

# A Discussion of Developing a Programming Education Portal

BALLA Tamás, KIRÁLY Sándor

**Abstract.** The dynamically developing IT industry is struggling with the lack of qualified software developers. The number of students in this area could be increased if more youngsters were familiar with programming in the public education system. The developed *kodolosuli.hu* portal offers interactive programming courses and coding challenges in four different programming languages: C++, C#, Java, and Python for free. Our goal with the design and development of *kodolosuli.hu* was to give the younger generation the opportunity to learn the basics of programming in a playful way. We started operating the portal in 2015. In this paper, we primarily present the experience gained over the last 5 years and the results of the analysis of the data collected during the operation of the portal. Based on the collected experiences, both the completed improvements and the possible future development directions are presented. This paper also outlines our future goals, which are primarily aimed at enhancing learning efficiency.

**Keywords:** educational games, adaptive learning, computer programming

## 1. Introduction

In IT education, the acquisition of algorithmization skills and the development of problem-solving thinking are becoming increasingly important. If we look at the current (2020) National Core Curriculum, it can be seen that this goal, this aspiration, is clearly emphasised. The National Core Curriculum defines the educational goal of the field of informatics (“Digital culture”) as “algorithmization and coding play an important role in the course, as it promotes the development of competencies such as problem solving in a digital environment, creativity, collaboration and logical thinking.” (Magyar Közlöny, 31 January 2020 No. 17: National Core Curriculum p. 140.) In the last few years, achieving this goal was extremely difficult due to the lack of sufficient lessons (45 minutes once a week). This problem has proved all the more acute, since there are many programmers missing from the job market, and it would be beneficial if secondary schools were to orient students towards the software development profession. To do this, it is essential to familiarize students with programming before applying to higher education institutions.

Recognizing this problem, we set ourselves the goal of creating courses where students can acquire programming skills online in the most popular (C++, C, Java, Python) programming languages. We examined the LMS systems that are available free of charge in order to determine which one to use during the preparation of the courses. The most important goal during the course development was that we would be able to check the source codes submitted by students in an automated way. We found that there is no appropriate, small and compact LMS framework in place to check practice-oriented tasks in all three languages. Therefore, we decided to build our own framework, which provides the above possibility, and it is definitely advantageous to implement our own software system due to the easier implementation of future improvements.

As a result, we created the *kodolosuli.hu* portal, which provides online courses in C#, C++, Java and Python programming languages for free (after registration). We have been operating the portal for about five years, and the aim of our article is to present the experiences, the completed improvements and to formulate future development directions. During the development of the portal, one of our goals was to create an e-learning environment that maintains a high level of student activity, student engagement, which enables guided discovery [3],[6],[10],[16],[17],[23][26]. Our goal during the development of the portal as well as the courses was to place a strong emphasis

on the involvement of playful applications in the courses, which increase motivation and help faster and easier processing and understanding, thus making the learning process more efficient [1],[4],[19],[20],[21],[28].

In our portal, the courses are gamified. They contain game elements in a non-game context: points, incentive and immediate feedback, and after completing the different sections, users get information and screenshots about the downloadable, reward program. This software becomes available for students after completing the course. We have not implemented leaderboards, since in our experience, it motivates the best N students, and others are more frustrated by leaderboards. We have also developed 19 games to encourage learning of programming.

There are numerous games types that can enhance the effectiveness of a programming course. Games that use some kind of computing machinery (e.g., personal computer or a smartphone) are called **digital games**. A **serious game** is a digital game created with the intention of entertaining and achieving at least one additional goal, for example learning or health [5]. According to the aforementioned definitions, our developed game programs are serious games with a developed didactic purpose. Unlike serious games, **educational** or **didactic games** are definitely educating tools serving a didactic purpose. They can be defined as interactive, competitive lessons with defined learning outcomes that enable students to have fun during knowledge acquisition. Their goal is not merely fun, but they also contain an educational component [2]. Playing with some of our programs in the portal, students can simulate the working of computer programming code structures following the given instructions that help them to understand the working of codes. Following the definition [25], these games are not only education games but also **Instructional Simulation Games** as shown in Figure 1. Learners using textboxes, buttons, and slide bars can run or stop the experiment, which is a simplified version of reality, and change the parameters of the phenomena within a framework of rules.

