Learning Binary Search Trees through Serious Games

Alberto Rojas-Salazar
Trinity College Dublin, Dublin, Ireland
rojassaa@tcd.ie

Paula Ramírez-Alfaro
University of Costa Rica, Alajuela, Costa Rica
paula.ramirez@ucr.ac.cr

Mads Haahr
Trinity College Dublin, Dublin, Ireland
mads.haahr@tcd.ie

Abstract

Data structures and algorithms are core topics in Computer Science, but they are difficult topics to grasp. Data structures and algorithmic concepts are abstract and difficult to relate to previous knowledge. To facilitate the learning process of these topics, learning tools that link new information with previous knowledge in an active way may be a useful approach to teach data structures and their algorithms. Furthermore, serious games have the potential to serve as a learning tool that accomplishes both objectives: to link new information with previous knowledge and to facilitate active learning. To tackle these issues, we developed *DS-Hacker*, an action-adventure serious game that utilizes the game elements to represent the Binary Search Tree (BST) properties and structure. In this paper, we report the results of a pilot experiment that compares the learning gains after completing two learning activities: (1) playing a serious game for learning Binary Search Trees, and (2) reading a summary and watching two video tutorials. Additionally, we report the results from a qualitative survey that evaluated the game usability, player satisfaction and the participants' perception about the means used by the game to deliver the BST concepts.

2012 ACM Subject Classification Applied computing → Education

Keywords and phrases Binary Search Tree, Data Structures, Serious Games

Digital Object Identifier 10.4230/OASIcs.ICPEC.2020.22

1 Introduction

Data structures and algorithms are core topics in Computer Science, and they are essential for the development of efficient software [10]. Due to the relevance of these topics, data structures and algorithms are included in the guidelines for undergraduate degree programs developed by the Association for Computing Machinery (ACM) [7]. Typically, universities teach the first introductory data structure course in the second year of their undergraduate Computer Science programs [8].

While a deep understanding of data structures is fundamental knowledge for computer scientists, advanced data structures and their algorithms are difficult topics to grasp [2]. Data structures and algorithmic concepts are abstract and difficult to relate to previous knowledge. From a constructivist point of view, it is important that new experiences and information link to previous knowledge in order to create new knowledge [6]. Educators should provide experiences and environments where the students can construct knowledge through reflection, critical thinking and their previous knowledge [6]. Therefore, learning tools that complement classes by linking new information with previous knowledge in an active way may be a useful approach to teaching data structures and their algorithms.

In view of the above, serious games have the potential to serve as a learning tool that accomplishes both objectives: to link new information with previous knowledge and to facilitate active learning. Many game genres are popular among teenagers and young adults, and well-crafted video games promote active learning [5]. Usually, video games offer challenges that require active engagement of the player. Therefore, serious games may take advantage of these characteristics in order to facilitate the association of new information with previous knowledge and active learning.

In this paper, we present a serious game for teaching Binary Search Trees (BSTs) called *DS-Hacker* (Data Structure Hacker). *DS-Hacker* aims to introduce BST concepts to college students by means of relating well-known game elements with BST concepts. These relations are presented to the learner through analogies embedded in the game. We also report the results of a pilot experiment that compares the learning gains after completing two learning activities: (1) playing *DS-Hacker*, and (2) reading a summary and watching two video tutorials. Finally, we report the results from a qualitative survey that evaluated the game usability, player satisfaction and the participants' perception about the means used by the game to deliver the BST concepts.

Results show that both learning approaches produces a learning effect and that there is no statistically significant difference between both activities. The qualitative survey suggests that participants perceived that they learned while playing the game, and that they could relate the BST concepts and structure with the *DS-Hacker* game elements.

2 DS-Hacker

DS-Hacker is a PC game developed with Unity 3D, and its target population are university students from Computer Science and Engineering Schools. The game is a third person 3D action-adventure game, a well-known genre. The aesthetics are sci-fi style, and its story takes place in a distant future where a corrupt corporation is harming the balance of society. In the game, the player takes the role of the robotic hacker that must traverse a maze composed of chambers and extract the information stored in the maze. A video file of the gameplay of the English version is available through the following link: `https://www.dropbox.com/s/8vavy0e7b9uywx6/DS-Hacker_Level1%26Level2.mp4?dl=1`

To achieve learning of the BST structure, *DS-Hacker* uses an analogy between the BST structure and the environment structure. According to the game plot, corporations hide and protect their information in places called "Data Systems" (our game environment). Data systems are mazes organized as well-known data structures, and to achieve our teaching objective, the Data System reflects the structure of a BST. Therefore, many elements of the game environment represent the most important elements of the data structure. For instance, in *DS-Hacker*, the maze's rooms represent the nodes; the portals of each room represent the links that points to other nodes; the room ID represents the comparable key; and the information stored in each room represents the associated values of each node. Furthermore, the chambers of the maze are organized following the BST property.

