

IMPORTANCE OF PROGRAMMING LANGUAGES FOR CHILDREN

Tamara Petković¹²
Svetlana Lazarević Petrović¹³
Mioljub Lazarević¹⁴

<https://doi.org/10.31410/itema.2018.33>

Abstract: *There is an innovative and interactive way of learning basic computer programming with special reference to the programming language ‘Scratch’, which has an advantage because it’s easy and intuitive, user interface makes programming as simple as using a simple block-like interface which can later make a code. The paper will consider the existence of blocks for all important program structures which can be seen/which exist in all contemporary/new programming languages, just like blocks for advanced concepts such as events, messages/texts, delegates, applicable/Usable methods, knots and sorting stimulate creativity, imagination, interest in problem solving and basic principles of programming through creating visual, interactive and multimedia applications (games, animations, sites). They also stimulate young people to apply the knowledge that they have acquired, not only in IT and school assignments, but in all other life aspects.*

Keywords: *Programming, computer language ‘Scratch’, school, animation, children, technology, computers*

1. What is object-oriented programming?

Object-oriented programming is a programming paradigm that uses objects as the basis for designing computer programs and various software applications. Solving the problem of object-oriented programming paradigm is very similar to human thinking and problem solving. It consists of identifying objects and setting objects to be used in the appropriate sequence for solving a particular problem. It is about the design of objects whose behavior as a unit and in their mutual interaction solves a certain problem. Interaction between objects consists in the exchange of messages, where a specific message directed to a particular object, triggers encapsulated operations in that object, which solves a part of a generally wider and more complex problem.

In general terms, object-oriented problem solving consists of four steps:

1. Identification of the problem,
2. Identification of the objects that are necessary for its solution,
3. Identify messages that objects will send and receive from each other,
4. Creating a message sequence for objects that will solve the problems.

¹² Agricultural School of Applied Studies, Šabac, Vojvode Putnika 56, Šabac, Serbia

¹³ Agricultural School of Applied Studies, Šabac, Vojvode Putnika 56, Šabac, Serbia

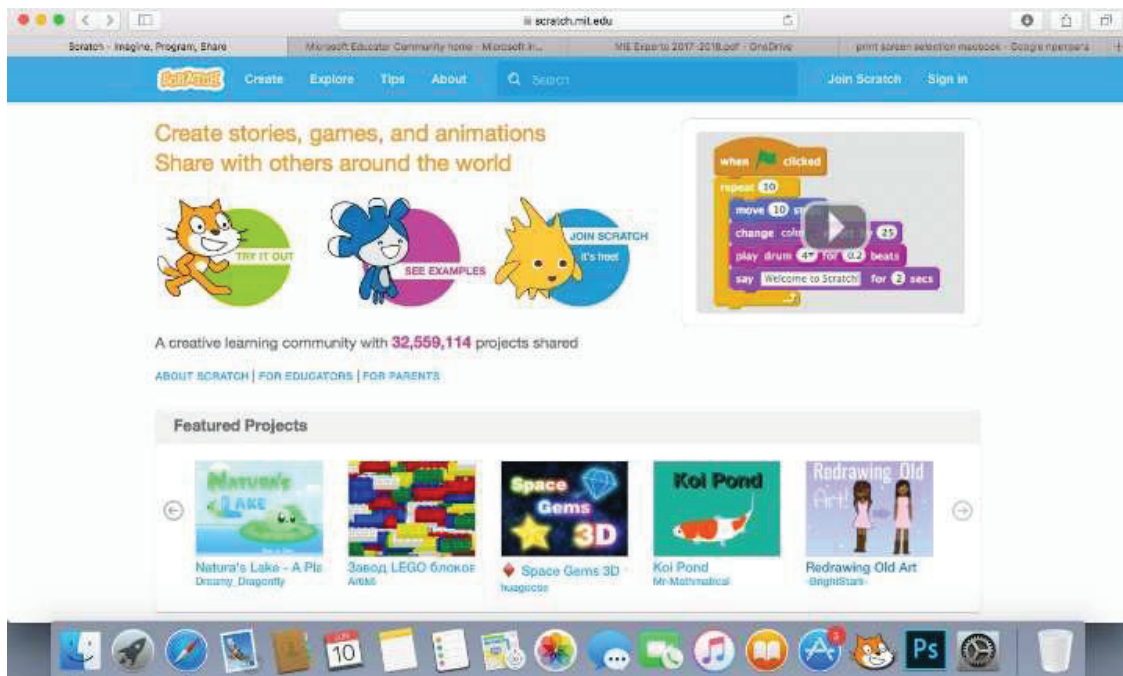
¹⁴ Faculty of Information Technology and Engineering, Serbia

2. Programming languages

Programming languages are used to facilitate communication with the computer when organizing and manipulating information, but also to accurately express algorithms. Some authors restrict the term "programming language" only to languages that can express all possible algorithms, and sometimes the term "computer language", which refers to more limited artificial languages, is used. A number of thousands of languages have been created, and new ones are created each year. Many programming languages require that the budget be specified in an imperative form [1] (for example, as the sequence of operations that must be executed), while other languages use other forms of program specifications, such as the declaration form [2] (e.g. the desired result is specified, and not how to achieve it). Some of the popular languages are:

- 1) Scratch is great for making games and programs that use many images; it is very easy to use,
- 2) Python is a language easy to learn, but it's also very powerful. It is used to coordinate programs for special effects e.g. in the film industry,
- 3) C++ is used to create programs that must work very quickly, including 3D computer games and platforms,
- 4) Java (Java) is a popular program for making games and applications for mobile phones mainly with an Android operating system and is often used in appliances such as air conditioners,
- 5) JavaScript is a language used to create interactive web sites, with a drop-down menu, text that changes, but also for making games on the Internet. Although they have a similar name, they are completely different from the Java programming language.¹⁵

Picture 1- Scratch online



¹⁵ Source: Private archive

3. The basic elements of the scratch project

The Scratch project is built from objects called Sprites. The appearance of a character can be changed by giving him a different costume from a library that can be accessed online via the Internet or if the software is downloaded and installed from the web address: www.scratch.mit.edu. The character can look like a person object, animal or anything else. Any picture can be used as a costume; users can draw their own drawing using the built-in drawing program "Paint Editor" or an image that is loaded from a hard disk or downloaded from a website. The characters join instructions indicating how they will move, how they will respond to other sprays, and what sound effects will follow it. A description of the behavior of a character is called a script (it is actually a program) and it is intended by placing graphic blocks one with the other in the stacks. When a double click is done on the script, Scratch executes actions specified by blocks from top to bottom. This process is actually running the program.

4. The Impact of the Use of the Scratch Programming Language for Young Adults

Several studies [3] have been made on the use of Scratch for learning programming in younger ages. In one of several studies where LOGO and Scratch programs were compared, it turned out that Logo is better for students' confidence, in greater interest in programming and better understanding of loops. Scratch showed an improvement in learning outcomes in conditional branching or repetition. The observations show that it is easier to work in Scratch. It is interesting that the students who were learning the Logo and Scratch were equal in the interpretation of loops. Logo logs for text commands, so students need to concentrate on details. This may be important to students in order to focus on important details that are usually neglected when using Scratch and thus compensate for the lack of a visual approach.

The most common research is done in summer schools and clubs where children are not in a particular school environment and do not have a load rating. In frequent researches during Harvard's Summer Computer Science School, they begin to learn programming in Scratch, and later switch to the Java programming language. Very few students had experience in programming. After completing the school, the students had very positive experiences with Scratch. The attendees who answered that Scratch had a negative impact on the experience with Java as one of the reasons stated that Java is much harder, less fun, and the results are how they expressed themselves are far more poor. There were also students who neutralized the experience with Scratch and they already had a previous programming experience. It was to these students that the programming language Scratch was more than easy to use.

The results that students achieve on the course of programming do not have a good correlation with their other successes in the school and the results of the competitions. A large number of studies were studied by the influence of factors such as spatial thinking and mathematical abilities, musical abilities, vocal skills, but these are factors that do not distinguish programming from other disciplines. More extensive research has indicated that the level of knowledge, mathematical knowledge and attribution of success can be considered good before the programmatic success rate. The results of the same research indicate that a serious approach to learning is in a positive relationship with the success of programming while the superficial attitude towards learning has a very negative attitude toward it. It is important for students to be able to achieve success and progress in programming. How do students experience programming? What, in their opinion, is needed to successfully master the programming tasks? They are very aware of the requirements of these tasks, so logical thinking and the ability to

solve the problem are considered the most powerful factors of success in programming. Very important are considered characteristics such as paying attention to details, considering certain alternatives, mathematical abilities and programming skills, and at the bottom of the scale, they have the ability to learn, knowledge of computing, modular engineering and planning. It is interesting that the skill of listening to work is not considered particularly interesting for programming discipline. Several different researches has confirmed the correlation between student's ability to solve problems and programming.

5. Conclusion

Teaching difficulties in children today generally hinder new technological and multimedia shifts, and programming is not particularly popular for long-term study at school. And in this particular case, when we have devoted attention to that in this work, we recognize more sociological reasons for which students lose interest in this discipline. Frequent perception of informatics as a socially isolated career, IT classes are often perceived as incompetent male environments of controlled behaviors, and students themselves rarely see the relevance of programming in real life. There is also an increase in digital violence, the great influence of social networks and overall intellectual development and access to computer science. The potential for attracting several different groups of people in the field of computer science is based on the creation of environments that deal with some of these sociological barriers to programming, supporting students as well as giving interesting reasons for programming and adequate education in these areas, with huge investments in computer science infrastructure and education.

6. REFERENCES

- [1] Michael Stevens (2011). „Programming paradigms and an overview of C - COMP3610 - Principles of Programming Languages: Object-Oriented programming” (PDF), Australian National University, p.5.
- [2] Lloyd, J.W., Practical Advantages of Declarative Programming
- [3] http://ims.mii.lt/ims/konferenciju_medziaga/SIGCSE%2710/docs/p346.pdf