Ion ROCEANU
Editor

16th International Conference eLearning and Software for Education

eLearning sustainment for never-ending learning

Volume I

Editors

Ion ROCEANU

Lucian CIOLAN
Cătălin RADU **Ștefan TRĂUȘAN MATU** **Mireille RADOI**
Ioana Andreea STEFAN

eLearning sustainment for never-ending learning

Volume I

eLearning and Software for Education Conference

Bucharest, April 30 – May 01, 2020

Publisher:
Editura Universitară
Director: Vasile Muscalu
Bd. Bd. Nicolae Bălcescu no. 27-33
Bucharest
Phone: +4021-315.32.47

ISSN 2066 – 026X
ISSN-L 2066 – 026X
ISSN – CD 2343 - 7669

Further information on all eLearning and Software for Education – eLSE – events can be found at:
www.elseconference.eu

All the papers published in this book were verified for plagiarism using the Plagiat-Sistem Antiplagiat prin Internet software

eLSE 2020

The 16th International Scientific Conference
"eLearning and Software for Education"

eLearning sustainment for never-ending learning

Volume I

Conference Chairman:

Professor Dr. Ion **ROCEANU**, “Carol I” National Defence University, Bucharest, Romania

Scientific Committee:

Dr. Sae **SCHATZ**, Director of Advanced Distributed Learning (ADL) Initiative, USA
Prof. Dr. Lucian Ion **CIOLAN**, Vice rector, University of Bucharest, Romania
Prof. Dr. Monica **STANESCU**,Vice Rector National University of Physical Education and Sport of Bucharest, Romania
Prof. Dr. Neculai Eugen **SEGHEDIN**, Vice-Rector "Gheorghe Asachi" Technical University of Iasi, Romania
Prof. Dr. Virgil **POPESCU**, University of Craiova, Romania
Dr. Doina **BANCIU**, National Institute for Research & Development in Informatics Bucharest, Romania
Prof. Dr. Alin **MOLDOVEANU**, Politehnica" University Bucharest, Romania
Dr. Adrian **ADASCALITEI**, "Gheorghe Asachi" Technical University of Iasi, Romania
Dr. Diana **ANDONE**, "Politehnica" University of Timisoara, Romania
Dr. Mihail **ANTON**, "Carol I" National Defence University, Romania
Dr. Jannicke **BAALSRUD HAUGE**, Bremer Institut fur Produktion und Logistik (BIBA), Germany
Dr. Dragos **BARBIERU**, "Carol I" National Defence University, Romania
Dr. Daniel **BELIGAN**, "Carol I" National Defence University, Romania
Dr. Mirela **BLAGA**, "Gheorghe Asachi" Technical University of Iasi, Romania
Dr. Vasileios **BARKOUKIS,** Aristotle University of Thessaloniki, Greece
Dr. Teodora Daniela **CHICIOREANU**, "Politehnica" University Bucharest, Romania
Dr. Carmen Elena **CIRNU**, National Institute for Research & Development in Informatics Bucharest, Romania
Dr. Anca Cristina **COLIBABA**, "Grigore T.Popa" University of Medicine and Pharmacy of Iasi, Romania
Dr. Stefan **COLIBABA**, "Alexandru Ioan Cuza" University of Iasi, Romania
Dr. Calin **CORCIOVA**, "Grigore T.Popa" University of Medicine and Pharmacy of Iasi, Romania
Dr. Maria-Iuliana **DASCALU**, University POLITEHNICA of Bucharest, Romania
Dr. Sara **DE FREITAS**, Serious Games Institute, Coventry University Technology Park, United Kingdom
Dr. Christian **GLAHN**, Open University of the Netherlands, Netherland
Dr. Alessandro **GLORIA**, University of Genoa, Italy
Prof. Dr. Vasilica **GRIGORE**, National
University of Physical Education and Sport, Romania
Dr. Carmen **HOLOTESCU**, "Politehnica" University Timisoara, Romania
Dr. Phil **ICE**, American Public University System, West Virginia, USA
Dr. Knud **ILLERIS**, Aarhus University, Denmark
Dr. Anca Daniela **IONITA**, University POLITEHNICA of Bucharest, Romania
Dr. Olimpius **ISTRATE**, International Federation of Red Cross and Red Crescent Societies, Switzerland
Radu **JUGUREANU**, Siveco Romania, Romania
Dr. Theo **LIM**, Heriot-Watt University, UK
Dr. Bogdan **LOGOFATU**, University of Bucharest - Faculty of Psychology and Educational Sciences, Romania
Dr. Cristina **MIRON**, University of Bucharest, Romania
Dr. Laura **MURESAN**, Academy of Economic Studies Bucharest, Romania
Dr. Cristina **NICULESCU**, Research Institute for Artificial Intelligence "Mihai Draganescu", Romanian Academy, Romania
Dr. Michela **OTT**, Institute for Educational Technology, CNR, Italy
Dr. Valeriu **PATRICIU**, Military Technical Academy, Romania
Dr. Florin **POPENTIU -VLADICESCU**, Academy of Romanian Scientists, Romania
Dr. Dragos Marian **POPESCU**, University of Medicine and Pharmacy of Craiova, Romania
Dr. Catalin **RADU**, ADL Romania Association, Romania
Dr. Emanuel **SOARE**, University of Pitesti, Romania
Ioana Andreea **STEFAN**, ADL Romania Association, Romania
Dr. Veronica **STEFAN**, "Valahia" University of Targoviste, Romania
Dr. Katheryna **SYNYTSYA**, IRTC ITS, Ukraine
Dr. Ștefan **TRAUSAN-MATU**, "Politehnica" University Bucharest, Romania
Dr. **Danut TURCU**, "Carol I" National Defence University, Romania
Dr. Marius **TURNEA**, "Grigore T Popa" University of Iasi, Romania
Dr. Marin **VLADA**, University of Bucharest, Romania

CONTENTS

The 16th International Scientific Conference
eLearning and Software for Education
Bucharest, April 23-24, 2020
10.12753/2066-026X-20-010

USING INTERACTIVE WEB-BASED ANIMATIONS TO HELP STUDENTS TO FIND THE OPTIMAL ALGORITHMS OF RIVER CROSSING PUZZLES

Ladislav VÉGH, PhD.
Department of Mathematics and Informatics, J. Selye University, Bratislavská cesta 3322, 945 01 Komárno, Slovakia
veghl@ujs.sk

***Abstract**: To acquire algorithmic thinking is a long process that has a few steps. The most basic level of algorithmic thinking is when students recognize the algorithms and various problems that can be solved with algorithms. At the second level, students can execute the given algorithms. At the third level of algorithmic thinking, students can analyze the algorithms, they recognize which steps are executed in sequences, conditions or loops. At the fourth level, students can create their algorithms. The last three levels of algorithmic thinking are: the implementation of the algorithms in a programming language, modifying and improving the algorithms, and creating complex algorithms. In preliminary research related to algorithmic thinking, we investigated how first-year undergraduate computer science students of J. Selye University can solve problems associated with the second, third and fourth level of algorithmic thinking. We chose these levels because these levels do not require to know any programming language. The tasks that students had to solve were for example: what will be the route of a robot when it executes the given instructions, how many times we need to cross a river to carry everyone to another river-bank. To solve these types of tasks requires only good algorithmic thinking. The results showed that students reached 81.4% average score on tasks related to the execution of given algorithms, 72.3% average score on tasks where they needed to analyze algorithms, and 66.2% average score on tasks where students needed to create algorithms. The latter type of tasks were mostly various river-crossing problems. Even though, that students reached a 66.2% average score on these tasks, if we had accepted only solutions with the optimal algorithms (minimal number of river crossing), they would have reached only a 21.3% average score, which is very low. To help students find the optimal algorithms of river crossing puzzles, we developed several interactive web-based animations. In the last part of this paper, we describe these animations, we summarize how they were created and how they can be used in education. Finally, we conclude and briefly mention our plans related to our future research.*

Keywords: *Algorithmic thinking, river crossing puzzles, interactive web-based animations.*

I. INTRODUCTION

One of the main subjects of first- and second-year undergraduate computer science students at J. Selye University is programming. Students learn programming in C language during the first two semesters of their study. Later, during the third and fourth semester, they learn object-oriented programming in C# programming language. To gain adequate programming skills, students need to learn the basic data types and structures, control structures, and the syntax of the chosen programming language. Except for these topics, the most important skill that students need to acquire is the algorithmic thinking. In the next chapter of this paper, we briefly summarize what algorithmic thinking is. Next, we show our preliminary research related to algorithmic and logical thinking. Finally, we describe some interactive web-based animations that we developed to help students to find the optimal algorithms of various river-crossing problems.

II. ALGORITHMIC THINKING

What are an algorithm and algorithmic thinking? An algorithm is a method to solve a problem that consists of exactly defined instructions [2]. Futschek defined algorithmic thinking as a pool of abilities that are connected to constructing and understanding algorithms. Among these abilities belong [2]:

- the ability to specify a problem precisely,
- the ability to find the basic actions that are adequate to the given problem,
- the ability to construct a correct algorithm to a given problem using the basic actions,
- the ability to think about all possible special and normal cases of a problem,
- the ability to improve the efficiency of an algorithm.

Ferrari at al. defined algorithmic thinking briefly as a method for solving problems based on the clear definition of the steps [1]. Moran at al. gave a similar definition of algorithmic thinking. According to them, algorithmic thinking is solving a problem in a logical, repeatable, step-by-step manner [3].

The algorithmic thinking has different levels, which are usually acquired by students sequentially [7]:

1. recognizing algorithms and various problems that can be solved with algorithms,
2. executing given algorithms,
3. analyzing algorithms, recognizing which steps are executed in sequences, conditions or loops,
4. creating algorithms,
5. implementing algorithms in programming languages,
6. modifying and improving algorithms,
7. developing complex algorithms.

Mastering the advanced levels is possible only with a lot of practice, solving tasks from the simple to more difficult ones [5, 7].

III. PRELIMINARY RESEARCH RELATED TO ALGORITHMIC THINKING OF FIRST-YEAR UNDERGRADUATE COMPUTER SCIENCE STUDENTS

A research related to algorithmic and logical thinking was conducted at J. Selye University at the beginning of the academic year 2019/2020. In the assessment were involved 96 first-year computer science students (75 students of applied informatics and 21 students of computer science education). Students had 60 minutes to solve the test containing 16 assignments. Among assignments were tasks related to execution of given algorithms (three robots, hexagon, drawing crosses, ladybug robot, airplane, colorful walls, maze), tasks related to analysis of algorithms (queuing, colorful blankets, water lilies), tasks related to creation of optimal algorithms (river crossing, weighing, prisoners), and few tasks related to problem solving and logical thinking (social site, knights and scoundrels, decoding) [6].

As an example, in Figure no 1. we can see an assignment (ladybug robot) related to the execution of algorithms. Students' task was to figure out which sequence of instructions should be used if we want the ladybug robot to reach the goal (red flag). The robot can go forward by one square (green forward arrow), turn right by 90 degrees (blue right arrow) or turn left by 90 degrees (blue left arrow). It can also repeat some instructions given times (yellow rectangle).

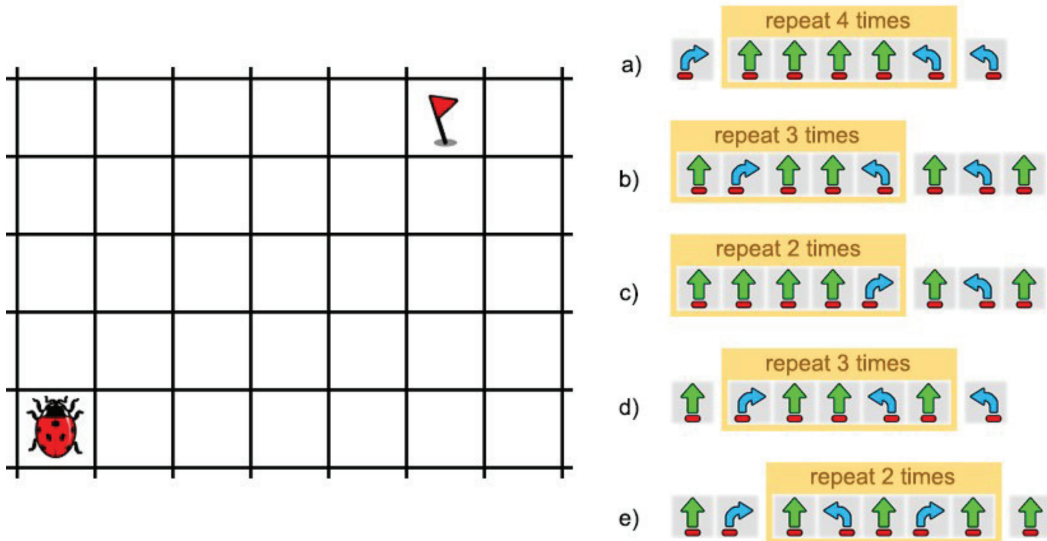

Figure no. 1. An assignment (ladybug robot) related to the execution of algorithms

The average scores of assignments of the test are shown in Figure no. 2. We can see in this graph that students solved some tasks with very high scores, but there were also tasks where students got very low scores. We examined in more detail what type of tasks were hard to solve for our students.

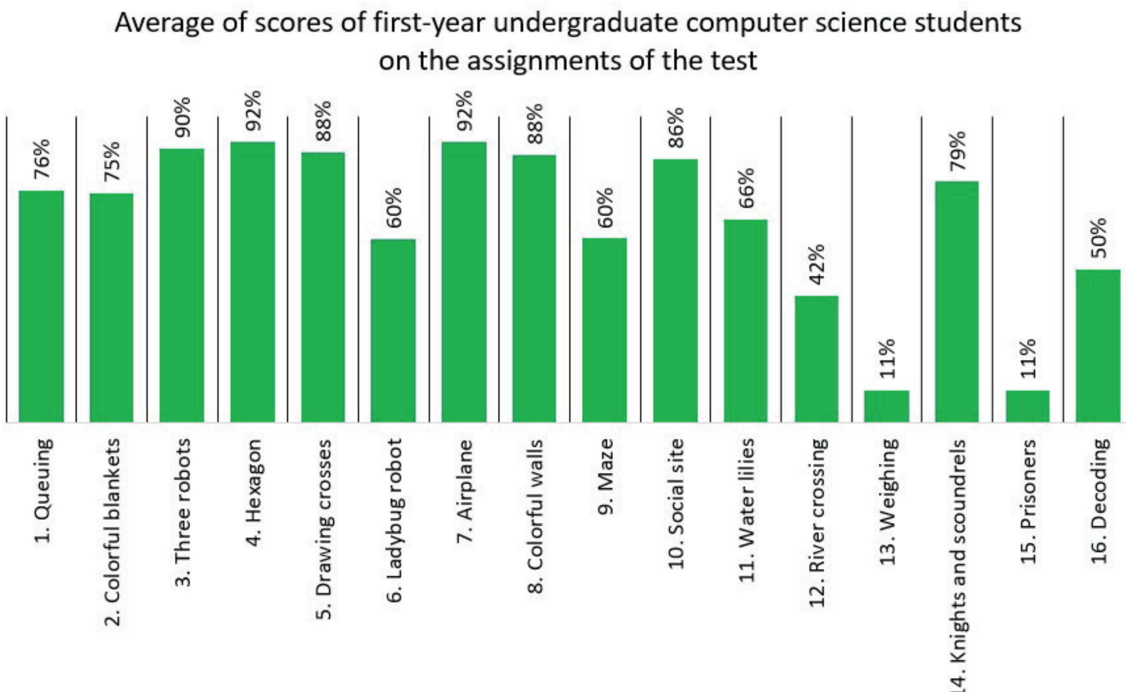

Figure no. 2. Average of scores of first-year undergraduate computer science students on the assignments of the test

Figure no. 3 shows the average scores of students on the different types of tasks. We can see in this chart that students reached the highest scores (81.4%) on tasks that were related to the execution of given algorithms (second level of algorithmic thinking). On tasks related to the analysis of algorithms (third level of algorithmic thinking), students reached 72.3% of scores on average. Tasks related to the creation of optimal algorithms seemed to be the hardest ones, students on these tasks reached only 21.3% of scores on average. This type of task contained mostly assignments related to river-crossing problems, where students had to figure out an algorithm with the minimum number of the river crossing. We need to mention that this type of task is not simply the fourth level of algorithmic thinking (creation of algorithms), because instead of creating algorithms students needed to find the optimal algorithms (with the minimum number of the river crossing). If we were accepted any correct algorithm (not only the optimal algorithms), the average score would be 66.2% instead of 21.3%.

