

**Two Wheelistic: Development of a High-Fidelity Virtual Reality Cycling Simulator for
Transportation Safety Research**

by

Ayah Hamad

**A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
(Human-Centered Design and Engineering)
in the University of Michigan-Dearborn
2021**

Master's Thesis Committee:

**Assistant Professor Fred Feng, Chair
Professor Bruce Maxim
Associate Professor Sang-Hwan Kim
Assistant Professor Feng Zhou**

Ayah Hamad

ayhamad@umich.edu

ORCID iD: 0000-0001-6776-5495

© Ayah Hamad 2021

Dedication

To Mama and Baba

Thank you for all the sacrifices you made for me to be where I am today.

Acknowledgements

I would like to start with thanking my committee chair, Dr. Fred Feng, for taking me in as a research assistant, believing in my abilities, and being a wonderful guide through each stage of the thesis process.

To Dr. Bruce Maxim, thank you for giving me a strong foundation in game design on which I have been able to build my research on.

To Dr. Sang-Hwan Kim, thank you for being a kind and supportive advisor who was always willing to answer all my questions and concerns.

To Dr. Feng Zhou, thank you for igniting my interest in the field and offering me an opportunity to develop a unique VR experience that informed my work on this thesis.

I am also grateful to the rest of my instructors in the Human Centered Design & Engineering department that have been instrumental in my success as a master's student.

Most of all, I am grateful for the support of my family and friends throughout my academic journey. I would not have made it this far without your support, encouragement, and reassurance of your faith in me. I am especially thankful to my parents, who have raised me to persevere through adversity and have been by my side through thick and thin.

Table of Contents

Dedication.....	ii
Acknowledgements.....	iii
List of Figures.....	vi
Abstract.....	vii
Chapter 1 Introduction.....	1
Chapter 2 Literature Review.....	4
2.1 Utilizing Simulators in Transportation Safety Research.....	4
2.2 Cybersickness.....	7
Chapter 3 Simulator Hardware Development.....	11
3.1 VR Headset System.....	13
3.2 Bicycle & Stationary Bike Trainer.....	14
3.3 Miscellaneous.....	14
Chapter 4 Simulator Software Development.....	15
4.1 VR Simulation Software - Unity®.....	15
4.1.1 Unity® Editor & Scene Creation.....	16
4.1.2 Environment & Scene Objects.....	18
4.2 Phase 1 – Interactive Simulation.....	20
4.3 Phase 2 – Dangerous Driver-Bicyclist Encounter Scenarios Simulation.....	22
4.3.1 360° Videos.....	25
4.4 Usability Improvement – Adding Simulation Menus.....	26

Chapter 5 Simulator Calibration & Validation	29
5.1 Setup and Calibration	29
5.2 Scenario Validation	30
5.3 Testing User Controls.....	31
5.4 Speed Validation	32
5.5 User Validation	33
Chapter 6 Discussion	35
6.1 Use Cases	35
6.2 Future Improvements	37
Appendix.....	39
References.....	41

List of Figures

Figure 3-1: Virtual Reality (VR) Cycling Simulator Setup.	12
Figure 3-2: System Components (note: features in gray are not currently implemented).....	13
Figure 4-1: The Unity® user interface, including scene and game views, hierarchy, inspector, and project windows.	17
Figure 4-2: Editor showing the VR Cyclist Object in detail.....	19
Figure 4-3: Examples of the dangerous encounter scenarios that one can experience in the cycling simulator.....	23
Figure 4-4: Active Recorder window ready to record in the Unity® editor.	25
Figure 4-5: Simulation Menu System.	27
Figure 5-1: Testing simulation controls (pedaling and controller input) in a Unity® scene with a simple environment.	31
Figure 5-2: Speed validation scene showing timer with set distance.	33

Abstract

This thesis presents the development of an immersive, high-fidelity virtual reality (VR) cycling simulator, where one can ride a stationary bicycle in a simulated virtual environment and interact with other road users (e.g., drivers). Inspired by driving simulators, a VR cycling simulator has potential to become a valuable tool for conducting traffic safety research involving bicyclists. The hardware and software development and integration were described in detail as a reference for others that may want to build similar systems. The VR simulation includes a representation of a real-world urban environment with a road network, and utilizes a VR headset coupled with an appropriate stationary bike system setup.

The first phase of development was focused on an immersive, interactive simulator, in which users are able to control their movements within the virtual environment. They control their speed by pedaling the stationary bike and can steer using buttons on the handlebar-mounted controllers. This aspect of the simulator is instrumental in applications that require observing participants' cycling behaviors in a safe, virtual environment. Validation was performed to ensure the cycling simulator functioned at realistic speeds and in accordance with the user's input. The second development phase focused on a use case for driver education and training, presented through a variety of common dangerous bicyclist encounters programmed into VR scenarios. Drivers who have limited bicycling experience can experience these scenarios in a safe, virtual setting to better understand the bicyclist's perspective.

Chapter 1 Introduction

Bicycles were first invented in the 19th century and have been around for over 200 years. As of 2000, there are an estimated over one billion bicycles in existence (Balton, 2021). Many around the world depend on bicycling as their principal means of transportation, while others are advocating for an increase in cycling as a promising method to tackle some of the biggest challenges society faces today, especially in the United States. Increasing our use of bicycles will help us fight climate change by reducing fossil fuel consumption and CO2 emissions from private motor vehicles (EPA, 2018), reduce air and noise pollution (Babisch et al., 2001; EPA, 2019), and make our cities and communities more healthy, affordable, and equitable.

In more recent years, there has been a growing trend of bicycling in the US (Pucher & Buehler, 2017), which is likely to continue increasing with urbanization and innovations such as shared, electric, and cargo bikes. An interesting phenomenon regarding bicycle usage is their sharp increase during the Covid-19 pandemic. More people wanted to exercise due to stay-at-home orders limiting their activities, which increased sales in both outdoor leisure bicycles and indoor bicycling equipment (Davies, 2020; The NPD Group, 2020). Inexperienced cyclists were more likely to ride on the roads with the decrease of vehicle traffic, as they felt safer doing so than in normal circumstances. Another aspect is wanting to stay safe; commuters preferred to use bicycles over public transportation to maintain social distancing (Davies, 2020). Although the Covid-19 pandemic was a catalyst for this phenomenon, it presents a dramatic shift towards higher demand for non-car transport accommodations in the form of pedestrian and cycling

infrastructure (Combs & Pardo, 2021). Research is currently underway on the implications this will have on future infrastructure efforts and designs after the pandemic.

Despite this growth in bicycle usage and the benefits it brings, cycling as a mode of transportation is still greatly underutilized in the US, with only 0.5% of American commuters choosing this option (Census Bureau, 2018). The safety issues that come with riding a bicycle on the road with motorized traffic have been one of the major deterrents for many people choosing bicycling as their transportation alternative (Jacobsen et al., 2009), as cyclists are inherently vulnerable road users who get little protection in the event of a collision with a motor vehicle. Every year in the US, approximately 800 bicyclists are killed and an additional 45,000 are injured from motor vehicle crashes (NHTSA, 2019, 2020).

Driving simulators have been effective in testing road-related safety solutions, which has inspired the development of cycling simulations for the same purpose. There are various solutions for improving bicycle safety that could be tested or implemented with the use of cycling simulators, which would allow users to experience bicycling safely in a simulated virtual environment. Proposed solutions include effective safety education and training programs, the introduction of bicycle safety regulations and laws, improving roads and bicycle infrastructure, and advances in safety technologies.

Virtual reality (VR) technologies have a wide range of applications, ranging from research, design, engineering, business, education and training, and entertainment. VR simulations are especially useful in research, as they can be used to conduct experiments in an immersive, safe, and replicable environment (Cipresso et al., 2018). The use of VR simulations for experiments that involve road users or traffic safety allows for observation and data collection based on a high-fidelity setting, without endangering any of the participating subjects.

Accordingly, the development of a cycling simulator equipped with state-of-the-art VR technologies provides an immersive, high-fidelity cycling experience that can be a valuable tool in many contexts. It can be used to understand bicyclists' behavior and their interactions with other road users (e.g., drivers, pedestrians) and their preferences on road and bike infrastructure. It can also be an effective tool for both driver and bicyclist education and training, as the simulator can provide a riding experience in a first-person perspective that allows users to better understand what a bicyclist may experience and encounter on the road.

This thesis presents our method of developing an immersive, high fidelity VR cycling simulator. The hardware components utilized in this simulation are all commercially available, and are described in detail in Chapter 3 – Simulator Hardware Development, and software development is also discussed in detail in Chapter 4 - Simulator Software Development, with the hope that these chapters could provide guidance for others looking to develop similar systems. Chapter 6 - Discussion covers some of the objectives met by this simulation and other possible purposes it could provide. A use case for education and training was presented, covering common dangerous bicyclist encounters created as VR scenarios for drivers to experience. The findings for this use case may have important implications on how to design effective driver education and training programs. Another use case presented was observing cyclist behavior, which could inform infrastructure design and urban development.

Chapter 2 Literature Review

This thesis describes the development of a VR cycling simulator that can be used in transportation safety research (Feng & Hamad, 2021). Virtual environments have been utilized in simulations due to the high-fidelity nature of this technology, and the vast possibilities in terms of research applications. Simulators have been especially important in studying transportation safety, as this field constitutes high-risk dangerous situations that are hard to recreate in real life without putting people in harm's way. Virtual reality simulators specifically have been more utilized recently due to their high-fidelity nature and ability to accurately represent reality to users. Unfortunately, VR usage comes with the possibility of causing cybersickness, but there have been efforts to examine its causes and how to mitigate it. In this chapter we explore literature on transportation safety research utilizing simulations, and discuss cybersickness as one of the major setbacks for VR usage in research.

2.1 Utilizing Simulators in Transportation Safety Research

Simulators have been a valuable tool in studying the behavior of road users in various contexts, such as collecting data in a safe environment, improving road safety for all users, and design analysis for new driving innovations. The most common simulation utilized in the context of road safety is driving simulations. In recent years, there have also been a number of studies pertaining to pedestrian and cycling behaviors as well.

“Driving simulators are one of the most sophisticated applications of computer-aided kinematic and dynamic simulations, as well as one of the biggest triumphs in their development,”

(Chang, 2013). They allow for not only driver training but also an instrumental tool in research applications. They are used in vehicle design, autonomous vehicle design, and human factors studies regarding many aspects of driving behavior, such as driving under the influence or in varying weather conditions. Another benefit of using simulators in driving applications is the ability to conduct experiments in a safe, cost-effective, and controlled environment.

Simulators in general can provide a realistic substitute for the actual experience in one or more aspects, as long as the validity of those aspects is met. One driving simulator study defines validity as “evoking the same behavior as would be shown in reality under similar circumstances”, and focused on establishing the validity of simulators as a reliable research tool (Kaptein et al., 1996). A Netherlands study utilized a driving simulator to train novice drivers and observe changes in their gaze behavior (Leeuwen et al., 2015). Results indicated short term changes in both visual behavior and driver skills after using the simulator for four 30-minute sessions, illustrating the effectiveness of simulations in driver education and training purposes. Another study utilized a driving simulation to study the behavior of drivers when overtaking bicyclists, which was used to examine the attitudes and biases of drivers towards bicyclists sharing the road, without endangering anyone in the process (Goddard et al., 2020).

