Research Article Journal of Pedagogical Research

Effect of Javascript on middle schoolers' attitute toward computer course

Ibrahim Bastug[1], Kagan Kircaburun[2]

[1] *Ministry of National Education, Kocaeli, Turkey*

[2] *Duzce University, Faculty of Education, Duzce, Turkey*

Article Info

Article History
Submitted: 9 November 2017
Revised: 7 December 2017
Published: 8 December 2017

Keywords
Scratch software
Programming instruction
code.studio.org
JavaScript
Attitude toward computer

Abstract

In recent years, the number of students who have chosen computer science related undergraduate departments has decreased rapidly. Moreover, students who prefer computer science and programming departments appear to be unsuccessful in programming classes due to the complexity and frustrating difficulties of programming learning. One of the solutions to overcome this problem is to make the students adapt and warm up to the programming issues from the lower grades. In order students, who are in the age of development at primary and junior high school level, to love and learn programming, various enjoyable methods are applied. Scratch software is one way of teaching programming to secondary school students. This study investigated how the attitudes of 5th and 6th grade students towards information technology course changed after Scratch software and programming language and Javascript programming language were used. A 2x2x2 factorial experimental design was used in the study. According to the analysis, it was found that both methods are effective on the students' attitudes. But there was no significant difference between them.

1. Introduction

Computers that have entered into every aspect of our lives from business to entertainment sector and software that enables us to control them have gradually increased in importance. In parallel to this phenomenon, there has also been a growing need for people who know how to improve software (Karabak & Güneş, 2013). However, the literature has elaborated in detail that it is not easy to learn the programming languages we need to develop software.

Research shows that the number of students choosing to study computer science has been in rapid decline in recent years (Heersink & Moskal, 2010; Hoegh & Moskal, 2009 as cited in Başer, 2013). One reason is that students are not interested in and do not have leanings towards software development. Additionally, software developers in movies and series are depicted as unattractive types with glassed and pimples, which have led young people to be very cool towards programming work (Rodger et al., 2009). Punctuation marks and design settings of common programming languages used to develop software also cause negative trends in students' attitudes and perceptions towards programming. Attitude is the status of being prepared to an object or a

Address of Corresponding Author

Kaan Kircaburun, Research Assistant, Duzce University, Faculty of Education, Computer Education and Instructional Technology, 81620, Duzce, Turkey

✉ kagankircaburun@duzce.edu.tr

iD 0000-0002-8678-9078

case (Gökdaş, 2008). Attitude influences how we learn and behave (Maio & Haddock, 2009 as cited in Özyurt & Özyurt, 2015).

Özyurt and Özyurt (2015) surveyed 325 students studying Computer Technology at a vocational school and reported that they had positive attitudes. However, according to Başer (2013) surveying 220 students studying computer teaching and computer engineering, it would not be safe to say that attitudes towards programming are positive. On the other hand, engineering students have a more favorable attitude compared to teaching students and male students have also a more favorable attitude compared to female students. In their study conducted with university students attending the information technology course, Hongwarittorrn and Krairit (2010) have found that there is no significant correlation between attitude and test scores (as cited in Özyurt & Özyurt, 2015). In order to clear programming of such negative attitudes, perceptions and disinterest as reported by previous research, it is important to takes efforts starting as early as the middle school years when students begin to make career choices (Rodger et al., 2009). The importance attached to programming education in some developed and developing countries has led to the introduction of programming education to middle school curricula (Karabak & Güneş, 2013). In Turkey, the information technology and software course was added as a compulsory course to the 5th- and 6th-grade curricula. Among the topics of this course, the way by which programming is taught depends on teachers' initiative and schools' facilities.

Research has indicated the great difficulty of programming even for undergraduate students (Feldgen & Clúa, 2004) and the negatives attitudes of even university students towards programming (Başer, 2013). Accordingly, a need to design a fun and simple environment has arisen to change this negativity (Kert & Uğraş, 2009). Many pieces of software have been developed to solve this problem (Karabak & Güneş, 2013). Among them, the Scratch software is considered suitable for middle school students and widely used.

Scratch software is a piece of software that uses a simple interface without punctuation marks allowing you to easily design games, animations, and simulations, and to create designs simply by dragging code blocks (Karabak & Güneş, 2013). Additionally, its interface allows children to design games as they wish and help them to learn more effectively through play. Indeed, learning through play is one of the most effective learning methods that allow students to acquire deep learning (Erekmekçi & Fidan, 2012). Since the process of game development involves instantaneous feedback to the problem-solving process, designing and playing computer games is highly motivational and students develop different skills and strategies (Collins et al., 1996 as cited in Hong, Fadjo, Chang, Geist & Black, 2010).

