos should be gathered. Next, the development should start with writing the code to build the tool and to put things together. To make sure that everything is working correctly, the software can be evaluated with user testing. Testing should include in-house testing, or Alpha testing, and end-user testing, or Beta testing. Results should be evaluated to see whether it is required to go back to the design and analyze phases or to proceed.

Implement: This phase involved launching the software tool, making it available to the end user, and letting people know about the product via social media or advertisement. Evaluation in this phase included tracking the product usage and performance by reviewing users' feedback and responding to them.

Evaluate. The evaluation phase consisted of two parts: formative and summative. Formative evaluation was present in each stage of the ADDIE process. Summative evaluation was done by usability testing with a large number of users and providing opportunities for feedback from them. There were different usability testing methods with specific laboratories for this (Shneiderman, Plaisant, Cohen, Jacobs, & Elmqvist, 2016). Testing could include the use of eye-tracking software, a "can-you-break-this" test, a "think-aloud" test, paper mockup testing, universal usability testing, and A-B testing. In the last one, two groups of users were given two different designs for the software product or two different designs for the toolbar of the tool and their responses are evaluated.

2.11 Summary

In this chapter, the theoretical frameworks and current literature on common styles of teaching novice learners about computer programming have been illustrated to help specify the variables needed for this study and to focus on the areas that need more research. Moreover, the characteristics of an interactive learning environment in addition to the human-computer interface guidelines and design processes have been studied to help in the design and the implementation of the treatment developed for this study. Chapter Three explains the research design and methodology. It also discusses the developed tool in detail.

Chapter Three: Research Design and Methodology

In this chapter, study variables hypotheses and research questions are illustrated first. The next sections explain the methodology, sample and population, data collection and survey design and validation, and data analysis. After those, the following section discusses the study validity that includes internal, external, and construct validity. The last section in this chapter illustrates the development tool that was used as the treatment in this study, in addition to its design and implementation process.

3.1 Hypotheses

Figure 3-1 shows the research design that includes all the study variables and hypotheses.

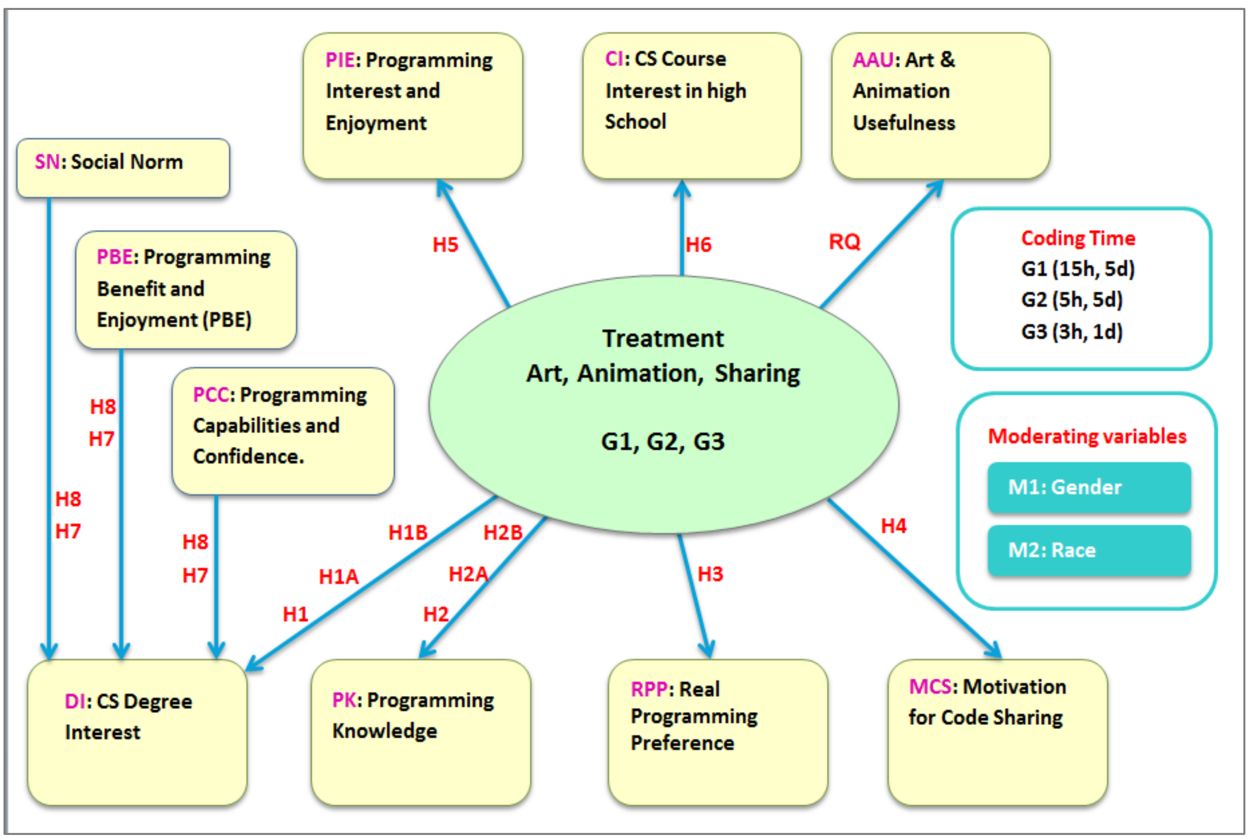


Figure 3-1: The research design.

The hypotheses that were tested in this experimental study are as follows:

H1: Integrating art and animation in teaching text-based computer programming increases students' interest in pursuing a CS degree.

H1$_o$: Integrating art and animation in teaching text-based computer programming has no significant effect on students' interest in pursuing a CS degree.

H1A: Integrating art and animation in teaching text-based computer programming increases female students' interest in pursuing a CS degree.

H1A$_o$: Integrating art and animation in teaching text-based computer programming has no significant effect on the female students' interest in pursuing a CS degree.

H1B: Integrating art and animation in teaching text-based programming increases the CS degree interest for students of different racial groups.

H1B$_o$: Integrating art and animation in teaching text-based programming has no significant effect on the CS degree interest for students of different racial groups.

H2: Integrating art and animation in teaching text-based programming increases students' knowledge in programming language.

H2$_o$: Integrating art and animation in teaching text-based programming has no significant effect on students' knowledge in programming language.

H2A: Integrating art and animation in teaching text-based programming increases female students' knowledge in programming language.

H2A$_o$: Integrating art and animation in teaching text-based programming has no significant effect on female students' knowledge in programming language.

H2B: Integrating art and animation in teaching text-based programming increases the knowledge in programming language for students of different racial groups.

$H2B_o$: Integrating art and animation in teaching text-based programming has no significant effect on knowledge in programming language for students of different racial groups.

H3: Integration of art and animation in teaching text-based programming increases high school students' preference to real programming language over block-based programming language.

$H3_o$: Integration of art and animation in teaching text-based programming has no significant effect on high school students' preference of real programming language over block-based programming language.

H4: Integrating art and animation in teaching text-based programming increases students' motivation to write and share more code.

$H4_o$: Integrating art and animation in teaching text-based programming has no significant effect on students' motivation to write and share more code.

H5: Integrating art and animation increases students' interest and enjoyment in text-based programming.

$H5_o$: Integrating art and animation has no significant effect on students' interest and enjoyment in text-based programming.

H6: Integrating art and animation in teaching text-based programming increases students' interest in taking a CS course in high school.

$H6_o$: Integrating art and animation in teaching text-based programming has no significant effect on students' interest in taking a CS course in high school.

H7: There is a statistically significant relationship between high school students' interest in pursuing a CS degree and Programming Benefit and Enjoyment (PBE), Social Norm (SN), and Programming Capabilities and Confidence (PCC).

H7$_o$: There is no statistical significant relationship between high school students' interest in pursuing a CS degree and Programming Benefit and Enjoyment (PBE), Social Norm (SN), and Programming Capabilities and Confidence (PCC).

H8: There is a significant prediction of high school students' interest in pursuing a CS degree by Programming Benefit and Enjoyment (PBE), Social Norm (SN), and Programming Capabilities and Confidence (PCC).

H8 $_o$: There is no significant prediction of high school students' interest in pursuing a CS degree by Programming Benefit and Enjoyment (PBE), Social Norm (SN), and Programming Capabilities and Confidence (PCC).

3.2 Research Questions

In addition to the study hypotheses, the following research questions were answered:

RQ1: What was the effect of integrating art, animation, and code sharing in teaching programming on the study variables for all students, for different genders, and for students of different racial groups?

RQ2: Was there any difference between the results of students with different amount of coding time?

RQ3: For high school students, which programming concept was easy, which was difficult, and which concept had the best improvement in the posttest?

RQ4: Was the Code Genie tool useful and easy to use?

RQ5: Was integrating art and animation in teaching text-based programming useful for high school students in understanding math functions, increasing their creativity and their programming skills?

RQ6: From the students' participation in the coding workshops, was there any difference in the students' interest to participate between different genders, and was there any difference among students of different racial groups?

RQ7: From the students' participation in the coding workshops, what was the percentage of the high school students who were interested in a free coding workshop?

3.3 The Study Variables

To test the study hypotheses, the variables shown in Table 3-1 were designed for this experimental study. These variables were measured and analyzed to accept or reject the hypotheses. The treatment and the coding time were the independent variables that affected the dependent variables shown in the Table 3-1. The last three variables in the table (PBE, SN, and PCC) were used as independent variables, and their relationships with the students' interest in a CS degree were tested in this study. Gender and race were used as the moderating variables for this study. Moderating variables influence the nature and the strength between the dependent and the independent variables (Leedy & Ormrod, 2013). For example, the level of enjoyment of using art with coding among female students may not be similar to that for male students and could also be affected by the different workshops' coding time.

Table 3-1 shows the variable name, the variable meaning, the variable type that indicates whether it is a dependent or independent variable, and the number of the survey items that were used to measure those variables.

Table 3-1
The Study Variables

Variable Name	Variable Meaning	Variable type	Survey Items
DI	CS Degree Interest	Dependent	5
PK	Programming Knowledge	Dependent	12
RPP	Real Programming Preference	Dependent	5
MCS	Motivation for Code Sharing	Dependent	11
IMCS	Intrinsic Motivation for Code Sharing	Dependent	6
EMCS	Extrinsic Motivation for Code Sharing	Dependent	5
CI	Computer Science Code Interest	Dependent	1
PIE	Programming Interest and Enjoyment	Dependent	5
AAU	Art and Animation Usefulness	Dependent	6
PBE	Programming Benefit and Enjoyment	Independent	7
SN	Social Norm	Independent	4
PCC	Programming Capabilities and Confidence	Independent	8
		Total	65

A 5-point Likert scale was used to measure the survey items for each variable, one for *strongly disagree* and five for *strongly agree*. The study variables were calculated as the averages of their specified survey items.

The Programming Knowledge (PK) variable was measured using students' scores for 12 programming questions. Each question was used to measure one programming variable. The letters PV stand for Programming Variable, and those variables are the sub-variables in this study. Table 3-2 lists those programming variables and their meanings.

Table 3-2
The Programming Knowledge Variable List

Variable Name	Variable Meaning
PV1	Understanding Variable Assignment
PV2	Understanding Variable Addition
PV3	Understanding Variable Multiplication
PV4	Understanding the for-loop statement
PV5	Understanding the if-statement
PV6	Understanding the if-else statement
PV7	Understanding the if-else statement (with art element)
PV8	Understanding the for-loop (with art element)
PV9	Understanding the switch-statement
PV10	Understanding the Math Function
PV11	Understanding the concept of Arrays
PV12	Understanding the concept of Function

3.4 Methodology

A quasi-experimental methodology was used for this study to explore how integrating art, animation, and code sharing in teaching text-based programming affects high school students' interest in pursuing a degree in CS and their programming knowledge. The study also included exploring the effect of the treatment on variables such as students' preference to real programming language, their motivation for code sharing, their programming interest and enjoyment, and their interest in enrolling in computer programming courses in high school. The art and animation usefulness were also explored in this study.

Furthermore, the study explored the relationship between the three factors that were suggested by the theory of planned behavior and the students' interest in a CS degree. These factors included students' programming enjoyment and whether they can find any benefit in a CS degree. The second factor was the social norms, which included the support and encouragement of the people around a student for this major. The third factor was the students' capabilities and confidence to overcome programming difficulties and their acceptance to the challenges.

The study aimed to answer the research questions listed earlier and to accept or reject the listed hypotheses. The advantage of an experimental study was that it provided real evidence to support the acceptance or rejection of the hypotheses.

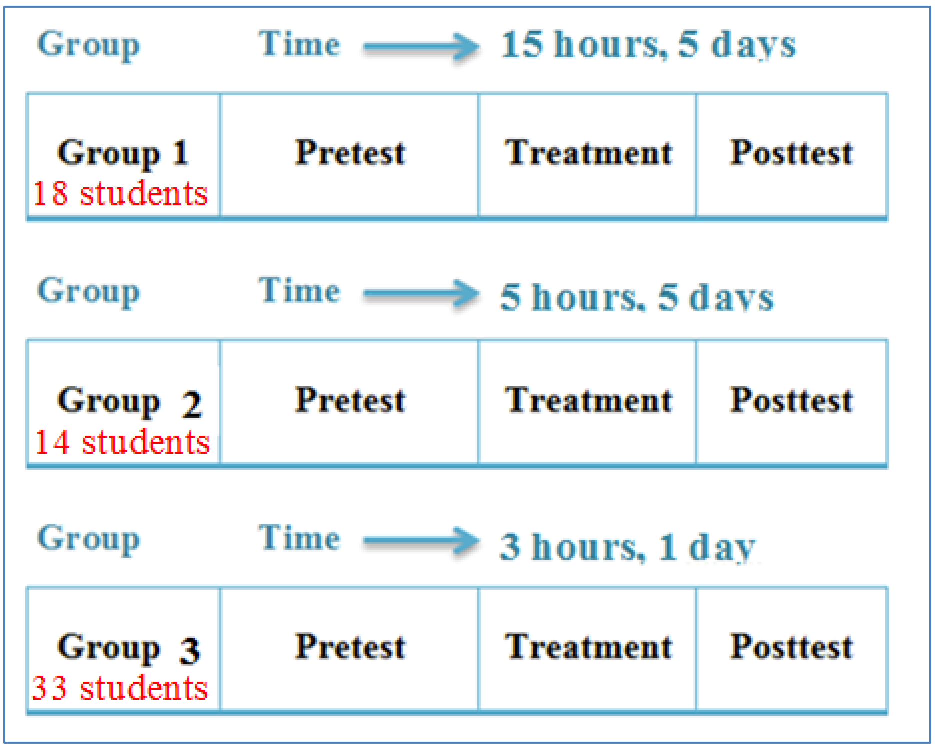


Figure 3-2. The research method.

One of the experimental designs that are mentioned in Leedy and Ormrod (2013) is the one-group pretest-posttest design. Figure 3-2 shows a modified version of that experimental design, which was used in this experimental study.

Three coding workshops were used to target high school students with different coding activities. A pretest–posttest survey questionnaire collected data at the beginning and at the end of the workshops. A classroom setting offered the environment where all students completed the online surveys together in a quiet room. The researcher administrated the data collection process and answered all students' questions.

Three groups of high school students (G1, G2, and G3) participated in three coding workshops/camp. The first was a five-day coding summer camp with three hours every day and 15 hours in total. The second was a five-day after school coding workshop with one hour every day and five hours in total. The third was a one-day coding workshop with three hours coding time. For G1 and G2, the pretest survey was conducted in the first day of the camp and the posttest survey was conducted in the last day of the camp. For G3, both tests were conducted on the same day. Students in G1 and G2 attended the workshops upon their own interest, while G3 student were exposed to coding as a school activity in the week of computer science (Computer Science Education Week, n.d.).

Pre- and posttests were used to measure all the study variables except for the last three variables shown in Table 3-1 which are PBE, SN, and PCC. The posttest was only used to measure these three variables.

3.5 Population and Sample

The population is all the high school students in the USA. The sample was 65 high school students from Ann Arbor public and private schools.-According to the National Center for Education Statistics (as cited in DiversityData.org, 2011), Ann Arbor is quite a diverse city. Table 3-3 shows the students' enrolment diversity for the 2010-2011 year (DiversityData.org, 2011). The diversity of Ann Arbor could reduce the threat of external validity problems and make it possible to generalize the research results.

The existence of the University of Michigan could be considered one of the reasons for Ann Arbor's diversity. Students from around the world and from all states move to Ann

Arbor to study in the highly reputed University of Michigan, and some of them choose to work and stay in Ann Arbor.

There are five public high schools in Ann Arbor that include Community High School, Pathways to Success Academic Campus, Huron High School, Pioneer High School, and Skyline High School. The following sections will explain more about the selected schools for this study.

Table 3-3
Composition of Ann Arbor Public School Enrollment by Race/Ethnicity

Race	Percentage
Non-Hispanic White	65.2%
Hispanic	4.3%
Non-Hispanic Black	19.3%
Non-Hispanic Asian/Pac. Islander	6.8%
Non-Hispanic American Indian	0.4%
Non-Hispanic Multi-Racial	3.9%

3.5.1 Pioneer High School. Pioneer High School was selected to conduct the study because it ranked academically as the first school in Ann Arbor, and it is number eight in Michigan rankings and number 556 in national rankings (Pioneer High School, 2014). Similar to other high schools, it accepts students from 9th grade to 12th grade. According to the statistic of the (2014-2015) school year, the total number of enrolled students was 1,671, with around 400 students in each grade. Figure 3-3 shows the number of enrolled students in each grade and the students' diversity. Sixty-six percent of the students were White, and 40% included all minority enrollments. It also shows that 51% were male students and 49% were female students (Pioneer High School, 2014).

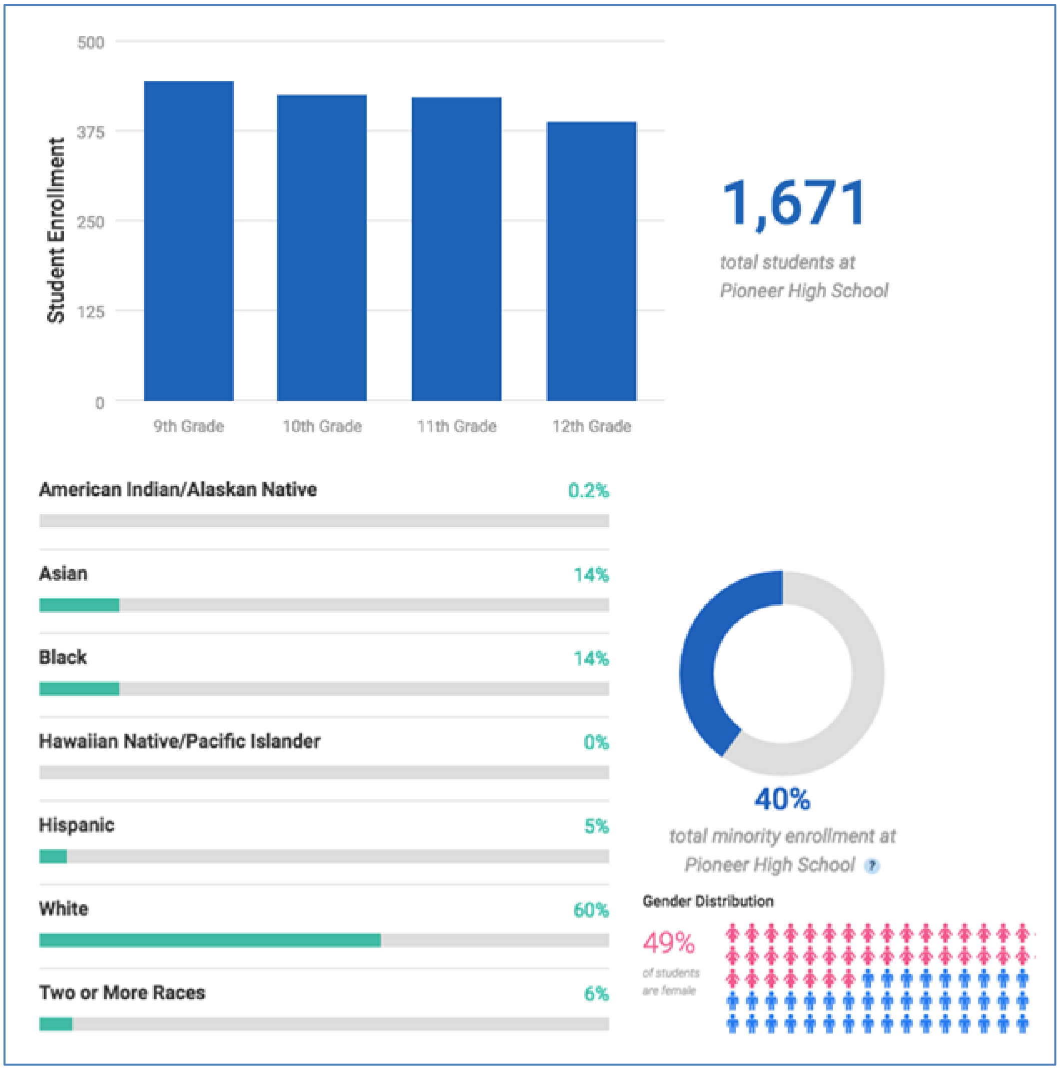
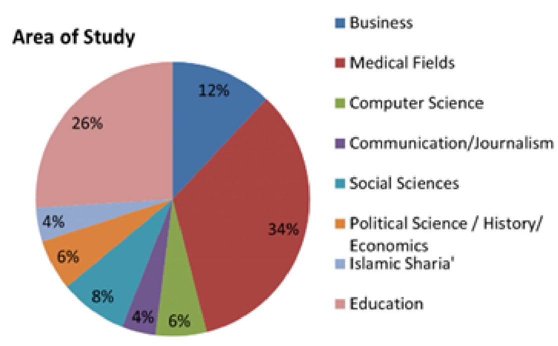


Figure 3-3. Pioneer High School students enrollment (Pioneer High School, 2014).

3.5.2 MIA private school. MIA, or Michigan Islamic Academy, is a private school in Ann Arbor, Michigan that serves the Muslim community in the area. The school is approved by the Michigan State Department of Education and accredited by AdvancEd (MIA-aa.org, n.d.). The demographic data for the school year 2016-2017 stated that most of the students were Middle Eastern with 64%, followed by Asian Indian students with 21%. The African American students were only 3%, 7% biracial students, and 5% listed as "Other" (MIA-aa.org, 2017). Figure 3-4 shows that most the graduates from MIA go to Eastern Michigan

University and most of them choose to study in the medical field, while only 6% choose the computer science major.



Country	State	College	Number
The United States	Michigan	Eastern Michigan University	31
		Michigan State University	1
		Central Michigan University	1
		University of Michigan	11
		University of Michigan - Dearborn	18
		Washtenaw Community College	4
		Wayne State University	2
	New York	Columbia University	1
	California	Zaytuna Institute	1
	Ohio	Youngstown State University	1
		Toledo University	1
	Georgia	Emory University	1
Japan		University of Tokyo, Japan	1
Jordan		American University of Amman	1
South Africa		Nelson Mandela Metropolitan University	1

Figure 3-4. Distribution of MIA Graduates by College Attended (MIA-aa.org, 2017).

3.5.3 Human subjects approval. Before conducting the study, an approval from the institutional review board (IRB) of Eastern Michigan University was required. To get the IRB's approval, high school approvals were needed. The researcher met with the principals of both schools to explain the study nature and get the schools' approvals. After getting the IRB approvals, consent agreements were required from students and their parents to allow the researcher to use their answers on the survey and for participation in the experimental component/training. The participant recruitment process is explained with more details in the next section. To maintain confidentiality, students' names did not appear in the data collected. In addition, the records were kept private and stored, and only the researcher has access to them. The result of this study will be shared with participants and to the public through publications and conferences.

3.5.4 Participant recruitment process for G1 and G2. After getting the approvals from the Human Subject Review Committee at Eastern Michigan University (see Appendix A), an email was sent to all Pioneer High School students and their families through the school messenger. The email contained the camp and the workshop details including the date, place and registration link. Emails also included flyers shown in Appendix A. The camp and the workshop were under the name "Art with Code," and they both were announced through emails several times with other school events to get the attention of as many parents as possible. The school cooperated nicely by accepting to host the camp and the workshop, and by sending the email announcement several times.

The Pioneer students were informed about the study purpose and the pretest-posttest survey and confidentiality before attending the coding summer camp and the fall workshop. The students and their parents were asked to sign the consent agreement to be able to participate in this study. Appendix A shows the consent agreement form that was included in the registration link. The students who attended the summer camp (G1) and the students who attended the fall workshop chose to participate upon their own interest.

The summer camp date was from July 10 to July 14, 2017, from 9:00 a.m. to 12:00 p.m. The after-school fall workshop date was from October 23 to October 27, 2017, from 2:44 pm. to 3:44 pm. The summer camp and the fall workshop took place in the computer lab at Pioneer High School.

The total number of students who attended and stayed to the end of the camp and the workshop were 32. The number of registered students for the summer camp was 27, but only 19 students actually participated. The number of the students who registered for the fall workshop was 21, but those who actually attended the workshop were 14.

3.5.5 Participant recruitment process for G3. After conducting the study for the second time with the fall workshop, the data were collected from 32 students. To increase the collected data and to be able to generalize the results, another coding workshop was arranged in December, 2017. The administrator of MIA private school in Ann Arbor was asked to host a one-day coding workshop. After getting the school approval, the researcher had to get the human subject approval from Eastern Michigan University again for subjects' participation; as they were not included in the original IRB application. The consent forms attached with the flyers (Appendix A) were given to all sophomore (10^{th} grade) and junior (11^{th} grade) students. The given form and flyer contain information about the study, the date, the time, and the place of the workshop. Students were asked to return the forms after having them signed by them and their parents before the workshop date. In this group, all students were asked to attend the workshop whether they were interested in coding or not. The workshop was held as a school activity during the first week of December, which was the computer science week, where many schools expose their students to computing skills (Computer Science Education Week, n.d.). This workshop took place on December 6 from 12:00 p.m. to 3:20 p.m. Thirty-four students attended the workshop. The time was less than four hours in this workshop, but the researcher tried to cover as much as possible of the most important programming concepts. Similar to the case in G2, the same teaching material was used, but it was more condensed for this workshop. However, the concepts of integrating art and animation with coding, and the concept of code sharing were covered adequately.

The students of G1 and G2 got more time to share their code and artwork. They had the chance to write some code and share it when they returned home because the workshop was a weeklong in both groups. In the third group, students had less time to share their artwork; however, each student had enough time to share a few artwork examples. The pretest and the posttest took place on the same day as the coding activity.

3.6 Data Collection

Pretest and posttest survey questionnaires were used to collect the data. The data was collected through an online pretest-posttest survey questionnaire on the first and ~~in the~~ last day of the coding camp/workshop. Google Forms was used to collect data with the survey questionnaire shown in Appendix B. The survey questionnaire was the measurement instrument that was developed to collect the data for this study. Some of the survey items were modified from other studies such as the Al-bow et al. (2009) and Pintrich (1991) studies, and some items were developed using the variables of the theoretical frameworks: the motivation theory (Ryan & Deci, 2000), the theory of planed behavior (Ajzen, 1991), and the TAM (Davis, Bagozzi, & Warshaw, 1989).

After conducting the study, the researcher received 130 (65 times two) Excel files for the pretest and posttest of each student. Since students were asked about their names, gender, and race, their information was kept confidential. All the names were hidden and coded names were used to hide students' identity.

3.6.1 The survey questionnaire. The developed survey questionnaire is shown in Appendix A, and it consists of several sections. Some sections include more than one part and some include only one part, and they are as follows:

1. Section One: This section was Part 1, which was used to collect the demographic data. This included student's name, grade, and gender and race questions. Race and Gender were used as the moderating variables for this study. The race options in the survey questionnaire were ordered alphabetically and included the following: American Indian or Alaska Native; Asian, Asian Indian; Black or African American;

Hispanic or Latino; Native Hawaiian or other Pacific Islander; Middle Eastern, Arab, or Persian, and White.

2. Section Two: Includes Part 2, which had 12 programming questions. These questions were used to measure students' knowledge in programming by measuring their understanding of different programming concepts. These were variable assignment, variable addition, variable multiplication, if-statement, switch-statement, for-loop, arrays, functions, and math functions. Some of the questionnaire items were modified from a previous study (Al-bow et al., 2009) and other questions were developed by the researcher to address the study variables. Multiple-choice options were used to answer the programming questions. The choices had the right, wrong, and the "I Don't Know" options.

```
n1=0;
n2=2;
n3=3;
for ( i = 1; i <= 4; i++)
{
    n1= n1 + 1;
    if (n1 > 4)
        n2 = n1 * n2;

    if (n2 == 2)
        n3=10;
    else
        n3=5;
}
```

Figure 3-5. One of the programming questions.

Programming questions were developed to follow the programming editor style or with the "Syntax Highlighting" feature where keywords and constant numbers have different colors for better code readability. In addition, the programming statements, such as if-else, for-loop, and switch statement, were written in the programming format or "Indentation Style." Logical and consistent indentation also increased the code readability, and this style was followed by professional programmers in the software industry. Figure 3-5 shows a code example that was included in this survey part.

3. Section Three: This section included four parts for Part 3 to Part 7 of the survey questionnaire, and it was used to measure eight study variables. These were DI, RPP, MCS, IMCS, EMCS, CI, PIE, and AAU. A 5-point Likert scale was used to answer questions in this section and in the following sections. Students were asked about their previous programming experience and skills to measure their understanding of the difference between block-based and text-based programming languages and their language preference. Other survey items were used to measure students' interest in taking programming courses in high school and their interest in pursuing a degree in CS after graduation. It also measured students' overall interest and enjoyment in programming. This section also included questions that were used to explore the effect of integrating art and animation with programming on students' interest and motivation toward CS, students' knowledge, and creativity from their perspectives. Questions measured the effect of using art and animation on students' motivation for code sharing. This included intrinsic and extrinsic motivation questions about collaboration between students, enjoyment of code sharing with others, and students' interest in coding. The last part in this section was designed to explore the usefulness of integrating art and animation with coding in understanding the math functions, increasing students' creativity and their programming skills.
4. Section Four: To explore the other factors that could affect the students' interest in pursuing a degree in computer science, planned behavior theory was used to guide the survey items designed for this section. Part 8, Part 9, and Part 10 of the survey

questionnaire were included in the posttest only and were used to measure student's attitude, social norms, and student's capability and confidence toward computer programming. In this section, students were asked about other factors that could increase their interest in a CS degree, such as how they feel about the CS degree and if it is beneficial to them. Questions were asked about social norms that indicated if students were encouraged by their family to pursue a CS degree or if they had siblings who were studying CS. Moreover, questions asked about behavioral control toward a CS degree such as students' confidence in their ability to complete this degree, even if they faced difficulties, and their acceptance to programming challenges.

5. Section Five: had one part, Part 11 that was only included in the posttest survey and it asked students about their overall experience in the coding camp/workshop and their opinions about the development tool. This part asked student about the ease of use and usefulness of the tool. It also asked them about their favorite feature of the tool. This section included 5-point Likert scale items, multiple choices items, and open-ended items for students' comments. This could help the researcher to improve the tool for future study and to introduce it to other high schools since the results revealed its usefulness.

3.7 Data Analysis

The data was analyzed to test the hypotheses and answer the research questions. The researcher performed descriptive statistics to measure normality, central tendency, and frequency. The demographic data was used to the effect of gender and race moderating variables. For example, the percentage of gender and race of students who were interested in a CS degree in the pretest was compared with the posttest percentage.

Since the experiment included repeated measures for the same group of students, a paired sample t-test was used to analyze the results and to compare the two responses for each student in the pretest and posttest (Landau & Everitt, 2004). The results of the pre- and posttests were compared to accept or reject the hypotheses. Null hypotheses were rejected when the results were statistically significant. Gender and race were the moderating variables in this study. The results were analyzed for the entire group, $G_{1,2,3}$, first, then G3 vs. $G_{1,2}$, and then for each group.

The paired sample t-test analysis was used to analyze the pretest and the posttest results for the six study variables (DI, PK, RPP, MCS, PIE, and CI). Appendix C shows the normality tests that is one of the t-test assumptions. The "skewness" for all variables was less than 0.8 and the "kurtosis" was less than 2 (George & Mallery, 2010). The histograms indicated that the data is approximately symmetrical. The assumptions for the t-test include that the dependent variables must be continuous, dependent observation or each subject should have two measurements, random sampling, and the differences between the pretest and the posttest scores should be normally distributed.

For the AAU variable, the descriptive statistic was used to answer the research question. For the last three variables, PBE, SN, and PCC, Pearson correlation analysis and regression analysis were used to analyze the data.

Correlation analysis can be used to find if there is any relationship between the dependent variable and independent variable and to measure the strength and direction of this relationship (Landau & Everitt, 2004). Pearson correlation was used to find the relationships between the three variables (PBE, SN, and PCC) and the students' interest in a CS degree. The other inferential statistic that was used to analyze the data was the linear regression. It was used to find predictors for the student interest in CS among the three variables.

3.8 Validity of the Study

Validity is to what extent the study findings are accurate and can be generalized and applied to other people in other situation (Brians, Willnat, Manheim, & Rich, 2016). External validity addresses the ability to generalize the results with confidence (Leedy & Ormrod, 2013). Internal validity explains the outcome of the study, and it can be assessed with content validity and construct validity (Roberts, Priest, & Traynor, 2006).

Validity of the study included a valid research design, valid sample, valid statistical analysis, and valid measurement tool. The advantage of an experimental study is that it provided real evidence to support the acceptance or rejection of the hypotheses. While findings from some descriptive studies were based on a few minutes of someone taking an online survey, this experimental study involved direct interaction with the study subjects for several hours. This provided the researcher with observations in addition to the data that were analyzed. The observations and the data results guided the study findings and improved the study validity. The statistical analyses were chosen based on the research design and are discussed more in the "Data Analysis" section. The sample in an experimental study is usually smaller than that in a quantitative study. The chosen sample was composed of high schoolers because this study is targeting these students and their opinions about choosing a college major. However, the same study could be repeated with middle schoolers. The measurement instrument validity, external and internal validity are discussed in the following sections.

3.8.1 Validity of measurement instrument. Face validity, pilot test, and reliability tests were used to validate the survey questionnaire (Leedy & Ormrod, 2013). The face validity was established by experts. Three professors from Eastern Michigan University were asked to review the survey questionnaire. One of the professors is an expert in sociology, and he reviewed the part of the questionnaire that related to his expertise. The other professors are experts in computer science and statistics, and they reviewed the whole survey and approved it. The pilot test was performed for construct validity by asking one high school student to fill the survey. The pilot test helped measure the time required to fill the survey and to make sure that the language was clear and not confusing for students at the high school level. It also helped to check that the instrument was measuring what it was supposed to measure (i.e., the designed variables). The student was asked to give his feedback, and some of the questions were modified to improve the language clarity. The assessment survey should produce stable and consistent results. To measure this, a reliability test was used. After collecting the results, the reliability test was performed to make sure that each group of items measured the same construct. Some of the survey items were removed for negative correlations in the reliability test, and for negative wording. The reliability test is explained in details at the beginning of Chapter Four.

3.8.2 Internal validity. For both the pretest and the posttest, the data were collected in a class room setting from all students in each group at the same time. Students answered the survey questions carefully in a quiet room, and all their questions about the survey items were answered by the researcher. This enhanced the internal validity of the study. Although the coding time was not equal for the three groups, the same teaching material was used for all students in the three groups to help them solve the programming questions that were used to measure the students' programming knowledge. The other study variables were affected by the coding time differences, and this effect is discussed in Chapter Four and Chapter Five. This is not considered as a threat to internal validity. It could be considered as part of the treatment, where coding times put the treatment in different levels.

According to Campbell and Stanley (1963), there are several threats to the internal validity of the experimental study. These threats and how the researcher tried to avoid them are discussed as follows:

History. This is when something happens to affect the results of the experiment other than the treatment. The duration of this experimental study was only one week. Nothing happened during the workshop time that affected the students' results other than the treatment, so the change in results was caused by the treatment only.

Maturation. This is when the subjects grow older and their responses to the measured dependent variable are changed due to their growing, not due to the treatment. The experiment took place in a short period of time, which was a week long, and this was not enough time for significant differences in students' maturation.

Experimental mortality. This is when the researcher loses some subjects before the end of the study. Since the sample for this experiment is students who are usually available in summer for the summer camp, the possibility of losing subjects was low. However, one student left the coding camp. Similarly, for the fall workshops, which were in the students' schools and since the experiment’s period was only five days no students were lost in the fall workshops. Out of 67 students, 65 students were able to complete the pre- and the posttests. One student was lost in G1 and one student was not able to fill the survey questionnaire on time in G3 for technical reasons.

Testing. This is when the subjects' test-taking skills are enhanced due to taking the same test multiple times. In this experimental study, the survey questionnaire was a little long and the test was taken two times only (pre and post). The possibility that students remembered their previous answers and tried to enhance them was not high. Moreover, most of the survey questions are asking students about their feeling and preference, where nothing can be learned from the pretest. The possibility that the students may learn from the pretest was minimized.

Instrumentation. This is when the results changed due to different use of the measurement instrument. In this study, the dependent variables were measured in the same way in both pretest and posttest.

Statistical regression. This threat appears when subjects are selected on the basis of their extreme scores. There was a tendency for the subjects who got extreme scores in the pretest to regress towards the mean in the posttest. The amount of statistical regression was inversely related to the reliability of the test (Michael, 2002). The results of the survey reliability, which are discussed in Chapter Four, indicated that the measurement instrument used for the pre- and posttest was reliable.

***Differential selection and selection–maturation interaction of subject*.** This threat is due to the bias in selecting the members in two or more groups. The members of different groups should be equivalent at the beginning of the study. The one-group pretest-posttest design was followed in this study. Subjects were selected randomly but they were from the same state and same city, they were similar in age, and they were in the same educational level to minimize this threat.

The John Henry effect. This threat is due to the competitiveness between the selected subjects of different groups. John Henry was a worker who tried to perform better than a machine because he knew that his performance was compared with a machine (Ohlund & Yu, n.d.). In this study, different groups are not competing with each other.

3.8.3 External Validity. After finishing the summer coding camp (G1, N = 18), two more coding workshops were arranged in fall 2017 (G2 and G3, N = 47) to increase the collected data and to enhance the external validity. Ann Arbor schools were chosen for this study for their diversity. Students were from different races, genders, and from more than one school to eliminate the threat to external validity and to be able to generalize the results. Michael (2002) mentioned other threats to external validity for experimental study. These threats and how to eliminate them are discussed as follows:

Interaction effect of testing. This is when the pretest interacts with the experimental treatment so that the changes in the dependent variables are not caused by the treatment only. In this study, the researcher claimed that the treatment caused the change in the dependent variables. The dependent variables changed differently with different coding time for different groups as is explained in Chapter Four and Chapter Five. This indicated that the change was affected by the treatment itself and not by the pretest.

Treatment and subject interaction. This is another threat to external validity where treatment effect may be different when applied to a different sample. This effect could be minimized by carefully choosing the sample to represent the population. The sample in this study was chosen randomly from a diverse city so that the treatment would have a similar effect if the sample was different. Also, the treatment had an effect for both gender samples. However, the effect of the treatment could be different if the student samples were different in their academic achievement. The researcher claimed that if the sample was from high student achievers, they might be able to produce more code and share them; hence their interest and enjoyment may be different than a sample of low achievers.

***Testing and subject interaction*.** The previous threat states that different samples may react differently to the treatment. This threat states that different samples may react differently to the test or pretest-posttest survey questionnaire. Most of the test items in this study were measuring personal preference, and the researcher thinks that the possibility that different samples would react to the test differently was minimal. Moreover, the sample was selected to be from different racial groups to minimize this threat as well to the other threats.

***Multiple treatments*.** This is when the same sample receives two or more different treatments. In this case the results could be affected by both treatments. Since one treatment was used for all students in this study, there was no threat of the multiple treatments' effect on the external validity. If there was a significant difference in students' performance between pretest and posttest, it could be because of the effect of the specified treatment. However, two levels of the same treatment were used; one with less and one with more coding time and the results were discussed for both levels.

3.9 Resources & Budget

The researcher received support of $650 from the graduate school at Eastern Michigan University to cover the expenses of the study. The money was used to buy a small thank you gift for each student who completed the pre and post survey. Each student received two small gifts for the two surveys. Also, during the summer camp and the fall workshops, students were offered small snacks and cold beverages every day.

3.10 Treatment: The Code Genie Development Tool

The following sections discuss the development environment that was used as the treatment in this experimental study. The first section discusses the motivation behind developing this tool while the second section explains function of the tool. The following three sections discuss the choice of the JavaScript language, the Code Genie development process, and the importance of the responsive design.

3.10.1 Why Code Genie? The development tool was developed for this study by the researcher, who is also a software developer, with an intention to make coding with a real programming language more interesting and fun for high school students. As far as the researcher knows, there aren't many tools that were designed and developed by a female developer to get the interest of high school students, especially the female students. The tool is a free web-based Integrated Development Environment (IDE), and it is available online under the domain name theCodeGenie.com. In this development environment, the researcher tried to focus on the art, animation, and code sharing features. Figure 3-6 shows a screenshot of the developed environment.

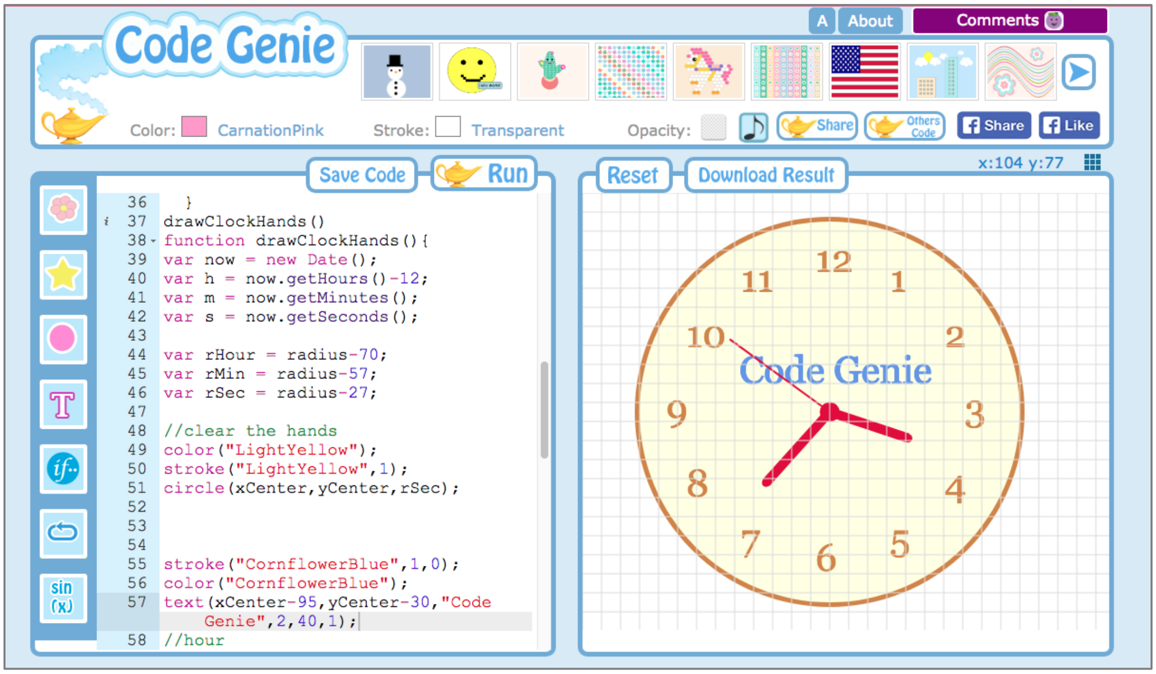


Figure 3-6. The Code Genie development environment (CodeGenie, 2018).

The idea came when the researcher attended one of the Hour of Code events for high school students in Ann Arbor, Michigan. The environment used in that event was offered by the Code.org and students used block-based language to write their programs. It was a fun environment for the students to get their interest in CS, but students were not using a real programming language. The researcher believed that this environment was suitable for elementary and middle school but not high school students who are in a stage of making decisions about their future careers. They should know what programming is really about because they are not going to write a program with blocks in the university course. The idea was to provide a fun and easy-to-use environment that encourages students to write a computer program with a real programming language used by developers.

3.10.2 What is Code Genie? Code Genie is a development environment targeting high school students to encourage them to write a computer program with a real programming language and to increase their interest in a CS degree after graduation. JavaScript was the programming language that was selected for this environment, and it can be used by students with different programming experience. A student can start programming using basic shapes and programming keywords and keep using the environment when he/she moves to intermediate or advanced levels.

Figure 3-7 shows an example of different programming levels. The advanced level in the figure included using several programming concepts such as the for-loop, JavaScript function, timer, math functions like cosine (cos), sine (sin), and the absolute value (abs). This code was written and shared by an Asian student in G1 in this experimental study, while the beginner level code was one of the code templates that can be found inside the environment.

The environment was developed to follow the human-computer interface rules that were discussed in the previous chapter. For example, to decrease the load on short-term memory, this environment categorizes the keywords so that a student doesn't need to study and memorize the keywords to develop his/her first program. In addition, browsing and the ease-of-composing features that exist in the block environment were added so that the student will know the language capabilities and what to do with the programming language. More HCI rules are discussed in the next section.

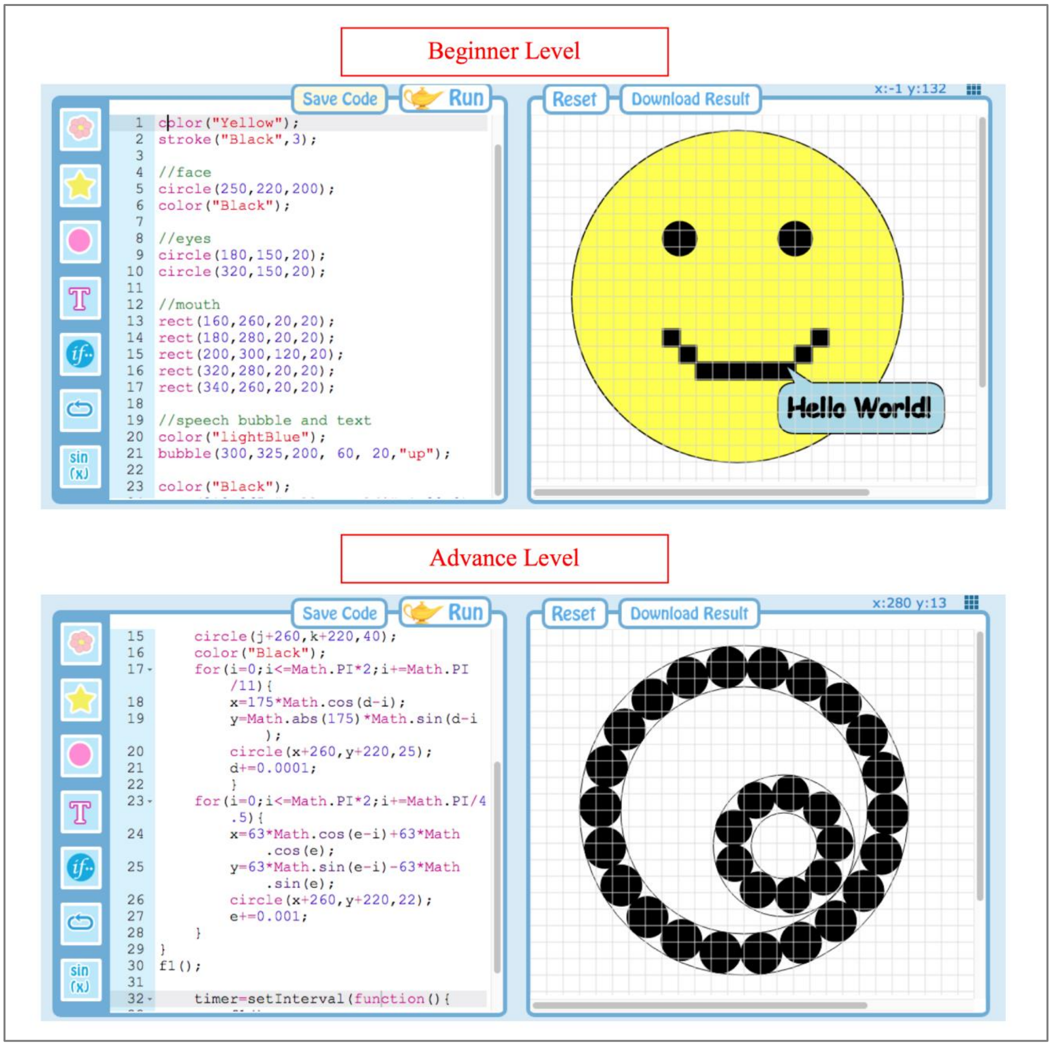


Figure 3-7. Programming levels at Code Genie IDE (CodeGenie, 2018).

The environment used art, animation, and code sharing features as motivation to write a program. The student started coding with simple shapes like squares, circles, and other basic shapes that can be found in the left-hand side toolbar as shown in Figure 3-8. The figure shows the main toolbar and the sub toolbars.

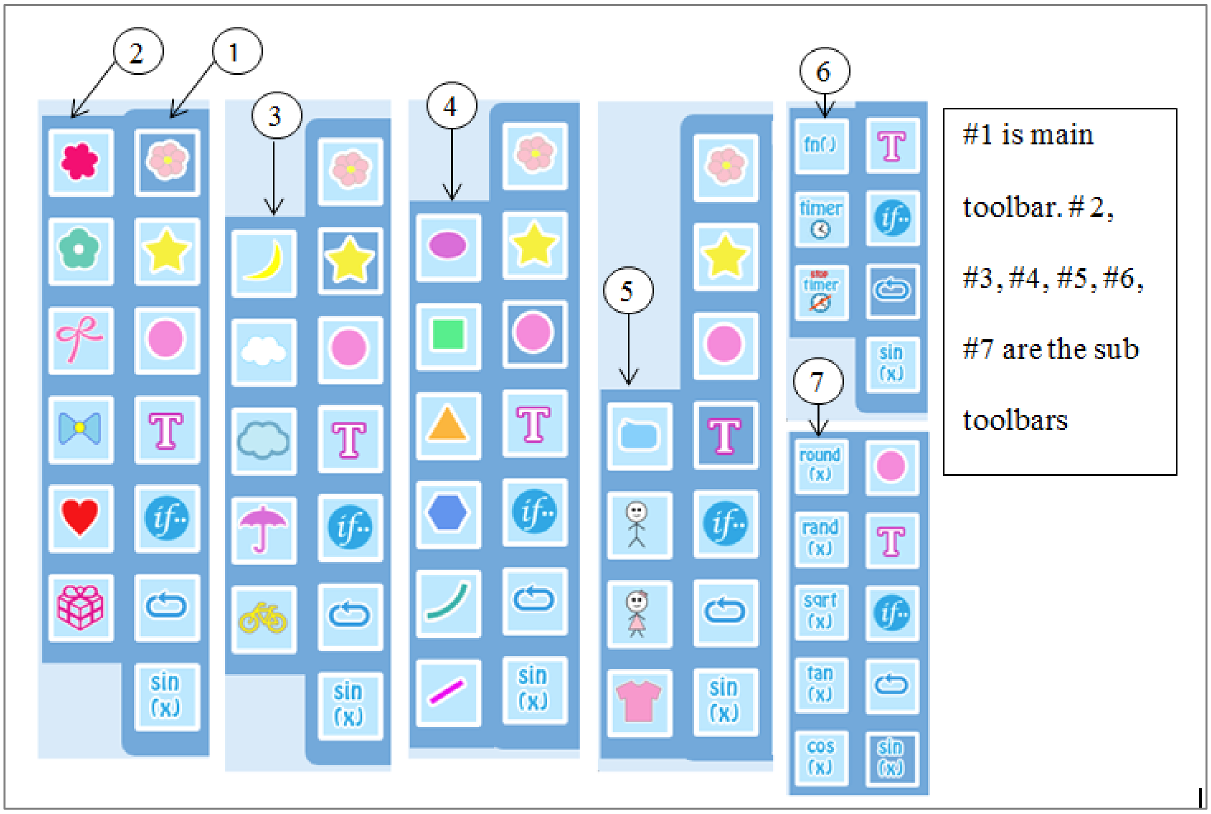


Figure 3-8. Code Genie toolbar (CodeGenie, 2018).

With this development environment, a student can learn more about math functions like square root, sine, cosine, and tangent. Figure 3-9 shows the template that includes the math functions. A student can also learn about the array keyword in JavaScript by inserting an array of colors as shown in the same figure. Array of colors could be added easily by selecting colors and clicking the "Add Array" button as shown in Figure 3-10. The stroke color and the opacity of the shapes can also be changed to provide more design features.

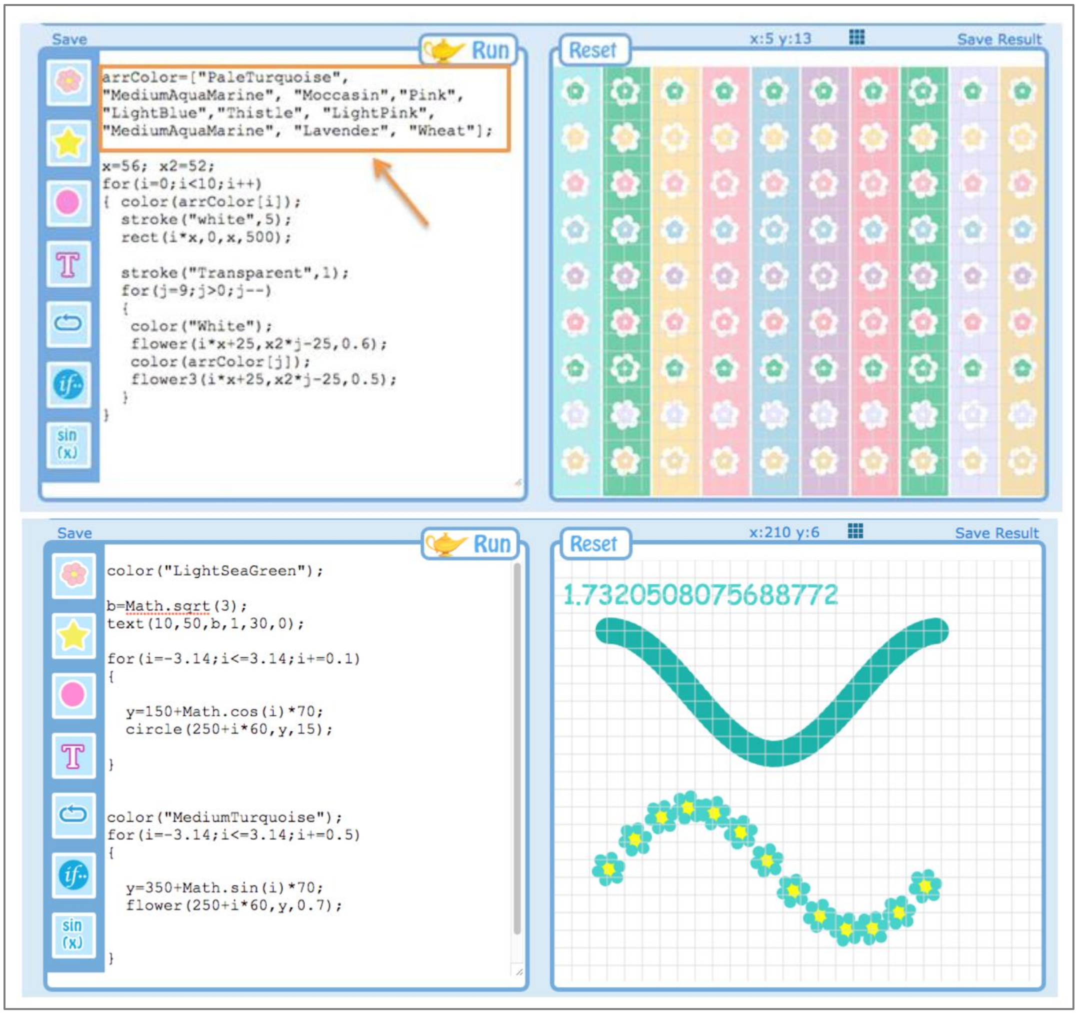


Figure 3-9. The use of Arrays and math functions in Code Genie(CodeGenie, 2018).

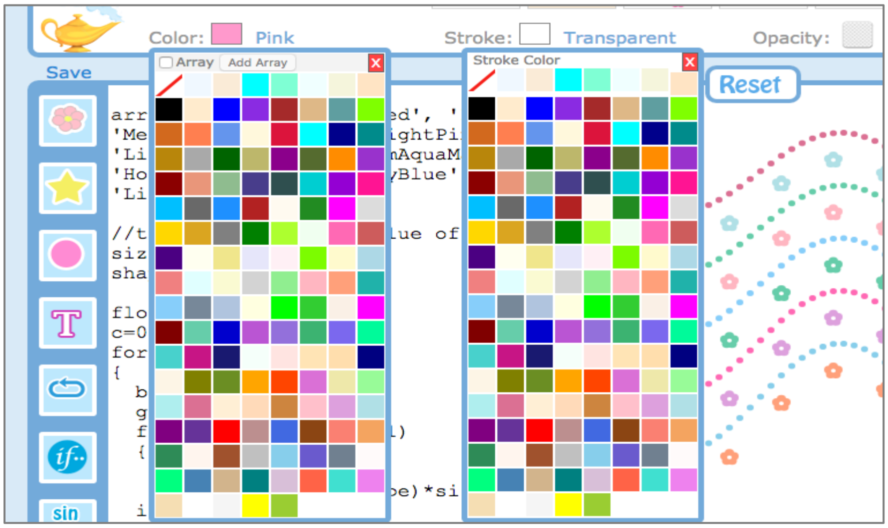


Figure 3-10. Color palettes and opacity setting buttons in Code Genie (CodeGenie, 2018).

Shapes and figures in Code Genie can be designed and implemented using the pixel drawing techniques. The right-hand side of the environment has a grid net that can be turned on and off through the "Grid" button. This button in addition to the X and Y coordinates can be used to help with pixel drawing in a way very similar to the embroidery net except that one would use lines of codes instead of a needle and thread. Figure 3-11 shows an example of pixel drawing.

