Olympiads in Informatics, 2022, Vol. 16, 13–22

DOI: 10.15388/ioi.2022.02

How Competitions Can Motivate Children to Learn Programming

Aliya KATYETOVA
Doctoral School, Faculty of Informatics, Eötvös Loránd University
Budapest, Hungary
e-mail: akatyetova@inf.elte.hu, katyetova@mail.ru

Abstract. In the present time, more and more children easily play with LEGOs, labyrinths, and puzzles, which help to master basic logical skills and algorithm creation. Playing the games they compete with peers at home and in a primary school to be faster and assemble a construction correctly, to set up and be able to manage a drone or a toy robot. Children are enthusiastic about competing and getting ahead of each other and with pleasure want to know how a drone works, how a robot is programmed, how a joystick is configured for a game. This article aims to provide teachers and parents with the evidence how competitions can motivate children to learn programming.

Keywords: competitions, motivation, children, programming.

1. Introduction

Motivation of students to learn is important for both teachers and parents. Competitions offer a convenient way to bring informatics concepts to students in a more different fashion than traditional teaching in schools. It could be said that the tasks are the heart of the competition. Therefore, designing tasks that support the goals of the competition is an important and demanding undertaking.

Nowadays, there are many different informatics competitions from small to worldwide events (Scratch Olympiad, IOI, Bebras, etc.). Moreover, the types of tasks vary from easy ones solved with a pen and some paper to complex problems dealing with large datasets and sophisticated algorithms. Many different types of events offer a wide range of possibilities for pupils to get involved in informatics.

Competitions can encourage students who want to learn informatics in depth. They can also be a source of motivation for students interested in programming (Dagienė, 2010).

2. The Importance of the Competitions

According to Manev *et al.* (2009), it is necessary to teach students to start to compete as early as possible. In many countries, there is a large gap between the knowledge and skills acquired through the regular curriculum and those required for informatics and algorithmic contests and competitions. Thus, teaching programming as well as preparing for informatics contests is basically an activity outside a school. There is a need for more independent work on the part of students and a more specific attitude from teachers and parents (Nemeth and Zsako, 2018).

Children at school and home became more interested in learning how to create new information communication technologies. They are easy to play LEGO, independently assemble mazes and puzzles that help them master the basic logical skills of creating algorithms. Competing with each other, children curiously examine robots, drones, joysticks, and other automated toys. They with burning eyes and interest ask their peers, parents, and teachers how they are arranged and how they work. Therefore, when schools announce the call of competitions and Olympiads in informatics for children, they willingly participate in them.

In the article "Kids Programming Marathon: A Step toward Better Engagement with Computer Science Education" (Taki and Alnahhas, 2019) authors state that computer science education is an extensive subject. It combines all the "STEM" subjects: science, technology, engineering, and math, and also includes design. It is important for students to master these skills because computer science is everywhere in our life. By expanding access to computer science for all young people as early as possible, it is necessary to help them prepare for current and future work. This education provides them the opportunity to become the next world innovators and programmers (Taki and Alnahhas, 2019).

Competitions are one of the essential ways to encourage pupils to discover new fields. They introduce children to various fields of science, including informatics, where the emphasis is on testing problem-solving and logical analysis skills through an exciting experience in which the participant learns programming, thereby satisfying their interests. Therefore, many countries are working on organizing informatics and also programming competitions for children, in order to develop not only the skills of creative thinking for pupils but algorithmic thinking as well. Furthermore, it is important for improving motivation to learn computer programming.

In addition, while writing this article the author wondered why many countries have informatics competitions for children, while some do not have such contests, particularly in programming for primary schoolchildren. At the same time some nations (Syria, etc.) are working on organizing a programming marathon for children and adolescents to increase the general scientific level and enhance the skills of analysis and creative thinking among pupils (Taki and Alnahhas, 2019). In Kazakhstan, competitions for children are held only from grades 7–8. It is a challenge to think and start organizing informatics contests by engaging primary schoolchildren. Such contests will make it possible to interest them, even more, to study computer science outside school, to learn programming with the help of educational games.

3. The Power of Motivation

The primary school years are important in children's lives. In this formative period, boys and girls profoundly affect their mental and emotional growth. Today's schools are challenged to provide meaningful experiences that will help these children realize their full potential. And in this case, motivation plays a huge role.

