

XR Rig Setup

Getting Started

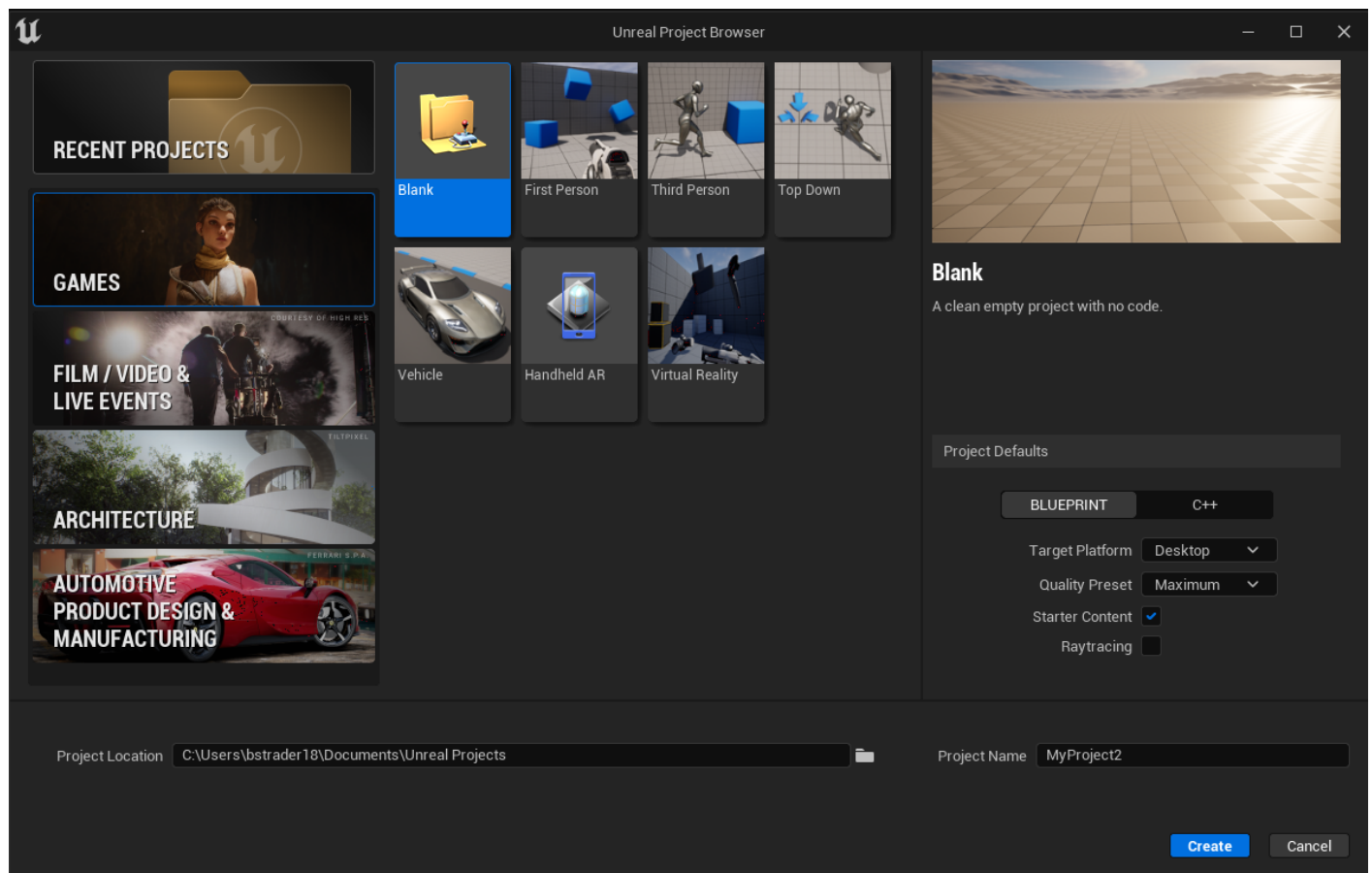
Start by launching Unreal Engine.

Select the **games** option on the left.

Select the **blank** template. Project Defaults should be set to **blueprint**. Raytracing should be **disabled**.

Consider your target device for Target Platform

Quality Preset for standalone requires **Scalable**. Desktop can be either **Maximum** or **Scalable**, but often Scalable is the best choice.



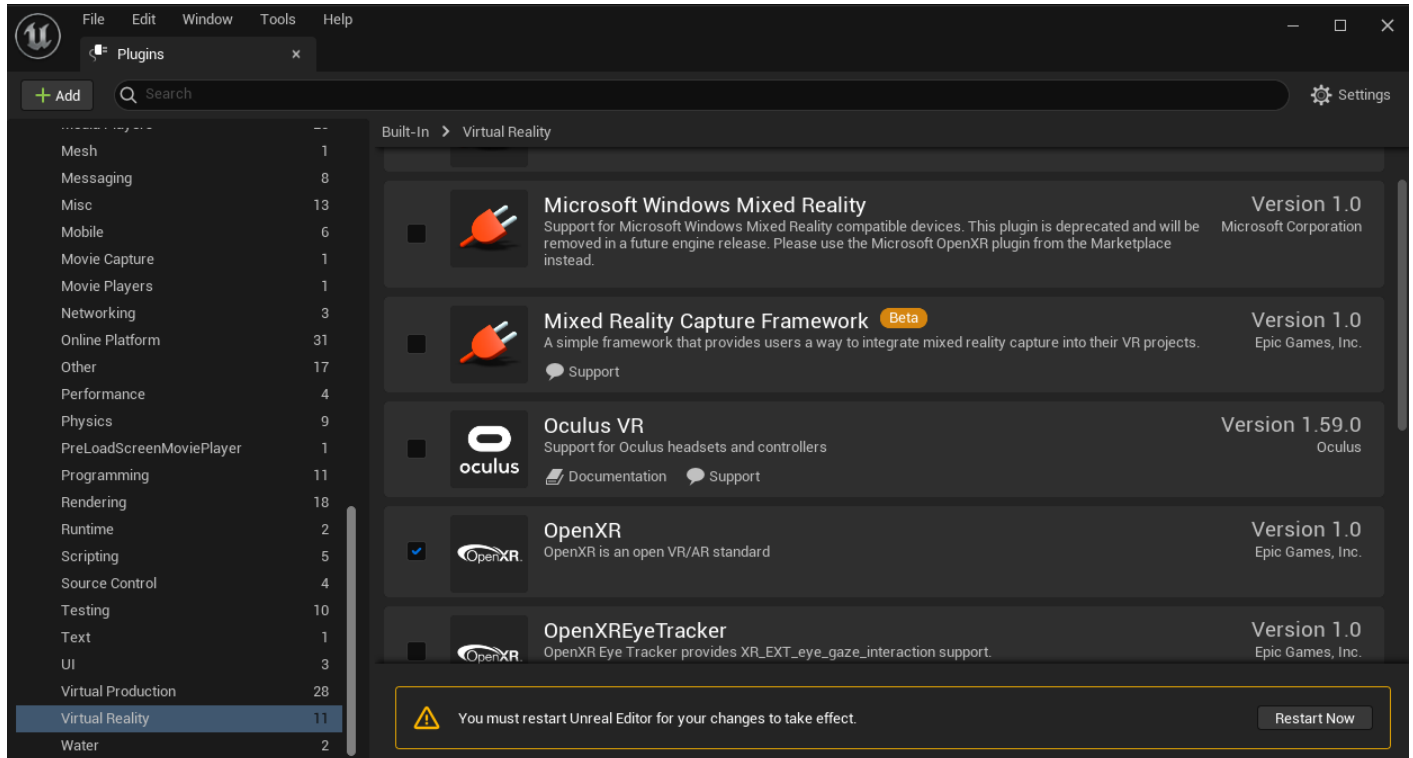
Select project location, name your project, and click **Create**. After some time the new project appears.

Select **plug-ins** in the Edit menu.

Select **Virtual Reality** in the left menu, under built-in.

Check OpenXR

A box outlined in yellow should appear at the bottom. Click **Restart Now**.



Once restarted, go to File > New Level

Select **Basic Level**

If this is your first time in Unreal Engine, it is generally *highly* recommended to **NOT** use **Open World** as a new level!

Go to **File > Save Current Level** and give your level a *meaningful* name.

