

Learn Programming In Virtual Reality? A Case Study of Computer Science Students

Benjamin Alexander
Computer Science and Engineering
San Bernardino, United States of America
benalexsc@gmail.com

Bilal Khan
Computer Science and Engineering
San Bernardino, United States of America
bilal.khan@csusb.edu

Yunfei Hou
Computer Science and Engineering
San Bernardino, United States of America
yunfei.hou@csusb.edu

Jennifer Jin
Computer Science and Engineering
San Bernardino, United States of America
jennifer.jin@csusb.edu

Abstract—This paper presents the development of a new learning platform in Virtual Reality to create a more immersive and intuitive learning experience for introduction of programming courses at an intermediate level. This platform is designed to create a central hub for interactive courseware and facilitate distance learning in our post COVID world. Utilizing Virtual Reality, the application teaches specific topics in Computer Science using scripted animations, tutorials, and interactive games. A pilot study was conducted to evaluate the user experience and learning outcomes. Participants of this study reported they were more engaged and motivated in learning programming concepts. We found the virtual learning modules helped to explain abstract concepts and provided better hands-on experiences.

Keywords—*Introduction to Programming, Virtual Reality, Gamification, Game Based Learning, Learning Experience*

I. INTRODUCTION

Virtual Reality (VR) has been a popular topic for study in regards to recent technology. Its proposed benefits have been researched extensively. Similarly, Gamification has also been quite influential. Taking a concept and making an interactive game has been quite appealing to the educational industry. A large number of previous studies and reviews have been conducted regarding both of these ideas, including prototypes, implementation, and proposed benefits. In this paper, our application, Computer Science Virtual Interactive Laboratory (CSVIL), attempts to make use of this past research in VR and Gamification to create a new learning experience.

By combining a VR environment with Gamification techniques, this application seeks to increase the effectiveness of learning different Computer Science topics. The educational industry has recently displayed an expanding interest in VR technology. Being able to create simulations of real-world concepts is becoming more appealing to both the public and private sectors. After reviewing literature and previous works, it seems that Computer Science does not have many options for utilizing VR courseware. We created this application to supplement this absence. Developing such an application also allows us to further study student interest in VR courseware and as such evaluate their response to the application's content. In addition, this application provides a better understanding of the design challenges and feasibility for developing VR courseware for different disciplines. Our hope is to create a VR courseware platform for the post

COVID community, where online and hybrid education will likely be a strategic priority.

The aim of this paper is to evaluate the effectiveness and usability of the CSVIL application. To supplement this, the paper will present existing research in both VR and Gamification techniques, the application's development process, and a student evaluation of the application. To study the student's perception of VR courseware, our evaluation focused on the user's experience and the proposed learning outcomes of the application.

II. LITERATURE REVIEW

We will briefly discuss the promising benefits of both Gamification and VR by summarizing the results from previous literature. The following review is the culmination of recent work regarding studies of similar topics.

Gamification, as Hernandez explains in his article, "is described as the process of game thinking and using its mechanisms to attract users to solve problems. This definition can be applied to any situation; however, in the educational field, Gamification refers to using elements of the game to engage students, motivate them to action, and promote learning and problem-solving. Gamification implies learning by doing and learning by interacting" [1]. Grimley confirms this in his investigation of whether learning through a game can improve student learning compared to a lecture approach. "Increased feelings of being active, involvement and a perception of challenge were reported when interacting with the game leading to higher engagement for students" [2]. In a similar study conducted by North and his co-authors, they report that creativity and creative thinking were also improved using a Gamification approach saying, "Disciplined imagination as being related to imaginative creations, risk-taking and participating in challenging tasks seemed to be amplified through game play. There seems to be consensus on how game based learning supports and potentially improves awareness and knowledge on certain subject areas contributing to learning effectiveness" [3]. Finally, in his study, Role of Gamification in Engineering Education, Anil states that, "electronic based Gamification tools stimulate interest, increase the participation among young generation engineers to the lectures, and enable professors to track the students' learning progress in every stage of curricula. Additionally, the findings highlight that Gamification increases students' critical thinking ability and efficiency in