What does motivation give to children? First of all, it is a good tool for education and development. Then there is the importance of computer science education. These are also new academic achievements and increased interest in the subject.

When children have a motivation they can come up to a number of things, including:

- Self-confidence.
- Lack of fear.
- Focusing.
- Comprehension.
- Good organization skills.

While some students are natural self-motivators, many children struggle to find the motivation needed to do their best. Therefore, competitions can help in this situation. At the same time teachers also help them in informatics classes.

3.1. *How can Competitions Motivate Children to Learn Programming?*

The goal of all teachers is to help pupils become self-motivated students. And competitions provided in the computer classes and/or online via Zoom, Microsoft Teams or Skype help them it this deal. To do this, it is important to involve parents in playing the games together with their sons and daughters which builds motivation. It can be done at a time convenient for parents, in the evenings or on weekends. All of these things help children to improve confidence and motivation to learn.

For example, competitions in Informatics organized by schools include different Informatics and ICT themes starting from easy tasks to difficult ones. It depends on whether the junior or senior pupil is participating and what the purpose of this event will be.

The following tasks features should be taken into account (Hakulinen, 2011):

- The problem should be clearly formulated.
- The tasks should be easy to understand.
- The algorithms solving the problem may be a modified version of the classical algorithm.
- There should be several different acceptable solutions of varying complexity and efficiency.
- The result should be clear and concise (depending on the complexity of the task).
- Tasks might be interactive and using questions.
- Short non-programming tasks can be used to attract new students.
- Other criteria.

These tasks provide pupils with logical and algorithmic thinking development. Algorithmic thinking is influenced by many human cognitive factors. This means not only abstract and logical thinking but also creative abilities and problem-solving competences, as well as the ability to think in structures. This complexity creates difficulties in learning and developing the algorithmic thinking of pupils. We need to reduce the complexity to the level, where the concepts of algorithmic thinking can be learned in a natural and playful way. The following is recommended: use of basic actions, natural description language for writing the algorithms, and an interactive environment with possibilities for experimentations also flexible for a run variety of the algorithms. The problems to be solved must be adequate to the pre-knowledge of the children-beginners.

And competition here is a good motivation tool for learning, particularly in programming.

In this part of the article, we can cite the following example from an article by Kubica and Radoszewski (2010), who proposed using tasks that require algorithmic thinking, but not programming, in order to attract novice students who know nothing about programming or algorithms. They claim that offering tasks or puzzles that require different levels of algorithmic thinking is a good way to popularize programming learning among younger schoolchildren. They also proposed a couple of tasks that require algorithmic thinking but are formulated in a purely mathematical way. The problems are designed in such a way that the desired solution minimizes the total time for its manual execution.

4. Learning Computer Programming

Being primary school students, children develop an interest in academic subjects, identify inclinations to various fields of knowledge, types of work, develop moral and cognitive aspirations. However, this process does not occur automatically, it is associated with the activation of cognitive activity of students in the learning process. One of the effective means of developing cognitive interest in computer science lessons in primary school is a game.

For a child of primary school age, playing is of the utmost importance: for them it is study, work, a serious form of education.

The inclusion of games and game moments in the lesson makes the learning process interesting and entertaining, creates a cheerful working mood for children, facilitates overcoming difficulties in mastering the educational material. Another positive side of a game is that it promotes the use of knowledge in a new situation, thus, the material assimilated by younger schoolchildren goes through a kind of practice, brings variety and interest to the educational process.

Playing is a natural and humane form of learning for a child. By teaching through games, we teach children not how it is convenient for us, adults, to give educational material, but how it is convenient and natural for children to receive it.

There are various types of games that contribute to the development of cognitive interest of younger schoolchildren. These are exercise games, competition games, story-role-playing games, educational travel games, etc.

The majority of students show an interest only when the lesson's topic is interesting to them or the teacher uses unusual teaching techniques, particularly, a game. A teacher should stimulate and develop the cognitive interest of younger students in each computer science lesson. It is necessary to strive to ensure that most of the class has a high level of cognitive interest, that is, that primary school students are active in every lesson. To do this, a teacher needs to include game elements or game situations in the structure of each computer science lesson and in informatics competitions.

