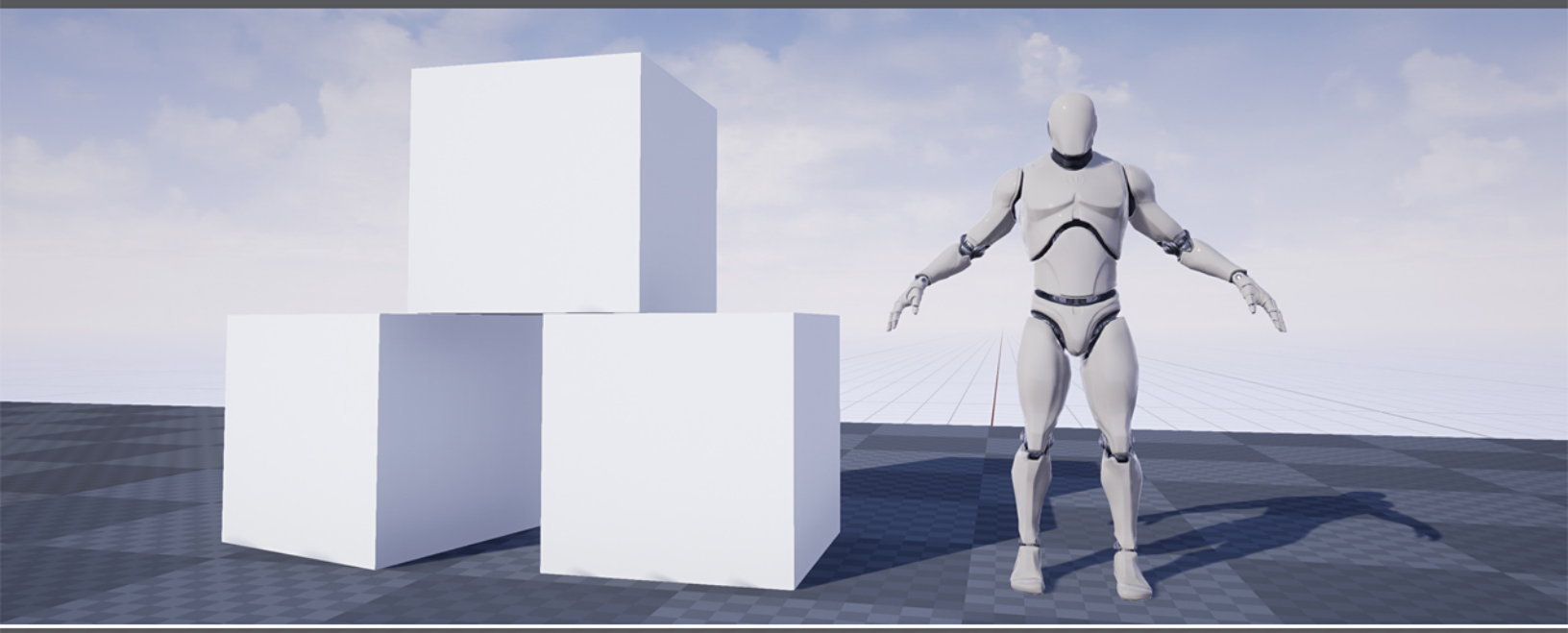
CONTAINS 150+ PAGES OF TECHNIQUES, TIPS, INSIGHT TO GETTING STARTED WITH UNREAL® ENGINE 4



UE4 BEGINNER'S

HOW TO START, LEARN & USE UNREAL® ENGINE 4

QUICK START GUIDE

WRITTEN BY: ALEX GALUZIN

V. 2

WORLD OF LEVEL DESIGN.COM

UE4 BEGINNER'S QUICK START GUIDE

HOW TO START, LEARN & USE UNREAL® ENGINE 4

WRITTEN BY: ALEX GALUZIN

V. 2

WORLDOFLEVELDESIGN.COM

“UE4 Beginner’s Quick Start Guide: How to Start, Learn and Use Unreal® Engine 4”

V1 Published: September 2015 (original name: “UE4 Beginner’s Crash Course: How to Start Learning and Using Unreal® Engine 4”)
V2 Published: July 2019

Version: 2.0
UE4 Version Used: 4.21
Cover Image: Starter/Example Maps in Unreal® Engine 4
Ebook Created and Published by World of Level Design LLC
www.WorldofLevelDesign.com

You may contact World of Level Design at: contact@worldofleveldesign.com

TABLE OF CONTENT

- Step 5: BP_Sky_Sphere
- Step 6: Directional Light and BP_Sky_Sphere
- Step 7: Sky Light
- Step 8: Inserting Atmosphere Fog
- Step 9: Inserting a Player Start
- Step 10: Inserting Reflection Capture Actor
- Step 11: Auto-Exposure/Eye Adaptation
- Step 12: Lightmass Importance Volume
- Step 13: Saving Your Level
- Step 14: Building Lights and Build All
- Step 15: Testing Your Level

Section 4: Transition from UDK to UE4

- Free Unreal Engine 4
- Unreal Engine/Epic Games Launcher
- Project Management
- Game Templates
- New Level Templates
- Keyboard Shortcuts
- Tabs And Fluid Interface
- Viewport Navigation In UE4
- Modes Panel
- Class Viewer
- Developer Tools
- Content Browser
- Working With Objects
- Details Panel
- Show All Advanced Details Property
- Scale
- Grid Setting Size
- World Outliner
- World Settings
- BSP Brushes
- PBR Material Creation
- Lighting
- Blueprint Replaces Kismet
- Importing And Storing Custom Assets
- Project Settings
- Current Project In UDK To UE4

AlexG here,

Thanks for visiting World of Level Design and signing up to download this "UE4 Beginner's Quick Start Guide".

If you happen to get this PDF from somewhere other than an email from WorldofLevelDesign.com then quickly head over to this page and **sign up for a FREE "WoLD Insider Emails"**. You'll join thousands of others who are on this **EXCLUSIVE and FREE Insider email list**.

I send out a lot of cool secret stuff, tutorials, guides and updates that aren't available anywhere else unless you are subscribed. I don't want you to miss out.

I love level design and game environment art. Pretty much all of my time consists of learning, testing, experimenting and creating inside various game engines. I take what I learn and create tutorials, videos, books and guides.

I do all the work, so you don't have to. This way you focus on creating!

So let's get to it.

Unreal® Engine 4 is a complete game engine. It is an extremely deep and complex piece of software that can be used to create variety of games, environments, cinematics and architecture/product visualizations.

Learning UE4 as a complete beginner is intimidating and requires a lot of your time.

There are thousands of tutorials, videos, documentation and advice already out there. But, you get pulled into too many different directions as you end up more confused and overwhelmed.

It doesn't have to be that way.

To learn UE4 from scratch comes down to two things:

1. You must know what you should focus on as a complete beginner
2. You must know what you should avoid until later.

Of course as a beginner you don't know what that is yet. But this "UE4 Beginner's Quick Start Guide" does.

So, I am going to teach you how to get started with UE4 today!

Make yourself many cups of coffee or tea and let's go!

FOLLOW WOLD FOR HIDDEN KNOWLEDGE

Below, I have listed how to keep in touch with WoLD to get more tutorials, guides, videos, tips and insight so you can become THE BEST level designer and environment artist.

WoLD Website (MAIN SOURCE):
https://www.worldofleveldesign.com/

WoLD Insider Newsletter and Updates:
If you haven't signed up for a free "WoLD Insider Newsletter and Updates" visit here and get a couple of free guides for signing up:
https://www.worldofleveldesign.com/wold-insider/

WoLD Facebook:
www.facebook.com/worldofleveldesign

WoLD Twitter Channel:
www.twitter.com/GameLevelDesign

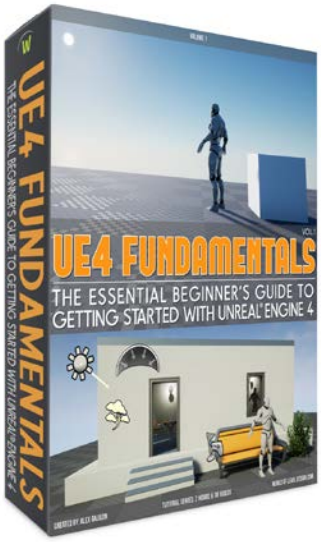
WoLD YouTube Channel:
www.youtube.com/WorldofLevelDesign

WoLD Instagram:
www.instagram.com/worldofleveldesign/

PREMIUM TUTORIAL GUIDES FOR UE4

This “UE4 Beginner’s Quick Start Guide” will get you started with Unreal Engine 4. But you’ll need more. I’ve created complete tutorial series below for deep dive into UE4 game engine and how to use it.

“UE4 FUNDAMENTALS VOL.1”



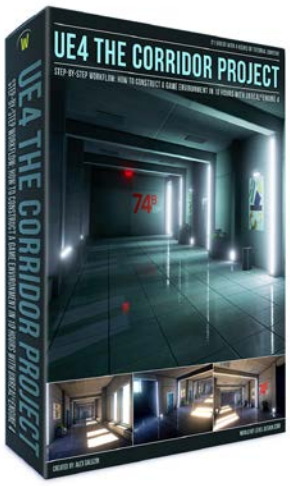
The essential complete beginner’s guide to learning and using Unreal® Engine 4 without any prior knowledge or experience.

Tutorial Series Includes:

- All New! Updated and revised
- 38 videos (7 hours)
- Instant Digital Download

Get the UE4 Fundamentals Vol. 1...

“UE4 THE CORRIDOR PROJECT”



UE4 The Corridor Project is an intermediate tutorial guide focused on constructing a game environment with provided custom Static Meshes. It’s in-depth guide for putting together an environment from start-to-finish.

Tutorial Series Includes:

- 21 videos (4 hours)
- Instant Digital Download

Get the UE4 Corridor Project...

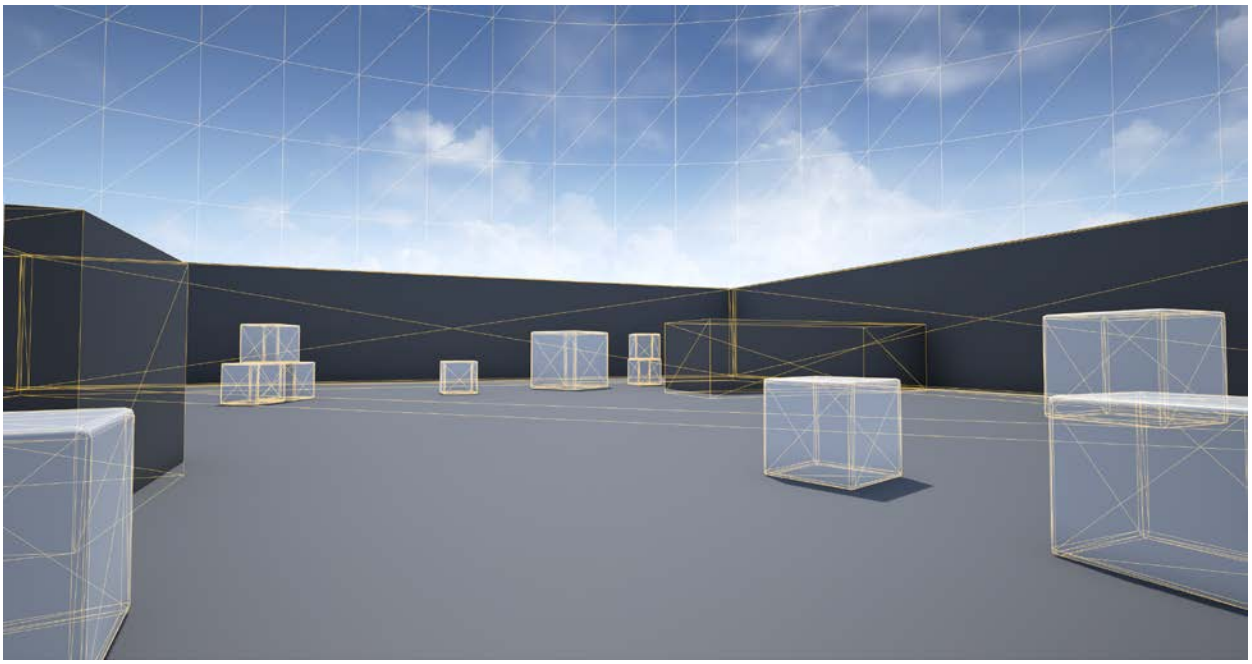
Premium Tutorials come with WoLD 30-Day Money Back Guarantee!

UE4 VERSION

The following “UE4 Beginner’s Quick Start Guide” uses Unreal Engine 4 version 4.21.

Most of the techniques and principles described in this guide will work in older or newer versions of the engine.

SECTION 1:
21 LESSONS TO QUICK START USING UE4



Unreal Engine 4 is a complete game engine; it's very expansive and it will take you years to master it.

#1 question as you learn UE4 is always been: **"Where do I even begin?"**

As a complete beginner you need to know exactly what you should focus on and what you should avoid.

The following section will get you started to use UE4 as a complete beginner without any prior knowledge or experience by the end of TODAY!

If you are interested in a methodical step-by-step home study course to learning Unreal Engine 4, you need "UE4 Fundamentals".

Let's begin!

1. DOWNLOAD AND INSTALL UE4 IN 3 EASY STEPS

UE4 is FREE to download and use. So let's download and install Unreal Engine 4.

Make sure that your computer hardware is up to specifications.

