

EmbarkVR: Outdoor Virtual Reality Experience

CS Senior Capstone

Final Report

Jake Jeffreys, McKenna Jones, Spike Madden, Sean Marty

May 31, 2017

Abstract

This document summarizes the Outdoor Experiences in Virtual Reality project for the Computer Science Senior Capstone class at Oregon State University. In the document you will see original technical documents, weekly blog posts, project documentation and other general information regarding the project. We hope that this document will be equally valuable to high school students, future CS Capstone Students and the OSU CS department.

CONTENTS

1	Project Overview	2
2	Requirements	2
2.1	Original Requirements Document	2
2.2	Requirements Document Changes	11
2.3	Final Gantt Chart	11
3	Design Document	11
3.1	Original Design Document	11
3.2	Design Document Changes	28
4	Tech Review	28
4.1	Original Tech Review	28
4.2	Tech Review Changes	49
5	Weekly Blog Posts	49
5.1	Fall Week 3	49
5.1.1	Jake	49
5.1.2	McKenna	49
5.1.3	Sean	50
5.1.4	Spike	50
5.2	Fall Week 4	50
5.2.1	Jake	50
5.2.2	McKenna	51
5.2.3	Sean	51
5.2.4	Spike	52
5.3	Fall Week 5	52
5.3.1	Jake	52
5.3.2	McKenna	53
5.3.3	Sean	53
5.3.4	Spike	53
5.4	Fall Week 6	54
5.4.1	Jake	54
5.4.2	McKenna	54
5.4.3	Sean	55
5.4.4	Spike	55

5.5	Fall Week 7	56
5.5.1	Jake	56
5.5.2	McKenna	56
5.5.3	Sean	57
5.5.4	Spike	57
5.6	Fall Week 8	57
5.6.1	Jake	57
5.6.2	McKenna	58
5.6.3	Sean	58
5.6.4	Spike	59
5.7	Fall Week 9	59
5.7.1	Jake	59
5.7.2	McKenna	59
5.7.3	Sean	60
5.7.4	Spike	60
5.8	Fall Week 10	61
5.8.1	Jake	61
5.8.2	McKenna	61
5.8.3	Sean	61
5.8.4	Spike	62
5.9	Winter Week 1	62
5.9.1	Jake	62
5.9.2	McKenna	62
5.9.3	Sean	63
5.9.4	Spike	63
5.10	Winter Week 2	63
5.10.1	Jake	63
5.10.2	McKenna	64
5.10.3	Sean	64
5.10.4	Spike	65
5.11	Winter Week 3	65
5.11.1	Jake	65
5.11.2	McKenna	65
5.11.3	Spike	66
5.12	Winter Week 4	66

5.12.1	Jake	66
5.12.2	McKenna	67
5.12.3	Sean	67
5.12.4	Spike	67
5.13	Winter Week 5	68
5.13.1	Jake	68
5.13.2	McKenna	68
5.13.3	Sean	69
5.13.4	Spike	69
5.14	Winter Week 6	69
5.14.1	Jake	69
5.14.2	McKenna	70
5.14.3	Sean	70
5.14.4	Spike	71
5.15	Winter Week 7	71
5.15.1	Jake	71
5.15.2	McKenna	71
5.15.3	Sean	72
5.15.4	Spike	72
5.16	Winter Week 8	72
5.16.1	Jake	72
5.16.2	McKenna	73
5.16.3	Sean	73
5.16.4	Spike	74
5.17	Winter Week 9	74
5.17.1	Jake	74
5.17.2	McKenna	74
5.17.3	Sean	75
5.17.4	Spike	75
5.18	Winter Week 10	75
5.18.1	Jake	75
5.18.2	McKenna	76
5.18.3	Spike	76
5.19	Spring Week 1	77
5.19.1	Jake	77

5.19.2	McKenna	77
5.19.3	Spike	77
5.20	Spring Week 2	78
5.20.1	Jake	78
5.20.2	McKenna	78
5.20.3	Spike	79
5.21	Spring Week 3	79
5.21.1	Jake	79
5.21.2	McKenna	80
5.21.3	Spike	80
5.22	Spring Week 4	80
5.22.1	Jake	80
5.22.2	McKenna	81
5.22.3	Sean	81
5.22.4	Spike	82
5.23	Spring Week 5	82
5.23.1	Jake	82
5.23.2	McKenna	82
5.23.3	Sean	83
5.23.4	Spike	83
5.24	Spring Week 6	83
5.24.1	Jake	83
5.24.2	McKenna	84
5.24.3	Sean	84
5.24.4	Spike	84
5.25	Spring Week 7	85
5.25.1	Jake	85
5.25.2	McKenna	85
5.25.3	Sean	86
5.25.4	Spike	86
6	Project Documentation	90
6.1	Technical Overview	90
6.2	How to Install and Run	90
6.3	Hardware Requirements	90
7	How We Learned	90

8	What We Learned	91
8.1	Jake	91
8.2	McKenna	92
8.3	Sean	93
8.4	Spike	93
9	Appendix	95
9.1	Essential Code Listings	95
9.1.1	FishingLineLogic.cs	95
9.1.2	FishLogic.cs	98
9.2	Images	102

1 PROJECT OVERVIEW

This project was sponsored and proposed by Intel and Columbia Sportswear. The two stakeholders in the project are Mike Premi from Intel and Tim Devlin from Columbia Sportswear. Mike Premi served as the technical adviser, and Tim Devlin served as the conceptual adviser. The motivation behind the project is to both inspire people to get outdoors, and promote the Performance Fishing Gear line of fishing apparel by creating an immersive and interactive fishing experience. The members of the team, Jake Jeffreys, McKenna Jones, Sean Marty, and Spike Madden all contributed equally to documentation and development of the project. This project serves as a strong proof of concept of what a VR experience could look like in a retail space.

2 REQUIREMENTS

2.1 Original Requirements Document

EmbarkVR: Outdoor Virtual Reality Experience

CS Senior Capstone

Software Requirements Specification

Jake Jeffreys, McKenna Jones, Spike Madden, Sean Marty

November 4, 2016

Abstract

Many outdoor activities these days initially require a large mental and economic investment to get started. This makes people less likely to try new outdoor activities. The goal of the project is to develop an interactive product demonstration with virtual reality to combat this issue. This project has the potential to inspire people to get outdoors and try new things by first getting them comfortable in new environments or performing new movements. It strives to make outdoor activities accessible to everyone no matter their experience level. This project will not only inspire but also improve the retail experience by making it more immersive, interactive, and informative while being entertaining. The main tool being used is Unity Gaming Engine in tandem with an HTC Vive Virtual Reality System. The objective of this project is to create a functional VR outdoor experience ready to be piloted in a Columbia retail store by May of 2017.

CONTENTS

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions	2
1.4	References	2
1.5	Overview	2
2	Overall Description	3
2.1	Product Perspective	3
2.2	Product Functions	3
2.3	User Characteristics	3
2.4	Constraints	3
2.5	Assumptions and Dependencies	4
2.6	Apportioning of Requirements	4
3	Specific Requirements	5
3.1	External Interfaces	5
3.2	Functions	5
3.3	Performance Requirements	5
3.4	Software System Attributes	5
3.4.1	Reliability	5
3.4.2	Availability	6
3.4.3	Security	6
3.4.4	Maintainability	6
3.4.5	Portability	6
4	Signatures	7

1 INTRODUCTION

1.1 Purpose

The main goal of the project is to make customers feel more inclined to purchase Columbia gear through the use of an immersive, outdoor Virtual Reality experience. This document exists both for development of the project and to provide a detailed description of the technical requirements for the clients.

1.2 Scope

We want to create an outdoor virtual reality experience for customers at a Columbia retail store. The application will consist mainly of visual, audio, and tactile experiences to create an outdoor world in which the user can navigate. The main activity available will involve fly fishing in one of the rivers within the environment. Users will also have the ability to interact with Columbia products while in the experience and gain specific product information.

1.3 Definitions

Virtual Reality (VR)	Artificial environment that is created with software
HTC Vive	A virtual reality headset produced by HTC
Unity Game Engine	The Unity Game Engine, developed by Unity Technologies is used in this project to develop the virtual reality simulation.
Avatar	An icon or figure representing a particular person.
Wands	Controllers that are used with the HTC headset.
Base Stations	These allow the Vive to track the movement and location of the wands and headset.

1.4 References

- [1] S. Michalak and E. Lind, "Virtual reality heuristics, results from user testing for prioritization and development," Sep 2016.
- [2] ——, "Virtual reality heuristics, results from user testing round 2," Oct 2016.
- [3] H. Corporation, "Vive ready computers," 2016.
- [4] C. Hall, "Sony to devs: If you drop below 60 fps in vr we will not certify your game," 2016. [Online]. Available: <http://www.polygon.com>

1.5 Overview

The next chapter of the document will give an overview of the basic functionality of the Virtual Reality application. It contains informal requirements to provide background for section three, Specific Requirements. Section three will provide more detailed requirements and is intended for a more technical audience such as developers.

2 OVERALL DESCRIPTION

2.1 Product Perspective

This VR product may be new to Columbia Sportswear but will still have ties to existing products. Within the experience, users will have the ability to view and interact with Columbia gear. Users will also be given the opportunity to wear Columbia gear while participating to learn how the clothes feel while executing certain movements.

The product will rely heavily on Virtual Reality Hardware. Specifically, the HTC Vive System. This system consists of the the headset, two wands, and two base stations. Additionally a Virtual Reality compatible computer is needed to actually run the software. In terms of software the product will rely on the Unity Game Engine. Unity will do the heavy lifting when it comes to rendering the virtual environment and making it look as realistic as possible.

2.2 Product Functions

The VR setup will allow the user to simulate outdoor experiences. Specifically, the user will be able to virtually see Columbia Sportswear gear in the environments they are intended to be used in. This will be done using 3D renderings of Columbia items which will then be placed within the environment. We will then be using Unity to design VR interaction capabilities on top of the renderings. The final product will allow the user to test a variety of clothes and equipment. This product will also give customers the ability to save the gear they liked in the VR, and access that information after the experience is over.

2.3 User Characteristics

The general type of user of this project will be a customer at a Columbia Sportswear retail store. Under this umbrella lies a few different types of customers. First there are customers who are inexperienced in the outdoor activity they are buying gear for. This target audience will benefit most from the VR expreience as it will allow them to experience the activity without a lot of economic or time commitment. Secondly, there are customers who are experienced in the activity they are buying gear for. This audience will benefit from the VR experience becauseit will allow them to view themselves actually using and testing out new gear.

2.4 Constraints

A virtual reality headset like the HTC Vive has some inherent restrictions. The first one is space. The HTC Vive tracks how much space you have set the system up in and creates virtual barriers. This limitation can be alleviated by using the controllers to move the users within VR environment. Besides physical space, space in the virtual display is also a concern. Information needs to be supplied to the user without obstructing the VR experience. A second limitation are the graphics within the VR

environment are not entirely realistic. According to studies done by Intel and Thug Design [1][2], the categories most important to the feeling of immersion are realistic interactions, responsiveness, graphic clarity, and smooth transitions in that order. Because we are trying to promote Columbia Sportswear Gear in a realistic environment it needs to be as authentic as possible.

2.5 Assumptions and Dependencies

An important assumption made in this requirements document is that the virtual reality experience will be run on a computer system that can run the HTC Vive software. The following are the minimum specifications to run Vive, as found on the HTC Vive website [3]:

- Processor: Intel Core i5-4590 or AMD FX 8350, or better
- Graphics: NVIDIA GeForce GTX 1060 or AMD Radeon RX480, or better
- Memory: 4GB RAM or better
- Operating system: Windows 7 SP1 or better

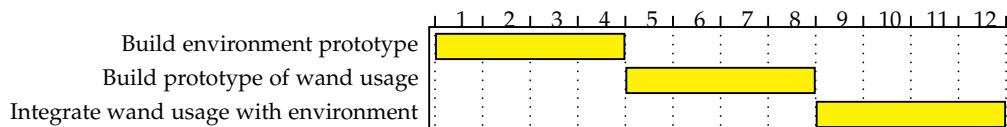
Design decisions and optimizations will be made so that a computer with the above specifications can run the experience with little noticeable lag, but if the machine drops below the minimum capabilities, the requirements regarding responsiveness will have to change.

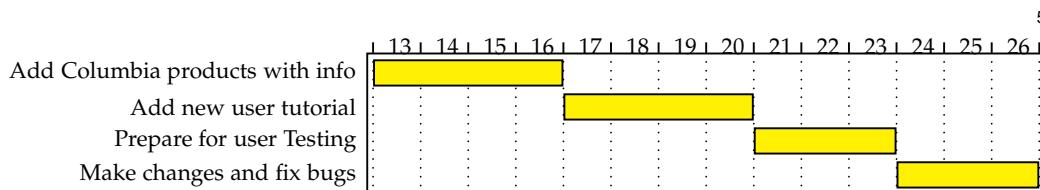
Also, the requirements often depend on the availability of a set of Columbia and Unity 3D assets. If either of those sources of assets is not available, the requirements about being able to see Columbia Sportswear gear in a realistic environment will have to change.

2.6 Apportioning of Requirements

One part of the project that will likely be delayed until later versions is the social aspect. Ideally the user would be able to share their VR experience on social media sites like Facebook or Youtube. This could be in the form of 360 degree images or videos. At the moment this requirement is not a high priority.

Gant Chart (measured in weeks)





3 SPECIFIC REQUIREMENTS

3.1 External Interfaces

- 360 degree view of outdoor scenario within VR experience using HTC Vive headset. This will contain optional user guidance (visual) and offer Columbia product information (visual).
 - Input: Movement of headset
 - Output: Visual data
- Immersive noises from outdoor VR experience. This includes audio from the optional user guidance.
 - Output: Audio through speakers and/or headphones.
- Ability for other users not using headset to see user's current view.
 - Output: Visual data on external monitor.
- Controller available to be held by user to interact within VR experience.
 - Input: HTC Wand movement

3.2 Functions

- Ability for users to interact with fly fishing equipment.
- Ability to see Columbia fishing apparel in use.

3.3 Performance Requirements

- Must maintain at least 60fps throughout experience.[4]

3.4 Software System Attributes

Our unity environment should be portable and work on all HTC Vive systems. Correctness can be evaluated by how authentic the real-world experience we're trying to replicate is. We should be able to adjust the scenes easily and accommodate for any changes that the client wants.

3.4.1 Reliability

The system will be considered reliable if it can provide the virtual reality experience consistently without failure. Failure can be defined as any technical issue that breaks immersion. This could include noticeable lag between the user and avatar's actions, distortion of the environment, or a software defect that causes the system to crash.

3.4.2 Availability

The system should be available to customers whenever the virtual reality station in the store is set up. There should be no downtime when the customer puts on the headset and the customer should almost immediately be placed in the immersive environment.

Many of the attributes of our system (such as availability and reliability) will be handled through the Unity game engine.

3.4.3 Security

The environment will be designed to not require a system with internet access, and will not need any user information. The expected implementation will be on a closed system in a Columbia store, where physical and remote access will be controlled by Columbia employees.

3.4.4 Maintainability

As time goes on it should be easy to adapt the product to easily include more Columbia Sportswear products and virtual reality environments to accompany them. Changes to the environment through the addition of assets (new Columbia gear) should not be difficult and an updated scene should be accomplished with a small patch.

3.4.5 Portability

The environment should work on all HTC Vive systems with the minimum requirements to be Vive Ready [3].

4 SIGNATURES

Document: Software Requirements Specification (EmbarkVR)

Intel Sponsor

Date

Columbia Sponsor

Date

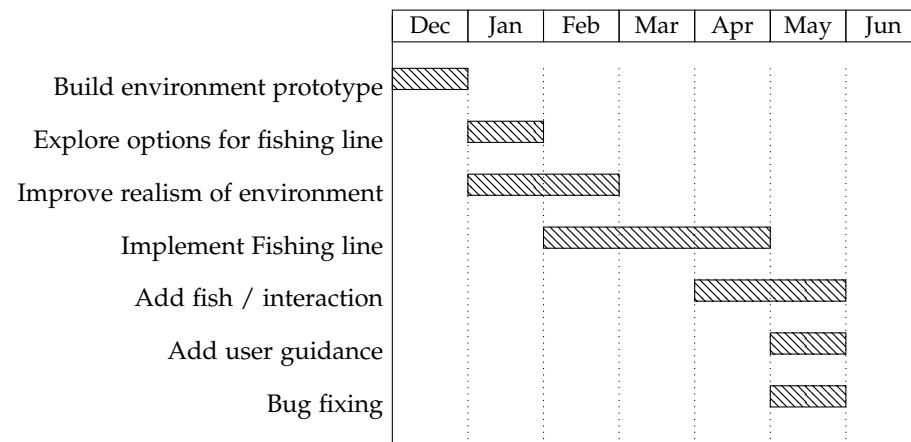
Student Team Members

Date

2.2 Requirements Document Changes

#	Requirement	What Happened To It	Comments
1	See Columbia gear in VR environment	We produced placeholders and examples, but no actual Columbia assets included	This changed mostly as a function of project time limit and roadblocks with Columbia
2	Fly-fishing experience	Changed to normal fishing	For easier physics and simpler activity
3	Optional audio guidance	Removed this requirement	Not a high priority in limited time
4	User Testing	Removed this requirement	Did not produce beta-ready product in limited time

2.3 Final Gantt Chart



3 DESIGN DOCUMENT

3.1 Original Design Document

EmbarkVR: Outdoor Virtual Reality Experience

CS Senior Capstone

Design Document

Jake Jeffreys, McKenna Jones, Spike Madden, Sean Marty

December 2nd, 2016

Abstract

This document describes the design components of Intel and Columbia's virtual reality project. It is structured as follows. A brief overview section will outline the purpose, intended audience, definitions and project context. This is followed by the body of the document which further details the design of the project. The design description defines the stakeholders, views and viewpoints defined in the IEEE 1016 format. The project was broken up into four components: static environment, animation, Columbia gear integration and fishing rod mechanics. Each of these sections are discussed in depth in terms of design description.

Intel Sponsor _____ Date _____

Columbia Sponsor _____ Date _____

Student Team Members _____ Date _____

CONTENTS

1	Overview	3
1.1	Scope	3
1.2	Purpose	3
1.3	Intended Audience	3
2	Definitions	3
3	Project Context	3
3.1	Hardware	3
3.2	Software	4
4	Design Description	4
4.1	Design stakeholders	4
4.1.1	Intel	4
4.1.2	Columbia Sportswear	4
4.2	Design views	4
4.2.1	Users	4
4.2.2	Intel Sponsor (Mike Premi)	5
4.2.3	Columbia Sponsor (Tim Devlin)	5
4.3	Design viewpoints	5
4.3.1	Context viewpoint	5
4.3.2	Composition viewpoint	6
4.3.3	Dependency viewpoint	7
4.3.4	Interface viewpoint	9
4.4	Design Rationale	9
5	Approach	10
5.1	Static Environment	10
5.1.1	Concerns	10
5.1.2	Approach	10
5.2	Improve Realism and Animate Environment	11
5.2.1	Concerns	11
5.2.2	Aspects	11
5.3	Tactile User Interaction	12
5.3.1	Concerns	12
5.3.2	Aspects	12

		2
5.4	Rod mechanics	13
5.4.1	Concerns	13
5.4.2	Aspects	14
	References	15

1 OVERVIEW

1.1 Scope

We want to create an outdoor virtual reality experience for customers at a Columbia retail store. The application will consist mainly of visual, audio, and tactile experiences to create an outdoor world in which the user can navigate. The main activities available will involve fly fishing in one of the rivers and interacting with virtual Columbia products while in the experience to gain specific product information.

1.2 Purpose

The main goal of the project is to make customers feel more inclined to purchase Columbia gear through the use of an immersive, outdoor virtual reality experience. This document exists both for development of the project and to provide a detailed description of the design plans.

1.3 Intended Audience

The intended audience of this design document are the student developers involved (EmbarkVR), project sponsors, and Capstone teachers. The development team will be using this report as a guide and will provide structure for the development process. The sponsors can use this document to understand the vision of the developers and it will give a platform to discuss design ideas. The teachers can benefit from this document by learning about the project as a whole.