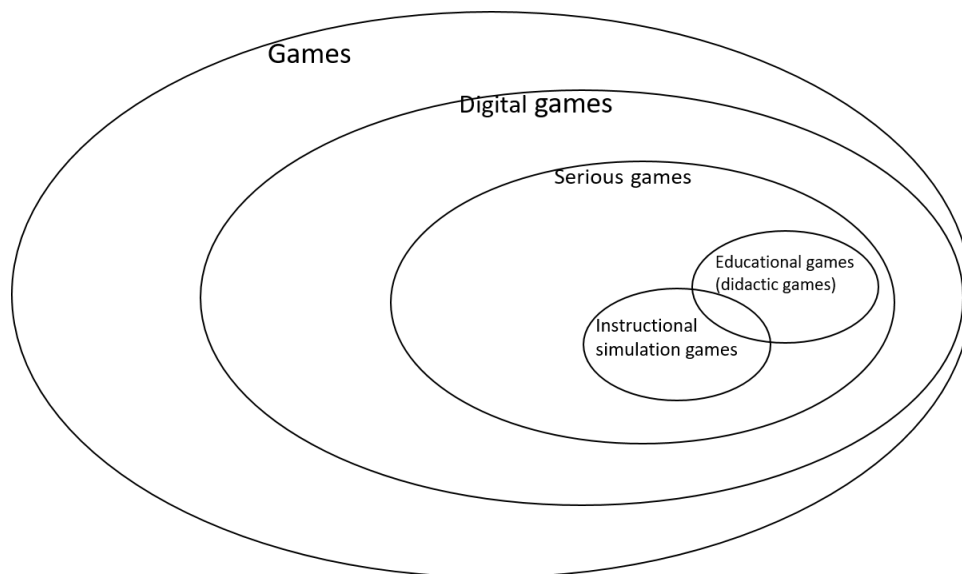


Figure 1: Game types

Our site offers real coding tasks related to the current lessons, contains animations and social media access (Facebook and Twitter).

## 2. The structure of the portal

The kodolosuli.hu portal offers free, online, C++, C#, Java, and Python programming languages courses in Hungarian. During the implementation of the framework, the LMS system was designed so that the courses could be structured as a series of chapters that consist of a series of lessons. Depending on the type, the lessons may include tasks as well as game programs, the purpose of which is to practice and deepen the acquired knowledge.

The website was written in PHP5 according to the MVC pattern. The curriculum, the users' data and their acts are stored in a MySQL database in 17 tables as opposed to the game applications that can be found in folders. Currently, there are two types of users who can access to the System: students and the administrators. Students can access only the curriculum unlike the administrator who has full access to both the materials and the users' data.

After login, students need to choose a language they wish to learn. The material to be learned can be found on the left side of the screen. The exercise belonging to the current topic is on the right side (see Figure 2).

Figure 2: The main user interface of the portal. The curriculum is on the left side, the exercise and the code panel are on the right side

The aim of the chapters is to integrate the logically related knowledge elements and lessons into one unit, as shown in Figure 3. The courses for the four programming languages are structured in such a way that by completing the course, students are thoroughly prepared from basic types of process-oriented programming. The most important chapters in the curriculum are *basic types, variables, I/O operations, operators, branches, loops, complex data structures (array, list, structure, class), methods,*

*text file management, exceptions.* The programming theorems, which play a key role in programming education, have been inserted into the individual lessons starting from the beginning of the chapter of Loops [24],[29].

Programozási nyelvek ⇌ Java

### Fejezetek listája


Sorszám	Cím	Fejezet leírása	Alfejezetek száma	Alfejezetek listája	Fejezet feldolgozása
1	Változók	Változók használata.	11	<a href="#">Alfejezetek</a>	<a href="#">Feldolgozás</a>
2	I/O műveletek	Hogyan lehet adatok beolvasni konzolról? És hogyan lehet a képernyőre írni adatokat?	6	<a href="#">Alfejezetek</a>	<a href="#">Feldolgozás</a>
3	Operátorok	A programozási nyelv által biztosított operátorok és használati eseteik.	7	<a href="#">Alfejezetek</a>	<a href="#">Feldolgozás</a>
4	Válaszút elé értünk!	Eddig azt tapasztaltuk, hogy a program utasításai egymás után kerülnek végrehajtásra, azaz a végrehajtás, futtatás szekvenciális. Hogyan lehet azt megoldani, hogy például egy változó értékétől tegyük függővé utasítás vagy utasítások végrehajtását? Erről szól a fejezet.	3	<a href="#">Alfejezetek</a>	<a href="#">Feldolgozás</a>
5	Ciklusok	Az előző lecke után világos, hogyan lehet bizonyos utasításokat végrehajtani vagy nem végrehajtani egy feltételtől függően. A kérdés most az, hogyan lehetne bizonyos utasításokat többször is végrehajtani egy adott feltételtől függően.	9	<a href="#">Alfejezetek</a>	<a href="#">Feldolgozás</a>

Figure 3: List of chapters (Java)

A linear sequence of lessons forms a chapter. We distinguish three types of lessons: **curriculum with assignments, independent assignment, independent curriculum**. Independent teaching materials may contain blocks of text, audiovisual elements to support the learning process, and games may also be included here.

### A macska defibrillátora

A következő programmal Kódoló Samunak (KS) kellene segítened.  
Van egy szívbeteg macskája, akinek időnként megáll a szíve. Ilyenkor KS-nek gyorsan el kell szaladnia egy macskák számára kifejlesztett defibrillátorhoz, majd azt a macskájához kell érinteni, amelynek így ismét elindul a szíve.



A feladatod olyan program írása, amely megmondja annak a defibrillátornak a sorszámát, amelyet KS-nek választania kell, hogy a leggyorsabban elérje a

Figure 4: An independent task with a picture that depicts the problem

It is possible to create lessons with short tasks, in which case, in addition to the curriculum, short sets of tasks appear on the right side to help students master the specific part of the lesson (see Figure 2). For most of the lessons, we have attached more than one practice.

