UE4 RealSense Plugin Getting Started Guide

This guide presumes that you have already installed Visual Studio, Unreal Engine 4.19 , the RealSense SDK (R4 or later) and the RealSense DCM(s) for whichever camera(s) you wish to use. If you do not, please install these programs before proceeding:

<https://software.intel.com/en-us/intel-realsense-sdk/download>

You need download and install SDK R2 2016 <http://registrationcenter-download.intel.com/akdlm/irc_nas/9078/intel_rs_sdk_offline_package_10.0.26.0396.exe>

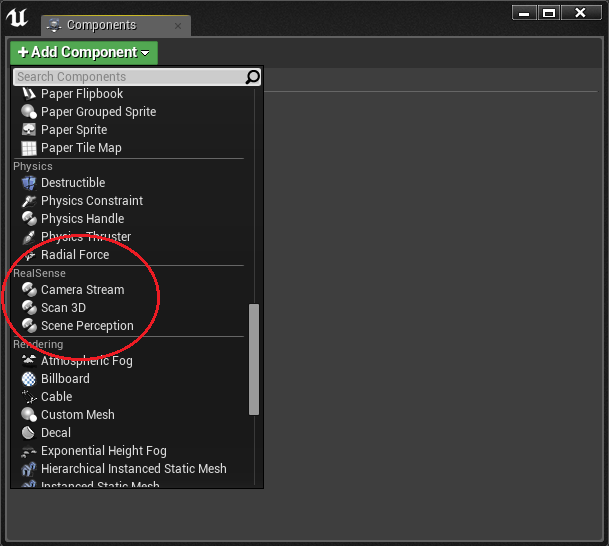
And you need install drivers for camera SR300 <https://downloadmirror.intel.com/25044/eng/intel_rs_dcm_sr300_3.3.27.5718.exe>

REAL SENSE SDK documentation <https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/index.html?doc_hand_hand_module.html>

General Using the plugin

Here are some common steps you will need to repeat to add some basic RealSense functionality.

1. With your project that you created in the last step open in the UE4 Editor, create a new Blueprint Actor.
2. Open that actor in the editor, and check out the Components windows. If this window is not visible, go to the “Window” tab on the file menu and make sure “Components” is selected.
3. Click the “Add Component” button and scroll down to find the category of components labeled “RealSense”.



**Shared properties of all RealSense Components**

Each RealSense component shares some basic functionality. They all contain the following camera properties:

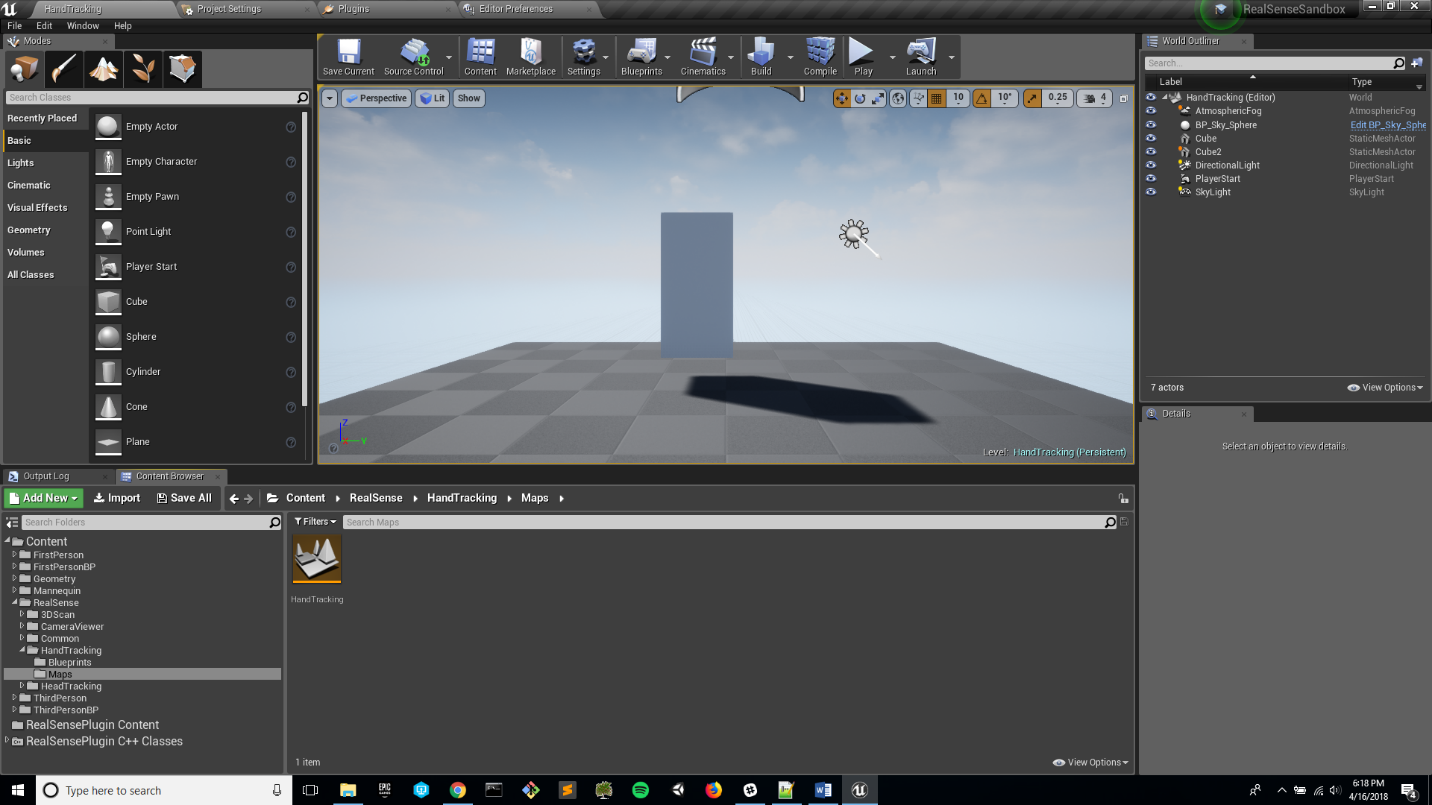
* Field of view of the color camera
* Field of view of the depth camera
* Model of the currently connected RealSense camera (F200, R200, SR300)
* Firmware version of the currently connected RealSense camera

And the following functions:

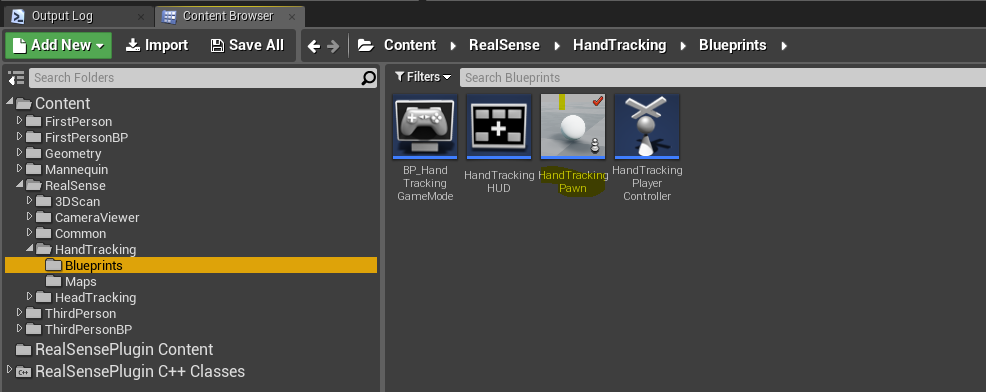
* StartCamera()
  + Turns on the RealSense camera
* StopCamera()
  + Turns off the RealSense camera
* IsCameraRunning()
  + Returns true if the camera is currently running (between calls to StartCamera and StopCamera)
* GetColorCameraResolution() / SetColorCameraResolution()
* GetDepthCameraResolution() / SetDepthCameraResolution()
* IsStreamSetValid()
  + Takes as input a Color and Depth Camera Resolution and returns true if the combination is a valid.
  + The RealSense SDK documentation has more information on valid sets of resolutions.

Hand Tracking Example

1. You can open project and it is automatically open HandTracking sandbox map



1. You can investigate Blueprint HandTrackingPawn. Where you can get how to receive hand tracking data



1. You can Investigate Blueprint HandTrackingHUD. Where you can understand how to het Depth and RGB frame from the camera
2. If you run game you should get result like this 
3. WARNING. There is some Issue in C++ with sharing data from the camera thread and game thread, sometimes it could crash the game. If game will stack you can unplug camera from usb for 5-10 sec and then plug it back. It should help