2 DEFINITIONS

- Virtual Reality (VR): Artificial environment that is created with software
- HTC Vive: A virtual reality system produced by HTC
- Base Stations: These allow the Vive to track the movement and location of the controllers and headset.
- Wands: Controllers that are used with the Vive headset.
- Unity Game Engine: The Unity Game Engine, developed by Unity Technologies is used in this project to develop the virtual reality simulation.
- GameObject: The base class for all entities in Unity scenes.
- GitHub: Web-based Git repository hosting service
- Git: version control system used for software development

3 PROJECT CONTEXT

3.1 Hardware

- Computer with the following specifications:

- Processor: Intel Core i5-4590 or AMD FX 8350, or better
- Graphics: NVIDIA GeForce GTX 1060 or AMD Radeon RX480, or better
- Memory: 4GB RAM or better
- Operating system: Windows 7 SP1 or better
- HTC Vive Headset: Used to track head movements and display application to users.
- HTC Wands (x2): Used to track the users hand movements and to give the user the ability to interact with virtual objects within the application.
- HTC Base Stations (x2): Used to track location of headset and wands. This information is then sent back to the computer in real time.

3.2 Software

- Unity Gaming Engine: Used to develop the application.
- Unity Asset Store: Used to find objects which can be imported into the application.
- GitHub: Used by developers to collaborate and share files.

4 DESIGN DESCRIPTION

The following will discuss the stakeholders, views, viewpoints, and rationale behind the design of the project.

4.1 Design stakeholders

4.1.1 Intel

In this project, Intel is working with Columbia Sportswear to help them meet their needs. Intel has graciously provided all necessary hardware to our team to allow us to create a successful application.

4.1.2 Columbia Sportswear

One aspect of Columbia Sportswear is their fishing apparel. Specifically, the Performance Fishing Gear (PFG) line of apparel. Columbia hopes to use the application we are developing in a retail store to showcase the PFG line in a new medium. The goal is to inspire customers to try new outdoor activities with Columbia gear.

4.2 Design views

4.2.1 Users

Users of the product expect this virtual reality experience to be as realistic and immersive as possible. We will be making the assumption that users will be experiencing this application without any previous virtual reality or fly fishing experience. We are making this assumption to account for

everyone who may be interested in participating. From their perspective, they will care most about their ability to quickly understand how to move around and interact with objects. This means that we need to create intuitive tools and controls. Users can also expect to find visual queues and instructions within the experience.

There are two main perspectives that users will have when using this product. Firstly, users will be hoping to gain an outdoor experience that they might not otherwise have the opportunity to try. Therefore, realism is key in this view. Secondly, users will expect interaction with Columbia gear in a meaningful way. The user should leave the experience with a feeling of how the Columbia gear would perform in a certain environment.

4.2.2 Intel Sponsor (Mike Premi)

The Intel sponsor of the project, Mike Premi, is concerned more with the technical side of the project. Things like which technologies are used, the technical performance, and overall technical design considerations are all important under this view. This view will guide the design process on a technical level.

One of the main goals of the project is to make the experience as immersive and realistic as possible. To achieve this we will first need to use a high performance computer. Second, we will need to constantly be aware of performance restrictions during the development process, mainly while improving realism. The realism techniques we discuss later in this document will all take a toll on performance speeds and application responsiveness.

4.2.3 Columbia Sponsor (Tim Devlin)

The Columbia Sportswear sponsor of the project, Tim Devlin, is concerned primarily with the how the user will interact with Columbia products in the Virtual Reality experience. This includes, how products are displayed, what information related to the products is shown, and the user interaction with said products. Ultimately, the goal of the product under this view is to create more sales for Columbia Sportswear. Therefore, that is what is most important under this view. This view will guide the design process at a higher level compared to the view of the Intel Sponsor.

4.3 Design viewpoints

4.3.1 Context viewpoint

This context is related to user interaction within the application in the environment. It provides a "black box" view of the project which can be useful from the perspective of users. From the perspective of the user, everything should be intuitive and they should rarely notice any issues with frame rate.

Design Concern: The main concern will come from users of the VR application. The ability to interact seamlessly with the application is essential. Users do not want to experience performance lag or errors in functionality. They also should not need to ever leave the application to ask questions about how to perform tasks or interact with objects. It must be an intuitive application that is easy to understand in less than 30 seconds given a minor introduction beforehand.

Analytical Methods: Once we have a fully functioning prototype we will put our application through user testing. During the testing phase the application will be evaluated on the basis of immersion, enjoyment, and possibly influence over customer purchase decisions. After users test the application we will ask for feedback related to their experience in the form of oral questions or an anonymous written survey depending on what the user prefers.

Rationale: We included this viewpoint due to the importance of user immersion. The user experience in virtual reality applications is very sensitive and there are many factors that can influence it. According to a study done by Intel and Thug[1], immersion needs graphical fidelity, not realism. They found that what was important was crisp and clean graphics at all times. They also found that a smooth experience, one without glitches and lost frames, was the most important aspect of immersion. They even went a step further to argue that photo-realism is often times worse because it makes inaccuracies more obvious.

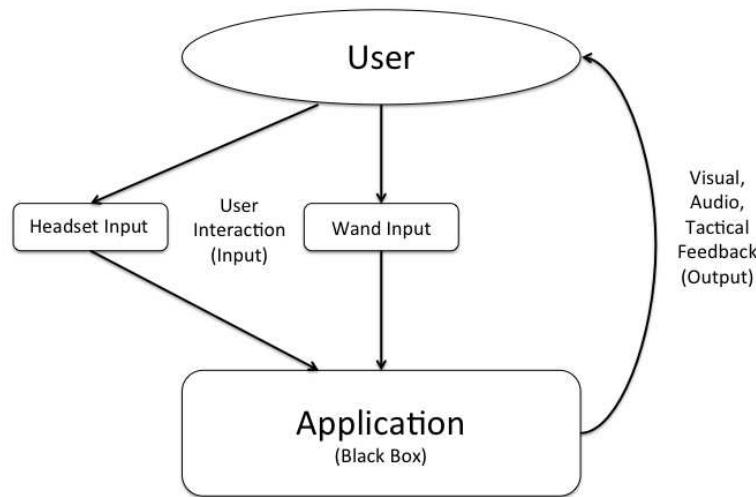
4.3.2 *Composition viewpoint*

Project composition is the organization of, and relationships between, each of the sections of our project. Each portion of the project will be explored in more detail in the Approach section below, but our virtual reality experience will be the sum of visual and audio assets, scripts, game objects, and HTC Vive hardware. All of this will be integrated to create a single, powerful retail experience.

Design Concern: The composition of the project affects ease of implementation, performance, and how well an outside developer can come in and familiarize themselves with the project. The Intel Sponsor Mike Premi will find this viewpoint important because of his role in guiding our technical development. Also, the Columbia Sponsor, Tim Devlin will find this viewpoint valuable because once the product is applied in a retail setting, a well-composed project will ease the transition and replication.

Analytical Methods: It will be hard to objectively analyze the success of this viewpoint, but the effects of a well or poorly composed project will undoubtedly be seen throughout development and implementation. A good test of quality composition will occur whenever parts of the project need

Fig. 1: Context Viewpoint Diagram



to be passed from one team member to another, or from the team to an outside developer. Project composition will be one of the factors in how well these transfers go.

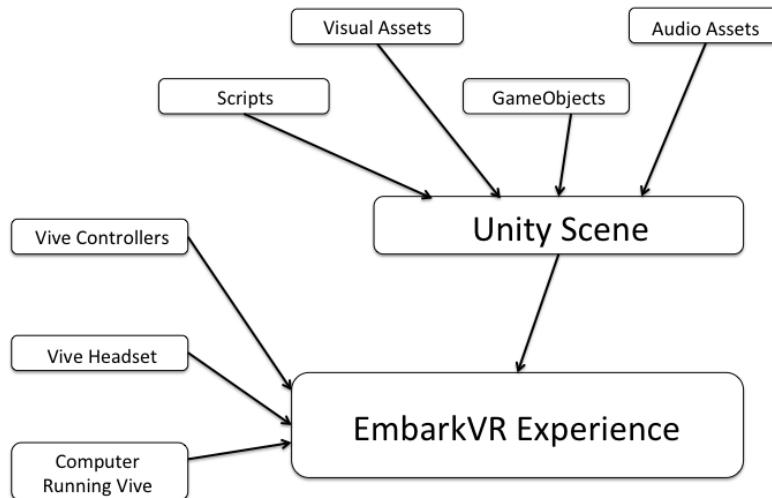
Rationale: We included the composition viewpoint because there are going to be a massive number of assets, scripts, and objects involved in the production of this virtual reality experience. It is vital to have our design keep an eye on composition so that nothing gets lost in the complexity of the undertaking. Also, as our team learns and improves our product, there will be periods of redesigning and modification that need to stay within the planned composition in order to not break other portions of the project.

4.3.3 Dependency viewpoint

The Dependency viewpoint specifies the relationships of interconnection and access among entities. These relationships include shared information, order of execution, or parameterization of interfaces. Establishing these relationships will be key in organizing the project's workflow.

Design Concern: Users will appreciate fluid and thoughtful interaction between assets within the virtual reality system. Both Intel sponsor Mike Premi and Columbia sponsor Tim Devlin will want

Fig. 2: Composition Viewpoint Diagram



clear dependencies to be outlined before any development begins to prevent any redundancies. This will save both time and resources during development and will result in a clear, aligned final product.

Analytical Methods: We will be able to find any dependency issues during development as certain components will not be functional without others. Development will halt any time a dependency issue arises that needs to be resolved. Ultimately, user testing with the final product will determine if the dependencies we outlined were implemented correctly.

Rationale: We included the dependency viewpoint because of the importance of planning ahead in a large software development project such as this. Design of the four major components, static assets, animation of the environment, rod mechanics, and Columbia gear integration are going to heavily rely on each other and need to be accounted for. The static environment will most likely be developed first, followed by the animation. Rod mechanics and Columbia gear integration will depend on both the created environment and available assets and animation. Successfully consolidating these elements will result in a polished final product.

4.3.4 *Interface viewpoint*

Within this project there are a number of interfaces, external and internal, that both the users and developers will be working with. The purpose of this viewpoint is to make it extremely clear how all of the interfaces work together to create a final product.

Design Concern: This viewpoint is primarily addressed at the technical side of the project. Therefore, the view with most interest in this viewpoint is the Intel Sponsor, Mike Premi. The main external interfaces are the HTC Vive system, which includes the headset, controllers, and base stations, the computer where the software is running, and any external monitors that are being used. As for internal interfaces, the project will involve the Unity Game Engine, SteamVR, and Visual Studio for C# development. In the short term, developers are most concerned with the interfaces, but long term, users will also see the impacts of the interface viewpoint. The orchestration of all interfaces is essential to creating a complete final project.

Analytical Methods: The creation of a product that works without any major issues depends on this viewpoint. All interfaces must play nicely with each of their dependencies for this to happen. During the design and development of the project, this viewpoint will guide the developers by providing an understanding of which interfaces of the project are important for each component. An indicator of good interface design will be if there are no interfaces which inhibit the functionality of other interfaces.

Rationale: This viewpoint was included because of the range of technologies that are used within the project. The way interfaces interact will affect users indirectly, and those with technical interests in the project directly.

4.4 Design Rationale

We chose to use the Unity gaming Engine to build our project. Here are some of the reasons for our decision:

- A large community base
- Compatible with Mac, Windows and Linux operating systems
- A relatively small learning curve, good for projects with a small time frame
- Built-in support for various VR related functions
- Includes a large, free asset store
- Uses C# as its scripting language, a language that our team has experience with

5 APPROACH

5.1 Static Environment

The basis of our virtual reality experience is the underlying static environment. Before we can start any animation, lighting, audio or physics work we need to build up a static terrain and collection of 3D objects. Most components of the environment will not just be static, but they will all start out that way when they are brought into our project. The basic project building block in Unity is a scene, which can be thought of as a level in a video game. [2] For our experience, the initial design will only have one scene because there isn't built-in user movement. This means that every object, texture, and script in our environment will exist within the scope of this single scene.

5.1.1 *Concerns*

This project component is most important from a dependency viewpoint. Before other aspects of the project can be completed, there needs to exist a static environment.

5.1.2 *Approach*

The main sections of the approach for this component are the static terrain, the realism of said terrain, static objects incorporated into the terrain, and finally assets used for the static environment.

Static Terrain: When building our static environment, we will start with the basic Unity terrain engine. A terrain object in Unity starts out as a flat plane, and then can be painted on with various effects.[3] These effects include heightmap effects and textures. In our environment, this will mean bringing out a riverbed, shore, and surrounding features such as hills with the heightmapping tool. Next up is adding base textures. Just adding a 2D texture for, say, grass will not provide nearly the level of authenticity we want the user to feel, so we will not use these textures for more than a base or background to 3D objects.

Static Realism: In the next section we will cover animation and realism through effects, but a certain amount of realism can be achieved just by careful choice of our static environment arrangement. We will do research on real world locations for fly fishing and draw from that as much as possible. For example, it is tempting to place trees in our environment in a semi-random, evenly spaced pattern or something similar. However, it will increase our authenticity if we base our placement choices on what actually occurs in nature. This is one example, but it demonstrates a larger theme, which is that because we are only producing a single scene and we want it to be as real as possible, attention to every reasonable detail is key.

Static GameObjects: Now that we have a terrain set up, the next step is to add static objects to the environment. At the most basic, Unity uses the GameObject type for all objects, and those objects have subobjects and components associated with them. [4] Without going into much detail about the actual workings of objects in Unity, there are two main aspects of game objects that we will use in our development to make our lives easier. Both of these features help solve problems that come with an environment that will contain lots of similar objects such as trees, stones, or grass. First, we will use prefabs to easily edit similar settings on a whole group of objects and make replication much easier. [4] Second, objects can be grouped into layers, which will provide us with the ability to toggle whole sections of our environment by theme while in development. [5] It will be helpful to be able to one-click toggle layers such as trees or small animals to clear up and expose underlying features.

Static Assets: It is perfectly good to be able to populate our environment with static objects but now we will need to actually create those visual 3D objects. In Unity, all project assets go into one central folder, which is brought into the editor for use. [6] There are all kinds of asset sources that we will explore, including specific Columbia asset files that will be provided by Columbia through Tim Devlin. Importing a quality set of assets will be the last step in creating a vibrant, realistic static environment to work with in our experience.

5.2 Improve Realism and Animate Environment

One of the main goals of our project is to make it as realistic as possible without compromising performance. Realism can come from a number of different techniques.

5.2.1 Concerns

From a dependency viewpoint it is important to have a static environment already created. However, in order to build a static environment we need to be aware of how dynamic textures work to effectively combine these two concepts.

5.2.2 Aspects

The three techniques we will be focusing on when it comes to improving realism are animation, audio, and lighting.

Animation: First we will be focusing on is environment animation. A majority of our application will take place in a river so we will need to make this river as animated as possible. This will involve an animation of the water moving passed the users and can be done using an open-source

animated water shading. A similar technique can be used to create movement of clouds in the sky.

Audio: The next step to improving realism is adding audio. Audio is crucial when it comes to immersion so not only will we need to add water noises but also noises related to wind and a wide range of animals. In Unity, sounds originate from Audio Sources attached to objects. Those sounds and audio clips can be found in any open-source audio library and easily imported into Unity.

Lighting and Shadowing: The last technique we will be focusing to improve realism is lighting and shadowing. This can be achieved using the built-in directional lighting tools within Unity.

5.3 Tactile User Interaction

The central idea behind this project is to promote Columbia gear in a realistic fly fishing experience. This requires the virtual reality project to allow for the user to interact with various pieces of Columbia gear. Along with direct interaction with apparel within the virtual reality world, the project should provide an overlay to display detailed information on specific Columbia Sportswear products.

5.3.1 *Concerns*

This section will examine how viewpoints affect the two components of tactile user interaction with Columbia gear: presence within the virtual reality environment and the product description overlay system.

Under the dependency viewpoint, presence of Columbia gear in the fly fishing experience will ultimately depend on the static environment and the animations. However, for development purposes, avatars with Columbia gear assets can be created independently. The product information panels, in a similar sense, will ultimately depend on the static environment. Once again, the overlay system can be developed independently as long as the product information is available. Integrating the panel would only be a matter of positioning and defining the correct context.

5.3.2 *Aspects*

The two aspects in this component are the the interaction with Columbia gear and the product information overlay.

Interaction with Columbia Gear: User interaction with Columbia gear will most likely be through various avatars placed in the virtual reality environment. Since the Vive doesn't allow for body tracking with just the wands, our sponsor steered us away from having Columbia apparel on the user's avatar. Users will be able to inspect and interact with the avatars that are showcasing the

Columbia gear. To achieve this, Columbia apparel assets and animations will be imported into the project. The avatars and Columbia apparel pieces will be animated using Unity's animation system, Mecanim. Mecanim uses Animation Clips which store information on how objects should adjust their position and other properties over time. These clips can come from third party digital content creation packages, motion capture studios or can be created within Unity. Animation Clips can be considered the building blocks for all animation sequences in Unity. These building blocks are organized into Mecanim's Animator Controller to chain various animations together. The Controller keeps track of the Animation Clips in a flowchart system and behaves like a state machine that keeps track of current Animation Clips and when they should change. The Controller is also capable of blending multiple clips to provide smooth transitions between animations. The Animation Clips and Animator Controller are aggregated into a GameObject through the Animator Component.[\[7\]](#)

Product Information Overlay: In addition to allowing users to interact with Columbia gear within the experience, Columbia product details need to be displayed to the user. This menu system needs to be unobtrusive in order to preserve the immersion of the experience. This will be done through the use of Unity's built-in UI element - the panel. The panels will include an image of the product, its specifications and an "Add to Cart" option.

5.4 Rod mechanics

In order to create a realistic fishing experience, the user will need to be able to interact with a virtual fishing rod. The user's interaction with the rod will be primarily based around the use of the HTC Vive controllers. Like other virtual reality simulations, in the game you will not see the Vive controllers, but instead virtual hands. The user will then be able to pick up the fishing rod using these virtual hands. To make this interaction as natural as possible the VR hands need to feel like an actual extension of the user's body. Once the user has picked up the fishing rod, it needs to behave as an actual rod would. This means that we will be using Unity's 3D physics engine extensively to create realistic movements with the fishing rod, line, and bait.

5.4.1 Concerns

This component of the project has interests in two different viewpoints. First, the dependency viewpoint will slightly determine when this component of the project can actually be started and completed. Mainly, it depends on the environment. It is possible to work on the basics of this component before the environment is completed; for example importing the required assets and the basic physics. However, the bulk of the work to create a realistic fishing interaction will need to be completed after the environment is in a finished state. Secondly, this component depends on the interface viewpoint.

The Vive external interfaces, primarily the headset and controllers, and required for this component. Therefore, whichever interfaces those are related to also come into play.

5.4.2 Aspects

The three aspects of this component are the HTC Vive controller models, the controller interaction with the Rod, and the physics of the fishing rod and line.

HTC Vive Controller Models: The hands assets can easily be downloaded from the Unity Asset store, where there are both free and paid options. Once these have been downloaded the next step is to map these to the Vive controllers. This is done by deleting the controller model in the controller GameObject and replacing it with the appropriate asset.

Controller Interaction With Rod: To pick up objects in the environment the user will use the trigger on the controller. This will be implemented by first monitoring for the trigger when the controller is near a GameObject. Unity provides the VRInteractiveItem component which can be attached to any GameObject, such as the rod.^[8] This component allows the user to interact easily with the object. When we detect an instance of this, the objects position can then be set to the same position as the Vive controller.

Fishing Rod and Line Physics: Creating ropes and cables in Unity is non-trivial. The preferred method is to use physics joints. The hinge joint GameObject is highly configurable, so the settings will need to be tweaked to create a realistic looking fishing line. For the fishing rod, hinge joints can also be used. Creating the physics to mimic the flex of a fishing rod is a matter of tweaking primarily the use limits of the hinge joints. The use limits determine the minimum and maximum angle to which a hinge joint may bend.^[9] In the case of the fishing line, the hinge joints have no limits because a fishing line can bend in any way. A fishing rod however, can only bend to a certain point. One can also control the bounciness of joints, which determines how much the object bounces when it hits the use limit. This will also be tweaked to control the flex of the rod.

REFERENCES

- [1] S. Michalak and E. Lind, "Virtual reality heuristics, results from user testing for prioritization and development," Sep 2016.
- [2] A. Tuliper, "Unity: Developing your first game with unity and c sharp," Aug 2014. [Online]. Available: <https://msdn.microsoft.com/en-us/magazine/dn759441.aspx>
- [3] U. Technologies, "Unity terrain engine documentation," 2016. [Online]. Available: <https://docs.unity3d.com/Manual/script-Terrain.html>
- [4] B. Moakley, "Unity tutorial part 2: Gameobjects," Aug 2016. [Online]. Available: <https://www.raywenderlich.com/142743/unity-tutorial-part-2-gameobjects>
- [5] U. Technologies, "Unity - layers video tutorial," 2016. [Online]. Available: <https://unity3d.com/learn/tutorials/topics/interface-essentials/layers>
- [6] ——, "Unity importing assets documentation," 2016. [Online]. Available: <https://docs.unity3d.com/Manual/ImportingAssets.html>
- [7] ——, "Unity animation," 2016. [Online]. Available: <https://docs.unity3d.com/Manual/AnimationSection.html>
- [8] ——, "Interaction in vr," 2016. [Online]. Available: <https://unity3d.com/learn/tutorials/topics/virtual-reality/interaction-vr?playlist=22946>
- [9] ——, "Physics joints," 2013. [Online]. Available: <https://unity3d.com/learn/tutorials/topics/physics/physics-joints>

3.2 Design Document Changes

Below is a list of the major changes that occurred between the original design document, and the final project we produced. The biggest theme, which will be evident throughout this document, is that we did not know exactly how much time and effort each feature would be, and overestimated what we could accomplish in nine around six months of development.