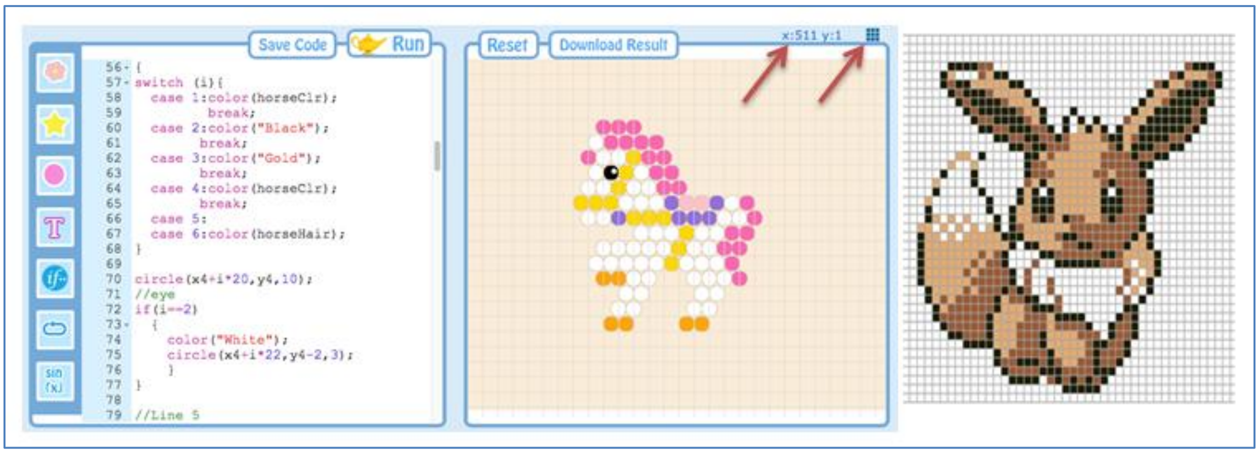


Figure 3-11. Pixel drawing in Code Genie (CodeGenie, 2018).

Code Genie includes several sample code examples or templates that will appear by clicking on the template's icon or "More" templates button. Figure 3-12 shows both art and animation templates.

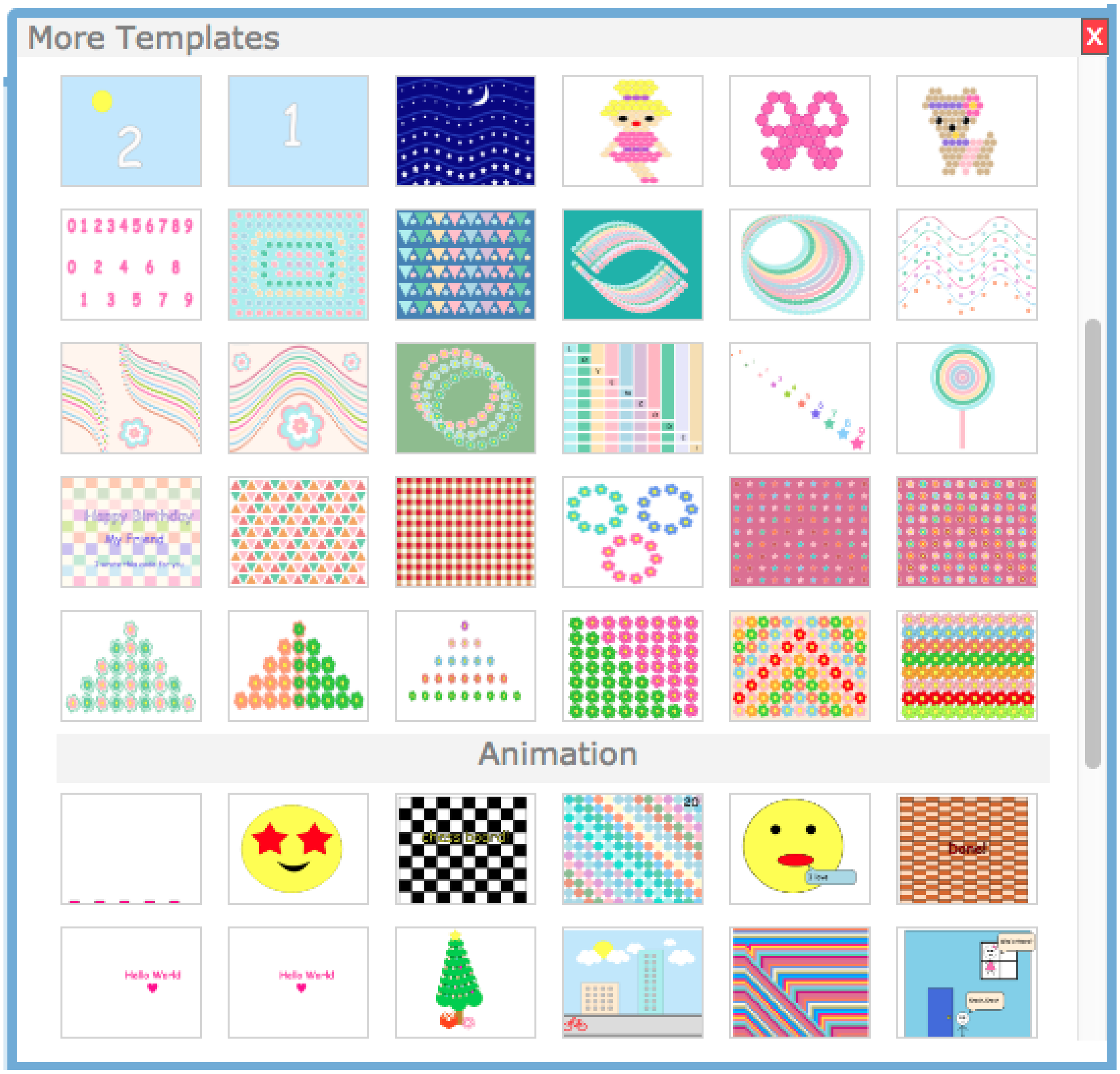


Figure 3-12. Code Genie templates (CodeGenie, 2018).

In addition to the templates, a user can share his/her code to serve as a template for other users. To do this, a user would click the "Share" button after running his/her code successfully. The shared code can be found by clicking the "Others Code" button. Figure 3-13 shows some of the students' shared code. Likewise, artwork produced by code can also be shared on social media, and this may attract other users to use the tool and learn coding.

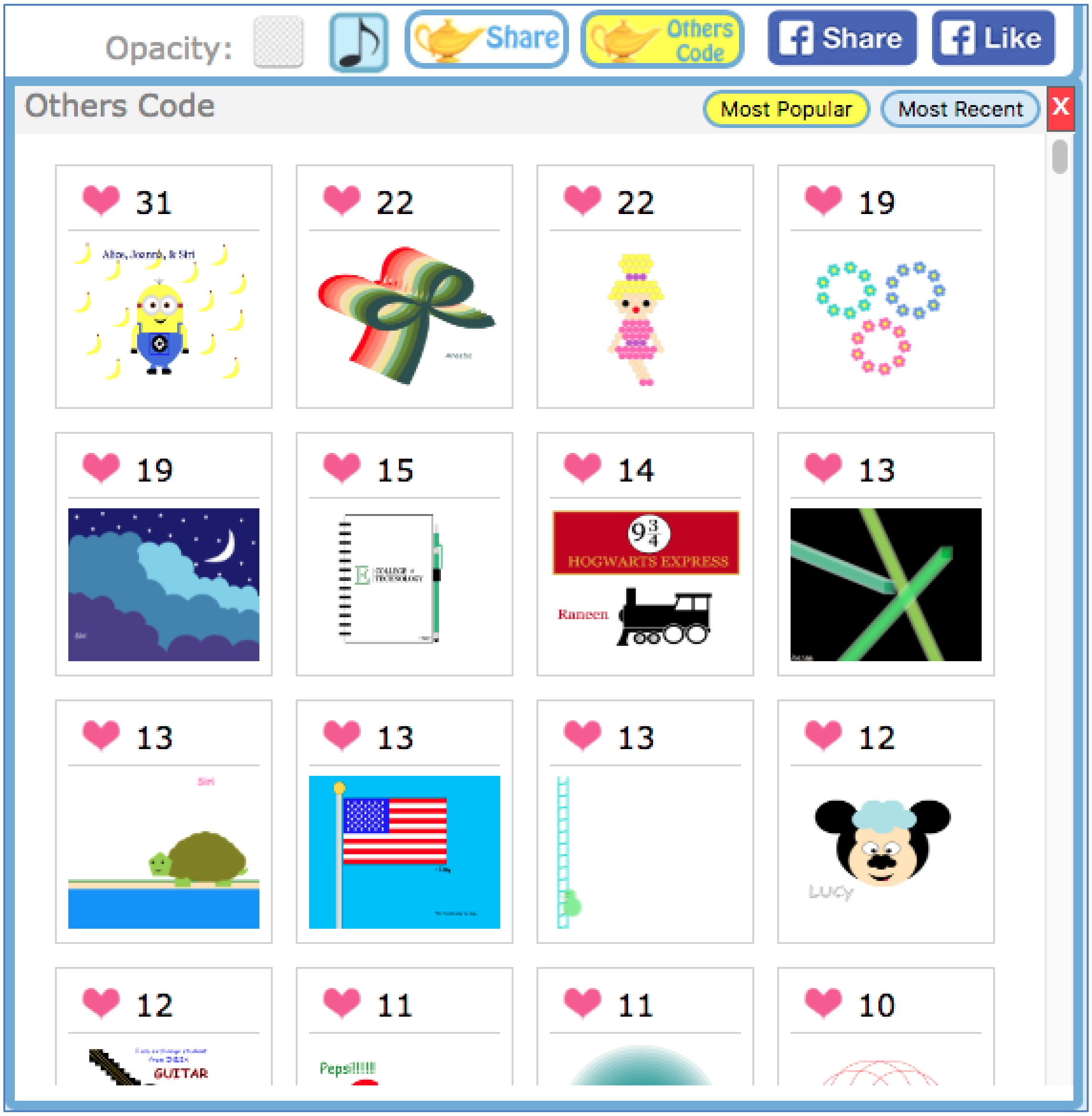


Figure 3-13. The sharing feature in Code Genie (CodeGenie, 2018).

3.10.3 Why JavaScript? As mentioned, the programming language used in this environment was JavaScript. As discussed in Chapter Two, JavaScript was the most popular programming language among developers in the StackOverflow annual survey. According to GitHub, JavaScript was also the most popular among the shared projects (Weinberger, 2017; GitHub Octoverse, 2017). GitHub has about 24 million users including employees from big tech companies like Apple, Google, and Facebook. Users from 200 countries are using

GitHub to share their projects, which are written in 337 different programming languages. Figure 3-14 shows the most popular languages at GitHub.

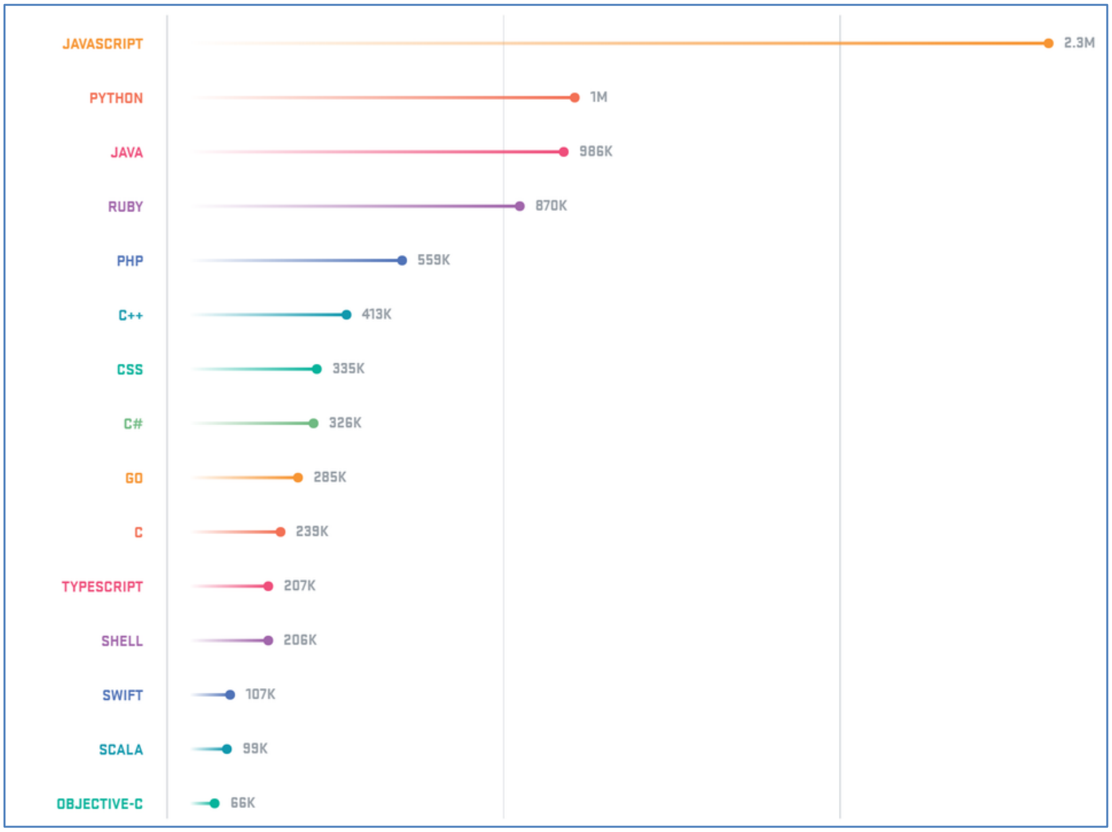


Figure 3-14. The most popular programming languages at GitHub (GitHub Octoverse, 2017).

JavaScript is a front-end developing language, and most of the software developers are front-end developers as discussed in the previous chapter. In addition, JavaScript language is considered one of the simplest languages where anyone with no prior experience in programming can program with it quickly. This language is also flexible and does not have as many restrictions as other languages. It does not require variable definition, and in many cases the developer can see some results even if there is some error with the program. While other languages, like Java or C, would not run if there was a small error. With JavaScript the developer did not need a special environment or a server. A text editor and Internet browser were enough to build a program with JavaScript, as compared to PHP (Hypertext

Preprocessor), which is a server-side scripting language that requires a PHP server to see the results (Bugs, 2009). Moreover, with the aid of mobile application development frameworks like PhoneGap or Ionic, JavaScript language is also used for mobile application development (Shaun, 2017). With those frameworks, a developer with knowledge of HTML, JavaScript and CSS can create an Android or iOS mobile application and upload it to the App Store or Google Play. For all these reasons, JavaScript was chosen for the Code Genie development environment.

3.10.4 Code Genie development process. Code Genie design and implementation followed the ADDIE model discussed in Chapter Two. Design was first implemented on a prototype using Photoshop then development started after evaluating the prototype with real users. Some of the Eight Golden Rules were followed in designing the user interface such as consistency, offering informative feedback and preventing error. For example, most of the shapes' code start with the x-axis and y-axis elements and most of them have the size element. To draw a circle this code " circle(x, y, radius); " is used, and this code " rect(x, y, width, height); " is used to draw a rectangle.

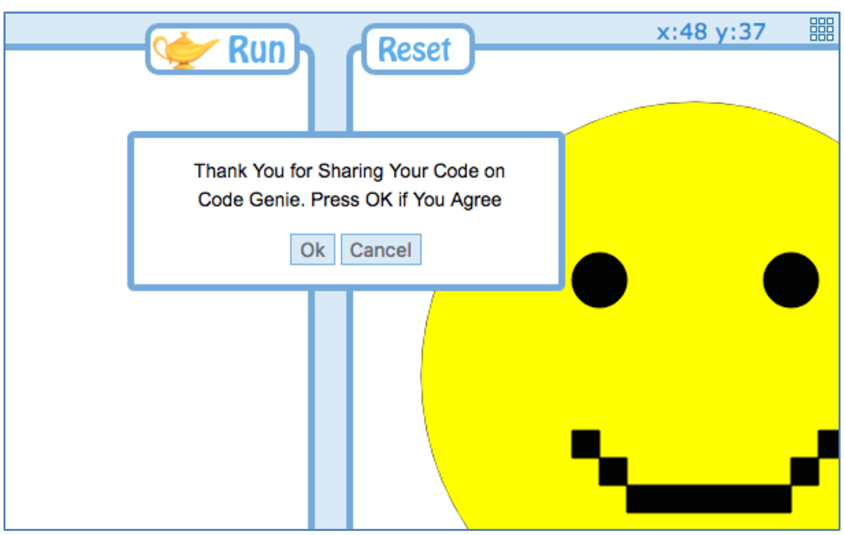


Figure 3-15. Offering informative feedback in Code Genie.

The other rule that was followed offered informative feedback. For example, when a user shared some art work, a confirmation box appeared asking the user to confirm his sharing, as shown in Figure 3-15.

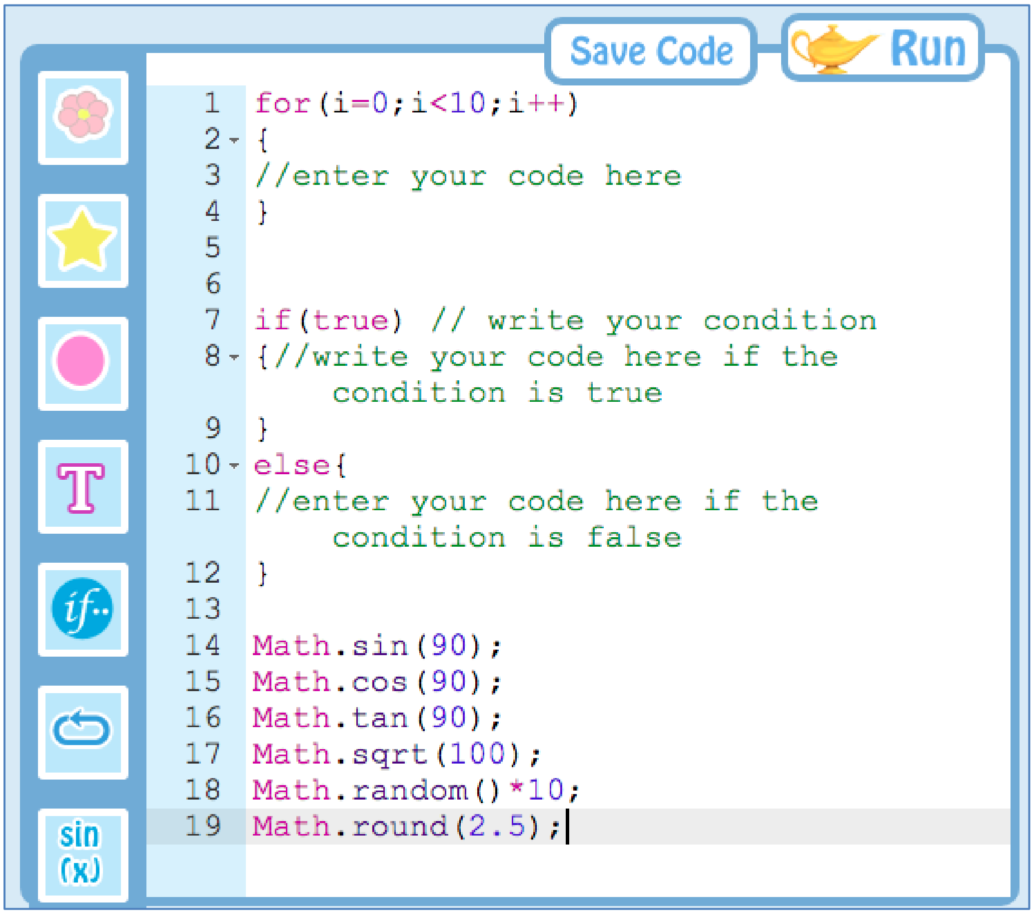


Figure 3-16. A Sample of pre-written code at Code Genie tool.

The use of pre-written code decreased the short-term memory load and enhanced the error prevention. "Prevent Errors" and "Reduce Short-Term Memory Load" were among the HCI golden rules that were followed in designing this tool. This was done by inserting a default working code when buttons were clicked. For example, if-statement, for-statement, or math functions, were entered when a user clicked on the corresponding buttons. Then the user could modify the inserted code. Providing some code to start reduced errors. Figure 3-16 shows a sample of default code. The figure also shows another feature that was added to increase the code readability. This was the "Syntax Highlighting" feature that was used to color the keywords, functions, constant, and comments with different colors. According to Sarkar (2015), adding this feature increased the program readability.

One of the important features that any programmer may need is the error messages when defining a variable is missing or when writing the wrong keyword. This could be classified under the user feedback rule of the HCI golden rules. Figure 3-17 shows an example of error messages.

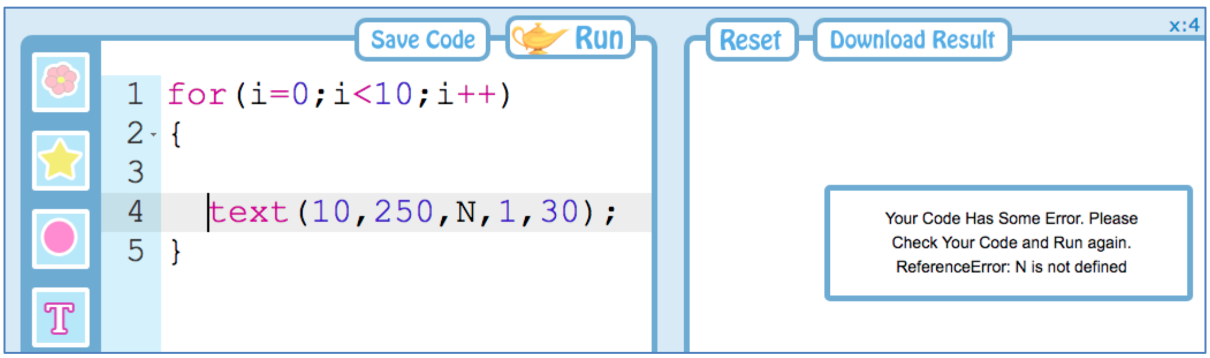


Figure 3-17. Error messages in Code Genie.

Design and development of the Code Genie learning environment required several months. The technologies used included several programming languages like HTML, CSS, JavaScript for front-end and PHP and MySQL for the back-end and the database.

3.10.5 Responsive design. To increase the students' access to the Code Genie tool, a responsive design was considered so that the tool could be used on smaller devices like smartphones. Any website or web application should be usable for different device sizes. The contents should stay the same for different devices while the design may be modified to fit the smaller screen.

Nowadays, most high school students have smartphones, and it is important to have the tool working on different devices. Although, the tool was designed for the desktop and it encourages students to program using the keyboard, it was also developed to work on desktop and smartphones such as iPhone 5, iPhone 6, and iPhone 6 Plus and any other smart phones with a 4.0 inch screen size or bigger. Designing for smaller screens involved some challenges. Figure 3-18 clarifies the idea of the responsive design and shows the difference in designing for desktop and smartphone.

Figure 3-18. Code Genie responsive deign.

The design should be changed to fit the smaller screen. Usually there isn't a physical keyboard during the interaction with the small device. Some buttons were added to handle the touch interaction. Some buttons were changed or merged with other buttons to fit the smaller screen and ensure the main purpose of the tool was not affected. For example, the "Clear" and "Copy" buttons were added, the sample example buttons on the top of the desktop design were removed and replaced by the "Example" button, and the "Back" button was added to go back to editor after running the code. The design should stay simple, responsive and user friendly, especially when the main target users are the school students who may get frustrated when they have to repeat the same task multiple times.

Figure 3-19 shows the tool's design for the smartphone. In this figure, the "Editor" area and the "Run" area were separated into two screens. First, a user sees the screen on the left then by pressing the "Run" button, the screen on the right would appear.

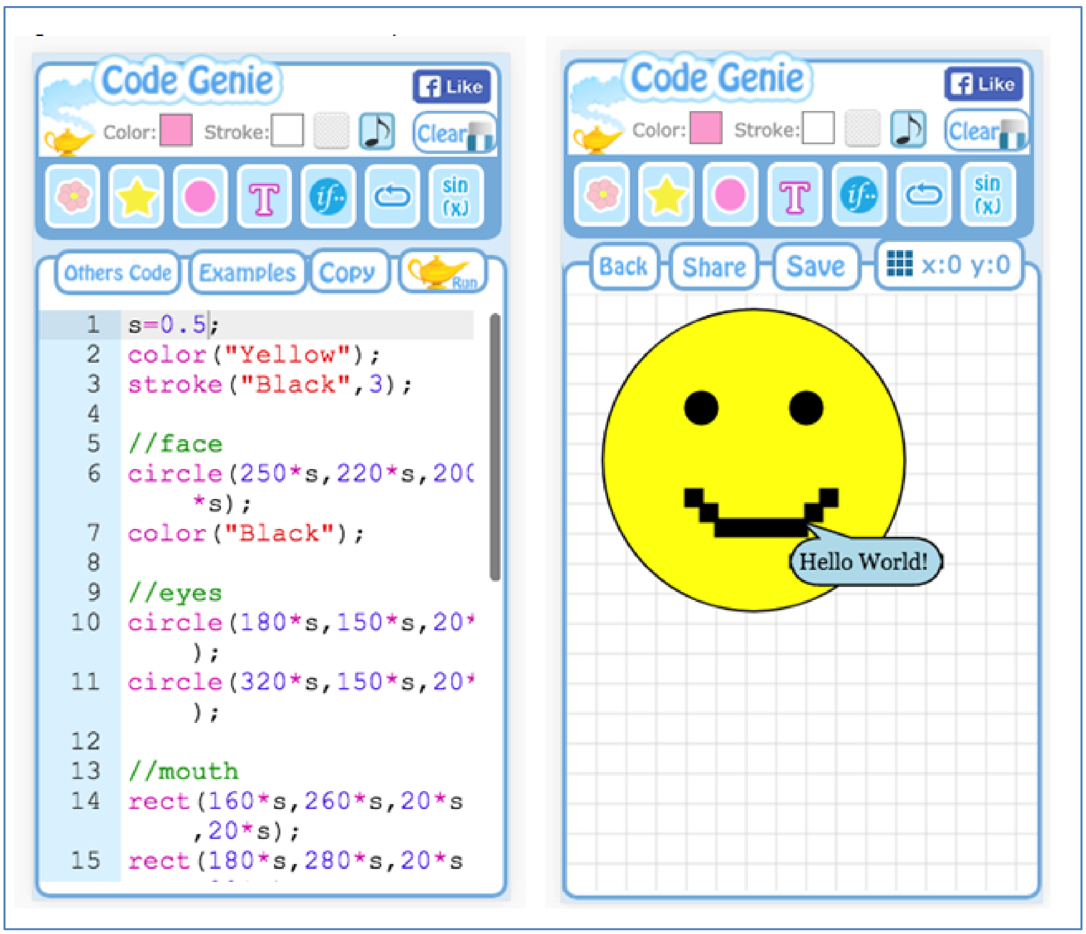


Figure 3-19. Code Genie for smartphones.

A user can save the artwork produced by coding and clicking the "Save" button in the output screen or the screen on the right in the above figure. The "Share" button was moved to the output screen or the "Run" area. The tool bar was moved to the top and the buttons were resized to fit the 4.7 inch device. Figure 3-20 shows the main toolbar and the sub-toolbars.

The tool is still responding to the users' error and specifies the errors as shown in Figure 3-21.

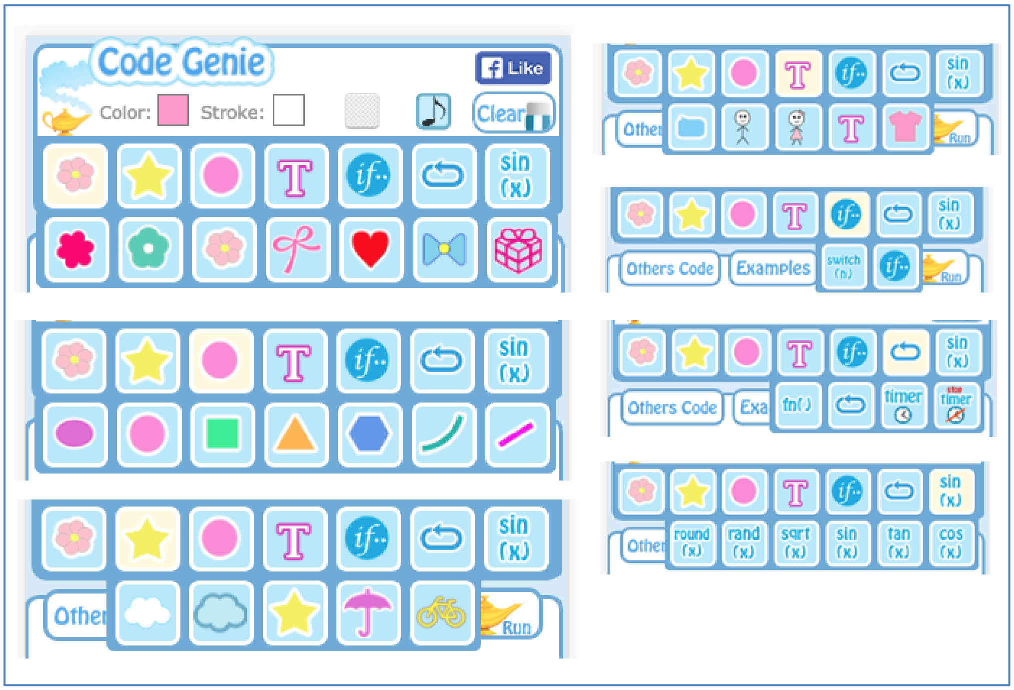


Figure 3-20. Code Genie tools menu bar for smartphones.

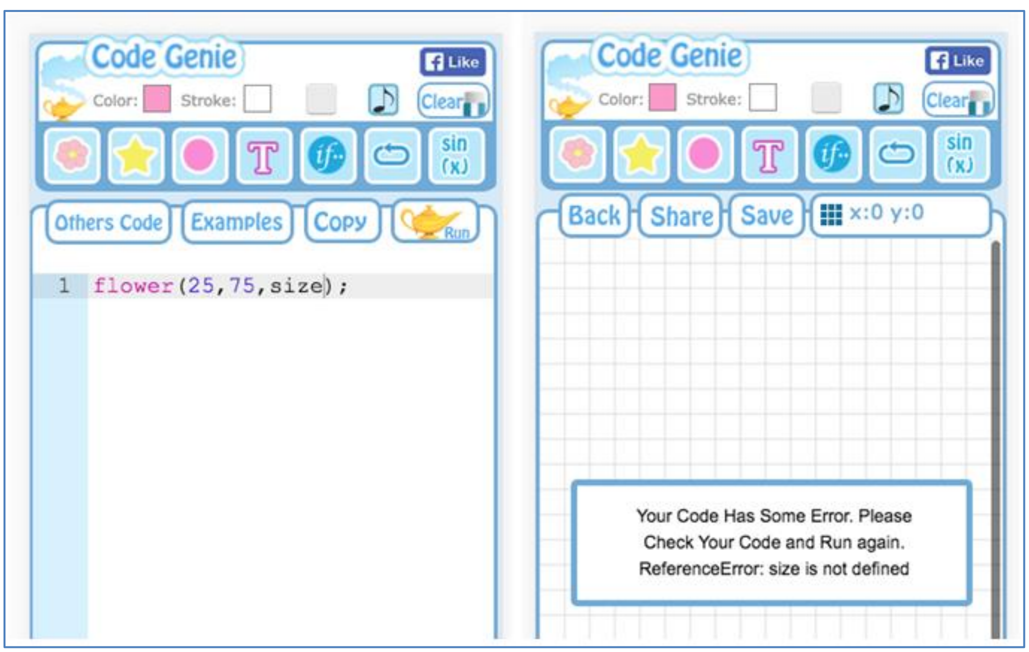


Figure 3-21. Code Genie error messages for smartphones.

3.11 Summary

This chapter explains the research design and the methodology that were used in this study. The chapter includes several sections for research questions and hypotheses, population and sample, the instrument used for data collection, the selected analysis methods, and the study validity. The chapter ends by demonstrating the online development environment that was built and used as the treatment in this experimental study.

The data was collected from high school students in Ann Arbor who participated in three coding workshops. Pretest-posttest design was used to collect the data at the beginning and at the end of the workshop. The analysis of the collected data answered the research questions and tested the hypotheses in chapters Four and Five. One of the main hypotheses is that the researcher wanted to explore whether or not the use of art and design in coding will increase students' interest in a CS degree.

Chapter Four: Result Analysis

In this chapter, the results of the survey questionnaire for the three groups are discussed in several sections. Before analyzing the data, two new groups were created, $G_{1,2,3}$ and $G_{1,2}$. The data of all students in the three groups were combined in $G_{1,2,3}$ to be analyzed together. The basic teaching material was the same for the three groups, and they were all similarly introduced to the treatment tool. However, G1 had 15 coding hours, and G2 had five coding hours in five different days, while G3 was exposed to three hours of coding in the same day. G1 and G2 had more time than G3 to explore the tool at home and try different programming examples. G1 and G2 were combined together in $G_{1,2,}$ and the results were analyzed as one ($G_{1,2}$). For all study variables, the sequence of the data analysis started with analyzing all groups together or $G_{1,2,3}$, then $G_{1,2}$ versus G3, and finally each group separately. Because the number of students in some racial groups was small, race results were calculated and discussed for $G_{1,2,3}$ only, while gender results were discussed for each group. This analysis was performed for the following six variables in this study: CS Degree Interest (DI), Programming Knowledge (PK), Real Programming Preference (RPP), Motivation for Code Sharing (MCS), Programming Interest and Enjoyment (PIE), and CS Course Interest (CI). The result of each of those was used to test the corresponding hypothesis. For the last variable, Art and Animation Usefulness (AAU), descriptive statistics of the posttest results were used to answer a research question.

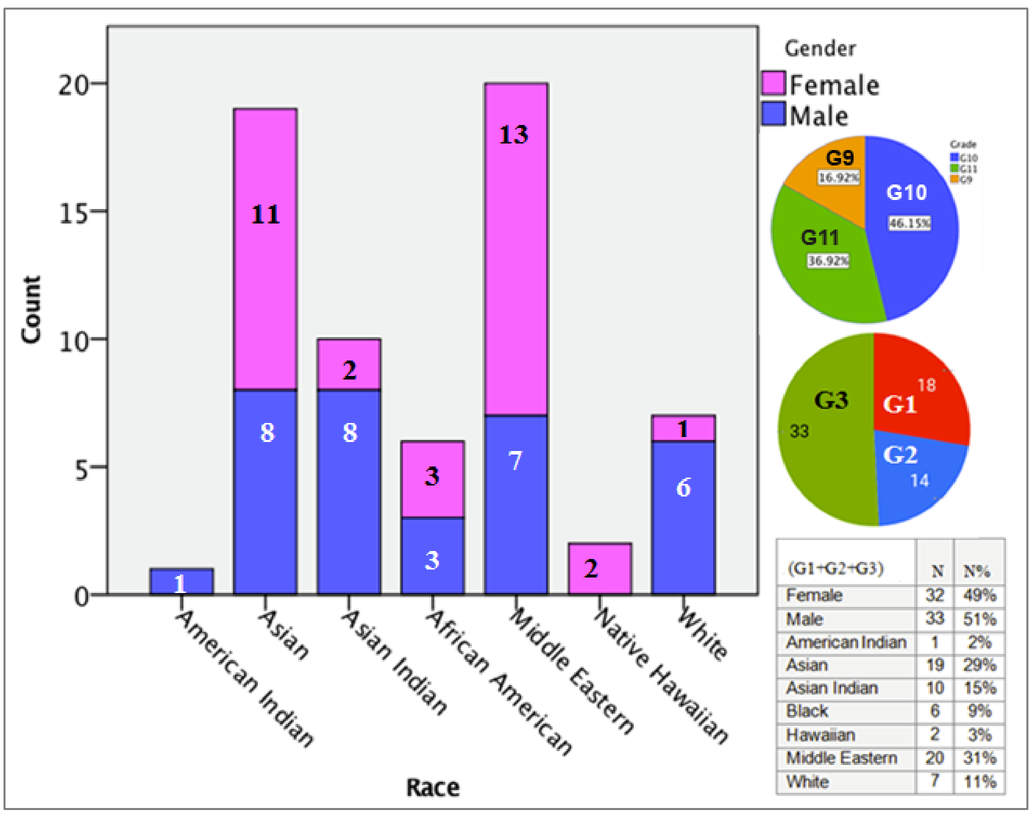
Correlation and linear regression were performed for DI and the following three variables: Programming Benefits and Enjoyment (PBE), Social Norm (SN), and Programming Capabilities and Confidence (PCC).

The results for programming knowledge questions are discussed in more details, and the students' scores are discussed for individual students in each group. Then, the relation between the coding time and programming knowledge is also discussed.

The last section in this chapter discusses the usability of the tool that was used as the treatment in this experimental study and the students' responses to this tool.

4.1 The Demographic Data

4.1.1 All groups ($G_{1,2,3}$). The participation of some racial groups was too low to get results for all racial groups. In addition, G2 had only four female students. To get better results, a new group was created to combine the data of all three groups together in one data file. This new group was called $G_{1,2,3}$, and all the data were analyzed as one group.



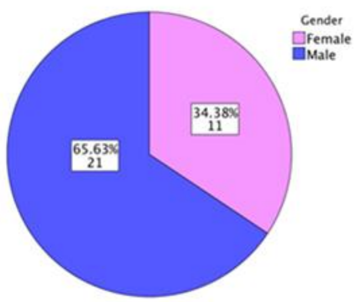
(G1+G2+G3)	N	N%
Female	32	49%
Male	33	51%
American Indian	1	2%
Asian	19	29%
Asian Indian	10	15%
Black	6	9%
Hawaiian	2	3%
Middle Eastern	20	31%
White	7	11%

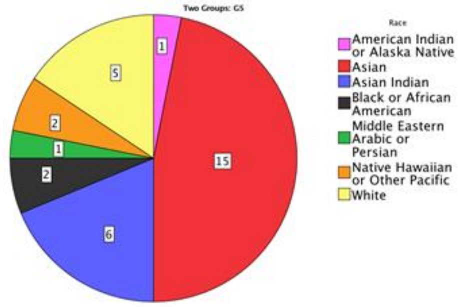
Figure 4-1. $G_{1,2,3}$ demographic data.

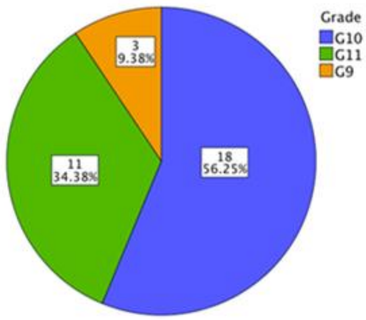
Figure 4-1 shows the demographic data of the resulting merged groups in one new group. The total number of students in all three groups was 65 students, 32 (49%) female and 33 (51%) male students. The largest racial group was Middle Eastern, 20 students or 31%, followed by the Asian group, 19 students or 29%. The other racial groups were Asian Indian, 10 students or 15%; White, 7 students or 11%; and Black, 6 students or 9%. The other two racial groups remained the same, one American Indian and two Hawaiian. Most of the students were sophomores (46%), 37% juniors, and 17% freshmen.

4.1.2 G1 and G2 together ($G_{1,2}$). The students of G1and G2 attended the summer camp and the fall workshop upon their own interest, while all junior and sophomore students of G3 were exposed to the coding workshop as a school activity. In addition, the three groups had different coding time. G1 had 15 hours of coding on five different days in the summer. G2 had five after school coding hours on five different days. G3 had three coding hours on the same day. The data of G1 and G2 was combined in $G_{1,2}$ to be analyzed together because students had more coding time and more days to explore the treatment tool. Figure 4-2 shows the demographic data of the resulting group $G_{1,2}$. The number of female students was 11 or 34%. The number of male students was 21 or 66%. The number of students in different racial groups was as follows: 15 Asian (47%), six Asian Indian (19%), five White (16%), two Hawaiian (6%), two Black (6%), one American Indian (3%), and one Middle Eastern (3%).

$G_{1,2}$ was compared with G3. $G_{1,2}$ was dominated by male and Asian students while G3 was dominated by female and Middle Eastern students, as shown in the G3 demographic section.







(G1+G2)	N	N%
Female	11	34%
Male	21	66%
American Indian	1	3.1%
Asian	15	47%
Asian Indian	6	19%
Black	2	6%
Hawaiian	2	6%
Middle Eastern	1	3%
White	5	17%

Figure 4-2. $G_{1,2}$ demographic data (G1 and G2 together).

4.1.3 Group One (G1): Pioneer summer camp group. The G1 pretest surveys showed that the number of all of the participants was 19 high school students: 7 female (36%) and 12 male students (63%). Figure 4-3 shows G1 demographic data. As shown in the gender pie chart, the number of female students was less than the number of male students. However, the difference is not big as compared to the statistic that states the male domination of the CS field. Participants were from different racial groups: 9 (47%) students were Asian, 5 (26%) Asian Indian, 2 (11%) White, 2 (11%) Black, and only one student was Middle Eastern (5%).

As shown in the race pie chart, most of the participants were Asian and Asian Indian. The researcher noticed that no white females were among the participants. Participants' school grade chart shows 9 sophomores (10^{th} graders), 9 juniors (11^{th} graders), and one freshman (9^{th} graders). Results show that senior students (12^{th} graders) were not interested in participating in the coding summer camp.

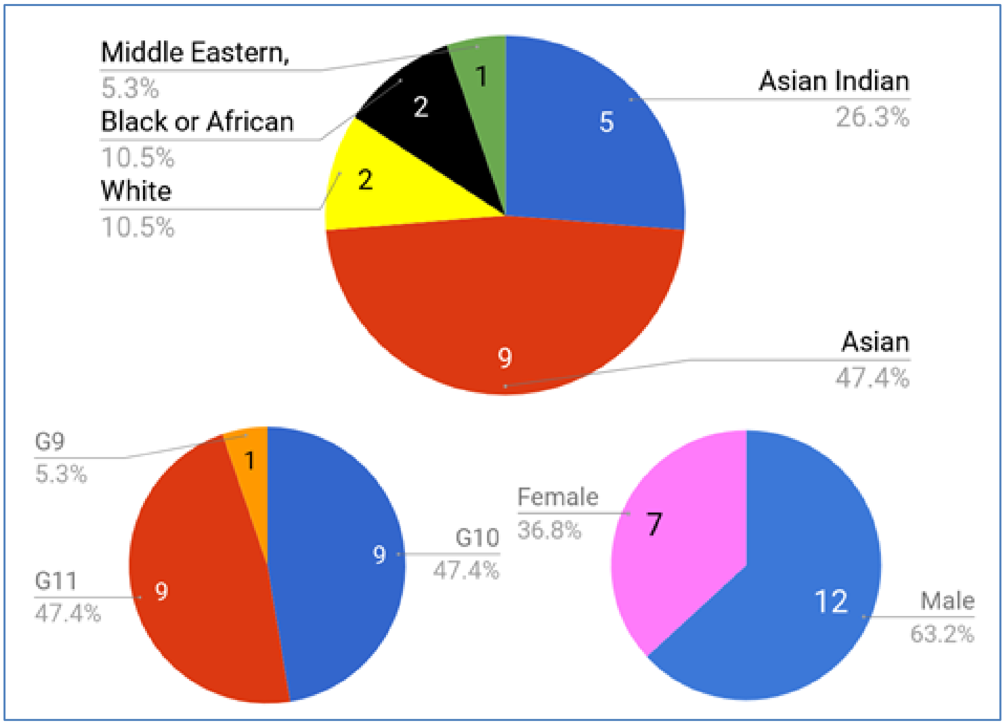


Figure 4-3. G1 demographic data.

Figure 4-4 shows that the female students were sophomores and juniors and not seniors or freshmen. It also shows that most of the female students were Asian with one being Asian Indian and most of the male participants were Asian Indian. Neither White females nor Black females were interested in a free coding summer camp. During school time, the researcher asked some Black female students to participate in the coding camp, but no one was interested in participation. Male participation had more race diversity than the female participation.

By the end of the coding camp, the researcher had lost one junior Black male student. The loss of that student agrees with Margolis' (2010) study that states that African Americans are less interested in CS and the number who earn undergraduate and advanced degrees in computer science is low. However, the lost student did a great job writing code that produced artwork, and he shared it with rest of the class. Also, he was able to run without errors all of the coding examples that were given as a teaching material. Moreover, attending the summer camp requires a transportation commitment, and the student may have left the camp for that commitment reason and not because of lack of interest. This is important because one of the goals in this study is to increase the programming interest of underrepresented minorities to encourage them to pursue a degree in computer science.

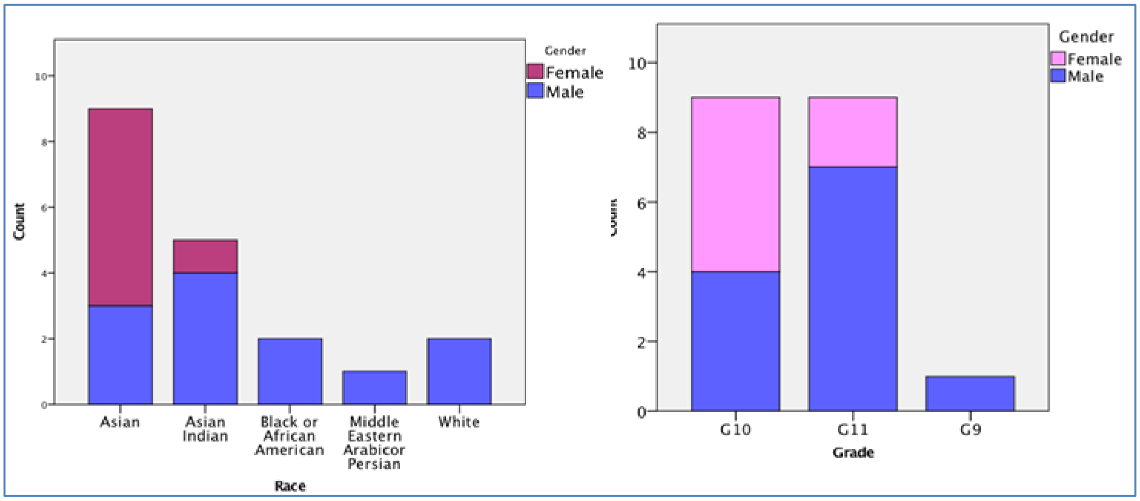


Figure 4-4. G1 participants' gender vs. race and grade.

4.1.4 Group Two (G2): Pioneer Fall Workshop Group. The number of participants in G2 in the pretest and posttest results was 14 (N = 14). Figure 4-5 shows the participants' demographic data. This time, the researcher did not lose any students, so the demographic data was the same in the pretest and the posttest survey. Similar to the summer camp, the number of female students was less than the number of male students, and the difference was relatively large this time. All participants were high school students, 4 female (29%) and 10 male students (71%). As shown in the race pie chart, students were from different racial groups: 6 (43%) were Asian, 1 (7%) Asian Indian, 3 (21%) White, 1 (7%) Black, 1 (7%) American Indian, and 2 (14%) Native Hawaiian. In G2, there was only one White female among the participants, as shown in Figure 4-6. Similar to the summer camp, most of the participants were Asian students. Participants' school grade chart states that 11 students were freshmen (9th graders), two sophomores (10th graders), and one junior (11th grader). All the female students were 9th graders, and this group had the most 9th grade students.

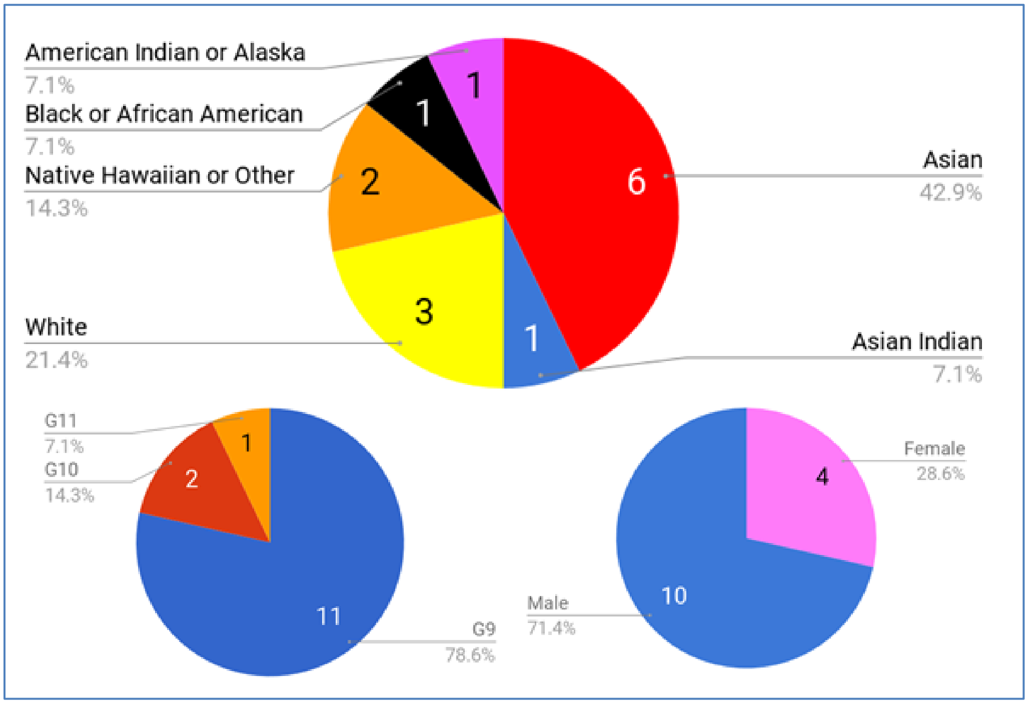


Figure 4-5. G2 demographic data.

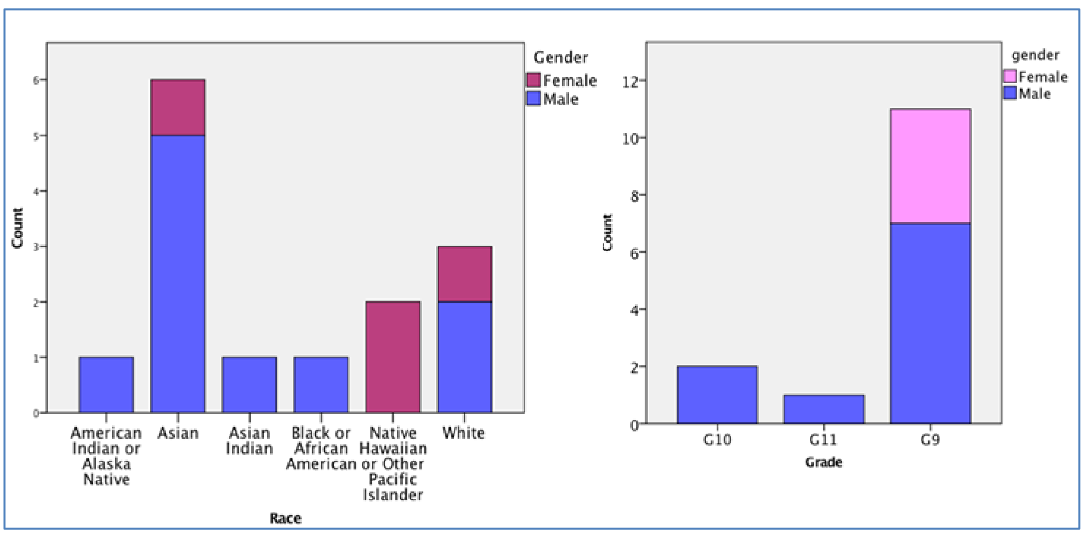


Figure 4-6. G2 participants' gender vs.race and grade.

In both the summer camp and in the fall workshop, there were no Black female students. Two Hawaiian, one White, and one Asian female student participated in the fall workshop.

4.1.5 Group three (G3): MIA fall workshop group. The number of the participants in the pretest and posttest results was 33; however, the actual number was 34. One student could not complete the pretest and posttest survey for technical reasons. All of the participants were high school students, 22 female (65%) and 12 male students (35%). Figure 4-7 shows the demographic data for this group. As shown in the gender pie chart, the number of female students was more than the number of male students in this group, taking into consideration that female students outnumber male students, in general, in this school.

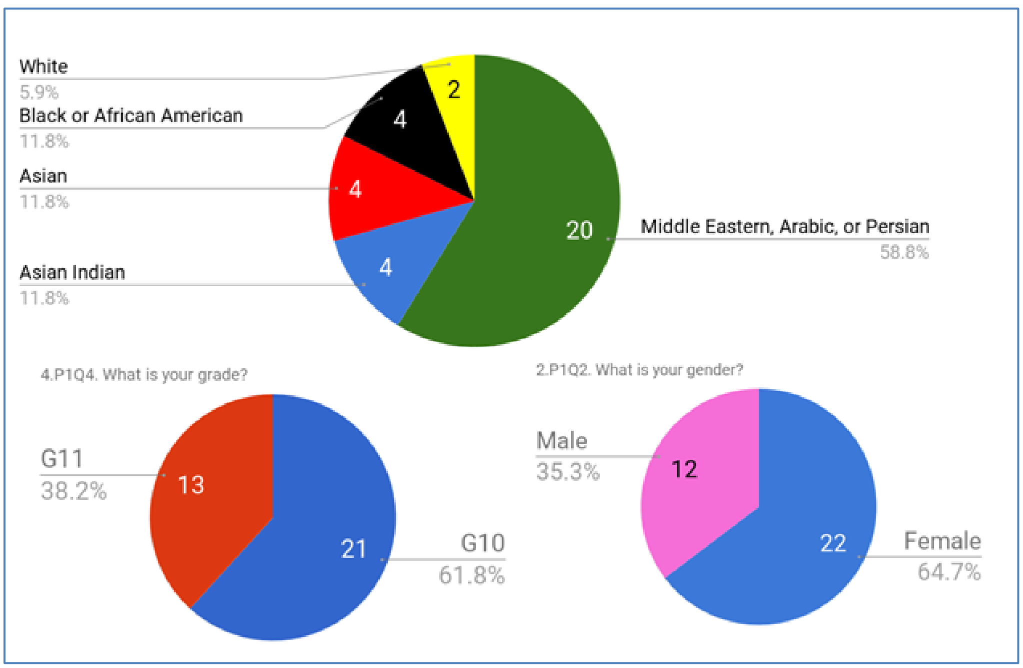


Figure 4-7. G3 demographic data.

Most of the participants were Middle Eastern students (59%), 12% Asian, 12% Asian Indian, 6% White, and 12% Black. Some Black females were in this group (G3), but there were no White females. The Middle Eastern female group had the largest number of students among other demographic groups, followed by Middle Eastern males, then Asian females. Sophomores accounted for 62% of the students, and 38% were juniors.

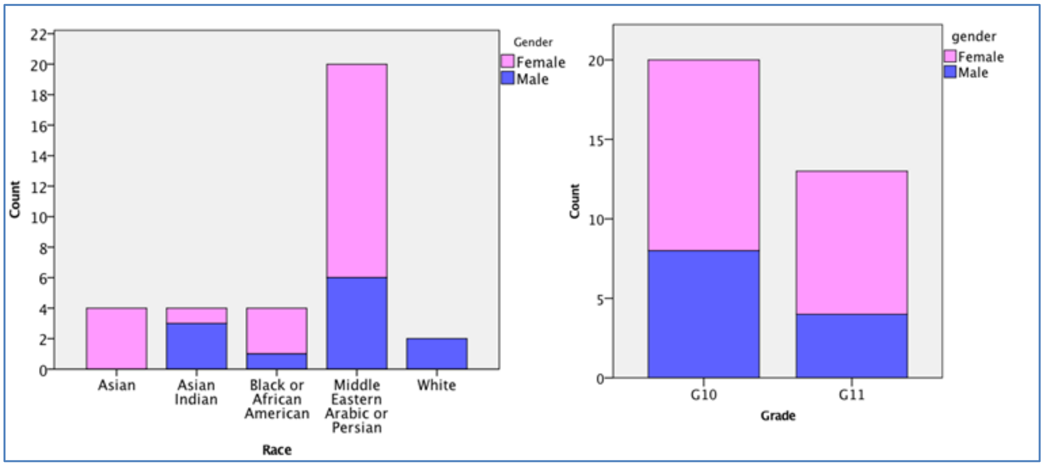


Figure 4-8. G3 participants' gender vs.race and grade.

4.2 Reliability Test and Validity

Before results were analyzed, a reliability test was performed to check the consistency of each group of survey items and to make sure that they measured the variable that they were designed to measure and were not negatively correlated with items in each group. The survey items for each variable are shown in Appendix B. The Cronbach's alpha coefficient of 0.7 or higher is considered acceptable (Nunnaly, 1978), and increasing the number of the survey items would increase the value of this coefficient. However, this value should not be higher than 0.95, which could indicate a redundancy in the scale items (Goforth, 2015). Table 4-1 shows the reliability test results for all the variables in this study.

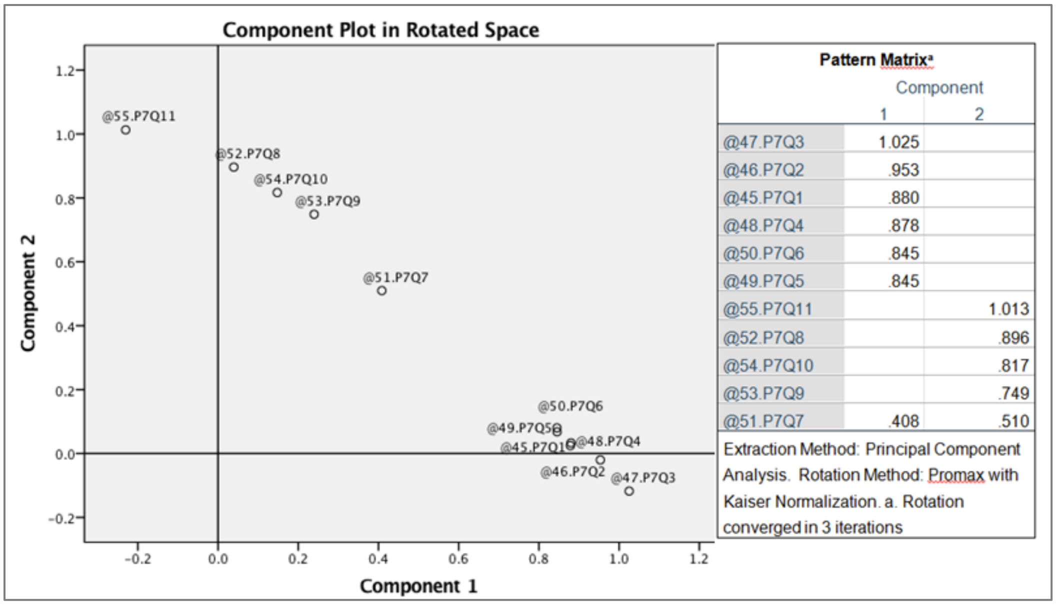
Confusing survey items and items negatively correlated with other items in the same construct were eliminated. The following two items are examples of the items that were eliminated. The first item was negatively correlated with other items, and the second item simultaneously measured programming enjoyment and interest in CS degree, yet with contradiction.

- I think men are better than women in the computer science field.
- I just enjoy programming now but I have other plans for my future job.

