Unity Package VRTK

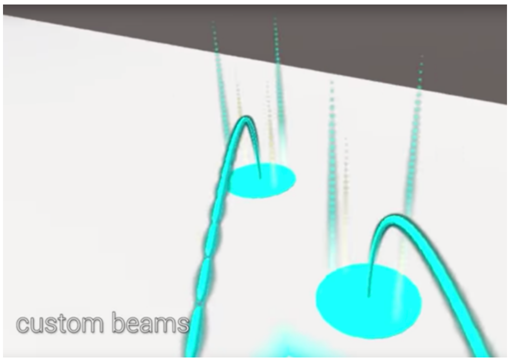
Software environment: Unity 2018.3.8
Hardware environment: Oculus Rift/HTC Vive

Revised 25 March 2019

Practice objectives:

1. Become familiar with a VR headset
2. Setup your first VR application
3. Discover different types of locomotion and interaction design for Virtual Reality
4. Fast prototyping
5. Make use of Assets provided in Unity

Description :



Goal: VRTK is a Unity package which combines several useful tools to help game developers working in VR, including interactions, means of locomotion, SDK managers, etc. In this practical work, you will learn how to implement basic interaction and locomotion using the VRTK assets. Then, you will be able to use this package in your class project in order to save a lot of time in your future VR projects. This tutorial is based on videos produced by the VRTK developers.

Getting Started:

You can either follow the videos in order to set up VRTK, or follow these instructions.

Video: Getting Started With SteamVR - https://youtu.be/tyFV9oBReqg

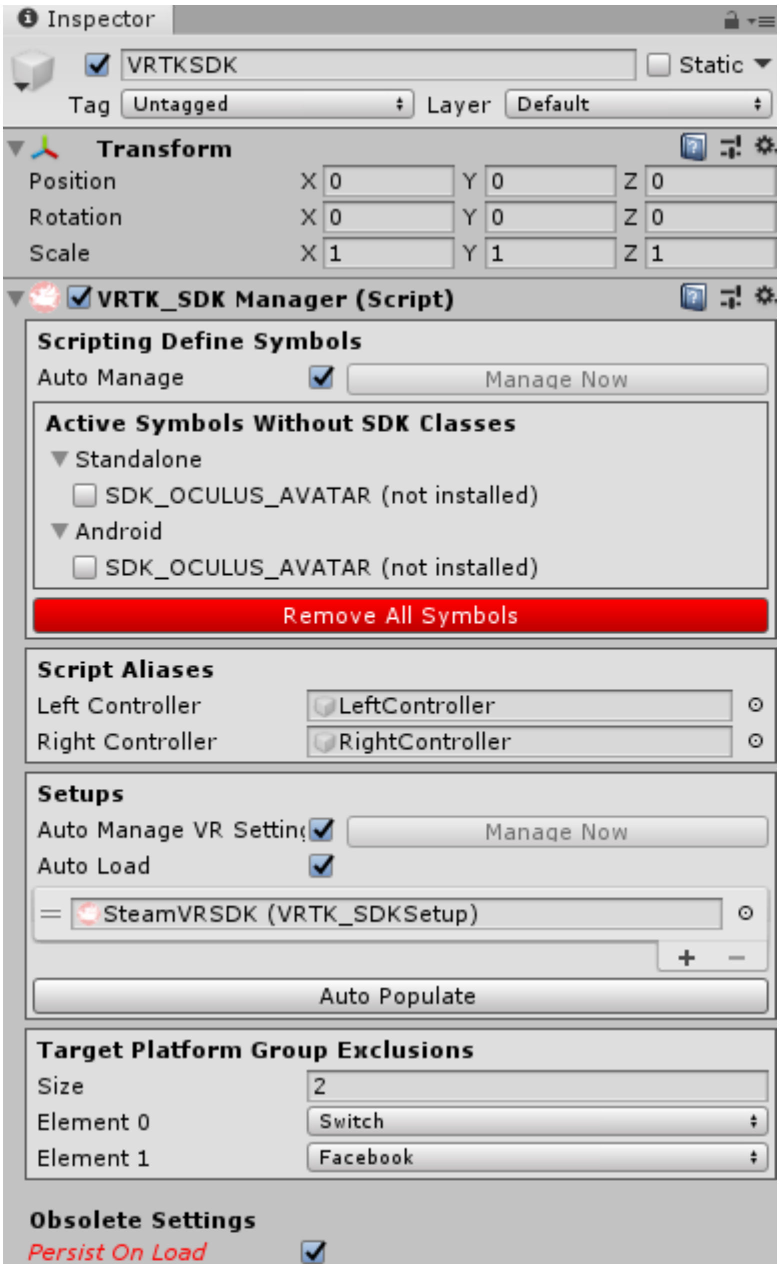
- **FIRST, Download SteamVR from here and import into Unity. Only the version 1.2.3 is compatible with VRTK :** https://github.com/ValveSoftware/steamvr_unity_plugin/releases/download/1.2.3/SteamVR.Plugin.unitypackage
- Download and install VRTK from the Unity asset store