- A simple, straightforward change that we made was the change from fly-fishing to normal fishing. We decided that the problems involved with movements and physics in fly-fishing would not be worth the small benefits of making a fly fishing experience over a normal fishing experience.
- The biggest feature of our project that we changed was the amount of Columbia gear interaction that was present in the experience. Our project shows some possible ways to include Columbia gear but does not fully flesh out any of those features. We still see there being a lot of product interaction in the complete experience, but we favored other features when developing the project on a limited timeline.
- In the design document we mention user testing, and we ended up not getting that far in the product time line. We felt like we produced an awesome proof of concept to build on, but that our project was not yet to the point where user testing would be reasonable.

4 TECH REVIEW

4.1 Original Tech Review

EmbarkVR: Outdoor Virtual Reality Experience

CS Senior Capstone

Technology Review and Implementation Plan

Jake Jeffreys, McKenna Jones, Spike Madden, Sean Marty

November 14th, 2016

Abstract

Virtual reality is just starting to enter all kinds of markets. Everyone is trying to figure out the best way to implement their VR applications. Each VR project is going to be looking for different things when it comes to hardware and development environments. For our project we have been given an HTC Vive which handles the hardware but we still need to choose appropriate software for use in development. There are a variety of options and we'll be looking for one that is inexpensive, easy to learn, powerful enough to develop a realistic VR application, and easily configurable with SteamVR. In this document we break down the different aspects of our project and compare possible technology against specific requirements related to each piece.

CONTENTS

1	Introduction	2
2	Project Sections	2
2.1	Environment	2
2.1.1	Create Static Environment	2
2.1.2	Animate Environment	3
2.1.3	Add Audio	5
2.1.4	Improve Realism and Immersion	6
2.2	Fishing Activity	7
2.2.1	Process User Wand Movement	7
2.2.2	Import Fishing Assets	9
2.2.3	Fishing Rod Interaction and Mechanics	10
2.2.4	Integrate Usage with Environment	11
2.3	Columbia Products	12
2.3.1	Create Avatars	12
2.3.2	Import Columbia Gear	14
2.3.3	Animate Avatars	15
2.3.4	Allow User Interaction with Products	16
3	Conclusion	17
	References	18

1 INTRODUCTION

The development experience in Virtual Reality is still in its infancy. However, gaming engines have been around for a while and fortunately some of them support SteamVR integration. There are currently multiple powerful and free VR gaming engines on the market to choose from. For our project we need to develop a realistic application that users are able to navigate and interact with a variety of objects and natural elements. When making a decision, the first thing we will be looking at is whether or not the engine is powerful enough to create a VR application. Next will be its integration with SteamVR and how easy it is to learn. We only have six months for our project so we need to make sure we can pick up the tools quickly. The three main aspects of our project are constructing the environment, integrating Columbia products, and create a fishing experience.

Authorship of Sections

- 2.1.1, 2.1.2, and 2.1.3: Sean Marty
- 2.1.4, 2.2.3, and 2.2.4: Jake Jeffreys
- 2.2.1, 2.2.2, and 2.3.4: McKenna Jones
- 2.3.1, 2.3.2, and 2.3.3: Spike Madden

2 PROJECT SECTIONS

2.1 Environment

2.1.1 *Create Static Environment*

To begin any virtual reality development, one must first create a basic static environment to work in. This generally involves a couple major steps, and every legitimate game development platform can do these basic things. That means that in order to research which platform is best, we need to look not just at whether basic environment creation is possible, but how easy and smooth it is. There are many game engines out there, and many facets involved in creating a static environment, but we had to narrow down our research to just three engines and a handful of the most important features. In the simplest sense, there are three steps to creating a static environment in most game engines. First, one needs to be able to explore an empty landscape through a virtual reality headset. In our case, we already have been given an HTC Vive so that will be our headset of choice. Second, one needs to place static objects in the empty landscape. Third, it is important to look at how easy and intuitive it is to work with the camera because that is very important to any virtual reality experience. The three game engines we will look at are the Unity game engine, the Unreal Engine, and Amazon Web Services' Lumberyard game engine.

To create a static environment in Lumberyard, one starts with creating a new level. These levels are the basis for game development in Lumberyard. [1] On creation, a developer is asked to fill out information about heightmaps, terrain texture, and color multipliers. Once a level has been created,

the next step is to populate the environment with static objects. Lumberyard breaks objects into a couple categories. These can include, among other things, brushes and entities. Brushes are objects that the user cannot interact with, while entities have the ability to be interacted with dynamically. [2] Lumberyard handles camera creation, views, and movement like the developer is shooting a cinematic scene (which sometimes they are). A camera is an object itself, and can be selected and moved about in the Track View editor. To move a camera, simply unlock it and use the mouse and keyboard to move it as it records. [2]

If one instead wants to go about creating an environment in the Unity game engine, simply create a new project and a basic empty scene will be generated. Adding objects to the scene can be done through the Object menu, where there is a list of 3D objects to choose from. [3] In order to interact with a basic camera object in Unity, one simply makes a parent object from the basic object class and moves that object, bringing the camera with it. This way, all the same movement and animation processes that apply to normal objects apply to cameras as well. [3]

Finally, if one wants to create a static environment in the Unreal Engine, they create a project and, like Lumberyard, add a new level. There is a specific empty level for virtual reality development, which adds certain basic settings and capabilities automatically. [4] To add static objects to the recently created level, one should add actors. Actors are any objects that can be placed into a level. Specifically, geometry brush actors are the simplest way to add geometry to a level. [4] Cameras are just a type of actor that can be added to a scene in Unreal Engine. Once added, cameras have their own set of attributes and methods that allow interaction.

All three game engines have certain appealing qualities for our project. However, the scene that we are creating is relatively simple in the sense that there will not be significant character movement and most objects won't need to be interacted with. Also, we will most likely only need one scene, so the ability to easily work with multiple levels does not have much appeal. With those constraints in mind, the Unity game engine would be best for simple static object creation because of its simplicity and the universal handling of 3D objects.

2.1.2 Animate Environment

Animating objects and characters in an environment is a key aspect of virtual reality development, just as with any game development. Unresponsive objects or unrealistic animations can negatively effect how real a virtual reality experience feels. Again looking at different game engines, we wanted to figure out which engines are the most intuitive and powerful. We chose two areas to focus on in regards to animation, so that we could narrow our comparisons. The first is linear animation, which is preset, and runs like a movie. User interaction does not effect linear animation. We will use some linear animation for the parts of our scene that the user cannot affect, such as swaying trees or distant water movement. Second, we want to look at interactive animation, which is a set of animations that

are performed based on user input. This is a huge section for virtual reality development, because the basis for a real experience is whether the world feels interactive.

For each engine listed in the table below, we will compare how the engine handles both kinds of animation.

Engine Name	Linear Animation	Interactive Animation
Unity	Animation Curves	Animation Events
Unreal Engine	AnimMontages	EventGraphs
Lumberyard	Track View and LimbIK Technical	AnimationEvent and XML file

When looking to animate objects in Lumberyard, there is the clearest split between linear and interactive animation. For linear animation, developers use the Track View to work through animations frame by frame. When animating people, Lumberyard developers can utilize Limb IK to set an endpoint for a hand or foot and have the engine calculate limb movement that achieves that end goal. [5] For interactive animation, Lumberyard has two categories to handle dynamic animated responses. Avatar controls deal with the character that a user is going to control. This means that movement commands come mostly from outside the game code. AI controls deal with non-user controlled characters. Every movement and decision made by these controls is housed in game code. Both of these are part of the bigger picture for interactive animation in Lumberyard that attempts to implement automatic motion synthesis. [5]

Animation in Unity is based on a collection of clips that can be combined in complex ways, controlled using animation curves, and triggered by animation events. For linear animation, Unity provides a flowchart-style window to manage animation clips and their interaction with each other. [6] Once a set of animations has been combined, Unity developers can use animation curves to connect other game content and parameters with an object in motion. [7] When an object needs to respond to user interaction, animation events come into play. These are simple functions with triggers that set up a specific animated response for an input action. [6]

The Unreal Engine handles animation with a combination of skeletal and vertex morphs, which are controlled through a visual graph of animation events. Linear and interactive animation events are handled in somewhat the same way, by adding interactions between events to the EventGraph. Developers can manage events themselves, their effects on the overall animation blueprint, and interaction between events all in one place. [8]

For our project, the animation needs in the general environment are broken into these two categories equally. We will have linear animation used to make the environment around the user come to life. Also, we will utilize interactive animation both to make close objects react to the user's presence, as well as handle item selection and manipulation by the user. After learning about all three game engines

and how they handle animation, it seems that the Unity game engine fits our needs the best. The other two engines are focused heavily on complex human animation and interaction, and we do not need all that for our game. The Unity engine breaks every animation down into manageable and reusable clips that we can populate our entire environment with.

2.1.3 Add Audio

Another key aspect of creating an immersion virtual reality experience is adding audio effects. A static soundtrack works well for 2D video, but for a virtual reality experience the audio has to be much more complex and dynamic. As Mona Lalwani mentions in her article *For VR to be truly immersive, it needs convincing sound to match; the biggest keys to realistic 3D sound with the technology we currently have are sound cues and three-dimensional sound.* [9] Sound cues are audio events that react to specific triggers. For example, if I move my foot through water that should elicit some sort of sound response. 3D audio can be created in a couple different ways, including sound attention and occlusion. Basically, this just means changing the audio in an experience based on user location and the other objects in the area. Finally, another key to developing audio in a game engine is how sound data is managed. Most engines have various ways to manage their sound data, each with benefits and drawbacks.

We will compare the usual three game engines: Unity, Unreal Engine, and Lumberyard.

Engine Name	Sound Cues	3D Sound	Managing Sound Data
Unity	AudioSource Effects	AudioSource Matrices	Audio Spacializer SDK
Unreal Engine	Visual Sound Cue Nodes	Attenuation Shapes	Audio Node Graph
Lumberyard	Audio Play/Stop Triggers	Raycasting	Audiokinetic Wwise LTX

In Lumberyard, 3D audio is handled by using raycasting through the Audiokinetic Wwise LTX audio system. Attenuation and occlusion are calculated by tracking a sound vector. This means that the path from a sound source to the user is calculated, and objects and distances in between affect the final sound. In order to implement responsive sounds, Lumberyard has Audio PlayTrigger and StopTrigger methods. This allows sounds with all of their properties to be triggered dynamically. [10]

Audio in the Unreal Engine is handled like many other things in the engine, with a node-based visual map. This map allows audio clips to play in any order, and interact with each other in complex ways. Each individual sound is a node on the chart that represents an audio clip bound to a Sound Cue Node. These sounds can then be made realistic by adding attenuation through either a simple distance algorithm or attenuation shapes. Attenuation shapes help create consistent, realistic soundscapes in an environment by calculating attenuation based on a geometric shape that might fit a certain location better than a simple distance calculation. [11]

Unity audio is a system of audio clips that are sent from an Audio Source to an Audio Listener. Both are attached to objects, the latter usually to the main camera object. Audio Sources generate

sounds when audio clips are triggered through AudioSource Effects, allowing for dynamic sound cues. To make those sounds more realistic, Unity uses the Audio Spacializer SDK to handle attenuation and other audio effects like echoing and the Doppler Effect. A system of matrices handles how the listener and source handle complex audio, although the spacializer performs good portions of the calculations. [12]

In the end, our choice in game engine for audio comes down to choosing a simple system and choosing a system that will most help bring realism to the soundscape of our environment. This is a hard balance, but it seems like the Unreal Engine finds the best middle ground. In the Unreal Engine there is no added outside software to handle complex sound manipulation. Also, the attenuation shapes feature of the Unreal Engine might be extremely useful to relatively quickly create a real sounding landscape.

2.1.4 Improve Realism and Immersion

The goal of our project is create a realistic, virtual, outdoor experience that makes users feel like they have been transported to a different location. After we've created the environment, animated objects, and added audio, the next step is to improve the visual realism.

The first thing that came to our mind was to increase the detail of the objects in the environment but this may actually make the experience less realistic from the perspective of the user. It turns out that detail is only one aspect of creating a visually successful game. The other two are frames per second and resolution. These three concepts make up what is called the graphical fidelity triangle[13]. According to a study done by Intel and Thug[14], immersion needs graphical fidelity, not realism. They found that what was important was that graphics were crisp and clean at all times. They found that a smooth experience, one without glitches and lost frames, was the most important aspect of immersion. They even went a step further to argue that photo-realism is often times worse because it makes inaccuracies more obvious.

In order to find out which of these tools will be the most effective we will be looking at simple particle systems, texture libraries, and clean texture mapping. It's important that which game engine we use that the asset store contains a wide variety of simple textures and materials so that we are able to create realistic environment objects. There are also going to be a lot of moving parts such as animals and river water. These need to be realistic enough to create immersion but not too detailed as to create graphical lag. The three tools I will be looking at are Unity, Unreal Engine, and Lumberyard.

Engine Name	Particle Systems	Texture Libraries	Clean Texture Mapping
Unity	Yes	Yes	Yes
Unreal Engine	Yes	Yes	Yes
Lumberyard	Yes	Minimal	Yes

Unity is an incredibly common tool for building in virtual reality, especially for the HTC Vive. It has managed to find a good balance of realism throughout the asset library[15]. This asset store offers a wide range of particle effects, textures, and materials. Within the Unity development environment, they have made it easy to map texture on to objects to create a truly crisp experience. They also offer a variety of lighting options to add even more outdoor realism. Unreal Engine also offers a wide range of particle animations and texture packages[4]. Unreal engine looks like it would be a much more effective tool for creating high end virtual reality games but for our usage it may be overkill. Overall it has a lot of the same capabilities if not more but they also come at a price. Most of the asset store costs money which is not what we are looking for. Lumberyard has a good variety of particle effects but struggles when it comes to textures and texture mapping[1]. Developer freedom within the environment is easy to learn but limited.

The best tool to create an immerse, realistic environment would be Unity. It offers a wide variety of free assets that can be used throughout projects to add that extra bit of realism. The Unity community also strives to create simple, crisp designs that dont have unnecessary details. They have recognized that these details can actual hinder the immersive experience instead of help it.

2.2 Fishing Activity

2.2.1 Process User Wand Movement

The first step to creating an immersive fishing experience will be collecting the users's movement from the virtual reality controllers. Since this is an essential component of the project, we will be examining how three different gaming engines go about doing this. For this specific project we will be using the HTC Vive so we are concerned with capturing the input from the two HTC Vive Wands. This is a core component of our system that other parts of the product will rely on so it should be simple and reliable at the same time.

We will consider three different gaming engines to accomplish this task, Unity, Unreal Engine, and Amazon's Lumberyard gaming engine. Two different aspects of each engine will be compared in order to reach a final verdict on which is the most useful for the task. First there is the consideration of which language, and tools are used to complete the task. Second, since this is a core function of our system we will consider if the gaming engine has native virtual reality support or not.

Engine Name	Language	Native VR Support
Unity	C#	Yes
Unreal Engine	Blueprint Visual	Yes
Lumberyard	Lua	No

First we will consider the Unity engine. Scripting in Unity is primarily done in C#. Most of the team working on this project has experience in C#, so that is an easy benefit of using Unity. Even for members of the team who do not have experience, C# is a fairly easy language to pick up. In order to handle controller input in Unity one can make use of built in SteamVR calls. For example there are SteamVR calls to access the controller itself, as well as various buttons on the controller [16]. This abstracts the process of retrieving controller input greatly, which is a good thing in this case. The generally accepted approach to capture the controller input in Unity is to create a controller class that makes use of these calls. Unity is officially supported for development for the Vive. Therefore much documentation exists on the subject of capturing controller movement.

The next engine to consider for this task is the Unreal Engine. Unlike Unity, Unreal does not make use of a traditional programming language. Instead, as a developer you use Blueprint Visual. Similar to C#, it is used to define object oriented classes. The only difference is that it is done visually instead of with code. The process for working with Motion Controllers in Unreal is much different than Unity. The developer picks from a list of Motion Controller-specific inputs in the Pallete panel of the Blueprint Editor[17]. From here you can simply drag and drop to attach a certain controller action to an action within the game. This simplifies the process greatly, but also has the drawback of not being as customizable as if we were writing this in code. The Unreal Engine has support for SteamVR, and therefore the Vive. That being said, the documentation available for VR related topics is not plentiful as with Unity.

The final option for this task is Amazon's Lumberyard Engine. What makes Lumberyard unique is that you can choose to use the Flow Graph System for visual scripting on the Lua language for code based scripting. As far as capturing controller input, it is possible with both. Using the Flow Graph one can create a VR:ControllerTracking node which provides up to date info regarding the controller's current position and status [18]. Alternatively, one can use Lua to access the TrackingState struct which contains the linear velocity, acceleration, and all other tracking info of each controller [19]. Lumberyard has this benefit of being flexible in the implementation. However, Lumberyard's VR support is currently still in beta and the documentation is lacking.

After considering the above options, Unity stands out as the best option to capture the controller movement. First of all, the language, C#, that Unity uses for scripting is ideal. Most of the team members are familiar with the language so there will be no time wasted learning a new language. Also, compared to Blueprint Visual and the Flow Graph of Unreal and Lumberyard, respectively, scripting in C#, offers more flexibility for capturing controller movement. Unity has the best support for the HTC Vive. This is important for the project specifically because as it will be in a retail setting, it needs to be as reliable as possible. Finally, Unity is the best option in terms of documentation which is important for development.

2.2.2 Import Fishing Assets

As none of the members of this team are artists, the project will rely heavily on downloadable assets. These assets will be used to create nearly all aspects of the virtual reality environment. Therefore the assets used play an essential role in creating the most realistic environment possible. A wide variety of assets will be used to create the environment. There is not a defined list of different medias that will be used but a few likely ones are still images (textures), animations, and audio.

As in other sections, the ability to handle assets in three different gaming engines, Unity, Unreal, and Amazon's Lumberyard engine will be compared. Three main categories will be considered in detail. Firstly, the access to a native asset store. This is preferable to searching for assets elsewhere. Secondly, the types of files that are supported for import. Since we are not certain of which assets we would like to use, the engine which has the widest support for file types is preferred here. Finally, we will examine each engine's method of organizing and importing assets. This project will make use of many assets so an organized system is crucial.

Engine Name	Asset Store	File Support	Asset Organization and Import
Unity	Yes	Images, 3D models, Animations, Audio	Robust
Unreal Engine	Yes(new)	Images, 3D models, Animations, Audio	Supported
Lumberyard	No	FBX files	Minimal

First we will consider Unity. One of Unity's selling points is its Asset Store. The Unity Asset store launched in 2010, making the oldest and most mature of the three gaming engines in discussion [15]. The asset store currently has over 15000 free and paid 3D assets to choose from. While browsing the store it becomes apparent that there has been an asset created for just about everything that the mind can imagine. Importing an asset into Unity from outside of the asset store is as simple as drag and dropping the asset into the Unity project window. Assets are organized in Unity in the Assets folder of the Project windows. Within the Assets folder assets are organized into subgroups of materials, textures, etc. This makes finding assets a painless process. Unity supports all major types of Images, 3D models, Animations, and Audio files.

Second we will consider the Unreal Engine. Like Unity, the Unreal engine also has an asset store, called the Unreal Engine 4 Marketplace. The Marketplace, however, is much younger than the Asset Store of Unity. The Marketplace opened to developers during 2014 [20]. This means that number of available assets is significantly smaller than the Unity Asset Store. As far as importing assets, the process in Unreal is straightforward, and guided by a GUI. Organizing assets in Unreal is a bit more manual than Unity. One must manual manage the folders where assets are stored, and there is not an assets folder by default. Unreal supports all major types of Images, 3D models, animations, and Audio

files. The preferred file format for importing assets FBX files, which is slightly limiting, as not all assets can be found in this format.

