*Florida International University*

*School of Computing and Information Sciences*

Software Engineering Focus

Final Deliverable

Project Title:

SKOPE VR 1.0 Oculus

**Team Members:** Parker Scott, Jose Maldonado

**Product Owner(s)**: Shahin Vassigh

**Mentor(s)**: Albert Elias

**Instructor**: Prof.Masoud Sadjadi , Dr.Francisco Ortega

The MIT License (MIT)

Copyright (c) *2016 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

*Unity has become one of the most widely known game and simulation development programs for cross-platform devices. Their integration with Virtual Reality and Augmented Reality systems is extensive. Contemporary Virtual Reality (VR) technology in particular is enabling new and innovative ways to interact with digital information and deliver engaging experiences. FIU’s SKOPE Project seeks to use this emerging technology to create an engaging interdisciplinary and collaborative learning experience for students. Besides utilizing the most innovative technologies, the software development process was using Agile methods and Scrum development with 6 sprints.*

**Table of Contents**

**Introduction** ……………………………………………………………………………………………………………………………….. 6

Current System …………………………………………………………………………………………………………………………………………….....6

Purpose of New System …………………………………………………………………………………………………………………………………………….... 6

**User Stories**

Implemented User Stories …………………………………………………………………………………………………………………………………………….....8

Pending User Stories …………………………………………………………………………………………………………..….. …………………………….11

**Project Plan**

Hardware and Software Resources ………………………………………………………………………………………….… ……………………………………………....11

Sprints Plan ……………………………………………………………………………………………………………………………………………..13

*Sprint 1*  …………………………………………………………………………………………………………………………... 13

*Sprint 2*  …………………………………………………………………………………………………………………………... 14

*Sprint 3*  …………………………………………………………………………………………………………………………... 14

*Sprint 4*  …………………………………………………………………………………………………………………………... 15

*Sprint 5*  …………………………………………………………………………………………………………………………... 16

*Sprint 6*  …………………………………………………………………………………………………………………………... 16

**System Design**

Architectural Patterns ……………………………………………………………………………………………………………………………………………..18

System and Subsystem Decomposition ……………………………………………………………………………………….…………………………………………………….20

Deployment Diagram ………………………………………………………………………………………………………….…………………………………. 21

Design Patterns ……………………………………………………………………………………………………………….…......................................22

**System Validation**  ……………………………………………………………………………………………………………………...23

**Glossary**  ………………………………………………………………………………………………………………………………….**.**37

**Appendix**  ………………………………………………………………………………………………………………………………….**.**38

Appendix A - UML Diagrams …………………………………………………………………………………………………………………………………………….. 38

Appendix B - User Interface Design ……………………………………………………………………………………….…...…………………………….…………………. 41

Appendix C - Sprint Review Reports ……………………………………………………………………………………...………………………………….………………….. 43

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents …………………………………………………………………….……………………... 47

**References** ……………………………………………………………………………………………………………………...………….49

# 

# 

# 

# 

# 

# Introduction

SKOPE VR 1.0 is a Virtual Reality application that focuses on building an engaging virtual learning environment for students in interdisciplinary areas of study. On this project, our team focused on using User Interface Design, Open World concepts and Immersive Experiences to tailor a student's learning environment around the architectural and functional aspects of FIU’s SIPA Building. This document provides a detailed overview of the design and architecture of the proposed application and its underlying technologies. First we describe the currently deployed SKOPE system by FIU Architecture and the purpose/vision of our new proposed system. Second, we provide the User Stories implemented throughout the development of the proposed system and their related tasks. Third, we provide a thorough overview of the hardware and software resources used through development and the Project Plan. Fourth, we provide a detailed look at SKOPE VR’s system design, architectural patterns, system/subsystem decomposition and accompanying diagrams. Lastly, we go over the accompanying test cases the proposed system.

## 

## Current System

FIU Architecture's SKOPE Project aims at facilitating interdisciplinary learning between students through technology mediated experiences. The current SKOPE Project iteration supports Augmented Reality (AR) and Mobile Virtual Reality (VR) learning experiences. While these experiences do deliver a more immersive experience, they fall short on providing intuitive user interaction and user interface for students in those experiences. Furthermore, the current SKOPE system suffered from a variety of stability issues inherited from VR and AR technology. Finally, the current system does not facilitate the addition of new learning experiences thus limiting the system to the current hard coded scenarios.

## Purpose of New System

Given the current system’s challenges and previous supporting work on the benefits of technology mediated learning environment, our team set out to build a new system for the SKOPE project using the Oculus Virtual Reality platform. We present an intuitive VR design approach that take advantage of Oculus’s hand and head tracking to support SKOPE’s original concept, while tacking it’s inherent challenges. Our primary focus lies on User Interface Design, Open World Design and Immersive Experiences. The resulting system, SKOPE VR 1.0, will immerse students in an interactive virtual environment of FIU’s SIPA building. Student will be able to use SKOPE VR’s new control system and User interface to explore the different architectural and functional aspects of SIPA. The system currently comes with a couple of pre-loaded experiences, however the system has the ability to add experiences to the Open World system at ease.

# User Stories

The following section provides the user stories that served as the basis for the implementation features of SKOPE VR 1.0. These user stories were approved by Ms. Vassigh, the product owner. This section also shows the user stories that are to be considered for future development.

## Implemented User Stories

**User Story # 241 - Learn how to use Mingle/Scrum/Agile**

**As a team developer, I want to** learn how to use Mingle for Scrum/Agile development **so that I** can complete my senior project.

* **Acceptance Criteria:** Able to effectively use Mingle for Scrum/Agile.

**User Story # 242 - Learn How to Use Unity**

**As a team developer, I want to** be able to learn Unity and its system model **so that I** can properly utilize all of the resources available with Unity.

* **Acceptance Criteria:** Able to effectively use Unity’s extensive library list.

**User Story # 243 - Understanding Current Assets**

**As a team developer, I want to** see the current assets available to us **so that I** can utilize these resources instead of recreating the models

* **Acceptance Criteria:** Able to visualize current assets and receive them from mentor.

**User Story # 244 - Learn How to Use C#**

**As a team developer, I want to** learn how to use C# **so that I** can use the official scripting language for Unity

* **Acceptance Criteria:** Competency in utilizing C# as it relates to Unity scripting.

**User Story # 245 - Unity Setup with OVR**

**As a team developer, I want to** setup the Oculus Rift VR **so that I** can properly communicate with the Oculus Rift.

* **Acceptance Criteria:** Able to effectively receive data from Oculus Rift.

**User Story # 246 - Learning with Augmented Reality Unity Asset Transfer**

**As a team developer, I want to** be able to utilize the previous assets available for the simulation **so that I** can utilize the resources instead of recreating the models.

* **Acceptance Criteria:** Able to visualize current SIPA assets in Unity scene editor.

**User Story # 247 - Add Geometry and Physics to Models**

**As a user, I want to** be able to use physics in SIPA **so that I** can move around the scene appropriately in VR.

* **Acceptance Criteria:** Collision Detection, Models have relevant and similar physics meshes

**User Story # 267 - Left Handheld**

**As a user, I want to** control parts of the scene **so that I** can interact with experiences within VR.

* **Acceptance Criteria:** In Game Model follows Controller, Controls Game Environment

**User Story # 268 - User Interface Buttons**

**As a user, I want to** have buttons that I can click **so that I** can initiate experiences.

* **Acceptance Criteria:** Button always faces towards user, Button is clickable and hoverable

**User Story # 275 - Teleporting**

**As a user, I want to** have buttons that I can click **so that I** can initiate experiences.

* **Acceptance Criteria:** Player moves to the spot where they are pointing

**User Story # 276 - Simulation Animation**

**As a user, I want to** be able to see animations in the simulation **so that I** can receive further knowledge on the construction of SIPA.

