

# Learning Binary Search Trees through Serious Games based on Analogies

Alberto Rojas-Salazar

School of Computer Science and Statistics, Trinity College  
Dublin, Dublin, Ireland  
rojassaa@tcd.ie

Mads Haahr

School of Computer Science and Statistics, Trinity College  
Dublin, Dublin, Ireland  
mads.haahr@tcd.ie

## ABSTRACT

Data structures and algorithms are core topics in Computer Science, and they are essential for the development of efficient software. However, 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. This paper presents work-in-progress on the development and evaluation of a serious game for teaching Binary Search Trees (BST) 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.

## CCS CONCEPTS

• Applied computing; • Education;

## KEYWORDS

Algorithms, Data Structures, Serious Games, Analogies

### ACM Reference Format:

Alberto Rojas-Salazar and Mads Haahr. 2020. Learning Binary Search Trees through Serious Games based on Analogies. In *International Conference on the Foundations of Digital Games (FDG '20)*, September 15–18, 2020, Bugibba, Malta. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3402942.3402999>

## 1 INTRODUCTION

Data structures and algorithms are core topics in Computer Science, and they are essential for the development of efficient software [25]. For this reason, software developers and computer scientists must possess a deep understanding of basic data structures and the impact that they have on computational systems performance. As an example of their importance, data structures and algorithms are topics included in the guidelines for undergraduate degree programs developed by the Association for Computing Machinery (ACM) [15]. Typically, universities teach the first introductory data structure course in the second year of their undergraduate Computer Science programs [19].

While a deep understanding of data structures is considered foundational knowledge, advanced data structures and their algorithms

are difficult topics to grasp [3]. 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 [11]. Furthermore, it is necessary that learners engage actively in the learning processes (active learning). Traditional approaches such as master class are not enough. Educators should provide experiences and environments where the students can construct knowledge through reflection, critical thinking and their previous knowledge [11]. Therefore, learning tools that help learners link new information with previous knowledge in an active way may be a useful approach to teach topics like 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. For instance, video games are quite popular among teenagers and young adults, and video game genres such as casual, action, adventure and role-playing games possess well-known game mechanics and environments. Additionally, well-crafted video games promote active learning [10]. Usually, video games offer problems in the form of 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.

This paper presents work-in-progress on the development and evaluation of a serious game for teaching Binary Search Trees (BST) 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. The rest of the article is organized as follows: In section 2, we present a review of previous games that focus on data structure learning. Then, section 3 describes the pedagogical theory on which our approach is based. Next, section 4 presents a detailed description of *DS-Hacker*. Finally, section 5 concludes this paper and discusses future work.

## 2 RELATED WORK

Serious games are becoming popular among researchers in the Computer Science field. Many games for teaching coding and Computer Science concepts have been created and evaluated. For instance, Petri and von Wangenheim [23] performed a systematic literature review and found 108 games and 117 evaluations. In another publication, Miljanovic and Bradbury [22] reviewed 49 programming serious games and their evaluations. Despite the increasing number of serious games that focus on computer science topics, a small



This work is licensed under a Creative Commons Attribution International 4.0 License.

*FDG '20, September 15–18, 2020, Bugibba, Malta*

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8807-8/20/09.

<https://doi.org/10.1145/3402942.3402999>

number of them target data structure topics [8, 22]. In this section, we present some of the most relevant serious games for learning data structures.

Among the reviewed games, the most popular data structure is the stack, and a considerable number of titles aims to facilitate learning of this data structure. For instance, Liu et al. [20] presented a game called *Star Chef* in which the player controls a chef who must attend to her clients following the stack principle: last in first out (LIFO). Additionally, the game possesses more levels that focus on other data structures such as queues and trees. Also targeting the stack, Ismail et al. [13] reported *Stack Em Up*, a serious game based on the Tower of Hanoi puzzle, and it provides three levels of difficulty. Other games based on the Tower of Hanoi puzzle are *HTML5 Hanoi Tower* [28] and the *Hanoi Tower Game* [6]. A more sophisticated game that possesses a design similar to *DS-Hacker* was reported by Dicheva et al. [7, 8]. The game, called the *Stack Game*, is an isometric action-adventure game developed in Unity in which the player takes the role of a robot that loses its spaceship in an accident. The robot must return home by walking through a path secured by locked gates. To unlock each gate, the player must solve a set of puzzles where the player must apply concepts related to stacks. Furthermore, the *Stack Game* has three levels, and each level tackles a learning objective. In the first level, the player learns the stack concepts. In the second, the player learns to use stacks to solve problems. In the third level, the player learns the implementation of the stack algorithms.