Finally we have Amazon's Lumberyard Engine. Unlike Unity and Unreal, Lumberyard does not have a place to download assets from within the engine. This means that you must look to other sources for assets, such as Unity's Asset Store, Unreal's Marketplace, or other websites. Like other parts of Lumberyard, the Asset Importer is still in preview release [?]. Currently it only supports FBX by default. If you would like to import other files you need to manually implement a new importer that will generate a SceneGraph for that particular file type [?]. Importing files requires navigating the installation location of Lumberyard and manually copying and pasting the files to the correct location.

After considering the three engines, Unity seems like the clear choice for importing and managing the assets needed to create our Virtual Reality environment. While both Unity and Unreal have places to download assets, Unity's Asset Store has the most assets by a long shot. Unity also has the benefit of having simple workflow for importing assets compared to Lumberyard. Unity has native support for importing many different file types, which is beneficial as it is not clear which assets we will be using yet. Finally, the process of importing and managing assets is organized and stable, which cannot be said about Lumberyard.

2.2.3 Fishing Rod Interaction and Mechanics

Within the outdoor virtual reality experience there will be the opportunity for users to go fly fishing in the virtual river. In order to achieve this capability, we will need to allow the user to first interact with a fishing rod. The mechanics of this process can get quite complicated and therefore it is important for us to decide on the correct tool to build this functionality. Not only will the user need to be able to pick up the fishing rod but they will also need the ability to cast and reel the line back in. These are the basic functionality of the fishing rod and will need to be as realistic as possible to create the illusion they are actually participating in the activity. In this document I will be comparing the virtual object mechanics within different, free gaming engines: Unity3D, Unreal Engine, and Lumberyard.

In order to find out which of these tools will be the most effective we will be looking at the ease of scripting mechanics and programming haptic feedback. Fly fishing is all about the smooth motion of the rod. In order to create a similar experience in virtual reality there needs to be some kind of haptic feedback (controller vibration). Fortunately, my team has been given an HTC Vive setup which comes with two wireless controllers. These controllers offer HD haptic feedback with 24 sensors to ensure accurate movement tracking[21]. This then brings up the question of software.

The first engine to discuss is Unity which is one of the most common tools for beginners developing virtual reality applications. Unity tools related to the physics engine and haptic feedback are incredibly

Engine Name	Language	Physics Engine	Haptic Scripting	Documentation
Unity	C#	Yes	Yes	Yes
Unreal Engine	Blueprint Visual	Yes	Yes	Yes
Lumberyard	Lua	Minimal	Minimal	Yes

easy to use for developers familiar with C#. There are extensive built-in libraries and a very intuitive structure[3]. Documentation and support is also strong for Unity programming which explains why Unity is a great choice for people new to physics engines and virtual object mechanics. The next tool to discuss is Unreal Engine which uses Blueprint Visual. This is an incredibly powerful engine and therefore offers an extensive physics engine[4]. The Blueprint Visual interface for developers makes development easy and visually clear. This could make the process of implementing physics simple and straightforward. Haptic feedback is implemented in a similar fashion with check boxes and drop down menus instead of having to write a single line of code. This would be great for those unfamiliar with programming fundamentals. The last engine is Lumberyard which uses the Lua scripting language. The physics engine and haptic feedback programming are still in their infancy but do offer some capabilities[2]. For a virtual reality project with a lot of moving pieces, the simplicity of lumberyard may be a hindrance and not offer enough freedom to developers.

For our project, the best tool to use is going to be easy to learn yet still have extensive physics functionality to give us enough freedom while developing. Based purely on the ease of use and capabilities of the physics engine and haptic feedback control, the best tool would be Unity. The main reason for this is that Unity is easier to get started on and given the timeframe of our project, it is important we are able to create a fundamental application as quickly as possible. The physics engine is clean and should give us enough freedom to create a realistic fly fishing experience.

2.2.4 Integrate Usage with Environment

Creating a realistic fishing rod that users are able to pick up and move around is one thing but to integrate these movements with the environment is incredibly important to creating a realistic experience. According to user studies done by Intel and Thug[14], realistic interaction is the most important heuristic when it comes to overall enjoyment and feeling of immersion with correlation coefficients of .49 and .57 respectively (1.0 is a perfect correlation). If these are the two standards we are looking at, in order to have a successful application we will need to allow users to easily interact with all aspects of the fishing environment. While fly fishing, people stand either in the water or on the bank. These locations have specific characteristics such as certain insects, fish, plants, water movement, and sounds. The user will then need to be able to interact with these objects as well as the other way around. In this document I will be comparing the virtual object mechanics within different, free gaming engines: Unity3D, Unreal Engine, and Lumberyard.

In order to find out which of these tools will be the most effective we will be looking at the ease of importing animals, of triggering sounds, and of animating these objects. The animation is the most important as it needs to not only give the illusion of realism but also react to user movements. For example, if the user steps into the water then not only will sounds need to occur but also certain fish animations may need to get triggered such as swimming away from the user. Sounds have an incredible power of creating immersion and therefore need to be comprehensive yet subtle. Subtlety is a big part of immersion as users should never feel overwhelmed by noises or animations.

Engine Name	Animal Assets	Animation Assets	Sound Assets	Object Assets
Unity	Yes	Yes	Yes	Yes
Unreal Engine	Yes(Paid)	Yes(Paid)	Yes(Paid)	Yes(Paid)
Lumberyard	No	Minimal	Minimal	Minimal

One of the biggest areas to look at here is the availability of free assets. The scope of this project is quite small so it is important to join a game engine community that supports this. Upon looking at Unity we found that there is a wide range of support for creating interactive games[3]. Objects are easily importable and interaction is easily programmable. Objects interact well and are able to demonstrate accurate collision mechanics. There is also a lot of flexibility when it comes to detecting user location and movements. Unreal engine is equally as powerful[8] but it doesn't provide nearly as many free assets. Realistic interactions require a lot of subtle objects which won't be possible in our time frame if we have to create everything from scratch. Lumberyard is an incredibly simple piece of software and offers very little when it comes generating realistic user interactions[2].

The amount of support behind Unity makes this tool much more effective for our needs. The community is made up of more enthusiasts and hobbyists which generates more free, quality assets. The flexibility and simplicity when programming user interaction will also give us a lot of freedom and the ability to create rapid activity prototypes.

2.3 Columbia Products

2.3.1 Create Avatars

Animated characters, or avatars, are going to be essential in building an immersive and realistic virtual reality environment for the Columbia/Intel Outdoor Gear project. Avatars are going to be especially useful in this project as they give us a means of displaying various Columbia gear to the VR user. The user themselves will be presented as an avatar wearing Columbia gear, and they'll also be able to see other avatars wearing Columbia gear. This variation of perspective will help the consumer get a better feel and understanding of the various gear choices available to them.

Three common game engines that are compatible with the HTC Vive are Valve's Unity game engine, Epic Games' Unreal Engine 4 and Amazon's Lumberyard. This section will examine each game engine's capabilities when it comes to avatars and character assets. All three game engines have support for character models and are all very similar in their implementation; they require a rigged and skinned model from a third party digital content creation package for character setup. The exported model is a collection of polygons in a 3D package. The process of rigging defines the skeleton and joint hierarchy of the model and is what determines the range of possible movements. Skinning is the process of connecting skin or mesh to the previously defined joint hierarchy[22]. Examples of these DCC packages include Autodesk 3ds Max or Maya, Poser, Makehuman, and Mixamo.

Engine Name	Asset Store Quality and Variety	Import assets from DCC
Unity	Robust	Yes
Unreal Engine 4	Moderate	Yes
Lumberyard	Does not exist	Yes

Unity's animation system, known as Mecanim, handles all humanoid creation and animation. In addition to being able to import models created by third party DCC packages, Unity also allows for downloading of character content from the Unity Asset Store. The Unity Asset Store has a wide variety of both free and purchasable character models. At the time of writing this document, the Asset Store has 3885 3D character models available for use. These include animals, humanoids and robots. Unity also supplies a service to create an animated humanoid character from scratch through the process of modelling, rigging and skinning within the client. The best case scenario for this project would be for Columbia to already have these complete models for their products, and to have them be easily imported into the Unity environment.

Unreal Engine 4 also supports the importing of art assets and animations from third party DCC packages[23]. It also has a Marketplace where developers can download various character assets for use in their environments. The selection is limited compared to Unity's Asset Store, and a lot of the models are paid; there are very few if any free ones.

Lumberyard also supports the importing of art assets and skeletal meshes from third party DCC packages[24]. There are several asset packages on their website but no designated asset store of any kind.

In conclusion, our team will likely be using Valve's Unity Game Engine. All three of the game engines allow for importing of assets from third party DCC packages, but the quality and quantity of available 3D avatars in Unity's Asset Store is unrivaled.

2.3.2 Import Columbia Gear

Arguably, the central focus of this outdoor virtual reality experience is to promote Columbia gear and lead the customer to a sale of a product. With that in mind, it's crucial to have Columbia gear presented as realistically and prominently as possible within the virtual environment. Columbia gear includes various apparel items such as coats, jackets, boots and other products from their fishing line.

Once again, the three game engines that will be compared are Unity, Unreal Engine 4 and Lumberyard. Each of these platforms will be evaluated on their asset stores, supported import file formats and ease of use. While it is unlikely that any of these platforms' asset stores have accurate and high quality models of Columbia gear, a strong selection of similar assets could be helpful in testing and development. This leads to the second point of supported file formats. The most favorable situation, and most likely, would be for Columbia to provide us with assets and animations of their products. In this situation, we would need the game engine to import external files; the game engine with the most supported file formats would be preferable. Lastly, since these assets are such a big part of the virtual reality experience, it's vital for us as the developers to be able to interact with these assets as easily as possible.

Engine Name	Asset Store Quality and Variety	File Support	User Friendly
Unity	Robust	3D models and animations	Yes
Unreal Engine 4	Moderate	3D models and animations	Yes
Lumberyard	Does not exist	FBX	No

Unity's Asset Store has over 15000 free and purchasable 3D assets available to users. The simple 5 star rating system of the store allows developers, at a quick glance, to find reputable asset packages that will fit any project's needs. If nothing in the store stands out, which may be the case with Columbia gear, Unity supports all major types of 3D models and animations. Models and animations can be created in third party digital content creation packages and imported into a Unity project. Unity also allows for the creation and animation of assets, within the client, if no suitable assets can be found anywhere. Asset access within the project is simple, and is organized clearly in an Assets folder in the Project window. Specific components within the Assets folder can also be browsed.

Unreal Engine 4 also has an asset store, named the Marketplace, but is not as diverse as Unity's Asset Store. The Marketplace is much newer[20] as Unity's Asset Store came out 4 years earlier, and as a result, has not built up the collection of quality and quantity that Unity offers. It also seems like a lot of the asset packages are paid; there are very few, quality, free asset packages. But once again, it is highly unlikely that any of the premade assets in the Marketplace will suffice for this project that wants to present Columbia gear as accurately as possible. Unreal Engine 4 offers support for all major file types as well, and the organization of assets within the menus is reasonable.

Amazon's Lumberyard is still in beta and is lacking a lot of features as expected. It does not have a native asset store and only supports FBX files by default. The Asset Importer requires the user to manually find the path of the project and paste the required files[25].

Since Columbia gear is a key part of this project, we want to be able to work with a gaming engine that allows us to effortlessly work with these assets. Due to Unity's large selection with the Asset Store, ability to support a majority of popular file formats and ease of use within the client, Unity seems like the best choice over Unreal Engine 4 and Lumberyard.

2.3.3 Animate Avatars

Proper animation of the avatars and environmental components in the virtual reality space is going to be key in creating a believable, immersive fly fishing experience. We need a game engine that is capable of animating the avatars, Columbia gear, and outdoor environment. As per our requirements document, the experience must maintain at least 60 frames per second.

Engine Name	Animation Assets from Store	Import Animations from DCC packages
Unity	Robust quantity and free	Yes
Unreal Engine 4	Moderate quantity and paid	Yes
Lumberyard	Does not exist	Limited

Unity's animation system, Mecanim, includes services that allow for retargetable animations, full control of animation weights at runtime, event calling from within the animation playback, sophisticated state machine hierarchies and transitions, and blending of shapes for facial animations[6]. In particular, retargetable animations for humanoid animations is going to extremely useful in this project. With retargetable animations, it is possible to apply the same set of animations on various character models using the humanoid models' bone structure. The animation workflow in its entirety is discussed in the next section.

Mecanim uses Animation Clips which store information on how objects should adjust their position and other properties over time. These clips can come from third party digital content creation packages, motion capture studios or can be created within Unity. Animation Clips can be considered the building blocks for all animation sequences in Unity. These building blocks are organized into Mecanim's Animator Controller to chain various animations together. The Controller keeps track of the Animation Clips in a flowchart system and behaves like a state machine that keeps track of current Animation Clips and when they should change. The Controller is also capable of blending multiple clips to provide smooth transitions between animations. The Animation Clips and Animator Controller are aggregated into a GameObject through the Animator Component[6].

Unreal Engine 4 also has a complete animation system and is similar in structure to that of Unity's. Unreal Engine 4's animation editing toolset, Persona, allows for editing of Skeletons and their

meshes, Animation Sequences and Animation Blueprints. Animation Sequences are similar to Unity's Animation Clips and is defined as a single animation asset that is associated with a Skeletal Mesh. Blueprints in Unreal Engine 4 are a gameplay scripting system. Object oriented classes are defined as nodes and developers can define interactions between the nodes[8]. The Animation Blueprints specifies the movement of bones and also allows for blending of animation. Unreal Engine 4 also has a Controller that moves or alters an avatar's properties due to a trigger.

Lumberyard's documentation on animation is limited, which is expected as it's still in beta. Lumberyard allows for the creation of both linear animation and interactive animations. Linear animations are predefined animations that do not interact with any outside influences. Interactive animations are influenced by the environment and are responsive[5].

Both Unity and Unreal Engine 4 have animation systems that are capable of handling the task of creating realistic animations that will help to build an immersive virtual reality experience. Both Unity and Unreal Engine 4 have very similar technologies that implement the animation workflow. Since Lumberyard is still in beta, Unity and Unreal Engine 4 seem like suitable choices for this project when it comes to animation.

2.3.4 Allow User Interaction with Products

In order for the user to consume information related to the Columbia Sportswear gear it will need to be displayed in some fashion within the VR environment. This will likely be done with a mixture of menus and dialogs. Different gaming engines go about doing this in different ways. The following section will compare support for different menu and dialog systems in Unity, Unreal, and Amazon's Lumberyard Engine.

There are a couple things to consider when creating menus and dialogs in VR. In traditional games, user interfaces are generally non-diegetic, meaning that it does not actually exist in the world. Think of a traditional heads-up display. Unfortunately this approach does not work well in Virtual Reality. It has the possibility of causing nausea and straining the user's eyes. Two other alternatives are diegetic UI and spatial UI. Diegetic UI elements are attached to something within the environment. Spatial UI elements are positioned in the environment. Another thing to consider is how the user will interact with the UI elements. In our case, using the Vive, it is possible for the user to interact with the UI either with the controllers or the headset. Therefore, we will consider which different types of UI and user interaction each gaming engine supports.

Engine Name	Supported types of UI	Supported types of user interaction
Unity	Diegetic, Spatial	Headset and Controller interaction
Unreal Engine	Widgets	Controller and Headset interaction
Lumberyard	No native VR user interfaces	No support

The first gaming engine to consider is Unity. Unity has very good support for VR related user interfaces. The main tool for creating VR user interfaces in Unity is the World Space UI. This UI system allows the developer to create UI elements that are either spatial or diegetic. The World Space UI allows you to create a virtual canvas where they can then place GUI interfaces [26]. As far as user interaction with user interfaces, Unity has virtual reality support for this built in and well documented. There are three main classes that can be used for interaction: VREyeRaycaster, VRInput, and VRInteractiveItem [27]. VREyeRaycaster casts a virtual ray in the environment and determines if it hits any colliders. This could be used for either the headset or the controllers. VRInput is a class that retrieves the button presses from the controllers of the VR system. Lastly, you can use VRInteractiveItem with any GameObject that you would like to be able to interact with in the VR environment.

The second gaming engine to consider is Unreal. Unreal does not have an default way to create VR user interfaces. However, the most commonly accepted way is to use Unreal's Motion Graphics UI Designer. This tool can be used to create 3D widgets. While these widgets are not necessarily designed to be used as menus or dialogs, it is possible to do so, but it is a bit of a workaround. It is possible to interact with these widgets with both the headset and the controllers, however, it is no easy task. Unreal does not have any classes or libraries to make this process easier. The developer will need to implement the user interaction from the ground up, using the Blueprint Editor.

The last engine to consider for this task is Lumberyard. As Lumberyard is still in beta, it is lacking when it comes to both user interfaces and user interaction with said interfaces. Lumberyard contains a UI editor which can be used to create most 2D interfaces. Interfaces that resemble heads up displays can easily be created in Lumberyard, but as discussed above this is not useful for VR. The other problem is that it has no support for 3D interfaces, and no native support for VR specific user interfaces. Because of this, Lumberyard also has no support for interaction with user interfaces within the VR environment.

After considering these three engines for the task of interacting with information regarding Columbia Sportswear gear found in the environment, Unity seems like a good pick for the issue. First of all in Unity, it is easy to create both diegetic and spatial interfaces using the World Space UI. On the other hand, neither Unreal or Lumberyard contain native support for VR specific user interfaces. Secondly, Unity has a set of classes which can be used to simplify the process of creating an interactive user interface. This process is complex in Unreal, and non existent in Lumberyard. Finally, Unity supports interaction with both the headset and the controllers. These factors make it a clear choice for displaying the information related to the Columbia Sportswear gear.

3 CONCLUSION

After considering the three different gaming engines discussed in this document, the best pick is clearly Unity, for a number of reasons. The fact that we are working with the HTC Vive VR system greatly

influences this decision. Unity is the accepted game engine for SteamVR development and therefore the Vive as well. Unity is also the engine that has built in support for various VR related functions, whereas in Unreal and Lumberyard, VR support is minimal and cumbersome. Finally, there is the added benefit of Unity using C# as its scripting language, a language that our team is experienced in. Overall, Unity will allow us to hit the ground running and develop our project as efficiently as possible.

REFERENCES

- [1] A. W. Services, "Lumberyard levels and environment," 2016. [Online]. Available: <http://docs.aws.amazon.com/lumberyard/latest/userguide/level-intro.html>
- [2] ——, "Lumberyard object and entity system," 2016. [Online]. Available: <http://docs.aws.amazon.com/lumberyard/latest/userguide/entities-intro.html>
- [3] U. Technologies, "Unity getting started with vr development," 2016. [Online]. Available: <https://unity3d.com/learn/tutorials/topics/virtual-reality/getting-started-vr-development>
- [4] E. Games, "Unreal engine editor manual," 2016. [Online]. Available: <https://docs.unrealengine.com/latest/INT/Engine/Editor/index.html>
- [5] A. W. Services, "Lumberyard animation," 2016. [Online]. Available: <http://docs.aws.amazon.com/lumberyard/latest/developerguide/animation-intro.html>
- [6] U. Technologies, "Unity animation," 2016. [Online]. Available: <https://docs.unity3d.com/Manual/AnimationSection.html>
- [7] ——, "Unity animation tutorial," 2016. [Online]. Available: <https://unity3d.com/learn/tutorials/topics/animation>
- [8] E. Games, "Unreal engine skeletal mesh animation," 2016. [Online]. Available: <https://docs.unrealengine.com/latest/INT/Engine/Animation/index.html>
- [9] M. Lalwani, "For vr to be truly immersive, it needs convincing sound to match," January 2016. [Online]. Available: <https://www.engadget.com/2016/01/22/vr-needs-3d-audio/>
- [10] A. W. Services, "Lumberyard audio system," 2016. [Online]. Available: <http://docs.aws.amazon.com/lumberyard/latest/userguide/audio-intro.html>
- [11] E. Games, "Unreal engine audio and sound," 2016. [Online]. Available: <https://docs.unrealengine.com/latest/INT/Engine/Audio/index.html>
- [12] U. Technologies, "Unity audio," 2016. [Online]. Available: <https://docs.unity3d.com/Manual/Audio.html>
- [13] T. Wilde, "Pc graphics options explained," Oct 2014. [Online]. Available: <http://www.pcgamer.com/pc-graphics-options-explained/>
- [14] S. Michalak and E. Lind, "Virtual reality heuristics, results from user testing for prioritization and development," Sep 2016.
- [15] W. Freeman, "Unity launches unity asset store," November 2010. [Online]. Available: <http://www.develop-online.net/news/unity-launches-unity-asset-store/0108154>
- [16] R. Reddy, "Steamvr: Handling vive controller input in unity," 2016. [Online]. Available: <http://www.leadingones.com/articles/vive-dev-unity-4.html>
- [17] I. Epic Games, "Motion controller component setup." [Online]. Available: <https://docs.unrealengine.com/latest/INT/Platforms/VR/MotionController/>
- [18] A. W. Services, "Setting up a basic virtual reality flow graph," 2016. [Online]. Available: <http://docs.aws.amazon.com/lumberyard/latest/userguide/virtual-reality-flowgraph-setup.html>
- [19] ——, "Vr lua functions," 2016. [Online]. Available: <http://docs.aws.amazon.com/lumberyard/latest/developerguide/lua-scripting-ref-vr.html>

- [20] D. Water, "Marketplace: Now open for business!" 2014. [Online]. Available: <https://www.unrealengine.com/blog/marketplace-now-open-for-business>
- [21] H. Corporation, "Buy vive hardware." [Online]. Available: <http://www.vive.com/us/product/>
- [22] U. Technologies, "Using humanoid characters," 2016. [Online]. Available: <https://docs.unity3d.com/Manual/UsingHumanoidChars.html>
- [23] E. Games, "Setting up a characters," 2016. [Online]. Available: <https://docs.unrealengine.com/latest/INT/Engine/Animation/CharacterSetupOverview/>
- [24] A. W. Services, "Setting up a characters," 2016. [Online]. Available: <https://docs.unrealengine.com/latest/INT/Engine/Animation/CharacterSetupOverview/>
- [25] —, "Asset importer (preview) technical overview," 2016. [Online]. Available: <http://docs.aws.amazon.com/lumberyard/latest/developerguide/asset-importer-intro.html>
- [26] U. Technologies, "Creating a world space ui," 2016. [Online]. Available: <https://docs.unity3d.com/Manual/HOWTO-UIWorldSpace.html>
- [27] —, "Interaction in vr," 2016. [Online]. Available: <https://unity3d.com/learn/tutorials/topics/virtual-reality/interaction-vr>

4.2 Tech Review Changes

The changes that we made to the Technology Review document were very similar to the changes we made to other documents. The first thing we changed was removing the presence of Columbia Gear within the experience. We also transitioned from fly fishing to normal fishing. Finally, we took into consideration the VR based physic libraries that each of the three gaming engines have to offer.

