VRTK - Virtual Reality Toolkit

A productive VR Toolkit for rapidly building VR solutions in Unity3d.

Supported SDK	Download Link
VR Simulator	Included
SteamVR Unity Asset	SteamVR Plugin
Oculus Utilities Unity Package	Oculus Utilities
*Google VR SDK for Unity	Google VR SDK for Unity

**experimental*

Documentation

The documentation for the project can be found within this repository in DOCUMENTATION.md which includes the up to date documentation for this GitHub repository.

Alternatively, the stable versions of the documentation can be viewed online at http://docs.vrtk.io.

Frequently Asked Questions

If you have an issue or question then check the FAQ.md document to see if your query has already been answered.

Getting Started

> *VRTK requires a supported VR SDK to be imported into your Unity3d Project.*

- Clone this repository `git clone https://github.com/thestonefox/VRTK.git`.
- Open `VRTK` within Unity3d.
- Add the `VRTK_SDKManager` script to a GameObject in the scene.

Instructions for using the VR Simulator

- Drag the `VRSimulatorCameraRig` prefab from the VRTK/Prefabs into the scene.
- Select the GameObject with the `VRTK_SDKManager` script attached to it.
  - Select `Simulator` for each of the SDK Choices.

    - Click the `Auto Populate Linked Objects` button to find the relevant Linked Objects.
- Use the Left Alt to switch between mouse look and move a hand.
- Press Tab to switch between left/right hands.
- Hold Left Shift to change from translation to rotation for the hands.
- Hold Left Crtl to switch between X/Y and X/Z axis.
- All above keys can be remapped using the inspector on the `VRSimulatorCameraRig` prefab.
- Button mapping for the VR control are as follows:
    - Grip: Left mouse button
    - Trigger: Right mouse button
    - Touchpad Press: Q
    - Button One: E
    - Button Two: R

Instructions for using the SteamVR Unity3d asset

- Import the SteamVR Plugin from the Unity Asset Store.
- Drag the `[CameraRig]` prefab from the SteamVR plugin into the scene.
- Check that `Virtual Reality Supported` is ticked in the `Edit -> Project Settings -> Player` menu.
- Ensure that `OpenVR` is added in the `Virtual Reality SDKs` list in the `Edit -> Project Settings -> Player` menu.
- Select the GameObject with the `VRTK_SDKManager` script attached to it.
    - Select `Steam VR` for each of the SDK Choices.
    - Click the `Auto Populate Linked Objects` button to find the relevant Linked Objects.
- Optionally, browse the `Examples` scenes for example usage of the scripts.

Instructions for using the Oculus Utilities Unity3d package

- Download the Oculus Utilities from the Oculus developer website.
- Import the `OculusUtilities.unitypackage` into the project.
- Drag the `OVRCameraRig` prefab from the Oculus package into the scene.
- Check that `Virtual Reality Supported` is ticked in the `Edit -> Project Settings -> Player` menu.
- Ensure that `Oculus` is added in the `Virtual Reality SDKs` list in the `Edit -> Project Settings -> Player` menu.
- Select the GameObject with the `VRTK_SDKManager` script attached to it.
    - Select `Oculus VR` for each of the SDK Choices.
    - Click the `Auto Populate Linked Objects` button to find the relevant Linked Objects.

Instructions for using the Google VR SDK for Unity

- Open a new or existing project in Unity (5.4.2f2-GVR13 or other version with Daydream integration).
- Import asset package GoogleVRForUnity you downloaded from Google.
- Build Settings:
  - Target platform: `Android`
- Player settings:
  - Virtual Reality Supported > Daydream
  - API Level: `Nougat`
  - Bundle Identifier and other settings for use with Android.
- In Hierarchy, create empty GameObject named `DaydreamCameraRig`.
  - Move or create a Camera as child of `DaydreamCameraRig`, reset its transform `position: 0,0,0`.
  - Add `GvrControllerPointer` prefab from `Assets/GoogleVR/Prefabs/UI`.
  - Add `GvrControllerMain` prefab from `Assets/GoogleVR/Prefabs/Controller/`.
  - Add `GvrViewerMain` prefab (enables view in editor play mode).
- Disable Daydream's native pointer tools.
  - Camera object, disable or remove GvrPointerPhysicsRaycaster component (if present).
  - GvrControllerPointer/Laser, disable or delete.
- In Hierarchy, create an empty GameObject named `[VRTK]`.
- Add component `VRTK_SDKManager`
- Add a child empty GameObject named `RightController`.
  - > Note: Daydream supports only one controller, LeftController will not be used. If present, can be disabled or deleted.
- SDK Selection
  - In Inspector, choose Quick Select SDK: Daydream
  - In Player Settings, ensure Scripting Define Symbols: `VRTK_SDK_DAYDREAM`
- Linked Objects:
  - Click `Auto Populate Linked Objects`, that should set:
  - Actual Boundaries: `DaydreamCameraRig`
  - Actual Headset: `DaydreamCameraRig/Camera`
  - Actual Left Controller: `empty`
  - Actual Right Controller: `DaydreamCameraRig/GvrControllerPointer/Controller`
- Controler Aliases:
  - Model Alias Left Controller: `empty`
  - Model Alias Right Controller: `DaydreamCameraRig/GvrControllerPoints/Controller`
  - Script Alias Left Controller: `empty`
  - Script Alias Right Controller: `[VRTK]/RightController`

What's In The Box

VRTK is a collection of useful scripts and concepts to aid building VR solutions rapidly and easily in Unity3d 5+.

It covers a number of common solutions such as:

- Locomotion within virtual space.
- Interactions like touching, grabbing and using objects
- Interacting with Unity3d UI elements through pointers or touch.
- Body physics within virtual space.
- 2D and 3D controls like buttons, levers, doors, drawers, etc.
- And much more...

Examples

A collection of example scenes have been created to aid with understanding the different aspects of VRTK.

A list of the examples can be viewed in EXAMPLES.md which includes an up to date list of examples showcasing the features of VRTK.

The examples have all been built to work with the SteamVR Plugin by default, but they can be converted over to using the Oculus Utilities package by following the instructions for using the Oculus Utilities package above.

> *If the examples are not working on first load, click the* `[VRTK]` *GameObject in the scene hierarchy to ensure the SDK Manager editor script successfully sets up the project and scene.*

Made With VRTK

Many games and experiences have already been made with VRTK.

Check out the MADEWITHVRTK.md document to see the full list.