SIU-CAVE

Cave Automatic Virtual Environment

Project Design

Version 1.0 (DRAFT)

Prepared for

Dr. Christos Mousas

By

JBU

on

March 2nd, 2018

TABLE OF CONTENTS

1. INTRODUCTION

This document provides a detailed description of the planned implementation of the CAVE. This document in divided into two main sections: architecture design and system design.

The first section of this document will describe the architecture design of the CAVE. This will cover the hardware and software components of the project. For the hardware section, this document will cover the setup of the CAVE and how everything is arranged in the room. A diagram will be included for further clarification. The software section of this document will provide detailed information about Unity and MiddleVR. These two pieces of software will work together in order to create an application that can be run on multiple screens. A diagram will be provided to explain how MiddleVR will be run on all three computers.

The second section of this document will cover the system design of the CAVE. In this section, use-cases of the CAVE will be discussed in detail along with the applications of it.

2. GENERAL OVERVIEW AND GUIDELINES

2.1 Assumptions / Constraints / Standards

Due to the limited budget we were given, there are a few constraints with the type of equipment we were able to order. The main constraint was the quality of projectors that we were able to order. Typically, the more expensive the projector is, the shorter its throw distance will be. Throw distance is the distance between the projector and the image it produces on the screen. It is very important in the design and implementation of a CAVE, because it determines how far away the projector needs to be from the screen in order to produce the desired image size. Since the projectors we ordered have longer throw distances and the room is a little on the small side, the size of the picture will not be as big as typical CAVE systems. Despite these limitations, the size of the screen will still approximately be 4 feet long by 4 feet high.

Another constraint for the design of the CAVE system is the gap in between each of the screens. With the budget we were given, we were able to buy a special type of screen material which is designed for rear projection. Rear projection allows for a projector to be behind the screen, and then the projector flips the image onto the screen. This eliminates shadows inside the CAVE. Along with the screen material, we purchased three frames to hold the screen material up. Due to the frames and their design, there is a small gap between each screen. In more high scale CAVEs, there is no gap between screens, the edges are seamless. Although we will minimize this gap as much as possible, there will inevitably be a slight gap in between screens due to the nature of the frames themselves.

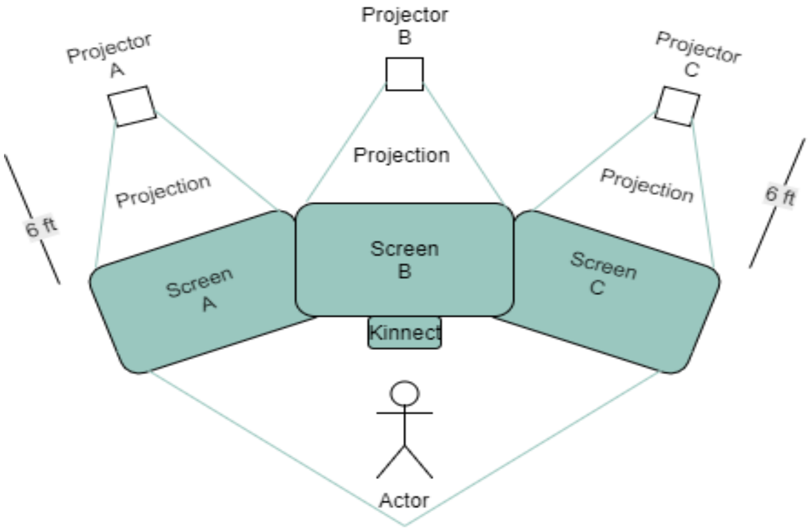
MiddleVR is a plugin for Unity that allows for the creation of VR applications. It has support for most VR systems along with a lot of interaction devices. However, it does not support every type of interaction device. For example, MiddleVR supports the Kinect, but it does not have built-in support for the Intel RealSense or the Wii remote. Because of this, we are limited with the type of interaction device we can use. Since it has built-in support for the Kinect, all we have to do is download the SDK for it and MiddleVR will automatically recognize it.

3. ARCHITECTURE DESIGN

.

