TDVR: Tower Defense in Virtual Reality: A Multiplayer Strategy Simulation

Andrew E. Munoz, Zach Young, Sergiu Dascalu, Fredrick C. Harris, Jr.
Department of Computer Science and Engineering
University of Nevada, Reno
Reno, NV United States
amunoz24@nevada.unr.edu, zachidk@gmail.com, dascalus@cse.unr.edu, fred.harris@cse.unr.edu

***Abstract*— Virtual Reality is a relatively new technology that has become a popular medium for games. This project is a virtual reality tower defense game that will allow for players to engage and play in a multiplayer environment. The users are given the option to either host a match or connect to a host player as a client. Once in the simulation, the users then compete against each other using various defense strategies. The game itself blends first person combat with top down view strategy mechanics. This paper details the implementation and design of the project as well as a detailed description and explanation of the current status of the project.**

***Index Terms*—Virtual Reality, Head Mounted Display (HMD), Multiplayer, Projectile, Network, Unity, HTC Vive**

I. Introduction

Virtual Reality (VR) technology provides immersive experiences that are currently unmatched and continue to improve as the technology develops. This is due primarily to the feeling of presence that Virtual Reality can offer to users that are simply not possible with other entertainment mediums. Through the use of a Head Mounted Display (HMD), VR technology places users into a virtual environment where the users can then have a multi-sensory experience that continues to become more immersive as the technology improves [1].

This project focuses on the development of a highly interactive and immersive simulation entitled Tower Defense in Virtual Reality. This simulation is that of a tower defense game that has a strict focus on multiplayer functionality. The main purpose of the game will be to allow players the opportunity to pick from different combat units, each equipped with different strengths and weaknesses, who attack the nearest enemy target to them. In a multiplayer environment, the users can compete against each other in an attempt to try and take down the opponents tower. Players can display their skill in both strategy of spawning combat units and dexterity by means of attacking enemy units with their own weapons. The simulation is also equipped with combat animations and health bars for both the towers and combat units that the user spawns. This project was developed and built using the Unity game engine. A more detailed breakdown of the design and implementation will be discussed further in sections III and IV.

The organization of the paper is as follows. In Section II similar work is mentioned and compared to our work and various studies are discussed that are relevant to multiplayer and social interaction in virtual environments. Section III gives a brief overview of the project as well as details the software design. Section IV will go into detail about the simulation and game play mechanics. This is followed by Section V which goes over discussion and overall analysis of the project development. Section VI then details the current status of the project. Lastly, Section VII covers the conclusion of the analysis and potential future development.

II. Background Review

This section provides an overview for background review and various related works. It is divided into multiple subsections with the first focusing on similar products that already exist and the second focusing on studies that relate to multiplayer experiences in virtual reality.

A. Related Applications

There have been a few similar applications developed within recent years that have each taken different approaches to tower defense game play in a VR environment. One such application is a game entitled Castle Must Be Mine [2] which gives the user a top down view of a pathway to a castle where users can setup defenses to ensure that enemies do not make it through to the castle. The game also includes an upgrade system where users can upgrade both their towers and their playable hero character. Another similar application is a game called Battle of Kings VR [3] which is a tower defense strategy game set in VR where the user can defend their kingdom as well as develop economics to fortify their defenses and upgrade their armies. Battle of Kings VR also affords the user with the ability to play in a campaign mode that includes different maps and themes or an online multiplayer mode where users can play against friends or other users.

There have also been a few games developed that give the user a first person view in the virtual environment. An application that utilizes the first person perspective is the game In Your Face TD [4] which allows users to battle enemies face to face with a variety of weapons including a shotgun, laser gun, as well as bombs. In Your Face TD offers a different experience from the other games mentioned due its futuristic setting and level design. Another first person based tower defense game is Alchemist Defender VR [5] which combines top down view defense mechanics with first person combat. This combination makes this game unique to some of the other

games mentioned as it blends different components of tower defense game play together. Our game will utilize some of these mechanics mentioned from other games; however, we will also add the options to select from different factions to build armies as well as being able to fight in an alliance with friends through multiplayer functionality.

