Software and Electronic Engineering VR headset + GAME Documentation

Project log

How does VR work?

VR glasses contain 2 lenses that help to create a 3D virtual image by angling 2 slightly different 2d images. light passes through the cornea, iris and lens up to finally at the retina. from the retina all the information travels to the brain and is processed

Tuesday, 8 October 2019:

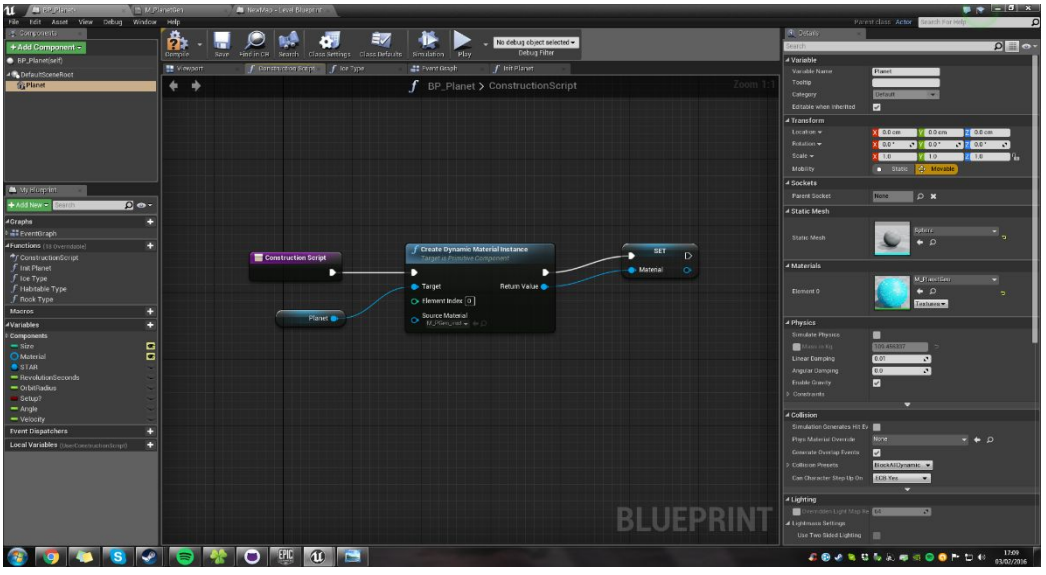
Submitted project proposal

Friday, 11 October 2019:

Ordered STM32 and Mpu6050 gyroscope and got 3d printed parts ordered and controller

Saturday 12 October 2019 Unreal engine:

Installed unreal engine and started using blueprint and C++ to start my game.



Monday 15 October 2019:

STM32 and gyroscope arrived

Started Implementing connection between gyroscope and Due to get reading from gyroscope for later use.

Wednesday 17 October 2019: STM32 hardware implementation

Started stm32 implementation with gyroscope

Friday 19 October 2019: Quaternions

Learning about Quaternions in video games

a quaternion is a complex number with 4 dimensions. But in game development, Quaternions are often used to describe a rotation in 3d space by encoding:

1. a rotation axis (in form of a 3-dimensional vector)
2. how far to turn around that axis

An alternative way to describe rotations is by describing how far to turn around the 3 fixed axis' x, y, and z (aka Euler angles) which only requires 3 numbers instead of 4 and is usually more intuitive to use. However, Euler-angles are subject to a problem called gimbal-lock: When you rotate 90° around one axis, the other two axis become equivalent. With quaternions, this problem does not occur

