ISSN 2367-8127 (CD-ROM), ISSN 2367-8151 (on-line)

Journal of
Innovations and Sustainability

Volume 2
Number 1
2016

Application of Block Programming and Game-Based Learning to Enhance Interest in Computer Science

Todorka Glushkova[1]

Plovdiv University "Paisii Hilendarski"

Faculty of Mathematics and Informatics - Bulgaria

Abstract

More rapid development of the information society poses increasingly acute issue to increase the effectiveness of training in the field of computer science. The fact is that students prefer more applied fields of information and communication technologies in comparison with the informatics and programming. To solve this problem, leading universities and software companies develop multiple environments for block programming, which is inherently more interesting, motivating, fun and practical. It is known that game-based learning significantly increased interest and activity of the students and significantly increases the effectiveness of the training. The article will present a model for application of block programming in the development of educational games for standard and mobile devices and their application in both computer science students and secondary school students. Will be presented the experience of the team in the use of e-learning environments and implementation of these applications in developed courses. The application of the model currently shows increasing interest of all groups of students to programming. All this gives grounds for continuing the work and research in this direction.

Key words: *block programming, game-based learning.*

[1] Corresponding author: Todorka Glushkova, Plovdiv University "Paisii Hilendarski", Faculty of Mathematics and Informatics, 4003 Plovdiv, 236, Bulgaria Blvd.
E-mail: glushkova@uni-plovdiv.bg

INTRODUCTION

The digital technologies impose new realities in contemporary society that require more dynamic and adequate changes in education that is inherently conservative. Every innovation and experience to change the traditional educational system requires more time for design, synchronizing the legal basis, approbation and implementation. This generates a continuous delay from rapidly increasing demands on the education system, for that is increasingly difficult to meet public expectations. From another perspective, the fact is that the activation of processes of automation and robotics in all sectors of production and management determine the radical change in attitudes to the modern labor market and determine the necessity of improving the quality and effectiveness of training in the field of computer science and technology.

Recent statistics show that fewer students want to engage in programming, unlike in more applied ICT fields. To solve this problem, teams from leading universities and software companies create different block-based programming environments, that are inherently visual, interactive, multimedia and attractive[2]. The code is also graphic and it is a set of colorful blocks that are associated with each other according to the desired algorithm. No syntax errors, problems with the structures of operators - also. All this gives confidence to students and allows them to focus on algorithms and scenarios, by which will implement them. In most cases, the block-based programming environments are Internet-based, that allowing access from anywhere, anytime and the ability to share created products with friends and followers. These features make it possible to use block-based programming for students with different level of knowledge and experience. Some of the most popular environments are presented in the following table (Table1.)

Table 1. *Some popular block-based environments*

Name	Internet site	Short description
SCRATCH imagine • program • share	http://scratch.mit.edu	Created by MIT (MIT-Massachusetts Institute of Technology) LLK Research Group.It represents a 2D platform with graphical and functional resources. Scratch 2.0 is

[2] Block-based programming environments: https://www.kidscodecs.com/resources/programming/education/.

Name	Internet site	Short description
		web-based and provides an environment for creating, sharing and commenting on developed projects. The programming language is similar to C (Source, 2011).
	http://code.google.com/p/blockly	Blockly is environment for block-based programming, designed by Google. Blockly is relatively new, and borrows many ideas from Scratch. For those who are familiar with Scratch is easy to work in the new environment. Can be exported Blockly program in JavaScript, Dart (object-oriented language by Google), Python or XML and to embed in other software products.
	http://snap.berkeley.edu	Snap! (formerly known as BYOB - Build Your Own Blocks) BYOB is the next step of Scratch, and is developed in the University of California at Berkeley. It added more new blocks, which every student alone can create, and new properties associated with object-oriented programming. Snap 4.0 is used instead C#. It is available on the Internet without installation.
		Stencyl is strongly influenced by Scratch, but is mainly focused on the creation of games. Games can be created and published for IOS, Android, Windows and Mac.
	http://ai2.appinventor.mit.edu	MIT App Inventor is a block-based programming environment for Android Mobile devices. Developed by a team of MIT in collaboration with Google. It is a web-based environment. As a result of

Name	Internet site	Short description
		programming is generated apk-file that can be published and shared in Google Apps.
	http://alice.org	Alice is a 3D environment, developed by a team at Carnegie Mellon University. This is perhaps the most popular and used block-based programming environment in recent years. In Alice we can program with 3D objects from libraries and to create our own objects. Since version 3.0. the users can develop their applications in blocks in Java, can also export a project in standard Java code.

