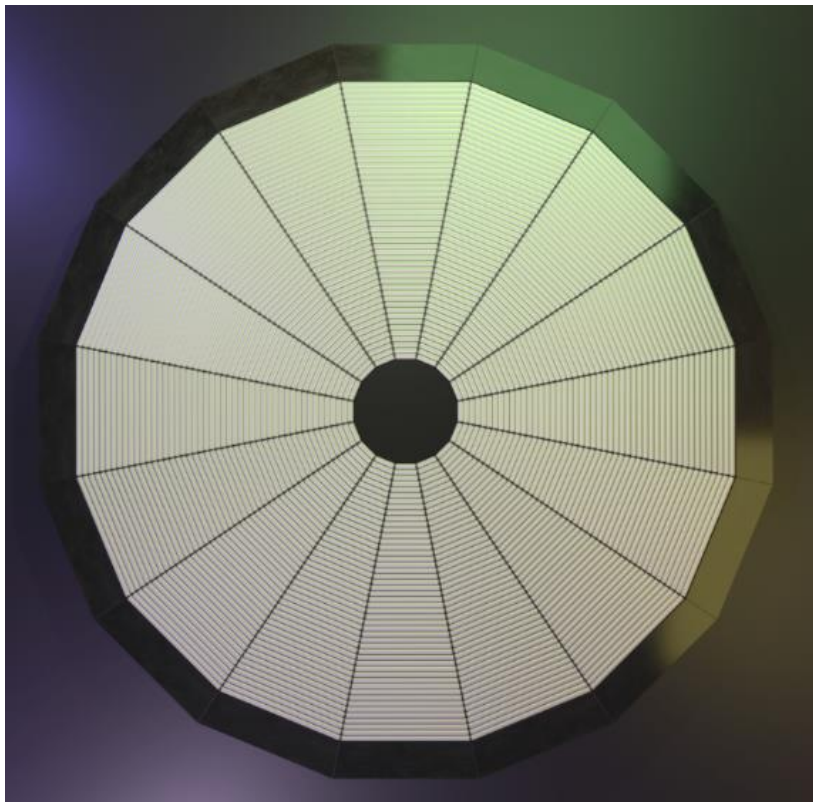


Controlling applications using the Omnideck



Copyright 2018 Omnifinity

Introduction

This document outlines the various ways you can control an application using the Omnideck. For details on each mode please contact us.

Ways to control the Omnideck

We can provide various ways to control a game. The preferred solution is to use our Omnideck API which is tailored for game engines.

Method	What	Integration quality level	Comment
A	Native integration using the Omnideck API	3	Available today: Unity API Available soon: Unreal API
B1 B2	B1 Touchpad emulation B2 Touchpad control with custom Omnideck controller driver	1 2	Available today: HTC Vive Omnideck driver
C	Custom integration with a specific application	Depends on functionality of application	
D	Keyboard and mouse	0	An emulator exists but this is not a recommended solution

Method A – Native integration using the Omnideck API

We provide an API for popular game engines allowing you to get best-in-class integration of the Omnideck in your application.

The Unity API can be freely downloaded from github:

<https://github.com/Omnifinity/Unity-SteamVR-API>

We currently use the HTC Vive system as our preferred Tracking/VR-system. If you use another Tracking/VR-system please contact us and we can help you with the integration.

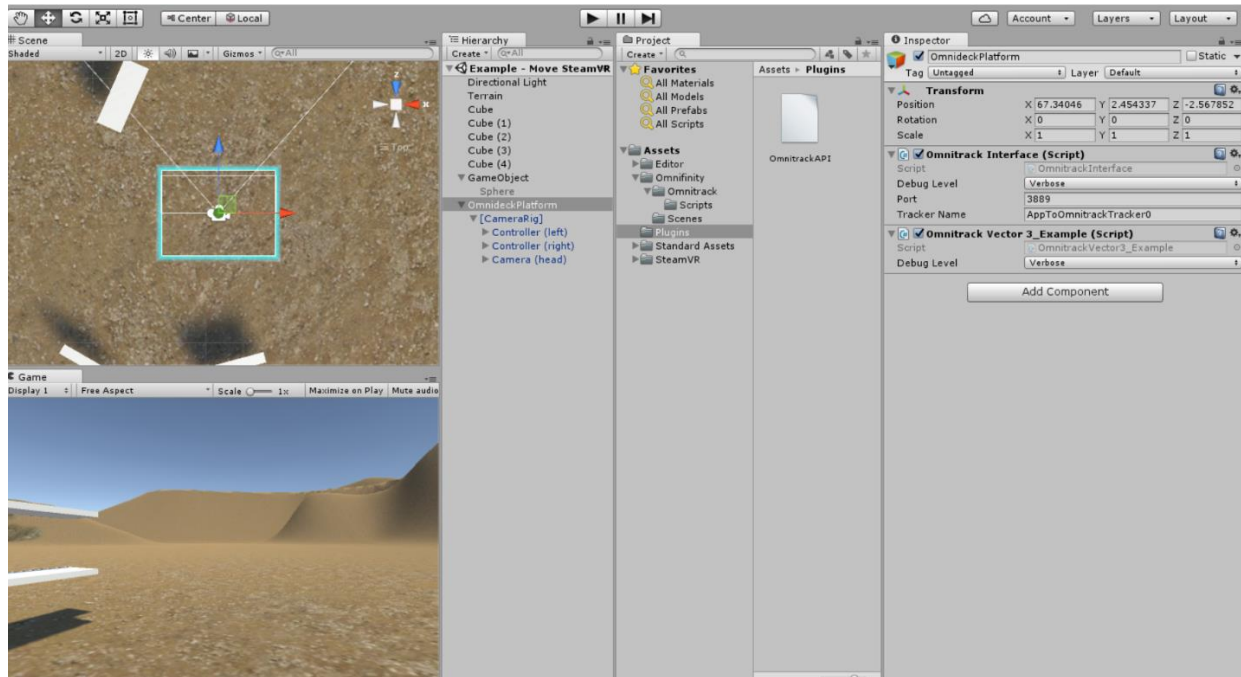


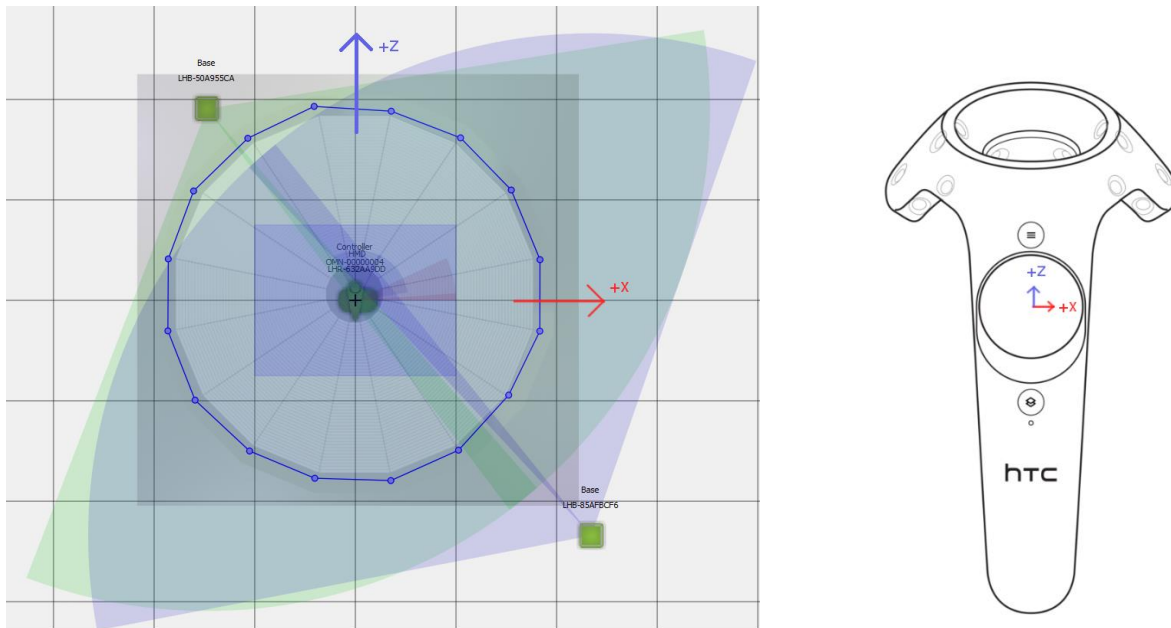
Figure 1 Unity API implementation example

Method B1 – Touchpad emulation

Note: Beta/experimental feature. This can break at any point of time.

We can provide a beta-driver for the HTC Vive system using touchpad emulation to control applications that already support touchpad locomotion today.

This is an experimental feature that can be used as an initial step to try your application on the Omniceck – hopefully with little overhead. Any serious implementations should however rely on Method A.



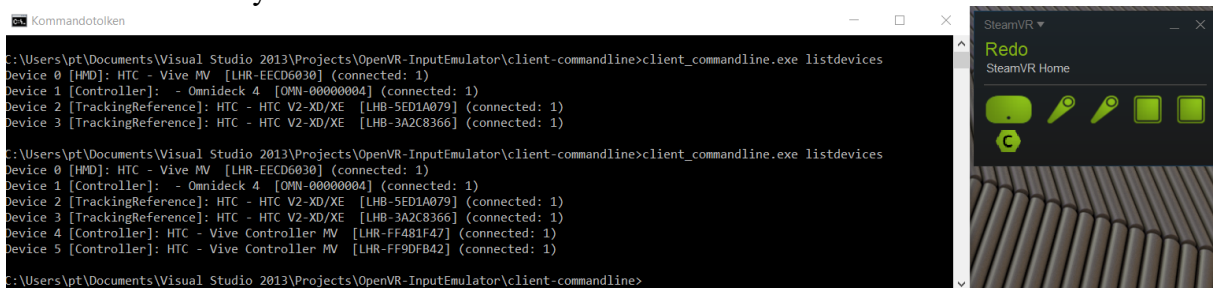
Figur 2 Overview of the HTC Vive tracking space and chaperone bounds

Method B2 – Touchpad control with custom Omniceck controller driver

Note: Beta/experimental feature. This can break at any point of time.

We can provide a beta-driver for the HTC Vive system that creates a new Omniceck controller device that sends out touchpad events. You can fetch this data in Unity or Unreal and control the movement of your character/camera. The implementation is similar to the path you would take when adding support for e.g. the Vive Tracker (<https://www.vive.com/us/vive-tracker/>).

This is an experimental feature that can be used as an initial step to try your application on the Omniceck with a bit more overhead compared to Method B1. Any serious implementations should however rely on Method A.



Figur 3 Custom HTC Vive driver for the Omniceck

Method C – Custom integration with a specific application

We can provide you with details on data formats to make it possible to control your application using the Omnideck.

Method D – Keyboard and mouse

Although we can provide emulators for keyboard movement (WASD) we do not recommend this discrete method of controlling any serious VR application since it is incompatible with how humans locomote in general.