Augmented Reality for Teaching Data Structures in Computer Science

Husnu S. Narman, Cameron Berry, Alex Canfield, Logan Carpenter, Jeremy Giese, Neil Loftus, and Isabella Schrader
{narman}{berry138}{canfield17}{carpenter190}{giese}{loftus6}{schrader13}@marshall.edu
Department of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV 25755

***Abstract*—Data structures course is the most essential and critical course for computing-related majors. In this course, data structures, their main differences, and their usages are explained. However, for computer science students and especially students who do not have programming experiences, learning data structures can be challenging, and many students change their majors if they are not successful in this course. Given the importance of data structures, it is a pressing issue that we need to work to meet student needs and improve computer science education in this area. With technological developments, teaching strategies based on technology are also varying. The student performance is improved at various levels with these strategies. Also, accessing lower price smart devices with lower prices makes previously inaccessible technology, like Augmented Reality (AR), more readily available for students. Therefore, in this paper, we have developed a practice and learning environment for students based on AR and compared AR approach with standard and web-based animation approaches according to students' comments. Results show that the AR approach not only helps students learn data structures better but is also found to be the most exciting approach. The results can help content-developers observe benefits of AR in computer science education.**

***Index Terms*—Augmented reality; teaching; computer science; data structures**

I. Introduction

The data structure course is the most essential and critical course for computing-related majors because it is one of the fundamental courses that needs to be taken by students. However, learning data structures and understanding the performance effects of distinct data structures on the same problem can be challenging. Mostly, students change their majors if they are not successful in this course in their first attempt. However, changing majors has a significant effect on the employment and economy. For example, the workforce requirements for computing-related jobs is increasing, according to the U.S Bureau of Labor Statistic report [1]. The number of available computing jobs through 2024 will be half a million, and the total number of available computing-related jobs will be more than one million [1]. Despite this, enrollment and graduation rates from computing-related disciplines have not increased to fulfill industry needs. This is largely because more than 59% of Computer/Information Sciences students either leave without obtaining degrees or change their majors to non-STEM fields [2], usually in the first year when data structures, programming and OOP courses are offered. In order to have a strong workforce and maintain economic growth, there is a critical need to identify effective teaching strategies for teaching fundamental concepts to students. In the absence of such strategies, finding required applicants for computing jobs will continue to be challenging and will likely impact U.S. economic growth negatively. Indeed, it is much more essential to have online and self-learning supportive teaching strategies due to COVID-19 now because many schools use online teaching platforms to deliver lectures upcoming semesters.

In the literature, there are many researchers to create either content, strategies, or support tools to improve the teaching of programming and data structures courses [3]. The computer-aided instruction in the course helps students understand the contents, especially abstract information [4], [5]. Therefore, several tools and visualization techniques have been developed to improve teaching data structures [6]–[16]. In [6], teachers and students develop Networked CAI Teaching System for data structures to engage students. In [7], students are encouraged to use JGRASP [17] to develop their programs in the courses so that students can easily write and debug their programs by visualizing components of programs such as data structures. In [8], an interactive e-book OpenDSA [18] is used, and students interactively use OpenDSA rather than traditional textbooks and work actively with the material on OpenDSA. In [9], BRIDGES, software infrastructure is designed to enable the creation of more engaging assignments for data structure students and allows them to create and use data structures with real-world data sets, such as those from popular social networks (e.g., Twitter, Facebook). In [10], two video series have been created for Binary Search and Selection Sorting algorithms and their code analysis to help students understand the concepts. In [11], the algorithm analysis visualization is used to engage students. In [12], [16], games are used to introduce the implementations of data structures such as Stack and Queue. In [13], visually interactive content similar to interactive books and that can be integrated into OpenDSA [18] are used to teach data structures. In [14], a domain-specific language for developing rich, interactive explanations of data structures and algorithms is created. In [15], a logical-mathematical object-oriented framework is used to help students understand the problem solution, by neither producing ambiguity nor uncertainty. There are also online web applications such as [19], [20] to visualize operations of data structures to teach data structures to students. All the above approaches are beneficial to students to understand the concepts of data structures.

