Available online at www.sciencedirect.com

ScienceDirect

Procedia - Social and Behavioral Sciences 116 (2014) 2962 – 2966

Procedia
Social and Behavioral Sciences

5th World Conference on Educational Sciences - WCES 2013

# Teaching sorting and searching algorithms through simulation-based learning objects in an introductory programming course

Georgi Tuparov [a, b] , Daniela Tuparova [a,] *Vladimir Jordanov [a]

[a]*South-Western University "Neofit Rilski", 66 Ivan Michailov Str., 2700 Blagoevgrad, Bulgaria*
[b] *Institute of Mathematics and Informatics, Bulgarian Academy of Science, Acad. Georgi Bonchev Str., Block 8, 1113 Sofia, Bulgaria*

**Abstract**

In this paper we present our experience in the development and use of interactive simulation-based Learning Objects (LOs) in an introductory course of programming. The focus of our research is on the teaching of one of the important topics in introductory courses of programming - Sorting algorithms and their programming implementation. The characteristics of the LOs developed and the scenarios used for their deployment are described. The results from the pilot study, discussed in the paper, demonstrate an increase of student interest and a level of understanding of the learning content.



*Keywords:* teaching programming, sorting algorithms; computer based simulation, vizualisation, learning objects;

## Introduction

Many educators and researchers in the area of computer science teaching methods argue that some of the freshman students in Computer Science have problems with understanding abstract concepts in programming – (Areias C. at all, 2007; Jenkins T, 2002; Lahtinen E., 2006; Roeßling G., 2010). "Often initial programming courses have high present dropout rates." (Areias C. at all, 2007).

The problems arise due to several reasons: in some universities classes are very big – 80-100 students and more, and the lecturer has no possibility to pay attention to every student. Students have different attitudes and motivation to study programming. They have different learning styles (Jenkins T, 2002) and previous knowledge about computing, algorithms, etc. The content in an introductory course in programming is too abstract and in some cases is "not understandable" by the students (Tuparov G.,at all 2012). Lecturers apply different pedagogical approaches to support students' achievements and to enforce effective knowledge mastery, to increase motivation of the students to continue their studies in the area of informatics (computer science), and to engage students in learning

* Correspondign Author: Daniela Tuparova. Tel.: ++359 898 441 341; fax: ++359 73 88 55 16
*E-mail address*: ddureva@swu.bg


doi:10.1016/j.sbspro.2014.01.688

activities. These approaches in most cases are based on the technology-enhanced learning by animation and simulation tools. The efforts of educators and researchers are presented in many of papers and software tools.

In this paper we present our experience in the development and use of interactive simulation-based Learning Objects (LOs) in an introductory course of programming. We focus our research on the one of the important topics in introductory course of programming - Sorting algorithms and their programming implementation.

# 1. Methods

## *1.1. Related Studies*

Most of the studies and projects regarding teaching a programming course to novices and experienced students are grounded on different types of visualizations, animations and simulations of algorithms and with the use of diverse programming languages. For example: ALGAE (Zeil S. J., 2011), allows C++ or Java code to be produced an "animated" version of that code.; TRAKLA 2 is a learning environment that supports algorithm animation and simulation tasks where the user directly manipulates data structures through a graphical user interface" (Korhonen A, at all 2003), H-SICAS (Marcelino M., at all, 2008) performs a handheld algorithm animation and simulation tool; Tutoring System for programming (Krushkov H. at all, 2009) – includes tasks with visualization of algorithms and interactive tests for the evaluation of students' knowledge; ProGuide (Areias C. M., at all, 2007) and ProLearn (Mendes A. J., 2006) are tools for the simulation of algorithms with flowcharts and code visualization.

In the framework of the project Codewitz (Project Codewitz, 2006) many LOs were developed. The demonstration LOs are Flash based. The student has to solve problems in a simulation environment and can input values for some variables and obtain feedback about a proposed solution of the problem. In Grozdev S. & T. Terzieva, (2011) authors discuss advantages of using the visualization of sorting algorithms in initial education in programming. Web sites developed by David R. Martin, (2007), Duane J. Jarc, (1998), Morris J, (1998) perform a diversity of visualizations through animations and simulations of sorting algorithms. There are various tools and environments for the development of simulation-based visualization of algorithms. The systems, tools and LOs discussed above are developed by diverse software environments for programming or authoring tools: Visual Basic, Java, Flash, C++, Delphi etc.

## *1.2. Requirements to simulation LOs*

The basic characteristics of most of the above mentioned visualizations of sorting algorithms contain animation of bars that represent values of the array's elements. Before preparing our interactive simulations we set a number of requirements to their functionality.

