HTC Vive Multi-Actor Framework

Design Document

Team Number: 04

Client: Nir Keren

Advisor: Nir Keren

Team Member/Roles:

Drew Buchta - Communications Lead

Tom Kiss - Team Lead

Marcus Eidahl - Design Co-Lead

Robert Slezak - Design Co-Lead

Nicholas Boos - Co-Webmaster

John Heiling - Co-Webmaster

Team Email: may1704@iastate.edu

Team Website: http://may1704.sd.ece.iastate.edu

Revised: 11/8/2016

Version 1

Contents

1 Introduction

1.1 Project statement

The project is to design a simulation for multiple people in a virtual reality environment (VRE), using the HTC Vive Head Mounted Display (HMD). The application will host training simulations while recording user actions for research and training purposes. It will include voice communication, multi-user interaction within an environment, multiple movement options, decision matrices (Figure 1), and authoring of said decision matrices.

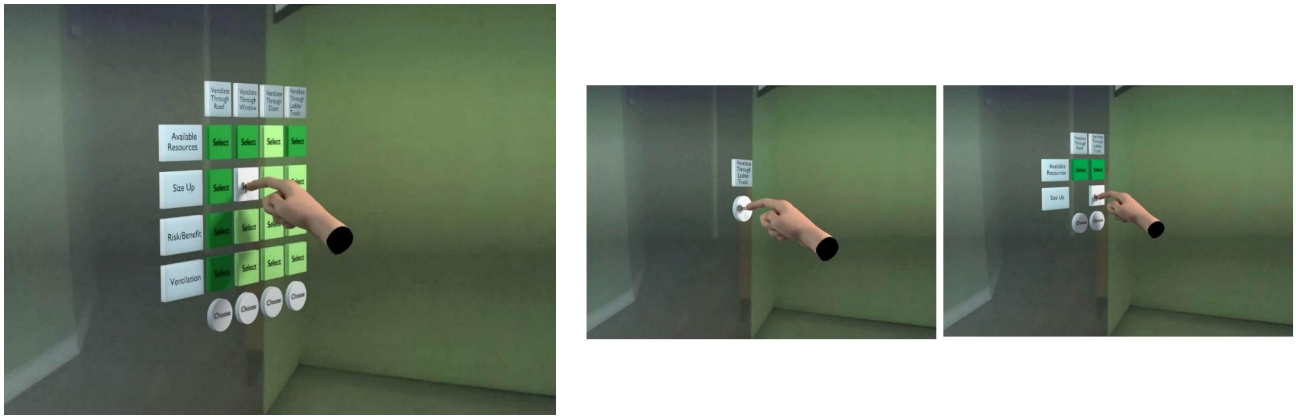

Figure 1

1.2 Purpose

Currently, the Vive is only designed for a single person in an environment. There are some instances of there being multiple people in the same environment, but not in the capacity that our client wants.

This platform will allow multiple people (e.g. astronauts, soldiers, firefighters, and law enforcement personnel) to interact with each other in a VRE. Along with this basic interaction, users will be able to engage in group decision making via voice communication. This would provide a new level of training to emergency personnel that will make a large impact on their actions in the real world. This platform can also host research simulations that research teams can have subjects go through and draw critical data about their decision making process.

This VR platform provides a more realistic feeling environment to the user and allows them to make decisions that more closely match how they will act in a real situation.