With their structures targeting thinking skills, programming languages offer great contribution to cognitive development (Kert & Uğraş, 2009). Additionally, programming learning also helps students to acquire high-level thinking skills and thus has a positive impact on success in other courses (Karabak & Güneş, 2013; Lau & Yuen, 2009).

1.1. Scratch

Scratch is an online community where children can program and share interactive stories, games and animations (Monroy-Hernández & Resnick, 2008). Users do not have to know a specific programming language. A program is written by putting code blocks together in the correct order. As it has such an easy and understandable interface, children and even primary school students can easily use Scratch. The most attractive feature of the Scratch is that it can easily produce games (Nikiforos, Kontomaris & Chorianopoulos, 2013). Students can share games they produce with the Scratch community. Members of the community can also review and improve these games. Resnick proposed using the advantage of community members' reviews and improvements in programming education (Brennan & Resnick, 2012). Reviewing and improving an available game is preferred rather than creating a new one.

1.2. Code.Studio.org

Code Studio, an online learning platform of Code.org, is used in more than 90 classes. It offers successive practices and artistic activities for students of all ages ranging from secondary to high school students. As it is reported, there are 3 million registered members of the site which allows students to use its applications without registration. Code.org site allows teachers to teach cycle, event, condition, algorithm, digital citizenship and more in both home and classroom settings (Wilson, 2015).

1.3. JavaScript

The web has become the target platform in social networking, gaming and other areas of uses. This situation causes languages used in web design such as JavaScript to come to the fore although they are old (Mikkonen & Taivalsaari, 2007). Thus, it is important to learn JavaScript to design applications that can work on any platform

Considering that programming has become one of the most important needs of our time, the purpose of this research is to investigate the effect of programming taught using the JavaScript programming language supported by the website code.studio.org and the Scratch software on secondary school 5th- and 6th-grade students' attitudes towards the computer course.

1.4. Research purpose and problems

The purpose of the research is to observed how 5th- and 6th-grade students' attitudes towards computer are changed by the Scratch software as an entertaining design environment, and the JavaScript programming language supported by code.studio.org

1.4.1. Sub-problems

- Is there a difference in the effects of the JavaScript programming language taught by the Scratch software and code.studio.org supported in changing students' attitudes towards computer?
- Does the availability of Internet at home have an effect on students' attitudes toward computer?
- Does the presence of computer at home have an effect on students' attitudes toward computer?
- Is there any difference in the attitudes of male and female students towards computer?

The researchers found no previous Turkish work on secondary school students affected by Scratch software and Code.studio.org in the literature. We did not find a related result for "scratch + computer + attitude" at scholar.google.com/. No related research has been found in the results with scratch and attitude word. It is an original work since there is no other experimental Turkish work that measures the programming effect of the scratching software and the effect of teaching the Javascript programming language supported by Code.studio.org on attitudes towards computer. Nikiforos et al. (2013) conducted a similar study using smaller groups and it was observed that Scratch software produced positive results in high school programming education.

2. Method

The study used the Attitudes toward Computer Scale consisting of 21 items and developed by Şerefhanoğlu, Nakiboğlu and Gür (2008). In the study, there are 2 independent variables whose effects are examined on a dependent variable. The dependent variable of the research is the attitude towards computer and the independent variables are the method and class levels used. Methods used in the independent variables are Scratch software and classical teaching method, while grade levels are 5th and 6th grades. Participants in the study were measured once in each experimental condition and unrelated measurements were obtained. The basic and common effects of these independent variables on the dependent variable were investigated. Taking these into consideration, the model of the research is a 2x2x2 factorial design.

Table1
Research Design

Group		Pre-test	Process	Post-test
5th Grade	G1	Attitudes toward Computer Scale	Teaching JavaScript code.studio.org	Attitudes toward Computer Scale
6th Grade	G2			
5th Grade	G3		Teaching Programming using the Scratch	
6th Grade	G4			

2.1. Study Group

Since the study was an experimental design, the study group was selected instead of the universe sample selection. In the studies involving many experimental models made by Kabaca and Erdoğan (2007, p. 58), the universe stated that the sample was selected and that the wrong generalizations were reached as a result. The fact that the results obtained from experimental studies are wrong for generalization to the environment is stated that the results are valid only for the applied groups (Coşar, 2013).

