# A Mobile Game to Teach AVL Trees

D. Šuníková*, Z.Kubincová* and M. Byrtus
* Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Bratislava, Slovakia
dana.sunikova@gmail.com
kubincova@fmph.uniba.sk
mirobyrtus@gmail.com

***Abstract*—Data structures and algorithms represent an important area of computer science. Sometimes, however, they are too abstract for students to understand them only from the lecture. Therefore we were looking for a way how to explain at least some of them to students.**

**In recent years, the use of digital games at all levels of education has come to the focus of researchers and educators. The advantage of this approach is, among others, learning in a natural and interesting way.**

**Since there are still very few good-quality educational games, we have decided to design, implement and test our own mobile game for teaching the algorithm of inserting new nodes to an AVL tree. In this paper, we describe designing of the game as a learning tool and introduce the first results from testing the game in a university course.**

## I. INTRODUCTION

Behind the idea of creating AVL Trees game was a desire to create one's own game available to the widest audience which would be different from the current games on the market, as well as the need to provide students with computer science concepts in a simple way.

Games are based on the fact that the user has to discover a pattern and learn it [1]. The game always takes place in a space which can be, for example, a map where characters move in an online game, or a grid, a chessboard, etc. As we wanted this game to be distinct from the others available on the market, we chose a different space - binary trees.

After reviewing the options, we have chosen AVL trees as a topic and designed a mobile logic game to work with this data structure. We made the choice to implement a mobile game because this kind of games seems to be the most accessible at the moment. The reason why this game is also logic will be explained later in the following sections. In this paper we also explain AVL tree concept, deal with educational games, describe our game design and methodology and report the first test results.

## II. AVL TREES

AVL tree [2] is a data structure that serves to store data that can be sorted. It supports binary search operation and all basic operations on this structure can be executed in a logarithmic time. It is a height-balanced binary search tree [2]. In addition to the basic criterion of the binary search tree (for each node N holds, that all values stored in its left sub-tree are less than the value stored in N, and all values in its right sub-tree are greater than the value in N), also the balance criterion applies in an AVL tree: for each node, the difference of the heights of its two sub-trees is at most one. This means that performing any operation changing the tree structure (inserting a new node, or deleting a node) may require the tree reconstruction (rebalancing) to preserve the balance criterion validity. The tree rebalancing is performed by rotating the nodes.

AVL tree is one of the structures that are taught in courses specialized in Algorithms and data structures in Computer Science programs. In order to better understand the correct functioning of operations performed on this structure as well as the reconstruction of the tree, students sometimes learn these operations and rotations by drawing them manually on the board, or on the paper. We have been thinking about how to motivate students to engage actively in such activities, as well as how to explain the operations on this structure in a fun way.

## III. EDUCATIONAL GAMES

In the recent years, the perception of computer games has changed significantly. More and more professionals recognize the potential of games in learning and developing different skills [3]. There are companies that have even employed people with the experience of online multiplayer games because the potential employees have gained high-quality leadership skills thanks to the game.

Also, a lot of research was done in the field of gamification in education in the past few years. Researchers try to align the use of game elements with existing educational and motivational theories. There are three main approaches to introducing computer games into classes [4]. In the first approach, pupils and students learn content while developing a game, mostly learning programming languages and training for problem solving. The second approach – to develop a computer game for teaching purposes – is resource-demanding and carries the risk that the game will not achieve both expected goals – it will not be entertaining or it will not teach the chosen subject well. The third approach – the use of commercial games for teaching – is difficult to prepare, as the teacher has to evaluate and analyze the game properly to find out what the game can teach as well as to recognize the parts of the subject that the game does not cover and to fill these parts by other suitably chosen activities.

In spite of the obvious obstacles to the introduction of digital games into the classroom teaching – which, apparently, stems in particular from the fact that there is still much to explore in this area – games can be considered educational tools. When playing a game, the player is working with the model lying in the background of the game until she masters it. For example, in fighting games the players profit less from precise shooting than from teamwork and tactics [1]. Games are effective in teaching for a number of reasons. Learning takes place in a meaningful context, a quick cycle of hypothesis formulation, testing, and revision facilitated by an instant feedback comes up during the interaction with the game, and others [4].