In recent years, many universities around the world organize education of students and teachers. App Inventor has great potential to transform Computer Science education and it is the most motivating tool for engaging students with computing (Wolber *et al.*, 2015).

It is especially effective for training IT students and teachers. By using the opportunities offered its last realization, this tool is powerful enough for students and professionals in the field of computer science (Benjamin *et al.*, 2015). At MIT and other leading universities conduct trainings for teachers to apply block-based programming environments to work with different groups of students. They believe that it is better to start with SCRATCH and to continue with Blockly, BYOB, Alice and App Inventor. The reason for the great success of App Inventor is that the animations and robots are interesting and entertaining, but with App Inventor the students can build anything, including apps that directly improve in their everyday life and those of their family and friends. To date, over 3.9 million users from 195 countries have created over 11.5 million projects with the MIT App Inventor[3].

Alfred Thompson - a leading specialist in Microsoft, and Aman Yadav from Michigan State University believe that for each students / teachers, according his background knowledge and personal aims may be offered a suitable environment for block-based

[3] MIT App Inventor - Explore MIT App Inventor, 2015. URL http://appinventor.mit.edu/explore.

programming[4]. Another reason for the success of this style of programming is the easy migration from one platform to another, and the rapid transformation of the code to traditional programming languages, for example from Alice- in Java; from Blockly to Dart; to JavaScript, C# and Python from another ones.

Despite good reviews from many renowned scientists from around the world, there are a lot of critical remarks to this style of programming. There have been studies in the USA among students in high school classes and students from many universities for their opinion in terms of the block-based and text programming (David & Wilensky, 2015). Aggregated data on the advantages and disadvantages of block-based programming are presented in Table 2.

Table 2

Advantages	Disadvantages
Easy programming style without syntax errors; quick result; work on natural language; different colored blocks; drag-and-drop mode	Difficulties in copying and pasting code between different objects in the application.
Suitable different programming environments for learners of all ages and level of training	It is difficult to find fragments of block-based code, because it is scattered among the common area. In many environments all events, methods and procedures are coded in a common area.
Possibilities for use of all basic algorithms in programming	Difficulty in creating of math expressions and in describing of math functions
Work online without installations	There are difficulties in creating your own objects in terms of object-oriented programming
Rich libraries of objects and resources with which to create application	Some problems associated with the use of Web databases
Opportunities to transfer the code to the standard language - java, Python, Dart, Javascript, etc.	

Obviously it makes sense to think about whether we really need only from the standard programming environments to teach students in programming skills and

[4] Alfred Thomson site: https://plus.google.com/116648179447008949472

whether to pay more attention to the environment for block-based programming, which significantly increases the interest and motivation of students. Experience in most colleges in different countries shows that lecturers organize such courses, but they are usually short and did not show the full potential of these programming environments. Then, they continues with traditional programming. For example, in the Faculty of Engineering of the University of Cagliari in introductory programming courses is used SCRATCH instead of C ++. Stefano Federici (2011) shows a model for this mode of training, make a comparative analysis and concluded that this type of training gives very good results. Similar results are reported for using an Alice instead of Java in the first year of training students in computer science.

From 2013, according to the US project The Mobile Computer Science Principles (Mobile CSP) funded by the National Science Foundation (NSF) in different universities are organized trainings of 10,000 teachers and professors of computer science, which will apply block-based programming for mobile devices in more than 10,000 high schools and colleges[5]. There are developed a complete set of course materials.The training lasts six weeks, and teachers are certified to work in these programming environments. Similar projects are being implemented in other countries such as Canada, Australia and so on.

Considering the realities in Bulgaria and the fact that the majority of students prefer a more applied areas of computer science and do not want to deal with programming, in 2014 and 2015 separate teams from Varna, Shumen, Plovdiv, etc. organized training courses with teachers and students to use block-based programming and to create real applications. We organized festivals, IT competitions and Workshops to promote this style of programming (Momcheva *et al.*, 2014).

In the region of Plovdiv in collaboration with Regional Inspectorate of Education in the spring of 2015 we organized training for IT teachers to work in environments SCRATCH 2.0 and App Inventor 2. The training lasted one day. The interest and willingness of teachers to implement this style of programming is high. Spontaneously they organized groups of students in many schools and started training process (Fig. 1).

[5] The Mobile CSP Project: http://mobile-csp.org

Fig 1. *Training students on block-based programming*

The results obtained show undoubtedly increase the interest in programming (Komsalova-Tabakova & Glushkova, 2015). According to teachers at the end of the school year the advantages of this type of work are: motivation and stimulation of cognitive activity of students; development of logical and algorithmic thinking; opportunities for the application of differentiated approach; capacity building of students to work independently; to apply and develop their skills for self-control and self-esteem.