Table 4-1
Reliability Test Results of the Study Variables

Variable Name	Variable Meaning	Number of Survey Items	Cronbach's alpha
DI	CS Degree Interest	5	.849
RPP	Real Programming Preference	5	.818
MCS	Motivation for Code Sharing	11	.926
IMCS	Intrinsic Motivation for Code Sharing	6	.885
EMCS	Extrinsic Motivation for Code Sharing	5	.892
PIE	Programming Interest and Enjoyment	5	.846
AAU	Art and Animation Usefulness	6	.924
PBE	Programming Benefit and Enjoyment	7	.85
SN	Social Norm	4	.85
PCC	Programming Capabilities and Confidence	8	.884

To measure the MCS variable, 11 survey items were used. This variable included intrinsic and extrinsic motivation (Pintrich, 1991). Factor analysis was used for construct validity and to determine the questions for the intrinsic and extrinsic motivation construct (Roberts, Priest, & Traynor, 2006). The results of the analysis revealed two components and the questions that were used to measure these two components, as shown in Figure 4-9. To measure the Intrinsic Motivation for Code Sharing (IMCS) variable, the first six questions (Q1to Q6) were used. To measure the Extrinsic Motivation for Code Sharing (EMCS) variable, other questions (Q7 to Q11) were used.



	Component 1	Component 2
@47.P7Q3	1.025	
@46.P7Q2	.953	
@45.P7Q1	.880	
@48.P7Q4	.878	
@50.P7Q6	.845	
@49.P7Q5	.845	
@55.P7Q11		1.013
@52.P7Q8		.896
@54.P7Q10		.817
@53.P7Q9		.749
@51.P7Q7	.408	.510

Extraction Method: Principal Component Analysis. Rotation Method: Promax with Kaiser Normalization. a. Rotation converged in 3 iterations

Figure 4-9. Factor analysis result for the items of MCS variable.

4.3 CS Degree Interest (DI) Variable

The results of the CS DI, which was one of the main variables in this study, are discussed in this section. In the survey questionnaire pretest and posttest, students were asked about their interest in pursuing a degree in CS. To measure this variable, five items used a 5-point Likert scale. The difference in this variable between the pretest and posttest shows the effect of the treatment (or the use of art, animation, and code sharing) on the students' interest in pursuing a degree in CS after graduation from high school.

To analyze the results, the average of the students' responses to the five survey items was calculated for both pretest and posttest. The resulting two columns were called DIPretest and DIPosttest.

The descriptive statistics and the t-test results for this variable are discussed in the following three sections.

4.3.1 CS degree interest for $G_{1,2,3}$. The percentage of the $G_{1,2,3}$ students who responded *agree* or *strongly agree* on questions that measured the DI variables increased from 40% in the pretest to 52% in the posttest. The agreement percentage for the female students rose from 28% to 50%, while it increased slightly for the male students from 52% to 55% in the posttest. The results indicate that, before the experiment, males were more interested in the CS degree than females, but the percentage for female interest had better improvement.

The White group had the highest agreement percentage in both tests, and it rose from 71% to 86%. The percentages increased in the posttest for most of the racial groups except for the Black group, which decreased from 33% to 17% in the posttest. Table 4-2 shows the descriptive statistics for $G_{1,2,3}$ students.

Table 4-2
Descriptive Statistic of the Students' Interest in CS Degree.

	Agreement Percentage		Neutral Percentage		Disagreement Percentage	
$G_{1,2,3}$	Pretest	Posttest	Pretest	Posttest	Pretest	Posttest
All students	40 %	52%	35%	31%	25%	17%
Female	28 %	50 %	34%	28%	38%	22%
Male	52 %	55 %	36%	33%	12%	12%
Asian	58 %	68 %	26%	26%	16%	5%
Asian Indian	30 %	40 %	60%	60%	10%	0%
Black	33 %	17 %	17%	33%	50%	50%
Middle Eastern	25 %	35 %	35%	35%	40%	30%
White	71 %	86 %	14%	0%	14%	14%

The percentage of the students who chose the *disagree* and *strongly disagree* options dropped from 25% in the pretest to 17% in the posttest. The percentage of the neutral option rose from 35% to 31%. This suggests that the disagree students became "Agree" or neutral towards the degree in CS. Figure 4-10 shows the difference between the pretest and posttest results for the CS degree interest. The disagreement percentage of the female students decreased from 38% to 22%, but it remained unchanged for the male students at 12% in both tests. The disagreement percentages for the Asian group were 5% and 0% for the Asian

Indian. The disagreement percentage for Black students was the same in both tests 50%, but it dropped 10% for the Middle Eastern students in the posttest.

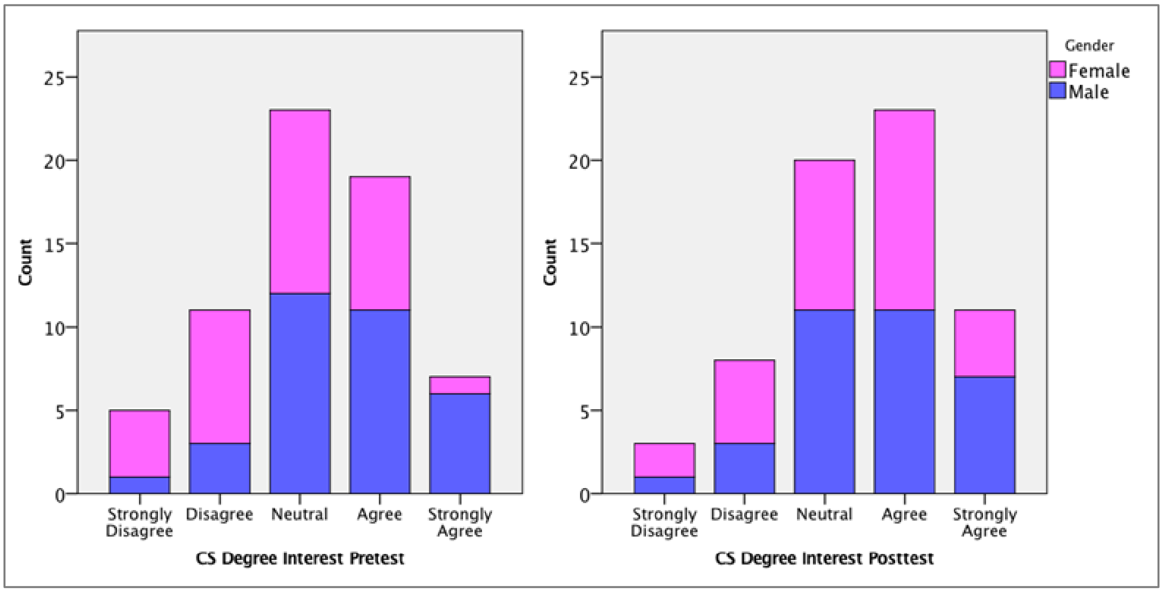


Figure 4-10. $G_{1,2,3}$ CS interest degree.

Table 4-3 shows the results of the paired samples t-test analysis for $G_{1,2,3}$, or all of the students. The analysis was first run for all students, and then the data file was grouped by gender and race. The analysis was run again, and the results were also calculated for each gender and race. The mean value for all students was above neutral in both tests ($M = 3.18$, $M = 3.48$), which suggests that most students had some interest in the CS degree. The t-test results, $t(64) = -2.681$, $p = .009$, showed a significant difference between the students' mean values in both tests. This means that the treatment was effective, and the students' interest in the CS degree increased in the posttest.

The mean values of the male students were higher than the mean values of the female students in both tests. This finding agrees with the previous suggestion that males are more interested in the CS degree than female students. However, the mean difference for the female students ($MD = -.531$) was higher than that for the male students ($MD = -.061$), which suggests better improvement in female students' interest.

Table 4-3
$G_{1,2,3}$ Mean Values of the CS Degree Interest

	Mean DI Pretest	Mean DI Posttest	N
$G_{1,2,3}$	3.18	3.48	65
Female	2.81	3.34	32
Male	3.55	3.61	33
American Indian	3.00	4.00	1
Asian	3.68	3.79	19
Asian Indian	3.20	3.60	10
Black or African American	2.83	2.83	6
Middle Eastern	2.65	2.90	20
Native Hawaiian	3.00[b]	4.00[b]	2
White	3.71	4.43	7

	MD (DI Pretest –DI Posttest)	t	df	Sig. (2-tailed)
$G_{1,2,3}$	-.292	-2.681	64	.009
Female	-.531	-3.418	31	.002
Male	-.061	-.421	32	.677
Asian	-.105	-.567	18	.578
Asian Indian	-.400	-1.309	9	.223
Black or African American	.000	.000	5	1.000
Middle Eastern	-.250	-1.097	19	.287
White	-.714	-2.500	6	.047

The t-test results for the female students was significant, $t(31) = -3.41$, $p = .002$. The p-value was less than .05, which indicates a significant difference in the mean values between the pretest and the posttest for the female students. The t- test result for the male students was not significant $t(32) = -.42$, $p = .677$. The results show that male students had prior interest in the CS degree and the treatment did not increase their interest significantly. However, the minus sign in the mean difference ($MD = -.061$) shows that male students' interest was slightly increased.

The mean values of the White group were the highest in both pretest ($M = 3.71$) and posttest ($M = 4.43$) in comparison with other racial groups. The t-test result was significant for the White group $t(6) = -2.500$, $p = .047$. In contrast, although the Asian group's mean values were also high in both tests ($M = 3.68$, $M = 3.79$), the t-test result was not significant for this racial group, which suggests that this group already had interest in the CS degree and the treatment slightly increased their interest ($MD = -.105$). The results for the other racial groups were not statistically significant.

4.3.2 CS degree interest for G3 vs. $G_{1,2}$. The descriptive statistics of G3 show that the percentage of the students' disagreement choices dropped from 42% in the pretest to 27% in the posttest. It also dropped from 52% to 33% for female students, and it dropped from 25% to 17% for male students. The agreement percentage of all students in G3 improved slightly from 30% to 33%, and the neutral percentage increased from 27% to 39%. This indicates an increment in the students' interest in CS degree for both genders in G3.

On the other hand, the interest in CS degree among the $G_{1,2}$ students was higher than the interest of G3 students. Statistics for $G_{1,2}$ show that the percentage of disagreement was lower than that of G3 in both tests. In the pretest, the percentage was 6% in $G_{1,2}$ while it was 42% in G3. This disagreement percentage for $G_{1,2}$ remained unchanged in the posttest (6%), but the agreement percentage for $G_{1,2}$ increased from 50% in the pretest to 72% in the posttest. No female student chose disagreement choices in the posttest, but the percentage slightly increased for male students from 5% to 10%. More students of both genders responded *agree* or *strongly agree* on their CS interest questions in the posttest. The agreement percentage for the females in $G_{1,2}$ increased from 46% to 73%, and it increased from 52% to 71% for the male students. The neutral percentage dropped from 44% to 22 % for $G_{1,2}$. Figure 4-11 shows the difference between the students' interest in CS degree in G3 and $G_{1,2}$.

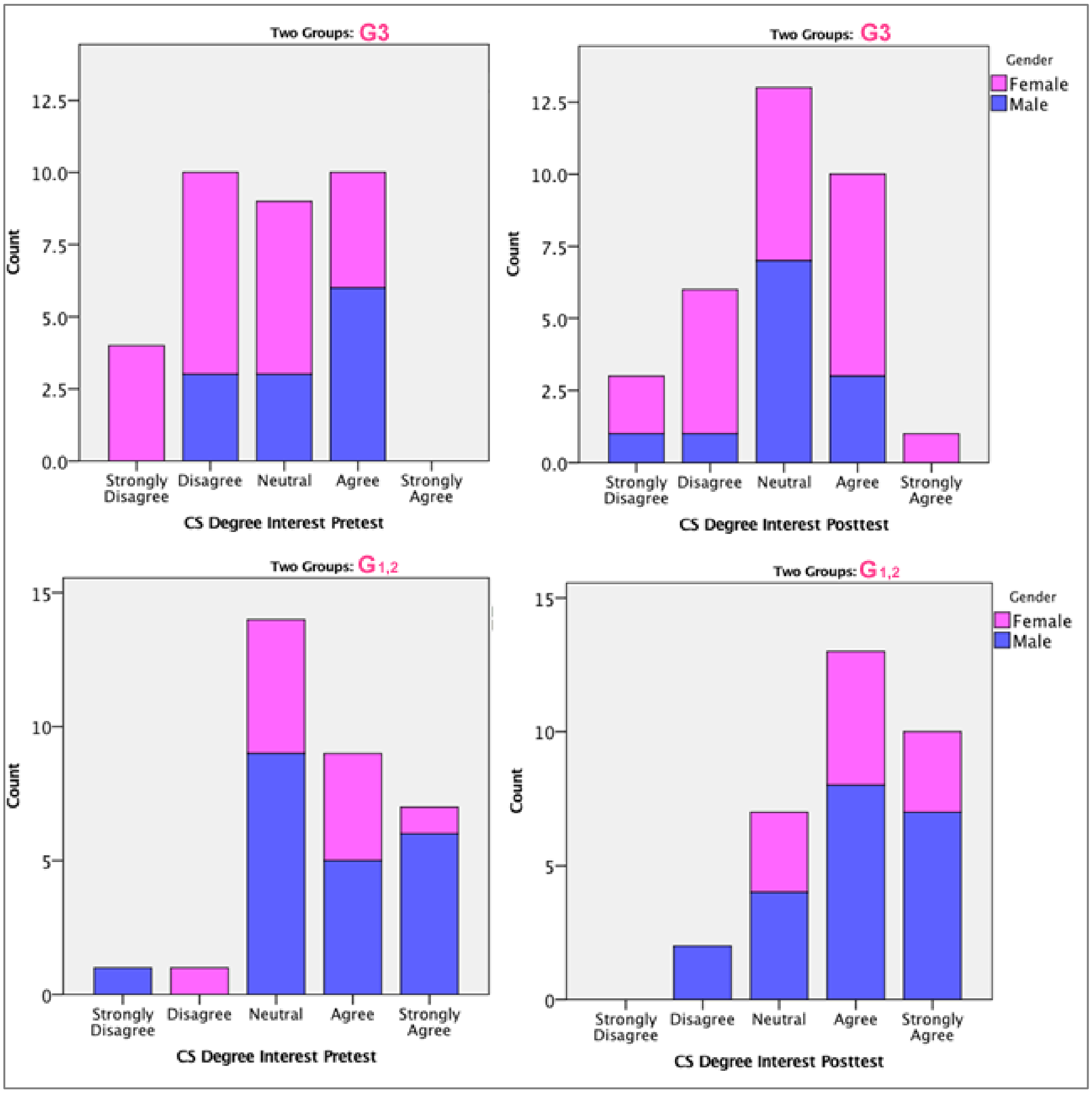


Figure 4-11. G3 vs. $G_{1,2}$ CS degree interest in pretest and posttest.

Table 4-4 shows the t-test results for G3 and $G_{1,2}$. The result was significant for G1,2 $t(31) = -2.470$, $p = .019$, but it was not significant for G3 $t(32) = -1.437$, $p = .160$.This means the students' interest in the CS degree increased significantly in G1,2, but it did not increase significantly for all students in G3. However, the female students in G3, who were 65% of the students, had a significant result $t(20) = -2.447$, $p = .024$ although the male students' result was not significant in G3. Similarly, the result for the female students in $G_{1,2}$ was also significant $t(10) = -2.63$, $p = .025$. The male students, who were the majority in

$G_{1,2}$, did not have significant results $t(20) = -1.313$, $p = .204$. In the posttest, the mean value of the female students in $G_{1,2}$ was highest among other groups.

Table 4-4
G3 and $G_{1,2}$ T-Test Results for the CS Degree Interest

	Mean DI Pretest	Mean DI Posttest		N
G3	2.76	3.00		33
Female	2.48	3.00		21
Male	3.25	3.00		12
$G_{1,2}$	3.63	3.97		32
Female	3.45	4.00		11
Male	3.71	3.95		21
	MD (DI Pretest –DI Posttest)	t	df	Sig. (2-tailed)
G3	-.242	-1.437	32	.160
Female	-.524	-2.447	20	.024
Male	.250	1.149	11	.275
$G_{1,2}$	-.344	-2.470	31	.019
Female	-.545	-2.631	10	.025
Male	-.238	-1.313	20	.204

4.3.3 CS degree interest for G1, G2, and G3. As mentioned, G1 and G2 attended the five-day coding workshop upon their own interest, while G3 were exposed to a one-day coding workshop as a school activity. This suggests that G1 and G2 initially had more interest in CS than G3, and the mean values of the three groups corroborate this assumption. The mean values of CS degree interest in the posttest for G1 ($M = 3.67$) and G2 ($M = 4.36$) were higher than the mean values of G3 ($M = 3.00$), as shown in Table 4-5.

The t-test result was significant for the female students in G2 and G3, and it was significant for all students in G1. In G3, the results for the female students were $t(20) = -2.44$, p = .024, and the mean values in both tests indicate a change from disagreement ($M = 2.48$) to neutral ($M = 3.0$). The result was not significant for the male students in G3 ($p = .275$), and the mean difference was positive, which indicates a slight decrease in their interest in the posttest. The result for the female students in G2 was close to being statistically significant $t(3) = -3.00$, $p = .058$. The result was also significant for G1, $t(17) = -2.40$, $p = .028$.

Table 4-5
G1, G2, and G3 T-Test Results for the CS Degree Interest

	Mean DI Pretest	Mean DI Posttest		N
G1	3.22	3.67		18
Female	3.43	3.86		7
Male	3.09	3.55		11
G2	4.14	4.36		14
Female	3.50	4.25		4
Male	4.40	4.40		10
G3	2.76	3.00		33
Female	2.48	3.00		21
Male	3.25	3.00		12
	MD (DI Pretest –DI Posttest)	t	df	Sig. (2-tailed)
G1	-.444	-2.406	17	.028
Female	-.429	-1.441	6	.200
Male	-.455	-1.838	10	.096
G2	-.214	-1.000	13	.336
Female	-.750	-3.000	3	.058
Male	.000	.000	9	1.000
G3	-.242	-1.437	32	.160
Female	-.524	-2.447	20	.024
Male	.250	1.149	11	.275

The agreement percentages for the three groups rose in the posttest. For G1, the agreement percentage increased from 28% to 50%. For G2, the agreement percentage was high in the pretest (79%) and increased in the posttest to 100%. For G3, the percentage increased slightly from 30% to 33%, but the neutral percentage increased and the disagreement percentage decreased in the posttest. Figure 4-12 shows the difference in the CS degree interest among the three groups.

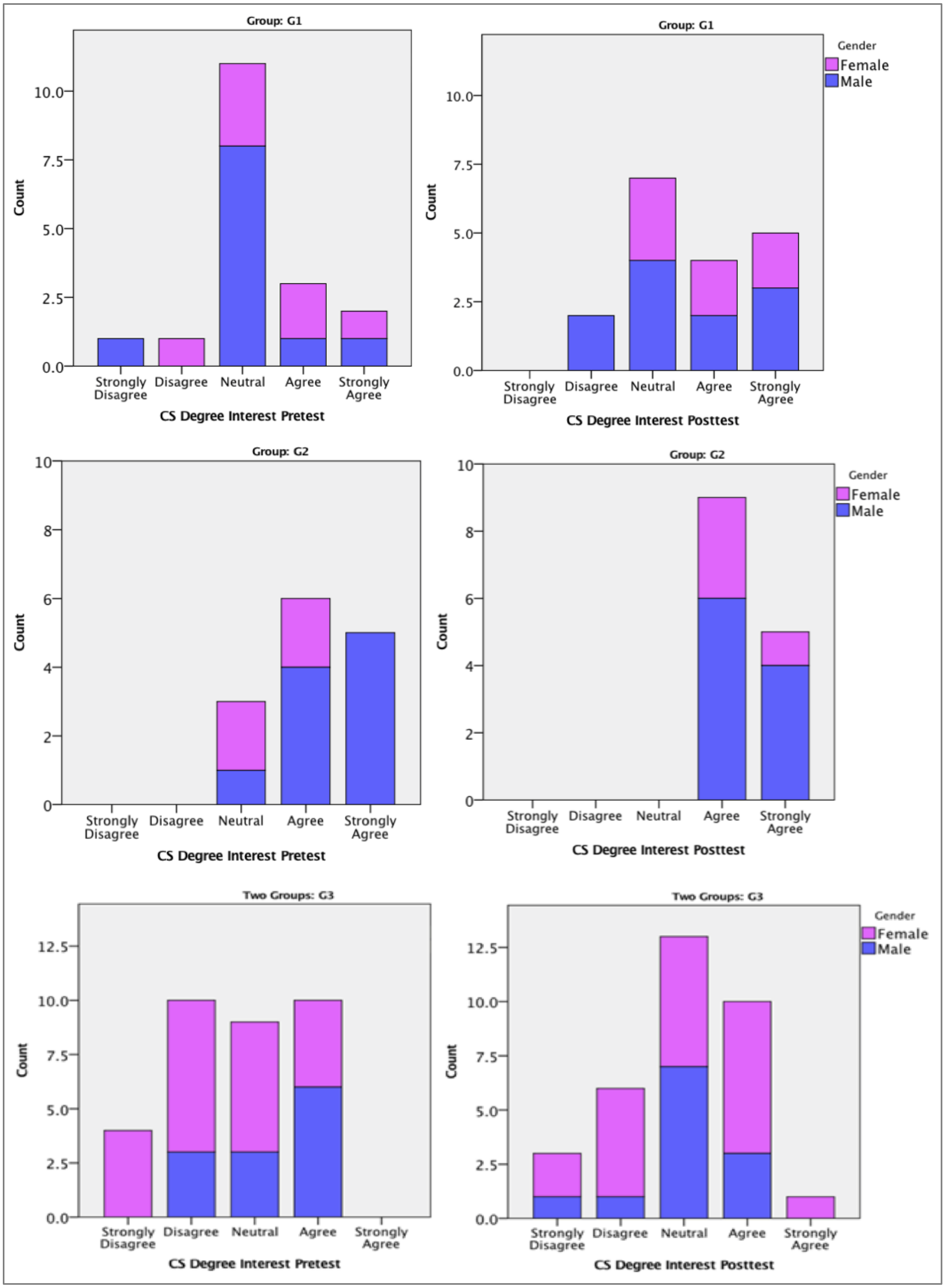


Figure 4-12. CS degree interest in pretest and posttest for G1, G2, and G3.

4.4 Programming Knowledge (PK)

The programming questions in the survey questionnaire were used to measure the students' Programming Knowledge (PK) variable. This variable included their understanding of the different programming concepts: variable assignment, variable addition and multiplication, for-loop, if-statement, switch-statement, arrays, function, and the math functions. The Students' knowledge was measured in terms of their understanding of the programming concepts used as sub-variables in this study. The results for the programming variables (PV1 to PV12) are discussed in detail at the end of this chapter. Students in G1 and G2 had more time to try more examples, but the same teaching material, which explains the measured programming concepts, was used and explained equally in all three groups. The results show improvement in all programming questions in the posttest. Figure 4-13 summarizes the results of the programming variables for all students in three groups in both pretest and posttest. As shown, the green color, which refers to the correct answers, increased for questions, while the orange color, which refers to the "don't know" option, decreased in the posttest.

To discuss the overall programming knowledge among the three groups, a students' identification column and a students' programming total scores column were created for each student in the data file. The identification column starts with a code followed by an alphabetical letter for each student. SSt, FSt and MSt are the codes that are used in the identification columns for Pioneer summer, Pioneer fall, and MIA groups respectively. The female students were listed first, and then the male students. For example, SStA and SStB are two female students in the summer group or G1. The score column was calculated to represent the total scores of the correct answers of the 12 programming questions. One score was assigned to each programming question then the total was calculated out of ten because G1 had only 10 questions. These two columns were used to discuss the results of the three groups in the following sections.

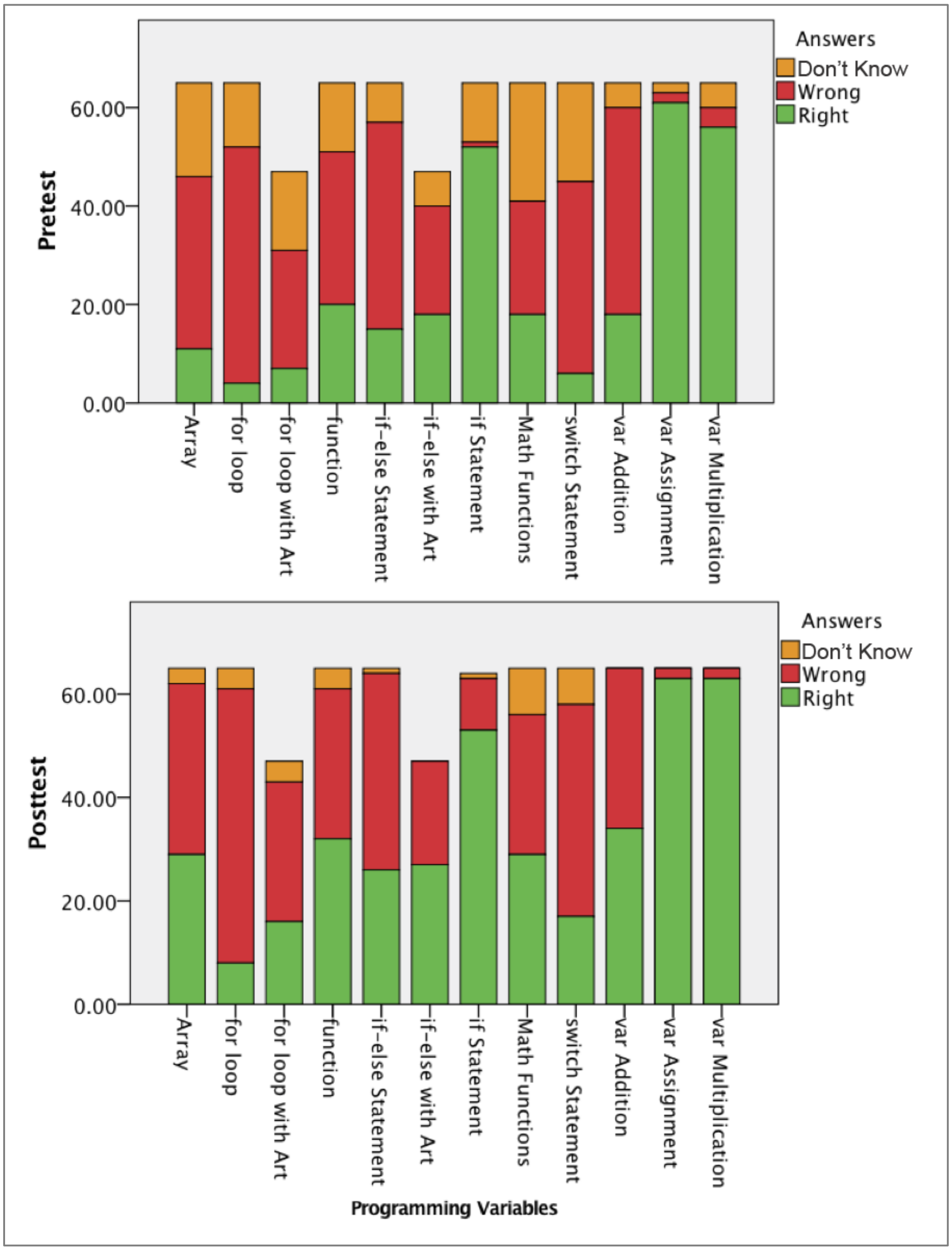
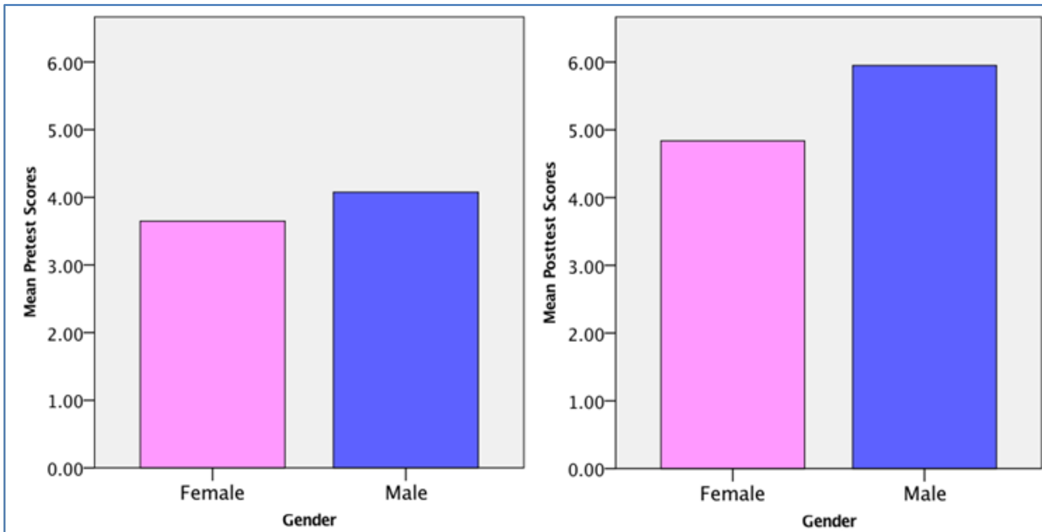


Figure 4-13. $G_{1,2,3}$ results of programming knowledge variables.

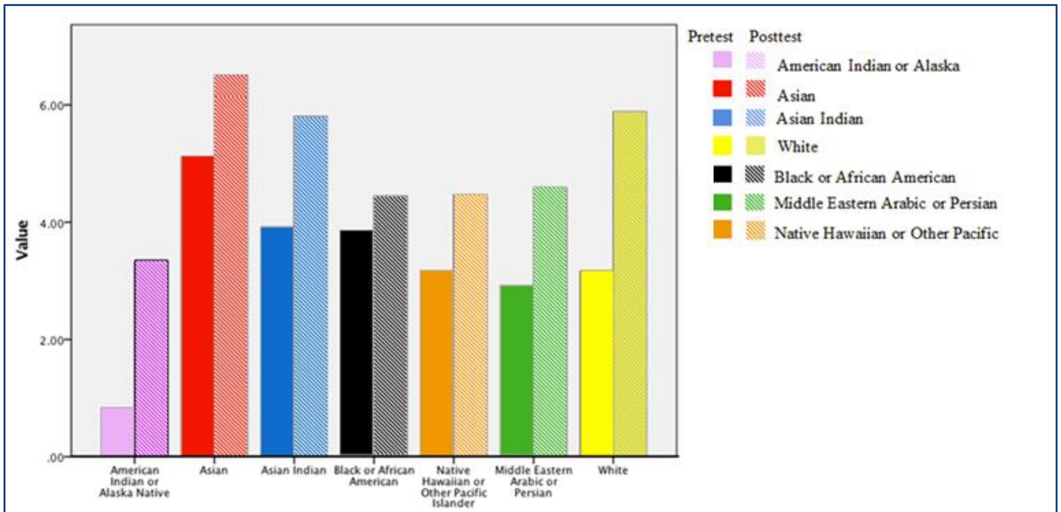
4.4.1 Programming knowledge for $G_{1,2,3}$. The Programming Knowledge (PK) variable was measured by the students' scores in the programming questions. In the posttest, the mean value of the students' scores in $G_{1,2,3}$ was higher than its value in the pretest; it increased by 16% (from 38% to 54%) for all students. The percentage of the students who were able to answer half or more questions rose from 31% in the pretest to 66% in the posttest. The mean value for the male students was higher than the mean value for the female students in both pretest and posttest. The female students' mean value increased by 12%, and the male students' mean value increased by 19% in the posttest. Figure 4-14 shows the descriptive statistic for $G_{1,2,3}$.



Programming Knowledge Mean G3 (out of 10)			
	G3, N=65	Females, N=32	Males, N=33
Pretest	3.8641	3.6458	4.0758
Posttest	5.4026	4.8385	5.9495

Figure 4-14. Gender vs. students' scores in pretest and posttest for $G_{1,2,3}$.

Figure 4-15 shows that the mean values of the Asian students were the highest among the other racial groups in both tests. The Asian group had the highest mean value, but the difference of the mean values (14%) between the pre- and posttests was not the highest. The size of the Asian sample was relatively large among the other racial groups (29%), so we can generalize the result. The best mean difference was for the White group (27%). However, the sample size was relatively small (11%), so we cannot generalize the result for all White students. Despite the shortest coding time for the Middle Eastern students, the mean value increased by 13% in the posttest. The sample size was relatively large (31%), so we can generalize the result. The increase in the mean value was different among the racial groups. It increased by 19% for the Asian Indian group, by 6% for the Black group, by 17% for the Hawaiian group, and by 25% for the American Indian group. The sample sizes of some racial groups were small, so the results cannot be generalized. However, the results give an indication that integrating art and animation with coding could increase students' knowledge in programming for those racial groups.



Programming Knowledge Mean G3 (out of 10)							
G4	American Indian N=1	Asian N=19	Asian Indian N=10	Black N=6	Middle Eastern N=20	Hawaiian N=2	White N=7
Pretest	0.8333	5.0965	3.9167	3.8611	3.1583	2.9167	3.1667
Posttest	3.3333	6.5000	5.8000	4.4444	4.4667	4.5833	5.8810

Figure 4-15. Race vs. mean value of the students scores.

Table 4-6 shows the results of the t-test analysis for all students in $G_{1,2,3}$ and then the results for different gender and racial groups.

Table 4-6

$G_{1,2,3}$ T-Test Results for the Programming Knowledge

	Mean PK Pretest	Mean PK Posttest	N	MD%
$G_{1,2,3}$ (All Students)	3.8641	5.4026	65	15%
Female	3.6458	4.8385	32	12%
Male	4.0758	5.9495	33	19%
American Indian	.8333	3.3333	1	25%
Asian	5.0965	6.5000	19	14%
Asian Indian	3.9167	5.8000	10	19%
Black	3.8611	4.4444	6	6%
Middle Eastern	3.1583	4.4667	20	13%
Native Hawaiian	2.9167	4.5833	2	17%
White	3.1667	5.8810	7	27%
	MD (PK Pretest – PK Posttest)	**t**	**df**	**Sig. (2-tailed)**
$G_{1,2,3}$ (All Students)	-1.53846	-7.703	64	.000
Female	-1.19271	-3.997	31	.000
Male	-1.87374	-7.275	32	.000
Asian	-1.40351	-4.036	18	.001
Asian Indian	-1.88333	-3.638	9	.005
Black	-.58333	-1.131	5	.310
Middle Eastern	-1.30833	-3.433	19	.003
Native Hawaiian	-1.66667	-1.000	1	.500
White	-2.71429	-4.871	6	.003

The result of the t-test for all students in $G_{1,2,3}$ was significant $t(64) = -7.703$, $p = .000$; the p-value was less than α or .05. The results were also significant for both genders $t(31) = -3.997$, $p = .000$ for female and $t(32) = -7.275$, $p = .000$ for male students. The results were significant for all racial groups, except the Black and Hawaiian groups, as shown in Table 4-6.This means that the difference in the mean values of the students' scores between the pretest and the posttest was statistically significant. In other words, the treatment increased students' knowledge in programming for both genders and for most racial groups in $G_{1,2,3}$.

Figure 4-16 shows the individual scores of all the students of $G_{1,2,3}$. The bigger green dot represents the pretest score, and the smaller red dot represents the posttest score. The line between the dots represents a student's progress, and, if the line is missing, then the student had the same score in both tests. Nine females had the same score in the pretest and the posttest, but only one male student had the same score. Most of the students improved their scores. However, four female students and two male students had the red dot below the green dot, which means they had lower scores in the posttest.

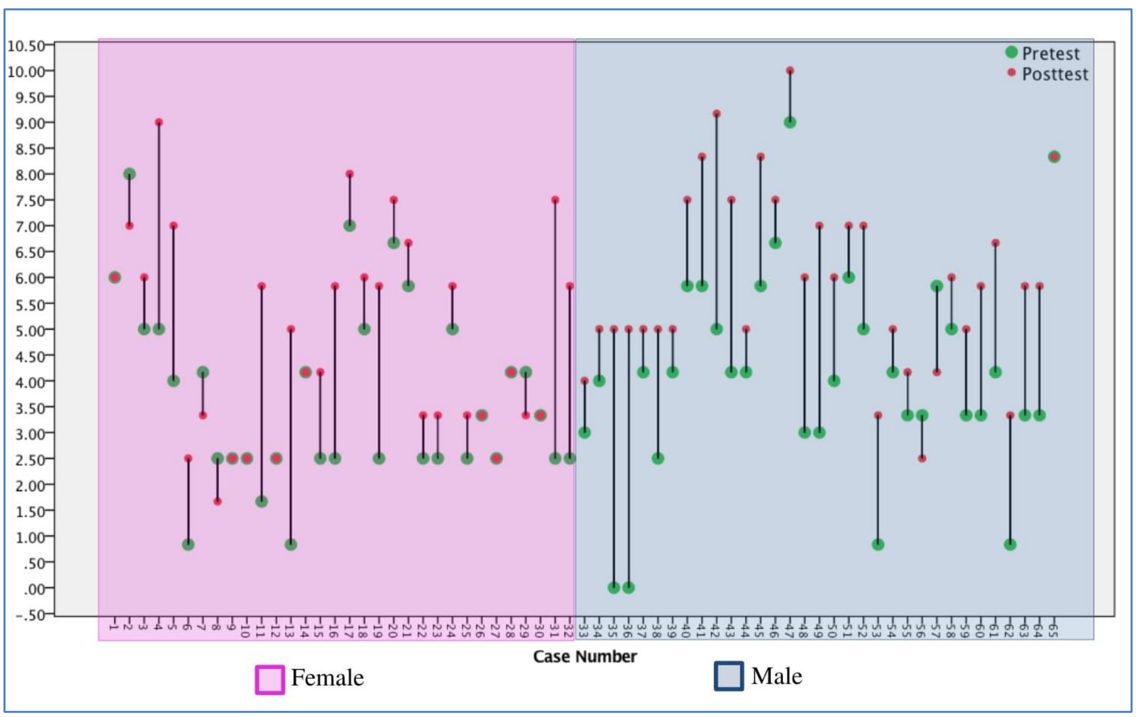


Figure 4-16. $G_{1,2,3}$ gender vs. individual score progress linechart.

In the pretest, the scores' range for the female students was 0.83–8 and it was 0–9 for the male students. In the posttest, these ranges rose slightly to be 1.67–9 for the female and 2.5–10 for the male students. The percentages of the students who were able to answer half or more questions in the pretest were 28% for female students and 33% for male students. These percentages improved in the posttest to 50% for females and 82% for male students.

Figure 4-17 shows the line chart of the individual scores for different racial groups in $G_{1,2,3}$. The percentages of the students who were able to answer half or more of the programming questions in the pretest were 63% Asian, 40% Asian Indian, 17% Black, 10% Middle Eastern, 14% White, and 0% for both American Indian and Hawaiian. In the posttest, these percentages rose to 84% Asian, 80% Asian Indian, 50% Black, 40% Middle Eastern, 100% White, and 50% Hawaiian. The students' scores ranges in the pretest were 0.83–9 for Asian, 0–6.7 for Asian Indian, 2.5–5.8 for Black, 0.83–5.83 for Middle Eastern, 0–5.83 for White, 2.5–3.3 for Hawaiian, and the only American Indian had .83. Those scores ranges rose in the posttest for most of the different racial groups to be 3.3–10 for Asia, 2.5–8.33 for Asian Indian, 1.67–8.33 for Black, 2.5–9.17 for Middle Eastern, and 5–7.5 for White.

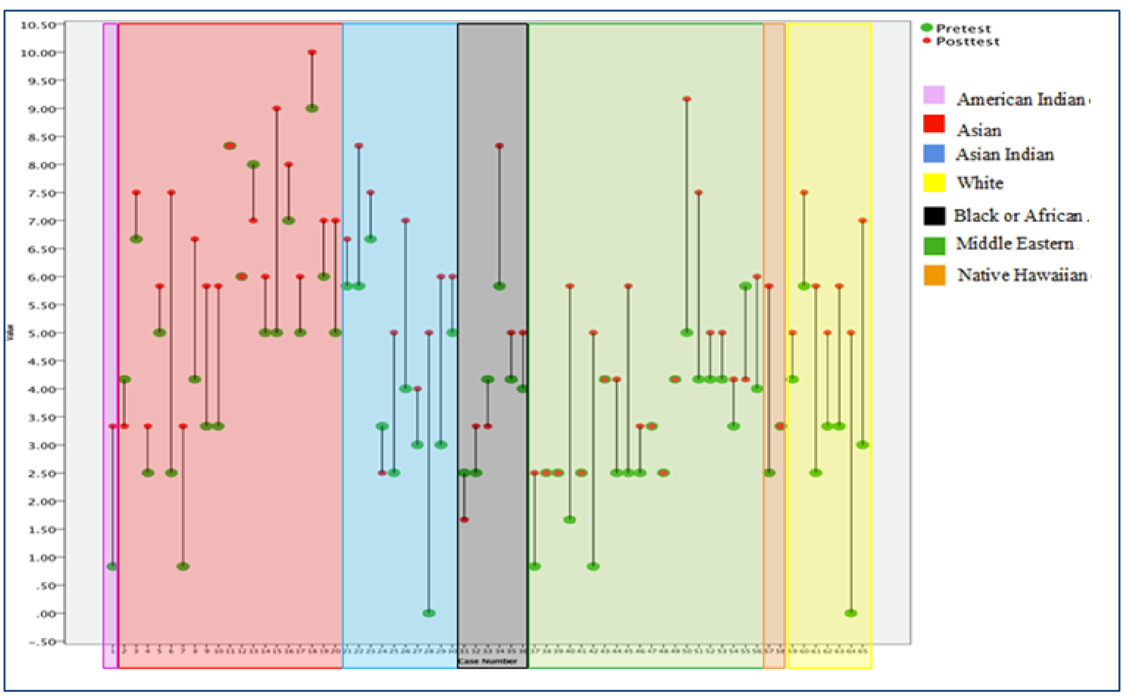


Figure 4-17. $G_{1,2,3}$ race vs. individual score progress linechart.

4.4.2 Programming knowledge for G3 vs. $G_{1,2}$. The mean values of the students' scores in $G_{1,2}$ were higher than the mean values in G3 in the pretest ($G_{1,2}$: M = 3.9688, G3: M = 3.7626) Similarly, the posttest mean value for the $G_{1,2}$ was higher than its value in G3 (G3: M = 4.7727, $G_{1,2}$: M = 6.0521) The female students in $G_{1,2}$ (pretest : M= 4.6212, posttest: M = 6.5000) had higher scores than the female students in G3 (pretest: M = 3.1349, posttest: M = 3.9683), but the male students in $G_{1,2}$ (pretest: M= 3.6270, posttest: M = 5.8175) had lower scores than the male students in G3 (pretest: M = 4.8611, posttest: M = 6.1806) despite fewer coding hours for G3.

The largest racial group in G3 was Middle Eastern (N = 19) while the largest one in $G_{1,2}$ was Asian (N = 15). Comparing the mean values of these two groups in both tests shows that the mean values for the Asian group (pretest: M = 5.2, posttest: M = 6.9) were higher than the mean values of the Middle Eastern group (pretest: M = 3.11, posttest: M = 4.38). The Asian Indian group had the highest mean values in G3 in both tests, and the Asian group had the highest mean in $G_{1,2}$. Figure 4-18, Figure 4-19, and Table 4-7 show the statistics for G3 and $G_{1,2}$ in the pretest and in the posttest.

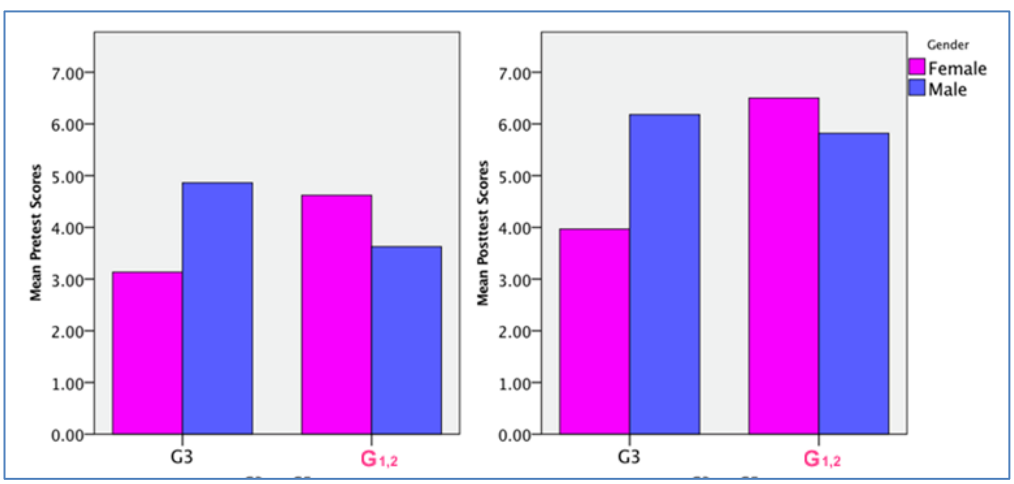


Figure 4-18. Gender vs. scores' mean in pretest and posttest for G3 and $G_{1,2}$.

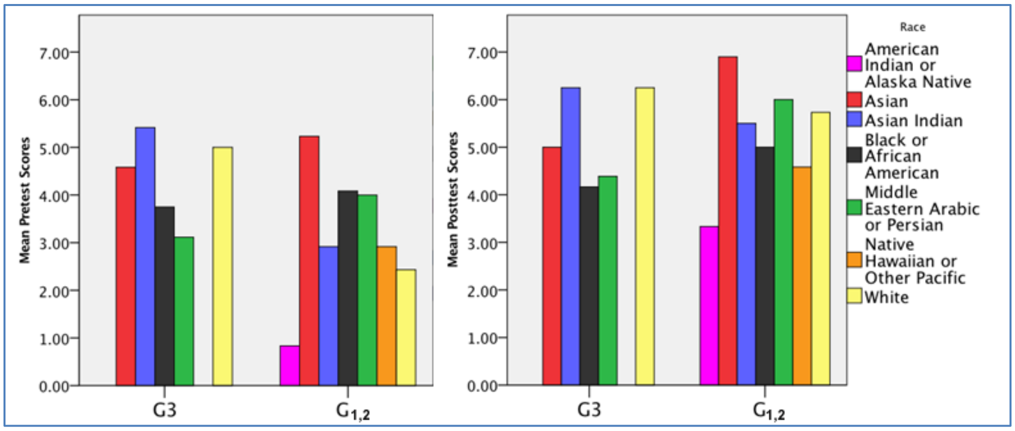


Figure 4-19. Race vs. scores' mean in pretest and posttest for G3 and $G_{1,2}$.

Table 4-7

G3 vs. $G_{1,2}$ Mean Values of the Programming Knowledge

	Mean PK Pretest	Mean PK Posttest	N
G3	3.7626	4.7727	33
Female	3.1349	3.9683	21
Male	4.8611	6.1806	12
Asian	4.5833	5.0000	4
Asian Indian	5.4167	6.2500	4
Black or African American	3.7500	4.1667	4
Middle Eastern Arabic or Persian	3.1140	3.1140	19
White	5.0000	6.2500	2
$G_{1,2}$	3.9688	6.0521	32
Female	4.6212	6.5000	11
Male	3.6270	5.8175	21
American Indian or Alaska Native	.8333	3.3333	1
Asian	5.2333	6.9000	15
Asian Indian	2.9167	5.5000	6
Black or African American	4.0833	5.0000	2
Middle Eastern Arabic or Persian	4.0000	6.0000	1
Native Hawaiian or Other Pacific	2.9167	4.5833	2
White	2.4333	5.7333	5

Most of the t-test results, shown in Table 4-8, were statistically significant. Both G3 ($t(32) = -3.799$, $p = .001$) and $G_{1,2}$ ($t(31) = -7.703$, $p = .000$) were significant with p equaling a value less than alpha. For female students, the results in both groups G3 ($t(20) = -2.603$, $p =$

.017) and $G_{1,2}$ ($t(10) = -3.222$, $p = .009$), were significant. Similarly, the results for the male students in both groups were also statistically significant. The result was $t(11) = -2.777$, $p = .018$ for G3 and it was $t(20) = -7.647$, $p = .000$ for $G_{1,2}$. Results show that the treatment was effective for both groups, G3 and $G_{1,2}$, and it increased the students' programming knowledge.

Table 4-8

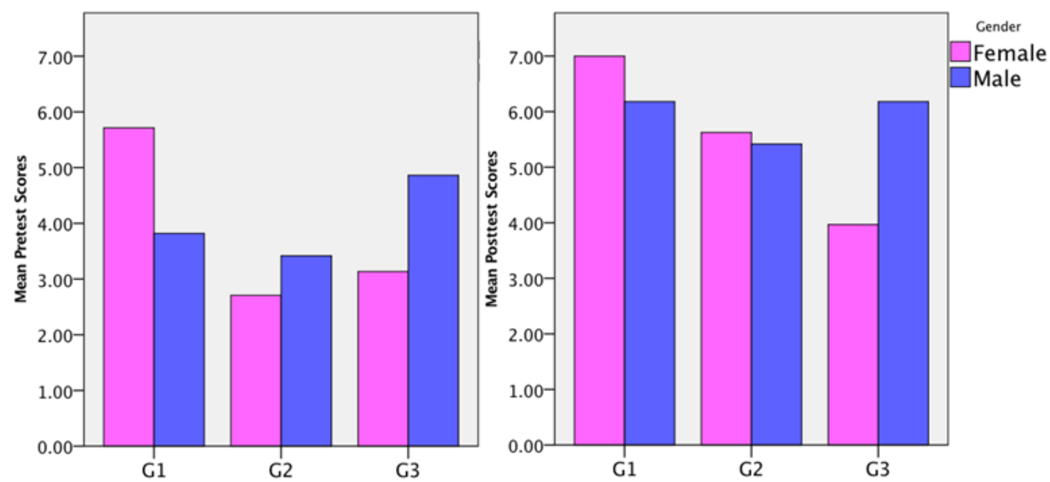
G3 vs.$G_{1,2}$ T-Test Results for the Programming Knowledge

	MD (PK Pretest – PK Posttest)	t	df	Sig. (2-tailed)
G3	-1.01010	-3.799	32	.001
Female	- .83333	-2.603	20	.017
Male	-1.31944	-2.777	11	.018
Asian	-.41667	-1.000	3	.391
Asian Indian	-.83333	-1.225	3	.308
Black or African American	-.41667	-.522	3	.638
Middle Eastern	-1.27193	-3.181	18	.005
White	-1.25000	-3.000	1	.205
$G_{1,2}$	-2.08333	-7.703	31	.000
Female	-1.87879	-3.222	10	.009
Male	-2.19048	-7.647	20	.000
Asian	-1.66667	-4.122	14	.001
Asian Indian	-2.58333	-4.226	5	.008
Black or African American	-.91667	-11.000	1	.058
Native Hawaiian	-1.66667	-1.000	1	.500
White	-3.30000	-5.706	4	.005

The Middle Eastern group was the only group to have significant results in G3 ($t(18) = -.181$, $p = .005$). All the racial groups in $G_{1,2}$ had significant results except the Hawaiian group ($t(1) = -1.000$, $p = .500$), but the sample size was very small for this racial group (N=2). For the largest racial group in $G_{1,2}$, which was the Asian group ($N = 15$), the result was significant ($t(14) = -4.122$, $p = .001$). It was also significant for the Black ($t(1) = -11.00$, $p = .058$) and the White ($t(4) = -5.706$, $p = .005$) groups.

4.4.3 Programming knowledge for G1, G2, and G3. In G1, the mean value of the students' scores increased from 4.5 in the pretest to 6.5 in the posttest, from 45% to 65%, and the mean value increased by 20%. In G2, the mean value of the students' scores increased by 22% in the posttest. It rose from 3.2 in the pretest to 5.4 in the posttest. This increment was little more than the mean increment of students in G1. The mean value of the students 'scores in G3 had the lowest increase among the three groups. It increased from 3.7 in the pretest to 4.7 in the posttest, or it increased by 10%.

The descriptive statistics, shown in Figure 4-20, show that the mean value of the female students in G1 (M = 5.71, M = 7) was more than the mean value for the male students (M = 3.81, M = 6.18) in both pretest and posttest.



G1- Programming Knowledge Mean (out of 10)			
G1	G1 N=18	Females N=7	Males N=11
Pretest	4.5556	5.7143	3.8182
Posttest	6.5000	7.0000	6.1818

G2 - Programming Knowledge Mean (out of 10)			
G2	G2, N=14	Females, N=4	Males, N=10
Pretest	3.2143	2.7083	3.4167
Posttest	5.4762	5.6250	5.4167

G3-Programming Knowledge Mean (out of 10)			
G3	G3, N=33	Females, N=21	Males, N=12
Pretest	3.7626	3.1349	4.8611
Posttest	4.7727	3.9683	6.1806

Figure 4-20. Programming knowledge scores in G1, G2, and G3.

However, the difference between the pretest and the posttest mean values of the male students was 2.87 (6.18 - 3.81), which is larger than the difference in mean values for the female students (7 - 5.7=1.3). This means that the treatment was more effective for the male students than for the female students. The researcher found this result interesting as the use of art with coding increased the male students' knowledge in programming.

For G2, the results show that the mean value for the female students improved more than the mean value of the male students. The mean value of the female students increased by 30% in the posttest, while the mean value of the male students increased by 20%. However, the mean value for the female students was lower than the mean value of the male students in the pretest, but it was a little higher in the posttest. It was also lower than the mean value of the female students of G1. Taking into consideration that the female students of G2 were 9th graders, while the females of the other two groups were 10th and 11th graders, this could explain the difference.

In G3, the mean values for male students (M = 4.86, M = 6.18) were higher than those for the female students (M = 3.13, M = 3.96) in both pre- and posttests. The mean value improved by 8% for the female students, and it improved by 13% for the male students.

Running the SPSS paired samples t-test analysis to compare the students' scores in the pretest and the posttest for the three groups gave the results shown in Table 4-9. One student was lost before the posttest, and the t-test compared the two samples of the same size so the score of the lost student was eliminated.

The paired sample t-test results were all significant for G1, G2, and G3 with $t(17) = -4.862$, $p = .000$ for G1, $t(13) = -6.388$, $p = .000$ for G2, and $t(32) = -3.799$, $p = .001$ for G3. The p-values for the three groups were less than α or .05, meaning that the differences in mean values of the students' scores were statistically significant between the pretest and the posttest in the three groups. In other words, the treatment increased students' knowledge in programming.

The result for the female students in G1 was $t(6) = -1.996$, $p = .093$. The p-value suggests that the difference in the mean values of the female scores between the pretest and the posttest was quasi-significant; it had a statistical trend toward significance (Martz, 2015). The result for the male students in G1, however, was significant $t(10) = -4.812$, $p = .001$. The results show that the treatment was more effective for male students in G1 (*MD* = -2.36364) than for females. However, results also showed that the female students' scores improved in the posttest (*MD* = -1.28571).

In G2, the result for the male students was significant ($t(9) = -7.060$, $p = .000$), but the t-test result for the female students was $t(3) = -2.782$, $p = .069$. This p-value was slightly larger than the .05, but it suggests that the difference in the mean values of the female scores between the pretest and the posttest approached the borderline of significance (Martz, 2015). The results also show that the increment in the mean value of the female students' scores in the posttest was higher than the increment for the male students and the mean difference for females was higher than that for male students (*MD* = -2.91667).

Table 4-9
G1, G2, and G3 T-Test Results for the Programming Knowledge

	MD (PK Pretest – PK Posttest)	t	df	Sig. (2-tailed)
G1	-1.94444	-4.862	17	.000
Female	-1.28571	-1.996	6	.093
Male	-2.36364	-4.812	10	.001
G2	-2.26190	-6.388	13	.000
Female	-2.91667	-2.782	3	.069
Male	-2.00000	-7.060	9	.000
G3	-1.01010	-3.799	32	.001
Female	-.83333	-2.603	20	.017
Male	-1.31944	-2.777	11	.018

In G3, the results were significant for both genders, $t(20) = -2.603$, $p = .017$ for the female students and $t(11) = -2.777$, $p = .018$ for the male students.

Table 4-10 and Figure 4-21 show the results for different racial groups. The race results for each group (G1, G2, and G3) were only discussed for this study variable (PK). For other variables, the race was discussed for $G_{1,2,3}$ only.

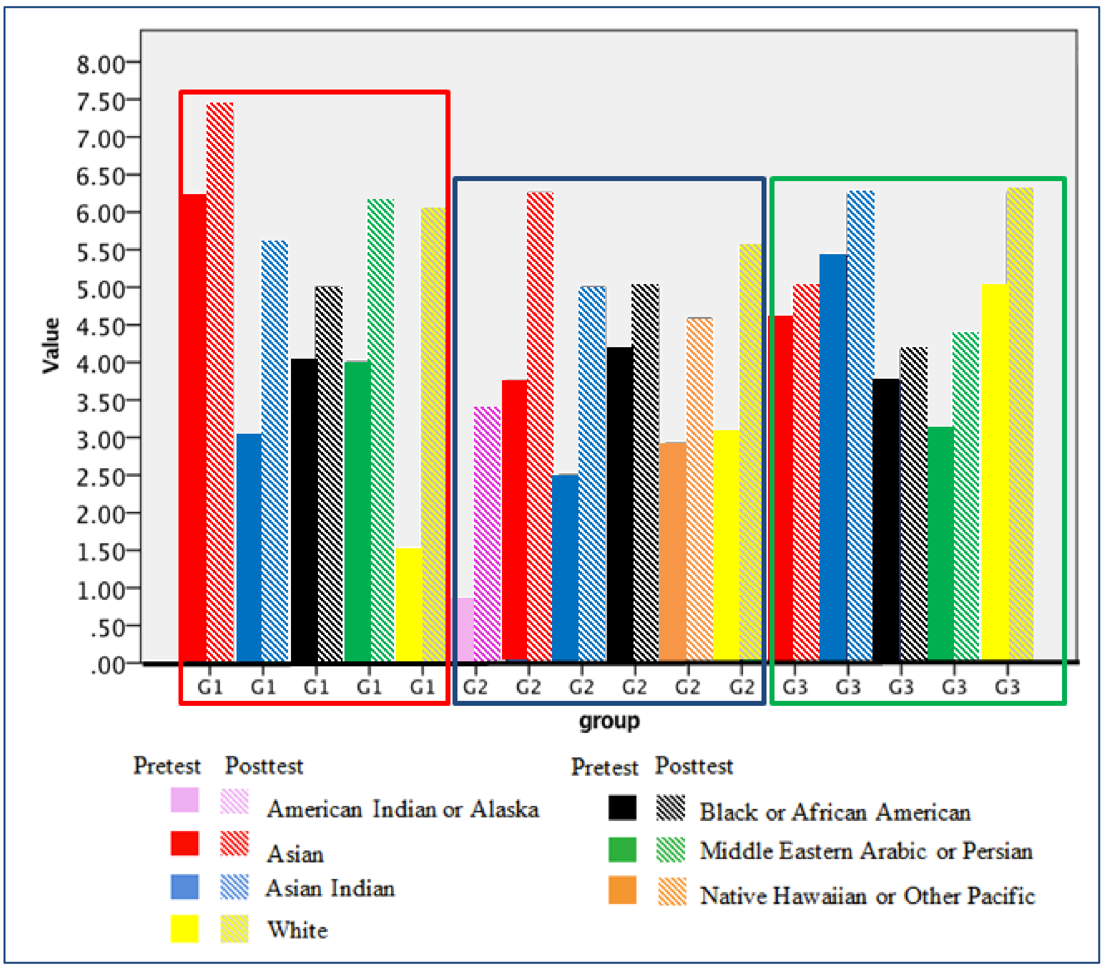


Figure 4-21. Race vs. the mean of the students scores for the three groups.