* **Acceptance Criteria:** Animations start over when actor focuses on them, Animations play properly

**User Story # 277 - Experiences**

**As a user, I want to** be immersed in my simulation and animations **so that I** experience the greatest amount of focus and learning as possible.

* **Acceptance Criteria:** Experiences can be created and stopped.

**User Story # 278 - Sunlight Changes**

**As a user, I want to** be able to change the sun’s location **so that I** can visualize the sun on the windows and the shadows created.

* **Acceptance Criteria:** Sunlight changes based on season and time of day.

**User Story # 288 - Camera Rotation**

**As a user, I want** to be able to rotate the camera **so that I** don’t have to rotate in real life in order to utilize the VR simulation.

* **Acceptance Criteria:** Camera rotates based on input from user.

**User Story # 289 - Left Handheld**

**As a user, I want** to be able to control parts of the experience **so that I** can interact properly with the VR simulation.

* **Acceptance Criteria:** Left Handheld rotates based on input, InfoPanel and OptionPanel are relevant to the experience being used.

**User Story # 290 - Waypoint Zones**

**As a user, I want** to be able to move to different areas of the map **so that I** don’t have to walk everywhere.

* **Acceptance Criteria:** Waypoint moves user to desired area.

**User Story # 291 - Remove Walls Button**

**As a user, I want** to be able to remove the walls of SIPA **so that I** can see the support system behind them.

* **Acceptance Criteria:** Button toggles walls out and into the scene.

**User Story # 301 - Left Handheld Map**

**As a user, I want** to be able to have map on my UI **so that I** can see where I am in the simulation.

* **Acceptance Criteria:** Map places user in the correct place.

## Pending User Stories

**User Story # 302 - Movement Right Handheld**

**As a user, I want** to be able to move around the scene with a walking animation **so that I** don’t have to teleport everywhere.

* **Acceptance Criteria:** The user properly moves through the scene as pressed by the joystick

# Project Plan

This section describes the planning that went into the realization of this project. This project incorporated Agile development techniques which required sprint planning. Sprint plans are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

## Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Hardware:

* **Oculus Rift:** The Virtual Reality Headset used in this project.
* **Oculus Rift Touch Controllers:** Virtual Reality controllers used in project.
* **Oculus Rift Sensors:** Virtual Reality sensors for location of user.
* **Windows/Mac Computer:** Used to run Unity and for connecting with the Oculus Rift.

Software:

* **Unity 2017.2.0f3:** The version of Unity that our project was developed within.
* **Unity OVRPlugin:** The plugin utilized within Unity for connecting with the Oculus Rift.
* **Microsoft Visual Studio 2017:** IDE used for developing Unity scripts.
* **MonoDevelop-Unity:** Alternative IDE used for developing Unity scripts.
* **3ds Max:** Program for editing models used in the project.
* **C#:** The programming language used in Unity scripts.
* **Mingle:** Software development management tool, used for planning.
* **GitHub:** Online Repository used for control and source management.
* **Unity Collaborate:** Online Repository used by Unity for sharing projects.
* **Google Drive:** Used to store and share documents among the team.
* **Skype:** Used for communication among the team.
* **Slack:** Used for communication among the team.
* **Google Hangouts:** Used for communication among the team.
* **Draw.io:** Used for UML diagram creation.
* **Windows Movie Maker:** Used to edit videos of demos.
* **Windows 10:** Used to run/develop/host the application.

**Sprints Plan**

### Sprint 1

(8/28/2017 - 9/18/2017)

The first sprint was utilized for learning the current technology that SKOPE was utilizing, deciding on an appropriate VR platform, and learning the components dealing with said VR platform: Oculus Rift.

**User Story # 241 - Learn how to use Mingle/Scrum/Agile**

* **Acceptance Criteria:** Able to effectively use Mingle for Scrum/Agile.
* **Related Tasks:**
  + **#248 Introduction to Mingle**
  + **#249 Brief on Google Drive Agile Documents**

**User Story # 242 - Learn how to use Unity**

* **Acceptance Criteria:** Able to effectively use Unity’s extensive library list.
* **Related Tasks:**
  + **#250 Unity Scripting Tutorials**
  + **#251 Unity Scene Editor Tutorials**
  + **#252 Install Unity 2017.2.0.f3**

**User Story # 243 - Understanding Current Assets**

* **Acceptance Criteria:** Able to visualize current assets and receive them from mentor
* **Related Tasks:**
  + **#253 Review Current Assets with Mentor**

**User Story #244 - Learn how to use C#**

* **Acceptance Criteria:** Competency in utilizing C# as it relates to Unity scripting
* **Related Tasks:**
  + **#254 Scripts as Behaviour Components**
  + **#255 Variables and Functions**
  + **#256 Scope and Access Modifiers**
  + **#257 GetComponent**
  + **#258 Data Types**

### Sprint 2

(9/19/2017 - 10/2/2017)

Sprint 2 focused on transferring assets from the previous SKOPE iterations and setting up the Oculus Rift libraries.

**User Story #245 - Unity Setup with OVR**

* **Acceptance Criteria:** Able to effectively use/receive data from Oculus Rift
* **Related Tasks:**
  + **#259 Install OVRPlugin**
  + **#260 Input from Oculus**

**User Story # 246 - Learning with Augmented Reality Unity Asset Transfer**

* **Acceptance Criteria:** Able to visualize current SIPA assets in Unity scene editor.
* **Related Tasks:**
  + **#261 Transfer from Github**
  + **#262 Transfer Prefabs to Scene Editor**
  + **#263 Transfer Terrain to Scene Editor**
  + **#264 Transfer Scripts to GameObjects**

**User Story # 247 – Add Geometry and Physics to Models**

* **Acceptance Criteria:** Collision Detection, Models have relevant and similar physics meshes
* **Related Tasks:**
  + **#265 Add Collision Meshes to Models**
  + **#266 Test Collisions**

### Sprint 3

(10/3/2017 - 10/16/2017)

Sprint 3 was mostly dedicated to continuing to transfer over assets from the previous projects and have them function properly. As well, development on a handheld UI and buttons began.

**User Story #267 - Right Handheld**

* **Acceptance Criteria:** In Game model follows controller, controller creates a raycast.
* **Related Tasks:**
  + **#269 Raycasting**
  + **#270 Teleportation**
  + **#271 Integration with Experiences**
  + **#272 Input for Creating Teleport Ray**

**User Story #268 - User Interface Buttons**

* **Acceptance Criteria:** Button always faces towards user, Button is clickable and hoverable.
* **Related Tasks:**
* **#273 UI Button Class Creation**
* **#274 UI Buttons Linking to Experiences**

### Sprint 4

(10/17/2017 - 10/30/2017)

Sprint 4 was primarily focused on expanding our current UI, and coordinating experiences with our UI in order to properly further more experiences.