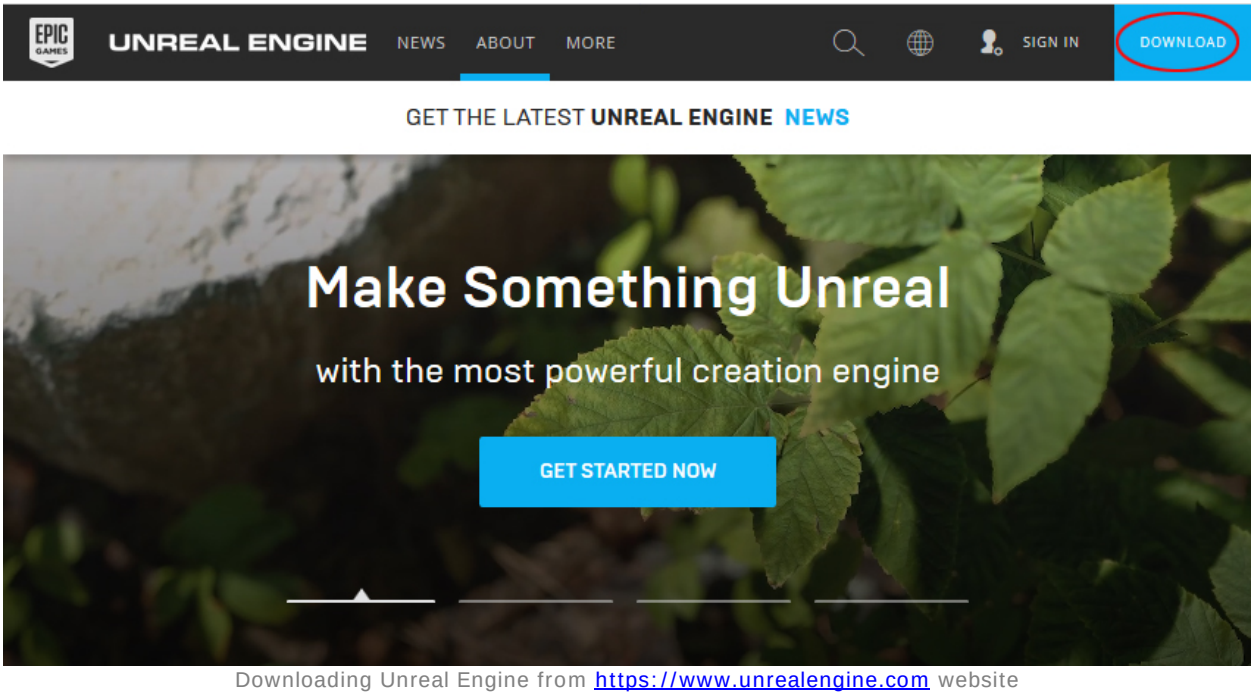
Recommended Specs:

- Operating System: Windows 10 64-bit (Windows 7 minimum)
- Processor: Quad-core Intel or AMD, 2.5 GHz or faster
- Memory: 8 GB RAM
- Video Card/DirectX Version: DirectX 11 or DirectX 12 compatible graphics card

Source: docs.unrealengine.com/en-us/GettingStarted/RecommendedSpecifications

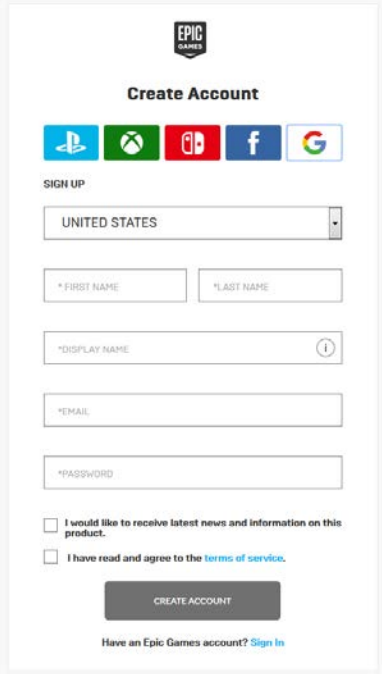
Step 1: Create Free Account

Go to https://www.unrealengine.com/ and click on **Download** icon on the top right of the website:



Downloading Unreal Engine from https://www.unrealengine.com website

You will need to create a log-in and register to access Unreal Engine 4:

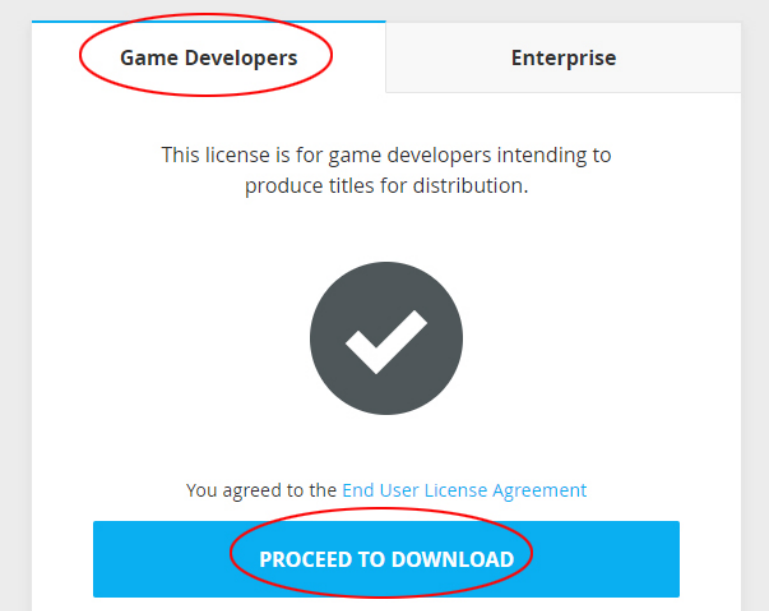


After you created the free account choose which UE4 version you want to download:

- Game Developers or Enterprise

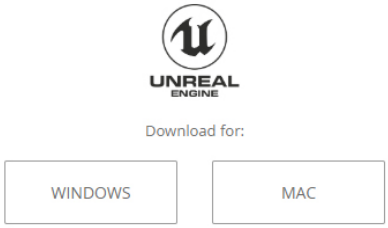
Game Developers will give you the UE4 you are looking for. Enterprise is something a bit different which gives you access to Unreal Engine Studio.

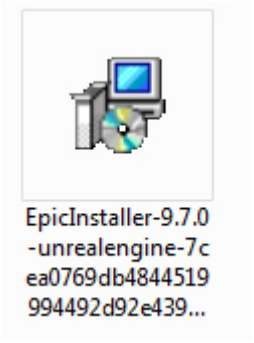
Choose **Game Developers** and Proceed to Download:



Step 2: Download and Install Epic Games Launcher

Choose Windows or Mac version of the installer:



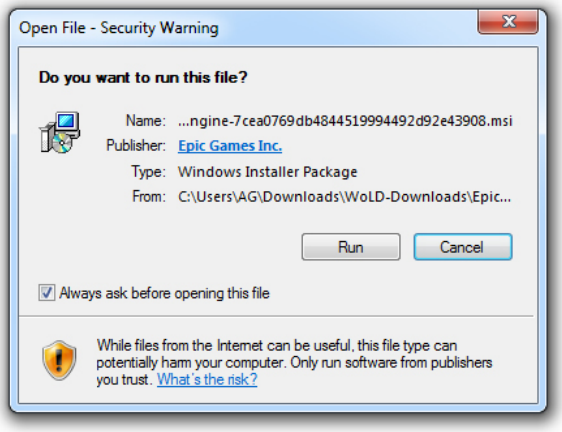


You will now download EpicInstaller. This installer is not UE4 yet.

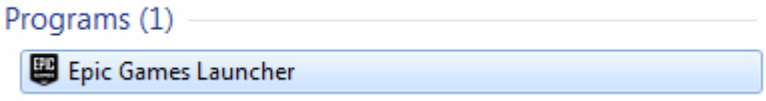
Epic Games Launcher is a portal through which you download and launch any Unreal Engine version you want.

Through this Launcher you also create and manage UE4 projects and download Marketplace Content.

Run to install **Epic Games Launcher**:

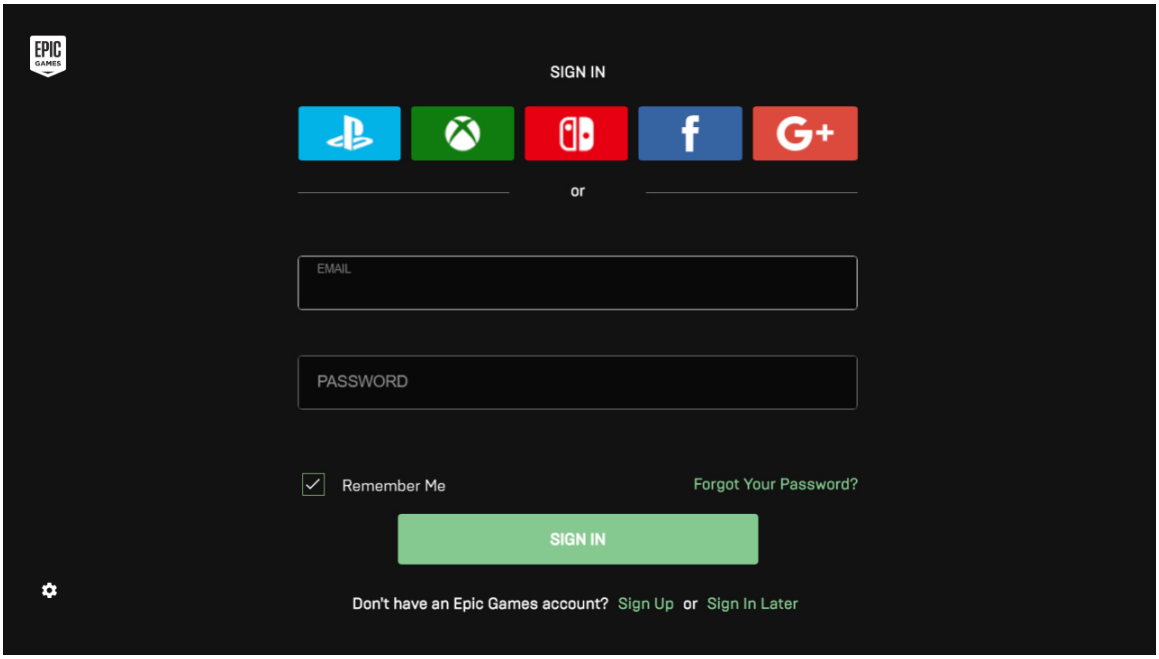


After installation, you should have Epic Games Launcher shortcut on your desktop. You can also access Epic Games Launcher through the Start icon on Windows.

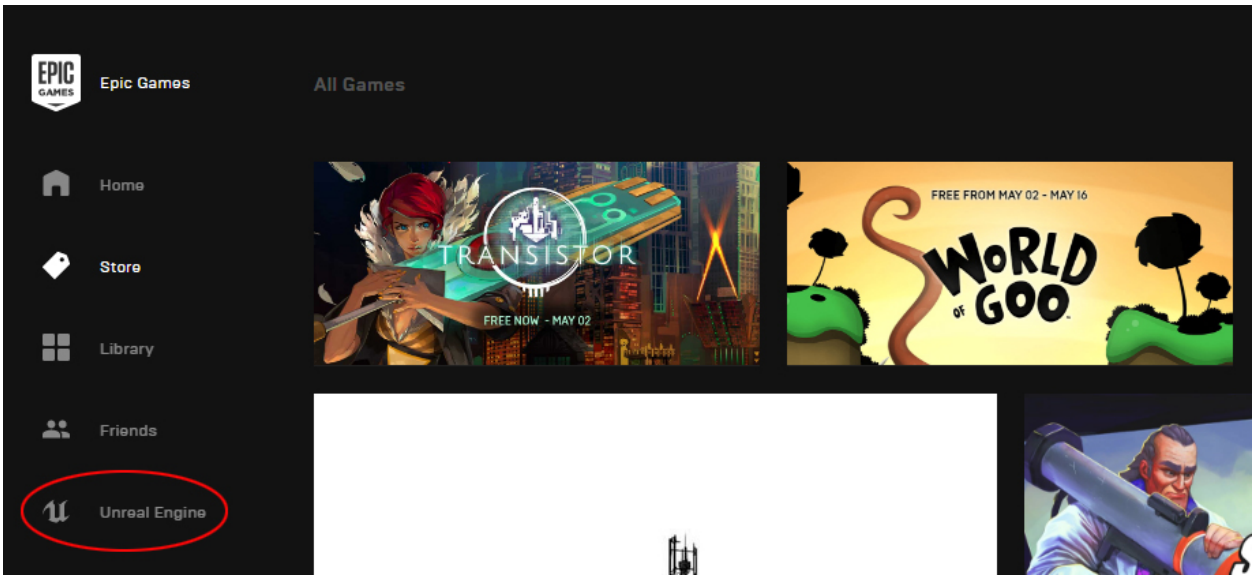


Step 3: Download and Install Unreal Engine 4

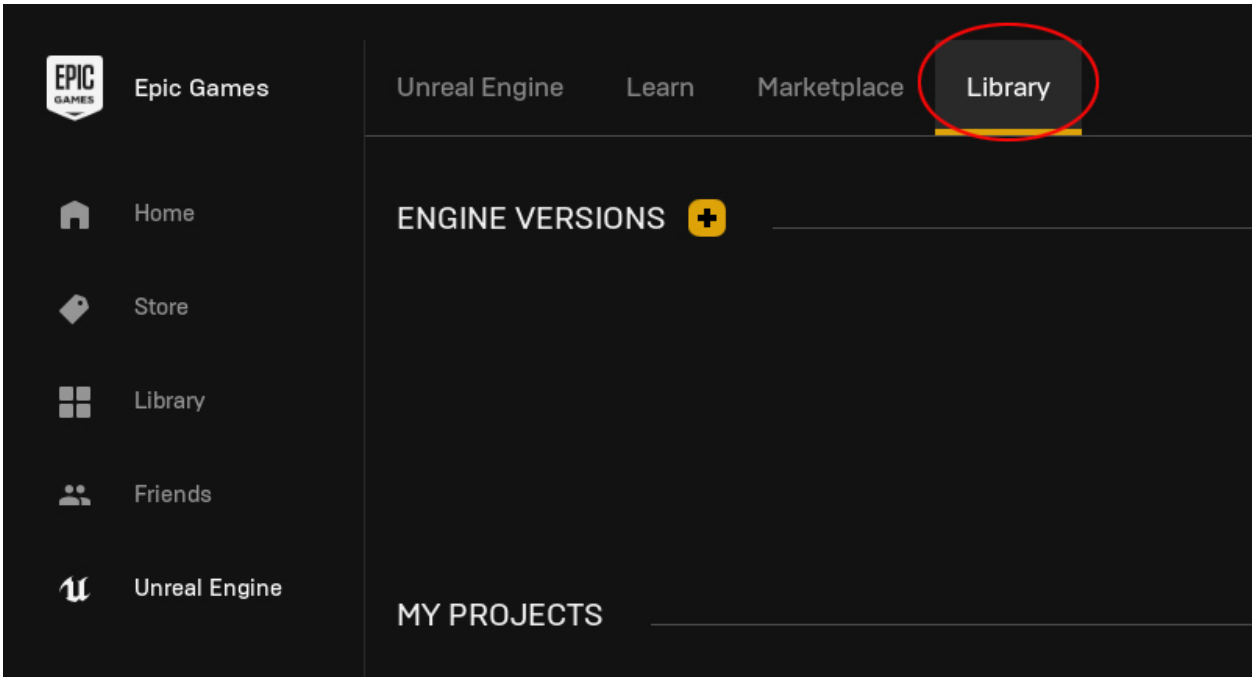
Open and log in to the Epic Games Launcher using the same credentials you used to register with.