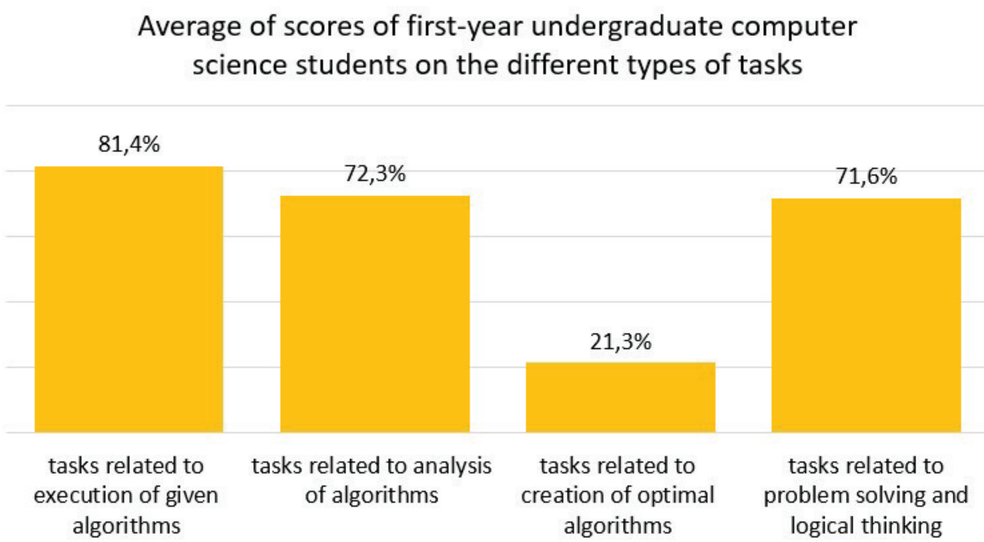

Figure no. 3. Average of scores of first-year undergraduate computer science students on the different types of assignments

The graph also shows that students reached 71.6% of scores on average on the tasks related to problem solving and logical thinking.

IV. INTERACTIVE WEB-BASED ANIMATIONS OF RIVER CROSSING PUZZLES

A river crossing puzzle is a type of puzzle in which the goal is to carry items from one river bank to another, usually in the minimum number of trips. The difficulty of the puzzle may arise from restrictions on which or how many items can be transported at the same time, or which or how many items may be safely left together [4].

To help students better understand and create optimal algorithms of various river-crossing problems, we created three interactive web-based animations. To develop these animations we used HTML5, CSS3, JavasScript and Konva JS library. The design and usage are the same for all three animations. Students can click on the items to add them to the boat or remove them from the boat. By clicking on the boat, if there is someone in the boat who can paddle, the boat will cross the river. The number of river crossings is shown at the bottom of the animations. All of three interactive web-based animations are publicly available at http://anim.ide.sk/crossing.php.

4.1. The wolf, goat, cabbage and the girl

The first web-based application that we created is the interactive animation of the puzzle called “The wolf, goat, cabbage and the girl” (see Figure no. 4), in which the girl returns from the market, where she bought the wolf, the goat, and the cabbage. On the way home, she must cross a river. The goal of the puzzle is to figure out how the girl can get the wolf, the goat and the cabbage on the other side of the river. However, there are some restrictions. Only the girl can row the boat. The boat is small and won't fit more than the girl and one of her purchases in it. She cannot leave the goat alone with the cabbage (because the goat would eat it), nor she can leave the goat alone with the wolf (because the goat would be eaten).

Figure no. 4. Interactive animation of "The wolf, goat, cabbage and the girl" puzzle

This problem is the easiest one from all three river-crossing puzzles, which can be solved with 7 one-way trips. Even though the interactive animation does not restrict to solve the puzzle with the minimum number of river crossings, we encourage our students to find the optimal algorithm with the minimum number of trips.

4.2. The soldiers and the children

The second web-based application that we created for students is the interactive animation of "The soldiers and the children" puzzle (see Figure no. 5). In this animation, there are three soldiers and two children who want to cross the river. The boat is small, it can carry only one soldier or a maximum of two children. The goal of the puzzle is to find out how the soldiers and the children can get to the other side of the river.

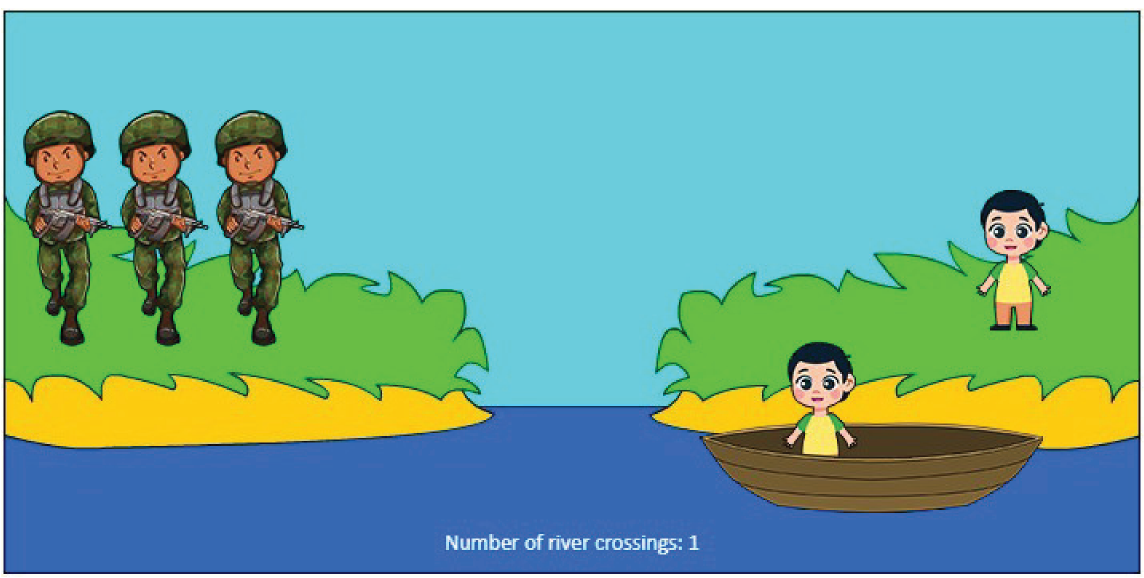

Figure no. 5. Interactive animation of "The soldiers and the children" puzzle

To solve this problem, first, students need to figure out, how 1 soldier can get to the other side of the river (this can be done with 4 one-way trips). Next, they need to repeat it for every soldier (3 times 4 one-way trips), and finally, the children must get to the other side of the river (1 one-way trip). To solve this puzzle with the minimum number of trips, 13 one-way trips are needed altogether. Similarly to the previous interactive animation, this animation also does not restrict to solve the puzzle with the minimum number of river crossings, but we encourage our students to figure out the optimal solution.

4.3. The monsters and the children

The third web-based application is the interactive animation of "The monsters and the children" puzzle (see Figure no. 6). In this puzzle, three monsters and three children want to get to the other side of a river. There is a small boat, which can fit only two of them. To prevent a tragedy, there can never be more monsters than children together (otherwise the monsters would eat the children). How can the monsters and the children get to the other side of the river?

