JOURNAL OF EDUCATIONAL
AND INSTRUCTIONAL STUDIES IN THE WORLD
November 2017, Volume: 7 Issue: 4 ISSN: 2146-7463

THE EFFECTS OF GAMING METHOD ON STUDENTS' PROGRAMMING ACHIEVEMENT

Dr. Raziye Demiralay
Alanya Alaaddin Keykubat University, Antalya- Turkey
raziyedemiralay@gmail.com

Abstract
In this study, the effects of using a game program developed for instruction of algorithms in programming lesson on students' programming achievement was investigated. The randomized pretest-posttest control group design was used. The study was carried out for 10 weeks in the Alanya Aladdin Keykubat University-Akseki Vocational School in the Computer Programming department students who are studying in the 2015-2016 academic year. The randomized sampling method has been applied in the determination of experimental and control groups In these groups, the same lecturer organized lessons. The data were collected by pre-test and post-test. For analyzing data were used SPSS 20.0 for descriptive statistics and t-test analysis with 5% significance level. It was determined that the achievement of the group that was supported by learning with game differed significantly from the control group.

Keywords: Programming education, gaming method, vocational school.

INTRODUCTION

Programming is the important skills of 21st century. Everything in our lives is programmable: for example, on wearable technologies, smartphones and tablets. Developed programs are applications developed to solve problems with programmable tools. They are developed in a special programming language. For this reason, programmers need to have programming/coding skills (Pea, Kurland, 1984; Salomon & Perkins, 1987).

Coding can be expressed in the form of problem solving with an abstract language (Blackwell, 2002). It is one of the basic subjects of computer science (CS). Although it is such an important subject, learning programming languages can be complicated and difficult for many students (Milne & Rowe, 2002). It requires creativity. This situation can lead students to develop negative attitudes towards lectures, develop their thoughts that they can not succeed, and consequently fail (Robins, Rountree & Rountree, 2003).

Some students use memorization instead of learning, and they are again failing. In order to increase the motivation of the students, digital game supported programming tools are effective. In this point, visual programming tools can be used as a solution (Maloney, Resnick, Rusk, Silverman & Eastmond, 2010).

Visual Programming Tools

When a classical programming language is started to be taught to students, a complexity perception of language can occur in students (Chang, 2005). If the environment in which the programming language is developed has a graphical interface, or if the language is a visual programming language, it can cause less complexity perception than classical language learning or can't cause complexity perception (Lewis, 2010). The simpler programming of visual blocks, without the need for coding, removes this problem almost entirely (Maloney, Peppler, Kafai, Resnick & Rusk, 2008).

Coding learning with visual tools allows students to see more concrete results of the applications they have developed. They have a design and production-oriented structure. Environments such as

Code.org, Scratch, Alice, and Code Game Lab are interactive visual programming environments that save students who use visual blocks from the complexity of the programming language semantics. It's easy to use with drag and drop technology. Language support is available. The students are trying to gain the programming skills with the related applications which are gained in the world of Coding Education (Resnick et al., 2009; Resnick, Martin, Sargent & Silverman, 1996; Rodger et al., 2009).

The most basic way to learn coding is to learn algorithms. Algorithm knowledge and skills of students are developing with these visual programming tools. Many countries in the world are still developing programming skills without improving their literacy skills (Kumar, 2014). For example, according to Code.org, with the Hour of Code activity in the code.org application, every year the program is being launched in the 180 countries (Eguchi, 2014). In these countries, these activities are supported at a high level (Layton, 2013). Participation in such visual programming activities is perhaps more than participation in traditional programming activities.

In many countries, like Spain, Hong Kong, Germany, programming education is included in school curricula (Wong, Cheung, Ching & Huen, 2015). At university level, classical programming trainings are carried out (Bennedsen & Caspersen, 2005). Generally, educators have used visual programming tools to facilitate lectures, to motivate and activate students, to provide individual learning and give homework (Asad, Tibi & Raiyn, 2016; Hwang, Shadiey, Wang & Huang, 2012).