The Asian students in G1 had the highest mean value among the other racial groups in both tests. They increased their mean value by 11% from 6.2 in the pretest to 7.3 in the posttest. The increment in the mean value of the Asian Indian group was more than the increment for the Asian group. It was 26% where the mean increased from 3.0 in the pretest to 5.6 in the posttest. The increment of the mean value for the White group was the highest, 45%, and the mean value rose from 1.5 to 6.0 in the posttest. Only one Black student and one Middle Eastern student participated in the coding camp, and their scores increased in the posttest. The results for G2 show that the mean values improved for some racial groups in the posttest. The Asian groups had the highest mean value in the posttest. The Asian and White students had the same mean increment percentage, 25%. American Indian, Asian Indian, and Black students had a small participation of only one student each. The mean values were the same as the students' scores for those small groups which were improved in the posttest. The mean values for all racial groups in G3 increased in the posttest, as shown in Table 4-10. The highest increment was for the Middle Eastern and White groups where the mean values increased by 13% for both groups, but the mean values increased by only 4% for the Asian and Black groups and by 8% for the Asian Indian group.

Running the dependent paired sample t-test for the data of different racial groups in the three groups separately gave the results shown in Table 4-10. Before running the t-test analysis, the data file was divided into different racial groups. SPSS does not give results for sample sizes less than one, which was the case for the Black and the Middle Eastern groups in G1 and the Asian Indian, American Indian, and Black groups in G2. The results for both Asian ($t(8) = -2.443$, $p = .04$) and Asian Indian ($t(4) = -3.474$, $p = .025$) in G1 were statistically significant. For the White group in G1 ($t(2) = -9.000$, $p = .070$), although the p-value is greater than .05, the difference in the mean value between the pretest and the posttest is relatively large (4.5). The result approached the borderline of significance, or the treatment is still effective for this racial group.

In G2, the results were significant for the Asian ($t(5) = -3.873$, $p = .012$) and White ($t(2) = -5.196$, $p = .035$) groups. No significant result was found for the Hawaiian group. In G3, the only significant result was found for the Middle Eastern group ($t(18) = -3.181$, $p = .005$). The study found that the treatment was effective and increased students' programming knowledge.

Table 4-10
G1, G2, and G3 PK T-Test Results for Different Racial Groups

Group	Race	N	Pretest Mean	Posttest Mean	Mean Increment %
G1	Asian	9	6.22	7.33	11%
	Asian Indian	5	3	5.6	26%
	Black	1	4	5	10%
	Middle Eastern	1	4	6	20%
	White	2	1.5	6	45%
G2	American Indian	1	.83	3.33	25%
	Asian	6	3.75	6.25	25%
	Asian Indian	1	2.5	5	25%
	Black	1	4.17	5	8%
	Native Hawaiian	2	2.92	4.58	17%
	White	3	3.06	5.56	25%
G3	Asian	4	4.58	5	4%
	Asian Indian	4	5.42	6.25	8%
	Black or	4	3.75	4.17	4%
	Middle Eastern	19	3.11	4.39	13%
	White	2	5	6.25	13%

		MD (PK Pretest –PK Posttest)	t	df	Sig. (2-tailed)
G1	Asian	-1.11111	-2.443	8	.040
	Asian Indian	-2.60000	-3.474	4	.025
	White	-4.50000	-9.000	1	.070
G2	Asian	-2.50000	-3.873	5	.012
	Hawaiian	-1.66667	-1.000	1	.500
	White	-2.50000	-5.196	2	.035
G3	Asian	-.41667	-1.000	3	.391
	Asian Indian	-.83333	-1.225	3	.308
	Black	-.41667	-.522	3	.638
	Middle Eastern	-1.27193	-3.181	18	.005
	White	-1.25000	-3.000	1	.205

4.4.4 Student's self-assessment of programming knowledge level. In addition to the programming knowledge assessment questions, students were asked in the pretest and the posttest to self-assess their programming knowledge level, on a scale of 1 to 5. The percentage of students who chose three or higher as their programming knowledge level rose from 15% in the pretest to 49% in the posttest, and the percentage of the students who chose a level of one dropped from 51% to 18% in the posttest (Figure 4-22). This item also indicates that half of the participants (51%) reported having no programming experience at the beginning of the experiment, and 34% reported having a little programming experience. That students reported having had little to no prior programming experience increased the internal validity of the study because it indicates that the improvement in programming knowledge was caused by the treatment itself, and not because students had had prior experience in programming. This item also suggests that the students' interest was also affected by the treatment itself. Because most of the students had had little programming experience at the pretest, this indicates that they had not had enough exposure to programming to know if they had interest or not.

This variable was not used to accept or reject any hypothesis; it is discussed as an additional finding. However, the t-test result, $t(64) = -6.480274$, $p = .000$, shows a significant difference in the mean values for all students. In other words, students indicated an increase in their programming level, which agrees with the study assessment of the programming knowledge.

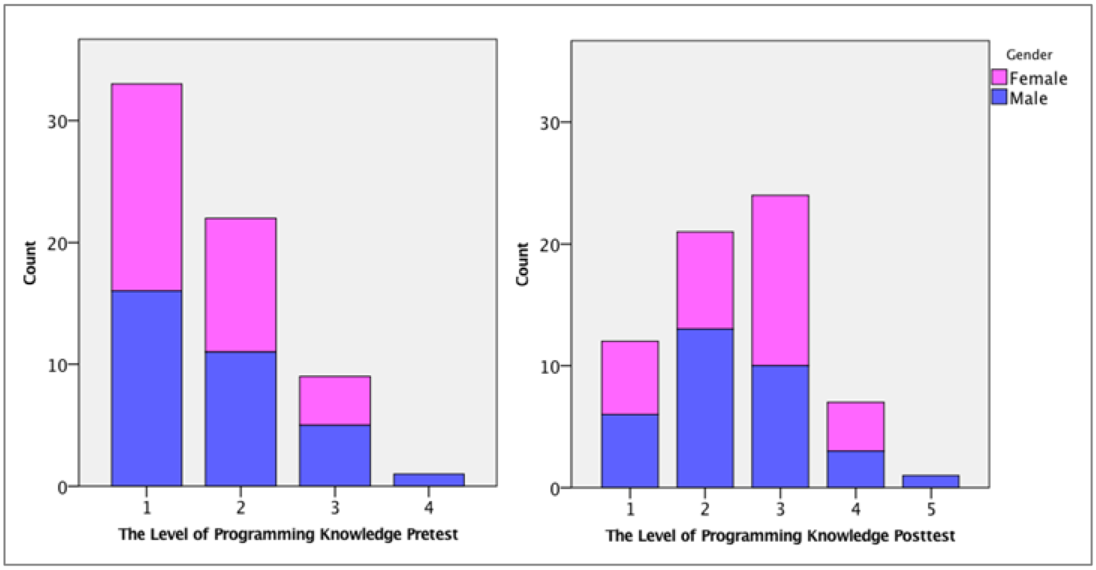


Figure 4-22. $G_{1,2,3}$ self-assessment of the programming level in both tests.

4.5 The Real Programming Language Preference

In this section, students were asked to answer questions on a 5-point Likert scale, 5 for *strongly agree* and 1 for *strongly disagree*. This scale was used to measure the effect of using art, animation, and code sharing on the students' preference of writing a program with a real programming language rather than a block-based programming language. JavaScript was the real programming language used in this experimental study. The results of the t-test analysis show a significant difference between the students' answers in the pretest and the posttest.

The students' answers to the five survey items were added together, and the averages were calculated into two new columns (RPP Pretest and RPP Posttest) where RPP stands for Real Programming Preference. Similar to other study variables, the results are discussed in the following three sections for $G_{1,2,3}$, for G3 versus $G_{1,2}$, and for each group.

4.5.1 Real programming preference for $G_{1,2,3}$. The RPP increased for all students in the posttest. The agreement percentage on the survey items that were used to measure this variable increased from 22% to 54%, and the disagreement percentage decreased from 28% to 12% in the posttest. The agreement percentage for the female students increased from 3%

to 50%, and it increased from 39% to 61% for the male students. The disagreement percentage decreased from 41% to 16% for the female students, and it decreased from 15% to 9% for the male students. Male students had higher preference to the real programming language than the female students in both tests.   Even though, the female students changed their preference more than the male students did.

The mean value of the RPP variable in the posttest (*M* = 3.46) was higher than its value in the pretest (*M* = 2.88) for all students in $G_{1,2,3}$. This means that more students agreed or strongly agreed with the survey items that measured this variable in the posttest. This indicates improvement in the students' preference for the real programming. Figure 4-23 shows the $G_{1,2,3}$ students' responses to this variable in the pretest and posttest.

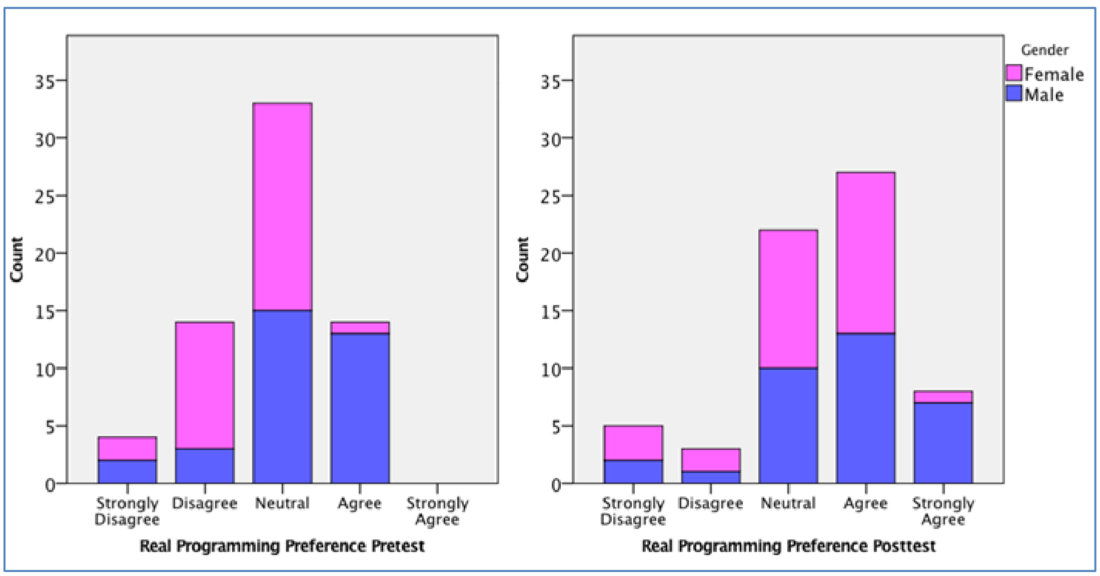


Figure 4-23. $G_{1,2,3}$ real programming preference in the pretest and posttest

Table 4-11 shows that the t-test result was significant for all students in $G_{1,2,3}$ for this variable ($t(64) = -5.141$, $p = .000$). The mean values for the male students were higher than the mean values for the female students in both tests. However, the mean difference for the female students was a little larger than that for the male students. The results for both genders were significant with $t(31) = -4.739$, $p = .000$ for female and $t(32) = -2.775$, $p = .009$ for male students.

Table 4-11

G1,2,3 T-Test of the RPP Variable

	RPP Mean in Pretest	RPP Mean in Posttest	N
$G_{1,2,3}$	2.88	3.46	65
Female	2.56	3.25	32
Male	3.18	3.67	33
American Indian	3.00	3.00	1
Asian	3.05	3.89	19
Asian Indian	3.00	3.80	10
Black or African American	2.67	2.83	6
Middle Eastern	2.55	2.90	20
Native Hawaiian	3.00	3.50	2
White	3.29	4.00	7

	Mean (RPP Pretest – RPP Posttest)	t	df	Sig. (2-tailed)
$G_{1,2,3}$	-.585	-5.141	64	.000
Female	-.688	-4.739	31	.000
Male	-.485	-2.775	32	.009
Asian	-.842	-4.800	18	.000
Asian Indian	-.800	-2.058	9	.070
Black or African American	-.167	-.415	5	.695
Middle Eastern	-.350	-2.101	19	.049
Native Hawaiian	-.500	-1.000	1	.500
White	-.714	-1.508	6	.182

The mean differences for all racial groups were negative, which indicates that students' agreement improved in the posttest. The result was significant for Asian students, ($t(18) = -4.800$, $p = .000$) for this variable. It was also significant for the Middle Eastern students, ($t(19) = -2.101$, $p = .049$). The result approached the border of significance for the Asian Indian students, ($t(9) = -2.058$, $p = .070$).

4.5.2 Real programming preference for G3 vs. $G_{1,2}$. The agreement percentage for G3 (33%) was lower than that for $G_{1,2}$ (75%) in the posttest. However, the percentage increased for both groups in the posttest. It increased from 18% to 33% for G3, and it increased from 25% to 75% for $G_{1,2}$. The disagreement percentage decreased from 52% to 24% for G3, and it decreased from 3% to 0% for $G_{1,2}$. Figure 4-24 shows the students' responses for both groups in the pretest and posttest.

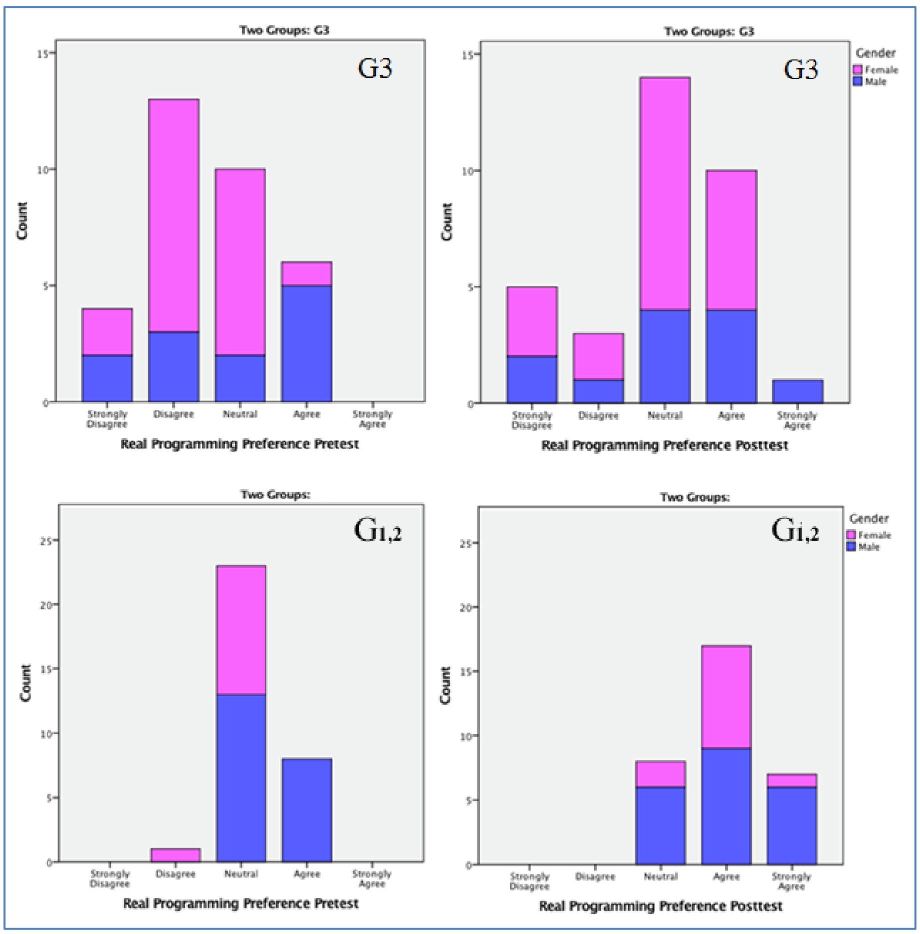


Figure 4-24. G3 vs. $G_{1,2}$ real programming preference in both tests.

The mean value of the students' responses to the real programming preference in $G_{1,2}$ was higher than the mean value in G3 for both tests as shown in Table 4-12, although the mean value did improve in both groups.

The mean difference for $G_{1,2}$ was higher than the mean difference for G3 (*MD* = -.42, *MD* = -.75). The t-value for $G_{1,2}$ was higher than the t-value for G3, and the p-value for $G_{1,2}$ was smaller, $t(31) = -.750$, $p = .000$ for $G_{1,2}$, and $t(32) = -2.514$, $p = .017$ for G3.

Table 4-12

The T-Test Results of the RPP in G3 and $G_{1,2}$

	RPP Mean in Pretest	RPP Mean in Posttest		N
G3	2.55	2.97		33
Female	2.38	2.90		21
Male	2.83	3.08		12
$G_{1,2}$	3.22	3.97		32
Female	2.91	3.91		11
Male	3.38	4.00		21
	Mean (RPP Pretest – RPP Posttest)	**t**	**df**	**Sig. (2-tailed)**
G3	- .424	-2.514	32	.017
Female	-.524	-2.750	20	.012
Male	-.250	-.761	11	.463
$G_{1,2}$	-.750	-5.036	31	.000
Female	-1.000	-5.244	10	.000
Male	-.619	-3.081	20	.006

A result is more significant for a larger t-value and a smaller p-value (Runkel, 2016). However, the results for both groups were significant. Both genders had significant results in $G_{1,2}$ ($t(10) = -5.24$, $p = .00$ for female students and $t(20) = -3.08$, $p = .006$ for male students). In G3, female students only had significant results ($t(20) = -2.75$, $p = .012$). The female students in G3 had the highest mean improvement for this variable (*MD* = -1.0), and the male students of $G_{1,2}$ had the lowest mean difference (*MD* =-.25) .The significant results for both groups G3 and $G_{1,2}$ indicate that the students' preference for the real programming significantly improved regardless of the different coding time.

4.5.3 Real programming preference for G1, G2, and G3. In the posttest, the mean values of G1 (M = 4.00) and G2 (M = 3.93) were higher than the mean value of G3 (M = 2.79). In other words, more students preferred the real programming over the block programming in G1 and G2 than in G3. However, the t-test results were significant for all three groups (p = .00 for G1, p = .047 for G2, and p = .017 for G3), and the students' preference improved significantly in all three groups. In G1, the results were significant for both genders, ($t(6)$ = -4.58, p = .004 for female students and $t(10)$ = -3.19, p = .010 for male students). For G2 and G3, only the female students had significant results as shown in Table 4-13.

Table 4-13

The T-Test Results of the RPP Variable in G1, G2, and G3

	RPP Mean in Pretest	RPP Mean in Posttest		N
G1	3.06	4.00		18
Female	2.86	3.86		7
Male	3.18	4.09		11
G2	3.43	3.93		14
Female	3.00	4.00		4
Male	3.60	3.90		10
G3	2.55	2.97		33
Female	2.38	2.90		21
Male	2.83	3.08		12
	Mean (RPP Pretest – RPP Posttest)	t	df	Sig. (2-tailed)
G1	-.944	-4.994	17	.000
Female	-1.000	-4.583	6	.004
Male	-.909	-3.194	10	.010
G2	-.500	-2.188	13	.047
Female	-1.000	-2.449	3	.092
Male	-.300	-1.152	9	.279
G3	-.424	-2.514	32	.017
Female	-.524	-2.750	20	.012
Male	-.250	-.761	11	.463

Figure 4-25 also shows that the agreement (agree and strongly agree) percentage improved in the posttest of all three groups.

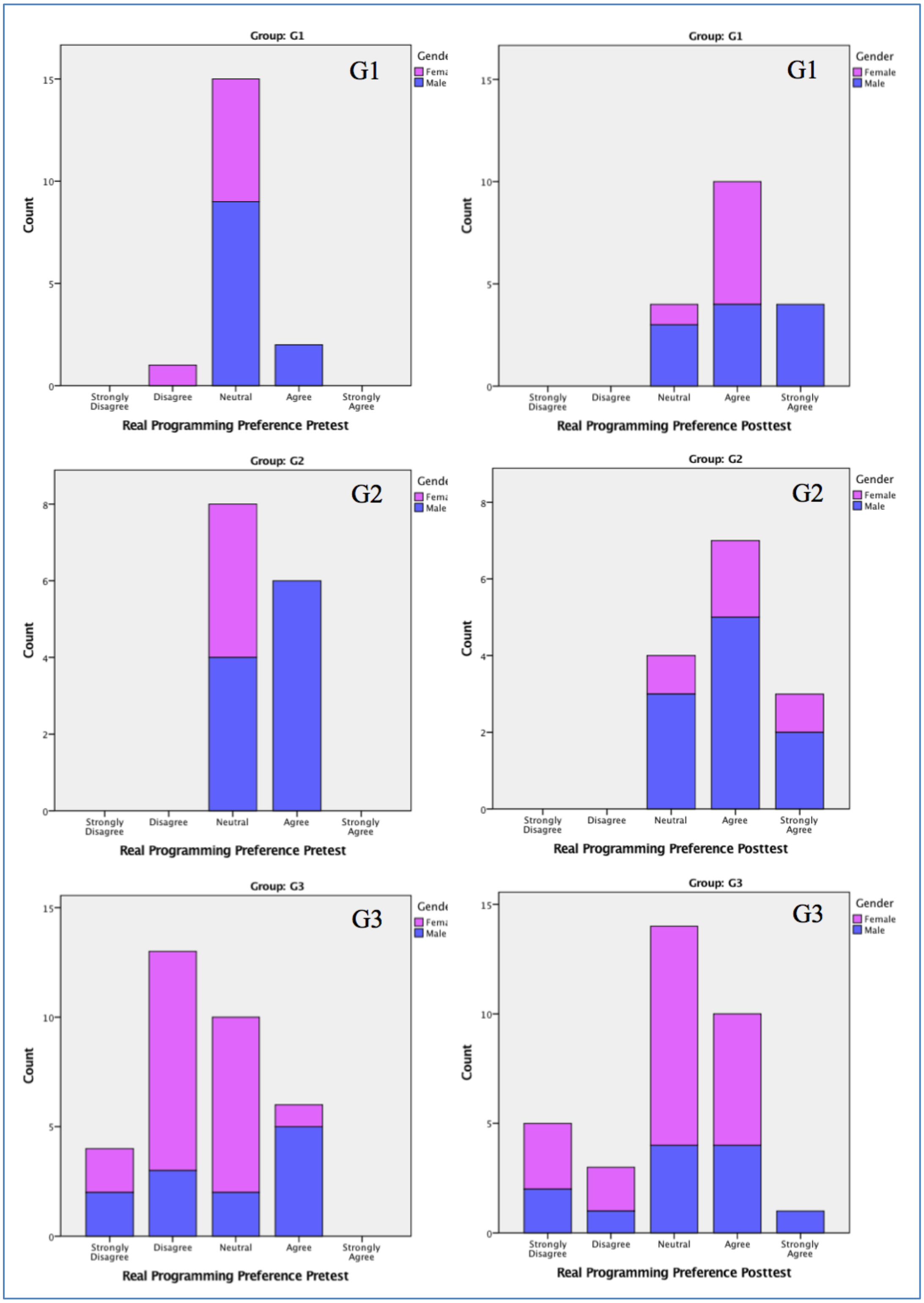


Figure 4-25. G1, G2, and G3 real programming preference.

The agreement percentage increased from 11% to 78% for G1, from 43% to 71% for G2, and from 18% to 33% for G3.

4.6 Motivation for Code Sharing

In this experiment, students wrote code that produced artwork and animated artwork. The sharing feature was explained to students, and some students chose to share their artwork. All students could browse the shared artwork. Table 4-14 shows the number of code lines that were shared in each group, the number of shared artworks, and the number of artworks that included animation. As shown, G1 had the most shared artwork and code lines because the students had more coding hours. Figure 4-26 shows a sample of the shared artwork by a junior Asian male student (SStl) in G1. The researcher noticed his advanced programming skills. The teaching material included introducing the students to the math functions and the for-loop, and student SStl was able to put them together and produce the shown artwork. The number of shared artworks was more than the number of students, which indicates that some students shared artwork more than once, even in G3 with its limited coding time. The number of shared code lines and the number of shared artworks give an indication of the effect of art and animation on the students' motivation for code sharing. However, the survey results were used and analyzed to accept or reject the hypothesis for the Motivation for Code Sharing (MCS) variable.

To measure the effect of integrating art and animation on students' motivation to write and share more code, 11 survey items were used. These items included intrinsic and extrinsic motivation variables from the motivation theory. In this part, students were asked about their coding enjoyment, competing with others, feeling proud to show work, and contribution.

The variable MCS was used to measure the effect of art and animation on the students' motivation for code sharing. The variable was calculated and analyzed as the average for all responses to the questions that were used to measure this variable in the survey questionnaire. As mentioned before, the first six items were used to measure the Intrinsic Motivation for Code Sharing (IMCS) variable, and the rest of the questions were used to measure the Extrinsic Motivation for Code Sharing (EMCS) variable.

Table 4-14
Number of the Shared Artwork

G#	N	Number Code Lines	Number of the Shared Artwork	Number of the Artwork that Include Animation
G1	18	4880	128	47
G2	14	546	22	8
G3	33	1202	60	40

```
var r=0;
for(i=0;i<=12*Math.PI;i+=0.01)
{
    r=3*Math.cos(5*i/6);
    circle(80*r*Math.cos(i)+300,80*r
        *Math.sin(i)+300,1);
}
```

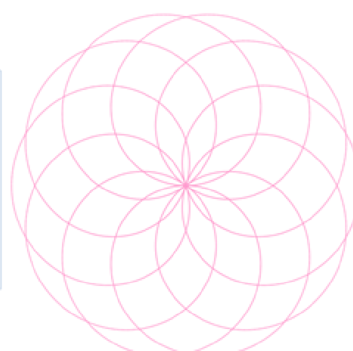
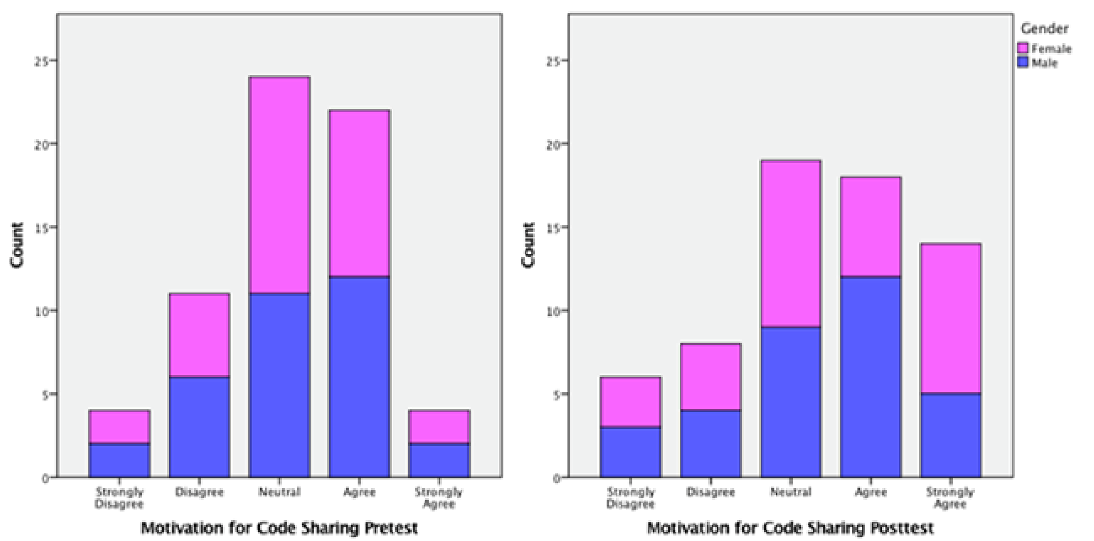


Figure 4-26. Sample of the shared artwork in g1 by student SStl.

4.6.1 Motivation for code sharing $G_{1,2,3}$. As shown in Figure 4-27, the majority of the students in $G_{1,2,3}$ agreed on the art and animation as a motivation for code sharing in both tests. The agreement percentage rose from 40% to 49% in the posttest. The agreement percentage for female students rose from 37% in the pretest to 47% in the posttest. The percentage of female students who chose the *neutral* choice dropped from 41% to 31% in the posttest, and the percentage of female students who chose *strongly agree* rose from 6% to 28% in the posttest. The figure shows this percentage's increment for the female students clearly. Similarly, for the male students, the agreement percentage rose from 42% to 52% in the posttest, and the *neutral* dropped from 33% to 27% in the posttest. The male students who chose the *agree* and the *strongly agree* were more than the females in both tests.



	Agree and Strongly Agree	Neutral	Disagree and Strongly Disagree
Pretest	40%	37%	23%
Posttest	49%	29%	22%

Figure 4-27. $G_{1,2,3}$ motivation for code sharing (MCS).

The mean values of the male and female students were close for this variable in both tests, and they were above the *neutral* option. The mean values for the Asian group were higher than the other racial groups in both tests. This indicated that their motivation for code sharing was higher than the other groups. Taking into consideration that the Asian group dominated G1 and G2 and had more time to share artwork than G3 could explain the high mean value for this racial group.

The t-test result for all students in $G_{1,2,3}$ (Table 4-15) was significant with p-value less than .05 ($t(64) = -2.034$, $p = .046$) and a mean difference of $MD = -.230$ between the pretest and the posttest. The result for the female students approached significance with $t(31) = -1.869$, $p = .071$ and $MD = -.28125$, which is more than the mean difference of all students.

The result for the male students, however, was not significant ($t(32) = -1.063$, $p = .296$, $MD = -.18182$). This suggests that the male students did not significantly change their choices in both tests, but the female students did..

Table 4-15
$G_{1,2,3}$ Mean Values for MCS, IMCS, and EMCS Variables.

		Mean MCS Pretest	Mean MCS Posttest	N
Motivation (MCS)	$G_{1,2,3}$	3.1692	3.4000	65
	Female	3.1563	3.4375	32
	Male	3.1818	3.3636	33
	American Indian	3.00	3.00	1
	Asian	3.79	4.11	19
	Asian Indian	3.30	3.50	10
	Black	3.17	3.17	6
	Middle Eastern	2.80	2.75	20
	Native Hawaiian	3.00	3.50	2
	White	2.43	3.43	7
Intrinsic Motivation (IMCS)	$G_{1,2,3}$	3.1538	3.3692	65
	Female	3.1250	3.3750	32
	Male	3.1818	3.3636	33
Extrinsic Motivation (EMCS)	$G_{1,2,3}$	3.3015	3.3877	65
	Female	3.3188	3.5375	32
	Male	3.2848	3.2424	33

The results of the t-test analysis for different racial groups in $G_{1,2,3}$ are shown in Table 4-16. Other than the Asian group, which had a certain trend toward significance ($p = .083$), no other racial groups had any significant results for the MCS variable.

Table 4-16
$G_{1,2,3}$ T-Test Results of MCS, IMCS, and EMCS Variables

		Mean (MCS Pretest – MCS Posttest)	t	df	Sig. (2-tailed)
Motivation (MCS)	$G_{1,2,3}$	-.23077	-2.034	64	.046
	Female	-.28125	-1.869	31	.071
	Male	-.18182	-1.063	32	.296
	Asian	-.316	-1.837	18	.083
	Asian Indian	-.200	-.802	9	.443
	Black	.000	.000	5	1.000
	Middle Eastern	.050	.326	19	.748
	Native Hawaiian	-.500	-1.000	1	.500
	White	-1.000	-1.620	6	.156
Intrinsic Motivation (IMCS)	$G_{1,2,3}$	-.21538	-1.809	64	.075
	Female	-.25000	-1.544	31	.133
	Male	-.18182	-1.030	32	.311
Extrinsic Motivation (EMCS)	$G_{1,2,3}$	-.08615	- .656	64	.514
	Female	-.21875	-1.307	31	.201
	Male	.04242	.210	32	.835

Figure 4-28 and Figure 4-29 show that the intrinsic motivation changed more than the extrinsic motivation in the posttest. The agreement percentage for the intrinsic motivation for code sharing questions of all students increased from 37% in the pretest to 52% in the posttest. For the female students, the agreement percentage increased from 38% in the pretest to 53% in the posttest. Similarly, the agreement percentage for the male students increased from 36% to 52%, and the disagreement percentage dropped from 25% to 19% for female students and remained unchanged (21%) for the male students.

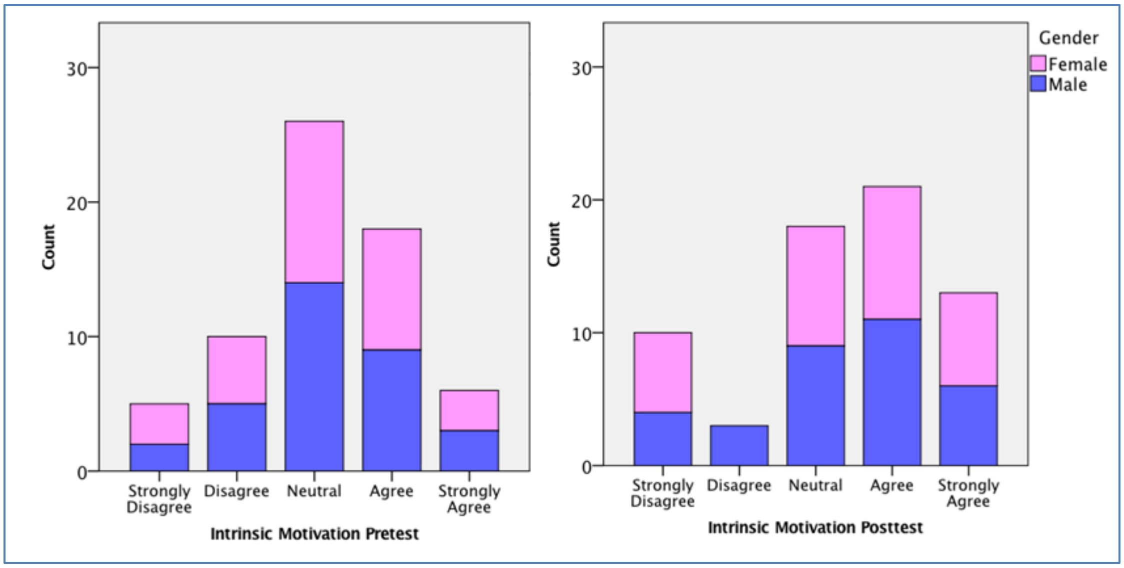


Figure 4-28. $G_{1,2,3}$ intrinsic motivation for code sharing (IMCS).

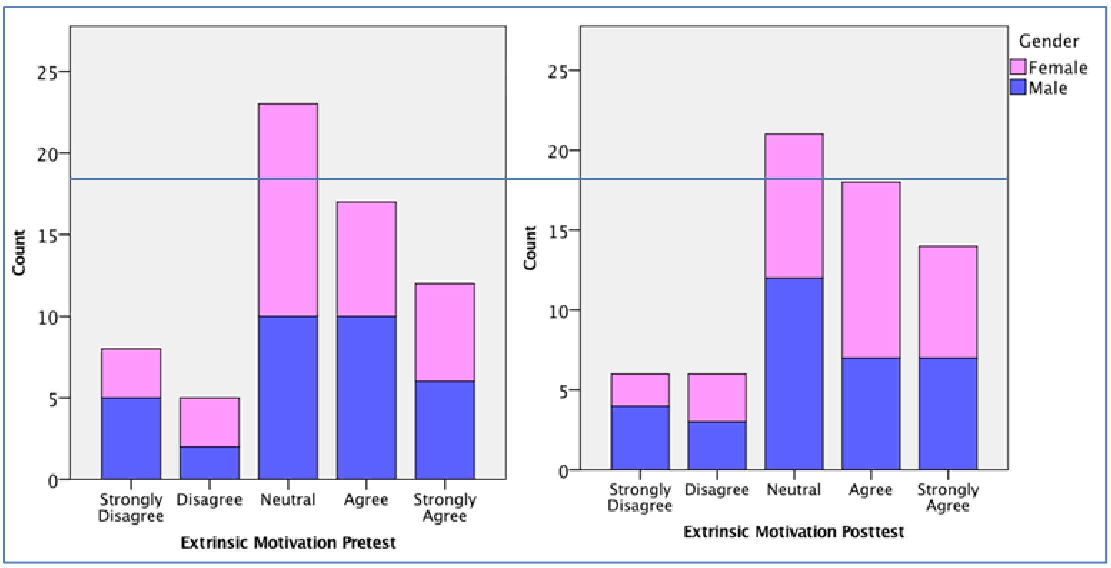


Figure 4-29. $G_{1,2,3}$ extrinsic motivation for code sharing (EMCS).

The agreement percentage of extrinsic motivation for code sharing was high from the beginning of the experiment. The majority of students (45%) agreed on external motivation, such as getting more "likes" or competing with friends who motivated them to share more code. This percentage rose in the posttest to 49% for all students in $G_{1,2,3}$. For the female students, the agreement percentage rose from 40% to 56% while it dropped slightly for male students from 48% to 42%, which was still high. The disagreement for the females dropped from 19% in the pretest to 16% in the posttest, and it remained the same for the male students. Few male students who chose the "agree" option changed to the neutral option in the posttest. The t-test results for individual survey items were significant for the following items:

- For intrinsic motivation:
  - Writing a program that produces art and design encourages me to share my code.
  - Writing a program that produces animation encourages me to share my code.
- For extrinsic motivation:
  - My friend shares her/his code, and I like to share my code.
  - I like to compete with my friends by writing a program that produces a cooler design and get more likes.

Table 4-16 shows that mean values of both variables (IMCS and EMCS) were above the neutral option, and the majority of the students agree that the use of art and animation in programming increased their motivation for code sharing. In $G_{1,2,3}$, the t-test result for intrinsic motivation was not significant for male students, and it approached significance for female students ($t(64) = -1.809$, $p = .075$). The results for the extrinsic motivation were not significant for both genders. The mean values of the IMCS and the EMCS for the female students were higher than the male students in the posttest.

4.6.2 Motivation for code sharing G3 vs. $G_{1,2}$. Coding time is an important factor in the MCS variable. The MCS variable was measured in five days for $G_{1,2}$ students, but it was measured in the same day for G3 students. Students of $G_{1,2}$ had more time to explore the treatment tool and to share more code from home. In G3, the coding time was 3 hours in one day, but the important factor was the number of students, which was more than the other two groups, and the competition to produce artwork and share it with a limited time was higher than the other two groups. Figure 4-30 shows the students responses in G3 vs. $G_{1,2}$ for the MCS variable.

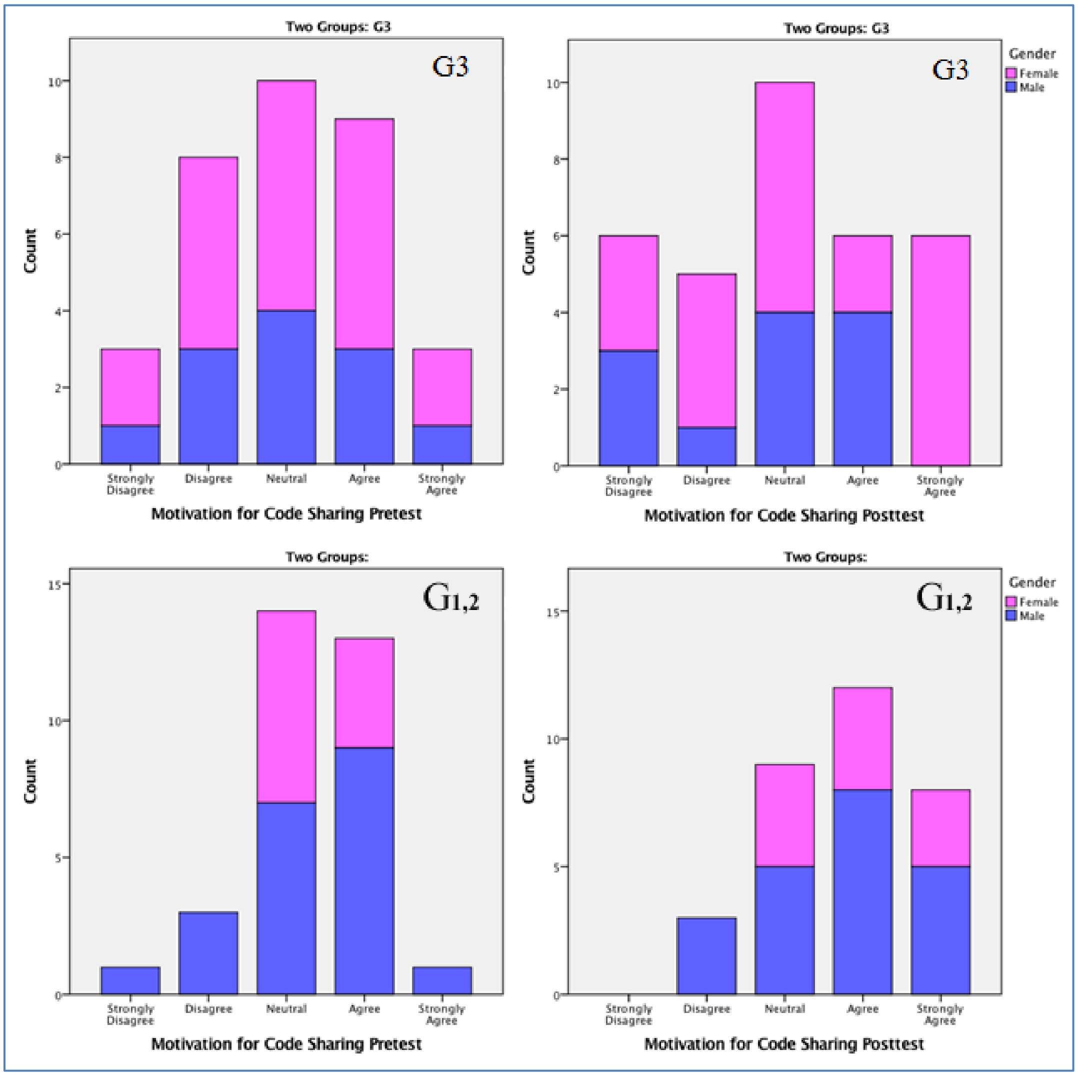


Figure 4-30. G3 vs. $G_{1,2}$ MCS variable.

Table 4-17 shows that the mean values of the MCS variable were higher in $G_{1,2}$ than G3 for all students and for both genders and in both tests, which means that more students in $G_{1,2}$ agreed on the art and animation as a motivation for code sharing. The students of $G_{1,2}$ attended the workshop on their own interest and had more time to write and share code than students who attended G3, who were exposed to a shorter coding workshop as a school activity.

Table 4-17

G3 vs. $G_{1,2}$ Mean values for MCS, IMCS, and EMCS Variables

	G3 vs. $G_{1,2}$	Mean Pretest	Mean Posttest	N
Motivation (MCS)	G3	3.03	3.03	33
	Female	3.05	3.19	21
	Male	3.00	2.75	12
	$G_{1,2}$	3.31	3.78	32
	Female	3.36	3.91	11
	Male	3.29	3.71	21
Intrinsic Motivation (IMCS)	G3	2.9394	2.9394	33
	Female	2.9524	3.0000	21
	Male	2.9167	2.8333	12
	$G_{1,2}$	3.3750	3.8125	32
	Female	3.4545	4.0909	11
	Male	3.3333	3.6667	21
Extrinsic Motivation (EMCS)	G3	3.2424	3.1515	33
	Female	3.3333	3.4286	21
	Male	3.0833	2.6667	12
	$G_{1,2}$	3.3750	3.7188	32
	Female	3.2727	3.8182	11
	Male	3.4286	3.6667	21

Table 4-18 shows that the t-test results for $G_{1,2}$ were significant for all students and for both genders. In other words, $G_{1,2}$ students' agreement percentage increased significantly in the posttest. The results for the intrinsic motivation variable (IMCS) were significant for male and female students in $G_{1,2}$, and the result of the extrinsic motivation was significant for $G_{1,2}$ female students (see Table 4-18).

In G3, the mean values of the female students increased for all three motivation variables (MCS, IMCS, EMCS) in the posttest, but the mean values of the male students slightly decreased in the posttest, or fewer male students in G3 found that art and animation increase their motivation for code sharing. The t-test results for G3 were not significant for all three motivation variables.

Table 4-18
G3 vs. $G_{1,2}$ T-Test Results of MCS, IMCS, and EMCS Variables

	G3 vs. $G_{1,2}$	(Mean Pretest – Mean Posttest)	t	df	Sig. (2-tailed)
MCS	G3	.000	.000	32	1.000
	Female	- .143	-.767	20	.452
	Male	.250	1.149	11	.275
	$G_{1,2}$	-.469	-2.792	31	.009
	Female	-.545	-2.206	10	.052
	Male	-.429	-1.910	20	.071
IMCS	G3	.000	.000	32	1.000
	Female	-.04762	-.237	20	.815
	Male	.08333	.364	11	.723
	$G_{1,2}$	-.43750	-2.441	31	.021
	Female	-.63636	-2.609	10	.026
	Male	-.33333	-1.375	20	.184
EMCS	G3	.09091	.423	32	.675
	Female	-.09524	-.346	20	.733
	Male	.41667	1.239	11	.241
	$G_{1,2}$	-.34375	-1.686	31	.102
	Female	-.54545	-2.631	10	.025
	Male	-.23810	-.815	20	.424

4.6.3 Motivation for code sharing G1, G2, and G3. G1 had a five-day summer camp with 15 coding hours. Students of G1 had more time to explore the tool and try to write more code and share it at home since the camp was in summer with no schoolwork. Students of G2 had less coding time (5 hours). However, their coding time was also in five days so they were able to explore the tool at home and write some code but, with school work and school time, they may have not found the time that students in G1 had.

Table 4-19, Table 4-20, and Figure 4-31 for the MCS variable suggest that G1and G2 students agreed on the effect of art as motivation for code sharing from the pretest, and their choices improved in the posttest.

Table 4-19
G1, G2, and G3 Mean of MCS, IMCS, and EMCS Variables.

		Mean Pretest	Mean Posttest	N
Motivation	G1	3.1667	3.5000	18
	Female	3.43	4.00	7
	Male	3.00	3.18	11
	G2	3.5000	4.1429	14
	Female	3.25	3.75	4
	Male	3.60	4.30	10
	G3	3.0303	3.0303	33
	Female	3.05	3.19	21
	Male	3.00	2.75	12
Intrinsic Motivation	G1	3.1111	3.6111	18
	G2	3.7143	4.0714	14
	G3	2.9394	2.9394	33
Extrinsic Motivation	G1	3.2556	3.3667	18
	G2	3.5429	3.9571	14
	G3	3.2242	3.1576	33

The descriptive statistics also show that the mean values of all groups for the three motivation variables (MCS, IMCS, and EMCS) were above the neutral option in both tests (Table 4-19). This indicates that the majority of students agreed on the questions or they agreed that the use of art and animation with coding increases their motivation for code sharing. For the MCS variable, the mean values for G2 were the highest among the three groups in both tests, and the t-test result was significant ($t(13) = -2.38$, $p = .033$), As shown in Table 4-20. The female students in G1 had a significant result ($t(6) = -2.82$, $p = .03$), and the result for the male students in G2 was also significant ($t(9) = -2.33$, $p = .045$). The mean values of the students' responses to this variable improved in the posttest for both genders in all groups except for the male students in G3, which slightly dropped from 3.00 to 2.75.

For the intrinsic motivation (IMCS), the result was significant for G1 only as the t-test results indicated (t(17) = -2.153, p = .046). No results in the three groups were significant for the extrinsic motivation (EMCS) variable.

Table 4-20
G1, G2, and G3 T-Test Results of MCS, IMCS, and EMCS

		(M Pretest – M Posttest)	t	df	Sig. (2-tailed)
Motivation	G1	-.33333	-1.558	17	.138
	Female	-.571	-2.828	6	.030
	Male	-.182	-.559	10	.588
	G2	-.64286	-2.386	13	.033
	Female	-.500	-.775	3	.495
	Male	-.700	-2.333	9	.045
	G3	.00000	.000	32	1.000
	Female	-.143	-.767	20	.452
	Male	.250	1.149	11	.275
Intrinsic Motivation	G1	-.50000	-2.153	17	.046
	G2	-.35714	-1.235	13	.239
	G3	.00000	.000	32	1.000
Extrinsic Motivation	G1	-.11111	-.561	17	.582
	G2	-.41429	-1.170	13	.263
	G3	.06667	.368	32	.715

The paired analysis t-test was run for each item, and this was used to measure MCS individually to find which item was more effective for each group. The t-test analysis results for all questions of MCS showed that the mean differences between the two tests were significant for the Q2, Q3, Q6, and Q10 for G1, Q10 for G2, and Q7 and Q11 for G3, as shown in Figure 4-32.

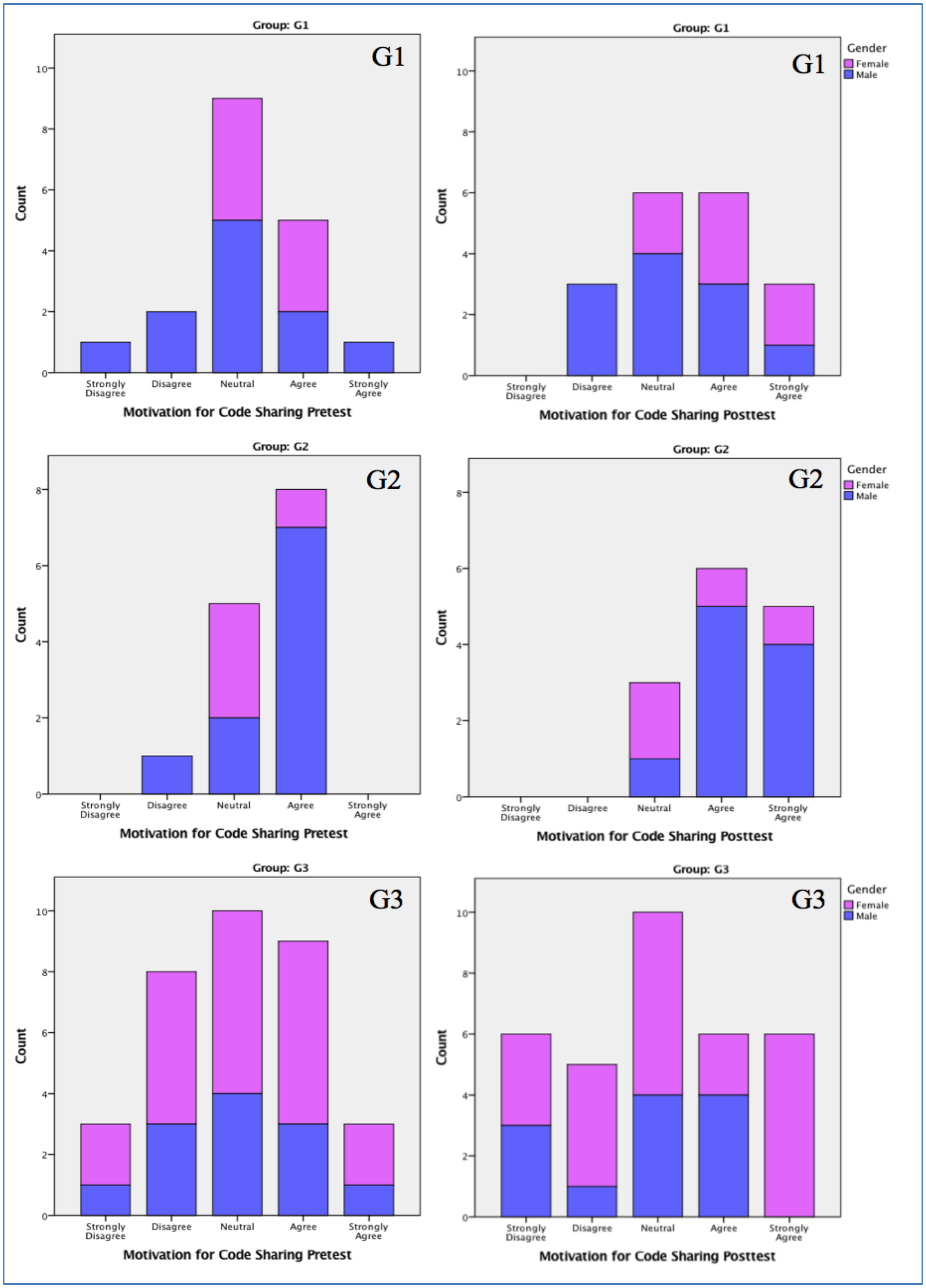


Figure 4-31. G1, G2, and G3 motivation for code sharing in both tests.

G1	Mean Pretest	Mean Posttest	N
Q1	3.61	3.61	18
Q2	3.06	3.61	18
Q3	2.22	3.61	18
Q4	3.11	3.33	18
Q5	3.50	3.72	18
Q6	3.17	3.56	18
Q7	3.61	3.61	18
Q8	3.44	3.28	18
Q9	3.44	3.56	18
Q10	3.00	3.56	18
Q11	2.78	2.83	18

G2	Mean Pretest	Mean Posttest	N
Q1	3.79	4.36	14
Q2	3.86	3.93	14
Q3	3.79	4.07	14
Q4	3.64	4.07	14
Q5	3.79	4.14	14
Q6	3.50	3.86	14
Q7	3.71	4.21	14
Q8	3.86	4.14	14
Q9	3.79	4.07	14
Q10	3.14	3.79	14
Q11	3.21	3.57	14

G3	Mean Pretest	Mean Posttest	N
Q1	2.91	2.85	33
Q2	2.73	2.88	33
Q3	2.94	2.88	33
Q4	2.79	2.82	33
Q5	2.94	3.09	33
Q6	2.79	2.85	33
Q7	3.70	3.12	33
Q8	3.24	3.09	33
Q9	3.33	3.15	33
Q10	2.94	3.00	33
Q11	2.91	3.42	33

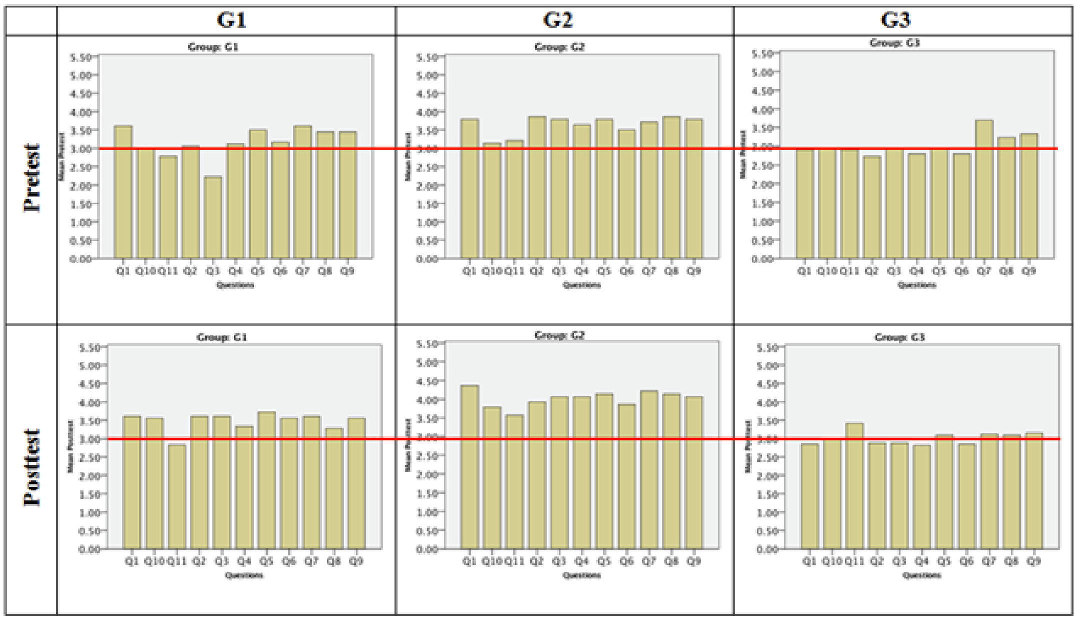


Figure 4-32. The mean values of the mcs questions in pretest and posttest.

4.7 Programming Interest Enjoyment

The Programming Interest and Enjoyment (PIE) variable was also measured in the pre- and post-survey questionnaire using 5-point Likert scales for five survey items. The results for the PIE variable are discussed for $G_{1,2,3}$, G3 vs. $G_{1,2}$, and for the three groups in the following sections.

4.7.1 Programming interest and enjoyment for $G_{1,2,3}$. The results for the PIE variable are shown in Figure 4-33. The agreement percentage of all students in $G_{1,2,3}$ improved from 39% in the pretest to 46% in the posttest while the disagreement percentage 22% remained unchanged, and the *neutral* percentage decreased from 40% to 32% for all students in $G_{1,2,3}$. In the posttest, the female agreement percentage rose from 31% to 38%, and it increased from 46% to 55% for male students.

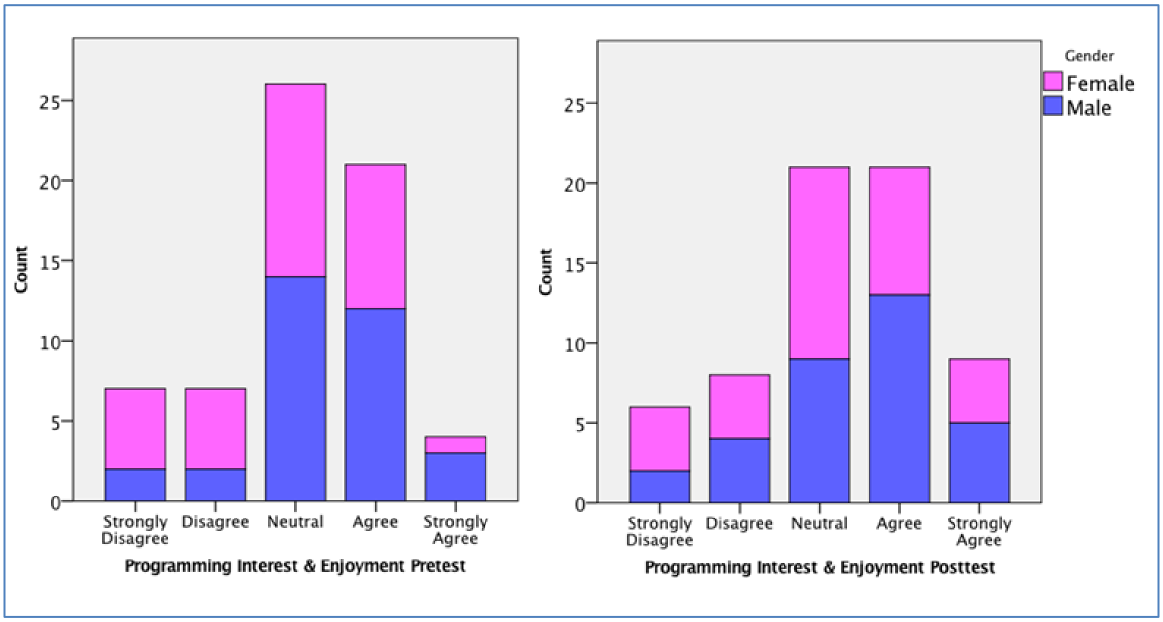


Figure 4-33. $G_{1,2,3}$ programming interest and enjoyment.

