Unity Portfolio – Esteban Segarra Martinez

Graduate Projects (2020 – 2026) from the University of Central Florida

This is a collection of works developed between 2020 to 2024. Most of these works are a combination of works that were developed towards my dissertation or a combination. I have had experiences with multiple programming languages, platforms, and paradigms.

Contents

Completed Dissertation Project: EFFECTS OF ENVIRONMENTAL AND CAMERA PROPERTIES ON SIMULTANEOUS LOCALIZATION AND MAPPING

Purpose

This project focuses on the development of a pipeline that can be used for rendering photorealistic images from a 3D editor environment to provide an easy-to-use environment for researchers to use as a platform for rendering a synthetic environment that can be used to evaluate Simultaneous Localization and Mapping (SLAM). The environment can be used to insert objects into the environment and develop additional tools for exact camera positioning and rendering. Features such as the camera trajectory, lighting, environment, and camera properties can be adjusted.

Additionally, for my dissertation, I had to incorporate a pipeline setup to evaluate the environment in a real-world SLAM. This was done using an open-source SLAM evaluation framework called SLAMFuse. In my implementation this was extended to account for additional information and then expand the data collection using my framework fork called ORBFuse.

Timeline:

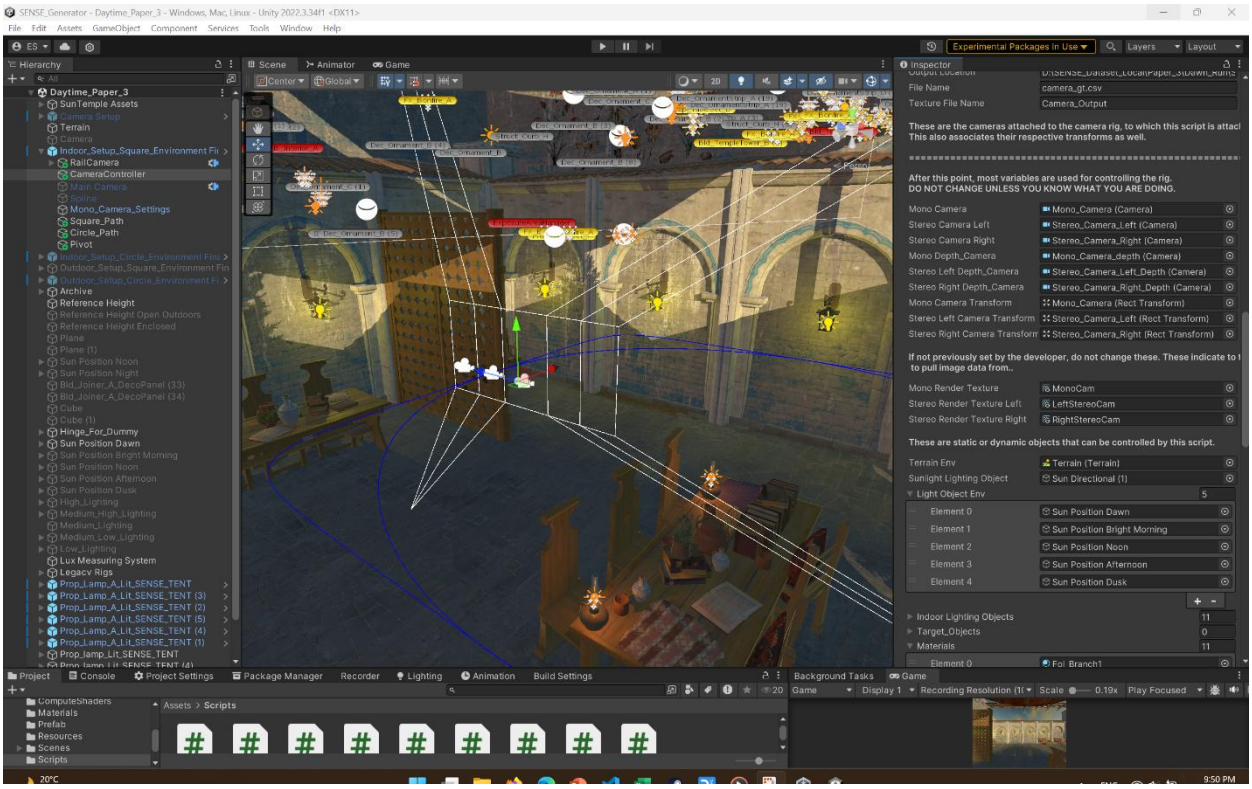
- Project time: 2023 – 2025

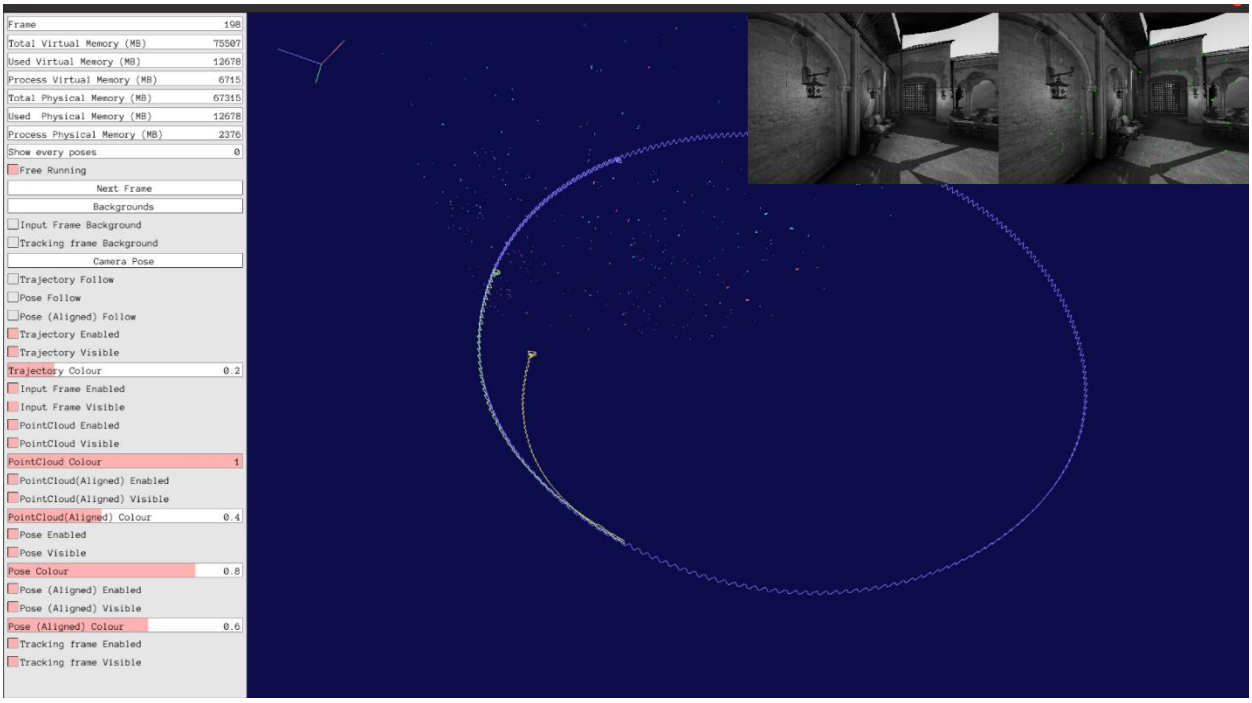
Features:

- Unity as target environment for rendering the environment.
- Developed scripts written in Python to organize and rename image files as required by the ORBFuse SLAM tester
- Extended the code used in recording data from ORB-SLAM 2 and ORB-SLAM 3 in ORBFuse.
- Created datasets using the synthetic environment of Sun Temple Unity environment freely available from the Unity store.
- Utilized the pipeline to systematically create and evaluate different image datasets created by the Unity rendering environment.

Contributions:

- Development of the SENSE synthetic rendering tool
- Development of the ORBFuse Evaluation tool
- Applied tool towards rendering a synthetic environment for use in SLAM


SENSE_Generator - Daytime_Paper_3 - Windows, Mac, Linux - Unity 2022.3.34f1 <DX11>
File Edit Assets GameObject Component Services Tools Window Help
Hierarchy
Daytime_Paper_3
SunTemple Assets
Camera Setup
Terrain
Camera
Indoor_Setup_Square_Environment Fli
RailCamera
CameraController
Main Camera