**User Story #275 - Teleporting**

* **Acceptance Criteria:** Player moves to the spot where they are pointing
* **Related Tasks:**
  + **#280 Marker Creation**
  + **#281 Right Handheld Input**

**User Story #276 - Simulation Animation**

* **Acceptance Criteria:** Animations start over when actor focuses on them, Animations play properly
* **Related Tasks:**
  + **#282 Import of Animations**
  + **#283 Animation Rewind and Play**

**User Story #277 - Experiences**

* **Acceptance Criteria:** Experiences can be created and stopped.
* **Related Tasks:**
  + **#284 Creation of Experiences Abstract Class**

**User Story #278 - Sunlight Changes**

* **Acceptance Criteria:** Sunlight changes based on season and time of day.
* **Related Tasks:**
  + **#285 Collection of Sunlight Data**
  + **#286 Sunlight Rotation**
  + **#287 Proper Sunlight Rotation Testing**

### 

### Sprint 5

(10/31/2017 - 11/13/2017)

**User Story # 288 - Camera Rotation**

* **Acceptance Criteria:** Camera rotates based on input from user.
* **Related Tasks:**
  + **#292 Handheld Input**
  + **#293 Camera and Children Rotate**

**User Story # 289 – Left Handheld**

* **Acceptance Criteria:** Left Handheld rotates based on input, InfoPanel and OptionPanel are relevant to the experience being used.
* **Related Tasks:**
  + **#294 Options Panel**
  + **#295 Info Panel**
  + **#296 Third Panel**

**User Story # 290 – Waypoint Zones**

* **Acceptance Criteria:** Waypoint moves user to desired area.
* **Related Tasks:**
  + **#297 Teleportation**
  + **#298 UI Integration**

**User Story # 291 – Remove Walls Button**

* **Acceptance Criteria:** Button toggles walls out and into the scene.
* **Related Tasks:**
  + **#299 Asset Tagging**
  + **#300 Removal Script**

### Sprint 6

(11/14/2017 - 11/27/2017)

**User Story #301 - Left Handheld Map**

* **Acceptance Criteria:** Map places user in the correct place.
* **Related Tasks:**
  + **#1136 [Reset Remote Game](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/mult_touch_midair_and_motion_f/cards/1136)**
  + **#1135 [Reset Local Game](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/mult_touch_midair_and_motion_f/cards/1135)**
  + **#1134 [Use Voice to Reset](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/mult_touch_midair_and_motion_f/cards/1134)**

**User Story #302 – Movement Right Handheld**

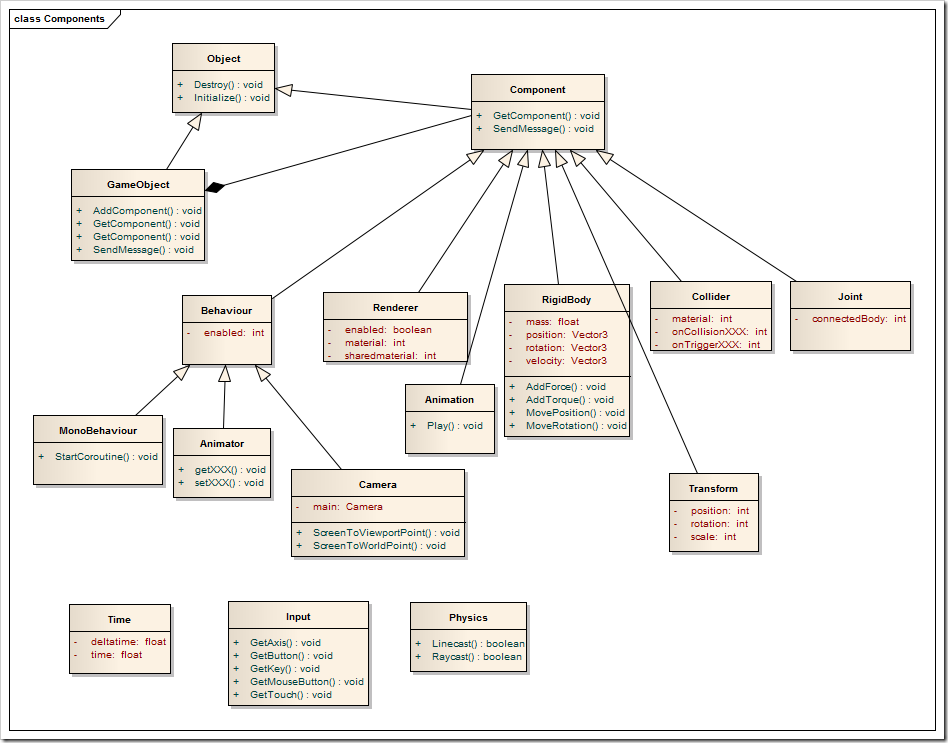
* **Acceptance Criteria:** The user properly moves through the scene as pressed by the joystick.
* **Related Tasks:**
  + **#1108 Develop the UI.**
  + **#1107 Watch tutorials on ways to change UI.**
  + **#1106 Import code from GitLab.**

# 

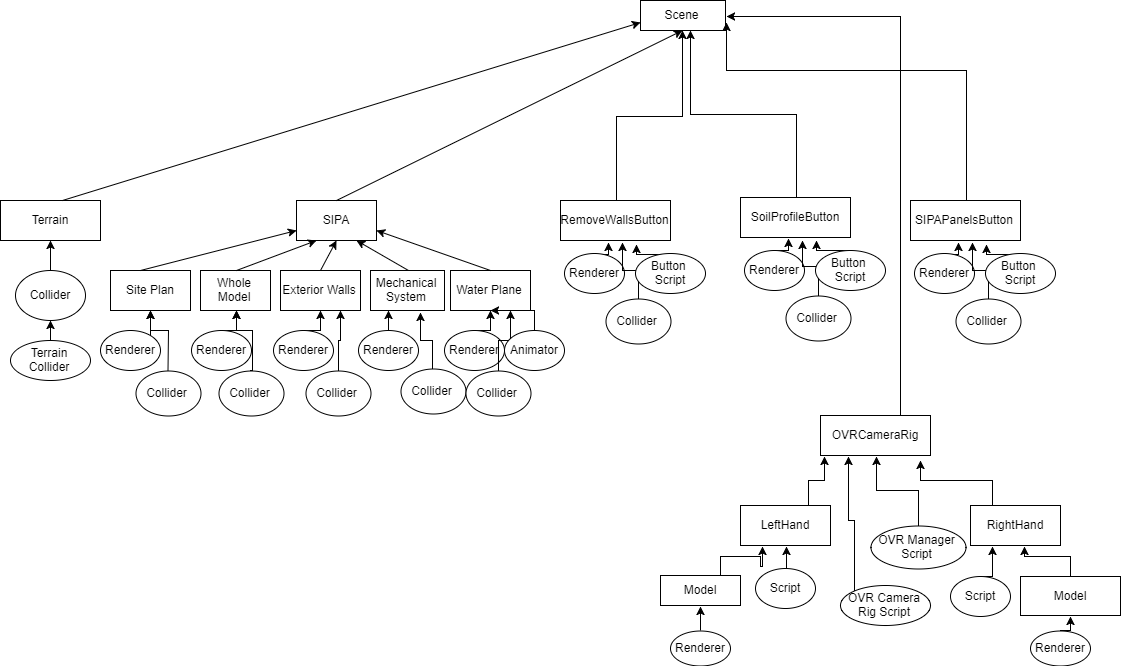
# 

# System Design

This section contains information on the design decisions that went into the SKOPE VR project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

