EmbarkVR: Outdoor Virtual Reality Experience
CS Senior Capstone
Design Document

Jake Jeffreys, McKenna Jones, Spike Madden, Sean Marty

November 17th, 2016

Abstract

Abstract Goes Here

CONTENTS

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 activity available will involve fly fishing in one of the rivers within the environment. Users will also have the ability to interact with virtual Columbia products while in the experience and 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 to 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 headset produced by HTC
- Base Stations: These allow the Vive to track the movement and location of the wands and headset.
- Wands: Controllers that are used with the HTC 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 scences.
- GitHub: Web-based Git repository hosting service
- Git: version control system used for software development

3 Project Context

3.1 Hardware

- Laptop Computers 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 is 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

4.1 Design stakeholders

4.1.1 Intel

Intel is working with Columbia Sportwear to help them meet their needs when it comes to this Outdoor Simulation Project. 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 Sportwear is their fishing apparel. Specifically, the Performace 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 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 with out any previous virtual reality or fly fishing experience. We are making the 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 techical performance, and overall technical design considerations are 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 real 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 responsivess.

4.2.3 Columbia Sponsor (Tim Devlin)

The Columbia Sportwear sponsor of the project, Tim Devlin, is concerened 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 gude the design process of a higher level compared to the Intel Sponsor view.

4.3 Design viewpoints

4.3.1 Context viewpoint

This context is related to user interaction with the application in its environment. It provides a ”black box” view of the project which can be useful from the perspective of developers.

- Design Concern: The main concern will come from users of the VR application. Users will only be able to see the outside of the application and their ability to interact seemlessly is essential. Users do not want to experience performance lag or errors in functionality.
- Analytical Methods: The design will be evaluated during user testing on the basis on immersion, enjoyment, and possibly influence over customers decisions. After users test the application we will ask for feedback related their experience.

- 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 a lot of factors that can influence it.

4.3.2 Composition viewpoint

[Composition viewpoint describes the way the design subject is (recursively) structured into constituent parts and establishes the roles of those parts.]

- Design Concern:
- Analytical Methods
- Rationale

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.]

- Design Concern:
- Analytical Methods
- Rationale

4.3.4 Interface viewpoint

[Interface viewpoint provides information designers, programmers, and testers the means to know how to correctly use the services provided by a design subject. This description includes the details of external and internal interfaces not provided in the SRS.]

- Design Concern:
- Analytical Methods
- Rationale

5 APPROACH

5.1 Static Environment

(terrain, static objects, Columbia gear assets)

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. The 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 one of the open-source animated water shadings. A similar technique can be used to create movement of clouds in the sky.

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 Source attached to objects. Those sounds and audio clips can be found in any open-source audio library and easily imported into Unity. 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

(user interaction with gear, Columbia gear info)

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.

The hands models 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 top 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. The next step is to allow interaction between the controller and the fishing rod. To pick up objects in the environment the user will use the trigger on the controller. This will be implimented by first monitoring for the trigger when the controller is near a GameObject. When we detect and instance of this, the objects position can then be set to the same position as the Vive controller.

Creating ropes and cables in Unity is non-trivial. The prefered method is to use physics joints. Physics joints, and specifically many hinge joints can be used to create the fishing line. The hinge joint game object 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. 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 controller 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.