

Computer Games Development

Project Thesis

Year IV

**Winter Wonder – VR Multiplayer Survival**

By

**Filip Rusiecki** [C00250721]

Submitted to the Department of Computing  
in partial fulfilment of the requirements for the degree of  
Bachelor of Science in Games Development  
at the  
South East Technological University  
May 2023  
© South East Technological University 2023. All rights reserved.

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  
 Department of Computing  
 May 1, 2023  
Certified by. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  
 Lei Shi  
 Assistant Professor  
 Thesis Supervisor

# C:\Users\tom\SkyDrive\Documents\HDip in Science in Computing\Electronic Plagiarism Declaration Undergraduates.png

|  |  |
| --- | --- |
|  |  |
|  | |



**Contents**

Acknowledgements**…………………………………………………..…..…1**

Project Abstract**………………………………………………….……………2**

Project Introduction and/or Research Question**….……………3**

Literature Review**……………………………………………….……………4**

Evaluation and Discussion**…………………………………………….…5**

Project Milestone**……………………………………………………………..6**

Major Technical Achievements**………………………………….……7**

Project Review**…………………………………………………………………8**

Conclusions**………………………………………………………………………9**

Future Work**…………………………………………………………………….10**

References**…………………………………………………………………….….11**

Appendices**………………………………………………………………………12**

# 

# Acknowledgements

I would like to thank the following people who assisted in completing this project including:

-Ben Williamson (Art Student 4th year) for creating 3D Models for my game.

# Project Abstract

The VR game genre became popular in recent years, slowly more and more people try to get their hands on VR technologies this allows users to plunge into virtual reality environment. The outcome of new VR headsets such as HTC VIVE, Oculus Rift, Samsung VR and many more stimulated the rapid VR development in the industry in recent years.

On the other hand, Survival game genre has been on the market for a long while yet always brining something new to excite people and making the companies have competition on which survival game is better yet always surprising players by brining something different to the market. Some survival games after realising they end up doing a version of the game for a VR which always attracts new players. With my game being made in VR I want to bring my own game to a great standard that will have some new features that will make my game unique.

I chose to create this game in a game Engine that is very known and it is Unity game Engine, I will try to create a very optimised game that will run smoothly on any VR headset and I was also wanting to bring something new to that game genre.

Thesis Supervisor: Lei Shi  
Title: Assistant Professor

# Project Introduction and/or Research Question

The Project I have decided to pick and do is a VR Multiplayer Survival game.

**Why VR?** I have never worked with VR or never seen how that process is made. This project also was chosen because I wanted to learn about VR games as in spare time, I’m trying to develop games/ modifications for different games and releasing them online.

**Why Survival game for the first game in VR?** Having learning about game development I wanted to do something that would challenge me and at the same time educate me and something that would expand my knowledge.

The game will present the player with a cold environment in which they will have to survive and beat few goals of the game to complete it. In the game the user will face different puzzles and different Advanced AI that will try to stop the player from completing the goals. The players main goal is to fix a crashed plane that they will use to escape the dangerous area in which they are trapped in.

**Why multiplayer?** I have never coded multiplayer game in unity engine. While playing any game almost definitely every game is better when you play it multiplayer. Its great to interact with other players and friends in a virtual environment. It is going to be challenging to make as VR and Multiplayer are hard to implement into games but I believe the I will manage to get the two systems working with each other.

**How will this project challenge me?** This project will challenge me in several different ways. First one is that I never worked with VR technology. This will require me to research into the topic on how to create a project file from scratch to suit a specific VR set (HTC VIVE), Input system will be difficult and controlling of any object by the player.

Another challenge that I will face is creating advanced options and controls for the player and also AI that player will be facing against. The AI will need different edge cases not to get stuck or bugged out to provide the player with smooth gameplay.

A big challenge that I will face is the graphics of the game, as many VR games if not done correctly they can make the user feel motion sickness so I have to make sure that the graphics of the game or camera movement will not cause it. Also while creating a game I need to remember that the game is created for VR, that cant be overloaded with very high graphics in order to run smoothly without interruptions.

Another part of the project which I will find challenging to do is multiplayer as I also never really looked into making games for multiplayer using unity engine.