20 October 2019: Quaternions in coding

1ST NOVEMBER 2019 OSVR Open-Source Virtual Reality

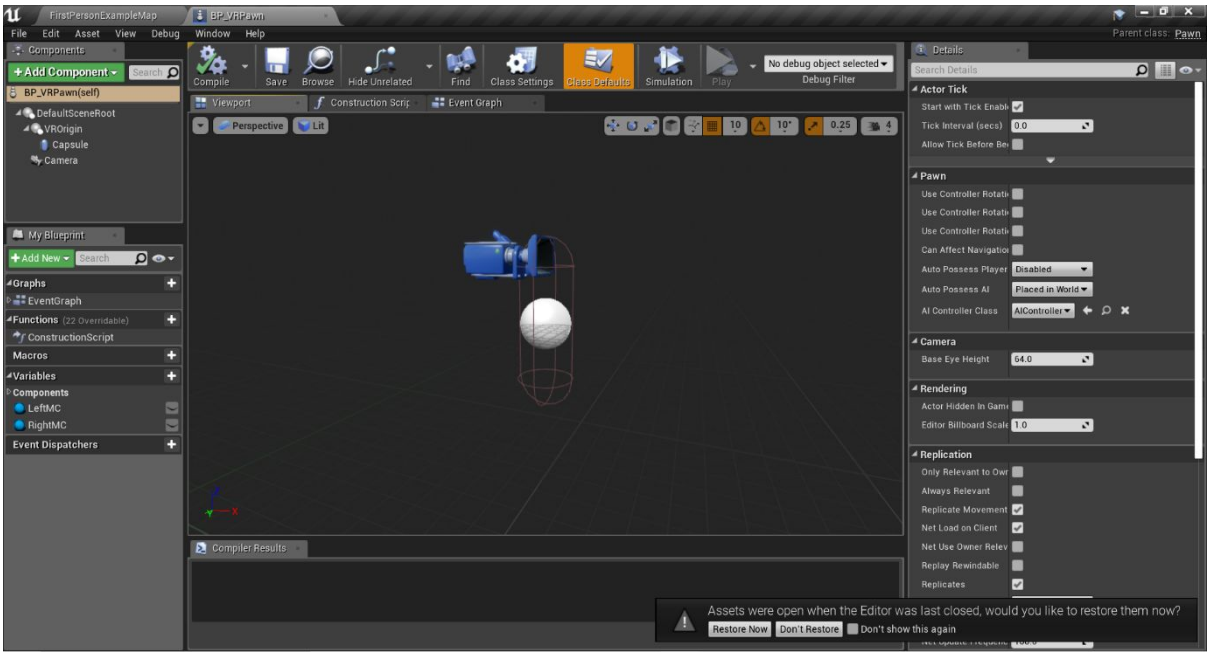
What is OSVR

Open Source Virtual Reality is an open-source software project that aims to enable headsets and game controllers from all vendors to be used with any games developed by Razer and Senses. It is also a virtual reality headset that claims to be open-source hardware using the OSVR software

OSVR is designed to work with several other head-mounted displays and is on a mission to establish an open standard so that existing devices and software can become interoperable.

Developers and consumers with OSVR-compatible HMDs, can already get started by configuring their equipment. The OSVR website features a perfect showcase of their software (also called “experiences”) that are compatible and ready to run. There are already games, media players, emulators and more! There is lots of SteamVR-powered content too.

1ST NOVEMBER 2019 continuation of Unreal engine



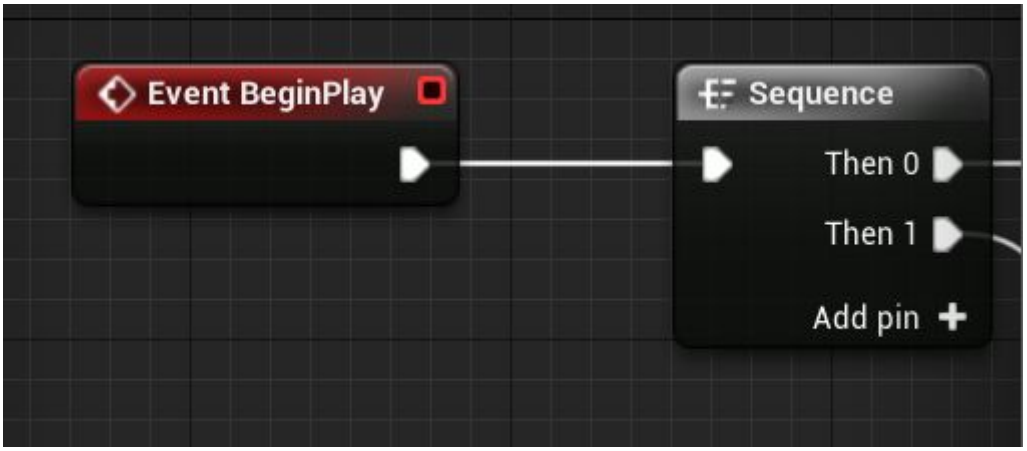
This the camera and how tall the player will be and how they will camera will be used and placed

So far, I've created two hands, a canvas and a pickup test. The goal is to test if I can pick up and drop-down items.

