VR Game Creator Program Structure

by Chris Kelley

# Overview

The VR Game Creator Progam allows VR games to be easily created. When using this program, the user will interface with the Unity Game Engine, and will write code for their games using Visual Studio. The program enables easy game creation through the use of a Unity extension and an API that is used to manipulate in-game assets.

# Unity Extension

The code for the Unity extension is located in the Unity files:

*Assets -> Editor -> CreationWindow.cs*

The Creation Window is a window that displays on the Unity interface. It has the following capabilities:

1. Create VR ready objects
2. Create API capable scripts
3. Change scenes

## Create Objects

The Creation Window displays many VR ready objects that may be created. These base objects are from the folder:

*Assets -> Resources*

From the base object, the creation window adds the components that allow for VR specific activities.

## Create Scripts

When an object is highlighted in the project hierarchy, the Creation Window will give the option to add a script to the object (or open an existing script). When a script is created, it is placed in the folder:

*Assets -> API -> UWBsummercampAPI -> customscripts*

When the script is created through the Creation Window, it inherets from the class ‘coreObjectsBehavior’, giving the script (and object) access to the custom API.

## Switch Scenes

The Creation Window allows a scene (level) to be selected. The scenes that may be validly selected are in the folder:

*Assets -> Scenes*

If a new scene is created, it must contain the following objects to function properly:

1. GameManager
2. Player
3. Teleporting
4. HUD

These objects may be simply copied from one of the already existing scenes into the newly created scene.

# API

## coreObjectsBehavior

When a scripts is created on an object through the creation window, it inherits from ‘coreObjectBehavior’, located at:

*Assets -> API -> UWBsummercampAPI*

Public methods in the file may be called on the child class (script created on the object), allowing for things such as: object movement, detecting controller actions, and detecting collisions.

## coreManagerBehavior

The coreManagerBehavior class is located at:

*Assets -> API -> UWBsummercampAPI*

It does not provide an API to be used on objects, rather it provides more of a background function. The purpose is to manage static variables and provide some inter-object communication.

The ‘GameManager’ in a given scene descends from ‘coreManagerBehavior’.

# Virtual Reality Capabilities

The program is currently compatable with Windows Mixed Reality and HTC Vive.

To enable virtual reality, the program uses the SteamVR 2.0 API.