The basic requirements to the array's sorting LOs are as follows: providing easy understanding of the learning content; allowing manual and automatic management of visualization; motivating students to study actively an introductory course in programming"; visualizing an array's elements in the form of colored boxes with relevant values; visualizing indexes of an array's elements; visualizing programming code with changes of variable values; giving the user freedom to choose a number of elements in array; using examples or inputting values of array elements;

The additional requirements have been set regarding the localization of the LOs interface (at this moment. in Bulgarian and English languages), performance of programming code in different programming languages, SCORM compatibility, and the possibility to use LOs by the mobile devices.

## *1.3. 2.3. Functionalities of simulation-based LOs for learning sorting algorithms.*

The proposed LOs respond to the requirements mentioned above. The user interface of each LO is divided into following blocks: Title of the sorting method; Setting of the examples; Elements input; Settings for the way of execution – automatic or manual; Sequence visualizations of the algorithm and visualizations of programming code.

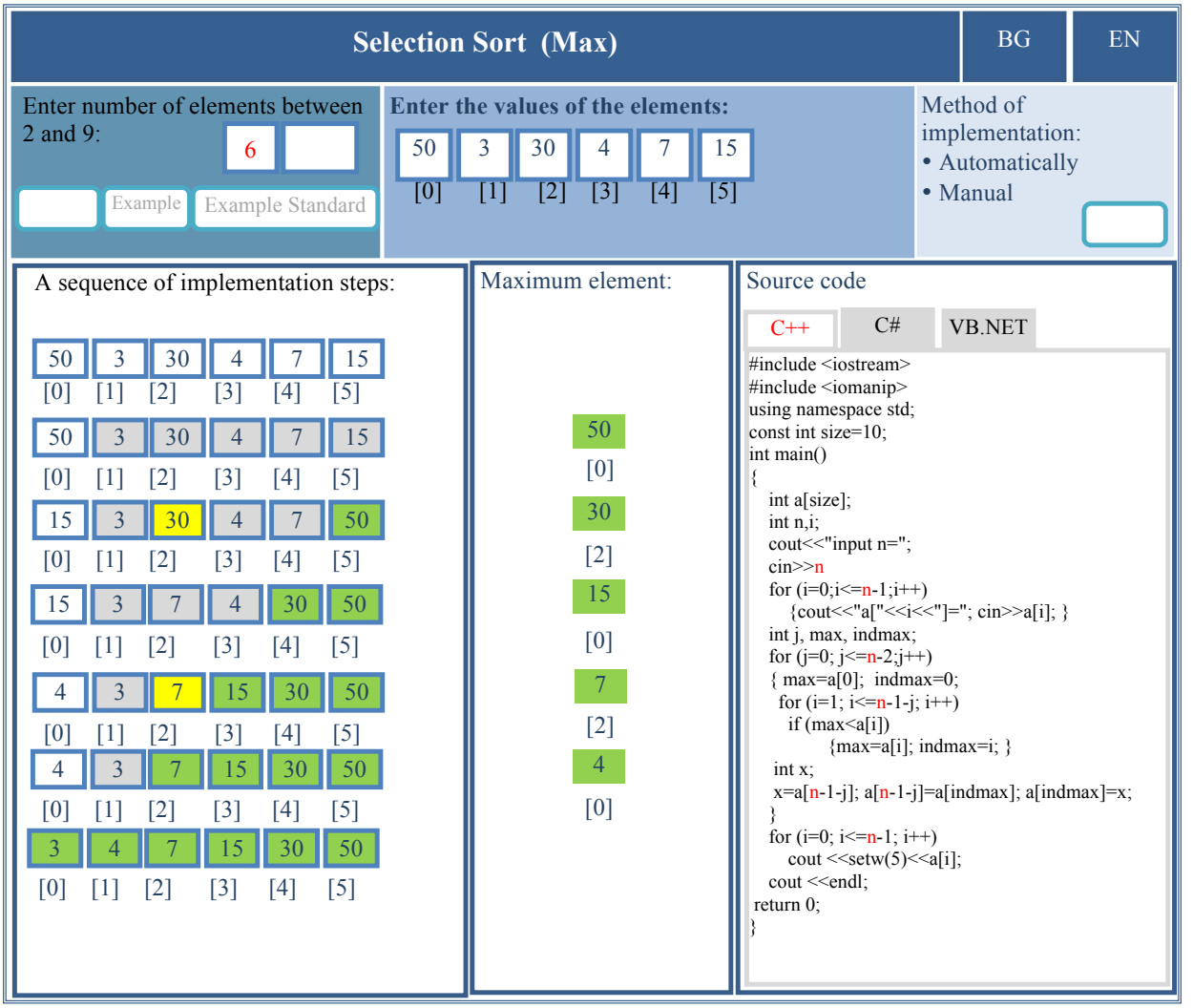


Figure 1. Schema of the interface of Selection Method

The sequence visualization block represents the array's elements as boxes; the relevant index of the element is mentioned under each box.