The linked list, another foundational data structure, is targeted by a considerable number of titles. Shabanah et al. [26] reported a prototype of serious game called *Singly-Linked List*. The aim of this game is to build a chain of connected nodes using the principles of the add and remove algorithms of the linked list. Costa et al. [6] modified the popular mobile game *Snake* to teach about the linked list's add algorithm. *Space Traveler* [31] is a 2D shooter-coding game where the player must code the missing sections of basic linked list operations, such as the add and remove algorithms. *Space Traveler* was designed to provide an environment where a student could practice and visualize the Linked List operations. Finally, *La Petite Fee Cosmo* is a 2D puzzle game developed by Kannappan et al. using Unity [16]. Like *Space Traveler*, *La Petite Fee Cosmo* aims to help players learn the algorithms for the basic linked list operations.

Regarding the tree data structure games, we found four games. *Elemental: The Recurrence* [5] is a 3D puzzle-coding game that intends to teach the recursive depth-first search (DFS) algorithm of a binary tree. In the game, the player must guide Ele (a programable avatar) through an environment that resembles a binary tree. To achieve this objective, the player must code the missing sections of the DFS algorithm. The second game we reviewed focuses on Adelson-Velsky and Landis (AVL) trees, and it is an adaptation of the classic game *Mario* [29]. The game has five levels, and each level covers a different concept. The AVL tree concepts covered by the game are the BST structure (level one), AVL tree rotations (level two), the AVL tree search algorithm (level three), the add algorithm (level four), and the delete algorithm (level five). The third game reviewed also targets the AVL tree data structure, and it is called the *AVL Tree Game* [27]. The *AVL Tree Game* is a mobile puzzle game that intends to teach the AVL tree rotation and add algorithms. Finally, Shabanah et al. [26] reported a game prototype

to teach the BST add algorithm. The game challenge is that the player should build a BST as fast as possible with a given set of nodes. Every time that the player adds a node in a correct place, the score increases. If the player adds a node in a wrong position, the score decreases.

Finally, some of the reviewed games are coding games that teach how to solve complex problems using data structures. For instance, *Critical Mass* [19] is a digital board game similar to *Checkers* or *Go*. Unlike traditional board games, it is designed to be played by an algorithm. In order to interact with the game, students should develop a program capable of playing the game. Then, students upload their code to a special website that allows them to challenge other students' programs. *Critical Mass* aims to motivate students to practice topics such as regular arrays, two-dimensional arrays, trees, and recursion. Similarly, *Resource Craft* [14] is a real-time strategy game that requires students to program an algorithm that plays the game. The objective of the game is to code an algorithm that maximizes the collection of resources. While playing the game, students practice the usage of arrays and dictionaries. Both *Critical Mass* and *Resource Craft* require advanced programming skills.

As shown above, several serious games focus on data structures; however, the reviewed works leave out aspects that are important to serious games for learning. One of these aspects regards the learning theories and pedagogical approaches that guide the learning process used by the game. Most of the reviewed articles only briefly mention learning principles to justify the learning process of their games. In particular, the articles do not discuss the learning principles in detail and also fail to explain how their games apply them. For instance, [13, 20, 26, 28, 31] suggest that their games may increase the motivation of the student, and consequently, their games may increase the student's engagement. However, these studies do not report the type of motivation that they expect (for example, goal-oriented motivation or task-oriented motivation), or how their games increase motivation. Another area that could be improved relates to the learning objectives that the player should achieve when playing the game. For instance, with the exception of [7], none of the reviewed articles explicitly state their learning objectives, and they only briefly explain the overall learning aim of the game. Finally, most of the reviewed games focus on basic data structures, such as the linked list and the stacks. Only some studies focus on more advanced trees and only one of these on BSTs, and this is a prototype designed to test a game engine [26]. This situation shows that there is a lack of research into serious games that target more advanced data structures, such as different types of trees and graphs.

