iCreate - Generative Design in Virtual Reality Progress Report Document

Hannah Solorzano
Software Developer

Rhea Mae Edwards
Software Developer

Nabeel Shariff
Software Developer

CS 461 - Group 61
Fall 2017

Abstract

The purpose of this document is to explain an overall progression of what the Generative Design in Virtual Reality (Group 61) has done and experienced throughout the Fall 2017 academic term at Oregon State University through writing. To accompany this written document is a slide presentation visually and audibly explaining similar contact that is also explained in this document. There will be struggles, achievements, and human interactions that will be described. Along with a general chronological format that will represent what and when a task has been done or have been experienced.

TABLE OF CONTENTS

1 Project Overview

The outcome of this Oregon State University Computer Science senior capstone project is a virtual reality program that allows the user to utilize generative design to develop complex architectural structure.

2 Problem Definition

The virtual reality (VR) application will utilize a virtual reality headset with input from the user via a controller or gesture recognition software. The VR headset will be used to look around in virtual space while the controllers or gesture recognition software will be used by the user to draw sketches. The application will also need to utilize the GPU in a computer to both run the VR application and render 3D objects in the virtual space. Additionally, the 3D modeling will be based on generative design techniques, and the assembly of the complex 3D designs will utilize mathematical equations and algorithms to derive the appropriate structure of the design.

3 Proposed Design and Solutions

3.1 Game Engine

Unity [1], a game engine known for it's easy learning curve offers several tools for programming simulations and video games for platforms, computers, and mobile phones. It utilizes the C# programming language whose classes and object oriented items allow for the easier creation of 3D items and interface. Unity also offers an abundance of libraries and tutorials for virtual reality as there is a large user base who share their 3D objects and libraries so that others may use those items in their own project.

After weighing the pros and cons of different engines, it was decided that Unity would be the best tool for us to develop the iCreate program. With it's large user base, numerous libraries available, and easy learning curve, this tool would be the easiest engine to use for development.

3.2 Graphics Card

The graphics processing unit (GPU) is required to effortlessly render the constantly updating program. As such, the minimum requirements are at least a GeForce GTX 970 [2] or an AMD Radeon R9 290 [3]. These GPU's are not very powerful in terms of their rendering power, but they are still capable of running a VR program. In addition to a strong GPU, it should also be noted that the computer needs at least a i5 core central processing unit (CPU), 8GB of RAM, and a 1.3 HDMI port. The GeForce GTX 970 graphics card is the least powerful GPU that is still capable to run VR programs. It is affordable and easy to use.

3.3 User Controls

This technology is how the user controls how they manipulate the 3D objects in the virtual space. Since this requires an extreme amount of precision, the team has decided that the best type of controller, would be bare hands. We found this to be the most precise of all the types of controllers, and there is no additional costs to including them.

3.4 Distribution Method

Once the VR application is developed, it has to find its way to the user via some channel of distribution. A distribution method is the way the user can access and install the VR application on their system. The main criteria for choosing the appropriate distribution method has to take into account the simplest and most convenient way for the user to access and install the iCreate VR application. It should also be compatible for the chosen operating system, hardware, and software chosen for development.

With these criteria, the team has decided to distribute iCreate via an executable file, available for download directly from the iCreate website. For now, the VR application will be distributed for free, so there is no need to go through a 3rd party channel at the moment. In the future, should the team decide to distribute iCreate more aggressively and for profit, the Steam Store will serve as a good backup option.

3.5 Operating System

Apart from the VR headset's propriety operating system (OS), the VR application also needs to run on the OS that is running on the computer itself. The OS manages the resources needed by the headset and VR software to interact with each other.The appropriate OS chosen for iCreate's development should give the team access to the chosen software tools and should be highly compatible with those tools. Additionally, the OS shouldn't hinder the performance of the VR application, and should also be compatible with the headset and its proprietary software.

The team has decided to develop the iCreate VR application on the Microsoft Windows operating system because it offers the most compatibly and support for the software tools chosen by the team, and also allows for the most efficient and effective development with a great deal of documentation available for Windows. Additionally, Windows is also the most commonly used OS for VR application use, and thus will give the team access to the widest share of users.

3.6 User Interface

The team has chosen to use diegetic UI as the main UI for the iCreate VR application. Diegetic UI can be used to attach small menus with sliders and buttons on the user's controllers, allowing the user to comfortably have their tools within arm's reach. Additionally, to relay notifications directly to the user, the team may use spatial UI, but the main UI will still be diegetic.

3.7 Connective Software

We chose to use the connective software, Leap Motion, as our initial main software to connect our main virtual program since it allows its users to use their bare hands when maneuvering through a virtual reality program. Allowing this type of availability for our user with generative designing will be a very convenient addition, and allow our users to experience a more realistic building of a virtual physical structural design with their own hands in virtual reality.

3.8 Programming Language

We chose to use C# as our main programming language because mainly with our initial chosen game engine, Unity, for our group to code in, C# is the more widely use language to code in. Many of the tutorials and examples when coding in Unity are in C# , along with C# begin the most often recommended programming language when creating games through the Unity game engine.

3.9 Headsets

The HTC Vive headset setup will be used on this project because its play area specifics and available resources we have through the internet and personal user experiences in using the device. With our virtual reality program in relation with generative design, using the HTC Vive provides a great amount of physical space for a user to move and design their elements within the program, and hardware specifics that allow a user to get a maximum virtual reality experience.

4 WEEKLY OVERVIEW

	Summary	Problems	Solutions
Week 1	The class considered which capstone team they might want to work on		
Week 2	Capstone team was formed; rough draft of problem statement began. We met with the client who went over the general description of the project. We were assigned a small program that would create a roman aqueduct.	The team did not know how to do the roman aqueduct as none of us have used Unity	The TA gave us resources to use that would help us learn Unity
Week 3	The team members were confused on how to advance on the sample program. We were assigned to produce a pseudocode that would explain how the program might work. The website for the project has been created.	The team struggled with creating the pseudocode for the program. Our client aided us in a meeting by providing more information and setting us on the right path.	After the meeting with the client, the team was able to finish the pseudocode and begin working on the actual implementation of the program.
Week 4	The team completed the pseudocode and submitted it to the client for approval. The general layout of the website was completed. The final draft of the problem statement		

	Summary	Problems	Solutions
Week 5	The team is working on completing the final draft of the Software Requirements document. The program for the client was only half completed.		
Week 6	We met with the client to go over additional features/requirements for the program. With this information, we will begin writing the System Requirements Document (SRS)		
Week 7	The SRS document was rejected by the client so the team needs to redo the document to satisfy the clients' specifications.	The team got confused as to how to redo the document correctly.	Our TA helped us with specific instructions on what we could improve on. We added more details and resubmitted the SRS.
Week 8	The team began working on the Tech Review document and dividing up which technologies each of us will cover.		
Week 9	The team submitted the final draft of the Tech Review paper.		
Week 10	Worked on Design Document and Progress Report.		Finished the Design document and submitted it on time.

5 RETROSPECTIVE TABLE

Positives	Deltas	Actions
There are plenty of developmental tools available for us to use on this capstone project	The team needs to evaluate which one is best suited for this project.	Research and sample programs must be completed in order to determine which tool we will end up using.
Communication amongst team members is great.	Communication with the client is not so great.	We will be more proactive on asking better questions during the meetings and visiting our client's office hours to get help or clarification.
The team members were able to acquire hardware that will allow us to begin development.	Only two team members possess that technology.	We will get the third member access to the technology.
The team members are learning a lot about different virtual reality tools and programs.	We need to decide on how we want to structure our avionics code.	