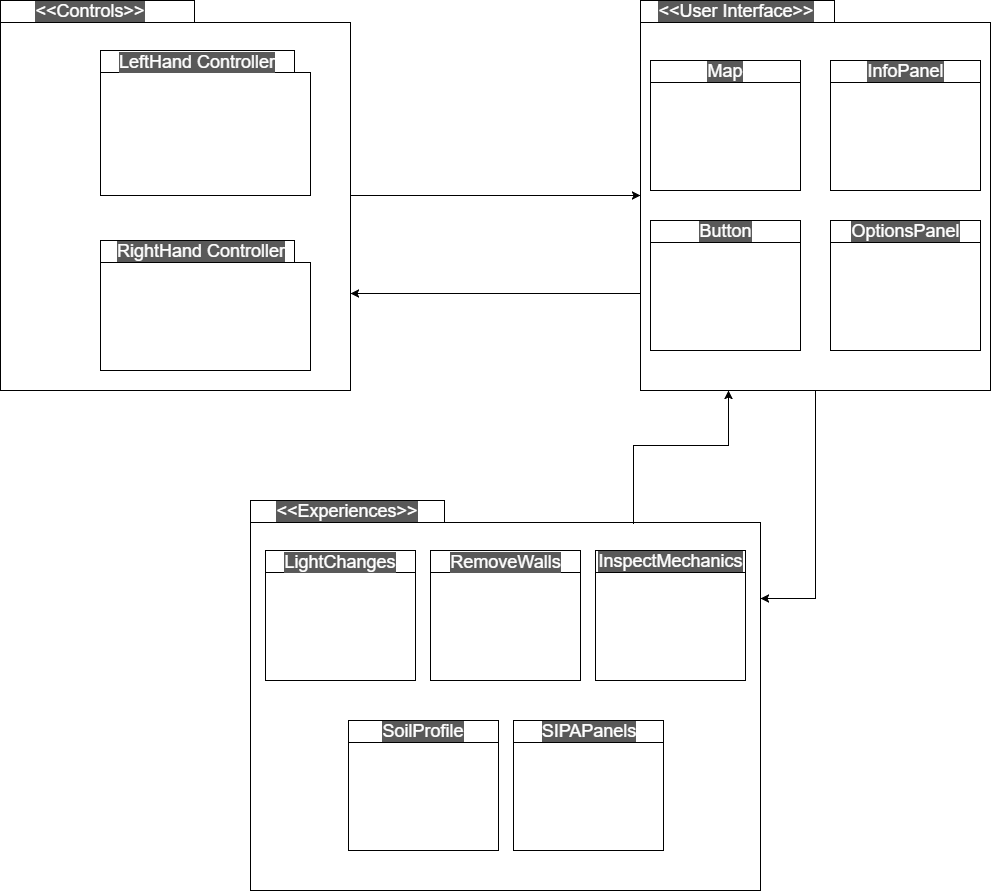
**Architectural Patterns**

Unity naturally employs the use of an entity-component based design wherein we assigned specific scripts to the GameObjects/Entities in order to give them a specific function. We decided to continue with this design instead of using the Model View Controller design, because our project wouldn’t become convoluted with the amount of GameObjects that we have since we’re creating a simulation. Shown above is the architecture for a GameObject which is merely an extension of an object that can be initialized or destroyed. The GameObject itself has multiple components assigned to it, such as in our case for most GameObjects, a renderer and a collider. However, the diagram below details each specific GameObject in our scene, its components, and how it relates to other GameObjects in the scene:



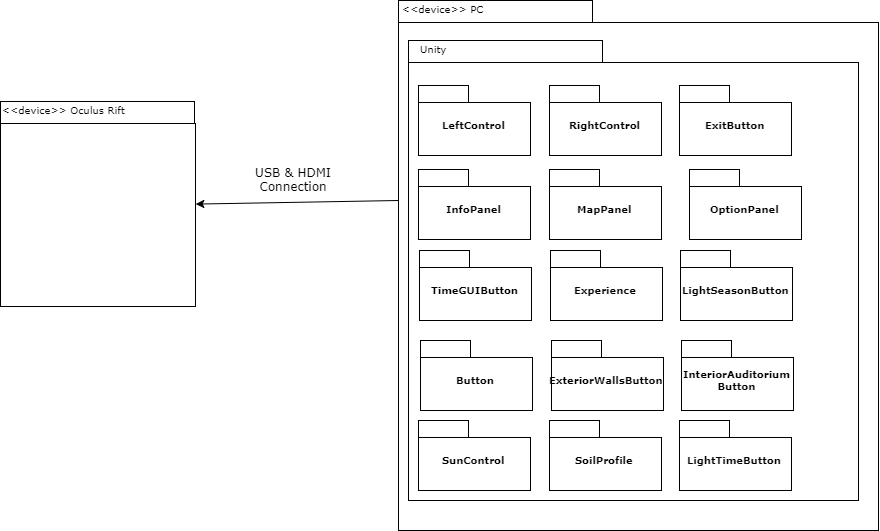
## 

## System and Subsystem Decomposition



As noticeable by the model above, SKOPE VR was developed using an interface between the controls and the experiences. We utilized interfacing between the experiences and controls instead of a manager, because it is the system design that coordinates best for our project and entity component styles. However, SKOPE VR does contain an OVRManager class which strictly manages the OVR headset functionality and the touch controls input. However, our simulation isn’t based around this manager; the manager does not control any components of the scene besides the OVR camera.

## Deployment Diagram



We have an executable application that was created by Unity for the project. Deployment is as simple as running the application on a PC that has enough graphical power to display the simulation and run the Oculus Rift. The Oculus program that synchronizes the Oculus Rift with Windows, which will then allow the Unity program to communicate with the Oculus Rift for the simulation.

## Design Patterns

**Singleton - creational:**

In the application, most scripts are singleton creation, because each script represents a specific entity, because the functions of each entity are detailed in the individual script.

# System Validation

**Unit Tests**

**User Story # 247 - Add Geometry and Physics to Models**

**Test Case 1: Sunny Day**

**Purpose:**

* To test that the geometry and physics added to the Unity models will affect or restrict the user when they are visibly walking into or touching the GameObject.

**Precondition:**

* SKOPE VR application has been executed
* The User is touching all GameObjects when walking around the map

**Test Procedure:**

1. The User moves around the map and visibly touches all GameObjects

**Expected Result:**

1. The user is stopped or hindered by all GameObjects; the user can not pass through the object.

**Actual Result:**

1. The user is stopped or hindered by all GameObjects; the user can not pass through the object.
   1. Pass

**Test Case 2: Rainy Day**

**Purpose:**

* To test that the *geometry and physics added to the Unity models will not touch. affect or restrict the user when they’re not visibly touching the GameObjects*  script will not allow the User to manipulate an object with a gesture tap if they are not looking at it.

**Precondition:**

* SKOPE VR application has been executed
* The User is not touching any GameObjects when walking around the map

**Test Procedure:**

1. The User moves around the map and doesn’t visibly touch any GameObjects

**Expected Result:**

1. The user is not stopped or hindered by any GameObjects.

**Actual Result:**

1. The user is not stopped or hindered by any GameObjects.
   1. Pass

**User Story # 267 - Left Handheld / Hub**

**As a user, I want to** control parts of the scene **so that I** can interact with experiences within VR.

* **Acceptance Criteria:** hub rotates according to joystick

**Test Case 1: Sunny Day**

**Purpose:**

* The hub consists of three panels that rotate up/down around a central point in the player’s hand. This test will check if the Hub properly rotates around the given central point to a desired panel.

**Precondition:**

* SKOPE VR application has been executed
* Left Oculus touch controller is being tracked by SKOPE VR.

**Test Procedure:**

1. The user pushes left hand controller joystick at the desired direction (up/down).

**Expected Result:**

1. The hub should rotate in according to the direction of the left Oculus touch joystick.

**Actual Result:**

1. The hub rotates in the direction of the left Oculus touch joystick.
   1. Pass

**Test Case 2: Rainy Day**

**Purpose:**

* The hub rotates along the physical rotation of the left controller as well. This test will check if the Hub rotates independent of the left touch controller joystick.

**Precondition:**

* SKOPE VR application has been executed
* Left Oculus touch controller is being tracked by SKOPE VR.

**Test Procedure:**

1. The player rotates the left Oculus Touch controller without pushing the joystick.

**Expected Result:**

1. Hub ` rotation should reflect the rotation of the left oculus Touch controller, independent of the joysticks direction.

**Actual Result:**

1. Hub rotation maps to the physical rotation of the left Oculus Touch controller.
   1. Pass

**User Story # 268 - User Interface Buttons**

**Test Case 1: Sunny Day**

**Purpose:**

* To test that the button class functions properly, and that when the user hovers over the button, it reacts. As well, when the user clicks the button, it initiates the proper experience.