In this study, a visual programming tool including with game learning was used in algorithm teaching, and the effect of this method on the success of the algorithm course of the students was investigated. The visual programming tool used in this study is "code.org". The internet can be connected to the system with any device. It's free. If you are a member, individual development can be followed by both the student and the teacher. It has a user-friendly interface. Graphical and auditory elements are utilized. Interaction with the user is provided at a high level (Kalelioğlu, 2015).

Code.org also has some features that are different from other visual programming tools. Because of consisting gaming method, this programming tool is more attractive and more choise than others. It can be said that the platform is like a puzzle as a game. According to Tutgun Ünal et al. (2013) in digital game platforms, puzzle style games are the most preferred games. Because puzzles are games in which cognitive skills are employed and pieces are appropriately brought together. In the teaching of algorithmic logic, students can also offer trial opportunities. In the current research, one of the puzzle-style electronic games-code.org has been used.

The Purpose of the Study

The purpose of the study is to investigate the impact of the game-based algorithm program code.org on the success of students entering the programming course. In addition, this study compares the achievements of students learning with code.org and students taught using traditional approach. The research question of this study is "Is there a significant difference between the achievement of students in the control group and the experimental group in the introduction to programming course?"

The answer was sought. In order to answer this question, the following sub-questions have been answered.

a) Is there a significant difference between pre-test results of experimental and control group?
b) Is there a meaningful difference between the pre-test and post-test results of the experimental group?
c) Is there a significant difference between the pre-test and post-test results of the control group?
d) Is there a significant difference between post-test results of experimental and control group?

METHOD

The quantitative method is used to determine the effect of the algorithm learning method on the success of the students at the entrance to the programming lesson. The randomized pretest-posttest control group design was used.

Setting

The teaching of the algorithm topic has been realized within the course labeled the basics of programming. This course is one of the main courses taken in the fall semester.

In the traditional design of programming language course, control group students were taught algorithms, variables, decision-control structures, cycles, functions and debugging subjects during the first four weeks.

In the experiment group, students should first be a member of code.org site and the interface of the site is explained. A road map was drawn to help students navigate their learning within the site. Students were informed about completing Lesson 3 and Lesson 4 activities. It was stated that they could complete these applications outside the course. If students complete this application, they will be supported with additional applications. Activities in Lesson 3 and Lesson 4 relate to variables, decision-control structures, loops, functions, and debugging issues.

Both groups completed the basic programming lessons and then wrote the program with the C # programming language, one of the most used languages in the World, for 6 weeks. Total instruction completed in 10 weeks.

Sample

The study was carried out in the Alanya Aladdin Keykubat University Akseki Vocational School in the Computer Programming department students who are studying in the 2015-2016 academic year. The students were taken on a voluntary basis for research. The randomized sampling method has been applied in the determination of experimental and control groups. A pre-test has been conducted to measure the students' programming knowledge. Experimental and control groups were determined in the direction of very close results.

Data Collection

The data were collected by tests. At the beginning of the course, different instructional strategies were applied in the control group and experimental group in order to understand the logic of the algorithm for the first 4 weeks.

Throughout of the course, in experimental group "code.org" was used. All students in the experiment group have completed all the activities on course3 and course4 on code.org. Individual controls were performed. Through the system, the progress of the students was followed. At the same time, students in the control group were taught algorithmic logic with traditional approaches. The researcher used the projector to provide examples of algorithms and showed flow charts on the board. Subsequently, both the control group and treatment group students took a C # programming language course together for a period of six weeks. After the instruction is completed, all students complete the C # achievement test.

Data Collection Tools

The data of the study were obtained by pre-test and post-test. While the pre-test is aimed to measuring algorithm knowledge of students, the post test is aimed to measure the ability to write programs using the c # programming language.

The pre-test questions developed by the researcher and revised according to the opinions of the four field experts consist of 3 classical questions.

These questions were created for the creation of equivalent groups by the researcher.The post test involves 3 classical questions that should be answered using the C # language. Questions were developed by the researcher and checked by experts.

Data Analysis
The quantitative data collected from the tests were analyzed using SPSS 20.0 for descriptive statistics and t-test analysis with 5% significance level.