The third type of task is the independent task, in which case we expect the solution of a complex practical task (see Figure 4). The learner solves the task with the help of the development environment, uploads the source code to the server, which translates it, and decides whether the solution is correct based on the output answers given to the input defined by the program (see Figure 5). The programs are run by the system, so we considered it important that no matter how simple the tasks are, the programs should run in a sandbox similar to the one used on the *progcheck.nejanet.hu* portal [15]. Accordingly, a run time limit of 1 second is set for all programs. Memory and hard disk cannot be accessed directly by programs, and programs that try to do so will not be run.

A 2. versenyző lőtt a legtöbbszőr, összesen 8x.

A 2. versenyző 4 találattal a legtöbb találatot érte el

### A megoldás megadása

A megoldás forrásállománya:

Tallózás... Nincs kijelölve fájl.

Megoldás elküldése

### Tesztek eredménye

Sorszám	Teszteset pontszám	Szerezett pont	Státusz
1	1	1	✓
2	1	1	✓

Tovább a következő feladatra ...

Figure 5: An independent task after grading the uploaded solution

### 3. Improving the effectiveness of courses

At the beginning, our courses were available at *kodolosuli.nejanet.hu*. Secondary school students tested the portal at this stage in 2014. Most of them completed the courses and gave feedback that enabled us to correct pitfalls.

In an online environment, students' performance can be influenced by several factors that stem from students' individual habits and characteristics. These include the ability of students to maintain their attention, their intrinsic motivation to learn [6],[12]. It is also possible to influence these factors in an online environment, as we have the opportunity to increase engagement, which can be defined as a student's cognitive process, active and emotional participation in the learning process [4],[20],[28]. All three factors of commitment (cognitive, behavioral and emotional) can be increased if the text of the curriculum is not only professionally correct, but also good in style and personal in tone [7]. The thought-provoking effect of the text can be facilitated by the inclusion of interesting examples, good metaphors, sometimes astonishing and provocative recordings, questions, contradictions, surprising and in many cases humorous twists [12],[13]. We have modified the text of the curriculum accordingly.

Then we moved to *kodolosuli.hu* and opened the courses for university students who also helped us in correcting errors in 2015. Then we made the portal available for anyone after registration.

After about one hundred registrations, we checked their interactions and submissions to find the weak spots of the courses. Since all the submissions are stored in a database, we were able to find the tasks that proved extremely difficult. The coding tasks that most students could only solve with additional help have been changed, as well as the study material belonging to that task. After implementing these improvements, users were able to solve coding tasks more efficiently which means that they were able to solve each task from fewer attempts.

The next step was that we have made our programming courses gamified. They contain game elements in a non-game context as we mentioned in the first section. The usefulness of the

electronic environment has a significant effect on student satisfaction, which is why it is worth striving to make the content applicable in real life [9]. Accordingly, exercises and coding tasks that students need to complete are applicable to real life to allow students to feel the usefulness of the portal [9]. Practical exercises, such as finding the closest defibrillator to save a life, writing a code that can control the descent of a spacecraft onto a planet, how to move an object from a given (X,Y) coordinate to another position as fast as possible or controlling the parking sensor of a car can increase the efficiency of the e-learning environment. The decision theorem should also be applied if the task is worded as follows: “Given a point, the center and radius of a circle, decide if the point is inside the circle”. To make this task sound more exciting, it is phrased as “We know the range of a laser cannon and the coordinates of the incoming enemy object. Let’s decide if it is worth firing with the cannon.” (Figure 6).

For difficult tasks, we created videos and animations to motivate students to come up with a solution.

