HTC Vive Multi-Actor Framework

Project Plan

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 2

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.

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

There exist two platforms that are widely available for which previous VR programming has been done. One is Unity3D and the other is Unreal Engine.

3.1 Previous work/literature

A program very much like this one, does already exist for the C6. This program was also made for our client, by people employed by ISU. We are working closely with this team.

3.2 Proposed System Block diagram

See Appendix A for flowchart.

3.3 Assessment of Proposed methods

Valve Corporation (the company that collaborated with HTC to provide software) has an asset for Unity called SteamVR which makes programming for the Vive relatively straight-forward. So, we will use Unity3D for all intents and purposes.

3.4 Validation

We will test them against the existing simulation in the C6.

4 Project Requirements/Specifications

4.1 Functional

Having multiple people in the same VRE, this will include all the networking features previously described.

Movement, this will include the various types of locomotion as well as the ability to select various types.

Decision matrices and input recording, this will include the two different types of matrices as well as the ability record all interactions with these matrices.

The playback of the simulation, this will include the ability to view from any angle, go to bookmarks, as well as pausing fast-forwarding and rewinding at various speeds.

4.2 Non-functional

Representation of people in the VRE, the client requires this, but it isn't necessary for the simulation to function properly.

Communication between people in the VRE, again, the client requires this, but it isn't necessary for the simulation to function properly.

Recording feedback about scene, also required by client, but not necessary for the simulation to run.

5 Challenges

There are two main challenges that we foresee with this project. The first is to do with the networking involved with the multiple people in the same VRE. The challenge here is that there is little support for this type of interaction. That which does exist is mostly proprietary and build from the ground up. Currently, we are using a Unity plug-in that helps in facilitating the connection between two people, however as it is right now, it is very user unfriendly and only works over a Local Area Network. The second challenge is based around the playback system. Again, there is little to no support for this. Currently, we are not quite sure how we will do this. We are exploring a method of logging all interactions with the environment into an XML file.

6 Timeline

See Gantt chart in Appendix B.

6.1 First Semester

Our plan for the first semester is to have a working simulation, albeit slightly rough around the edges. This means that all the core functionality, including multiple people in the same VRE at one time, movement around the VRE, the decision matrices, and possibly the recording of inputs.

6.2 Second Semester

In second semester, we plan to "polish" the simulation. This would include such things as the representation of people in the VRE to others. We also plan to do the communication between two people in this semester as well as finishing (if not already done) the recording of inputs from the users, the feedback recording from the user about the scene they just interacted with, and finally the playback of the simulation.

7 Conclusions

Our main goal is to make it possible for multiple people to interact with each other in the same VRE. Beyond that however, we need to make this interaction as smooth as possible, and the way in which the connection is made must be simple as well. To represent people in the VRE we will most likely use some sort of avatar. To facilitate communication between two people we will most likely use some sort of third party software that we will control through Unity. We will have two main types of locomotion (P2V and Teleportation) that will be able to be switched between by the user. We will have the two different types of matrices (full release and diffusive) that all interactions the user makes with will be recorded to an XML file for use in the playback of the simulation. Lastly, we will have an interface that the user will be able to interact with and report feedback about the scene they just interacted with.

8 References

The Scope document given to us by our client.

