Integrating Art and Animation in Teaching Computer Programming for High School Students

Experimental Study

Hadeel Mohammed Jawad
Ph.D. Candidate
College of Technology
Eastern Michigan University
Michigan, USA
hmohamme@emich.edu

Samir Tout, Ph.D.
Professor
College of Technology
Eastern Michigan University
Michigan, USA
stout@emich.edu

Munther Abualkibash, Ph.D.
Assistant Professor
College of Technology
Eastern Michigan University
Michigan, USA
mabualki@emich.edu

Yichun Xie, Ph.D.
Professor and Director
Institute for Geospatial Research and Education
Eastern Michigan University
Michigan, USA
yxie@emich.edu

***Abstract* – This paper discusses the results of an experimental study that was conducted to explore the effect of integrating art and animation in teaching computer programming on high school students' interest and knowledge in programming. The study aimed to explore the students' interest in pursuing a degree in Computer Science (CS) after graduation. Three groups of high school students were targeted with educational programming sessions, and the study variables were measured through pre and posttest surveys. A new web-based programming tool was developed and used as the treatment in this study. The developed tool includes the use of art, animation and code sharing to increase students' motivation in learning computer programming. The results of the study showed that the use of art, animation and code sharing increased students' knowledge, enjoyment, and motivation in learning computer programming, and hence, increased their interest in pursuing a degree in CS after graduation.**

***Keywords* -- programming for high school students, computer science for girls, art with coding, animation, code sharing, STEM tool for K12.**

I. INTRODUCTION

As oil is the fuel of the industrial society, software is the fuel of our current information society. Today, people use software all the time and everywhere. Future economies are expected to require more automation, which will increase the demand for software developers. According to the Bureau of Labor Statistics, there will be more demand for computing jobs in the future [1]. By 2024, more than one million computing jobs will be available. Statistics show that there is more demand for computing jobs than there is supply from universities [2]. Moreover, statistics from big tech companies showed that the CS field is currently dominated by White and Asian males. The Google diversity report of 2017 showed that the percentage of males and females in tech jobs were 80% and 20% respectively. The employee racial percentages were 53% White and 39% Asian and only 1% Black [3]. Similarly, the Apple diversity report [4] showed that the male employee percentage was 77% while its female counterpart lingered at 23%. Most employees were White (52%) and Asian (31%). The third report from Facebook was not different. The male percentage was 81% and the female was 19%, while 45% of the employees were White and 49% were Asian [5].

To address these problems, this study aims to increase the interest of high school students in pursuing a CS degree. It also aims to encourage and increase the motivation of female and underrepresented racial groups towards computer programming.

Many educators found Block-based programming languages to be useful and suitable for introducing school students to coding. However, these are not real programming languages that are used by real developers, nor ones that are taught in university courses. Such languages could be suitable for middle school students, while high school students are ready to code with a real programming language. This could help them make the right decision about their university degree. Programming with a real language like JavaScript may not be as fun and easy as Block-based programming, however, it is more powerful and flexible as a student moves from beginner to intermediate and advanced levels.

II. LITERATURE REVIEW

There are many tools that have been designed to encourage K12 students in coding like Scratch [6] and General Purpose (GP) from MIT [7] and Code Studio from Code.org [8]. Many educators and designers support Block-based programming as the best and easiest way to teach computer science. A study that was conducted to compare interactive Block-based programming and text-based

978-1-5386-5398-2/18/$31.00 ©2018 IEEE

programming found that most of the high school students preferred Block-based programming and found it easier than text-based programming [9]. However, the study states that Block-based programming has some drawbacks. It is less powerful, slower than text-based programming, and inauthentic. Another study discussed other drawbacks [7]. It stated that Block programming takes more screen space than real programming language, and modifying a long program with dragging and dropping is slower than modifying the text-based program. DiSalvo (2014) conducted a study with African American high school students that included exposing students to Block-based and text-based programming languages. Alice visual drag and drop programming language was used to teach coding in the first four weeks, while Jython, a text-based version of Python, was used in the next four weeks. Students' interviews showed that most of the students found the text-based language more suitable for their age [10].