5 WEEKLY BLOG POSTS

5.1 Fall Week 3

5.1.1 Jake

Week's Progress

I spent the week putting together a Problem Statement which aimed to define the problem and make sure that everyone on the team and the sponsors are all on the same page. To help us write it we spoke with Mike and Tim over the phone and discussed their visions and the scope of the project.

Problems Encountered

Only problem we've faced so far is finding time that we are all free to meet and discuss the project with the sponsors. We have managed to find time but we'll need to make a big effort to be efficient we meet up.

Plans for the Coming Week

Next week I see the team starting to research the programs we'll be using and looking into getting our hands on a an HTC Vive.

5.1.2 McKenna

Week's Progress

This week we spent most of our time working on our problem statement. Overall this was a very smooth process. In order to complete this assignment we had a conference call with both of our sponsors: Mike Premi from Intel and Tim Delvin from Columbia. Our conversation with them served as the first meeting with our entire team. Before the meeting we only had a vague idea of what the project would entail. After the meeting the entire team was finally on the same page about the general details of the project. I would call this first meeting a huge success.

Problems Encountered

This week we did encounter any major road blocks. The main thing that held our team up was finding a time that was suitable for all of us to meet. With our of our busy schedules this was easier said than done.

Plans for the Coming Week

For this coming week we will continue planning our project. This will most likely be in the form of completing upcoming assignments for class, and talking with our sponsors more. The other thing we will most likely do is dive into VR and the Unity Engine. Currently another team which is working on a VR project with Intel has the HTC Vive and some other VR gear. Our team needs to coordinate a system for sharing this gear within the coming week.

5.1.3 Sean

Week's Progress

This week I worked with the team to formulate our Problem Statement. This included meeting up as a team, and having a conference call with our sponsors Mike Premi and Tim Devlin (referred to in the future as just Mike and Tim). We had to distill a bunch of information and big picture goals down into about a page of content explaining what we want to do.

Problems Encountered

No true problems faced this week. A hard part of writing the Problem Statement was keeping our goals and deliverables vague enough to allow for the creativity and variability that is inherent in the project while also having something concrete to strive for.

Plans for the Coming Week

The biggest tasks I see coming this next week are to continue planning and following the process outlined by our capstone assignments, and learning about the Unity gaming engine. Ramping up on Unity will help ensure I can hit the ground running once true development starts.

5.1.4 Spike

Week's Progress

Our team worked on the Problem Statement for our project which included an abstract, problem definition, proposed solution and performance metrics. To complete this week's assignment, the team had a conference call with Mike Premi from Intel and Tim Devlin from Columbia. We went over the vision and the goals of the project and discussed strategies on how to move forward.

Problems Encountered

We were unsure about the expected deliverables for this project but we were able to clear up our concerns during the conference call. Other than that, we didn't really have any large issues to deal with.

Plans for the Coming Week

We'll continue to brainstorm ideas for the project and work on the expected milestones for the class. The biggest technical short term goal for the team is to ramp on the Unity Engine so we're all familiar with its capabilities. We'll also have to contact the other Intel sponsored groups to work out times we can share the HTC Vive so we can do our own testing.

5.2 Fall Week 4

5.2.1 Jake

Week's Progress

I spent some time this week researching the software needed to create a Virtual Reality application. After finding the correct tools I looked into the hardware required to run these programs.

Problems Encountered

I currently work on a MacBook Pro from early 2011 which has the following stats:

- 2.3 GHz Intel Core i5 - Intel HD Graphics 3000 512 MB - 8GB RAM

In the past I have had issues with Hd video editing but so far haven't had any major issues with Unity. I have experienced some lag so I will continue to keep an eye on it. Fortunately I will be investing in a new Computer with upgraded specs within the next month.

Plans for the Coming Week

Next week I will be working on the Requirements document and I will try to get in contact with Erik Watterson (a student on one of the other Intel VR Projects) to discuss usage of the HTC Vive gear.

5.2.2 McKenna

Week's Progress

The first thing I worked on this week was revising our problem statement. Luckily our problem statement was used as an example during class so we received some very valuable feedback from Kirsten. The main issue was our Problem Definition. We incorrectly thought that our problem was to create a VR experience. However, the root problem is to sell more Columbia Gear, which will be solved with our VR experience.

The other thing I did was work on getting the Unity Game Engine and Editor set up on my machine. For the most part this was a smooth process, and I have started to go though some basic tutorials to acquaint myself with the software.

Problems Encountered

The small problem I ran into this week was getting Unity on my laptop. I run Linux as my primary operating system, however, the Unity software is only officially supported by MacOS and Windows. I have both Windows and Linux installed on my machine so I was originally planning on having to dual boot to Windows every time I would want to do Unity development. Luckily, after a bit of searching I found a Unity Beta that had been released for Linux. It is the same version number as the official release so I should be fine developing on Linux. That being said, I am still unsure if my laptop is up to the task as it has no dedicated graphics card.

Plans for the Coming Week

For the coming week we plan to keep revising our problem statement. We will then also meet with our sponsors and discuss the changes we made.

5.2.3 Sean

Week's Progress

This week we worked to make edits to our Problem Statement. Our document was used as an example in class, so we got some extra pointed feedback to go along with the feedback from Kevin. Our biggest issue is a clear concise definition of what part of our problem statement actually needs to be our main "problem". This mostly manifests as an ordering issue. We have most all the info we need there, just not put together correctly.

Problems Encountered

No major problems faced this week other than being busy. We need to edit our Problem Statement, but that was everyone in the class so I do not feel as though that is a major issue.

Plans for the Coming Week

Two big tasks for the coming week. First, work to nail down requirements. This keeps being stressed as super important for future success on the project so I don't want to put too little effort into it. Second, I want to get familiarized with Unity game engine because that is the main area that we will be developing in once that starts up.

5.2.4 Spike

Week's Progress

We revised our Problem Statement slightly to accommodate for the changes that Kirsten suggested during class. I also downloaded the Unity Game Engine and messed around with some of the tools. It seems to run well which is really good.

Problems Encountered

I didn't really run into any problems this week since the only real task we had was to try to get Unity on our laptops. I didn't try any graphic intensive simulations so my GPU might become a problem, but so far it looks ok.

Plans for the Coming Week

We need to finish revising the Problem Statement so we can resubmit it for credit. We'll have to email our sponsors to outline the changes to make sure we're all on the same page. We should also continue to explore Unity.

5.3 Fall Week 5

5.3.1 Jake

Week's Progress

I met with Erik Watterson to learn about his project and to understand how to set up the HTC Vive. We also discussed the logistics of sharing the equipment. My team and I also created a rough draft of the requirements documentation for the project.

Problems Encountered

Erik and I came to the conclusion that it would be difficult for our two teams to share the same equipment for the entire project. The room the other team is using is also quite small so my team will probably need to find another location to set up our equipment.

Plans for the Coming Week

Next week I will work to create a final draft of the requirements document. I will also be looking for a location on campus my team can use to store our equipment once we get it.

5.3.2 McKenna

Week's Progress

This week we made progress in a few areas. First of all we finished up our Problem Statement based on suggestions from Kirsten, the TA's, and our clients. Secondly we developed a rough draft of our Requirements Document. Finally, we have made plans to get our own virtual reality gear from our clients. Before this there had been discussion of us sharing VR gear with another Capstone team who is also doing a VR project with Intel. While this would have been doable, it will be much easier if we have our own VR gear.

Problems Encountered

This week we did not run into any road blocks. The one thing that held us up a bit is that we still have many questions for our clients about the specifics of our project requirements. Once we get this information our Requirements Doc should come together nicely.

Plans for the Coming Week

For the coming week we plan to meet with our clients again and finish up our Requirements Doc.

5.3.3 Sean

Week's Progress

Met with our sponsors over Webex. Talked about getting our own equipment instead of sharing with another Intel VR group, and about finalizing our Problem Statement. We turned our problem statement in on time. We worked some on the requirements document, still need a lot of work on it.

Problems Encountered

The requirements document is going to be very hard. Just need to spend time thinking about it and working out each question. The better we do with it, the easier the rest of the year will be.

Plans for the Coming Week

Big plans are to just make sure that we keep in good communication with our sponsors and keep up with the documents. Also, need to continue ramping up on the Unity Game Engine. There will be more info in next week's blog post, this week was relatively slow.

5.3.4 *Spike*

Week's Progress

We finished our revision of the Problem Statement after another meeting with our clients. We also worked on a Requirements Document with our clients and turned in a rough draft. We think we convinced our sponsor that our own VR system would be beneficial for the development of the project.

Problems Encountered

We didn't run into any huge problems this week. We still have some questions about the Requirements Document but we'll be able to get those answered as we figure out more of the in depth details of the project.

Plans for the Coming Week

We'll need to continue to work on the Requirements Document to clear up any of the ambiguous points. We should be getting access to our own VR system sometime soon, and it'd be great to set that up either on campus or one of our apartments.

5.4 Fall Week 6

5.4.1 *Jake*

Week's Progress

This week I was able to meet with Christian from one of the other Intel teams. I'm hoping we'll be able to do some collaboration through the year. He was able to bring down an HTC Vive, the stands, and a PC on Thursday for my team. That evening my Team and I were able to set up the HTC Vive and mess around with a few free applications. This week we were also able to complete the Software Requirements Specification for the project.

Problems Encountered

We had some issues completing the Software Requirements Specification. We are all new to VR and therefore lacked the knowledge needed to put this documentation together. Fortunately we were able to discuss this with our sponsors and find a few resources to help us.

Plans for the Coming Week

Next week we will be ramping up with Unity development. Hopefully we'll be able to setup up an environment for us to work in as well as view our application from within the VR headset.

5.4.2 *McKenna*

Week's Progress

This was a busy week for our team. First we got our problem statement back which we did not do as well as we thought we did. Luckily we can still revise it and resubmit it. Secondly, we received our VR gear. This was huge for us. It was the first time that all of us got to try VR. Intel

was very generous with the gear that they lent us. Finally we spent most of the week finishing up our requirements document.

Problems Encountered

We encountered one road block this week related to our requirements document. We struggled to nail down quantitative performance requirements. Us and the clients agreed that the VR experience should be as realistic as possible. The problem is that this is a hard thing to measure. The one quantitative performance requirement that we came up with is to make sure that our experience runs at at least 60 frames per second.

Plans for the Coming Week

For the coming week we plan on exploring VR much more and getting ready for the tech review.

5.4.3 Sean

Week's Progress

We updated our rough draft and turned in a final copy. This was more of a struggle than we had thought it would be, mostly in the area of finding quantitative measures for how "real" our experience is. In the end we settled on our major "real-ness" measure being frames per second. Also, we were delivered our virtual reality hardware on Thursday. We signed up for Vive software and set up the physical headset. The first demo we ran through was AWESOME!

Problems Encountered

The requirements document proved to be very difficult for our group, although working through them was a great exercise. We have plenty of subjective requirements but not as many objective ones. I don't know if it counts as a problem for this week, but before we got our HTC Vive, it was a little hard to put what we were writing and thinking about in context.

Plans for the Coming Week

I am so excited to get better at Unity and be able to build something and then bring it to life in our Vive! Getting our VR setup working will open up so much more for our ramping on Unity. I also want to get more ahead on the documentation so that we aren't cutting it so close on the deadline this next time.

5.4.4 Spike

Week's Progress

We completed our Requirements Document after revising the draft we submitted for feedback. We met with our TA and clients to discuss specifics within the document and I think we've done a good job of outlining our requirements. A huge step in progress came on Thursday when we actually received VR gear including the headset and a VR ready laptop. Setup went smoothly and the components seem to be working well.

Problems Encountered

We had some issues when it came to the performance metrics in the Requirements Document. We couldn't figure out how to quantify how realistic or how immerse the VR environment we were creating would be . After meeting with our TA, we did some extra research and tried to keep our performance metrics as quantifiable as possible.

Plans for the Coming Week

We have the VR station set up so we'll continue to ramp on Unity and start to develop scenes that we can actually test on the system. A basic environment in Unity that can be tested with the Vive could be a possibility.

5.5 Fall Week 7

5.5.1 Jake

Week's Progress

This week we were able to experiment with Tilt Brush by Google. This application does an amazing job of handling the user experience. Tilt Brush offers an incredibly user friendly way to interact with a 3d paint brush. It is intuitive and every single person on my team was able to understand exactly what to do within 10 seconds. Moving forward we will strive to incorporate similar ideas into our project. This week I also wrote 3 pieces for the Technology review.

Problems Encountered

The main problem I had this week was finding time to setup an development environment. I think the Technology review could be helpful for some groups but for us, since we've already spent time choosing our technology, is a complete waste of time. It has taken away from us being able to move forward on the project.

Plans for the Coming Week

Next week I hope to complete the Technology Review and have a completed Gantt chart. This gantt chart will be incredibly valuable in helping us plan out our project to get an idea of we is possible in the time frame.

5.5.2 McKenna

Week's Progress

This week we continued to experiment with VR. We've been playing some of the most popular titles from Steam Store. Playing these polished games has given us a lot of ideas for features that we would like to implement in our game. For example, we have seen many different ways to display menus, some good and bad. Similarly, we have seen a couple different ways that developers deal with teleporting in game. Most of our other time this week has been spent working on the Tech Review

Problems Encountered

We didn't experience any major problems this week. The one thing we struggled with a bit was how to structure our Tech Review. For most of our project we will be making use of the Unity Game Engine. Therefore, for our Tech Review we simply decided to compare how we would implement features in different gaming engines.

Plans for the Coming Week

For the coming week we hope to implement a "Hello World" type of program in Unity. This will be our first try at developing in Unity.

5.5.3 Sean

Week's Progress

This week was a little bit rough. Mostly worked to figure out what the heck we are going to do for our tech review. We divided up tasks into four sections, one for each of us to write about. This also included working through some questions about which sections would require the most work. Also kept ramping up on Unity when time allowed.

Problems Encountered

I faced two main problems. First, I think our group is a little bit behind. It feels like we are reaching just to complete each document, and that is taking up most of our available time that we should be using more of to ramp up on Unity and get a prototype out. Second, it feels like those of us also in other classes with McGrath are getting a huge load of work dumped on us kind of out of nowhere. This is making it stressful to get everything done on time.

Plans for the Coming Week

Just to work as fast as possible to get the tech review done and start busting out more learning on Unity. Hopefully I can get a Hello World sort of environment with some basics going by the time I leave for Thanksgiving break. We will see.

5.5.4 Spike

Week's Progress

We spent a lot of this week getting more familiar with the Vive and trying out new games. We got assigned the Technology Review document so the team brainstormed some tasks. The big sections we came up with are: the environment, wand/rod usage, Columbia products and the UI and user guide.

Problems Encountered

The Technology Review document is a little troubling because our team has settled on using Unity with the Vive as that's what our clients suggested. We'll be looking at different development environments as well as different strategies within Unity but this assignment seems a little pointless.

Plans for the Coming Week

We'll be finishing up the Technology Review document this weekend and attempt to build a basic working prototype in Unity. We're looking to build a simple environment with maybe some objects the user can interact with.

5.6 Fall Week 8

5.6.1 Jake

Week's Progress

This week we completed the Technology Review. We were also able to set up a development environment to begin building our VR application. Fortunately there is a SteamVR plug-in in the Unity asset store which makes testing our application within the VR experience quick and easy.

Problems Encountered

No major problems this week but we will need to figure out how to quickly, effectively, and cheaply create a realistic outdoor environment. It looks like a lot of the more realistic assets cost money so we will need to do some research into other ways.

Plans for the Coming Week

Next week I hope to figure out what our environment should look like and how it should all be laid out. It would also be great if we were able to find free tools that can autogenerate landscapes and environments which we could then use as a template to build on.

5.6.2 McKenna

Week's Progress

This week we made progress in a few areas. First we finished out Tech Review on Monday. For us, this document ended up being very repetitive because most of our project will be completed in Unity and we were pretty much set on using Unity before starting the document. However, it was still a good exercise to explore how other gaming engines implement certain features. This week we also completed a "Hello World" type program in Unity. We were able to create a simple environment using basic assets from the Unity Asset Store and view that in the HTC Vive headset. This was a big step for us.

Problems Encountered

We did not encounter any problems this week.

Plans for the Coming Week

For the coming week we plan to start our Design Document and continue experimenting with Unity.

5.6.3 Sean

Week's Progress

Finished and turned in tech review. Got a real "hello world" environment ported into SteamVR so we can explore it in our Vive headset. Started exploring more of the documentation and tutorials for building experiences in Unity.

Problems Encountered

The tech review was hard. We did it, but it was hard. For me, it was kind of a weird exercise because we were doing 500-1000 words for each section, and to feel like I did a thorough job on a topic like I tackled, I would need to do many times that and spend weeks. But it was still good to start working through documentation and delving deep into the nitty gritty.

Plans for the Coming Week

Build up a working prototype of the environment just so we can teleport around and explore while in virtual reality. Also, settle on whether we want to have our project documented for OSU. We need to set up a meeting for the week after Thanksgiving with our sponsors just to check in with them and ask any questions. Nothing specific to ask, more just to keep them updated and keep up good communication.

5.6.4 Spike

Week's Progress

We got a couple tasks done this week. We finished up our tech review document on Monday. A lot of it was repetitive where we talked about different gaming engines (Unity, Unreal Engine 4, Lumberyard) for the three different technologies. Our clients are asking us to use Unity so the document was a little tedious. We also were able to load a basic environment, made in Unity, onto the Vive.

Problems Encountered

There were no problems we ran into this week.

Plans for the Coming Week

We'll be working on the Design Document this week and improving the environment we've built.

5.7 Fall Week 9

5.7.1 Jake

Week's Progress

I put together a statement about my fears and excitements for our project. I also talked with Mike Premi about our Technology Review and about getting NDA agreements for our team so we can collaborate more with Erik's team.

Problems Encountered

The only problem I encountered was Erik's team is a little hesitant to collaborate with us since we haven't signed NDA agreements.

Plans for the Coming Week

Next week I plan to work on the Design Document and begin thinking about the final Progress Report.

5.7.2 McKenna

Week's Progress

This week we continued to develop our prototype environment in Unity. We are learning new aspects of the engine rapidly from online tutorials. We also continued to work on the Design Doc, but we still have a lot of work to do before the due date next Friday.

Problems Encountered

We did not encounter any problems this week.

Plans for the Coming Week

For the coming week we plan to spend most of our time finishing up the Design Document. One of our project sponsors also suggested that we look into the Singray Gaming Engine, so we will experiment with that a bit.