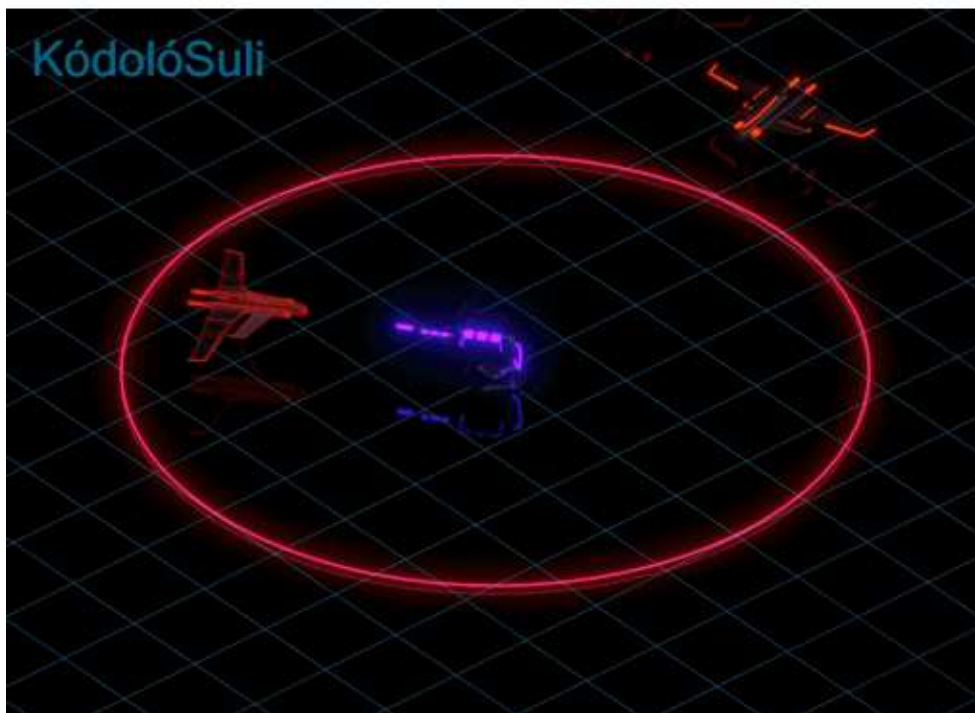


Figure 6: An animation to motivate students to come up with a solution

When creating an effective e-learning environment, the developers should strive to make the learner feel that the teacher, the instructor is present, accessible, so that the feeling of cognitive presence is realized [9]. Therefore, we have integrated the Disqus system into the portal, which is suitable for building communities. Social media is also available, most specifically FaceBook and Twitter.

In 2016, after gaining more than 400 registered users, we started to develop 19 educational game programs that cannot only foster motivation during the playing process but provide faster understanding, since students needed less attempts to solve coding tasks. Most of them are educational simulation games, but we have also implemented games that help in memorisation (memory cards, board game and hangman) and also ones that measure how well the learner has understood the material.

They were written in HTML5 and JavaScript and added to the course right before a coding task. In this paper, we introduce two of them with their learning objectives and learning activities.

### 3.1 The developed educational games

According to Vihavainen et al. in [27], the intervention of educational games in programming courses raises pass rates by 10.8% on average. We were curious, whether developing and adding educational games to our portal can improve its effectiveness.

#### 3.1.1 Counting factorial by calling a function

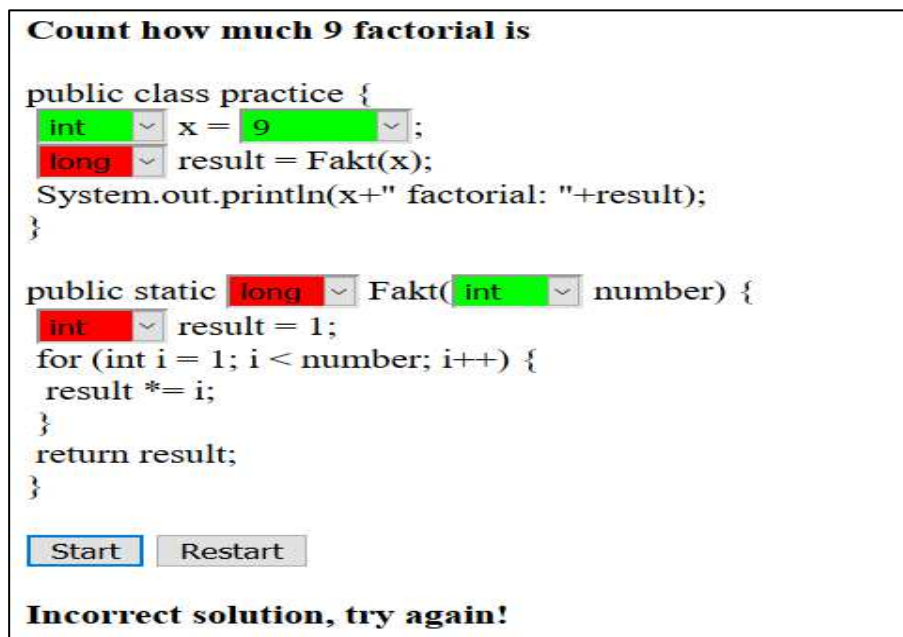


Figure 7: Writing a function with one parameter and calling it. Parameter types can be set by user

**Learning goal:** how to call a function with one parameter, and what happens after calling this function. In the game, the line of the programming statement to be currently executed is blue. If the parameter types are not correct, a warning message is displayed and the program cannot be executed (see Figure 7). Students can observe the flow the execution.

**Activity:** player has to set the value and the type of the current parameter, the type of the formal parameter and the type of the return value.