Given the above, this article intends to improve the areas mentioned above and makes three contributions in the field of serious games for learning data structures. First, we report on a serious game that intends to teach Binary Search Trees to college students. Second, we present a detailed description of how the player will learn while playing the game and the learning theory and learning principles that support the learning process utilized by the game. Finally, we explicitly define the learning objective that the player should achieve and the learning activities that the player must perform to achieve the learning objectives. The last two contributions are crucial for any serious game or learning tool that intends to

support formal education, since the learning objectives and activities facilitate the evaluation of the learning process by defining how and what the player will learn. Furthermore, the learning objectives should guide the design and development of the knowledge assessment. Additionally, in case that the game does not obtain the expected results, it will be easier to determine the cause, e.g., whether the assessment tool is inadequate for the scope of the game, or if the learning activities are not properly aligned with the learning objectives.

### 3 PEDAGOGICAL APPROACH

Learning tools, such as serious games, should provide a means to link new information and familiar knowledge in order to create new knowledge. Additionally, learning tools should facilitate active engagement and active learning. Both features are principles of the constructivist paradigm. Constructivism suggests that human beings create and re-create their knowledge actively using their previous experiences, mental structures, and beliefs [11]. This means that educators cannot simply transfer the information to the learners, and that the learners will not understand the transferred information at the same level as the educator [11]. Instead, learners must interpret and organize their learning experience in order to create meaning and transform into knowledge.

Under the constructivist paradigm, Kolb's Experiential Learning Theory (KELT) [17, 18] suggests that experience plays a fundamental role in every learning process and that the learner should be in the center of the learning system, and it is based on six principles [17]:

1. Learning is a process.
2. Learning is re-learning.
3. "Learning requires the resolution of conflicts between dialectically opposed modes of adaptation to the world" [[17]:1216]. These modes of adaptation are action/reflection and experience/abstraction.
4. Learning is a holistic process.
5. Learning is the result of transactions between the learner and the environment.
6. Learning is a process of creation knowledge.

According to the KELT, learning occurs as cyclical or spiral process where learners transform their experience into knowledge (Figure 1). The learning cycle consists of four interrelated stages, and ideally, the learner should pass through all the stages in order to maximize the learning experience [18]. Simply put, the cycle works like this: A learner experiences a concrete or immediate experience. Then, the learner observes and reflects about it. Then, she integrates her reflections with previous knowledge and beliefs and translates those reflections into abstract concepts and theories. Finally, these abstracts concepts can be used or tested in new applications that at the same time will generate a new experience.

According to Whitton [30], video games may embody several aspects of the KELT. In video games, learning is a cycle of probing [10]. According to Gee, the player experiences the playable aspects of the video game; then, the player reflects on her experience, and then formulates theories and hypotheses from those experiences; finally, the player probes the new ideas that will provide new experiences. Another important characteristic is that video games

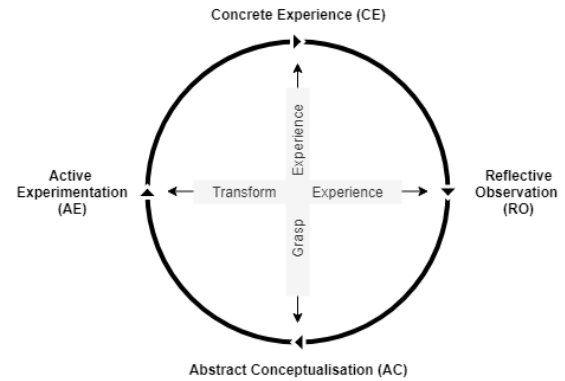


Figure 1: Experiential Learning Cycle [17]

allow learning of procedural knowledge [10]. Unlike other types of learning experiences that only provide conceptual and reflected knowledge, learning through video games can happen by means of repetition of concrete experiences and active experimentation. Finally, video games provide immediate feedback necessary for the experiential learning cycle [30].