problem solving since they retain information and apply the new learned concepts in an integrated learning approach” [4]. Gamification and game based learning is thus a tool developers can use to take different topics and create immersive interactive experiences to increase the learning outcome. Based on the numerous studies on the topic, it is evident that Gamification is quite an effective tool. Increasing participation and interest in topics that are generally tedious or tiresome will undoubtedly increase the learning outcome and experiences of the student participants. For this application Gamification is applied similarly to how the authors describe above. Taking different topics in Computer Science, we can create different interactive tutorials. Rather than simply listening to a verbal explanation of how a specific algorithm is performed, we can apply Gamification to create captivating and interactive animations to visualize and present flat two dimensional (2D) topics in a three dimensional (3D) space.

In contrast to Gamification, VR is a relatively new technology. This technology allows users to immerse themselves into applications and games creating an entirely new way to learn. Traditional game based learning is done in a 3D space using a mouse and keyboard for input. With VR we are placed in the world using headsets and sensors. Instead of just clicking on buttons, we can now move within the application to grab virtual objects and interact with them. VR offers a much higher level of immersion than was previously possible. Stender, Paehr and Jambor explain additional benefits in more detail. “VR applications generate a higher degree of immersion compared to other media thus potentially improving the learning experience. VR technologies make it possible to provide learners with computer-generated virtual, 3D content, in addition to reality to support learning processes. VR also allows learners to interact with these virtual elements. One of the well described didactic benefits of VR is that it enables learners to gain a better spatial ability of phenomena, machines or technical installations. In particular, learners with a lower spatial ability benefit from a reduction of mental load. Also, the applications allow for a more intuitive interaction than other learning media placing additional information and the learning subject in a direct spatial context so that fewer transfer steps have to be taken during learning. By visualizing additional virtual content and information in AR-applications, learners' attention can be focused on the current work step and the learners can be guided through the workflow. Work processes can be learned and understood by doing them directly. And if learning content is linked to movement, the retention rate for process learning improves.” [5]. From Stender’s analysis, it is evident that this increase in immersion and physical interaction helps students to truly digest and divulge themselves in the VR courseware. Removing outside sources of distraction and providing a much more immersive and interactive experience allows students to achieve a much greater grasp on the concepts in which they are presented. In a similar paper to our own, Hurtado and his co-authors created a VR simulation to create a Universal Robot 5 e-Series in virtual reality. Their application allowed them to learn and use the robot in VR without interacting with their real world counterparts [6]. Similarly, our application allows students to utilize and interact with technology without using compilers or developing programs. Instead users can interact with or view algorithms, and concepts, as they are executed. Our application allows users to learn and experience Computer Science topics without the need of a supporting lecture or

hands on coding. Students can learn and experience the topics in real time with the ability to watch data and code being manipulated as if they were viewing the internal system of the compiler. Our application and Hurtado’s are similar in the sense that we allow students to utilize and learn how to use a Computer Science based technology, except we provide additional tutorials and resources for better cementing and learning the concepts we provide.

Based on this review of literature, it is evident that the increase in interest and interaction through Gamification, in addition to the increase of immersion and access to a controllable virtual environment through VR, will not only promote but also increase retention, understanding, and motivation of each topic presented in the virtual environment. Our hope is to effectively utilize this past research to deploy a successful learning platform that will hopefully influence future learning applications.

III. APPLICATION

The CSVIL application was designed over the course of a year on the Unity game engine. Utilizing Steam VR, Unity’s primary scripting language C#, and the HTC Vive Virtual Reality Headset, we developed a suitable learning platform for college students to learn Computer Science topics. The CSVIL application was designed as a learning platform for Computer Science students to study and learn about topics utilizing Virtual Reality. During development we adopted a modified Agile software development model with sufficient time to define requirements, develop, and test, followed by scheduled meetings to evaluate progress.