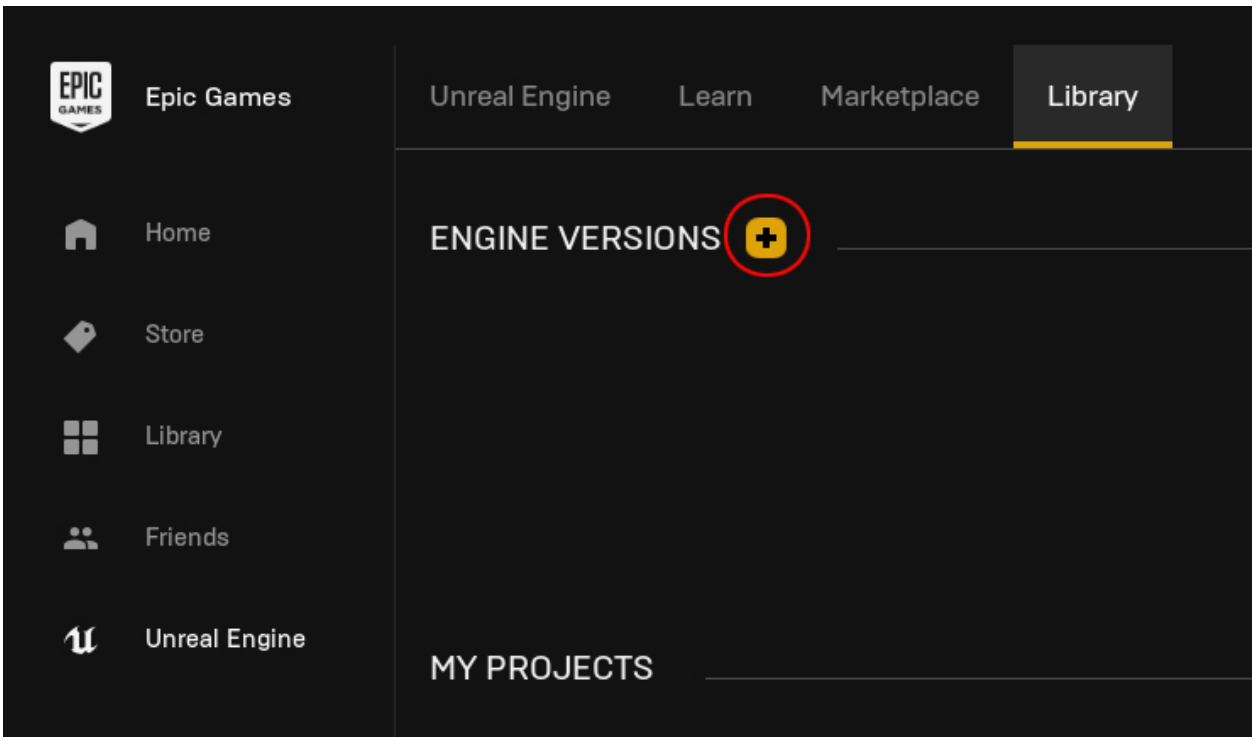
Click over to **Unreal Engine** tab on left hand side:



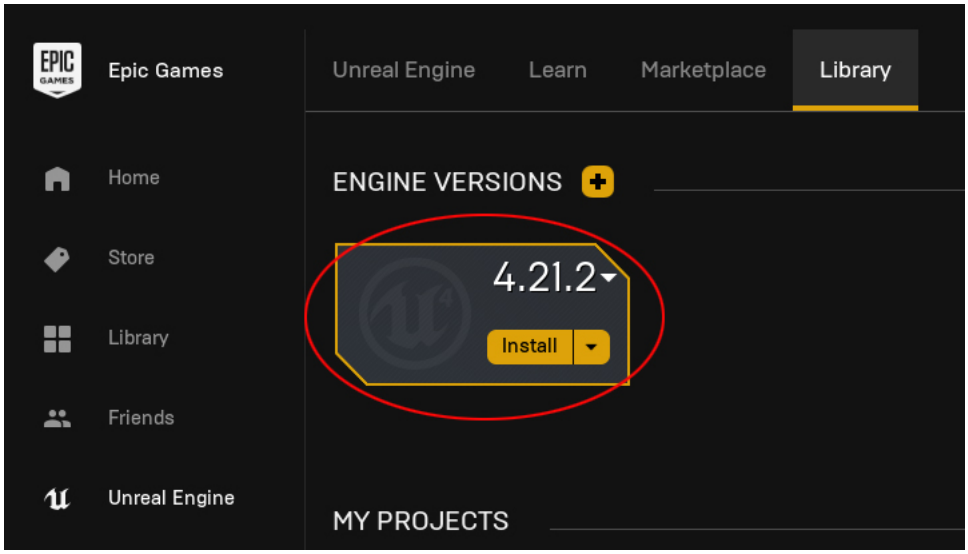
Then click over to **Library** tab:



Under Engine Version click on the plus icon:



Using the drop down menu, choose which Unreal Engine version you want to download and install. Choose any version you want. Most likely it will be the latest available version:



Downloading and installation will take a bit of time, but once it is done, you now will have Unreal Engine 4 on your computer - ready to use.

PS. For selling your game and royalties details visit here.

2. HOW TO CREATE YOUR FIRST PROJECT AND LAUNCH THE EDITOR

Everything you work on in UE4 is contained within a project.

Projects have their own folder where all of that project's content is contained. This content will often include Static Meshes, textures, materials, animations, particle effects, created levels and so on that are relevant to one specific project.

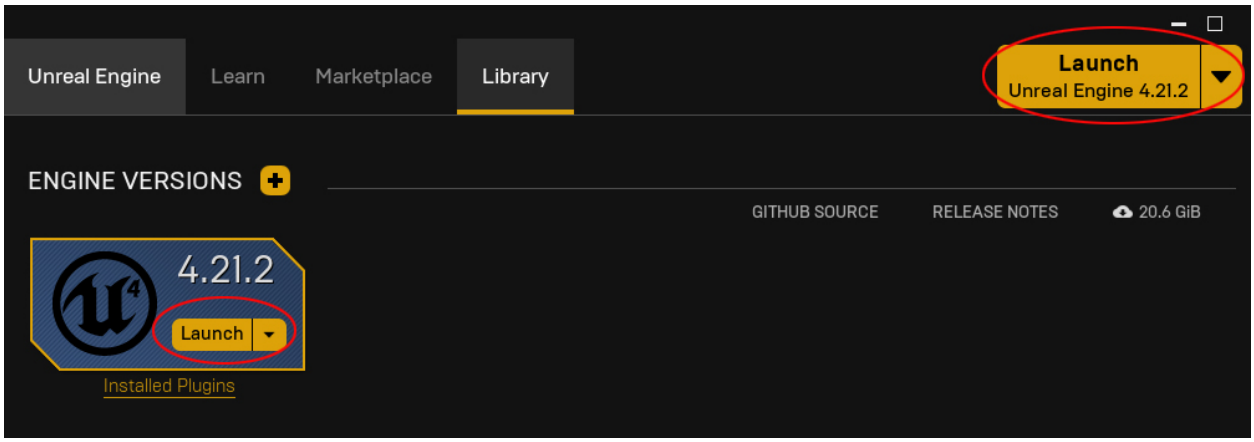
One project can be an entire game or it can be just a single level.

In order to start Unreal Engine 4 editor you will have to:

- Create a new project
- Open already existing project
- Open a project that you downloaded from Learn/Marketplace section

Let's create our first project.

Click on **Launch** button on top right or in the middle under Engine Versions. The version you launch will be the one you downloaded and installed.



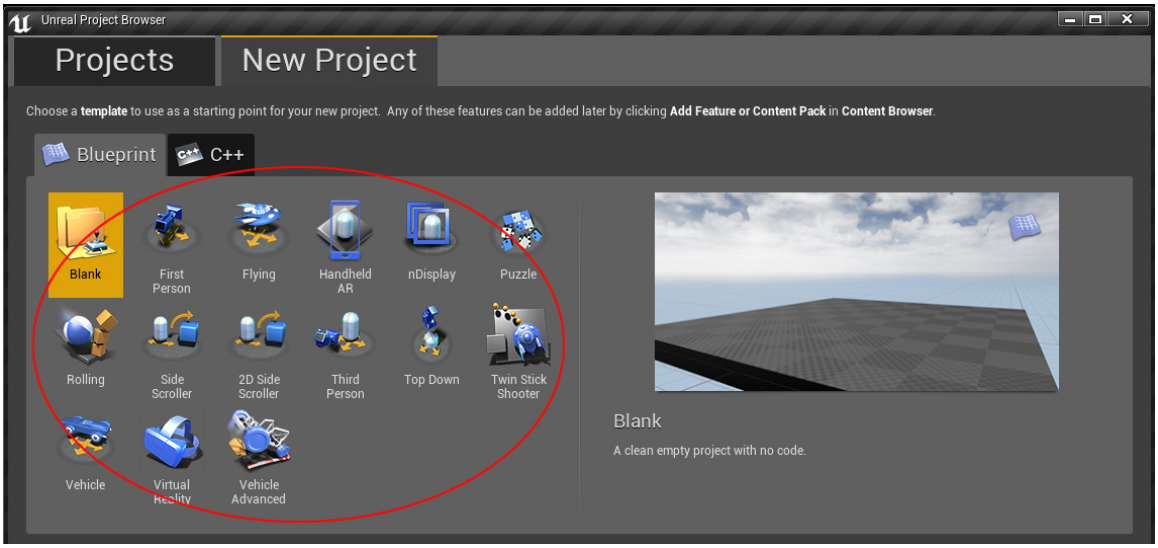
Remember: you can have more than one UE4 version installed. Whichever version you launch, is the one that Project will be created for.

When **Unreal Project Browser** opens up, switch tab to **New Project** and select **Blueprint** tab instead of C++:

Blueprint is a very powerful visual scripting language that requires no programming knowledge and you won't have to compile any C++ code. You can create entire games or just simple level interactions using Blueprint.

Once you get more advanced with UE4 you can learn C++ programming. But to keep things simple, begin with Blueprint. It will do most of the functionality you will ever need to start with - especially if you are a level designer or a game environment artist.

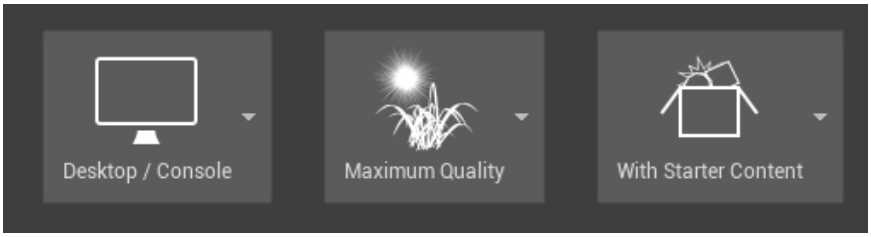
Choose a game template to use. You can choose FPS, third-person or any other available game templates. Blank will not add any templates. But to start things off, choose one of the game templates. It will be easier to learn UE4 when you have something to work with.



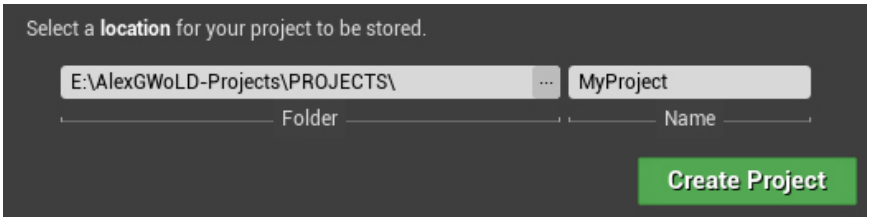
Next, choose the following:

- Desktop/Console
- Maximum Quality
- With Starter Content

You can change these 3 settings later through Project Settings any time. I will show you how later.



