SUPPORTING THE LEARNING AND TEACHING OF PROGRAMMING BY USING INTERACTIVE ANIMATION MODELS EMBEDDABLE INTO ELECTRONIC LEARNING MATERIALS

THESES OF THE DOCTORAL DISSERTATION

LADISLAV VÉGH

PhD School of Informatics of Eötvös Loránd University
PhD Program: Basics of informatics and its methodology

Leader of the PhD School: Dr. Erzsébet Csuhaj Varjú
Doctor of MTA, university full professor

Leader of the PhD Programme: Dr. János Demetrovics
Doctor in Mathematical Science, university full professor

Advisors:

Prof. Ing. Veronika Stoffa, CSc.
university full professor

Dr. Márta Turcsányi-Szabó
associate professor

2017

Introduction

Programming is an important part of computer science education. However, learning programming and acquire algorithmic thinking is one of the hardest tasks for first-year computer science undergraduate students.

Algorithms describe processes that are abstract and dynamic, but the methods which are used for teaching algorithms are not. Using animations in education can help to understand algorithms easier, because according to the following assumptions:

- graphical representation is better than a textual explanation;
- dynamic graphics is better than a static one.

However, many of the prior pedagogical experiments do not support these assumptions in every case (Hansen, Narayanan, & Hegarty, 2002; Hundhausen, Douglas, & Stasko, 2002).

During his research, Young also dealt with the time that students spent studying beyond teaching hours. In his publication, he described that today's students are so accustomed to distracting their attention (e.g. by advertising, images, videos in the media) that they can concentrate harder than students in the past (Young, 2002). One of the advantages of visualization-supported education is that students can better focus on processes illustrated in animations (Grissom, McNally, & Naps, 2003).

Carefully designed animations and visualizations increase students’ motivation, make learning more enjoyable, help students acquire new, short and long-term knowledge, and reduce students’ dropouts at universities (Urquiza-Fuentes & Velazquez-Iturbide, 2013).

Motivation and goals

Our many years of experiences of teaching programming also shows that the biggest problem of first-year undergraduate students is to understand algorithms and develop algorithmic thinking. To facilitate the understanding of algorithms, we tried to apply different visualizations and animations in education, including some of our own developments. Our preliminary experiences and results in this field have been continuously published: (Stoffa & Végh, 2006a, 2006b, 2014; Stoffová & Végh, 2007, 2010, 2014; Végh, 2006a, 2006b, 2010a, 2010b).

The aim of the doctoral dissertation and the related research is to summarize the impact of interactive algorithm animations embeddable into electronic learning materials and to evaluate the usage of animations in education among beginner programmers (first-year undergraduate students). Our specific goals:

- Summarizing the principles and methodological suggestions from different literature sources that can be used for creating high-quality interactive animations in education.
- Assessing the possibilities of using virtual worlds for teaching and learning programming; developing a virtual educational space in Second Life environment, creating interactive animation models in this environment that illustrate some basic sorting algorithms.
- Developing an own, multilingual web portal and filling it up with algorithm animations and visualizations that can be embedded into electronic learning materials and meet most of the principles summarized from the literature. The portal can be a useful starting point for students and teachers. Students can browse the collected animations while teachers can embed them into their own electronic learning materials and web pages.
- Developing game-based interactive animations that help students to understand algorithms using everyday objects (such as playing cards, wooden boxes), discover these algorithms, their features and differences, but do not go into details. By experimenting with these game-based animations, students can playfully create the steps of the solutions (algorithms) and develop their algorithmic thinking.
- Developing a JavaScript library that supports creating micro-level interactive animations. Our goal was to create animations using this library, demonstrate some basic algorithms and various algorithms on one-dimensional arrays, including the sorting algorithms. Using these micro-level animations, students can learn algorithms in details, in lower abstraction level. The interactive animations also include the pseudocode or source code of the algorithms.
- Another goal of our pedagogical research was to assess students' views and attitudes towards using interactive animations for learning algorithms and programming.

Applied methods

In the first part of the dissertation, we summarized the principles and suggestions from literature that can be used to create educational animations of algorithms. One part of these recommendations focuses on the graphical representation, the other part on the interactivity of animations.