One of the first known motivational games was Zombies, Run! [5], in which a player runs while listening to a story put into a world where monsters are hunting her. The story and the music are set in a way which keeps the runner at the right speed at the right intervals.

Till now, very little research was conducted in gamification in education in our geographic area. There were several educational games for children and teenagers developed as bachelor or diploma theses in Slovak and Czech universities and certain studies proposing the use of gamification elements in university courses were published. However, there are projects that bring games (especially digital games) to the classroom teaching at all levels that can be found online. GalaxyCodr [6] is an educational game for teaching algorithmic thinking and basics of programming. Project Chymeros [7] aims to teach children and teenagers creativity, language literacy, reading skills and many more by following and participating in a tale of a young book character. It combines online and offline activities within a webpage, social network and several offline events. Otvorená hra [8] aims to teach adults communication skills through gamified online courses and several offline events. The VÍčatá.sk project [9] is also worth mentioning. It is an internet portal where articles on "meaningful games and technologies that deserve the attention of parents, teachers and all who meet young players" are regularly published. A lot of articles on computer games that can be used directly to teach a particular school subject, or to mediate complex emotions, or to guide parents how to use computer games to educate their children can be found here.

## IV. AVL Trees Game Design

We have decided to use the benefit of games – teaching skills and knowledge – and create a game for users who need to learn the AVL insertion algorithm. We wanted the game to be entertaining and playable, thus we designed it as a game that should entertain users with no interest in AVL trees as well.

As there is no widespread training of teachers regarding the use of digital games in the classroom, we have developed a methodology for using our game in a university course. We decided to let the teacher explain the algorithm and to link the formal knowledge to the game experience. To teach the complex theory within the game, we would either have to invent a very complex game or incorporate the amount of theoretical text in order to fully take up the role of the teacher.

Therefore, we have decided to create a game in which students either consolidate their knowledge after a theoretical teacher's explanation or discover new knowledge for themselves. Later in this paper we will describe the first experience, in which we used the game in a university course after the formal explanation to consolidate the student's knowledge through the drill. In the near future, we also plan to develop a lesson plan for high school Informatics class. We believe that the game can be used without the thorough explanation if the goal of the activity is just to meet such a data structure without learning much theory. We believe that both cases of use need a game with strong playability, so the students will be more interested and motivated to think more and to learn more.


(a) (b)

Figure 1. (a) Home screen, (b) Selection of game instances

We have decided to develop a logic game, because in this genre, we can let the user perform the selected algorithm – inserting elements into AVL trees – directly. For the game to be entertaining, users have to find their own solutions. However, for the game to be playable, we have to present them the basic rules of how the algorithm works. Thus we included a tutorial into our game, but users have to apply the rules from the tutorial by themselves, even in most difficult cases.

As the algorithm can be divided into few steps we introduced levels in the AVL Trees game. Each level corresponds to one algorithm step and has many game instances in which users solve a problem of the same type. Thus the player is learning a partial step of the algorithm by solving several game instances of the same level.

The goal of the first level is to learn how to insert a node with a given value to the right position in a binary search tree. In order to keep the AVL tree balanced all the time, we are letting the user insert only the nodes that will not make the tree unbalanced.

In the second level, the user has to mark the root of an unbalanced subtree. After the root of an unbalanced subtree is successfully marked, the application performs all needed rotations. However, the user has to insert the node first – to solve the problem from the first level type.

In the third level, the user inserts a node then marks all nodes that will rotate and then performs the rotation by dragging the new root of the previously unbalanced tree to its correct place.

In the first version of the game, users could not drag the node to its correct place. We added this mechanic because our first test users stated that it would be much more interesting, if they could actually perform the rotation instead of just marking nodes on which the action is performed.

To maximize both user motivation and learning outcomes, we have used different game elements according to best practices and available literature. Considering levels, they allow students to acquire new knowledge gradually, however, they serve as motivation as well. Each level is a partial goal and after fulfilling this goal, the feeling of success appears together with a desire to continue and to conquer the next level. Level difficulty usually increases [10].