**Precondition:**

* SKOPE VR application has been executed
* The user is hovering over the button
* The user clicks the button

**Test Procedure:**

1. The user hovers over the daylight button
2. The user clicks the daylight button
3. The experience begins

**Expected Result:**

1. Console output by the experience that it has begun.
2. The daylight UI appears

**Actual Result:**

1. Console output by the experience that it has begun.
   1. Pass
2. The daylight UI appears
3. Pass

**Test Case 2: Rainy Day**

**Purpose:**

* To test that the button class functions properly, and that when the user doesn’t focus on the button or click the button, then it doesn’t react.

**Precondition:**

* SKOPE VR application has been executed
* The user doesn’t hover or click a button

**Test Procedure:**

1. The user doesn’t hover over any of the buttons

**Expected Result:**

1. The button should not react or initiate any experience.

**Actual Result:**

1. The button does not react or initiate any experience.
   1. Pass

**User Story # 275 - Teleporting**

**As a user, I want to** have buttons that I can click **so that I** can initiate experiences.

* **Acceptance Criteria:** Player moves to the spot where they are pointing

**Test Case 1: Sunny Day**

**Purpose:**

* To test the Pointers teleportation capability, which should teleport the player to a valid pointed location in the world space when “A” is pressed and released..

**Precondition:**

* SKOPE VR application has been executed
* Oculus Right touch controller is being tracked by SKOPE VR.
* Pointed location is less than 10 meters away from the player.
* Pointed location is a valid teleportation spot in the world space.

**Test Procedure:**

1. The user presses and holds “A” on the right Oculus Touch controller
2. The user points at a valid teleportation spot in the world space
3. Player releases “A” on the right Oculus Touch controller

**Expected Result:**

1. Player is teleported to the pointed location once “A” is released.

**Actual Result:**

1. Player is teleported to the appropriate location.
   1. Pass

**Test Case 2: Rainy Day**

**Purpose:**

* A player might accidently attempt a teleportation to an invalid location such as a vertical wall, the hub or a location 10 meters away from the player. This test will check if the system will avoid teleporting to invalid locations.

**Precondition:**

* SKOPE VR application has been executed
* Oculus Right touch controller is being tracked by SKOPE VR.
* Pointed location is more than 10 meters away from the player **OR** Pointed location is an **invalid** teleportation spot in the world space.

**Test Procedure:**

1. The user presses and holds “A” on the right Oculus Touch controller
2. The user points at an **invalid** teleportation spot in the world space
3. Player releases “A” on the right Oculus Touch controller

**Expected Result:**

1. Teleportation is aborted and the player stays at the current location

**Actual Result:**

1. The player did not teleport, stayed at the current location.
   1. Pass

**User Story # 276 - Simulation Animation**

**As a user, I want to** be able to see animations in the simulation **so that I** can receive further knowledge on the construction of SIPA.

* **Acceptance Criteria:** Animations start over when actor focuses on them, Animations play properly

**User Story # 277 - Experiences**

**As a user, I want to** be immersed in my simulation and animations **so that I** experience the greatest amount of focus and learning as possible.

* **Acceptance Criteria:** Experiences can be created and stopped.

**User Story # 278 - Sunlight Changes**

**As a user, I want to** be able to change the sun’s location **so that I** can visualize the sun on the windows and the shadows created.

* **Acceptance Criteria:** Sunlight changes based on season and time of day

**Test Case 1: Sunny Day**

**Purpose:**

* To test that the sunlight reacts correctly when you click on the User Interface buttons.

**Precondition:**

* SKOPE VR application has been executed
* The user is within the sunlight experience

**Test Procedure:**

1. The user clicks the season or time of day buttons on UI

**Expected Result:**

1. The UI updates properly to the season and time of day displayed on the UI.

**Actual Result:**

1. The UI updates properly to the season and time of day displayed on the UI.
2. Pass

**Test Case 2: Rainy Day**

**Purpose:**

* To test that the sunlight reacts correctly when you click on the User Interface buttons.

**Precondition:**

* SKOPE VR application has been executed
* The user is within the sunlight experience

**Test Procedure:**

1. The user clicks the season or time of day buttons on UI

**Expected Result:**

1. The UI updates properly to the season and time of day displayed on the UI.

**Actual Result:**

1. The UI updates properly to the season and time of day displayed on the UI.
   1. Pass

**User Story # 288 - Camera Rotation**

**As a user, I want** to be able to rotate the camera **so that I** don’t have to rotate in real life in order to utilize the VR simulation.

* **Acceptance Criteria:** Camera rotates based on input from user.

**Test Case 1: Sunny Day**

**Purpose:**

* To test player camera/view rotation independent of player head movement.

**Precondition:**

* SKOPE VR application has been executed
* Oculus Rift is being tracked by SKOPE VR.

**Test Procedure:**

1. The user pushes left OR right on the right Oculus Touch controller’s joystick.

**Expected Result:**

1. The players view should either turn left or right based on the direction of the joystick, independent of the player’s head movement.

**Actual Result:**

1. The player view does change in the direction the player is pushing the right joystick, independent of the player’s head movement.
   1. Pass

**Test Case 2: Rainy Day**

**Purpose:**

* A player might attempt to rotate the camera by pressing up and down on the right oculus touch joystick. This test checks whether the Y axis of the joystick has any effect on the rotation.

**Precondition:**

* SKOPE VR application has been executed
* Oculus Rift is being tracked by SKOPE VR.

**Test Procedure:**

1. The user pushes up OR down on the right Oculus Touch controller’s joystick.

**Expected Result:**

1. The players view should not change, unless triggered by head movement.

**Actual Result:**

1. The player view did not change based on the right oculus touch joystick Y axis.
   1. Pass

**User Story # 289 - Left Handheld**

**As a user, I want** to be able to control parts of the experience **so that I** can interact properly with the VR simulation.

* **Acceptance Criteria:** Left Handheld rotates based on input, InfoPanel and OptionPanel are relevant to the experience being used.

**Test Case 1: Sunny Day**

**Purpose:**

* The Hub’s displays relevant options to the player based on their current experience through the Option Panel. This test checks if the correct panel is loaded when a user enters an experience.

**Precondition:**

* SKOPE VR application has been executed.

**Test Procedure:**

1. The player enters an experience through a world space button.
2. The player rotates the Hub to the Options Panel.

**Expected Result:**

1. The relevant options, specific to that experience, should have loaded into the Option Panel.

**Actual Result:**

1. The correct options load into the Option Panel for each world space experience.
   1. Pass

**Test Case 2: Rainy Day**

**Purpose:**

* Some experience might not have options to choose from, thus we need to check that we do not receive an error for not having any options to load.

**Precondition:**

* SKOPE VR application has been executed

**Test Procedure:**

1. The player enters an experience through a world space button that does not have any options, such as remove a wall.
2. The player rotates the Hub to the Options Panel.

**Expected Result:**

1. The Option Panel should be empty and void of any errors.

**Actual Result:**

1. The Option Panel is empty. No errors are reported.
   1. Pass

**User Story # 290 - Waypoint Zones**

**As a user, I want** to be able to move to different areas of the map **so that I** don’t have to walk everywhere.

* **Acceptance Criteria:** Waypoint moves user to desired area.

**User Story # 291 - Remove Walls Button**

**As a user, I want** to be able to remove the walls of SIPA **so that I** can see the support system behind them.

* **Acceptance Criteria:** Button toggles walls out and into the scene.

**User Story # 301 - Left Handheld Map**

**As a user, I want** to be able to have map on my UI **so that I** can see where I am in the simulation.

* **Acceptance Criteria:** Map places user in the correct place.