There are also studies that utilize simulations in focusing on the perspective of pedestrians on the road. One such study utilized a pedestrian simulator to examine pedestrian-driver interactions at busy intersections (Deb et al., 2017), and another study examined pedestrian crossing behaviors and interactions with autonomous vehicles (Jayaraman et al., 2020).

In recent years, more cycling simulation research projects have emerged, focusing on different simulation aspects and with varying levels of fidelity according to each simulator’s

needs. Cycling simulators have been utilized in a large range of use cases, such as studying road safety, testing infrastructure design, and creating a safe way to gather accurate cyclist data. One study in Japan utilized VR in creating a cycling simulator with the purpose of studying the effectiveness of a safety education program in reducing bicyclists' traffic law violations (Suzuki et al., 2018). There was also a VR cycling simulator study focused on researching child bicycling behavior, including social interactions, traffic crossing, riding behaviors (such as stopping or speeding up), and gaze behaviors (Babu et al., 2011). A recent Australian study showed that drivers who also rode bicycles were faster at detecting cyclists on road (Beanland & Hansen, 2017), and an American study found that drivers who believed bicyclists weren't legitimate road users overtook cyclists with higher speeds (Goddard et al., 2020).

Cycling simulations have also been useful in experimenting with bicycle facility designs and street configurations in a low-cost, safe manner. One study achieves this purpose by constructing a bicycle simulator using a system of sensors to allow for realistic steering (Sun & Qing, 2018), while another study was able to utilize ready-made products for the same purpose (Schramka et al., 2017). A couple of studies in South Korea utilized a cycling simulator as a rehabilitation system, one to improve the sense of equilibrium in its users (Jeong et al., 2005), and the other to rehabilitate motor functions and balance capabilities in elderly patients (N.-G. Kim et al., 2006).

There are also research efforts dedicated to creating cycling simulators that are as realistic as possible in meeting the needs of each study. An Oregon State study used a commercially available bike computer to measure the speed of a bike and compared it to the bike in the cycling simulation to validate that the simulation is representative of real cycling (Horne et al., 2018). A Japanese study focused on developing a cycling simulator with realistic tilt controls

for data collection (Yamaguchi et al., 2018). A study in the Netherlands focused on building a mechanical cycling simulator that emulated realistic terrain conditions (Dialynas et al., 2019). The resulting prototype was a mountain bike placed on top of a bike roller that was fitted with a haptic steering device. This simulator did not provide any visual cues to the rider, as it focused more on the mechanical aspects of the simulation.

Overall, there is considerable literature on simulations dedicated to the field of transportation safety and design. With their large variety of applications, more simulations will continue to be developed and utilized in similar research applications, especially as the technology continues to evolve.

2.2 Cybersickness

There are major advantages to using virtual environment simulations and virtual reality applications when it comes to research and education regarding road safety. One of the main benefits is that users can experience dangerous situations and understand their contexts without ever needing to be in danger themselves. It is also less expensive and safer to create staged dangerous encounters using software than it is staging them in the real world. VR applications also come with interactivity that would benefit users in training a lot more than just watching a video or seeing a demonstration.

However, the benefits of VR usage come with the possible drawback of VR-induced motion sickness, or “cybersickness”. Cybersickness is a phenomenon where users will feel symptoms akin to motion sickness (i.e. nausea, dizziness, lightheadedness) as a result of using a VR device (LaViola, 2000). The most likely reason behind this occurrence is known as the “sensory conflict theory” which states that the excessive mismatch between the motion a user perceives visually and the lack of motion in their body cause a conflict the body doesn’t know

how to handle (Hettinger & Riccio, 1992; H. G. Kim et al., 2019; LaViola, 2000). This occurs due to the disparity between the visual system and vestibular system, which is the sensory system responsible for providing our brain with information about motion, head position, and spatial orientation (Jones et al., 2009). Another hypothesis for cybersickness is the “ecological hypothesis”, which states that postural instability occurs when humans are not able to perceive or react to new dynamic situations (Stanney et al., 2020).

A virtual experience doesn't always come with cybersickness, but there are several factors that may increase the likelihood of users experiencing this issue. Individual factors include a user's predisposition to motion sickness, fatigue, or nausea (LaViola, 2000; Rebenitsch & Owen, 2014), prolonged VR exposure, and how adapted a user is to VR applications (Harm et al., 2007). It has also been observed that users experience less cybersickness symptoms when sitting as opposed to standing, symptoms were worse when experiencing high-speed games or simulations, and controlling the simulation makes the user less susceptible to symptoms than being a passive participant (Hakkinen et al., 2002; LaViola, 2000; Merhi et al., 2007). Technical factors include issues like position tracking errors (better head tracking reduces symptoms), detectable lags (delays in the visual display can cause symptoms), and flicker in the visual display (Hettinger & Riccio, 1992; LaViola, 2000). Accordingly, applications using VR technology may be compromised to the extent that users experience these uncomfortable symptoms, which presents a critical issue for future development and utilization of virtual simulations (Hettinger & Riccio, 1992; Stanney et al., 2020).

To mitigate VR-induced cybersickness, a perfect system would have absolutely no lag (0 ms of latency), 8K resolution and 90-120 Frames Per Second (FPS) (Jung & Whangbo, 2017). However, this is too idealistic, so the goal is to get as close to this standard as possible with a

system's hardware. Following the sensory conflict theory, one approach is to reduce the disparity between perceived visual motion and what users physically feel. This can be achieved using a motion platform that reduces sensory conflict, but the disconnect between the physical and visual feedback could cause motion sickness as well as cybersickness if not applied correctly (Casali, 1985; LaViola, 2000). Researchers have also looked into direct vestibular stimulation, where a user could wear a device that sends electrical signals to a nerve that tricks the vestibular system into believing there is motion, reducing or even eliminating cybersickness, but studies on this subject are still in their infancy.

Another approach suggests the use of “rest frames”, where users are allowed to observe the environment in rest and determine what is stationary and what isn't (Prothero, 1998). If the background is in agreement with inertial cues, cybersickness symptoms reduce in users (LaViola, 2000; Stanney et al., 2020). Adaption is another method that may help users; by slowly adding linear or rotational acceleration, users will gradually adapt to the virtual environment and are less likely to experience symptoms (LaViola, 2000; McCauley & Sharkey, 1992). One simple solution some researchers used was asking participants to take medicine (such as Dramamine) to counteract any motion sickness symptoms that may occur (Jung & Whangbo, 2017).

One common solution in mainstream VR applications is the implementation of teleportation as a movement mechanism to reduce or even eliminate cybersickness. Continuous motion is especially problematic for stationary user configurations, so this approach has been widespread to increase usability (Habgood et al., 2018). However, teleportation can also disorient users and reduce their sense of presence in the virtual environment, so this method should be used sparingly or designed in a way in which users are more aware of their surroundings.

Ultimately, it is vital that these factors and design considerations are taken into account when developing a VR simulation. It is necessary to reduce the possibility of cybersickness as much as possible so that users and researchers can benefit from the use of such simulators without experiencing adverse effects.

Chapter 3 Simulator Hardware Development

The development of a realistic VR cycling simulation depends on having the appropriate hardware and software components working together seamlessly. One of the goals of this thesis was developing a simulation in a manner that was easily duplicated, as we wanted researchers to be able to build similar simulations with attainable components. Accordingly, we opted for equipment readily available on the market rather than customizing our own.

We used a regular bicycle (Trek® FX 4) that was mounted on a wheel-on bicycle trainer (Saris® M2) and a motion platform (Saris® Nfinity) for the cycling simulation. The VR headset used for our simulation was the HTC Vive Pro Eye (HTC, 2021b) and we used a PC that had appropriate specifications for VR use. A smart fan (Wahoo® KICKR) was also used to simulate air flow based on the speed of the user's pedaling.

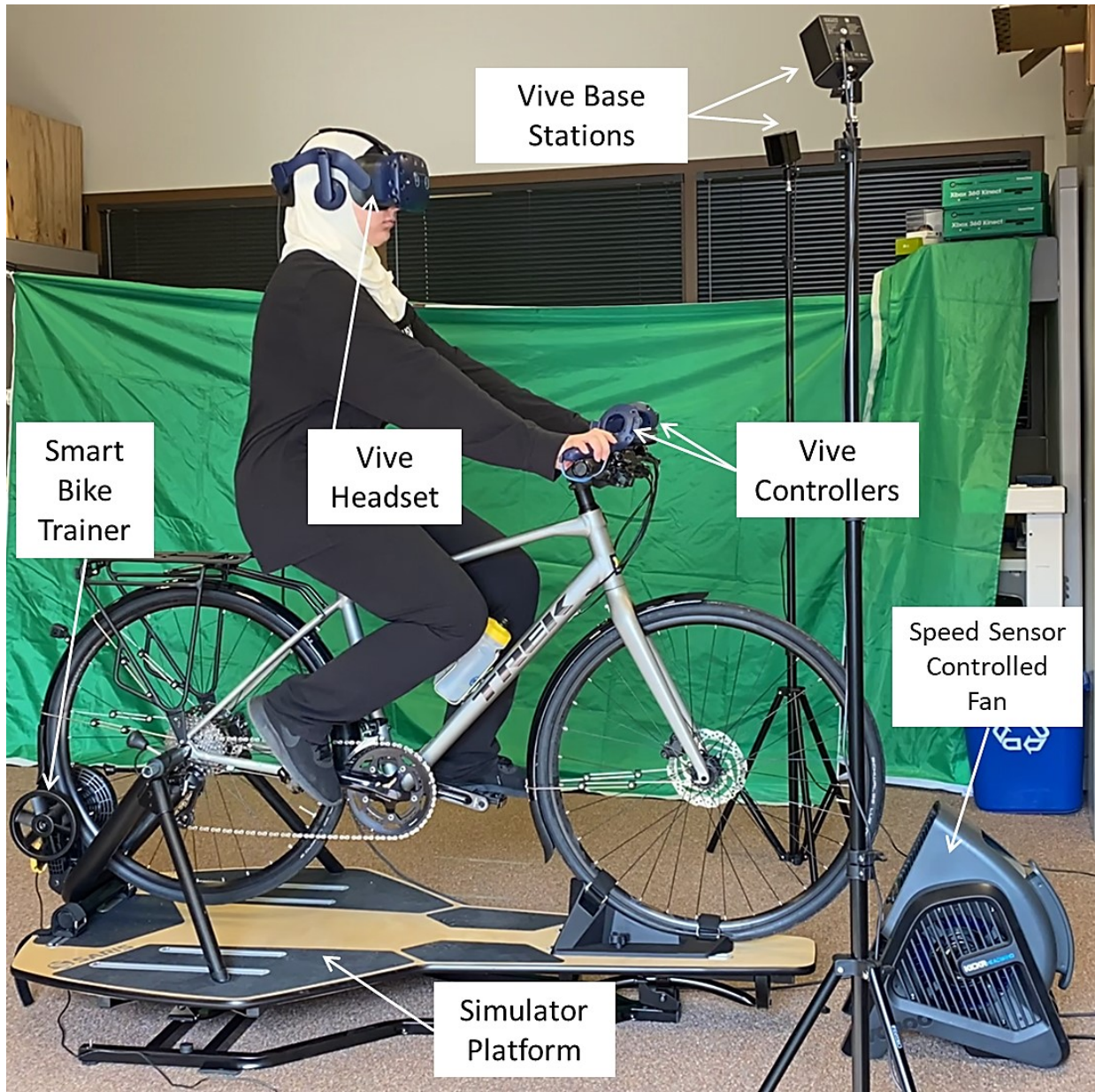


Figure 3-1: Virtual Reality (VR) Cycling Simulator Setup.