Select a location for your project to be stored. I choose to store all of my Unreal projects on a different drive other than my main C drive. Then name your project and click **Create Project**:



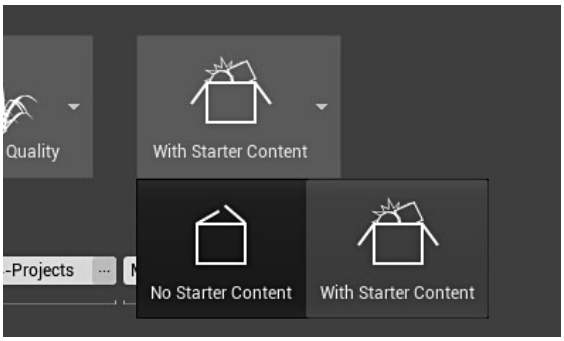
This will launch Unreal Engine editor and open your newly created project.

Creating your first UE4 project, remember 3 things:

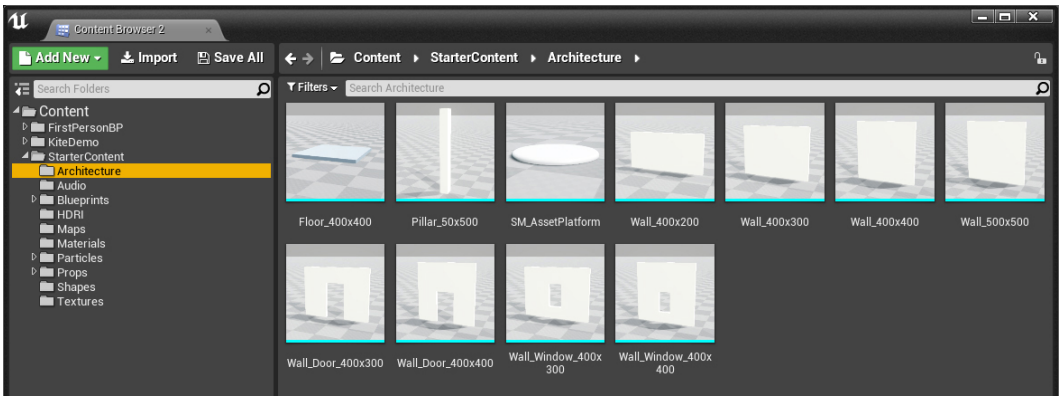
1. Use one of the available game templates such as First Person Shooter, Third Person Shooter, Top Down or Side Scroller template. Using these will give you basic gameplay mechanics with your project. You could reverse engineer them for learning and build on the existing functionality by adding your own changes.
2. Select to use Blueprint. Blueprint is a visual scripting language inside UE4 and does not require you to have any C++ programming knowledge in order to incorporate custom behavior in your project. Blueprints are extremely powerful and you can use it to create your own games and game types without having to learn C++ just yet.
3. Include Starter Content with you project. This will give you a few assets (Static Meshes, textures and materials) to use with your work.

3. USE STARTER CONTENT

When you choose to include **With Starter Content** option, this will add a series of assets into your project such as Static Meshes, material, textures, audio and effects that you can use to construct a simple environment with.

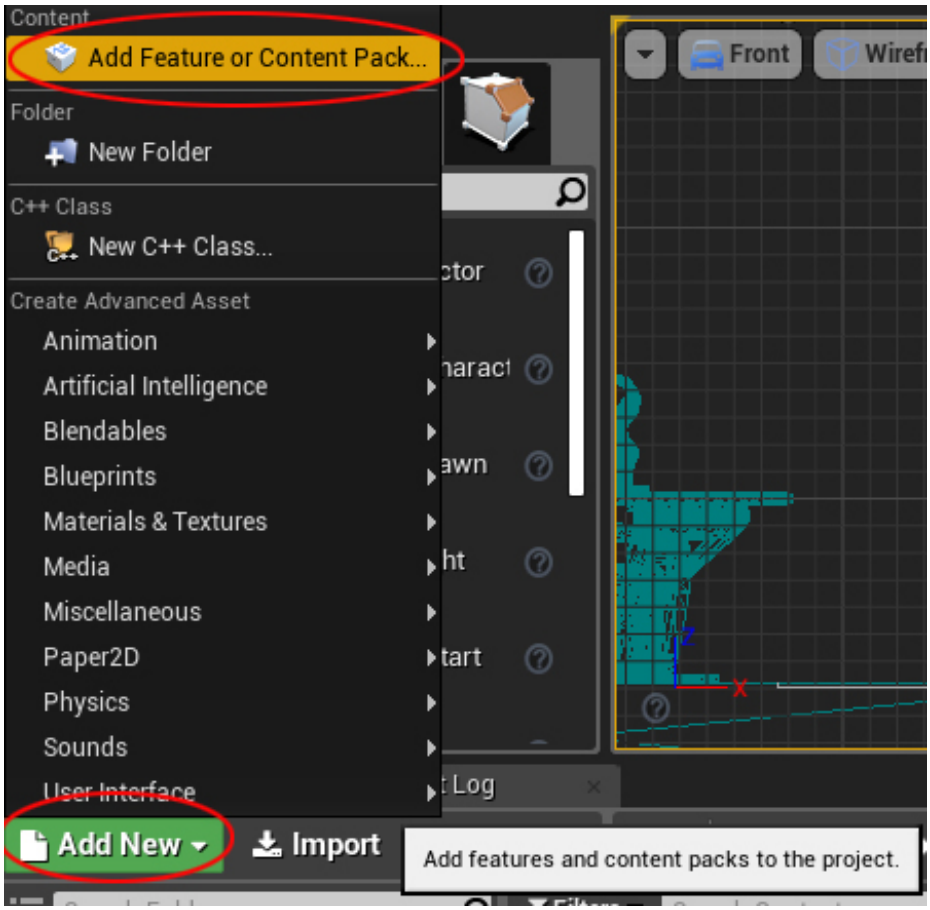


As a beginner, this gives you something to work with, rather than having to import and create assets yourself.



If you did not add Starter Content during project creation screen, you can add it at any time later within UE4 editor.

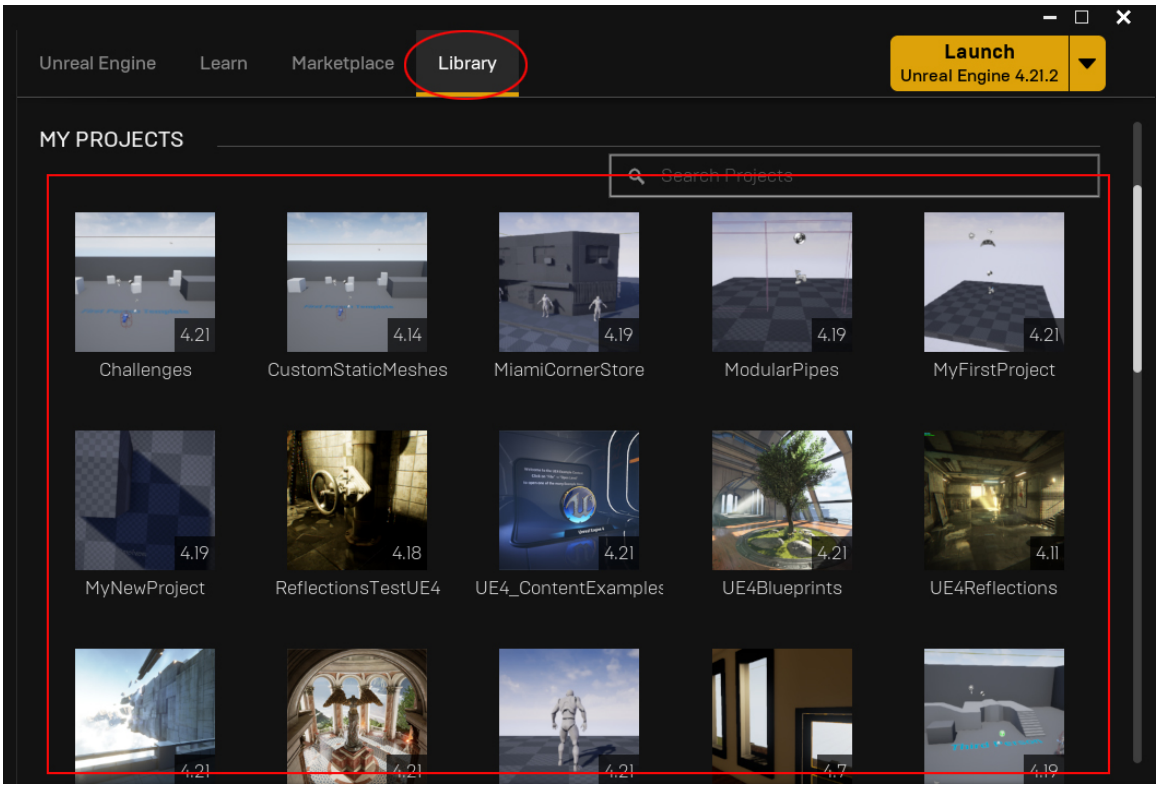
In the Content Browser, click on **Add New** and at the top choose **Add Feature or Content Pack**:



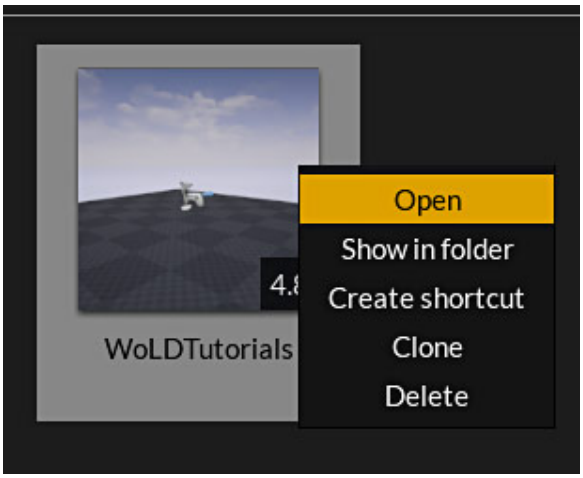
4. PROJECT MANAGEMENT IN UE4

As mentioned before, everything inside UE4 is contained within a project.

In the **Library** section you can open or delete any already created projects. Double click on the icon within the Library tab:



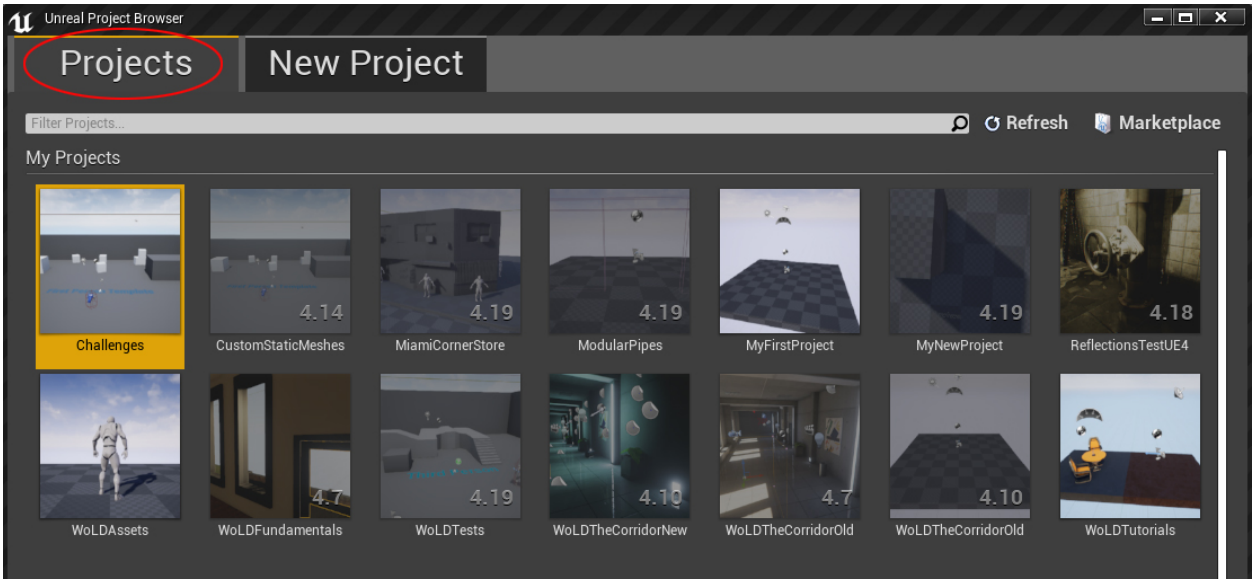
You can right click on the thumbnail with additional options such as Open, Show in Folder or Delete:



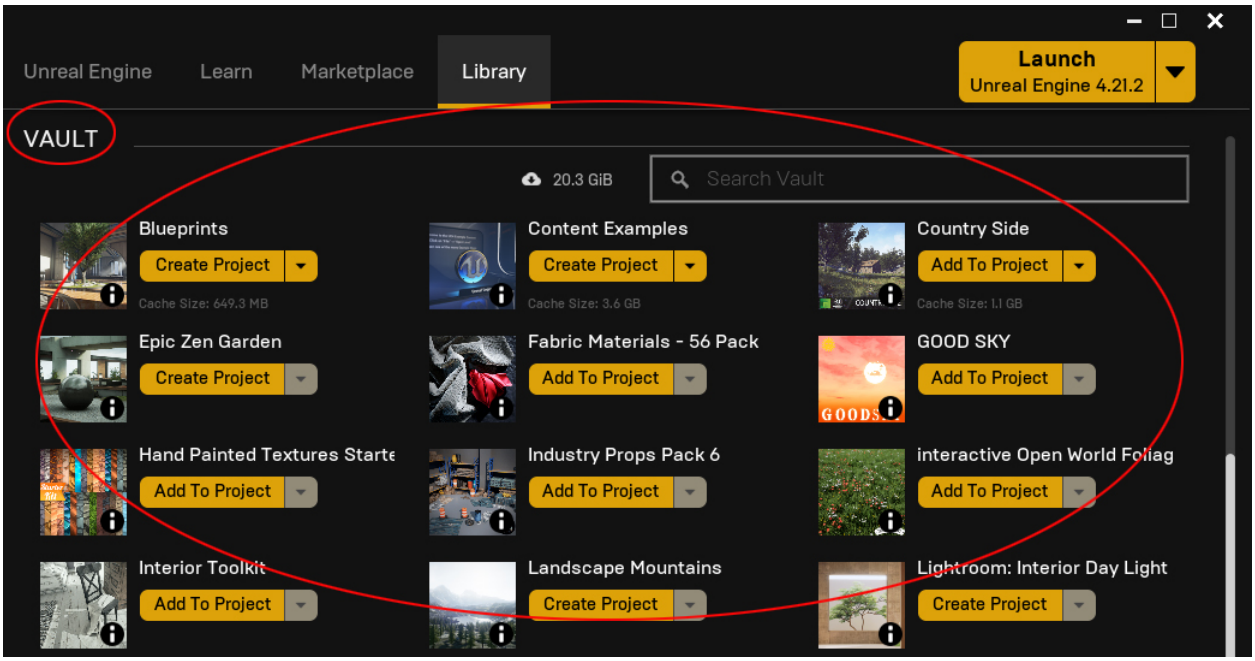
Show in Folder will take you to the location on the drive where your project is stored.