3.1 Hardware Architecture

The CAVE system architecture design consist of three screens lined up forming a C shaped canvas for projection. The projectors are placed behind the screens each at a distance of six feet in order to maximize the size of the projection. The Kinect motion sensor is placed underneath the middle screen. The space in front of the screens is for the user to interact with the CAVE system. Each projector is connected to its own computer, where all three PC's are networked together using a switch, two of which acting as clients to one master. The diagram below demonstrates the architecture design of the CAVE system.



3.2 Software Architecture

-MiddleVR

In order to properly utilize our different types of hardware for development, we need a unified system manager. That is what the MiddleVR software brings to the project and helps bridge the gap between the CAVE's different elements. However it is more of a "back-end" type of software, as the actual application will be developed in the Unity game engine. As seen in Figure 1.1 and 1.2, MiddleVR is used to create a config file that is then imported into Unity with an asset package. The config file contains all of the information edited within MiddleVR, such as a device manager for inputs, viewport options for outputs, and cluster management for the networked devices. A viewport can be assigned to any output device, in our case projectors, with each viewport also being assigned cameras. Those cameras are then adjusted to a screen based on your settings in Figure 1.1, which are a representation of the real-world CAVE screens. Final adjustments are made in the cluster management section, as the PC's are assigned a projector and calibrated automatically in the network.

-Unity

In order to best showcase the CAVE's features, we believe an application needs to be interactive and localized to a user. A simple game application would work best for our needs as it accomplishes both in a very efficient manner. Unity's flexible environment as well as it's compatibility with MiddleVR made it our natural choice for development of the game application.. Our mentor, Dr. Mousas, is also very experienced in the Unity environment and can provide extra resources.

-Kinect SDK and VRPN

The Kinect SDK will allow for more fine tuning of the user's input for our application. It allows for early testing and monitoring of the input before it is implemented into the full application. The VRPN, or virtual reality peripheral network, is a set of class libraries and set of servers that can implement an interface between applications and physical devices. It "converts" data from devices into three categories; tracker, analog, and buttons. This allows for many different devices to be able to be used with our application as an input or controller. VRPN supports the use of a Wiimote device, and VRPN itself is supported by the MiddleVR config file.