B. Literature Review

With virtual reality continuously growing, it has been utilized in a variety of studies to test and research areas pertaining to learning, reaction time, and movement through simulation. For example, a study conducted in [6] presented a mobile framework that was used to capture user movements and translate them into a virtual environment. The authors employed the use of smartphone cameras to track the user and their movements and developed a multiplayer first person game for their virtual environment as a proof of concept.

Another area of research in virtual reality are different means of control that can help increase a users sense of presence within the virtual space. One such study conducted by Sra, Xu, and Maes [7] presents breathing as a possible means of controlled input for VR games. The experiment was designed in correspondence with two game modes: a First Person Shooter and a Ball Game due to them both being easy and could be quickly learned. The authors created four breathing mechanisms that the users could use as controlled inputs and therefore increase their sense of immersion. Based on the results, the experiment is shown to be successful as users reported to have felt an increased sense of presence when using the breath control input as opposed to non-breathing control input.

One of the biggest issues in the development of VR applications is motion sickness. In reference to VR, motion sickness is often referred to as cybersickness and results in symptoms such as nausea, headaches, and disorientation among others. It has been studied that cybersickness may be in part to the design of an application and therefore needs to be tested for usability to ensure that cybersickness can be minimized [8]. Another study was conducted that looks into three different methods of locomotion mechanics which include node-based, continuous, and arc-based teleport locomotion. The results from the study produced evidence that indicated continuous movement between nodes resulted in less instances of cybersickness [9]. Since the virtual world in our game will likely be larger than the play area for most users, a form of locomotion to travel is needed for the user. The locomotion method of using nodes to travel to specific important locations will be used based on [9]. This method is effective for our purposes because the user will only need to travel to specific predetermined locations such as a weapon.

While VR has some drawbacks such as cybersickness, there are many areas where VR technology can be used to help users through training simulations and learning. For instance, research conducted in [10] shows users gaining experience in a virtual application and how the experience can help gauge their ability to learn. The study itself combined VR, kinematic tracking, and Electroencephalography (EEG) to investigate the cognitive and neural mechanisms of motor skill for the user. The main task was a marksmanship task that took place over the span of a few days to test users reaction time and adaptability as well as provide a platform to develop mobile brain-body imaging. This is similar to our game in that aiming is required so there will be a learning curve to the game. This allows users to have varying levels of skill that can be improved upon over time in a competitive environment.

III. Project Overview and Design

In this section, we will discuss how Tower Defense in Virtual Reality was built such as the main hardware components used as well as the software tools used. During the early stages of project development, our team broke down the design elements to create a development plan that established a structure to meet the project goals. Our team accomplished this by breaking up certain elements to each work on separately. For the design elements that required more time, our team employed peer programming to have more effective development. In order to discuss the project development thoroughly, this section is separated into three subsections: Hardware, Software, and High Level Design.

A. Hardware

The main hardware component used to build and test the simulation was the HTC Vive VR base system. The Vive Virtual Reality System includes a headset, two wireless controllers, and lighthouses or sensors that track the user within the specified play area. The Vive headset is an HMD that has a front facing camera to fully immerse the user into a virtual environment and also contains an adjustable strap to ensure that the user has a certain level of comfort while in the virtual space. The Vive controllers have a unique design that is specific to VR with easy to understand controls and haptic feedback to add to the users sense of presence. Fig. 1 shows images of the full HTC Vive VR system.

Fig. 1. (Left) HTC Vive Headset. (Right) HTC Vive Controllers with adjustable straps.