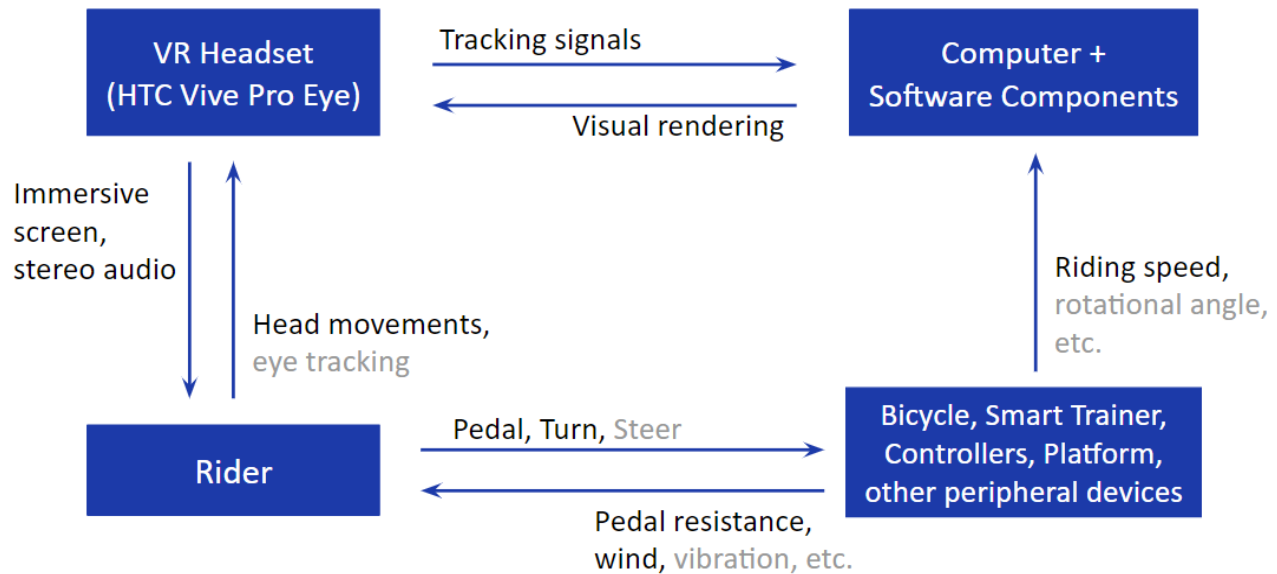


Figure 3-2: System Components (note: features in gray are not currently implemented).

3.1 VR Headset System

There is a number of various VR headsets and technologies currently available on the market for personal and professional use, such as the Oculus Series, PlayStation VR, Valve Index, and the HTC Vive. After carefully reviewing several VR Headsets, the HTC Vive Pro Eye was selected as the most appropriate hardware for this simulation project. This decision was largely attributed to the Vive’s high-quality graphic displays, high-resolution headphones with 3D spatial sound, and large room-scale tracking that could accommodate multiple headsets should the need arise. The headset has a dual screen display (one screen per eye) with a combined resolution of 2880x1600 and a field of view of 110 degrees.

The specifications on the Vive are the best currently available for providing an immersive experience to the users. The realistic surrounding ambience add realism to the experience, while having a large tracking area accommodates a variety of stationary bike setups. The wide-area tracking is a feature in the Vive base station, which has wireless syncing capabilities and can track an area up to 1000 square feet (93 square meter). The presence of these small base stations is essential for tracking the exact locations of both the headset and the controllers (See figure 3-

1), making the experience as seamless and realistic as possible. The Vive comes with two controllers, which were used to implement steering capabilities for this simulation. Turning is achieved by clicking the trigger buttons on the right and left controller, both of which are mounted onto the handlebars of the bicycle. The headset tracks head movement in all directions, which allows users to look around and explore the virtual environment while experiencing the cycling simulation. The eye tracking capability is also potentially useful for research purposes in measuring a user's visual attention and visual scan patterns (e.g., at an intersection).

3.2 Bicycle & Stationary Bike Trainer

For the stationary bicycle setup, we opted for a regular adult-sized bicycle that was mounted onto a wheel-on smart bike trainer. This setup allows users to pedal the bicycle while the bike trainer provides realistic pedaling resistance, giving users an experience similar to cycling in the real world. The smart bike trainer also broadcasts the instantaneous riding speed, pedaling cadence, and pedaling power (force), all of which can be read and used in the VR software via Bluetooth or ANT+ connection (Garmin Canada Inc., 2012). Both the bicycle and the bike trainer are mounted onto a Saris® Nfinity platform that provides side-to-side and fore-aft movements when riding the bicycle, further increasing the realism of the simulated experience.

3.3 Miscellaneous

A VR-ready, high-performance desktop PC was used to create and run the cycling simulator. The key specifications of this PC include an Intel® Core i9 3.6GHz CPU, a NVIDIA® GeForce RTX 2080 Ti graphics card, 64GB DDR4 RAM, and a 1.5 TB SSD hard drive. The PC was not equipped with a built-in Bluetooth or ANT+ receiver, so separate USB adapters were purchased to fulfill these connections. A speed-sensor controlled fan was also used to simulate varying levels of headwind based on the riding speed.

Chapter 4 Simulator Software Development

VR technology offers a great method of building and simulating 3D environments in an accessible and cost-efficient way. One of the most attainable options for using VR in building simulations is the utilization of game engines, such as Unity[®], Unreal Engine 4, or CryEngine, each with its own unique features and native scripting languages. Selecting the appropriate software to build a VR Simulation depends on the type of experience one wants users to have and the goals the simulation aims to achieve. For this VR cycling simulator, Unity[®] was selected as the development environment. This chapter covers in detail the process in which the Unity[®] engine environment was used to develop our simulation, and how different simulation purposes can be achieved in one application.

4.1 VR Simulation Software - Unity[®]

The first step to creating the virtual reality simulation was deciding on a suitable program to work with. Unity[®] was selected as the development environment due its wide array of applications. Using a real-time development platform like Unity[®] provides larger capabilities than those provided by traditional software development methods, and is especially accessible for developers that have less coding experience. Some of the advantages of using Unity[®] include developing simulations visually in a 3D-world view (shown in figure 4-1 below), compatibility with various 3D model formats, a physics engine that provides realistic interactions, and the ability to create and play animations within the 3D simulation environment (Unity Technologies, 2021a). Unity[®] also supports development in VR and the ability to continuously test the

simulation prior to building the final executable program. Accordingly, Unity® was the most appropriate development method for the goals of this project.

By using Unity®, it was possible to develop an immersive VR simulation with a variety of built-in applications, including both passive and active simulation experiences on driver-bicyclist interactions. C# was used as the primary programming language (as it is Unity's native scripting language), with several native code libraries (plug-ins) added and used for developing specific functionalities. Unity® 2020 depreciated built-in XR (umbrella term for virtual, mixed, and augmented reality) platform integration, so we decided to work with Unity® 2019.4 LTS (long term support), as it is currently the most up-to-date version of Unity® that supports Vive-specific VR development.

The primary plug-in used for developing VR elements was the SteamVR plugin from Valve, which provides built-in VR functionality for the Unity® engine (Valve, 2020). Some of these functions include the ability to look around a virtual environment just by moving the headset and built-in controller button mapping.

4.1.1 Unity® Editor & Scene Creation

A scene in Unity® is where a user would work with content, and each unique scene can be considered its own level (Unity Technologies, 2017). Essentially, a scene represents any background scenery, static and moving objects, and the surfaces on which these objects exist. Within Unity®, multiple scenes were created to fulfill the needs of the simulation. Different scenes can use the same objects, but have different functionalities and controls, depending on the purpose that scene is meant to serve. For example, one scene could implement character movement using a keyboard, while another scene could have the character move with a joystick. Another useful feature that can be used with scenes is the Prefab system (Unity Technologies,

2018). Prefabs are game objects that can be saved with specific values and configurations. The benefit of prefabs is the ability to reuse these items multiple times in a scene or across multiple scenes in a project, and being able to easily make broad changes to these game objects without having to edit each instance.

For the purposes of this simulation, each scene was set in an urban environment, which is common for bicycle traffic and any bicyclist-driver interactions. The Unity® asset Windridge City (Unity Technologies, 2019) was used to provide a realistic urban environment, in conjunction with 3D vehicle models used to populate the roads with both background vehicles and vehicles used in pre-programmed interactions.