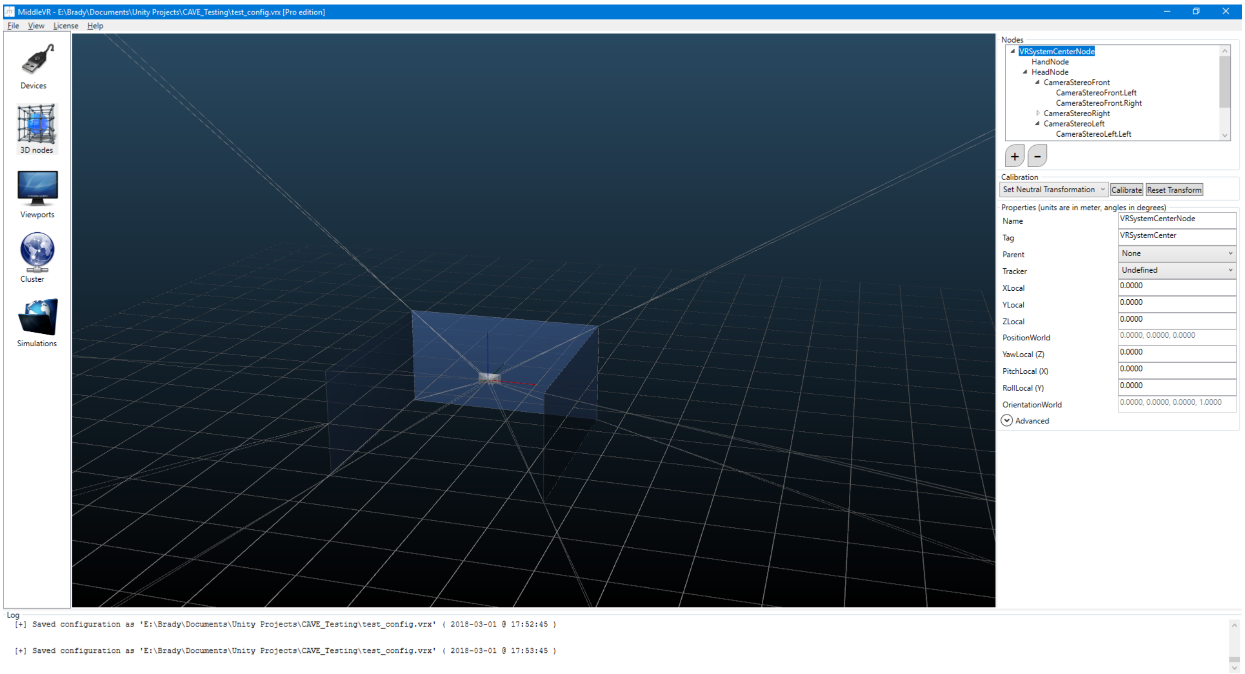
4. SYSTEM DESIGN

4.1 Use-Cases

The CAVE system was built in the most cost efficient way. The system will be a proof of concept of the high end CAVE system which has variety of applications in the field of automotive engineering and aerospace engineering and design. CAVE is also used in different simulations and trainings such as flight simulations for pilots and combat simulations for army. Video game designing is also popular applications of a CAVE system. Simply put, a CAVE is beneficial to anyone who wants an immersive experience in order to further evaluate and analyze data. The CAVE gives the user an experience unlike any other 2D screen or any other type of virtual reality system that uses head-mounted displays. This is because the user has a sense of awareness while being in the CAVE. Also, multiple people can experience the CAVE at once in more advanced systems.

