

Muse: A Musically Inspired Game To Teach Arrays and Linked Lists

Vaishali SHARMA^{1*}, Raghib MUSARRAT^{2*}, Sridhar CHIMALAKONDA³,
& Y Raghu REDDY⁴

¹*Indian Institute of Information Technology Sri City*

²*Indian Institute of Information Technology Sri City*

³*Indian Institute of Technology Tirupati*

⁴*International Institute of Information Technology Hyderabad,*

¹vaishali.s15@iiits.in, ²raghib.m14@iiits.in

Abstract: Data structures is an important area of knowledge in computer science that is often difficult to understand for high school students. There has been a lot of research in computer science education around the subject of programming. Hence, it becomes imperative to explore the domain of data structures as well. With this work, we have looked into an active pedagogical approach for facilitating learning of data structures in school. This approach has been implemented by means of an interactive game. The game aims to teach arrays and linked lists through activities based on the principle of compare and contrast. The innovation in the game is the use of musical notes to teach operations in arrays and linked lists. The game was evaluated with 27 students across 8th, 9th and 10th grades using a questionnaire. The results of the evaluation showed that the group of students who learnt about arrays and linked lists through the game scored an average of 81.5% of the total marks while the other group scored an average of 72% of the total marks.

Keywords: Active learning, Data Structures, Gamification

1. Introduction

In the field of computing, data structures is a fundamental course that lays a critical foundation for the rest of the curriculum and is a solid basis for industry projects. A sound understanding of data structures, hence is indispensable for any student aiming to pursue a career in this field, which is supported by many competitive programming websites such as *TopCoder*, *CodeChef*, *HackerRank*.

Data structures course is perceived to be a difficult subject for students, particularly the ones in high school because it is difficult for them to visualize how the data structures are structured and analyze its properties; hence they do not get the required motivation to pursue it in the way it should be (Luo, X., & Liu, B., 2011). So, there is an ardent need to improve the understanding of data structures amongst the students. Researchers have addressed this problem from several perspectives. There have been some theories and proposals which suggest that learning experience can be improved with the adoption of gamification in educational practices (Budd, T. A., 2006). This theory was applied to a competitive game for teaching data structures (Lawrence, R., 2004). Furthermore, there was another solution which addressed this concern by using games which explained the basic functionality of some of the data structures like arrays, linked lists, stacks, queues and trees (Costa, Toda, Mesquita & Brancher, 2014). Another line of research used a combination of deep parametric abstraction techniques with interactive abstraction manipulation to provide a better understanding of the details of the operations done on data structures (Ou, Vechev & Hilliges, 2015). In this paper, we experiment with the use of music to teach data structures through an innovative game, specifically focusing on using the principle of compare and contrast to teach arrays and linked lists.

We discuss an overview of related work in Section 2. We present a theoretical basis for the game in Section 3 and elaborate the design and development of the game in Sections 4 and 5 respectively. We discuss evaluation in Section 6 followed by discussions and conclusion.

2. Related Work

Gamification in education has been understood as the addition of game-like rules, interactivity, user experience to the educational setting. It has shown positive results in quite a number of domains like marketing, politics, health and fitness. Gamification can have major influences on the users in three aspects - cognition, emotion and social (Detering, Khaled, Nacke & Dixon, 2011). The 'learn by exploration' feature in games broadens the player's perception about learning, besides teaching the users the process of thinking (Detering, Khaled, Nacke & Dixon, 2011). In the game proposed in the paper, this feature of games is exploited to explain to the users the technicalities of the operations on arrays and linked lists. The user learns how to perform a certain operation on an array or a linked list right from the scratch through the method of trial and error and the element of music in it keeps the user interested throughout the process of learning. This is explained in Section 4.