2nd November 2019

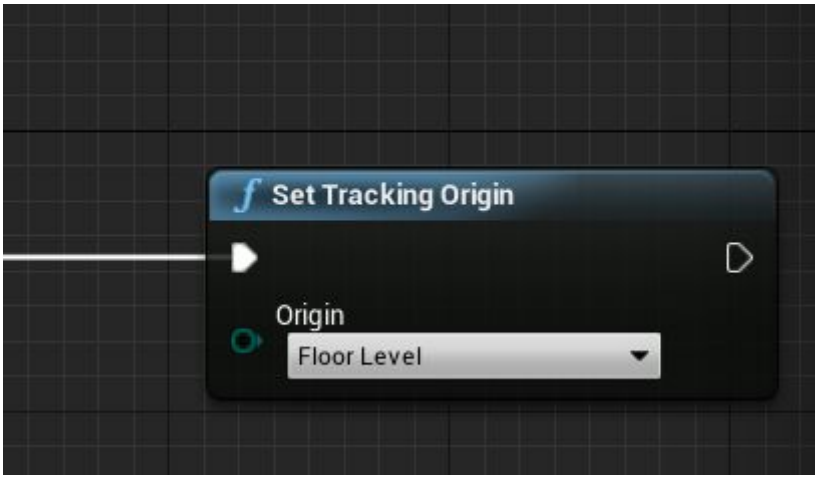
Unreal documentation for BP_VPAWN

Reacting to the environment



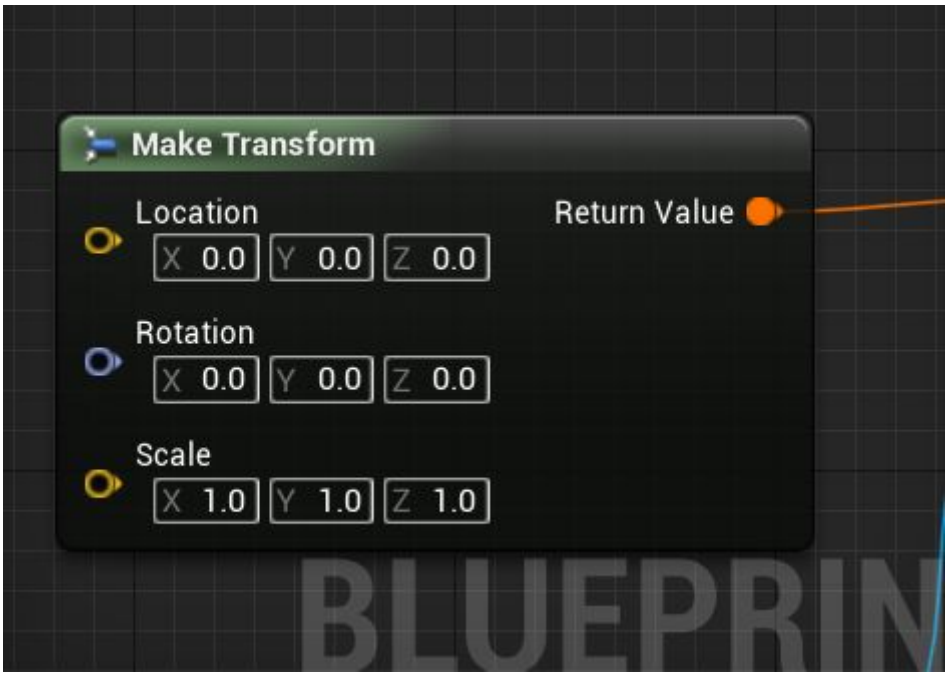
Event Begin Play
This **event** is triggered for all Actors when the game is started, any Actors spawned after the game is started will have this called immediately. Upon **beginning play**, this Actor will set its Health to 1000 and Score to 0.

The **Level Sequence** is the "container" for your cinematic scenes and must be created in order to begin working inside of the Sequencer Editor.