5. Figures

5.



MiddleVR - E:\Brady\Documents\Unity Projects\CAVE_Testing\test_config.vrx [Pro edition]

Nodes

- VRSystemCenterNode
  - HandNode
  - HeadNode
    - CameraStereoFront
      - CameraStereoFront.Left
      - CameraStereoFront.Right
    - CameraStereoRight
    - CameraStereoLeft
      - CameraStereoLeft.Left

Calibration: Set Neutral Transformation

Properties (units are in meter, angles in degrees)

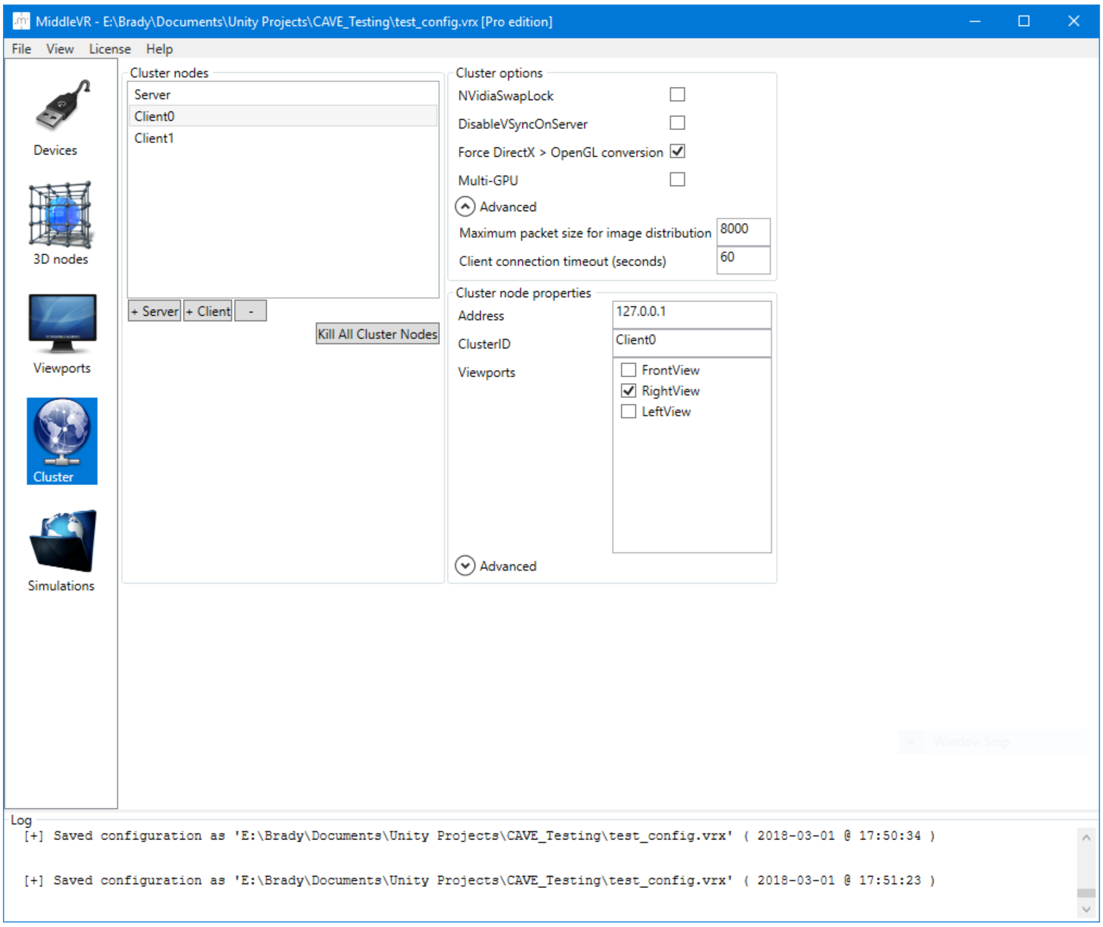
Name	VRSystemCenterNode
Tag	VRSystemCenter
Parent	None
Tracker	Undefined
XLocal	0.0000
YLocal	0.0000
ZLocal	0.0000
PositionWorld	0.0000, 0.0000, 0.0000
YawLocal (Z)	0.0000
PitchLocal (X)	0.0000
RollLocal (Y)	0.0000
OrientationWorld	0.0000, 0.0000, 0.0000, 1.0000