1.3 Goals

- Multi-user
    - Voice Communication, public or specific to targeted user(s)
    - Interaction with other users through game environment
- Decision Matrices
    - Authoring system

    - Video, audio, or text content
    - Randomization per simulation instance
    - Logging of interactions and choices made
    - Functionality for full or diffusive release of content
- Playback system
    - Recording user actions in simulation for real-time playback from alternate perspective
    - Bookmarking of points in simulation, ability to skip through bookmarks
    - Speed up and slow down time
- Multiple forms of locomotion through environment

2 Deliverables

- Application with functionality achieving above goals and following the below functional and nonfunctional requirements
- Website highlighting the project
- Documentation for how to interact with the application

3 Design

At this point in our project we can have multiple users interacting with each other and decision matrices in a virtual reality environment across a local network. Our approach to many of the problems given to us by the client is to start with something simple and build on that to accomplish the end goal. We have used this strategy with movement, decision matrices, playback, and networking. We have tried many forms of movement in VR including position to velocity, touchpad movement, teleportation, climbing, and walking in place. In the end we found that P2V and teleportation are best for our project. The client has a very specific idea on how the decision matrices should work so we have been building it based on frequent feedback from our client. Networking is a big portion of this project so for a start we have just implemented multi-user functionality within LANs. We are now working on expanding what we have to work over the internet so users can be anywhere in the world and interact with each other. We have also just begun work on the logging and playback portion of this project. Currently we can log user input to an XML file.

3.1 System specifications

- Locomotion
    - Position to velocity movement
    - Teleporting
    - More forms of movement if necessary
    - Ability to step up objects
- Network framework
    - Smooth interaction between users
    - Voice chat
    - Interactable objects synced between users
    - Ability to connect to users anywhere in the world

- Decision matrices
    - Authoring system
    - Randomized location of choices
    - Logging of interactions and choices made
    - Choices described with either text, audio, or video
    - Functionality for full or diffusive release of content
- Playback system
    - Record users actions in order to be able to play them back in real time
    - Bookmark of points in simulation with ability to skip to them
    - Varying speeds of replay

3.1.1 Non-functional

- Locomotion
    - Scalability of movement speeds based on the size of the scene
    - Movement is user friendly
- Network framework
    - Stable and reliable network connection with as few disconnects as possible
    - Need users movements to appear smooth to other users for optimal experience
    - Voice chat sounds clear
- Decision matrices
    - Available whenever the user wants them in a scene

3.1.2 Functional

- Locomotion
    - Position to velocity movement which is movement that is controlled based on how far the user is outside of the center area. The user will then continue moving in that direction.
    - Teleportation movement where the user points to a location and clicks a button and that is the new location of the user
    - Ability to step over objects that are of realistic height to step over
- Network Framework
    - Voice chat where the user is able to select which users they can communicate with
    - Interactable objects synced between users so that it feels like they are in the same place together
    - Ability to connect to players anywhere in the world so that these simulations can be ran between users no matter where they are located
- Decision matrices
    - Authoring system for users to create decision matrices outside of the scene and import them into the scene later
    - Choices in the matrix are shuffled to appear in different locations each time so there is no user bias
    - Logging of interactions and choices made, so that researchers can easily look at the results from a simulation
- Playback system
    - Playing back user's actions in real time, so that key decision points can be further evaluated

- Ability to skip to bookmarked times makes the replay system more user friendly

3.2 PROPOSED DESIGN/METHOD

Our team has decided to implement the movements necessary for a good user experience with variable speeds for movement. We have integrated this with our network framework. With our network framework we decided to use linear interpolation to make sure that movements between users appear as smooth as possible. We plan on adding voice chat so that users can easily communicate with each other. Along with this we have implemented user interactable objects that sync between users. Decision matrices are being worked on in order to give people choices and record their actions in a simulation scene. Using recorded actions a playback system will be implemented in order to rewatch simulations.

3.3 DESIGN ANALYSIS -

So far, we have implemented various types of movement, a decision matrix authoring and viewing system, smooth networking between multiple users, and a user input logging system. The features we have implemented so far are working as intended and are a base for us to build on to complete this project. We went through many types of movement and have found the ones that work the best for our situation. We have had many iterations of the decision matrix authoring and viewing system, but we now know what the client wants and can work on completing it. LAN multi-actor is working well now after smoothing changes were made to the asset we used. Now multi-actor functionality needs to be implemented across the internet.