5.7.3 Sean

Week's Progress

This week was relatively quiet as a group. Most of my time was spent just doing some research and work on a basic prototype. I started looking into the Design Doc but I have not had a ton of time to work on it.

Problems Encountered

First off, this wiki post was supposed to be done this last Friday, but being the day after Thanksgiving and this weekend being kind of hectic I forgot. Second, I was away from the HTC Vive headset and all of our VR equipment so this week was a little bit hard to get much progress done on actual prototype development. Lastly, I am scared that we haven't done enough implementation to truly write a good Design Document.

Plans for the Coming Week

I am going to try and push some implementation this Monday and Tuesday, while also buckling down on Design Doc work. Hopefully have the Design Doc done and ready for review by Mike and Tim Wednesday night by the latest. Also, I want to spend time each day this week mapping out topics, content, etc for our poster and term presentation.

5.7.4 Spike

Week's Progress

We worked on our prototype of the natural environment in Unity. I looked up some VR fishing games on YouTube to check out the interaction of the wands and rod. We did a little work on the Design Document but we'll have to finish that up this week.

Problems Encountered

We didn't encounter any problems this week.

Plans for the Coming Week

We'll be working on the Design Document this week. We'll probably also improve on the prototype of the basic environment that Sean created in Unity.

5.8 Fall Week 10

5.8.1 Jake

Week's Progress

This week I created a design document to plan out the entire development process. My team also reached out to the other VR team to discuss future collaboration opportunities.

Problems Encountered

The only problem we encountered this week was trying to figure the requirements for our design document. The instructions were incredibly unclear and it seemed like everyone was telling us something different.

Plans for the Coming Week

Next week I plan to start working on the prototype. This will continue into the following week.

5.8.2 McKenna

Week's Progress

This week all of our time was devoted to developing our design document. We had hoped to spend some time working on our prototype in Unity, but with the chaos of dead week, this didn't happen.

Problems Encountered

We struggled quite a bit with the format of our design document. First of all, the IEEE format document was not clear at all. This was made worse by the fact that some of the professors recommended a completely different format. In the end we ended up with something that made sense to read, but may not fit the IEEE format correctly.

Plans for the Coming Week

For the coming week we plan to spend time working on our progress report. After that is done we are going to spend a few days of our winter break flushing out a prototype because we will all have free time for once.

5.8.3 Sean

Week's Progress

This week we wrote our entire design document, which was super helpful for getting the details of our project into focus. Also, we finally got a real good meeting with Erik Watterson and his team to collaborate on our similar VR projects.

Problems Encountered

Not too many problems actually! We need to sign the NDA for Erik's project so that we can more easily collaborate, but other than that no issues!

Plans for the Coming Week

We have the final projects due, and then just work a bunch on the actual implementation!

5.8.4 Spike

Week's Progress

We started and finished our design document. We haven't gotten the signatures yet but our finished document was turned in on time. We didn't have any time to work on the prototype.

Problems Encountered

We ran into a lot of troubles with formatting the design document. The IEEE format wasn't clear, and the professors said to disregard a lot of the information provided on the assignment sheet. We were able to complete it, and hopefully it's in a passable format.

Plans for the Coming Week

We need to finish up the progress report. This includes the written presentation and a recorded presentation. After that, we should have some time to work on our prototype in Unity.

5.9 Winter Week 1

5.9.1 Jake

Week's Progress

Over the break we were able to create our first prototype. It involved a terrain, trees, water animation, first person VR view, and minor touch mechanics with the VR controllers. The first week of this term we fixed an issue with our trees, we expanded our terrain to improve realism, and started adding direction to the water. As a team we also decided on weekly meeting times throughout each week to work on the project.

Problems Encountered

One problem we ran into was adding direction to the water that would match the curves of the river. We will continue to research solutions and try combining multiple water objects.

Plans for the Coming Week

Next week I plan to meet with the TA for the first time since last term. We will also reach out to our clients to set up more regular conference calls. We will continue to develop the terrain and improve our river feature.

5.9.2 *McKenna*

Week's Progress

This week we continued work on the prototype that we had started over winter break. We made progress by enlarging the terrain and making it generally more realistic. We also sat down and figured out good meeting times for our group throughout the term.

Problems Encountered

We are still a little shaky on how we will be actually incorporating the Columbia Sportswear gear into our experience. We will be meeting with our project advisors soon to figure this out.

Plans for the Coming Week

This coming we will continue to work on our prototype. We are also planning on meeting with our project advisors and possibly the other VR team at OSU to see where they are in their project.

5.9.3 *Sean*

Week's Progress

I did not get a lot done this week, but we met as a team and set up when we wanted to meet and how often. We also talked a little bit about our general strategy for attacking this term.

Problems Encountered

Not many problems encountered, mostly just that we only have 10 weeks and we need to get rolling on this. We also need to get our NDA with Intel sorted out if we are going to work with Erik Waterson's team to combine knowledge.

Plans for the Coming Week

This week I plan to watch a ton of tutorials and just get more versed in basic development for VR, as well as how to create an awesome static environment. Just basically attacking every problem I can this week.

5.9.4 *Spike*

Week's Progress

We have a working prototype of the static environment in Unity and we spent a little time this week working on that. We met up as a group to talk about meetings and also created a rough list of tasks we want to accomplish soon.

Problems Encountered

We had some trouble with water textures and the trees within the virtual environment. We got those sorted out but we'll have to continue working on that.

Plans for the Coming Week

We'll continue to work on our environment in Unity. We also have plans of setting up a conference call with our sponsors to catch up on our progress and figure out what's next.

5.10 Winter Week 2

5.10.1 Jake

Week's Progress

This week we first reached out to our clients to begin setting up weekly or biweekly meetings. We also all met as a team multiple times to create a plan moving forward and to set out initial design goals. These mainly involved creating a new river and adding new rocks and plants.

Problems Encountered

The main problem we ran into was that we are unsure whether or not to commit a lot of time to building our own assets or requesting a budget to purchase prebuilt assets so that we can focus on more import development.

Plans for the Coming Week

Next week I hope to talk with Mike and Tim to discuss assets and a possible budget. I hope we are also able to develop a routine moving forward where each team member has their own section of the project to be working on.

5.10.2 McKenna

Week's Progress

This week we made progress in a couple areas. First of all we began working on making the water of river more realistic. We found one free Unity package that seems promising. We also began making the foliage more realistic. This will primarily be in the form of bushes and grass around the river. Finally, we approached our clients about setting up biweekly meetings.

Problems Encountered

We have realized that without a budget to buy assets, we will spend most of our time manually building assets. It would be much more beneficial to spend our time working on other interesting aspects of the project.

Plans for the Coming Week

This coming week we need to meet with our clients to discuss a possible budget to buy assets. We would also like to meet with them to discuss the specifics of incorporating Columbia gear into the project.

5.10.3 Sean

Week's Progress

We met a couple times as a team to just start attacking our development head on. Built out and refined the landscape a bit, and learned a lot more about availability and usability of external assets.

Problems Encountered

We are going to need either assets from Columbia/Intel, or a budget to buy nicer asset packages. We need to do this to bring our environment to life, and to include Columbia products in our experience.

Plans for the Coming Week

Build out the environment a bit with what we have, but don't go too far until we nail down final assets etc. Just mainly put in as many hours as possible learning about the whole development process and each facet.

5.10.4 Spike

Week's Progress

The team met up a couple times to work on the environment. We've been experimenting with different water shaders and tree/rock combinations. We emailed our sponsors to update them on our progress.

Problems Encountered

We're having trouble incorporating Columbia products within the VR environment. We're either going to need to get assets from our sponsors or figure something out with them in the coming weeks. It'd also be nice to have a small budget to purchase some high quality assets for the fly fishing environment.

Plans for the Coming Week

We'll continue to work on our Unity project but we really need to have a conference call with our sponsors to sort out the problems we're running into.

5.11 Winter Week 3

5.11.1 Jake

Week's Progress

This week we worked on building up our environment with foliage and rock features. We also started creating a vision for how we want our final application to look.

Problems Encountered

One problem we encountered was the rendering distance while in the development view. We were having a hard time adding grass to our terrain because we couldn't see where we had already placed it.

Plans for the Coming Week

Still haven't heard from Mike or Tim so next week I hope to talk with them to discuss assets and a possible budget. Just like last week I hope we are also able to develop a routine moving forward where each team member has their own section of the project to be working on.

5.11.2 *McKenna*

Week's Progress

This week we continued to improve the realism of our terrain. This was primarily in the form of grass and rocks around the river. We also made some initial progress in implementing user interaction with objects in the environment. To do this we used a Unity package called NewtonVR which is a physics based VR interaction system that supports the Vive very well.

Problems Encountered

We have yet to hear from our project sponsors after emailing them over a week ago. This is a little concerning because there are some things that we really need to discuss with them in order to begin making more progress on the project.

Plans for the Coming Week

This coming week we hope to meet with Mike and Tim. We need to discuss a budget to buy assets, and the details regarding how we will include Columbia gear into the experience. We will also continue to work on the user interaction and realism parts of the project.

5.11.3 *Spike*

Week's Progress

We continued to work on the environment and a little work on interaction with objects with NewtonVR. Jake had a cool idea of creating a "campsite" location in the VR environment to showcase the Columbia gear. We'll start implementing that.

Problems Encountered

The environment is looking decent but paid assets, especially for the water, could be really helpful. We also don't have access to any Columbia assets yet so it's hard to incorporate that part of the project into our design.

Plans for the Coming Week

We emailed our sponsors but we haven't gotten a response yet. It'd be really good to have a conference call to share out progress and to have Tim send us some Columbia assets that we could use. It'd also be nice to have a budget so we could purchase some paid assets from Unity's Asset Store.

5.12 Winter Week 4

5.12.1 *Jake*

Week's Progress

This week we thought about our end goal. We then proceeded to divide up the project into four major sections and assign these to group members. I am responsible for the campsite which is the location where the user will start out. This will be where the user gets comfortable and can inspect

Columbia gear. So far I've created a flat location and added some free example assets. I also carved out a road that leads to the campsite.

Problems Encountered

One problem I encountered was that there are very few quality, free assets available. I placed sample assets throughout the campsite as placeholders.

Plans for the Coming Week

Plan to talk with Mike and Tim on Monday and discuss a possible budget. As a team we also plan on consolidating our work and making more plans for the future.

5.12.2 McKenna

Week's Progress

This week was primarily a planning week for our team. We all got together and made sure that it was clear who is responsible for each part of the project. I think this was a big step for us. I will primarily be working on the fishing rod. We also made a decision to make the main part of our experience be a campsite where the user can view and interact with Columbia gear. This gives us much more freedom compared to having the experience be centered around fly fishing.

Problems Encountered

We heard back from one of our clients, but have yet to hear back from the other client. Therefore, we were once again unable to meet to discuss the project this week. This has been a significant roadblock for us. It has been a large amount of time since we met with our clients.

Plans for the Coming Week

This coming week we will finally meet with Mike and Tim to discuss the project. We will also all continue to work on our individual aspects of the project.

5.12.3 Sean

Week's Progress

Met with Vee on Monday (finally, I had missed the previous one so this was my first). Also went to class for the first time in a while. Good to get back and locked in. My team corresponded with our sponsors to set up a regular winter term weekly conference call.

Problems Encountered

Still facing an issue of not having the funds for good asset packages. Hopefully a conference call with our sponsors this coming week will clear things up in that manner.

Plans for the Coming Week

This weekend I plan to get my personal laptop all up to speed so I can begin working on my individual section of the project. I want to make a bunch of progress this week while I have a relatively quiet week.

5.12.4 *Spike*

Week's Progress

This was a slower week but we got some good planning done. We created a rough schedule of certain tasks that need to be completed to be able to produce a working beta by the end of this term. I'm in charge of Columbia products and the product display interface so I did some research on menus and other UI elements in Unity.

Problems Encountered

We didn't run into any problems this week.

Plans for the Coming Week

We have a conference call with our Columbia sponsor, Tim, next Monday. We'll be able to ask a lot of the questions we've been mentioning in our previous blog posts. The biggest things are: a budget for Unity assets, Columbia product assets and scheduling a time to visit.

5.13 Winter Week 5

5.13.1 *Jake*

Week's Progress

This week we had a conversation with Tim about project goals and direction. We will be moving away from displaying gear and more towards a physical experience (fly fishing, cutting wood, skipping a rock). I also spent time working on building up the campsite as well as helping to scale the landscape.

Problems Encountered

One problem we encountered was that the direction we were going was slightly off since Columbia already has VR applications that display gear. Instead we will be creating interactive experiences.

Plans for the Coming Week

I plan to finish the campsite boundaries and begin working on object interaction. I hope we can talk with Mike and Tim again to bounce some ideas off of them.

5.13.2 *McKenna*

Week's Progress

This week we were able to meet with one of our sponsors, Tim, from Columbia. This meeting was very productive. In this next couple days Tim will be providing us with assets of Columbia apparel that we will be able to use in our project. We also discussed the possibility of a small budget for our team, which will allow us to buy some higher quality assets, that will improve the realism of our project. Finally, we decided that user movement, in the form of fishing and possibly other activities should be the focus of our project.

I also made some small strides on the fishing aspect of the project. I am able to "cast" a line. It's a simple rigid line for now, but it's a start.

Problems Encountered

The meeting we had this week did steer our project in a slightly different direction than we were initially working towards. However, it is better to change direction now rather than in a few weeks.

Plans for the Coming Week

This coming week I plan to continue working on the fishing aspect of the project.

5.13.3 Sean

Week's Progress

We met briefly with Vee on Monday, not a lot there. Much more importantly, had a conference call with Tim (Columbia client) and got a lot more clarity on how to go forward. This was good because it had been a little while since we had spoken to our client "in person" (not email).

Problems Encountered

No major issues. A little bit of a change of direction based on the talk with Tim, but that is a good thing not really a problem. We still have the issue of not enough assets, but we worked out a tentative fix with Tim.

Plans for the Coming Week

This week the main focus is continued development and also work on a set of assignments for the class portion of capstone. For development, I want to finish up the environment using the assets we currently have. Get it to a state where we can take some screen shots or even video as an update for clients and Github.

5.13.4 Spike

Week's Progress

We had a call with our Columbia sponsor, Tim Devlin, to catch up with our progress. We discussed his vision for the project, the availability of Columbia assets and a budget for other assets to improve the aesthetic of our environment.

Problems Encountered

After talking with Tim, it looks like we have to change the direction of our project. We originally thought the main goal of this VR project was to promote Columbia gear, but it seems like the interaction/fishing portion is what's really important. We'll have to make some minor adjustments to account for this.

Plans for the Coming Week

We'll continue to work on the fishing experience as there's a bigger focus on that now. We should be getting the Columbia assets sometime in the upcoming week so we can work to add those to the scene as well.

5.14 Winter Week 6

5.14.1 Jake

Week's Progress

This week I continued developing the campsite. In general, development was put on hold this week to work on our midterm progress report. We created a progress report and a video to accompany it showing what we have done so far.

Problems Encountered

No major roadblocks this week as we spent most of our time on documentation rather than development.

Plans for the Coming Week

This upcoming week I hope to finish the boundaries of the campsite, experiment with sky textures, and get approval for some low cost assets.

5.14.2 McKenna

Week's Progress

This week we spent the majority of our time finishing up both the written and video portions of our progress report. This was a bit frustrating because it took away a lot of time from development, however it was beneficial to reevaluate where we are in the project. I also made some small strides with the fishing rod.

Problems Encountered

We are struggling to make large strides when it comes to the fishing rod. I think that more of our attention will be focused on user interaction in the following weeks.

Plans for the Coming Week

This coming week we will continue our development, and work on our elevator pitches, as we have class on Tuesday.

5.14.3 Sean

Week's Progress

Conference call with Tim Devlin went well, we are all set to get some assets from him soon and to include Mike Premi again on the call next week. Most of this week was spent on documents, but got a little more done on the physics end (mostly McKenna so far on that).

Problems Encountered

No big problems encountered. It was a little difficult to figure out how to make edits to our documents to remove a whole portion of functionality from the original plan, but I think we got it mostly figured out.

Plans for the Coming Week

This weekend I plan to do a major overhaul of the terrain and landscape in order to bring the magnitude down a little. We are having some frame rate issues, and distance rendering issues, and we think it might be because of our large landscape. Then after that is done hopefully I can move to some more final touches to the landscape and start catching up to McKenna in knowledge on the physics and fishing rod stuff.

5.14.4 *Spike*

Week's Progress

We spent most of our time this week working on our progress report for Winter term. This included a video and written report on our current progress and future plans.

Problems Encountered

It's been rough adjusting to the change in project scope to focus more on the fishing mechanics. We didn't have much time to develop in Unity this week but we should be good for next week.

Plans for the Coming Week

We really need to ramp up on development and get the interactive fishing portion of the project working.

5.15 Winter Week 7

5.15.1 *Jake*

Week's Progress

Since we have two locations within our VR experience (campsite and river), we needed to figure out a way to allow users to transfer between them. One solution we came up with was to add some teleport objects which will allow users to teleport on contact.

Problems Encountered

Major roadblock we found this week is our fps while running the application. It is lower than we had hoped so now we have to start trying to understand the major causes and start eliminating them.

Plans for the Coming Week

this upcoming week I hope we can adopt our new environment and start cleaning everything up. I also hope we can start using some new assets.

5.15.2 *McKenna*

Week's Progress

This week I continued to work on the fishing rod. Specifically I worked on the line physics of the rod. We found a project on GitHub called Cable-Component which we will be utilizing for now in order to create the fishing line.

It uses verlet physics attached to a simple LineRender. This approach is fairly straightforward and should work for our needs. We also made some progress in figuring out how to 'teleport' the user to different locations in our VR environment.

Problems Encountered

No major problems were encountered this term. We have yet to receive the Columbia assets from our project client, but hopefully we will get them sometime this term.

Plans for the Coming Week

For the coming week we plan to continue making development progress. The three main areas we are working on are the fishing mechanics, improving the framerate, and creating a menu system to teleport the user to various locations.

5.15.3 Sean

Week's Progress

This last week I pretty much only worked on redoing the landscape so that the render distances would be shorter and the area would look more like Smith Rock State Park. I am just finishing that up today and tomorrow morning, and will then do an FPS test to see whether there is a major difference from the old landscape.

Problems Encountered

No big problems encountered, mostly just the fact that redoing the entire landscape is more work than I thought it would be. Coming along pretty nicely though!

Plans for the Coming Week

This coming week I am going to finish up the landscape, and get myself looped back in with the rest of the team on other topics that I have gotten behind them on.

5.15.4 Spike

Week's Progress

We spent a good amount of our time working on the fishing line in Unity. We found this Cable-Component on GitHub - a cable component implementation using verlet physics that we based our fishing line on. We also worked on a teleportation system that we'll use when we want to move the user from the campsite area to the fishing area.

Problems Encountered

We ran into some problems with the cable component but we got our bugs fixed and it's looking good. We still have framerate issues, most likely due to the complexity of the environment, so we'll have to work on fixing that soon.

Plans for the Coming Week

Keep working on the fishing experience and look to add menus or other UI elements if needed.

5.16 Winter Week 8

5.16.1 Jake

Week's Progress

I finished the teleport pads and began researching how to program the fish within our river. I found a simple solution that involves creating a grid of objects and having the fish gameobject move between them. On contact with one, the fish will randomly choose another one to move to. Once this is setup, all I will need to do is add fish animation and make sure to change the direction of the fish based on a vector of the source and destination.

Problems Encountered

Biggest problem we ran into this week was trying to import a new terrain object into our environment. I worked with Sean to brainstorm how to do this and we were able to get it done by the end of this week.

Plans for the Coming Week

Next week I hope to finish the campsite boundaries as well as resize some of the objects within the campsite to a more realistic size. I also plan to implement the teleport pads.

5.16.2 McKenna

Week's Progress

This week we continued making progress with the fishing rod. We purchased a new fishing rod which greatly improves the realism. We also finally nailed down the orientation of the rod when the user picks it up. We also continued working on the fishing line.

We had a productive meeting with our clients. The biggest thing we finally resolved is that we will be able to get a small budget to purchase higher quality assets.

Problems Encountered

No major problems encountered this week.

Plans for the Coming Week

This coming week we will be incorporating the new assets that we purchase into our project. The one I am most excited for is the new rope asset. It will allow us to actually reel in the fishing line, if everything goes according to plan.

5.16.3 Sean

Week's Progress

Finally brought the new landscape over to the main project, and got awesome FPS results! Over 100 FPS almost exclusively. Also we realized that part of the issue is render distance and clumps of trees.

Problems Encountered

Had trouble moving just a scene or landscape from one scene to another. Other than that no major issues.

Plans for the Coming Week

Just mostly finishing up details on the environment. I want to get caught up on the rod mechanics code other team members have been developing so I can hopefully help.