Log

```
[+] Saved configuration as 'E:\Brady\Documents\Unity Projects\CAVE_Testing\test_config.vrx' ( 2018-03-01 @ 17:52:45 )

[+] Saved configuration as 'E:\Brady\Documents\Unity Projects\CAVE_Testing\test_config.vrx' ( 2018-03-01 @ 17:53:45 )
```

Figure 1.1: MiddleVR’s Config. file editor



MiddleVR - E:\Brady\Documents\Unity Projects\CAVE_Testing\test_config.vrx [Pro edition]

Cluster nodes

- Server
- Client0
- Client1

Cluster options

- NVidiaSwapLock □
- DisableVSyncOnServer □
- Force DirectX > OpenGL conversion ☑
- Multi-GPU □
- Maximum packet size for image distribution: 8000
- Client connection timeout (seconds): 60

Cluster node properties

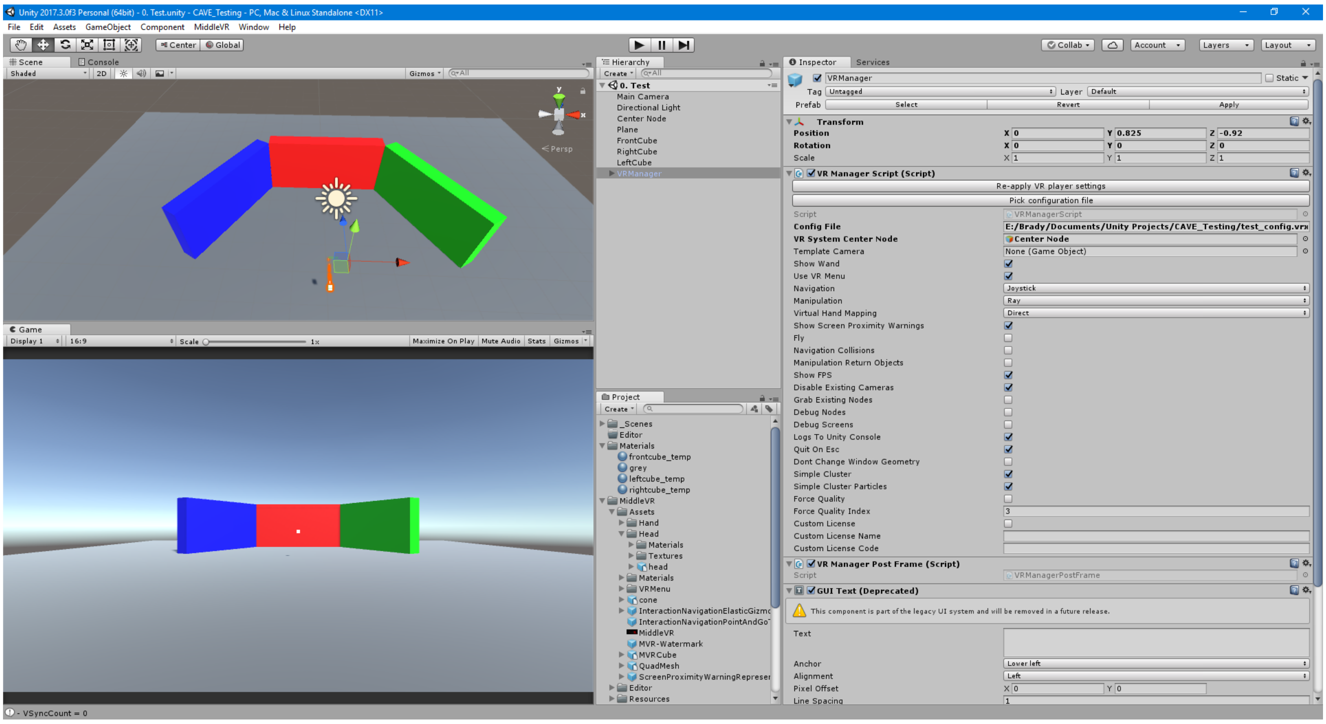
- Address: 127.0.0.1
- ClusterID: Client0
- Viewports: □ FrontView ☑ RightView □ LeftView

Log

```
[+] Saved configuration as 'E:\Brady\Documents\Unity Projects\CAVE_Testing\test_config.vrx' ( 2018-03-01 @ 17:50:34 )

[+] Saved configuration as 'E:\Brady\Documents\Unity Projects\CAVE_Testing\test_config.vrx' ( 2018-03-01 @ 17:51:23 )
```

Figure 1.2: MiddleVR’s cluster editor



☑ VRManager	□ Static
Tag	Untagged
Layer	Default
Prefab	Select / Revert / Apply

Transform

	X	Y	Z
Position	0	0.825	-0.92
Rotation	0	0	0
Scale	1	1	1

☑ VR Manager Script (Script)

Re-apply VR player settings

Pick configuration file

Script	VRManagerScript
Config File	E:/Brady/Documents/Unity Projects/CAVE_Testing/test_config.vrx
VR System Center Node	Center Node
Template Camera	None (Game Object)
Show Wand	☑
Use VR Menu	☑
Navigation	Joystick
Manipulation	Ray
Virtual Hand Mapping	Direct
Show Screen Proximity Warnings	☑
Fly	□
Navigation Collisions	□
Manipulation Return Objects	□
Show FPS	☑
Disable Existing Cameras	☑
Grab Existing Nodes	□
Debug Nodes	□
Debug Screens	□
Logs To Unity Console	☑
Quit On Esc	☑
Dont Change Window Geometry	□
Simple Cluster	☑
Simple Cluster Particles	☑
Force Quality	□
Force Quality Index	3
Custom License	□
Custom License Name	
Custom License Code	

☑ VR Manager Post Frame (Script)

Script	VRManagerPostFrame

☑ GUI Text (Deprecated)

This component is part of the legacy UI system and will be removed in a future release.