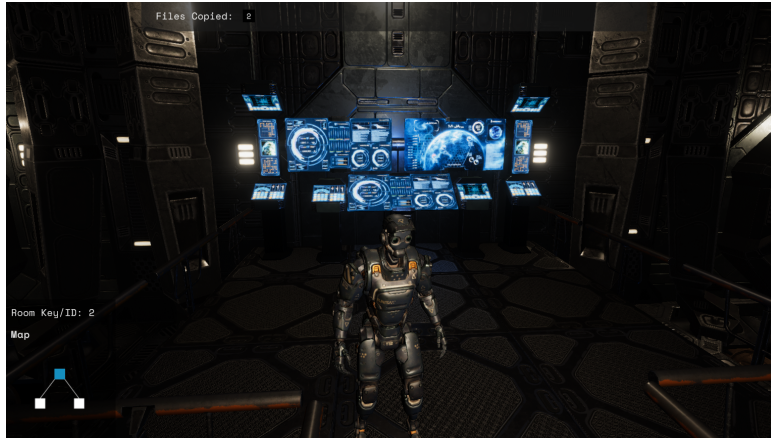
Additionally, our project uses analogies as a pedagogical approach to facilitate the construction of new knowledge. According to Duit [9], analogies are comparisons of structures between two domains; those structures must share symmetrical relations among some of their components. In educational environments, analogies are utilized extensively to create relations between non-intuitive concepts and familiar concepts. For this reason, analogies are quite popular in science education; a field of knowledge that possesses many abstract and non-intuitive concepts.

Finally, some scholars [4, 21] suggest that a good pedagogical design ensures consistency among the learning objectives, the learning activities, and the assessment. Furthermore, the objectives, activities and assessment must be aligned with a learning theory. For the design of *DS-Hacker*, we utilized the KELT; and for the formulation of the learning objectives and activities, we used the revised Bloom's Taxonomy [2]. The alignment between learning objectives and learning activities will be shown in the next section. However, the assessment is beyond the scope of this work-in-progress paper.

### 4 DS-HACKER

*DS-Hacker* (Figure 2) is a PC game developed with Unity 3D and designed to be played by college students. The game is a third person 3D action-adventure. The game's 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 on the role of a robotic hacker who must traverse a maze composed of rooms and extract information stored in each of the maze's rooms. A video file of the gameplay is available through the following link ([https://www.dropbox.com/s/8vavy0e7b9uywx6/DS-Hacker\\_Level1%26Level2.mp4?dl=1](https://www.dropbox.com/s/8vavy0e7b9uywx6/DS-Hacker_Level1%26Level2.mp4?dl=1)).

The rationale for adapting the action-adventure genre is its flexibility for teaching purposes and its popularity. The genre combines game elements from the action genre and the adventure genre. For



**Figure 2: Screenshot of the first level of DS-Hacker.**

instance, the action-adventure genre borrows the story-driven and conceptual puzzle elements from the adventure genre, and it borrows the physical challenges and fast pace from the action genre [24]. These characteristics have the potential to deliver conceptual and procedural knowledge. For instance, it is possible to introduce and explain concepts and theory through the game story narrative. Furthermore, the physical challenges and conceptual puzzles can be used to create scenarios where the player can practice the concepts delivered by the story. Finally, the action-adventure genre is a well-known genre among teenagers and young adults, and it is considered even more popular than the action and adventure genres [1]. This characteristic may decrease the learning curve of the game mechanics to potential players who are already familiar to the genre.

Regarding the game design, *DS-Hacker*'s environment reflects the structure of a BST. According to the game plot, a robotic hacker controlled by the player must enter and traverse a maze called a "Data System" (our game environment). Data Systems are places where corporations hide and protect their information, and the structure of these systems follow the same organization of well-known data structures. In *DS-Hacker*, 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 rooms 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) called Anonymous who always appears at the beginning and end of each level. Anonymous introduces the missions and the necessary BST concepts 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 example, in the first two levels, the NPC

informs the player about the relation between the game environment's structure and the BST structure. In the last three levels, Anonymous presents the relation between the game challenges and the Get algorithms. We anticipate that analogies between BST concepts and game elements may assist during the creation of new knowledge.

Currently, *DS-Hacker* has five levels, and each level focuses on different concepts of the BST data structure. Levels one and two cover topics related to the basic structure of the BST, its properties, and the structure of its nodes. Levels three, four and five cover topics related to Get 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 embody the learning activities that the player should perform. They provide opportunities to apply the concepts given by Anonymous and to experience the structure of the BST in a concrete manner. Table 1 presents the topics covered by the different levels, as well as the learning objectives and the learning activities/missions that the player must perform during a gaming session. The missions are located under each learning objective.

## 5 ANALYSIS AND FUTURE WORK

*DS-Hacker* is a serious game that intends to teach BST by applying two principles. First, the game aims to link new information with familiar knowledge. According to the constructivist paradigm, this principle is crucial for creating new knowledge. To achieve this objective, the game takes advantage of a well-known game genre (action-adventure) in order to provide environments and game mechanics that are familiar to learners. In the game, these game elements are utilized to reflect the structure of the BST. Furthermore, to facilitate the understanding of the relation of the two domains (game elements and data structure concepts), *DS-Hacker* utilizes analogies embedded in the game story. As shown above, analogies are useful for teaching abstract concepts, such as data structures, because they illustrate the relation between two symmetrical domains. We expect that by using these three approaches, the learners may create links that will facilitate the understanding of the BST concepts and the creation of new knowledge.

**Table 1: Levels, topic, learning objective (LOs) and learning activity (LAs).**

Level	Topics covered	Learning Objective (LOs) and Learning Activities (LAs)
Level 1	Binary Search Tree structure and its nodes.	LO.1. Recognize the basic structure of the BST. LA.1.1 The player must traverse all the rooms of the Data System and extract the information of each chamber. LA.1.2 The player must relate the structure of the environment and the BST structure. LO.2. List the basic elements of a BST node. LA.2.1 The player must listen and read the definitions of the node's elements embedded on the game story. LA.2.2 The player must experience the game elements (e.g., the chambers' IDs, the computer that stores the associated values, the portals that points to other rooms, etc.) that reflect the BST elements.  LO.3. Recall the BST root node definition. LA.3.1 The player must listen and read the definition of the root node embedded on the game story. LA.3.2 The player must experience the game elements that represent the root node.
Level 2	Binary Search Tree structure and property.	LO.4. Recall the binary search tree property. LA.4.1 The player must listen and read the definitions of the BST property. LA.4.2 The player must remember the BST property through the relation between the environment and the BST property. LO.5. Distinguish a binary search tree from a binary tree. LA.5.1 The player must listen and read the definitions of the binary tree. LO.6. Apply the BST property to recognize a binary search tree. LA.6.1 The player must apply the BST property to traverse the maze chambers without activating the alarms.
Level 3	Binary Search Tree Get algorithm.	LO.7. Formulate the BST Get algorithm. LA.7.1 The player must intuitively formulate the Get algorithm using the BST property to solve the level mission. LA.7.2 The player must apply the BST property to traverse the Data System without activating the portal alarms.
Level 4 Level 5	Binary Search Tree Get algorithm: Search hit and search miss.	LO.8. Recall the Get algorithm. LA.8.1 The player must listen and read the explanation of the pseudo code of the Get algorithm. LO.9. Apply the Get algorithm. LA 9.1 The player must apply the Get algorithm in order to complete the level mission. LA 9.2 The player must apply the Get algorithm to traverse the Data System without activating the portal alarms. LO.10. Recognize the two possible outcomes after executing the Get algorithm LA 10.1 At the end of the mission, the player must relate the outcome of the mission with one of the two possible outcomes of the Get algorithm.

Second, *DS-Hacker* aims to promote active learning. To achieve this, *DS-Hacker* is designed to offer a learning environment that embodies Kolb's experiential learning cycle. For instance, at the beginning of each level, the game offers conceptual information to the player. The player should use this information to plan a strategy to solve the missions. Then, the player should execute the strategy that will provide new experiences. Then, the player should consider and rationalize these new experiences, restarting the cycle. We anticipate that *DS-Hacker* ensures active learning by means of this principle.

