



Problem

Originally designed to be a mode of implementation for an interactive painting software, we have decided to create a framework for Virtual Reality (VR) programs. The project, now aimed towards individual work, has the goal of building a framework for VR that can be used for educational programs. Taking advantage of the VIP design process, we would like to create a grounds for future programming that adheres to three main functions:
1. Uses VR Technology
2. Builds a framework for future iterations of projects
3. Can be used for educational purposes

Current System

Currently, the VR framework is in a completed state, ready to be iterated upon by a new generation of programmers. Features include:
1. Unity project that provides a base to create games within
2. Test showcase that can demonstrate physics and movement

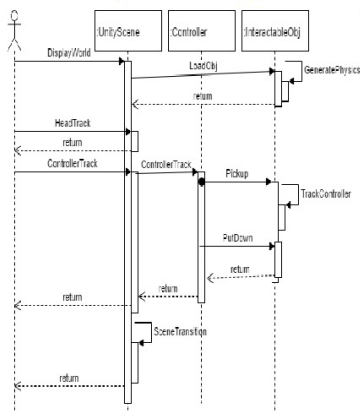
Requirements

- 1. Framework that can be built upon using Unity engine
2. System that takes advantage of the HTC Vive headset/controller

Solution

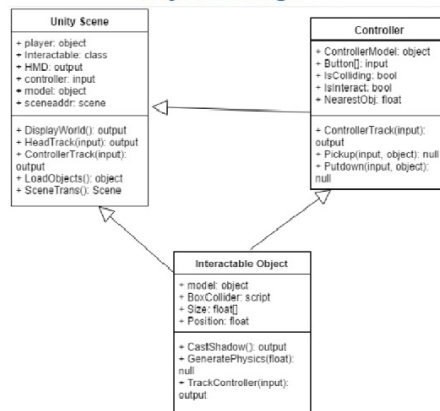
Our implementation is a project that takes advantage of Unity and its associated plugins for VR games development. Our first iteration of the system uses the HTC Vive as our VR hardware, but plans for other devices are possible in future iterations.
1. Using HTC Vive, develop a prototype system within Unity
2. Create a framework that can be developed for educational purposes
3. SteamVR and Virtual Reality Toolkit (VRTK) plugins are used (All work is the first version/iteration of this project)

System Design



Functions for Interactable Objects follow a unique sequence. The unity engine is aware of all actors in the system.

Object Design



Class diagram of a Unity scene that uses the framework

Implementation

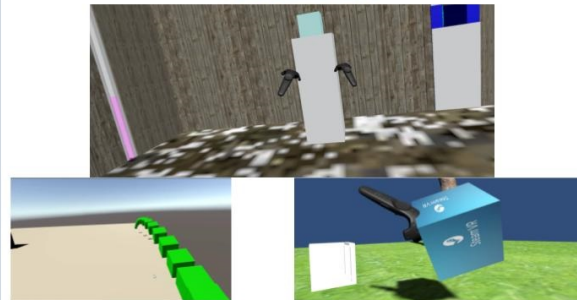
- 1. Physics showcase with firing range and interactable objects
2. Maze game that takes advantage of movement and controls



Verification

Test Case 1: Pick up object
Purpose: Test to see if object framework can pick up object and track the controller
Preconditions: user has no other objects picked up
Action: User moves controller over the object's model, presses the pickup button, then moves the controller
Expected result: Object should start tracking the controller's movements: velocity, rotation, trajectory when the button is released.
Actual result: Same as Expected Result.
Test Case 2: Teleport
Purpose: Test to see functionality of teleportation as a mode of movement
Preconditions: Case 1: User has no object picked up Case 2: User has an object picked up
Action: Case A: User holds the teleport button and points to a location where the pointer is red Case B: User holds the teleport button and releases it when the pointer is green
Expected result: Case 1-A/2-A: Camera remains in place Case 1-B: Camera teleports to location Case 2-B: Camera teleports, object is dropped
Actual results: All as expected

Screenshots



Summary

The VR Framework, while having little to show off in its current implementation, provides an excellent base for future programmers to create educational Virtual Reality experiences.

Acknowledgement

The material presented in this poster is based upon the work supported by Jacob Leschen. I am thankful to the help that I received from my group members, Alain Galvan, Jose Morgan, Justin Alvarez.