Once a project is created, it will always be available in the Library.

You can also manage projects through **Unreal Project Browser**:



Any content you download from Learn and Marketplace will appear in Vault section. More on Marketplace and Learn Examples content later in this guide.

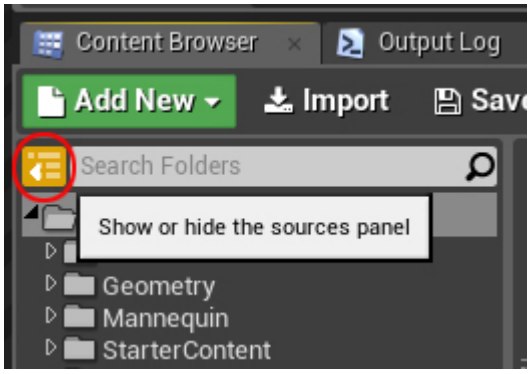


5. FIRST TIME INSIDE THE EDITOR

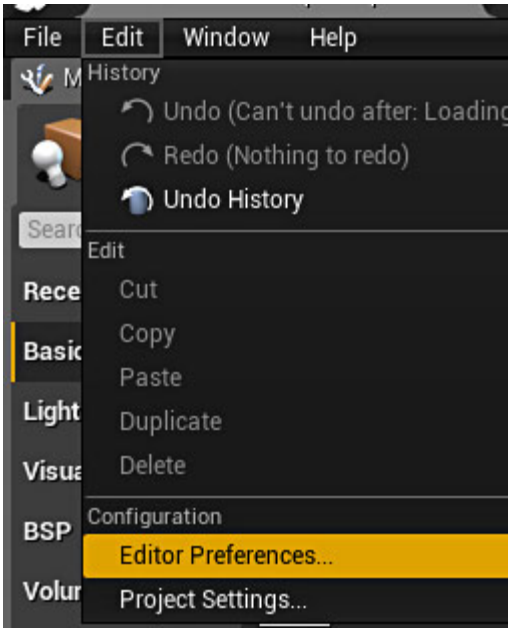
Once a new project has been created, UE4 editor will be launched automatically.

Here are some important steps to take before you do anything.

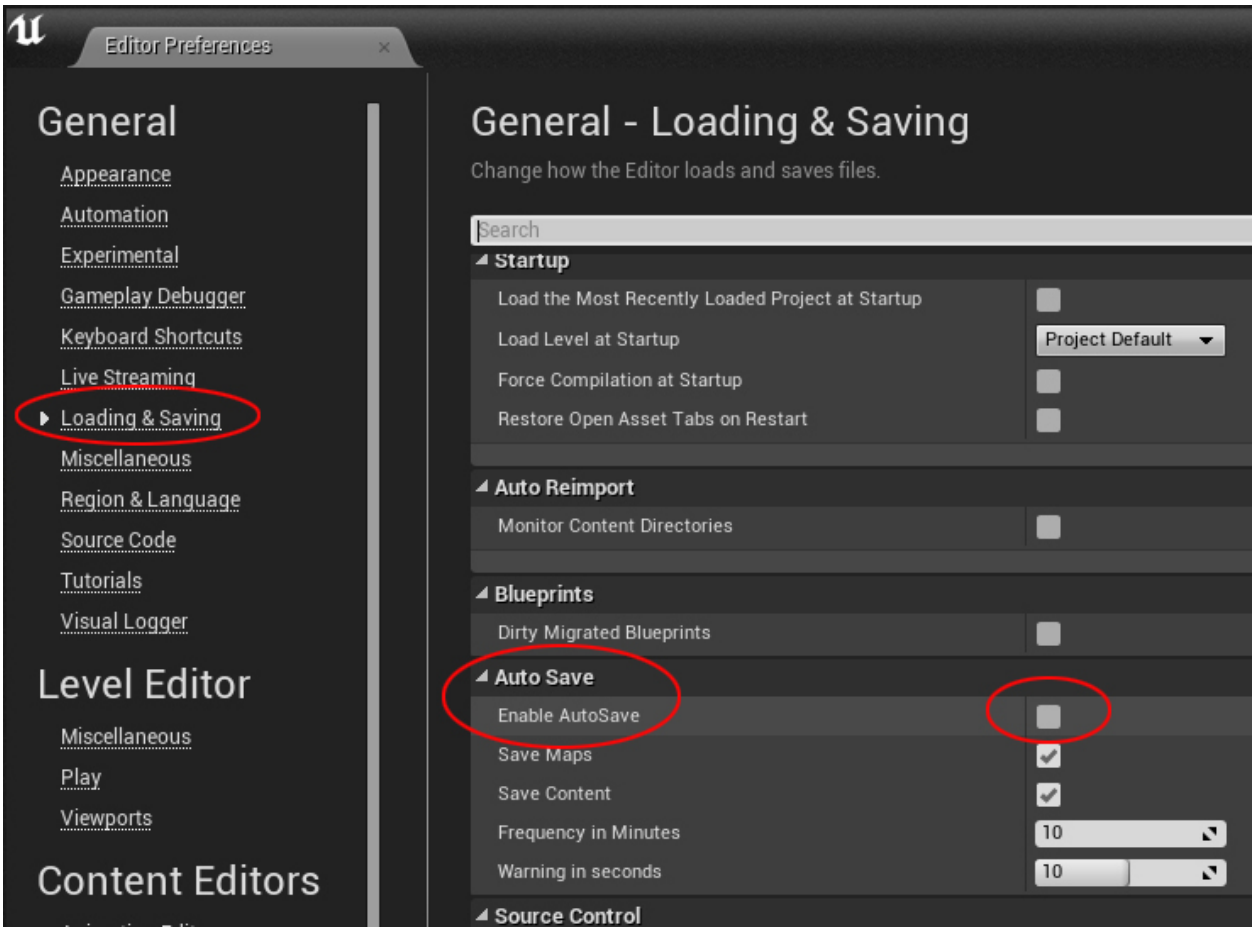
In the Content Browser click on **Extended View** icon to **Open Source Panel**, this makes it easier to navigate around your content through the Content Browser (covered later).



Next go to **Edit → Editor Preferences**:

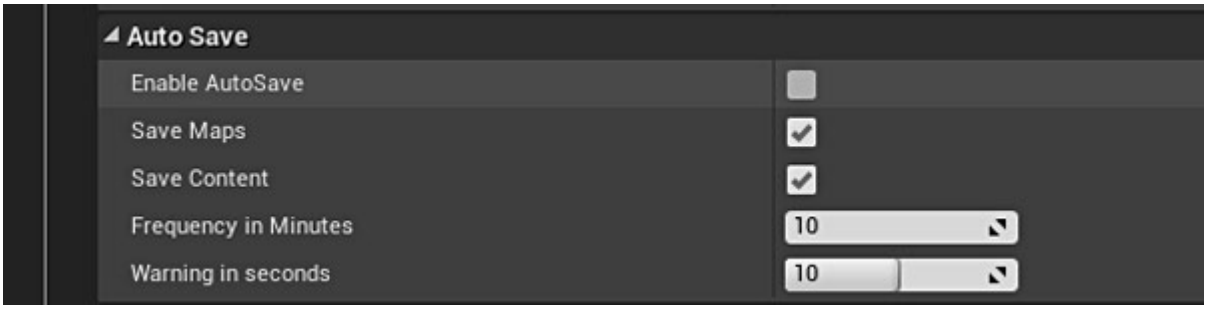


Under **Loading & Saving** and **Auto Save** section, choose to **Disable AutoSave**:

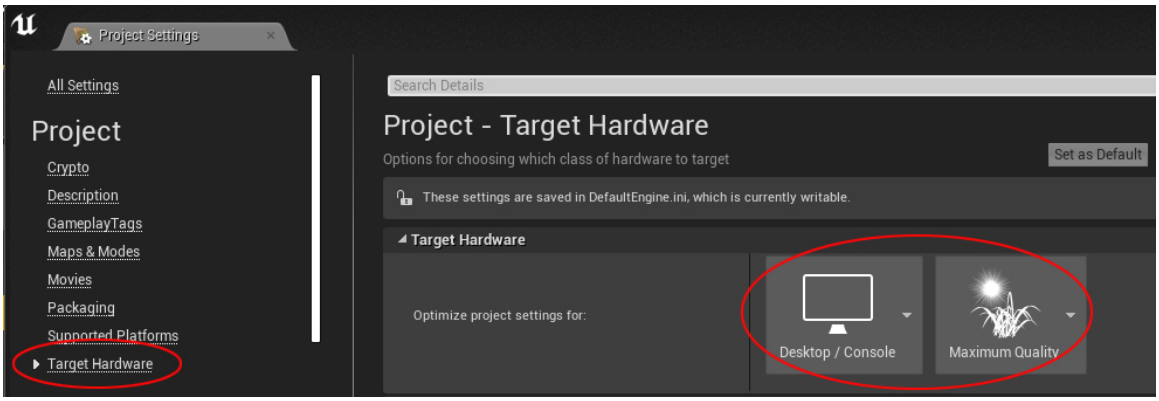


Now, this option is a personal preference. I disable auto save functionality because I like to control when I save my work. But, this is a very useful function to keep enabled, if you want.

If you choose to keep Auto Save enabled, set how often you want the editor to automatically save your work and how often to warn you about when it is saving. You also have additional options what to save Maps and Content as well as the frequency and warning times.

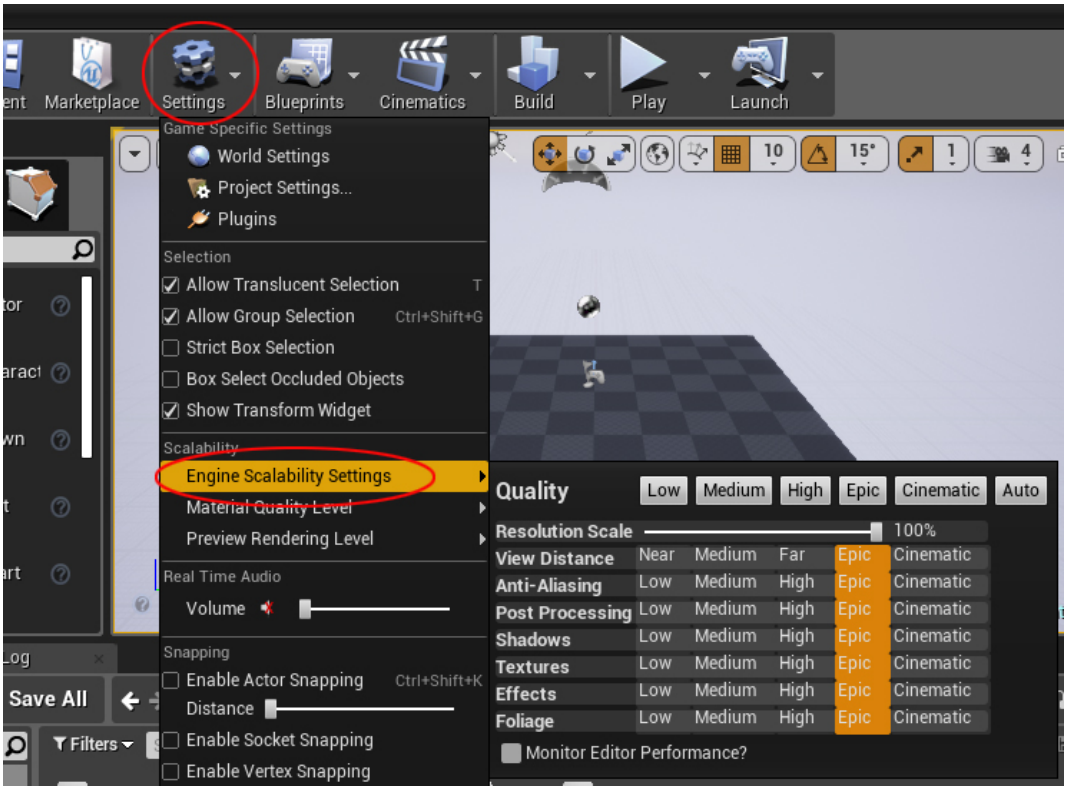


During New Project Creation we had options to choose from. You can update them by going to **Edit → Project Settings → Target Hardware**:



6. ENGINE SCALABILITY SETTINGS

Engine Scalability Settings allow you to adjust the quality of various features in order to maintain the best performance for your game on different platforms and hardware. To adjust **Engine Scalability Settings** click on the Settings button and scroll down the Engine Scalability Settings options:



Lowering the settings can often help to make the editor more responsive while working. Remember that you need to change it back before playing if you want to have the full effect of your work appear.