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 emphasize the role of the keyboard in programming [11]. They state that students should get to use the keyboard when coding to prevent them from getting bored when they move to the intermediate and professional levels. They suggest activating the keyboard's role in the blocking environment.

In a study funded by a National Science Foundation grant, Al-bow et al. (2009) found that the integration of art and design in a project-based learning model increased student interest and knowledge in computer programming. The Greenfoot development environment was used as a treatment in that study [12]. Greenfoot is an open source development environment developed by researchers at the University of Kent in the UK. They have developed an environment that combines the best between text and block programming [13]. Greenfoot simplified programming with an object-oriented programming language like Java and made it more interesting for the school students. This development environment is free, but it is not an online tool. Students need some instruction to install it. The researchers in the University of Kent provided tutorials to install and use this object-oriented development environment.

To measure students' motivation, the literature was reviewed to find the suitable theoretical framework. Motivation theories that include intrinsic and extrinsic motivation were used in designing the survey items. Intrinsic motivation measures the student's engagement in the learning process by internal reasons such as challenge, curiosity, mastery, or enjoyment. A high rank on this scale shows an interest in learning a programming language. On the other hand, the extrinsic motivation measures the student's engagement in the learning process by external reasons such as grades, rewards, performance, and evaluation by competition with others [14].

III. METHODOLOGY

Three groups of students from Ann Arbor, Michigan, public and private schools were targeted in this experimental study. The first group (G1) had 15 teaching hours in a summer camp week. The second group (G2) had five afterschool teaching hours on five different days. The third group (G3) had three teaching hours on the same day. Students in G1 and G2 attended the coding camp and afterschool workshop based on their own interest, while students in G3 were exposed to the coding workshop as a school activity. Human subject approvals were obtained before conducting the study. Parents' approvals were also required for school students. Ann Arbor was chosen for its diversity. Pioneer High School is the number one school in academic performance in the city and ranked among the top ten schools in Michigan. This school has around 1600 students [15]. MIA is a small private school in Ann Arbor. It was targeted for more diversity.

A pretest-posttest survey was used to measure the study variables before and after taking a programming course. The demographic data were also collected and analyzed. Gender and race were used as moderating variables in this study.

A. Study Variables and Hypotheses

The study variables are shown in Table I and the following hypotheses were examined in this study:

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

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

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

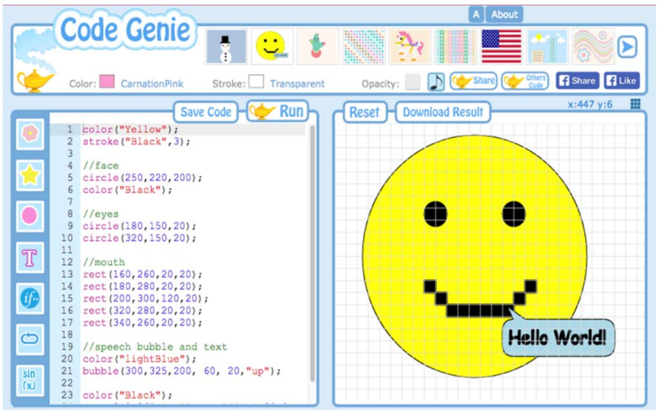
H4: Integrating art and animation in teaching programming increases students' motivation to write and share more code.

