An Experience of Game-Based Learning in Web Applications Development Courses

Míriam Antón-Rodríguez
University of Valladolid, Spain
mirant@tel.uva.es

María Ángeles Pérez-Juárez
University of Valladolid, Spain
mperez@tel.uva.es

Francisco Javier Díaz-Pernas
University of Valladolid, Spain
pacper@tel.uva.es

David González-Ortega
University of Valladolid, Spain
davgon@tel.uva.es

Mario Martínez-Zarzuela
University of Valladolid, Spain
marmar@tel.uva.es

Javier Manuel Aguiar-Pérez
University of Valladolid, Spain
javagu@tel.uva.es

Abstract

Preparing graduates for working in the software engineering industry is challenging and requires effective learning frameworks and methodologies. More specifically, the challenge of teaching programming languages and paradigms is a very complex task that needs innovative educational tools. This paper presents a game-based educational tool named eLiza, developed and used to support the teaching and learning of programming languages and paradigms related to the development of web applications. eLiza was initially developed as a Moodle-based web application because Moodle is the educational eLearning platform used at the University of Valladolid, but as the use of mobile devices is constantly increasing, Android and iOS versions were created later in order to facilitate the access of the students to the games. This paper describes the main elements and the mechanics in playing eLiza. And it also describes an experience of its use in two engineering courses related to web programming applications development, offered to students in two different engineering study programs at the University of Valladolid, during the academic years 2017-2018 and 2018-2019. The great majority of the students, more than 75%, considered that the use of the eLiza game-based educational tool was positive to improve the teaching and learning process of the topics covered by the courses.

2012 ACM Subject Classification Social and professional topics → Computer science education

Keywords and phrases eLearning, mLearning, Game-based Learning, Programming Languages, Web Applications Development

Digital Object Identifier 10.4230/OASIcs.ICPEC.2020.3

1 Introduction

There are a lot of courses whose objective is to teach different programming languages and paradigms to the students. These courses are mainly offered to University students. The learning of programming languages and paradigms is tough and requires a lot of practice.

First International Computer Programming Education Conference (ICPEC 2020).
Editors: Ricardo Queirós, Filipe Portela, Mário Pinto, and Alberto Simões; Article No. 3; pp. 3:1–3:11

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This discipline is also very different to others in which students take memorization procedures as a base. Programming requires a lot of additional work which is developed within the classroom, especially when compared to more theoretical fields of study [4]. According to Vega et al. [19] students have the perception that programming is difficult and it is quite frequent to hear about problems related to frustration and lack of motivation. Moreover, programming is hard to teach, in fact, there are professors that think that programming requires abilities more than knowledge [17].

In such a complex context, gamification can play an important and positive role. Teachers can use game mechanics in non-game contexts to engage students in solving problems and increase their motivation and academic performance. Students who participate in games develop more intellectual abilities than those who do not [6]. Some studies have shown that the part of spare time that students spend gaming exceeds the part of spare time that they spend watching television [7]. For this reason, to take advantage of the benefits of games for educational purposes would open a lot of new possibilities.

In order to explore the possibilities of using game-based educational tools in the formal teaching of programming languages and paradigms, a game-based educational tool eLiza has been developed and tested in different courses offered at the University of Valladolid for engineering students. eLiza was initially developed as a Moodle-based tool. The reason is that Moodle is the educational eLearning platform used by the University of Valladolid to support the teaching and learning process in formal courses. Moreover, more recently Android and iOS versions have also been implemented to facilitate the access of the students to the games.

eLiza uses different strategies to increase students' motivation. The objective is that students really enjoy themselves as they do when they play a game they like in their spare time. The experiences have been carried out in two engineering courses, related to web programming applications development, during the academic years 2017-2018 and 2018-2019. The analysis of the experiences has been based in both qualitative and quantitative procedures.

Therefore, the objective of this paper is to present a game-based educational tool that aims to support the teaching and learning process of programming learning and paradigms. As well as to test the tool in a university academic environment in order to determine the usefulness of gamification and more specifically of this educational tool in such a context for both students and teachers.

2 Teaching Programming Languages and Paradigms

Besides the inherent difficulty of programming as a discipline, the focus of the problem could be in the use of inefficient and inadequate methodologies and educational tools to teach this subject. For this reason, during the last decades, researchers have searched for ways to improve the academic performance of students, especially in the case of newcomer students [12].