Also sometimes the environments in the editor don't look as good as they should. Engine Scalability Settings were lowered due to your computer hardware or quality chosen during project creation.

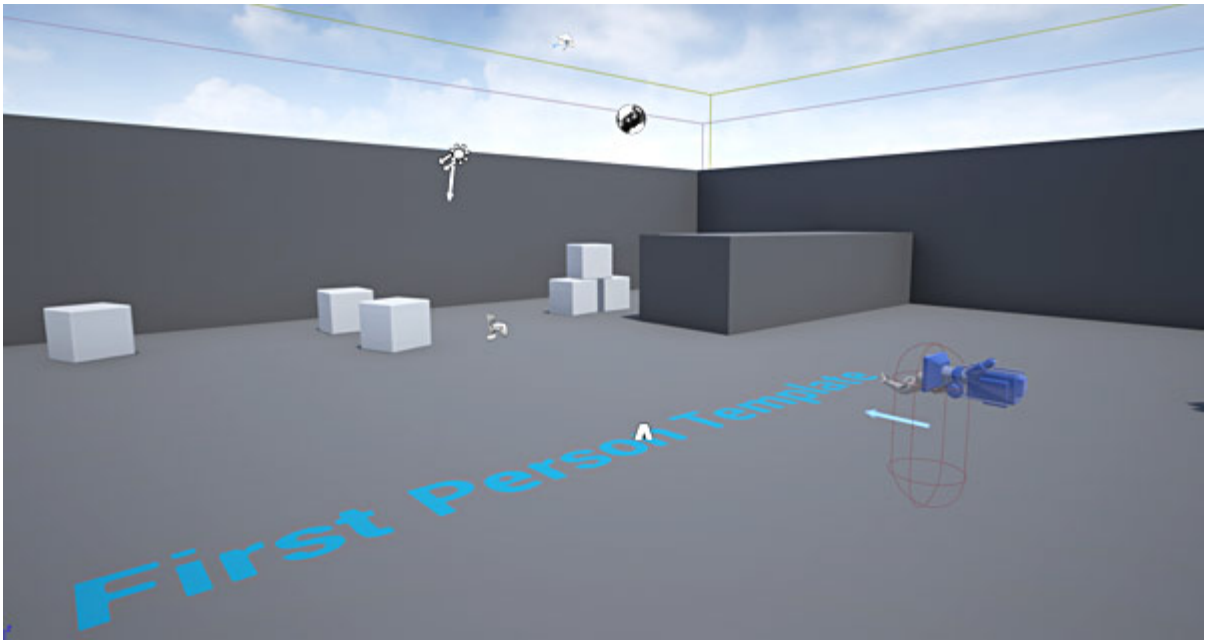
I keep the following options on as I work:

a. **Engine Scalability Settings:** Epic
b. **Material Quality Level**: High
c. **Preview Rendering Level**: Shader Model 5

7. CREATING, SAVING AND OPENING LEVELS

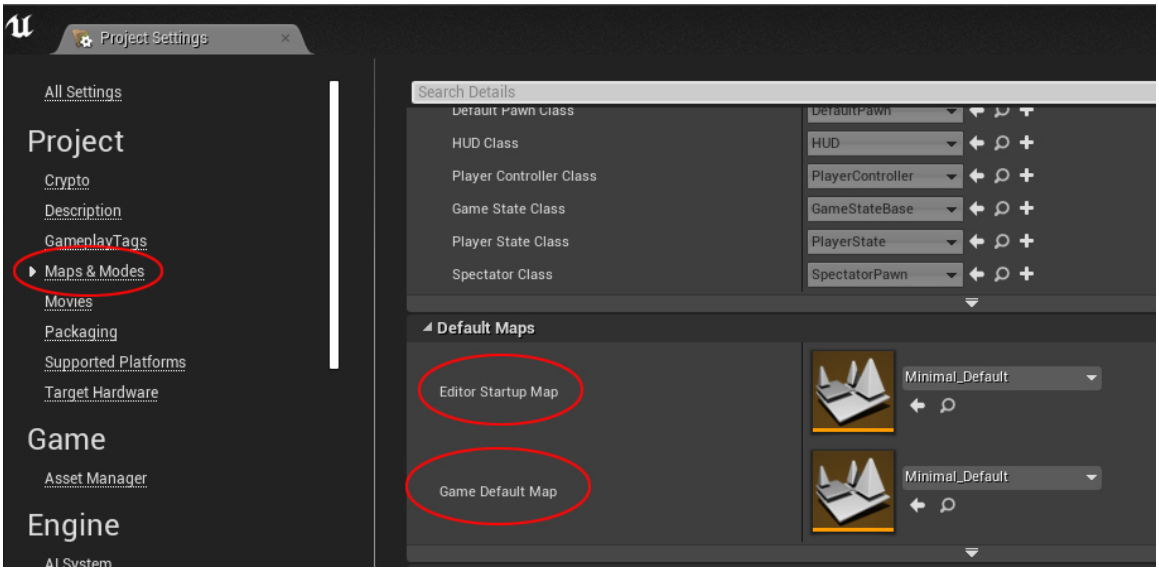
As you launch a project, you will have a default level/map open automatically. Depending on the type of game template you chose, this starter map will vary.

Below is the default starter map from FPS Shooter Game Template:

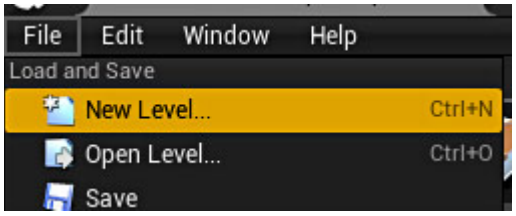


If you chose a Blank project, this map will be the default opened map:

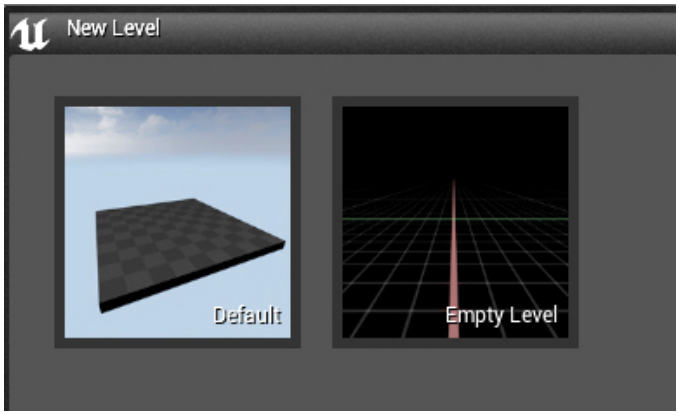
You can change starter default map to any other map. To update this go to **Edit → Project Settings** and under **Maps & Modes** and Default Maps you will have an option to switch to Game Default Map and Editor Startup Map:



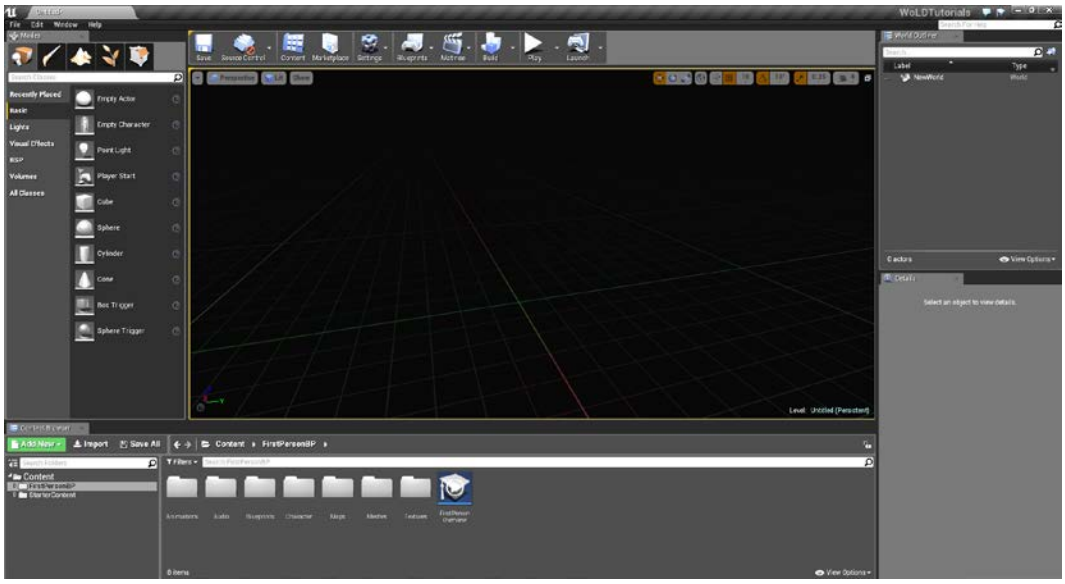
Start a new blank map without any actors go to **File → New Level**:



Choose between **Empty Level**:



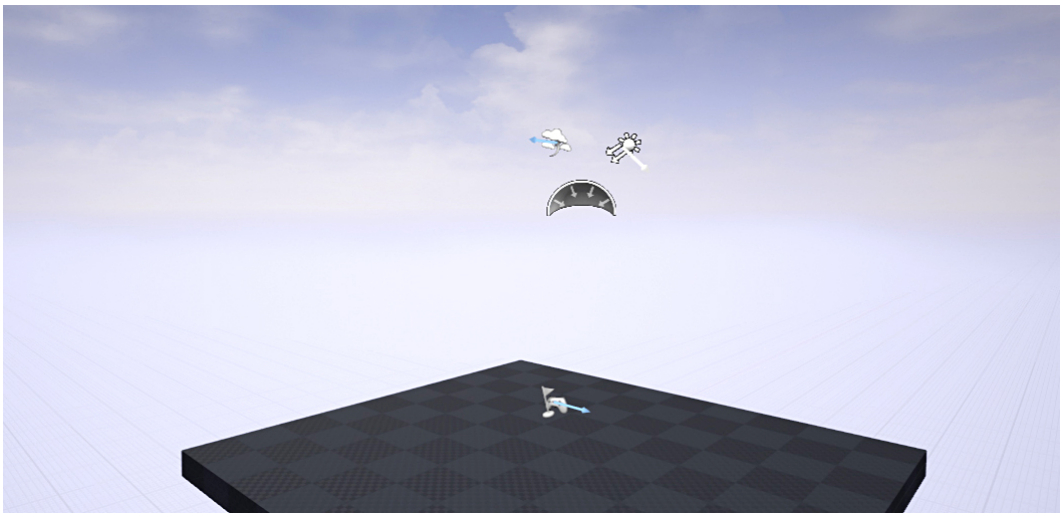
Empty Level will be a blank map without anything in it:



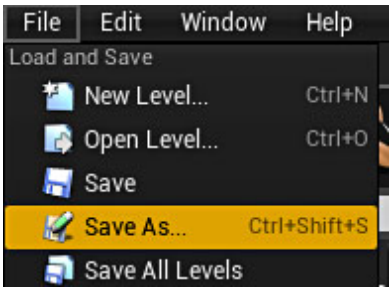
But to get things started, choose a Default Level.

Default Level gives you necessary actors to start with such as:

- Ground plane
- Light source (Directional Light and Skylight)
- Player Start
- Sky Sphere
- Atmospheric Fog



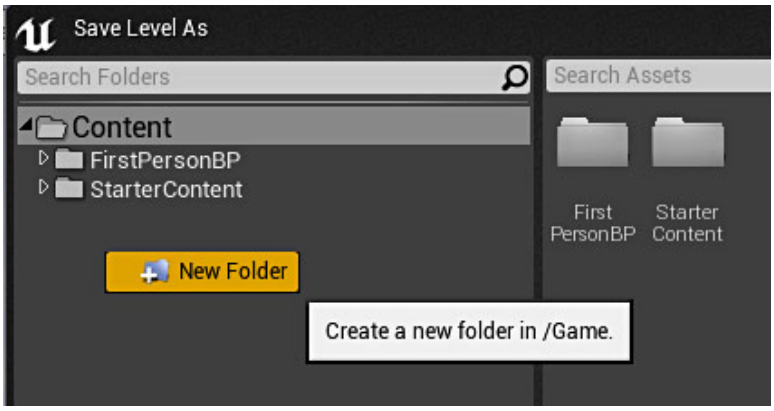
Save any level you are currently working on, go to **File → Save As**:



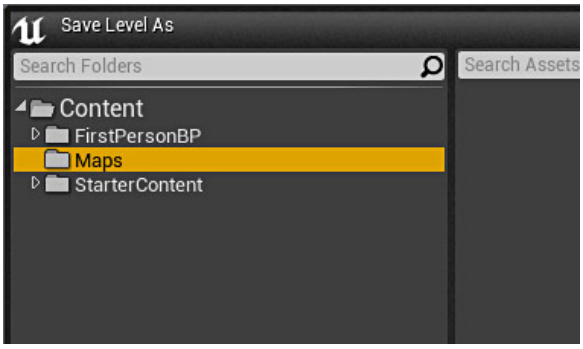
Choose a folder to store your map in, name it and choose Save:

Important: make sure that all of maps and content you save are inside the Content folder of your project**.** Do not save anything outside Content folder.