The development of the application was challenging and took some time to solidify a design and style. Since this application is the first of its kind, it was very difficult to find similar works, ideas, or concepts to utilize. Researching topics was simple, but creating interactive tutorials and animations was challenging with the sparsity of similar VR courseware. Our User Interface (UI) design was also changed multiple times during development. Unfortunately, there were multiple instances where the entire layout of sections or topics had to be completely changed. With the multitude of different algorithms and concepts, it became a challenge when a fix or change was required. When such an instance occurred, then all similar tutorials for each topic had to be modified as well, which sometimes caused a further list of issues to reveal themselves. It became quite evident that topics and animations should have been developed in sequence, one after another, instead of parallel to each other. During our meetings we had more to show, but as a result some concepts may have taken longer as opposed to if we had approached development more sequentially. These challenges provided an amazing learning experience, and with the help of online resources, including Unity documentation, Steam VR documentation, Brackeys, and VR with Andrew, we were able to develop a suitable application.

As a result of our efforts, the project provides college level students with a multitude of tutorials, animations, and explanations for students to observe and study. Each module provides these students with access to different functions or topics for them to utilize.

A. Main Hub

Upon starting the application, students will find themselves in the Main Hub. This Main Hub acts as the central

passageway to the three learning modules that the application provides, i.e., Sorting Algorithms, Programming Paradigms and Lecture Hall. This hub includes tutorials that give controller layout information and provide an introduction for each module.

B. Module 1: Sorting Algorithms

The first of these modules is a virtual room for learning sorting algorithms. To supplement this section, we created multiple tutorials and animations for five different sorting algorithms. These algorithms include Merge Sort, Quicksort, Selection Sort, Insertion Sort and Bubble sort. Each sorting algorithm provides students with a multitude of panels for key concepts of each algorithm. These panels provide scripted tutorials, text explanations, and interactive games for students to utilize. See the below figure 1 for a picture of the sorting room.

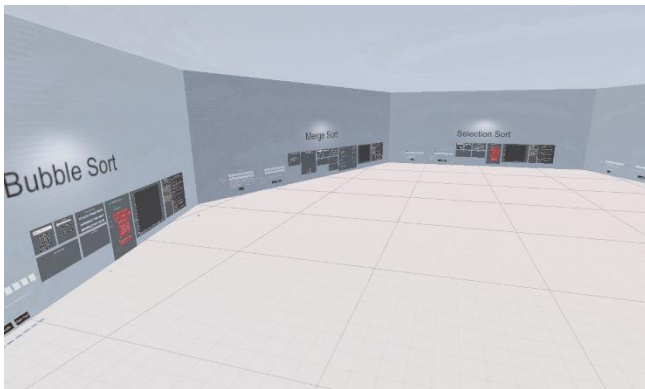


Fig. 1 Image of the Sorting Algorithm Room.

Each sorting algorithm consists of 5 different tutorials. See figure 1 for an image of the Sorting Algorithm room. The first panel is a text tutorial that verbally describes how the algorithm is performed. Students may also switch pages to view the explanation of the time complexity for the algorithm. This text tutorial was designed for students who prefer a verbal or text-based explanation of how the algorithms work.

The next panel provides an animation that allows students to run an array of their choosing through the sorting algorithm. They can use a number pad to push and pop numbers on and off the array, or they can push 5 random numbers, up to a maximum of thirty array elements. Next, they can then run the sorting algorithm over the array they created. At any time, students can pause the algorithm, step back to the previous step, or step forward to the next step of the algorithm's execution. The animation also provides students with a visual representation of the array as it is being sorted, a view of the code, including the current line of code being run, and a 3D graph that displays the array and how it changes over time. This graph is meant to give students a visualization of how the array looks when it changes. Providing a graphical interface is a unique way for students to examine the array over time. See figure 2 for an example of the first Sorting tutorial panels.