Spline
Mono_Camera_Settings
Square_Path
Circle_Path
Pivot
Indoor_Setup_Circle_Environment Flas
Outdoor_Setup_Square_Environment Fin
Outdoor_Setup_Circle_Environment Fl
Archive
Reference Height
Reference Height Open Outdoors
Reference Height Enclosed
Plane
Plane (1)
Sun Position Noon
Sun Position Night
Bld_Joiner_A_DecoPanel (33)
Bld_Joiner_A_DecoPanel (34)
Cube
Cube (1)
Hinge_For_Dummy
Sun Position Dawn
Sun Position Bright Morning
Sun Position Noon
Sun Position Afternoon
Sun Position Dusk
High_Lighting
Medium_High_Lighting
Medium_Lighting
Medium_Low_Lighting
Low_Lighting
Lux Measuring System
Legacy Rigs
Prop_Lamp_A_Lit_SENSE_TENT
Prop_Lamp_A_Lit_SENSE_TENT (3)
Prop_Lamp_A_Lit_SENSE_TENT (2)
Prop_Lamp_A_Lit_SENSE_TENT (5)
Prop_Lamp_A_Lit_SENSE_TENT (4)
Prop_Lamp_A_Lit_SENSE_TENT (1)
Prop_lamp_Lit_SENSE_TENT
Scene
Animator
Game
Inspector
Experimental Packages In Use
Layers
Layout
File Name
camera_gt.csv
Texture File Name
Camera_Output
These are the cameras attached to the camera rig, to which this script is attaci
This also associates their respective transforms as well.
After this point, most variables are used for controlling the rig.
DO NOT CHANGE UNLESS YOU KNOW WHAT YOU ARE DOING.
Mono Camera
Mono_Camera (Camera)
Stereo Camera Left
Stereo_Camera_Left (Camera)
Stereo Camera Right
Stereo_Camera_Right (Camera)
Mono Depth_Camera
Mono_Camera_depth (Camera)
Stereo Left Depth_Camera
Stereo_Camera_Left_Depth (Camera)
Stereo Right Depth_Camera
Stereo_Camera_Right_Depth (Camera)
Mono Camera Transform
Mono_Camera (Rect Transform)
Stereo Left Camera Transform
Stereo_Camera_Left (Rect Transform)
Stereo Right Camera Transform
Stereo_Camera_Right (Rect Transform)
If not previously set by the developer, do not change these. These indicate to
to pull image data from..
Mono Render Texture
MonoCam
Stereo Render Texture Left
LeftStereoCam
Stereo Render Texture Right
RightStereoCam
These are static or dynamic objects that can be controlled by this script.
Terrain Env
Terrain (Terrain)
Sunlight Lighting Object
Sun Directional (1)
Light Object Env
5
Element 0
Sun Position Dawn
Element 1
Sun Position Bright Morning
Element 2
Sun Position Noon
Element 3
Sun Position Afternoon
Element 4
Sun Position Dusk
Indoor Lighting Objects
11
Target_Objects
0
Materials
11
Element 0
Foi Branch1
Project
Console
Project Settings
Package Manager
Recorder
Lighting
Animation
Build Settings
Background Tasks
Game
Assets > Scripts
ComputeShaders
Materials
Prefab
Resources
Scenes
Scripts
Display 1
Recording Resolution (1
Scale
0.19x
Play Focused
20°C
9:50 PM