TABLE I. THE STUDY VARIABLES

Variable Name	Variable Meaning
DI	CS Degree Interest
PK	Programming Knowledge
RPP	Real Programming Preference
MCS	Motivation for Code Sharing

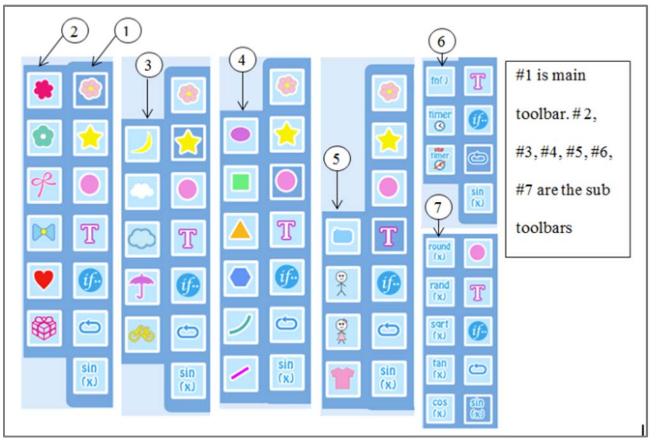
B. Treatment

This study introduces a new web-based development tool that high school students can use to learn a real programming language with the integration of art, animation and code sharing. The tool is called Code Genie, and it is an open source development tool that is available online under the domain name theCodeGenie.com. In this development environment, the main focus was on the art, animation, and the share features. Fig. 1 shows a screenshot of the development environment and Fig. 2 shows the submenus of the main menu on the left hand side of Fig. 1.



```
color("Yellow");
stroke("Black",3);

//face
circle(250,220,200);
color("Black");

//eyes
circle(180,150,20);
circle(320,150,20);

//mouth
rect(160,260,20,20);
rect(180,280,20,20);
rect(200,300,120,20);
rect(320,280,20,20);
rect(340,260,20,20);

//speech bubble and text
color("lightBlue");
bubble(300,325,200, 60, 20,"up");

color("Black");
```

Fig. 1. Code Genie Development Environment



2
1
3
4
5
6
7
#1 is main toolbar. #2, #3, #4, #5, #6, #7 are the sub toolbars

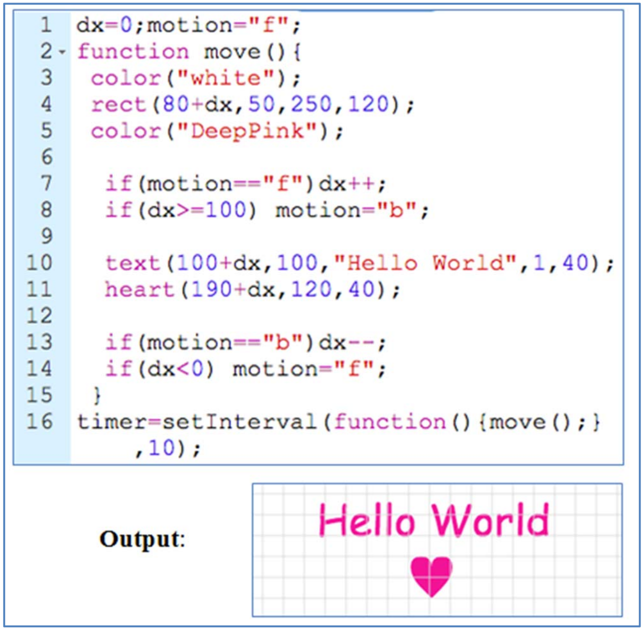
Fig. 2. Submenus in the Code Genie Development Environment

JavaScript is the language that the students can learn using this tool. JavaScript was chosen for its simplicity and popularity among web and frontend developers. The Stack Overflow 2017 Annual Survey shows that JavaScript is one of the most popular languages among developers [17]. Similarly, the data from GitHub indicated that JavaScript was the most popular among the shared projects [18]. Moreover, the researchers of this paper think that JavaScript should be taught instead of Java for its simplicity and compatibility with most Internet browsers. Almost every website uses JavaScript which is supported by all Internet browsers. Unlike other languages, JavaScript does not need a special development environment.

The software that was used to build the Code Genie tool includes HTML, CSS, JavaScript for the front-end, and PHP, MySQL for the back-end and the database.

The Human Computer interface (HCI) rules were considered in the design stage. This includes consistency, usability, simplicity, informative feedback, preventing errors, permitting easy reversal of actions, keeping users in Control, and reducing short-term memory load [16].