The study groups consist of a total of 114 students. A total of 58 students, 32 of which are 5th grade (17 females and 15 males) and 26 of which are 6th grade (10 females and 16 males) constituted the control group. The experimental group consisted of 56 students, 32 of which are 5th grade (17 females and 15 males) and 24 of which are 6th grade (11 females and 13 males) students. The reason for choosing Scratch software for the experiment group is that it is a fun option to learn how to program. In the Scratch experiment group, programming training with scratch software was given for 6 weeks. In the group trained with Code.studio.org, students were allowed to use the applications at code.studio.org for 2 weeks, then 4 weeks.

Table 2
Study Groups

Group			Female	Male	Total
5th Grade	G1	Scratch	17	15	32
6th Grade	G2	Scratch	10	16	26
5th Grade	G3	code.studio.org + javascript	17	15	32
6th Grade	G4	code.studio.org + javascript	11	13	24
Total			55	59	114

2.2. Data collection tools

The study used the 21-point Likert-type Attitudes toward Computer Scale developed by Şerefhanoğlu (2008) et al. to measure students' attitudes. The The Kaiser-Meyer-Olkin (KMO) coefficient of the scale used was 0.88 and the Barlett coefficient was found significant. It is stated by the developer that the items are gathered under 4 factors and that they have the titles of "trust", "willingness", "reluctance" and "belief", respectively. The Cronbach's Alpha total reliability coefficient of the scale was 0.87. When the reliability coefficient of the subscales of the scale have been examined; .75, .72, .72, and .72 have been found for the first, second, third, and fourth factor respectively. The total variance explained as a result of the scale development study is 42.6%.

(Şerefhanoğlu et al., 2008). The highest score that can be taken from the scale is 105 and the lowest score is 21.

2.3. Data collection

Before entering the programming topic, all students were asked to write Şerefhanoğlu (2008) and others. After applying the attitude scale for the computer with 21 questions developed by the Ministry of Education, the subject of programming started to be given.

All the experimental groups participating in the study were students with information technology and software course programming with scratch software.

2.4. Data analysis

There are 8 students who are at the beginning of the study but who are subsequently taken out of school for various reasons. The data belonging to these students were not included in the analyses. SPSS 20 package program was used in the analysis of the data.

Table 3
Normality Test

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	*p*	Statistic	df	*p*
Pre-test Attitudes Mean	.07	106	.20*	.99	106	.27
Post-test Attitudes Mean	.07	106	.20*	.98	106	.13

First, the assumptions necessary for the application of the ANOVA statistic to the data obtained are met. For this purpose, the normality test was performed to see whether the pretest-posttest measures of the dependent variable had a normal distribution at the stage of the belonging, and the Kolmogorov-Smirnov test results were examined because the size of the group was large. The *p* values for the pretest and post-test were found to be greater than .05. Here, it is found that the data show a normal distribution.

Table 4
Levene's Test of Equality of Variances

F	df_1	df_2	*p*
1,354	1	104	.25

Levene's homogeneity test was used to find out if the observed groups had equal variance in the population they belonged to and *p* value was found .25. Accordingly, the groups have equal variance in the population they belong to. Observations are independent of each other. According to these assumptions, ANOVA statistical method could be conducted.

3. Results

Data collected in the study were analyzed according to the sub-problems and presented below in order.

3.1. Is there a difference in the effects of the JavaScript programming language taught by the Scratch software and code.studio.org supported in changing students' attitudes towards computer?

Table 5
Post-test Attitudes Mean

Method Used	Mean	Standard Deviation	n
Scratch	3.94	.50	52
Code.studio.org+ javascript	3.75	.68	54
Total	3.84	.60	106

Table 5 summarizes the students' post-test scores of attitudes toward computer. Analysis of covariance (ANCOVA) was performed for the significance of the difference observed in attitude scores of students according to the method used.

Table 6
ANCOVA Test Results of Post-test Scores Corrected Based on the Pre-Test Scores of the Groups

Source of Variance	Sum of Squares	df	Mean of Squares	*F*	*p*	*η2*
Pre-test attitudes mean	9.521	1	9.52	35.289	.00	.26
method	.051	1	.05	.189	.67	.00
Error	27.790	103	.27			
Total	1603.947	106				

When the students' pre-test attitude scores and post-test attitude scores were examined, there was no significant difference between the effects of modifying the attitudes in the two methods used in the programming topic (p> 0.05). With the help of Code.studio.org supported Javascript programming language, it has been found that there is no difference in the influence of the use of Scratch software on the computer, both of which have the same effect without changing the attitudes of the students. This may be due to the Code.studio.org website, which has been used for two weeks to draw students' attention.