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

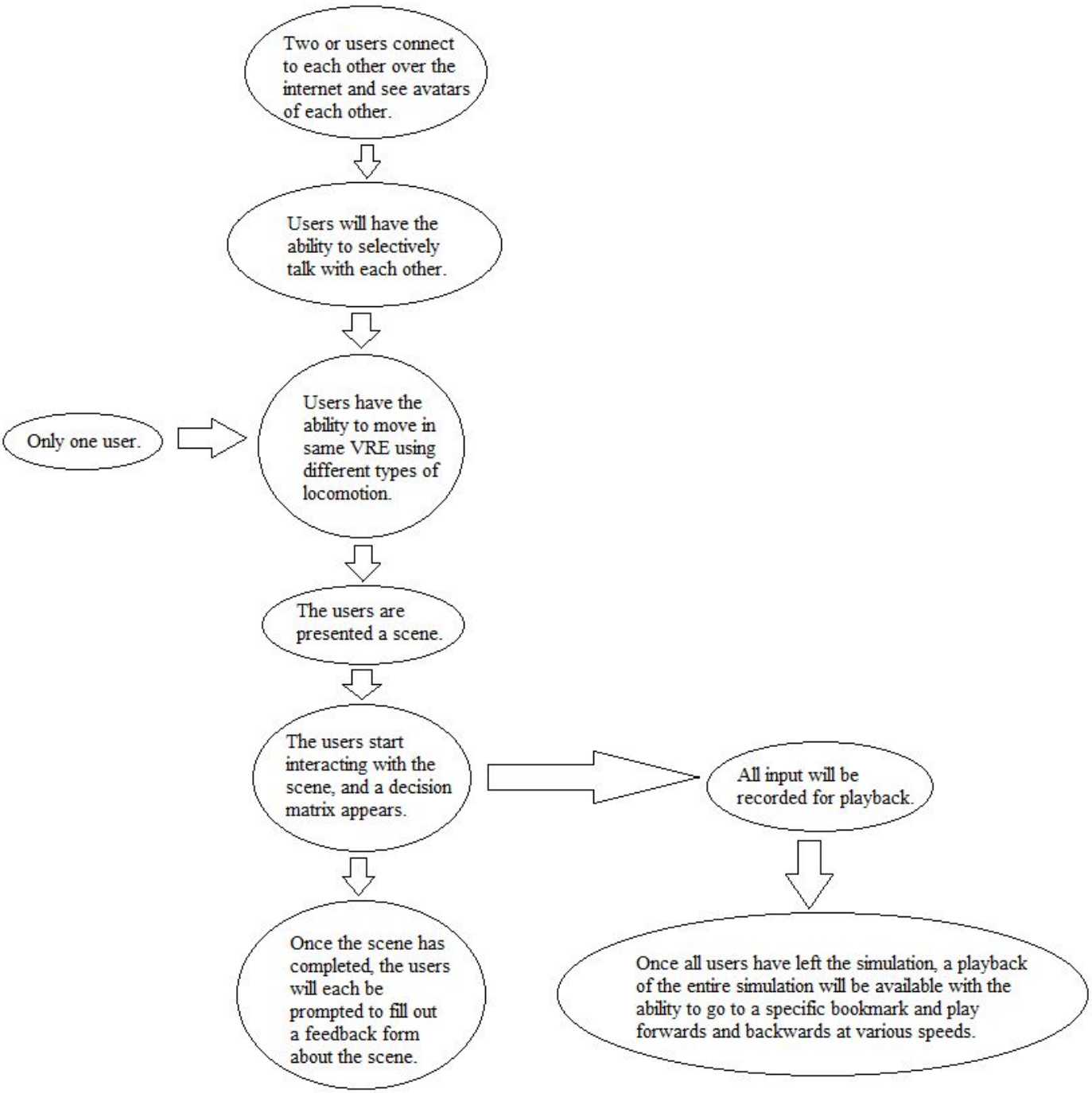
Unity Game Engine

https://unity3d.com/

9 Appendices

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. You may also include your Gantt chart over here.

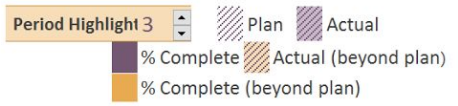
Appendix A



Appendix B

Virtual Reality Environments

r



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Locomotion	7	2	7	2	75%
Scene Building	9	1			
Scene Triggers	10	1			
Smooth Networking	7	3	7		50%
Create Server	10	2			
Editor Scripts	7	1	7	1	100%
Show and Hide Matrices	7	1	7	1	50%
Reveal Content	7	1	7	1	50%
Audio and Video Files	7	1	7	1	25%
Pre-Determine Dimensions	8	1			
Edit Canvases Individually	8	1			
Allow Randomization	8	1			
Diffusive Revealing Matrices	9	1			
Scenario Shifting	9	3			
Audio/Video/Images over Network	10	2	9		0%
Import Decision Matrix to XML File	9	1			
Export Decision Matrix from XML File	9	1			
Record User Interactions with Matrix in XML File	7	2	7		50%
Create Independent Bookmarks in Simulation and Store in XML File	9	1			
Log All User Actions in Simulation (not just inputs)	7	2	7		50%
Replay Environment Setup (moving camera and controls)	12	2			
Playback Interface	12	2