Serious game in science education: How we can develop mathematical education

M. Hosťovecký*, I. Šalgovič*, Roderik Virágh**
* Department of the Applied Informatics, University of Ss. Cyril and Methodius, Trnava, Slovakia
** Department of Informatics, Slovak University of Agriculture in Nitra, Slovakia
marian.hostovecky@ucm.sk
ivan.slagovic@google.sk
roderik.viragh@gmail.com

***Abstract* — We see a growing interest in serious games in science education. Natural science education is the field where it is particularly suitable to implement educational games. Mathematics is one of those fields where it is useful to implement other than traditional forms of education in order to make the subject more attractive. Therefore, we decided to design and develop a computer game. The game aims to support the development of logical thinking in primary school pupils. The development of logical thinking in a pupil cannot be guaranteed (various aspects ranging from physiological and psychological to mental ones may be an obstacle). It is, however, highly desirable to keep creating suitable forms and look for ways of creating the development of logical thinking. Logical thinking in children should be formed from an early age. That's why the proposed game can be used from lower to higher grades of primary school. The game is still being developed, so its first levels are suited for the lower grades and its difficulty will be gradually increased.**

I. INTRODUCTION

Educational games are nowadays a subject of active research and their popularity in the computer game industry is constantly rising (Alvarez, 2008). In the field of educational games, most would credit Abt with inventing educational games for coining the term; however, considering the rich history of purposeful games, elements of educational games can be traced back to the work of Plato. It could be said that educational games are a modern manifestation of centuries-old theory and practices.

Serious games are not just developed and employed exclusively for entertainment purposes, but have been successfully incorporated as learning and training tools for a broad range of different areas [1, 2].

John Huizinga belongs to the authors who contributed to the development and promotion of the games in education. The game is characterized as "voluntary activity that is carried out within the fixed time and spatial boundaries, based on voluntarily accepted but absolutely binding rules which has its objective in itself and it is accompanied by feelings of tension and joy and knowledge of other than everyday life." [3] Miloš Zapletal, one of the most important Czech writers dealing with the phenomenon of games, described the game as: "active dynamic process employing a lesser or greater mental and physical abilities that are simultaneously trained and developed."[4] Definition of didactic games by T. Houška: "Activity in which the child spontaneously applies cognitive activities carried out under the primary influence of the relevant rules, which make the learning and teaching take place peacefully and in the second plan." [4]

Despite a relatively widespread belief in society that computer games are of no use whatsoever and are a so-called "waste of time", research has shown that playing computer games affects a wide spectrum of development of cognitive skills. Players who played action games, specifically FPS, have shown faster and more accurate allocation of attention and better spatial imagination [5]. One possible solution is to integrate problem-solving learning in form of digital computer games into education. Unlike traditional learning, where a teacher gives students just theoretical knowledge, in problem-solving learning a teacher gives students some problem tasks. They motivate pupils or students, direct the search for new ways of solving the tasks, through which pupils acquire new knowledge and new methods of operation. Students discover themselves through solving the problem. There is a team of investigators composed of two to five members. This kind of education supports thinking skills and students learn to apply theoretical knowledge in practice.

Problem-based learning as a method of instruction stands firm within the rationalist and, hence, is strongly influenced by cognitive psychology [6]. Another definition of problem-based teaching says: "teacher not convey pupils in the final form of knowledge, places for pupils tasks containing unfamiliar to them knowledge and methods of operation, motivates them, directs the search for ways and means to solve problems [7].

Research shows that students in problem-based curricula are indeed learning facts and concepts and the skills needed for critical problem solving and self-learning [6, 7, 8]. Other research shows that students participating in these kinds of learning activities are more motivated to learn, what they learn is more usable than the knowledge learned by students carrying out rote activities, and that they tend to acquire better higher-order thinking skills than students in other learning situations [8, 9, 10].

Problem-solving education should be one of the main points for developing this kind of learning and supporting analytical and critical skills. It is important to integrate game to learning process, but attention also needs to be paid to the type of game. A game should be effective and

useful for students. In last twenty years, integrating a game to learning has been rather untypical. However, specific needs need to be developed in pupils and students of every age.