Games invoke a variety of emotional experiences - from frustration to joy. Hence, students learn to see failure as an opportunity and in turn try to find out the cause of the error instead of feeling helpless about it (Lee & Hammer, 2011). This feature of gamification grips them up with a better understanding and command over the subject. We exploit this idea in the proposed game as discussed in Section 4. Lee et al. proposed the use of game-like rules system, player experiences and cultural roles to shape learners' behavior (Lee, & Hammer, 2011). It pointed out three major areas where gamification could serve as an intervention - cognition, emotional and social. The authors asserted that well-designed gamification system could help students think about schools and the process of learning differently. Denny conducted a study on the effect of a badge based achievement system on the user participation, with two groups (Denny, 2013). The activity was designed on the lines of the approach of "student pedagogy" where the students take responsibility for creating and moderating learning resources and producing peer feedback. The result was that the group having access to the badge system outperformed the other group.

Ramon Lawrence designed a method which used competitive programming with game development for an introductory data structures course (Lawrence, 2004). This method was with a class of 55 students implement the game intelligence functionality for a board game. A user survey found that 40% of the surveyed students logged onto the game site more than 20 times, and 88% logged on at least six times; thus reflecting increased motivation in the students to learn and explore more. Estevan B. Costa et al. demonstrated an interactive web learning environment, called DSLEP (Data Structure Learning Platform) to teach some of the data structures like stacks, queues, lists, arrays and trees (Costa, Toda, Mesquita & Brancher, 2014). Decker and Lawley created a game based achievement system for the undergraduate freshmen of the Computer Science course (Detering, Khaled, Nacke & Dixon, 2011). The primary intention of the system was to increase students interaction with one another, thereby promoting teamwork while trying to achieve a particular task of the game and gaining points or awards for the same. The result of this experiment was that a record-breaking 91% of the class showed an improvement in their performance.

However, to the best of our knowledge, we could not find any related work that uses music to teach data structures, which is the core idea proposed in this paper.

3. Theoretical Basis

Designing learning environments and games without theoretical basis is a futile effort (Goodyear et al., 2004). The designed game is based on active learning which can enable students to read, write, discuss, and engage in problem-solving and to maximize their potential for intellectual growth (Slavich, & Zimbardo, 2012). *Muse* can be considered as a practical application of this approach that teaches the operations on arrays and linked lists through music. There is a mapping between the musical notes and their corresponding tones (SouthCalMusic), (Do Re Mi Isn't Just Child's Play: How Solfege Training Can Improve Your Ear). This mapping is exploited for explaining the concepts of arrays and linked lists as shown in Table 1. The implication of this mapping is that different sequences of musical notes produce different tunes. Also, it is implicit that the various operations on arrays and linked lists of musical notes manipulate the sequence of notes differently. So, this difference in the sequences or the effect of the operations is brought out through different tunes. Hence, the similarity and the difference in the musical tunes is analogous to the difference in

arrays and linked lists.

Table 1: Mapping of the concepts in arrays and linked lists with music.

Concepts	Arrays	Music	Linked List	Music
Accessing an element	Element at any index of the array can be directly accessed.	Directly play the desired musical note.	The desired element can be accessed in a sequential manner.	All the notes up to the desired note will be played in order.
Inserting an element	Cannot insert an element at a given position because the size of the array is fixed .	The musical sequence cannot be altered here.	An element is inserted by first traversing to the desired position; then the appropriate links are made.	The new melody will have the sound of the inserted note as well.
Deleting an element	Cannot delete an element directly at a given position because the size of the array is fixed.	The musical sequence cannot be altered here as well.	Traverse to the position of the desired element; alter the appropriate links.	The new melody will not have the sound of the deleted musical note.

The way music is used here can be aligned with the principle of compare and contrast. It is this principle that forms the basis of the activities designed in the game to demonstrate the concepts. It has also been assessed that teaching using the principle of compare and contrast builds on the critical thinking skills of the learner naturally (Faisal, 2006). The fact that comparative thinking is a natural form of reasoning, this principle was an apt choice for the game explained in the paper.

4. Design of Game

The game teaches the various operations on arrays and linked lists along two levels, each of which discusses operations on a particular structure - the first level teaches about the operations on arrays while the second one deals with the operations on lists. The game focuses on three operations namely *update*, *insert* and *delete*. At each level, the player is asked to do some task associated with a certain operation and a certain structure. On completion of the task, the next level unlocks and the player moves up. The different operations of arrays and linked lists in the game are shown in Figure 1, Figure 2 and Figure 3.