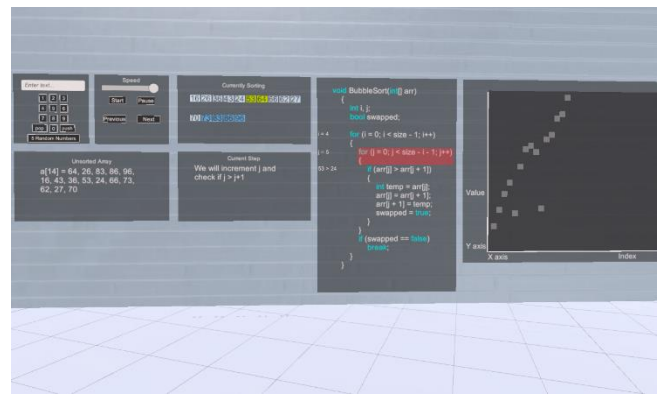


Fig. 2 Image of the first Sorting Tutorial Panels.

Our next panel provides a similar animation. This panel runs the algorithm over a random array. It will perform animated swaps, and it will change colors of each element as it is being iterated over by the algorithm. This panel is designed to allow students to visualize the sorting pattern that each algorithm has. All five algorithms have a specific pattern or rhythm that they are designed to follow. This tutorial puts emphasis on this pattern and attempts to expose this to the user without using any code.

The following panel allows students to perform the algorithm on their own. Students must perform each step of the algorithm in order to complete the tutorial. Students can grab different array elements and perform the algorithm themselves without a compiler. Students have access to a guidance mode and a test mode. Guidance mode will highlight the index that students must interact with or swap. Test mode requires students to interact and perform swaps without assistance of any highlighting. This mode will also record correct answers, incorrect answers, and number of retries. See figure 3 for the interactive guidance mode.

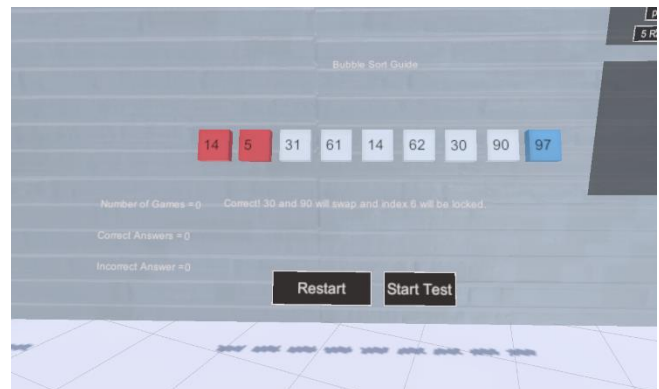


Fig. 3 Image of the Guidance Mode.

The final panel for this topic is only present for sorting algorithms that utilize recursion. Specifically, Merge sort and Quicksort. This panel provides students with a look at which recursive method is being called at any given time, in addition to a visual representation of how recursion affects the array during the algorithm's execution. The array in this panel is designed to resemble a descending tree and gives students a new representation and visualization of how recursion is performed for these algorithms. See figure 4 for the Recursion tutorial panel.

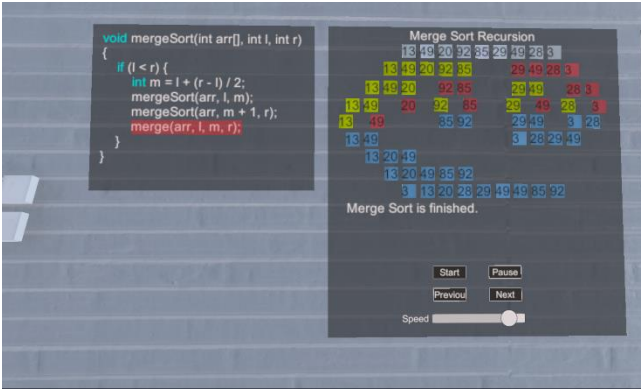


Fig. 4 Image of the Recursion Tutorial.