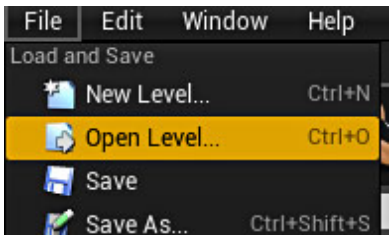
You can organize your maps better by creating a new folder inside the Content folder. Right-Click in empty gray space and choose New Folder:



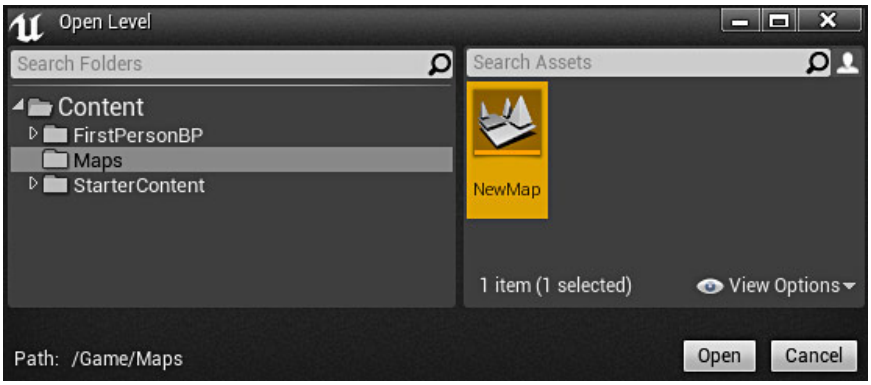
Name the folder Maps:



Open any existing/saved level, go to **File → Open Level**:



Navigate into the folder where the map is saved and click Open:

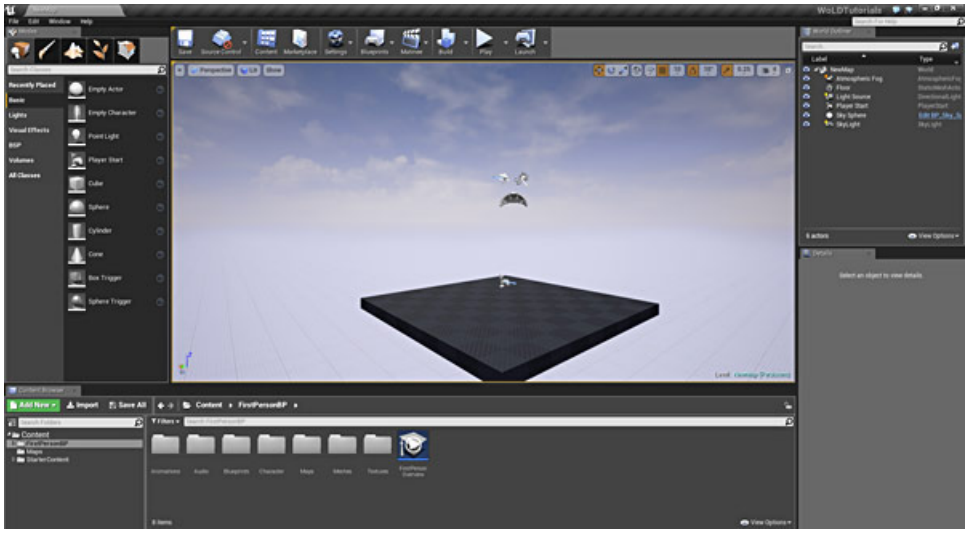


- **Ctrl + S** = Save Current
- **Ctrl + Alt + S** = Save Current As
- **Ctrl + Shift + S** = Save All (will save all unsaved content such as maps, Static Meshes, Blueprints, Materials etc.)

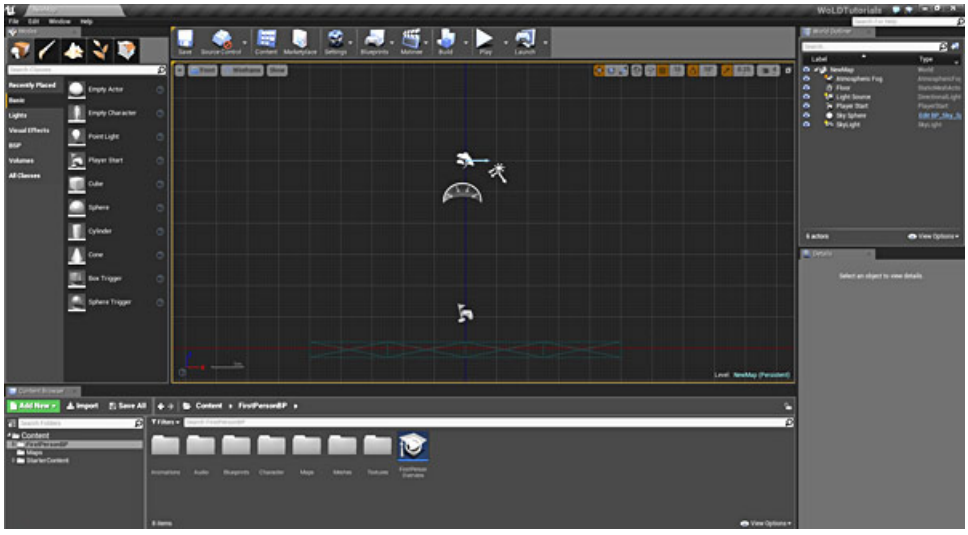
8. HOW TO WORK WITH VIEWPORTS

There are two types of viewports: **perspective and orthographic**.

Perspective is the world view. It is how your level looks from the point of the player inside the game:



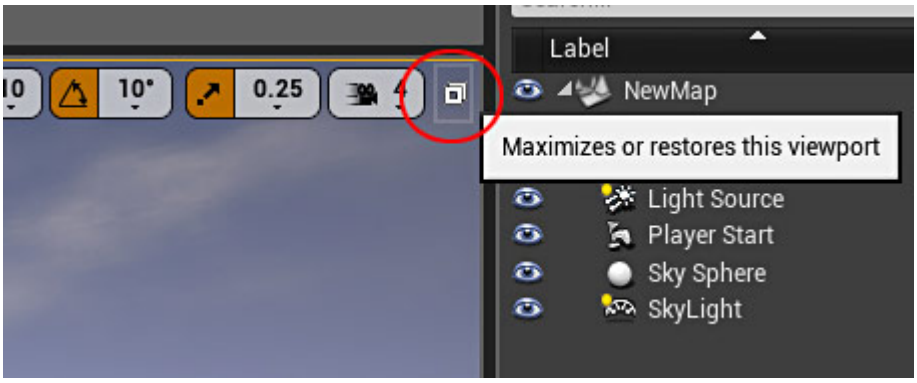
Orthographic are schematic wireframe grid views (front, side, top):



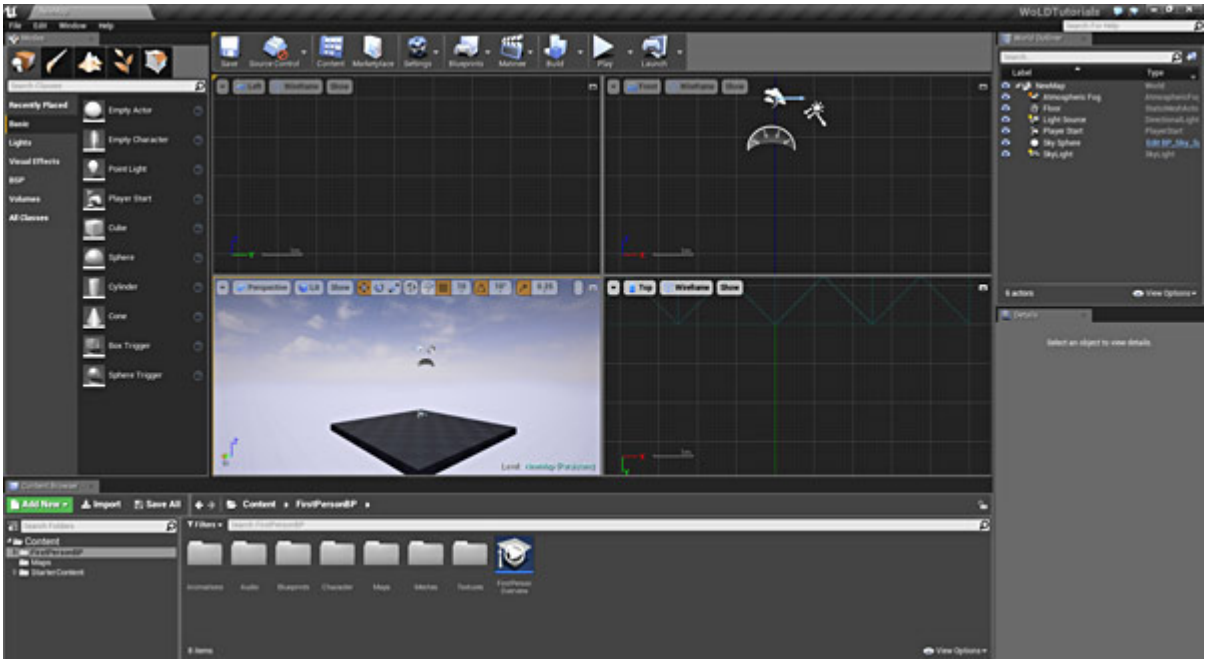
Perspective is going to be your primary view through which you work and construct your map.

Switch to 2x2 view of the editor by restoring the perspective viewport.

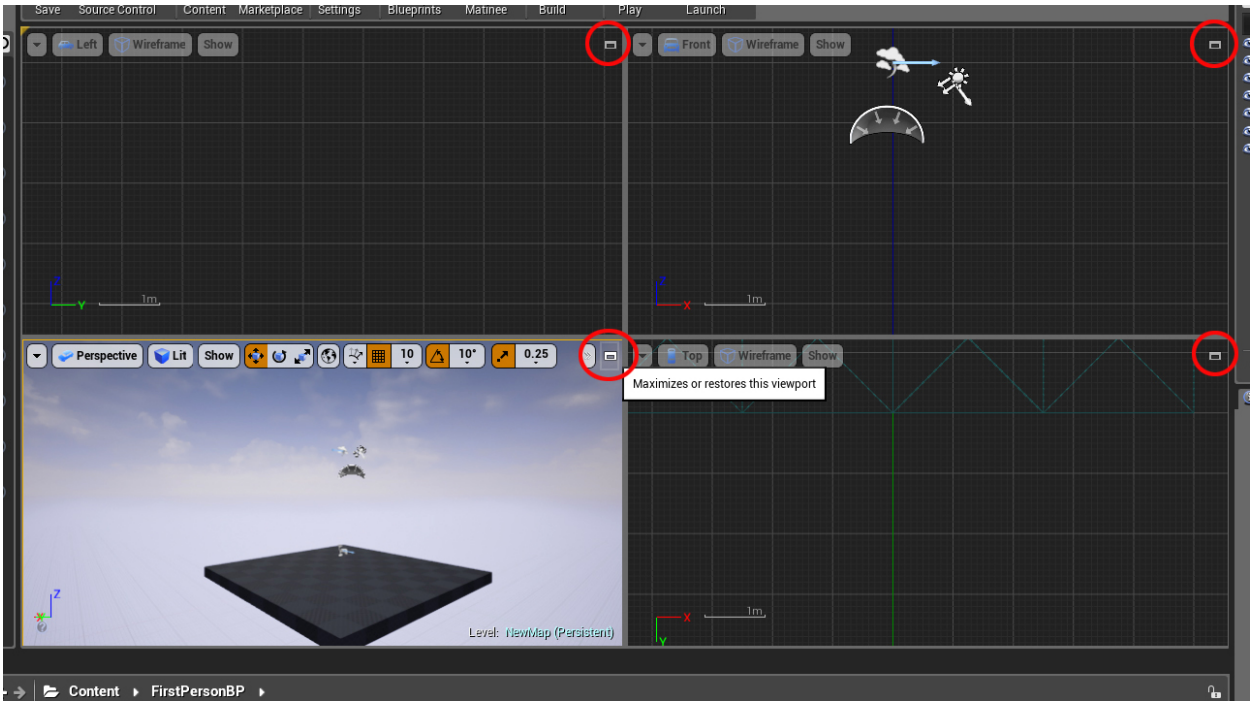
Click on **Maximize / Restore** icon:



You will now have all 4 views on the screen at the same time.



Maximize any viewport to full window by clicking on the **Maximize / Restore this viewport** icon within each viewport:



Use the shortcut **Alt + X to Maximize/Restore viewports**.

- **Alt + X** = Maximize/Restore Viewports

There will be a yellow border highlight around each viewport, telling you that viewport is active.

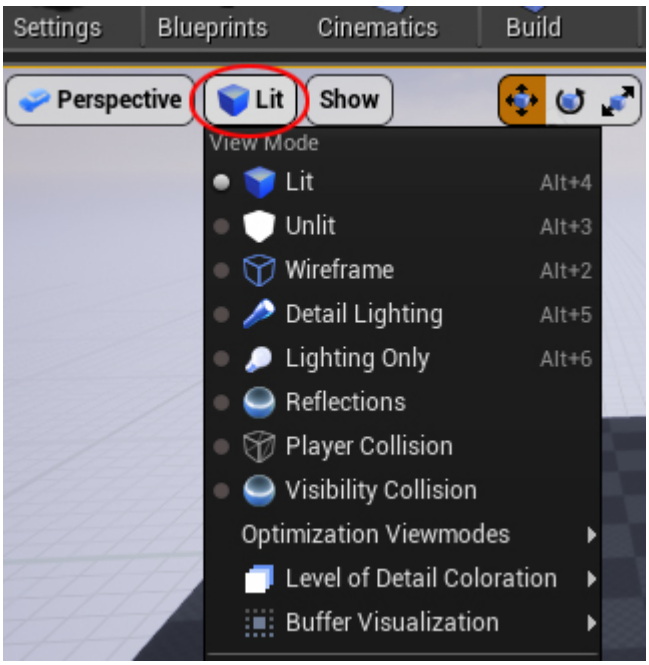
Make viewport active by Left Click or Middle Mouse Click inside that viewport:

- Left Click or Middle Mouse Click in Viewport = Make Viewport Active

I often use the following shortcuts as I work in perspective view to switch to any orthographic viewport:

- **Alt + G** = Perspective
- **Alt + H** = Front
- **Alt + J** = Top
- **Alt + K** = Left/Side
- **Alt + Shift + K** = Right/Side

You can change the perspective viewport to use different view modes, such as Unlit, Wireframe, or Detail Lighting etc. Use the drop down menu to choose:



Each view mode has its own hot key. For beginners, keep your perspective viewport to Lit (Alt+4).