### 3.1.2. Throw different exceptions

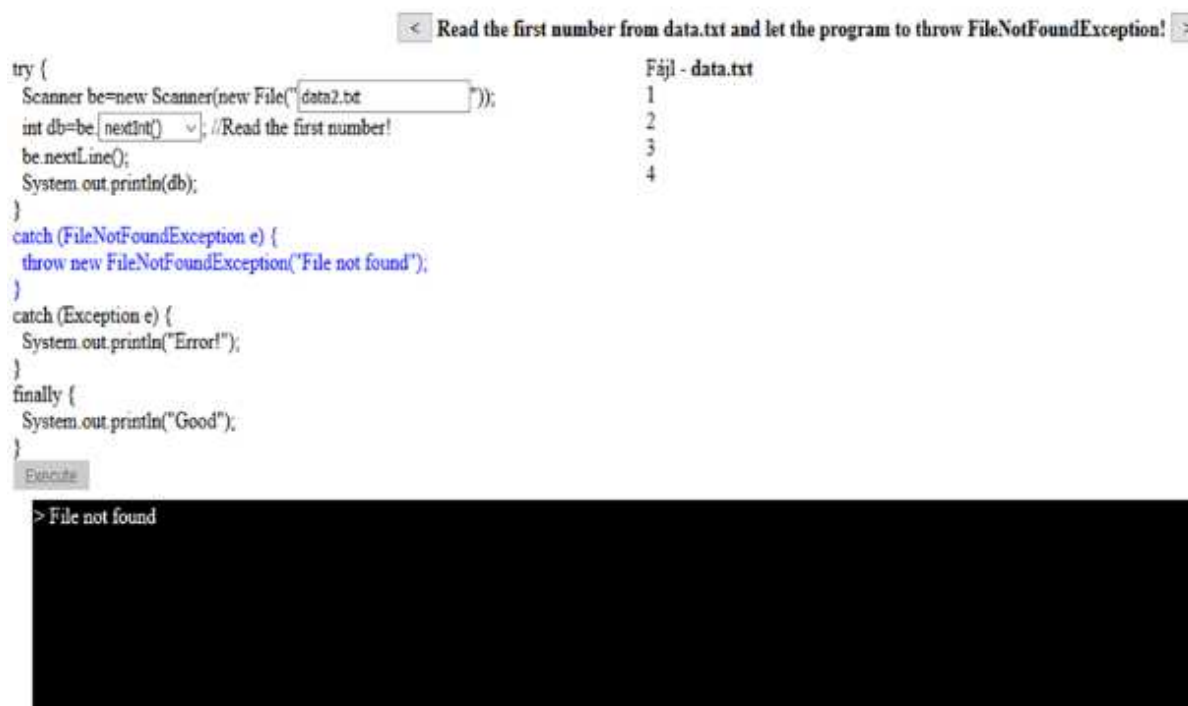


Figure 8: A game for better understanding of exceptions.

**Learning goal:** how to catch exceptions and how to trigger an exception. In the game, the line of the programming statement to be currently executed is blue. Students can observe the flow of the execution: what causes an exception and when, and which catch block will be executed after throwing an exception.

**Activity:** player has to modify statements in the code to throw the requested exception (see Figure 8).

## 3.2 Measuring the effect of the developed games

We have also implemented traditional board games such as Match Pairs Memory Game, Hangman and two own-developed board games, but because they were placed in the portal more than once with different contents, we could not measure their efficiency. Fifteen educational games were placed immediately before an exercise which are followed by another exercise on the current topic for which the game was created, everything else remained unchanged.

We compared the performance of the last 200 students (control group) who registered and completed the last chapter in the curriculum before the game programs had been added into the Java course, and the first 200 students (experimental group), who registered and completed the last chapter in the curriculum, after the addition of the games in the supported coding tasks. We examined how many students were able to solve the first and the second exercises after the first or second try from the 200, before adding the games to the portal (control group) and after adding them to it (experimental group).

The results were impressive and proved that with our thematic didactic game programs, users were able to solve coding tasks more effectively (with less attempts). We published the results in an other paper [14].

#### 4. Directions for further development, opportunities

Our most important goal is to implement new, advanced courses in all four languages that move from the process-oriented world to the basics of object-oriented programming. Obviously, this involves the development of new curricula, games and tasks.

Due to the popularity of the Python language, we have added it to the list of courses, therefore it has been available since 1st September 2020. It was not possible to add all the developed games to the Python course due to the peculiarities of the Python language compared to the other three languages. It is planned to develop educational games that take into account the specifics of the Python language.

Another important goal is to examine how effective the structured curricula is, along with exercises that help to practice and deepen skills. We consider it important to implement functions in the framework with the help of which the efficiency of the curriculum and the reliability of the tests can be measured. This way, the framework also becomes suitable for building tests that really measure knowledge about the course. Here it is necessary to develop computational functions related to the topic of test theory, which can ensure the reliability, authenticity and objectivity of the tasks.

On the other hand, expanding the toolbar can also be important. At the moment, the programs implemented by the users are examined in terms of (in the case of stand-alone tasks) what output it produces for a pre-determined input. It is obviously enough for beginners to test a task at this level, but in the long run it is much more useful to build the expression tree for the source based on the language rules and subject it to preliminary analysis. In the long run, we will also be able to check whether the required subroutines can be found in a program with the appropriate parameters, or in the case of new courses it is possible to determine whether the right object class with the appropriate fields and methods exists. This step can also be important for efficiency analysis, as the end goal is to achieve not only the right but also the right and effective resources. The efficiency analysis of algorithms and programs can be performed classically in three areas:

- (1) Execution time is an important factor for both the user and the server programs. This is also relatively easy to measure at the moment, only the difference between the start time and the stop time of the program after compilation should be taken into account.
- (2) Storage requirements, in which we examine how much memory space the program uses during its operation. This is difficult to examine in the current implementation, in the future a measurement methodology must be created in the framework.
- (3) Complexity indicators, which assigns some degree of complexity to the algorithm of the implemented program. Obviously, the above expression tree construction is required for this analysis.