Frame 198
Total Virtual Memory (MB) 75507
Used Virtual Memory (MB) 12678
Process Virtual Memory (MB) 6715
Total Physical Memory (MB) 67315
Used Physical Memory (MB) 12678
Process Physical Memory (MB) 2376
Show every poses 0
Free Running
Next Frame
Backgrounds
Input Frame Background
Tracking frame Background
Camera Pose
Trajectory Follow
Pose Follow
Pose (Aligned) Follow
Trajectory Enabled
Trajectory Visible
Trajectory Colour 0.2
Input Frame Enabled
Input Frame Visible
PointCloud Enabled
PointCloud Visible
PointCloud Colour 1
PointCloud(Aligned) Enabled
PointCloud(Aligned) Visible
PointCloud(Aligned) Colour 0.4
Pose Enabled
Pose Visible
Pose Colour 0.8
Pose (Aligned) Enabled
Pose (Aligned) Visible
Pose (Aligned) Colour 0.6
Tracking frame Enabled
Tracking frame Visible

Past Dissertation Project: CAPTURING AND EVALUATING POINT CLOUDS FOR COMPUTER VISION APPLICATIONS

I-SPIES: InfraStructure for Photorealistic Image and Environment Synthesis - Award Abstract # 2120240

Purpose

This project focuses on the development of a pipeline that can be used for rendering photorealistic images from a 3D editor environment so as to provide an easy-to-use environment for researchers to use as a platform for rendering point clouds, inserting objects into the environment, and developing additional tools for exact camera positioning and rendering.

Timeline:

- Project time: 2022 – 2025

Features:

- 3DsMax as target environment
- Developed scripts written in Python, used native wrapper MxtoPy to convert Python scripts
- Developed scripts for measuring distances of paths in 3DsMax, scripts to output the total specifications of paths
- Optimized the ray-tracing pipeline for use with Arnold 5.4.3.10
- Optimized the environment to be as photorealistic as possible
    - This was verified by running image quality assessment metrics resulting in a score between 82% – 90%
- Created datasets using the point clouds we collected

Contributions:

- Development of the I-SPIES pipeline
- Development of the RecolorCloud tool

CLEVER Capturing and Logging Ecological Virtual Experiences and Reality (CLEVER) - Award Abstract # 2232448 –

CLEVER Tool: CLOVR: Collecting and Logging OpenVR

Purpose:

The purpose of this project was to develop an open-source tool that can be used to capture motion capture data, video, and audio data. This project was published in IEEE VR 2024 as a workshop paper. We also included a dataset compromised of six games that were recorded using the tool. The tool was developed in Unity.

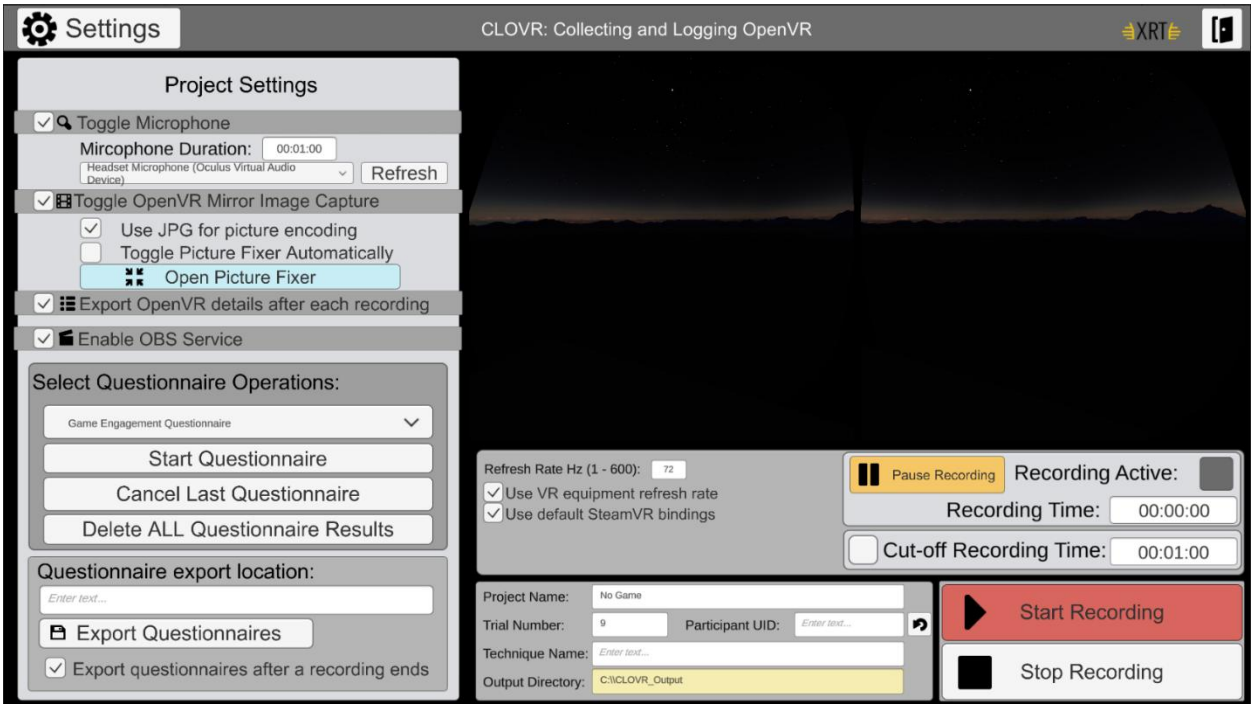
Timeline:

- October 2023 – March 2024

Features:

- Open source; Github link:
- Developed in Unity
- Uses Open Broadcasting Software for recording audio and video
- Motion capture software for OpenVR
- Any device can be recorded so long as it can connect to OpenVR
- User interface that can control the recording of the session
- Uses a in-VR user interface for participant survey collection

Pictures:



Additional Projects:

I-SPIES Tool: Final Cloud v1 (RecolorCloud)

Purpose:

This is the first implementation of the RecolorCloud tool. This tool is intended to edit point clouds in such a way that it could be easily done simultaneously for a large density point cloud. The density of this point cloud has been tested up 650M points, however, there is no theoretical limit to the density of the point cloud except hardware requirements to open and edit the point cloud.

Features:

- Written in Python 3.6.1
- Open3D point cloud interface interaction
- Qt5 Python interface
- No limit to the size of point cloud being edited
- Supported LabelCloud bounding boxes for editing/labeling
- Automated recoloring via algorithmic selection of points and recoloring
- User-selected range recoloring for points
- Solid-color recoloring segmentation (For segmentation versions of point clouds)
- Automated removal of chunks of undesired points on point cloud

Pictures:

I-SPIES Tool: Final Cloud v2

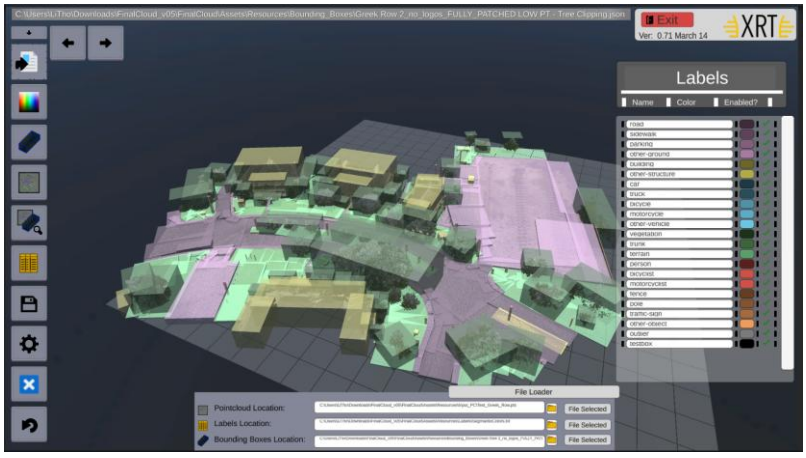
Purpose:

This is an extension to RecolorCloud, which is to have a more extensive user interface which could replace and remove the requirement of having an extrenal tool that could create the bounding boxes that limit the editing space of the point cloud. This tool was developed in Unity, which managed the bounding boxes, labels, and also managed to edit the point cloud directly using Open3D.

Features:

- Developed on the Unity Engine x Python interface
- Reused many of the python scripts used in RecolorCloud for automated recoloring
- Developed the easiest form of visualizing the point cloud in-application without having to open LabelCloud or a slow interface
    - The point cloud was automatically resampled at a lower resolution to allow real-time visualization of the point cloud at an acceptable frame-per-second speed
    - Supports the following key features:
        - Recoloring the point cloud by bounding boxes
        - Edited and created new bounding boxes in the editor (Removes the dependence on LableCloud)
        - Can create individual labels for the bounding boxes and recolor sections of the point cloud utilizing per-label colors
        - Supported all automated and manual recoloring tools as in RecolorCloud
            - Circular, average, or ranged recoloring of the point cloud

Pictures:



Academic/Technical Unity Projects

Project name: FinalCloud: Point Cloud Editing Tool

Date: Fall of 2022

Website Link to Source:

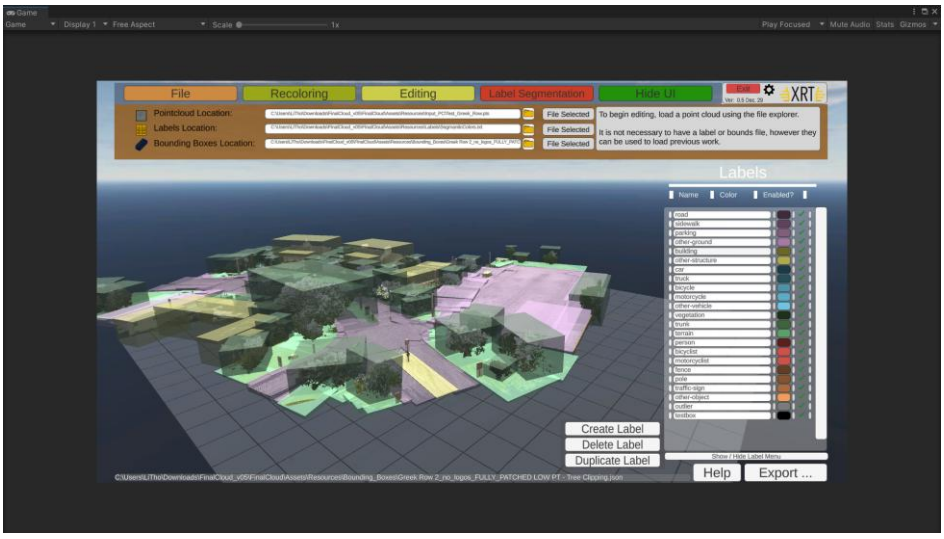
- Source will be released when the project has attained viable publication.