Video: Getting Started With Oculus - https://youtu.be/psPVNddjgGw

- FIRST, Download the ***Oculus Integration*** package directly from the asset store.
- Download and install VRTK from the Unity asset store

Configure the Scene

- Create a new Scene, then create an empty GameObject, called VRTKSDK
- Add the *VRTK_SDK Manager* script to the *VRTKSDK* object
- Add an appropriate GameObject for either Oculus or Steam, under the *VRTKSDK* object
    - Oculus
        - Name the new object OculusSDK
        - From Oculus>VR>Prefabs drag and drop OVRCameraRig inside OculusSDK. (Select floor level)
        - From Oculus>Avatar>Content>Prefabs drag and drop LocalAvatar into OculusSDK and make sure SDK_OCULUS_AVATAR is enabled in VRTKSDK
    - Steam
        - Name the new object SteamVRSDK
        - From SteamVR>SteamVR>Prefabs drag and drop SteamVR and CameraRig into SteamVRSDK
- Within this new object
    - Click Add Component
    - Search for and add VRTK_SDK Setup
    - Under *SDK Selection* click Quick Select, and choose either Steam VR or Oculus Rift
    - Click the *VRTKSDK* object, and under Setups, click the + button. Select the appropriate option for the SDK Setup, either OculusSDK or SteamVRSDK
    - Be sure to click the *Persist on Load selection* (in RED)
    - Under the object VRTKSDK, click AutoPopulate
    - Under the SteamVRSDK or OculusSDK object, click Populate Now
- At the top level of the scene, Right click and create a 3D Object>Cube
    - Name it Floor
    - Make it flat – Scale 100, 0.01, 100
    - Apply a color, or ideally, the texture Default_Floor (search for it in your VRTK assets)



Inspector

☑ VRTKSDK □ Static

Tag Untagged Layer Default

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

☑ **VRTK_SDK Manager (Script)**

Scripting Define Symbols

Auto Manage ☑ Manage Now

Active Symbols Without SDK Classes

Standalone

□ SDK_OCULUS_AVATAR (not installed)

Android

□ SDK_OCULUS_AVATAR (not installed)

Remove All Symbols

Script Aliases

Left Controller LeftController

Right Controller RightController

Setups

Auto Manage VR Settin(☑ Manage Now

Auto Load ☑

SteamVRSDK (VRTK_SDKSetup)