The task to be performed at every level of the game is to reproduce a given musical tune in terms of its sequence of Solfege syllables - *Do, Re, Mi, Fa, So, La, Ti*. The player does this by rearranging the musical blocks using the allowed operations. Music serves the purpose of an error finder in the game. At each level, the player is given a reference to the tones of each of the seven Solfege syllables, with the help of which the blocks should be arranged in the correct order based on

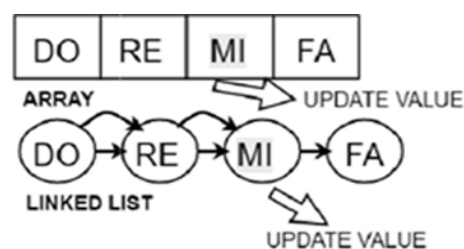


Figure 1: Updating notes in array, linked list.

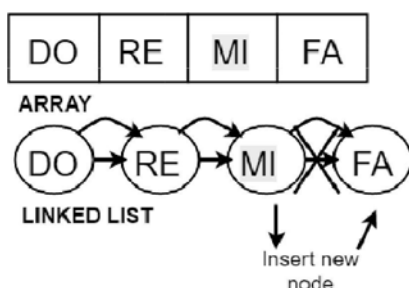


Figure 2: Inserting notes in array and linked list

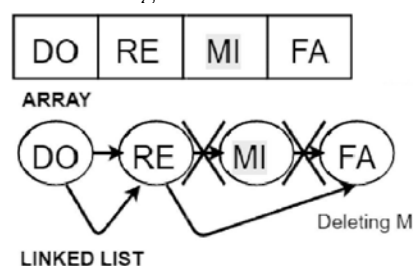


Figure 3: Deleting notes in array and linked list.

4.1 Operations on Arrays

The scene in the game opens up with a musical tone that needs to be reproduced by the player to proceed to the next level. The scene also has some musical blocks with addresses displayed, arranged in certain order. The number of musical blocks in the sequence is fixed. This implies that the memory allocated to arrays is fixed and cannot be changed later. To access any block, the players can specify the address they want to jump to and then, jump to the block at that location. Using the given three operations: insert, delete, update, the player needs to reproduce the given sequence. In the case of *update* operation, the players can "jump" to any element in the array which they want to change; then update it with the desired musical note from the available seven musical notes as shown in Figure 4. The ability to "jump" given to the player here relates to the fact that any element of an array can be accessed straightaway by referring to its index. The next operation is *insert* operation. For this operation, if the players try to insert any musical note, they are prompted with a message stating that the array is of fixed size; hence insertion of any musical note is forbidden. The last operation is *delete* operation. The players, when try to delete a note in an array by jumping to its position, they are prompted with a message which says that deleting a cell of memory directly in an array is forbidden because the size of the array is not allowed to be changed.

4.2 Operations on Linked Lists

The setting in this mode is same as in the "array" mode of the game. However, the players, here cannot jump to a block because the addresses of the blocks are not revealed to them. To access any block, they walk to the first block, which then forwards it to the second block and this continues until they reach the desired block. A scene in the linked list level of the game is shown in Figure 5. In this Figure, the student (player) is represented as a person holding a sword and has to rearrange the blocks using the insert operation till the sequence plays the correct music; the green links mean they are linked to another musical block while the grey ones mean that they aren't. The player has to insert the appropriate block and make links from the previous block to the current block and from the current block to the next block. In the case of *update* operation, they need to traverse in steps of one from the first musical block to the desired one; then replace it by selecting the appropriate note from the available seven notes. The next operation is the *insert* operation. For this operation, the sequence of musical blocks has some holes in it. Now, the player has to insert appropriate musical block in the holes such that the sequence sounds same as the given musical tune. Once the players reach the desired hole, they can insert the appropriate block from the available seven notes. Next, they make the link between the previous block - the current block and the current block - next block. For the *delete* operation, the player has to change the sequence of the musical blocks using the delete operation only so that it produces the given musical tune. Once the players reach the desired block to be deleted, they can simply can pull it away from the list and then make links between the previous block and the next block of the currently deleted block. The player finishes the task when the sequence in the linked list plays the musical tune specified in the task.