Each game instance in all three levels starts with an empty tree and a list of nodes that the user needs to add to the tree. Each game instance has a different number of nodes that are supposed to be added to the tree and different rotations that have to be performed. The list of nodes is generated before the game instance starts, which means that the user can play games with the same characteristics (number of nodes and rotations) as many times as she needs to and she always has to add different nodes.

After performing a good action (correct insert), the user continues by inserting next node from the pre-generated list. After a bad action (incorrectly placed node or wrong rotation), the user gets feedback in the form of vibration, loses one life and has to repeat the action with the same node. There are 3 to 5 lives in each game (depending on the game instance difficulty). The user loses the game if she loses all lives in this game.

When deciding on the consequences of bad action, we also looked at the variants in which either the game correctly inserts the node for the user (and a life will be lost) or the user inserts a different node afterwards. We have chosen the current version because the possibility of repetition after the wrong action is an important game element, which makes the mistake a convenient way of exploring the possibility space and therefore learning [10]. Here, the user can insert the particular node correctly even after she could not do it at first and thus to earn a sense of success. In other words, the trial and error way of solving is embraced.

Apart from the aforementioned, the most apparent feedback in our game is that the node connects to the right place when inserted correctly and the whole tree „grows". After finishing the game, user can see a pop-up window with her points and stars, which are derived from the number of mistakes she has made. These stars are also displayed on the screen where users can choose the level and the game. To earn all the stars, each game has to be solved without mistakes.

Time pressure plays an important role in our game as well. It occurs here in two variations, it is a speed of the falling node – the time dedicated to locate the correct position of the node and it is also a time-bar, which indicates how much time remains for the user to determine the root of the unbalanced subtree or mark the nodes which are to be rotated. Only the rotation itself is not time-limited. Time pressure is important motivator in games [10].

## V. TESTING THE APPLICATION IN TEACHING

### A. Testing Sample

Our game was tested during one lab of the Algorithms and Data Structures course, which is part of the bachelor's study program for Informatics teachers. There were five students present. In the preceding lecture, AVL trees were explained (with two of the students present). In the lab, students were expected to practice insert and delete operations on AVL trees. Students were instructed to install our application on their personal mobile devices in advance.

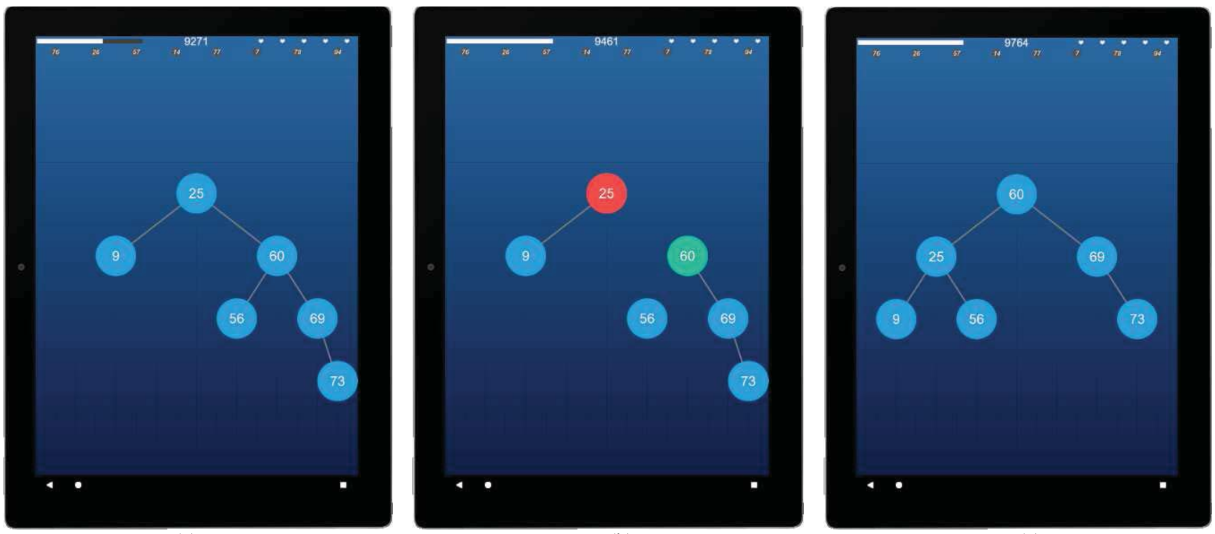
### B. Procedure

During the practice, the teacher shortly repeated the theory of AVL trees and how the insert operation works. Then she explained the partial operations in particular levels, after explanation of each level, students were playing the game, where they trained the respective operation. The teacher was walking around the class monitoring students' progress and explained whenever an ambiguity occurred. After all students understood the level, the next one was trained. At the end of the lesson, one exercise on the board was made, where students had to insert nodes into an AVL tree and delete some of them afterwards.