The Vive headset has a Dual AMOLED 3.6" diagonal screen with a resolution of 1080 x 1200 pixels per eye. It also includes an integrated microphone that can be used for players to communicate with each other in multiplayer VR experiences. The headset utilizes a lens distance adjustment for eye relief to provide users with a sense of comfort and maintain a strong sense of immersion while in a virtual environment. The controllers allow for a variety of input sources such as track pad, grip buttons, trigger, system button, and a menu button. As for tracking, the lighthouses can have a maximum tracked play area space of 11'5" x 11'5" [11].

B. Software

Tower Defense in Virtual Reality was created using the Unity game engine [12]. During development of the project, our team used Unity version 2018.3.4 to create and build the simulation. We ran Unity on Windows 10 and wrote C# scripts for the simulation using Microsoft's Visual Studio. Our team utilized Unity Collaboration to work in tandem on different aspects of the project development and to provide an easy means of version control.

SteamVR [13] was the main software tool used to connect with the hardware and track the users movement as it keeps track of the play space as well as tracks the various components of the Vive system. SteamVR is able to effectively interface with the HTC Vive VR hardware system and connect to the main computer system running the simulation.

One of the biggest components of the project was being able to develop multiplayer functionality. In order to add multiplayer capabilities, we utilized the high level scripting API (HLAPI) called Mirror [14]. The Mirror networking HLAPI allowed for the use of a Network Manager and the operation of so-called "client-hosted" games where a player acts as a host for a multiplayer game. It also allows for the spawning of non-player objects on the server which both host and client can see within the virtual space, i.e. defense units sent to attack the opposing player.

C. High Level Design

The design diagram shown in Fig. 2 details the overall design flow of a user setting up and running the Tower Defense simulation. Upon starting the application, the user will be first shown a menu screen where they have the option to run or quit the simulation. In order to run the simulation, the user can choose to either be the host of the multiplayer game or connect to another player host as a client. If the user decides to start the simulation as the client they need to first ensure that the host player is online, otherwise the user will not have a server to connect to. Once the simulation starts the user is then placed on top of a tower and must prepare their defense strategy. Game play details and a description of the simulation itself will be discussed further in Section IV.

IV. Simulation

At the core the simulation is simply a tower defense game where one player fights another using various resources and combat units. As stated in the previous section, Tower Defense in Virtual Reality was developed using the Unity 3D game engine and tested using the Steam VR software to connect with the HTC Vive hardware. This section presents the game play mechanics, the defense units, the game world, and the weapons a user has at their disposal.

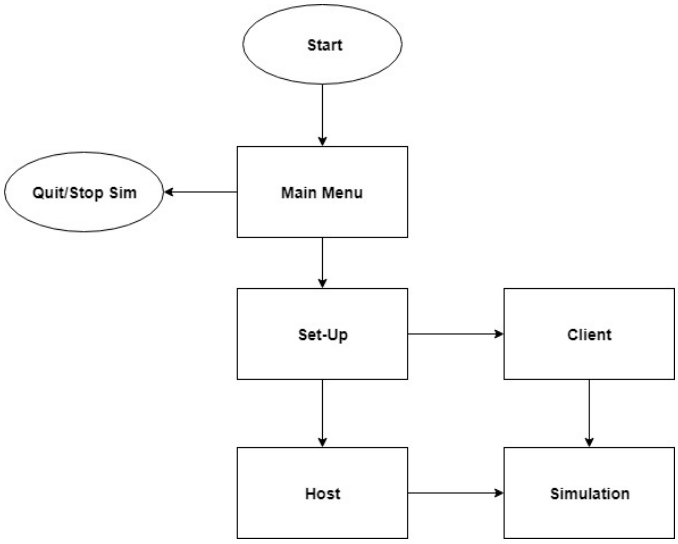

Fig. 2. High Level System Design Diagram that shows the design flow of the Tower Defense simulation.