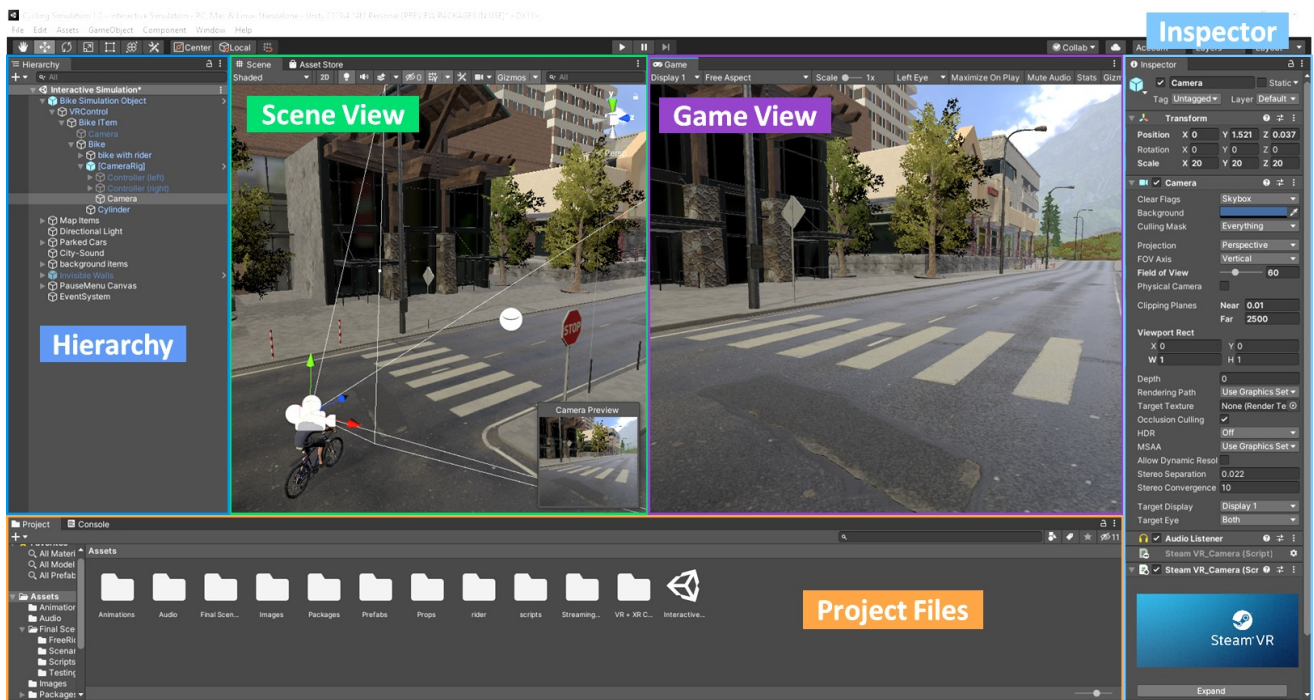


Figure 4-1: The Unity® user interface, including scene and game views, hierarchy, inspector, and project windows.

Figure 4-1 shows an example of the Unity® editor being used to develop a scene. On the left of the editor is a hierarchy list showing all the contents present in the active scene. The middle window contains the scene view and the game view. The scene view allows one to

visually navigate the scene and make any changes directly to the active object (such as moving it or changing its size), while the game view provides a testing screen that shows what the final simulation will look like. Because this VR simulation is a first-person experience, the game view will display the scene on screen from a first-person perspective, providing an effective tool for testing even without using a VR headset. The far right of the editor contains the inspector, which allows users to view and edit the properties of the selected object from the scene or hierarchy. In the bottom is the project window, which shows all the assets available to use in the active Unity® project, such as scenes, scripts, 3D models, downloaded assets, and more.

4.1.2 Environment & Scene Objects

As mentioned previously, the urban environment for the simulation is provided by The Windridge City asset. This asset comes preloaded with common static models, including buildings, roads, sidewalks, signs, street markings, plants, and more. This was a useful base to build the simulation on, as it allowed my main efforts to be focused on the development of the simulation itself rather than building a virtual environment from scratch. Modifications were made to accommodate any specific circumstances needed for some of the scenes, such as adding a new parking lane, modifying and/or adding street markings and traffic signs, and moving some objects like street poles, trees, or benches as needed. Other necessary 3D model objects such as a bicycle and various vehicles were then downloaded and added to the relevant scenes. Alongside the existing Windridge City environment objects, parked vehicles and moving vehicles were added sparingly in some scenes to make the environment look more populated and realistic.

Another key aspect of every scene is the inclusion of a user-defined game object (VR Cyclist object) that contains the bicycle model, a rigged 3D model of a cyclist, and most importantly the Camera Rig prefab object from the SteamVR asset pack. The first object is a

solid 3D bicycle model (Filippov, 2017), but other models could be used in the future if animation of wheels turning, steering with the handlebars, or pedaling is desired. The second object is a 3D model of a person (Taha, 2020) that was rigged so it looks like they are riding the bicycle. A rigged 3D model is one that has a skeleton that can be moved and deformed into different shapes, which is especially useful when using characters in specific poses or positions. These two 3D models were added to give users context to where they are relative to their environment, as they can see the bike they are riding and a representation of a virtual body where theirs would be. Custom 3D models can be created using 3D modeling software, but for the sake of simplicity we utilized free-to-use premade models. The last part of the VR cyclist object is the Camera Rig. The presence of the Camera Rig removes the need for the built-in standard The Unity® virtual camera, allowing the user to experience the simulation from a first-person perspective. It consists of a cube-shaped play area, two controller items representing the right and left VR controllers, and a virtual camera that moves based on the VR headset's tracking.

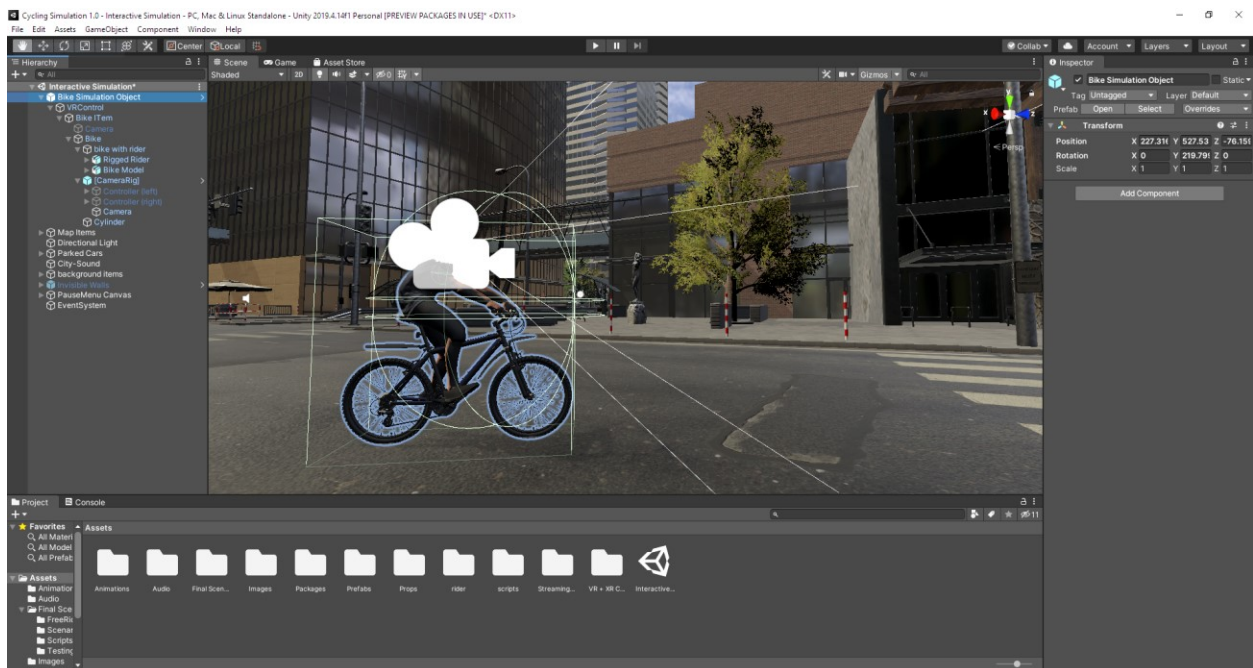


Figure 4-2: Editor showing the VR Cyclist Object in detail.

4.2 Phase 1 – Interactive Simulation

The first phase of the simulation development focused on a high-fidelity VR cycling experience fully controlled by the user. One motivation for this aspect of the simulation was aiming to reduce motion sickness in users by adding physical interactivity via pedaling. Another purpose an interactive simulation fulfills is the ability to study the cycling behavior of users as they will be in control. To start, a scene was created using an urban setting, and background moving vehicles were added to increase the realism of the simulation and present some form of an obstacle on the road for users to navigate around. The VR Cyclist object was also added, but was changed as needed.

The first functionality addressed was reading a speed value from the bike trainer to Unity[®]. This is a necessary step in order to have the in-simulation bike speed match that of the user's pedaling. The bike trainer supports both Bluetooth and ANT+ protocol, but we opted for using ANT+ since it is capable of connecting seamlessly to multiple devices at once. To connect to Unity[®], we used the Advanced ANT+ Asset (2TH prod, 2020) available for purchase on the Unity[®] Store. The Asset comes with several scripts that allow for the use of different fitness devices in Unity[®], which serves the exact purpose needed from the trainer. An ANT+ fitness display script that reads data from the bike trainer was then connected to the VR cyclist object in order to be able to reference the speed value needed to control the in-game bicyclist.

The next step was adding controls to the bicycle, which started with adding a “Character Controller” component (Unity Technologies, 2021b) to the VR Cyclist object. This adds a collider to the bicycle object, as well as a class of movement functions that can be customized for our purposes. A C++ script called “VRController” was then created to add movement functionalities (script can be found in Appendix). The script is able to reference the speed value

found from the fitness display script and use it to control movement speed. The movement is set to always be active and forward-facing, with the movement speed equal to the speed referenced from the ANT+ script tracking the user's pedaling.

Steering was implemented using a snap rotation function that rotates the virtual bicycle a certain number of degrees left or right (i.e., yaw angle), depending on which (i.e., left or right) controller is used. The reason why steering was implemented as a snap rotation is because smooth rotation and continuous motion in VR can cause a lot of motion sickness for users, especially with their physical bodies not rotating with what they experience visually (as mentioned in section 2.2 regarding the sensory conflict theory). The left and right Vive controllers are mounted on the bicycle handlebars facing down so that the user can easily use the trigger buttons to rotate in the desired direction. The current rotation angle is set at 5 degrees, but can be changed to any other value if necessary. By rotating the entire VR Cyclist object, the user (represented by the Camera Rig) will rotate with the bicycle and the forward-facing direction will change according to the current rotation angle.

Another element of this phase is adding constraints on the map. Due to the colliders added to the environment elements and the character controller, the VR Cyclist object (and thus the user) isn't able to fall through the road or go through buildings thanks to Unity[®]'s physics engine. However, the user is still able to ride onto sidewalks and other areas off-road, which could be an issue if the simulation is focusing on studying bicyclist behavior on the road only. This can be rectified with the use of "invisible walls" around restricted areas, which is implemented by adding more colliders to the map and making them invisible by turning off mesh renderers on the collider objects. A toggle option on the menu allows for these invisible walls to be turned on or off as needed for the purposes of the simulation.

4.3 Phase 2 – Dangerous Driver-Bicyclist Encounter Scenarios Simulation

The second phase of the simulation development focused mostly on developing specific scenarios that represent common dangerous encounters experienced by bicyclists, and was set in the same urban setting as the interactive part of the simulation, but without the interactive user controls. These scenarios are based on findings from a previous naturalistic study (Feng et al., 2018). Below is a brief description of each of these scenarios (also shown in Figure 4-3):

1. Close pass by a sedan: the bicyclist is overtaken by a sedan with close distance and high speed
2. Close pass by a van: similar to scenario 1, but the bicyclist is overtaken by a larger vehicle. For scenarios 1 and 2, It is easy to change the type of overtaking vehicle (e.g., a bus or RV) in Unity® to generate a variety of close pass scenarios.
3. Dooring: the bicyclist is riding next to a parking lane, when one of the parked vehicle's door suddenly opens (typically due to the vehicle occupant failing to check any approaching traffic before opening the door).
4. Right hook: the vehicle is making a right turn at an intersection and cutting off the bicyclist who is on the right-hand side traveling in the same direction.
5. Fail to yield I: the bicyclist is crossing the road on a cross walk at an unsignalized intersection, and the driver is making a left turn and failing to yield to the bicyclist who has right-of-way.
6. Fail to yield II: the bicyclist is going straight at an unsignalized intersection, and the driver of an oncoming vehicle is making a left turn and failing to yield to the bicyclist. For both scenarios 5 and 6, the vehicle is programmed to brake hard at the last second to avoid collision with the virtual bicycle.