## VI. RESULTS

During the testing of our mobile game we served as a lecturer and as an observer as well. Therefore, we can make an assessment of the use of AVL Trees game from both points of view.

From the perspective of the teacher the use of the mobile game during the lab brought a more active engagement of students in the learning process. Students enjoyed the game and were willing to spend more time training the particular game levels compared to previous




(a) (b) (c)

Figure 2. (a) Unbalanced tree, (b) Marking of vertices for rotation, (c) Rotation

labs dedicated to this thematic unit. Previously, the students often did not solve given tasks by themselves, they rather looked over the neighbor's shoulder or just took a solution written on a blackboard by somebody else. This kind of behaviour was not forbidden during the labs (contrary to the exam), however it did not lead to the effective learning. When working with the mobile game this could not happen as every student was solving a task different from the tasks of other students. Nevertheless, all students worked and nobody gave up.

There were two types of interactions between students recorded while using the mobile game: two of them were friendly baiting each other a couple of times, and one student helped the other in another couple. Thus the fact that everyone worked on her own equipment and was immersed in the game did not mean that no interactions between the students that could be helpful to their learning were possible.

Despite the fact that only two out of five students who participated in the testing were present on the lesson where the AVL trees were explained, none of the students had significant problems in the step-by-step practice of inserting a node into the AVL tree using this mobile game. Students, who attended the lesson, were initially a bit faster, but after several repetitions of the required steps at particular levels, the other students understood the principles and managed the game as well.

In the anonymous questionnaire, which was answered by three students, it was confirmed that they liked the new form of learning, they were willing to work during the labs and they have learned and understood the topic.

About the game itself, we noticed a few minor, easy to fix shortcomings during the lab. We also received a few tips from the students in the questionnaire. In the future, we want to add player levels so that the user initially chooses whether she is a beginner, an advanced user or an expert, and hence will have the time set for the individual actions according to the chosen player level. We also plan to add a font size setting and contrast setting into the game to enable also users with small screen devices (or lightly visually impaired users) to enjoy the game.

## VII. Conclusions

In this paper, we presented the design of the mobile logic game AVL Trees, which was developed to attract the users, engage them, and to teach them the given data structure. We described the design choices we made with regard to the up to date literature from the fields of game-based learning and gamification. We also developed corresponding methodology that can be used in higher education.

We implemented the game and tested it in one lab of the Algorithms and data structures course, which is part of the bachelor’s study program for Informatics teachers. The students worked independently, engaged in the practice, and seemed to enjoy it. They all mastered the required curriculum.

## Acknowledgment

This research was supported from the Slovak national VEGA project no. 1/0797/18 as well as from the bilateral APVV project no. SK-TW-2017-0006.

## References

[1] R. Koster, *A theory of fun for game design.* O'Reilly Media, Inc., 2013.

[2] N. Wirth, *Algorithms+ Data Structures= Programs.* Prentice-Hall Series in Automatic Computation. Prentice Hall, 1976.

[3] J. McGonigal, *Reality is broken: Why games make us better and how they can change the world.* Penguin, 2011.

[4] R. Van Eck, “Digital Game-Based Learning: It's Not Just the Digital Natives Who Are Restless,” *EDUCAUSE review, 41(2), 2006, p.16.*

[5] Zombies, Run! https://zombiesrungame.com

[6] https://www.galaxycodr.com/sk

[7] https://chymeros.sk/

[8] https://www.otvorenahra.sk/

[9] Vĺčatá.sk. http://vlcata.dennikn.sk,

[10] K. M. Kapp, *The gamification of learning and instruction: game-based methods and strategies for training and education.* John Wiley & Sons, 2012.