4.1. *Teacher Engagement*

One-to-one programming is probably the best form of teaching programming because the teacher can focus on one student and promptly give feedback and correct the student's misunderstandings. However, in real conditions, one teacher often teaches ten, twenty, or even more students at the same time. In this case, quizzes and tests with appropriate feedback can help to identify misunderstandings of the students about programming concepts and to correct them (Brown and Wilson, 2018).

Using live coding can also help students. If the teacher creates a program in front of their students instead of using and showing presentation slides, the teacher can react more sensitively to students' "what if?" questions. Also, learners can see that making mistakes during programming is normal and they can learn to find and correct these errors. In addition, a teacher may ask their students several times during a live coding lesson to predict the results before the application is executed (Brown and Wilson, 2018). Despite the fact that live coding might be slower than using prepared slides, Stoffova, V., and Vegh, L. believe that it is worth taking the time to try at least a few times during a programming lesson (Stoffova and Vegh, 2019).

Pair programming, when two students share one computer, is also a good practice in teaching computer programming. One of the students types the code, while the other comments, prompts a classmate and makes suggestions. It is important to change the role several times per lesson. During pair programming, students can help each other, they can explain each other's misconceptions (Brown and Wilson, 2018).

It is also important that teachers give students motivational tasks that interest them. Since almost all students like computer games, it can be a good choice to assign learners to create some simple games. A computer game as a project adds an element of fun to programming, captivates children, and they stay interested from the beginning to the end (Doherty and Kumar, 2009). However, because children in primary school are not proficient in programming, it would be difficult for them to write a complex computer game from Scratch. Framework-assisted computer programming, where children use a prepared application framework for creating games, might be an effective solution to this problem. Using interactive online e-learning platforms that merge the possibility of learning programming skills while building digital games (Ivanova, 2016) can be a good solution, as well.

In this case, the Scratch programming environment can help.

4.2. *Scratch as a Motivation Tool*

Scratch as a visual environment for programming and creating games is being used successfully to teach programming skills to novices. It is used in primary and secondary schools in Kazakhstan and other countries as well. Scratch is freely available and easy to install and use for teachers and students. Scratch teaches computational concepts to students in a fun and engaging way. The student engagement with Scratch is superior to the level of engagement while studying ICT literacy skills and motivation levels are also high.

For instance, in Scratch, you program largely visually: Programs are put together from colored building blocks, which allow children at any performance level a very easy entry into programming. At the same time, Scratch can also introduce advanced programming concepts such as object orientation, concurrent processes, or event handling in a very natural way. Scratch is, therefore above all, a tool for conveying important and cross-programming concepts. Verifiable learning objectives are important to us. The acquired knowledge can be applied or expanded in (partly further) tasks directly following the material. The successful completion of the assessment tasks (in the form of a game) at the end of each learning unit guarantees that the essential concepts have been understood. In this case, the children are happy, perform tasks with interest, compete with each other to see who is more beautiful and performed the algorithm better or made an easy game according to the template by themselves.

Teachers can include tasks that have some initial code, which have to be improved. Here Scratch can be used to motivate younger students to participate in a competition and include tasks with some initial game elements that needed to be improved.

Often the games chosen for the competition may not have a known ideal solution. This is how they encourage competitors to think and develop their own original ideas, rather than implement well-known algorithms.

4.3. *Working in Tynker Platform*

Tynker is a creative computing platform that helps kids develop computational thinking and programming skills in a fun, intuitive, and imaginative way. As they are guided through interactive game-based courses, kids quickly learn fundamental programming concepts. With Tynker every child can apply their coding skills as they build games, tell stories, create apps, control drones and robots, and more. The platform even offers a parent dashboard where mothers and fathers can follow their child's success and share their creations.

This opportunity is especially interesting for those children who already like to play with LEGO. They will be more genuinely excited about the opportunity to integrate Tynker with these interests, expanding their potential for games as they learn.

The significant advantage of this platform is the possibility of working in it for primary school students aged 5–10 (Fig.1).