With such a complex context it is important to have a clear view of which are the main problems that the students face when approaching to the theory and practice of the programming languages and paradigms. The educational tools oriented to learning programming should take into account theses needs and provide resources and strategies to manage them.

The first main issue is the fact itself of having to use technology-based educational tools. At this moment, the focus is not on the use of the computer itself as younger generations are used to use them, but on the anxiety that generates the use of certain complex software

applications. Anxiety and a negative attitude negatively affect learning and conditions the use of the computer [13]. For this reason, it is important that students can quickly and easily adapt to these type of systems.

A first conclusion is that a good educational tool focused on programming must be easy to use so the students invest their effort and time in learning programming languages and paradigms and not in learning how to use the tool itself. Another central issue is motivation. Students must spend lots of hours practising if they want to learn how to program, and this is not possible if they are not very motivated [11]. Students and teachers both think that programming is difficult to learn, specially for new students. Students must be adequately motivated during the whole process. Improving the practises of teaching and learning programming is not possible without paying special attention to the motivation of students for learning [8]. According to Brito & Sá-Soares [4], motivation is the most important factor to succeed when learning how to program. A motivated student will succeed no matter other circumstances including a bad teacher or a bad structure of the course. In the same way, if a student is not motivated he will probably not succeed no matter how favourable are other circumstances.

Most teachers apply different approaches to support the learning process of students and to adequately motivate them. Most times this approach is based in the use of technology [18]. The results of different studies also suggest that technology-based educational tools that are easy to use can improve motivation and efficiency in the learning process [11].

When teaching programming languages and paradigms it is very important to identify concepts with very precise definitions. Students must understand concepts in order to be able to solve programming challenges [9].

In order to prepare students for programming it is important that they first know the basic concepts of algorithmic thinking. This means that students must be able to clearly define a problem, to divide it into smaller parts that are easier to solve and to determine the steps to solve the entire problem. In order to fulfil this objective students must be able to distinguish the essential characteristics from the unnecessary details [5].

So programming requires that students understand the problem, define the solution and finally translate the solution into code by using a programming language. Different studies show that students find serious difficulties in every step of this process [10].

During the last years, teachers and researches had tried to improve the teaching of programming. To achieve this objective, they have focused on the different elements of the process. One of the aspects in which they have focused is methodologies. In fact, choosing the right methodology to teach programming is one of the main issues of the debate of teaching programming languages and paradigms [1, 2, 3].

A challenge is also to convert programming into an activity that is mainly developed in groups instead of individually. With the idea of promoting workgroup in the learning process of languages and paradigms some projects have been developed like Nucleo [15]. This project promotes that students acquire social skills and abilities for working in groups and that they adopt a more active role in the learning process. This is important as in the software industry projects are mainly undertaken by groups and not by individual programmers.

3 eLiza

eLiza is a game-based educational tool initially developed for the educational eLearning platform Moodle. eLiza is a competitive game which main objective is that students learn while playing. To achieve this goal, eLiza challenges students with questions classified in different levels of difficulty. eLiza promotes competitiveness among students by offering information about the students' performance.

eLiza is expected to have a positive impact into the learning process by increasing the involvement of students. Teachers can establish prizes at different score levels, so students are encouraged to reach those score levels. Moreover, teachers can use the results obtained by students in eLiza as an assessment element to evaluate them. On the other side, students can use the educational game for their own self-assessment. eLiza was initially developed as a module for the educational eLearning platform Moodle. For this reason, eLiza has been mainly used from desktop browsers, but as it does not have any special requirements, it could have been played from browsers in mobile devices such as tablets and smartphones. Moreover, as the use of mobile applications has grown exponentially in recent years [14], Android and iOS versions have recently been created in order to facilitate the access of the students to the games.

When accessing eLiza, the teacher will see a main screen with five buttons which give him access to the different sections of the application: Let's Play, Let's Play in Groups, Management, Add a new Question and Statistics. The student can access the same sections that the teacher, except for the Management section (see Figure 1).