(a) Close pass by a sedan



(b) Close pass by a van



(c) Dooring



(d) Right hook



(e) Fail to yield I



(f) Fail to yield II

Figure 4-3: Examples of the dangerous encounter scenarios that one can experience in the cycling simulator.

The goal of this part of the simulation is to educate users on road safety and some of the common dangerous encounters bicyclists may experience on the road. This is especially important in the US, where there is a much lower cycling population, and drivers overestimate how much space or time they give to cyclists sharing the road due to their lack of experience.

To guarantee a uniform experience for all users in each of these scenarios, this part of the simulation was designed as a passive experience. Users are able to wear the headset and experience the simulation from a first-person perspective, but they cannot control the speed nor direction of the bicycle. This is implemented by using an asset called the Bezier Path Creator (Lague, 2019), in which complex paths can be defined and edited in a graphical user interface. C# path follower scripts are then added to the relevant objects, whether it is the VR Cyclist object, scenario vehicles, or background vehicles, allowing them to follow their designated paths with configurable settings such as moving speed and end of path instruction (e.g., looping, stopping).

For each of the dangerous encounter scenarios described above, an appropriate vehicle model was added to the scene and programmed to follow a predefined path and speed profile. The behavior of the vehicle can also be configured based on the desired interactions (e.g., behavior change triggered by the location). For instance, in the “dooring” scenario (see Figure 4-3(c) above) one of the parked vehicle’s doors suddenly opens when the bicyclist is approaching. In this scenario, a vehicle model with separate parts was used, and the vehicle door was programmed to open using animations. In order to play the animation at an appropriate time, a trigger is implemented. A trigger object in Unity[®] is related to the “collider” objects from the built-in physics engine. When an object with a collider property touches another object with a trigger property, the trigger becomes capable of executing script. Using these properties, a collider object was added to the bicyclist, and an invisible trigger cube was added in the bicyclist’s path. Once the bicyclist reaches the cube, a script activates the vehicle door animator, and the door animation starts to play.

4.3.1 360° Videos

The main goal of developing common dangerous encounters into VR scenarios is to educate drivers on the perspective of bicyclists on the road. With this purpose in mind, we thought it useful to not only have these scenarios be experienced in the full-fledged VR simulator at our lab, but to also have them available as 360° videos that anyone can experience from their smart devices or computers anywhere in the world.

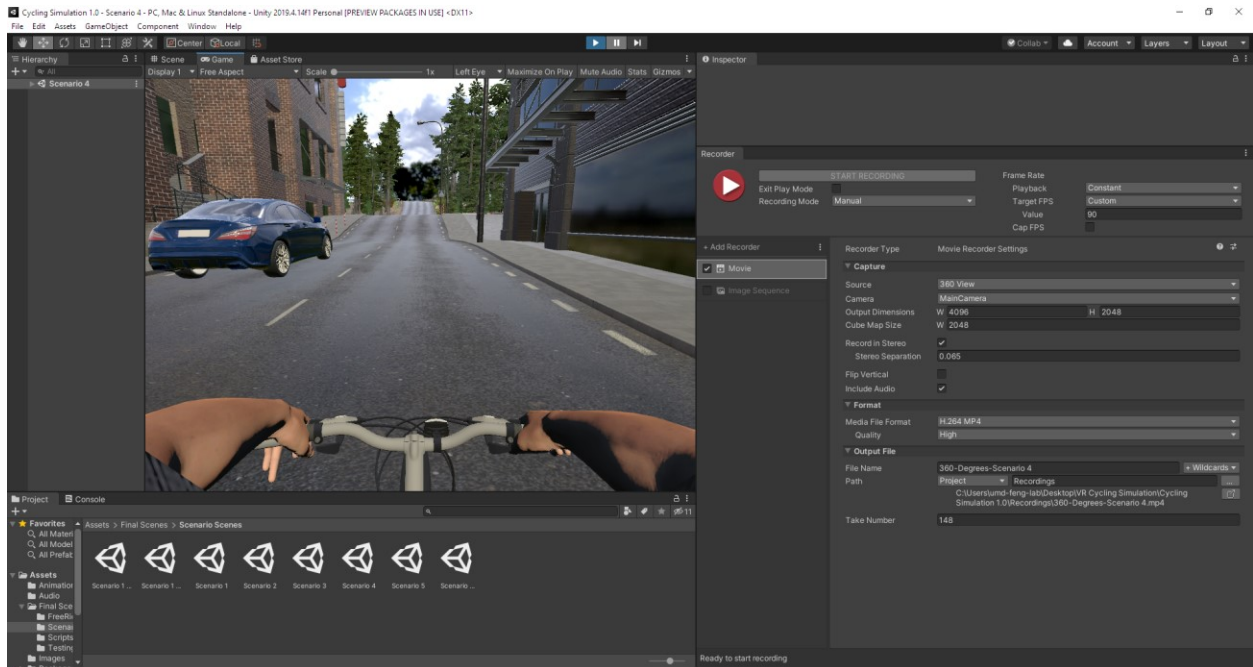


Figure 4-4: Active Recorder window ready to record in the Unity® editor.

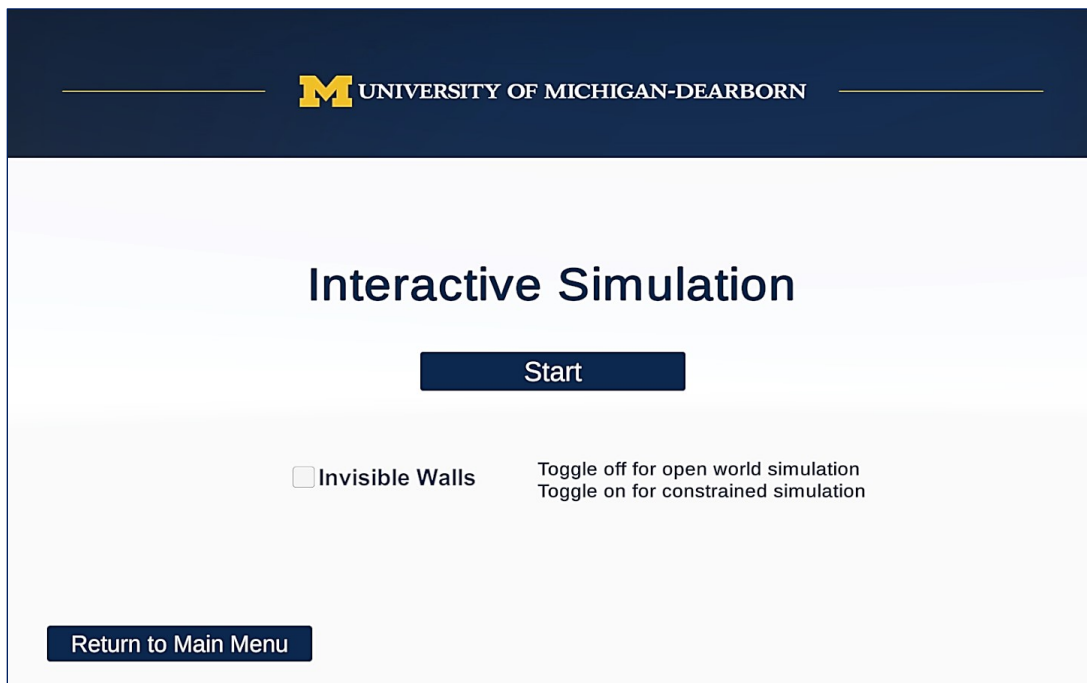
These videos were recorded from within Unity® using a built-in video recorder. Settings can be adjusted to different qualities, can include audio, and the recorded view can be set to the perspective of the in-scene camera or a complete 360° view of every frame. Figure 4-4 shows the Unity® editor window with the Recorder window active and set to record a 360-degree capture of the simulation.

4.4 Usability Improvement – Adding Simulation Menus

Menus were added in order to streamline and organize the different aspects of the simulation, which is especially important when building the simulation as a Unity® executable.



(a) Main Menu



(b) Interactive Simulation Menu

Scenarios (Passive Simulation)

- Scenario 1a Close Pass - Sedan
- Scenario 1b Close Pass - Van
- Scenario 2 Dooring
- Scenario 3 Right Hook
- Scenario 4 Fail to Yield I - Crosswalk
- Scenario 5 Fail to Yield II - Intersection

Return to Main Menu

(c) Scenario Menu



(d) Pause Menu

Figure 4-5: Simulation Menu System.

The main menu (Figure 4-5 (a)) allows for selection between the passive scenario experience and the user-controlled interactive experience. When selecting the interactive

simulation option from the main menu, the interactive simulation menu (Figure 4-5 (b)) will appear, allowing you to toggle between having the invisible walls on or off before entering the simulation. When selecting the scenario option, the scenario menu (Figure 4-5 (c)) appears with a list of the available scenarios and a short description of each one. Clicking on any of these scenario buttons will take you to the corresponding scenario, and once it is complete the scenario menu will automatically pop back up. During any part of any simulation, users can pause, which will pull up a pause menu screen (Figure 4-5 (d)) containing the options to resume, return to the main menu, or quit the simulation.

Chapter 5 Simulator Calibration & Validation

Calibration and validation of computer simulations is necessary to ensure that the final product serves as an accurate and realistic model of what the simulation represents. This VR cycling simulator presents an approximate reproduction of riding a bicycle in real life, so it is important to calibrate it, so it performs as designed, and validate that it does in fact imitate reality enough that it is useful for research and education purposes.

Some of the main factors of the cycling simulator to validate include whether or not the environment is scaled properly and looks realistic enough, if the users can control the simulation the way they would expect a bike to move in real life, and if there is any VR-caused cybersickness that may inhibit the users or simulation in any way. Subjective validation is necessary to ensure that the simulation acts the way users expect, but unfortunately due to the Covid-19 pandemic, subjective validation was limited to myself and my repeated testing of the simulator during and post development. The confirmation of the simulator's validity is threatened by the lack of data due to Covid-19 restrictions, but once circumstances improve and it is safe to do so, more testing will be conducted. Other participants would be included in the validation process and offer feedback on how to improve the validity and realism of the cycling simulator.