Figure 4: Updating elements in array.

5. Development

The development of the game began with the contextualization phase, which consisted of creation of activities for each of the operations on arrays and linked lists. The activity was sketched out on the principle of compare and contrast. Such an activity created an apt setting for the implementation of the concept of active learning. As for the gamification concepts, a brief explanation of it can be seen below (i) *Stages*: Gained by finishing activities (ii) *User Profile*: Used to see the statistics of utilization of the platform. The application is developed using *Unity*, a cross-platform game engine.

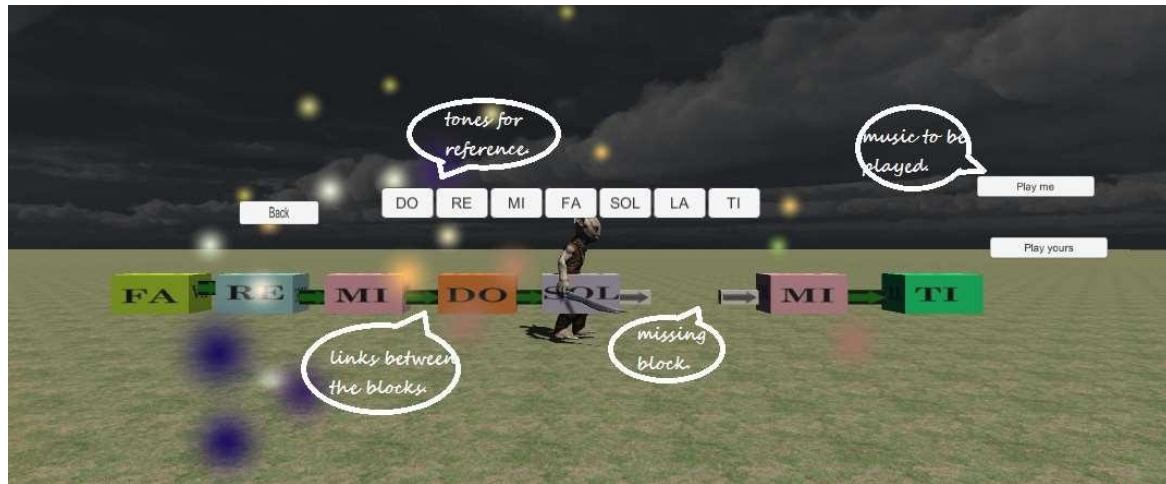


Figure 5: Inserting missing blocks in linked list.

6. Evaluation

The evaluation for the game was carried out in the school, Chinmaya Vidyalaya at Sri City, Andhra Pradesh, India. Students from grades 8th, 9th and 10th played the game and their performance was tested through a questionnaire. The questionnaire was reviewed by two teachers of the school. A total of 27 students participated in the evaluation of the game. It was ensured that the only those students participated who did not have any prior knowledge of arrays and linked lists for a fair evaluation of the hypothesis. The students were divided into two groups - one was taught arrays and linked lists through the traditional method of chalk and talk while the other was taught through the game. 14 students were in the first group while the remaining 13 were in the other group.

The effectiveness of the game was analyzed through a questionnaire on arrays and linked lists. The questionnaire had 10 questions on the operations on arrays and linked lists. Some of the questions that were included in the questionnaire were – *You have an array $A = \{2,3,1,4,6,7\}$. Update the third element to 8 and write the resultant array. You have a linked list $B = 2 \rightarrow 3 \rightarrow 5 \rightarrow 9 \rightarrow 10$. Insert 7 at the third position and write the new linked list.* The result was that the group which learnt arrays and linked lists through the game scored an average of 81.56% of the total marks. The other group had an average of 72.2% of the total marks. A more detailed analysis of the evaluation could not be included due to space constraints. The students were also interviewed about their experience with the game. The response was a positive one and many of them were eager to play the game again. Some of them were positive about the idea of exploring other data structures through games.

