VRTK - Virtual Reality Toolkit

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

Supported SDK	Download Link
VR Simulator	Included
SteamVR	SteamVR Plugin
Oculus	Oculus Utilities
*Ximmerse	Ximmerse Unity SDK
*Daydream	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 offers a VR Simulator that works without any third party SDK, but VR device support requires a supported VR SDK to be imported into the Unity project.*

Using the example project & scenes

- Download or clone this repository.
- Open the folder in Unity to load the project.
- Have a look at the included example scenes.

The example scenes support all the VRTK supported VR SDKs. To make use of VR devices (besides

the included VR Simulator) import the needed third party VR SDK into the project.

Using VRTK in your own project

- Download or clone this repository.
- Import the `Assets/VRTK` folder into your Unity project.
- Add the `VRTK_SDKManager` script to a GameObject in the scene.

The SDK Manager handles setting up everything to use the various supported SDKs via `VRTK_SDKSetup`s. To use a VR SDK the following steps are needed:

- Download and import the SDK into the project.
- Create a new empty game object.
  - Add `VRTK_SDKSetup` to it.
  - Add the VR SDK's game objects (e.g. the Camera Rig) as children.
  - On the SDK Setup set the `SDK Selection` to the respective VR SDK.
  - Make sure all the `Object References` are set correctly by `Auto Populate` or set them manually.
- On the SDK Manager under `Setups` add a new slot and select the SDK Setup for that slot.

Repeat the steps above to add additional SDK Setups to the SDK Manager.

If the `Auto Load` setting on the SDK Manager is enabled the SDK Setups are automatically loaded in the order they appear in the list: * The first Setup that is usable (i.e. initializable without errors and the HMD is connected) will be used. * If a Setup can't be used the next one will be tried instead. * If no Setup is usable VR support is disabled.

The SDK Manager allows switching the used SDK Setup at runtime. To add a simple overlay GUI to do so add the `SDKSetupSwitcher` prefab from `VRTK/Prefabs` to the scene.

Read the rest of this document for more detailed instructions on how to set up and use the supported VR SDKs.

VR Simulator

Instructions for using the VR Simulator

- Follow the initial steps above by adding the `VRSimulatorCameraRig` prefab from `VRTK/Prefabs` as a child of the SDK Setup game object.
- 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 Ctrl to switch between X/Y and X/Z axis.
- Additional button mappings can be found on `SDK_InputSimulator` on the prefab.
- All button mappings can be remapped using the inspector on the prefab.

SteamVR

Instructions for using SteamVR

- Import the SteamVR Plugin from the Unity Asset Store.
- Follow the initial steps above by adding the `[CameraRig]` prefab from the plugin as a child of the SDK Setup game object.

Oculus

Instructions for using Oculus Utilities

- Download the Oculus Utilities from the Oculus developer website.
- Import the `.unitypackage`.
- Follow the initial steps above by adding the `OVRCameraRig` prefab from the Utilities as a child of the SDK Setup game object.
- On the `OVRCameraRig` in the scene find the `OVRManager` and set its `Tracking Origin Type` to `Floor Level`.

Ximmerse *(experimental)*

Instructions for using Ximmerse

- Download the Ximmerse Unity SDK from the Ximmerse SDK Github page.
- Import the `.unitypackage`.
- Follow the initial steps above by adding the `VRCameraRig` prefab from the SDK as a child of the SDK Setup game object.
  - It is recommened to use `Floor Level` as the `Tracking Origin Type`, with the position of `VRCameraRig` set to `(0f, 0f, 0f)`. `Eye Level` can also be used, in this case it is recommended to set the position to `(0f, 1.675f, 0f)`.
  - Make sure `SimplePicker` is **not** attached to any of the game objects `cobra02-L` and `cobra02-R`. `SimplePicker` is provided by the Ximmerse SDK but using the script may break VRTK's grab functionality.
- Switch the build settings in `File > Build Settings...` to `Android`.

> Currently Ximmerse 6DOF tracking is only supported on Android. 3DOF tracking is supported on both iOS and Android. Ximmerse are getting MFI certification from Apple at the moment.

Daydream *(experimental)*

Instructions for using Daydream

- Open a new or existing project in Unity that offers Daydream integration.
- Download the Google VR SDK from the Google developer website.
- Import the `.unitypackage`.
- Switch the build settings in `File > Build Settings...` to `Android`.
- In `Edit > Project Settings > Player` set the following:
  - API Level to `Nougat`.
  - Bundle Identifier and other settings for use with Android.
- In the `Hierarchy` window, create a new empty GameObject named `DaydreamCameraRig`.
- Add the following as children of `DaydreamCameraRig`:
  - A new `Camera`.
  - The `GvrControllerPointer` prefab from `GoogleVR/Prefabs/UI`.
  - The `GvrControllerMain` prefab from `GoogleVR/Prefabs/Controller`.
  - The `GvrViewerMain` prefab from `GoogleVR/Prefabs` (enables the view in editor play mode).
- Disable Daydream's native pointer tools by removing or disabling `DaydreamCameraRig/GvrControllerPointer/Laser`.
- Follow the initial steps above by adding the `DaydreamCameraRig` object as a child of the SDK Setup game object.

> Note: Daydream supports only one controller, the left scripting alias controller of VRTK will not be used.

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 example scenes support all the VRTK supported VR SDKs. To make use of VR devices (besides the included VR Simulator) import the needed third party VR SDK into the project.

Made With VRTK

Many games and experiences have already been made with VRTK.

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