Implementing Unity Mirror Multiplayer can be challenging for game developers, especially for those who are new to multiplayer game development. The technology requires a solid understanding of networking, synchronization, and game optimization, which can be difficult to master. Game developers need to ensure that the game is optimized for multiplayer, and that the game logic is properly synchronized across all players. They also need to consider the potential for lag or network interruptions, which can affect the gameplay experience. However, with the right knowledge and expertise, implementing Unity Mirror Multiplayer can provide a highly rewarding multiplayer gaming experience.

# Literature Review

Survival game genre has been on the market for a long while yet always brining something new to excite people and making the companies have competition on which survival game is better yet always surprising players by brining something different to the market. Some survival games after realising they end up doing a version of the game for a VR which always attracts new players or players that have already played their game on a non-VR platform. With my game being made in VR I want to bring my own game to a great standard that will have some new features that will make my game unique.

There are already existing games that made similar type of projects which were a success such as game called Green Hell VR, they used VR but not multiplayer technologies. [(References)](#_References)

One of the most significant projects that influenced my own project is a game called VR Chat. This game is a virtual reality experience that allows players to connect with one another and engage in a wide range of activities, including playing games, chatting, and exploring virtual environments. [(References)](#_References)

One of the things that makes VR Chat so unique is its ability to support both VR and traditional keyboard input. This feature makes it accessible to a wider range of players, as those without VR equipment can still enjoy the experience. The game was built using the Unity Engine, which is one of the most popular game engines in use today.

As a game developer, I found VR Chat to be an inspiring and informative project to study. I was particularly interested in the game's networking capabilities, which allow players to connect with one another in real-time. This feature was crucial to the success of my own project, as I wanted to create a multiplayer experience that was seamless and enjoyable for all players.

I also studied the game's user interface and gameplay mechanics to gain insights into what makes a successful virtual reality experience. VR Chat's user interface is intuitive and easy to navigate, making it accessible to players of all skill levels. The gameplay mechanics are engaging and interactive, encouraging players to explore and experiment with different features.

Overall, VR Chat served as a valuable source of inspiration and guidance for my own project. By studying this game and others like it, I was able to gain a deeper understanding of what makes a successful virtual reality experience. I was also able to learn from the successes and failures of other game developers, allowing me to make informed decisions and avoid common pitfalls.

VR chat also has “The VRChat Documentation Hub” where they explain how they did all the things in the project and it can be found here. [(References)](#_References)

# Evaluation and Discussion

Creating a project, whether it's a game, a software program, or any other kind of project, can be a challenging task that requires a lot of time, effort, and expertise. While creating the project, you can encounter various problems that can be catastrophic to the project. As you're likely aware, even the smallest of issues can snowball into major headaches that consume valuable time and resources.

As an example, when I was working on creating multiplayer functionality for my game, I ran into a significant problem. Specifically, a package that I had integrated into my project, which was Fish-net Networking, ended up breaking my entire project. This was a frustrating setback for me, as I had to perform multiple reverts in order for the project to work properly again. This was a time-consuming process that set my project back significantly.

After that issue, I had to take a step back and reassess my approach. I knew that I needed to look for another library which provided me with the same things as the other one. I was fortunate that I was able to find another library that worked well, but it could have been a lot worse.

After finding the correct open-source library to integrate into my project, I thought the worst was behind me. However, I soon discovered that yet another issue was on the horizon. Specifically, the new library that I had added to the project, called Mirror Networking, clashed with another library that I had inside the project which controlled all my VR scripts.

As you might imagine, this was a major problem, and one that required a significant amount of time and effort to resolve. I spent hours digging through all of my scripts, trying to locate the specific lines of code that were causing the issue. This was a frustrating and time-consuming process, as there were dozens of scripts that needed to be examined.

Once I had identified the problematic scripts, I had to edit them to ensure that they were compatible with the new library. This was another tedious process that required a great deal of care and attention to detail. I had to go through each line of code, making sure that everything was properly aligned and that there were no conflicts.

After making the necessary edits, I had to go through a lengthy testing process to ensure that everything was working properly. This was a time-consuming and tedious process, as I had to test each and every feature of my game to make sure that it was functioning as intended. It was important to ensure that the game was stable and that there were no lingering issues that could cause problems down the line.

In conclusion, finding the correct open-source library to integrate into a project can be a challenging process. Even after finding the right library, there can be issues that arise that require a great deal of time and effort to resolve. In my case, I had to spend a significant amount of time editing and testing scripts to ensure that everything was working properly. It was a tedious process, but one that was necessary to ensure the stability and functionality of my project.