5.1 Setup and Calibration

The HTC Vive requires the use of base stations in order to track the headset and controllers. In order to test and run the simulation using the headset, two HTC Vive VR base

stations were positioned on either side of the stationary bike using tripods to cover a large “play area” (as is seen in figure 3-1 bike setup). The play area is defined as the virtual boundaries of the Vive, in which users are able to experience the VR simulation (HTC, 2021a). The Vive headset is then calibrated with the center of the play area set to where the bicycle seat is located, allowing a user to experience the simulation and have their height be translated accordingly for a realistic experience.

Calibration of the Vive is performed using SteamVR Room Setup (Valve, 2021) which allows for a quick calibration of the play area in which the simulation will occur. During development, the room setup was useful in figuring out where to place the Camera Rig in reference to the 3D bicycle model. Testing can easily be performed by using the game view option in the Unity[®] editor, which provides a preview of what the simulation will look like through the headset. In order to limit any translational movement that would affect the experience, the size of the Camera Rig’s play area was reduced to only allow rotational head movement. This is an especially important step to ensure that any users experiencing the simulation will have a similar view of their surroundings in the virtual environment. It is also possible to leave the play area’s size as the default and continuously calibrate the middle, but it would take several test runs and some trial and error to reach a point where the camera within the Camera Rig is always appropriately placed where the bicyclist’s head would be.

5.2 Scenario Validation

After completing each of the scenario scenes, we would continuously playtest them from within the Unity[®] game window both with and without the Vive headset. This helped assure that each of the bicyclist-driver interactions was running smoothly and looked realistic and accurate

compared to the use cases they are based on. It also helped with determining what issues needed to be fixed before any other users experienced the simulation.

One major issue that we experienced was motion sickness even after a short playtest, which we realized was caused by the abrupt turning in some parts of the bicycle paths. This problem was eliminated by changing the paths so that the user doesn't need to turn in order to experience each scenario.

5.3 Testing User Controls

When developing the user-controlled simulation, the biggest obstacle was getting the user's pedaling to control the speed and adding the steering functions. For the sake of simplicity, a testing scene was created, containing a simple environment made out of a plane and walls. In this scene, different scripts were tested out on the VR Cyclist object for each of the functions that needed to be added.

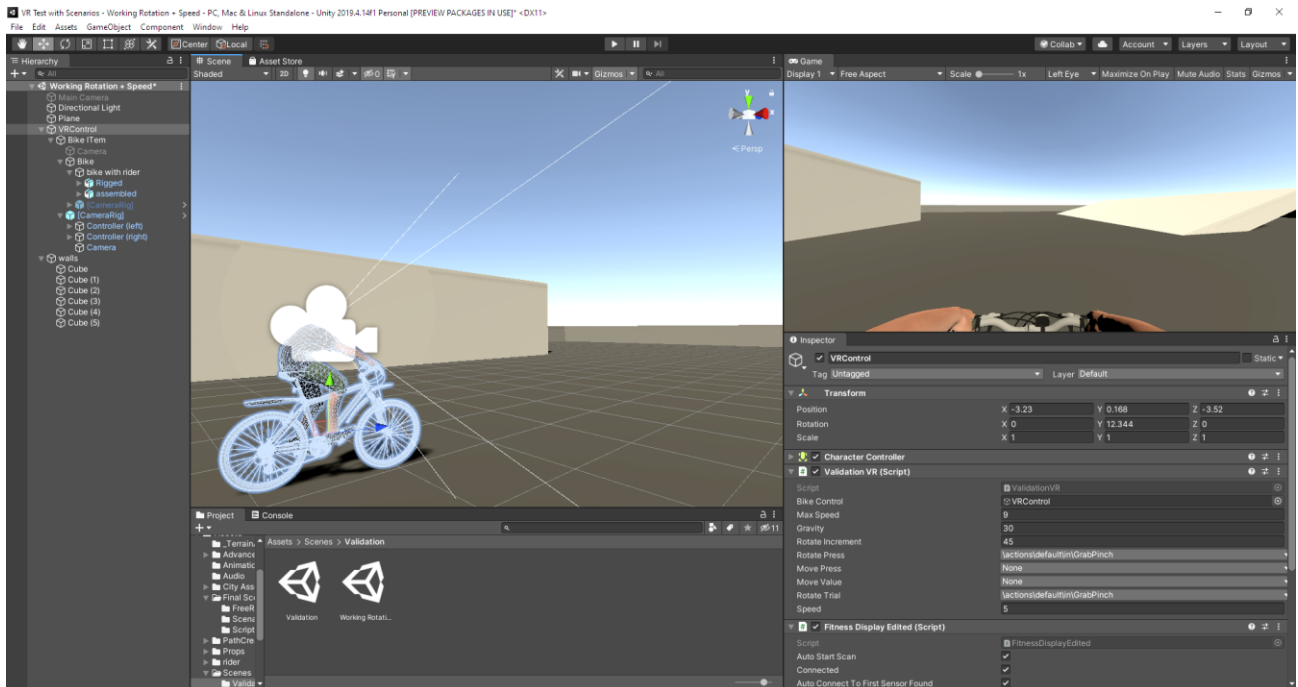


Figure 5-1: Testing simulation controls (pedaling and controller input) in a Unity® scene with a simple environment.

The initial focus was on testing the ANT+ script and making sure the speed value from the bike trainer was accurately displayed in the Unity® editor. Another script was created to set the movement at a constant speed so that the steering can be tested separately from the speed. Once the steering was properly implemented, changes to the script were made so that it referenced the speed value from the bike trainer. When trials in the testing scene eventually resulted in the VR Cyclist object working as needed, the same configuration was then used in the user-controlled interactive simulation.

5.4 Speed Validation

Another aspect that needed validation was the riding speed received from the bike trainer. In order to measure whether the speed value is accurately scaled to the environment, the testing scene was used again. One unit in Unity® is equivalent to one meter in reality (Unity Technologies, 2021c), so this was a good starting point. A cube was added to the scene with one of the dimensions = 100 units, followed by the addition of the VR Cyclist object next to it, so that the bike's movement would be parallel to the 100-meter distance. A validation script was then created based on the VRController script, but deactivated the steering functionalities and set the speed to a constant value. A timer script was also created and added to the scene. This script simply sets a timer that starts at the beginning of the simulation or playtest, and returns a value in seconds when the simulation ends. For testing purposes, the speed was defined at a constant 5 meters/second, and the playtest was allowed to run until the bicycle collided with the wall to get an accurate timing. The test ended with the timer at 20.21 seconds, with the small error probably due to the sizing of the colliders around the bicycle. With this in mind, we were able to validate that the speed in the script was scaled properly to the environment.

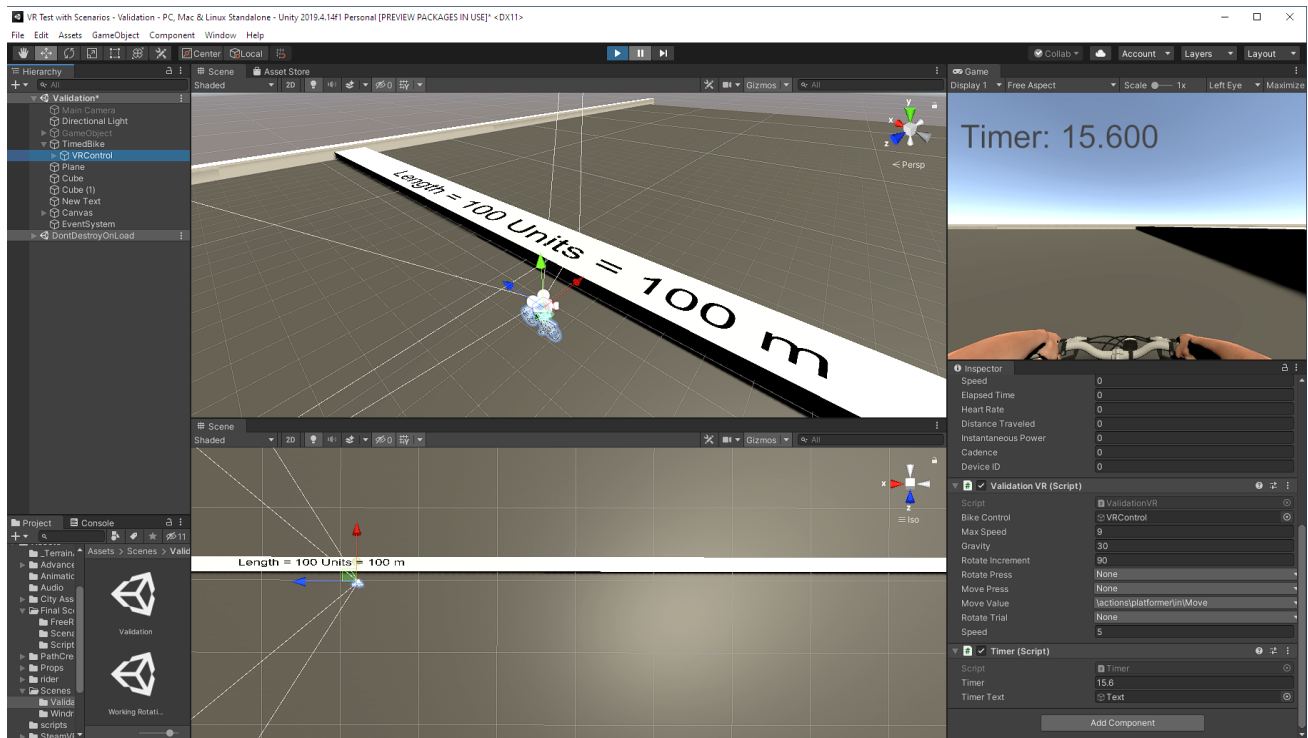


Figure 5-2: Speed validation scene showing timer with set distance.

When switching back to the VRController script that referenced the bike trainer's speed value, the simulation was tested again to see if the speed value was scaled to the environment correctly. After multiple rounds of testing, we determined that the speed value was in kilometers per hour but was being used in a context designed for meters per second, so the speed was much faster than it should be. Accordingly, the speed value being accessed in the VRController script was divided by 3.6 to convert from km/h to m/s, so the current speed read from the bike trainer is also in meters per second.

5.5 User Validation

Unfortunately, due to the Covid-19 Pandemic, validation of the simulator was limited to myself, as others were not allowed to access the simulator in the lab. As a result, the simulator's validation is threatened by the lack of user data. Under better circumstances, the validation

process would include having participants test different aspects of the simulator, including but not limited to:

- Subjective validation of the interactive simulation and each of the simulator scenarios.
- Experimenting with different user control schemes to determine which is the most comfortable and intuitive to users.
- Mitigating cybersickness, in which user testing can be performed with timed trials, questionnaires, utilizing physiological sensors, adding and/or removing peripheral hardware devices, prolonged exposure, and other appropriate methods as needed.

Once it is safe to do so, we look forward to inviting test subjects to the lab to continue validation efforts and build on the existing simulation framework.

Chapter 6 Discussion

VR cycling simulators have the potential to become valuable research and education tools for improving bicycle safety. This thesis presents the hardware and software development of an immersive, high fidelity VR cycling simulator, where users can ride a bicycle in a simulated virtual environment. All hardware elements of this project are commercially available, and the development for both hardware and software were described in detail for others who may want to build similar systems. The developed simulator has several use cases, and allows for future additions and improvements to be implemented as needed.