Status: In Development

Purpose: This is a tool being developed for the XRT Lab at UCF for Dr. Ryan McMahan. The intent is to have a Unity-based tool that can edit point clouds as well as having the ability to be expanded and modified by users for additional features. The primary goal of this tool was to implement an open-source solution for editing point cloud files. This tool, being part of my dissertation and currently the sole developer for the tool. Key features of the project are: being able to select individual boxes of the point cloud and edit them by selecting relevant points, recoloring the points, and segmenting the point cloud into individual point clouds.

Features:

- Incorporates Python into Unity by locating or allowing Python to be found from the system
- Python is used to edit the point cloud while Unity handles the interfaces and positions of the bounding boxes and labeling system.
- Users can import their bounding boxes from third party tools or start a new file
- Users can label the bounding boxes with their own labeling system
- Tool can automatically downscale a large point cloud to be visualized on the Unity tool
- Designed to be built-and-ran as a standalone application with Python being the only dependency



Project name: UCF Camp Connect Point cloud Viewer Application

Date: July 2022

Website Link to Source:

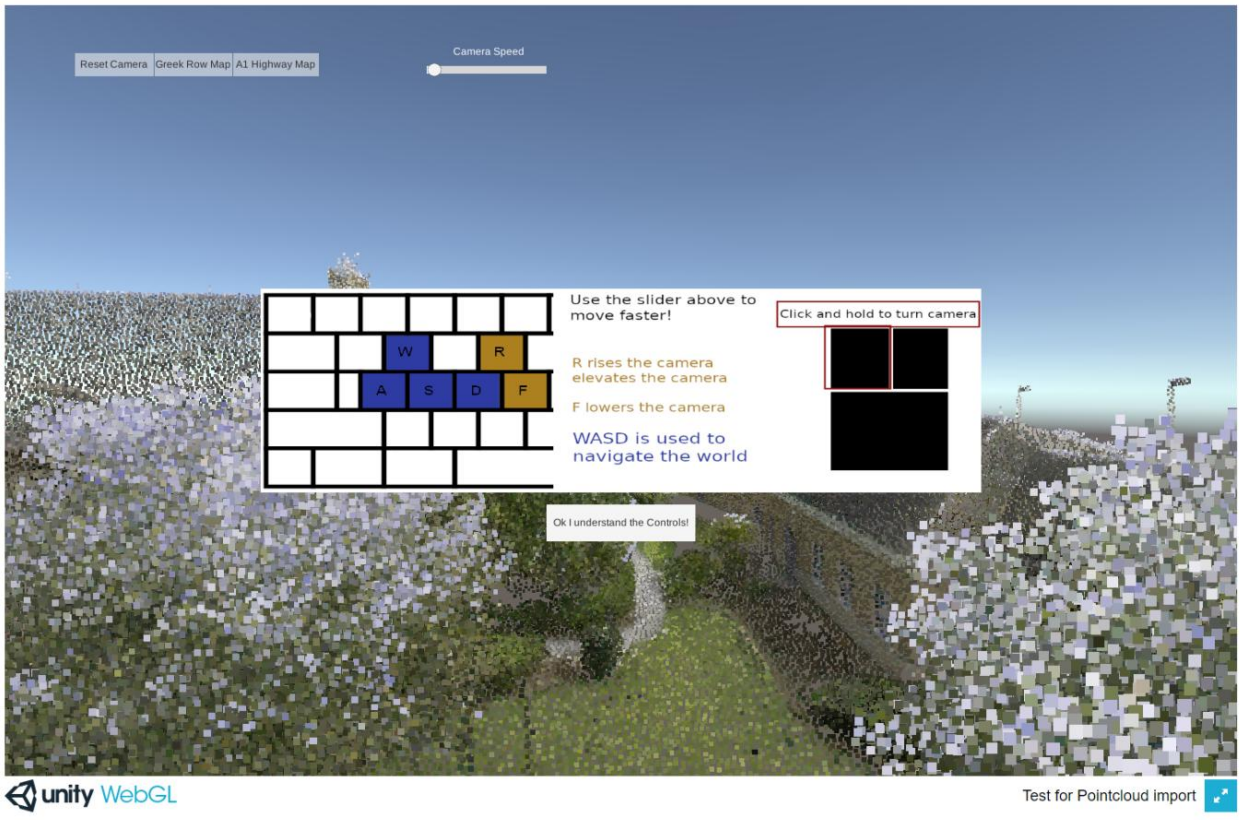
- https://github.com/OvercodedStack/UCF-Camp-Connect-2022-PointCloud-Viewer-Game

Status: Completed

Purpose: This was a proof-of-concept Unity project that was tossed together in a week for an event that happens once a year at UCF called Camp Connect. The proof of concept was used for showing students what are point clouds and how are they represented in a 3D virtualization. This demo later was utilized in another project named FinalCloud, which is a point cloud editing tool also developed by myself.