7. Discussion

The problem of teaching operations on arrays and linked lists can be addressed via a variety of approaches including games. The idea of illustrating the concept of arrays and linked lists through a music-based game is substantiated by certain arguments. First, there is a clear mapping between the Solfege syllables of music, i.e. *do, re, mi, fa, sol, la, ti* and their corresponding tones. In addition to

this, music can be used as an error checker because the error in music can be easily noticed. This in turn helps the players to debug through the steps they take while doing the specified task in the game. Second, using music can promote user engagement and makes the learning interactive.

Our approach and game have certain limitations. The mapping of musical notes to computing concepts is based on our intuition as we did not have a musical expert in our group as of now. However, we plan to work with a musicologist to further strengthen our approach. We limited our evaluation initially with a set of 27 students but we plan to extend it with more subjects.

We see that the game can be extended to include other data structures, activities and to even other courses in computing. As a short-term future work, we plan to design activities for some of the other data structures like stack, queue, trees, heaps etc. This game could also be extended to teach some of the simple sort and search algorithms like binary search, selection sort, insertion sort etc. Based on the initial game proposed in this paper, we are also keen to explore the idea of combining music, storytelling and gamification to teach computing to school students.

8. Conclusion

The paper presented *Muse* as an interactive and innovative game that uses music to teach arrays and linked lists to high school students based on compare and contrast principle. The game discussed how the sequence of musical notes could be rearranged to bring out similarities and differences between arrays and linked lists. The results of evaluating the game with 27 students were promising and also show promise that the game can be extended for other data structures like stack, queue, trees, heaps and so on. The broader impact of this game was to explore the possibility of using the natural interest of music listening of people to visualize how data structures are structured and ultimately, motivate them to develop innovative applications inspired from the fundamentals.

References

- Budd, T. A. (2006). An active learning approach to teaching the data structures course. *ACM SIGCSE Bulletin*, 38(1), 143-147.
- Costa, E. B., Toda, A. M., Mesquita, M. A., & Brancher, J. D. (2014). DSLEP (Data Structure Learning Platform to Aid in Higher Education IT Courses). *World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 8(4), 1143-1148.
- Denny, P. (2013, April). The effect of virtual achievements on student engagement. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 763-772). ACM.
- Detering, S., Khaled, R., Nacke, L. E., & Dixon, D. (2011). Gamification: Toward a Definition. In *Proceeding of the CHI 2011 Gamification Workshop, Vancouver, Canada*.
- Goodyear, P., Avgeriou, P., Baggetun, R., Bartoluzzi, S., Retalis, S., Ronteltap, F., & Rusman, E. (2004, April). Towards a pattern language for networked learning. In *Networked learning* (pp. 449-455).
- Lawrence, R. (2004). Teaching data structures using competitive games. *IEEE Transactions on Education*, 47(4), 459-466.
- Lee, J. J., & Hammer, J. (2011). Gamification in education: What, how, why bother? *Academic exchange quarterly*, 15(2), 146.
- Luo, X., & Liu, B. (2011). Educational and Learning Technologies for "Data Structure" Course. In *Education and Educational Technology* (pp. 381-387). Springer Berlin Heidelberg.
- Ou, J., Vechev, M., & Hilliges, O. (2015, April). An interactive system for data structure development. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 3053-3062). ACM.
- Faisal, K. A. (2006). Infusing Critical Thinking Skill Compare and Contrast into Content of Data Structures Course.
- Slavich, G. M., & Zimbardo, P. G. (2012). Transformational teaching: Theoretical underpinnings, basic principles, and core methods. *Educational Psychology Review*, 24(4), 569-608.
- "Do Re Mi Isn't Just Child's Play: How Solfege Training Can Improve Your Ear". url{<http://www.easyeartraining.com/learn/do-re-mi-isnt-childs-play/>}. Accessed May 14,2017.
- "SouthCalMusic". url{<http://www.southcalmusic.com/solfege.php>}. Accessed: May 16,2017.