**Test Case 1: Sunny Day**

**Purpose:**

* The Hub’s Map Panel displays world space location of the player. This test will check if the Map’s Cursor accurately points the player location in the world space

**Precondition:**

* SKOPE VR application has been executed.

**Test Procedure:**

1. The player will begin at the starting position once the game executes.
2. The player checks starting position in the Map Panel.
3. The player teleports to a desired location.
4. Player check new position in the Map Panel.

**Expected Result:**

1. Map Panel’s cursor location should reflect the player’s location in the world space.

**Actual Result:**

1. The Map panel’s cursor location does reflect the player’s location in the world space.
   1. Pass

**Test Case 2: Rainy Day**

**Purpose:**

* A player might exit the world space to enter an experience at a separate location. This test check if the Map Panel is disabled when entering an experience outside the open world space.

**Precondition:**

* SKOPE VR application has been executed

**Test Procedure:**

1. The player will begin at the starting position once the game executes.
2. The player checks starting position in the Map Panel.
3. The player enters any experience that lies outside the world space.
4. Player check Map Panel.

**Expected Result:**

1. The Map’s cursor should be disabled, not visible in the world space.

**Actual Result:**

1. The Map’s cursor is disabled and not visible to the player until they exit the experience.
   1. Pass

# 

# Glossary

**Agile:** Describes a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams.

**Virtual Reality:** the computer-generated simulation of a three-dimensional image or environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment,

**C#:** Object-oriented programming language used to develop in Unity.

**Oculus Rift:** a photographic recording of a [light field](https://en.wikipedia.org/wiki/Light_field), rather than of an image formed by a [lens](https://en.wikipedia.org/wiki/Lens_(optics)), and it is used to display a fully [three-dimensional](https://en.wikipedia.org/wiki/Three-dimensional_space) image of the holographed subject.

**World Space:** An open world that is explorable by the player.

**Experience:** Modular learning unit that can be accessed in the world space.

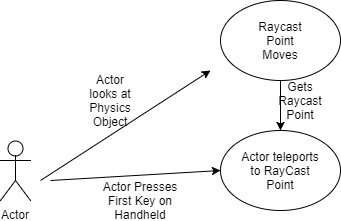
**Mingle:** An Agile project management application from ThoughtWorks Studios for teams that need to make real-time decisions using real-time data. Mingle provides project intelligence and supports all team activity.

**Unity:** A cross-platform game engine developed by Unity Technologies and used to develop video games for PC, consoles, mobile devices and websites.