FINDINGS

Findings Related with Research Question 1
Findings related to pre-test scores to measure programming success of experimental and control groups are below in the Table 1.

Table 1: Pre-test Scores to Measure Programming Achievement of Experimental and Control Groups

Groups	n	$\bar{x}$	Ss	Sd	t	p
Experimental	50	35,83	20,06	60	0.037	0.986
Control	50	35,98	20,01			

A difference of 0.15 between the arithmetic mean of the experimental and control groups. T-test for understanding whether this difference is meaningful t = 0.037 and P = 0.986> 0.05 (at the 95% confidence interval) ,there was no significant difference between the experimental group and the control group. herefore, experimental and control groups are equivalent.

Findings Related with Research Question 2
Findings related to pre-test and post test scores to measure programming success of experimental group are below in the Table 2.

Table 2: Comparison of Pre and Post Test Scores of the Experimental Group

Tests	n	$\bar{x}$	Ss	Sd	t	p
Pre-test	50	35,83	20.06	30	-6,961	0.000
Post-test	50	75,06	35.06			

The difference between the pretest-posttest arithmetic mean of the experimental group was 40.03. T-test for understanding whether this difference is meaningful t = -6,961 and P = 0.000 (at the 95% confidence interval) There was a significant difference between pre-test and post-test scores.

Findings Related with Research Question 3
Findings related to pre-test and post test scores to measure programming success of control group are below in the Table 3.

Table 3: Comparison of Pre and Post Test Scores of the Control Group

Tests	n	$\bar{x}$	Ss	Sd	t	p
Pre-test	50	35,98	20,06	30	-4,628	0.000
Post-test	50	50,89	10,81			

The difference between the pretest-posttest arithmetic mean of the control group was 14.91. T-test for understanding whether this difference is meaningful t = -4,628 and P = 0.000 (at the 95% confidence interval) There was a significant difference between pre-test and post-test scores.

Findings Related with Research Question 4

Findings related to post-test scores to measure programming success of experimental and control groups are below in the Table 1.

Table 4: Post-test Scores to Measure Programming Achievement of Experimental and Control Groups

Groups	n	$\bar{x}$	Ss	Sd	t	p
Experimental	50	75,06	35,06	60	4,666	0.002
Control	50	50,89	10,81			

The t-test results showed that the experimental group for gaming method implemented had a higher mean score than the control group for which traditional methods were implemented. There is a significant mean difference between the experimental group (x = 75.06, SD = 35.06) and the control group (x = 50.89, SD = 10.81) with the mean difference of 24.17; t (60) = 4,666, p <.05.
There was a significant difference between the arithmetic mean of the experimental group and the control group. As you can see, the difference was favored by the experimental group.

DISCUSSION AND RESULTS

In this study, the effects of using a game developed for instruction of algorithms in programming lesson on students' programming achievement was investigated. The achievement of students using game was compared with the achievement of students learning through the traditional approach. The quantitative method is used to determine the effect of the algorithm learning method on the success of the students at the entrance to the programming lesson. The randomized pretest-posttest control group design was used. Results showed that the experimental group, game was used, had a higher mean score than the control group, traditional methods were implemented. A significant mean difference between the groups was found. It has been seen that gaming method has been applied in the teaching of programming and it has been positively affecting the success of the student.

Studies of Bergin & Reilly (2005) and Shellington, Humphries, Morsi & Rizvi (2015) overlap with this research results. Shellington et al. (2015) highlights that students enjoy games when especially learning programming and students' skills develop related with identify correct and incorrect syntax. Likewise, Dogan & Kert (2016) stated that learning with games has a positive effect on critical thinking skills and algorithmic achievements according to the classical methods.

Gaming method which is active in the process of student learning, encourages independent learning individuals to encourage their own or group solving of the problems that lead to their ideas; can be

used alone or in combination with other methods in teaching programming as a teaching method among the new approaches.