II. SERIOUS GAME IN MATH

Serious games present themselves as one of the more interesting and also promising means of improving cognitive abilities, particularly in pupils [11]. Recently, research has consistently shown that several aspects in cognition such as visual short-memory, multitasking and spatial cognition can be enhanced by game play [12, 13, 14]

The body of academic literature on web-based games dedicated to increasing mathematical knowledge is still in its infancy compared with other well-developed research streams in education [15].

III. ATTRIBUTES OF THE PROPOSED GAME

The success of a game is something that cannot be guaranteed or bought. Nowadays, computer games and their design involve multiple aspects: from the story through interactivity and comprehensibility to UX design. All these parameters are important for a game to be successful. These properties will be briefly discussed and characterized below. The following ideas are based on the recommendations from game designers, game developers, etc.

Story – the storyline plays a crucial role for the success of a game. A great story can keep players immersed in the created virtual world. Players should feel attached to game characters and it is important to arouse curiosity in them as to how the story will develop.

Interactivity – a good story does not always equal to a challenging experience. Games should be interactive in their nature and a player should be constantly in contact with the story and game environment. New computer games containing elements of artificial intelligence are capable of adjusting to players, i.e. every player may experience a different storyline, different opponents, etc. However, developing such algorithms is incredibly complex and time-consuming.

Comprehensibility – it does not matter whether playing is complex or simple; a player needs to feel that they are doing what they are supposed to. Controlling has to be clear with a good feedback to the player. It is particularly important to understand the game. It is not recommended to constantly change the plot during the game. Such games may seem chaotic and may rather cause player's unwillingness to adapt.

Visuality – graphics is one of the most important aspects of every game. A balanced graphics affects a player like a magnet. It brings them joy, but also experience. The player likes to come back to play the game. Perception comes through the eyes - that's why it constitutes a relatively sensitive aspect. Every uneven balancing, exaggerated and sharp colours, asymmetry, imperfect composition may cause player's unwillingness to come back to play the game. The visual side of the game consists of the environment, characters, lighting, objects, etc. Graphics often belongs to the most complex game element in terms of computing and it is becoming increasingly realistic as hardware is developing.

Adequacy – a game needs to be a challenge for a player. The goal has to be achievable, but not too easily so as to keep the player interested. The end of the game should not be achievable quickly and easily. Players should have the feeling of success and achievement after fulfilling a task or accomplishing a mission.

A good example of a game fulfilling the aforementioned criteria is, for example, Grand Theft Auto 5 developed by Rockstar Games. The game is set in the modern world, within a fictional American city resembling Los Angeles. The story is told through 3 lead characters - Michael de Santa, Trevor Phillips and Franklin Clinton. Players can choose only one of the lead protagonists at a time and switch between them freely. There is also a possibility to customise their appearance and outfits. The plot of the game starts with a bank robbery in North Yankton, where the player gets familiar with two lead protagonists of the game. The game also has a multiplayer mode.

IV. DESIGN AND DEVELOPMENT OF THE GAME

A. Game proposal and target group

After analysing different game genres in the previous stage and discussion in the scholarly community, we decided to design a game with elements of adventure. On that basis, the goal of the game is as follows.

The game aims to support the development of logical thinking and cognitive skills in pupils of upper primary schools. In order to maximize the game experience, we decided that the game will be played from the first-person perspective. The development of logical thinking and cognitive skills in pupils will be supported by means of logical problems located at various places in a virtual world. To avoid monotonousness, a virtual world that will attract player's attention needs to be created. Attention is very important when developing an educational game. It may decide whether players will play a game or not, whether they will resume the game or not, etc. Therefore, various mechanisms to attract player's attention are applied in games, e.g. sophisticated graphics of the environment, characters, objects, etc.

The primary target age group of the proposed computer game are eighth- and ninth-graders of primary schools and high school students. The secondary target age group may be pupils of lower grades of primary schools, university students and anyone else interested in the development of logical thinking in both funny and educational way provided by a computer game.

B. Tools for development

The game development is not less important than the concept design and the selection of genre, strategy and game mechanism. Right at the beginning of the design, one has to think about the means that will be used for development. The selection of tools is made easier by the maximum amount of details characterized in the design possible. Of course, one of the key parameters in the selection of tools is price, which determines the suitability of tools to be used. That is one of the reasons why we selected open source tools when developing a game. They may or may not be always suitable.