# Appendix

## Appendix A - UML Diagrams

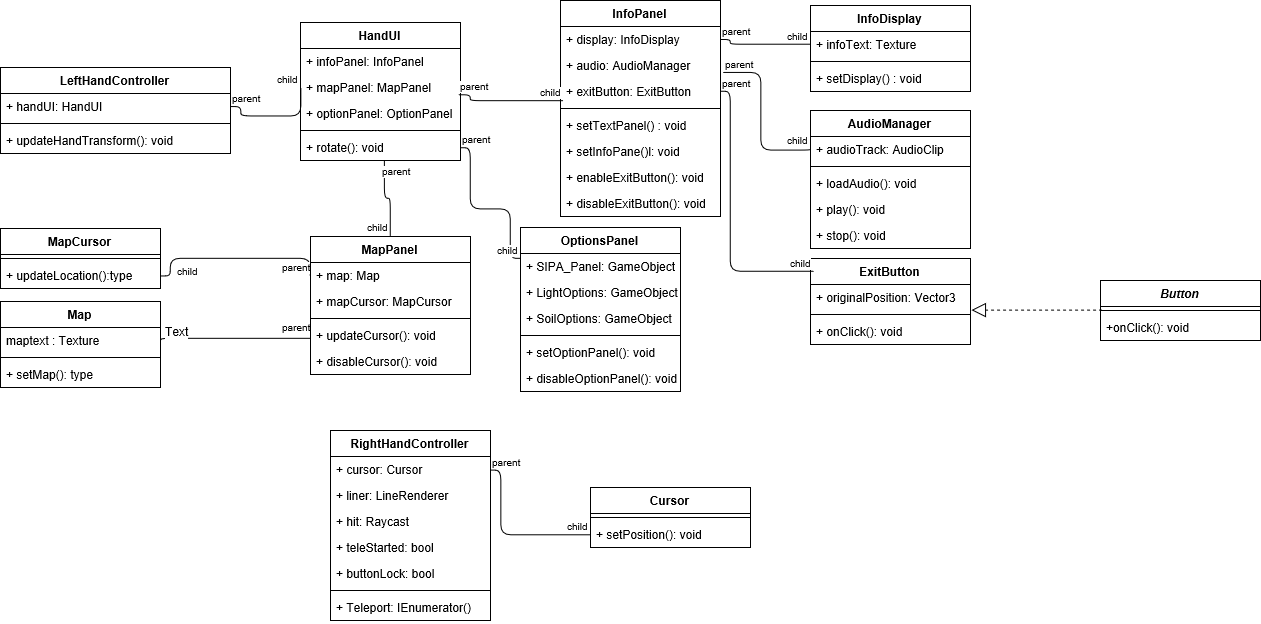
**Use Case Diagrams:**

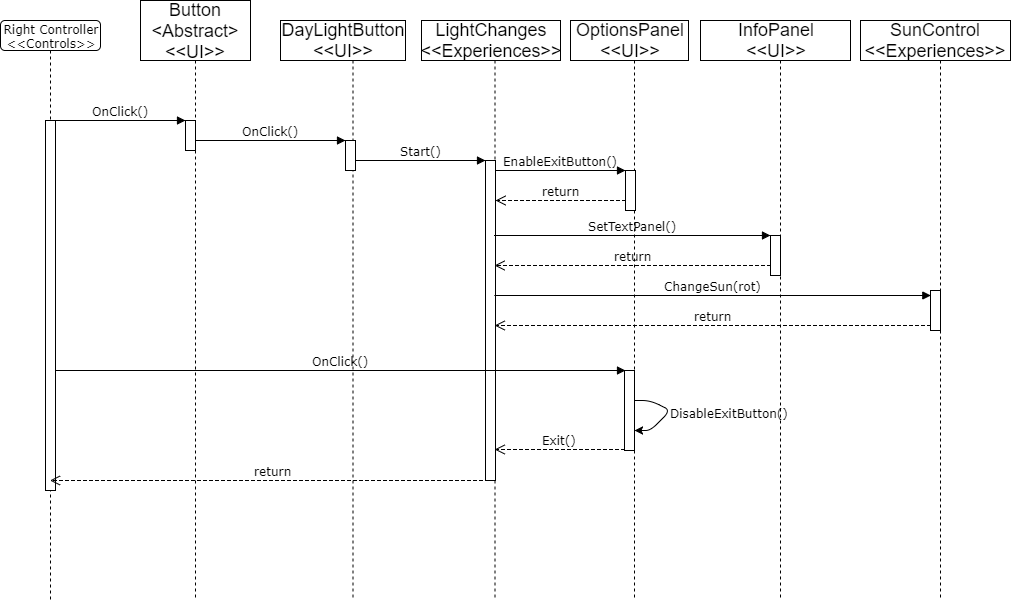


**Use case Diagram : Movement**

**CLASS DIAGRAMS**

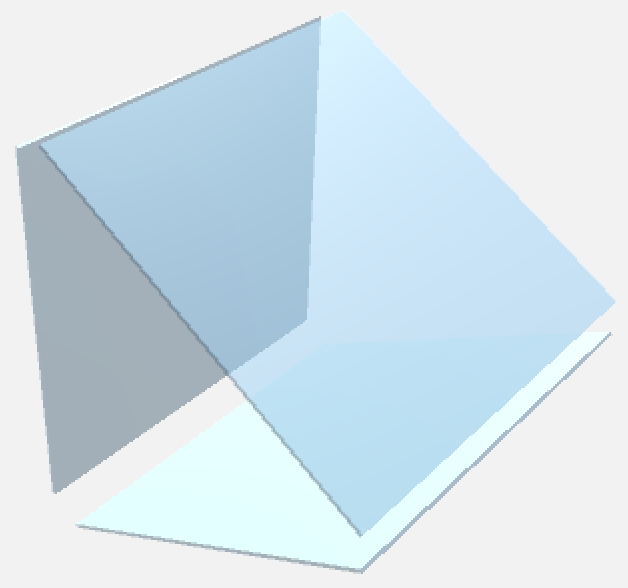
**Full Class Diagram:**

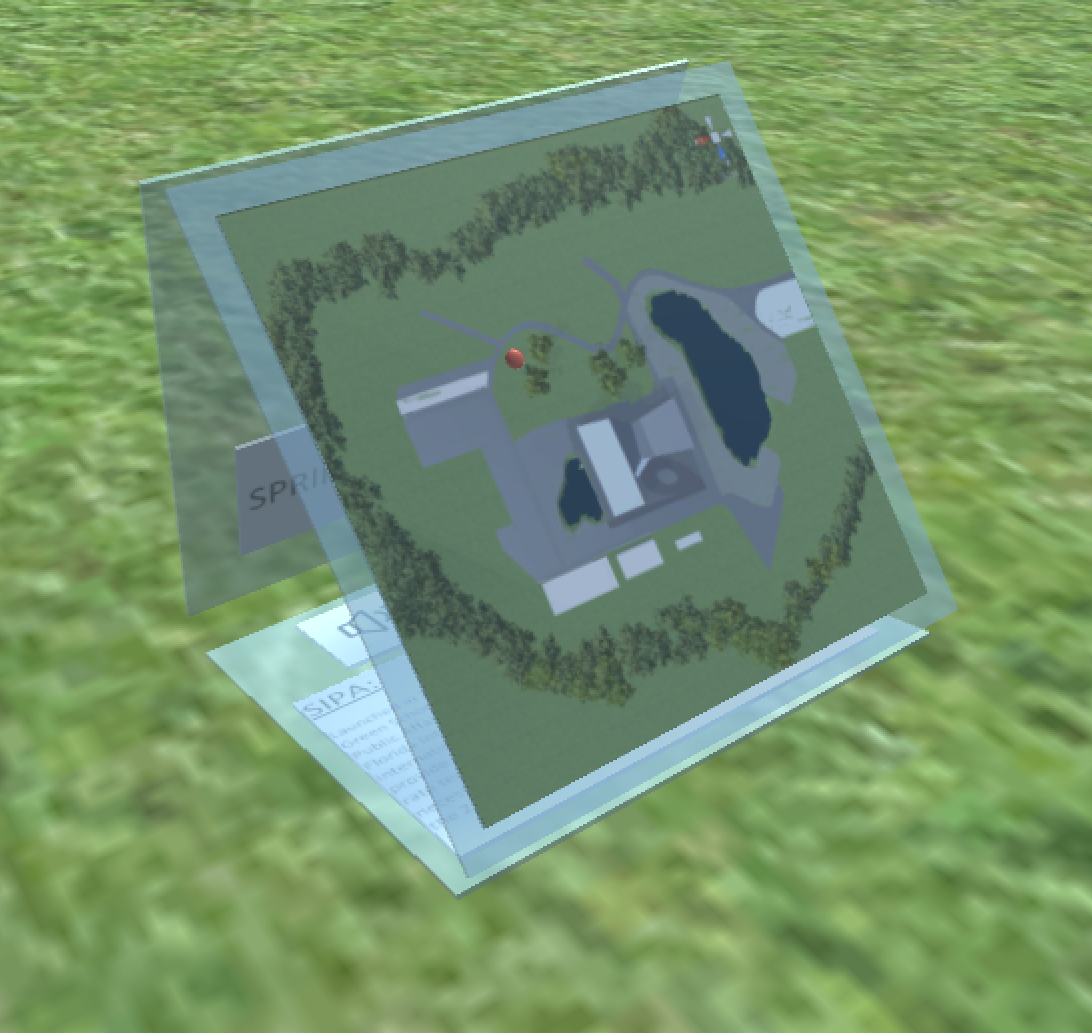
****



**Daylight Experience Sequence Diagram:** Above

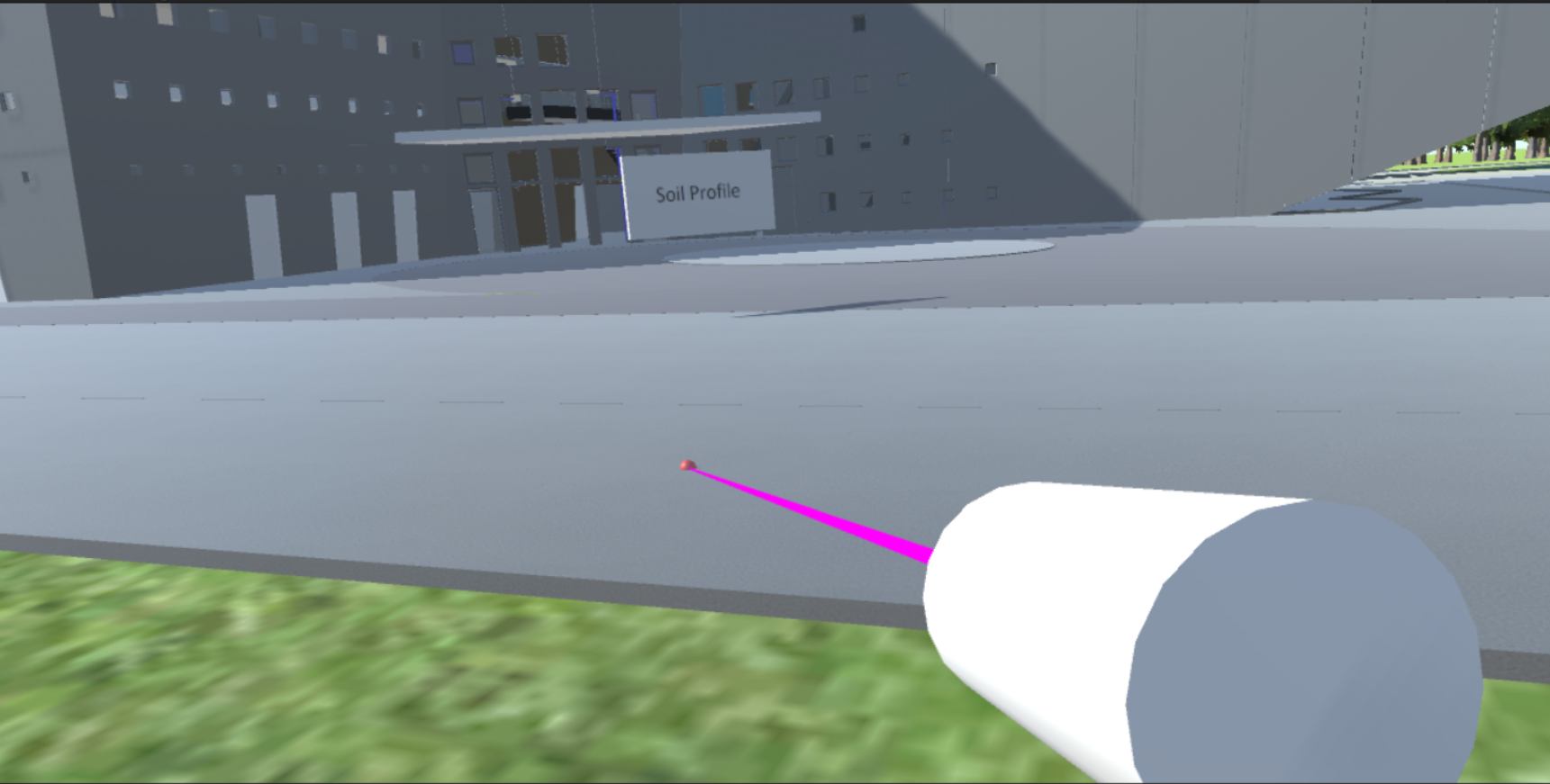
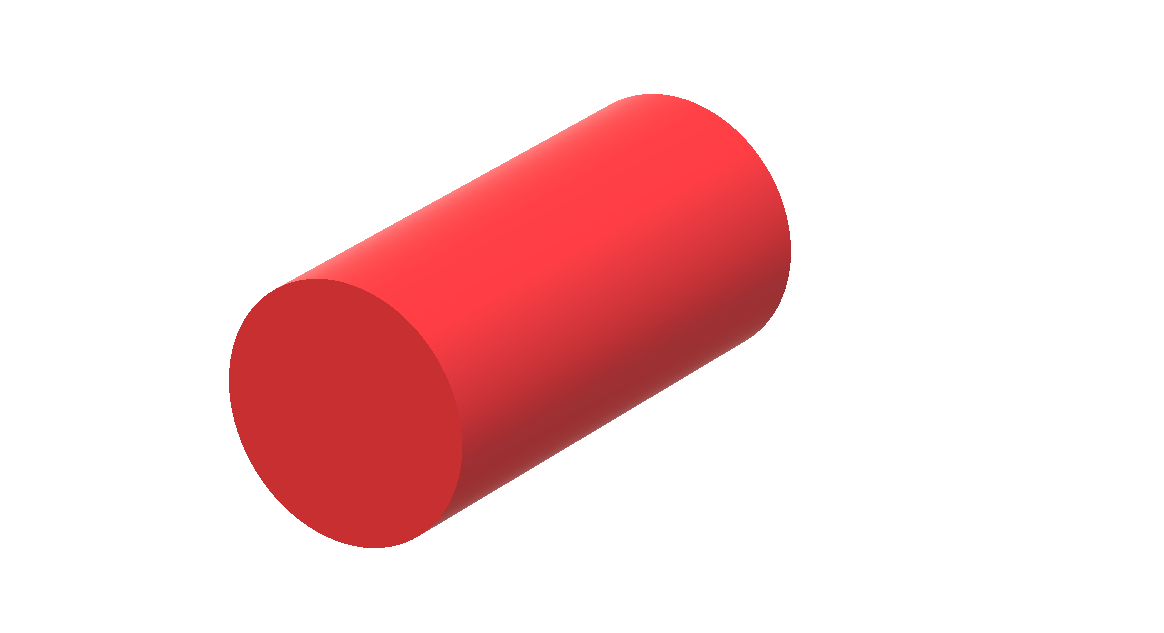
## Appendix B - User Interface Design