Auto Populate

Target Platform Group Exclusions

Size 2

Element 0 Switch

Element 1 Facebook

Obsolete Settings

Persist On Load ☑

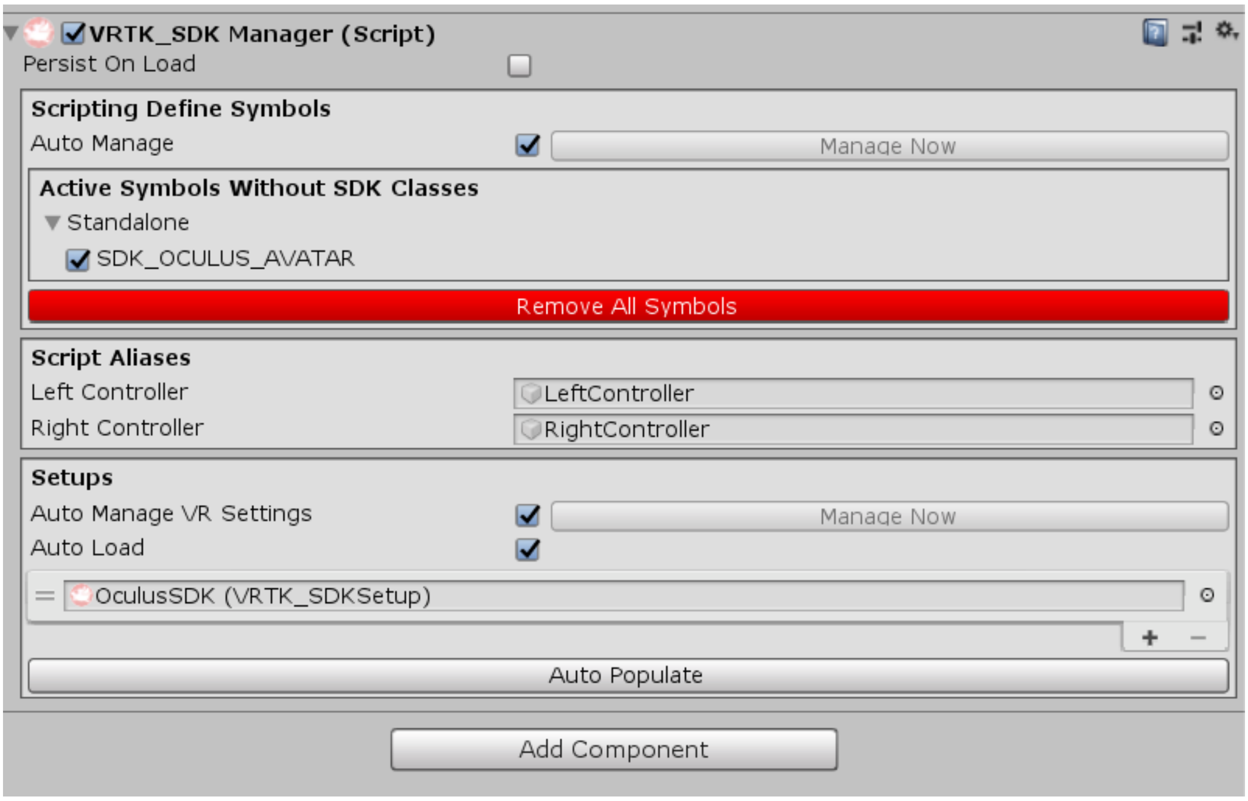
For Oculus, the Oculus SDK Should be selected, not SteamVR

Prepare the Controller and Teleportation Scripts

- Create an empty GameObject and name it VRTK_Scripts
    - Create another empty GameObject and name it LeftController
    - Add the following scripts (components) to the object
        - Controller Events script

        - Pointer Script
        - Straight Pointer Renderer
            - Drag and drop the Straight Pointer Renderer name into the field *Pointer Renderer* in the VRTK_Pointer script

  - Duplicate LeftController, and name it RightController
  - Drag the two controllers into their appropriate fields in the SDK Manager script, so that Unity can identify your controller objects
- Under the Object VRTK_Scripts, add another empty GameObject and call it PlayArea
  - Within PlayArea, add the VRTK_Height Adjust Teleport or VRTK_Basic Teleport scripts (experiment with one at a time)



☑ VRTK_SDK Manager (Script)