Finally, we also consider it very important to increase the adaptability of the system because as M. Nadas said... "in adaptability in pedagogy we mean adaptive development/evolution that takes into account the aspects of unity and differentiation, whether we approach it from the needs of individuals, groups or institutions." [18] In other words, it is a system of pedagogical activities that strives to take into account the needs of all participants in the education system at the same time.

It can be interpreted as a dynamic interaction of the concepts of change-reflection-learning/innovation [22]. To make our portal adaptive, the first step can be developing a lighter version of educational games. With these versions, students have fewer and simpler tasks during the game, and in many cases they do not have to type any code, it is enough to select the appropriate code line from the menu, as shown in Figure 9.

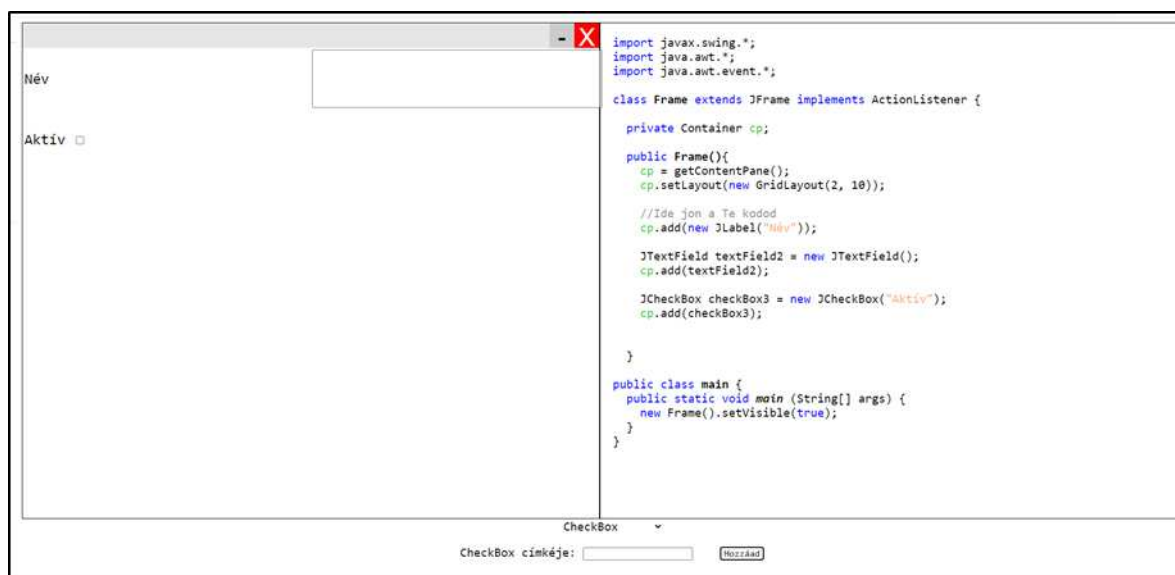


Figure 9: Frame construction by adding components. The selected component appears on the Frame and the appropriate instructions are added to the code.

We want to store the practice tasks in a question bank with a level number from 1-3 while increasing the number of tasks for each level. In the case of each lesson, students start a task at level 1, and after solving it without help, they are allowed to continue with another task at level 2. In case of failure or using help, students need to solve another task at level 1. This solution could lead to adaptive problem solving.

Achieving full adaptability is an extremely labor-intensive task, and the curriculum as well as the LMS also require major modifications. Its implementation, even when using a Moodle system, requires a lot of work from developers [11]. The first step in this work is to identify the initial parameters (age, prior knowledge, learning goals, etc.) of learners using the systems. Further measurements are needed of users' interactions with the digital environment: how much time they spent reading a lesson, solving a task, how many times they improved the solution, what order they followed, and so on. They are related to the possibility of contextual data collection and attach increasing importance to data mining technologies [8].

If we implement a learning algorithm (neural network or inference engine) that can "learn the learner's behavior", then the system is able to operate more efficiently: recognizing the important elements of the tasks, the algorithm can decide if the learner has properly learned the material. From then on, the number of tasks to be solved will be personalized. Moreover, by describing relationship networks, the direction of the step can even be determined between tasks and groups of tasks.

## 5. Conclusion

In the second section of this paper, we have outlined the structure and the functionality of a gamified learning portal for computer programming. We have also presented the ways and methods of improving the effectiveness of this site by using ourself-developed LMS. A well-designed LMS is suitable for tracking student activity. Since we store both the coding tasks of the course and the users' solutions in a database, it is possible to make the task solution part of the courses adaptive, which allows faster and more personalized progress in the system. Today more than 900 users have registered on the portal, who spent a significant amount of time learning the courses. The developed educational games enhanced the effectiveness of the programming course. The practical and independent tasks placed between the chapters of the courses greatly help to master the curriculum. Analysing the data, we found that there are areas in programming education that pose a challenge for learners. We will conduct further research in relation to these, and then develop solutions that will help to effectively address problems.