The simulation itself was designed to be a competitive multiplayer experience with two players that allows for gameplay that challenges an individual strategically and physically. One user would act as the host server while the other user would connect to the server as a client. Once in the simulation, the users are then placed on opposing towers that are facing each other and they are armed with a slingshot and have access to a top down board view of the map. The board view contains various types of units that user can then deploy to attack the opposing player. The board with units is designed to be the strategic aspect of the simulation while the slingshot and future iterations of weapons are designed to be the dexterity aspect. It is then the goal of each user to destroy the opposing users defenses and main tower. The various types of units, weapons, and rules will be described in further detail later in the section.

To begin the simulation, the user will be placed in a Menu environment. In Fig. 3, a screenshot from the menu area can be seen. The environment itself is simply a white plain because the user has to connect through a computer station. The user can enter the simulation from this screen by clicking on either the LAN Host or LAN Client GUI buttons shown in the left hand corner of Fig. 3. This setup of the simulation is accomplished from a computer station prior to entering the virtual environment and in order for the user to connect via client that must first ensure that the host player is online prior attempting a connection.

Once the users enter the simulation, they are placed on opposing watch towers and then immediately must prepare their defense strategy. The users have access to a Mini Board

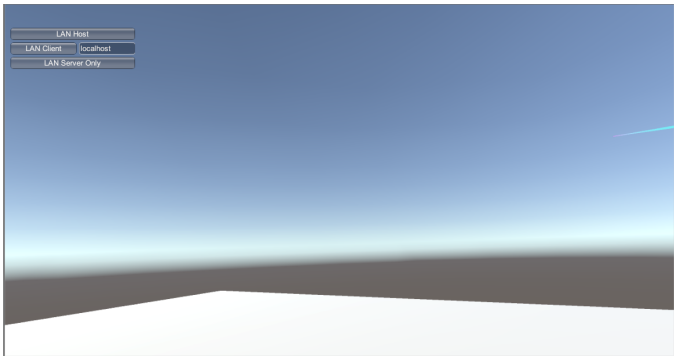

Fig. 3. Screenshot of Menu screen when the game is first initialized. In top left corner, provides user with the option to be the LAN Host or LAN Client on a particular game.

which has drag and drop spawn units that the user can place onto the battlefield. Using the Mini Board, the user can place the units at three different spawn points in front of their watch tower. These spawn points include the left, middle, and right of the tower facing toward the enemy player. The units, shown in Fig. 4, that the user can spawn include a goblin class and a wizard class. The goblin class acts a common combat type that the user can place as their basic unit. The wizard class, on the other hand, is a stronger mystic class that can deal high amounts of damage with increased range.

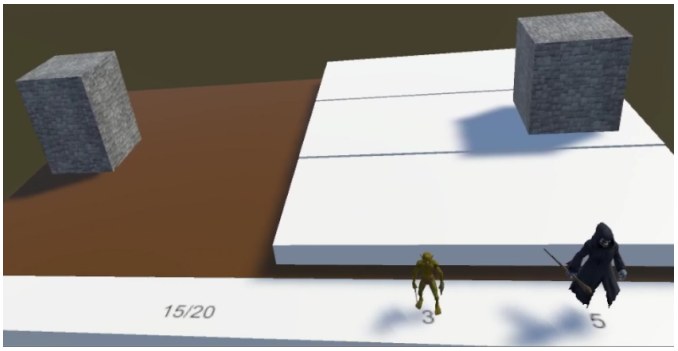

Fig. 4. Screenshot of in-game Mini Board with Wizard and Goblin drag and drop units. Numbers to the left refer to the number of resources obtained out of the max amount. The numbers under the units refer to the amount of resources that will be used per spawn.

As mentioned previously, the goblin and wizard classes both have differences in their ability to deal damage. Out of the two classes, the wizard is stronger and has a higher range for attack. This is reflected in the attack animations for each class. The goblin class has a basic attack animation of swinging their weapon, which requires close combat in order to be effective. The wizard class uses a spell as the main form of attack and this therefore allows them to be more of a ranged specialist. Each units health is shown via a health bar or slider that hovers above the units as can seen in Fig. 5. In order to differentiate between each user, the health bar for the units is green for friendly units and red for enemy units. The health bars are oriented to always look towards the local player so that both the host and the client can clearly see the health of each unit. Each users tower also has a similar health bar that hovers above the front of the tower to make it easy for the user to view in the middle of the simulation by simply looking up.

