

Code School: Learn to code by practicing

Sándor Király, Tamás Balla, Gábor Kusper

`kiraly.sandor@uni-eszterhazy.hu`, `balla.tamas@uni-eszterhazy.hu`,
`gkusper@aries.ektf.hu`

Institute of Mathematics and Computer Sciences
Eszterházy Károly University
Eger, Hungary

Abstract

Today the IT industry provides an attractive career choice to novice software developers, but even then, majority of the students pursuing higher education in computer programming field are not finishing with enough competencies and capabilities to meet the industry expectations. It has been observed that such students are not sufficiently focused on computer programming during their school studies. The main reason for unconcern in computer programming is identified as the lack of motivation and engagement of students in learning the programming concepts. `kodolosuli.hu` is a learning portal that offers interactive programming courses in three different programming languages: C++, C# and Java. It has been developed for students who are interested in computer programming but do not know how to start. Students, who are not offered programming lessons at school and teacher help are welcome. The framework of this site has an extensive grader tool that helps students test their codes without intervention from a teacher, thus providing a flexible learning experience. The gamification in this portal is expected to increase students' engagement and motivation in learning. The developed LMS (Learning Management System) offers the authors the ability to improve the courses according to the solved problems and the comments of the forums. Tasks are stored in an exercise repository to make creating tests and exams easier. The course is available in Hungarian.

Keywords: gamification, Learning Management System, computer programming, game-based learning

1 Introduction

Today in Hungary, there are about 20,000 unfilled IT jobs and most of them are software development positions. The lack of properly trained staff is a problem for hundreds of companies although thousands of software engineers and software developers get a degree in every year.

The IT sector could even be a key sector in Hungary since its value-added ratio is 90% and it produces 12% of

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: E. Vatai (ed.): Proceedings of the 11th Joint Conference on Mathematics and Computer Science, Eger, Hungary, 20th – 22nd of May, 2016, published at <http://ceur-ws.org>

the Hungarian GDP (Gross Domestic Product). That is why it would be particularly important that high school students choose this profession in larger numbers and, moreover, the demand for high-skill software developers is now growing faster than the supply.

To achieve this, IT must be made more attractive for students, and it is particularly important that the high school students who are interested in this area learn to program. In our experience, students who have already learnt computer programming can work in the IT sector in their first academic years even as software developers, gain work experience and acquire this professional status faster than those students who only start to learn programming in the first academic years.

The main problem is that during learning programming both algorithms and coding are needed to be learned at the same time. Being familiar with the syntax of a computer language and writing programs in it are challenges for youngsters. To understand algorithms is also a big challenge. It is no accident that coding and making algorithms are different subjects in higher education. Most Hungarian secondary schools prepare students for only the mid-level exam of informatics that does not include knowledge of programming. The reduced 1 lesson per week is not enough to teach students both the usage of office programs and programming although there are thousands of talented students who are interested in programming but not able to learn it by themselves.

Though, books and notes in Hungarian and courses in English are available it seems insufficient for most of them. Portals such as CodeAcademy, CodeSchool or Code Avengers that teach different programming languages in an interactive way in English might be useful. In Hungary, this type of portal, that teaches C++, Java and C#, is not available, for instance learning C++ or C# languages in Hungarian for free is impossible. For example, memmoc.hu portal of University of Miskolc only offers C and Java language courses [Kusper2016]. Algorithms can be learned from books but gaining experience for their usage requires solving problems and coding them in a computer language. To do so, however, the knowledge of a programming language syntax is elementary.

kodolosuli.hu portal wishes to solve this problem. In this website, students can learn how to code basic algorithms in a chosen language that can be C#, C++ or Java for free. The main objective of this site is to teach students the basic programming structures that are necessary to be able to pass the advanced level exam of informatics. After completing the course on the website they will be able to use an IDE (Integrated Development Environment) and solve programming tasks. The programming knowledge gained from this portal can be enough for students to understand programming resources on the internet and they can achieve more in this field.

kodolosuli.hu teaches and trains the usage of the essential lexical unit and the control structures, how to make and use procedures and functions including the built-in functions of languages, and the way of applying standard I/O statements, methods. It also introduces the way of storing data in records, and that they can be stored even in arrays or lists. The tools of exception handling are also discussed in all the three courses through practical problems. Gamification applications provide motivation and make the understanding easier and faster [Sailer2013]. The practical problems are mostly related to the field of making computer game programs. This portal is developed for expressly Hungarian students.

2 Learning and practicing in this site

In an online environment students' achievements can be influenced by several factors that stem from their features. These factors can be for example their ability to maintain their attention or their intrinsic motivation [Biggs2012]. One of the most important factors that must be considered in terms of their activity is students' engagement: this can be defined as the student's cognitive process, active participation and emotional involvement in the learning process [Pellas2014]. In an online environment it is also important to increase the student's cognitive involvement [Wolf2007]. According to various e-learning objectives e-learning environments can be differentiated [Clark-Mayer2011]. In kodolosuli.hu directive e-learning is preferred since the material of this site is supposed to be new for the participants.

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. (Figure 1)

Students are asked to write the proper code in the panel on the right side. After the coding is completed, clicking on the Send button uploads the solution. If it is correct the next unit will be displayed. If the code is not correct a message including the number of the line that contains the error will be displayed. By clicking on Help button the program makes instructions available for the student to be able to correct the mistake. If the student fails an exercise he can move on to another unit for now and come back later to try again. Typically, there is more than one exercise for each topic. Students' attempts are stored in a database so teachers can even



Figure 1: 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

help them with the solution.

The goal is to learn coding in a chosen language alone in order to be able to code the basic computer programming problems (searching, selection, counting, etc.). There are several units that do not contain theoretical material just practical exercises from real life. For example, finding the closest defibrillator to save a life; controlling the parking sensor of a car; how to move a character from a given X,Y coordinate to another position as fast as possible.

19 gamified applications have been developed for students to test their knowledge in a given topic during playing with them. One of them can be seen on Figure 2. These software not only foster the motivation but provide faster understanding [Domínguez2013][Khaleel2015].

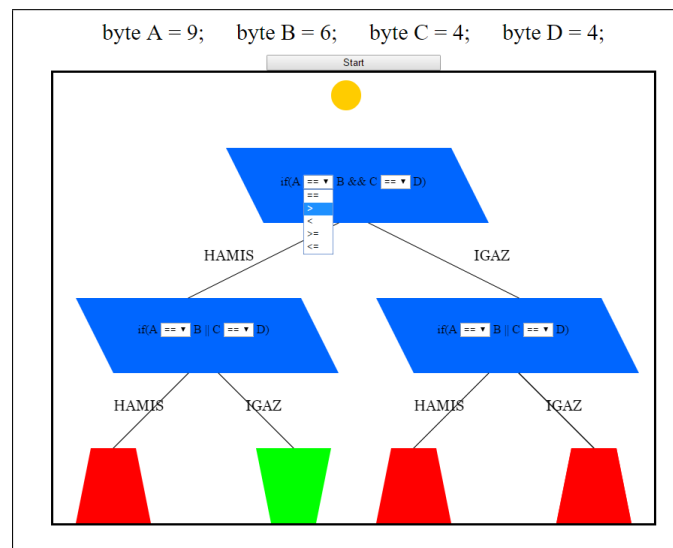


Figure 2: Drop the ball into the green bucket. IGAZ=TRUE, HAMIS=FALSE.

Naturally, the portal contains a Forum for users to communicate with each other on a variety of subjects and

to share their experience gained during coding, or they can collaborate with each other [Arefin2015][McDowell-Werner-Bullock-FernaldKumar2012][Teague-Roe2008]. By reading this forum, teachers can help students, improve the system and remove bugs from the programs.

The exercise repository that is part of the system contains several tasks that are not available for the students during the course except for tests and exams. Exercises can be selected by either teachers or the system. Students who completed all the tasks can download a computer game for both Android and Windows platforms. Naturally, different game programs have been developed for both platforms for all the three languages.

3 The developed Learning Management System

The website has been 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 administrator. Students can reach only the curriculum unlike the administrator who has full access to both the materials and the users' data.

3.1 The admin interface

Users logged in as administrator can create new languages, new curriculums and new exercises or edit the previous ones and change the order of the units presented for students. (See Figure 3) It is possible to see the students' solutions, and this way the difficulty level of the exercises and the mistakes can be analysed as well. It also records that how many times a unit was opened and how often tried a student to solve an exercise.

Cím	Fejezetek száma	Statusz	Statusz átminősítése	Nyelv törölése	Fejezet átmozgatása	Részletek
Változók	5	✓	Passzív	Töröl	Le	Részletek megtekintése
IO műveletek	4	✓	Passzív	Töröl	Le Fel	Részletek megtekintése
Operátorok - aritmetikai	3	✓	Passzív	Töröl	Le Fel	Részletek megtekintése
Választó előkezdők	2	✓	Passzív	Töröl	Le Fel	Részletek megtekintése
Ciklusok	6	✓	Passzív	Töröl	Le Fel	Részletek megtekintése
Tömbök - vektorok	4	✓	Passzív	Töröl	Le Fel	Részletek megtekintése
Beolvasás szöveges fájlból - 1	2	✓	Passzív	Töröl	Le Fel	Részletek megtekintése
Metódusok - I	6	✓	Passzív	Töröl	Le Fel	Részletek megtekintése

Figure 3: Access to all chapters

The system logs all the actions users perform so the publication of the achievements becomes easier, and it helps the developers improve the curriculum. (Figure 4)

3.2 The graders

The most important parts of the system besides the curriculum are the graders. There are two types of graders: the first one examines the code line by line to see whether it is the same as the solution. This is the reason why exercises are defined precisely, e.g. the name of the loop variable is also specified. (Figure 5) The purpose of this method is to get the students to use "speaking variable names" and practice the right coding style. However, in some cases this grader could cause problems especially when longer codes are given, because it might not accept

Tananyag kezelése Tananyagok feldolgozása Felhasználók kezelése Tesztek készítése Fórum Tesztek

A C# programozási nyelvhez tartozó leckék

Feladat szövege:

Deklarálj egy statikus függvényt, annak fejrészét, neve faktor, visszatérési értéke long!
Két formális paramétere van, az egyik param1, típusa long, a másik param2, típusa int és referenciát vár, olyan változóval, melynek még nincs kezdőértéke!

Feladat megoldása:

static long faktor(int param1,out int param2)

A programozási nyelvhez kapcsolódó tanulható fejezetek

Fejezet címe	Begépett megoldás	Helyes a megoldás
2016-01-16 13:51:46	static long faktor(long param1, int param2)	✗
2016-01-16 13:53:01	static long faktor(long param1; out int param2)	✗
2016-01-16 13:53:13	static long faktor(long param1, out int param2)	✗
2016-01-16 13:54:15	static long faktor(int param1, out int param2)	✓

Figure 4: Tracking a user's activities

codes with correct syntax and semantics. Furthermore, the description of the exercise might be too long due to the very strict specification.

Listák - I

Mi a baj a vektorral és a mátrix-szal? Az, hogy ha nem tudjuk, hogy hány adatot kell eltárolni benne, akkor deklarálni sem tudjuk. Illetve tudjuk, csak éppen akkor is 100 elemű lesz a vektor, ha 10 elemet tárolunk benne, és akkor is, ha 40-et.

Erre jó a List a C#-ban. Ebben is adatokat tudunk tárolni, csak éppen mérete dinamikusan változik. Csak annyi helyet foglal el, amennyi adatot foglalunk.

Persze ehhez az elején meg kell adni a névtérrel, ahol megtalálható:

```
using System.Collections.Generic;
```

Azaz a lista egy **generikus adatszerkezet**. Fontos, hogy elemei kb. akármilyen típusúak lehetnek!

Pelda:

```
List<int> a=new List<int>();
```

Ez egy int típusú elemek tárolása szolgáló lista. 0 eleme van kezdetben.

```
a.Add(10);
```

Most már 4 bájtot foglal, ha az int 4 bájtos.

```
a.Add(20);
```

Most már 8 elemű a lista, 8 bájtot foglal.

Írj olyan kódot, amely deklarál egy t nevű listát, melynek elemei int-ek. A második sorban egy int típusú, db nevű változóba "másold bele" a lista elemeinek a számát!

✗ Hiba a megoldás 2. sorában!

```
1 List<int> t=new List<int>();
2 int db=t.Length;
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

Elküld

Segítség

Feladatlista

Figure 5: Grader in action. The curriculum is on the left side, the exercise and the code panel are on the right side. In this case, the second line is incorrect.

When users are getting familiar with coding algorithms they need to solve more complex practical problems, therefore this grader cannot be used. That is why another type of grader is used for longer codes and exercises. It first compiles the submitted code, and then, if no compile error have occurred, it executes this code for different test data stored in files. For each task, input and output files are available and are used by this grader to test the code. Naturally, users are informed about the correctness of the submitted programs.

Programs, coded and submitted by students, are executed (after the compilation) in a sandboxed environment also used in the portal of progcheck.nejanet.hu [Király-Szekely2015]. The default memory and execution time limit are taken into account and enforced but these constraints can be changed for each task. Students are informed about the memory and execution time their submissions use up while if they overstep the limits an error message is presented for them. The submitted programs cannot access neither the memory nor the hard disk directly. Programs that try to break the rules are not executed at all.

4 The gamified curriculum

Students sometimes find it hard to learn a programming language. They are often confronted with unfamiliar programming terms and are required to visualize the processes that happen in computer memory. Others find this a burden and end up memorizing the processes without understanding them.

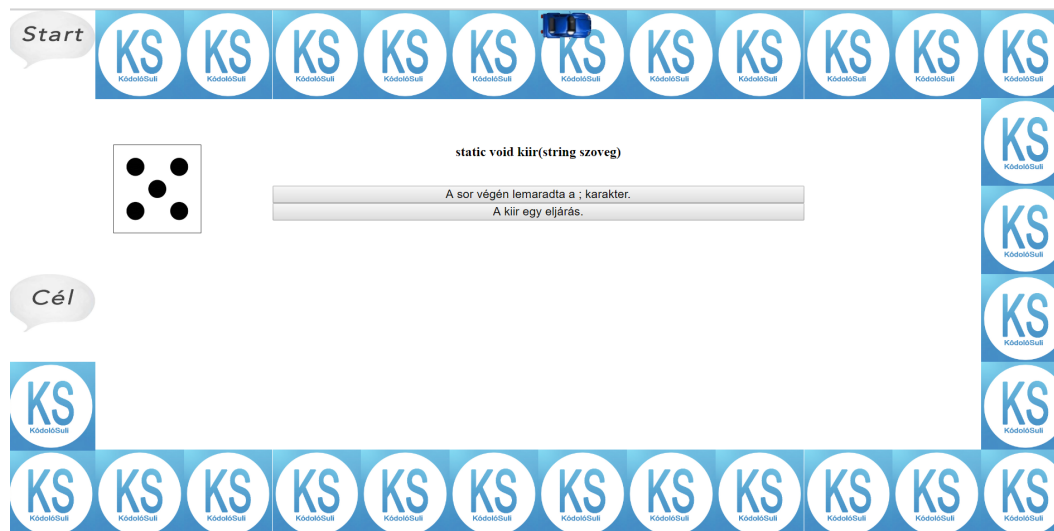


Figure 6: Boardgame that helps in summarizing the topics

This situation invariably leads students to get low grades in their programming subjects and abandon learning programming. Studies have shown that applying gamification elements in websites engage users [Fotaris-Mastoras-Leinfellner-Rosunally2015]. Games are engaging because they have the potential to satisfy basic psychological needs for competence and autonomy [Li-Dong-Unitch-Chasteen2016].

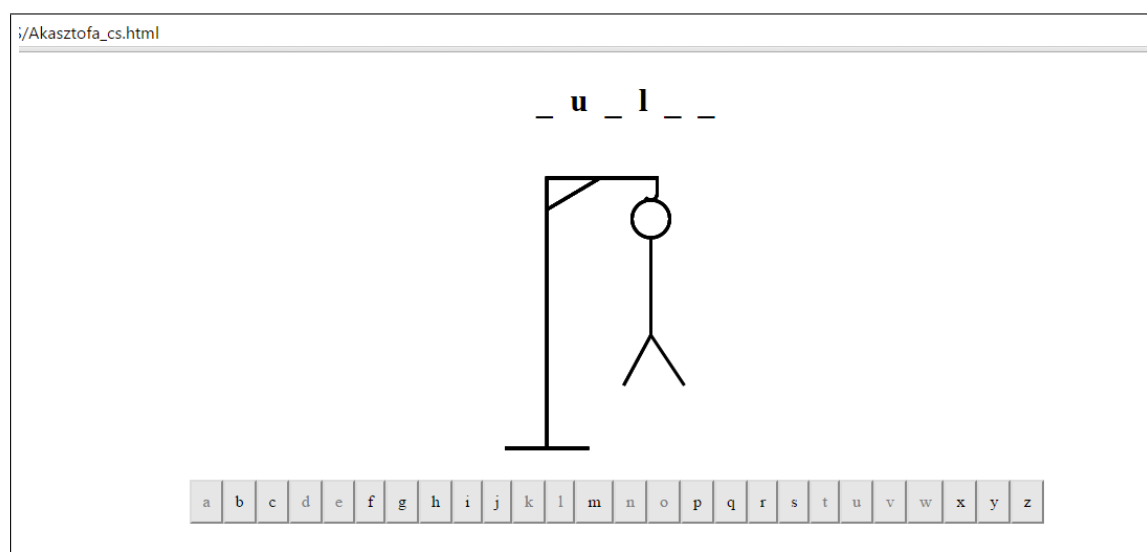


Figure 7: Hangman for memorizing keywords

The most important games have been implemented in HTML5 and JavaScript and put in iframe. The images of games are either drawn on a HTML5 Canvas during execution or loaded from the hard drive before running the code. For the most games, the JQuery library is also used.

Games are not only motivating but also filled with educational content. The experience gained from the current

game is not enough, this experience must be transformed to educational experience [Garris-Ahlers-Driskell2002]. After games, the site offers real coding tasks relating to the game. When students are playing with these games they not only play but also learn the given topic [Ben-Ari2013][Kumar2012]. Traditional games such as Boardgame (Figure 6) Hangman (Figure 7),) and Memory cards have also been implemented. The more complicated exercises involve students to code algorithms of computer games. For example, they need to write a code that can control the descent of a spacecraft onto a planet. The submitted solutions for these problems are evaluated by the second type of grader.

5 Conclusions and Future Work

The aim of this study is to demonstrate a novel website, kodolosuli.hu, that is an online interactive platform that offers free coding classes in 3 different programming languages. The goal is to reach youngsters who have little-to-no coding experience, but wish to learn. The website starts by presuming no prior knowledge at all, and lets students work through small exercises and slowly growing in complexity. This website offers gamified curriculum. Gamification refers to the use of game elements in a non-game context to increase engagement between humans and computers. The implemented game programs not only increase the effectiveness of learning but enhance students' understanding. As the more complicated exercises are related to the implementation of different game programs they also increase the engagement of students in learning programming. The successful students who complete the course of a given computer language are allowed to download a game program for both Android and Windows platforms that is not available anywhere else.

The developed LMS and Forum support keeping track of student progress during lessons and foster the improvement of the material.

It is planned to give the system the following function: instead of the two types of user solution (student and administrator) there will be a teacher type of user who cannot create curriculums, but they can create classes and add students to the system by themselves. They can upload their own tests for practice, or they can give homework, which are corrected by the system automatically.

References

- [Arefin2015] Ahmed Shamsul Arefin, Arefin A.S. 2015. Pedagogy of Computer Programming: An Interactive and Collaborative Learning Approach. *Macquarie University Postgraduate Certificate of Higher Education EDCN 871 Final Project*.
- [Ben-Ari2013] Ben—Ari, M. (2013). Visualization of Programming. *Improving Computer Science Education*, 52.
- [Biggs2011] Biggs, J.,& Tang, C. (2011). Teaching for quality learning at university. *McGraw-Hill International*.
- [Clark-Mayer2011] Clark, R.C., Mayer, R.E. (2011). E-learning and the science of instruction. *Pfeiffer, San Francisco, 2011*.
- [Fotaris-Mastoras-Leinfellner-Rosunally2015] P. Fotaris, T. Mastoras, R. Leinfellner, Y. Rosunally (2015). Who wants to be a Pythonista? Using Gamification to Teach Computer Programming. *7th International Conference on Education and New Learning Technologies*, Pages: 2611-2619.
- [Domínguez2013] Adrián Domínguez, Joseba Saenz-de-Navarrete, Luis de-Marcos, Luis Fernández-Sanz, Carmen Pagés, José-Javier Martínez-Herráiz (2013). Gamifying learning experiences: Practical implications and outcomes. *Elsevier, Volume 63*, April 2013, pp 380–392.
- [Garris-Ahlers-Driskell2002] GARRIS, R., AHLERS, R., Driskell, J.E, (2002). Games, motivation and learning, A research and practice model. *Simulation and Gaming*, 33(4), 441-467.
- [Khaleel2015] Firas Layth Khaleel, Noraidah Sahari, Tengku Siti Meriam, Amirah Ismail (2015). The study of gamification application architecture for programming language course. *ACM IMCOM 2015 - Proceedings. Association for Computing Machinery, Inc, 2015*. a17.
- [Kiraly-Szekely2015] Sándor Király, Szilveszter Székely (2015). How to use our own program evaluation system to streamline teaching computer programming. *Teaching Mathematics and Computer Science* 13:(1) pp. 73-80. (2015)

- [Kumar2012] Balraj Kumar (2012). Gamification in education - learn computer programming with fun. *International Journal of Computers and Distributed Systems*, Vol. No.2, Issue 1, December 2012, pp. 46-53.
- [Kusper2016] Kusper G., Percze G., Kovács L., Nehéz K., Radványi T., Király S. (2016). A MeMOOC online informatikai egyetem koncepciója, The concept of the MeMOOC Online IT University. *Workshop 2016 (CD)*. Budapest: Nemzeti Információs Infrastruktúra Fejlesztési Intézet (NIIFI) 2016. p. CD. (ISBN:978-963-88335-6-3)
- [Li-Dong-Untch-Chasteen2016] C. Li, Z. Dong, R. H. Untch, and M. Chasten (2013). Engaging Computer Science Students through Gamification in an Online Social Network Based Collaborative Learning Environment. *International Journal of Information and Education Technology*, Vol. 3, No. 1, February 2013.
- [McDowell-Werner-Bullock-FernaldKumar2012] McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). The effects of pair-programming on performance in an introductory programming course. *SIGCSE '02 Proceedings of the 33rd SIGCSE technical symposium on Computer science education*: Pages 38-42.
- [Pellas2014] Pellas, N. (2014). 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.
- [Sailer2013] Michael Sailer, Jan Hense Heinz Mandl and Markus Klevers. Psychological Perspectives on Motivation through Gamification, *Interaction Design and Architecture(s) Journal - IxD&A*, N.19,, 2013, pp. 28-37.
- [Teague-Roe2008] Teague, D., & Roe, P. (2008). Collaborative learning: towards a solution for novice programmers. *Paper presented at the Proceedings of the tenth conference on Australasian computing education-Volume 78*.
- [Wolf2007] WOLF, M. (2007). Learning to Think in a Digital World. In: Bauerlein, M. (ed.) (2011): *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.