# Project Milestone

**Reaching the first milestone** of a project is always an exciting achievement. In this particular project, the first milestone was accomplished after successfully adding Steam VR to the project. This was a major milestone because it opened up a whole new world of possibilities for the project, allowing for the creation of an immersive and interactive virtual reality experience. The process of integrating Steam VR into the project was not an easy one. It required a significant amount of research and experimentation to ensure that the player was able to move around the virtual environment without any issues. One of the primary concerns was making sure that the player did not fall through the ground, which required careful attention to the physics engine and the creation of a new player controller.

After the initial setup was completed, I was able to create a simple test area to ensure that everything was working as intended. This involved testing the physics engine to ensure that the player could move around the environment without clipping through objects or experiencing any strange behaviour. This was an important step in the development process, as it allowed me to identify any potential issues early on and make adjustments as necessary. In order not to clip through the objects It was decided to make a script for the VR camera so that the player collider will follow the players camera which is players head in our situation. Overall, reaching the first milestone of the project was a significant accomplishment that required a lot of hard work and dedication. It laid the foundation for the rest of the project and set the me up for success in creating an engaging and immersive virtual reality experience.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

**Reaching the second milestone** was a crucial step in the development of the game. After successfully integrating Steam VR into the project, it was time to create a more immersive gameplay experience for the player by allowing them to shoot with guns in the game.

In order to achieve this, I created a customizable gun script that would allow me to create any weapon I desired. This script saved me a lot of time as it could be attached to any object, making it a weapon with any desired properties. It was a versatile tool that could be used in various ways and added to different weapons to make them unique.

However, creating the gun script came with its own set of challenges, particularly in regards to VR input. As the game was being developed for VR, it was essential that the guns were controllable with VR input. I had to consider how the gun would be held and fired in VR, and how the controls would feel to the player.

To overcome these challenges, I had to experiment with different input mappings and configurations. I had to adjust the settings until I found the perfect balance between intuitive controls and realistic gameplay. It was a trial and error process that required a lot of patience and perseverance, but ultimately, it paid off.

With the customizable gun script and the VR input controls in place, the player could now shoot in the game, adding a new level of interactivity and excitement to the gameplay. It was a significant achievement that brought the game one step closer to completion.

A screenshot of a computer program

Description automatically generated with low confidenceA screenshot of a video game

Description automatically generated

A screen shot of a computer program

Description automatically generated with medium confidence

A screen shot of a computer program

Description automatically generated with medium confidence

**Reaching the third milestone** of the project involved designing and creating the game environment, which was an enjoyable experience for the development process. The decision was made to create a large-scale world for players to explore, with a focus on creating an immersive experience. However, I had to be cautious about the performance impact of rendering large amounts of foliage in a VR headset, and thus ensured that all the foliage had a low level of detail when viewed from a distance.

To optimize the game's performance, I also created the project with low poly models, which not only made the game run more smoothly, but also added to its overall aesthetic appeal. The process of creating the world was challenging yet rewarding, as it allowed the me to experiment with various design elements and explore different artistic styles.

One of the most exciting aspects of creating the world was the opportunity to collaborate with a 4th year art student who provided some of the models. This collaboration allowed the me to access a wider range of assets and to interact with someone who had a different perspective and approach to art and game design. This experience helped the developer to learn and grow as a game developer.

In summary, the process of creating and designing the game world was a rewarding experience for the me as a developer. It provided an opportunity to experiment with different techniques and styles while also learning how to optimize performance for VR. The end result was a beautiful and immersive world that players could explore and enjoy.

**Reaching the fourth milestone** of the project was a critical moment for me as it marked a significant leap forward in the game's design and functionality. The decision was made to create an interactable plane that players could control and fly, adding a new level of excitement and immersion to the game.

To ensure the plane was easy to control and enjoyable to fly, It was put a lot of thought into the details. For example, they designed the plane to have a turning speed that wasn't too fast, as this would make the game difficult to control and potentially frustrating for players. Additionally, it was made sure that the plane's movements weren't too extreme - for example, the plane wouldn't be able to spin vertically upwards or downwards fully. This was done to prevent players from feeling disorientated or getting motion sickness from playing the game.