5.16.4 Spike

Week's Progress

The group made really good progress with the environment and improved the framerate from 20 fps to 100 fps. We had a conference call with our sponsors and we were able to get a budget for some new assets. The biggest thing is the rope editor which should allow for easier management of the fishing line.

Problems Encountered

We didn't run into any problems this week.

Plans for the Coming Week

We're going to incorporate the new assets we're getting which should improve the realism of our scene. We'll continue working on the fishing rod mechanic which should be easier to manage with the rope editor.

5.17 Winter Week 9

5.17.1 Jake

Week's Progress

The first thing i did this week was complete laying out a border for the campsite. This made it a lot easier to then go in and apply texture to the terrain. I also added more content to the campsite to make it feel more realistic. I then proceeded to resize all the items to make sure they looked normal from the VR perspective. Throughout the week i also helped integrate the new rod and rope mechanics. Towards the end of the week I started adding text displays to the environment with instructions to new users.

Problems Encountered

Biggest problem I ran into this week was rescaling all the items to work with VR. We ended up resizing some of the items to make them smaller along with the VR character itself to make it bigger.

Plans for the Coming Week

Next week I hope to have the teleport feature finished which means we need to finalize the locations on the map where we want the players to be. I would also like to start working towards adding AI fish to the water.

5.17.2 *Mckenna*

Week's Progress

This week we purchased a new rope asset called Ultimate Rope Editor. Before we were using manual vertlet physics in an attempt to create a realistic looking fishing rod. This new rope asset will allow us to create a coil in order to mimic the reel of a fishing rod. We also continued to make improvements to our environment.

Problems Encountered

We are still waiting to get the funds to buy some assets, but we should get them sometime this coming week

Plans for the Coming Week

This coming week we plan to tie up all lose ends in our project before spring break. Then the next big thing we will be working on is creating the interaction with the fish assets that we purchase.

5.17.3 *Sean*

Week's Progress

Basically just more work on improving the environment. Had a good talk with Vee about what we have coming up, and more importantly what we should have done by the end of this term. There is a little bit of a sense of impending doom, but I think that part of that is just the scope of the project.

Problems Encountered

No major problems. Just the usual bumps and issues that come with learning new techniques and methods in Unity every week.

Plans for the Coming Week

This weekend and next Monday/Tuesday/Wednesday I plan to finish up the environment with the current assets we have. I leave for a quick trip second half of dead week, so I want to get ahead on some of the documents as well. I feel like I have been lagging a little and letting other parts of my life take precedence over Capstone, I want to fix that this week.

5.17.4 *Spike*

Week's Progress

We got the Ultimate Rope Editor from the Unity Asset Store so we've been developing our fishing line using that. It looks promising as the coil mechanic of the fishing line is built in to the editor. I also did some researching on fish movement AI and found some tutorials that could be useful in figuring out the fish/fishing line interaction.

Problems Encountered

We're having issues with collisions in the rope object which is causing the fishing line to spazz out. We've been able to reduce this a little bit by tweaking some settings but we'll have to work on this to improve realism.

Plans for the Coming Week

More work on the fishing line and work to incorporate the fish AI.

5.18 Winter Week 10

5.18.1 Jake

Week's Progress

Spent this week doing research related to AI fish. I think we now have a good idea of how to complete this aspect of the project. I also added in user instructions and signs.

Problems Encountered

No big problems this week. Only issue was finding time to work on the project since everyone is incredibly busy during dead week.

Plans for the Coming Week

Next week hope to have the poster finalized and complete the end of the term report. These progress reports normally take us a good amount of time to complete. This along with finals next week means development will probably slow down.

5.18.2 McKenna

Week's Progress

Since this week was dead week, none of us had much time to any actual development, unfortunately. We did spend some time working on the first draft for our poster. We also received funds from our Columbia sponsor to purchase some higher quality assets. We will do this next week or during spring break.

Problems Encountered

No problems encountered this week, other than all of us being busy with other classes.

Plans for the Coming Week

This coming week we hope to purchase new assets and incorporate some of them into the project, if we can find time inbetween studying for finals and wrapping up other final projects. We will also need to spend time working on our final progress report and video presentation, which will most likely occupy most of our time.

5.18.3 Spike

Week's Progress

We spend the little time we had this week working on our poster draft. We didn't have much time for development but we did receive the funds for assets so we can work towards incorporating higher quality items into our scene. I implemented a basic ai that follows the player within a certain range but I'll need to keep working on it and eventually port it over to the scene.

Problems Encountered

No problems encountered this week.

Plans for the Coming Week

I'm not sure if we'll have any time for development this week as we have finals. We'll need to work on our video and individual papers for the final progress report of the quarter.

5.19 Spring Week 1

5.19.1 Jake

Week's Progress

We spent the last week implementing new purchased assets into our unity project and working to finalize terrain plans. Two of the new assets we received were water and grass assets which have greatly improved our projects realism as we are no longer seeing any issues with reflection or animations. We also met with Tim Devlin and his colleague Chris Lundy. Afterwards we began figuring out how to most effectively share the project. We managed to create a build of our project with a file size of only about 20mb which was a relief.

Problems Encountered

No big issues this week but we are all a little nervous about the upcoming expo deadline.

Plans for the Coming Week

Next week I hope to improve the fishing location and add instructions to aid in the fishing experience. It is currently not very intuitive so we will need to make some changes.

5.19.2 McKenna

Week's Progress

This week we picked up on the development progress we had made before Spring break. This first thing we worked on was incorporating the new water asset that we purchased, called AQUAS. The process of adding the asset to our project was relatively painless. We just needed to perform some tweaks to the flow direction, and the color of the water. We also made progress on the process of actually catching a fish. Currently the fish follows our hook, when it is close, and it will latch to the hook. The issue is that the fish is freaking out because it is connected by a rigid joint, the the hook, and the hook is moving quite a bit.

We also had a chat with one of Tim's colleagues at Columbia. He is going to help us find an appropriate place to upload our work so the folks at Columbia can see our progress.

Problems Encountered

As mentioned above, we are having some issues with the fishing line. At times the line will freak out for no apparent reason. I do not currently have any fixes for this problem, but we will continue to investigate it.

Plans for the Coming Week

This coming week we plan to continue working on the fishing interaction, specifically the action of actually catching the fish. We are also going to work on uploading our work to Columbia's servers.

5.19.3 Spike

Week's Progress

We made decent progress this week in developing our fishing experience. We purchased some paid assets including new water and grass for our environment. We also got a fish model and implemented the line of sight fish pathfinding script. We're at a point where we can cast, have the fish hook on and reel back the fish.

Problems Encountered

We're having some issues with the fishing line - the same ones we've had since last quarter. The movement and physics of the line is unreliable and freaks out. We mentioned this to our sponsor and he made a comment about maybe getting us in touch with some developers at Columbia who could help with this.

Plans for the Coming Week

We'll continue to work on the fishing component of our project. Chris from Columbia asked us to upload our work to their servers so we'll work on getting that done as well.

5.20 Spring Week 2

5.20.1 Jake

Week's Progress

Last week we worked on finishing up our terrain and added instructions to aid in gameplay. Additionally, we talked with a Columbia developer about our project and future plans he has for virtual reality within the company. We also completed a first build of our project which we shared with the team at Columbia.

Problems Encountered

Only big problem we encountered was making the fishing line look perfectly realistic. We are going to continue manipulating the line segment masses, number of line segments, and air drag.

Plans for the Coming Week

Next week I plan to brainstorm ideas related to fixing our fishing line. Some possible plans may include manipulating the velocity of the fishing line itself when the hook collides with the water.

5.20.2 *MCKENNA*

Week's Progress

This week we made progress in a couple areas. Jake created menus that are visible around the fishing location, that will inform the user on how to fish. Spike and I continued to work on improving the fishing experience. One addition we made was allowing the user to reel in the fishing rod at different speeds. We all continued to debug the fishing line, but it is still proving to be problematic.

Problems Encountered

As mentioned above, the fishing line is our main problem. We have yet to figure out what exactly is making it "freak out", for lack of a better term. The line is constructed of many line segments, each one carrying a rigid body, and collier component. One thing we have observed is that if we do not include a collier on each segment, things begin to improve.

Plans for the Coming Week

This next week we need to finish our poster draft by Monday. The rest of the time will be spent developing, focusing on the fishing line.

5.20.3 *SPIKE*

Week's Progress

A lot of development time was spent on working on the casting of the line. We improved the casting speeds and added a feature to allow for the user to take the caught fish off the hook. Jake added some more menus to help the user with the fishing experience.

Problems Encountered

The fishing line is still spastic and we're still not sure how to handle that. We made some progress with steadyng the line but we'll have to keep trying different things to make it look more realistic.

Plans for the Coming Week

We need to finish up our poster draft and continue development on the fishing experience.

5.21 Spring Week 3

5.21.1 *JAKE*

Week's Progress

This week we continued trying to debug out fishing line as we were seeing some issues during gameplay. We also worked on creating a build of our project and uploading it to the Azure storage space setup up for us by Columbia.

Problems Encountered

We are still struggling a bit to make the fishing rod behave exactly how we want it to be we are seeing improvements. Another issue we started running into was that we couldn't seem to upload

anything except the executable of our project to the azure space. We tried to zip all the assets to share with Columbia because we kept getting errors on all of our systems. Eventually I was able to get upload the folder with our build data but it still came through a few errors. We are now currently waiting to see if Chris at Columbia can access all the information from his end.

Plans for the Coming Week

Next week I hope to make other objects in the campsite interactable including the shovels and maybe some stones that are laying around. It would also be interesting to add some clouds to the sky. This week, as a team, we will start thinking about what we want to upload to github. We haven't been using this tool at all so we need to make sure we put something up there before the code freeze.

5.21.2 McKenna

Week's Progress

This week we didn't make as much progress as I would have hoped on the development side of things. We did make progress in other areas though. We uploaded our initial build of our project to a server at Columbia Sportswear, so some of the folks there can test it out. Mike, our sponsor from Intel was gracious enough to drop off a VR viewer for us, similar to Google Cardboard. This will be very useful for our Expo setup, as we will not be having our actual Vive system setup.

Problems Encountered

As with previous weeks, we are still having issues with the fishing line. At this point, we are just trying to make it as stable as possible. It will most likely not be perfect, before our deadline.

Plans for the Coming Week

This coming week we will just be scrambling to prepare for our code freeze on May 1st, recommended by Kevin. We will need to wrap up all loose ends.

5.21.3 Spike

Week's Progress

We didn't make too much progress with our Unity project, we're still having a lot of trouble with keeping the fishing line taut. We did reach out to Chris from Columbia who's helping us out with pushing our project over to their server. We uploaded our executable and project folder; we're still waiting to hear back how that went.

Problems Encountered

We're still having issues with the fishing line. We're at the point where we're not really sure how to fix it since we've been working on it for so long.

Plans for the Coming Week

We just have to finish up some development before the code freeze.

5.22 Spring Week 4

5.22.1 Jake

Week's Progress

This week I implemented potential product placement within the VR application. In the campsite I added a mannequin along with product information such as features and materials. The goal of this was to create a bridge between our application and Columbia's existing projects.

Problems Encountered

No major issues this week. We worked hard to make our poster accurately represent the enthusiasm we have about the potential of this project.

Plans for the Coming Week

Over the next week we will be preparing for the github code freeze. Our application requires Virtual Reality hardware to run which adds some difficulty in creating effective compile instructions. The last things we need to add to our project are interactive objects within the campsite. Intractability plays a huge role in creating realism when it comes to Virtual Reality so this is a top priority for us this week.

5.22.2 McKenna

Week's Progress

This week we have been going through our requirements documentation and our current project, and making sure that everything lines up nicely. We realized that we are missing some small things like sounds, which we added this week. We have also been working on the finishing touches of our project like the user instructions, and various bug fixes. This fishing interaction has improved slightly thanks to Sean, and I think it's at the best state it has been at. When you cast, it now takes in to account how fast you are moving the rod to determine the speed of the cast. This adds to realism.

Problems Encountered

Since our project cannot simply be cloned from GitHub and run on any computer we have been having a lot of discussion as to how we should submit out project. We need to flush this out before May 1st.

Plans for the Coming Week

This coming weekend we will be working to get everything in order before the code freeze. After the code freeze we will begin working on making an awesome demo video for Expo.

5.22.3 Sean

Week's Progress

This week I helped more with the team's push to bug fix the fishing rod casting mechanics. This was great for me because I got to dive deeper into parts of the project code that I normally don't touch.

We also as a team did a good amount of work finishing up the poster, getting it signed off by our clients, and reviewing our poster layout with Group 8.

Problems Encountered

We didn't encounter any specific issues this week. At least on my part, there is still a sense that I won't be ready for various deadlines between now and Expo, but I get the send that many other groups feel the same way.

Plans for the Coming Week

The big task this coming week is to prepare for the Monday Code Freeze. This means getting all our code in Github, making sure the executable will run correctly, and writing a good "recipe" for compiling everything.

5.22.4 Spike

Week's Progress

We went through our requirements document and created an updated version of our poster. We got approval from both our sponsors and went into Kirsten's office hours to go over it as well. We added noises to our environment to improve immersion. Sean also made some changes to the fishing line so that it adjusts to the cast speed of the user.

Problems Encountered

We have a code freeze on May 1st so we'll work to upload our project by then. It might be difficult as the Unity project is really big, but we'll figure something out - maybe just upload an executable and some of the scripts.

Plans for the Coming Week

We'll working on finishing up our code and pushing to GitHub.

5.23 Spring Week 5

5.23.1 Jake

Week's Progress

This week we got everything ready for the code freeze. Our project is a little too big for github so instead of uploading everything we just uploaded our executable and a series of screenshots. We also plan to get our project up on one of our ENGR spaces with a link on our github for anyone interested in looking at all the assets and source code that was generated. We also finalized our poster and submitted it for printing.

Problems Encountered

Only issue we had was in uploading our project to github. It's too big to fit everything in our repo so we only uploaded the essentials.

Plans for the Coming Week

This week we will be start on our final report and on our video for expo. Since we won't have enough room at expo to setup the VR, a video is necessary to showcase our project.

5.23.2 *McKenna*

Week's Progress

This week was fairly slow for our team, except for Monday. On Monday we were busy all day fixing any last minute bugs in preparation for the code freeze at midnight. Overall I am happy with what we were able to submit for the code freeze. We also polished our poster and sent it off for printing. The rest of the week the only thing we were really working on was our WIRED articles.

Problems Encountered

No problems encountered this week. It's a miracle.

Plans for the Coming Week

Next week we will be creating an awesome video for Expo.

5.23.3 *Sean*

Week's Progress

This week the main item of work was the big push to clean up and present our project on Github for the final code freeze. We feel pretty good about being able to show our project off, but we will have to see what Kevin thinks.

Problems Encountered

Two big problems. First, our project folder is too big to upload to Github completely. That means we had to upload scripts and screenshots, then provide a link to a web space with the bigger asset folders. Second, it is REALLY hard to grasp what we have accomplished without putting on a headset and actually using our product.

Plans for the Coming Week

Our big focus now is to create an awesome demo video of our project for the midterm report and for expo. Although a headset is the only way to really experience our project, a killer demo video will definitely help.

5.23.4 *Spike*

Week's Progress

We finished cleaning up our code and adding small features before the code freeze. I'm happy with the product we came up with and it'll be cool to demo it at expo. The rest of the week we spent time working on our WIRED interview articles.

Problems Encountered

No problems this week.

Plans for the Coming Week

We'll be creating our 360 VR video for expo this week.

5.24 Spring Week 6

5.24.1 Jake

Week's Progress

This week I completed my section of the Spring term progress report and started on the report slides. I didn't get a chance to start on the expo video yet but I have started brainstorming how the video will be structured. As a team we have also made plans for expo as to how we will be laying out our table.

Problems Encountered

No significant issues this week as development is pretty much done.

Plans for the Coming Week

This week we will complete our video for expo and present at expo on Friday. This next week will be pretty busy because I know a couple of us also have midterms.

5.24.2 McKenna

Week's Progress

This week was slow for our team, our team was simply spent working on the midterm progress report, and presentation. We have also begun to start working on our demo video, which will be displayed at the Expo, since we will not be setting up the Vive System.

Problems Encountered

No major problems. We would like to spend all of our time working on our demo video, so the progress report is a bit of an annoyance.

Plans for the Coming Week

This coming week we will be finishing the progress report, and finishing up our demo video.

5.24.3 Sean

Week's Progress

After the code freeze, now we are focusing on our Expo demo video, and the midterm report due on Monday. We planned out and divided tasks for the midterm report, and talked through how we can best display our product without actually putting people in a headset.

Problems Encountered

I feel like we are repeating a LOT of stuff from our final winter term report to this midterm report. There are a couple ways we can adjust what we present, but we don't know how far we can or should stray from the general form in order to make it new and interesting.

Plans for the Coming Week

For Monday night we just want to create the best product for the midterm report that we can. Then, we will take the video content from the midterm report and improve and combine it to create a short(ish) effective video for Expo. I also think we should talk through common questions we might get and how we will respond to them.

5.24.4 Spike

Week's Progress

We spent our time working on the midterm progress report and presentation. We started working on the demo video that we'll be presenting at Expo.

Problems Encountered

No problems this week.

Plans for the Coming Week

We'll finish up our deliverables and our demo video.

5.25 Spring Week 7

5.25.1 Jake

Going into this project I had no experience working with Virtual Reality but I was intrigued by the project. The applications of VR are growing exponentially and I knew this would be a great opportunity to learn about the community and the VR development process. During the first month I worked really hard to learn the basics so I could dive in as quickly as possible. Throughout this project I made lots of mistakes but that was expected since I didn't have any VR knowledge and was learning on the go. I had an incredible time working on this project with the team and I wouldn't change much of anything.

Before the project assignments I had reached out to the client of this project, Mike Premi, to get more information about it. He told me about the direction he saw it going and the kind of team he was looking for. I was eager to take on this assignment so he told me I could have it if I was able to get a team together. This allowed me to recruit some of my close friends and the best developers I know. One of my biggest fears for going into capstone was getting placed on a team with people I did not get along with or who were not motivated to make something really memorable. That's why I worked so hard the first few weeks but I also got really lucky with how it all worked.

Our Virtual Reality application had a wide range of challenges but we were always able to work through them as a team. We had a great understanding and in the end I think we created something pretty cool. It's only a proof of concept but I think we shown Columbia the potential Virtual Reality could have as a marketing tool and as an interactive and immersive way to engage with customers. Moving forward we will be passing this project off to a current sophomore at Oregon State who will be interning at Columbia this summer. I was able to meet him at the engineering expo and I believe

he's perfect for the job. I told him if there is anything he has questions about involving the project he can always reach out to us.

5.25.2 McKenna

Expo was a big success for our group. Even though we were not able to have our complete VR setup, it was nice to be able to showcase something with our demo video. This along with the fact that we were doing a VR project made us quite a popular booth.

Looking back at the entire process of working on this project I think that we have all learned some valuable skills. Going into the project we had some really high expectations. Throughout the process we needed to lower/and or refine some of our expectations. It is a very valuable skill to come up with realistic expectations. While I do not think that any of us would like to get into game development after college, the technical portion of the project was still a great learning experience for everyone. Being able to pick up any new technology in a reasonable amount of time is a great skill to have.

I hope that our project continues to be worked on in some fashion either by Columbia or maybe as a future capstone project. I know that our VR gear will be donated to the new VR club here at OSU so I'm sure we will see some exciting things come out of OSU related to VR. The next big step for our project would be to begin the process of incorporating actual Columbia gear into the project. This was our goal from the start, but it never happened for a number of reasons. That addition would really bring the project together.

5.25.3 Sean

There are a number of little things that I would change about my Capstone execution, but here are the big three. 1. Start developing WAY earlier. I don't think that I should have made major progress Fall Term, but it would have helped my Fall Term documentation tremendously if I had just started hands-on exploring earlier. 2. Take the tech review document more seriously. I obviously gave solid effort on the tech document, but I wish I had dug more deeply into the choices available and really found out what people in the industry think of major VR tools. 3. Push harder to have a face-to-face meeting early on with my clients.

I really enjoyed the experience of Capstone overall. To start, I will say what I bet 80% or more of the class is going to say, which is that I liked the development and didn't like the documentation. To delve deeper than that, I would say that I really enjoyed interacting with our clients and working out what they needed from us and what we could provide. I also got a lot out of the weekly blog posts. I didn't write much for them and even missed some, but the blog posts helped keep me focused on the big picture, and accountable to take a step back and look at progress at least once a week.