Moreover, there are other approaches to improve programming skills before and during college education. Some of the more widely used tools for beginners are MIT App Inventor [21], Sonic PI [22], Alice [23], [24], Scratch [25] and ScratchJr [26], Made with Code [27], Tynker [28], Lego [29] with EV3 Programming, Vex [30] with Modkit and RobotC, Arduino [31], Humming Bird [32], Finch Robot [33], CoderZ [34], Blockly Games [35], Make Wonder [36], and even Minecraft [37]. There are other tools such as Robot Turtles [38], Kodable [39], CodeCombat [40], Kodu [41] Lightbot [42], Game Star Mechanic [43], Sphero [44], Capture the flag [45] etc. The purpose of all of the above tools is to motivate and stimulate visual interest in learners for coding activities. Every year thousands have an opportunity to practice and experience coding by using aforementioned or similar tools.

However, our approach is different from above works that combines real-world environment with virtualization by using Augmented Reality (AR), and then we observe the effects of AR by comparing AR-based visualization approach with the pure lecturing and lecturing with animations. AR is defined in [46] as “3D virtual objects are integrated into a 3D real environment in real-time”. The *objective* of this paper is to develop AR visualization contents for data structures and analyze the effects of AR-based visualization and animation techniques for teaching data structures course. The key *contributions* of this paper can be listed as follows:

- We first developed an AR application that visualizes some operations of data structures such as Array list, Linked list, and Stack. In order to develop an AR application, we have followed AR requirements such as integration of virtual objects or environment into the real environment; and real-time interactivity of the virtual and real objects at the same time.
- Then, we analyzed the effects of AR visualization on student engagement by comparing it with pure lecturing and lecturing with animations. We have used VisuAlgo [19] for visualized animations of data structures. We selected VisuAlgo [19] rather than other data structure visualization techniques in previous papers and online like [20] due to lecturing mode, modifiable examples mode, and explanations of code steps for each operation.

The *results* show that the AR application is more attractive to students than other methods and can be used effectively as an interactive aid to explain complex, abstract subjects in the classroom.

The rest of the paper is organized as follows: In Section II, the system model and the design of the developed application are explained. In Section III, the experiment with analysis is presented. Section IV has the concluding remarks, and finally, Section V includes the possible future improvements.

II. System Model and Design of the Application

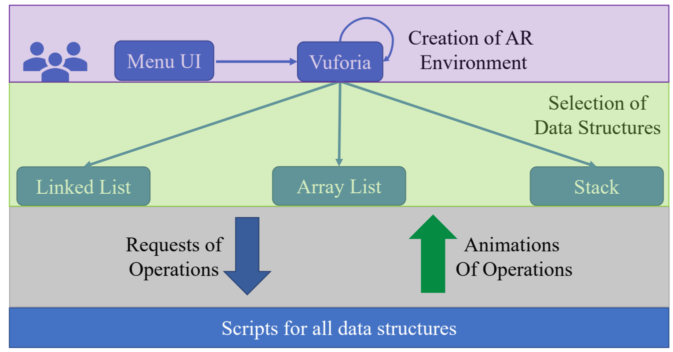

Fig. 1: The model of the developed augmented reality application.

Fig 1 shows the system diagram for how the application works on smartphones. Initially, the user interface collects information about what type of data structure the user wants to use. With this information, the Vuforia software [47] then creates an AR environment in which our code will display and animate data structures through the camera of users’ devices. As shown in Fig 1, from this point, the AR application then moves into the desired data structure module, whether that be Linked list, Array list, and Stack. The data structure interface then accepts requests from the user and passes those requests to the underlying code or “Script” that is waiting for requests. The script then processes the request by doing operation an element from the data structure and sends a message back to the interface when the animation of the request is completed. The Linked list, Array list, and Stack operations are broken into three completely separate modules. With this, developers can operations such as add and remove to/from a data structure without affecting the others. This also allows for the easy addition of more data structures in the future without affecting previous work.

A. Tools and Augmented Reality

Our developed application uses the Vuforia [47] software to handle the augmented reality functionality. The application operates as image recognition AR, which means that the application needs a specific image to aim the camera for the AR objects to be displayed appropriately. We use the Marshall University logo as our “target image”. When the user points their camera at this image, Vuforia recognizes the image, and the application can correctly place the data structure objects on top of the image, as if in real space.