Text	
Anchor	Lower left
Alignment	Left
Pixel Offset	X 0 Y 0
Line Spacing	1

Figure 1.3: MiddleVR’s package inside the Unity editor

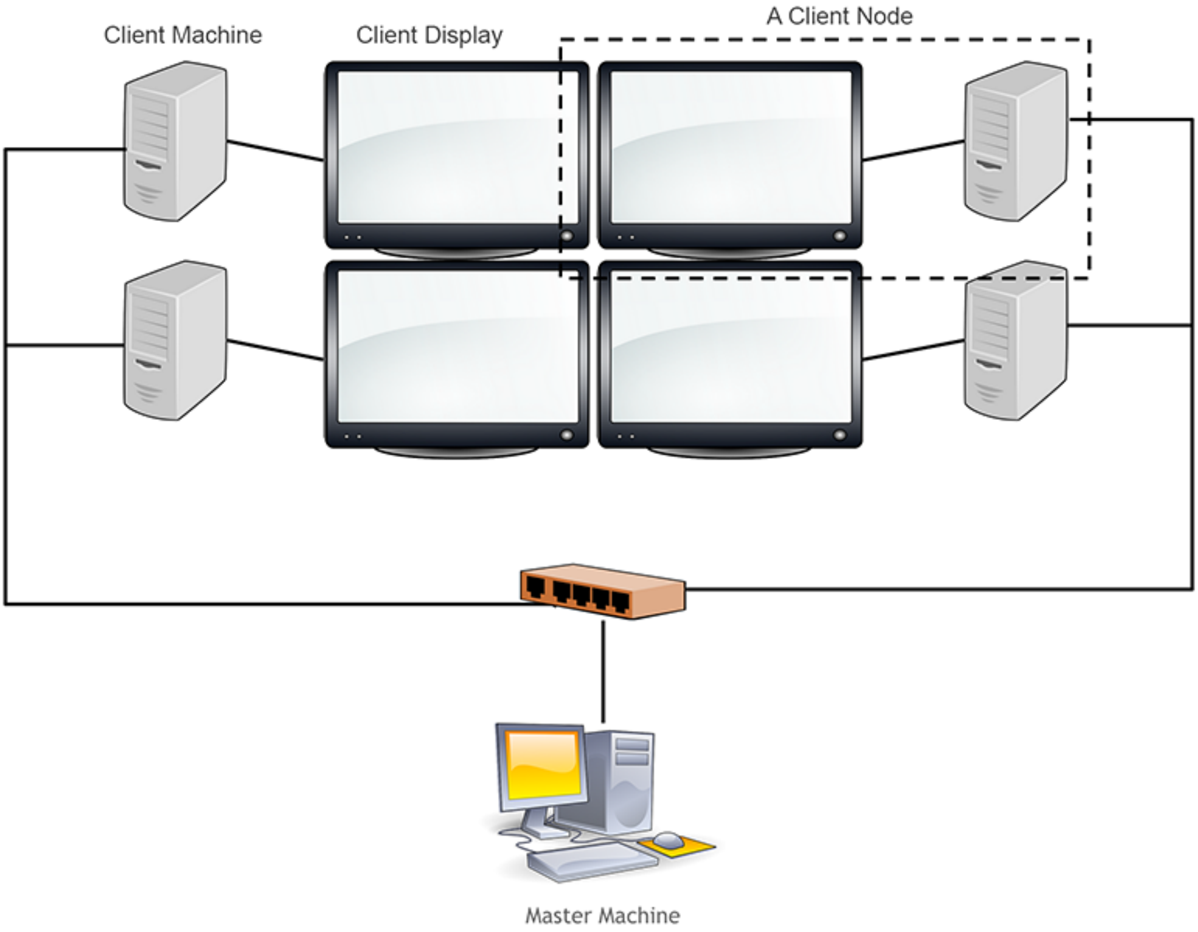


Figure 1.4: An example of a common CAVE hardware setup, similar to our own

6. CONTACT INFORMATION

The JBU Team:

Josh Maier	joshmaier33@siu.edu
Brady Sprinkle	bsprinkle@siu.edu
Utsav Dhungel	utsav@siu.edu

Our mentor:

Dr. Christos Mousas	christos.mousas@siu.edu

JBU website: www2.cs.siu.edu/~jbu

Date: March 2nd, 2018

Approved by: Dr. Christos Mousas

Approver Signature:

Mentor Name: Dr. Christos Mousas

Mentor Signature: