CYBERITH SDK

Unity Project - Integration Guideline

Cyberith GmbH
Arbeitergasse 25
3130 Herzogenburg
Austria
FN 410899p

For any question, please contact

Pierre Mercy
Software Developer
p.mercy@cyberith.com
+43 660 71 76 828

Table of Contents

Prerequisites

DirectX

https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=35

Unity3D Game Engine

http://unity3d.com/

First Steps

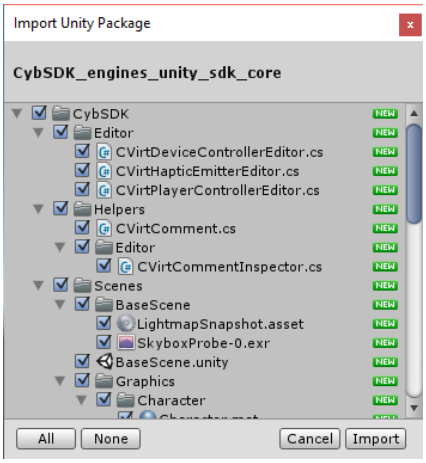
First of all, please note that:

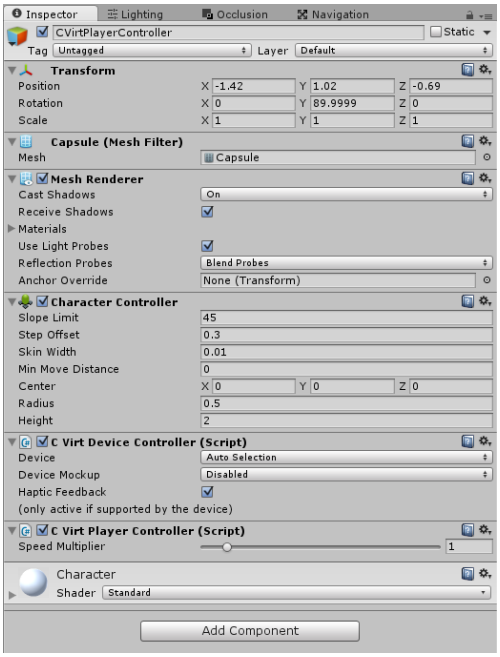
- If you don't have a Virtualizer plugged in, the SDK will detect it and switch automatically to a mouse+keyboard or gamepad controller (Xbox360 gamepad is recommended).

The Cyberith SDK

The Unity package you received works totally independently so you can import it in an empty Unity project and you should be able to try it using the example scene located in "CybSDK/Scenes/ BaseScene".

There is an example Character Controller that you will find in this scene named "CVirtPlayerController".





The HMD

Drag and drop your HMD prefab / Camera directly in the CVirtPlayerController child called "CameraHolder" to have an already Virtualizer ready demo.

If you are able to move your character with WASD keys (movement) and Q/E to rotate… In short, if your demo looks good with these, you are good to go with a Virtualizer!

The Character's body

If you want to add a body to your character, please drop your model inside the "ForwardDirection" child of the CVirtPlayerController. It will be rotated when the player rotates in the device.

SDK Documentation

In the CVirtPlayerController script, you'll find an example on how to implement the SDK and use all the Virtualizer capacities.

Here's the description of them, written in the CVirtDevice script.

GetPlayerHeight

Returns: float.
Gives back the current height of the ring construction in centimeters.
0 is default height (upon reset), <0 is when you're getting lower (crouching) and >0 is upper (on your toes, jumping).

ResetPlayerHeight

Returns: void.
Resets the default height of the ring construction.
Basically, you will call this function when pressing a "reset" button for example.

GetPlayerOrientation

Returns: Vector3.
Gives back the orientation as being the forward vector of the body.

Exemple of integration

```
Quaternion rotation = new Quaternion();
rotation.SetLookRotation(virtOrientation, Vector3.up);
transform.localRotation = rotation;
```

GetMovementSpeed

Returns: float.
Gives back the speed of the player's movement in meter per second.

GetMovementDirection

Returns: Vector3.
Gives back the Movement Direction as a normalized vector of structure: (right, 0, forward).

ResetPlayerOrientation

Returns: void.
Currently disabled.