The mean values for both tests were above the *neutral* option, meaning most of the students enjoyed and were interested in programming (M = 3.1, M = 3.3). The mean values for the male students were higher than the mean values for the female students in both tests.

The t-test result (Table 4-21) was significant for all students in $G_{1,2}$ ($t(64) = -2.599$, $p = .012$). It was also significant for female students ($t(31) = -3.676$, $p = .001$), but it was not significant for male students because the agreement percentage of the male students, which was high in the pretest, did not increase significantly in the posttest.

In the posttest, the mean value of the Asian group was the highest among other racial groups, followed by the mean values of the White and Asian Indian groups. This indicates that these groups had high interest in programming. No significant results were found for any racial group in this variable.

Table 4-21
$G_{1,2,3}$ T-Test Results for PIE Variable

	Mean PIE Pretest	Mean PIE Posttest	N
$G_{1,2,3}$	3.1141	3.3167	65
Female	2.8516	3.1901	32
Male	3.3687	3.4394	33
American Indian	3.3333	4.4167	1
Asian	3.6974	3.8377	19
Asian Indian	3.4333	3.5917	10
Black	2.4583	2.4028	6
Middle Eastern	2.5708	2.7292	20
Hawaiian	3.0000	3.5833	2
White	3.1905	3.7381	7

	Mean (PIE Pretest – PIE Posttest)	t	df	Sig. (2-tailed)
$G_{1,2,3}$	-.20256	-2.599	64	.012
Female	-.33854	-3.676	31	.001
Male	-.07071	-.580	32	.566
Asian	-.14035	-1.774	18	.093
Asian Indian	-.15833	-.772	9	.460
Black	.05556	.305	5	.773
Middle Eastern	-.15833	-1.032	19	.315
Hawaiian	-.58333	-2.333	1	.258
White	-.54762	-1.355	6	.224

4.7.2 Programming interest and enjoyment for G3 vs. $G_{1,2}$. Figure 4-34 shows the improvement in the programming enjoyment variable in the posttest for both G3 and $G_{1,2}$. Programming interest and enjoyment was higher for $G_{1,2}$ than for G3 students in both tests. The agreement percentage for the PIE variable in $G_{1,2}$ increased from 53% in the pretest to 66% in the posttest, and it increased slightly from 24% to 27% for G3 students. Figure 4-34 and Table 4-22 show the descriptive statistic for the PIE variable in both tests for G3 and $G_{1,2}$.

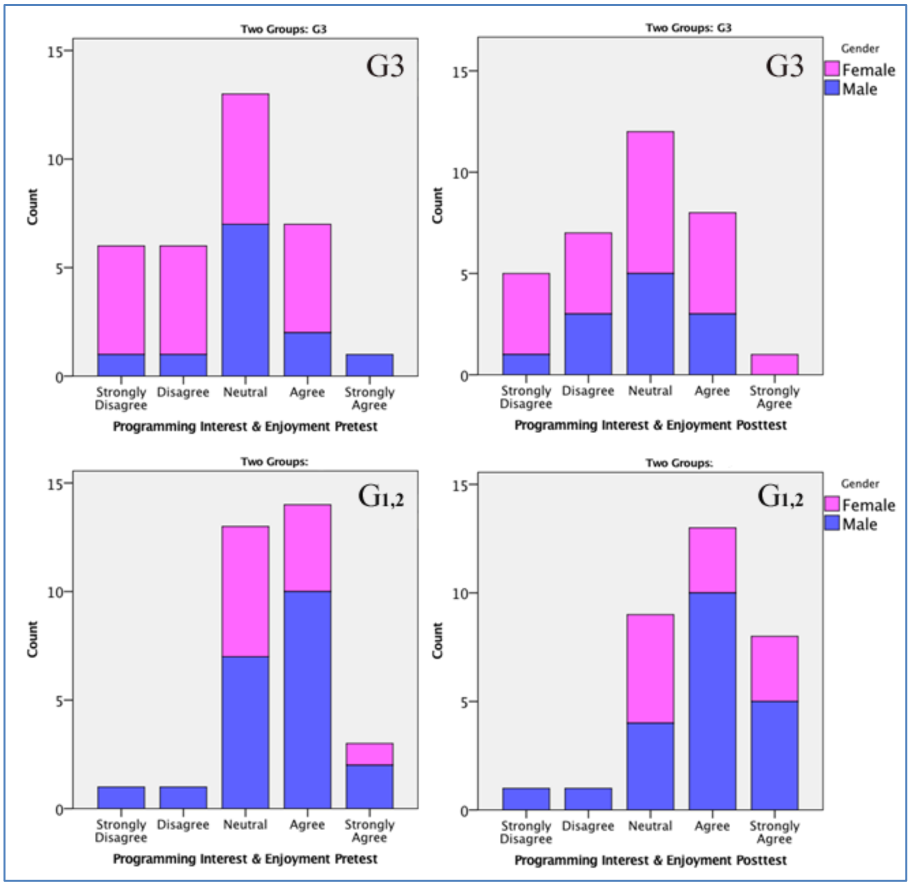


Figure 4-34. G3 vs. $G_{1,2}$ programming interest and enjoyment.

Table 4-22

G3 vs. $G_{1,2}$ Descriptive Statistics for PIE Variable

		Agreement %		Disagreement %		Neutral %	
		Pretest	Posttest	Pretest	Posttest	Pretest	Posttest
G3	All students	24%	27%	37%	37%	39%	36%
	Female	24%	29%	47%	38%	29%	33%
	Male	25%	25%	17%	33%	58%	42%
$G_{1,2}$	All students	53%	66%	6%	6%	41%	28%
	Female	45%	55%	0%	0%	55%	45%
	Male	57%	71%	10%	10%	33%	19%

In G3, the programming interest and enjoyment increased for female students more than the male students in the posttest. The agreement percentage increased from 24% to 29% for the female students, but it remained unchanged for the male students (25%) in the posttest. In $G_{1,2}$, the agreement percentage for the female students rose from 45% to 55%, and it rose from 57% to 71% in the posttest for the male students. Table 4-23 shows that the mean values for the $G_{1,2}$ were higher than the mean values for the G3 in both tests.

Table 4-23
G3 vs. $G_{1,2}$ T-Test Results for PIE Variable

	Mean PIE Pretest	Mean PIE Posttest	N
G3	2.7374	2.8207	33
Female	2.5397	2.8175	21
Male	3.0833	2.8264	12
$G_{1,2}$	3.5026	3.8281	32
Female	3.4470	3.9015	11
Male	3.5317	3.7897	21

	Mean (PIE Pretest – PIE Posttest)	t	df	Sig. (2-tailed)
G3	-.08333	-.750	32	.459
Female	-.27778	-2.195	20	.040
Male	.25694	1.455	11	.174
$G_{1,2}$	-.32552	-3.050	31	.005
Female	-.45455	-3.941	10	.003
Male	-.25794	-1.710	20	.103

The mean value for the male students was higher than the mean value for the female students in the pretest for both groups, and it was the same as the female students' mean for G3 and a little lower than the female students' mean in the posttest for $G_{1,2}$. The t-test results were significant for the female students in both groups, but no significant results were found for male students. In other words, the female students' agreement percentage for the programming interest and enjoyment questions increased significantly while it did not increase significantly for the male students.

4.7.3 Programming interest and enjoyment for G1, G2, and G3. As mentioned, G1 and G2 attended the camp and the workshop upon their own interest. The difference in the PIE variable between the groups appears clearly in Figure 4-35.

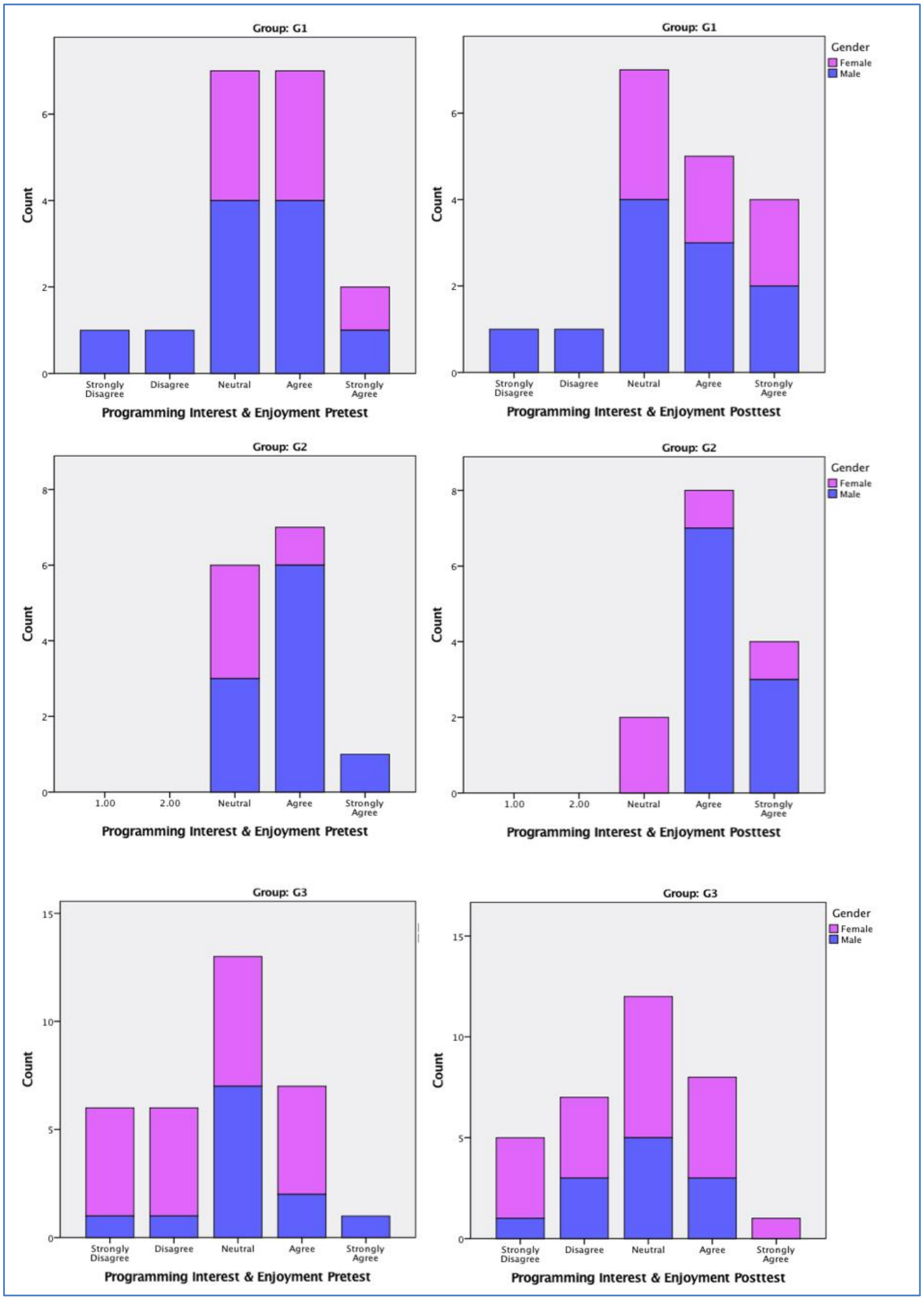


Figure 4-35. G1, G2, and G3 programming interest and enjoyment.

It can be seen that the majority of the students in G1 and G2 indicated their programming enjoyment. In contrast, the responses of the students in G3 were distributed between the agreement, neutral, and disagreement options. The agreement percentage for the

G1 students remained 50% for both tests, but many of the students who chose the *agree* option in the pretest changed to *strongly agree* in the posttest. For G2, the agreement percentage improved from 57% to 85%, and it rose slightly for G3 from 24% to 27% in the posttest.

Table 4-24 shows that the mean values increased for all groups in the posttest. G2 had the highest mean value in the posttest (M = 4.16), and G3 had the lowest mean value (M = 2.8). The mean value improved significantly in the posttest for all students in G2 (t(13) = -2.52, p = .025). The mean value for the female students improved significantly in G1 and G3 with p = .028 and p = .04, respectively.

Table 4-24
T-Test Results for PIE Variable for G1, G2, and G3

	Mean PIE Pretest	Mean PIE Posttest		N
G1	3.3750	3.5694		18
Female	3.5952	3.9524		7
Male	3.2348	3.3258		11
G2	3.6667	4.1607		14
Female	3.1875	3.8125		4
Male	3.8583	4.3000		10
G3	2.7374	2.8207		33
Female	2.5397	2.8175		21
Male	3.0833	2.8264		12
	Mean (PIE Pretest – PIE Posttest)	t	df	Sig. (2-tailed)
G1	-.19444	-1.790	17	.091
Female	-.35714	-2.873	6	.028
Male	-.09091	-.585	10	.572
G2	-.49405	-2.523	13	.025
Female	-.62500	-2.724	3	.072
Male	-.44167	-1.676	9	.128
G3	-.08333	-.750	32	.459
Female	-.27778	-2.195	20	.040
Male	.25694	1.455	11	.174

4.8 Interest in CS Courses

Morgan and Klaric's (2007) study states that students who had a CS course in high school are more likely to major in CS in university. To measure the Course Interest (CI) variable, all students were asked about their interest in taking a CS course in their schools in both pretest and posttest, specifically before and after having the art with code workshop as a treatment in this experimental study. The CS course interest variable was measured with 5-point Likert scales: 1 for *strongly disagree* and 5 for *strongly agree*.

4.8.1 Interest in CS courses for $G_{1,2,3}$. Figure 4-36 shows the difference between the students' responses in the pre- and the posttests for the CI variable. The agreement percentage for all students in $G_{1,2,3}$ for this variable increased slightly in the posttests, rising from 31% in the pretest to 34% in the posttest. The disagreement dropped from 43% to 31%, and the neutral percentage increased from 26% to 35% in the posttest. The male students' agreement percentages were higher than the female students' percentages in both tests. In the pretest, 55% of the male students were interested in taking a CS course in high school, yet the percentage for the female students was only 6%. The female students' agreement percentage improved to 16%, and it slightly dropped for the male students to 52% in the posttest. The disagreement percentage dropped for both genders in the posttest from 66% to 50% for female students and from 21% to 12% for male students. The percentage for the *neutral* option increased for both genders in the posttest; it increased from 28% to 34% for females and from 24% to 36% for male students.

The interest in a CS course among male students (M = 3.55) was higher than the interest among the female students (M = 1.97) in both tests, but, in the posttest, the improvement in a CS course interest was significant for female students (M = 2.44) and not significant for male students (M = 3.64). In other words, the relatively high interest of the male students did not change much in the posttest, but, after the treatment, some female students changed their minds, and their interest in taking a CS course improved in the posttest.

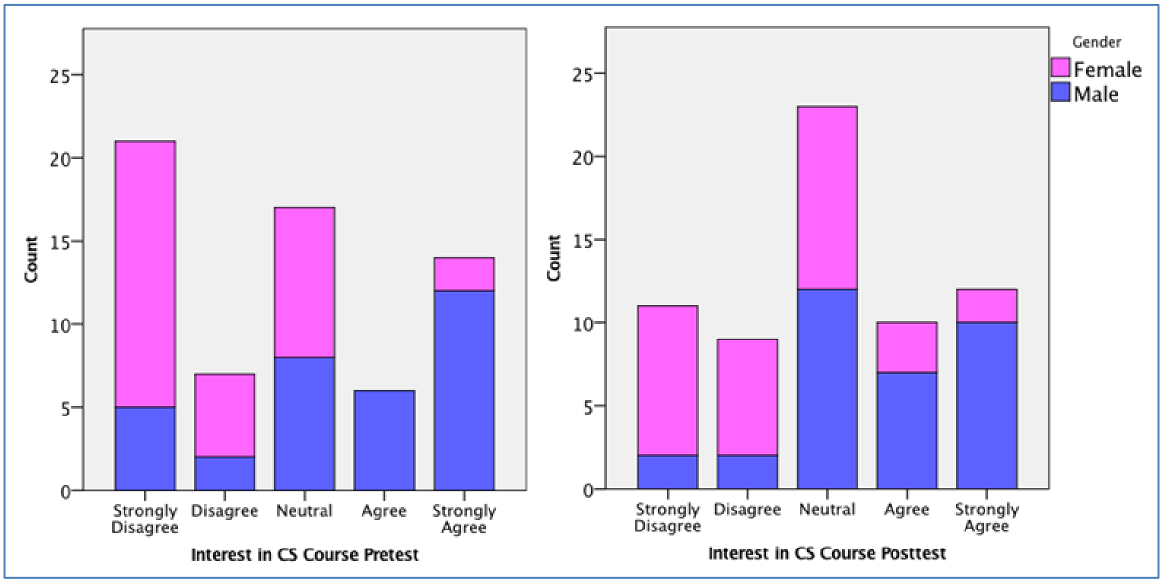


Figure 4-36. $G_{1,2,3}$ interest in CS courses.

The results of paired samples t-test analysis (Table 4-25) agreed with significant increment in the female interest in taking CS course in high school. The result was significant for the female students ($t(31) = -2.462$, $p = .020$), but it was not significant for the male students ($t(32) = -.392$, $p = .697$). The total result for all students approached the border of significance ($t(64) = -1.832$, $p = .072$).

The mean value for the White racial group was the highest among other groups in both tests, followed by the mean values of the Asian Indian then Asian group. The highest increment in the posttest was for the Middle Eastern group ($MD = -.45$). The mean values for the CS course interest increased in the posttest for most of the racial groups; however, no significant results were found for any racial group.

Table 4-25
$G_{1,2,3}$ T-Test Results for the CS Course Interest

	Mean CI Pretest	Mean CI Posttest	N
$G_{1,2,3}$	2.77	3.05	65
Female	1.97	2.44	32
Male	3.55	3.64	33
American Indian	4.00	4.00	1
Asian	2.89	3.16	19
Asian Indian	3.00	3.20	10
Black	2.17	2.50	6
Middle Eastern	2.20	2.65	20
Hawaiian	3.00	2.00	2
White	4.00	4.29	7

	Mean (CI Pretest – CI Posttest)	t	df	Sig. (2-tailed)
$G_{1,2,3}$	-.277	-1.832	64	.072
Female	-.469	-2.462	31	.020
Male	-.091	-.392	32	.697
Asian	-.263	-.925	18	.367
Asian Indian	-.200	-.802	9	.443
Black	-.333	-1.000	5	.363
Middle Eastern	-.450	-1.229	19	.234
Hawaiian	1.000	1.000	1	.500
White	-.286	-1.549	6	.172

4.8.2 Interest in CS courses for G3 vs. $G_{1,2}$. In $G_{1,2,}$ the students' interest in taking a CS course was higher than the interest in G3, but it increased for both groups in the posttest. The agreement percentages increased from 15% to 18% for G3 and from 47% to 50% for $G_{1,2}$. The *neutral* option percentage increased from 24% to 39% for G3 and from 28% to 31% for $G_{1,2}$. The disagreement percentage dropped from 61% to 43% for G3 and from 25% to 19% for $G_{1,2}$. Figure 4-37 shows the descriptive statistics for both groups.

In G3, the agreement percentage of the female students improved from 0% to 10% in the posttest. The agreement percentage of the male students, which was much higher than the female students in the pretest (42%), dropped in the posttest to 33%, but their disagreement percentage remained unchanged, 25% in both tests, and their neutral percentage increased from 33% to 42%. In $G_{1,2}$, the female students' agreement percentage increased from 18% to 27% in the posttest, but the male students' agreement percentage, which was already high in the pretest (62%), remained unchanged.

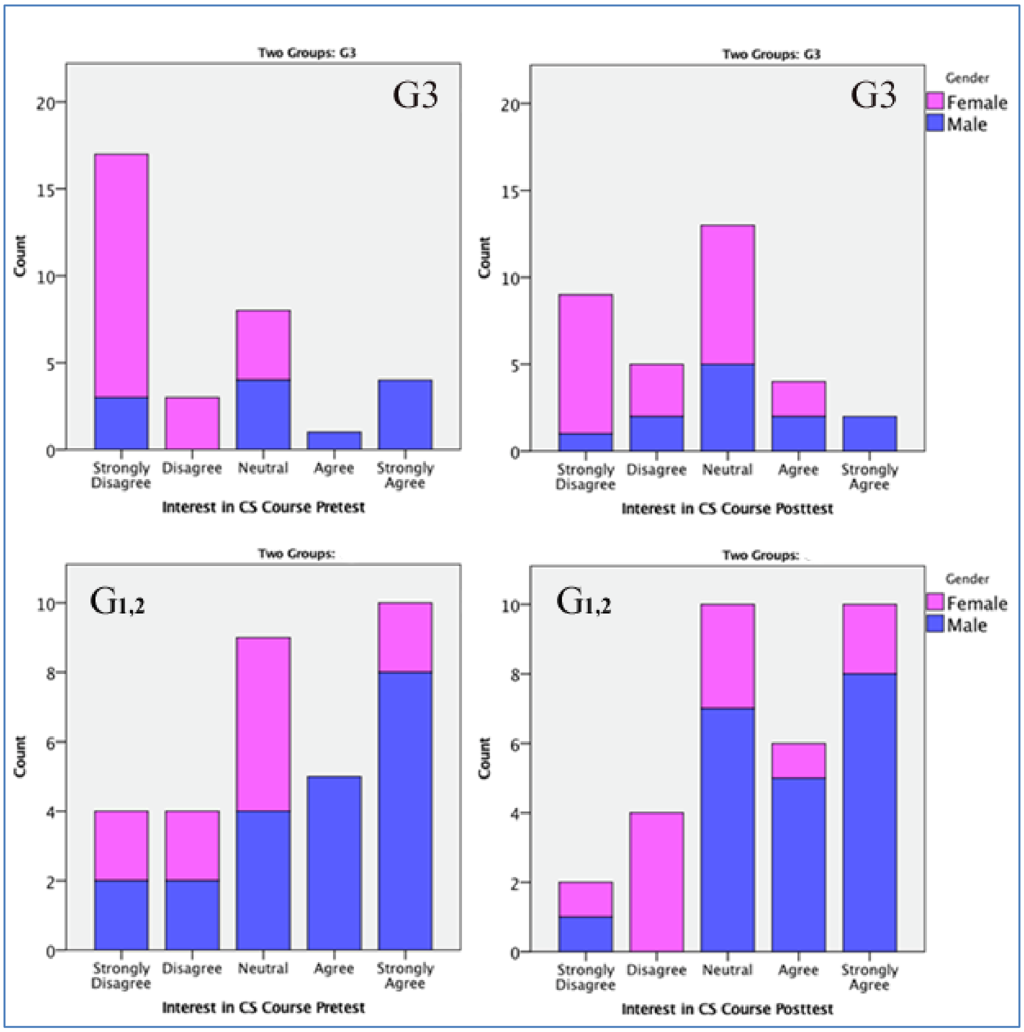


Figure 4-37. G3 vs. $G_{1,2}$ students' interest in CS courses in high school.

Table 4-26 shows that the mean value for $G_{1,2}$ in the posttest (M = 3.56) was higher than the mean value of G3 (M = 2.55). The mean value for the male students of $G_{1,2}$ in the posttest was the highest (M = 3.9), followed by the mean value of the male students in G3 (M = 3.17). The male students were more interested in taking a CS course in the school than the female students. However, the t-test result was significant for the female students of G3 ($t(20) = -2.87$, $p = .009$), which indicates a significant increase in interest in taking a CS course.

Table 4-26

G3 vs. $G_{1,2}$ T-Test Results for the CS Course Interest

	Mean CI Pretest	Mean CI Posttest		N
G3	2.15	2.55		33
Female	1.52	2.19		21
Male	3.25	3.17		12
$G_{1,2}$	3.41	3.56		32
Female	2.82	2.91		11
Male	3.71	3.90		21
	Mean (CI Pretest – CI Posttest)	t	df	Sig. (2-tailed)
G3	-.394	-1.683	32	.102
Female	-.667	-2.870	20	.009
Male	.083	.172	11	.866
$G_{1,2}$	-.156	-.818	31	.420
Female	-.091	-.289	10	.779
Male	-.190	-.777	20	.446

4.8.3 Interest in CS courses for G1, G2, and G3. The percentage of students who chose *disagree* and *strongly disagree* dropped from 39% in the pretest to 22% in G1. It also dropped from 61% to 42% in G3, but it rose from 7% to 14% in G2. However, most of the students in the three groups either agreed or were *neutral* but had not disagreed to take a CS course in high school.

Table 4-27 shows the results of the three groups. The result was $t(17) = -2.062$, $p = .055$ for G1, $t(13) = .186$, $p = .856$ for G2, and $t(32) = -1.683$, $p = .102$ for G3. The results were significant for G1 and not significant for G2 and G3. In other words, the students' interest in taking a CS course increased significantly for G1, and it did not change significantly for G2 and G3. However, Figure 4-38 shows a decrement in the number of students who chose the *disagree* and the *strongly disagree* options in G3.

The mean value of G3 was the highest among all of the groups (M = 3.93). The male students of G2 had the highest mean for this variable in both tests (M = 4.2, M = 4.4). The mean values of the students' responses improved for both genders in all three groups, except for the male students of G3, which slightly dropped from 3.25 to 3.17 in the posttest. Despite that drop, the mean value of the male students' responses in G3 was still higher than the female students' mean in the same group (M = 2.19). However, the result for the female students in G3 was significant ($t(20) = -2870$, $p = .009$).

Table 4-27

G1, G2, and G3 T-Test Results for the CS Course Interest

	Mean CI Pretest	Mean CI Posttest	N
G1	2.94	3.28	18
Female	2.43	3.00	7
Male	3.27	3.45	11
G2	4.00	3.93	14
Female	3.50	2.75	4
Male	4.20	4.40	10
G3	2.15	2.55	33
Female	1.52	2.19	21
Male	3.25	3.17	12

	Mean (CI Pretest – CI Posttest)	t	df	Sig. (2-tailed)
G1	-.333	-2.062	17	.055
Female	-.571	-1.922	6	.103
Male	-.182	-1.000	10	.341
G2	.071	.186	13	.856
Female	.750	1.567	3	.215
Male	-.200	-.408	9	.693
G3	-.394	-1.683	32	.102
Female	-.667	-2.870	20	.009
Male	.083	.172	11	.866

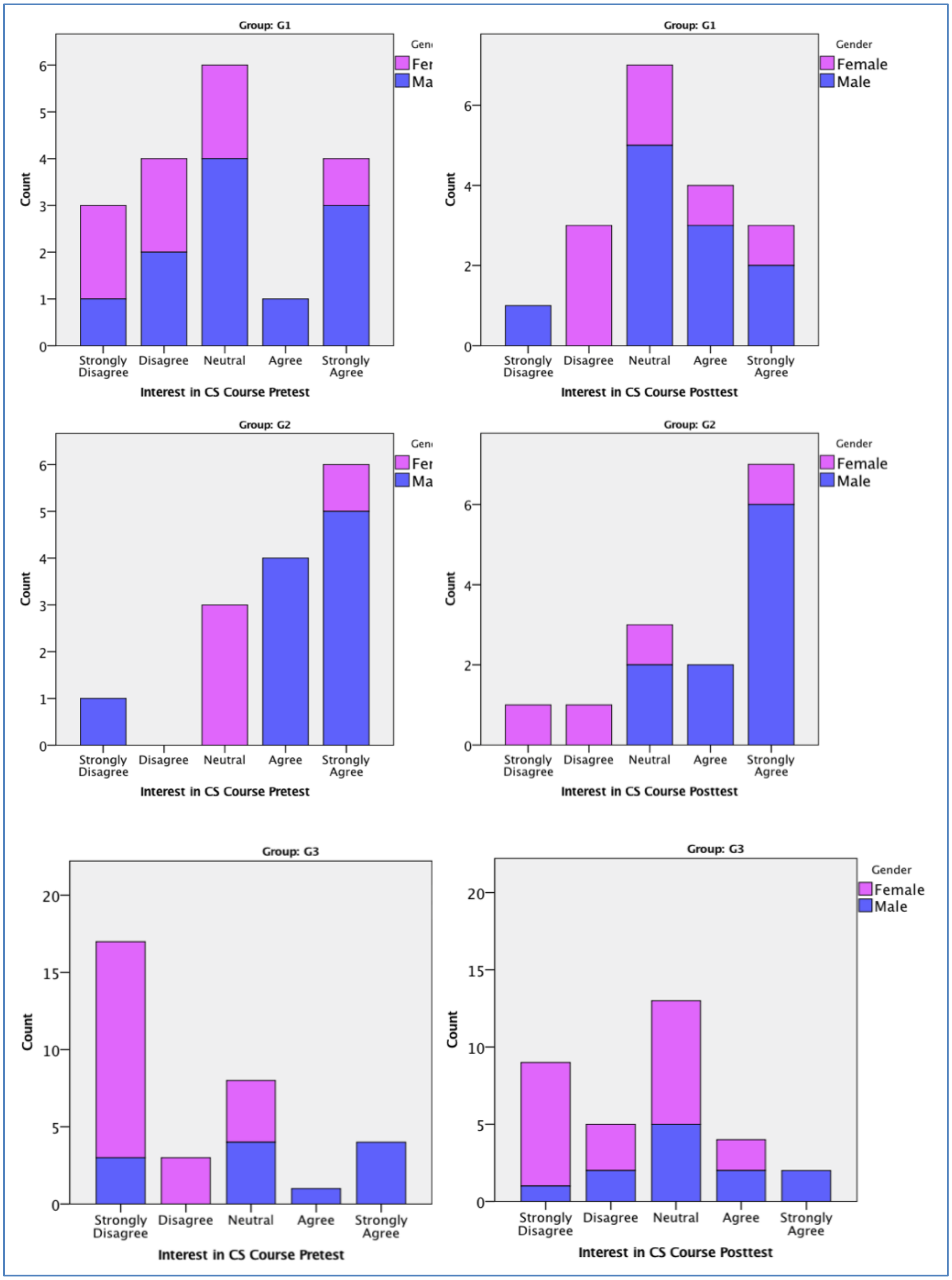


Figure 4-38. G1, G2, and G3 interest in CS courses.

4.9 Art and Animation Usefulness

Unlike the previous six variables, only posttest survey items were used to measure the Art and Animation Usefulness (AAU) variable. Six survey items were used to measure the

AAU variable in learning math functions and increasing students' creativity and programming skills.

4.9.1 Art and animation usefulness for $G_{1,2,3}$. Figure 4-39 shows the students' responses to the AAU in learning programming. In the posttest, the agreement percentage of all students was 50% (31% for *agree* and 19% for *strongly agree*). The disagreement percentage was 16%, and the neutral percentage was 34%. This indicates that most of the students agreed that art and animation were useful in learning programming, and only 16% of the students disagreed in the posttest. The agreement percentage for the female students was 53%, while their disagreement and neutral percentages were16% and 31%, respectively. For the male students, the agreement percentage was 46%, their disagreement percentage was 18%, and their neutral percentage was 36% in the posttest.

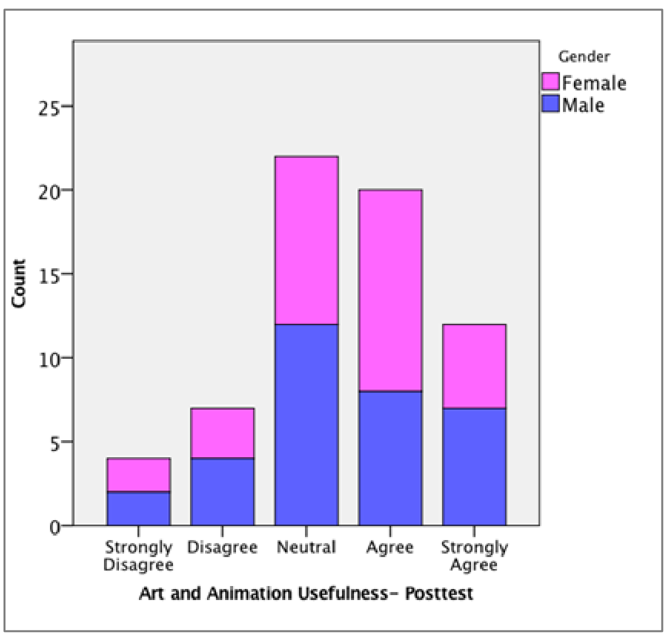


Figure 4-39. $G_{1,2,3}$ students' responses to the art and animation usefulness.

As shown in Table 4-28, the mean value for the female students (M = 3.38) was higher than the mean value for the male students (M = 3.32). The mean value of the Asian students (M = 3.84) and their agreement percentage (79%) were higher than the mean values and the agreement percentages of other racial groups. The White students' mean value was the next highest mean (M = 3.76) as was their agreement percentage (71%).

Table 4-28
$G_{1,2,3}$ Posttest Results for the AAU Variable

	Agreement %	Neutral %	Disagreement %	Mean AAU	N
$G_{1,2,3}$	50%	34%	16%	3.3538	65
Female	53%	31%	16%	3.3802	32
Male	46%	36%	18%	3.3283	33
American Indian	100%	0%	0%	4.0000	1
Asian	79%	21%	0%	3.8421	19
Asian Indian	50%	50%	0%	3.6500	10
Black	17%	50%	34%	2.4722	6
Middle Eastern	20%	45%	35%	2.8083	20
Hawaiian	50%	50%	0%	3.5833	2
White	71%	0%	28%	3.7619	7

4.9.2 Art and animation usefulness for G3 vs. $G_{1,2}$. In comparison between G3 and $G_{1,2}$ for the AAU variable, the results show that the agreement percentage in $G_{1,2}$ (72%) was higher than in G3 (27%). However, that difference does not necessarily indicate that all students in G3 did not find the art and animation useful because their neutral percentage was 46%. The disagreement percentage of $G_{1,2}$ was 6%, and it was 27% in G3.

The agreement percentage for the female students in G3 (38%) was higher than for the male students, which was only 8%. The neutral percentage for the female students was 38%, and their disagreement percentage was 24%. The neutral percentage of the male students was 58%, and their disagreement percentage was 33%.

In $G_{1,2}$, both male and female students agreed on the usefulness of art and animation in learning math functions and increasing students' creativity and programming skills. For the female students in $G_{1,2}$, the agreement percentage was 82%, which was the highest percentage among the genders in both groups. For the male students, it was 67%. The disagreement percentage was zero for the female students, and it was 10% for the male students in $G_{1,2}$. Figure 4-40 shows the descriptive statistics for this variable for both groups.

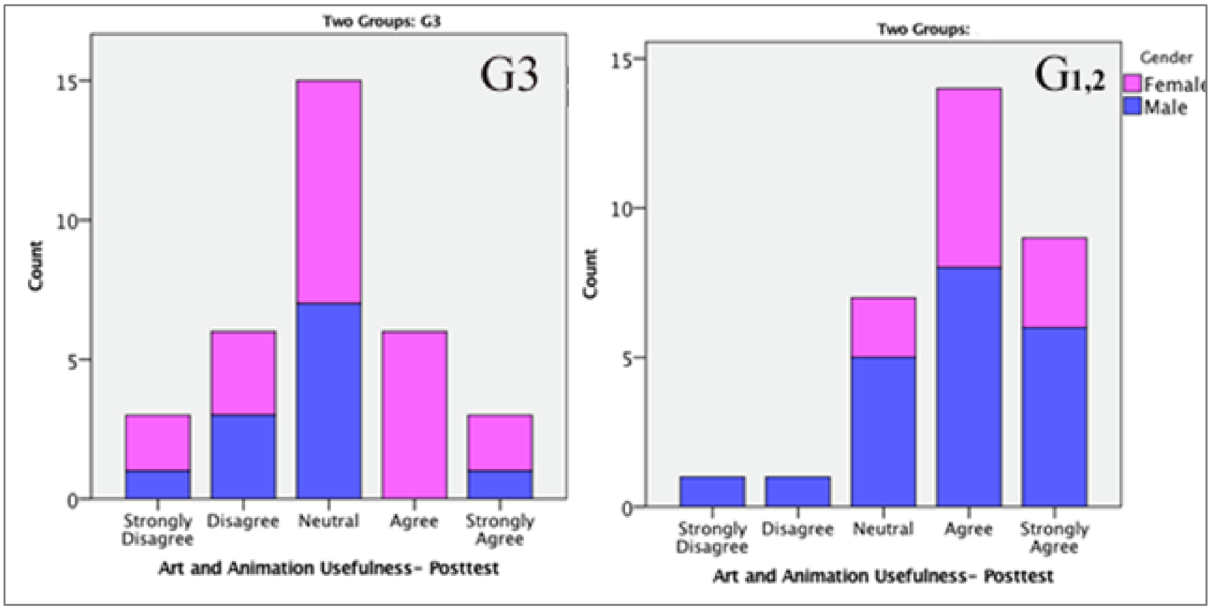


Figure 4-40. G3 vs. $G_{1,2}$ art and animation usefulness.

Table 4-29 shows that the mean value of the female students in G3 (M = 3.10) was higher than the male students' mean value (M = 2.5) in the posttest. The mean value for the $G_{1,2}$ (M = 3.8) was higher than the mean value of G3 (M = 2.9). The mean value of the female students in $G_{1,2}$ (M = 3.9) was slightly higher than the mean for male students in that group (M = 3.75).

Table 4-29
G3 vs. $G_{1,2}$ Posttest Results for the AAU Variable

	Agreement %	Neutral %	Disagreement %	Mean AAU	N
G3	27%	46%	27%	2.9141	33
Female	38%	38%	24%	3.1032	21
Male	8%	58%	33%	2.5833	12
$G_{1,2}$	72%	22%	6%	3.8073	32
Female	82%	18%	0%	3.9091	11
Male	67%	24%	10%	3.7540	21

The descriptive statistics of $G_{1,2}$ shows that most of the students agreed on items that were used to measure the AAU variable. The female students' agreement was higher than the male students' agreement in both groups G3 and $G_{1,2}$.

4.9.3 Art and animation usefulness for G1, G2, and G3. Table 4-30 shows that the agreement percentage was 61% for G1, 27% for G3, and 86% for G2, which was the highest among the three groups. The male students in G2 had the highest agreement percentage (90%) for both genders among the three groups. The female students' agreement percentage was higher than that for male students in G1 and G3.

Table 4-30
G1, G2, and G3 Posttest Results for the AAU Variable

	Agreement %	Neutral %	Disagreement %	Mean AAU	N
G1	61%	28%	11%	3.6111	18
Female	86%	14%	0%	3.9048	7
Male	46%	36%	18%	3.4242	11
G2	86%	14%	0%	4.0595	14
Female	75%	25%	0%	3.9167	4
Male	90%	10%	0%	4.1167	10
G3	27%	46%	27%	2.9141	33
Female	38%	38%	24%	3.1032	21
Male	8%	58%	33%	2.5833	12

The female students in G1 and G2 shared the same high mean value (M = 3.9). The mean value for G2 (M = 4.0) was the highest among all of the groups, while G3 had the lowest mean value for this variable (M = 2.9).

The agreement percentages for G1 and G2 were more than 50%, indicating that most of the students in these two groups agreed on the usefulness of art and animation in programming learning. Figure 4-41 shows the results for the AAU variable in the three groups.

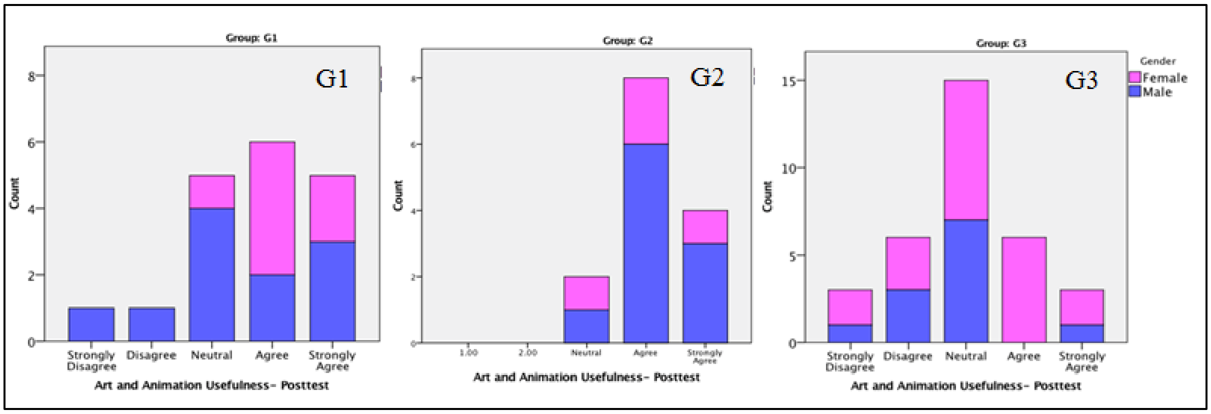


Figure 4-41. G1, G2, and G3 art and animation usefulness.

4.10 The TPB Factors and Interest in CS Degree

To explore the relationship between the factors suggested by the theory of planned behavior (TPB) and the students' interest in pursuing a degree in CS, students responded to 19 survey items in the posttest: seven items for the Programming Benefits and Enjoyment (PBE) factor, four items for the Social Norm (SN) factor, and eight items for the Programming Capabilities and Confidence (PCC) factor. The average for each factor was calculated before running the analysis. The CS degree interest variable was calculated again after removing the items that contained the art and animation elements and one item that related to the SN variable. The average for the resulting variable was calculated and named CS Degree Interest (CSDI).

The Correlation analysis was run between the three factors and the CSDI variable discussed earlier. The results were significant for the three variables or factors, as shown in Table 4-31.

The Pearson correlation coefficient between the CSDI and the PBE was $r=.720$, $p = .000$, which indicates a positive strong relationship. The relationship was also strong and positive between the other two factors and CSDI, $r=.581$, $p = .000$ for SN and $r=.700$, $p = .000$ for PCC.

Table 4-31
Correlation Results of CSDI and the PBE, SN, and PCC Variables

Variable	CS Degree Interest (CSDI) Pearson Correlation (r)	Sig. (2-tailed)
Programming Benefits and Enjoyment (PBE)	.720**	.000
Social Norm (SN)	.581**	.000
Programming Capabilities and Confidence (PCC)	.700**	.000
** Significant		

In other words, if a student enjoys programming and sees benefit in a CS degree, his/her interest in pursuing a CS degree will increase. Similarly, if people around a student have a CS degree or encourage it, his/her interest will increase. Finally, if a student has confidence in his/her capabilities, accepts challenges, and thinks he/she is capable of overcoming the difficulties, then this will increase interest in pursuing a CS degree. The correlation between the PBE and CSDI was the strongest among other variables.

A simple linear regression was calculated to predict students' interest in a CS degree based on the three factors (PBE, SN, and PCC). A stepwise method was selected to include the significant predictors only (SPSS Stepwise Regression, 2017). The results showed two models. Table 4-32 shows the regression analysis results.

The first model includes only one predictor, which is PBE. A significant regression equation was found (f (1, 63) = 67.874, p < .000) with an R2 of .519. The constant coefficient is not significant, and students' interest in a CS degree was equal to -.182 + .898 PBE. Students' interest in a CS degree increased by .898 points for each point on the 5-point Likert scale of the students' PBE.

The second regression model included two predictors, PBE and PCC. A significant regression equation was found (f (2, 62) = 38.409, p < .000) with an R2 of .553. Students' interest in a CS degree was equal to -.211 + .558 PBE + .389 PCC. Students' interest in a CS degree increased by .558 points for each point on the 5-point Likert scale of the students'

PBE. It also increased by .389 points for each point of PCC. SN was excluded from the two models.

Table 4-32
Regression Results DI and the PBE, SN, and PCC Variables

	Model	B Coefficient	Standardized Coefficients Beta	t	Sig.
1	Constant	-.182		-.458	.649
	PBE	.898	.720	8.239	.000
2	Constant	-.211		-.547	.587
	PBE	.558	.447	2.971	.004
	PCC	.389	.331	2.197	.032

a. Dependent Variable: CSDI

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	51.682	1	51.682	67.874	.000[b]
	Residual	47.971	63	.761		
	Total	99.654	64			
2	Regression	55.146	2	27.573	38.409	.000[c]
	Residual	44.508	62	.718		
	Total	99.654	64			

a. Dependent Variable: CSDI , b. Predictors: (Constant), PBE, c. Predictors: (Constant), PBE, PCC

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.720[a]	.519	.511	.87261
2	.744[b]	.553	.539	.84727

a. Predictors: (Constant), PBE, b. Predictors: (Constant) PBE, PCC

4.11 Code Genie User Experience (UX)

The technology acceptance model (TAM) framework defines two variables as the psychological factors that affect technology acceptance; these are ease of use (EOU) and usefulness (Davis, Bagozzi, & Warshaw, 1989). A similar concept is found in the usability rule in the human-computer interface (HCI) field where user experience (UX) designers perform a usability test to measure the user's satisfaction (Shneiderman, Plaisant, Cohen, Jacobs, & Elmqvist, 2016). These two variables were applied to measure the tool's usability.

The development environment that was used in this experimental study included several features or components, such as the use of art, animation, sharing of code and produced artwork, and rating the artwork with a "like" button. In the posttest, students were asked about the usefulness of the treatment components to measure the Usefulness and EOU of the suggested environment. The following survey items were used to measure the variables:

- I think using the Code Genie development environment in learning programming is easy, and I didn't face any difficulty using it.
- I think the Code Genie development environment is a very useful tool in learning programming language.
- The Code Genie development environment has several features. Which of the following features is more useful in learning programming? (*art, animation, sharing, all of the above, none*)

Figure 4-42 shows the students' responses to the first two questions that were used to measure the tool's EOU and Usefulness. Table 4-33 shows descriptive statistics of these two variables. The results showed that the agreement percentage for the tool's EOU was 43% among all students, which was more than double the disagreement percentage (20%). Most of the students found the tool easy to use. The agreement percentage for the female students (44%) was slightly higher that the agreement percentage for the male students (42%). However, the disagreement percentage of the females was also higher (22%) than for the male students (18%).

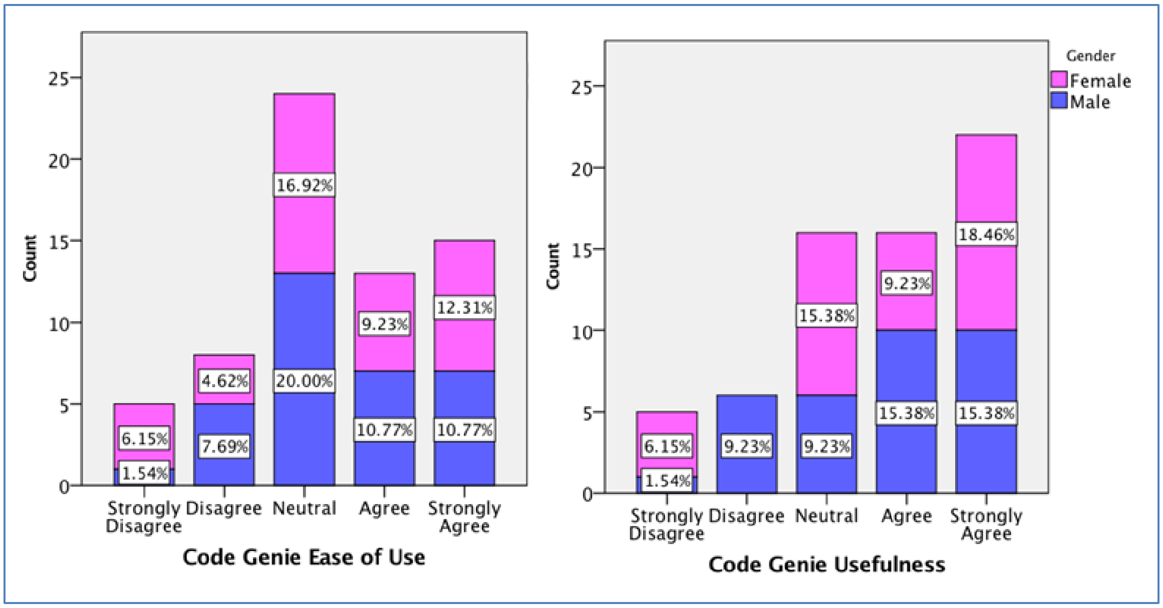


Figure 4-42. Code Genie EOU and Usefulness.

The students' responses to the usefulness variable were a little higher than for the EOU. The results showed that the agreement percentage for the tool's usefulness was 58% among all students, which was more than three times the disagreement percentage (17%). In other words, the majority of the students found the tool useful as a learning environment. The agreement percentage for the male students (61%) was slightly higher than the agreement percentage for the female students (56%). However, the disagreement percentage of the male students was also higher (21%) than that of the female students (13%).

Table 4-33
Students Responses to the EOU and Usefulness Variables

	Responses	All students	Female	Male
EOU *M* = 3.38	Agreement	43%	44%	42%
	Neutral	37%	34%	40%
	Disagreement	20%	22%	18%
Usefulness *M* = 3.68	Agreement	58%	56%	61%
	Neutral	25%	31%	18%
	Disagreement	17%	13%	21%

The mean value for usefulness (M = 3.68) was slightly higher than the mean value for the EOU variable (M = 3.38). The mean values for each group of the three workshops were also calculated, as shown in Table 4-34.

Table 4-34
Mean Values of the EOU and Usefulness for the Three Groups

		G1	G2	G3
EOU	All Students	3.44	3.86	3.15
	Female	3.57	3.50	3.24
	Male	3.36	4.00	3.00
Usefulness	All Students	4.11	3.86	3.36
	Female	4.14	4.00	3.48
	Male	4.09	3.80	3.17

For the EOU variable, the male students of G2 had the highest mean value (M = 4.00). The mean values of the female students (G1: M = 3.57, G3: M = 3.24) were higher than the values of the male students for G1 (M = 3.36) and G3 (M = 3.00). G2 had the highest mean values among the three groups. For the Usefulness variable, the mean value of the female students in G1 was the highest (M = 4.14). Also, the mean values of the females were higher than the mean values of the male students in the three groups. This may imply that the females found the tool more useful the longer they had to work with the tool.

Most of the students found usefulness in the development tool, except for two female students, one Black and one Middle Eastern in G3, who chose the *none* option. These two responses may be attributable to the fact that G3 had a three-hour coding workshop in one day. Other groups had more coding time in five-day workshops, so students had more time to explore the development environment in G1 and G2.

In the third survey item, students were also asked to indicate their favorite element in the tool. They were asked to choose between *art, animation, sharing, all of the above,* and *none.* In response to this question, 51% or most of the students chose the *all of the above* choice, 25% chose *animation*, 17% *art*, 5% *sharing*, and 3% chose the *none* option.

For the gender preference (see Figure 4-43), 50% of the female students chose *all of the above*, 22% *art*, 16% *animation*, 6% *sharing*, and 6% *none*. Male students found animation more useful than art, but some female students preferred art over animation. Of the male students, 52% chose the *all of the above* option, 33% chose *animation*, 12% *art*, 3% *sharing*, and 0% the *none* option.

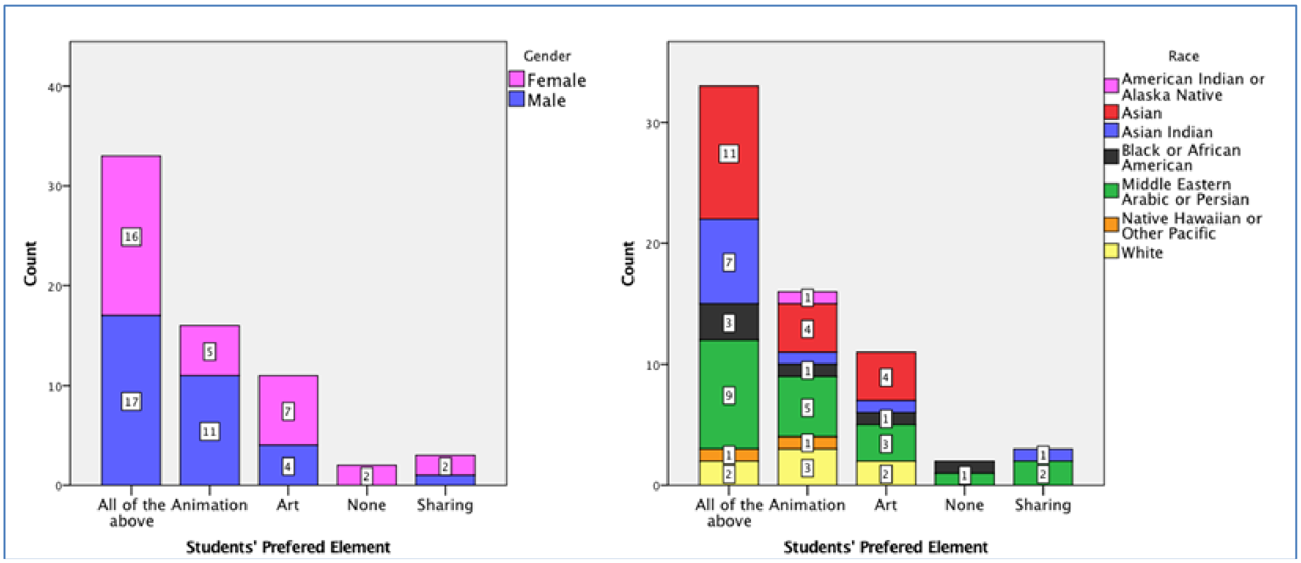


Figure 4-43. Students preferred element classified by gender, and race.

The American Indian student chose the animation. For the Asian group, 58% chose *all of the above*, 21% *animation*, and 21% *art*. For the Asian Indian students, 70% chose *all of the above*, and *art*, *animation*, and *sharing* each received 10% of the responses. Of the Black students, 50% chose *all of the above*, and 17% preferred *art*, 17% *animation*, and 17% chose *none*. Of the Middle Eastern group, 45% chose the *all of the above* option, 25% *animation*, 15% *art*, 10% *sharing*, and 5% chose *none*. For the two Hawaiian students, one chose the *all of the above* option, and one chose *animation*. Of the White students, 29% chose *all of the above*, 43% *animation*, and 29% *sharing*.

Figure 4-44 and Table 4-35 show the descriptive statistics for the preferred element among the three groups.

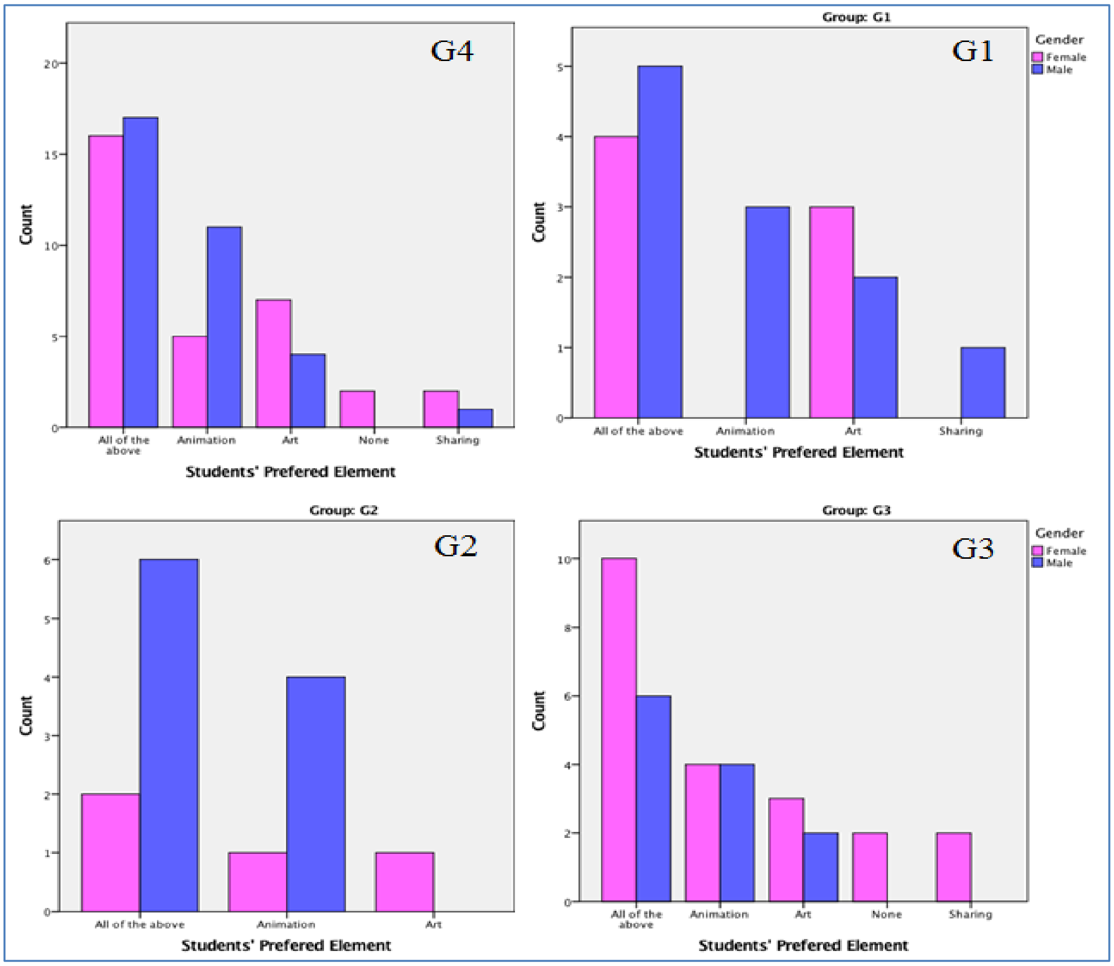


Figure 4-44. Students preferred element classified by coding time.

Table 4-35
Students' Preferred Element in the Tool

	Frequency Percent %			
	$G_{1,2,3}$	G1 (N=18)	G2 (N=14)	G3 (N=33)
Art	17%	28%	7%	15%
Animation	25%	27%	36%	24%
Sharing	5%	6%	0%	6%
All of the above	51%	50%	57%	49%
None of the above	3%	0%	0%	6%
Total	100%	100%	100%	100%

Only two students in G3 chose the *none* option. No student in G2 chose the *sharing* option. However, *sharing* is included in the *all of the above* option, which was selected by the majority in all three groups.

4.12 Programming Knowledge Sub-Variables

This section sheds light on the programming questions that were used to measure students' programming knowledge. The students' responses to these questions are discussed in detail in the following subsections.

The programming knowledge variables PV1 to PV12 are the sub-variables in this study. There were no hypotheses to be accepted for those variables. However, the following sections demonstrate the descriptive statistics for each programming variable among the three groups and the results analysis in the pretest and the posttest for those variables. In the last two subsections, the scores for individual students in each group and the relationship between the coding time and programming knowledge are discussed.

4.12.1 Programming variables PV1, PV2, and PV3. Table 4-36 shows the questions used to measure students' understanding of the variable assignment (PV1), variable addition (PV2), and variable multiplication (PV3).