The AR application was built using the Unity platform [48]. This is traditionally thought of as a software used to make games, but the built-in animation features made for easier development of animated data structures. Unity also has built-in augmented reality support with Vuforia software package [47] to handle the augmented reality side of the project. Unity also allows for the cross-platform deployment; thus, the developed application is available on both Android and iOS with minimal changes.

B. User Interface

Fig. 2 shows the main menu of the developed AR application. The main menu and sub-menus are kept as simple as possible to decrease user interface complexity. Users can select one of the data structures from the main menu to experience the concepts.

Fig. 2: The main menu of the developed augmented reality application.

C. Data Structures with Augmented Reality

In this section, some of the data structures which are used in AR are explained.

1) Linked List: A Linked list is a list of elements that are held together by "pointers" or "references", each of which points from one object in the list to the next. When objects are added or removed to this Linked list, the pointers have to be manipulated to allow this operation. For new programmers who implement Linked list, it can be very easy to misplace a pointer and accidentally cut off half or more of the list. The Linked list module in our application is designed to help a student visualize the Linked list by displaying elements and pointers. As elements are added and removed from the list, students see how these pointers are moving in the list such that no elements are lost. Although there are many kinds of Linked list, we have visualized the single Linked list, which is a base of understanding other types.

Fig. 3 shows how a user can add and remove data from anywhere within their list in the developed application. To represent data, we used "objects" in the form of shapes. Shapes can have different sizes, colors, and forms. Specifically, we offer spheres, cubes, and cylinders in red, blue, and green. When a user adds an element to their list, they choose what shape they want to add in which position. This allows the user to visualize the Linked list as not just storing data, but storing objects, a critical Object-Oriented Programming concept. To illustrate the pointers that connect objects on a Linked list, we used arrows pointing from one object to another. When objects are added and removed, arrows are shifted, added, and removed just as the pointers would be in an actual Linked list. The arrows in this visualization precisely mirror the movement of pointers in a Linked list. The user also tracks iteration over the visualized application to understand the runtime of each operation. The Linked list also shows the location of the "Head" of the list. The Head dynamically moves with the first element of the list to illustrate this concept.

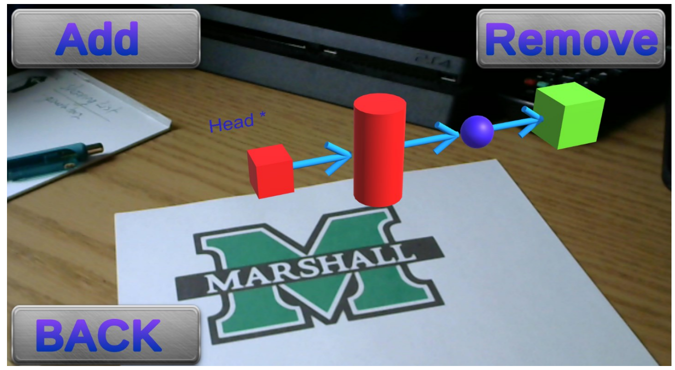

Fig. 3: The screen capture of the Linked list Module of the developed augmented reality application.

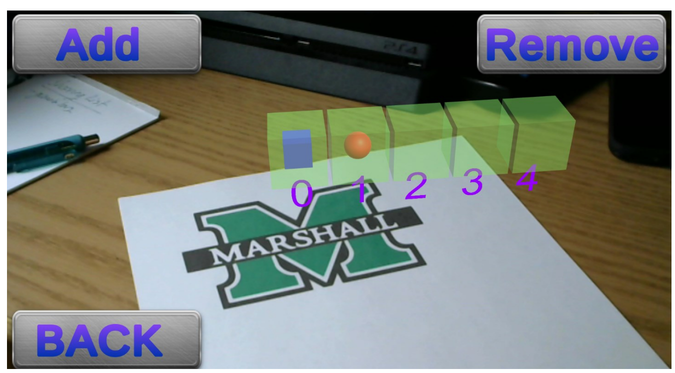

Fig. 4: The screen capture of the Array list Module of the developed augmented reality application.