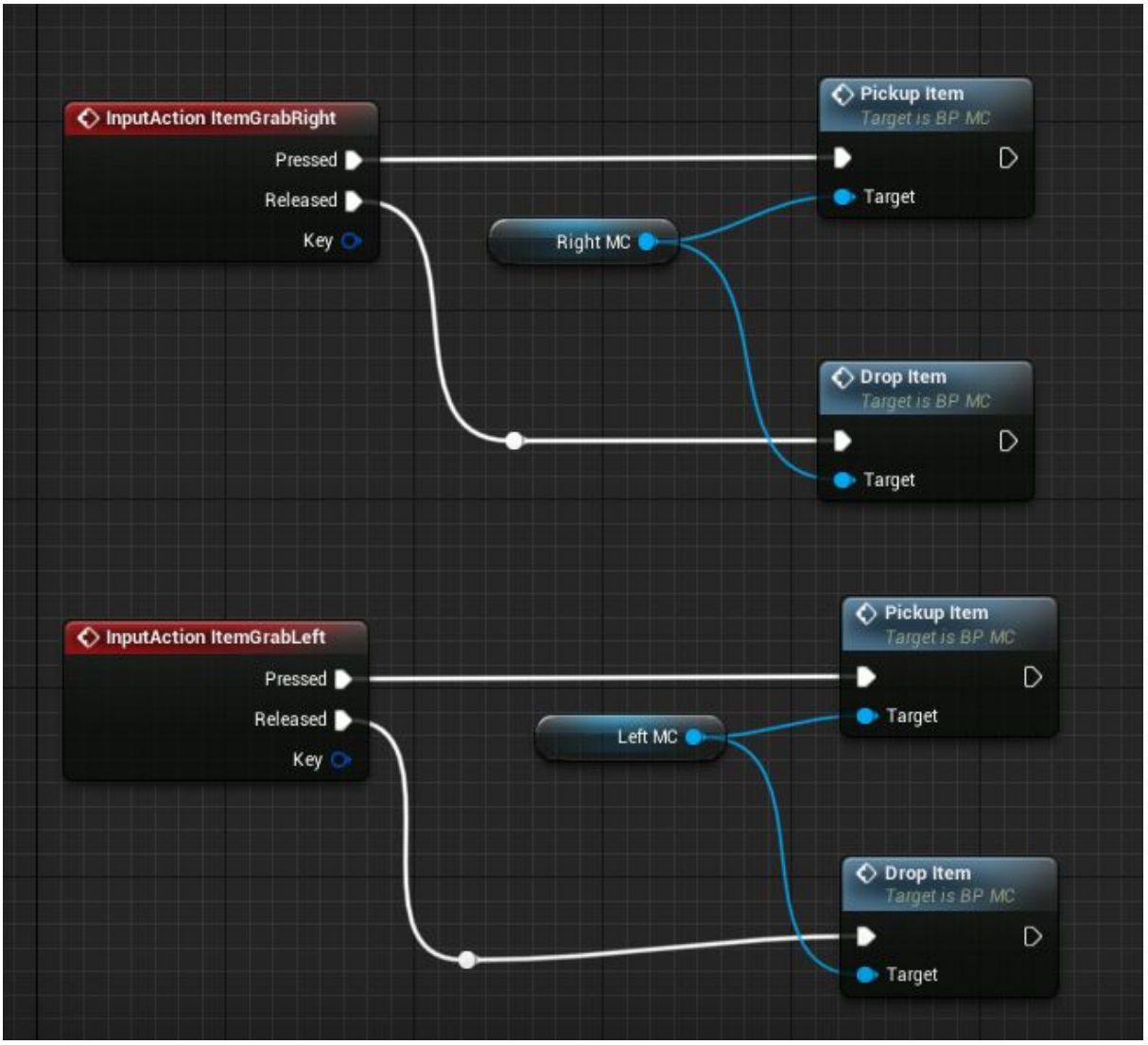


Sets current tracking origin type (eye level or floor level).

Target is Head Mounted Display Function Library



make transform is just location rotation and scale combined as quaternion



Action and Axis Mappings provide a mechanism to conveniently map keys and axes to input behaviours by inserting a layer of indirection between the input behaviour and the keys that invoke it. Action Mappings are for key presses and releases, while Axis Mappings allow for inputs that have a continuous range.