The first author, who has more than six years computer programming teaching experience, taught in the three coding camp/workshops. PowerPoint presentations and the Code Genie tool were used to explain the basic programming concepts in JavaScript language with art and animation examples. The teaching material included explaining variables and arithmetic operators such as variable assignment, multiplication, division, modulus, increment, and decrement operations. The two types of control statements were also explained. These are the conditional statements: "if", "if-else", and "switch" statements, and the loop statements: the "for" and "while" statements, in addition to the logical and comparison operators. Arrays, String, Function, Math Functions, and Timer concepts were also explained with art examples. Fig. 3 shows an example that includes animation where the output keeps moving forward and backward. Function, if-statement, and Timer were used in this example.



```
dx=0;motion="f";
function move(){
  color("white");
  rect(80+dx,50,250,120);
  color("DeepPink");

   if(motion=="f")dx++;
   if(dx>=100) motion="b";

   text(100+dx,100,"Hello World",1,40);
   heart(190+dx,120,40);

   if(motion=="b")dx--;
   if(dx<0) motion="f";
 }
timer=setInterval(function(){move();}
     ,10);
```

Output:

Hello World

Fig. 3. Teaching Example in the Coding Workshops

C. Survey Questionnaire

In the pretest and in the posttest, students were asked several questions to measure their CS Degree Interest (DI), their Preference to Real Programming language (RPP), and their Motivation for Code Sharing (MCS). These variables were measured using survey items with a 5-point Likert scale: one through five, representing "Strongly Disagree" and five "Strongly Agree" respectively.

TABLE II. PROGRAMMING QUESTIONS

Question #	Programming Concepts
Q1	Understanding Variable Assignment
Q2	Understanding Variable Addition
Q3	Understanding Variable Multiplication
Q4	Understanding the for-loop statement
Q5	Understanding the if-statement
Q6	Understanding the if-else statement
Q7	Understanding the if-else statement with art element
Q8	Understanding the for-loop with art element
Q9	Understanding the switch-statement
Q10	Understanding the Math Function
Q11	Understanding the concept of Arrays
Q12	Understanding the concept of Function

To measure students' programming knowledge, the survey questionnaire included different programming questions that were used to reveal students' understanding of the different programming concepts. Table II shows the programming concepts that were measured in this experimental study. It also shows the question numbers that were used to measure those programming concepts.

D. Sample and Demographic data

The demographic data of the students who participated in this study is shown in Table III. "All Students" column represents the combination of students in all three groups.

TABLE III. DEMOGRAPHIC DATA

Race	**N=65**	**All Students (G1+G2+G3)**	**Female=32 49%**	**Male=33 51%**
American Indian	1	2%	0	1
Asian	19	29%	11	8
Asian Indian	10	15%	2	8
Black	6	9%	3	3
Hawaiian	2	3%	2	0
Middle Eastern	20	31%	13	7
White	7	11%	1	6

The numbers of male and female students were almost equal. Students of G1 and G2 were mostly Asian and Asian Indian while G3 students were mostly Middle Eastern. The number of the students in G1, G2, and G3 were 18, 14, and 33 respectively. The total number of students was 65 students: 33 males and 32 females. The number of Asian students was 19 and the Middle Eastern was 20 students. The numbers of other racial groups are shown in Table III. Most of the females who attended the camp upon their own interest were Asian.

IV. RESULTS

This section discusses the results of the four variables in this study to accept or reject the four hypotheses. 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. Gender and race are the moderating variables in this study. The results are discussed for each variable in the following sections.

A. CS Degree Interest

To examine students' interest in pursuing a degree in computer science after graduation from high school, five survey items were used to measure the Degree Interest (DI) variable. Table IV shows that the mean value for the male students was higher than the mean value for the female students for this DI variable. However, the results for the male students was not significant (p=0.667). In other words, male students did not change their answers significantly in the posttest. The results for the female students were significant t(31) = -4.06, p=0.002. Among different racial groups, White students had the highest mean value for the DI variable in both tests. The result was statistically significant for this group t(6) = -2.50, p=0.047, while it was not significant for any other racial group.