Game engine

Game engine is one of the key factors in the development of a computer game. The selection of engine is determined by a number of key areas such as technical possibilities, target platforms and, of course, licensing. In our case, we preferred not only the range of possibilities and tools for the development of a game, but an important role was played by add-ons, controllability, working with the game engine, etc. Game engines enable one to develop a game for several operating systems at the same time. It means for developers that they can develop games for a much wider audience. However, unless the price is reasonable, it does not matter which platforms are supported by the engine. Based on that, we decided to use the Unreal Engine (UDK).

Programming languages

Just as the price of a game engine certainly may affect the decision, game engine technologies also play an important role in the process of selection and decision based on their suitability. The Unreal Engine 4 uses C ++ and Unity uses mainly C # or JavaScript. The Unreal Engine 4 also features visual scripting called Blueprint Visual Scripting („Blueprint"). This scripting language is based on so-called nodes. Programming takes place directly in the engine editor. Theoretically, not a single line of code needs to be ever written. This is great for a quick creation of prototypes and it is even possible to create the entire game using only blueprints. Of course, C++ is a more suitable choice for creating large and complex systems.

Optimization tools (Profiler)

One thing to remember is that the free version of Unity has no optimization tools (Profiler). These tools are included only in paid versions of Unity. Profiler enables to optimize a game, play a game and test the game performance and fluency in the process. These tools show percentage of time spent performing particular functions and modules, e.g. rendering and animation during the game. Therefore, it may be difficult to optimize a game in the free version of Unity.

Graphic tools

As far as graphics is concerned, we decided to work with freely available solutions under the GNU General Public License. Below, we describe the following:

Unreal Engine – in terms of graphics, the Unreal Engine 4 is really a top-quality game engine. It is capable of creating graphics on the same level as seen in games of other large studios, from particle simulation systems to advanced dynamic lighting. The Unreal Engine 4 can create any type of visual style, 2D or 3D. It really comes down to personal preference when deciding what program is more easily usable. Despite the fact that the developers of the Unreal Engine 4 put in a great effort in creating a user-friendly user interface, Unity is generally considered a more intuitive and more easily comprehensible game engine.

Blender –software with an open source code shared under the GNU General Public License. It is designed especially for modelling and 3D graphics rendering.

Gimp - GNU Image Manipulation Program is a free multiplatform computer application for editing and creating raster graphics. It is used mainly for photo editing and creation of web graphics. Gimp is available for free including source codes under the General Public License.

Krita – is an open-source editor for raster graphics designed especially for the purposes of creation of digital images and animations. It is available for operating systems Microsoft Windows, Linux and MacOS.

C. Game characteristics

Game genre – it would be difficult to clearly classify the game into one genre; however, it combines elements of logical, problem-solving and educational games. As there are no complex rules or any criteria for the assessment of how successful a problem has been solved in the game and as it does not contain any competitive elements either, it could be classified into the category of so-called casual games.

The concept of problems – problems in our computer game are based on solving logic circuits similar to those found in real computers. Logic circuits function based on the principles of Boolean algebra. The player's task is to set inputs into the circuit so that an imaginary current flows through the circuit to its output. After successfully solving a problem, the player may continue to the next level.

Game environment – The initial idea was to create a 2D game environment, where players would regulate a logic circuit by clicking on objects representing inputs into the circuit. An example of a similar game is a mobile game called Circuit Scramble. However, our desire to create an attractive and captivating game world made us drop the idea of a 2D virtual world and create a 3D world. One advantage/disadvantage of a 3D world in comparison to a 2D world in our present implementation that will be described in the following chapters is the fact that the player is forced to memorize the logic structure of a problem, which would most likely not be the case in a 2D game, as the player would see all parts of the logic structure of a problem at a time on the screen. When designing a 3D environment, graphic style of individual models is important. We decided for the style similar to the Ancient Roman architecture. These models are a part of the Sun Temple package available at the Unreal Engine Marketplace. Models with futuristic elements are also incorporated into the game environment.