4 Testing/Development

4.1 INTERFACE SPECIFICATIONS

We are using the HTC Vive for the VR application. To develop for this hardware we are using the Unity 3D game engine.

4.2 HARDWARE/SOFTWARE

The only hardware this project is intended to use is the HTC Vive. So to test our solution we also use the HTC Vive headset and Unity3D.

4.2 PROCESS

- Locomotion
  - Locomotion was tested by trial and error. Locomotion was primarily evaluated by ease of movement, precision of movement, and how intuitive it was to use.
- Networking
  - We tested networking by using two HTC Vives and verifying that the input and actions caused by one client affected the other client.
- Decision Matrix

    - We tested the decision matrix by using the HTC Vive headset in VR to verify that the decision matrix could accurately display text, video, and audio.
- Playback System
    - To test the playback system we have to verify that the scene that is played back after recording is the same.

5 Results

The project that we are working on is relatively unlike anything that currently exists. The closest thing would a program fairly similar in the C6 which has elements that are slightly resemblant to things that our client wants us to implement. With that being said, it is difficult for us to create any sort of testing as we have little to test against, and the only way we know if something is correct is by getting approval from our client. Any sort of testing that we have done has mostly been trial and error testing to see if the things we are currently working on do what we want them to. Below is a list of the main things we have worked on along with a short description of whether or not they are complete.

Networking - The core of this has been successfully tested and completed There are few things remaining but they mainly concern with how people connect to each other, not whether or not they can connect.

Locomotion - This has been tested and successfully completed.

Input logging - This is a work in progress as it is one of the more extensive things that we are working on. What testing we have done has been successful.

Decision matrices - This is also a work in progress as it is another rather extensive part of our project. There has been a fair amount of trial and error with this; the concept of decision matrices is one that does exist outside the scope of our project, so we are able to somewhat test it in regards to how they are generally expected to work, but ultimately, in the context of VR, this has never really, to our knowledge, been done, making it still kind of difficult to test. Our client has played a fairly large in role in ironing out exactly how these should work.

For the most part, our method of simple trial and error has worked quite well and we intend to continue doing it in this fashion. For the things that are somewhat unique to our project, we will consult our client on how exactly he expects them to work.

6 Conclusions

Networking between two actors in VRE and locomotion have been tested and successfully completed with decision matrices and input logging currently being worked on. We have four main goals to accomplish: multi-user functionality, decision matrices functionality, playback system functionality, and multiple forms of locomotion.

Multi-user functionality requires multiple actors in a single VRE and interact and communicate together. Decision matrices functionality must have multimedia playback and text, an authoring system to create decision matrices, randomized location of decisions, decision and interaction logging, and have full or diffusive content. The playback system must feature the ability to record multiple users' actions and replay them in real time from different perspectives, bookmark certain points in the scenario allowing users to skip to certain events like decision making, and speed up or slow down replays. And finally, for locomotion, users must be able to teleport and use touchpad and position to velocity to move around in the VRE.

To accomplish these four main goals, we divided ourselves into groups of two or work alone and selected which component to complete first and perform methods of trial and error. Once the main components of the project are completed, we decided to merge our work together and fine tune the components to get a working prototype. We believe this is the best solution for this project because we had little to no experience with virtual reality and Unity game engine and dividing the project into important components allowed us to focus on one particular component for development before moving onto the next component or merging the project. This solution also allowed us to share knowledge about the project and tools in order to make progress.

7 References

Our scope document given by our client, Nir Keren.

8 Appendices

The following tools and assets are used in this project:

HTC Vive VR Headset

https://www.vive.com/us/

SteamVR API By Valve Corporation

http://store.steampowered.com/steamvr

https://www.assetstore.unity3d.com/en/#!/content/32647

SteamVR Network Essentials By LRX GmbH

https://www.assetstore.unity3d.com/en/#!/content/63969

Unity Game Engine

https://unity3d.com/