WJEIS's Note: This article was presented at 6th World Congress on Educational and Instructional Studies- WCEIS 2017, 26-28 October 2017, Antalya-Turkey and was selected for publication for Volume 7 Number 4 of WJEIS 2017 by WCEIS Scientific Committee.

REFERENCES

Asad, K., Tibi, M., & Raiyn, J. (2016). Primary School Pupils' Attitudes toward Learning Programming through Visual Interactive Environments. *World Journal of Education*, *6*(5), 20.

Bennedsen, J. & Caspersen, M. E. 2005. Revealing the programming process. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, St. Louis, Missouri, USA, February 23-27, 2005, pp. 186-190.

Bergin, S., & Reilly, R. (2005, February). Programming: factors that influence success. In *ACM SIGCSE Bulletin* (Vol. 37, No. 1, pp. 411-415). ACM.

Blackwell, A. (2002). What is programming? In *14th workshop of the Psychology of Programming Interest Group* (pp. 204-218).

Chang, S. E. (2005). Computer anxiety and perception of task complexity in learning programming-related skills. *Computers in Human Behavior*, *21*(5), 713-728.

Doğan, U., & Kert, S. B. (2016). Bilgisayar Oyunu Geliştirme Sürecinin, Ortaokul Öğrencilerinin Eleştirel Düşünme Becerilerine ve Algoritma Başarılarına Etkisi.*Boğaziçi Üniversitesi Eğitim Dergisi.33(2).*21-42.

Eguchi, A. (2014, July). Robotics as a learning tool for educational transformation. In *Proceeding of 4th International Workshop Teaching Robotics, Teaching with Robotics and 5th International Conference Robotics in Education, Padova, Italy* (pp. 27-34).

Hwang, W. Y., Shadiev, R., Wang, C. Y., & Huang, Z. H. (2012). A pilot study of cooperative programming learning behavior and its relationship with students' learning performance. *Computers & Education*, *58*(4), 1267-1281.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior*, *52*, 200-210.

Kumar, D. (2014). Digital playgrounds for early computing education. *ACM Inroads*, *5*(1), 20-21.

Layton, L. (2013). Week-Long "Hour of Code" Campaign Lures Millions of US Students to Computer Coding.

Lewis, C. M. (2010). How programming environment shapes perception, learning and goals: logo vs. scratch. In *Proceedings of the 41st ACM technical symposium on Computer science education*(pp. 346-350). ACM.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). *Programming by choice: urban youth learning programming with scratch* (Vol. 40, No. 1, pp. 367-371). ACM.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 16.

Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies*, *7*(1), 55-66.

Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New ideas in psychology*, *2*(2), 137-168.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67.

Resnick, M., Martin, F., Sargent, R., & Silverman, B. (1996). Programmable bricks: Toys to think with. *IBM Systems journal*, *35*(3.4), 443-452.

Robins , A., Rountree, J. & Rountree, N. (2003) Learning and Teaching Programming: A Review and Discussion, Computer Science Education, 13(2), 137-172.

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

Salomon, G., & Perkins, D. N. (1987). Transfer of cognitive skills from programming: When and how?. *Journal of educational computing research*, *3*(2), 149-169.

Shellington, W. A., Humphries, T. O., Morsi, R., & Rizvi, M. A. (2015, October). Syntax circuitry: A mobile game for practicing programming language syntax. In *Frontiers in Education Conference (FIE), 2015 IEEE* (pp. 1-4). IEEE.

Tutgun Ünal, A., İnan, F., Kaya, M. T., Fırat, M., Güzelbaba, Z., & Bahadır, A. (2013). Öğretmen Adaylarının Bilgisayar Oyunu Oynama Alışkanlıkları, Amaçları ve Oyun Tercihlerinin İncelenmesi: Maltepe Üniversitesi Örneği. *AJIT-e: Online Academic Journal of Information Technology*, *4(12).*29-52.

Wong, G. K., Cheung, H. Y., Ching, E. C., & Huen, J. M. (2015). School perceptions of coding education in K-12: A large scale quantitative study to inform innovative practices. In *Teaching, assessment, and learning for engineering (TALE), 2015 IEEE international conference on* (pp. 5-10).