InstantVR Free Support

Version: 3.1
Date: August 28, 2015

Features

The Free version of InstantVR supports the following input:

- keyboard/mouse: using standard WASD input and mouse look functions
- game controller: standard controller input is supported with left analog stick for walking around and right analog stick for looking around.
- Oculus Rift DK1 & DK2: Oculus rift is detected automatically. When it is connected the usual stereoscopic view and distortion are applied. Moreover, the mouse look and/or right analog stick input are replaced by the rotational input movements from the Rift.
- Google Cardboard for Android. Only works when playform is set to Android. Only rotational tracking is supported.

Prerequisites

- The current distribution only supports development on Windows and Android..
- Unity 5.1 or higher.
- Oculus Runtime for Windows version 0.6.0.1 or higher is required. The runtime can be downloaded from the Oculus VR site.

Usage

To use InstantVR you need an environment first. The minimum is a flat terrain with a directional light, but you can make it as complex as you want with lots of meshes, rigidbodies and colliders. You are only bound by the limits of Unity and the computer used to drive the game.

You should not include a camera of other first or third person objects in your game. This is fully handled by InstantVR.

To complete the scene you should drag one of the prefabs named ‘MH_…’ from the folder InstantVR into your scene.

Now you can press play and walk around your wonderful environment.

General configuration

InstantVR

The instantVR contains references to the 6 transforms which move the avatar. These transforms can be placed anywhere within the Hierarchy.

IVR_Walking

The walking script implements walking around using input.

Walking forward and backward is implemented using the standard keys:

- on keyboard: w key = forward, s key = backward
- on gamepad: left analog stick up/down
- on Razer Hydra: left analog stick up/down

The direction in which you walk depends on the body orientation.

Sidestepping is supported using the following keys:

- on keyboard: a key = left, d key = right
- on gamepad: left analog stick left/right
- on Razer Hydra: left analog stick left/right

The direction of stepping depends on the body orientation.

Body rotation uses the same keys as sidestepping:

- on keyboard: a key = left, d key = right
- on gamepad: left analog stick left/right
- on Razer Hydra: left analog stick left/right

The script has the following options:

- Walking: enables forward and backward walking
- Sidestepping: enables left and right sidestepping
- Rotating: enables body rotation
- Rotation speed rate: sets the speed for rotating the body
- Proximity speed: makes the walking speed dependant on the proximity of statis object
- Proximity speed rate: the amount proximity speed influences the walking speed.

  Note that when sidestepping and rotation are selected together, body rotation and sidestepping take place simultaneously.

Without this script physical walking is still possible using the Oculus Rift DK2.

IVR_Body Movements Free

This script translates the movements of the targets to the actual body positions.

Extensions

A number of extensions is included which can be added to the gameObject with InstantVR which add support for one or more input devices or other target controllers.

Extensions can be added to every GameObject with an InstantVR script attached to extend the tracking functionality. Extensions will typically implement support for specific input

devices like the Oculus Rift or Microsoft Kinect, but also include extensions for animations or networking.

The free version supports the Oculus Rift, Traditional and Animator extensions. More extensions are available in the Advanced edition of InstantVR. These include the Razer Hydra, Microsoft Kinect 1 & 2 and Leap Motion.

Multiple extensions can be added to a single InstantVR gameObject. The extensions which will be used at play time depends on:

- availability: is the hardware associated with the extension present?
- priority: the extensions with the highest priority are chosed over lower priority extensions.
- tracking: is the hardware currently tracking?

In the example below, we have added 5 extensions: Oculus Rift, Leap Motion, Kinect 2, Traditional (Xbox 360/mouse/keyboard) and Animator. (note: not all extensions are available in the free edition)

Dynamic behaviour

During gameplay with all supported hardware available the behaviour above is as follows:

- Oculus Rift will always be used for head tracking. It has the highest priority and is always tracking.
- Leap Motion is used for tracking the hands when the hands are in the field of view of the Leap Motion camers. When the hands are outside the view they cannot be tracked by the Leap Motion so it will not be used then. Hand tracking will drop down to the next hand tracker in the hierarchy
- Kinect supports tracking of all 6 tracking targets, but in this case it will not be used for head tracking, because the Oculus Rift has a higher priority. It will be used for hand tracking when the hands are outside the view of the Leap Motion camera. All other body parts are tracked by Kinect all the time.
- Traditional input is can be used for walking around
- The animator is not used, as all targets are tracked by Kinect.

When the game is played with just the Oculus Rift available, it will behave like this:

- Oculus Rift will always be used for head tracking. It has the highest priority and is always tracking.
- Leap Motion and Kinect are not available, so will not be used
- Traditional input is can be used for walking around
- The animator is used to move all targets except the head, which is tracked by the Oculus Rift. This results in leg and arm movements during walking and rotation movements.

Traditional and Animator are typically used when no or limited VR hardware are available and it is good practice to have them as the lowest 2 priority spots as it is shown here.

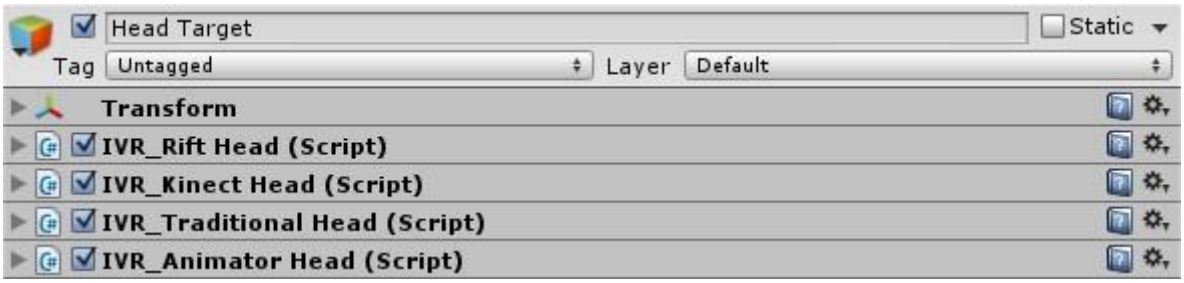
During play mode in the editor it is possible to view which extensions are currently used for each target by switching to debug view:

In this case we have the extensions for Oculus Rift, Razer Hydra, Microsoft Kinect, the traditional and animator extensions configured. In the debug view of InstantVR you can see the various controllers used: Rift for the Head Target, Hydra for the Hand Targets and Kinect for the Hip and Feet Targets.

Controllers

Every extension implements one or more controllers for targets. The Oculus Rift extension has one controller: the Rift Head Controller which is used solely for the head target.

If we look into the Targets themselves we can see the controllers currently associated with that target. Here we have an example of the Head Target (note: not all controllers are available in the free edition)



You can see the Rift and Kinect Head Controllers. Like the priority for extensions the order in which the controllers appear determines the priority. This order is in fact controlled by the ordering of the extensions. So you need to reorder the extension when you want to change the priority of the Head Controllers. Direct changes in the order of controllers will be undone automatically.

Note: the controller priorities and positions are not updated when the inspector is in Debug mode. This is a known issues and is reported to Unity. The priorities and ordering will be corrected when the inspector is switched back to Normal mode.

It is possible to disable specific controllers. For instance: if you do not want to use Kinect Head tracking when the Rift is not available, but you do want to use Kinect for the rest of the body, you can disable the Kinect Head Controller. This is done by disabling the IVR_Kinect Head by unchecking the script in the example above. If you want to disable all controllers of an extension, it is better to remove the extension altogether.

IVR_Animator

The animator implements procedural animations for legs and arms. It is typically used as the last extension in the list of extensions as a fallback when these are not driven by a input device

Hand Targets

Follow head: when enabled, the hands will follow the X/Z position of the hips. Needs to be enabled for physical walking using the Rift and when using the IVR_Walking script

Hip Target

Follow head: when enabled, the hip will follow the X/Z position of the head. Needs to be enabled for physical walking using the Rift.

Rotation method determins how the rotation along the Y axis is determined:

- NoRotation: speaks for itself
- LookRotation: the body rotates to the direction in which the player is looking. Good option when using the Oculus Rift.
- HandOrientation: is not supported in the free version. It will behave the same as LookRotation

IVR_Cardboard

Google Cardboard enables a wide range of mobile phones to be used as a head mounted display. Cardboard is only supported on Android.

Important: the option ‘Virtual Reality Supported’ needs to be unchecked in the Edit menu, Project Settings, Player. This is in contrast to the IVR_Rift setting.

Head Target

Rotational tracking is supported for all mobile phones compatible with Google Cardboard.

IVR_Rift

Adds support for the Oculus Rift DK1 or DK2.

Important: the option ‘Virtual Reality Supported’ needs to be checked in the Edit menu, Project Settings, Player.

Head Target

Full rotational tracking is supported for all versions of the Oculus Rift. Positional tracking is only supported for the Oculus Rift DK2. Note that positional tracking is only possible when the headset is in view of the Oculus IR camera.

Calibration

The head tracking can be calibrated using the calibration keys offered by any other input controller.

IVR_Traditional

Adds support for Xbox controllers or mouse/keyboard.

Calibration

All tracking of other controllers can be recalibrated by pressing ‘back’ and ‘start’ on the Xbox controller simultaneously or by pressing <Tab> on the keyboard.

Left Hand Movements

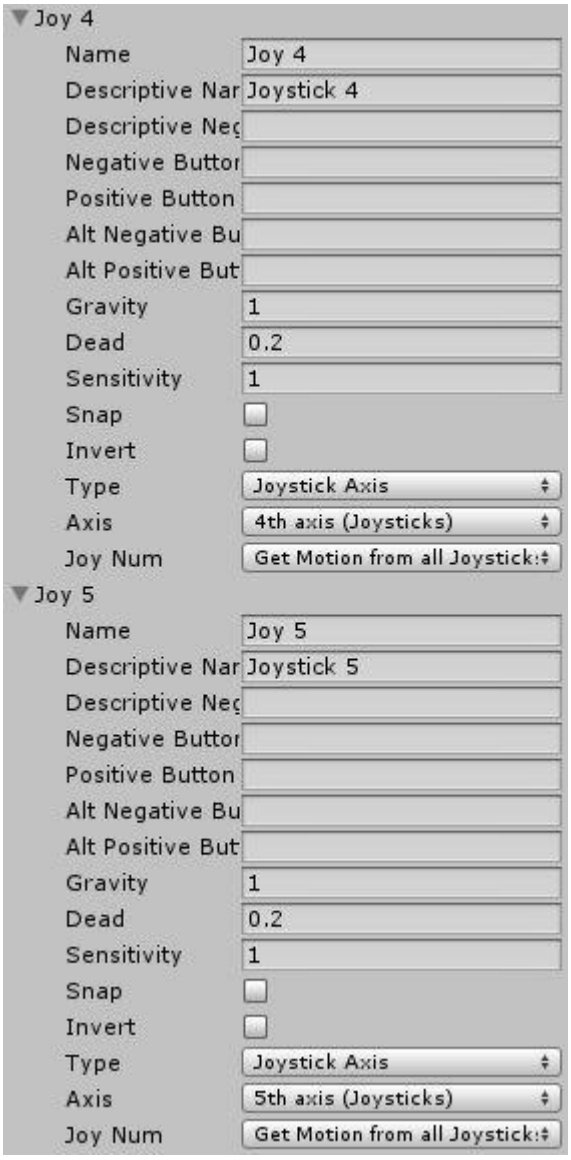
	Xbox controller	Mouse/Keyboard
Thumb	DPad down/right	-
Index finger	left bumper	-
Middle finger	left trigger	-
Ring finger	left trigger	-

Little finger	left trigger	-

Right Hand Movements

Thumb	A/X buttons	-
Index finger	right bumper	left mouse button
Middle finger	right trigger	right mouse button
Ring finger	right trigger	right mouse button
Little finger	right trigger	right mouse button

To support Xbox controllers, you have to add two entries in the InputManager. You can open this from the Edit – Project Settings – Input menu option. The necessary values can be found in the picture below.



Walking

Physical walking is supported in combination with the Oculus Rift DK2.

Known issues and limitations

The following issues are known to the current version of InstantVR:

- In InstantVR v3, the priority of the controllers is not updated from the extensions priority when the inspector is in Debug Mode. This is a Unity issue which has been reported. The ordering will be corrected again when the inspector is switched back to Normal mode.
- When the Razer Hydra is not working and/or when you see the message that the sixense.dll is missing, although it is in the plugins folder, please update to version 1.0.6 of the Sixense Unity Plugin. This is a known issue with version 1.0.4 of the Sixense Unity Plugin.
- When the walking speed is extremely slow after hitting an object: make sure that a layer “MyBody” is available in “Project Settings/Tags and Layers”.
- Leg walking animation does not work when Kinect is selected for the foot targets

Version history

Version 1

- Initial release

Version 1.1

- Oculus Rift SDK 0.4.3 implemented with Unity Free support
- Objects can now be throwed
- Included options to enable/disable controllers
- Look rotation using Rift is now working properly
- Leg orientation improved with high foot positions
- Free and Advanced code bases merged

Version 2

- Oculus Rift SDK 0.4.3.1
- Physical drift correction
- Free walking support
- Side stepping
- Removed Character Motor and Character Controller

Version 2.0.1

- Issue solved: right leg not moving well when avatar was not at origin

Version 2.0.2

- Deleted unnecessary OVR files

Version 2.0.3

- Fix: calibrating with HandRotation and not looking forward teleports the avatar

Version 2.1

- Proximity based walking speed
- Improved leg animation with thumbstick walking
- Improved Rift calibration. No need to face the Rift camera anymore.
- Oculus Rift SDK 0.4.4

Version 3.0

- Unity 5/5.1 support
- Modular architecture
- Body movements in editor
- Realtime switching between input controllers
- Oculus Rift SDK 0.5.0.1

Version 3.1

- Unity 5.1 support
- Google Cardboard support

More information

The lastest and detailed support information can be found at the Passer VR website: http://serrarens.nl/passervr/support/instantvr-support/

Contact

You can contact me via email: support@passervr.com or use the contact form on my website: http://serrarens.nl/passervr/contact/

Thank you for supporting my work!
Pascal.