2) Array List: The Array list is often one of the first data structures that students learn. The Array list is a powerful tool for students who are new to programming, especially when learning the Java programming language. Most students start using the Array list long before they understand how the Array list works. Understanding the Array list's inner workings and being able to develop one on their own is an essential task in learning data structures.

Unlike the Linked list, which is truly dynamic, the Array list simulates being dynamic with a layer of abstraction on top of a static array. This static vs. dynamic attribute is one of the main focuses of the Array list visualization. The primary concern with visualizing an Array list is the importance of showing that beneath an Array list is a static, typical array that every student is familiar with. To show this, we use transparent green cubes to represent the "spaces" in an array, as shown in Fig 4. There are no pointers in this module, like in Linked lists, since elements in Array lists simply occupy adjacent memory locations. As students add objects to the Array list, objects move into these cubes just as they would populate the array beneath an Array list. For removing items, it was important to show that items in the array have to shift to keep their order correct. For example, if a user removes an item from the middle of the list, the rest of the items shift to fill in the empty space while maintaining the order of the list. While this does not show the student any code, it puts the idea in their mind that when they remove an item, other items will need to shift to fill in the array.

Another important visualization that this module provides is the resizing of an Array list. An Array list does not have a limit to how many items it can hold, but the underlying array does have a limit. Resizing is the key concept to the unrestricted size of an Array list. When the underlying array reaches max capacity, another array is created that is double in size. Each element is then moved into the new, larger array, and the old array is thrown away.

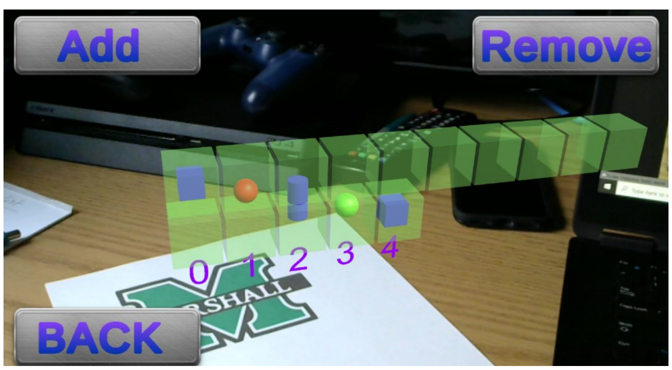

Fig. 5: The screen capture of the Array list resizing of the developed augmented reality application.

Fig 5 shows how this module illustrates this scenario by creating an array that is double the size of the current one. The student can then observe as the data objects move from the old array into the new one. This gives the student an idea about how the Array list can achieve dynamic resizing by using a static array.

3) Stack: Stack is a simple data structure, but it has many vital applications in computer science. In a sense, a Stack can be thought of as literally stacking objects one on top of the other. With a Stack, there are two primary operations: push and pop. Push is placing an object on top of Stack, and pop is taking an object off the top of Stack. In this sense, Stack follows a "First In, Last Out" pattern.

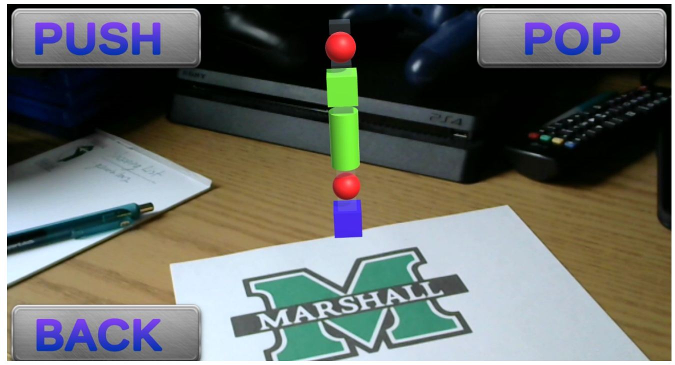

Fig. 6: The screen capture of the Stack Module of the developed augmented reality application.