C. Module 2: Programming Paradigms

The second module is a virtual room for learning programming paradigms. This room is separated into different sections depending on the paradigm. This module offers information on Object Oriented paradigms, Functional paradigms, Logical paradigms, and Procedural paradigms. See Figure 5 for an image of the Programming Paradigms room.

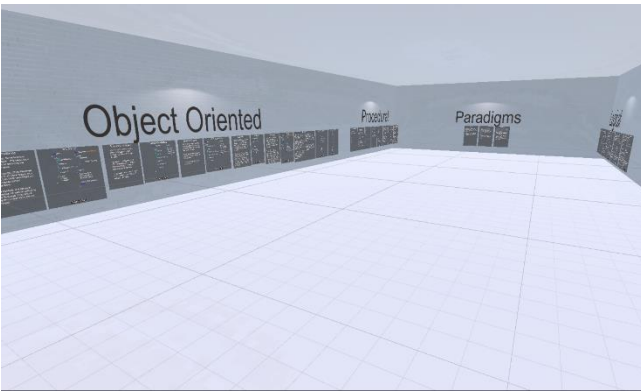


Fig. 5 Image of the Programming Paradigms Room.

For each different paradigm we include multiple different panels covering the differences for each paradigm. The first of these panels covers what languages are included or used for each paradigm. Additional panels include different features and characteristics that the programming paradigm offers. Each of these panels includes a text description of the feature and a small example program written in a language for that paradigm. These example programs can be paused and resumed at any time during their execution. They will attempt to show how the specific feature for that paradigm can be utilized in an actual real-world problem. See figure 6 for an image of an example program.

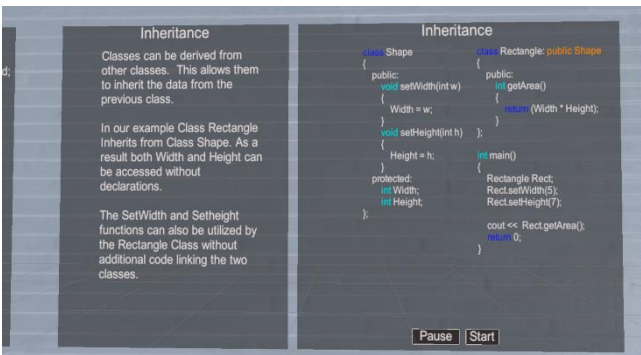


Fig. 6 Image of an Object Oriented sample program.

D. Module 3: Lecture Hall

The third and final module is a virtual Lecture Hall where students or professors will be able to hold interactive lectures or presentations. The host of the room will be able to upload presentations or videos and display them to an audience of their choosing. Students will have access to a Lecture Hall to supplement this module. See figure 7 for an image of the Lecture Hall room.



Fig. 7 Image of the Lecture Hall.

IV. EVALUATION

In order to evaluate the effectiveness and useability of the application, we conducted a testing and evaluation session to gather and record the student experiences and perceptions. Students were brought into a lab and asked to test the application. Students were given fifteen minutes to explore the game and try all tutorials that interested them. After this testing period, participants were given a short survey to record their experiences. The research questions of this evaluation include:

- To what extent are users comfortable with using VR equipment to interact with the application?
- How long does it take for users to begin feeling fatigued or distracted?
- What are the users' overall experiences in learning computer programming in VR?
- To what extent does VR affect or improve the users' learning motivations and outcomes?

For this evaluation we recruited volunteers from different Computer Science classes. We managed to recruit fourteen participants to assist with evaluation and testing of the application. This number suffered because of how difficult it was to recruit participants due to COVID-19 restrictions on lab access. The majors by participants include Computer Science (71.4%), Computer Engineering (7.1%) and Computer Systems (21.4%). The survey students were administered included a fourteen question google form with multiple choice and short answer questions. This test came in two major sections entitled User Experience and Learning outcomes.