Finally, each level of the game embodies a set of learning objectives with their respective learning activities. This characteristic determines the learning scope of the game, and it will facilitate the creation of the knowledge assessment tool.

Concerning the limitations, we foresee three major limitations. First, the interactive elements of the game, such as the aesthetics, the 3D game environment, the music and sound effects could be a distraction that may affect the learning process of the player. Interactive elements may increase engagement, but at the same time, they can increase the cognitive load [12]. Second, even if the selected game genre is popular, students who are not familiar with the genre, or with PC games in general, can experience difficulties learning the game controls and mechanics. Again, such characteristics may impede the learning process. Finally, realistic 3D videogames like *DS-Hacker* are computationally demanding. Therefore, students playing on a computer that has insufficient CPU or GPU power may experience a lag between the input system and the display system. This may be frustrating for the player, which may decrease the engagement and enjoyability of the game.

As noted, the *DS-Hacker* project is a work in progress. Currently, we are developing the assessment tools, and subsequently, we will perform an evaluation in order to measure the learning gains and the perceived engagement of the players. In addition, we will develop new levels covering more BST operations, such as the in-order traversal algorithm, and the add and remove operation.

## REFERENCES

- [1] Ernest Adams. 2014. *Fundamentals of Game Design* (3rd ed.). New Riders Publishing, Thousand Oaks, CA, USA.
- [2] Lorin W. Anderson and David R. Krathwohl. 2001. *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*. Longman, New York.
- [3] Katrin Becker and Melissa Beacham. 2000. 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, Consortium for Computing Sciences in Colleges, USA*, 65–71. Retrieved from <http://dl.acm.org/citation.cfm?id=369274.369319>
- [4] John Biggs and Catherine Tang. 2007. *Teaching for Quality Learning at University* (3rd Edition ed.). Open University Press, New York, NY.
- [5] Amanda Chaffin, Katelyn Doran, Drew Hicks, and Tiffany Barnes. 2009. Experimental Evaluation of Teaching Recursion in a Video Game. In *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games (Sandbox '09)*, ACM, New York, NY, USA, 79–86. DOI:<https://doi.org/10.1145/1581073.1581086>
- [6] Estevan B. Costa, Armando M. Toda, Marcell A. A. Mesquita, Fabio T. Matsunaga, and Jacques D. Brancher. 2014. Interactive Data Structure Learning Platform. In *Computational Science and Its Applications – ICCSA 2014*, Springer International Publishing, Cham, 186–196.
- [7] D. Dicheva and A. Hodge. 2018. Active Learning Through Game Play in a Data Structures Course. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*, ACM, New York, NY, USA, 834–839. DOI:<https://doi.org/10.1145/3159450.3159605>
- [8] D. Dicheva, A. Hodge, C. Dichev, and K. Irwin. 2016. On the design of an educational game for a Data Structures course. In *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 14–17. DOI:<https://doi.org/10.1109/TALE.2016.7851763>
- [9] Reinders Duit. 1991. On the role of analogies and metaphors in learning science. *Science Education* 75, 6 (1991), 649–672.
- [10] James Paul Gee. 2007. *What Video Games Have to Teach Us about Learning and Literacy* (2nd ed.). Palgrave Macmillan, New York.
- [11] Aytac Gogus. 2012. Constructivist Learning. In *Encyclopedia of the Sciences of Learning*, Norbert M. Seel (ed.). Springer US, Boston, MA, 783–786. DOI:[https://doi.org/10.1007/978-1-4419-1428-6\\_4049](https://doi.org/10.1007/978-1-4419-1428-6_4049)