The stack module creates a vertical tower of objects, as shown in Fig 6. As students "push" objects onto Stack, the tower gets taller, and students can remove objects from the top of the tower with the "pop" method. Stack is also the only module that simulates gravity. The objects are assigned mass and literally fall onto the top of Stack with a small spacer in between each element.

III. Analysis and Experiment

To gauge the effectiveness of the AR-based application, we conducted a study with beginner computer science students. Specifically, this study investigated how well the application engaged students with data structure topics.

A. Evaluation Methodology

To conduct this study, we used a group of 13 computer science students who had never been taught data structures. These students were freshmen computer science majors, so they had not yet learned data structures. Many of these students had used data structures before, such as Array list, but did not know how the data structures worked.

The class was split into three separate groups, and each group was lectured by the same lecturer for the entire time of teaching. This eliminated the effects of the lecturers as a variable in our results since each group dealt with the same lecturer for each data structure. Each group was then given 15 minutes per data structure to learn that topic with a given teaching aid. Each group got exposure to all three data structures using three different teaching tools, including traditional whiteboard, VisuAlgo [19], and our application. The schedule prevented groups from seeing any data structure or teaching method twice. This ensured that knowledge or experience from a previous session did not impact their opinion of any particular teaching aid.

After the students finished learning each data structure, they took a survey about the teaching methods. The survey's primary goal is to gauge which teaching method best engaged the students with the material. This survey was not a test of primarily understanding. Instead, our study focuses on how well each learning method engaged the students with the topic. This might be a stretch, but it would seem reasonable to assume that students who are better engaged with the topic are likely to learn it better, which is the goal of this paper [49].

B. Survey Questions

The survey includes eight questions, and five of them are related to tools and their engagements. The five questions are as follows:

- Q1: Which teaching method did you like the best?
- Q2: Which teaching method did you find the most interesting?
- Q3: Which tool do you feel best helps you understand how each data structure works?
- Q4: Would you like to see more AR/VR incorporated into the classroom?
- Q5: On a scale of 1-10 (10 being very likely), how likely would you be to use an AR application as a learning tool?

C. Survey Results

In this section, we analyze the students' answers to the five questions mentioned above with their significance.

Fig. 7 shows the results of students' answers for Q1. 54% of students selected our developed AR-based teaching application as the best out of the three teaching strategies tested. Interestingly, only 15% of students selected VisuAlgo as the best, which is lower than the percentage of students who selected the traditional whiteboard technique as the best.

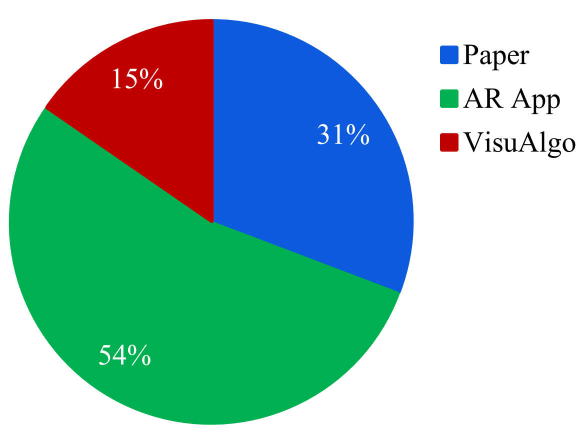

Fig. 7: The survey results for question "Which teaching method did you like the best?".

Fig. 8 shows the results of students' answers for Q2. 84% of students selected our developed AR-based teaching application as the most interesting technique out of the three teaching strategies tested. This shows that the AR application gets the attention of most of the students. Both VisuAlgo and traditional whiteboard techniques found to be the most interesting by a total of 16% of students (8% for each).

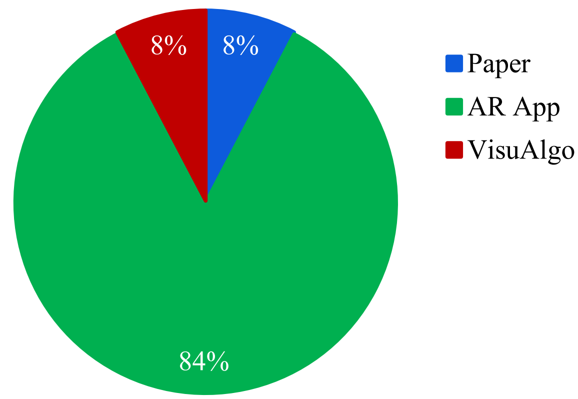

Fig. 8: The survey results for question "Which teaching method did you find the most interesting?".