The results for all students was significant t(64) = -2.681, p=0.009. The p-value is less than α or 0.05. Hypothesis H1 is accepted and the null hypothesis is rejected. This indicates that the integration of art and animation in teaching computer programming increased students' interest in pursuing a degree in CS. Fig. 4 shows that the agreement improved or the interest in the CS degree increased in the posttest.

TABLE IV. T-TEST RESULTS FOR THE DI VARIABLE

		Mean		**t-test**	
	N	**Pretest**	**Posttest**	**t**	**p**
All Students	65	3.18	3.48	-2.681	0.009
Female	32	2.81	3.34	- 4.063	0.002
Male	33	3.55	3.61	-0.421	0.677
Asian	19	3.68	3.79	-0.567	0.578
Asian Indian	10	3.20	3.60	-1.309	0.223
Black	6	2.83	2.83	0.000	1.000
Middle Eastern	20	2.65	2.90	-1.097	0.287
White	7	3.71	4.43	-2.500	0.047

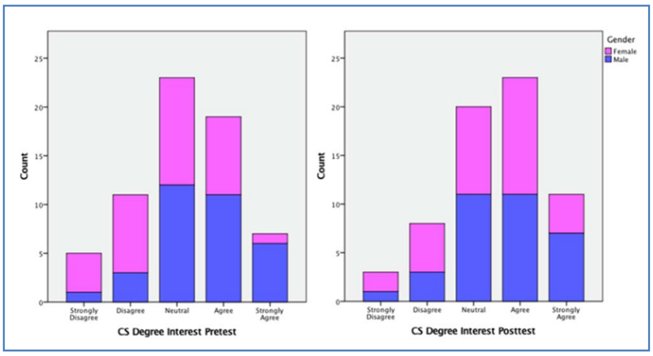


Fig. 4. CS Degree Interest Variable in Pre and Posttests

B. Programming Knowledge

The results show that the students' programming knowledge improved in the posttest. Fig. 5 shows that the percentages of students who gave the correct answer in the posttest were higher than the percentages in the pretest for all programming questions. It also shows that the percentages of the "Don't Know" answers in the posttest were lower than the percentages in the pretest for all programming questions. In other words, more students gave answers in the posttest or their willingness to give answers improved in the posttest and fewer students chose to not give answer or the "Don't Know" option. To measure the programming knowledge (PK) varibale, the total scores were collected by adding the correct answers for each student. The results show that the mean value of the students' programming knowledge variable increased from 3.86 in the pretest to 5.4 in the posttest. The mean values of the male were higher than the mean values for the female students in both tests. The mean increment for the male was higher than the mean increment for the female students. The mean value for the female students increased from 3.6 to 4.8, while it incresed from 4.07 to 5.94 for male students, as shown in Fig. 6.

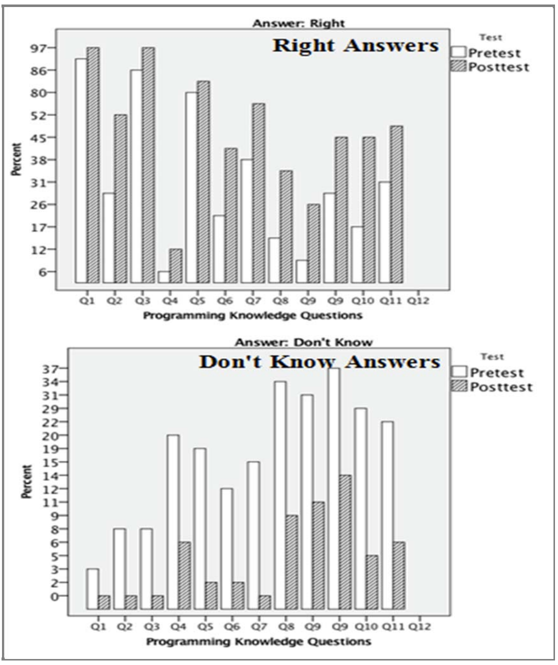


Fig. 5. Programming Questions, the Right and the Don't Know Answers.