Fig. 5. Screenshot of spawned goblin unit with green slider that signifies health.

A key game play component to maintain balance and to provide for more strategic opportunities is the implementation of player resources. As each user plays the game, they are able to collect a certain number of resources that max out once a total of 20 resources have been reached. This total can be seen on the Mini Board, shown in Fig. 4, next to the drag and drop units. As a user places and spawns units, they use up resources. The amount of resources used depends on which unit is spawned. For example, the goblin class uses a total of 3 resources per spawn while the wizard class uses 5 resources per spawn. As long as the users resources aren't maxed out they will replenish over time, so the users need to ensure that their resources aren't going to waste.

Aside from the wizard and goblin units, the user has another weapon that they can use to defend their tower from enemy attacks. The user is armed with a slingshot, which can be seen in Fig. 6, that can be used to shoot rocks at the enemy units to provide aid to their spawned units. This also allows the user to continue their defense efforts while waiting for the collection of resources. Once enough resources have been collected, the user can then deploy additional support units. There is no limit to the amount of rocks that can be shot by the user from the slingshot; however, the shooting mechanics aim to be as realistic as possible which can prove to make aiming difficult within the simulation. However, due to this users are unable to damage their own units if they accidentally hit them.

In addition to the game play, our team designed a surrounding environment to engage the user and add to the immersive experience. While the battle would capture the users attention, it is important to ensure that the user feels a sense of presence and therefore make the virtual environment feel as authentic as possible. The game world was created using a Unity terrain map and designed to be a valley surrounded by hills and mountains in the distance. Fig. 7 shows the user viewpoint from the top of their tower at the hills and mountains that

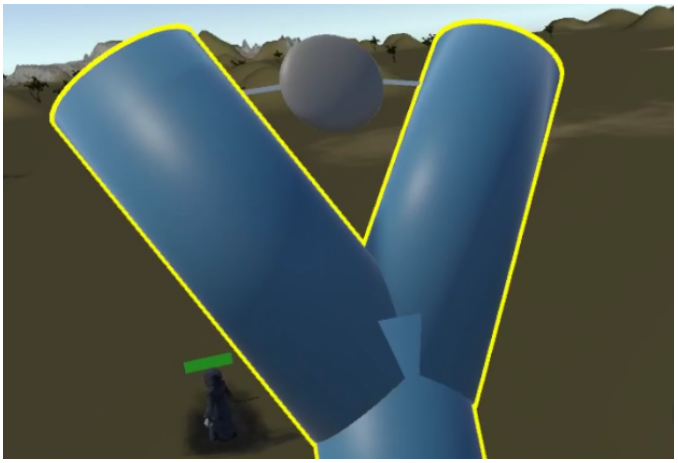

Fig. 6. Displays user preparing to use slingshot weapon to attack a unit.

encompass the area. Grass, dirt and rock textures were added to the terrain map as well as 3D trees from the Unity asset store to provide a sense of realism.

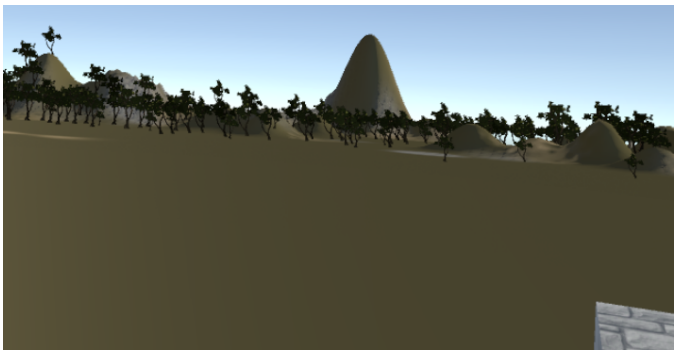

Fig. 7. Shows in-game user perspective of surrounding environment from the top of the tower.