Figure 1: Models from the Sun Temple package

Player character – a less visible, but not less important game element than the game environment is a player character. Controlling a player character has a great influence on the overall experience with a computer game. As already mentioned, the game is played from the first-person perspective. One of the advantages of such a perspective is the fact that as long as there are no mirrors in the game, the character does not need to have a 3D model. Another advantage is that when playing from the first-person perspective, the virtual world may have a more captivating effect on a player. They see objects more closely than they would with other game camera types. Additionally, it is simpler to implement a first-person perspective game into the virtual reality platform when opting for such an extension.

Controlling a player character – a player character is controlled using traditional keys used for moving vertically oriented characters. *The "W"* key moves a player character forward, *the "S"* key backwards, *the "A"* key to the left and the *"D"* key to the right. Pressing a spacebar prompts a character to jump. The player is also supposed to manipulate the objects on the scene with the physical model on using the left mouse button. Pressing the left mouse button makes the player lift the object and drop it after pressing the same button again.

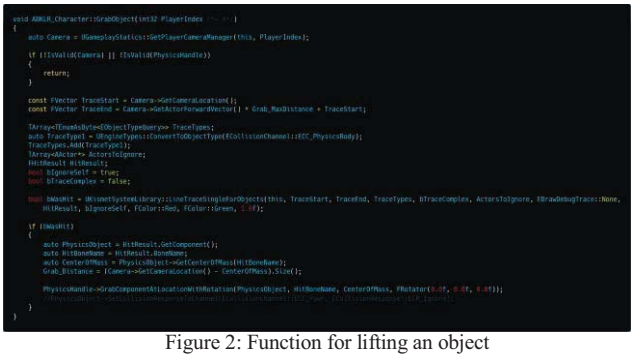

Figure 2: Function for lifting an object

Level design – the game consists of three levels with each corresponding to one of three basic levels of difficulty – easy, medium and hard.

- ✓ *Level 1* (easy) is called *Gardens* and serves as an introduction to the game. In this level, a player gets familiar with controlling a game character as well as with basic game mechanisms. *Gardens* is the only level set outdoors; other two levels are played indoors.

Figure 3: Level 1 as seen from above

- ✓ *Level 2* is called *Four Doorways*. Here, the player encounters more difficult problems that need to be solved to be able to continue to the next, i.e. the final level.

Figure 4: Preview of Level 2

- ✓ *Level 3* is entitled *The Big Hall.* This level includes the most difficult problem of the game. After successfully solving it, a door to a portal opens to the player. The game ends by entering the portal and a congratulatory screen about successfully finishing the game is shown.

Figure 5: The Big Hall preview

A logic circuit consists of three objects: the LogicGate, the Cable and the PowerGenerator, an input into a logic circuit. To ensure interaction between the player and the logic circuit, another class – the Cube – was created. The task of the class Door is to prevent the player from continuing to the next level without solving a given problem. Each class will be described individually in the following lines.

Cube – an object that can be manipulated by the player. Its only task is to enable the player to set inputs into the logic circuit. It occurs in several types. Its type is determined by the CubeType variable. The CubeType variable is the ECubeType enumeration consisting of six members: Red, Green, Blue, Cyan, Magenta and Yellow.

```
UENUM(BlueprintType)
enum class ECubeType : uint8
{
    CT_Red      UMETA(DisplayName = "Red"),
    CT_Blue     UMETA(DisplayName = "Blue"),
    CT_Green    UMETA(DisplayName = "Green"),
    CT_Cyan     UMETA(DisplayName = "Cyan"),
    CT_Magenta  UMETA(DisplayName = "Magenta"),
    CT_Yellow   UMETA(DisplayName = "Yellow")
};
```

Figure 6: ECubeType enumeration

LogicGate – represents a logic gate. It can have a random number of inputs and outputs; however, two inputs and one output occur most frequently. The game incorporates four basic logic gates: AND, OR, NOT and XOR.