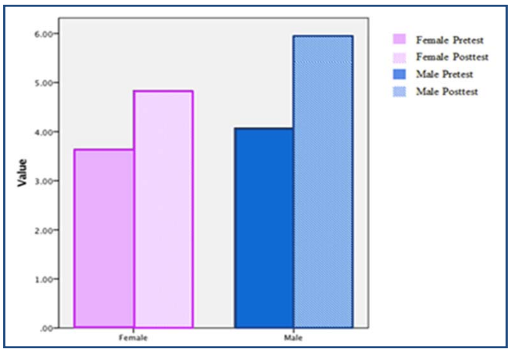


Fig. 6. Gender vs. the Programming Knowledge Mean in Pre and Posttests.

The t-test results for the programming knowledge (PK) variable are shown in Table V. The results were significant for all students t(64)=-7.7, p=0.000. The mean values for all students increased significantly in the posttest. The results were also statistically significant for female t(31)=-3.99, p=0.000 , and male students t(32)=-7.27, p=0.000. The results were significant for most of the racial groups. This includes the Asian, Asian Indian, Middle Eastern and White groups. The mean value for the Black students was close to the mean value of the Middle Eastern students in both tests, but it was not increased significantly in the posttest, and the t-test result was not significant for the Black Students.

TABLE V. T-TEST RESULTS FOR THE PK VARIABLE

		Mean		t-test	
	N	Pretest	Posttest	t	p
All Students	65	3.86	5.40	-7.703	0.000
Female	32	3.64	4.83	-3.418	0.000
Male	33	4.07	5.94	-7.275	0.000
Asian	19	5.09	6.50	-4.036	0.001
Asian Indian	10	3.91	5.80	-3.638	0.005
Black	6	3.86	4.44	-1.131	0.310
Middle Eastern	20	3.15	4.46	-3.433	0.003
White	7	3.16	5.88	-4.871	0.003

Since the mean value of programming knowledge increased significantly for all students, hypothesis H2 is accepted and the null hypothesis is rejected.

C. Real Programming Preference

Similar to the DI variable, five survey items were used to measure the students' preference to the real programming language over the Block-based programming language. Fig. 7 shows the descriptive statistics for this variable (RPP).

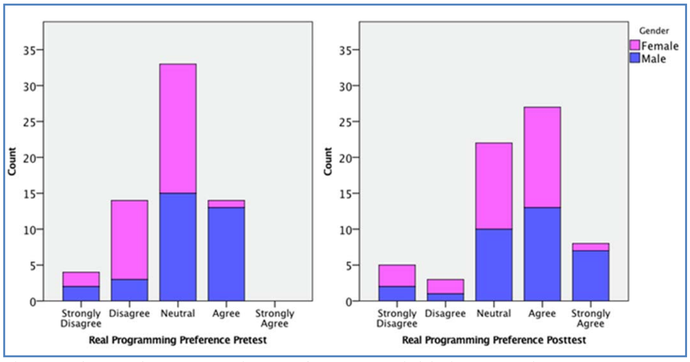


Fig. 7. The Real Programming Preference Variable in Pre and Posttests.

The mean value for this variable increased significantly in the posttest, or more students chose the "Agree" and the "Strongly agree" options in the posttest. Table VI shows that the t-test results were significant for all students t(64) = -5.1, p=0.00 and for the female t(31) = -4.73, p=0.00, and male students t(32) = -2.77, p=0.00. The results were significant for the Asian and Middle Eastern students.