The sorted elements in each algorithm are colored in green. The elements with values that have to be exchanged are colored in yellow. The LOs for Selection sort methods consist additional visualization part with information about the current maximum (minimum) element. The source code block allows the user to change the programming language and to observe the current variable values. The variables with values that have to be exchanged during the simulation are colored in red. The basic programming languages supported in the LOs are C++, C#, VB.Net.

For the development of LOs, we have chosen Java Script and Jason. This approach allows the LOs to be used by different mobile devices and with different languages.

### *1.4. Pilot study of pedagogical efficiency of the proposed Los*

A pilot study was conducted in the course "Introduction to Programming" with freshman students in the specialty "Informatics" and "Pedagogy of Teaching Mathematics and Informatics". The course is divided in lectures, seminars and lab exercises. About 80 students were involved in the course. Even though those students have studied

the subject "Informatics" in high school, most of them have demonstrated a low level of basic knowledge about algorithms. Therefore we accepted that in the course "Introduction to Programming" we will start with the basic concepts in algorithms and programming languages. In this course we use C++ as a programming language for the description of algorithms. Before they started studying the topic of Sorting Algorithms, students have had experience with the use of simulation based LOs. The results of this study are presented in Tuparova D., at all, (2012).

The Sorting Algorithms topic covers the most popular algorithms: Bubble Sort, Bubble Sort with improvement, Selection Sort, Merge sort. Also two search algorithms are performed – Ordinary Search of elements and Binary Search. The proposed LOs were delivered through the e-Learning environment Moodle for self-learning and were used with different learning scenarios during the face to face lectures.

Scenario 1. The lecturer explains the basic steps of the algorithm, demonstrates the algorithm by the LO and focus students to the sequence visualization part. At the end, the lecturer stresses the programming code.

Scenario 2. The lecturer demonstrates the simulation with different input data and sets the task "Explain the steps in the proposed algorithm." Students observe the behavior of the array's elements with different data. During the discussion, the students and the lecturer summarize the content of the algorithm. The lecturer introduces the programming code of the algorithm and discusses basic points in the code with the students.

## 2. Findings and Results

Our observations during the performance and use of simulation-based LOs in the framework of classes showed that students were impressed by the simulations. The simulations gained the students' attention. Even the students with low interest to the lectures started following the simulations and participate in the discussions.

For the evaluation of some issues of pedagogical and technical usability of proposed the LOs, we conducted surveys with the students from the course "Introduction to Programming". The terms pedagogical and technical usability of multimedia learning materials and e-learning environments are discussed in the papers of Brodahl C., & B. Smestad, (2009), Hadjerrouit S.,(2010), Nokelainen, P. (2006), TEOH Belinda Soo-Phing & Tse-Kian NEO, (2007), and include basic constructs such as Understandability and Motivation.

We proposed an anonymous survey for the evaluation of all simulation-based LOs. The survey consists of 17 items. In the first item we ask students how often they used the simulation-based learning material. The items 2-15 are presented in a Like Scale with responses "Not applicable"(1), "Definitely No" (2), "No" (3), "Yes"(4), "Definitely Yes"(5). Item 16 is directed to the self-evaluation of knowledge about the syntax of C++ language mastered before using the LOs. The last item is an open one. The students have been asked to make suggestions for the improvement of the LOs.

The items in the surveys were divided into three groups regarding pedagogical usability criteria– Understandability (nine indicators) and Motivation (three indicators) and regarding technical usability criteria– Graphical layout (two indicators). We obtained 34 responses. The Cranach's alpha coefficient of reliability of the survey is 0,94. The students' answers indicated that learning content is presented in an understandable way in learning objects – the boundaries of the percentages of the students with answers "Yes" and "Definitely Yes" for the indicators in the Understandability criteria are from 79% to 94%. The Cranach's Alpha coefficients for the category Understandability are 0,89. The LOs motivated students to participate actively in the educational process. More than 90% of students declare that the use of LOs increased their interest to the course "Introduction to Programming". Cranach's Alpha for Motivation criteria is 0,75. The students are satisfied with the Graphical Layout of the LOs. Main suggestions given in the item 17 are directed to the development of similar multimedia LOs for other topics of the course.