Table 4-36
PK Questions for Variables PV1, PV2, and PV3

Variables	Code	Questions

1. Variable Assignment (PV1)
 	
2. Variable Addition (PV2)
 	
3. Variable Multiplication (PV3)
	Assume the following block of code for the following three questions
	
	`n1=3;`
	`n2=8;`
	`n3=n1*n2;`
	`n2=n1+n2;`
		(5- P2Q1)
		After the above code, n1contains {1, <u>3</u>, 8, 11, Don't Know}
		
		(6- P2Q2)
		After the above code n2 contains {24, 3, 8, <u>11</u>, Don't Know}
		
		(7- P2Q3)
		After the above code, n3 contains {<u>24</u>, 3, 8, 11, Don't Know}

To demonstrate students' results, pie charts were used for each programming variable. The right answer is shown in green, wrong answers are in red, and the "Don't Know" answers are in orange. Pretest and posttest results show that students' programming knowledge increased. The number of students in the green area of the pie charts increased in the posttest survey, and the green area increased in all questions. On the other hand, the orange area in the pie charts decreased or disappeared in the posttest, which means that fewer students indicated that they did not know the answers, suggesting their confidence in answering the programming questions improved.

Figure 4-45 shows the pretest and posttest results of PV1, PV2, and PV3 for the three groups. The percentage was rounded to the nearest integer number. The numbers outside the pie chart are the answering options, and the numbers inside it are the number of students who gave the specified answer. For example, in the first chart, 17 students chose 3 from the answering options.

The variable assignment question (PV1) was the easiest question among the three groups. Most of the students gave the correct answers in both pre- and posttests. No students chose the "Don't Know" option in the posttest. However, two students in G3 gave the wrong answers. Students' knowledge in variable assignment increased, but the difference between the pretest and the posttest result was not significant ($t(64) = -1.271$, $p = .208$).

The PV2 question measures students' understanding of variable addition in JavaScript. Results showed that this question was not as easy as the previous question. However, the percentage of students who gave the right answer rose in the posttest, and no students indicated the "Don't Know" option. This means that students' knowledge of variable addition improved.

Figure 4-45. Results of PV1, PV2, and PV3 variables.

Students found PV3, the variable multiplication question, easier than the variable addition question. The result was better for this variable. Most of the students gave correct answers in the posttest. Two students gave wrong answers, and no students indicated the "Don't Know" answer.

4.12.2 Programming variables PV4, PV5 and PV6. Table 4-37 shows the questions used to measure students' understanding of the for-loop (PV4), if-statement (PV5), and if-else statement (PV6). The same code was used to measure the three variables.

Table 4-37
PK Questions for Variables PV4, PV5, and PV6

Variables	Code	Questions

4. for-loop (PV4)
 	
5. if-statement (PV5)
 	
6. if-else statement (PV6)
	Assume the following block of code for the following two questions:
	`n1=0;`
	`n2=2;`
	`n3=3;`
	`for ( i = 1; i <= 4; i++)`
	`{`
	`    n1= n1 + 1;`
	`    if (n1 > 4)`
	`        n2 = n1 * n2;`
	
	`    if (n2 == 2)`
	`        n3=10;`
	`    else`
	`        n3=5;`
	`}`
		(8-P2Q4)
		After the above code n1 contains
		{0, 3, 4, 5, Don't Know}
		
		(9-P2Q5)
		After the above code,n2 contains
		{2, 4, 16, 8, Don't Know}
		
		(10-P2Q6)
		After the above code, n3 contains
		{10, 5, 3, 2, Don't Know}

Figure 4-46 shows the results for PV4, PV5, and PV6. The PV4 question measures the students' understanding of the for-loop. For this variable, the red areas were larger than the green, and this means that students did not find the question easy, and it required more thinking than the previous questions. Results showed that the for-loop question was the most difficult question for the students in the three groups.

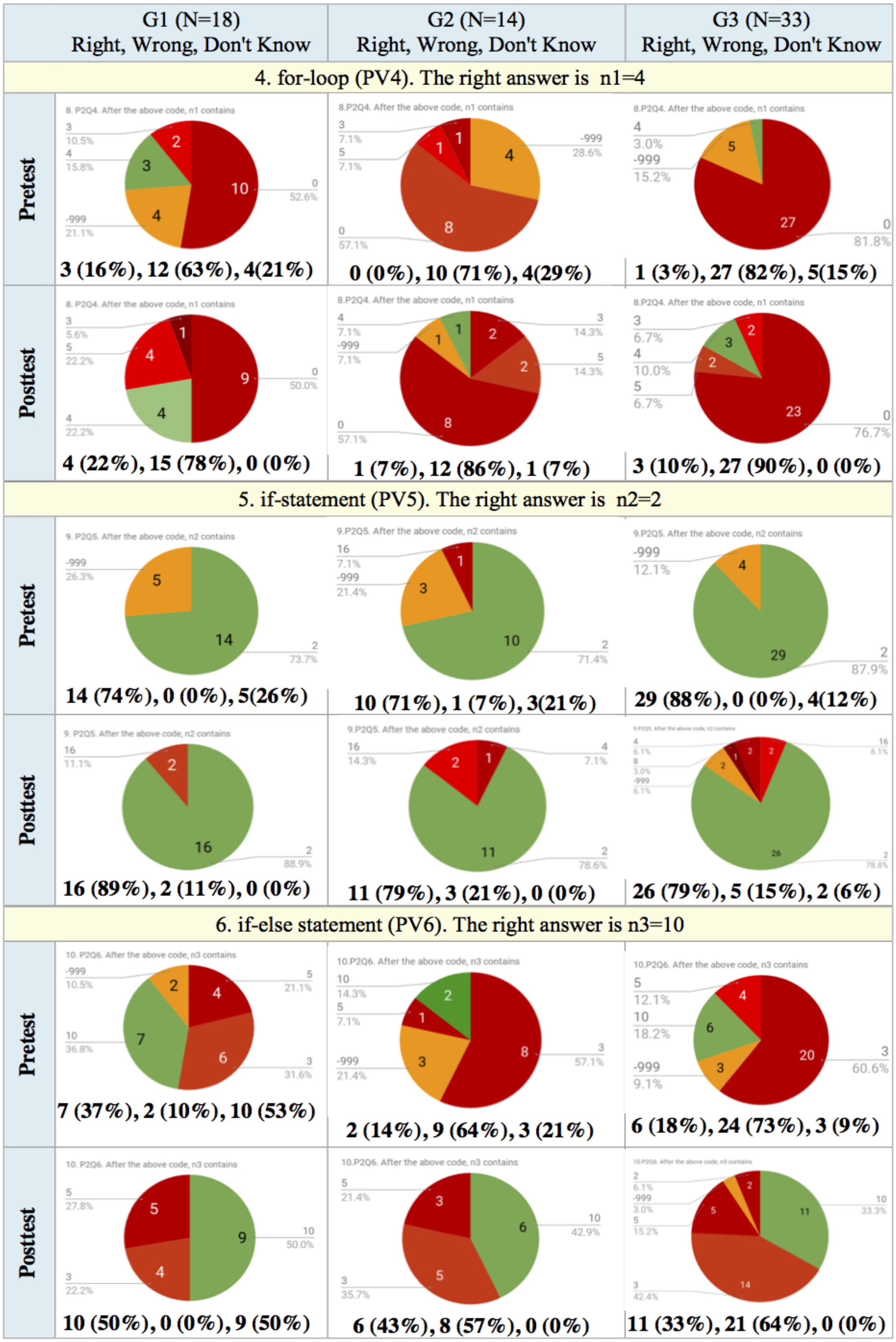


Figure 4-46. Results of PV4, PV5, and PV6 variables.

In the posttest, only one student in G2 chose the "Don't Know" option. However, only a small percentage of the students were able to answer this question correctly. G1 had the highest percentage of correct answers for this question (22%). PV4, PV5, and PV6 were measured using the same question, and this may have made it not an easy code.

Although the same question was used to measure students' understanding of the if-statement, results showed that students of the three groups performed better on this question, which represents the PV5 variable in this study. In the posttest, 89% of the G1 students and 78% of G2 and G3 students chose the correct answers. Only two students out of 65 students indicated the "Don't know" option.

The percentage of the students who gave the right answers for the if-else statement question of PV6 was smaller than for PV5; only 3% of students in G3 chose the "Don't know" option. The decrease in the orange area in the posttest indicates an increase in students' confidence to answer this programming question. Students' understanding of the if-else statement also improved in the posttest, but not as much as the previous question. The orange area almost disappeared in the posttest. Only one student chose the "Don't know" option in G3. The number of students who gave the right answer rose from 15 to 26 students in the three groups.

4.12.3 Programming variables PV7 and PV8. PV7 and PV8 are two new variables that were added to the two fall workshops groups, G2 and G3, to measure students' understanding of the if-else and for-loop statements with the use of art functions. These two variables were not measured in the summer camp. Table 4-38 shows the questions that were used to measure these two new variables.

The results of the previous section showed that students found the for-loop question (PV4) difficult and only a small percentage was able to give the correct answer. This could be because the for-loop, if-statement, and if-else statements were measured using the same piece of code. To increase simplicity, code readability, and to measure the effect of integrating art with code, the following two questions (Table 4-34) were developed and added to measure the students' understanding of the for-loop and the if-else statement in a separate smaller code for G2 and G3.

Table 4-38

PK Questions for Variables PV7 and PV8

Variables	Code	Questions

7. if-else statement with art (PV7)	Assume the following block of code
	`n1=4;`
	`if (n1 == 3)`
	`    circle(100,100,25);`
	`else`
	`    star(100,100,1);`
		(11-P2Q7)
		The above code will draw
		{ Circle, <u>Star</u>, Circle and Star , Nothing, Don't Know}

8. for-loop with art (PV8)	Assume the following block of code
	`for ( i = 1; i < 5; i++)`
	`{`
	`   circle(20*i,100,10);`
	`}`
		(12-P2Q8)
		The above code will draw
		{One Circle, Five Circles, <u>Four Circles</u>, Nothing, Don't Know}

PV7 is the variable that measures the students' understanding of the if-else statement with the use of art functions. The percentage of students who answered the PV7 question correctly increased from 43% to 79% in G2, though it only increased by 3% in G3. The orange area disappeared in both groups, as shown in Figure 4-47.

Comparing the results of PV4 and PV8 shows that students performed much better in the for-loop question with art (PV8) than with the previous for-loop question (PV4). The percentage of students who answered correctly rose from 7% to 43% in G2 and almost doubled in G3. However, the code was also simpler in the PV8 question.

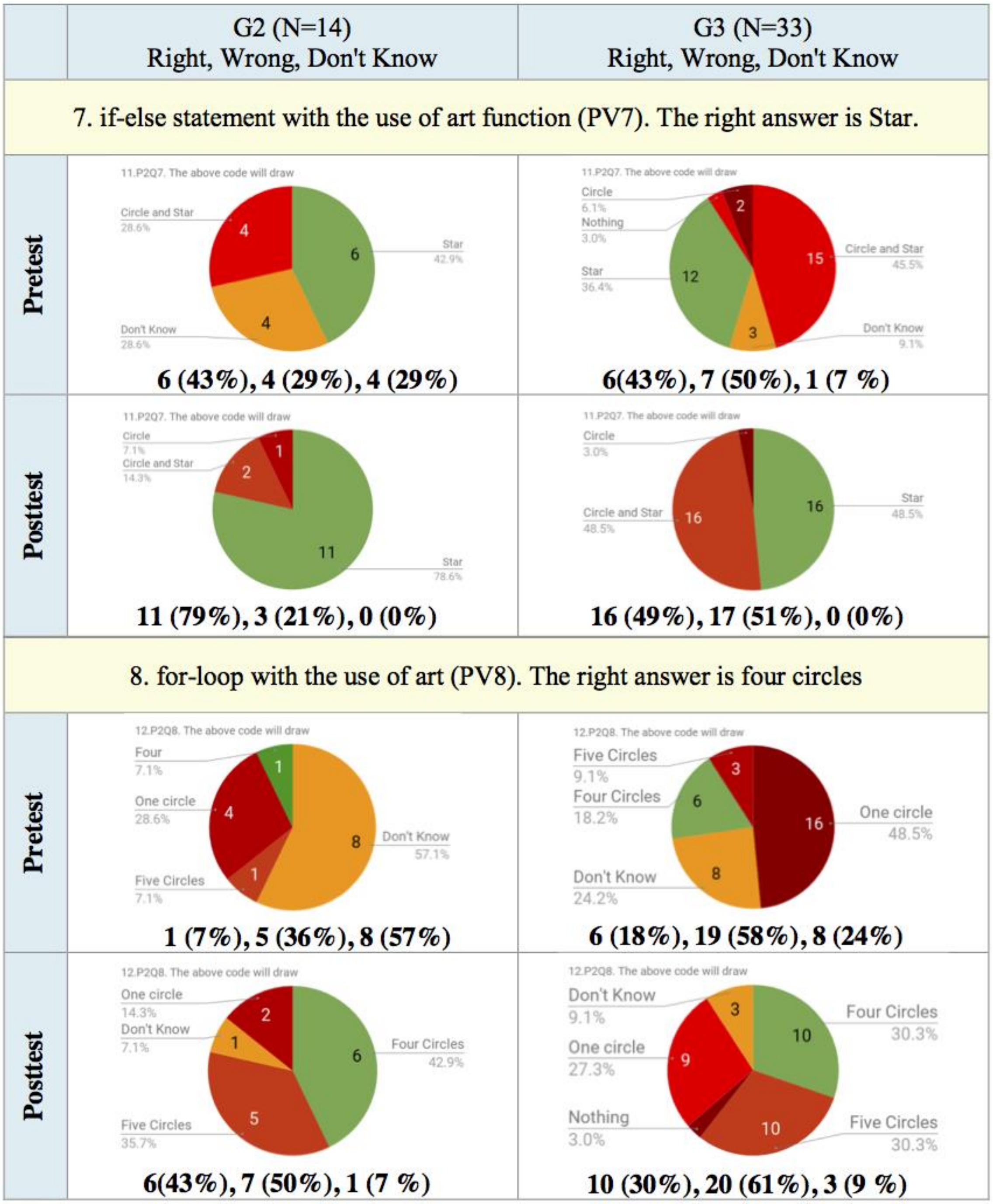


Figure 4-47. Results of PV7 and PV8 variables.

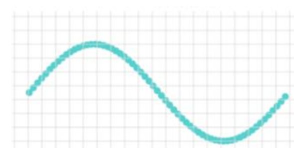
4.12.4 Programming variables PV9 and PV10. The variable PV9 measures the students' understanding of the switch-statement in JavaScript, while PV10 measures their understanding of the math functions. The following two questions were used to measure those two variables (Table 4-39).

Table 4-39

PK Questions for Variables PV9 and PV10

<table>
<tr><th>Variables</th><th>Code</th><th>Questions</th></tr>
<tr><td>9. switch-statement (PV9)</td><td>Assume the following block of code<pre>
n1=0;
n2=2;
switch (n2)
{
      case 0: n1=5; break;
      case 1: n1=10; break;
      case 2: n1=8; break;
      default: n1=5;
}
</pre></td><td>(13-P2Q9)
After the above code n1 contains
{10, 5, 3, 8, Don't Know}</td></tr>
<tr><td>10. Math Function (PV10)</td><td>Assume the following two piece of Code A& B and the following figure:

Code A<pre>
for (i = -3.14; i <= 3.14; i+= 0.1)
{
  y=150 + Math.cos (i) * 70;
  circle (250 + i * 60 , y , 5);
}
</pre>Code B<pre>
for (i = -3.14; i <= 3.14; i+= 0.1)
{
  y= 350 + Math.sin (i) * 70;
  circle (250 + i * 60 , y , 5);
}
</pre></td><td>(14-P2Q10)
Which code produces the shown figure?
{A, B, Both, None, Don't Know}

</td></tr>
</table>

PV9 measured the students' understanding of the switch-statement. Students of the three groups faced difficulty with this question. In the posttest, the number of students who gave the "Don't Know" answer in G2 was relatively large. However, the percentages of students who gave the correct answer improved in all three groups (see Figure 4-48).

The total number of students who gave the right answer for the PV10 question rose from 18 to 29 students across all groups. This math function question was not an easy question. In G2 and G3, 78% and 30%, respectively, of the students indicated the "Don't know" option. However, this percentage decreased to zero in G2 and to 20% in G3 in the posttest.

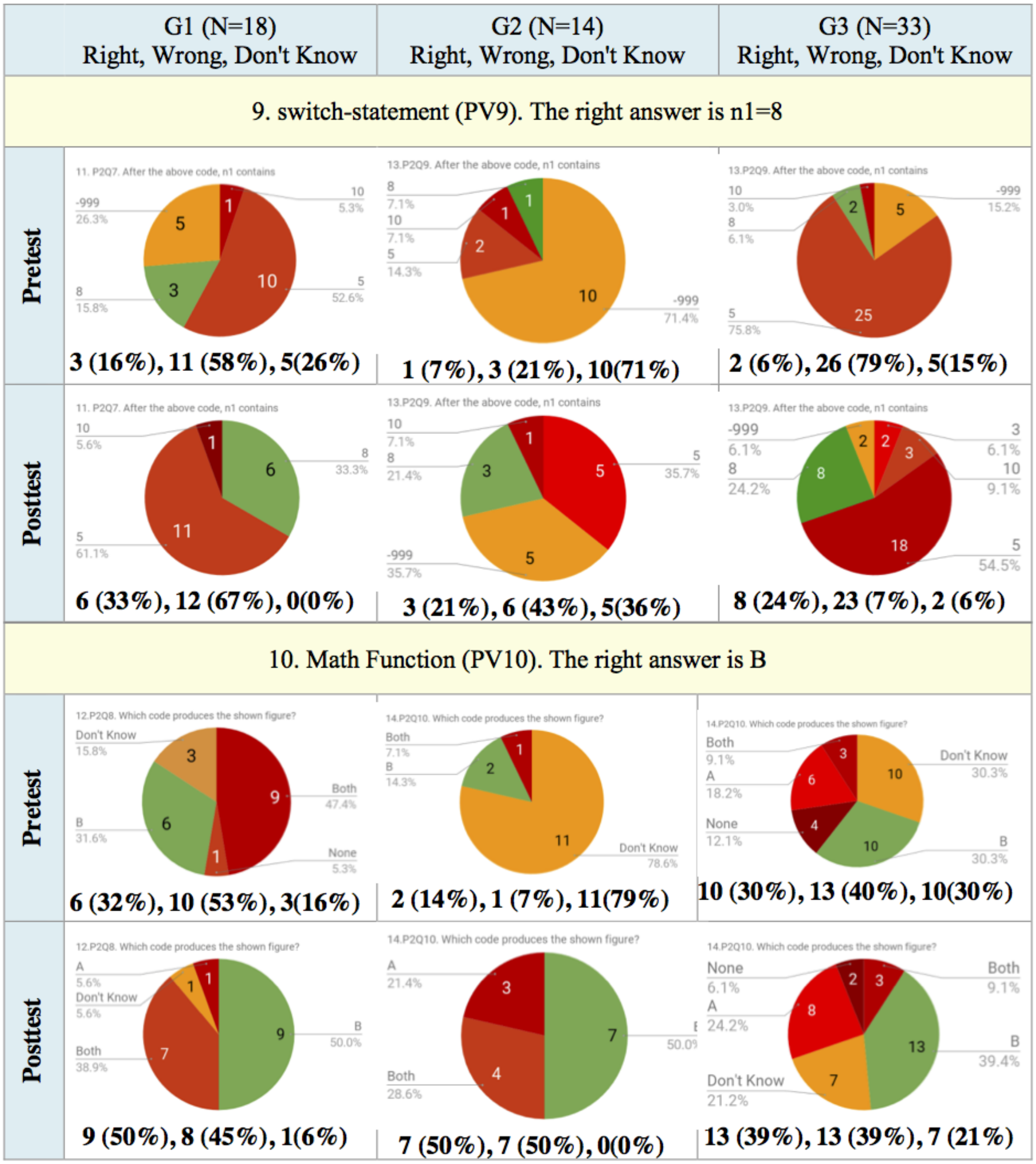


Figure 4-48. Results of PV9 and PV10 variables.

4.12.5 Programming variables PV11 and PV12. The variable PV11 measured the students' understanding of the Array concept in JavaScript. PV12 was used to measure students' understanding of the function concept. Table 4-40 shows the two questions that were used to measure those two variables.

Table 4-40

PK Questions for Variables PV11 and PV12

Variables	Code	Questions

11. Arrays (PV11)	The following code draws 10 stars
	`arrColor = ['red', 'green', 'purple',`
	`'yellow', 'pink', 'brown', 'gray', 'blue',`
	`'skyBlue', 'black' ] ;`
	
	`for ( i=0 ; i<10 ; i++)`
	`{`
	`  color ( arrColor [ i ] ) ;`
	`  star ( 20 + i * 50, 100 , 1);`
	`}`
	
		(15- P2Q11)
		What is the value of " i " in " color (arrColor [i]) ; " that gives the 'LightPink' color ?
		{0, 1, 2, 3, Don't Know}

12. Function (PV12)	Assume the following block of code
	`function fn (y)`
	`{`
	`    return y * y;`
	`}`
	`x= fn (3);`
	`z= fn (4);`
		(16-P2Q12)
		After the above code x contains
		{4, 16, 9, 3, Don't Know}

In the Array question, students' performance improved in the posttest for all three groups. The "Don't Know" option, or the orange area, decreased from 57% to 7% in G2; it decreased from 18% to 6% in G3, and it disappeared in G1.

Figure 4-49 shows the results of the PV11and PV12 variables. The last programming question was used to measure students' knowledge of functions. The percentage of students who gave the correct answers improved in G1 and in G2, but it decreased in G3. The orange area, or the "Don't Know" option, disappeared in G1, and it decreased from 43% to 14% in G2, but it was not affected much in G3.

To summarize, the overall results show that the green area in the pie charts increased in the posttest for most of the programming questions for the three groups. However, for G3, the percentage of the correct answers in the posttest was smaller than the pretest for two variables, PV5 and PV12. In the posttest, the orange area decreased in all of the pie charts, but it also disappeared in many questions, especially for G1students who had more programming hours. In general, the results show that the students' programming knowledge improved after the coding workshop week, and their confidence to answer the questions increased.

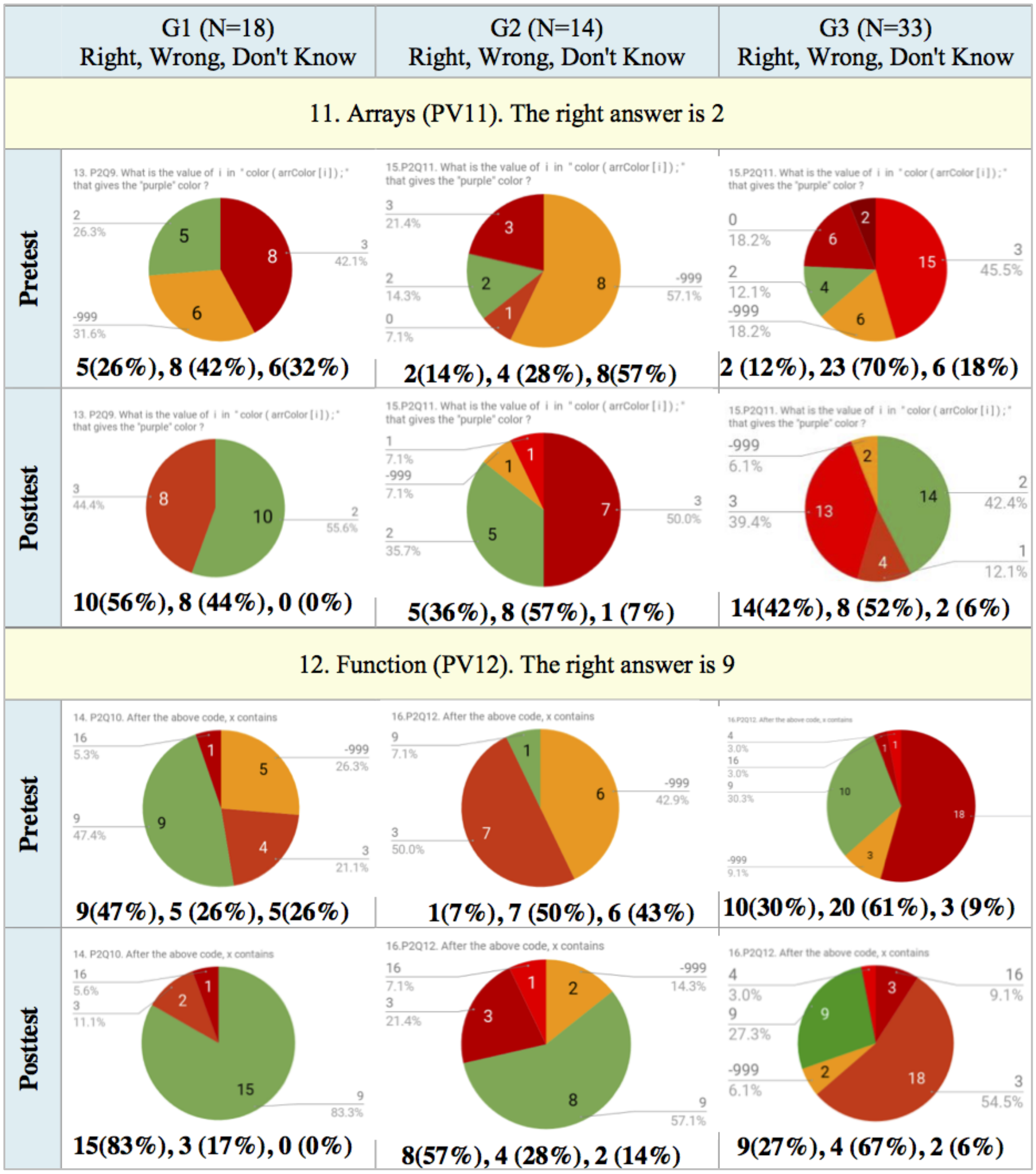


Figure 4-49. Results of PV11 and PV12 variables.

4.12.6 Programming variables results analysis. Before running the paired sample analysis, new columns were added for each response to each programming question that was used to measure the PV1 to PV12 variables. The new columns indicated if the answer was right, wrong, or "Don't Know." These columns were used to compare the number of right answers, wrong answers, and the "don't know" answers to find if the results were statistically significant between the pretest and the posttest. Table 4-41 and Figure 4-50 show that the percentage of correct answers increased in the posttest for all questions.

Table 4-41

The Percentages of the Students' Answers for PK Questions in $G_{1,2,3}$

	Right			Wrong		Don't Know	
	Pretest	Posttest	increment	Pretest	Posttest	Pretest	Posttest
PV1	94%	97%	3%	3%	3%	3%	0%
PV2	28%	52%	24%	65%	48%	8%	0%
PV3	86%	97%	11%	6%	3%	8%	0%
PV4	6 %	12%	6%	74%	82%	20%	6%
PV5	80 %	82%	2%	2%	15%	19%	2%
PV6	23 %	40%	17%	65%	59%	12%	2%
PV7	38 %	57%	19%	47%	43%	15%	0%
PV8	15 %	34%	19%	51%	57%	34%	9%
PV9	9 %	26%	17%	60%	63%	31%	11%
PV10	28 %	45%	17%	35%	42%	37%	14%
PV11	17 %	45%	28%	54%	51%	29%	5%
PV12	31 %	49%	18%	48%	45%	22%	6%

The best percentage increment was 28% for the PV11 variable, which was used to measure the array understanding, followed by PV2 (24%) and PV7 (19%), which were used to measure variable addition and if-else-statement respectively. PV1, PV3, and PV5 had the highest percentages of correct answers in the posttest, and they were used to measure variable assignment, multiplication, and if-statement, respectively.

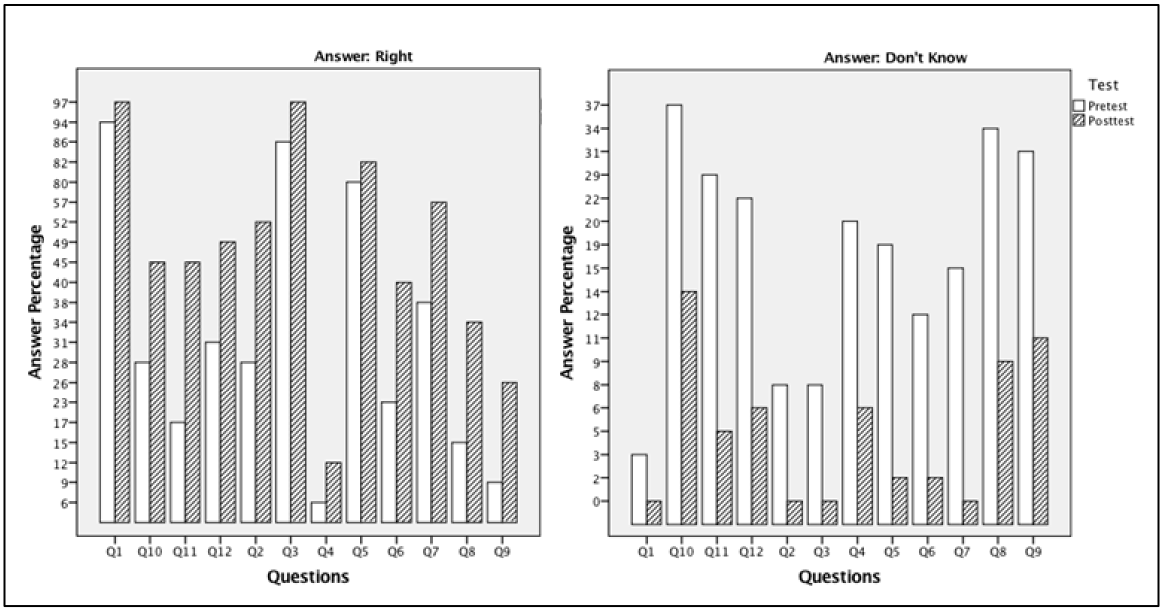


Figure 4-50. Students' answers to the programming knowledge questions.

Table 4-42

$G_{1,2,3}$ T-Test Results for Sub-Variables PV1 to PV12

	Paired	Mean Differences	t	df	Sig. (2-tailed)
Pair 1	PV1 Pretest - PV1 Posttest	-.12308	-1.271	64	.208
Pair 2	PV2 Pretest - PV2 Posttest	-.64615	-4.235	64	.000
Pair 3	PV3 Pretest - PV3 Posttest	-.36923	-2.551	64	.013
Pair 4	PV4 Pretest - PV4 Posttest	-.40000	-2.611	64	.011
Pair 5	PV5 Pretest - PV5 Posttest	-.40625	-2.200	64	.031
Pair 6	PV6 Pretest - PV6 Posttest	-.55385	-3.207	64	.002
Pair 7	PV7 Pretest - PV7 Posttest	-.68085	-3.491	46	.001
Pair 8	PV8 Pretest - PV8 Posttest	-.89362	-4.691	46	.000
Pair 9	PV9 Pretest - PV9 Posttest	-.73846	-4.137	64	.000
Pair 10	PV10 Pretest - PV10 Posttest	-.80000	-3.592	64	.001
Pair 11	PV11 Pretest - PV11 Posttest	-1.04615	-6.138	64	.000
Pair 12	PV12Pretest - PV12 Posttest	-.54839	-2.655	64	.010

The t-test results for all the programming variables were statistically significant with p-value less than 0.05, except for the first variable (PV1), which was easy, and students gave correct answers in both tests (Table 4-42). This statistical significance indicates that the treatment was effective, and it significantly increased the programming knowledge of all students in $G_{1,2,3}$ in the programming concepts that were measured in this study.

4.12.7 Individual Score Progress for G1, G2, and G3. The individual scores for each student in the three groups are discussed in this section. In addition, the scores for different races and genders are also discussed.

Figure 4-50 shows the individual score progress in the pretest and posttest for G1 grouped by gender and race. Most of the students improved their scores in the posttest. However, one female student had the same scores in both tests, one female student had a higher score in the pretest, and one male student was lost by the end of the experiment and did not take the posttest survey. In the paired samples t-test analysis, the scores of the lost student were ignored.

In the pretest, the scores of the female students were higher than the scores of the male students. The lowest pretest score for the female students was 4, and the highest was 8. In contrast, the lowest score for the male students was 0, and the highest was 9. In the posttest, the scores of both groups improved. The lowest score for female students was 6, and the highest was 9, and the lowest score for male students was 4, and the highest was 10. The percentage of the female students who were able to answer half or more of the questions correctly improved from 85% in the pretest to 100% in the posttest. The percentage for male students improved from 36% in the pretest to 91% in the posttest. This suggests that the treatment was effective for both genders and increased their scores and hence their programming knowledge in the posttest.

In Figure 4-51 and in the line chart for the students' progress grouped by race, it can be seen that the highest scores were by the Asian group in the pretest.

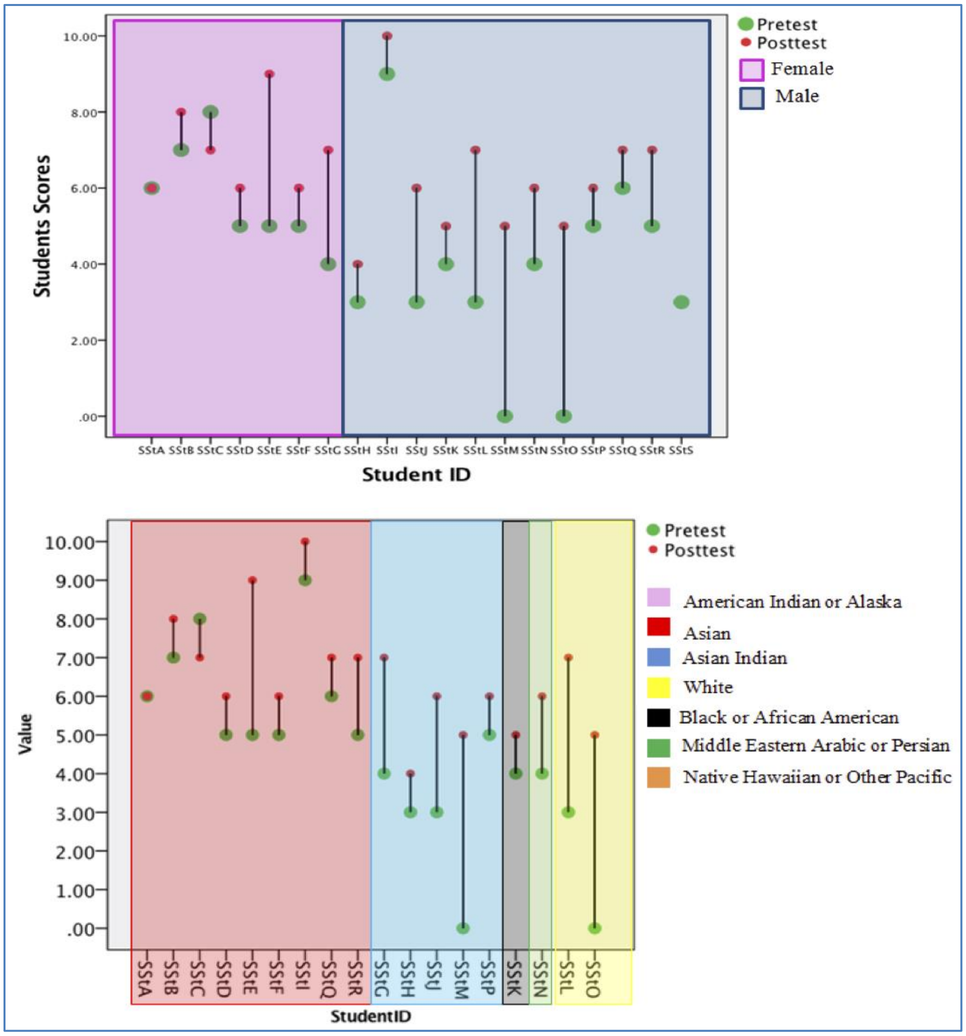


Figure 4-51. G1 Gender and race vs. individual score progress.

Those scores ranged between 9 and 5, while the Asian Indian scores ranged between 5 and 0. The highest score, which was 9, belonged to an Asian student. Two students, one Asian Indian and one White, scored the lowest scores, which were 0. The percentage of students who were able to answer half or more questions correctly was 100% among the Asian group. It was 20% among the Asian Indian group and 0% among other racial groups. In the posttest, the Asian group also had the highest score range, ranging between 10 and 6. The Asian Indian group's scores ranged between 7 and 4, and the White group's scores ranged between 7 and 5. The same Asian student who had had the highest score in the pretest also scored the highest score in the posttest, a 10. An Asian Indian student scored the lowest score in the posttest, a 4. The percentage of the students who were able to answer half or more of the questions correctly in the posttest was 100% among all racial groups.

Figure 4-52 shows the students' scores in G2. Most of the students improved their scores in the posttest.

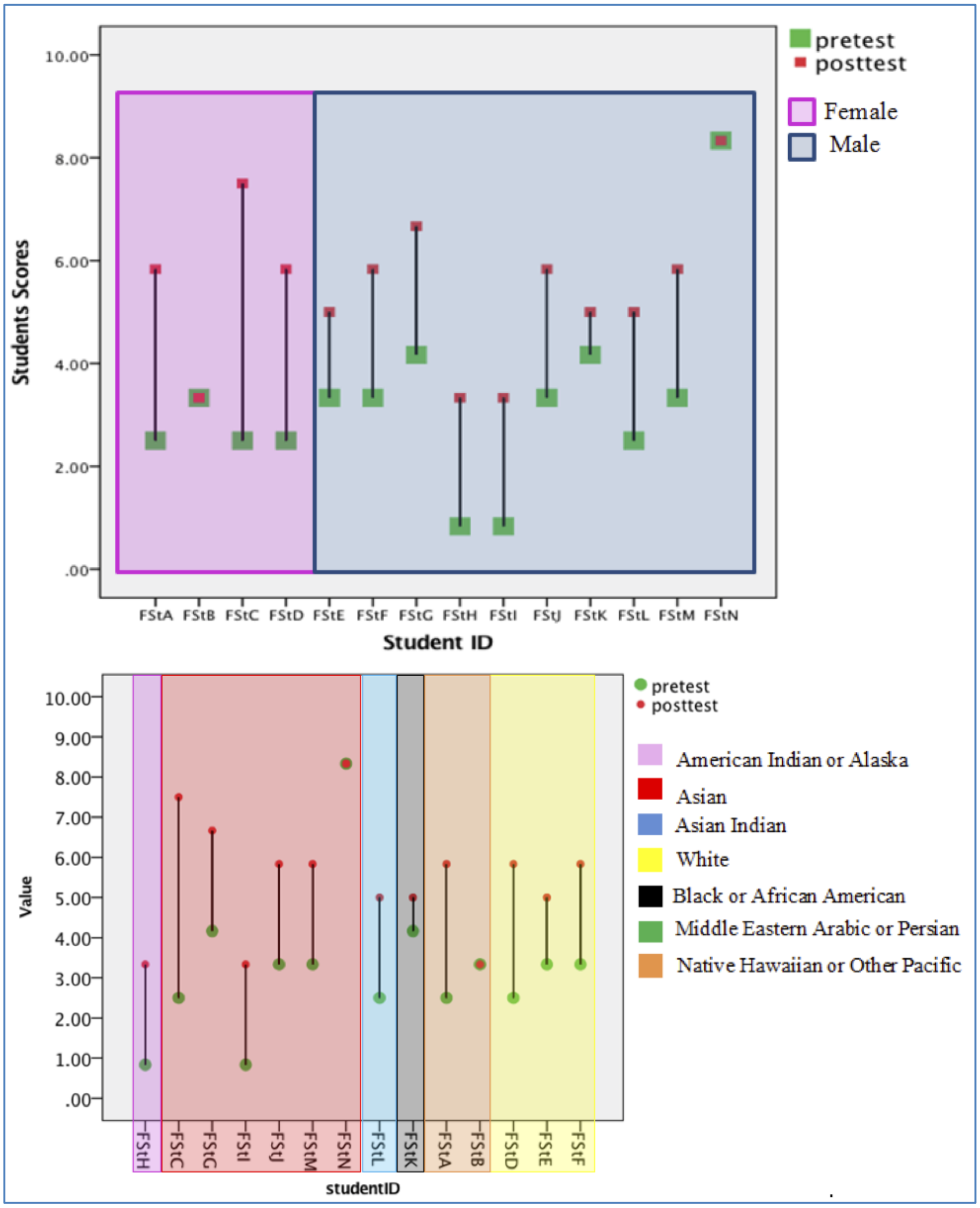


Figure 4-52. G2 Gender and race vs. individual score progress linechart.

However, one female and one male student had the same scores in both tests. In the pretest, the lowest score was 2.5, and it was the same for both male and female students. The highest score was 3.3 for the female students and 8.33 for the male students. In the posttest, the lowest score for the female students was 3.3, and the highest was 7.5. For the male students, the lowest score was 3.3, and the highest was 8.33.

In the pretest, the percentage of students who were able to answer half or more of the questions correctly was 0% for female students, and it was 10% for male students. In the posttest, these percentages improved to 75% for the female students and 80% for the male students. Similar to the case in G1, the treatment was also effective for both genders in G2, and it increased the students' scores and their programming knowledge in the posttest.

In Figure 4-53 of G2, the line chart grouped by race shows that most of the students improved their scores, except one Hawaiian student and one Asian student who had the same scores in both tests. The Asian student's score was also the highest score in both tests for this group, and it was 8.33. The one American Indian student shared the lowest score with another Asian student in the pretest. The overall Asian students' scores ranged from .83 to 8.33. The White students' and the Hawaiian students' scores ranged between 2.5 and 3.33. There was only one Black student and only one Asian Indian student, and their scores were 4.1 and 2.5, respectively. In the pretest, the percentage of the students who were able to answer half of the questions or more correctly was 17% for the Asian group and 0% for the other racial groups. In the posttest, this percentage improved to 100% for the Asian Indian, White, and Black groups. It was 83% for Asian and 50% for Hawaiian.

The statistics for G3 show that 21 students, or 63% of all the students, improved their scores in the posttest. In contrast to G1, Figure 4-52 shows that in G3 the scores of the male students were better than the scores of the female students. Also, more male students were able to improve their scores in the posttest. Of the female students, 57% improved their scores, but 29% female students had the same scores in both tests, and 14% had lower scores in the posttest. Of the male students, however, 84% improved their scores in the posttest and 16%, or only two male students, had lower scores.

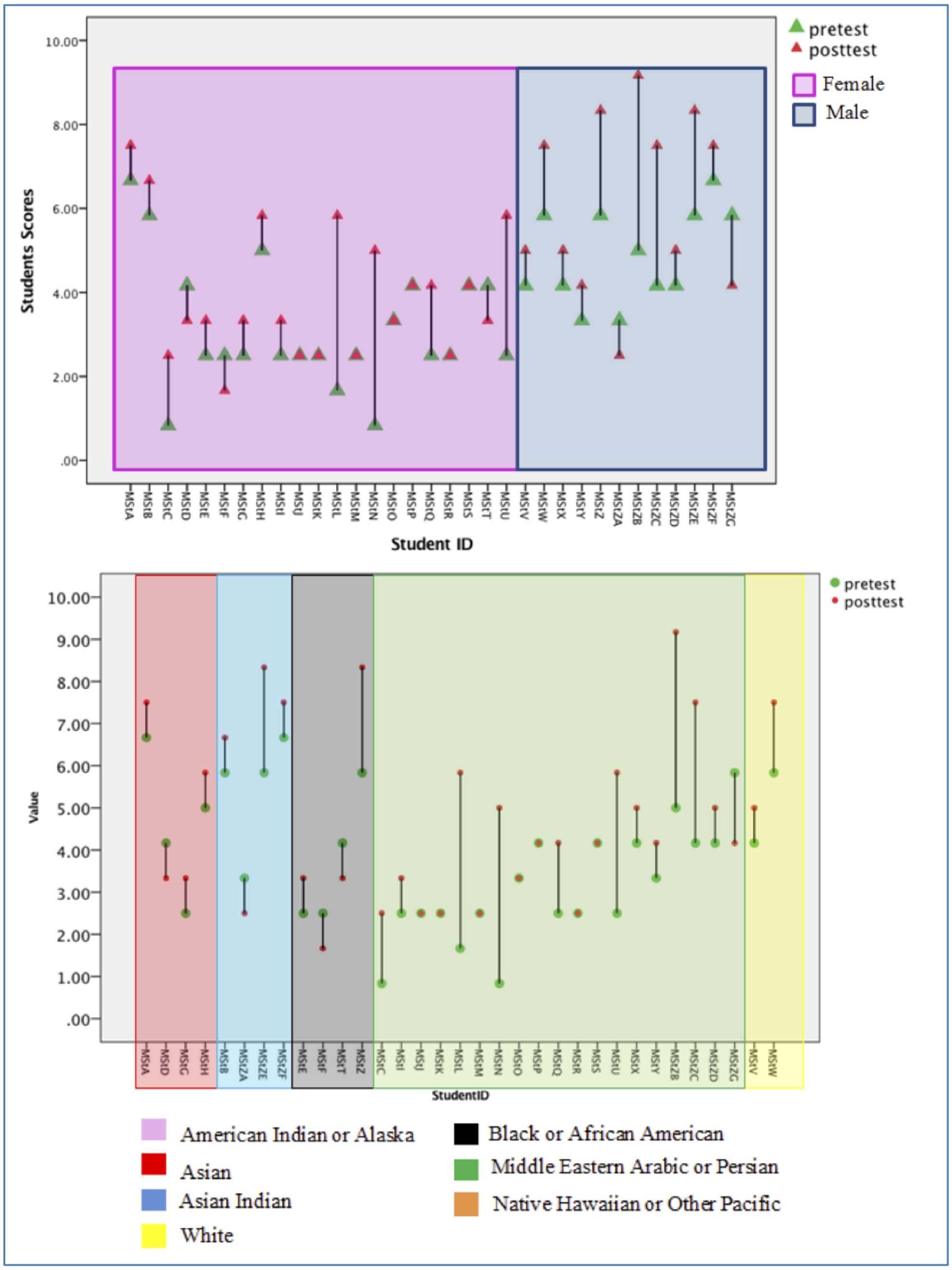


Figure 4-53. G3 Gender and race vs. individual score progress linechart.

In the pretest, the lowest score for the female students was .83, and the highest was 6.67. The lowest score for the male students was 3.3, and the highest was 6.67. In the posttest, the lowest score for the female students was 1.67, and the highest was 7.5. The lowest score for the male students, in the posttest, was 2.5, and the highest was 9.17.

The percentage of the female students who were able to answer half or more of the questions correctly in the pretest was 14%, but it was 50% for the male students. In the posttest, these percentages improved to 29% for female students and 75% for male students.

The statistics of G3, grouped by race, show that most of the racial groups were able to improve their scores, except one Asian, one Asian Indian, one Middle Eastern, and two Black students. These students had lower scores in the posttest than the pretest. Seven Middle Eastern students had the same scores for both tests. In the pretest, the score ranges for different racial groups were as follows: Asian scores ranged between 2.5 and 6.7, Asian Indian between 3.3 and 6.7, Black between 2.5 and 5.8, Middle Eastern between .83 and 5.83, and White between 4.7 and 5.83. In the posttest, the score ranges improved as follows: Asian between 3.3 and 7.5, Asian Indian between 2.5 and 8.33, Black between 1.7 and 8.33, Middle Eastern between 2.5 and 9.17, and White between 5 and 7. In the pretest, the percentages of the students who were able to answer half or more of the questions correctly were as follows: Asian 50%, Asian Indian 75%, Black 25%, Middle Eastern 11%, and White 50%. These percentages improved in the posttest for only two racial groups: the Middle Eastern percentage improved to 37%, and White improved to 100%. Other racial groups had the same percentages as in the pretest.

4.12.8 Coding hours and PK mean difference. The three groups were exposed to different numbers of coding hours but the same teaching materials that were used to measure student understanding in the programming knowledge question were used for all groups.

However, groups with more coding hours and days had the chance to practice more coding in class and at home.

The difference between the pretest and posttest scores was computed for each student in a new variable (Score Difference). Figure 4-54 shows the mean difference of scores for each group. It also shows the score difference for each student in the three groups. In the first chart, the mean of the score difference for G2 (M = 2.22) was higher than that for G1 (M = 1.94) and for G3 (M = 1.01). In the second chart, the point below the zero line means that the student had lower scores in the posttest than in the pretest. As shown in G1 (red line), the number of students who had lower scores in G1 was only one, no students for G2, and 5 students for G3. The figure also suggests that there is no clear relation between the amount of time and the score difference. G2 had less coding time than G1, but the score difference mean was a little higher than that for G1.

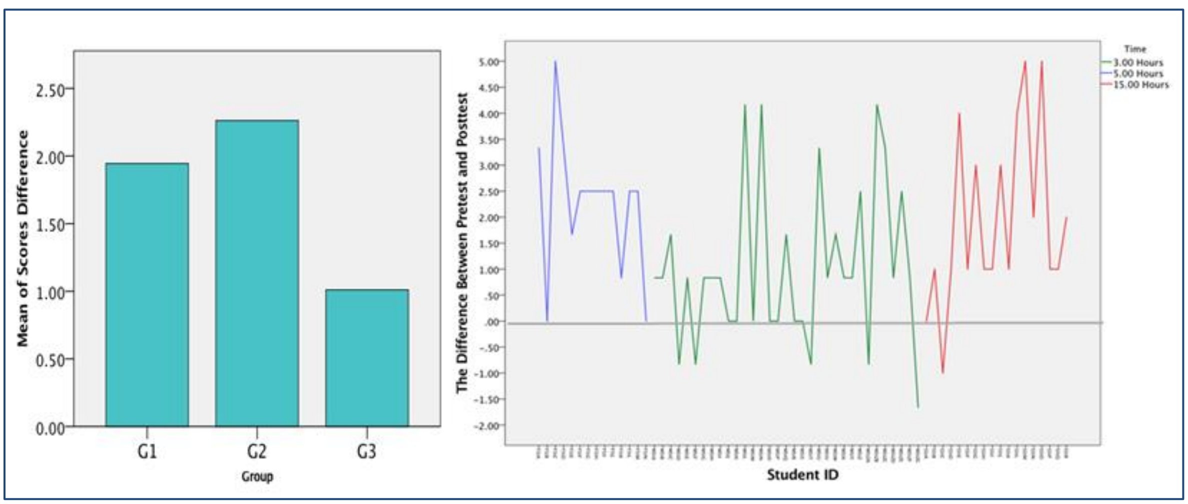


Figure 4-54. Students' score difference vs. workshop time.

Correlation analysis was used to see if there was any relation between the amount of coding time and the difference between the students' scores in the pretest and posttest for the three groups. The results show a small Pearson correlation coefficient (r =.201, n=65, p = .10), indicating a positive weak correlation between the variables. This means when the coding time increased, the difference in the programming knowledge scores slightly increased for the specified programming questions that were used in this study.

4.13 Results Summary and Conclusion

Table 4-43 summarizes the students' responses for the study variables, except the PK variable that was measured by the students' scores in both tests instead of students' agreement or disagreement. For the PK variable, the percentage of students who were able to answer half or more questions correctly rose from 31% in the pretest to 66% in the posttest for all students ($G_{1,2,3}$). The t-test was significant ($t(64) = -7.7$, $p = .000$). As shown in Table 4-43, the agreement percentages increased, and the disagreement percentages decreased in the posttest for all the study variables. The neutral percentage decreased, except for the CI variable.

Table 4-43
$G_{1,2,3}$ Summary of Responses and the T-Test for All Variables

	Agreement %		Disagreement %		Neutral %		t-test	
$G_{1,2,3}$	Pretest	Posttest	Pretest	Posttest	Pretest	Posttest	t	p
DI	40%	52%	25%	17%	35%	31%	-2.68	.009
RPP	22%	54%	28%	12%	50%	34%	-5.14	.000
MCS	40%	49%	23%	22%	37%	29%	-2.03	.046
PIE	39%	46%	22%	22%	40%	32%	-2.59	.012
CI	31%	34%	43%	31%	26%	35%	-1.8	.072

The paired sample t-test results were significant for four variables (DI, RPP, MCS, and PIE), and it approached the borderline of significance for the CI variable.

Table 4-44 summarizes the p-value results of the t-test analysis for the six study variables. For the race results, only the p-values that are less than one were listed in the table.

For female students, the results were significant for all of the six variables, but for the male students, the results were only significant for the Programming Knowledge (PK) and Real Programming Preference (RPP).

Table 4-44
$G_{1,2,3}$ Summary of the T-Test P-Value Results for the Study Variables

Variables	All Students	Female	Male	Asian	Asian Indian	Black	Middle Eastern	White
DI	.009	.002	.667					.047
PK	.000	.000	.000	.001	.005		.003	.003
RPP	.000	.000	.009	.000	.07		.049	
MCS	.046	.071	.296	.08				
PIE	.012	.001	.566	.093				
CI	.072	.02	.69					
* The mean value decreased in the posttest								

Most of the male students agreed with the survey items from the pretest, and their responses did not change significantly in the posttest. In contrast, the responses of the female students changed significantly in the posttest for all of the six study variables. The results for these two variables (PK and RPP) were significant for the Asian, Asian Indian, and Middle Eastern students. For the PK variable, the result was also significant for the White students.

Table 4-45 shows the students' response differences between G3 students who had fewer coding hours and who were exposed to the coding workshop, and $G_{1,2}$ students who had more coding hours and who joined the workshop upon their own interest.

Table 4-45
G3 vs. $G_{1,2}$ Summary of Responses for All Variables

		Agreement %		Disagreement %		Neutral %	
		Pretest	Posttest	Pretest	Posttest	Pretest	Posttest
$G_{1,2}$	DI	50%	72%	6%	6%	44%	22%
N=32	RPP	25%	75%	3%	0%	72%	25%
	MCS	44%	63%	13%	9%	44%	28%
	PIE	53%	66%	6%	6%	41%	28%
	CI	47%	50%	25%	19%	28%	31%
G3	DI	30%	33%	42%	27%	27%	39%
N=33	RPP	18%	33%	52%	24%	30%	43%
	MCS	37%	37%	33%	33%	30%	30%
	PIE	24%	27%	37%	37%	39%	36%
	CI	15%	18%	61%	43%	24%	39%

The agreement percentages increased for all variables in $G_{1,2}$, and the results were statistically significant for all variables, except CI. The t-test result for PK was significant for $G_{1,2}$ ($t(31)$ = -7.7, p = .000) and G3 ($t(32)$ = - 3.79, p = .001). The agreement percentages increased for most of the variables in G3, except for MCS, which remained unchanged. The t-test results were only significant for PK and RPP in G3.

Table 4-46 shows the agreement percentages of the six variables in both tests for the female and male students in $G_{1,2}$ and G3. The agreement percentages for $G_{1,2}$ increased for all variables and for both genders. In G3, the female agreement percentages either increased or remained unchanged while the percentages for the G3 male students either decreased or remained unchanged.

Table 4-46
G3 vs. $G_{1,2}$ Agreement Percentages of Both Genders for All Variables

		Female Agreement %		Male Agreement %	
	Variable	Pretest	Posttest	Pretest	Posttest
$G_{1,2}$	DI	46 %	73 %	52 %	71 %
N=32	RPP	0 %	82 %	38 %	71 %
	MCS	36 %	64 %	48 %	62 %
	PIE	45 %	55 %	57 %	71 %
	CI	18 %	27 %	62 %	62 %
G3	DI	19 %	38 %	50 %	25 %
N=33	RPP	5 %	29 %	42 %	42 %
	MCS	38 %	38 %	33 %	33 %
	PIE	24 %	29 %	25 %	25 %
	CI	0 %	10 %	42 %	33 %

Table 4-47 shows the p-values for both genders in $G_{1,2}$ and G3. For $G_{1,2}$, the results were significant for most of the variables, except CI. For G3, the responses of the students significantly changed for two variables only (PK and RPP).

Table 4-47

G3 vs $G_{1,2}$ Summary of the T-Test P-Value Results for all Variables

	G3			$G_{1,2}$		
Variables	All Students	Female	Male	All Students	Female	Male
DI	.16	.024	.275*	.019	.025	.204
PK	.001	.017	.018	.000	.009	.000
RPP	.017	.012	.463	.000	.000	.006
MCS	1.00	.45	.27*	.009	.052	.071
PIE	.459	.04	.174*	.005	.003	.103
CI	.102	.009	.866*	.42	.77	.44

* The mean value decreased in the posttest

Female students' responses significantly changed for five variables in both groups, but the responses significantly changed for only two variables for male students. In the posttest, the responses improved for female students in the three groups and for the male students in $G_{1,2}$. But the responses of the male students in G3 dropped for several variables in the posttest, except for the PK and RPP variables. However, for those students, the mean values did not drop significantly for the six variables.

In comparison between the three groups, Table 4-48 shows that the results were significant for four variables in both G1 and G2, while in G3 the responses changed significantly in the posttest for two variables only (PK and RPP).

Table 4-48

G1, G2, and G3 Summary of the T-Test P-Value Results for all Variables

Variables	G1	Female	Male	G2	Female	Male	G3	Female	Male
DI	.028	.20	.09	.33	.058	1.00	.16	.024	.275*
PK	.000	.093	.001	.000	.069	.000	.001	.017	.018
RPP	.000	.004	.010	.047	.092	.279	.017	.012	.463
MCS	.138	.030	.588	.033	.495	.045	1.00	.452	.275*
PIE	.091	.028	.572	.025	.072	.128	.459	.042	.174*
CI	.055	.103	.341	.856*	.215*	.693	.102	.009	.866*

* The mean value decreased in the posttest

Females of G3 had the most significant results among the different genders in the three groups despite their shortest coding workshop. The mean values for most variables improved in the posttest for all groups, except for the male students in G3 where their mean values for four variables slightly dropped. The mean value for the female students in G2 also dropped slightly in the posttest, and that drop affected the mean value of all students in G2.

Correlation analysis showed that the relation between the students' interest in a CS degree and the three factors suggested by the planned behavior theory were positive and strong. The regression model showed that the Programming Benefit and Enjoyment (PBE) factor and Programming Capabilities and Confidence (PCC) were two predictors that had an effect on students' interest in pursuing a CS degree.

The results were significant for all individual programming questions. Most students indicated that the Code Genie learning environment was useful and easy to use, and most of them liked the elements used in this experiment (Art, Animation, and Code Sharing). The students seemed to enjoy the overall experience and will potentially code again with Code Genie.

4.14 Summary

In this chapter, the demographic data were first analyzed, and then reliability of the results was checked in sections 4.1 and 4.2, respectively. From section 4.3 to section 4.9, the results were analyzed for the study variables in three stages. The results for all students were analyzed first. Second, the results for the students who had more coding workshop time and more time to explore the developed tool versus students who had less time were analyzed. Finally, the results were also analyzed for each group. The paired sample t-test was mainly used to analyze the results in addition to the descriptive statistics.

In section 4.10, correlation and regression analyses were used to explore the relation between the students' interest in pursuing a CS degree and the three factors suggested by the theory of planned behavior. Section 4.11 discussed the students' responses to the tool's usability and their preferred elements in the tool. Section 4.12 discussed the programming knowledge in detail and demonstrated the individual scores for each group. This section also discussed the relation between the amount of coding time and the programming knowledge scores' difference. Section 4.13 summarized the results analyses for the study variables. Chapter Five uses the results of Chapter Four to accept or reject the study hypotheses and to answer the research questions.