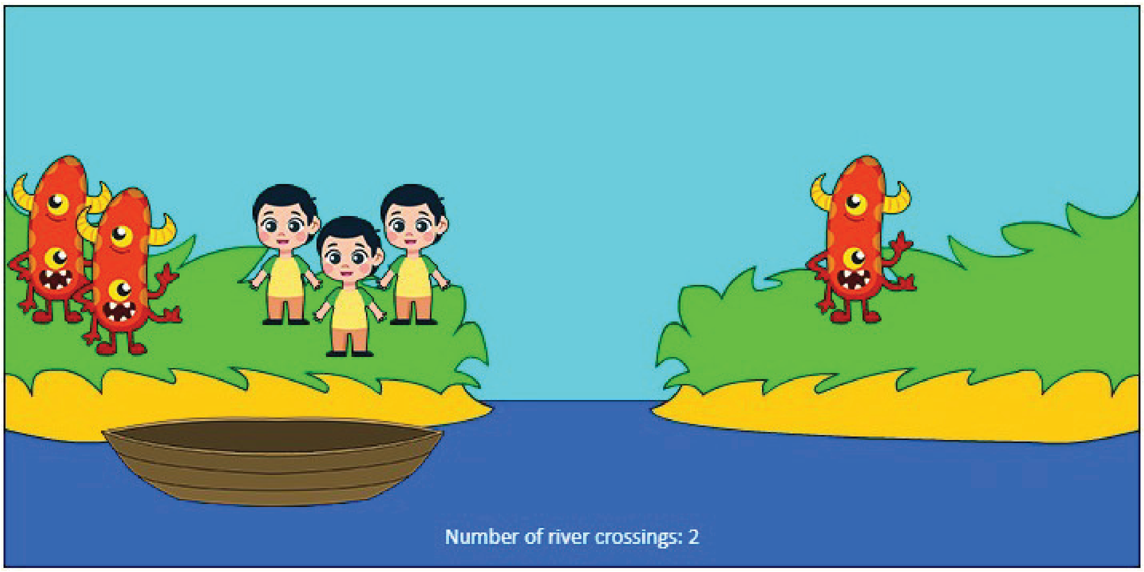

Figure no. 6. Interactive animation of "The monsters and the children" puzzle

This is probably the hardest puzzle of all three river-crossing challenges. The problem can be solved with 11 one-way trips. Similarly to the previous interactive animations, this animation also does not restrict to solve the puzzle with the minimum number of river crossings, but we encourage our students to find the optimal algorithm for this problem, as well.

V. CONCLUSIONS

In this paper, we briefly mentioned what is algorithmic thinking, which is a required skill to master computer programming. Next, we showed the results of our preliminary research related to algorithmic and logical thinking. Because the results showed that the hardest tasks for students were to create optimal algorithms of problems similar to river-crossing puzzles, we developed three interactive web-based animations. We believe, these animations can help students to create algorithms with the minimum number of trips for the mentioned puzzles. In the future we would like to conduct a pedagogical experiment using these animations, to observe to what extent our animations can help students to develop the optimal algorithms of river-crossing problems.

Acknowledgments

The paper was supported by projects KEGA 012TTU-4/2018 "Interactive animation and simulation models in education" and VEGA 1/0663/19 "Analysis of natural and mathematical education at high schools and innovation of the content of subject methodology".

Reference Text and Citations

[1] Ferrari, Alberto; Poggi, Agostino; Tomaiuolo, Michele; Computational Thinking in Innovative Computational Environments and Coding, *Encyclopedia of Information Science and Technology*, 2018.

[2] Futschek, Gerald; Algorithmic Thinking: The Key for Understanding Computer Science, *ISSEP 2006, Informatics Education – The Bridge between Using and Understanding Computers, Lecture Notes in Computer Science*, vol. 4226/2006, Springer-Verlag, Berlin, Heidelberg, pp. 159-168.

[3] Moran, Renee; Roberston, Laura; Tai, Chihche; Keith, Karin J.; Price, Jamie; Meier, Lori T.; Hong, Huili; Preparing Pre-Service Teachers for the Future: Computational Thinking as a Scaffold for Critical Thinking, *Handbook of Research on Integrating Computer Science and Computational Thinking in K-12 Education*, 2020.

[4] River crossing puzzle, *Wikipedia*, https://en.wikipedia.org/wiki/River_crossing_puzzle

[5] Stoffová, Veronika; Educational computer games in programming teaching and learning, *eLearning and Software for Education (eLSE)*, Carol I National Defence University Publishing House, Bucharest, Romania, 2019. pp. 39-45.

[6] Végh, Ladislav; Gubo, Štefan; Takáč, Ondrej; Az elsős informatika szakos hallgatók programozói tapasztalatainak és informatikai gondolkodásuknak felmérése, *SJE Nemzetközi Tudományos Konferencia*, J. Selye University, Komárno, Slovakia, 2019.

[7] Zsakó, László; Szlávi, Péter; Informatikai kompetenciák: Algoritmikus gondolkodás. *InfoDidakt 2010*. Szombathely, Hungary, 2010, https://people.inf.elte.hu/szlavi/InfoDidact10/Manuscripts/ZsL_SzP.pdf