

Developing games while learning programming

Galina Teodosieva¹ and Teodosi Teodosiev²

¹Profiled high school of natural sciences and mathematics, Shumen,
Bulgaria

²Department of Mathematics and Computer Science, Shumen University,
Bulgaria

Abstract

The article proposes the development of games in the learning process as a possible approach to increase the motivation of students. The well-known approach of connecting learning to play has been chosen. Linguistic and algorithmic elements are considered through interesting tasks - games. It turns out that games can be programmed, even with basic programming knowledge. Students take on the role of game developers, which gives them the opportunity to practice a professional activity very similar to real one. Based on several selected game tasks, it is discussed how games can be developed in parallel with learning basic programming structures. The development of computer games during the training can spark interest in new topics and stimulate the memorization of what has already been learned.

Keywords: programming, game, learning, teaching, development, introduction to programming.

1 Introduction

In recent years, the trend of including games in education in various disciplines has intensified [1], [2]. This is caused by on the one hand the lack of motivation for learning among young people, and on the other hand the huge interest of young people in computer games. A few years ago, the World Health Organization recognized computer and video game addiction as a disease. It is obviously tempting to use this interest to motivate learning and increase learning outcomes. There is also a danger of using games to convince students that learning is a game, which is not the case at all. Training requires focused effort and is inevitably associated with challenges.

Programming training poses at least three very complex questions. First, how to teach students the syntax and semantic of a new language of expression. Second, how to develop algorithmic thinking in students. Furthermore, how to break the solution into a sequence of steps. Training in other subjects is mainly in the use of pre-made algorithms. This also leads to difficulties in solving geometric and practical tasks. There are no pre-made algorithms, as in programming. Third,

as with practical tasks, there is another factor complicating learning. This is the problem with modelling - compiling a mathematical and computer model respectively. All these difficulties lead to many difficulties for the learners in mastering the subject and to loss of interest in programming. The last difficulty is often overcome by solving only etude tasks. There the model is given and it remains to be algorithmized and coded. However, this approach renders solving practical tasks meaningless.

The idea we offer is especially suitable for initial programming training. Interest in computer games can be used to showcase practical applications of programming.

2 Algorithms Games in programming training

Programming courses are necessary for students because they are basic for the development of applications in their future activities. In general, the purpose of these courses is to allow students to master the knowledge: to present the task in a set of specific steps, and then to encode them in a programming language.

Programming training faces serious problems. Many students give up programming because they cannot understand the concepts taught in the Introduction to programming course. Among the reasons for this problem are: low motivation, lack of ability to think abstractly, low level of mathematical knowledge [3].

These difficulties cause many students difficulties in mastering the subject and loss of interest in programming. On the other hand, game development is an opportunity for students to perform actual professional activity and can therefore serve as an incentive for their motivation and activity.

The choice of tasks in the introductory programming course is challenging. On the one hand, the task should be easy to understand to focus the students' attention on the algorithmic and linguistic elements. On the other hand, it is good for the task to be interesting to increase their motivation. Here, however, there are problems with lack of knowledge on the subject area and therefore with compiling the model. In that sense, games seem like a good choice. The games are familiar to students (many of them have played them on paper). Significant knowledge and habits can be acquired through playing games. The game element creates interest and at the same time stimulates the overcoming of technical difficulties, which could otherwise demotivate students for the new activity at an early stage.

An alternative to generating success in the learning process is the use of games, as they can add benefits in the educational context such as motivation, enjoyment, interdisciplinarity and development of cognitive abilities. Two ways can be used to include computer games in training.

The first is known as "serious games" in which students participate as players, i.e., they must solve the task collectively or individually. Vahldick et al. [4] presents in their review article 40 games related to the subject and competencies in the introductory programming course. The idea is to provide the learners with games, i.e., the students encounter pre-set problems and must overcome them. It is good to set out considerations of how the game program is organized, what techniques are used to create it, how it can be improved.

Based on these considerations, the student can try to modify the game on their own. The next step in this direction will be to create your own games, and perhaps

completely serious practical programs.