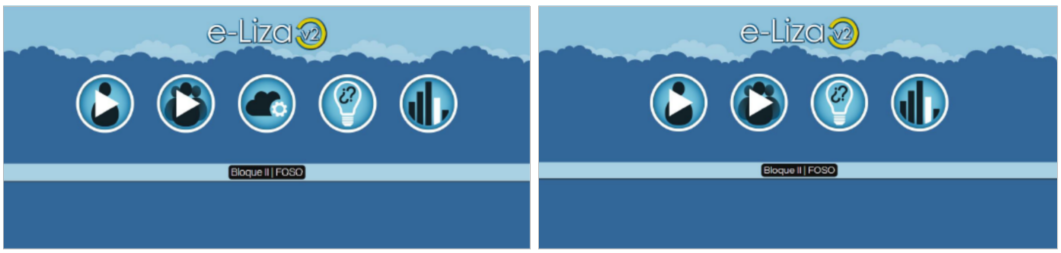

Figure 1 eLiza main access screen for teachers (left) and students (right).

The teacher can manage the questions bank of a game. As shown in Figure 2, to add a new question it is only necessary to complete a form providing the name, the type of question (multiple choice or true/false), the content and the possible answers indicating for each case if the answer is a correct or an incorrect one. Another element that is necessary to define a question is the labels associated to that question. It is possible to add as many labels as desired in order to have the question adequately categorized. Labels must have previously been created. Finally it is necessary to indicate the time the student will have to answer the question during the game, and the value of that question in points, that is the points that the student will be assigned during the game in case he answers the question correctly. Regarding the value of the question it is important to mention that it is also possible to assign a penalty which is a number that indicates the percentage of the value of the question in points that will be subtracted in case the student answers that question incorrectly during the game.

The teacher can also view and manage all the questions proposed by the students for a course. Each of the questions proposed by the students has a state that can be: pending (the question is still waiting for the teacher's approval), approved (the question has been accepted by the teacher) and rejected (the question has been rejected by the teacher). The teacher can see the content of any of the pending questions in order to decide to approve or to reject the question.

Figure 2 Questionnaire to add a new question.

In the Statistics section, the teacher can see the activity of the students in the games. The global statistics include all the students that have participated in the games of a course. The systems presents these statistics by using two type of graphs that are the pie chart and the temporary graph (see Figure 3). The pie chart shows more clearly the percentage of success, failures and not answered questions. While the temporary graph shows more clearly the average score of the students of the course, grouped by months during the period in which those students have taken part in games. The teacher can also view the statistics per student, the statistics per game and the statistics per question.

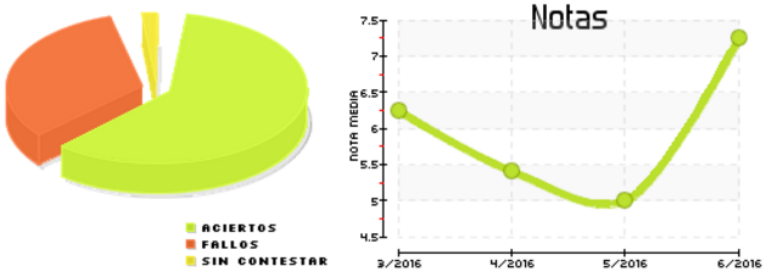

Figure 3 Global statistics (all students & all games).

The student has also access to different functionalities which are summarised in the eLiza main screen shown to him when he accesses the application. One of the most important buttons for the student is the Let's Play button which allows him to start a new game. When the student starts the game he must answer all the questions. After finishing the game the student can see the result including the number of questions answered correctly, the number of points obtained over the maximum number of points that is possible to obtain, and his final score which is the number of points obtained weighted over ten (see Figure 4).

Another possibility available for the student is to play in group. First the student will see the list of all the group games in which he can take part, for example in the case he has been challenged by other student. As in an individual game the student has to answer the different questions proposed within the time provided for each question. Any student can

Figure 4 Information offered to a student after finishing a game.

create a group game indicating the students that are challenged selected from the list of students enrolled in the course. The system announces the winner of a group game only after all the students challenged in that group game has played.