The game story serves a major function because it delivers the conceptual knowledge. The game story is delivered through a non-player character (NPC) named Anonymous who always appears at the beginning and at the end of each level. Anonymous introduces the missions and the necessary BST concept to accomplish them. In order to facilitate the understanding of the BST concepts, Anonymous takes advantage of analogies between the game elements and the BST elements. For instances, in the first two levels, Anonymous informs the player about the relation between the game environment structure and the

BST structure. In the last three levels, Anonymous presents the relation between the game challenges and the search algorithm.

Currently, *DS-Hacker* possesses five levels, and each level focuses on different concepts of the BST data structure. Level one and two cover topics related to the basic structure of the BST, its properties, and the structure of its nodes. Level three, four and five cover topics related to the search algorithm such as the sequence of the algorithm's steps and its outcomes. Furthermore, each level has a mission, and each mission possesses one or more challenges. Missions provide opportunities to apply and solve problems using the concepts given by Anonymous and to experience the structure of the BST in a concrete manner.

3 Method

Our pilot experiment follows a "switching replications" experimental design. In the study, participants are randomly divided into two groups, G1 and G2. Both groups must complete two activities (the treatment and the control activity) and answer three tests (pre-test, mid-test and post-test). The study is organized as follows: First, all participants take a pre-test; then G1 performs the treatment, and G2 performs the control activity; then, all participants answer the mid-test; then, G1 and G2 switch and perform the other activity; finally, all participants take the post-test. Switching replications design may decrease social threats to validity, since it allows all participants equal access to the treatment activity [4]. However, the learning effect due to the overexposure to the test may lead to a testing threat [4].

The experiment was carried out as a workshop during a class of the 2020 Summer Term in University of Costa Rica. Participants were randomly divided into two groups and assigned to a computer with the game already installed. Participants of G1 played the Spanish version of *DS-Hacker* (the treatment); meanwhile, participants of G2 completed the control activity. Then, G1 performed the control activity, and G2 played the game.

The control activity included two popular teaching methods: a written summary of the BST concepts and three video tutorials. The summary was a Spanish translation of the book *Algorithms* by Sedgewick and Wayne [10]. The first video tutorial[1] was about BST structure and characteristics, and the second[2] was about the search and insert operations (the insert operation was not evaluated). The third video[3] tutorial was a general summary about the BST basic concepts and operations.

The pre-test, mid-test and post-tests were designed to assess the learning gains. The tests have 23 questions and cover the first four levels (remember, understand, apply, and analyse) of the revised version of the Bloom's taxonomy [1]. The questions were multiple-choice, and their construction followed the guidelines suggested in [9]. Furthermore, the questions verify factual, conceptual and procedural knowledge. Besides the tests, participants took a demographic survey at the beginning of the experiment and a qualitative survey to evaluate the game at the end of the experiment. All surveys (and tests) were performed using Google Forms.

Initially, 32 students participated in the experiment; however, we excluded 5 participants from the analysis because they did not complete one of the tests. Therefore, we only take into consideration the 27 participants who completed all the evaluations. Group 1

[1] https://youtu.be/Bh61AvHAf90
[2] https://youtu.be/DVKDQcJOqy8
[3] https://youtu.be/mTMrszfrNtI

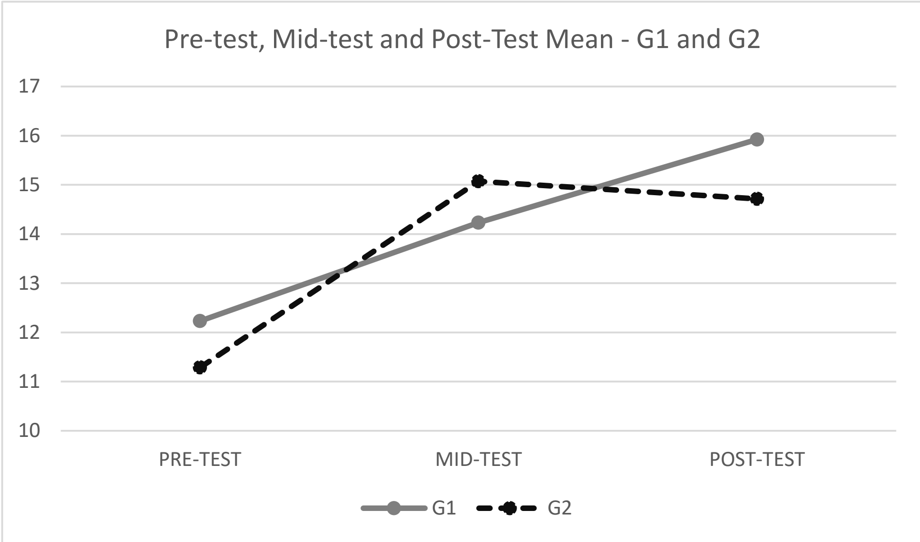

Figure 1 Mean of the pre-test, mid-test and post-test of G1 and G2.