To verify the hypothesis that this type of programming is appropriate for university students in computer science, in the spring trimester of the last school year at Faculty of Mathematics and Informatics of Plovdiv University "Paisii Hilendarski" was declared training course "Block-based programming"[6]. The training was attended by about 60 students from all disciplines of the faculty. Learners with interest and desire participated in training process and developed their projects in App Inventor, Scratch and Alice. The course was directed to the possibilities for processing of data structures, working with different types of databases, creation of procedures and own blocks, using and management of sensors for mobile devices, creation of 3D animations and interactive applications. In the final stage of training, students developed their own applications and submit them to their colleagues. They indicated their willingness to continue their education in the next academic year. Similar are the results of work with students from specialty Telematics in Physics Faculty of University of Plovdiv during the current school year.

Game-based learning has proven its effectiveness in the learning process (David & Wilensky, 2015). Increasingly, this didactic technology is placed in the focus

[6] Faculty of mathematics and informatics: http://fmi-plovdiv.org/index.jsp?ln=1&id=2208

of the new educational doctrine and "serious games" become one of the most effective teaching tools. It is believed that the game is an important part of cognitive and social development of the students; that they learn through play with other people and so increase their degree of cognitive and emotional development; that the symbolic using of different objects in the game is prerequisite of abstract thinking (Whitton & Moseley, 2012). From one point of view game-based learning can be seen as a means to increase interest, motivation and effectiveness of training and on the other hand as an opportunity for simulation of real situations (Glushkova, 2014).

The Google team has chosen to use game-based learning in preparing students for block-based programming. For this purpose is created a special environment - Blockly games (https://blockly-games.appspot.com) After registration each participant given the opportunity to play and compete with others[7]. The scenarios of the games are different - to search the way in the maze, to draw shapes and pictures, to create scenarios and stories and so on. For moving to the next level, the student needs to create the right algorithm and to build the program from the provided blocks. In many of the tasks is required to optimize the created algorithm and to achieve a result with a limited number of steps. After solving of each task, the students pass to the next level, earn points and badges. If he wants, the block- written program can be transformed to some standard programming language with a text interface. A similar idea is of game-based environment Code.org (https://code.org), supported by Infosys Foundation, Google and Microsoft[8]. To use more convenient by students of all ages, we provided an interface in Bulgarian language of this environment. The teachers form the groups, determine the tasks, and chose the games for the lesson (Fig. 2).

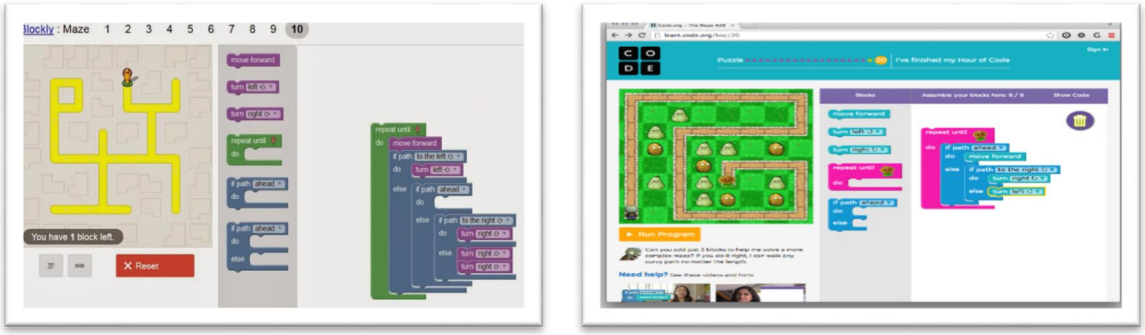

Fig. 2. *Game-based learning in Blockly games and Code.org*

[7] Blockly games:https://blockly-games.appspot.com

[8] An Hour of Code for every student: http://code.org

These environments can be used both for training and for verification and evaluation. After passing all levels in the games, students receive certificates and badges.
On the other hand, in the course of training in block-based programming clearly is highlighted the willingness of all groups of students to develop educational games. About 80% of the students in the schools, at the end of the training course, created projects, that are games; for university students this percentage is around 70%. Even more interesting fact is that about 70% of all developed games are educational games and are intended for direct use in the learning process. For example, the game "For excellent students and not only" is designed to prepare students for national external evaluation in biology, history and geography in seventh grade. The game is mobile application on App Inventor 2 (Fig. 3) Such is the game "Fun School", which is available over the Internet and is a game for preparation of students for national external evaluation in fourth grade. Both games were awarded from the team of Student Institute of Mathematics and Informatics of the Bulgarian Academy of Sciences in 2015.