Finally the student has the possibility to access the statistics about his own participation in the different games (see Figure 5. In this case the student can see information about the percentage of times he answered questions correctly and incorrectly, the percentage of times he did not answer a question, the number of complete and incomplete games, that is the number of games in which he answered all the questions, and the number of games in which he did not answered all the questions foreseen. Finally the student can also view the average time that took him to answer a question, and the average and maximum scores that he obtained in the games he took.

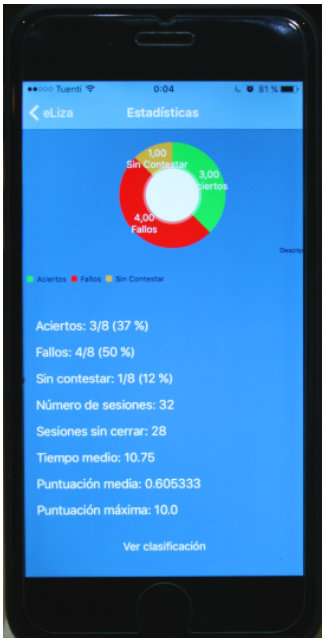

Figure 5 Student's statistics in the eLiza iOS version.

As explained before, eLiza is also available as Android and iOS native applications for mobile devices. The reason to develop also these versions, apart from the desktop version, was to facilitate the access of the students to the games. Most of the students have their own laptops and carry them to the University. However, for obvious reasons, laptops are not continuously turned on as it happens with mobile devices.

4 Results

The courses in which the game-based educational tool eLiza has been used are courses focused on programming languages and paradigms that are part of different engineering study programs offered by the University of Valladolid. The experiences have been carried out during the academic years 2017-2018 and 2018-2019. The students were encouraged to use eLiza and then were invited to talk about their user experience, the benefits obtained, the problems found, the upgrades suggested, etc., in a forum. The great majority of the students, more than 75%, considered the use of the eLiza game-based educational tool was positive to improve the teaching and learning process of the topics covered by the course. Moreover, twenty-five students that participated in the experience using the Android version of eLiza, completed the MARS (Mobile Application Rating Scale) questionnaire [16]. The global results of the experience are shown in Table 1.

Table 1 Global statistics of the experience with the Android version of eLiza.

Statistics Item	Value
Successes	1891/2504 (75.52%)
Faults	588/2504 (23.48%)
Unanswered	25/2504 (1%)
Number of sessions	513
Number of sessions per user	20.52
Number of sessions without closing	22
Average response time	19.64 s.
Average grade	7.1
Maximum grade	10

The results of the survey for each question were the following:

1. Entertainment: Is the app fun/entertaining to use? Does it use any strategies to increase engagement through entertainment (e.g. through gamification)?

   - Dull, not fun or entertaining at all ..... 0.0 %
   - Mostly boring ..... 12.5 %
   - OK, fun enough to entertain user for a brief time (< 5 minutes) ..... 37.5 %
   - Moderately fun and entertaining, would entertain user for some time (5-10 minutes total) ..... 50.0 %
   - Highly entertaining and fun, would stimulate repeat use ..... 0.0 %

2. Interest: Is the app interesting to use? Does it use any strategies to increase engagement by presenting its content in an interesting way?

   - Not interesting at all ..... 0.0 %
   - Mostly uninteresting ..... 12.5 %
   - OK, neither interesting nor uninteresting; would engage user for a brief time (< 5 minutes) ..... 25.0 %
   - Moderately interesting; would engage user for some time (5-10 minutes total) ..... 62.5 %
   - Very interesting, would engage user in repeat use ..... 0.0 %

3. Interactivity: Does it allow user input, provide feedback, contain prompts (reminders, sharing options, notifications, etc.)? Note: these functions need to be customisable and not overwhelming in order to be perfect.