Next, we created an educational place in Second Life virtual world. In this virtual school, some animated models of sorting algorithms are located, as well. These interactive animations of sorting algorithms were later redesigned into JavaScript animations on a website that can be easier accessible for students.

In the next part of the doctoral dissertation, we have collected animations for teaching and learning algorithms. This collection also contains interactive animations developed by us. The educational effectiveness of our animations and their impact on the acquired knowledge were assessed by pedagogical experiments:

- During the pedagogical experiment regarding the interactive animation "sorting cards" (http://anim.ide.sk/sortingcards.php), students' knowledge were assessed using pre-tests and post-tests. We made several paired sign tests and a McNemar test to determine the educational effectiveness of animations.
- During the pedagogical research regarding the interactive animation "sorting boxes" (http://anim.ide.sk/sortingboxes.php), students filled in a questionnaire and the results of their experiments with the didactic game were saved into a MySQL database. Students' task was to develop such a sorting algorithm (without previous knowledge about the complexity of algorithms), that do not make any redundant comparison. We analyzed students' answers and used Friedman test on the collected data.
- For the easier development of micro-level animations, we developed a JavaScript library. Using this library we created several interactive animations of sorting algorithms. During the pedagogical experiment, the first group of students used these interactive animations to understand the algorithms, while the second group of students used graphical representations of the essential steps of the algorithms. In both groups, after the explanations and discussions, students filled in several tests to assess their knowledge. To compare the answers of two groups of students, we used Mann-Whitney U tests.

Summary of Results

The output and the result of the doctoral dissertation are as follows:

- **We collected and summarized those suggestions, methods, and principles from literature**, that can be useful for developing and using educationally effective interactive animations for teaching and learning algorithms (Végh, 2011a).
  Next, we evaluated the validity of our hypotheses.
- **Hypothesis about students' opinion:**

Hypothesis no. 1: The majority of the inquired undergraduate computer science students prefer to study algorithms using animations instead of static images or textual explanations.

  We measured students' opinion using a questionnaire. 56 undergraduate computer science student from J. Selye University were involved in the survey during the winter semester of the academic year 2014/15.

The results show that significant majority of students (85,7%) prefer using animations for learning algorithms instead of using static images or textual explanations, **χ^2(2, N = 56) = 70.107, p < 0.0005** (SPSS Statistics, 2013) (Végh & Stoffová, 2016).

- **We developed an educational place in Second Life virtual world (http://luckstonesuli.blogspot.com).** The results of a questionnaire and our experiences show that students like to study in virtual worlds (Végh, 2012, 2014b, 2016d; Végh & Turcsányi-Szabó, 2017). In this virtual space, we also created interactive models some of the sorting algorithm ("sorting cards"), which were later improved and redesign into the form of JavaScript animations that can be embeddable into electronic learning materials.
- **We developed a website, where we collected animations for teaching and learning algorithms (http://algoanim.ide.sk).** The portal can be a great help for students learning computer science algorithms (Végh, 2016a, 2016b). The collection also contains some of our interactive animations, which were used in different pedagogical experiments. The results of these pedagogical researches are summarized in the following paragraphs.
- **Hypothesis related to the interactive animation „sorting cards" (http://anim.ide.sk/sortingcards.php):**

Hypothesis no. 2: Using the interactive animation „sorting cards", students can recognize the main features and differences of basic sorting algorithms.

Using these interactive animations, we were conducted a pedagogical experiment at J. Selye University during the summer semester of academic years 2014/15 and 2015/16 within the course „Algorithmization and programming". In the experiment were involved 92 first-year undergraduate computer science students.

The research was realized using questionnaires (tests). Using a pre-test, we measured student's prior knowledge about the sorting algorithms; afterward, students experimented with interactive animations; and finally, they filled in a post-test (Végh, 2016e; Végh & Stoffová, 2017; Végh & Takáč, 2017).