Persist On Load □

Scripting Define Symbols

Auto Manage ☑ Manage Now

Active Symbols Without SDK Classes

Standalone

☑ SDK_OCULUS_AVATAR

Remove All Symbols

Script Aliases

Left Controller: LeftController

Right Controller: RightController

Setups

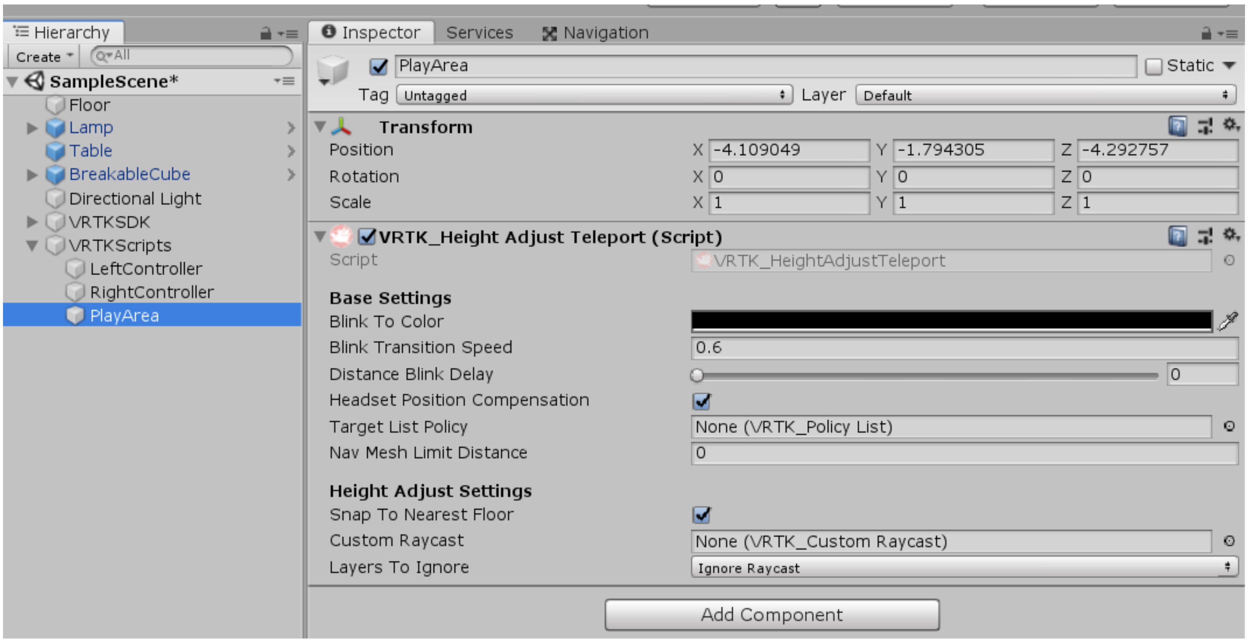
Auto Manage VR Settings ☑ Manage Now

Auto Load ☑

OculusSDK (VRTK_SDKSetup)

Auto Populate

Add Component



SampleScene*
- Floor
- Lamp
- Table
- BreakableCube
- Directional Light
- VRTKSDK
- VRTKScripts
  - LeftController
  - RightController
  - PlayArea

☑ PlayArea □ Static

Tag Untagged Layer Default

Transform

Position	X	-4.109049	Y	-1.794305	Z	-4.292757
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

☑ VRTK_Height Adjust Teleport (Script)

Script: VRTK_HeightAdjustTeleport

Base Settings

Blink To Color

Blink Transition Speed 0.6

Distance Blink Delay 0

Headset Position Compensation ☑

Target List Policy None (VRTK_Policy List)

Nav Mesh Limit Distance 0

Height Adjust Settings

Snap To Nearest Floor ☑

Custom Raycast None (VRTK_Custom Raycast)

Layers To Ignore Ignore Raycast

Add Component

Testing Teleport:

Play your project. Now, when you press the touchpad, rays should be cast from the controllers.

Now by clicking on the touchpad and pointing at somewhere on the floor the ray will become green and by releasing it you will be able to teleport to that location.

You can change different options such as the activation button, color, shape of the ray (using for example VRTK_Bezier Pointer Renderer Script instead of VRTK_Straight Pointer Renderer Script). For more information refer to https://youtu.be/2IWAwFDhX2M and watch the videos about teleporting.

Interaction (Grabbing, Touching...):

Look at the different scenes from Assets>VRTK>Examples in order to understand how the different interactions, pointers and locomotion work using VRTK. Try it with your Oculus Rift/HTC Vive!

Have a look at the different scripts used by each Game Object, especially in the scenes **[004 - Locomotion] Teleporting, [005 - Interactions] InteractableObjects** and **[006 - Locomotion] MovementTypes**

Implementation:

Your goal is to create a small scene to test what you have just learned.

Your scene should contain:

- At least one locomotion technique (teleportation, joystick walking, etc...)
- At least 3 interactable objects (using different scripts)

When finished, show your small scene to the TAs.
If you have any questions during the session, do not hesitate to ask the TAs!

Good Luck!