Creating a plane that was both fun and comfortable to fly required a lot of experimentation and trial-and-error. I also had to think creatively and come up with innovative solutions to ensure that players wouldn't get nauseous or feel sick when playing the game. Also had to consider the various factors that could contribute to motion sickness, such as the plane's speed, altitude, and orientation.

One of the most exciting aspects of designing the plane was the opportunity to experiment with new technologies and control schemes. For example, the implementation of VR touchpad system that allowed players to control the plane using specific positions on the touchpad. This was a new experience for the creation process, and it took a lot of time mapping different positions on the touchpad to different functions within the game. However, the effort paid off, and the VR touchpad system added a new level of immersion and excitement to the game.

Overall, reaching the fourth milestone of the project was a significant achievement for the development. The creation of an interactable plane added a new dimension to the game and required a lot of thought and experimentation to get right. However, by carefully considering the various factors that could contribute to motion sickness and implementing innovative control schemes, I was able to create a plane that was both fun and comfortable to fly.

**Reaching the fifth milestone** of the project was a critical moment for the development as it marked a significant leap forward in the game's design and functionality. The decision was made to enable multiple players to connect to a single world, allowing them to play together in real-time. Players would be able to connect to a host using IP. While this feature added a new level of excitement and social interaction to the game, it also presented some significant challenges for the development. Another challenge was designing the game world in a way that would be interesting and engaging for multiple players. However, building this functionality was not without its challenges, and I had to invest a significant amount of time and effort into ensuring a smooth and secure multiplayer experience.

**Reaching the sixth milestone** of the project was a crucial moment for the development, as decided to implement a waiting lobby that would improve the overall user experience of the game. This feature enabled players to either choose to host a game or connect to a game using an IP address.

The waiting lobby provided a central hub where players could gather and interact with each other before the game started. Once players were connected to the waiting screen, they had to ready up before the game could start. This feature ensured that all players were prepared and ready to play, preventing any delays or disruptions during the game.

To make the waiting lobby even more user-friendly, I implemented a "Ready" button that appeared next to each player's name. Players could click this button to indicate that they were ready to start the game. When all players had clicked the button, the game would start automatically.

To prevent any unfairness or confusion during the waiting period, I made sure that players could only ready up themselves and not other players. This ensured that each player had control over their own readiness status, and that no one could start the game prematurely.

The waiting lobby also provided an opportunity for players to interact with each other and build anticipation for the game. This social aspect of the game added an extra layer of enjoyment and excitement for players, as they could chat, make plans, and get to know each other before the game started.

Overall, the implementation of the waiting lobby was a significant achievement for the development. This feature improved the overall user experience of the game, ensuring that players could connect and prepare for the game without any delays or disruptions. By making the waiting lobby user-friendly and social, I was happy to add an extra layer of enjoyment to the game, making it more engaging and exciting for players.

**Reaching the seventh milestone** of the project was a critical moment, as it marks a significant milestone in the development process. In this project, reaching the seventh milestone was especially important, as it marked a major turning point in the project's progress.

The next step had decided to implement a new feature that would allow players to experience objects in the multiplayer world in real time. This meant that players would be able to interact with objects in the game environment, and other players would be able to see those interactions in real time.

To achieve this level of interactivity, I had to ensure that each object that players could interact with was synced across all players' screens. This was no small feat, as it required a significant amount of attaching lots of components to objects to ensure that all players actions were accurately and quickly transmitted to all other players in the game.

The result of this effort was a more immersive and engaging multiplayer experience, where players could pick up objects, move them around, and see the impact of their actions on the game world in real time. I was proud of what I had achieved.

Overall, reaching the seventh milestone in this project was a significant achievement, as it marked a turning point in the development process and paved the way for a more engaging and immersive multiplayer experience. I had worked hard to ensure that the syncing of objects across all players' screens was seamless and accurate and the end result was a game that was more fun and interactive than ever before.

# Major Technical Achievements

As a developer, it is important to constantly expand one's knowledge and skills in order to stay up to date with the latest trends and technologies. In this case, I had the opportunity to gain valuable experience and knowledge in the areas of multiplayer game development and virtual reality (VR) systems.

Developing a multiplayer game requires a deep understanding of network architecture, game synchronization, and player interaction. By working on a multiplayer game project, I as a developer was able to gain hands-on experience with these topics, as well as learn about the challenges and opportunities that come with developing games for multiple players.