The first part of the tests contained 35 algorithm-statement combinations (7 statements x 5 sorting algorithms). Students' task was to mark those algorithm-statement combinations which are true. During the experiment, the number of correctly marked true algorithm-statement combinations increased by 53.1%, from 654 (54.7%) to 1001 (83.7%). The number of incorrectly marked false algorithm-statement combinations decreased by 39.7%, from 335 (18.2%) to 202 (11.0%). To determine whether the increase in the number of correctly marked answers and the decrease in the number of incorrectly marked answers are significant, we made two paired sign tests.

For true algorithm-statement combinations, the statistically significant increase in the median of the differences is 4 marks (pre-test median: 7 marks, post-test median: 11 marks), **$z = 8.087$, $p < 0.0005$** (SPSS Statistics, 2013).

For false algorithm-statement combinations, the statistically significant decrease in the median of the differences is **-1** mark (pre-test median: 4 marks, post-test median: 2 marks), **$z = -4.037$, $p < 0.0005$** (SPSS Statistics, 2013).

In the second part of the questionnaire, we asked students to pair sorting algorithms to their pseudocodes. We used McNemar tests to compare students' answers in pre-test and post-test (SPSS Statistics, 2013). The results do not show any significant changes (simplesort: **$N = 88$, $\chi^2(1) = 2.370$, $p = 0.124$**; bubblesort: **$N = 89$, $\chi^2(1) = 0.000$, $p = 1.000$**; insertion sort: **$N = 87$, $p = 1.000$**; minsort: **$N = 88$, $p = 1.000$**; maxsort: **$N = 89$, $p = 1.000$**).

Based on the results of the statistical tests, we can say that even though the interactive animations of sorting cards helped students to understand the essential steps and differences of the algorithms, students did not learn the algorithms in-depth (however, learning the algorithms in-depth was not the goal of using these animations with conceptual view).

- **Hypothesis related to the interactive animation „sorting boxes" (http://anim.ide.sk/sortingboxes.php):**

Hypothesis no. 3: Using the interactive animation „sorting boxes", students can create their own sorting algorithm, and reduce in their algorithms the number of unnecessary comparisons.

Using the didactic game "sorting boxes", we were conducted a pedagogical experiment at J. Selye University during the summer semester of the academic years 2014/15 and 2015/16. 99 first-year undergraduate computer science students were involved in the experiment: 50 in the academic year 2014/15, and 49 in the academic year 2015/16. During the experiment, we used a MySQL database to save the number of comparisons of students' solutions. Furthermore, students also filled in a questionnaire, where instead of their name they wrote an identification number that they got in the didactic game. Thus, we were able to connect students' answers to the data in the MySQL database.

After examining the data in the database, we saw that students were able to reduce the number of unnecessary comparisons during their first six successful attempts while they experimented with the animation. The result of Friedman test showed that the number of redundant comparisons in students' successful solutions was statistically significantly different at different attempts during the pedagogical experiment, **$\chi^2(5) = 35.560$, $p < 0.0005$** (SPSS Statistics, 2013). Pairwise comparisons with Bonferroni correction for multiple comparisons revealed statistically significant differences in the number of redundant comparisons between the **1st attempt (Mdn = 7)** and **5th attempt (Mdn = 0) (p = 0.001)**, **1st attempt (Mdn = 7)** and **6th attempt (Mdn = 1) (p < 0.0005)**, **2nd attempt (Mdn = 3)** and **5th attempt (Mdn = 0) (p = 0.027)**, and **2nd attempt (Mdn = 3)** and **6th attempt (Mdn = 1) (p = 0.004)** (SPSS Statistics, 2013).

After reviewing and analyzing students' answers, we recognized that the significant majority of the students (47%) used a modified version of the bubblesort algorithm (bubblesort without redundant comparisons) to sort the boxes: χ^2**(3, N = 70) = 25.657, p < 0.0005** (SPSS Statistics, 2013). 20 students used the original version of the bubblesort algorithm, 13 students used the quicksort algorithm, and 4 students used a modified version of the mergsort algorithm (Végh, 2016e; Végh & Stoffová, 2016).