## Bibliography <sup>1</sup>

1. T. Balla, S. Király: *Online programozás a kódolásuliban* (in Hungarian). In: Szlávi, Péter; Zsakó, László (Eds.) *INFODIDACT 2017*, Budapest, Magyarország. Webdidaktika Alapítvány, (2017) Paper: 8. <https://people.inf.elte.hu/szlavi/InfoDidact17/betolt.html>
2. Blumberg, F., Debby E. Almonte, Jared, S. Anthony, and Hashimoto, N: *Serious Games: What Are They? What Do They Do? Why Should We Play Them*. In: Dill, K.E. (Hrsg.): *The Oxford Handbook of Media Psychology*, Oxford et al. 2013. S. 334-351., (2013). [DOI: 10.1093/oxfordhb/9780195398809.013.0019](https://doi.org/10.1093/oxfordhb/9780195398809.013.0019)
3. P. de Byl, J. Hooper: *Key Attributes of Engagement in a Gamified Learning Environment*. In H. Carter, M. Gosper and J. Hedberg (Eds.), *Electric Dreams. Proceedings, ASCILITE Sydney*, pp.221-230., (2013).
4. R.C. Clark, R.E. Mayer: *E-learning and the science of instruction*. Pfeiffer, San Francisco, (2011)
5. Dörner, R., Göbel, S., Effelsberg, W., Wiemeyer, J.: *SeriousGames- Foundations, Concepts and Practice*. 1st ed.; Springer International Publishing: Basel, Switzerland, (2016). [DOI: 10.1007/978-3-319-40612-1](https://doi.org/10.1007/978-3-319-40612-1)
6. B., Faragó: *Tanulói aktivitás, aktív tanulás és tevékenység online környezetben* (in Hungarian). In: Papp-Danka, Adrienn; Lévai, Dóra (szerk.) *Interaktív oktatásinformatika*, (2015).
7. K. Héjja-Nagy: *Tanulási stratégiák és a tanulói aktivitást befolyásoló egyéni feltételek online környezetben* (in Hungarian). In: Papp-Danka, Adrienn; Lévai, Dóra (szerk.) *Interaktív oktatásinformatika* p. 33-49., (2015).
8. L. Hülber: *Az adaptív online környezet lehetőségei az egyén fejlesztésében*. (in Hungarian) In: Papp-Danka, Adrienn; Lévai, Dóra (szerk.) *Interaktív oktatásinformatika*, p.138-151., (2015)
9. Joo, Y.J., JOUNG, S. & KIM, E.K.: *Structural Relationships among E-learners' sense of Presence, Usage, Flow, Satisfaction, and Persistence*. In: *Educational Technology and Society*, 16(2), 310-324., (2013).

---

<sup>1</sup> The data as indicated in the source are to be followed. The main principle is to make the source reachable.