In addition, I also had the opportunity to work with VR systems and input devices. VR is an exciting and rapidly growing area of game development, with new technologies and devices being introduced all the time. By gaining experience with VR systems, I had learned about the unique challenges and opportunities that come with designing and implementing games for VR, as well as the different input devices that can be used to create immersive and engaging experiences.

Overall my experience in working on multiplayer games and VR systems has provided valuable insights and knowledge that can be applied to future projects. By continuing to learn and expand their skills in these areas, I can stay at the forefront of the latest trends and technologies in game development and continue to create innovative and engaging experiences for players.

# Project Review

As I continued to work on the project, I encountered a plethora of obstacles that I had not anticipated when I first began. The complexities of working with VR and multiplayer technologies became more apparent, and I realized that I needed to dig deeper and learn more in order to succeed. It was a challenging journey, but it ultimately helped me to grow as a developer.

One of the biggest challenges that I faced during the project was related to the Fish-net Networking library that I had initially chosen to use. While it had been recommended to me as a popular choice among Unity developers, I found that it was quite complex and difficult to work with. It was particularly frustrating to encounter the issue with cloning a copy from GitHub that caused my entire project to break. I was disheartened and felt like I was taking steps backward rather than moving forward.

(This error has fixes to it but no tutorial or fix was working in my situation) A screenshot of a computer

Description automatically generated

However, I did not let this setback defeat me. Instead, I decided to switch to the Mirror Networking library, which was a more user-friendly and straightforward option. This change required significant modifications to my code, but I felt it was worth it. I was pleased to see that the project was no longer crashing, and I was able to proceed with implementing multiplayer functionality.

Another significant challenge that I faced was related to testing the game in VR with parallel sync. It was a complex and time-consuming process to ensure that everything worked correctly in both VR and multiplayer. It was particularly frustrating to discover that all the players were trying to use the same VR headset, which made testing impossible. However, I persevered and found a workaround that allowed me to continue testing despite this difficulty.

Despite the challenges, I am grateful for the experience that this project gave me. I learned a great deal about working with VR and multiplayer technologies, and I gained valuable insights into network synchronization and latency. I am confident that this knowledge will be useful in future projects, and I am excited to continue exploring these fields further.

During my project, I encountered another significant challenge related to the integration of Steam VR and Mirror networking libraries. I had originally planned to use both of these libraries to enhance the VR and multiplayer aspects of the game. However, as I began to implement both libraries, I quickly realized that they were clashing and not working together as I had hoped. The main issue was that both libraries had their own references to the keyword "Player" in their respective scripts. As a result, when I tried to run the project, there were conflicts between the two libraries as they were trying to access the same properties simultaneously. This made it impossible to run the project without first addressing this issue. To solve the problem, I had to make significant changes to the VR properties in the Steam VR scripts. This was a daunting task as I had to go through each script and modify it to ensure that there were no conflicts with the Mirror networking library. It was a time-consuming process, but it was necessary to get both libraries working together and to ensure that the project was functional. In retrospect, this challenge taught me the importance of careful planning and consideration when integrating multiple libraries and frameworks. It highlighted the importance of understanding the intricacies of each library and the potential conflicts that may arise when using them together. While it was a difficult experience, it was also a valuable lesson that will help me to approach future projects with more caution and forethought. [(click here)](#_Appendices)

Overall, I am proud of the work that I accomplished and the progress that I made. It was a journey filled with ups and downs, but I emerged from it with a greater understanding of these technologies and a renewed sense of determination.

Every development project has its share of challenges, and one of the biggest is often time management. Despite the best efforts of development, there are often tasks that need to be left until later or pushed aside entirely in order to meet deadlines For this particular project, I had made great progress in creating an engaging and immersive gameplay experience, but as the project deadline approaches, there are a few important features that may need to be left out in order to deliver the game on time.

One of these features is the ability for players to choose their preferred input method, such as a keyboard, controller, or VR headset. While this is certainly an important feature that would add to the game's accessibility and appeal, it may not be possible to implement it in time for the project's deadline. This could be due to a variety of reasons, such as the complexity of the control scheme or the need for extensive testing and optimization to ensure that the different input methods work seamlessly together.

