Using Vive Tracker and Controllers without a headset

MOTIVATION

There are several reasons why one would want to use Vive Trackers and Controllers without a headset. These devices are sub-millimetric positional trackers, which are useful for dozens of applications: low-cost motion capture, marker-less camera tracking, pointing devices, althlete performance assesment, among others.

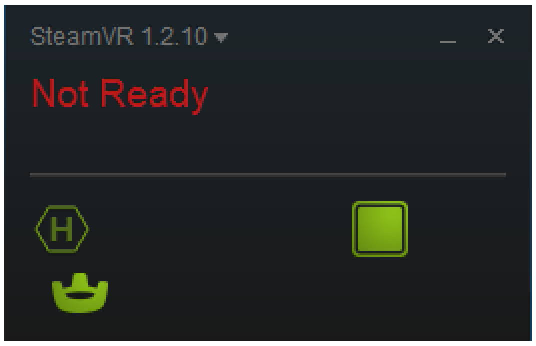
Unfortunately, this use case is not officially supported by Valve so many of Valve's tools (like the Vive Input asset) do not work if one has the headset disconnected. Valve's staff will not answer any questions regarding these topics on Viveport forums.

This document is a guide to using HTC Vive Tracker's without the headset in Unity. The scripts in this package will allow you to:

- Get the tracker/controller data.
- Get button input from controllers.
- Interact with Unity UI.
- Record data from trackers to a json file.
- Build your project without manually having to copy OpenVR dlls every time you build.

STEAMVR CONFIG

1. Make a backup and edit the **<Steam_Directory>/steamapps/common/SteamVR/resources/settings/default.vrsettings** file to the following settings:
   - **"requireHmd": false**
   - **"forcedDriver": null**
   - **"activateMultipleDrivers": true**
2. Make a backup and edit the file **<Steam_Directory>/steamapps/common/SteamVR/drivers/null/resources/settings/default.vrsettings.** Set **"enable": true.**
3. Restart SteamVR if it was open. Unplug your headset and Linkbox if they were plugged.
4. Plug your dongle if it if you hadn't done it previously. Make sure your tracker/controller is properly synced. You should see something similar to this (ignore the *Not Ready* warning):



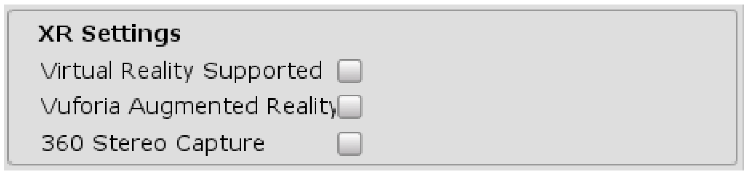
DOWNLOADING STEAMVR PACKAGE

This package works with SteamVR version 1.2.3, which is an old release. Get it from here. If you get the latest version (currently 2.2.0) you will be able to use the Vive Tracker, but not the controllers. If you only want to use the Vive Tracker, you're fine with version 2.2.0, but you may have to delete the files that have compile errors.

Download and import the SteamVR package **before** downloading this project or package.

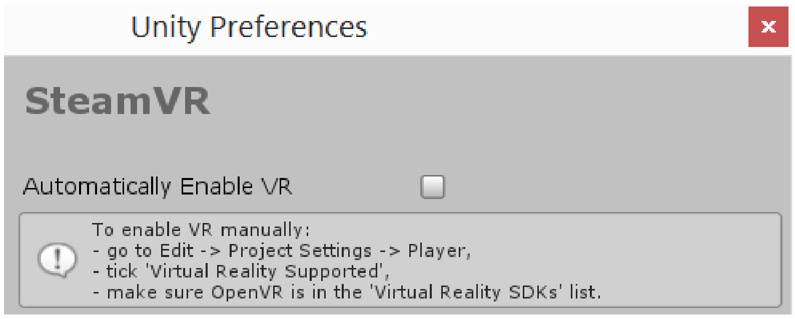
UNITY CONFIG

Make sure **Virtual Reality Supported** is **unchecked** in Unity's player settings. **This is important.** If checked, Unity will crash. Also, uncheck the "**Automatically Enable VR**" checkbox in Unity's preferences (SteamVR section).



XR Settings

Virtual Reality Supported	☐
Vuforia Augmented Reality	☐
360 Stereo Capture	☐



Unity Preferences

SteamVR

Automatically Enable VR ☐

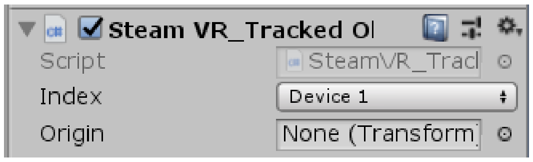
```
To enable VR manually:
- go to Edit -> Project Settings -> Player,
- tick 'Virtual Reality Supported',
- make sure OpenVR is in the 'Virtual Reality SDKs' list.
```

SAMPLE SCENE

The SampleScene uses a tracker to interact with a UnityUI. If you click on the **Record Data** toggle, the data from your controller will be recorded to a .json file.

GETTING POSITION DATA FROM HTC VIVE TRACKER

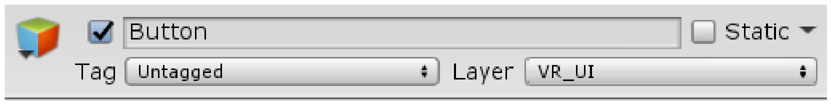
1. Add the **TrackerEvents** prefab to your hierarchy. This will get the poses from the CVRSystem and forward them to all **SteamVR_TrackedObject** components.
2. Add the **ControllerUI** or the **SimpleTracker** prefab to your hierarchy.
3. Run the scene, and increment the device index, until your GameObject is moving.


Steam VR_Tracked Ol
Script SteamVR_Tracl
Index Device 1
Origin None (Transform)

GETTING BUTTON DATA AND INTERACTING WITH GUI

1. Use the **ButtonListener** example as a starting point if you want to use button clicks on MonoBehaviours.

2. The **ControllerEvents** script, which should be added to the controller gameobject (which as a **SteamVR_TrackedController** component), sends Raycasts to the objects in its **layerMask**. In the examples, the layer is called “VR_UI”. Therefore, you should follow these steps to interact with UI objects:
   - Make sure your controller object has a **ControllerEvents** component, set to the correct device index.
   - Add a Button to a Canvas, and set its layer accordingly to your **ControllerEvents** component.


Button
Static
Tag Untagged
Layer VR_UI

   - Add a thin box collider to your button, or use the **Tools/ViveUtilities/Add colliders to buttons** or **Tools/ViveUtilities/Add colliders to element** command to generate the collider for you. The former will generate colliders for all buttons in with the “**VR_UI”** layer mask. The latter will do so for an element that you have selected (blue) in the hierarchy.

3. Note that you can add events for any canvas element, or any 3D object that has a collider and has the correct layer mask. Only **OnPointerEnter**, **OnPointerExit** and **OnSubmit** events are implemented.

BUILDING THE PROJECT

If you want to build your project, you must assure the **openvr_api.dll** is included in the **Plugins** folder of your build. You can copy it manually. Otherwise, edit the **CopyOpenVR.cs** script and point it to the correct location and it will be done automatically.