Table 1 G1 and G2 Pre-test, Mid-test and Post-test means and standard deviation.

	G1 Pre-test	G1 Mid-test	G1 Post-test	G2 Pre-test	G2 Mid-test	G2 Post-test
Mean	12.231	14.231	15.923	11.286	15.071	14.714
StDev	3.395	2.948	2.691	3.832	3.452	3.730

(the experimental) had 13 participants, and Group 2 (the control) had 14. In terms of background, 11 students were from the Computer Science School; 10 students were from the Industrial Engineering School; and 5 students from the Electrical Engineering School and the Mathematics School.

4 Results and Discussion

Figure 1 shows the average of the scores obtained during the pre-test, mid-test and post-test of G1 and G2. The chart presents a learning effect for both activities. After the first round of activities, G2's participants performed better in the mid-test than G1's participants. On average, G1's participants (who played the game) increased by 2 points. Meanwhile, G2's participants (who watched the video tutorials) increased by 3.79 points. After switching and completing the second round of activities, G1's participants performed better in the post-test than G2's participants. G1's participants increased by 1.69 points; G2's participants decreased 0.36 points. After both activities, G1's participants increased by 3.69 points, and G2's participants increased by 3.43 points. Table 1 presents the mean and the standard deviation of each test.

Additionally, we performed a t-test analysis to verify whether the difference between the means of the pre-test, mid-test, and post-test were statistically significant. Table 2 presents the results, showing no significant difference. We also verified whether the difference between mid-test and pre-test and post-test and pre-test of G1 and G2 were statistically significant. To achieve this, we per-formed a series of two-tailed paired t-tests with an alpha of 0.05. In addition, to verify the magnitude of the difference, we calculated the Cohen's d

Table 2 Mean, p-value and t statistic of the difference of scores.

	N	Pre-test	Mid-test	Post-test
Group 1 - Mean	13	12.231	14.231	15.923
Group 2 - Mean	14	11.286	15.071	14.714
Difference P-Value		0.503	0.502	0.341
T Statistic		0.679	-0.682	0.971

Table 3 T Statistic, p-value, and effect size of the difference between Mid-test and Pre-test, and Post-test and Pre-test.

	P Value	T Statistic	Effect Size
Difference: Pre-Test and Mid-test of Group 1	0.0218	-2.6331	0.63
Difference: Pre-Test and Mid-test of Group 2	0.000009	-5.7303	1.04
Difference: Pre-Test and Post-test of Group 1	0.000095	0.7012	1.21
Difference: Pre-Test and Post-test of Group 2	0.000094	-5.5511	0.91

that quantifies the effect size. In our case, it determines the magnitude of change in scores. Cohen's d result larger than 0.80 is considered a large size effect; a result around 0.50 is considered a medium size effect, and around 0.20 a small size effect [3]. Table 3 shows the results of the calculations.

The previous results indicate that both learning activities were effective. The differences between pre-test and mid-test of G1 and G2 are statistically significant. However, the results show that the control activity (reading and watching video tutorials) was more efficient than playing *DS-Hacker*. For instance, the effect size of the difference between the mid-test and pre-test of G2 is higher than the size effect of G1. Additionally, the average score of the mid-test of G2 is almost the same as the average score of the post-test of G1. Even though, the difference between mid-test and post-test averages are not statistically significant.

Another interesting finding is that G2 slightly decreased its performance during the post-test (after playing the game) and that G1 increased the scores in the mid-test and post-tests. This discovery suggests that the order of the treatment may affect the learning gains. Further studies regarding the order of the learning activities may lead to promising results.

The qualitative survey has 19 four-point Likert-scale questions divide into three categories. The first category (Q1-Q6) assesses the participant's perception about learning and the means used by the game (environment, story and challenges) to deliver the BST concepts. The second category (Q7-Q15) assesses the usability. The third category (Q16-Q19) assesses the enjoyability. We only present the answers of the 27 participants who completed all the tests. Table 4 presents the percentage of the positive answers ("Strongly agree" and "Moderately agree") of the qualitative survey.

Most of the answers of the qualitative survey were positive. However, three questions received a considerable number of negative responses, indicating that the game has some problems. Q3 responses indicate that half of the participants could not understand the search algorithm principles while playing the game. This is an indication that we should improve the content and levels that cover this topic. Second, Q8 and Q9 answers suggest that participants had trouble dealing with the game controls. The cause of this problem was the low performance of the computers utilized during the experiment. We should consider this factor, and we should optimize the game to make it appropriate for low performance

Table 4 Distribution of the positive answers of the qualitative survey.