- **Interactive animation for creating singly linked lists (http://anim.ide.sk/list.php):**
  Using this animation, students can easier understand the steps of creating singly linked lists and visualize the processes in memory.
  During the academic year 2014/15, we used a questionnaire to survey students' opinion about using this interactive animation in education. 55 undergraduate computer science students from J. Selye University were involved in the survey. Students used a 5-point Likert scale to express their opinion about different features of the web application. The majority of students rated the animation positively; all mean ratings of questions were between 3.75 and 4.58.
- **JavaScript library for creating interactive animations (http://inalan.ide.sk/):**
  The JavaScript library that we developed contains several functions for creating interactive animations illustrating algorithms on one-dimensional arrays (Végh, 2016c). The library helped us to develop various micro-level animations of algorithms, which were used to conduct a pedagogical experiment described in the following section.
- **Hypothesis related to the interactive micro-level animations of sorting algorithms (http://ani.ide.sk/):**

Hypothesis no. 4: Students who use our interactive animations for learning sorting algorithms get better results on tests than students who use static graphical illustrations.

In the experiment were involved 71 first-year undergraduate students from J. Selye University. The pedagogical research was conducted within the course "Algorithmization and programming" during the academic year 2015/16.

Before the experiment, students filled in a test to assess their prior knowledge. During the pedagogical experiment, two group of students (35 students) used interactive animations to understand the sorting algorithm, while other two group of students (36 students) used static graphical illustrations. First two group of students (groups using interactive animations) used the http://ani.ide.sk/ website, the second two group of students (groups using graphical representations) used the http://gra.ide.sk/ website to learn the sorting algorithms during the course. The interactive animations and graphical representations were completed with the teacher's explanation in voice. After the explanations, students had time to experiment with the animations or think over the steps of the algorithms represented by static images. They could ask questions for clarification of the steps of the algorithms. Explaining the algorithms and answering the questions usually took longer in the study groups where static pictures were used. When students did not have any more questions, they were asked to fill in a test related to the given sorting algorithm.

We used Mann-Whitney U tests to evaluate the scores of the tests; the following table shows the results for different sorting algorithms.

Sorting algorithm	Group using INTERACTIVE ANIMATIONS	Group using GRAPHICAL REP-RESENTATIONS	Results
Simplesort	mean rank = 40.19	mean rank = 27.62	U = 350, z = -2.715, p = 0.007
Bubblesort	mean rank = 38.76	mean rank = 29.09	U = 399, z = -2.115, p = 0.034
Bubblesort2	mean rank = 43.29	mean rank = 23.62	U = 208, z = -4.245, p < 0.0005
Insertsort	mean rank = 37.03	mean rank = 29.32	U = 402, z = -1.713, p = 0.087
Insertsort2	mean rank = 40.05	mean rank = 26.57	U = 308, z = -2.913, p = 0.004
Minsort	mean rank = 34.75	mean rank = 24.37	U = 280, z = -2.365, p = 0.018
Maxsort	mean rank = 32.91	mean rank = 26.56	U = 339, z = -1.467, p = 0.142
Quicksort	mean rank = 32.37	mean rank = 20.00	U = 184, z = -2.964, p = 0.003

To evaluate the scores of the test focusing on the understanding of mergesort algorithm, we used independent-samples t-test (because the distribution of scores in both groups was close to normal). The results show that students who used interactive animations to learn mergesort algorithm reached higher scores in test **(mean: 70.25 ± 20.32)** than students who used static images **(mean: 51.74 ± 22.90)**, statistically significant difference **18.51 (95% CI, 5.71 to 31.32), t(45) = 2.912, p = 0.006**.

In conclusion, students who learned sorting algorithms using our interactive micro-level animations got better results on tests than students who used static images. The results were statistically significant for seven sorting algorithms, for two algorithms (insertsort, maxsort) there was no statistically significant difference between the groups (Végh, 2016c).

Publications relevant to theses

Végh, L. (2011a). *Animations in Teaching Algorithms and Programming (Animácie vo vyučovaní algoritmov a programovania).* Paper presented at the Nové technologie ve vzdělávání, Olomouc, CZ.