TABLE VI. T-TEST RESULTS FOR THE RPP VARIABLE

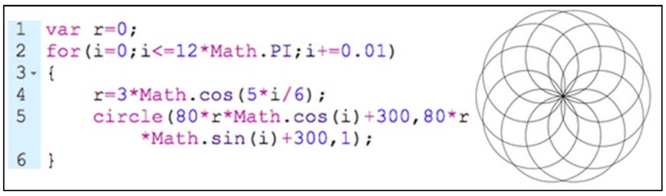
		Mean		t-test	
	N	Pretest	Posttest	t	p
All Students	65	2.88	3.46	-5.141	0.000
Female	32	2.56	3.25	-4.739	0.000
Male	33	3.18	3.67	-2.775	0.009
Asian	19	3.05	3.89	-4.800	0.000
Asian Indian	10	3.00	3.80	-2.058	0.070
Black	6	2.67	2.83	-0.415	0.695
Middle Eastern	20	2.55	2.90	-2.101	0.049
White	7	3.29	4.00	-1.508	0.182

D. Motivation for Code Sharing

Eleven survey items that include intrinsic and extrinsic motivation were used to measure the MCS variable. The mean value of this variable was higher than the other two variables (RPP and DI). It was 3.16 in the pretest and rose to 3.4 in the posttest. Most of the students agree that writing a code that include art and animation increases their motivation to share their artwork. Table VII shows the number of artwork that was shared in the three coding workshops. It also shows the number of code lines that were written and shared during the workshops. Fig. 8 shows a sample of the artwork that was shared by an Asian, male student in G1. Fig. 9 shows the improvement on the agreement for the MCS variable.