Questions	Agree %	Questions	Agree %
Q1. The game help me to understand BST structures.	77.78	Q11. The game tutorial was useful and clear.	81.48
Q2. The game help me to understand the nodes' structure.	81.48	Q12. The voice and way of talking of the NPC ware clear.	81.48
Q3. The game help me to understand the search algorithm.	59.26	Q13. Game missions were clear.	81.48
Q4. I could relate concepts presented in the game story with the BST concepts.	88.89	Q14. The game GUI was easy to un-derstand and intuitive.	85.19
Q5. I could relate the game environment with the BST structure.	85.19	Q15. The game menu has useful op-tions.	81.48
Q6. The game allows me to practice the previously learned BST concepts.	85.19	Q16. I enjoyed playing DS-Hacker	70.37
Q7. The game was easy to learn.	85.19	Q17. I like the way that BST concepts were presented during the game.	81.48
Q8. The game controls were easy to learn.	66.67	Q18. I think that video games increase my motivation towards computer science topics.	85.19
Q9. The game controls respond smoothly.	48.15	Q19. I would like more serious games to be used to teach data structures.	81.48
Q10. The map was easy to understand.	77.78		

computers.

Regarding the learning approach of the game, participants reported that they could relate the BST concepts with the game environment and the story. Additionally, results indicate that participants think that the game provides an environment that allows them to practice the BST concepts. Finally, participants reported that they felt that they were learning while playing the game, and that in general, they enjoyed the game.

5 Conclusion and Future Work

The article also presented the results of a pilot experiment and a qualitative evaluation of *DS-Hacker* which aims to facilitate learning of the BST data structure and associated algorithms by linking new information with previous knowledge and facilitating active learning. The results of the pilot experiment show that the treatment (playing the game) and the control activity (reading a summary and watching video tutorials) produced learning gains on the participants. Differences between the scores obtained by the treatment group and the control group were not statistically significant. However, results from the mid-test suggest that the control activity is slightly more efficient than playing the game.

Our qualitative evaluation showed that the participants could relate game elements (game story, environment and challenges) with the BST concepts. This finding suggests that these game elements may be used to delivery educational information. Additionally, participants felt that they learned while playing the game. Regarding the usability of the game, we must optimize the game to run on low performance computers. In general, participants reported that they enjoyed playing the game.

In the future, we plan to redesign the levels that cover the search algorithm and add more level covering other BST operations such as the tree traversal algorithms and the insert algorithm.

References

1 Lorin W. Anderson and David R. Krathwohl. *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*. Longman, New York, 2001.

2 Katrin Becker and Melissa Beacham. A Tool for Teaching Advanced Data Structures to Computer Science Students: An Overview of the BDP System. In *Proceedings of the Second Annual CCSC on Computing in Small Colleges Northwestern Conference*, pages 65–71, USA, 2000. Consortium for Computing Sciences in Colleges. event-place: Oregon Graduate Institute, Beaverton, Oregon, USA. URL: `http://dl.acm.org/citation.cfm?id=369274.369319`.

3 D. Dicheva and A. Hodge. Active Learning Through Game Play in a Data Structures Course. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, SIGCSE '18, pages 834–839, New York, NY, USA, 2018. ACM. event-place: Baltimore, Maryland, USA. URL: `http://doi.acm.org.elib.tcd.ie/10.1145/3159450.3159605`, `doi:10.1145/3159450.3159605`.

4 Michael Eagle and Tiffany Barnes. Experimental Evaluation of an Educational Game for Improved Learning in Introductory Computing. In *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09, pages 321–325, New York, NY, USA, 2009. ACM. event-place: Chattanooga, TN, USA. URL: `http://doi.acm.org.elib.tcd.ie/10.1145/1508865.1508980`, `doi:10.1145/1508865.1508980`.

5 James Paul Gee. *What Video Games Have to Teach Us about Learning and Literacy*. Palgrave Macmillan, New York, 2nd edition, 2007.

6 Aytac Gogus. Constructivist Learning. In Norbert M. Seel, editor, *Encyclopedia of the Sciences of Learning*, pages 783–786. Springer US, Boston, MA, 2012. `doi:10.1007/978-1-4419-1428-6_4049`.

7 Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, New York, NY, USA, 2013.

8 R. Lawrence. Teaching data structures using competitive games. *IEEE Transactions on Education*, 47(4):459–466, November 2004. `doi:10.1109/TE.2004.825053`.

9 Rafael Moreno, Rafael Martínez, and José Muñiz. Directrices para la construcción de ítems de elección múltiple. *Psicothema*, 16(3):490–497, 2004.

10 Robert Sedgewick and Kevin Wayne. *Algorithms*. Addison-Wesley, 4th edition, 2014.