## 3. Conclusions

We present in brief our experience with the use of interactive simulations in teaching sorting and searching algorithms. The results encouraged us to develop new LOs for learning abstract content of programming and

algorithms. Our future work will be directed to study relations between students' achievements, their learning styles and attitude towards using interactive simulation in introductory programming course.

**Acknowledgements**

The paper is published with support of the project „Science and Business", BG051PO001/3.3-05-0001, Operational Programmed "Human Resources Development", funded by European Social Fund

## References

Albreht D. (2007) Animations for Data Structures and Algorithms, http://www.csse.monash.edu.au/~dwa/Animations/index.html,

Areias C. M., A. J. Mendes & A. J. Gomes,(2007) Learning to Program with ProGuide, In International Conference on Engineering Education – ICEE 2007, Coimbra, Portugal

Brodahl C., & B. Smestad, (2009) A Taxonomy as a Vehicle for Learning, Interdisciplinary Journal of E-Learning and Learning Objects, Volume 5, 2009

David R. Martin, (2007), Sorting Algorithm Animations, http://www.sorting-algorithms.com/, Last accessed at 30.11.2012

Duane J. Jarc, (1998) Interactive Data Structure Visualizations, http://www.student.seas.gwu.edu/~idsv/idsv.html, Last accessed at 30.11.2012

Galles David, Data Structure Visualizations, (2011), http://www.cs.usfca.edu/~galles/visualization/ Last accessed at 30.11.2012

Grozdev S. T. Terzieva, (2011) Visualization of sorting methods, Electronic Journal "Information environment for eucation and science", Volume 5, 2011, (in Russian), URL: http://www.iiorao.ru/iio/pages/izdat/ison/publication/ison_2011/num_5_2011

Hadjerrouit S.,(2010) Developing Web-Based Learning Resources in School Education: A User-Centered Approach, Interdisciplinary Journal of E-Learning and Learning Objects Volume 6, 2010

Jenkins T, (2002), On the Dificulty of Learning to Program, 3rd Annual LTSN-ICS Conference, Loughborough University, http://www.ics.heacademy.ac.uk/Events/conf2002/tjenkins.pdf, Last accessed at 28.11.2011

Korhonen A, L. Malmi, J. Nikander & P. Tenhunen, (2003), Interaction and Feedback in Automatically Assessed Algorithm Simulation Exercises, Journal of Information Technology Education, Volume 2, 2003

Krushkov H., M. Krushkova, V. Atanasov, & M. Krushkova, (2009), A Computer –Based Tutoring System for Programming, Mathematics and Mathematical Education, 2009, (in Bulgarian)

Lahtinen E., (2006), Integrating the Use of Visualizations to Teaching Programming, MMT2006 Proceedings, http://www.codewitz.net/mmt2006proceedings.php, Last accessed at 28.11.2011

Marcelino M., T. Mihaylov, & A. Mendes, (2008) H-SICAS, a Handheld Algorithm Animation And Simulation Tool To Support Initial Programming Learning, 38th ASEE/IEEE Frontiers in Education Conference, October 22 – 25, 2008, Saratoga Springs, NY

Mendes A. J.(2006) , ProLEARN, A Platform to Support Programming Learning, MMT2006 Proceedings, http://www.codewitz.net/mmt2006proceedings.php, Last accessed at 28.11.2011

Morris J, (1998), Animated Algorithms, http://www.cs.auckland.ac.nz/~jmor159/PLDS210/alg_anim.html, Last accessed at 30.11.2012

Nokelainen, P. (2006) An empirical assessment of pedagogical usability criteria for digital learning material with elementary school students. Educational Technology & Society, 9 (2), 178-197.2006

Project Codewitz, (2006) (http://www.codewitz.net/minerva/ , Last accessed at 28.11.2011

Roeßling G., (2010), A Family of Tools for Supporting the Learning of Programming, Algorithms 2010, 3, 168-182;

TEOH Belinda Soo-Phing & Tse-Kian NEO, (2007), Interactive Multimedia Learning: Students Attitudes and Learning Impact in Animation Course, The Turkish Online Journal of Educational Technology – TOJET October 2007 ISSN: 1303-6521 volume 6 Issue 4 Article 3

Tuparov G., D. Tuparova, A. Tsarnakova, (2012), Using Interactive Simulation-Based Learning Objects in Introductory Course of Programming, Procedia - Social and Behavioral Sciences, Volume 46, 2012, Pages 2276-2280, ISSN 1877-0428,

Zeil S. J., (2011), ALGAE - Algorithm Animation Engine, AlgAE (Algorithm Animation Engine) Reference Manual Version 3.0, 2011, http://www.cs.odu.edu/~zeil/AlgAE/referenceManual.pdf, Last accessed at 30.11.2012