# Team 5 - Shaun Stangler and William Susanto

11/22/2021 - CSS451A Au21

# **Final Project Proposal**

VR Dynamic Vertex Environment Mapping + Future LIDAR Scan Support

### General Overview:

Multiple VR players are in a game world (Or a single VR player with multiple NPC avatars). The game world reacts dynamically to real-world objects and barriers in the physical spaces. The players will see unpassable game world objects instead of artificial barrier lines and can interact with barriers. For the class final, only the manipulatable world and avatar control will be in scope. LIDAR support will come in the next POC.

#### Controls:

- Camera tumble based on VR headset positioning
- Left hand/Right hand movement based on VR headset positioning
- Position teleport
- Position movement w/VR sticks
- Position movement w/physical movement
- Right hand object position/rotation manipulation

### App Structure

- Unity
  - SceneNode
  - VRCameraController
  - PlayerController
  - WorldModel
  - $\circ$  NodePrimitive
  - WorldMesh
    - VertexHeightMap
    - VertexColorMap
  - Network Listeners (Outside of final scope)
  - ConversionTools (Outside of final scope)

- RoomspaceToWorldspace
- IphoneApp (Outside of final scope)
  - LIDARMapController
  - Roomspace Model
  - ApplicationController

Architecture Diagram (TBD)

## Tasks

POC 1 (Final Project)

- Learn basic Unity WMR development
- Build ADO/Github instances (Finished) <u>CSS 451 Final Repos (azure.com)</u>
- Create ADO Work Items
- Integrate Unity WMR SDK into base project
- Add SceneNode/NodePrimitive support
- Build Player and NPC Hierarchies
- Build Player and NPC textures
- Build barrier object textures
- Build World and barrier object textures
- Integrate VR Camera into Player/NPC Head
- Integrate VR Controllers into Player arms
- Integrate NodeControl and selection into solution
- Build World vertex controller
- Build World vertex base maps
- Build collision detection into Player and NPC movement
- Build lighting functionality
- Build initial worldmap positioning of objects and players

POC 2 (Beyond Class)

- Familiarize with Iphone App development
- Create LIDAR maps of a room
- Convert LIDAR maps to height maps
- Build Service code for World Control
- Refactor Unity app to get positioning and world data from server/service
- Build Server/Service instance
- Build World map update service classes
- Build Room scale to world scale classes

## Sketch

# BEHAVIOR



## Schedule

#### POC 1 (Project Final)

Date	Plan
11/28	Unity WMR/VR initial research complete and VR project created
12/3	Basic functionality up to MP4 level features complete
12/5	Player VR camera control and node selection complete
12/8	Have a minimum working example for Progress Demo
12/10	Lighting and collision detection complete
12/13	Have a complete project scope for Final Demo

POC 2 (LIDAR Integration) TBD

# Meeting Technical Requirements

Tech Req 1: SceneNodes

- Player Hierarchy
- NPC Hierarchy
- Obstacle Hierarchy

Tech Req 2: Object Interaction

- World map / Player interaction Collision detection and height/position changes
- Player to NPC Node manipulation
- Obstacle interaction Collision detection and behavior changes
- External manipulation of world map nodes

Tech Req 3: Illumination & Texture

- Player/NPC textures and illumination
- Object Textures and illumination
- Worldmap illumination

Tech Req 4: Camera Manipulation

- VR Player Camera manipulation

Tech Req 5: Working in VR

- WMR support