V. Discussion

Following the implementation, our team began to prepare for the final demo of the simulation. In order to prepare, our team played the game to determine fun factor and test for any potential bugs or issues that needed to be fixed prior to the demo. One of the major issues that made it difficult to refine and polish the game was giving users the ability to spawn combat units in the game world. The primary issue was that the client would spawn units but the host would not see the units and therefore would not be able to create an efficient defense. Once this issue was resolved, our team was able to continue and finish preliminary development on the simulation.

As for fun factor, the simulation itself was tested by our team to ensure that everything ran smoothly and was fun for users to play. This was later validated during the final demo showcase. The game was setup using two VR ready computer stations and a connection was made using SteamVR. The live final demonstration was shown following a brief presentation that explained the purpose of the simulation and how it was developed. Feedback was mostly positive with many users giving valuable insights on different ways to improve the simulation and how to make it more immersive and efficient in terms of game play and usability. These improvements and capabilities will be discussed further in Section VII.

Overall, the game is not as fleshed out as we would have liked it to be. Multiplayer was the major roadblock for the development of the game due to VR not being well supported in both Unity and Mirror's networking capabilities. For instance, one of the major obstacles was having the client and objects connect to the server so that the host could see both the opponent and enemy units that were attacking. Due to these development issues, much of the development time was spent trying to ensure the game works in multiplayer rather than being able to add content and functionality to further refine the game prior to the final demo.

VI. Current Status

As it stands currently, Tower Defense in Virtual Reality is still a work in progress. The current iteration has core functionality implemented and some basic game play elements. The core functionality includes multiplayer connection, spawning game objects on the server, and some basic physics to allow for realistic projectile attacks. The main game play elements completed include spawning defense units through the use of an in game mini board of the map, a resource management system, and shooting projectiles at enemy defense units with a slingshot. In addition to the game play, the combat units each have unique attack animations that make the battles between users more immersive and exciting. The simulation also has a terrain map with grass, rock, and dirt textures surrounding the play area to add to the users overall immersion in the game world.

While the core elements of the simulation are complete, there are some additional elements that are currently being added during the writing of this paper. The first of these is the addition of a more interactive user interface. This includes a more interactive and visually appealing menu screen as well as a pause screen that allows users to adjust their settings or exit a match. Voice chat and voice recognition are also elements that are currently being looked into as additional features. More specifically, our team is currently looking into adding voice commands with Watson [15]. Conceptually, this would afford users the ability to use voice commands to spawn defense units as opposed to the drag and drop method that is currently developed. This would make the simulation more efficient and make the game play much more fast paced.

VII. Conclusion And Future Work

This paper describes the development and implementation of a VR simulation entitled Tower Defense in Virtual Reality. The hardware and software used to build the simulation as well as the software design and physical implementation are explained in great detail. Overall feedback and the current status of the simulation are also touched upon. The main idea behind the simulation itself was to see how well multiplayer

can be integrated in VR simulation and how it effects a users sense of presence within the virtual world.

There is much that can be added to improve the experience of the game through future work and development. For instance, a voice chat system would be useful as it would allow the two competitors to communicate as they play the game. In addition to this audio in general would also improve the game in the future. A god view camera would be beneficial to add in the future to allow spectators with or without VR devices to watch the game from a top down view. Game play wise we would like to implement a single player mode where players are able to hone their skills to better compete in the online game play by providing an area to practice their aiming. We would also like to add more weapons and units with different strengths and weaknesses to add more strategy to the game. In addition to this better balancing for the units and slingshot. An easier means of rematching the other player is also necessary for a better user experience. As far as artwork goes we would like to create our own units in the future along with a more immersive and cohesive art style overall. Lastly we didn't need to implement locomotion for the play area we are using but for someone with a smaller play area we wish to implement the node based locomotion.