A. User Experience

The User experience section was used to record the user's opinion of interactive systems during the use of the application. User experience plays particular attention to the user's perception of controls, UI, and environment. These are questions posed to evaluate the application's design rather than its effectiveness. The first question we asked participants was

to rate their familiarity with VR simulations. Most students reported some familiarity with VR. See figure 8 for the results of our first question.

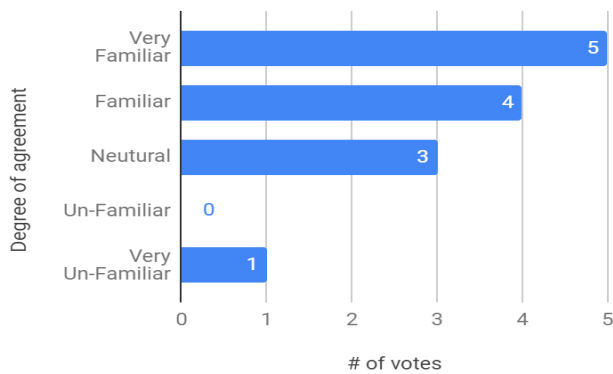


Fig. 8 Question: Rate How familiar you are with Virtual Reality Simulation.

In addition to their familiarity with VR, we also asked participants how well they perceived the control layout for the application (refer to figure. 9). Controls for VR applications can sometimes be obscure and confusing at first. The VR technology is not as common as a mouse and keyboard, so it is possible that some may find the new control scheme difficult. The evaluation yielded results that seem to correlate with our first question's responses. Given the small number of participants in the pilot study, we did not conduct correlation analysis between users' prior experience on VR and their perception on learning.

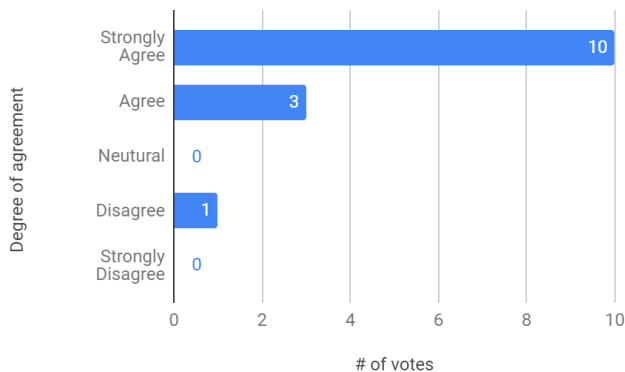


Fig. 9 Question: Were VR controls simple to use and easy to understand?

Following these questions, we asked students to rate their opinion of the environment and decor in simulations focused on learning. Students reported that they believe environment and decor to be a necessity in such platforms. This suggests that the final version of the application may need more interior design or environmental clutter to make students more satisfied or comfortable with the simulation. Possibly an empty simulation room is perceived as unnerving and dull, which could affect user experience.

To finish off the section, we asked students to express any critiques with interface and panel design, in addition to if they felt fatigued or dizzy at all during the simulation. Most students did not feel fatigued but six did report slight dizziness after 15 minutes of testing. Most students' suggestions and critiques involved changing font size of different tutorials. VR has a much smaller resolution depending on the hardware utilized, so it is possible the headset affected how clear the font appeared for participants during testing.

B. Learning Outcome

The learning outcome section of the survey focused on the users' perception of the modules. This section was designed to evaluate whether students thought the tutorials were useful as educational learning materials. The first question of this section asked students to rate how well the application taught sorting algorithms. Since the Sorting Room was a major section of the project, with most of our interactive modules and animations, this question would heavily determine if further development of such tutorials would be worthwhile. During testing many of the participants expressed their intrigue at watching a 2D object be manipulated in a 3D space. Most participants were very interested in the many ways that our learning modules represented sorting algorithms. Where before students could only view a sorting algorithm through code, our tutorials attempted to provide a new way to look at the algorithms during their execution. Figure 10 shows student perception on tutorial effectiveness.