Features:

- First project where point clouds were used for visualization
- Simple interactions
- WebGL build and support
- Support for two point cloud maps (Was available information on hand)




Reset Camera
Greek Row Map
A1 Highway Map
Camera Speed
unity WebGL
Test for Pointcloud import

Project name: VR Selection Techniques Prototype

Date: Spring of 2022

Website Link to Source:

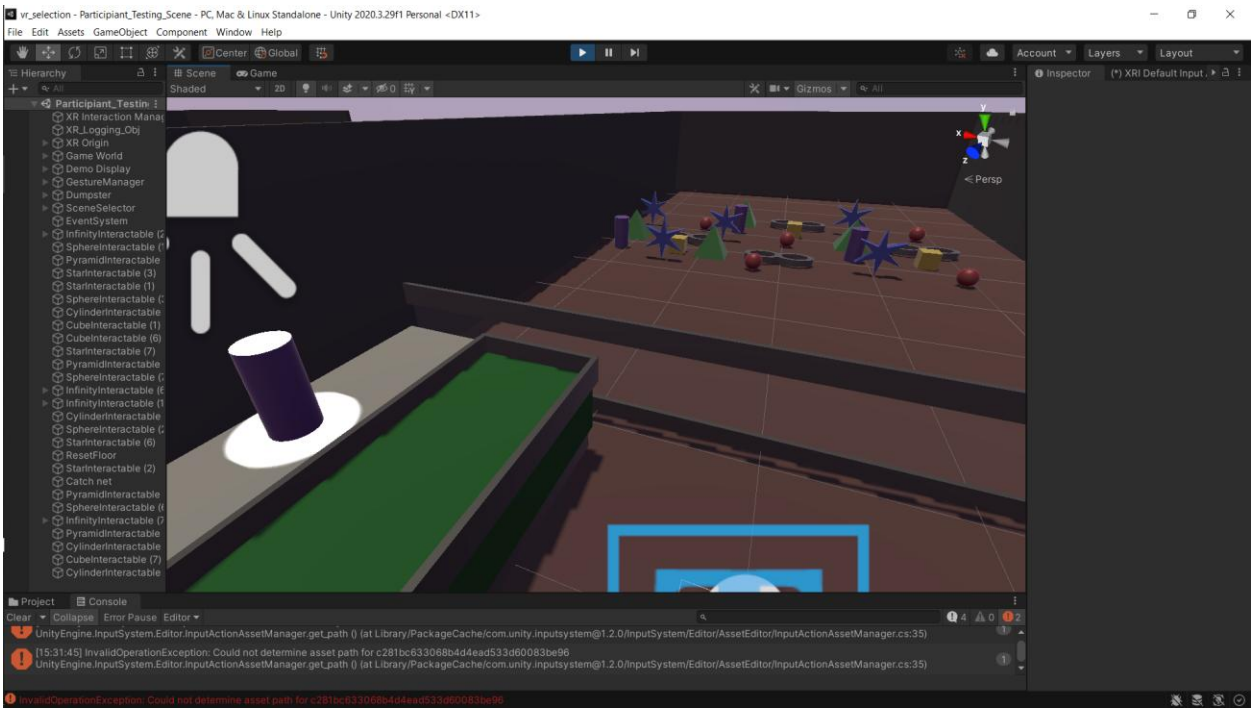
- Source available on request

Status: Completed for assignment – Development continued privately

Purpose: The purpose of this project was part of a class for creating a project that could be used for interfacing with the virtual environment in order to select objects using different selection techniques in virtual reality. This could be either through a gestural circle or a flashlight selection technique. I was collaborating with a teammate; Mykola Maslych, I developed the interface, and Mykola incorporated a package for gesture recognition.

Features:

- Developed using the Oculus Quest 2 VR headset
- Recorded data using CSV files and instantaneously objects would be selected based on the selection technique used
- Application was used for a small batch of participants on campus to record technique performance and preferences for techniques



Project name: Gesture Recording Studio (Mixed Reality Project)

Date: Fall 2021

Website Link to Source:

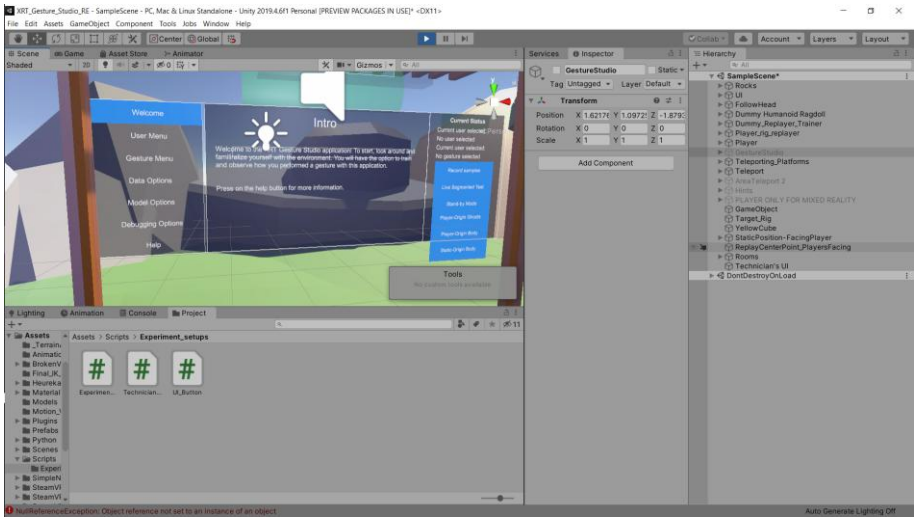
- Source Available on Request

Status: Completed for assignment – Still being developed for testing trials.

Purpose: This project was designed to record the interactions that a user would have with gestures recorded by VR equipment. This project was a collaboration between me and another teammate; Austin Mathew. I worked primarily with the interface and the Unity-side interfacing between the Python backend developed by my teammate. Austin primarily was working on a machine learning algorithm using an altered version of a gesture recognition technique.