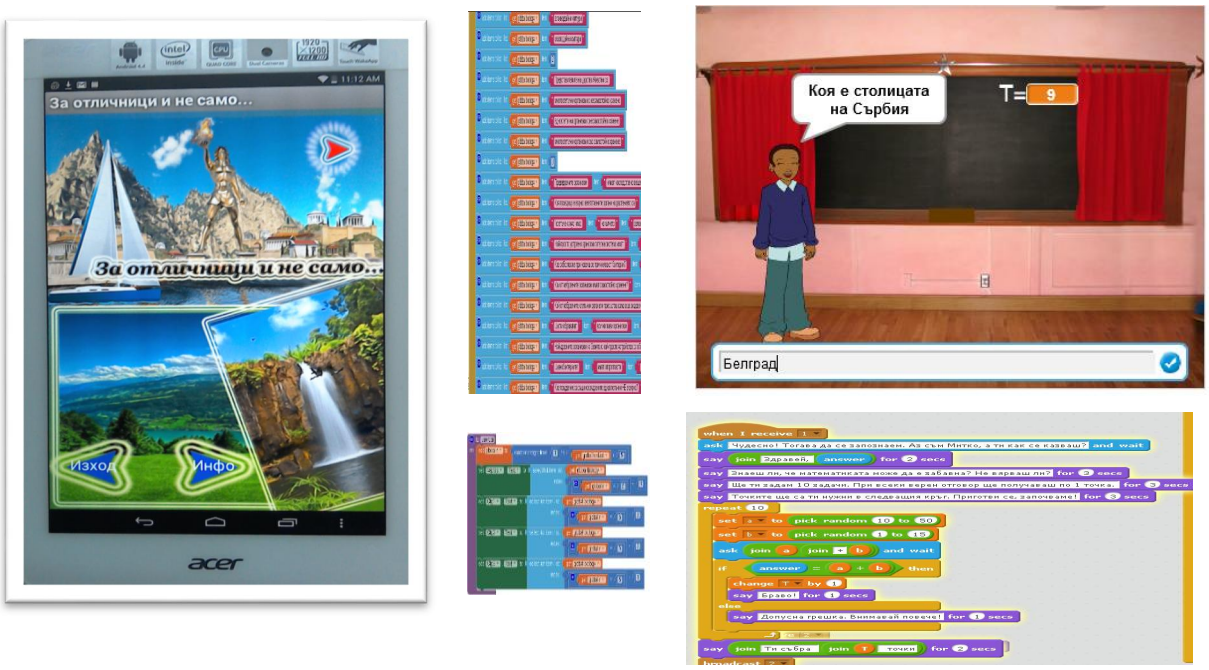

Fig. 3. *Learning games, developed from school students*

We got similar results and with the university students. Educational game "You should know this" is mobile application. It aims are to help students in their

preparation for the state exam. Other games are designed to facilitate the teaching of economic disciplines, computer graphics, etc. (Fig. 4).

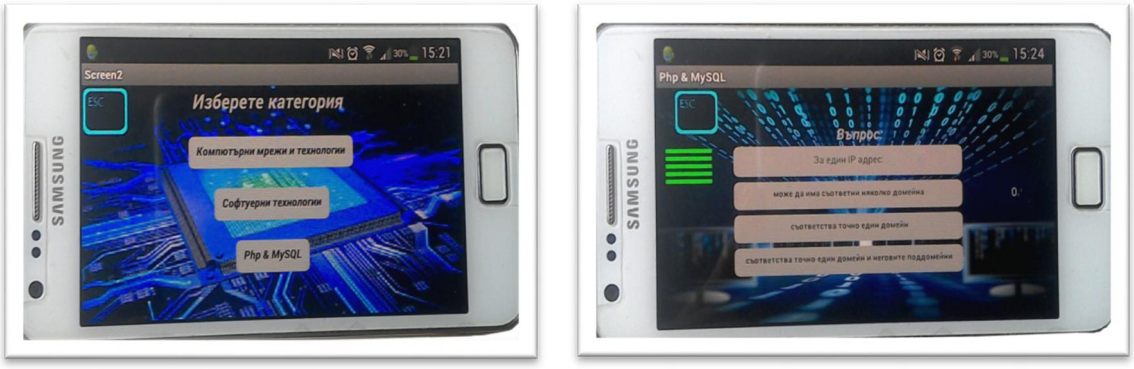

Fig. 4. *Educational mobile games, developed from university students*

In some groups, the students were encouraged to participate in role play "Software company". Participants were placed in almost real situation - they made analyze, plan, develop, test and install the developed software. In the "Software company" is formed teams of graphic designers, programmers, analysts, group for testing, installation and maintenance. The fact is that most thereby created software products were educational games. On the other hand created games were immediately presented to the students from different grades for using in the real training situation. Thus the activity and responsibility of the "programmers" are significant increased. They are satisfied with the fact that the result of their labor immediately finds its application in the real learning process and just completed one project already planning its improvement and expansion.