6.1 Use Cases

There are plenty of use cases for the VR cycling simulator, some of which have already been developed. One use case presented was using the simulator for driver education and training. Dangerous encounters that bicyclists commonly face were created as VR scenarios that drivers can experience via the simulator. One hypothesis is that drivers may develop awareness, knowledge, and empathy towards bicyclists after experiencing these scenarios from a bicyclist's perspective. For example, drivers sometimes think they are giving bicyclists enough distance when they're not, and experiencing that situation themselves might help them understand how it feels from the cyclist perspective. This awareness and empathy may further lead drivers to behave differently when interacting with bicyclists on the road. If this hypothesis is true, it may have important implications on how to design effective educational and training programs for

drivers. Further research is needed to determine how this experience would affect a driver's behavior.

To reach a broader audience and provide educational resources to communities, organizations and the general public, we have also created stereoscopic 360° videos of pre-recorded virtual scenarios using the Unity® Frame Recorder. These videos are shared online, where people can experience them anywhere by playing them on their personal computers or smart devices without needing a dedicated VR headset or a full-fledged cycling simulator. The 360° videos of the programmed scenarios are available at www.umich.edu/~fredfeng/cyclingsim.

Other use cases can be applied with some changes made to the current simulator setup. One use case is to study cyclist visual scanning behavior (e.g., at intersections) while using the simulator. The HTC Vive Pro Eye has eye tracking capabilities that can be used to scan eye movements, and study where participants look, if there are any scanning patterns, and even comparing novice to expert cyclists. Another use case is to study cyclists' preferences of different infrastructure designs. The virtual environment can be modified so participants can experience different designs within the same environment in VR. This method of testing infrastructure is much cheaper and easier to evaluate in VR than it would be in real life. participants can provide feedback on their preferences regarding different bikeway types (e.g., protected bike lanes, marked bike lanes), the presence of physical barriers between bike paths and the road, different usages of bike icons on shared roads, and other bike facilities that could be implemented. The simulator could also be used to test the effectiveness of on-bike safety technologies, such as the Garmin® Varia radar system which warns bicyclists about vehicles approaching from behind. We plan to explore some of these use cases in the future.

6.2 Future Improvements

There are a number of improvements and additions that can be made to the current simulator setup to make it more representative of real life. One improvement would be adding realistic steering capabilities to the simulator. As of now, one of the main limitations of the simulator is that turning is controlled via button presses. The ability to move the handlebars to steer would increase the fidelity of the simulator and make the VR experience more realistic. Another improvement is to improve the visual aspects of the simulator in order to mitigate cybersickness symptoms. Decreasing the latency, increasing the rendering speed, and improving the virtual environment are some of the aspects that could be improved on in the near future to make users less susceptible to cybersickness. Testing is needed to pinpoint any issues the current simulator may have in this regard.

Another way to increase realism is by adding a variety of other road users to the virtual environment. This would include pedestrians, other cyclists, motorcyclists, more cars, etc. It may also be worth exploring multi-agent simulator development by combining the VR cycling simulator with a driving simulator or a VR pedestrian simulator (such as the one used in (Jayaraman et al., 2020)). In the combined simulator, a subset of virtual vehicles, bicycles and pedestrians can be controlled by multiple human subjects. This would enable researchers to study more complex interactions between all these road users in the same virtual environment.

The simulator allows for open-world exploration, in which users can ride on the roads, sidewalks, and bump into map items such as buildings or trees. Invisible walls can be added as needed to restrict movement, such as only allowing cyclists on the roads, or limiting the area in which participants can travel.

There is currently no haptic feedback implemented in the simulator, but this can be added by utilizing some current sports technologies. Pedaling resistance can be simulated by controlling the smart bike trainer via Unity[®], for example going uphill would cause higher pedaling resistance. Simulating the road gradient can be achieved by physically adjusting the bicycle's front-end position (such as with the Wahoo[®] Kickr Climb). The simulator can also adjust according to pavement smoothness (e.g., asphalt vs cobblestone) using a motor inside the smart bike trainer that can generate micro vibrations in different intensities (Tacx[®] Neo). It would also be possible to simulate lateral wind using a controllable fan or low-budget wind tunnel, which would be useful for simulating lateral wind force when close-passed by a vehicle for example.

Appendix

VRController C# Script:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Valve.VR;

public class VRController : MonoBehaviour
{
    public GameObject bikeControl; //this is the VR Cyclist game object
    public float maxSpeed = 9f; //cap speed at 9 m/s

    public float gravityValue = 30.0f;
    //gravity value for Unity's physics engine. Number is based on trial and error

    public float rotateIncrement = 5f; //rotation angle = 5 degrees every press, can be
    changed if needed

    public SteamVR_Action_Boolean rotatePress= null; //initializes rotation action in
    SteamVR action set

    public float speedValue; // speed value to be assigned = to trainer speed

    private CharacterController charControl = null;
    private Transform camRig= null; // tracks location and movement of the camera rig
    private Transform headVR = null; // tracks location and movement of VR headset

    private void Awake()
    {
        charControl = GetComponent<CharacterController>();
    }

    // Start is called before the first frame update
    void Start()
    {
        camRig= SteamVR_Render.Top().origin; //assigns camRig variable to the origin
        point of the Camera Rig
        headVR = SteamVR_Render.Top().head; //assigns headVR variable to the top of the
        tracked headset
    }

    // Update is called once per frame, smoother than using void Update()
    void FixedUpdate()
    {
        CalculateMovement(); //calls Calculate Movement function
        Rotation();//calls Calculate Movement function
    }
}
```

```

// function for lateral movement, controlled by user's pedaling
void CalculateMovement()
{
    //this section tracks current direction in which movement will occur
    Vector3 orientationEuler = new Vector3(0, camRig.eulerAngles.y, 0);
    Quaternion orientation = Quaternion.Euler(orientationEuler);
    Vector3 movement = Vector3.zero;

    //user's speed equals the speed value from the smart trainer, divided by 3.6 to
    convert km/h to m/s
    speedValue = bikeControl.GetComponent<FitnessDisplayEdited>().speed/3.6f;

    //movement in the current orientation at the calculated speed value
    movement += orientation * (speedValue * Vector3.forward);

    //add gravity to Cycling object
    movement.y += -gravityValue * Time.deltaTime;

    //character controller movement
    charControl.Move(movement * Time.deltaTime);
}

// rotational movement function
private void Rotation()
{
    float snapValue = 0.0f; // start at rotation = 0

    //turn left
    if(rotatePress.GetStateDown(SteamVR_Input_Sources.LeftHand))
    {
        snapValue = -Mathf.Abs(rotateIncrement);
    }

    //turn right
    if (rotatePress.GetStateDown(SteamVR_Input_Sources.RightHand))
    {
        snapValue = Mathf.Abs(rotateIncrement);
    }