Chapter Five: Findings and Discussion

While the previous chapter discusses each study variable in detail, this chapter summarizes the results, addresses the hypotheses, answers the research questions, and discusses the study findings. Students' comments and engagement are also discussed in this chapter. The findings of other studies are compared to findings of this dissertation study in the "Discussion and Conclusion" section. Finally, the chapter ends with the future work and the possible research areas or domains.

5.1 Hypotheses Discussion

The paired sample t-test results of all students ($G_{1,2,3}$) were used to test the study hypotheses and to reject or accept the null hypotheses. Each hypothesis is discussed in the following sections.

5.1.1 Hypothesis H1: Interest in CS degree. Three hypotheses are discussed in this section; one main and two sub-hypotheses. The results for the DI (Degree Interest) were used to address these three hypotheses.

> **H1:** Integrating art and animation in teaching computer programming increases students' interest in pursuing a CS degree.
>
> **$H1_o$:** Integrating art and animation in teaching computer programming has no significant effect on students' interest in pursuing a CS degree.

The students' agreement percentage on the questions that measured their interest in a CS degree rose from 40% in the pretest to 52% in the posttest. The results of the paired sample analysis of the DI variable for all students were statistically significant, $t(64) = -2.681$, $p = .009$. This means that the treatment was effective and the use of art and animation in teaching computer programming increased the high school students' interest in pursuing a CS degree. Hypothesis H1 was accepted and the null hypothesis $H1_o$ was rejected.

> **H1A:** Integrating art and animation in teaching computer programming increases <u>female</u> students' interest in pursuing a CS degree.
>
> **$H1A_o$:** Integrating art and animation in teaching computer programming has no significant effect on the <u>female</u> students' interest in pursuing a CS degree.

As discussed in Chapter Two, statistics from big companies showed that the computer science field and the tech jobs in the big companies such as Google, Apple, and Facebook are dominated by male employees. One of the goals of this study was to encourage females to learn computer programming and to pursue a degree in CS. The results of this experimental study showed that the female students' interest in a CS degree increased significantly. The coding camp was effective enough to change the agreement percentages of female students from 28% in the pretest to 50% in the posttest. The t-test result for the females in $G_{1,2,3}$ was $t(31) = -3.418$, $p = .002$. The result was statistically significant ($p < .05$), which means that the treatment was effective enough to increase the female students' interest in a CS degree. H1A hypothesis was accepted and the null hypothesis $H1A_o$ was rejected.

The agreement percentage for the male students was 52% in the pretest and it rose to 55% in the posttest. However, the change was not statistically significant for the male students for the DI (CS Degree Interest) variable ($p = .677$).

H1B: Integrating art and animation in teaching programming increases the CS degree interest for students of different racial groups.

$H1B_o$: Integrating art and animation in teaching programming has no significant effect on the CS degree interest for students of different racial groups.

For the DI variable, the result of the White racial group was the only significant result among other racial groups, $t(6) = -2.500$, $p = .047$. The results for the other racial groups were not significant. The participation of underrepresented groups like the Black group was low (N=6). Two Black students attended the summer camp (G1); one of them left in the fourth day and did not complete the camp. There was only one Black student in the fall workshop (G2), and he showed high interest in coding. The third workshop (G3), which was the shortest, had four Black students.

The well-represented Asian and Indian Asian groups indicated high interest in a CS degree (DI variable) from the beginning of the experiment (pretest), and their level of interest did not increase significantly. However, the results for the female students, who were mostly Asian in G1 and mostly Middle Eastern in G3, were significant.

Since there was not any significant result for any racial group other than the White group, Hypothesis H1B was rejected and the null hypothesis was accepted. However, the researcher thinks if a larger number of Black students were exposed to a long enough coding workshop, the result might be different. The result for this hypothesis might be affected by the low participation of the Black and other underrepresented racial groups in G1 (*N* = 1) and G2 (*N* = 1), and it is also affected by the greatest number of Black students having participated in the shortest coding workshop in G3 (*N* = 4).

5.1.2 Hypothesis H2: Programming knowledge.

H2: Integrating art and animation in teaching programming increases students' knowledge in programming language.

$H2_o$: Integrating art and animation in teaching programming has no significant effect on students' knowledge in programming language.

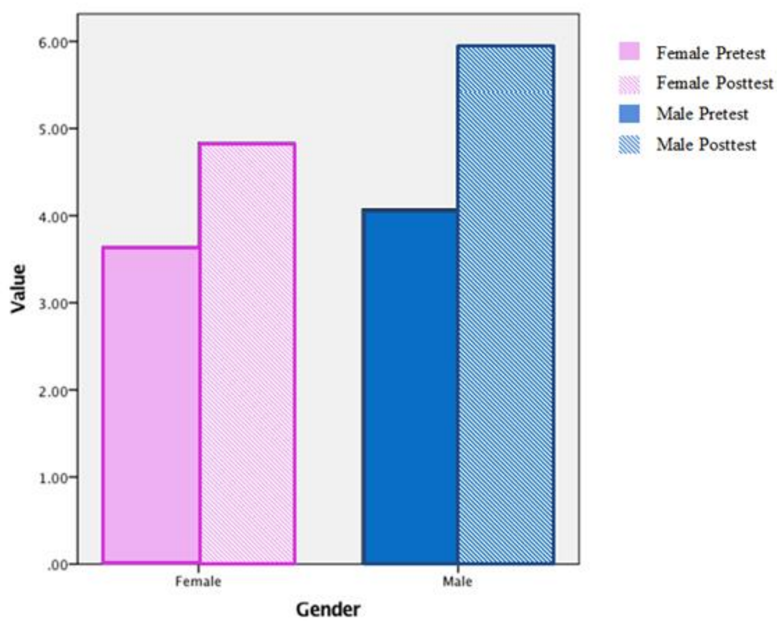
The result of the t-test analysis showed a significant difference between the pretest and the posttest scores for all students in $G_{1,2,3}$, $t(64) = -7.7$, $p = .000$. The percentage of the students who were able to answer half or more programming questions correctly doubled in the posttest. The programming knowledge had improved in $G_{1,2,3}$, and the mean value of the students' scores increased by 15% in the posttest.

The significant difference between the pre- and posttests scores suggested that the treatment was effective and increased the students' knowledge in programming. The null hypothesis $H2_o$ was rejected, and H2 was accepted.

H2A: Integrating art and animation in teaching programming increases <u>female</u> students' knowledge in programming language.

H2A$_o$: Integrating art and animation in teaching programming has no significant effect on <u>female</u> students' knowledge in programming language.

Figure 5-1 shows the mean values of the students' scores in the pretest and in the posttest for both genders. The mean values of the programming knowledge scores improved for all students in $G_{1,2,3}$ in the posttest. The mean values of the male students were higher than that of the female students in both tests. Also, the mean increased by 19% for the male students while it increased by 12% for the female students.



Gender	N	N%	Mean Difference %
Male	33	51%	19%
Female	32	49%	12%

Figure 5-1. The mean values of the students' scores in both tests.

The interesting finding that this study revealed was that the integration of art and animation with coding increased the knowledge of the male students more than the female students' knowledge. The result of the paired sample t-test was statistically significant for all students in $G_{1,2,3}$. It was also significant for the male and female students separately. For the female students it was $t(31) = -3.997$, $p = .000$, and since the mean value increased significantly in the posttest for female students, hypothesis H2A was accepted, and the null hypothesis was rejected.

H2B: Integrating art and animation in teaching programming increases the knowledge in programming language for students of different racial groups.

H2B$_o$: Integrating art and animation in teaching programming has no significant effect on knowledge in programming language for students of different racial groups.

The results were significant for most of the racial groups except the Native Hawaiian, American Indian, and Black students. The mean values improved in the posttest, and the mean differences were significant for the students of the following racial groups: Asian (p = .001, N = 19), Asian Indian (p = .005, N = 10), Middle Eastern (p = .003, N = 20), and White (p = .003, N = 7).

Hypothesis H2B was accepted for the Middle Eastern, Asian, Asian Indian, and White groups, while it was rejected for the Black and other racial groups.

The findings of this study did not necessarily mean that integrating art and animation with coding had no effect on the knowledge of Black and other racial groups that did have significant results. The results were not significant because some racial groups had low participation as compared to other groups. For better results, the study should be conducted by exposing larger numbers of Black and other racial groups to coding with art and animation workshop. This could be future work for this study.

5.1.3 Hypothesis H3: Real programming preference.

H3: Integration of art and animation in teaching programming increases high school students' preference to real programming language over block-based programming language.

$H3_o$: Integration of art and animation in teaching programming has no significant effect on high school students' preference of real programming language over block-based programming language.

The results were statistically significant for the students' preference for the real programming language over the block-based programming language. The results indicated that most of the students know the difference between the two languages, and they were ready to learn programming with real programming language. The results also showed that the use of art and animation made real programming easy to learn. In addition, the study found that high school students realized that they could do much more with real programming than they could with block-based programming language. This suggested that students know that their imagination and creativity are the only limit to what they can do with real programming. For high school students, this study recommended using real programming in events like the Hour of Code instead of the block-based programming currently offered by the event initiators' website Code.org. High school students are in the stage of making decisions about their university degree. The use of block-based programming may confuse a student with the fun and ease of programming, while in reality the student may face some difficulty in university courses where real programming is usually used. A student should have a clear idea about what programming is before making the decision about pursuing a degree in computer science. This study suggested utilizing block-based programming for elementary and middle school students but not for high school students.

The t-test results were significant for all students in $G_{1,2,3}$ with $p = .000$. For female and male students, the results were significant with $p = .000$, and $p = .009$, respectively. The results were significant for Asian ($p = .000$) and Middle Eastern ($p = .049$) groups. The result approached the borderline of significance for the Asian Indian ($p = .07$) and White ($p = .062$) students. The significant results indicated the improvement in the students' preference to the real programming language. Hypothesis H3 was accepted, and the null hypothesis was rejected.

5.1.4 Hypothesis H4: Motivation for code sharing.

H4: Integrating art and animation in teaching programming increases students' motivation to write and share more code.

$H4_o$: Integrating art and animation in teaching programming has no significant effect on students' motivation to write and share more code.

The effect of integrating art and animation with programming on students' motivation to write and share more code was measured by several survey items. These items measured variables of both intrinsic and extrinsic motivation like enjoying code sharing, getting more likes, feeling proud, competing with peers, and contributing and helping others.

The paired sample t-test result was significant for all students in $G_{1,2,3}$, $t(64) = -2.03$, $p = .046$. This indicated that the use of art and animation increased students' motivation in writing and sharing more code. Hypothesis H4 was accepted, and the null hypothesis was rejected.

5.1.5 Hypothesis H5: Programming enjoyment.

H5: Integrating art and animation increases students' interest and enjoyment in programming.

$H5_o$: Integrating art and animation has no significant effect on students' interest and enjoyment in programming.

Programming Interest and Enjoyment (PIE) was one of the variables that were measured in this study. The result was significant for this variable $t(64) = -2.599$, $p = .012$ for all students, and it was also significant for female students ($p = .001$). The use of art and animation increased students' interest and enjoyment. Hypothesis H5 was accepted, and the null hypothesis was rejected.

5.1.6 Hypothesis H6: Interest in CS course in high school.

H6: Integrating art and animation in teaching programming increases students' interest in taking a CS course in high school.

$H6_o$: Integrating art and animation in teaching programming has no significant effect on students' interest in taking a CS course in high school.

For this Course Interest (CI) study variable, the results of the paired sample t-test approached the border of significance for all the students ($p = .072$), and were significant for the female students $t(31) = -2.462$, $p = .020$. Male students had high interest in taking a CS course in both tests. The result was also significant for the students in G1who had more coding hours, $t(17) = -2.062$, $p = .05$. No significant result for any specific racial group was found. The hypothesis H6 was accepted, and the null hypothesis was rejected for the female students and for the students who had 15 hours of coding with art and animation.

5.1.7 CS Degree interest and other factors. Hypothesis H8 and hypothesis H9 are discussed in this section.

H8: There is a statistically significant relationship between high school students' interest in pursuing a CS degree and Programming Benefit and Enjoyment (PBE), Social Norm (SN), and Programming Capabilities and Confidence (PCC).

$H8_o$: There is no statistical significant relationship between high school students' interest in pursuing a CS degree and Programming Benefit and Enjoyment (PBE), Social Norm (SN), and Programming Capabilities and Confidence (PCC).

Correlation analysis was used to discuss this hypothesis. The results showed a significant strong positive relationship between the CS degree interest and the three factors suggested by the theory of planned behavior. Hypothesis H8 was accepted, and the null hypothesis was rejected.

H9: There is a significant prediction of high school students' interest in pursuing a CS degree by Programming Benefit and Enjoyment (PBE), Social Norm (SN), and Programming Capabilities and Confidence (PCC).

$H9_o$: There is no significant prediction of high school students' interest in pursuing a CS degree by Programming Benefit and Enjoyment (PBE), Social Norm (SN), and Programming Capabilities and Confidence (PCC).

Regression analysis revealed two models. The first model included one predictor, which was the PBE, and the second model included two variables, which were PBE and the

PCC. This indicated that if students enjoy programming and see benefit in it, then their interest in a CS degree would increase. Similarly, if students had confidence in their capabilities, accept challenges, and think they were capable of overcoming the difficulty, then this would increase their interest in pursuing this degree. The Social Norm (SN) variable was excluded from the model. Hypothesis H9 was accepted for two variables (PBE and PCC) and was rejected for the SN variable.

5.2 Research Questions

The five research questions were answered in the following five sub-sections:

5.2.1 Research Question One.

RQ1: What was the effect of integrating art, animation, and code sharing in teaching programming on the study variables for all students, for different genders, and for students of different racial groups?

As mentioned, the Code Genie development environment was used to provide the treatment for this experimental study, which was integrating art, animation, and code sharing in teaching computer programming.

The students' agreement percentages increased for all the study variables in the posttest, as shown in Figure 5-2. In the posttest, RPP had the highest agreement percentage (54%) followed by the DI variable with (52%). More than half of the students agreed on their preference to the real programming and on their CS degree interest. The agreement percentage for MCS was 49%, and it was 46% for PIE. The agreement percentages for those three variables were relatively high, or more than 40% of the students agreed on the art motivation for code sharing, the art and animation usefulness, and on their programming enjoyment. CI had the lowest agreement percentage (34%) among other variable or more than 30% of the students were interested in taking a CS course in high school. For the PK variable, the percentage of the students who were able to answer half or more questions in the posttest correctly was 66% while it was 31% in the pretest.

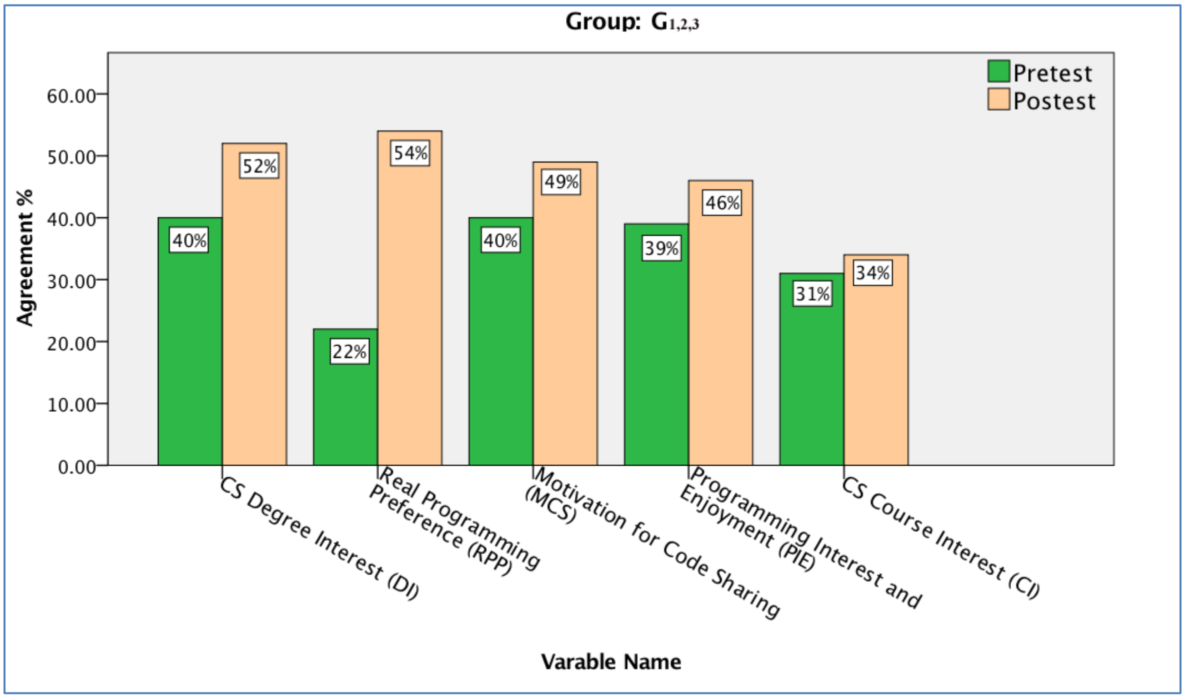


Figure 5-2. Agreement percentages for the study variables.

To answer this question, it is important to know the variables that had significant results for all students and for both genders. Table 5-1 shows the significant results for the study variables. The results for the paired sample t-test were significant for all students for most of the study variables and it was approaching the borderline of significance for the Course Interest (CI) variable.

For the female students, the results were significant for most of the variables and approached the borderline of significance for the Motivation for Code Sharing (MCS) variable. For the male students, the results were significant for only two variables, Programming Knowledge (PK) and Real Programming Preference (RPP).

Table 5-1

The T-Test Results for All Students and for Both Genders

Variables	All Students	Female	Male
DI	$t(64) = -2.68$, $p = .009$	$t(64) = -3.41$, $p = .002$	
PK	$t(64) = -7.70$, $p = .000$	$t(64) = -3.99$, $p = .000$	$t(64) = -7.27$, $p = .000$
RPP	$t(64) = -5.14$, $p = .000$	$t(64) = -4.73$, $p = .000$	$t(64) = -2.77$, $p = .009$
MCS	$t(64) = -2.03$, $p = .046$	$t(64) = -1.86$, $p = .071$	
PIE	$t(64) = -2.59$, $p = .012$	$t(64) = -3.67$, $p = .001$	
CI	$t(64) = -1.83$, $p = .072$	$t(64) = -2.46$, $p = .02$	

Table 5-2 summarizes the t-test results for different racial groups that have a p-value less than one. The Programming Knowledge (PK) variable had the most significant results for four racial groups. These were Asian, Asian Indian, Middle Eastern, and White groups. The Real Programming Preference (RPP) variable was significant for two racial groups which were the Asian and Middle Eastern groups, and it was approaching the borderline of significance for the Asian Indian group. The White group was the only group who had significant results for the Degree Interest (DI) variable.

For the Motivation for Code Sharing (MCS) variable, the results for the Asian students had a certain trend toward significance. The result for the Programming Interest and Enjoyment (PIE) was quasi-significant for the Asian group. The Asian group had the most significant results among other racial groups.

Table 5-2

The T-Test Results for Different Racial Groups

	Asian	Asian Indian	Middle Eastern	White
DI				$t(6) = -2.5$ $p = .047$
PK	$t(18) = -4.03$ $p = .001$	$t(9) = -3.63$ $p = .005$	$t(19) = -3.43$ $p = .003$	$t(6) = -4.87$ $p = .003$
RPP	$t(18) = -4.80$ $p = .000$	$t(9) = -2.05$ $p = .07$	$t(19) = -2.1$ $p = .049$	
MCS	$t(18) = -1.83$ $p = .08$			
PIE	$t(18) = -1.77$ $p = .093$			

To answer the research question, the effect of the treatment was increasing student's interest in pursuing a degree in CS. It also increased their programming knowledge, their preference to real programming, their motivation to write and share more code, and their programming interest and enjoyment. In addition to the mentioned effect, the treatment also increased the female students' interest in taking programming courses in high school, and female students found that the use of art and animation in programming was useful. For the male students, the treatment increased their programming knowledge and their preference to code with real programming language. For the treatment effect on students of different racial groups, the study found that integrating art and animation in teaching programming increased the programming knowledge for the Asian, Indian Asian, Middle Eastern and White students. The treatment also increased the Asian and Middle Eastern students' preference to real programming language, and it also increased the interest of White students in a CS degree.

5.2.2 Research Question Two.

RQ2: Was there any difference between the results of students with different amount of coding time?

To answer the second research question, it is important to discuss the coding time effect on different variables for G3 and $G_{1,2}$. The agreement percentages and the mean differences between the pre- and posttest results for all study variables were affected by the coding time. Most of the study variables were measured through a 5-point Likert scale except the Programming Knowledge variable, which was measured by programming scores. The coding time effect will be discussed for the six-study variables and for the programming knowledge variable in the following two sections.

5.2.2.1 Study variables. Figure 5-3 shows a difference between G3 and $G_{1,2}$ agreement percentages of the six study variables for all students, and Figure 5-5 shows the differences between the two genders.

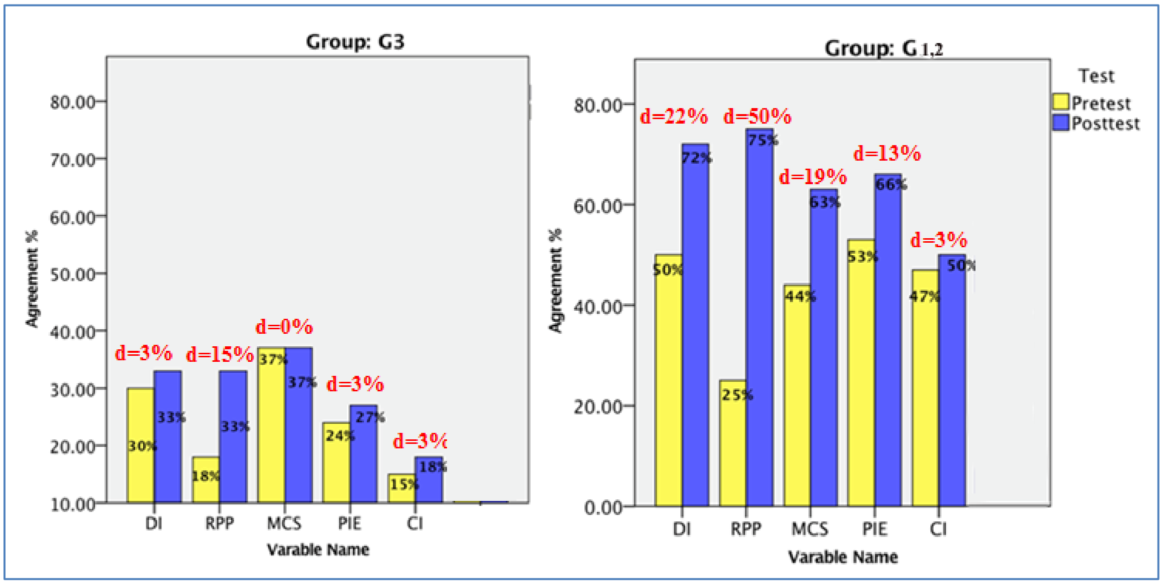


Figure 5-3. G3 vs. $G_{1,2}$ agreement % of the study variables for all students.

The agreement percentages for all variables in $G_{1,2}$ were higher than those in G3, and they increased for all variables in the posttest in G3, where it dropped by 19%. The differences between agreement percentages in the pretest and posttest were higher in $G_{1,2}$. In other words, the responses for the students with more coding time improved more than that for the students with less coding time in the posttest. The improvement ranged from 3% and 50% in $G_{1,2}$, while it ranged from 3% to 15%, in G3. The improvement of the agreement percentage for the RPP variable in the posttest was the highest among the other variables' improvement in G3 and $G_{1,2}$.

The agreement percentages improved in the posttest for all variables and for both genders in $G_{1,2}$. In G3, the agreement percentages improved or remained unchanged for the female students, while they remained unchanged or decreased for the male students, as shown in Figure 5-4 .

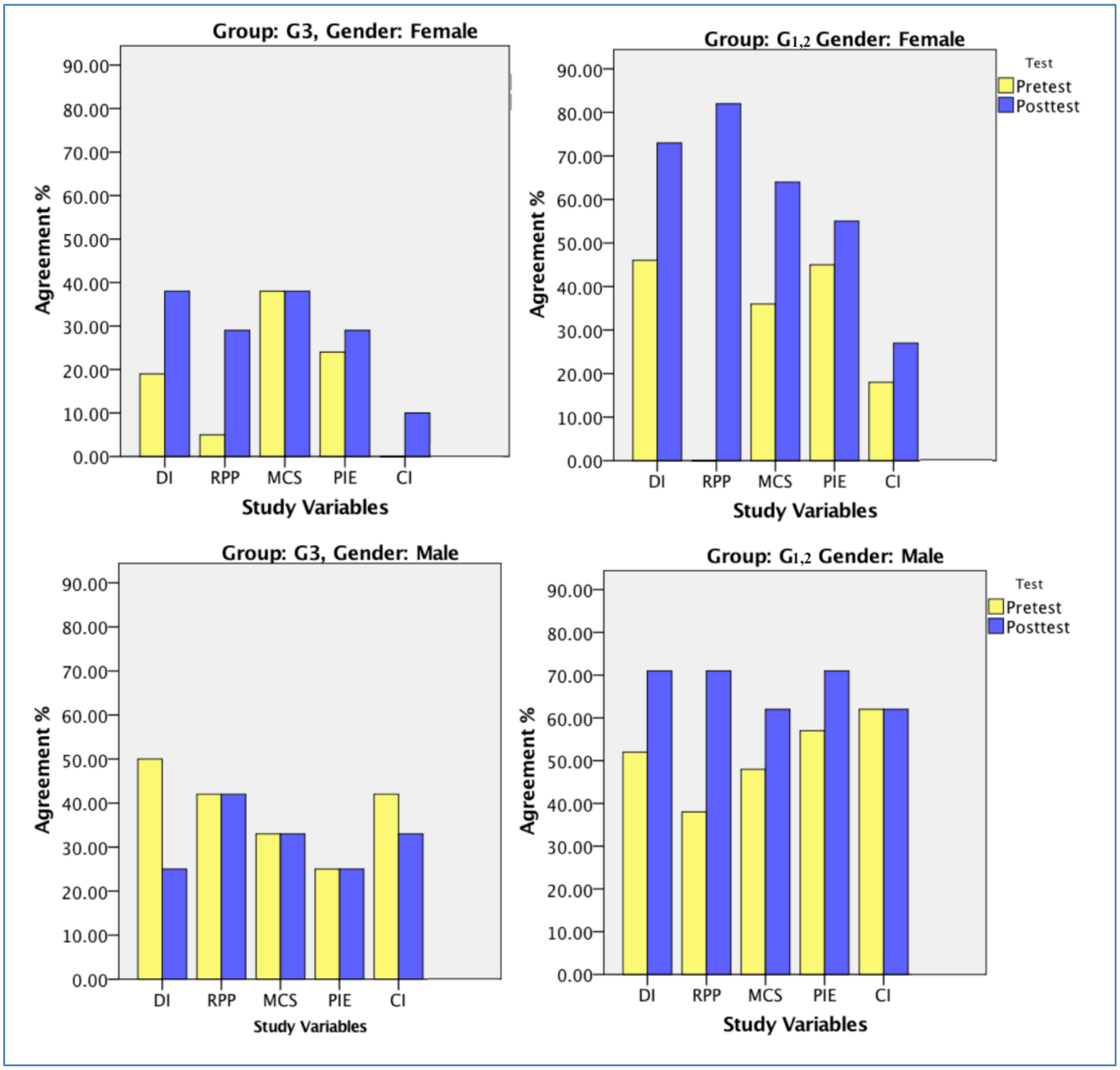


Figure 5-4. G3 vs. $G_{1,2}$ agreement % of the study variables vs. genders.

In the pretest, the agreement percentage of the female students was zero for interest in taking a CS course in high school (CI) in G3, and it rose to 10% in the posttest. Similarly, the agreement percentage for the female students was zero for the real programming preference (RPP) in $G_{1,2}$, and it rose to 82% in the posttest. This finding was interesting since the female students were able to program with the text-based language, or real programming language, and preferred it over the block-based programming language. In $G_{1,2}$, the female agreement percentages in the posttest were higher than the male percentages for all variables except the programming enjoyment (PIE) and CS course interest (CI) variables, where males had higher percentages. In G3, the female percentages in the posttest were also higher than the male percentages except for the real programming preference (RPP) and CS course interest (CI) variables. Male students had higher interest in taking a CS course in high school than female students.

As shown in Table 4-47 in the previous chapter, the results of paired sample t-test analysis for all students in G3 were statistically significant for only one variable among the six other variables. This was the real programming preference RPP variable. For the female students in G3, the results were significant for two variables, the CS degree interest (DI), and the RPP variable, while no result was significant for the male students in G3.

While the result for only one variable was significant for all students in G3, four variables had significant results for all students in $G_{1,2}$. These were DI, RPP, MCS, and PIE. For the female students in $G_{1,2}$, the results were also significant for the same four variables, while only one variable had significant results for the male students in $G_{1,2}$ which was the RPP.

5.2.2.2 Programming knowledge. The workshop coding time for the three groups was sufficient to explain the main programming concepts that were measured in this study. However, students with more coding time were able to write more coding examples and share more artwork.

The mean value of the students' Programming Knowledge (PK) variable increment in $G_{1,2}$ was double its increment in G3. It increased by 20% in $G_{1,2}$, while it increased by 10% in G3 for all students in the two groups (see Figure 5-5). For male students, the mean value increased by 13% in G3 and by 22% in $G_{1,2}$. For the female students, it increased by 8% in

G3 and by 18% for females of $G_{1,2}$. The mean value for the male students in G3 was higher than the mean value of the male students in $G_{1,2}$ despite the shorter coding time. This indicated that the effect of coding time on the male students' responses for the programming knowledge variable was less than its effect on the other variables discussed in the previous section. However, female students in $G_{1,2}$ had higher mean values that the females in G3 for both tests.

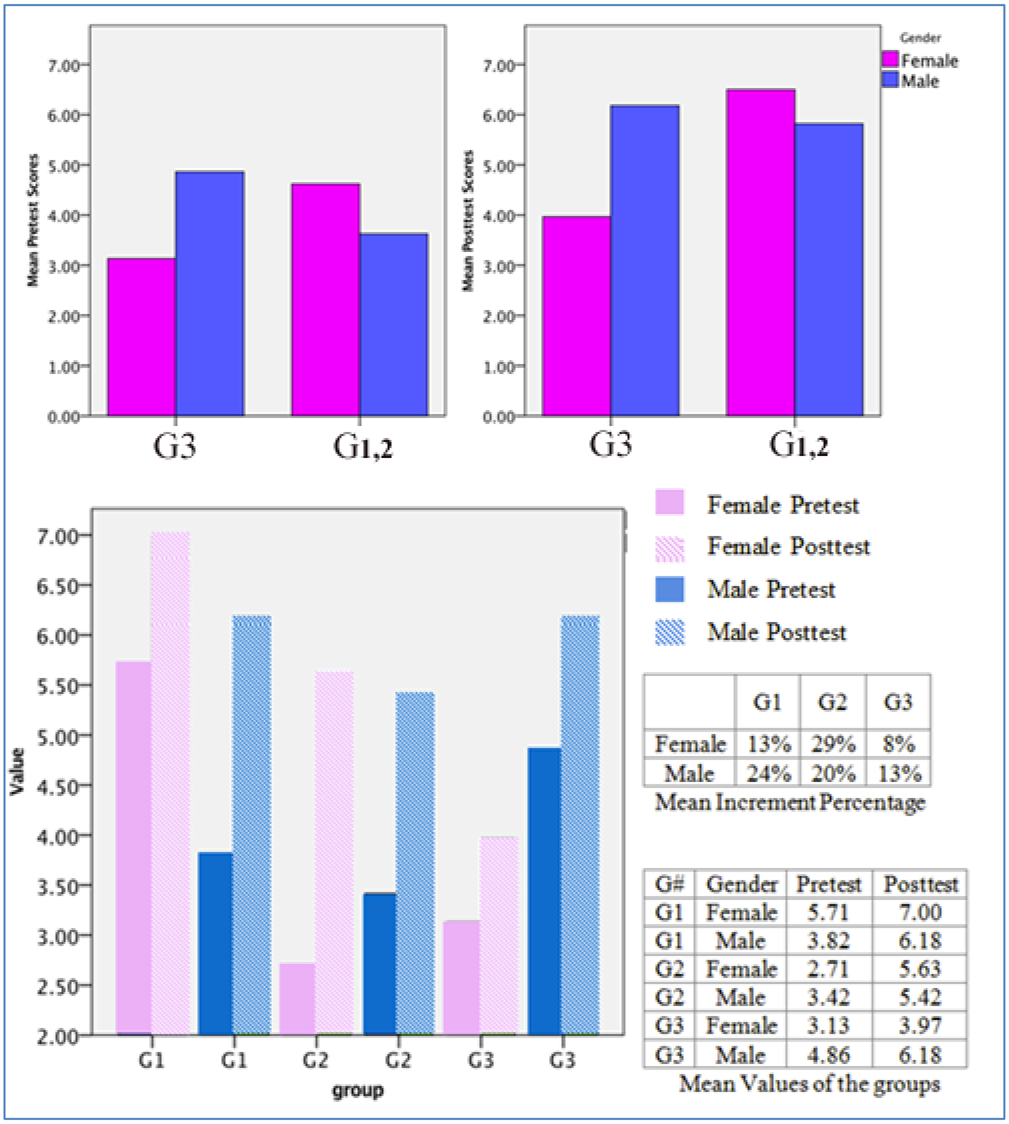


Figure 5-5. Gender vs. the mean of the students scores for the three groups.

The improvement of the PK variable was not equal among the three groups. To compare between the three groups, G2 had the highest increment in the mean value, which was 20%. The mean value increment was 19% in G1 and 10% in G3. Figure 5-6 shows the mean increment percentage of the study variables for different groups. The descriptive statistics show that the scores of the female students were better than the scores of the male students in G1, while they were better in male than female students in G3. The improvement in female students of G2 was the best among the other groups (29%) followed by the improvement in the male students of G1, which was 24%. The improvement in the mean value of the female group of G3 (8%) was the lowest among other groups. Similarly, the improvement in the scores of the females of G1 (13%) was also small compared to other groups.

The female students in G1 had the highest mean value among other groups. The mean values for the male students in G3 and G1 were the same (6.18) despite the different coding time.

The results of the paired sample t-test analysis for the PK variable were significant for both G3 and $G_{1,2}$ and for both genders in each group. This indicated that the coding time had less effect on the PK variable than its effect on the other study variables discussed in the previous section.

To answer the Research Question Two, there wasn't any difference for the students' programming knowledge and the students' preference to the real language variables between the students who had longer ($G_{1,2}$) and shorter (G3) coding time. The results were both significant for G3 and $G_{1,2}$ for these two variables.

The results were different for the other study variables. The results were significant for three more variables for the students who had longer coding time, while the results were not significant for students with shorter coding time. These variables showed interest in a CS degree, motivation for code sharing, and programming interest and enjoyment. For the female students in both groups, the results were close and significant for most of the variables except for the motivation for code sharing in G3 and the CS course interest in high school in $G_{1,2}$. For the male students in both groups, the result was different for real programming preference, which was significant in $G_{1,2}$ only. For the art and animation usefulness, the results were similar and not significant for both groups.

5.2.3 Research Question Three.

RQ3: For high school students, which programming concept was easy, which was difficult, and which concept had the best improvement in the posttest?

The students' understanding of the different programming concepts was improved for all programming questions in the posttest, as shown in Figure 5-6.

The easiest programming concepts were the "Variable Assignment" which was measured by Q1, and the "Variable Multiplication," which was measured by Q3. Ninety-seven percent of students were able to answer these two questions correctly in the posttest. The most difficult programming concept was the "for-loop," which was measured by Q4. The percentage of the students who were able to answer that question correctly doubled in the posttest.

However, it was the lowest percentage among the percentages for the other variables. The array question, Q11, had the best improvement percentage in the posttest. The difference between the pretest and posttest percentages was 28%, which was the highest compared to the differences for the other programming questions. In contrast, the improvement for Q5, which measured the students' understanding of the "if-statement" programming concept, was 2%, which was the lowest percentage improvement.

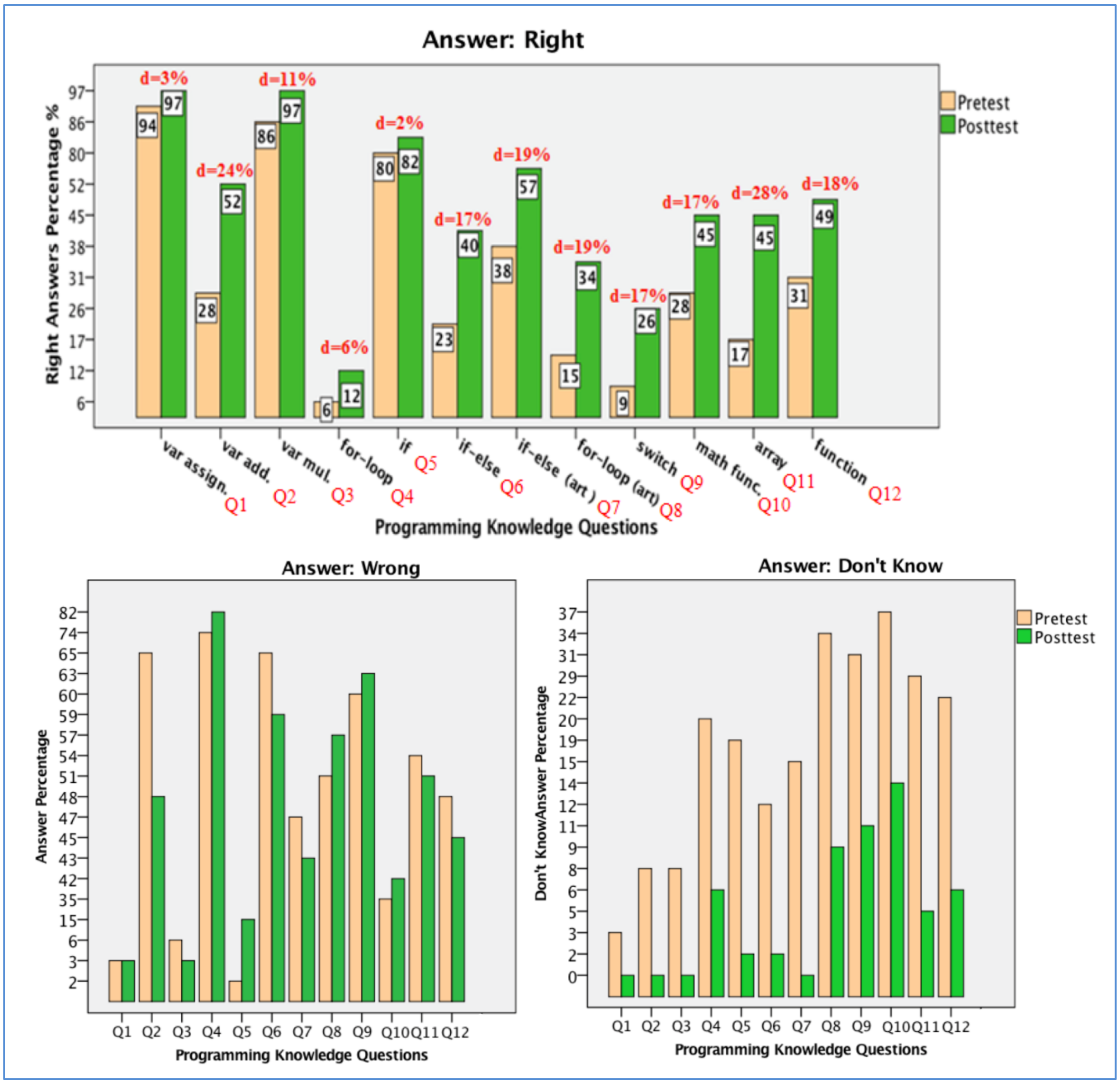


Figure 5-6. Answers for the different programming concepts.

The "Don't Know" answers decreased in the posttest, which indicated that students became more confident in answering the programming questions (Al-bow et al., 2009).

5.2.4 Research Question Four.

RQ4: Was the Code Genie tool useful and easy to use?

The mean values of the students' responses for the Ease of Use (EOU) and Usefulness variables were above the neutral option. The EOU mean value was 3.38, and it was 3.68 for Usefulness. The majority of the students agreed on the tool usefulness (58%), while only 17% of the students disagreed. Forty-three percent of the students agreed that the tool was easy to use, 20% disagreed, and 37% chose the *neutral* option. The students who agreed on the EOU were double those who disagreed.

5.2.5 Research Question Five.

RQ5: Was integrating art and animation in teaching text-based programming useful for high school students in understanding math functions, increasing their creativity and their programming skills?

Section 4.9 discusses the results for the AAU variable, which was used to answer this research question. The descriptive statistics showed that 50% of all students in $G_{1,2,3}$ agreed on the usefulness of the art and animation in teaching CS. The disagreement percentage was 16%. This indicated that most of the students agreed on the usefulness of art and animation in understanding math functions, increasing their creativity and enhancing their programming skills. The agreement percentage for the female students (53%) was higher than the male students (46%) in $G_{1,2,3}$. Asian (79%), and White (71%) students had the highest agreement percentages among the other racial groups in $G_{1,2,3}$. Female students had a higher agreement percentage in G1 (86%) and G3 (38%), while the male agreement percentage was higher in G2(90%). The coding hours had a significant effect on the AAU variable. The difference in the agreement percentage between G3 and $G_{1,2}$ was very high. The agreement percentage was

72% in $G_{1,2}$ and 27% in G3. However, the neutral percentage for the G3 was 46%. In comparing between the three groups, G2 had the highest agreement percentage (86%) and G3 had the lowest (27%). G1 agreement percentage was also high (61%). This suggested that students who had more coding time found integrating art and animation useful in teaching programming more than students who had less coding hours.

5.2.6 Research Questions Six and Seven.

RQ6: From the students' participation in the coding workshops, was there any difference in the students' interest to participate between different genders, and was there any difference among students of different racial groups?

Statistics from the big tech companies like Google and Apple showed that tech jobs are dominated by White and Asian males. The male to female percentage ratio was usually 80:20 in these companies (Naughton, 2017; Apple, 2017). As mentioned before, $G_{1,2}$ (G1 and G2) represented the students who attended the coding camp/workshop upon their own interest. Table 5-3 summarizes the demographic data of $G_{1,2}$. The percentage of the female students was 34%, and it was 66% for the male students. As compared to the male-dominated tech jobs, this female percentage was not small, and this indicated that female students were also interested in computer programming.

Table 5-3
$G_{1,2}$ Demographic Data

	G1 , N=18		G2, N=14		$G_{1,2}$ (G1+G2), N=32	
	N	%	N	%	N	%
Female	7	39%	4	29%	11	34%
Male	11	61%	10	71%	21	66%
American Indian	0	0%	1	7%	1	3.1%
Asian	9	50%	6	43%	15	47%
Asian Indian	5	28%	1	7%	6	19%
Black	1	5.5%	1	7%	2	6%
Hawaiian	0	0%	2	14%	2	6%
Middle Eastern	1	5.5%	0	0%	1	3%
White	2	11%	3	22%	5	16%

The largest percentage among racial groups in $G_{1,2}$ was the Asian group (47%) followed by the Asian Indian group (19%). The White group came into third place with 17% attendance. The participation of Black and other racial groups was low while the participation of Asian and Asian Indian groups dominated. This study found that the Asian group for both genders was more interested in coding than other racial groups.

One of the observations was the lack of programming interest among Black students. Although Pioneer High School's demographic data indicated 14% of the students were Black or African American (Pioneer High School, 2014), a visitor to the school can see a higher percentage of Black students, both male and female. However, no Black female students were interested in the coding camp or workshop, while a very small percentage of Black male students were interested in coding. The researcher focused on informing some of the Black females and asked them in person to participate before the workshop started, but no Black females showed any interest in participation. Similarly, White female participation was also low; only one White female participated in the fall workshop. Most of the female students who attended the coding workshop were Asian and Asian Indian. Their programming scores were also high. The study has found that females of these two racial groups are more interested in programming than the other racial groups at Pioneer High School.

The researcher was asked by Pioneer High School administration to teach using the same Code Genie tool during the Computer Science Education Week (CSEdWeek) (Computer Science Education Week, n.d.), which was the first week of December, 2017. During that week, 750 students were exposed to coding with art and animation for one hour per group. No data were collected, but the researcher observed that most students' engagement and interest was very high, and they were creative in producing and sharing the artwork that they created with coding. Comparing this with the summer camp participation, which was dominated by Asian and Asian Indian groups, the CSEdWeek revealed high interest by Black students of both genders and White female students.

The study suggested exposing a larger number of students to coding with art and animation as a school activity whether they are interested or not, similar to what this study did with G3. Students may have an interest in programming, but they may not be aware of their interest. As a future study, a summer camp or an after-school workshop could be offered again for the students who were exposed to the coding workshop to find if the interest in participation will change or not.

RQ7: From the students' participation in the coding workshops, what was the percentage of the high school students who were interested in a free coding workshop?

The study found that only a small percentage of the high school students were interested in computer programming and ready to attend a free coding summer camp or an after-school workshop. The camp and workshop emails were sent several times through the school messenger. The school has more than 1,600 students, but only a small percentage of students showed an interest in participation. There were only 32 students who were interested in learning programming for free in the summer and during the fall workshop. This means only 2% of the students were interested in attending the free coding workshop/camp at Pioneer High School.

The transportation commitment could be one of the reasons for relatively low enrollment in the summer camp. Parents needed to provide transportation for their children in order to attend the coding camp. However, it was a free coding camp and many parents usually register their children for similar activities in summer. In addition, the fall workshop was an after-school activity and the students were already in the school, but the enrollment was also low as compared to the school size. This indicated that few high school students are interested in coding or have other activities to attend such as sports or band.

This lack of interest agrees with findings of other studies (Kessler, 2017; Snyder, Brey & Dillow, 2016), and it is expected to continue in the future unless some solution will be proposed to increase students' interest in coding. This was one of the main objectives of this study.

5.3 Students' Comments and Engagement

This section sheds light on the students' engagement in the coding workshops especially for underrepresented groups. This section also discusses students' comments and feedback on their coding experience.

In the two fall workshops (G2 and G3), students got the same amount of programming information to measure their programming knowledge using the same teaching materials that were used in the summer camp (G1). However, for the other coding examples, material was more condensed for students in G2 and G3. Also, they had less time to cooperate and share their coding artwork. Their engagement could be relatively less than students' engagement in the summer camp. However, G2 and G3 coding engagement was not low. As shown in Table 4-14 in the previous chapter, the number of artworks that were shared by G1 was 128. It was 22 artworks by students in G2, and 60 artworks by students in G3. The total number of code lines that were written by the three groups was 6,628 lines of code. This indicated sufficient students' coding engagement.

The posttest survey questionnaire had a free-field comment where students could express their general experience in the coding workshop and add their suggestions for the used tool. As mentioned before, 65 students participated in the study, 32 females and 33 males. Before discussing the students' comments, it is important to know the number of students in each racial group as shown in Table 5-4.

Table 5-4
Students Distribution among Different Racial Groups

	Female	Male	All students
American Indian	0	1	1
Asian	11	8	19
Asian Indian	2	8	10
Black	3	3	6
Middle Eastern	13	7	20
Hawaiian	2	0	2
White	1	6	7

5.3.1 Female students' comments. Most of the female participants were Asian and Middle Eastern in this study. Their comments and the comments of females from other racial groups are demonstrated in this section. The Black students' comments, males and females, are discussed in the next section.

As mentioned before, the White female participation was very low. There were not any White females in G1 and G3, and there was only one White female in G2. However, this White female student (FStD) showed high interest in coding by sharing her artworks and by the comment she made at the end of the workshop where she said,

> I think that learning code through art really sparked my interest in coding. My dad is a computer programmer and I thought computer programming was boring before this workshop. Now that I know that art can also be involved, I think I will be more apt to pick this field. (student FStD, 2017)

The researcher thought this comment was interesting and met the goal of this study and the goal of the development tool that was built as a treatment for this experimental study.

Two Hawaiian females participated in the fall workshop (G2). Both said, "It was fun." One liked the tool as it is, and the other suggested it should "have more labels so everything can be found faster and easier."

Other female students had different suggestions. An Asian female suggested adding an "Undo" button. Student SStG, who was Asian Indian, suggested adding the "Save" button. The "control +z" works as the undo in the tool; however, the "Undo," "Redo" buttons could also be added to the tool. To add the "Save" button, a login account should be added for each student. To keep the tool simple and ready for immediate use, a user account is not currently required. However, this feature could also be added to the tool. Student SStb suggested adding more shapes and an image library where an image could be inserted to the artwork. Student SStc suggested adding a non-equilateral triangle and pentagon. Students FStC, FStD, MStC, MstI, Mstk, and MstM suggested adding tutorial pages and more explanation, accessing lessons in a more user-friendly manner, organizing the templates in a sequence, making the process easier, and adding more activities.

Several female students liked the tool as is, and they had no suggestions. Many females indicated that the workshop was fun, educational, and gave a chance for collaboration. Student SStG said, "The camp was a lot of fun, and I like that everyone was able to save and share their code. It helped everyone else learn from what one person did and make their own coding skills better." Student MStH said, "It was Great and Wonderful." The word "fun" repeated ten times, "nice" six times, "cool" and "enjoyable" four times, and "interesting" three times in different comments for the female students.

A number of female students indicated that they learned new things. Student SStb said, "Thank you for hosting this camp! I enjoyed it a lot and I learned a lot of new things I didn't know before." Student MStO said, "learned new things," and student MstP said, "It was interesting and something new that I did not try before." Student MstU said, "Very educational, I learned a lot." Student SStF shared a similar comment: "It was a lot of fun and I learned a lot." The word "learned" was repeated nine times in nine different comments.

Students MStJ said that she preferred the block-programming, and one student said, "I don't know."

5.3.2 Black students' comments. Three Black males and three Black females participated in this study. By the beginning of the summer camp, there were two Black male students; one left the camp on the fourth day and the other was somewhat engaged but did not share as much artwork as the other students, even though there was enough time to share. He also indicated that his parents wanted him to attend this camp. However, his comment revealed some interest in coding where he suggested to add procedures on how to code, and he indicated that the camp experience was "All Right," which showed some level of coding satisfaction.

On the other hand, although there was only one Black student in the fall workshop, this student showed high interest in coding, and he was very creative in modifying and sharing the existing artwork. He also developed and shared his own artwork despite the limited time of the fall workshop. In his comment, he said "It was fun," which indicates a sufficient level of satisfaction. He also said that he had no suggestion for the tool and he liked it as is.

The third Black male student in G3 who was exposed to the coding workshop said "it was nice, but the tool was pretty complex." This suggested that the student liked the coding experience, but he faced some difficulty using the development environment.

No Black females participated upon their own interest in G1 or G2; however, three Black females were exposed to the coding workshop in G3. One said, "I thought it was fun and interesting workshop and the tool was good." The other student said, "It was a nice experience and I had a fun time doing this workshop."

The overall experience of the Black students and their feedback is considered positive in general. However, to be able to generalize this positive feedback, the study will need to be conducted with a larger number of Black participants in the future.

5.3.3 Male Students' Comments. Most of the White male students liked the workshop. Student FStE said, "I liked it a lot and it made it easier to get better at coding while still learning," and Student MStv said, "It was an interesting learning experience." Student FStF suggested adding more shapes.

Most of the Asian students enjoyed the workshop. Student SStl, who shared many artworks, said, "This was a very fun experience." Another student, SStR, suggested making the tool more professional. He said, "I think Code Genie can be more like an environment that more professional [sic], like Apple PlayGround. This camp let me know how to use code to make art." Student FStN said,

> Code Genie allows the students to make animations, which improves the ability of creating things of the student. I really liked this workshop. If there is a similar workshop about programming in the future, I'll try my best to participate in it. The teacher explained every topic thoroughly. (student FStN, 2017)

Student FStJ suggested adding the lessons page or tab when he said, "The tool was nice but if there was a lesson plan tabs that could allow us to more easily access other statements or functions. I liked the workshop and I think it may help me in the future."

The Asian Indian male students shared the workshop enjoyment. Student SStP said, " I loved this camp and will come back next year!" Student MStZF stated a similar comment, "This was a fun workshop. I'm motivated to go to another one." He also shared the suggestion of adding more shapes and different orientation when he said, "There ought to be more ways to express different designs as in different shapes with more orientations." Student SStJ said, "I learned a lot."

Most of the Middle Eastern male students indicated that the workshop was a good experience. Student MStZD said, "It was a good learning experience." Two students suggested adding a "Help" tab and one student suggested adding more colors. The American Indian student had no comments; however, he shared several artworks during the workshop.

5.4 Discussion and Conclusion

Many students face difficulty in computer programming courses (Lahtinen, Ala-Mutka, & Järvinen, 2005), and many educators are trying to make it easier by providing user friendly development environments. This study introduced a new development environment that integrated art and animation in learning text-based programming language. The new developed environment was used as a treatment in this experimental study, and different study variables were used to measure the effect of using art, animation, and code sharing on students' interest in programming. This study found that integrating art and animation increased interest in pursuing a degree in CS. It also found that it increased students' programming knowledge. This finding agreed with the study findings of Al-bow et al. (2009)

where the researchers found that the Greenfoot coding environment increased student programming knowledge. The Greenfoot learning environment was developed to increase the interest in learning Java object-oriented language by adding animation elements (Kölling, 2012). The difference between this dissertation study and Al-bow et al. study was the programming language and the coding environment.

Many educators encourage block-based language for easier programming experience. Block-based programming language is fun and easy, and it encourages students with no programming background to start coding. However, it is not a real programming language. As high school students near university admission, they should begin to use industry-based programming language or face frustration in their first university computer science course. Students who code with real programming language will not be surprised when they have to write a program in a university class. Moreover, by writing a program with standard programming languages like JavaScript, Java, C++, etc., a student can communicate with a larger community for information exchange and can create a real software product. The other finding of this dissertation study was that the use of art and animation increased students' preference to the real programming language that is also a text-based programming language. This study finding agrees with DiSalvo's (2014) study where students preferred the text-based language, Jython, which is a version of Python language, over the block-based programming language, Alice. In contrast, Weintrop and Wilensky (2015a) found that high school students preferred the Snap block-based programming over the Java language. Java object-oriented language and Snap block-based programming have two different levels of difficulties. Block-based programming is usually easier than object-oriented programming. This could explain the students' preference in the Weintrop and Wilensky (2015a) study.

In a survey of 64,000 programmers, 48% indicated that they program as a hobby (StackOverflow, 2017). Many developers like to contribute in software fields by sharing their code at code sharing websites such as GitHub, CodeShare, and JSFiddle (Uzayr, 2016). One of findings of this dissertation study was that the use of art and animation increased students' motivation to write and share code and increased their interest and enjoyment in programming. By sharing their code, students were able to collaborate by adding their code together to create one artwork. The "Like" button in the Code Genie development environment created an extrinsic motivation for the students to compete and get more likes on their artwork. According to Griffin (2006), competition between students was one of the motivations for high academic achievement among Black students. The students' programming enjoyment, their desire to contribute and share their code with others, and their acceptance of the challenges were the intrinsic motivations. The motivation for the code sharing variable included both intrinsic and extrinsic motivation.

Carter's (2006) study found that some high school students do not choose the CS major because they do not know what is it, or they have an incorrect perception of what computer scientists do. The study suggested that exposing high school students to computer science courses in the high school could increase the enrolment in this major. According to Morgan and Klaric (2007), female students who had computer science courses in high school were 10 times more likely to major in CS in the university. The study also found that underrepresented racial groups, like Black and Hispanic, were seven times more likely to major in computer science if they had CS courses in high school. The results of this dissertation study showed that female students' interest in taking a CS course in high school increased significantly by the end of the workshops, while the male students' interest, which was three times higher than the female interest, did not change significantly. This finding agreed with the Al-bow et al. (2009) study, where the number of female students who were interested in a computer science major increased in the posttest.

The coding time had an effect on the students' responses for all variables. In the posttest, the agreement percentages of the students with longer coding time were higher than the percentages of the students with shorter coding time, and the results were more significant for the measured seven variables. This indicated that more coding time increased students' motivation for computer programming. The study itself was one of the few studies that explained the difference in results caused by different coding time.

The students' participation in this dissertation study showed that the students who participated in the coding workshops upon their own interest were mostly Asian and Indian Asian of both genders. Statistics from large companies such as Google, Apple, and Facebook showed that tech jobs were dominated by White and Asian males (Naughton, 2017; Apple, 2017; Williams, 2017). The Asian and Indian Asian students' participation agreed with those statistics; however, this study found that White students' participation was relatively low.

Among the three factors that were suggested by the theory of planned behavior (Ajzen, 1991), Programming Benefit and Enjoyment (PBE) and Programming Capabilities and Confidence (PCC) were significant predictors for students' interest in pursuing a degree in CS. In other words, if students found a benefit and enjoyment in programming, and if they were confident that they could overcome the programming challenges, then their interest in a CS degree would increase. Bandura's (1997) study stated that more confident people put more effort into performing a new task than less confident people. The third predictor, which was the social norms, was excluded from the regression analysis. However, a positive and strong relation was found between CS degree interest and the programming social norms variable, which implied that parent and relative encouragement and support for pursuing a degree in CS could increase a student's interest in this major.

The Ease of Use and Usefulness variables that were suggested by the technology acceptance model or TAM (Davis, Bagozzi, & Warshaw, 1989) were used to test the Code Genie usability. This model was widely used to test the usability of any new system or tool. The results revealed that the majority of the students in this experimental study found that the tool was easy to use and useful.

Code Genie was different than the other tools by providing a free online IDE that focused on encouraging students to program with a real programming language or text-based language. For example, the Pencil Code development environment had both block-based and text-based programming modes (Bau et al., 2015). However, its default and main interface or mode was the block mode, and a user of the text mode should switch to the block mode to get a new keyword. Moreover, the Pencil Code supported the "Syntax Highlighting" feature, but the function names were abbreviated and not meaningful which could have had a load on the short-term memory and decreased the code readability.

Code Genie text editor supported the "Syntax Highlighting" feature and the tool used meaningful function names. These two features increased the code readability. Syntax coloring was an important feature that was available in several programming editors or integrated development environment (IDE). Sarkar (2015) found that this feature increased program readability, and it decreased the time needed to read and understand the code (Sarkar, 2015). Code readability was one of the important features in the software source code. A study found that readability had an impact on software quality, such as reusability, maintainability, reliability, complexity, and portability (Tashtoush, Odat, Alsmadi, & Yatim, 2013).

The other available programming environment for K-12 students that supported real programming was the Greenfoot environment (Kölling, 2012). This environment encouraged students to program with Java language. The tool was free and open sourced; however, it was not available online and instructions should be followed to install and use the tool. The researcher of this dissertation study thought Java was not an easy language for beginners, and it was more suitable for students in university courses or high school students in the advanced or at least intermediate level. JavaScript was chosen as the real, or text-based, programming language for the Code Genie online IDE for its simplicity. Statistics showed that JavaScript was one of most popular industry-based languages and has a large community (O'Grady, 2018; StackOverflow, 2017).