There is another way to use the games. Using interest in games, game development is included during training. In this case, the linguistic and algorithmic elements are considered through interesting tasks. It turns out that games can be programmed, even with basic programming knowledge.

From the programmer's point of view, rules are a set of procedures defined over data structures (a set of variables). The procedures and data together embody the "inner world" of the game program.

In the process of playing, the inner world is constantly changing firstly, according to its own laws, secondly, under the influence of the player (players). Thus, at any given time, the game is in a specific state, perceived by the player as the current situation.

In the second way, students take on the role of game developers, which allows them to practice a professional activity very similar to a real one.

3 A set of games

Text of Results and Discussion... Problem solving is a mandatory element of the content of programming training. By solving problems, students master skills and habits for using theoretical knowledge in their practice. Moreover, it is the ability to solve problems, i.e., performing certain actions with the information from the conditions of the task, means mastering knowledge.

The development of games during the introductory programming course provides students with experience that they can harness throughout their careers. Positive and proactive attitudes during game development are obvious especially when it comes to seeking new knowledge.

Based on several selected game tasks, it is commented on how games can be developed in parallel with learning basic program constructions and structures. For this reason, the recommendations for the selection of tasks are shown and commented as part of each of the main topics in the course "Introduction to Programming".

3.1 Variables, Data Types and Expressions. Conditional Statements. Multiple Selection and Switch Statements.

It can start with a game suitable for young students.

✓ *"Calculator" [5].*

A simple training program that uses the "innate" abilities of the computer - computing. The program generates two pseudo-random numbers from a given interval and selects an arithmetic operation (+, -, *, /), then allows the student to answer and checks the answer.

This is a training or exercise in a game environment. The goal is to acquire some useful habit or knowledge. Here learners can get acquainted with the functions for generating a pseudo-random number. Show how to reformat a numeric interval. When considering the control structures, the most important point is the correct construction of the conditions (Boolean expressions) and their correct use.

After entering operations with integers and real numbers and branching programs, the following task can be solved.

✓ *"Calendar".*

This program on an entered date for an arbitrarily set year (1592..4902). displays the day of the week. For each date of the specified range, the number of the day of the week (0 - for Sunday, 6 - for Saturday) is equal to the remainder in integer division by 7 of the expression

$$84 + [2,6.m-0,2] + d + y + [y / 4] + [c / 4] - 2.c, (1)$$

where d - number of the day in the month; m - number of the month in the year (1 - March, January and February - 11 and 12, but from the previous year); c - the number formed by the first two digits of the year; y - the number formed by the last two digits of the year.

Here the model of the task is given in advance to concentrate the attention on the algorithmic and linguistic elements. This program makes it possible for students to focus on the connection between the types of variables and the expression (1). In addition, integer operations are performed. The task provides an opportunity to comment on the features of the arithmetic division operation. The Switch Statements is also used.

3.2. Repetition structure (Loops).

The implementation of loops processes is a fundamental topic in programming. The genesis of these processes is the repeated calculation. Each loop process has four main elements: initial initialization, end condition, preparation for the next iteration, loop body.

One of the difficult moments in compiling cyclic programs is the initial initialization of the loop parameters. These values require careful selection because incorrect initial values can lead to erroneous results in an algorithmically correct program. In addition, this selection must ensure the completion of the loop.

A simple logic game can be used to master cyclic processes.

✓ *"Guess the number"*.

This is an evolving logic game. The essence of the game is that the computer guesses the number chosen by the user with the least tries. Optimal guess of the number is ternary search.

Failure to initialize a variable can lead to incorrect program operation (undefined result value). To avoid such an error, all used variables must be defined in the code.

In this topic it is appropriate to implement (maybe as an independent work) a variation of the "Calendar" task, in which instead of entering a date only a month and a year are entered and a calendar for the whole month of the year is displayed.

3.3. Arrays. Strings.

✓ *"Bulls and Cows"*.

This is also a logic game. It is more complex than the previous one and requires, like all logic games, not only speed of thinking, but also depth and accuracy, skills to analyse options. The program generates a four-digit number (without repeating digits). The player must assume at each step about the number generated. The program tells him how many known digits (cows) and how many of them are in the correct position (bulls). The task must be solved with a minimum number of attempts.