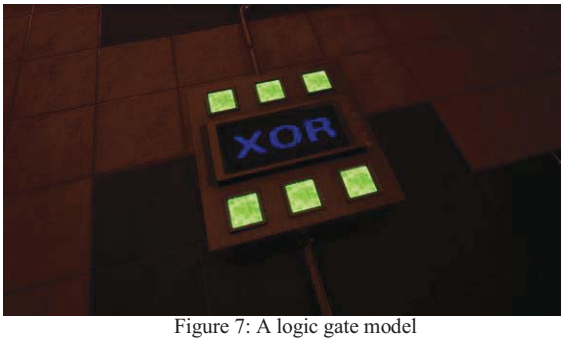

Figure 7: A logic gate model

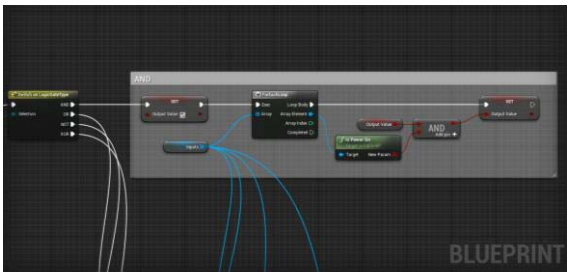

Figure 8: Setting the output value for the logic gate AND

PowerGenerator – is a class, the task of which, along with the class Cube, is to enable the player to set inputs into the logic circuit. It can have two states: on or off. If it comes to a collision of the generator with a cube of the same type, the generator state turns into on. As the state changes, a signal is sent to all outputs. State changes signalling takes place through delegates. The generator can have a random number of outputs. Outputs are made of indicators to objects of the Cable type. When the collision is resolved, the generator turns off. The type of the generator is represented by the ECubeType enumeration.

Door – represents the player's mission. When they have successfully solved a problem, a door opens and enables them to continue to the next level. It has only one input made of an indicator to an object of the Cable type.

Cable – a class enabling communication among other objects in the logic circuit.

V. TESTING

Iterative and *incremental* software development methodology was used when developing our computer game. Iterative and incremental software development is a software development method modelled around gradual adding of functionalities – increments. Iterative and incremental software development process is one of the agile software development methodologies.

Testing during the entire development period revealed many bugs and glitches. For example, there was a bug that occurred only after designing and developing the introductory level of the game. Before developing Level 1, the testing took place only in a testing level, where a player was located in a closed room. After creating an open space in the first level of the game, we discovered several difficulties of such a design. In the original version of the game, the player could throw a held object using the right mouse button. One of the objects that could be thrown was also the Cube. The player was thus enabled to throw away the cube to a location that could not be accessed, which resulted in the inability of the player to complete the level successfully. The first attempt to solve this bug was done by adding invisible walls that prevented the player from throwing the cube to inaccessible locations. However, this solution did not remove another bug related to the way the level was designed. The level is divided into individual rooms separated by a door. In each room, there is a simple problem – a logic circuit with one or two inputs. The bug consisted in the fact that the player could throw the cube from the room they currently found themselves in to the previous one in a way that the cube fell on the input of the logic circuit in the previous room and changed its output from the resolved state to unresolved, which caused the door between the previous and the current room to close and the player got stuck without the possibility to go back or continue. The only way out was to restart the level from the beginning. When dealing with this bug, we came up with two solutions. The first solution was to redesign the level in a way that would make such a situation impossible to happen. The second solution was to remove the functionality of throwing a cube. As the former solution is much more time-consuming, we decided for the latter. However, it did not remove another bug. By holding and moving one cube, the player was able to throw away another cube, which again resulted in all the difficulties described above. Due to lack of time, this problem remains unsolved.

VI. CONCLUSION

The theoretical part discussed the characteristics of computer games and various definitions of this term. We created a didactic computer game, which focuses on the development of logical thinking. The practical part involved creating a simple and funny game that meets the didactic aim of mastering the curriculum. The game is designed so that it can be further developed with added objects and settings. In the following months, the game will be tested at primary schools.

Additionally, there is a possibility of integrating a special kind of technical movement known as motion capture technology [17] which can improve movements and graphics of serious games. Besides, it would be great if the grid or cloud systems could be used to dynamically allocate resources during the game design in order to minimize total maintenance costs [18]. It will be very important to obtain some results about the impact of communication and opinions from pupils and teachers during the use of the game. Moreover, it will be important to find out what kind of role and tasks a teacher would have and how games could help pupils find employment [19, 20].