Students' comments by the end of the coding workshops indicated high interest in coding, and several students stated that they would attend similar workshops again if they were available in the future.

5.5 Future Work

After conducting this study, the researcher knew that a new mobile development programming course would be available at Pioneer High School for the next school year (2018-2019). This could be a good opportunity for conducting a similar study to explore the effect of mobile development, which usually involves art and design, on students' interest in a CS degree. Also, data could be collected from the students who participated in this study after two years to find if they chose the CS field after their graduation or not. The researcher should have the emails of the participants who will be adults after two years, and no parental approvals will be needed to collect data.

The study could be conducted again with a different sample such as exposing a larger number of Black students or other underrepresented groups to coding a workshop to find if the results will be different. College students or middle school students could be also targeted as a suggested sample for future studies. Interviews could also be added to the research design to collect more data and measure the students' acceptance to the tool and the treatment. Interviews could include students' suggestions to improve the tool or more ideas for the programming template that could be added to the tool.

The students' artwork that was produced by coding could be analyzed qualitatively in a future research study. The students' artworks that were produced by code had different coding styles depending on their programming experience. Students with more experience were able to write more structured code while the beginners' code was less structured or what is known as "Spaghetti Code." "Skill Level" and "Creativity Level" measurements could be developed and used to classify and analyze the students' artworks. The qualitative analysis of the students' code may or may not reveal different styles, skill levels, and creativity levels for different genders or even race.

The Code Genie tool could be improved by adding tutorial pages, and its user interface could be more interactive. Gamification rewarding and elements could be added such as badges, points, or scores. Usability could be performed through different testing method recommended by the human-computer interface experts such as a field observation test, a "Thinking Aloud" test, or a questionnaire designed for the tool usability (Holzinger, 2005). In the usability test, users or students are asked to perform different tasks, and then their activities are observed. They are also asked to think aloud about the sequence of actions that they intend to do to perform a specific task. Future work could be a combination research in the areas of CS, human-computer interface (HCI), STEM, and integrated development environment (IDE) for K-12.

5.6 Summary

This chapter tested the hypotheses in section 5.1 and answered the research questions of this study in section 5.2. Students' comments and engagement were discussed in section 5.3 in this chapter. The chapter compared the findings of this study with the finding of other studies in the "Discussion and Conclusion" section. Finally, the chapter ends with future work and what could be done after this dissertation study.

References

Ahmad, K. N. (2012). *Measuring the impact of app inventor for android and studio-based learning in an introductory computer science course for non-majors* (Doctoral dissertation, Ball State University). Retrieved from http://ezproxy.emich.edu/login?url=http://search.proquest.com/docview/1035149298?accountid=10650

Ajzen, I. (1991). The theory of planned behavior. *Organizational Behavior and Human Decision Processes, 50*(2), 179-211.

Al-Bow, M., Austin, D., Edgington, J., Fajardo, R., Fishburn, J., Lara, C., ... & Meyer, S. (2009). Using game creation for teaching computer programming to high school students and teachers. *ACM SIGCSE Bulletin*, *41*(3), 104-108.

Al-Bow, M., Austin, D., Edgington, J., Fajardo, R., Fishburn, J., Lara, C., ... & Meyer, S. (2008, August). Using Greenfoot and games to teach rising 9th and 10th grade novice programmers. *In Proceedings of the 2008 ACM SIGGRAPH symposium on Video games* (pp. 55-59).

Alvarado, C., & Dodds, Z. (2010, March). Women in CS: An evaluation of three promising practices. *In Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 57-61).

Apple. (2017). Inclusion & Diversity. Retrieved January 12, 2018, from https://www.apple.com/diversity/

AZ Quotes. (n.d.). Retrieved from http://www.azquotes.com/quote/844498

Bandura, A. (1997). *Self-Efficacy: The Exercise of Control. Macmillan*.

Bau, D. (2015). Droplet, a block-based editor for text code. *Journal of Computing Sciences in Colleges*, *30*(6), 138-144.

Bau, D., Bau, D. A., Dawson, M., & Pickens, C. (2015, June). Pencil code: block code for a text world. *In Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 445-448).

Boud, D., & Feletti, G. (1997). Changing problem-based learning: Introduction to the second edition. *The challenge of problem-based learning*, 1-14.

Brians, C. L., Willnat, L., Rich, R., & Manheim, J. (2016). Elite and Specialized Interviewing: Discussing to Garner Knowledge. *Empirical Political Analysis*.

Brown, N. C., Altadmri, A., & Kölling, M. (2016, March). Frame-Based Editing: Combining the Best of Blocks and Text Programming. In *Learning and Teaching in Computing and Engineering (LaTICE), 2016 IEEE International Conference* (pp. 47-53).

Brown, N. C., Kolling, M., & Altadmri, A. (2015, October). Position paper: Lack of keyboard support cripples block-based programming. In Blocks and Beyond Workshop (Blocks and Beyond), 2015 IEEE (pp. 59-61).

Bugs, G. T. (2009). *The intersection of people, technology and local space: PPGIS and Web in practice for participatory planning* (Master Thesis, Universidade NOVA de Lisboa, Portugal).

Bureau of Labor Statistics. (2016). Employment Projections. Retrieved from http://www.bls.gov/emp/tables.htm

Bureau of Labor Statistics. (2017). *Occupational outlook handbook: Computer and information technology occupations*. U.S. Department of Labor. 2016-17 Edition.

Retrieved October 13, 2017, from https://www.bls.gov/ooh/computer-and-information-technology/home.htm

Bureau of Labor Statistics. (2018). *Occupational outlook handbook, computer and information technology, software developers*. Retrieved from https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm#tab-6

Byrne, J. R., Fisher, L., & Tangney, B. (2015, July). Computer science teacher reactions towards raspberry Pi Continuing Professional Development (CPD) workshops using the Bridge21 model. In *Computer Science & Education (ICCSE), 2015 10th IEEE International Conference* (pp. 267-272).

Campbell, D. T., & Stanley, J. C. (1963). Experimental and quasi-experimental designs for research. Handbook of research on teaching. Chicago, IL: Rand McNally.

Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. ACM SIGCSE Bulletin, 38(1), 27-31.

Code.org Infographic Source Data. (2015). Retrieved from https://docs.google.com/document/d/1wIZY4_0zQSoZy0sO7FL7yK3hlLGt0o6a0CQ6XoloO7g/pub

Code.org Infographic Source Data. (2016). Retrieved from https://docs.google.com/document/d/1gySkItxiJn_vwb8HIIKNXqen184mRtzDX12cux0ZgZk/pub

Code.org. (2013a).Retrieved from https://code.org/about

Code.org. (2013b). What Most Schools Don't Teach [Video File]. Retrieved from https://www.youtube.com/watch?v=nKIu9yen5nc

Code.org. (2013c). President Obama asks America to learn computer science [Video File]. Retrieved from https://www.youtube.com/watch?v=6XvmhE1J9PY

Code.org. (2018). Learn Computer Science. Retrieved from https://studio.code.org

Code.org. (n.d.-a). Hour of Code Artist Application. Retrieved from https://studio.code.org/s/artist/stage/1/puzzle/1

Code.org. (n.d.-b). All hour of code tutorials. Retrieved from https://code.org/educate/allhourofcode

Codecademy.com. (2018). Hour of Code. Retrieved from: https://www.codecademy.com/hour-of-code

Codecombat.com. (n.d.). Retrieved from https://codecombat.com/hoc

CodeGenie. (2018). Retrieved from http://thecodegenie.com/

Computer Science Education Week. (n.d.). Retrieved from https://csedweek.org/about

Cueto, J. (2015, July 13). Race and gender among computer science majors at Stanford. Retrieved November 30, 2017, from https://medium.com/@jcueto/race-and-gender-among-computer-science-majors-at-stanford-3824c4062e3a

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management science*, *35*(8), 982-1003.

DiSalvo, B. (2014). Graphical qualities of educational technology: Using drag-and-drop and text-based programs for introductory computer science. *IEEE Computer Graphics and Applications*, (6), 12-15.

DiversityData.org . (2011). Data for Diverse and Equitable Metropolitan Areas, Ann Arbor, MI. Retrieved from http://www.diversitydata.org/Data/Profiles/Show.aspx?loc=104&notes=True&rgn=None&cat=4

Durruthy, R. (2017). *LinkedIn workforce diversity report*. Retrieved January 12, 2018, from https://careers.linkedin.com/diversity-and-inclusion

Ericsson Mobility Report. (2016, November 07). Ericsson. Retrieved from https://www.ericsson.com/mobility-report

George, D., & Mallery, M. (2010). *SPSS for windows step by step: A simple guide and reference, 17.0 update* (10a ed.) Boston: Pearson.

GitHub Octoverse. (2017). Retrieved March 20, 2018, from https://octoverse.github.com/

Goforth, C. (2015, November). Using and interpreting Cronbach's alpha. Retrieved, February 14, 2018, from http://data.library.virginia.edu/using-and-interpreting-cronbachs-alpha/

Griffin, K. (2006). Striving for success: A qualitative exploration of competing theories of high-achieving Black college students' academic motivation. *Journal of College Student Development*, *47*(4), 384-400.

HackerRank. (2008). About us. Retrieved from https://www.hackerrank.com/aboutus

HackerRank. (n.d.). Frequently Asked Questions. Retrieved from https://www.hackerrank.com/faq

Hall, Z. (2017, October 10). Tim Cook believes learning to code is more important than English [Video]. Retrieved October 13, 2017, from https://9to5mac.com/2017/10/10/tim-cook-interview-video/

Holzinger, A. (2005). Usability engineering methods for software developers. *Communications of the ACM*, 48(1), 71-74.

HourOfCode.com. (2018). Retrieved from https://hourofcode.com/

Hundhausen, C. D., Narayanan, N. H., & Crosby, M. E. (2008). Exploring studio-based instructional models for computing education. *ACM SIGCSE Bulletin*, *40*(1), 392-396

Huysse, J. (2014, July). Building a Twitter we can be proud of. Retrieved January 12, 2018, from https://blog.twitter.com/official/en_us/a/2014/building-a-twitter-we-can-be-proud-of.html

Information Technology and Innovation Foundation. (2013). The U.S. STEM workforce: The real story. Retrieved February 01, 2018, from https://itif.org/publications/2013/05/16/us-stem-workforce-real-story

Kessler, S. (2017, March 10). You probably should have majored in computer science. Retrieved from https://qz.com/929275/you-probably-should-have-majored-in-computer-science/

Khanacademy. (n.d.). Retrieved from: https://www.khanacademy.org/hourofcode

Knake, L. (2015, Aug 19). *3 Ann Arbor-area high schools ranked in Newsweek's top 500*. Retrieved from http://www.mlive.com/news/ann-arbor/index.ssf/2015/08/3_ann_arbor-area_high_schools.html

Koitz, R., & Slany, W. (2014, October). Empirical comparison of visual to hybrid formula manipulation in educational programming languages for teenagers. In *Proceedings of the 5th Workshop on Evaluation and Usability of Programming Languages and Tools* (pp. 21-30).

Kölling, M. (2012). Joy of code with GreenFoot. Retrieved from http://blogs.kcl.ac.uk/proged/joc/

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, *37*(3), 14-18.

Landau, S., & Everitt, B. (2004). *A handbook of statistical analyses using SPSS*. Washington D. C.: CRC Press LLC.

Leedy, P. D., & Ormrod, J. E. (2013). *Practical Research: Planning and Design (10th Ed.).* Buffalo, NY: Pearson Higher Ed.

Lehman, K. J., Sax, L. J., & Zimmerman, H. B. (2016). Women planning to major in computer science: Who are they and what makes them unique? *Computer Science Education*, *26*(4), 277-298.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2010). *Stuck in the shallow end: Education, race, and computing*. Cambridge MA: MIT Press.

Martz, E. (2015, December 03). What can you say when your p-value is greater than 0.05? Retrieved November 20, 2017, from http://blog.minitab.com/blog/understanding-statistics/what-can-you-say-when-your-p-value-is-greater-than-005

Mazaika, K. (2017, January 20).Will the demand for developers continue to increase? Retrieved October 13, 2017, from https://www.forbes.com/sites/quora/2017/01/20/will-the-demand-for-developers-continue-to-increase/#38e1bb8433ee

McDowell, G. (2018). *2018 women in tech report*. Retrieved March 21, 2018, from https://research.hackerrank.com/women-in-tech/2018/

McKay, F., & Kölling, M. (2013, September). Predictive modelling for HCI problems in novice program editors. In *Proceedings of the 27th International BCS Human Computer Interaction Conference* (p. 35). London, UK: British Computer Society.

MIA-aa.org. (2017). MIA Demographic Data. Retrieved from http://www.mia-aa.org/about-us/mia-demographic-data/

MIA-aa.org. (n.d). Retrieved from http://www.mia-aa.org/about-us/

Michael, R. S. (2002). Threats to internal & external validity: Y520 strategies for educational inquiry [PowerPoint slides]. Retrieved from http://www.indiana.edu/~educy520/sec5982/week_9/520in_ex_validity.pdf

Microsoft. (2017). Inside Microsoft. Retrieved January 12, 2018, from https://www.microsoft.com/en-us/diversity/inside-microsoft/default.aspx

Monig, J., Ohshima, Y., & Maloney, J. (2015, October). Blocks at your fingertips: Blurring the line between blocks and text in GP. In *Blocks and beyond workshop (Blocks and Beyond)*, 2015 IEEE (pp. 51-53).

Morgan, R., Klaric, J. (2007). *AP students in college: An analysis of five-year academic careers*. New York: The College Board

Morrison, B. B., & DiSalvo, B. (2014, March). Khan academy gamifies computer science. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 39-44). Atlanta, GA, USA.

Moss, S. (2012, July 5). *Steve Jobs says everyone should learn to program* [Video File]. Retrieved from https://www.youtube.com/watch?v=mCDkxUbalCw

National Science Foundation. (2012). Science and Engineering Indicators. In. Retrieved from: http://www.nsf.gov/statistics/seind12/appendix.htm

Naughton, E. (2017, June 29). Making progress on diversity and inclusion. Retrieved January 11, 2018, from https://blog.google/topics/diversity/making-progress-diversity-and-inclusion/

Nickelsburg, M. (2017, October 11). Amazon consumer chief announces $10M donation to Code.org, says STEM education is a key factor in HQ2 search. Retrieved October 13, 2017, from https://www.geekwire.com/2017/amazon-consumer-chief-announces-10m-donation-code-org-says-stem-education-key-factor-hq2-search/

Nunnaly, J. (1978). *Psychometric theory*. New York: McGraw-Hill.

O'Grady, S. (2018, March 07). The Redmonk programming language rankings: January 2018. Retrieved March 08, 2018, from http://redmonk.com/sogrady/2018/03/07/language-rankings-1-18/

Ohlund, B., & Yu, C. (n.d.). Threats to validity of Research Design. Retrieved March 20, 2018, from http://web.pdx.edu/~stipakb/download/PA555/ResearchDesign.html

Partovi, H. (2015). A comprehensive effort to expand access and diversity in computer science. ACM Inroads, 6(3), 67-72.

PencilCode. (n.d.). Retrieved from https://pencilcode.net/edit/first

Peterson, C. (2003). Bringing ADDIE to life: Instructional design at its best. *Journal of Educational Multimedia and Hypermedia*, *12*(3), 227-242.

Pintrich, P. R. (1991). *A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ)*. Michigan: The regents of the University of Michigan.

Pioneer High School. (2014). *U.S. News & World Report*. Retrieved from http://www.usnews.com/education/best-high-schools/michigan/districts/ann-arbor-public-schools/pioneer-high-school-9694/student-body

Post, J. (2016, December 29). 9 companies founded by amazing young entrepreneurs. Retrieved from http://www.businessnewsdaily.com/5051-young-entrepreneurs.html

Promote Computer Science. (2016). Retrieved from https://code.org/promote

Rincón, R. (2017, December). Is there a shortage of stem jobs to stem graduates? It's complicated. Society of Women Engineers (SWE). Retrieved February 01, 2018, from http://alltogether.swe.org/2017/12/is-there-a-shortage-of-stem-jobs-to-stem-graduates-its-complicated/

Roberts, P., Priest, H., & Traynor, M. (2006). Reliability and validity in research. *Nursing Standard*, *20*(44), 41.

Rothwell, J. (2012). The need for more STEM workers. Brookings institution. Retrieved from: https://www.brookings.edu/blog/the-avenue/2012/06/01/the-need-for-more-stem-workers/

Runkel, P. (2016, November 04). What are t values and p values in statistics? Retrieved January 24, 2018, from http://blog.minitab.com/blog/statistics-and-quality-data-analysis/what-are-t-values-and-p-values-in-statistics

Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, *25*(1), 54-67. doi:10.1006/ceps.1999.1020.

Sarkar, A. (2015, July). The impact of syntax colouring on program comprehension. In *Proceedings of the 26th Annual Conference of the Psychology of Programming Interest Group* (ppig 2015) (pp. 49-58).

Scratch. (n.d). Retrieved from https://scratch.mit.edu

Shaun, E. (2017, March 02). 5 most popular frameworks for mobile development. Retrieved November 29, 2017, from https://medium.com/@eduoshaun/5-most-popular-frameworks-for-mobile-development-4089ffbf536f

Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., & Elmqvist, N. (2016). *Designing the user interface: strategies for effective human-computer interaction (6th ed.).* USA: Pearson.

Siminoff, J. (2017, January). Building a more inclusive Twitter in 2016. Retrieved January 12, 2018, from https://blog.twitter.com/official/en_us/topics/company/2017/building-a-more-inclusive-twitter-in-2016.html

Sloan, R. H., & Troy, P. (2008, March). CS 0.5: A better approach to introductory computer science for majors. In *ACM SIGCSE Bulletin.40*(1). (pp. 271-275).

Smith, A., & Anderson, J. (2014). AI, robotics, and the future of jobs. Pew Research Center. Retrieved from http://www.fusbp.com/wp-content/uploads/2010/07/AI-and-Robotics-Impact-on-Future-Pew-Survey.pdf

Smith, M. (2016, January 30). Computer science for all. Retrieved from https://www.whitehouse.gov/blog/2016/01/30/computer-science-all

Snyder, T., Brey, C., & Dillow, S. (2016, December). *Digest of Education Statistics 2015 (Rep.)*. Retrieved from https://nces.ed.gov/pubs2016/2016014.pdf

SPSS Stepwise Regression. (2017). Retrieved February 28, 2018, from https://www.spss-tutorials.com/spss-stepwise-regression-example-2/

StackOverflow. (2008). Learn, Share, Build. Retrieved from https://stackoverflow.com/company/about

StackOverflow. (2016). Developer survey results 2016. Retrieved from https://insights.stackoverflow.com/survey/2016

StackOverflow. (2017). Developer survey results 2017. Retrieved from https://stackoverflow.com/insights/survey/2017

Study.com. (n.d.). What is JavaScript? - Function & Uses. Retrieved from http://study.com/academy/lesson/what-is-javascript-function-uses-quiz.html

Tashtoush, Y., Odat, Z., Alsmadi, I., & Yatim, M. (2013). Impact of programming features on code readability. *International Journal of Software Engineering and Its Applications*, *7*(6). pp.441-458.

Teo, T. (2010). Examining the influence of subjective norm and facilitating conditions on the intention to use technology among pre-service teachers: A structural equation modeling of an extended technology acceptance model. *Asia Pacific Education Review*, *11*(2), 253-262.

Uzayr, S. B. (2016, September 30). 15 useful code sharing websites for web developers. Retrieved March 08, 2018, from https://www.designbombs.com/code-sharing-websites/

Vincent, J. (2013, November) Kano: A Raspberry Pi computer that snaps together like Lego in 107 seconds. *The Independent*. Retrieved from http://www.independent.co.uk/life-style/gadgets-and-tech/news/kano-a-raspberry-pi-computer-that-snaps-together-like-lego-in-107-seconds-8953923.html

Weinberger, M. (2017, October 11). The 15 most popular programming languages, according to the 'Facebook for programmers'. Business Insider. Retrieved March 20, 2018, from http://www.businessinsider.com/the-9-most-popular-programming-languages-according-to-the-facebook-for-programmers-2017-10

Weintrop, D., & Wilensky, U. (2015a). To block or not to block, that is the question: students' perceptions of block-based programming. In *Proceedings of the 14th International Conference on Interaction Design and Children* (pp. 199-208).

Weintrop, D., & Wilensky, U. (2015b). Using commutative assessments to compare conceptual understanding in block-based and text-based programs. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research, ICER, 15.* (pp. 101-110).

White House. (2016, January 30). *Weekly Address: Giving Every Student an Opportunity to Learn Through Computer Science For All* [Video File]. Retrieved from https://www.youtube.com/watch?v=8sthaV8ddJ4

White House. (2017, September 27). Ivanka Trump Promotes Access to High-Quality Coding and Computer Science Education Programs. Retrieved September 29, 2017, from https://www.whitehouse.gov/blog/2017/09/27/ivanka-trump-promotes-access-high-quality-coding-and-computer-science-education

Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education*, 12(3), 197-212.

Williams, M. (2017, August). Facebook diversity update: Building a more diverse, inclusive workforce. Retrieved January 11, 2018, from https://newsroom.fb.com/news/2017/08/facebook-diversity-update-building-a-more-diverse-inclusive-workforce/

Wohlsen, M. (2016, May 23). What we'll be doing once the robots take our jobs. Retrieved from http://www.wired.co.uk/article/robot-work

Wojcick, S. (2014). Things you love are Made with Code. Retrieved from https://www.blog.google/outreach-initiatives/diversity/things-you-love-are-made-code/

Wu, W. (2015, July 10). Race and gender among computer science concentrators at Harvard. Retrieved November 30, 2017, from https://medium.com/@winniewu/race-and-gender-among-computer-science-concentrators-at-harvard-1c1943a20457

Xue, Y., & Larson, R. C. (2015). STEM crisis or STEM surplus: Yes and yes. Retrieved from https://www.bls.gov/opub/mlr/2015/article/stem-crisis-or-stem-surplus-yes-and-yes.htm

Zweben, S. (2011). Computing degree and enrollment trends. Computing Research Association. Retrieved from: http://cra.org/uploads/documents/resources/taulbee/CRA_Taulbee_CS_Degrees_and_Enrollment_2011-12.pdf

APPENDICES

Appendix A: Human Subject Approval and Informed Consent Form

After getting the human subject approvals from Eastern Michigan University, the workshop/camp flyer and the consent from were sent to students through the school messenger and the schools' administrators.

RESEARCH @ EMU

UHSRC Determination: EXPEDITED INITIAL APPROVAL

Date: June 21, 2017

To: Hadeel Mohammed Jawad
Eastern Michigan University

Re: UHSRC: # 1079881-1
Category: Expedited Category 7
Approval Date: June 21, 2017
Expiration Date: June 20, 2018

Title: Integrating Art & Design in Teaching Computer Programming for High School Students

Your research project, entitled **Integrating Art & Design in Teaching Computer Programming for High School Students,** has been approved in accordance with all applicable federal regulations.

This approval includes the following:
1. Enrollment of *up to 40* subjects to participate in the approved protocol.
2. Use of the following study measures: *online survey questionnaire items.*
3. Use of the following stamped recruitment materials: *flier.*
4. Use of the stamped *consent form.*

Renewals: This approval is valid for one year and expires on 6/20/2018 If you plan to continue your study beyond 6/20/2018, you must submit a Continuing Review Form by 5/10/2018 to ensure the approval does not lapse.

Modifications: All changes must be approved prior to implementation. If you plan to make any minor changes, you must submit a **Minor Modification Form.** For any changes that alter study design or any study instruments, you must submit a **Human Subjects Approval Request Form**. These forms are available through IRBNet on the UHSRC website.

Problems: All major deviations from the reviewed protocol, unanticipated problems, adverse events, subject complaints, or other problems that may increase the risk to human subjects **or** change the category of review must be reported to the UHSRC via an **Event Report** form, available through IRBNet on the UHSRC website

Follow-up: If your Expedited research project is not completed and closed after **three years**, the UHSRC office requires a new **Human Subjects Approval Request Form** prior to approving a continuation beyond three years.

Please use the UHSRC number listed above on any forms submitted that relate to this project, or on any correspondence with the UHSRC office.

Good luck in your research. If we can be of further assistance, please contact us at 734-487-3090 or via e-mail at human.subjects@emich.edu. Thank you for your cooperation.

Sincerely,

Sonia Chawla, PhD
Research Compliance Officer

Group One (G1)

Human Subjects Informed Consent Form

Study Title

Integrating Art and Design in Teaching Computer Programming for High School Students.

Study Purpose and Rationale

This research study is designed to help us understand the effect of integrating art and design with computer programming on students' knowledge in computer science (CS) and on their interest to pursue a degree in CS after graduation from high school. The study will also measure effect other factors on students' interest in pursuing a degree in CS. these factors include students' attitude toward CS, social norm and behavioral control. Findings from this study may lead to improved methods for teaching computer programming.

Study Procedures

Participants will complete a 20-minute online survey questionnaire at the beginning and end of the summer camp. Participants grant permission to use their survey results and demographic data as part of this study. To be eligible to participate in this study, students and their parents must sign this form.

Risks or Discomforts

This study has no foreseeable risks.

Benefits

Your child may not directly benefit from this study. However, Participants will learn about computer programming by attending this summer camp. In addition, participants who have some knowledge in computer programming could improve their programming skills. This research may lead to improved methods for teaching Computer Science.

Confidentiality and Data Storage

The consent forms will be collected electronically through Google Forms and will be stored in password-protected files on the researcher personal computer. Prior to analysis of the collected data, each participant will be assigned an arbitrary identifier. This will be used to transcribe your survey results and the demographic data (gender, ethnicity, age). Participant's name will be removed from the records, and no personally-identifying information will be disclosed as part of this study.

Approved by the Eastern Michigan University Human Subjects Review Committee
UHSRC Protocol Number: 1079881-1
Study Approval Dates: 6/21/2017 – 6/20/2018

About Participation

Your participation in this study is voluntary and you have the right to discontinue participation at any time without prejudice from the investigator or loss of benefit to which your child is otherwise entitled. If you choose to participate and later withdraw your consent, your child will not be able to continue with the camp.

Compensations

Upon completing each survey your child will receive a small gift.

Parent or Legal Guardian Agreement

I read the above consent form. The nature, demands, risk, and benefits of the project have been is clear to me. I am aware that I have the opportunity to ask questions about this research. I may withdraw my consent and discontinue my child's participation at any time without penalty. By registering my child, I consent to the participation in this research.

Student's Agreement

I agree to participate in this research project and I have received a copy of this form.

For Questions about your rights as a research participant please contact:
Eastern Michigan University, Human Review Committee
734 - 487 - 3090
human.subjects@emich.edu

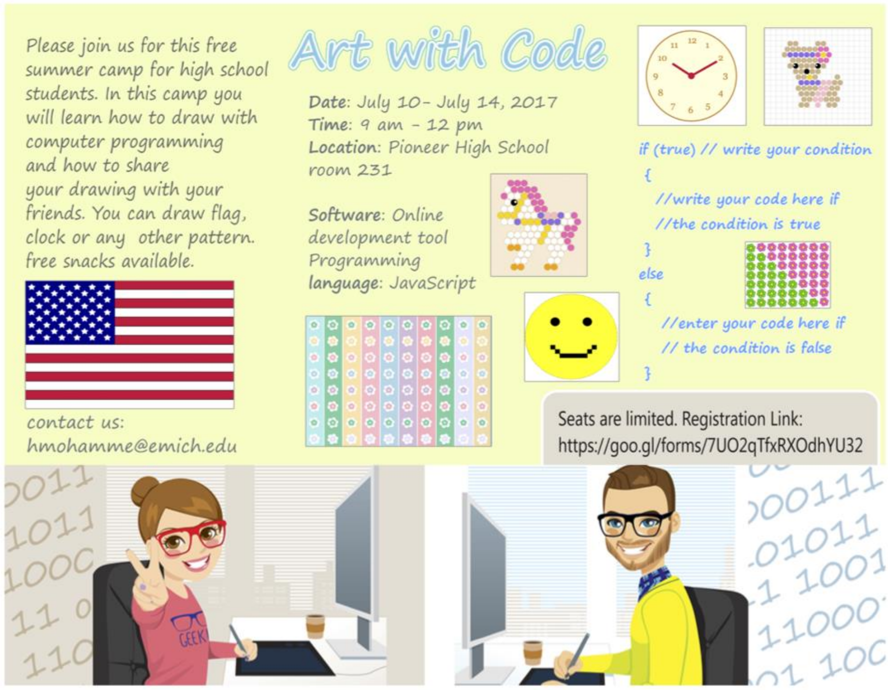
Contact Information

Hadeel Mohammed Jawad
College of Technology
Eastern Michigan University
Ypsilanti, Michigan 48197
hmohamme@emich.edu

Yichun Xie, Ph.D.
Institute for Geospatial Research and Education
125 King Hall
Eastern Michigan University
Ypsilanti, Michigan 48197
Telephone: 734- 487-7588 Email:
yxie@emich.edu,

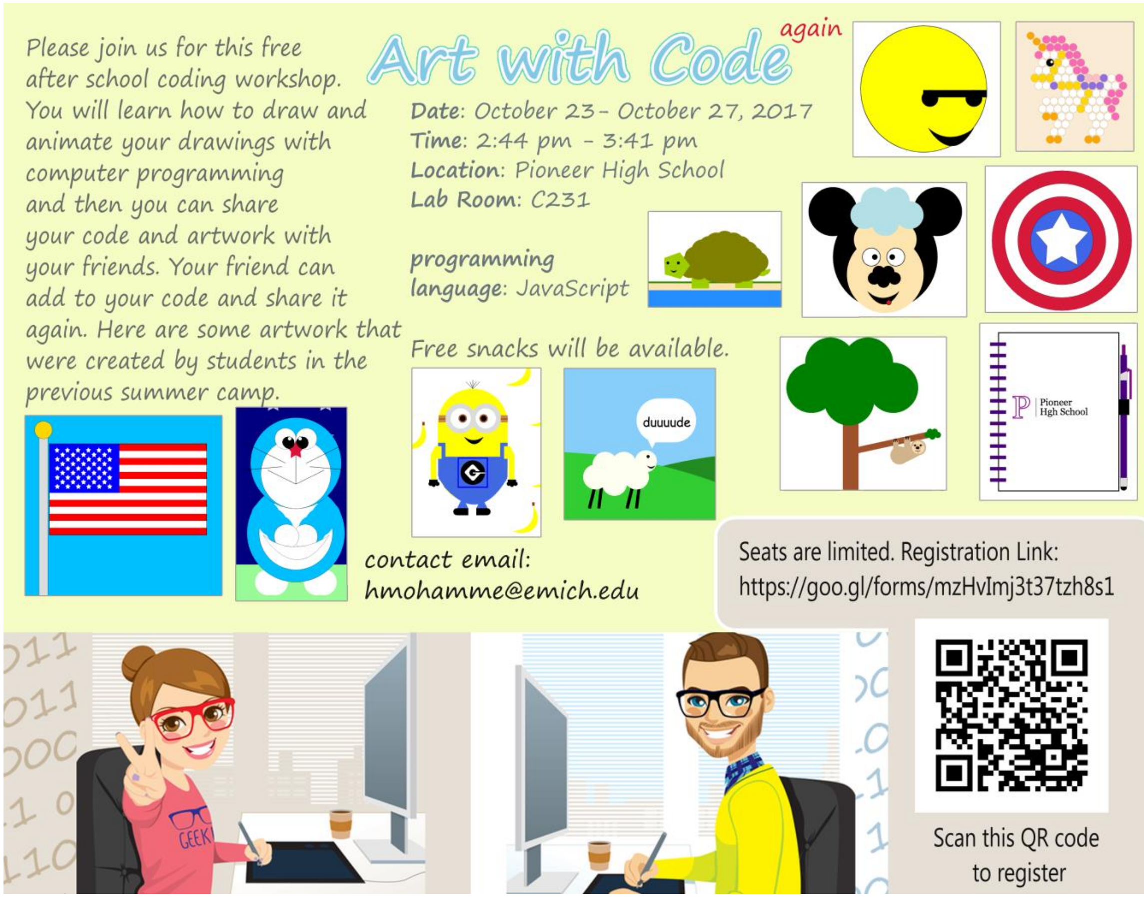
Approved by the Eastern Michigan University Human Subjects Review Committee
UHSRC Protocol Number: 1079881-1
Study Approval Dates: 6/21/2017 – 6/20/2018

Camp Flyer



Approved by the Eastern Michigan University Human Subjects Review Committee
UHSRC Protocol Number: 1079881-1
Study Approval Dates: 6/21/2017 – 6/20/2018

Group Two (G2)


Please join us for this free after school coding workshop. You will learn how to draw and animate your drawings with computer programming and then you can share your code and artwork with your friends. Your friend can add to your code and share it again. Here are some artwork that were created by students in the previous summer camp.
Art with Code again
Date: October 23 - October 27, 2017
Time: 2:44 pm - 3:41 pm
Location: Pioneer High School
Lab Room: C231
programming language: JavaScript
Free snacks will be available.
duuuude
Pioneer High School
contact email: hmohamme@emich.edu
Seats are limited. Registration Link: https://goo.gl/forms/mzHvImj3t37tzh8s1
Scan this QR code to register

Group Three (G3)

Human Subject Informed Consent Form

Study Title

Integrating Art and Animation in Teaching Computer Programming for High School Students.

Study Purpose and Rationale

This research study is designed to help us understand the effect of integrating art and animation with computer programming on students' knowledge in computer science (CS) and on their interest to pursue a degree in CS after graduation from high school. The study will also measure effect other factors on students' interest in pursuing a degree in CS. These factors include students' attitude toward CS, social norm and behavioral control. Findings from this study may lead to improved methods for teaching computer programming.

Study Procedures

Participants will complete an online survey questionnaire at the beginning and end of the coding workshop. Participants grant permission to use their survey results and demographic data as part of this study. To be eligible to participate in this study, students and their parents must sign this form.

Risks or Discomforts

This study has no foreseeable risks.

Benefits

Your child may not benefit from this study. However, Participants will learn about computer programming by attending this coding workshop. In addition, participants who have

Approved by the Eastern Michigan University Human Subjects Review Committee
UHSRC Protocol Number: 1079881
Study Approval Dates: 11/21/17 – 06/20/18

some knowledge in computer programming could improve their programming skills. This research may lead to improved methods for teaching computer science.

Confidentiality and Data Storage

The consent forms will be collected from the student before the experimental workshop and will be stored safely by the researcher for the duration of this school year. Prior to analysis of the collected data, each participant will be assigned an arbitrary identifier. This will be used to transcribe your survey results and the demographic data (gender, ethnicity, age). Participant's name will be removed from the records, and no personally-identifying information will be disclosed as part of this study.

About Participation

Your participation in this study is voluntary and you have the right to discontinue participation at any time without prejudice from the investigator or loss of benefit to which your child is otherwise entitled. If you choose to participate and later withdraw your consent, your child will not be able to continue with the workshop.

For Questions about your rights as a research participant please contact:

Eastern Michigan University, Human Review Committee
734 - 487 - 3090
human.subjects@emich.edu

Approved by the Eastern Michigan University Human Subjects Review Committee
UHSRC Protocol Number: 1079881
Study Approval Dates: 11/21/17 – 06/20/18

Student Agreement

I agree to participate in this research project and I have received a copy of this form.

Student's Name (please print)	Date
Student's Signature	Student's Email

Parent or Legal Guardian Agreement

I read the above consent form. The nature, demands, risk, and benefits of the project have been clear to me. I am aware that I have the opportunity to ask questions about this research. I may withdraw my consent and discontinue my child's participation at any time without penalty. By registering my child, I consent to the participation in this research.

Parent's Name (please print)	Date
Parent's Signature	Parent's Email

Contact Information

Hadeel Mohammed Jawad
College of Technology
Eastern Michigan University
Ypsilanti, Michigan 48197
Email: hmohamme@emich.edu

Samir Tout, Ph.D., PMP, GIAC-GREM, ITIL, TOGAF
Professor of Information Assurance
School of Information Security & Applied Computing (SISAC)
206 F Roosevelt Hall
Eastern Michigan University
Ypsilanti, MI 48197 USA
Phone: 734.487.4906
E-mail: stout@emich.edu

Approved by the Eastern Michigan University Human Subjects Review Committee
UHSRC Protocol Number: 1079881
Study Approval Dates: 11/21/17 – 06/20/18


Art with Code
Please join us for this free coding workshop.You will learn how to draw and animate your drawings with computer programming and then you can share your code and artwork with your friends. Your friend can add to your code and share it again.
Here are some artwork that were created by students in the previous coding camp.
Date: December 6th, 2017
Time: 12:00 pm - 3:20 pm
Location: MIA Private School
Prog. Language: JavaScript
duuuude
Pioneer Hgh School
contact email:
hmohamme@emich.edu
Seats are limited.
Please sign the attached form
Approved by the Eastern Michigan University Human Subjects Review Committee
UHSRC Protocol Number: 1079881
Study Approval Dates: 11/21/17 – 06/20/18

Appendix B: Online Survey Questionnaire

The following is the survey questionnaire that was used in the pretest and the posttest.

Part 1: Demographic Data	
1.	What is your full name?
2.	What is your gender? ◯ Male ◯ Female
3.	What is your race? ◯ American Indian or Alaska Native ◯ Asian ◯ Asian Indian ◯ Black or African American ◯ Hispanic or Latino ◯ Native Hawaiian or Other Pacific Islander ◯ Middle Eastern, Arabic, or Persian ◯ White
4.	What grade are you going next school year? ◯ G9 ◯ G10 ◯ G11 ◯ G12

Part 2 : Programming Knowledge	

Assume the following block of code for the following three questions
`n1=3;`
`n2=8;`
`n3=n1*n2;`
`n2=n1+n2;`	

5.	After the above code, n1contains (1, 3, 8, 11, Don't Know)
6.	After the above code n2 contains (24, 3, 8, 11, Don't Know)
7.	After the above code, n3 contains (24, 3, 8, 11, Don't Know)

Assume the following block of code for the following questions:
`n1=0;`
`n2=2;`
`n3=3;`
`for ( i = 1; i <= 4; i++)`
`{`
`    n1= n1 + 1;`
`    if (n1 > 4)`
`        n2 = n1 * n2;`
 	
`    if (n2 == 2)`
`        n3=10;`
`    else`
`        n3=5;`
`}`	

8.	After the above code n1 contains (0, 3, 4, 5, Don't Know)
9.	After the above code,n2 contains (2, 4, 16, 8, Don't Know)
10.	After the above code, n3 contains (10, 5, 3, 2, Don't Know)

11.	Assume the following block of code
	`n1=4;`
	
	`if (n1 == 3)`
	`    circle(100,100,25);`
	`else`
	`    star(100,100,1);`
	
	The above code will draw
	(Circle, Star, Circle and Star , Nothing, Don't Know)

12.	Assume the following block of code
	`for ( i = 1; i < 5; i++)`
	`{`
	`    circle(20*i,100,10);`
	
	`}`
	
	The above code will draw
	(One Circle, Five Circles, Four Circles, Nothing, Don't Know)

13.	Assume the following block of code

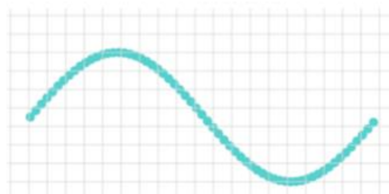
```
n1=0;
n2=2;
switch (n2)
{
      case 0:  n1=5; break;
      case 1:  n1=10; break;
      case 2:  n1=8; break;
      default: n1=5;
}
```

After the above code n1 contains
(10, 5, 3, 8, Don't Know)

14. Assume the following two piece of Code A& B and the following figure:

Code A

```
for ( i = -3.14; i <= 3.14; i+= 0.1)
{
  y=150 + Math.cos ( i ) * 70;
  circle ( 250 + i * 60 , y , 5);
}
```



Code B

```
for ( i = -3.14; i <= 3.14; i+= 0.1 )
{
  y= 350 + Math.sin ( i ) * 70;
  circle ( 250 + i * 60 , y , 5);
}
```

Which code produces the shown figure?
(A, B, Both, None, Don't Know)

15. The following code draws 10 stars

```
arrColor = ['red', 'green', 'purple',
'yellow', 'pink', 'brown', 'gray', 'blue',
'skyBlue', 'black' ] ;

for ( i=0 ; i<10 ; i++)
{
  color ( arrColor [ i ] ) ;
  star ( 20 + i * 50, 100 , 1);
}
```

What is the value of " i " in " color (arrColor [i]) ; " that gives the 'LightPink' color ?
(0, 1, 2, 3, Don't Know)

16.	Assume the following block of code
	`function fn (y)`
	`{`
	`    return y * y;`
	`}`
	`x= fn (3);`
	`z= fn (4);`
	
	After the above code x contains
	(4, 16, 9, 3, Don't Know)

17.	From 1 to 5, how would you rate the level of your programming knowledge and skills? 1 ◯ 2 ◯ 3 ◯ 4 ◯ 5 ◯
18.	Writing a program that includes art and design increases my knowledge in programming language. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
19.	Writing a program that includes animation increases my knowledge in programming language. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
Part 3 : Interest in CS Degree and CS Course	
20.	I have an interest in pursuing a degree in computer science. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
21.	I enjoy computer programming and I want to be a good programmer in the future. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree

22.	My sibling/relative has a degree in computer science and I like what he/she does 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
23.	Writing a program that includes art and design increases my interest in pursuing a degree in Computer Science. 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
24.	Writing a program that includes animation increases my interest in pursuing a degree in Computer Science. 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
25.	I have an interest in taking a computer science courses in high school next year 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
Part 4 : Real Programming Preference	
26.	Writing a program with a real programming language is different than writing a program with a block-based programming language. 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
27.	I prefer writing a program with a real programming over writing it with a block-based programming language. 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
28.	With real programming language, I have much more to do than with the block-based programming language

	1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
29.	I think students in high school are ready to write a program with a real programming language. 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
30.	I think writing a computer program with a real programming language is easy and I can do it 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
Part 5 : Programming Interest and Enjoyment	
31.	I am very interested in learning more about computer programming 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
32.	Writing a program that includes art and design increases my interest and motivation towards computer programming. 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
33.	Writing a program that includes animation increases my interest and motivation towards computer programming. 1 ○ Strongly Disagree 2 ○ Disagree 3 ○ Neutral 4 ○ Agree 5 ○ Strongly Agree
34.	I enjoy writing a computer program.

	1 Strongly Disagree ○ 2 Disagree ○ 3 Neutral ○ 4 Agree ○ 5 Strongly Agree ○
35.	I enjoy writing a program that produces art. 1 Strongly Disagree ○ 2 Disagree ○ 3 Neutral ○ 4 Agree ○ 5 Strongly Agree ○
Part 6 : Art and Animation Usefulness	
36.	Writing a program that includes art and design increases my knowledge in programming language. 1 Strongly Disagree ○ 2 Disagree ○ 3 Neutral ○ 4 Agree ○ 5 Strongly Agree ○
37.	Writing a program that includes animation increases my knowledge in programming language. 1 Strongly Disagree ○ 2 Disagree ○ 3 Neutral ○ 4 Agree ○ 5 Strongly Agree ○
38.	Writing a program that includes art and design makes writing a program with a real programming language easy. 1 Strongly Disagree ○ 2 Disagree ○ 3 Neutral ○ 4 Agree ○ 5 Strongly Agree ○
39.	Writing a program that includes art and design is useful in learning programming and math functions.

	1 2 3 4 5 Strongly Disagree Disagree Neutral Agree Strongly Agree
40.	Writing a program that includes animation is useful in learning programming and math functions. 1 2 3 4 5 Strongly Disagree Disagree Neutral Agree Strongly Agree
	Writing a program that produces art and animation increases my creativity. 1 2 3 4 5 Strongly Disagree Disagree Neutral Agree Strongly Agree
Part 7 : Motivation for Code Sharing	
41.	Sharing my code with others is useful in learning programming. 1 2 3 4 5 Strongly Disagree Disagree Neutral Agree Strongly Agree
42.	Writing a program that produces art and design encourages me to share my code. 1 2 3 4 5 Strongly Disagree Disagree Neutral Agree Strongly Agree
43.	Writing a program that produces art and animation encourages me to share my code 1 2 3 4 5 Strongly Disagree Disagree Neutral Agree Strongly Agree
44.	Sharing art work produced by code with others increases my motivation to learn programming.

	1 2 3 4 5 Strongly Disagree / Disagree / Neutral / Agree / Strongly Agree
45.	Sharing my code will help other people to solve similar problem and encourage them to share their code. 1 2 3 4 5 Strongly Disagree / Disagree / Neutral / Agree / Strongly Agree
46.	I enjoy sharing my program with others. 1 2 3 4 5 Strongly Disagree / Disagree / Neutral / Agree / Strongly Agree
47.	I enjoy when people like my shared code 1 2 3 4 5 Strongly Disagree / Disagree / Neutral / Agree / Strongly Agree
48.	Getting more likes on my shared code motivates me to write more code and share it. 1 2 3 4 5 Strongly Disagree / Disagree / Neutral / Agree / Strongly Agree
49.	I feel proud when I write a program and share the results with others. 1 2 3 4 5 Strongly Disagree / Disagree / Neutral / Agree / Strongly Agree
50.	My friend share her/his code and I like to share my code. 1 2 3 4 5 Strongly Disagree / Disagree / Neutral / Agree / Strongly Agree
51.	I like to compete with my friends by writing a program that produces a cooler design and get more likes.

	1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
Part 8 : Programming Benefits and Enjoyment	
52.	Writing a computer program increases my confidence in problem solving 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
53.	Facing errors and fixing them while programming increases my persistence and makes me more determined to attain my goals. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
54.	I think knowing how to write a computer program will empower me, and even if I didn't find a job, I can write my own apps and sell them in the app store, or I can build my own website and get money from commercial ads. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
55.	I think getting a degree in computer science will increase my chance to get a good job with a good salary since there will be more demand for computing jobs in the future. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
56.	Taking a computer science course in high school will improve my overall GPA. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
57.	Writing a program and running it successfully gives me a sense of accomplishment and makes me feel happy. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree

58.	I enjoy writing a computer program. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
Part 9 : Social Norms	
59.	My parents and my siblings encourage me to get a degree in computer science 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
60.	Future job markets will require many computer jobs, and many of my friends are thinking of computer science degree. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
61.	My sibling/relative has a degree in computer science and I want to have a degree in CS. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
62.	My parents encourage me to take a computer science course in high school. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
Part 10 : Programming Capabilities and Confidence	
63.	I think writing a computer program with a real programming language is easy and I can do it. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
64.	I am confident that having a computer science degree is the right choice for me. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree

65.	Programming needs patience and I have patience. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
66.	When I face errors while programming, I don't give up easily. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
67.	If I face a difficulty in writing a program, I am sure I will find a solution on the internet. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
68.	I know programming is not easy but I accept challenges and I feel happy when I solve difficult problems. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
69.	When I face errors while programming, I don't give up easily. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
70.	I think computer programming is easy to learn. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
	Part 11 : About the Workshop and the Tool
71.	I like this workshop and I want to take similar one again. 1 ◯ Strongly Disagree · 2 ◯ Disagree · 3 ◯ Neutral · 4 ◯ Agree · 5 ◯ Strongly Agree
72.	I would recommend this workshop to my friends

	1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
73.	I think using the Code Genie development environment in learning programming is easy and I didn't face any difficulty using it. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
74.	I think the Code Genie development environment is a very useful tool in learning programming language. 1 ◯ Strongly Disagree 2 ◯ Disagree 3 ◯ Neutral 4 ◯ Agree 5 ◯ Strongly Agree
75.	Code genie development environment has several features. Which of the following features is more useful in learning programming? ◯ Art ◯ Animation ◯ Sharing ◯ All ◯ None
76.	Add your comment or suggestion on Code Genie development environment
77.	Add a general comment on your experience in this workshop

Appendix C: Normality Tests for Study Variables

The following are the results of the normality test for the six study variables where t-test analysis was used to explore the effect of the treatment. Normality test is one of the assumptions for the paired samples t-test. The differences between the protest and posttest scores were tested for normality. The "Skewness" should less than 0.8 and the "Kurtosis" should be less than two (George & Mallery, 2010). The six study variables were within the acceptable range for normal distribution. Also, the differences histograms (first column in Table C-2) look approximately symmetric and bell-shaped.

Table C-1
Normality Test Results for the Six Study variables.

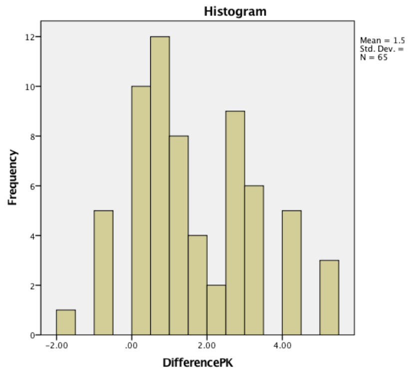
			Statistic	Std. Error
DifferencePK	Mean		1.5385	.19972
	95% Confidence Interval for Mean	Lower Bound	1.1395	
		Upper Bound	1.9374	
	5% Trimmed Mean		1.4986	
	Median		1.0000	
	Variance		2.593	
	Std. Deviation		1.61018	
	Minimum		-1.67	
	Maximum		5.00	
	Range		6.67	
	Interquartile Range		2.08	
	Skewness		.393	.297
	Kurtosis		-.493	.586
DifferenceDI	Mean		.2692	.09112
	95% Confidence Interval for Mean	Lower Bound	.0872	
		Upper Bound	.4513	
	5% Trimmed Mean		.2457	
	Median		.5000	
	Variance		.540	
	Std. Deviation		.73462	
	Minimum		-1.00	
	Maximum		2.50	
	Range		3.50	
	Interquartile Range		.75	
	Skewness		.425	.297
	Kurtosis		.415	.586
DifferenceRPP	Mean		2.5846	.49947
	95% Confidence Interval for Mean	Lower Bound	1.5868	
		Upper Bound	3.5824	
	5% Trimmed Mean		2.5726	
	Median		2.0000	
	Variance		16.215	
	Std. Deviation		4.02683	
	Minimum		-6.00	
	Maximum		12.00	
	Range		18.00	
	Interquartile Range		5.50	
	Skewness		.130	.297
	Kurtosis		-.476	.586
DifferenceMCS	Mean		1.7846	1.11405
	95% Confidence Interval for Mean	Lower Bound	-.4410	
		Upper Bound	4.0102	
	5% Trimmed Mean		1.3462	

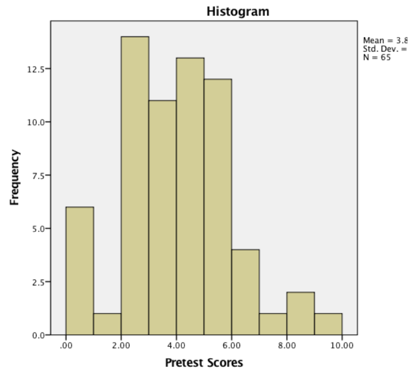
	Median		1.0000	
	Variance		80.672	
	Std. Deviation		8.98174	
	Minimum		-14.00	
	Maximum		30.00	
	Range		44.00	
	Interquartile Range		11.50	
	Skewness		.701	.297
	Kurtosis		1.101	.586
DifferencePIE	Mean		.6769	.21607
	95% Confidence Interval for Mean	Lower Bound	.2453	
		Upper Bound	1.1086	
	5% Trimmed Mean		.6282	
	Median		1.0000	
	Variance		3.035	
	Std. Deviation		1.74201	
	Minimum		-4.00	
	Maximum		6.00	
	Range		10.00	
	Interquartile Range		2.00	
	Skewness		.242	.297
	Kurtosis		1.150	.586
DifferencCI	Mean		.2769	.15113
	95% Confidence Interval for Mean	Lower Bound	-.0250	
		Upper Bound	.5788	
	5% Trimmed Mean		.2179	
	Median		.0000	
	Variance		1.485	
	Std. Deviation		1.21845	
	Minimum		-2.00	
	Maximum		4.00	
	Range		6.00	
	Interquartile Range		1.00	
	Skewness		.995	.297
	Kurtosis		2.025	.586

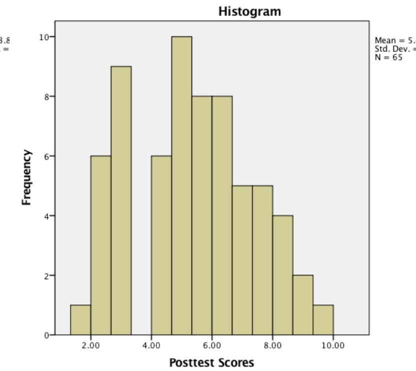
Table C-2

The Differences Histograms for the Six Study variables.

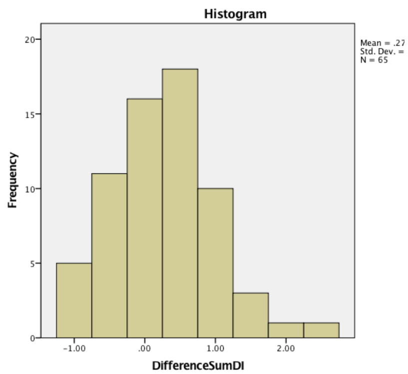
Difference	Pretest Data	Posttest Data
Programming Knowledge		

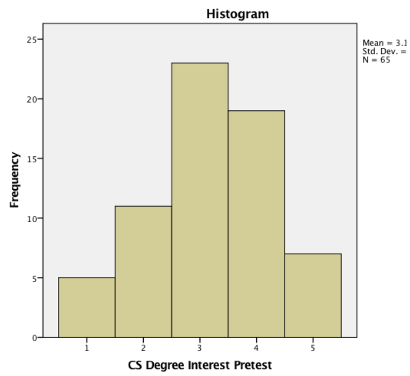


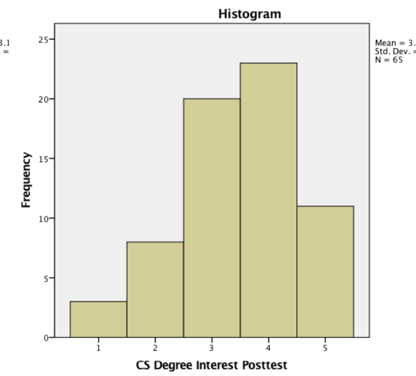




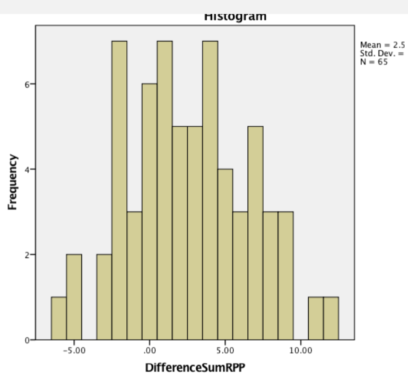
CS Degree Interest

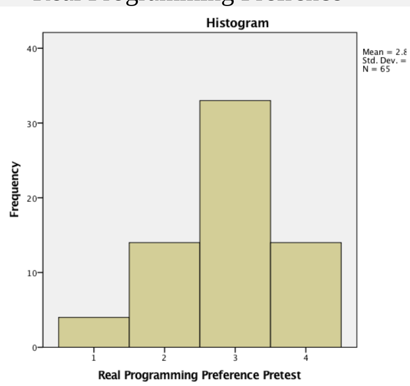


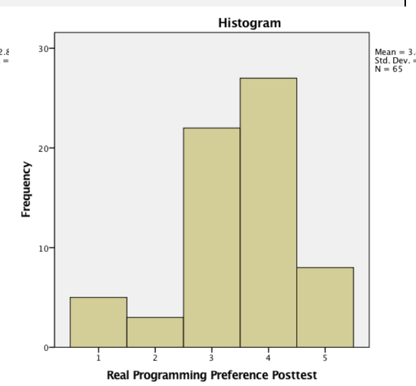




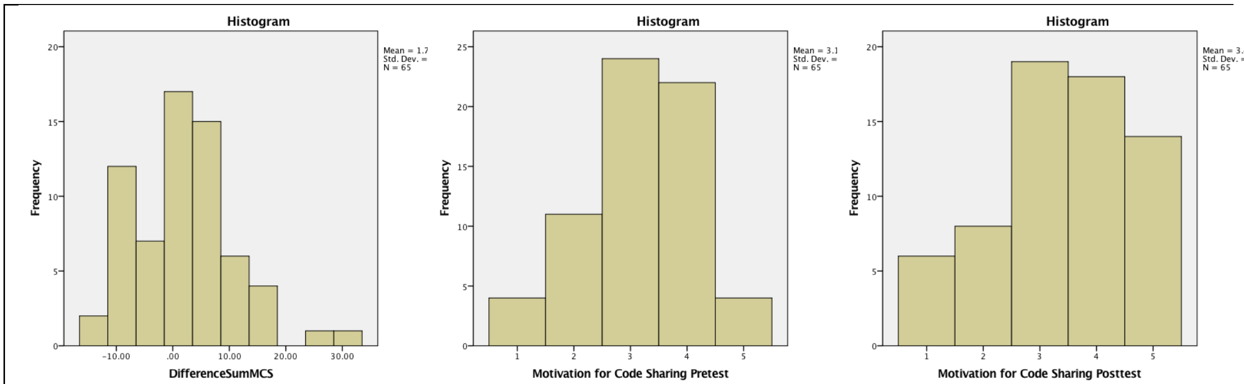
Real Programming Prefrence



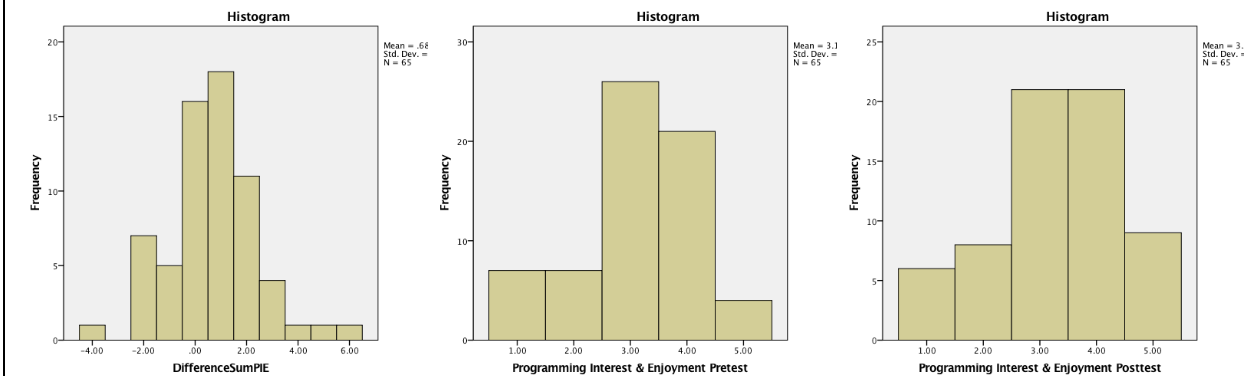




Motivation For Code Sharing



Programming Benefit and Enjoyment



CS Course Interest