Features:

- Record and back-play gestures after recording
- Mannequin was designed to be controlled by the recorded gesture
- A second box was shown to the user to see the positions of the boxes and recorded gestures
- SteamVR support



Project name: Virtual Reality Engineering Course

Date: Fall of 2020

Website Link to Source:

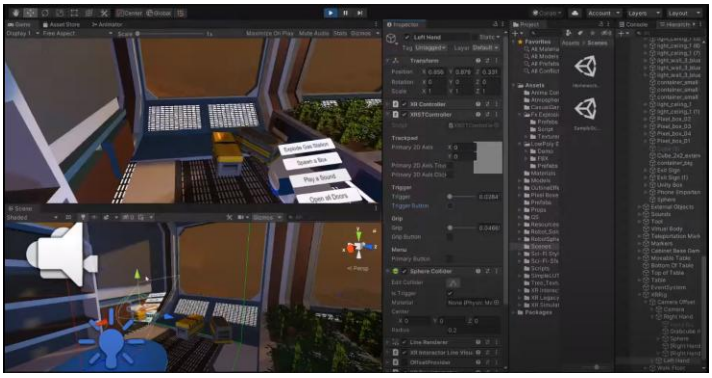
- https://www.youtube.com/watch?v=hVaOlzvyhOQ&list=PLdsxhOIzXlc7LJzR1QDxFaHfMwHxZY-tr
- Source available on request (Backed up on drives)

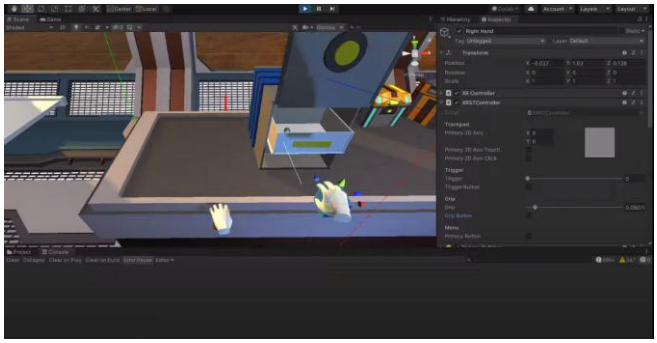
Status: Completed

Purpose: This was a VR project that was completed for a class project that was started and ended throughout the semester. The assets were obtained from the Unity Store but were arranged to create an immersive and atmospheric environment. Class points were arranged for producing a environment that produced a highly immersive environment

Features:

- Full VR environment using SteamVR
- Immersive with sound and other VR features such as interactions and events based on interacting with the hand-held UI.
- Environment was ready for gameplay in the event it was necessary
- Virtual interactions were possible with some objects





Project name: Android Application in Unity Engine

Date: August 2019

Website Link to Source:

- https://github.com/OvercodedStack/ANDROID_UNITY_UI-Summer-2019-NIST
- https://github.com/OvercodedStack/CRPI_MIDDLEWARE_INTEGRATION-Summer-of-2019-NIST
- https://github.com/OvercodedStack/MOTION_CAPTURE_UNITY-Summer-of-2019-NIST
- https://github.com/OvercodedStack/CRPI-UI-DOCUMENTATION-Summer-of-2019

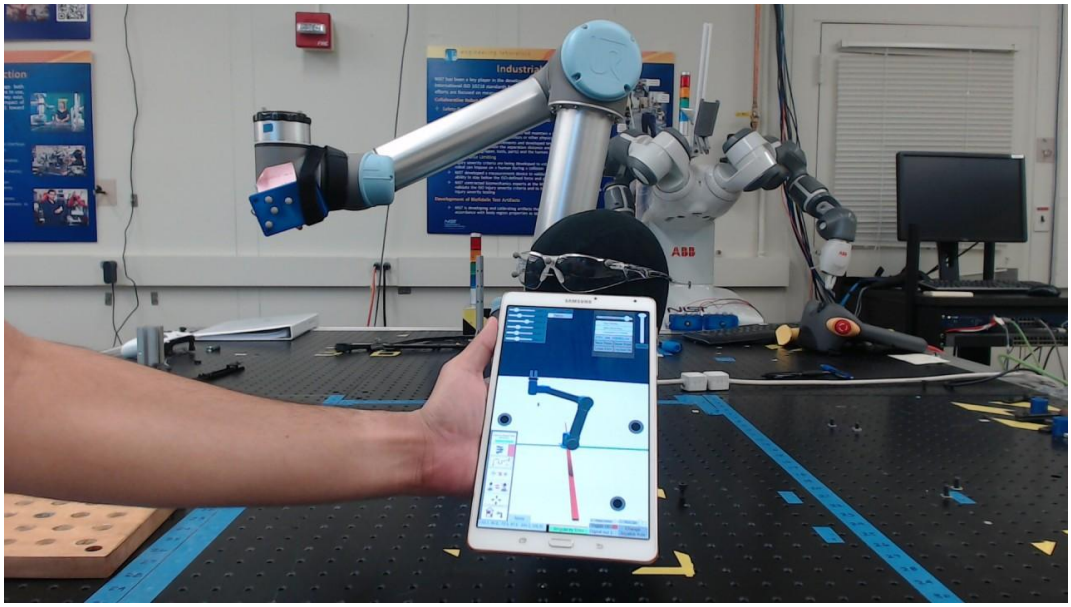
Status: Completed

Purpose:

This project was also produced while working at NIST SURF 2019 for an Android application that controlled a UR5 robot while holding a tablet or compatible device. The intent was to be able to measure user performance and interactions with the user interface while handling the robot's commands.