However, if there were enough time, the next developer could certainly prioritize implementing this feature if project was to continue. Giving players the ability to choose their preferred input method can greatly enhance the overall gameplay experience and make the game more accessible to a wider audience. For example, players who are more comfortable with traditional keyboard and mouse controls may prefer to use this input method, while others who are more familiar with console gaming may prefer to use a controller. Additionally, players who have invested in a VR headset would be able to fully immerse themselves in the game world, using their own body movements and gestures to interact with the environment.

Overall, allowing players to choose their preferred input method is an important feature that can greatly enhance the accessibility and appeal of any game. While it may not always be possible to implement this feature due to time constraints, I will certainly strive to do so if there is enough time available. Ultimately, the goal is to create a fun and engaging game that can be enjoyed by as many players as possible, regardless of their preferred input method.

In the world of game development, setbacks are an unfortunate but inevitable part of the process. Sometimes, despite the best intentions and efforts of developers, a project can run into unexpected challenges or roadblocks that derail progress and set back timelines. For me as a developer of this particular project, one of the key lessons learned was the importance of doing thorough research and planning before embarking on a new project.

Looking back on the experience, I realized that there were a few key areas where more research and planning could have helped to prevent some of the issues that arose during development. One of these areas was the choice of libraries to use for the project. While I had done some preliminary research into different libraries, I realized in hindsight that more extensive research and testing could have helped to identify potential issues or drawbacks before starting the project. As a result, when the project hit a major setback, for me as a developer I was forced to spend a significant amount of time and effort troubleshooting and finding workarounds to get the project back on track.

Another key lesson learned was the importance of careful time management and planning. In any project, it's important to set clear timelines and milestones, and to build in extra time for unexpected setbacks or delays. By doing so, I can ensure that I have enough time to complete all necessary tasks and can avoid the stress and pressure of rushing to meet deadlines. In this particular case, I have realized that more careful planning and time management could have helped to avoid some of the issues that arose during development and could’ve helped to ensure that project was completed on time.

# Conclusions

In conclusion, Unity Mirror Multiplayer technology has revolutionized the multiplayer gaming experience in VR, providing game developers with the tools they need to create immersive and engaging multiplayer games. However, implementing this technology is not without its challenges, as game developers must navigate the complexities of networking, synchronization, and game optimization.

One of the biggest challenges of developing multiplayer games in VR is the need to ensure seamless gameplay. Even minor delays or interruptions in network connectivity can lead to a poor user experience, which can be frustrating for players. As a result, game developers must work tirelessly to account for potential lag or network interruptions and ensure that their games are optimized for performance.

Additionally, creating multiplayer games in VR requires a different set of skills and expertise than developing single-player games. Game developers must familiarize themselves with new VR systems, tools, and technologies and must work to optimize their games for the unique requirements of the VR environment.

Despite these challenges, Unity Mirror Multiplayer technology has enabled game developers like myself to create engaging and immersive multiplayer games that provide players with a highly rewarding experience. By leveraging the power of this technology, we can create games that allow players to connect with each other in real-time and explore new worlds together.

Of course, developing multiplayer games in VR is not without its setbacks. During the development process, I encountered several challenges, including issues with networking and game optimization. However, with perseverance and determination, I was able to overcome these obstacles and deliver a game that I am proud of.

In retrospect, I believe that my lack of prior knowledge of VR or multiplayer technologies made this project even more challenging. However, with the help of online resources and forums, I was able to learn the necessary skills and knowledge to bring this project to fruition.

Overall, I believe that Unity Mirror Multiplayer technology has opened up a world of possibilities for game developers, enabling us to create innovative and immersive multiplayer games in VR. Despite the challenges, I believe that the rewards are well worth the effort, as multiplayer games in VR provide players with an unparalleled gaming experience that they will never forget.

# Future Work

If a student were to undertake a project in the area of Unity VR multiplayer, there are several interesting next steps that they could consider exploring:

1. Allowing for players to play with any input they want such as Controller, Keyboard or VR. So that any players playing the game could have a choice between different inputs they want to play on so the game wouldn’t be limited to only play it with VR.

2. Proper AI integration: The integration of AI into Unity Mirror Multiplayer could provide an exciting opportunity for game developers to create highly engaging and challenging multiplayer experiences. The implementation of AI-controlled non-player characters could enhance the realism of the game and create new challenges for players. Even creating a bot that can do anything that player can.