10. F. Layth Khaleel, N. Sahari, T. Siti Meriam, A. Ismail: *The study of gamification application architecture for programming language course*. In: ACM IMCOM 2015 -Proceedings. Association for Computing Machinery, Inc, 2015. a17, (2015).  
[DOI: 10.1145/2701126.2701222](https://doi.org/10.1145/2701126.2701222)
11. S. Király: *Adaptív oktatás a moodle segítségével* (in Hungarian), In: Új köznevelés (2016).  
<https://folyoiratok.oh.gov.hu/uj-kozneveles/adaptiv-oktatas-a-moodle-segitsegevel>
12. S. Király: *Tanulás támogatása digitális környezetben* (in Hungarian). In: OKTATÁS-INFORMATIKA 2016: 1 pp. 29-40., 12 p., (2016)
13. S. Király: *How to Implement an E-learning Curriculum to Streamline Teaching Digital Image Processing*, In: ACTA DIDACTICA NAPOCENSIA 9: 2 pp. 13-22., 10 p., (2016)
14. S. Király, T. Balla: *The effectiveness of a fully gamified programming course after combining with serious games in a computer programming portal*, In: ACTA DIDACTICA NAPOCENSIA 13: 1 pp. 65-76., 12 p., (2020). [DOI: 10.24193/adn.13.1.7](https://doi.org/10.24193/adn.13.1.7)
15. S. Király, Sz. Székely: *How to use our own program evaluation system to streamline teaching computer programming*. In: Teaching Mathematics and Computer Science 13:(1) pp. 73-80., (2015). [DOI: 10.5485/TMCS.2015.0384](https://doi.org/10.5485/TMCS.2015.0384)
16. B. Kumar: *Gamification in education - learn computer programming with fun*. In: International Journal of Computers and Distributed Systems, Vol. No.2, Issue 1, December 2012, pp. 46-53., (2012)
17. Lengyelne, Molnár Tünde: *IKT mint oktatás támogató rendszer* (in Hungarian). In: Bárdos, Jenő; Kis-Tóth, Lajos; Racsko, Réka (szerk.) XIII. Országos Neveléstudományi Konferencia: Változó életformák - Régi és új tanulási környezetek: Absztraktkötet, (2013)
18. M. Nádas, M.: *Adaptív nevelés és oktatás* (in Hungarian). Magyar Tehetségsegítő Szervezetek Szövetsége. Budapest, (2010)
19. Muratet, M., Torguet, P., Viallet, F. & Jessel, J.-P.: *Experimental feedback on Prog&Play: a serious game for programming practice*. in: L. Kjeldahl and G. Baronosk (Eds.), EUROGRAPHICS1–8., (2010).
20. N. Pellas: *The influence of computer self-efficacy, metacognitive self-regulation and self-esteem on student engagement in online learning programs*. Evidence from the virtual world of Second Life Computers in Human Behaviour, 35, 157-170, (2014).
21. R. Racsko, L. Kis-Tóth: *Új tanulási környezetek a köznevelésben: a személyes tanulási terek és az interakcióelemzés elméleti háttere* (in Hungarian). In: Námesztovszki, Zsolt; Vinkó, Attila (szerk.) XXI. Multimédia az oktatásban és II. IKT az oktatásban konferencia = XXI Naučna konferencija „Multimediji u obrazovanju” i II Naučna konferencija „IKT u obrazovanju”. Szabadka, Szerbia : Újvidéki Egyetem Magyar Tannyelvű Tanítóképző Kar, pp. 253-258. 6 p., (2015)
22. N. Rapos, K. Gaskó, O. Kálmán, Gy. Mészáros, Gy. (2011): *Az adaptív-elfogadó iskola koncepciója* (in Hungarian). Oktatókutató és Fejlesztő Intézet. Budapest, (2011).  
<http://mek.oszk.hu/13000/13021/13021.pdf>
23. M. Sailer, J. Hense, H. Mandl, M. Klevers: *Psychological Perspectives on Motivation through Gamification, Interaction Design and Architecture(s)*. Journal - IxD&A, N.19, 2013, pp. 28-37., (2013)

24. P. Szlávi, L. Zsakó, G. Törley: *Programming Theorems Have the Same Origin*. In: Central-European Journal of New Technologies in Research, Education and Practice Volume 1: Number 1, pp. 1-12., 12 p., (2019) [DOI: 10.36427/CEJNTREP.1.1.380](https://doi.org/10.36427/CEJNTREP.1.1.380)
25. Sauv e, L., Renaud, L., Kaufman, D., & Marquis, J. S.: *Distinguishing between games and simulations: A systematic review*. Educational Technology & Society, 10 (3), 247256., (2007).
26. Tsai, M.-J., Huang, L.-J., Hou, H.-T., Hsu, C.-Y., Chiou, G.-L.: *Visual behavior, flow and achievement in game-based learning*. In: Computers & Education. 98, pp 115–129., (2016). [DOI: 10.1016/j.compedu.2016.03.011](https://doi.org/10.1016/j.compedu.2016.03.011)
27. Vihavainen, A., Luukkainen, M. & Kurhila, J.: *Multi-faceted support for MOOC in programming*. In: Proceedings of the 13th Annual Conference on Information Technology Education, ACM, New York, pp. 171–176, (2012).
28. M. Wolf: *Learning to Think in a Digital World*. In: Bauerlein, M. (ed.): *The digital divide: arguments for and against Facebook, Google, texting, and the ages of social network*. Jeremy P. Tarcher/Penguin, New York. 34-37., (2007).
29. L., Zsak o, P. Szl avi,  . Harangoz o: *Joining Programming Theorems, a Practical Approach to Program Building*. In: Annales Universitatis Scientiarum Budapestinensis de Rolando Eotvos Nominatae, Sectio Geologica 17: 1 pp. 155-172. 18 p. (1998).

## Authors

### **BALLA Tam as**

Eszterh azy K aroly University, Faculty of Informatics, Institute of Mathematics and Informatics, Department of Computational Science, Eger, Hungary,  
e mail: balla.tamas@uni-eszterhazy.hu

### **KIR ALY S andor**

Eszterh azy K aroly University, Faculty of Informatics, Institute of Mathematics and Informatics, Department of Information Technology, Eger, Hungary,  
e mail: kir aly.sandor@uni-eszterhazy.hu

## About this document

### **Published in:**

CENTRAL-EUROPEAN JOURNAL OF  
NEW TECHNOLOGIES IN RESEARCH,  
EDUCATION AND PRACTICE

Volume 2, Number 2. 2020

ISSN: 2676-9425 (online)

### **DOI:**

10.36427/CEJNTREP.2.2.833

## License

Copyright   BALLA Tam as, KIR ALY S andor. 2020

Licensee CENTRAL-EUROPEAN JOURNAL OF NEW TECHNOLOGIES IN RESEARCH, EDUCATION AND PRACTICE, Hungary. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license.

<http://creativecommons.org/licenses/by/4.0/>