Table 7
Post-test Attitude Scores Corrected Based on the Pre-Test Scores of the Groups for Each Method Used

Method Used	Mean	Standard Error
Scratch	3.87^a	.07
Code.studio.org+ javascript	3.82^a	.07

The Scratch software and Javascript software supported by Code.studio.org showed that there was a difference between the pre-test attitude mean scores of groups. For this reason, the pre-test attitude scores were found to be 3.87 for the Scratch group and 3.82 for the javascript group supported by Code.studio.org as a result of the ANCOVA analysis of covariates.

Table 8
The pretest-posttest Attitude Mean Scores of the Scratch Group

	Mean	n	Standard deviation	Standard Error Mean
Pre-test Attitude Mean Scores	3.77	52	.49	.07
Post-test Attitude Mean Scores	3.94	52	.50	.07

Pre-test and post-test attitude scores of the group that was given programming training with Scratch software are shown in the table. According to the data, Scratch group's pre-test attitude mean was 3,77 while post-test attitude mean was 3,94. This indicates that the Scratch software has changed the attitudes of the students positively. A t-test was conducted to determine whether this change was meaningful.

Table 9
T-test results of the attitude scores of the Scratch Group

	Paired Differences			*t*	df	*p*
	Mean	Standard Deviation	Standard Error Mean			
Pre-test Attitude Mean Scores - Post-test Attitude Mean Scores	-.18	.51	.07	-2.483	51	.02

The results of the t test on the mean scores of the pretest-posttest attitude scores of the group to which the scratch software is applied are presented in tablature. According to the result of the t-test, the difference between the pretest-posttest results of the group that applied scratch software is significant (*p*<.05). As a result, Scratch software positively affects students' attitudes toward computers. This may be due to the simple, simple, fun nature of Scratch software and the willingness of students to develop games.

Table 10
The pretest-posttest Attitude Mean Scores of the group using the Javascript programming language supported by Code.studio.org

	Mean	n	Standard Deviation	Standard Error Mean
Pre-test Attitude Mean Scores	3.51	54	.52	.07
Post-test Attitude Mean Scores	3.75	54	.68	.09

The group pretest-posttest mean scores described in the programming topic using Code.studio.org supported Javascript programming language are presented in the table. According to these results, the mean of the pretest attitude scores was 3.51 while the mean of the post-test attitude scores was 3.75. When the attitude scores are compared, it is seen that there is a positive change. Whether or not this change is significant is examined by t-test.

Table 11
T-test results of the attitude scores of the group using the Javascript programming language supported by Code.studio.org

	Paired Differences			*t*	df	*p*
	Mean	Standard Deviation	Standard Error Mean			
Pre-test Attitude Mean Scores - Post-test Attitude Mean Scores	-.23	.60038	.08	-2.855	53	.01

The results of the t test on the mean of pre-test and post-test attitude scores of the group using the JavaScript supported by Code.studio.org are presented in the table above. According to the t-test results (*p*<.05), the difference between the pre-test and post-test results of the group implementing the JavaScript programming language supported by code.studio.org is significant. As a result, code.studio.org supported JavaScript programming language positively affects students' attitudes towards computers. This can be caused by the presentation of applications in the form of games in programming site at code.studio.org.

3. 2. Does the availability of Internet at home have an effect on students' attitudes toward computer?

The analysis results showed that the presence of internet access in the students' homes has a positive effect on the attitude towards computer. Sixty four students had internet access in their home, while 42 did not. Their pretest-posttest attitude scores were compared and it was observed that internet access at home was related to computer.

Table 12
Do you have internet access at home?

	Do you have internet access at home?	N	Mean	Standard Deviation	Standard Error Mean
Pre-test Attitude Mean Scores	Yes	64	3.65	.53	.07
	No	42	3.63	.52	.08
Post-test Attitude Mean Scores	Yes	64	3.80	.56	.07
	No	42	3.91	.67	.10

As seen in the table above, the attitude pre-test scores of those who have internet access at home were 3.65 while the post-test scores were 3.80. This shows that the average has changed positively. For those without internet access at home, the pretest-pretest scores were 3.63, while the posttest scores changed positively as 3.91.