Fig. 9 shows the results of students' answers for Q3. 38% of students considered that our developed AR-based teaching application helps them to understand the concepts better. However, we observed that the rest of the students (31% for each) considered that the other strategies helped them learn the explained data structures better. We believe that there is an effect of students' learning styles on the results of this question.

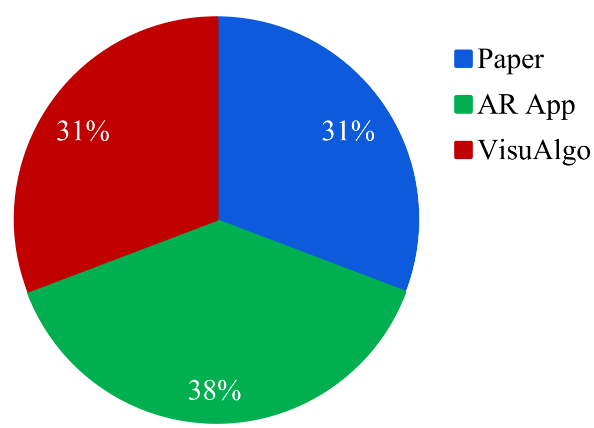

Fig. 9: The survey results for question "Which tool do you feel best helps you understand how each data structure works?".

Fig. 10 shows the results of students' answers for Q4. 85% of students wanted to see more AR-based content in the classroom. That is a significantly high percentage that exceeds our expectations. However, the result of this question shows that students are interested in advanced technology AR in the classroom while explaining the content.

Fig. 11 shows the results of students' answers for Q5. It is interesting to see that students would like to use AR-based applications as a learning tool. Such information must be considered by lecturers to be able to engage students.

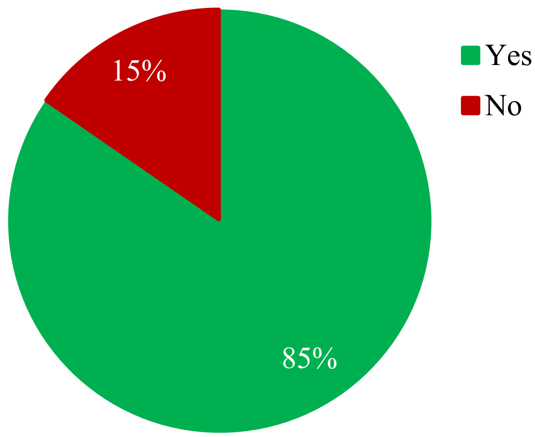

Fig. 10: The survey results for question "Would you like to see more AR/VR incorporated into the classroom?".

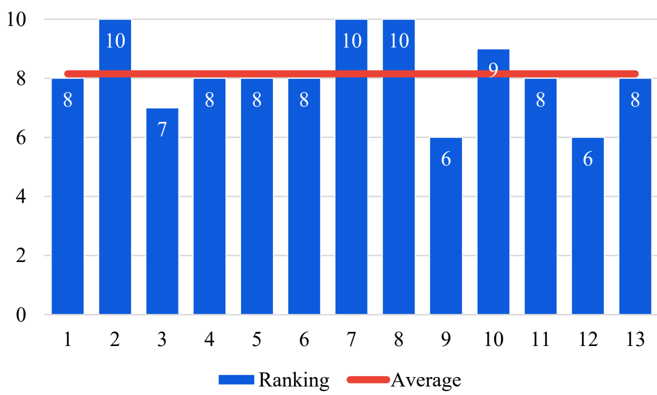

Fig. 11: The survey results for question "On a scale of 1-10 (10 being very likely), how likely would you be to use an AR application as a learning tool?".