- [12] Celia Hodent. 2017. *The Gamer's Brain: How Neuroscience and UX Can Impact Video Game Design*. CRC Press, Boca Raton, FL.
- [13] Marina Ismail, Najwa Ghafar, and Norizan Diah. 2013. Realization of Conceptual Knowledge Through Educational Game. DOI:[https://doi.org/10.5176/2251-1679\\_CGAT13.06](https://doi.org/10.5176/2251-1679_CGAT13.06)
- [14] H. C. Jiau, J. C. Chen, and K. Ssu. 2009. Enhancing Self-Motivation in Learning Programming Using Game-Based Simulation and Metrics. *IEEE Transactions on Education* 52, 4 (November 2009), 555–562. DOI:<https://doi.org/10.1109/TE.2008.2010983>
- [15] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. ACM, New York, NY, USA.
- [16] V. T. Kannappan, O. N. N. Fernando, A. Chattopadhyay, X. Tan, J. Y. J. Hong, H. S. Seah, and H. E. Lye. 2019. La Petite Fee Cosmo: Learning Data Structures Through Game-Based Learning. In *2019 International Conference on Cyberworlds (CW)*, 207–210. DOI:<https://doi.org/10.1109/CW.2019.00041>
- [17] Alice Y. Kolb and David A. Kolb. 2012. *Experiential Theory Learning*. In *Encyclopedia of the Sciences of Learning*, Norbert M. Seel (ed.). Springer US, Boston, MA, 1215–1219. DOI:[https://doi.org/10.1007/978-1-4419-1428-6\\_4049](https://doi.org/10.1007/978-1-4419-1428-6_4049)
- [18] David A. Kolb. 2014. *Experiential Learning: Experience as the source of learning and development* (2nd ed.). Pearson, New Jersey.
- [19] R. Lawrence. 2004. Teaching data structures using competitive games. *IEEE Transactions on Education* 47, 4 (November 2004), 459–466. DOI:<https://doi.org/10.1109/TE.2004.825053>
- [20] T. Liu, Y. Chu, and T. Tan. 2012. Using computer games in a computer course to improve learning. In *Proceedings of IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) 2012, W2C-16-W2C-19*. DOI:<https://doi.org/10.1109/TALE.2012.6360301>
- [21] T. Mayes and Sara de Freitas. 2004. Review of e-learning theories, frameworks and models. Joint Information Systems Committee, London.
- [22] Michael A. Miljanovic and Jeremy S. Bradbury. 2018. A Review of Serious Games for Programming. In *Serious Games*, Springer International Publishing, Cham, 204–216.
- [23] Giani Petri and Christiane Gresse von Wangenheim. 2017. How Games for Computing Education Are Evaluated? A Systematic Literature Review. *Comput. Educ.* 107, C (April 2017), 68–90. DOI:<https://doi.org/10.1016/j.compedu.2017.01.004>
- [24] Andrew Rollings and Ernest Adams. 2003. *Andrew Rollings and Ernest Adams on Game Design*. New Riders Games, USA.
- [25] Robert Sedgewick and Kevin Wayne. 2014. *Algorithms* (4th ed.). Addison-Wesley.
- [26] S. S. Shabanah, J. X. Chen, H. Wechsler, D. Carr, and E. Wegman. 2010. Designing Computer Games to Teach Algorithms. In *2010 Seventh International Conference on Information Technology: New Generations*, 1119–1126. DOI:<https://doi.org/10.1109/ITNG.2010.78>
- [27] D. Šuniková, Z. Kubincová, and M. Byrtus. 2018. A Mobile Game to Teach AVL Trees. In *2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 541–544. DOI:<https://doi.org/10.1109/ICETA.2018.8572263>
- [28] Daniel Vasić, Emil Brajković, and Tomislav Volaric. 2014. Experimental evaluation of teaching recursion with HTML5 game. DOI:<https://doi.org/10.13140/2.1.1669.2481>
- [29] D. Wassila and B. Tahar. 2012. Using serious game to simplify algorithm learning. In *International Conference on Education and e-Learning Innovations*, 1–5. DOI:<https://doi.org/10.1109/ICEELI.2012.6360569>
- [30] Nicola Whitton. 2014. *Digital Games and Learning: Research and Theory*. Routledge, New York.
- [31] J. Zhang, M. Atay, E. R. Caldwell, and E. J. Jones. 2015. Reinforcing student understanding of linked list operations in a game. In *2015 IEEE Frontiers in Education Conference (FIE)*, 1–7. DOI:<https://doi.org/10.1109/FIE.2015.7344132>