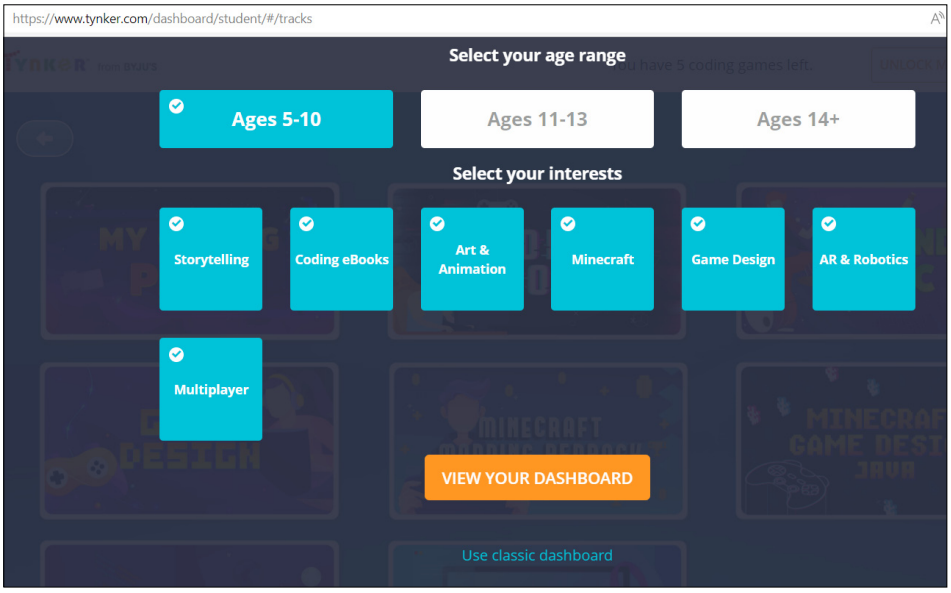

Fig.1. Age range of participants.

Children begin programming using Tynker's block-based visual language, which helps them recognize patterns and master programming concepts such as sequence, loops, conditional logic, and algorithmic thinking. They can show their creativity by animating their games and telling stories using code.

4.4. *Involving into Playing Algorithms*

There are several ways to engage children to playing algorithms not only using Scratch and Tynker platform but also other computer applications.

Understanding algorithms is one of the difficulties that students met when starting to learn programming. For beginners in programming, it is important to understand the principles of algorithms along with the ability to find or create their own algorithms for new problems and tasks. An algorithm prescribes exactly what to do in the given possible situations. This is one of the main educational goals that beginners should know. So, the algorithm manages all possible situations, it is a sequence of steps that leads to the result. It is a possibility to experience in a game that exactly follows such steps of a self-developed algorithm (Czakoova, 2020).

Children understand the principles of algorithms by solving different tasks in competitions organized by schools. Animations of algorithms are used. According to Vegh and Stoffova (2016) in these approaches, the students play algorithms. In this way they get a better understanding of given algorithms.

In the article "Developing algorithmic thinking by educational computer games" Czakoova (2020) suggests an idea of involving students into playing algorithms. The task of the teachers is to motivate the students to improve their algorithms whilst finding more efficient solutions. An example of a computer game in which there are three levels of difficulty to find an effective solution is given. The students take on the roles of a figure and a navigator. They can determine the progress of the game using algorithms. In this way, students will learn more about basic algorithmic thinking. While the students find good solutions, they can learn a lot about sequential algorithms. It is a form of explorative learning, where the students can test algorithms by playing. The goal is to bring fun, pleasure, and motivation to the programming learning process by using games. All this can be used at computer science Olympiads as assignments.

A much greater motivation arises when students get the opportunity to invent their own algorithms to solve a specific problem. A necessary condition is the correct choice of tasks that need to be solved. The best option is to gamify the problem. Gamification opens the way to the introduction of game elements in a non-game situation. Such an educational approach at competitions motivates students to learn using game design and game elements in learning environments. The competitive spirit during the competition and games should inspire children to continue learning new and interesting things. Games in any form increase motivation through engagement. All this is observed as more and more important in education. The goal is to maximize fun as well as engagement by attracting student interest.

Problem methodology is effectively used in a playful way in competitions for the development of algorithmic thinking of primary school students. Students should be directed to become creative problem solvers, experimenters, and creators of alternative solutions. By playing games, the student receives immediate feedback so that he/she can correct his/her actions to get the right solution (Czakoova, 2020).

For novices in programming, educational computer games with the help of programmable toys are suitable tools for teaching the basics of programming. They will be highly motivated by the educational computer game to learn programming (Stoffova and Czakoova, 2019).