   - No interactive features and/or no response to user interaction ..... 0.0 %
   - Insufficient interactivity, or feedback, or user input options, limiting functions 0.0 %
   - Basic interactive features to function adequately ..... 25.0 %
   - Offers a variety of interactive features/feedback/user input options ..... 37.5 %
   - Very high level of responsiveness through interactive features/feedback/user input options ..... 37.5 %

4. Target group: Is the app content (visual information, language, design) appropriate for your target audience?

   - Completely inappropriate/unclear/confusing ..... 0 %
   - Mostly inappropriate/unclear/confusing ..... 0 %
   - Acceptable but not targeted. May be inappropriate/unclear/confusing ..... 0 %
   - Well-targeted, with negligible issues ..... 50 %
   - Perfectly targeted, no issues found ..... 50 %

5. Performance: How accurately/fast do the app features (functions) and components (buttons/menus) work?

   - App is broken; no/insufficient/inaccurate response (e.g. crashes/bugs/broken features, etc.) ..... 0 %
   - Some functions work, but lagging or contains major technical problems ..... 0 %
   - App works overall. Some technical problems need fixing/Slow at times ..... 0 %
   - Mostly functional with minor/negligible problems ..... 50 %
   - Perfect/timely response; no technical bugs found/contains a “loading time left” indicator ..... 50 %

6. Ease of use: How easy is it to learn how to use the app; how clear are the menu labels/icons and instructions?

   - No/limited instructions; menu labels/icons are confusing; complicated ..... 0.0 %
   - Usable after a lot of time/effort ..... 0.0 %
   - Usable after some time/effort ..... 12.5 %
   - Easy to learn how to use the app (or has clear instructions) ..... 12.5 %
   - Able to use app immediately; intuitive; simple ..... 75.0 %

7. Navigation: Is moving between screens logical, accurate, appropriate, uninterrupted; are all necessary screen links present?

   - Different sections within the app seem logically disconnected and random/confusing/navigation is difficult ..... 0 %
   - Usable after a lot of time/effort ..... 0 %
   - Usable after some time/effort ..... 0 %
   - Easy to use or missing a negligible link ..... 50 %
   - Perfectly logical, easy, clear and intuitive screen flow throughout, or offers shortcuts ..... 50 %

8. Layout: Is arrangement and size of buttons/icons/menus/content on the screen appropriate or zoomable if needed?

   - Very bad design, cluttered, some options are impossible to select/locate/see/read device display not optimised ..... 0.0 %
   - Bad design, random, unclear, some options difficult to select/locate/see/read . 0.0 %
   - Satisfactory, few problems with selecting/locating/seeing/reading items or with minor screen-size problems ..... 12.5 %
   - Mostly clear, able to select/locate/see/read items 37.5%
   - Professional, simple, clear, orderly, logically organised, device display optimized. Every design component has a purpose. 50.0%

9. Graphics: How high is the quality/resolution of graphics used for buttons, icons, menus, content?

   - Graphics appear amateur, very poor visual design; disproportionate, completely stylistically inconsistent ..... 0.0 %
   - Low quality/low resolution graphics; low quality visual design; disproportionate, stylistically inconsistent ..... 0.0 %
   - Moderate quality graphics and visual design (generally consistent in style) ... 37.5 %
   - High quality/resolution graphics and visual design; mostly proportionate, stylistically consistent ..... 50.0 %
   - Very high quality/resolution graphics and visual design; proportionate, stylistically consistent throughout ..... 12.5 %

10. Visual appeal: How good does the app look?

    - No visual appeal, unpleasant to look at, poorly designed, clashing/mismatched colours ..... 0 %
    - Little visual appeal; poorly designed, bad use of colour, visually boring ..... 0 %
    - Some visual appeal; average, neither pleasant, nor unpleasant ..... 50 %
    - High level of visual appeal – seamless graphics – consistent and professionally designed ..... 25 %
    - As above + very attractive, memorable, stands out; use of colour enhances app features/menus ..... 25 %

From the results of the study a series of conclusions were drawn. The majority of the students considered the app was highly entertaining and fun, but a small percentage considered that the app was boring or only entertaining for a brief time. None of the students considered that the app was very interesting, but most of them considered it to be moderately interesting. Most students considered the app to be interactive, but a small group commented that the interactivity characteristics available at the app were very basic. Most students thought that the app was well targeted and considered that the functions available worked correctly, or else encountered minor problems when using it. Finally, most students considered the application to be very easy to use and that the navigation was perfectly logical, or at least easy enough to understand and navigate.

References

1 Owen Astrachan, Kim Bruce, Elliot Koffman, Michael Kölling, and Stuart Reges. Resolved: Objects early has failed. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '05, page 451–452, New York, NY, USA, 2005. Association for Computing Machinery. doi:10.1145/1047344.1047359.