****

****

**Figure B1 - Hub**

## 





**Figure B2 - Pointer**

## Appendix C - Sprint Review Reports

**Sprint Review 1**

Attendees: Parker, Jose, Ms. Vassigh

Start time: 5:30pm

End time: 5:45pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* Learn How to use Mingle/Scrum/Agile #241
* Learn How to use Unity #242
* Understanding Current Assets #243
* Learn How to use C# #244

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* None

**Sprint Review 2**

Attendees: Parker, Jose, Ms. Vassigh

Start time: 5:30pm

End time: 5:50pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* Learning with Augmented Reality Unity Asset Transfer #246
* Add Geometry and Physics to Models #247

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* Unity Setup with OVR #245 (Story required more time to complete)

**Sprint Review 3**

Attendees: Parker, Jose, Ms. Vassigh

Start time: 5:30pm

End time: 5:50pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* Unity Setup with OVR #245
* Right Handheld #267
* User Interface Buttons #268

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* None

**Sprint Review 4**

Attendees: Parker, Jose, Ms. Vassigh

Start time: 5:45pm

End time: 6:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* Teleporting #275
* Simulation Animation #276
* Experiences #277
* Sunlight Changes #278

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* None

**Sprint Review 5**

Attendees: Parker, Jose, Ms. Vassigh

Start time: 5:35pm

End time: 6:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* Camera Rotation #288
* Left Handheld #289
* Waypoint Zones #290
* Remove Walls Button #291

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* None

**Sprint Review 6**

Attendees: Parker, Jose, Ms. Vassigh

Start time: 5:30pm

End time: 5:40pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

* Left Handheld Map # 301

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* Camera Movement # 302 (User Story will be moved to the product backlog as it currently causes nausea when utilized)

## 

## 

## 

## 

## Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

## User Manual:

1. The user will start at the World Space start position
2. To move about the world space, the player must:
   1. Press “A” on right Oculus touch controller
   2. Point the right Oculus Touch controller to a desired location to move to
   3. Release pressing “A”
   4. The player should have teleported to the desired location.
3. To move to a desired panel in the Hub, the player must
   1. Push the Joystick Up or Down on the Left Touch Controller
   2. The Hub should rotate around its center to the

desired panel.

1. To “Push” a button on the World Space or on the Hub, the player must:
   1. Point the right Oculus Touch controller to the button of interest.
   2. Press the back trigger button on the right Oculus controller.
   3. The corresponding behavior should execute.
2. A player must move around the world space and select from a choice of experiences laid out in the worldspace.

## Installations and Maintenance

To run the application, proceed with the following installation steps:

1. Use a computer with Windows 8.1 and above Operating System.
   1. Computer must meet the following minimum hardware specifications
      1. Processor: Intel i5-4590 / AMD Ryzen 5 1500X or greater
      2. Graphics card: NVIDIA GTX 970 or AMD Radeon R9 290
      3. RAM: 8GB
      4. Video output: HDMI 1.3
      5. USB: One USB 3.0 and two USB 2.0
2. Acquire Oculus Rift system, which contains:
   1. Oculus Rift Head Mounted Display
   2. Touch Controllers (Left and Right hand)
   3. Two Oculus sensors at a minimum.
3. Connect Oculus Rift system to Computer:
   1. Oculus rift must be connected to a HDMI 3.1 port connected to the Graphics card and a USB 3.0 port.
   2. Each Oculus sensor must be connected to an individual USB 3.0 port.
4. Install Oculus Software from Oculus site:
   1. <https://www.oculus.com/setup/>
5. Download Project from GitHub to a desirable folder location.
   1. https://github.com/FIU-SCIS-Senior-Projects/Learning-with-Augmented-Reality-4.0
6. Open Unity Scene file “Main”
7. Put Oculus Rift on with touch controllers and press play on Unity.

We encourage any future developers working on this project to use Unity cloud services, which allow teams to work on a single project over a cloud instance. For more details, please check out: https://unity3d.com/unity/features/collaborate

## Shortcomings and Wishlist

While our team focused on building a robust system for SKOPE VR 1.0, we had little time to add new experiences or supporting functionality. For future work, we would like to see the following be implemented into SKOPE VR:

* Add experiences through close collaboration with FIU Architecture.
* Multi player support to have students collaborate in the world space.
* Add video playback functionality to experiences.
* Improve animations and UI (Visual Assets).
* Improve stability and performance.

# 

# References

* https://www.oculus.com/rift/
  + Oculus Rift Official Homepage
* https://developer.oculus.com/
  + Oculus Rift development homepage.
* <https://unity3d.com/>
  + Unity HomePage.
* <https://developer.oculus.com/documentation/unity/latest/concepts/book-unity-gsg/>
  + Oculus Rift Documentation Page.