The situation is quite different when students have to program a game. This motivation towards games could be used to promote informatics competitions for children and also to get students interested in algorithms. As they learn, children create mini-games, solve puzzles, create programming projects, earn exciting badges (prizes) and discover new characters. That's why kids love to learn by playing – even though they are learning important programming concepts, they feel like they are just playing a game.

5. Conclusion

In this paper from all the mentioned above, it is obvious how important the role of the competitions is for the student's motivation to learn computer programming. They do not only involve children in an interesting and exciting environment but also develop a competitive and team spirit in children to complete tasks. It also helps to instil interest

in studying further topics in computer science and programming, because children are already familiar and know that it will not be boring, but fun and useful.

In competitions, the use of games and game situations using a computer and in the process of teaching computer science is also appropriate and relevant at the stages of programming training.

For many countries, it will be meaningful and useful to think about organizing competitions for younger schoolchildren, especially to engage kids from grades 2–4. Children's computer science competitions will be promising competitions that should be organized where they are not held and supported in order to promote computer science education, as well as for the future development of children's digital literacy.

References

Brown, N.C., Wilson, G. (2018). Ten quick tips for teaching programming. *PLoS Computational Biology*, 14(4), e1006023.

Czakoova, K. (2020). Developing algorithmic thinking by educational computer games. *eLearning & Software for Education*, 1.

Dagienė, V. (2010). Sustaining informatics education by contests. In: *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*. Springer, Berlin, Heidelberg, pp. 1–12.

Doherty, L., Kumar, V. (2009). Teaching programming through games. In: *2009 International Workshop on Technology for Education*. IEEE, pp. 111–113.

Hakulinen, L. (2011). Survey on informatics competitions: Developing tasks. *Olympiads in Informatics*, 5, 12–25.

Ivanova, S. (2016). Learning computer programming through games development. In: *The 12th International Scientific Conference "eLearning and Software for Education"*. "Carol I" National Defence University Publishing House, Bucharest, Romania, pp. 492–497.

Kubica, M., Radoszewski, J. (2010). Algorithms without programming. *Olympiads in Informatics*, 4, 52–66.

Manev, K., Sredkov, M., Bogdanov, T. (2009). Grading systems for competitions in programming. In: *Proceedings of the XXXVIII. Spring Conference of the Union of Bulgarian Mathematicians*.

Németh, Á.E., Zsako, L. (2018). Grading systems for algorithmic contests. *Olympiads in Informatics*, 12, 159–166.

Stoffová, V., Végh, L. (2019). Learning object-oriented programming by creating games. In: *The 15th International Scientific Conference eLearning and Software for Education* (Vol. 1, pp. 20–29). "Carol I" National Defence University.

Stoffová, V., Czakóová, K. (2019). A Playful form of Teaching and Learning using Micro-World Applications. In: *The International Scientific Conference eLearning and Software for Education* (Vol. 1, pp. 110–115). "Carol I" National Defence University.

Taki, M., Alnahhas, A. (2019). Kids programming marathon: A step toward better engagement with computer science education. *Olympiads in Informatics*, 13, 225–235).

Végh, L., Stoffová, V. (2016). An interactive animation for learning sorting algorithms: How students reduced the number of comparisons in a sorting algorithm by playing a didactic game. *Teaching Mathematics and Computer Science*, 14(1), 45–62.

Websites

A.D. Katyetova, doctoral student of the Faculty of Informatics of Eötvös Loránd University, Budapest, Hungary, Master of Informatics. She is the author of scientific and methodological papers and articles in Kazakhstan on the informatization of education, distance learning, and children's development. She is the author of the educational and methodical manual for the course "Fundamentals of algorithmization and programming".

She was doing a research internship on the program "Improving University Leadership, Management and Teaching and Learning" in the Reading University, UK, under the "Bolashak" International Program of Kazakhstan (2013). She has been awarded a Stipendium Hungaricum Scholarship (2021).

She worked in the Strategic department in a Kazakhstan educational company, which is a National operator of WorldSkills Kazakhstan competition.

Her research interest includes didactics and methodology of teaching informatics (digital literacy) in primary school, teaching programming languages and databases.

Her current research interest is the digital literacy development of primary schoolchildren.