After getting acquainted with the "array" structure, you can pay attention to basic operations with arrays. The development of this game can be realized in the

following steps:

- Search for an element in an array.
- Using the above construction, a loop is organized to generate a four-digit number without repeating digits.
- Loop to check if there are identical elements in two arrays.
- Finally, this loop is placed in a cycle counting the attempts to know the number.

3.4. Functions. Definition. Call. Parameters.

The most important idea of structural programming is the idea of modularity. This means that the whole program must be divided into modules. In addition, the principles of design ("top-down") and step-by-step detailing must be considered. The algorithm for solving the problem is based on the mathematical model. No time should be saved from developing and studying the properties of this model. This will help to better understand the task and find the most natural way to solve it. The solution algorithm is carefully chosen. The data representation corresponding to the task is used.

✓ *HangMan* [6].

A word is given only with the number of letters. The goal is to recognize the word with a certain number of attempts. This is an example of teaching and training games.

When considering the topics of character arrays and functions, it is appropriate to solve this task in the following steps:

- Function to check if a character is in a string.
- Check if the letter has been entered using the above function.
- Check if the entered letter is part of the word and in what position using the above function.
- Attempts counting loop.

4 Conclusion

Of course, this is not about absolutizing such an approach, but only as further maintaining the interest of students in programming. In many of these game tasks, the mathematical knowledge and skills of the students can be seriously enriched. The problems in the introductory programming course are mainly due to factors such as lack of knowledge about the subject area and inability to think abstractly. However, when there is motivation, interest and proactivity in relation to some content, some of the barriers can be overcome.

Computer games have many fans among young people. This interest can be used to increase the effectiveness of initial programming training. The proposed few games, which from a programmer's point of view are not complicated, can be easily implemented by students. On the other hand, these games-programs can be improved in the direction of a more friendly interface. The games can be set as a project and then the students work independently, solving the problems themselves. This raises their self-esteem and prepares them for the real workplace. These tasks are solved after mastering the necessary tools.

The idea that is proposed is especially suitable for initial programming training. Students' interest in computer games can be used to see practical

applications of programming. The development of computer games during the training can motivate the students, arouse interest in new topics, facilitate the training and stimulate the memorization of what has already been learned. Games should not mislead students that learning is a game. That's why it's good to show them the other side, it's not easy to create an interesting game. There is a good opportunity for this in programming training.

Acknowledgements

This work is partially supported by the Scientific fund Fund "Scientific Research" at the University of Shumen, Bulgaria – Contract № RD-08-107/02.02.2021.

References

- [1] Martins, VF, de Almeida Souza Concilio, I, de Paiva Guimarães, M. Problem based learning associated to the development of games for programming teaching. *Comput Appl Eng Educ.* 2018; 26: 1577– 1589. <https://doi.org/10.1002/cae.21968>.
- [2] Nikolova, E., Tuparova, D., Teaching Informatics Through Game Programming - Pilot Study of Students Attitude, *Proceedings of the National Conference on "Education and Research in the Information Society"*, Plovdiv, June, 2018, 99-106pp.
- [3] Piteira, M., Costa, C., Learning computer programming. In *Proceedings of the 2013 International Conference on Information Systems and Design of Communication - ISDOC '13*, Lisboa, Portugal: ACM Press, (2013), 5–80, doi:10.1145/2503859.2503871.
- [4] Vahldick, A., Mendes, A. J., Marcelino, M. J. A review of games designed to improve introductory computer programming competencies, 2014 IEEE *Frontiers in Education Conference (FIE) Proceedings*, Madrid, (2014), 1-7.
- [5] Gnezdilova G. A., *Personalnii computer v igrakh i zadachah*, Moskva, 1988.
- [6] Klejman G. M., *Shkola budushego: komputeri v procese obuchenia*, Moskva, 1987.

Copyright © 2021 Galina Teodosieva and Teodosi Teodosiev. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.