3. Dynamic environments: The development of Unity Mirror Multiplayer games that feature dynamic environments, such as changing weather or destructible terrain, could provide an exciting and engaging multiplayer experience. This could involve the implementation of advanced physics systems, such as Unity's Physics System or NVIDIA's PhysX, to create more realistic and interactive environments. Procedural world generation could be included into this which would create a much better experience for the players as it would continuously create a random chunks of world for the player to explore. This feature would add to the replay ability to the game.

# References

* Unity XR Input, Unity Technologies (2023) [Online] Available at: <https://docs.unity3d.com/Manual/xr_input.html> (Accessed:24-04-2023)
* Input for OpenVR controllers, Unity Technologies (2023) [Online] Available at: <https://docs.unity3d.com/560/Documentation/Manual/OpenVRControllers.html> (Accessed:24-04-2023)
* William, How to Perform VR Interaction With Objects [Unity Tutorial] (2023) [Online]Available at: <https://arvrtips.com/vr-interaction-with-objects/> (Accessed:24-04-2023)
* Green Hell VR on Steam (2023) [Online] Available at: <https://store.steampowered.com/app/1782330/Green_Hell_VR/> (Accessed:24-04-2023)
* Open Source Networking for Unity (2023) [Online] Available at: <https://mirror-networking.com/> (Accessed:24-04-2023)
* Verior Pies, ParrelSync: (Unity3D) Test multiplayer without building (2023) [Online] Available at: <https://github.com/VeriorPies/ParrelSync> (Accessed:24-04-2023)
* VRChat, Game website [Online] Available at: <https://hello.vrchat.com> (Accessed:25-04-2023)
* VRChat, The VRChat Documentation [Online] Available at: <https://docs.vrchat.com> (Accessed:25-04-2023)

# Appendices

Error! Not a valid bookmark self-reference.Using the VR input system was a trail and error as things didn’t go as easy as plannedA screenshot of a computer

Description automatically generated with medium confidence

The VR scripts I had to change in order to run the project.

[PlayerVRRR.cs]

A screenshot of a computer program

Description automatically generated with medium confidence

[Hand.cs]

A screen shot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated with low confidence

A picture containing text, screenshot

Description automatically generated

[HandPhysics.cs]

A picture containing text, screenshot, font, software

Description automatically generated

[SnapTurn.cs]

A screen shot of a computer program

Description automatically generated with low confidenceA picture containing text, screenshot, font, software

Description automatically generatedA picture containing text, screenshot, software, font

Description automatically generated

[DebugUI.cs]

A screenshot of a computer program

Description automatically generated with medium confidence

[RenderModels.cs]

A screen shot of a computer code

Description automatically generated with medium confidence

[Throwable.cs]

A screenshot of a computer program

Description automatically generated with medium confidence

[ControllerButtonHints.cs]

A screen shot of a computer

Description automatically generated with low confidence

A picture containing screenshot, text

Description automatically generated

[Arrow.cs]

A picture containing screenshot, text

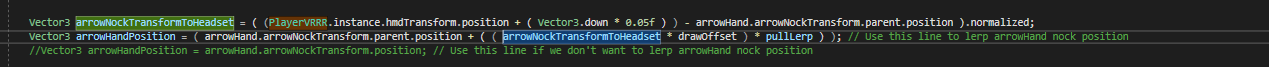
Description automatically generated

A picture containing text, screenshot

Description automatically generatedA picture containing text, screenshot, font

Description automatically generated

[Longbow.cs]



[JoeJeffGestures.cs]

A picture containing text, screenshot, software, multimedia software

Description automatically generated

[ControllerHintsExamples.cs]

A screen shot of a computer

Description automatically generated with medium confidence

[FlowerPlanted.cs]

A picture containing text, screenshot, font, software

Description automatically generated

[Planting.cs]

A picture containing text, screenshot, font

Description automatically generated

[SkeletonUIOptions.cs]

A screen shot of a computer program

Description automatically generated with medium confidence

[SnapTurn.cs]

A screen shot of a computer program

Description automatically generated with medium confidence

[Teleport.cs]

A screen shot of a computer

Description automatically generated with low confidenceA picture containing text, screenshot, font

Description automatically generated

[TeleportArc.cs]

A screen shot of a computer program

Description automatically generated with low confidence

[TeleportPoint.cs]



A screen shot of a computer

Description automatically generated with low confidence