2 Frances Bailie, Mary Courtney, Keitha Murray, Robert Schiaffino, and Sylvester Tuohy. Objects first - does it work? *Journal of Computing Sciences in Colleges*, 19(2):303–305, December 2003.

3 John Bergin, Jutta Eckstein, Mary Lynn Manns, and Eugene Wallingford. Patterns for gaining different perspectives. In *8th conference on Pattern Languages of Programs (PLoP 2001)*, Illinois, USA, 2001.

4 Miguel A. Brito and Filipe De Sá-Soares. Assessment frequency in introductory computer programming disciplines. *Computers in Human Behavior*, 30:623–628, January 2014. doi:10.1016/j.chb.2013.07.044.

5 Jill Denner, Linda Werner, and Eloy Ortiz. Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers and Education*, 58:240–249, 2012.

6 Juan Alberto Estallo. *Los videojuegos. Juicios y prejuicios.* Planeta, Barcelona, 1994.

7 Francisco José García-Peñalvo, Mark Johnson, Gustavo Ribeiro Alves, Miroslav Minović, and Miguel Ángel Conde-González. Informal learning recognition through a cloud ecosystem. *Future Generation Computer Systems*, 32(C):282–294, March 2014.

8 Tony Jenkins. The motivation of students of programming. In *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '01, page 53–56, New York, NY, USA, 2001. Association for Computing Machinery. doi:10.1145/377435.377472.

9 Cagin Kazimoglu, Mary Kiernan, Liz Bacon, and Lachlan MacKinnon. Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, 9:522–531, 2012. doi:10.1016/j.procs.2012.04.056.

10 Maria Kordaki. A drawing and multi-representational computer environment for beginners' learning of programming using c: Design and pilot formative evaluation. *Computers & Education*, 54(1):69–87, 2010. doi:10.1016/j.compedu.2009.07.012.

11 Kris M.Y. Law, Victor C.S. Lee, and Y.T. Yu. Learning motivation in e-learning facilitated computer programming courses. *Computers & Education*, 55(1):218–228, 2010. doi:10.1016/j.compedu.2010.01.007.

12 Jan Moons and Carlos [De Backer]. The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism. *Computers & Education*, 60(1):368–384, 2013. doi:10.1016/j.compedu.2012.08.009.

13 Aysen Gurcan Namlu. The effect of learning strategy on computer anxiety. *Computers in Human Behavior*, 19(5):565–578, 2003. doi:10.1016/S0747-5632(03)00003-7.

14 William T Riley, Daniel E Rivera, Audie A Atienza, Wendy Nilsen, Susannah M Allison, and Robin Mermelstein. Health behavior models in the age of mobile interventions: are our theories up to the task? *Translational Behavioral Medicine*, 1(1):53–71, February 2011. doi:10.1007/s13142-011-0021-7.

15 Pilar Sancho-Thomas, Rubén Fuentes-Fernández, and Baltasar Fernández-Manjón. Learning teamwork skills in university programming courses. *Computers & Education*, 53(2):517–531, 2009. doi:10.1016/j.compedu.2009.03.010.

16 Stoyan R Stoyanov, Leanne Hides, David J Kavanagh, Oksana Zelenko, Dian Tjondronegoro, and Madhavan Mani. Mobile app rating scale: A new tool for assessing the quality of health mobile apps. *JMIR mHealth uHealth*, 3(1):e27, March 2015. doi:10.2196/mhealth.3422.

17 Jun Tan, Xianping Guo, Weishi Zheng, and Ming zhong. Case-based teaching using the laboratory animal system for learning C/C++ programming. *Computers & Education*, 77:39–49, 2014. doi:10.1016/j.compedu.2014.04.003.

18 Georgi Tuparov, Daniela Tuparova, and Anna Tsarnakova. Using interactive simulation-based learning objects in introductory course of programming. *Procedia - Social and Behavioral Sciences*, 46:2276–2280, 2012. doi:10.1016/j.sbspro.2012.05.469.

19 Carlos Vega, Camilo Jiménez, and Jorge Villalobos. A scalable and incremental project-based learning approach for cs1/cs2 courses. *Education and Information Technologies*, 18(2):309–329, June 2013. doi:10.1007/s10639-012-9242-8.