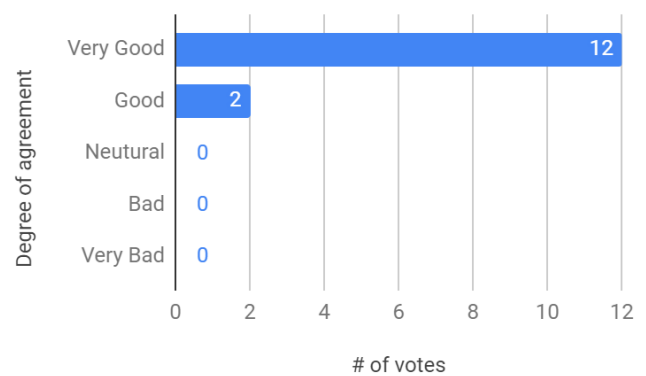


Fig. 10 Question: how well did the simulation teach Sorting Algorithm concepts?

Now that students had reported their evaluation of our tutorials, we wanted to compare these interactive tutorials with others that could be accessed online. To supplement this question, we asked students to express whether they would prefer an animated tutorial in VR or a traditional tutorial much like those found on coding websites. Three students reported that they still prefer traditional tutorials, but the remaining students reported that they preferred the VR method instead. See figure 11 for student opinion on using traditional or VR tutorials.

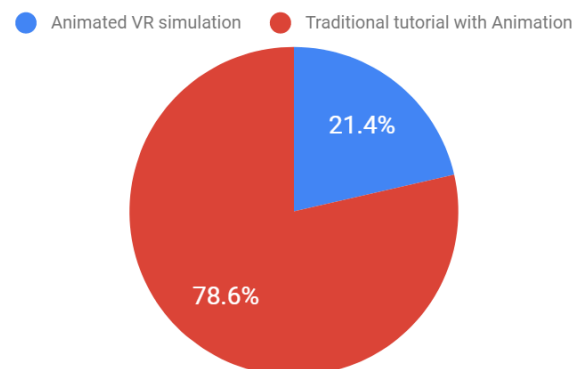


Fig. 11 Question: Which type of learning tool would you prefer?

We also asked whether VR affected the student experience with learning Computer Science. We received very similar results from our first question of this section with most students agreeing that VR increases interest in Computer

Science topics. Participants found the ability to view data in a virtual space incredibly interesting and expressed this verbally during their evaluation. Many students enjoyed watching the arrays being manipulated, as this allowed them to watch the algorithms in action. Providing the visual aids and representations seemed to captivate and motivate students to pay more attention to the animations and tutorials.

The next question we posed to the participants allowed them to report their overall experience with the simulation (refer to figure 12). Most students reported a satisfactory overall experience. However, this time we did have one student that was not particularly satisfied with the application. It is possible the student's overall experience was affected due to the lack of environment detail and text polish. Unfortunately, we were not able to inquire more due to the anonymity of the survey results. This application is still considered to be in an alpha state. As a result, many changes will be made before an official release, and we will continuously work on improving user experience.

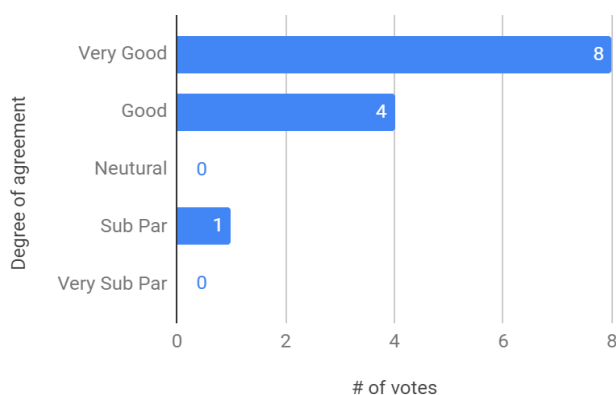


Fig. 12 Question: How would you rate your overall experience with the simulation?