ACKNOWLEDGMENT

The paper was supported by the project KEGA 011UCM-4/2018. Thanks to Mr. Halás for language correction.

REFERENCES

[1] Breuer, J. - Bente, G. (2010) "Why So Serious? On the Relation of Serious Games and Learning". In: Eludamos. *Journal for Computer Game Culture*. (2010) Vol. 4, No. 1, 7-24

[2] Ritterfeld, Ute, Cody, Michael J.; Vorderer, Peter (2009) Serious Games: Mechanisms and effects. New York: Routledge.

[3] Huizinga, J. *Homo ludens: o původu kultury ve hře.* 2. vydanie. Praha: Dauphin, 2000. 297 s. ISBN 80-7272-020-1

[4] Zapletal, M. *Hry v klubovně.* 1. vyd. Praha: Olympia, 1986. 567 p. ISBN 27-053-86

[5] Green, C. S., Bavelier, D. 2012. *Learning, attentional control, and action video games*. [online][cit. 16.3.2018]. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3461277>

[6] Norman, G.T. – Schimdt, H.G. 1992. The psychological basis of problem-based learning: A review of the evidance. *Academic Medicine* 67, 557-65.

[7] Petlák, E. (1997) Všeobecná didaktika. In Bratislava: Iris. ISBN 80-88778-49-2

[8] Hmelo, C. E. (1995) Problem-based learning: Development of knowledge and reasoning strategies. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society.* Hillsdale NJ: Erlbaum.

[9] Blumenfeld, P., Soloway, E., Marx, R., Krajcik, J., Guzdial, M., & Palincsar A.(1991) Motivation Project-based Learning: Sustaining the Doing, Supporting the Learning. In> *Educational Psychologist*, vol. 26 (3&4), pp. 369-398.

[10] Kolodner, J. L. – Hmelo, C. E. – Narayanan, N. H. (1996) Problem-Based Learning Meets Case-Based Reasoning. Proceedings of the 1996 In: *International conference on Learning sciences. pp.*188-195

[11] Castellar, E. N. – All, A. – de Marez, L. – Looy, J. V. (2015) Cognitive abilities, digital games and arithmetic performance enhancement: A study comparing the effects of a math game and paper exercises. In: *Computers & Education*. 85 (2015), pp. 123-133

[12] Štubňa, J., 2016a. Selected Determinants Influencing on Student Motivation in Creating a Relationship Towards of Science. In: *Acta Humanica*. 13, (1) (2016), pp. 68-77

[13] Štubňa, J., 2016b. The importance of educational game as a part of teacher students preparation in the context of developing intersubject relationships in science subject. In: *Acta Humanica* 13 (2) 2016, pp. 39-45.

[14] Bavelier, D., Green, C. S., Pouget, A., & Schrater, P. (2012) Brain plasticity through the life span: learning to learn and action video games. In *S. E. Hyman* (Ed.), Annual review of neuroscienceVol. 35, pp. 391-416.

[15] Fellnhofer, K. 2016. All-in-one: impact study of an online math game for educational purposes. In: *International Journal of Technology Enhanced Learning*. Volume 8, Issue 1

[16] Alvarez, J. Michaud L. 2008, Serious games, Advergaming, edugaming, training and more [online]. Available at: <http://ja.games.free.fr/ludoscience/PDF/EtudeIDATE08_UK.pdf >

[17] Ölvecký, M., and Gabriška, D. Motion Capture as an Extension of Web-Based Simulation. In Applied Mechanics and Materials, 2014, Vol. 513, pp. 827-833.

[18] Šimon, M., Huraj, L., Siládi, V.: Analysis of Performance Bottleneck of P2P Grid Applications. In: Journal of Applied Mathematics, Statistics and Informatics. Volume 9. Issue 2, 2013. ISSN:1336-9180. (IET Inspec).

[19] Mišútová, M., Mišút, M. 2012: *Impact of ICT on the quality of mathematical education* 2012 IMSCI 2012 - 6th International Multi-Conference on Society, Cybernetics and Informatics, Proceedings, pp. 82-86.

[20] Mišút, M., Pribilová, K. 2013: *Communication impact on project oriented teaching in technology supported education*. In Lecture Notes in Electrical Engineering, 152 LNEE, pp. 559-567.