As mentioned above I didn't like documentation. Part of that was because we were being asked to nail down requirements, but we really didn't get much from our clients. It just constantly felt like we were being pushed to pull BS out of thin air to fill in around the requirements we actually knew.

5.25.4 *Spike*

I think we did a really good job with this project as a proof of concept for Columbia. We had a lot of people come by at expo and our sponsors seemed pretty excited with what we had. I think our team worked really well together and creating this project together was a lot fun.

If we were to redo the project from Fall term, it would've been helpful to explicitly set our requirements for this project. While it was cool that our sponsors gave us a lot of freedom to create our project, at points we got confused on what our clients specifically wanted. The biggest skill I've learned this term is definitely working with Unity and writing C#. I liked the idea behind our project and how it all came together; I think our expo video and 360 video are both really well made. It would've been nice if we purchased some higher quality assets after looking at the other VR team's projects. I think it could've added a lot to the experience.

Like I said, both Tim from Columbia and Mike from Intel seemed pretty happy with our product at expo. I'd be pretty satisfied with our product. The fishing line definitely needs work but I think the core idea behind the experience was developed well. If we were to continue to work on this project, I'd want to work on the following: adding higher quality assets and improving the visual quality of the environment, incorporate Columbia gear within the experience and fix the fishing line.

FINAL EXPO POSTER

INTRODUCTION

Many outdoor activities these days require a large initial mental and economic investment to get started. This makes people less likely to try new outdoor activities. In collaboration with Columbia Sportswear and Intel Corporation, we set out to develop an interactive product demonstration with virtual reality to combat this issue.

This project has the potential to inspire people to get outdoors and try new things by first getting them comfortable in new environments or performing new movements. It strives to make outdoor activities accessible to everyone regardless of their experience level. This project will not only inspire but also improve the retail experience by making it more immersive, interactive, and informative while being entertaining at the same time.

BACKGROUND

The current retail experience for outdoor gear is disconnected from the experience that consumers have when using the gear in the outdoors. Additionally, the current experience is neither interactive nor immersive.

An ideal retail experience needs to be informative yet entertaining to inspire consumers to get outdoors. Columbia Sportswear has already created a few projects that involve displaying clothing virtually so the main focus of our project was to incorporate a physical aspect into the experience. We also hoped to create a bridge between our new ideas and Columbia's existing projects.

While the overall idea of this project could be translated to many different outdoor activities, the focus of this project is fishing, in order to promote the Performance Fishing Gear (PFG) line at Columbia Sportswear.

Outdoor Experiences in Virtual Reality

Simulated Fishing in the Retail Space to Generate Consumer Confidence and Engagement



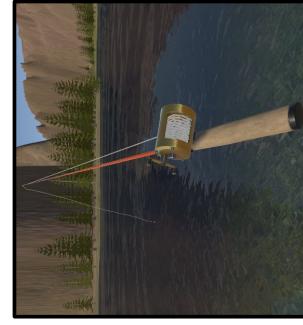
PROJECT DESCRIPTION

We developed the project using the Unity Game Engine and the HTC Vive. The primary components of our virtual reality experience are the terrain and the user interaction. We modeled our environment off of Smith Rock State Park in Central Oregon. More specifically, the iconic sheer cliffs and vegetation were the inspiration behind our design process.

We also put an emphasis on heavy user interaction within the experience as we found this was one of the most effective ways of aiding in immersion in virtual reality. We broke down the experience into two sections: a campsite area and a fishing area. Users will start in the campsite area where they can pick up and play with various items. When the fishing rod is picked up, the user will be teleported to the river where they can go fishing.

RESULTS

- Added gameobject touch-logic and animation to increase user engagement and in-turn improve virtual realism.
- Harnessed the Unity physics engine alongside the Ultimate Roped Editor to develop an interactive fishing rod and line.



LOOKING FORWARD

Our main goal throughout the project was developing virtual realism. We found that the most important factors that influenced realism were realistic interactions, responsiveness, and graphical clarity.

With our limited virtual reality background and knowledge we completed an experience that can certainly help Columbia and many other companies. Looking forward, this project can be taken in a variety of directions and we know there is huge potential in the retail space.

- We were able to create an immersive, virtual retail space in which users can partake in a outdoor fishing experience alongside virtual product lines.

- A physical activity users can engage in while wearing outdoor gear gives them the opportunity to test clothing while performing realistic movements.

- Our application could help improve the retail experience by making it fun, interactive, and informative.

- This project has demonstrated the great potential of Virtual Reality to redefine and transform outdoor gear shopping.

Oregon State
UNIVERSITY

- Took advantage of collision dynamics and velocity manipulation to create catchable fish with realistic movement animations and random swimming behaviour.
- Used position vectors and transforms to create teleportation pads within the environment to allow more user flexibility despite space limitations.



CLIENTS

- Tim Devlin
Mike Premi
Columbia Sportswear
Intel Corporation

6 PROJECT DOCUMENTATION

6.1 Technical Overview

Our development mainly took place within the Unity Software. All the environment modeling was done using Unity tools and scripting was done in Visual Studio. The built-in Unity Asset store was great for finding the assets we needed for our environment.

6.2 How to Install and Run

- Setup HTC Vive hardware and software by following guide on Vive website: <https://www.vive.com/us/setup/>
- Make sure the executable and the data folder are in the same location.
- Import executable into Steam:
 - Games -> Add a Non-Steam Game to My Library
 - Browse to saved executable
 - Add selected programs
- Run executable from within Steam Library

6.3 Hardware Requirements

- Steam and SteamVR
- HTC Vive Hardware (headset, 2 controllers, 2 base stations)
- Computer that fulfills the following minimum specifications:
 - Processor: Intel Core i5-4590 or AMD FX 8350, or better
 - Graphics: NVIDIA GeForce GTX 1060 or AMD Radeon RX480, or better
 - Memory: 4GB RAM or better
 - Operating system: Windows 7 SP1 or better

7 HOW WE LEARNED

The main technology that we had to learn was Unity. Unity was our IDE and 3D modeling space for the physical environment and the fishing experience. We spent a lot of time in the editor figuring out what different tools were available as well as looking through the Unity documentation to gain a better grasp of the capabilities of the game engine.

Below is a short list of Unity manpages that we found to be useful for our project.

- <https://docs.unity3d.com/Manual/index.html>
- <https://docs.unity3d.com/Manual/VROverview.html>
- <https://docs.unity3d.com/Manual/terrain-UsingTerrains.html>
- <https://docs.unity3d.com/Manual/UICanvas.html>

- <https://docs.unity3d.com/Manual/AnimationOverview.html>

While we did not end up collaborating heavily, we did meet with another Intel VR group earlier in the Capstone course. We met up a couple times to discuss strategies for ramping on Unity and some VR concepts that carried over between the two projects. The other group had signed an NDA so we could not go into specifics of their project. We were not aware of any students or professors who specialized in Unity so most if not all of our research was done on our own, online.

8 WHAT WE LEARNED

8.1 Jake

I have very little knowledge of Virtual Reality development prior to starting this project. The first couple months I worked hard to absorb as much information as I could and to figure out which tools would be best for us to use. We decided that Unity would be the most appropriate development tool because of the relatively small learning curve and the strong community behind it. Throughout the project I learned all about Unity game development and C# scripting. I learned how to use game logic to create realistic interactions and behaviors. The great part of this project and our team was that we all worked incredibly well together so everyone was able contribute to each aspect of the project. This gave us all a wide range of development experiences from environment modeling to user interaction to object scripting.

From a non-technical perspective, I learned a huge amount involving how to effectively communicate with clients and how to work efficiently on a team. I had already worked for 6 months at an internship so I had a basic understanding of how a professional workplace runs and how to work on a scrum team. For a Senior Design Project, we definitely didnt want that much forced structure but we did take ideas from it. We made sure to have weekly meetings where we talked about what we did over the last week, where we are now, what we have left to do, and what new stuff we can start working on. Basically we just made sure to touch in with each other regularly to make sure everyone was on the same page and that we were on track to finishing the project on time. Implementing these practices into our team made us much more effective and helped make sure no one was being left behind.

We also made sure to touch in with our clients on a regular basis. We had clients from both Intel and Columbia which made it a little more difficult to always find times where everyone was available but this wouldnt be uncommon in the real world so it was great preparation. Overall, our clients were incredibly helpful and usually quick to respond to questions. There were a few times when it would take us a little while to get feedback from them so we had to roll with it but thats expected in the real world. Having a scheduled meeting time each week or every other week is really important so that the

development team isn't ever diverging from the clients vision. However, it's also important to note that clients can make mistakes as they're just human so you have to be flexible as a developer and work extra hard to make sure you are understanding the client correctly. Strong communication is essential to a smooth project.

If I could do it all over again there is very little I would change. I had a strong team and great clients so working on this project was a blast. If there is one thing I would have worked harder at it would be to clarify the clients vision a little better. We were mainly on the same page but there were some things we didn't know because we didn't think to ask. Mainly this has to do with other projects Columbia was already working on. Unfortunately, we didn't know this was taking place so there was a little bit of overlap in what we were all working on. Clarifying scope is incredibly important early on. Overall though, Capstone was a consistently positive experience.

8.2 McKenna

This project was an invaluable learning experience for me. Throughout the process I have learned numerous technical and non technical skills along the way. Overall, I think it was a well rounded project and experience when considering the range of skills I learned.

Technical Skills

- Unity Game Engine Basics
- Virtual Reality development within Unity
- SteamVR development within Unity
- Unity C# scripting
- Basic physics modeling in Unity

Non-Technical Skills

- Writing software documentation
- Explaining a complex project to a less informed audience (Elevator Pitches)
- Team problem solving skills
- Project time management
- Forming and revising project goals/purposes

This project has also been a great learning experience regarding project management and working on a team of engineers. If I was to do this project again I would be much more careful when designing the specific requirements of the final product. We definitely overestimated the amount of work that we thought we would be able to complete. This is not inherently bad, as it probably pushed us to complete

more than if we had set our expectations low. But the biggest thing that I will now remember is to think realistically when designing a project.

As the project progressed we all learned the importance of working as a more cohesive team. For example, as we began to do more paired programming type work our project began to ramp up much more quickly. As a group I would say that we are all procrastinators at heart. However, as the project went on we did begin to try and finish things a little bit before the final due dates. This definitely reduces stress and results in higher quality work. As far as issues with team members carrying equal weight, the best solution for us was to simply attempt to divide up the work as evenly as possible. This seemed to work with little to no issues.

8.3 Sean

I learned a lot during the last nine months of Capstone. I learned aspects of computer science and team development that I enjoy, as well as aspects that I definitely don't enjoy. Overall I feel that I am a better engineer and developer after the senior design experience.

The most basic thing I learned is how to develop a virtual reality experience in the Unity game engine. This started with exploring how to bring a terrain and an environment to life. I learned how to mix the creative side of painting a realistic scene around the user with the technical side of render distance, performance, and

8.4 Spike

Capstone as a whole has taught me a lot and this project in particular has helped me develop a new set of skills, both technical and non-technical. I had not worked on a project of this size before but I had a great time with the team and I am happy with the product we developed.

On the technical side of things, I came into this VR project with very little experience. I had never interacted with any VR related technologies and had not had any exposure to the Unity Game Engine. Even the scripting, which was done in C#, was all new to me. With documentation and experimentation within Unity, we were able to ramp up on what we needed to know fairly quickly. We learned how to edit assets through the inspector and we were onto adding scripts to start creating our outdoor experience after a short time.

The nontechnical skills I picked up were as, if not more, relevant than the technical skills I learned during the course of this project. Communication was the most important skill that I worked on. We were always in contact with various people in the Capstone process. Our group was really close and worked really well together because of how easy it was to communicate to each other. We had to manage communication with both of our clients as well as with our professors and TA. We had some struggles with communicating with our sponsors but everything turned out to be all right

after clarifying some ideas and adapting our approach to the project. We spent a lot of time writing documentation for this class. While it was repetitive at points, it kept us in line with the work we were putting out and ensured clarity in our development.

Capstone was a unique experience that really emphasized the importance of group cohesion. I really enjoy working on projects as a group. It lets the group members bounce ideas off each other and work cooperatively towards a single goal. Everyone is going to have different strengths and weaknesses, and group projects allow for individuals to play to their strengths. I do not think I would change anything if I were to do this all over again. Our group worked really well together, we had fun and we met our requirements. The only slight issue we really ran into was a misunderstanding of project requirements with our client. We had to shift the project focus a little to accommodate for the changes but it was manageable.

9 APPENDIX

9.1 Essential Code Listings

9.1.1 *FishingLineLogic.cs*

This C# script handles the fishing line, including the reeling in and out motion. It makes use of the UltimateRopeEditor package from the Unity asset store.

```
using UnityEngine;
using System;
using NewtonVR;

namespace fishingLineLogic
{
    public class FishingLineLogic : MonoBehaviour
    {
        public UltimateRope Rope;
        public Rigidbody FishingRod;
        public float castingSpeed;

        static float m_fRopeExtension;

        void Start()
        {
            m_fRopeExtension = Rope != null ? Rope.m_fCurrentExtension :
                0.0f;
        }

        // Called every frame
        void Update()
        {
            bool casting = false;
            bool reelHand = false; // True is right, false is left

            int mag = (int)Math.Round(FishingRod.velocity.magnitude);
            if (mag > 1)
            {
```

```

        castingSpeed = mag * 2;
    }

else
{
    if (castingSpeed > 0)
    {
        castingSpeed -= 0.1f;
    }
}

if (castingSpeed > 0)
{
    Debug.Log(castingSpeed);
}

// The reel hand is set to the opposite hand of the one that is
→ holding the fishing rod.

// The user is casting, when the touchpad of the hand that is
→ holding the rod is being pressed.

if (NVRPlayer.Instance.LeftHand.IsInteracting)
{
    casting =
        → NVRPlayer.Instance.LeftHand.Inputs[NVRButtons.Touchpad].IsPressed;
    reelHand = true;
}
else if (NVRPlayer.Instance.RightHand.IsInteracting)
{
    casting =
        → NVRPlayer.Instance.RightHand.Inputs[NVRButtons.Touchpad].IsPressed;
    reelHand = false;
}

// If the user is casting, set the extension speed

if (casting)
{

```

```

        m_fRopeExtension += Time.deltaTime * castingSpeed;
    }

// Find the reel in speed by getting the position of the users
→ thumb on the touchpad.

if
    → (NVRPlayer.Instance.LeftHand.Inputs[NVRButtons.Touchpad].IsTouched
    → && reelHand == false)
{
    Vector2 leftAxis =
        → NVRPlayer.Instance.LeftHand.Inputs[NVRButtons.Touchpad].Axis;
    reelIn(leftAxis);
}

else if
    → (NVRPlayer.Instance.RightHand.Inputs[NVRButtons.Touchpad].IsTouched
    → && reelHand == true)
{
    Vector2 rightAxis =
        → NVRPlayer.Instance.RightHand.Inputs[NVRButtons.Touchpad].Axis;

    reelIn(rightAxis);
}

// Extend the rope

if (Rope != null)
{
    m_fRopeExtension = Mathf.Clamp(m_fRopeExtension, 0.0f,
        → Rope.ExtensibleLength);

    → Rope.ExtendRope(UltimateRope.ERopeExtensionMode.LinearExtensionIncrement
    → m_fRopeExtension - Rope.m_fCurrentExtension);
}
}
```

```

// This function sets the speed to reel in the rope, based on the
→ user's position on the touchpad.

public static void reelIn(Vector2 axis)
{
    float reelingSpeed;
    if (axis.y > -1 & axis.y < -0.33)
    {
        reelingSpeed = 0.25f;
    }
    else if (axis.y > -0.33 && axis.y < 0.33)
    {
        reelingSpeed = 0.5f;
    }
    else
    {
        reelingSpeed = 0.75f;
    }
    m_fRopeExtension -= Time.deltaTime * reelingSpeed;
}
}
}

```

9.1.2 FishLogic.cs

This script handles the motion of the fish(es), including the catching interaction of the fish.

```

using NewtonVR;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class FishLogic : MonoBehaviour
{

    public GameObject hook;
    public GameObject lineEnd;
    public bool caught = false;
}

```

```

public bool userIsFishing = false;
public GameObject FishParent;
public bool fishDead = false;

Vector3 initialPosition;
FishLogic[] fishList;
bool otherFishCaught;
// Use this for initialization
void Start()
{
    fishList = FishParent.GetComponentsInChildren<FishLogic>();
    initialPosition = transform.position;
}

// Update is called once per frame
void Update()
{
    // Check if another fish is currently on the hook
    otherFishCaught = false;
    foreach (FishLogic fish in fishList)
    {
        if (fish.caught == true)
        {
            otherFishCaught = true;
        }
    }

    // Check if fish has not been caught, the user is fishing (hook in
    → water),
    // another fish is not currently on the hook, and the fish isn't
    → dead
    if (!caught && userIsFishing && !fishDead && !otherFishCaught)
    {
        // If hook is 50 units from fish, fish will begin to follow
    }
}

```

```

if (Vector3.Distance(hook.transform.position,
    → this.transform.position) < 50)
{
    Vector3 direction = hook.transform.position -
        → this.transform.position;

    direction.y = 0;
    this.transform.rotation =
        → Quaternion.Slerp(this.transform.rotation,
        → Quaternion.LookRotation(direction), .2f *
        → Time.deltaTime);

    if (direction.magnitude > 5)
    {
        this.transform.Translate(0, 0, 0.1f);
    }

// When hook is close enough, lock onto hook, with a
→ character joint
// if another fish is not currently not on the hook.

else if (direction.magnitude <= 5 && !otherFishCaught)
{
    caught = true;
    this.gameObject.AddComponent<CharacterJoint>();
    CharacterJoint joint =
        → this.GetComponent<CharacterJoint>();
    joint.autoConfigureConnectedAnchor = false;
    joint.connectedAnchor = new Vector3(0, 0, 3f);

    Rigidbody lineEndRigid =
        → lineEnd.GetComponent<Rigidbody>();

    this.transform.position = lineEnd.transform.position;
    joint.connectedBody = hook.GetComponent<Rigidbody>();
}

```

```

Rigidbody fishRigid = this.GetComponent< Rigidbody >();
fishRigid.isKinematic = false;


// Move every other fish back to their initial position
foreach (FishLogic fish in fishList)
{
    if (fish.gameObject.name != this.name &&
        ↳ !fish.fishDead)
    {
        fish.gameObject.transform.position =
            ↳ fish.initialPosition;
    }
}

// If there is a character joint on the fish, it is currently
↳ attatched to the hook.
// Vibrate the controller, while it is.

if (this.GetComponent< CharacterJoint >())
{
    if (NVRPlayer.Instance.LeftHand.IsInteracting)
    {
        NVRPlayer.Instance.LeftHand.TriggerHapticPulse(1500,
            ↳ NVRButtons.Touchpad);
    }
    else if (NVRPlayer.Instance.RightHand.IsInteracting)
    {
        NVRPlayer.Instance.RightHand.TriggerHapticPulse(1500,
            ↳ NVRButtons.Touchpad);
    }
    return;
}
}

```

{}

9.2 Images

Below are some images that show the current state of our virtual reality experience.

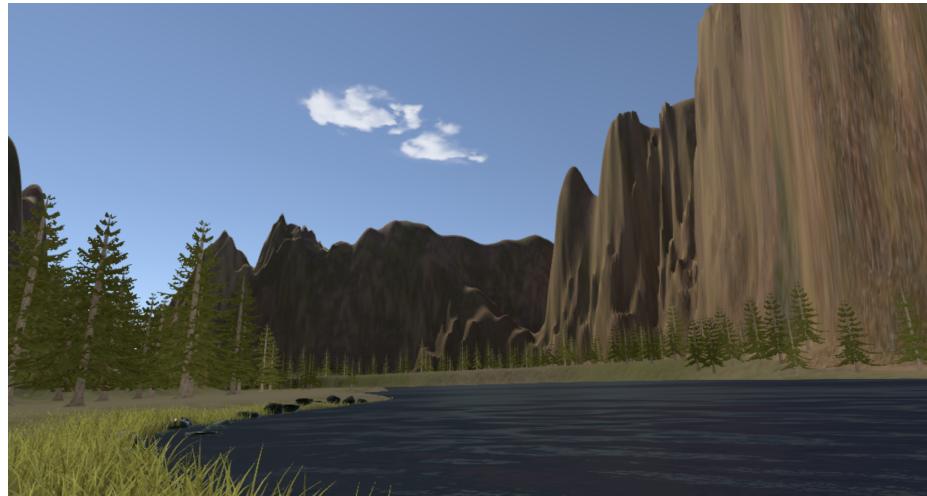


Fig. 1: Image of our landscape



Fig. 2: Fishing rod and line with fish on it



Fig. 3: Campsite with placeholder gear information