    //turn according to snap value from above if statements
    transform.RotateAround(headVR.position, Vector3.up, snapValue);
}
}
}

```


References

- 2TH prod. (2020, May 14). *Advanced ANT+ | Network | Unity Asset Store*.
<https://assetstore.unity.com/packages/tools/network/advanced-ant-71980>
- Babisch, W., Fromme, H., Beyer, A., & Ising, H. (2001). Increased catecholamine levels in urine in subjects exposed to road traffic noise: The role of stress hormones in noise research. *Environment International*, 26(7–8), 475–481. [https://doi.org/10.1016/s0160-4120\(01\)00030-7](https://doi.org/10.1016/s0160-4120(01)00030-7)
- Babu, S., Grechkin, T., Chihak, B., Ziemer, C., Kearney, J., Cremer, J., & Plumert, J. (2011). An Immersive Virtual Peer for Studying Social Influences on Child Cyclists' Road-Crossing Behavior. *IEEE Transactions on Visualization and Computer Graphics*, 17, 14–25.
<https://doi.org/10.1109/TVCG.2009.211>
- Balton, J. (2021, March 26). *Bike Statistics & Facts*. Bicycle Guider.
<https://en.rayhaber.com/2020/05/How-many-bicycles-are-there-in-the-world/>
- Beanland, V., & Hansen, L. J. (2017). Do cyclists make better drivers? Associations between cycling experience and change detection in road scenes. *Accident Analysis & Prevention*, 106, 420–427. <https://doi.org/10.1016/j.aap.2017.07.013>
- Casali, J. G. (1985). *Vehicular Simulation-Induced Sickness. Volume 1. An Overview*. Virginia Polytechnic Inst and State Univ Blacksburg Dept of Industrial Engineering and Operations Research. <https://apps.dtic.mil/sti/citations/ADA173904>
- Census Bureau. (2018). *Commuting characteristics by sex—2018: American Community Survey 1-Year Estimates Subject Tables*.
<https://data.census.gov/cedsci/table?q=S0801%3A%20COMMUTING%20CHARACTERISTICS%20BY%20SEX&hidePreview=true&tid=ACSST1Y2018.S0801>
- Chang, K.-H. (2013). Chapter 3—Motion Analysis. In K.-H. Chang (Ed.), *Product Performance Evaluation with CAD/CAE* (pp. 121–204). Academic Press.
<https://doi.org/10.1016/B978-0-12-398460-9.00003-2>

- Cipresso, P., Giglioli, I. A. C., Raya, M. A., & Riva, G. (2018). The Past, Present, and Future of Virtual and Augmented Reality Research: A Network and Cluster Analysis of the Literature. *Frontiers in Psychology, 9*. <https://doi.org/10.3389/fpsyg.2018.02086>
- Combs, T. S., & Pardo, C. F. (2021). Shifting streets COVID-19 mobility data: Findings from a global dataset and a research agenda for transport planning and policy. *Transportation Research Interdisciplinary Perspectives, 9*, 100322. <https://doi.org/10.1016/j.trip.2021.100322>
- Davies, E. (2020, May 15). What do bikes and toilet paper have in common? Both are flying out of stores amid the coronavirus pandemic. *Washington Post*. https://www.washingtonpost.com/local/what-do-bikes-and-toilet-paper-have-in-common-both-are-flying-out-of-stores-amid-the-coronavirus-pandemic/2020/05/14/c58d44f6-9554-11ea-82b4-c8db161ff6e5_story.html
- Deb, S., Carruth, D. W., Sween, R., Strawderman, L., & Garrison, T. M. (2017). Efficacy of virtual reality in pedestrian safety research. *Applied Ergonomics, 65*, 449–460. <https://doi.org/10.1016/j.apergo.2017.03.007>
- Dialynas, G., Happee, R., & Schwab, A. (2019). Design and hardware selection for a bicycle simulator. *Mechanical Sciences, 10*, 1–10. <https://doi.org/10.5194/ms-10-1-2019>
- EPA. (2018). *Sources of Greenhouse Gas Emissions* [Overviews and Factsheets]. United States Environmental Protection Agency. <https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions>
- EPA. (2019). *Smog, Soot, and Other Air Pollution from Transportation* [Overviews and Factsheets]. United States Environmental Protection Agency. <https://www.epa.gov/transportation-air-pollution-and-climate-change/smog-soot-and-local-air-pollution>
- Feng, F., Bao, S., Hampshire, R. C., & Delp, M. (2018). Drivers overtaking bicyclists—An examination using naturalistic driving data. *Accident Analysis & Prevention, 115*, 98–109. <https://doi.org/10.1016/j.aap.2018.03.010>
- Feng, F., & Hamad, A. (2021). *Development of a High Fidelity Virtual Reality Cycling Simulator for Road Safety Education and Research*. Transportation Research Board 100th Annual Meeting.

- Filippov, S. (2017, November 30). *3D Stark Temper Cross-Country Mountain Bike* | CGTrader. Cgtrader.Com. <https://www.cgtrader.com/free-3d-models/vehicle/bicycle/stark-temper-cross-country-mountain-bike>
- Garmin Canada Inc. (2012, November 15). *ANT / ANT+ Defined. This Is ANT.* <https://www.thisisant.com/developer/ant-plus/ant-antplus-defined/>
- Goddard, T., McDonald, A. D., Alambeigi, H., Kim, A. J., & Anderson, B. A. (2020). Unsafe bicyclist overtaking behavior in a simulated driving task: The role of implicit and explicit attitudes. *Accident Analysis & Prevention, 144*, 105595. <https://doi.org/10.1016/j.aap.2020.105595>
- Habgood, J., Moore, D., Wilson, D., & Alapont, S. (2018, March 18). Rapid, Continuous Movement Between Nodes as an Accessible Virtual Reality Locomotion Technique. *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. <https://doi.org/10.1109/VR.2018.8446130>
- Hakkinen, J., Vuori, T., & Paakka, M. (2002). Postural stability and sickness symptoms after HMD use. *IEEE International Conference on Systems, Man and Cybernetics, 1*, 147–152. <https://doi.org/10.1109/ICSMC.2002.1167964>
- Harm, D. L., Taylor, L. C., & Bloomberg, J. J. (2007, January 1). *Adaptive Changes in Sensorimotor Coordination and Motion Sickness Following Repeated Exposures to Virtual Environments*. NASA Technical Reports Server (NTRS). <https://ntrs.nasa.gov/citations/20070006524>
- Hettinger, L. J., & Riccio, G. E. (1992). Visually Induced Motion Sickness in Virtual Environments. *Presence: Teleoperators and Virtual Environments, 1*(3), 306–310. <https://doi.org/10.1162/pres.1992.1.3.306>
- Horne, D., Abadi, M. G., & Hurwitz, D. S. (2018). Bicycling Simulator Calibration: Proposed Framework. *Transportation Research Record, 2672*(37), 11–18. <https://doi.org/10.1177/0361198118790328>
- HTC. (2021a). *Play Area*. Vive Support. https://www.vive.com/eu/support/vive/category_howto/play-area.html
- HTC. (2021b). *VIVE Pro Eye Overview* | *VIVE™*. <https://www.vive.com/eu/product/vive-pro-eye/overview/>
- Jacobsen, P., Racioppi, F., & Rutter, H. (2009). Who owns the roads? How motorised traffic discourages walking and cycling. *Injury Prevention : Journal of the International Society*

- for *Child and Adolescent Injury Prevention*, 15, 369–373.
<https://doi.org/10.1136/ip.2009.022566>
- Jayaraman, S., Tilbury, D., Yang, X. J., Pradhan, A., & Robert, L. (2020, March 22). Analysis and Prediction of Pedestrian Crosswalk Behavior during Automated Vehicle Interactions. *2020 IEEE International Conference on Robotics and Automation (ICRA)*.
- Jeong, S. H., Piao, Y. J., Chong, W. S., Kim, Y. Y., Lee, S. M., Kwon, T. K., Hong, C. U., & Kim, N. G. (2005). The Development of a New Training System for Improving Equilibrium Sense Using a Virtual Bicycle Simulator. *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2567–2570.
<https://doi.org/10.1109/IEMBS.2005.1616993>
- Jones, S. M., Jones, T. A., Mills, K. N., & Gaines, G. C. (2009). Anatomical and Physiological Considerations in Vestibular Dysfunction and Compensation. *Seminars in Hearing*, 30(4), 231–241. <https://doi.org/10.1055/s-0029-1241124>
- Jung, S., & Whangbo, T. (2017). Study on inspecting VR motion sickness inducing factors. *2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, 1–5. <https://doi.org/10.1109/CAIPT.2017.8320660>
- Kaptein, N., Theeuwes, J., & van der horst, R. (1996). Driving Simulator Validity: Some Considerations. *Transportation Research Record*, 1550, 30–36.
<https://doi.org/10.3141/1550-05>
- Kim, H. G., Lim, H., Lee, S., & Ro, Y. M. (2019). VRSA Net: VR Sickness Assessment Considering Exceptional Motion for 360° VR Video. *IEEE Transactions on Image Processing*, 28(4), 1646–1660. <https://doi.org/10.1109/TIP.2018.2880509>
- Kim, N.-G., Kim, Y.-Y., & Kwon, T.-K. (2006). Development of a Virtual Reality Bicycle Simulator for Rehabilitation Training of Postural Balance. In M. Gavrilova, O. Gervasi, V. Kumar, C. J. K. Tan, D. Taniar, A. Laganá, Y. Mun, & H. Choo (Eds.), *Computational Science and Its Applications—ICCSA 2006* (pp. 241–250). Springer.
https://doi.org/10.1007/11751540_26
- Lague, S. (2019). *Bézier Path Creator | Utilities Tools | Unity Asset Store*.
<https://assetstore.unity.com/packages/tools/utilities/b-zier-path-creator-136082>
- LaViola, J. J. (2000). A Discussion of Cybersickness in Virtual Environments. *SIGCHI Bulletin*, 32(1), 47–56.

- Leeuwen, P. M. van, Happee, R., & de Winter, J. C. F. (2015). Changes of Driving Performance and Gaze Behavior of Novice drivers During a 30-min Simulator-based Training. *Procedia Manufacturing*, 3, 3325–3332. <https://doi.org/10.1016/j.promfg.2015.07.422>
- McCauley, M. E., & Sharkey, T. J. (1992). Cybersickness: Perception of Self-Motion in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 1(3), 311–318. <https://doi.org/10.1162/pres.1992.1.3.311>
- Merhi, O., Faugloire, E., Flanagan, M., & Stoffregen, T. A. (2007). Motion sickness, console video games, and head-mounted displays. *Human Factors*, 49(5), 920–934. <https://doi.org/10.1518/001872007X230262>
- NHTSA. (2019). *Bicyclists and other cyclists: 2017 data*. (Project Report DOT HS 812 765). National Highway Traffic Safety Administration.
- NHTSA. (2020). *Overview of Motor Vehicle Crashes in 2019* (Traffic Safety Facts Research DOT HS 813 060). National Highway Traffic Safety Administration.
- Prothero, Jerald D. (1998). *The role of rest frames in vection, presence and motion sickness* [University of Washington]. http://papers.cumincad.org/data/works/att/diss_prothero.content.04134.pdf
- Pucher, J., & Buehler, R. (2017). Cycling towards a more sustainable transport future. *Transport Reviews*, 37(6), 689–694. <https://doi.org/10.1080/01441647.2017.1340234>
- Rebenitsch, L., & Owen, C. (2014, October). Individual variation in susceptibility to cybersickness. *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. <https://doi.org/10.1145/2642918.2647394>
- Schramka, F., Arisona, S., Joos, M., & Erath, A. (2017). Development of Virtual Reality Cycling Simulator. *Arbeitsberichte Verkehrs- Und Raumplanung*, 1244. <https://doi.org/10.3929/ethz-b-000129869>
- Stanney, K., Lawson, B. D., Rokers, B., Dennison, M., Fidopiastis, C., Stoffregen, T., Weech, S., & Fulvio, J. M. (2020). Identifying Causes of and Solutions for Cybersickness in Immersive Technology: Reformulation of a Research and Development Agenda. *International Journal of Human–Computer Interaction*, 36(19), 1783–1803. <https://doi.org/10.1080/10447318.2020.1828535>
- Sun, C., & Qing, Z. (2018). Design and Construction of a Virtual Bicycle Simulator for Evaluating Sustainable Facilities Design. *Advances in Civil Engineering*, 2018. <https://doi.org/10.1155/2018/5735820>

- Suzuki, M., Sato, K., Hosoya, K., Miyanoue, K., & Yai, T. (2018). *A Fundamental Study of Validation of Cycling Simulator and Designing Safety Education Programs* (No. 18–02484). Article 18–02484. Transportation Research Board 97th Annual Meeting Transportation Research Board. <https://trid.trb.org/view/1495332>
- Taha. (2020, May 6). *Rigged Adrian Gochila 3D Rigged Character man* | CGTrader. <https://www.cgtrader.com/free-3d-models/character/man/adrian-gochila-3d-rigged-character>
- The NPD Group. (2020, June 16). *2020 Bicycle Market | Sales Statistics*. NPD. <https://www.npd.com/wps/portal/npd/us/news/press-releases/2020/cycling-industry-sales-growth-accelerates-in-april/>
- Unity Technologies. (2017). *Unity Manual: Scenes*. Unity Documentation. <https://docs.unity3d.com/560/Documentation/Manual/CreatingScenes.html>
- Unity Technologies. (2018, July 31). *Unity - Manual: Prefabs*. <https://docs.unity3d.com/Manual/Prefabs.html>
- Unity Technologies. (2019). *Windridge City | 3D Roadways | Unity Asset Store*. <https://assetstore.unity.com/packages/3d/environments/roadways/windridge-city-132222>
- Unity Technologies. (2021a). *Solutions | Unity*. <https://unity.com/solutions>
- Unity Technologies. (2021b). *Unity Manual: Character Controller*. Unity Documentation. <https://docs.unity3d.com/2019.4/Documentation/Manual/class-CharacterController.html>
- Unity Technologies. (2021c). *Unity—Manual: Preparing Assets for Unity*. <https://docs.unity3d.com/2019.3/Documentation/Manual/BestPracticeMakingBelievableVisuals1.html>
- Valve. (2020). *SteamVR Unity Plugin Documentation*. SteamVR API Documentation. https://valvesoftware.github.io/steamvr_unity_plugin/api/index.html
- Valve. (2021). *HTC Vive Installation Guide*. Steam Support. https://support.steampowered.com/steamvr/HTC_Vive/
- Yamaguchi, T., Yamada, A., Fujiwara, T., Takahashi, R., Shimada, T., Ishida, Y., Ueda, K., & Kaneda, S. (2018). Development of Bicycle Simulator with Tilt Angle Control Tilt Angle. *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 02, 247–252. <https://doi.org/10.1109/COMPSAC.2018.10238>