IV. Conclusion

In this paper, we first develop Augmented Reality (AR) based visualization contents for data structures and analyze the effects of AR-based visualization and animation techniques for teaching data structures course. Many valuable insights can be taken from the results of this paper. A summary of the results can be listed as: (i) The survey results show that an augmented reality application is more interesting to students than other methods. Such apps can better engage students in the classroom. (ii) Augmented reality can be used effectively as an interactive aid to explain complex, abstract subjects in the classroom. (iii) Out of the three teaching methods in our study, students enjoyed using augmented reality more than other methods. (iv) Many students want to see more augmented/virtual reality in the classroom. As a teaching tool, augmented reality gets students excited about learning.

These observations show that our developed AR-based application could be valuable in teaching data structures. The results also indicate that such applications would better engage students with difficult topics and get them excited about learning.

V. Future Works

There is much improvement that can be made in the future in terms of development and experiments. This section will detail some of the ideas that have been considered for features that future groups can develop.

A. Expanded Object Addition

The ability to choose from a wider array of objects to add to a data structure would improve the user experience. For example, being able to add objects like bread, tomato, lettuce, cheese, etc. to a data structure can be more attractive because students can "make a sandwich" type things with their list. This would work exceptionally well with the Stack data structure. This would be a simple way to improve user engagement with the application. Adding basic shapes to a data structure is interesting, but adding some more realistic objects to the program would be an exciting feature. This would require some work with the user interface as well as in the Unity Asset Store to find these objects.

B. Expanded Data Structure Selection

The current application supports Linked List, Array List, and Stack. There are many other types of data structures that students need to learn. Adding some other data structures like AVL Tree, Heap, Queue, and HashMap would benefit students. Currently, we are working on adding Binary Search Tree and Queue.

C. Improved AR Stability and Recognition

As it stands, the application requires the use of a "target image" to function. We chose the Marshall University Logo as our target image. This is partly a limitation of the Vuforia framework. Along with the elimination of the target image requirement, improved stability for the objects in the AR environment will increase the usability of the application.

D. Testing

Due to the limited number of students, we experimented with a low number of students. We aim to increase the number of students in our future works to observe the effects on more students with a variety of data structures.

References

[1] S. Fayer, A. Lacey, and A. Watson, "Stem occupations: Past, present, and future," 2017. [Online]. Available: https://www.bls.gov/spotlight/2017/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future/pdf/science-technology-engineering-and-mathematics-stem-occupations-past-present-and-future.pdf

[2] X. Chen and M. Soldner, "Stem attrition: College students' paths into and out of stem fields," 2013.

[3] D. B. Silva, R. d. L. Aguiar, D. S. Dvconlo, and C. N. Silla, "Recent studies about teaching algorithms (cs1) and data structures (cs2) for computer science students," in *IEEE Frontiers in Education Conference (FIE)*, 2019, pp. 1–8.

[4] R. Li and X. Lin, "The application of computer aided instruction in course of elements of information theory," in *7th International Conference on Information Technology in Medicine and Education (ITME)*, 2015, pp. 631–633.

[5] F. Yusupov, I. Shamuratova, D. Yusupov, and T. Khudayberganov, "Improving the effectiveness of teaching the course "data structure and algorithms" based on structuration and integration of the discipline," in *International Conference on Information Science and Communications Technologies (ICISCT)*, 2019, pp. 1–4.

[6] J. Wenli, L. Yongjun, and W. Songtao, "Design and realization of networked cai teaching system for data structure in digitized learning environment," in *International Conference on Electrical and Control Engineering*, 2011, pp. 6508–6511.

[7] J. Cross, D. Hendrix, L. Barowski, and D. Umphress, "Dynamic program visualizations: An experience report," in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, 2014, p. 609–614.

[8] T. Färnqvist, F. Heintz, P. Lambrix, L. Mannila, and C. Wang, "Supporting active learning by introducing an interactive teaching tool in a data structures and algorithms course," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. Association for Computing Machinery, 2016, p. 663–668.

[9] D. Burlinson, M. Mehedint, C. Grafer, K. Subramanian, J. Payton, P. Goolkasian, M. Youngblood, and R. Kosara, "Bridges: A system to enable creation of engaging data structures assignments with real-world data and visualizations," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. Association for Computing Machinery, 2016, p. 18–23.