Features:

- First app in Unity Engine with Android support
- UR5 interaction and support
- Performance measurements recording and storing user finger presses and buttons pressed
- Data exported to a CSV or compatible file format



Project name: HoloLens Integration with PioneerDX

Date: 2017 - 2018

Website Link to Source:

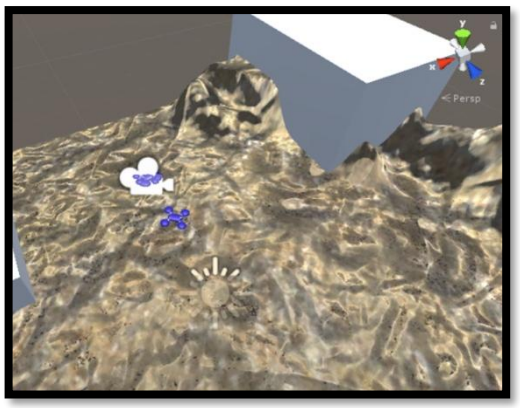
- https://github.com/OvercodedStack/HoloLens-Drones-ROS-Operation-Florida-Polytechnic-University
- https://github.com/OvercodedStack/UDP_SERVER_FLPOLY
- https://github.com/OvercodedStack/SharedExperience_FLPOLY

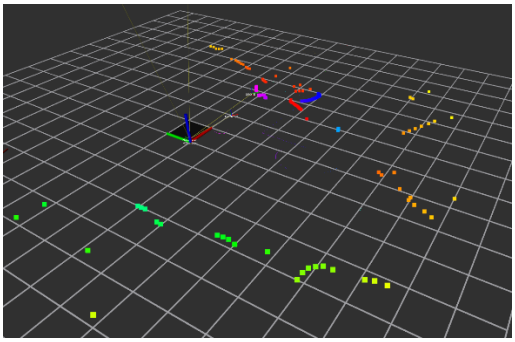
Status: Completed

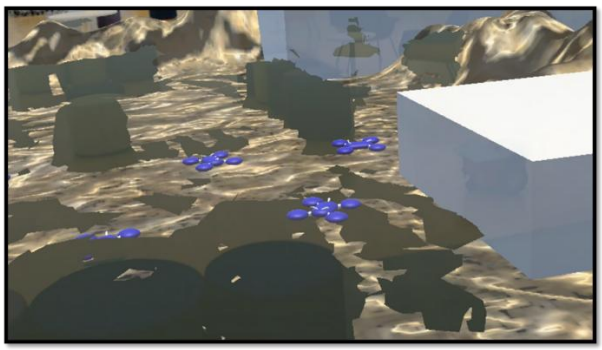
Purpose: The purpose of this project was to utilize a HoloLens MXR headset and use it as a pass-through sensor to spatially sense the environment around it and pass the information to a on-board computer for mimicking a LiDAR sensor to drive a PioneerDX robot utilizing a simple algorithm to avoid walls.

Features:

- Used ROS Bridge to communicate with a ROS-enabled system
- Utilized (At the time) HoloLens API for tapping into spatial sensors
- Virtual objects interacted with the spatial sensing to “confuse” the real world robot’s navigation
- An extension of the project allowed virtual drones to collaborate or play in the environment







Date: August 2018

Website Link to Source:

- https://github.com/OvercodedStack/C-Plus-Plus-API-Implementation-Manus-VR
- https://github.com/OvercodedStack/C-Sharp-API-Implementation-ManusVR
- https://github.com/OvercodedStack/CRPI-UR5-Manus-VR

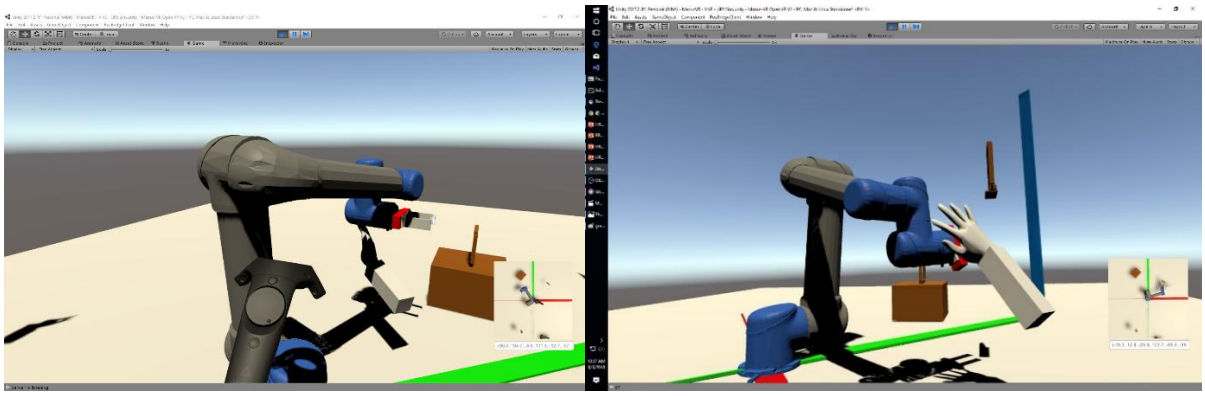
Status: Completed

Purpose: The purpose of this project was to develop a Unity-based project that could implement the Manus VR finger-tracking sensors for controlling a virtual AND real representation of a UR5 robotic arm. Users who interacted with the robot would be interacting with the robotic representation of the robot in the virtual environment and Unity would translate the position and rotations of the virtual robot to the UR5 robot controller. The intent of this project was to create an interface that would allow control between the user and the robot and have a way to record the interactions between the user and the user interface and provide means of measuring worker performance with different interfaces.

This project was completed during a NIST SUFT internship during summer of 2018. This project involved both, VR components and developing components between interacting with SteamVR and the actual UR5 robot.

Features:

- Incorporated Manus VR
- SteamVR
- Controlled a real-world robot using an NIST-developed API



Honorable Mentions:

FruitNinja (2022): A proof-of-concept where users can slash fruit with a sword and play a game where they can reach a high score by slashing a variety of enemies and fruit approaching the player. This project was developed in collaboration with a teammate; Ryan Ghamandi.