Végh, L. (2016a). *Interaktív algoritmus animációk az oktatásban.* Paper presented at the XXIX. DIDMATTECH 2016, Budapest, HU.

Végh, L. (2016b). Interaktívne animácie vo vyučovaní algoritmov (Interactive animations in teaching and learning programming). *Edukacja – Technika – Informatyka (Education – Technology – Computer Science), 15*(1), 207-211. doi:10.15584/eti.2016.1.29

Végh, L. (2016c). Javascript library for developing interactive micro-level animations for teaching and learning algorithms on one-dimensional arrays. *Acta Didactica Napocensia, 9*(2), 23-32.

Végh, L. (2016e). Using Interactive Game-based Animations for Teaching and Learning Sorting Algorithms. *eLearning and Software for Education, 1*(2016), 565-570. doi:10.12753/2066-026X-16-083

Végh, L., & Stoffová, V. (2016). An interactive animation for learning sorting algorithms: How students reruced the number of comparisons in a sorting algorithm by playing a didactic game. *Teaching Mathematics and Computer Science, 14*(1), 45-62. doi:10.5485/TMCS.2016.0415

Végh, L., & Stoffová, V. (2017). Algorithm Animations for Teaching and Learning the Main Ideas of Basic Sortings. *Informatics in Education, 16*(1), 121-140. doi:10.15388/infedu.2017.07

Végh, L., & Takáč, O. (2017) Using interactive card animations for understanding of the essential aspects of non-recursive sorting algorithms. In*: Vol. 511 AISC. Federated Conference on Software Development and Object Technologies, SDOT 2015* (pp. 336-347): Springer Verlag.

Other publications related to doctoral dissertation

Csízi, L., & Végh, L. (2011). *Virtuális világok az oktatásban.* Paper presented at the III. Oktatás-Informatikai Konferencia, Budapest.

Gubo, Š., Jaruska, L., & Végh, L. (2012). *The Possibilities of Using Virtual Worlds in Teaching Basics of 3D Modelling and Simulation (Možnosti využitia virtuálnych svetov vo vyučovaní základov 3D modelovania a simulácie).* Paper presented at the XXX. International Colloquium on the Management of Educational Process, Brno, CZ.

Kristóf, Z., Végh, L., & Bodnár, K. (2011). Felsőoktatásban alkalmazott Sloodle eszközrendszer használati tapasztalatai. Egy saját eszköz bemutatása. *Oktatás-Informatika, 2*(3-4), 68-76.

Stoffa, V., & Végh, L. (2006a). A programozás tanításának és tanulásának elektronikus támogatása. *Eruditio-Educatio, 3 (2006)*, 105-113.

Stoffa, V., & Végh, L. (2006b). *Guided animation of dynamic data structures*. Paper presented at the Third Central European Multimedia and Virtual Reality Conference, Eger, Hungary.

Stoffa, V., & Végh, L. (2014). *Didaktikai kutatásra szolgáló adatbegyűjtő információs rendszer.* Paper presented at the Agria Media 2014, Eger, HU.

Stoffová, V., & Végh, L. (2007). *Tvorba animačno-simulačných modelov v rôznych prostrediach*. Paper presented at the Informatika v škole a v praxi, Ružomberok, Slovakia.

Stoffová, V., & Végh, L. (2010). *Szemléltető animációk a programozásban.* Paper presented at the INFODIDACT 2010, Szombathely.

Stoffová, V., & Végh, L. (2013). *Grafické modely dynamických údajových štruktúr a ich význam vo vyučovaní programovania (Visual models of dynamic data structures and their importance in teaching programming).* Paper presented at the Trendy ve vzdělávání 2013, Olomouc, CZ.

Stoffová, V., & Végh, L. (2014). *Prieskum vhodnosti používania animačno-simulačných modelov algoritmov vo vzdelávaní (Survey of Effectiveness of Animation-Simulation Models of Algorithms in Education).* Paper presented at the XXVII. DIDMATTECH 2014, Olomouc, CZ.