TABLE VII. The Shared Artwork

G#	Coding Time (in Hours)	N	Number of the Written Code Lines	Number of the Shared Artwork	Number of the Artwork that Include Animation
G1	15	18	4880	128	47
G2	5	14	546	22	8
G3	3	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);
}
```

Fig. 8. A Sample of the Shared Artwork by an Asian Male Student in G1.

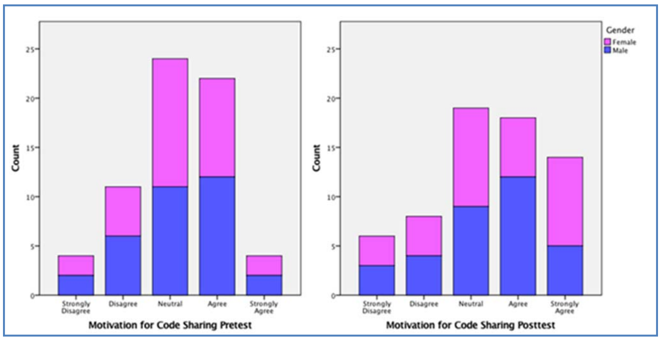


Fig. 9. The Motivation for Code Sharing Variable in Pre and Posttest

Table VIII shows that the t-test result was significant for all students t(64) = -2.03, p=0.046. The result approached the line of significance for the female students t(31) = -1.86, p=0.071. The mean values for the male students were higher than females in both tests. However, the results were not significant for male students. The male students' agreement percentage improved, but it didn't change significantly in the posttest. For the female students, the result approached the level of significance, t(64) = -1.869, p=0.071. No significant result was found for any racial group in this variable.

TABLE VIII. T-test Results for the MCS variable

		Mean		t-test	
	N	Pretest	Posttest	t	p
All Students	65	3.16	3.40	-2.034	0.046
Female	32	3.15	3.43	-1.869	0.071
Male	33	3.18	3.36	-1.063	0.296
Asian	19	3.79	4.11	-1.837	0.083
Asian Indian	10	3.30	3.50	-0.802	0.443
Black	6	3.17	3.17	0.000	1.000
Middle Eastern	20	2.80	2.75	0.326	0.748
White	7	2.43	3.43	-1.620	0.156

V. Conclusion

This study introduced a new development environment for teaching real programming language for high school students. The suggested development environment focuses on learning JavaScript language. This programming language was chosen because it is a popular and simple programming language that is suitable for beginners.

The results of this study show that integrating art and animation in teaching computer programming for high school students increases the students' interest in programming and in pursuing a degree in CS after graduation from high school. It also increased their programming knowledge, their preference to real programming over the Block-based programming language, and their motivation to write and share coding. The paired sample t-test analysis results accepted the four research hypotheses and rejected the null hypotheses. This indicates that integrating art and animation in teaching programming could encourage high school students to pursue a degree in CS, and therefore, close the gender and racial gap in the software field.

Acknowledgment

The researchers of this study would like to thank the administrations, students and their parents of the two schools participated in this study.

References

[1] "Software Developers : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics", Bls.gov, 2018. [Online]. Available: https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm. [Accessed: 05- Feb- 2018].

[2] "The U.S. STEM Workforce: The Real Story", Itif.org, 2018. [Online]. Available: https://itif.org/publications/2013/05/16/us-stem-workforce-real-story. [Accessed: 05- Feb- 2018].

[3] "Making progress on diversity and inclusion", Google, 2018. [Online]. Available: https://blog.google/topics/diversity/making-progress-diversity-and-inclusion/. [Accessed: 05- Feb- 2018].

[4] "Inclusion & Diversity", Apple, 2018. [Online]. Available: https://www.apple.com/diversity/. [Accessed: 05- Feb- 2018].

[5] "Facebook Diversity Update: Building a More Diverse, Inclusive Workforce | Facebook Newsroom", Newsroom.fb.com, 2018. [Online]. Available: https://newsroom.fb.com/news/2017/08/facebook-diversity-update-building-a-more-diverse-inclusive-workforce/. [Accessed: 05- Feb- 2018].

[6] "Scratch - About", Scratch.mit.edu. [Online]. Available: https://scratch.mit.edu/about. [Accessed: 05- Feb- 2018].

[7] Monig, Y. Ohshima and J. Maloney, "Blocks at your fingertips: Blurring the line between blocks and text in GP", 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond), 2015.

[8] "Code.org - Learn on Code Studio", Studio.code.org, 2018. [Online]. Available: https://studio.code.org. [Accessed: 05- Feb- 2018].

[9] D. Weintrop and U. Wilensky, "To block or not to block, that is the question", Proceedings of the 14th International Conference on Interaction Design and Children - IDC '15, 2015.

[10] B. DiSalvo, "Graphical Qualities of Educational Technology: Using Drag-and-Drop and Text-Based Programs for

Introductory Computer Science", IEEE Computer Graphics and Applications, vol. 34, no. 6, pp. 12-15, 2014.

[11] N. Brown, M. Kolling and A. Altadmri, "Position paper: Lack of keyboard support cripples block-based programming", 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond), 2015.

[12] M. Al-Bow, D. Austin, J. Edgington, R. Fajardo, J. Fishburn, C. Lara, S. Leutenegger and S. Meyer, "Using Greenfoot and games to teach rising 9th and 10th grade novice programmers", Proceedings of the 2008 ACM SIGGRAPH symposium on Video games - Sandbox '08, 2008.

[13] N. Brown, A. Altadmri and M. Kolling, "Frame-Based Editing: Combining the Best of Blocks and Text Programming", 2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE), 2016.

[14] R. Ryan and E. Deci, "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions", Contemporary Educational Psychology, vol. 25, no. 1, pp. 54-67, 2000.

[15] "Pioneer High School Student Body". usnews.com, 2014. [Online]. Available: http://www.usnews.com/education/best-high-schools/michigan/districts/ann-arbor-public-schools/pioneer-high-school-9694/student-body. [Accessed: 05- Feb- 2018].

[16] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs and N. Elmqvist, Designing the user interface, 6th ed. Pearson, 2016.

[17] "GitHub Octoverse 2017", GitHub Octoverse 2017, 2018. [Online]. Available: https://octoverse.github.com. [Accessed: 10- Apr- 2018].

[18] "Stack Overflow Developer Survey 2017", Stack Overflow, 2018. [Online]. Available: https://stackoverflow.com/insights/survey/2017. [Accessed: 10- Apr- 2018].