Now is a good moment to **APPLY VR SETTINGS**

Create the XR Rig

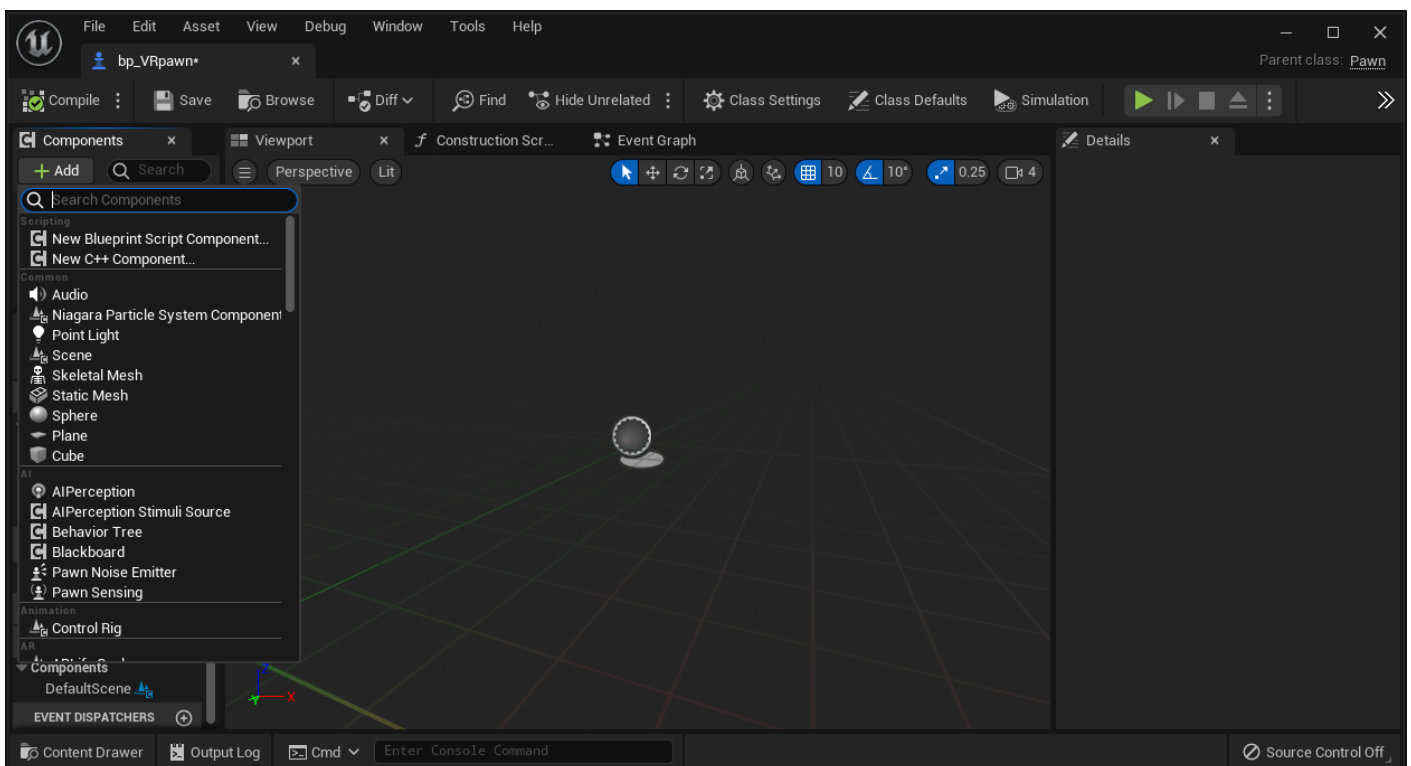
Right click in the **Content Drawer** (at the bottom of the screen) and choose **Blueprint Class**.

When prompted, choose **Pawn**. Give it a meaningful name, such as **VRPawn_BP**.

Double click the new pawn to open and edit it.

A new window should appear- this is the **Pawn Object Editor**.

In the upper left corner of the Pawn Object Editor under Components, click the Add button (marked by a green +).



Choose **scene** and rename it to **CameraOffset**.

With **CameraOffset** selected, press Add again and add a **camera**. Leave the name as "camera".

Select **CameraOffset** once again, press Add, and add a **motion controller**. Name it "Motion_Controller_L".

Repeat the previous action, this time naming it "Motion_Controller_R".

Select "Motion_Controller_L". With this component selected, add a component by either clicking the **+Add** button or right-clicking and add the component **XR Device Visualization**. This should now be a child of Motion_Controller_L. Rename the component to something like "XRDeviceVisualization_L". In the **Details** panel, under **Collision**, Change **Can Character Step**

Up On to **No**, and change the **Collision Presets** from **BlockAllDynamic** to **NoCollision**. Finally, move to the **Visualization** section of the *Details* panel and check **Is Visualization Active**.

Repeat with "Motion_Controller_R".

Make sure the **Motion Source** is set to **Right** under the **Motion Controller** section (directly under Visualization) for "Motion_Controller_R". Do not select Display Device Model (it is deprecated and replaced with a component as mentioned above).

Finally, switch to the **Event Graph** tab and find the node listed **Event BeginPlay**. Right click and add a **Set Tracking Origin** node and under origin verify "Floor Level" is selected. From the **BeginPlay** node, drag a white connector pin to the newly added node.

Press **compile** at the top left of the screen.

Press **save** and return to the level view.

Add your pawn to the level

Add your newly created player pawn to the scene. Select your pawn and in the details panel, under Pawn, change **Auto Possess Player** from **Disabled** to **Player 0**. You can also change these values within the pawn blueprint itself, especially if you are using it in multiple maps.

In the same panel as the play button, click the three dots and select **VR Preview** (if you don't see VR Preview, leave Unreal, run the Meta Link software,, confirm VR is working through the headset, and restart the Unreal editor).

Test the scene in your headset. (Adding an object and/or lighting point may be helpful for testing).

Where to go from here?

The wiki is full of guides, but it is important to learn basic teleport as it covers additional basics for both Unreal and VR.