Table 12
The effect of having internet access at home on attitudes score

	t-test for equality of means				
	t	df	*p*	Mean Difference	Standard Error Difference
Pre-test Attitude Mean Scores	.20	104	.84	.02	.10

The difference between the pre-test attitude scores was not found to be significant (*p*> .05). The t-test was used to determine whether the difference between post-test test scores was significant.

Table13
The effect of having internet access at home on attitudes score

	t-test for equality of means				
	t	df	*p*	Mean Difference	Standard Error Difference
Post-test Attitude Mean Scores	-.858	104	.39	-.10	.12

Looking at the table, there was no significant difference between the pre-test attitude averages of the internet access users and those without internet accesses (*p*>.05). This situation is thought to have been caused by the existence of centers that provide internet access to students by the municipality.

3.3. Does the presence of computer at home have an effect on students' attitudes toward computer?

Table 14
Attitude Mean Scores of those with and without a computer at home

	Do you have a computer at home?	n	Mean	Standard Deviation	Standard Error Mean
Pre-test Attitude Mean Scores	Yes	69	3.69	.54	.07
	No	37	3.54	.48	.08
Post-test Attitude Mean Scores	Yes	69	3.86	.60	.07
	No	37	3.82	.63	.10

Considering the attitude scores according to the presence of computers at students' home, the pre-test attitude score of those who have a computer at home was 3.69 while the pre-test attitude score of those who do not have a computer at home was 3.54. There was no significant difference between pre-test scores; posttest attitude scores were examined in order to determine whether the presence of a computer at home has an effect on students' attitudes towards computer.

Table 15
The difference between the Attitude Mean Scores of those with and without a computer at home

	t-test for equality of means				
	t	df	*p*	Mean Difference	Standard Error Difference
Pre-test Attitude Mean Scores	1.372	104	.17	.15	.12
Post-test Attitude Mean Scores	.327	104	.74	.04	.12

According to the t test results (*p*> .05) between the post-test attitude scores, there was not a significant difference between the attitude scores of the students with and without computers at home

3.4. Is there any difference in the attitudes of male and female students towards computer?

Table 16
Mean Scores by Gender

	Gender	n	Mean	Standard Deviation	Standard Error Mean
Pre-test Attitude Mean Scores	Female	50	3.62	.50	.07
	Male	56	3.65	.55	.07
Post-test Attitude Mean Scores	Female	50	3.83	.62	.09
	Male	56	3.85	.59	.08

Female students' pre-test attitude score for computer was 3.62 while male students' score was 3.65, female students' post-test attitude score was 3.83 while male students were 3.85. A t-test was conducted to examine the effect of the gender factor on the computer.

Table 17
Difference between mean scores by gender

	t-test for equality of means				
	t	df	p	Mean Difference	Standard Error Difference
Pre-test Attitude Mean Scores	-.278	104	.78	-.03	.10
Post-test Attitude Mean Scores	-.205	104	.84	-.02	.12

Both the pre-test and post-test attitude scores of girls and boys are close to each other. There was no significant difference between the pre-test and post-test attitude scores according to the t-test to examine the effect of the gender factor. It can be said that this is the effect of changing social roles.

4. Conclusion

Analyses of the collected data show that both the scratch software and the JavaScript programming language supported by the code.studio.org site have a positive effect on the computer's attitude toward the computer. However, these methods cannot be said to be a meaningful difference in terms of changing the attitude towards computers. Wang and Chen (2010) reported that the method in classroom learning with the scratch and classical method has no effect on students' understanding of programming concept but increases motivation.

Considering the post-test mean score of the attitudes toward computer was examined, there was no significant difference between the attitude score averages of the ones who were not at home computers. There was also no significant difference in the pretest-posttest attitude scores of computer users with and without internet access at home.

Considering students' pre-test ad post-test attitude scores of the students, there was no significant difference between them according to their genders. Nikiforos et al. (2013) reported that the students' attitudes toward programming were affected by scratch software. According to the results of the research, the girls have more positive attitudes.

The present study investigated whether there is a significant difference between the Scratch software and code.studio.org-supported JavaScript programming language in terms of altering attitudes towards computer. The following recommendations can be offered for further studies:

- A study comparing the traditional method and the Scratch software in altering secondary school students' attitudes towards programming,
- A study comparing the traditional method and the website code.studio.org in altering secondary school students' attitudes towards programming,
- A study comparing the traditional method and the JavaScript programming language in altering secondary school students' attitudes towards programming,
- A study comparing the traditional method and the Scratch software in altering secondary school students' attitudes towards computer,
- A study comparing the traditional method and the website code.studio.org in altering secondary school students' attitudes towards computer,
- A study comparing the traditional method and the JavaScript programming language in altering secondary school students' attitudes towards computer.