CONCLUSIONS AND FUTURE WORK

All data currently undoubtedly confirm our expectations for raising interest of students of all ages to programming. The lightweight style of work in the block-based environments stimulate students and allows them to concentrate mainly on the algorithm and the script of the project. This greatly develop their logical and algorithmic thinking and frees them from the oppression associated with syntax errors. Combining the use of block-based programming and game-based learning multiply the effect of the training and increases its efficiency, stimulate interest, motivation and activity of students. All data to date indicate that this style of programming is particularly successful for school students and beginner

programmers, as giving them a home base for future development as professionals. This gives grounds for the continuation of experiments and studies in this direction.

ACKNOWLEDGMENTS

This paper is supported by Project IT15-FMIIT-004 "Research in the domain of innovative ICT oriented towards bussiness and education" of the Scientific Fund of the University of Plovdiv "Paisii Hilendarski".

REFERENCES

[1] Alfred Thomson site: https://plus.google.com/116648179447008949472, Accessed on 10.11.2015.

[2] An Hour of Code for every student: http://code.org, Accessed on 30.11.2015.

[3] Benjamin, X., Shabir, I., & Abelson, H. (2015). Measuring the Usability and Capability of App Inventor to Create Mobile Applications. 2015 ACM SIGPLAN Conference on Systems, Programming, Languages and Applications: Software for Humanity (SPLASH), pp. 1-8, New York, USA, ISBN: 978-1-4503-3908-7.

[4] Block-based programming environments: https://www.kidscodecs.com/resources/programming/education/, Accessed on 25.11.2015.

[5] Blockly games: https://blockly-games.appspot.com, Accessed on 29.11.2015.

[6] David, W., & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming, IDC '15 Proceedings of the 14th International Conference on Interaction Design and Children, pp. 199-208, New York, USA, ISBN: 978-1-4503-3590-4.

[7] Faculty of mathematics and informatics: http://fmi-plovdiv.org/index.jsp?ln=1&id=2208, Accessed on 19.11.2015.

[8] Feberici, S. (2011). A minimal, extensible, drag-and-drop implementation of the C programming language, Proceeding SIGITE'11 conference on Information technology education, pages 191-196, New York, NY, USA, ISBN: 978-1-4503-1017-82011.

[9] Glushkova, T. (2014). Model for Application of Game-based learning in Secondary School, in proc. of International conference "Informatics in Science Knowledge", Varna, 26-29 June 2014, ISSN 1313-4345, http://www.vfu.bg/dokladi_ot_konferentsii/files/Informatika-konf-2014.pdf.

[10] Komsalova-Tabakova, V., & Glushkova, T. (2015). Block-based programming for everybody, "Education and Technology", Issue 6, ISSN 1314-1791, Burgas, BG, http://itlearning-bg.com/magazines/Spisanie2015.

[11] MIT App Inventor - Explore MIT App Inventor, 2015. URL http://appinventor.mit.edu/explore. Accessed on14.11.2015.

[12] Momcheva, D., Spassova, C., & Pavlova, E. (2014). The competition TECHNOVATION -training in creating mobile applications and enterprise, IV international scientific-conference "Modern trends of cooperation between school and family", Varna, 2014.

[13] Source, W. (2011). Scratch Programming, publ. General Books, ISBN 1234853191, 9781234853198.

[14] The Mobile CSP Project: http://mobile-csp.org, Accessed on 10.11.2015.

[15] Whitton, N., & Moseley, A. (2012). Using games to enhance learning and teaching, Routledge, ISBN-10: 0415897726, ISBN-13: 978-0415897723.

[16] Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2015). App Inventor 2- Create your own Android apps, O'Reilly Media, USA, ISBN-13: 978-1491906842, ISBN-10: 1491906847.

Internet sites

[17] http://scratch.mit.edu

[18] http://code.google.com/p/blockly

[19] http://snap.berkeley.edu

[20] http://ai2.appinventor.mit.edu

[21] http://alice.org