[10] B. J. Schreiber and J. P. Dougherty, "Assessment of introducing algorithms with video lectures and pseudocode rhymed to a melody," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. Association for Computing Machinery, 2017, p. 519–524.

[11] M. F. Farghally, K. H. Koh, H. Shahin, and C. A. Shaffer, "Evaluating the effectiveness of algorithm analysis visualizations," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. Association for Computing Machinery, 2017, p. 201–206.

[12] B. Park and D. T. Ahmed, "Abstracting learning methods for stack and queue data structures in video games," in *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2017, pp. 1051–1054.

[13] M. Mcquaigue, D. Burlinson, K. Subramanian, E. Saule, and J. Payton, "Visualization, assessment and analytics in data structures learning modules," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, 2018, p. 864–869.

[14] J. Young and E. Walkingshaw, "A domain analysis of data structure and algorithm explanations in the wild," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, 2018, p. 870–875.

[15] P. Gonzalez, "Teach data structures by using logical-mathematical object oriented formalism and several programing languages," in *38th International Conference of the Chilean Computer Science Society (SCCC)*, 2019, pp. 1–6.

[16] V. T. Kannappan, O. N. N. Fernando, A. Chattopadhyay, X. Tan, J. Y. J. Hong, H. S. Seah, and H. E. Lye, "La petite fee cosmo: Learning data structures through game-based learning," in *International Conference on Cyberworlds (CW)*, 2019, pp. 207–210.

[17] JGRASP. [Online]. Available: https://www.jgrasp.org/

[18] OpenDSA. [Online]. Available: https://opendsa-server.cs.vt.edu/

[19] "Visualgo." [Online]. Available: https://visualgo.net/en

[20] "Data structure visulizations." [Online]. Available: https://www.cs.usfca.edu/%7Egalles/visualization/Algorithms.html

[21] "MIT App Inventor." [Online]. Available: http://appinventor.mit.edu

[22] "Sonic PI." [Online]. Available: https://sonic-pi.net/

[23] "Alice." [Online]. Available: https://www.alice.org/

[24] C. Kelleher, "Motivating programming: using storytelling to make computer programming attractive to middle school girls," November 2006.

[25] "Scratch." [Online]. Available: https://scratch.mit.edu/

[26] "Scratchjr." [Online]. Available: https://www.scratchjr.org/

[27] "Made with code." [Online]. Available: https://www.madewithcode.com/projects/

[28] "Tynker." [Online]. Available: https://www.tynker.com/

[29] "Lego." [Online]. Available: https://www.lego.com/

[30] "Vex." [Online]. Available: https://www.vexrobotics.com/

[31] "Arduino." [Online]. Available: https://www.arduino.cc

[32] "Hamming bird." [Online]. Available: http://www.hummingbirdkit.com/

[33] "Finch robot." [Online]. Available: https://www.finchrobot.com

[34] "CoderZ." [Online]. Available: https://gocoderz.com/

[35] "Blockly games." [Online]. Available: https://blockly-games.appspot.com/

[36] "Make wonder." [Online]. Available: https://www.makewonder.com/

[37] "Minecraft." [Online]. Available: https://education.minecraft.net/

[38] "Robot turtles." [Online]. Available: http://www.robotturtles.com/

[39] "Kodable." [Online]. Available: https://www.kodable.com/

[40] "CodeCombat." [Online]. Available: https://codecombat.com/

[41] "Kodu." [Online]. Available: https://www.kodugamelab.com/

[42] "Lightbot." [Online]. Available: http://lightbot.com/

[43] "Game star mechanic." [Online]. Available: http://gamestarmechanic.com/

[44] "Shhero." [Online]. Available: https://www.sphero.com

[45] "Capture the flag platform." [Online]. Available: https://github.com/facebook/fbctf

[46] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators & Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[47] "Vuforia." [Online]. Available: https://developer.vuforia.com

[48] Unity. [Online]. Available: https://unity.com/

[49] G. D. Kuh, "Assessing what really matters to student learning inside the national survey of student engagement," *Change: The Magazine of Higher Learning*, vol. 33, no. 3, pp. 10–17, 2001.