Turcsányi-Szabó, M., Csízi, L., & Végh, L. (2012). Virtual Worlds in Education - best practice, design and research consideration. *Teaching Mathematics and Computer Science, 10*(2), 309-323. doi:10.5485/TMCS.2012.0308

Vegh, L. (2011). *From Bubblesort to Quicksort with Playing a Game*. Brno: Univ Defence, Fac Econ & Management.

Végh, L. (2006a). *Elektronická podpora vyučovania dynamických údajových štruktúr (Electronic support of teaching dynamic data structures)*. Paper presented at the XIX. DIDMATTECH 2006, Komárno.

Végh, L. (2006b). *Vizualizácia algoritmov vo vyučovaní programovania*. Paper presented at the Informatika v škole a v praxi, Ružomberok, Slovakia.

Végh, L. (2010a). *Tvorba animácií pre e-learning (Creating animations for e-learning)*. Paper presented at the XXII. DIDMATTECH 2009, Trnava, Slovakia.

Végh, L. (2010b). *Vyučovanie algoritmizácie a programovania hravou formou (Teaching algorithmization and programming plazfully)*. Paper presented at the XXVIII International Colloquium on the Management of Educational Process, Brno, Czech Republic.

Végh, L. (2011b). *From Bubblesort to Quicksort with Playing a Game (Hravou formou od bublinkového triedenia po rýchle triedenie).* Paper presented at the XXIX. International Colloquium on the Management of Educational Process, Brno, CZ.

Végh, L. (2012). *Informatikai tantárgyak oktatásának lehetőségei a virtuális világokban.* Paper presented at the IV. Oktatás-Informatikai Konferencia, Budapest.

Végh, L. (2014a). *Methods of Creating Educational 3D Animations Models in Virtual Worlds* Paper presented at the XXVI. DIDMATTECH 2013, Győr.

Végh, L. (2014b). *Programozás alapjainak oktatása az LSL szkriptnyelv használatával.* Paper presented at the Oktatás és tudomány a XXI. század elején, Komárno.

Végh, L. (2016d). *Second Life virtuális világban kialakított oktatási környezet felhasználási lehetőségei.* Paper presented at the SJE Nemzetközi Tudományos Konferencia 2016, Komárno, Slovakia.

Végh, L., & Csízi, L. (2010). *Using Virtual Worlds in Education (Využitie virtuálnych svetov vo vzdelávaní).* Paper presented at the Education and Technology / Edukacja i Technika, Radom, PL.

Végh, L., Gubo, Š., & Takáč, O. (2015). *Interactive Algorithm Animations in Virtual Worlds (Interaktívne animácie algoritmov vo virtuálnych svetoch).* Paper presented at the XXXIII. International Colloquium on the Management of Educational Process, Brno, CZ.

Végh, L., & Turcsányi-Szabó, M. (2017). Using a Virtual School for Teaching and Learning the Basics of 3D Modeling and LSL Scripting in Second Life. In *Could technology support learning efficiency?, Vol I* (pp. 572–579). Bucharest: Carol I Natl Defence Univ Publishing House.

References

Grissom, S., McNally, M. F., & Naps, T. (2003). *Algorithm visualization in CS education: comparing levels of student engagement.* Paper presented at the Proceedings of the 2003 ACM symposium on Software visualization, San Diego, California.

Hansen, S., Narayanan, N. H., & Hegarty, M. (2002). Designing educationally effective algorithm visualizations. *Journal of Visual Languages and Computing, 13*(3), 291-317. doi:10.1006/s1045-926x(02)00027-7

Hundhausen, C. D., Douglas, S. A., & Stasko, J. T. (2002). A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing, 13*(3), 259-290. doi:10.1006/s1045-926x(02)00028-9

Urquiza-Fuentes, J., & Velazquez-Iturbide, J. A. (2013). Toward the effective use of educational program animations: The roles of student's engagement and topic complexity. *Computers & Education, 67*, 178-192. doi:10.1016/j.compedu.2013.02.013

Young, J. R. (2002). Homework? What Homework? Students Seem to Be Spending Less Time Studying Than They Used To. *The Chronicle of Higher Education, 49*(15), A35-A37.