The final question in this section asked participants to report how likely they would be to recommend the application to other students. Most students reported that they would recommend the application. With the new way to observe and learn different topics in computer science, students seemed hopeful for the future of the application. Their suggestions mentioned adding a guide that helped touring the different sections of the applications, much like a butler that would escort participants and show them how to use algorithms. Their enthusiasm provided a confident and optimistic perception for the future of the platform.

With this survey, we can see many different areas in which we can improve the application. A final question was posed to allow students to give suggestions for the application. Many gave good advice that will be very beneficial for further development. Student participation was incredibly helpful, and all participants provided great feedback for us to improve the platform.

V. CONCLUSION

This project presents one of the first platforms for learning college level Computer Science topics in VR. Through the interactive tutorials and game style quizzes, students were given the opportunity to learn about five different sorting algorithms and four different programming paradigms. Each topic provides multiple tutorials and panels of information, in addition to multiple interactive modules for students to utilize. Through our evaluation, it can be concluded that the

application was successful in both improving the learning outcome and providing a new interesting and immersive experience. Most participants expressed an optimistic opinion for the application, and by the evaluation it was evident that the program was both usable and increased the learning experience for each topic. By providing captivating animations and interactive tutorials, we are able to immerse students in a virtual learning environment that was not previously possible. The participants that had the opportunity to test the application expressed interest in future developments, and most were very satisfied with the new style of learning. Traditional lectures simply cannot provide hands-on interactive experience to such a degree. With the ability to present flat 2D topics in a 3D space, this platform gives students a new and unique way to view Computer Science topics. Seeing these topics for the first time had participants interested and focused. Observing a sorting algorithm on paper is flat and uninteresting, but watching the array transform over time provides a fascinating alternative that has not been utilized previously.

This VR platform does have its limitations however, and the most obvious is the fact that VR equipment is cumbersome to use. VR headsets can be quite expensive which makes it difficult for institutions to provide a multitude for their students. A non-VR version was made to supplement this fact, but it is evident that the level of immersion will be nowhere close to the original style we had intended. For future studies, we would also prefer a much larger sample size for evaluation.

REFERENCES

- [1] Hernandez, M. I., Lezama, R. M., & Gomez, S. M. (2021). Work-in-progress: The road to learning, using gamification. 2021 IEEE Global Engineering Education Conference (EDUCON).
- [2] Grimley, M., Green, R., Nilsen, T., Thompson, D., & Tomes, R. (2011). Using computer games for instruction: The student experience. *Active Learning in Higher Education*, 12(1), 45–56.
- [3] North, B., Diab, M., Lameraz, P., Zarak, J., Philippe, S., Muller, J., & Fischer, H. (2021). Developing a platform for using game-based learning in vocational education and training. 2021 IEEE Global Engineering Education Conference (EDUCON).
- [4] Anil Yasin, A., & Abbas, A. (2021). Role of gamification in engineering education: A systematic literature review. 2021 IEEE Global Engineering Education Conference (EDUCON).
- [5] Stender, B., Paehr, J., & Jambor, T. N. (2021). Using AR/VR for technical subjects in vocational training – of substantial benefit or just another technical gimmick? 2021 IEEE Global Engineering Education Conference (EDUCON).
- [6] Hurtado, C. V., Flores, A. R., Elizondo, V., Palacios, P., & Zamora, G. (2021). Work-in-progress: Virtual reality system for training on the operation and programming of a collaborative robot. 2021 IEEE Global Engineering Education Conference (EDUCON).
- [7] Evangelou, S. M., Stamoulakatou, G., & Xenos, M. (2021). A serious game for mobile phones used in a software engineering course: Usability Evaluation and educational effectiveness. 2021 IEEE Global Engineering Education Conference (EDUCON).
- [8] Moser, G., Vallon, R., Bernhart, M., & Grechenig, T. (2021). Teaching software quality assurance with gamification and continuous feedback techniques. 2021 IEEE Global Engineering Education Conference (EDUCON).