References

Başer, M. (2013). Developing attitude scale toward computer programming. *The Journal of Academic Social Science Studies*, 6(6), 199-215.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada* (pp. 1-25).

Coşar, M. (2013). *Effects of computer programming studies on academic success, critical thinking skills and programming-based attitudes in problem-based learning environment*. Unpublished Doctoral Thesis, Gazi University Educational Sciences Institude, Ankara.

Erekmekçi, M. ve Fidan, Ş. (2012). The effect of game design platforms to education and cultural change. *Batman University Journal of Life Sciences*, 1(1), 851-861.

Feldgen, M. & Clúa, O. (2004, October). Games as a motivation for freshman students learn programming. Paper presented at the *34th Annual Frontiers in Education* (pp.1079-1084).

Gökdaş, İ. (2008). [Attitudes toward computer]. (Editor: D. Deryakulu). [Socio-psychological variables in computer technologies] (125-150). Ankara: Maya Academy.

Heersink, D., & Moskal, B. M. (2010, March). Measuring high school students' attitudes toward computing. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 446-450). ACM.

Hong J., Fadjo C., Chang C.-H., Geist E., & Black J. (2010). Urban culture and constructing video games. In J. Herrington & C. Montgomerie (Eds.), *Proceedings of ED-MEDIA 2010--World Conference on Educational Multimedia, Hypermedia & Telecommunications* (pp. 1334-1337). Toronto, Canada: Association for the Advancement of Computing in Education (AACE).

Hongwarittorrn, N., & Krairit, D. (2010, April). Effects of program visualization (jeliot3) on students' performance and attitudes towards java programming. In *The spring 8th International conference on Computing, Communication and Control Technologies* (pp. 6-9).

Kabaca, T., & Erdoğan, Y. (2007). Examining of statistical properties of the thesis studies in the fields of science, computer and mathematics education. *Pamukkale University Faculty of Education Journal*, *2*(22), 54-63.

Karabak D. & Güneş A. (2013). Curriculum proposal for first class secondary school students in the field of software development. *Journal of Research in Education and Teaching, 2*(3), 163-169

Kert, S. B. & Uğraş, T. (2009). Programlama eğitiminde sadelik ve eğlence: Scratch örneği [Simplicity and fun in programming education: Scratch example]. *1st International Congress of Educational Research. 1st International Congress of Educational Research.*

Lau, W. W. F. & Yuen, A. H. K. (2009). Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy. *British Journal of Educational Technology*, 40(4), 696-712.

Mikkonen, T., & Taivalsaari, A. (2007). *Using JavaScript as a real programming language.* Mountain View, CA: Sun Microsystems.

Monroy-Hernández, A., & Resnick, M. (2008). Empowering kids to create and share programmable media. *Interactions, 15,* 50–53.

Nikiforos, S., Kontomaris, C., & Chorianopoulos, K.(2013). MIT scratch: A powerful tool for improving teaching of programming. *In Proc. 5th Conference on Informatics in Education.*

Özyurt, Ö., & Özyurt, H. (2015). Learning style based individualized adaptive e-learning environments: Content analysis of the articles published from 2005 to 2014. *Computers in Human Behavior*, *52*, 349-358.

Rodger, S. H., Hayes, J., Lezin, G., Qin, H., Nelson, D., Tucker, R. (2009). Engaging middle school teachers and students with alice in a diverse set of subjects. *In ACM SIGCSE Bulletin, 41*(1), 271-275.

Şerefhanoğlu, H., Nakiboğlu, C., & Gür, H. (2008). An investigation on computer attitudes of elementary school students: Balıkesir sample. *Elementary Education Online*, *7*(3), 785-799.

Wang, L. C., & Chen, M. P. (2010, April). Learning programming concepts through game design: A PCT perspective. *In Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2010 Third IEEE International Conference on (pp. 219-221).*

Wilson, C. (2015). Hour of code-a record year for computer science. *ACM Inroads*, *6*(1), 22-22.

How to cite this article:

Bastug, I. & Kircaburun K. (2017). Effect of Javascript on middle schoolers' attitute toward computer course. *Journal of Pedagogical Research, 1*(1), 43-53.