The project developed has great opportunity for a user study going forward, particularly after more of the future work is implemented. When the game becomes more fleshed out there are many factors that can be looked at in this game. For balancing purposes, extensively testing the game with multiple users will give great insight into which units and weapons are too weak and too strong. Testing would also be helpful in determining the right amount of resources needed and time to replenish those resources. We could also get insight into the game by learning what aspects of the game people like and don't like to provide a better overall experience. Furthermore, we can evaluate aspects such as evolution of strategy and evolution of skill level over time.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation under grant numbers IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] P. Cipresso, I. A. C. Giglioli, M. A. Raya, and G. Riva, "The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature," *Frontiers in Psychology*, vol. 9, p. 2086, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2018.02086

[2] TheMiddleGray, "Castle Must Be Mine." [Online]. Available: https://store.steampowered.com/app/542770/Castle_Must_Be_Mine/ (Last accessed 1/7/2020)

[3] Battle of Kings Team and Wenkly Studio Sp.zo.o., "Battle of Kings VR." [Online]. Available: https://store.steampowered.com/app/778250/Battle_of_Kings_VR/ (Last accessed 1/7/2020)

[4] BitBreak I/S, "In Your Face TD." [Online]. Available: https://store.steampowered.com/app/564330/In_Your_Face_TD/ (Last accessed 1/7/2020)

[5] TreeView Studios, "Alchemist Defender VR." [Online]. Available: https://store.steampowered.com/app/602160/Alchemist_Defender_VR/ (Last accessed 1/7/2020)

[6] T. D. Schepper, B. Braem, and S. Latre, "A virtual reality-based multi-player game using fine-grained localization," in *2015 Global Information Infrastructure and Networking Symposium (GIIS)*, Oct 2015, pp. 1–6.

[7] M. Sra, X. Xu, and P. Maes, "Breathvr: Leveraging breathing as a directly controlled interface for virtual reality games," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: ACM, 2018, pp. 340:1–340:12. [Online]. Available: http://doi.acm.org/10.1145/3173574.3173914

[8] S. Davis, K. Nesbitt, and E. Nalivaiko, "A systematic review of cybersickness," in *Proceedings of the 2014 Conference on Interactive Entertainment*, ser. IE2014. New York, NY, USA: ACM, 2014, pp. 8:1–8:9. [Online]. Available: http://doi.acm.org/10.1145/2677758.2677780

[9] M. P. Jacob Habgood, D. Moore, D. Wilson, and S. Alapont, "Rapid, continuous movement between nodes as an accessible virtual reality locomotion technique," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, March 2018, pp. 371–378.

[10] J. M. Clements, R. Kopper, D. J. Zielinski, H. Rao, M. A. Sommer, E. Kirsch, B. O. Mainsah, L. M. Collins, and L. G. Appelbaum, "Neurophysiology of visual-motor learning during a simulated marksmanship task in immersive virtual reality," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, March 2018, pp. 451–458.

[11] HTC Corporation, "Vive VR System." [Online]. Available: https://www.vive.com/us/product/vive-virtual-reality-system/ (Last accessed 1/7/2020)

[12] Unity Technologies, "Unity." [Online]. Available: https://unity3d.com/unity (Last accessed 1/7/2020)

[13] Valve Corporation, "SteamVR." [Online]. Available: https://store.steampowered.com/steamvr#WhyItMatters (Last accessed 1/7/2020)

[14] Mirror, "Mirror - Documentation." [Online]. Available: https://vis2k.github.io/Mirror/ (Last accessed 1/7/2020)

[15] Watson, "VR Speech Sandbox with Watson Services." [Online]. Available: https://github.com/IBM/vr-speech-sandbox-vive (Last accessed 1/7/2020)