- **Alt + 2** = Wireframe
- **Alt + 3** = Unlit
- **Alt + 4** = Lit
- **Alt + 5** = Detail Lighting
- **Alt + 6** = Lighting Only

There are other handful of options under Show and Arrow drop down menu; including Realtime, Game Mode and Full View/Immersive Mode.

Game mode will show you how the environment will look during gameplay. All editor actors become hidden. Press G for Game Mode.

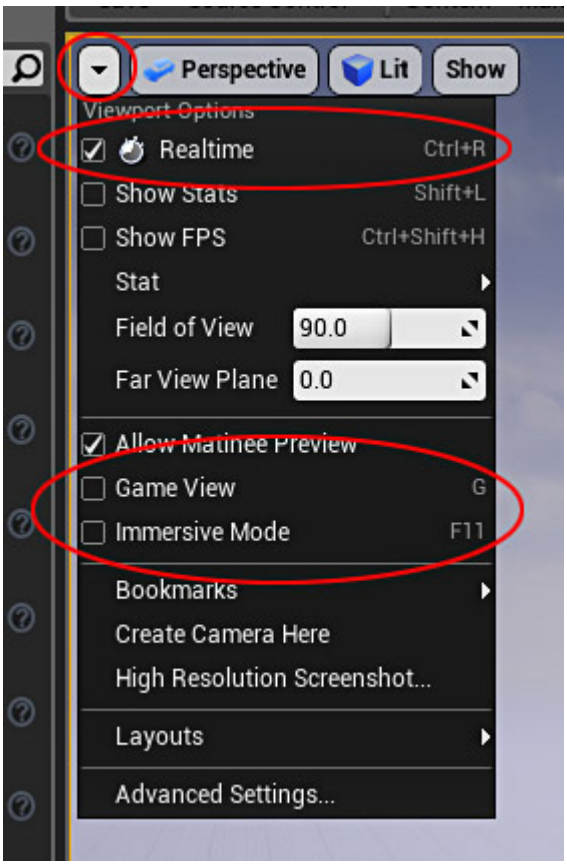
- **G** = Game Mode

Real time mode will display animated and real time effects such as materials and particles. Press Ctrl + R for real-time mode.

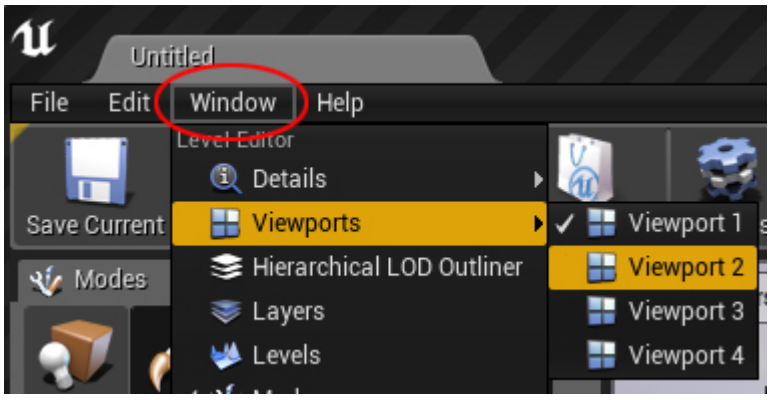
- **Ctrl + R** = Real Time Mode

Immersive mode will make the active viewport fill the entire screen.

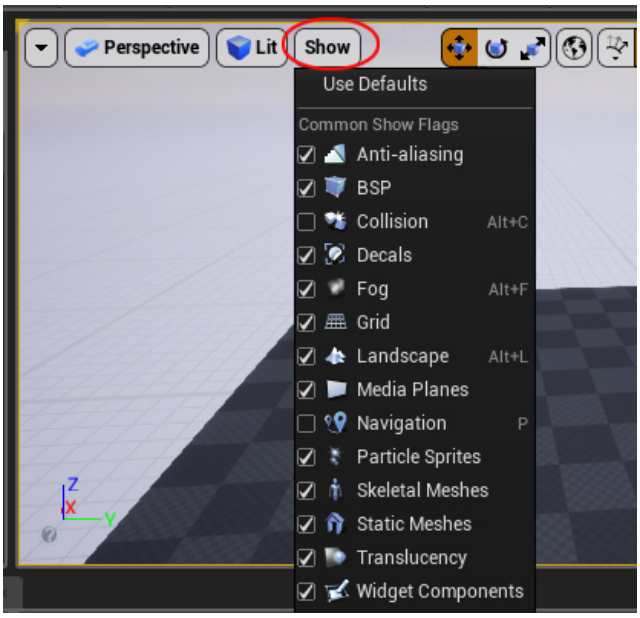
- **F11** = Immersive Mode



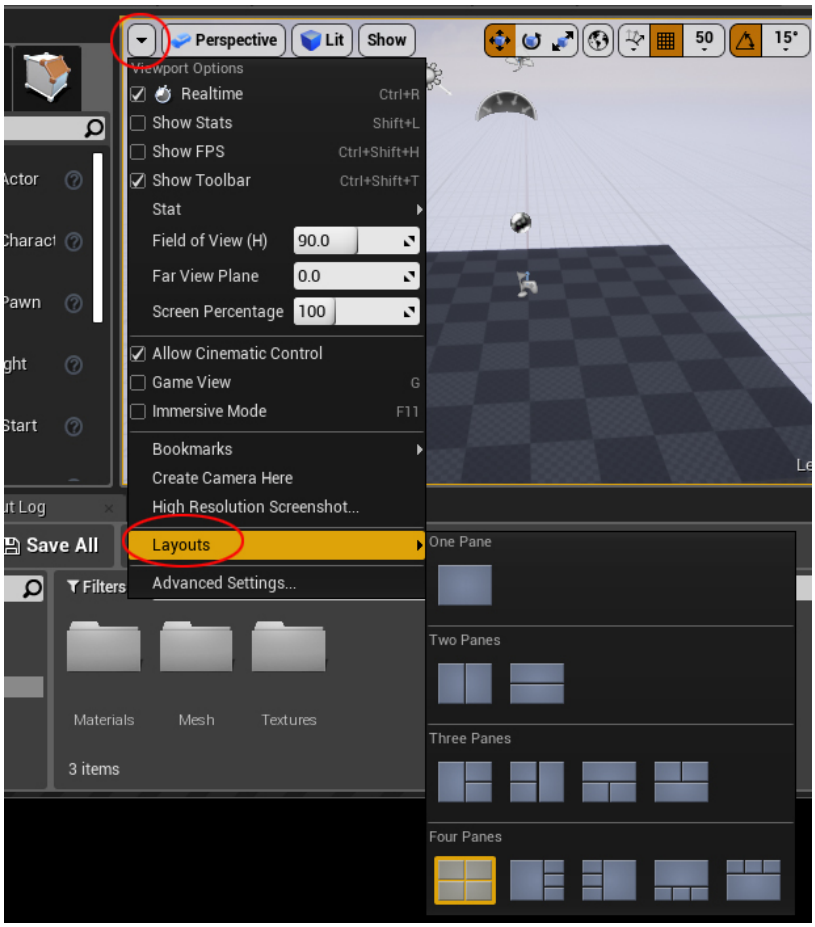
At any time you can open a new perspective window by going to **Window → Viewports**:



You can hide various assets from view under **Show** within the viewport drop-down menu - this is very useful:



If you don't like 2x2 configuration layout you can choose others you feel more comfortable with. Go to **Viewport Arrow → Layouts → One, Two, Three, Four Pane Layouts**:



9. VIEWPORT NAVIGATION

Let's cover everything you need to know for navigating each viewport.

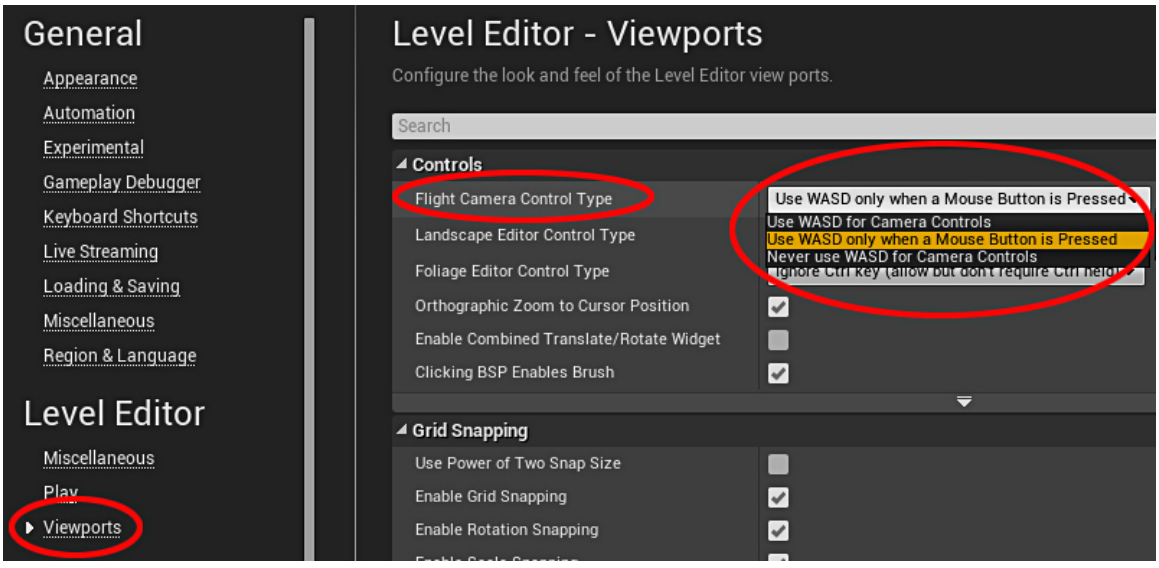
Few hot key abbreviations we'll be using:

- **RMB** = Right Mouse Button
- **LMB** = Left Mouse Button
- **MMB** = Middle Mouse Button

8 Ways Navigating Perspective Viewport:

- **Hold RMB + Move Mouse** = Look Around
- **Hold RMB + W, A, S, D (keys)** = Move Forward/Back/Side-to-Side
- **Hold LMB + Move Mouse** = Move Forward/Back; Look Left/Right
- **Hold MMB + Move Mouse** = Pan Up/Down/Left/Right
- **LMB + RMB + Move Mouse** = Pan (Same as MMB + Drag)
- **Mouse Wheel** = Move Forward or Back
- **Hold RMB and Press Z or C** = Zoom In/Out
- **Hold RMB and Press Q or E** = Move Camera Up/Down

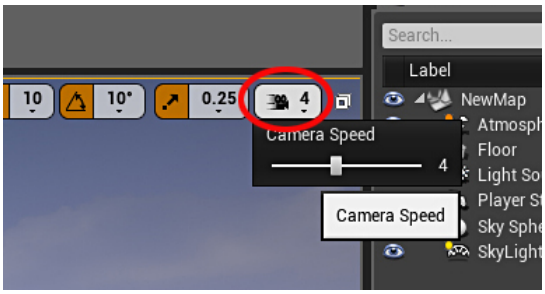
You can disable holding Right Mouse Button as you press WASD keys to navigate, go to **Edit → Editor Preferences** and under Viewports, Flight Camera Control Type choose **Use WASD for Camera Controls**:



Although I keep this at default and I use holding down the right mouse button.

Perspective View Camera Speed:

In perspective viewport, camera speed at which you look around can be adjusted. Use the slider to lower number (slower) or higher number (faster):



You can also adjust camera speed in real-time by holding down the Right Mouse Button (while using WASD controls) and scrolling the mouse wheel forward or back to speed up or slow down the camera's movement.

- **Hold RMB + WASD then Scroll Mouse Wheel Forward/Back** = Fine Tune/Adjust Camera Movement Speed

4 Navigating In Orthographic Viewport:

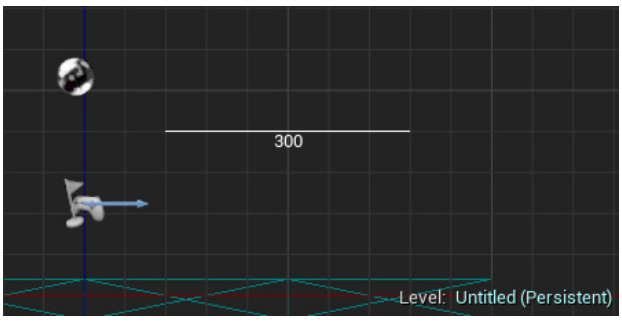
Navigation in orthographic viewports is a lot simpler.

- **Hold RMB + Mouse Move** = Pan
- **Hold LMB + Mouse Move** = Marquee Selection
- **Hold LMB + LMB and Mouse Move Forward/Back** = Zoom In/Out
- **MMB** = Measuring Tool

Measuring Tool:

Measuring Tool is very useful to measure distance in the editor. You can access this tool only in Orthographic viewports.

- **Hold and Drag MMB** = Measuring Tool



10. WORKING WITH ACTORS/OBJECTS

Any object you place in the level is called an actor this includes Static Meshes, audio, decals, volumes, brushes etc. Actor and object are used interchangeably as they mean the same thing.

- Actor = Object

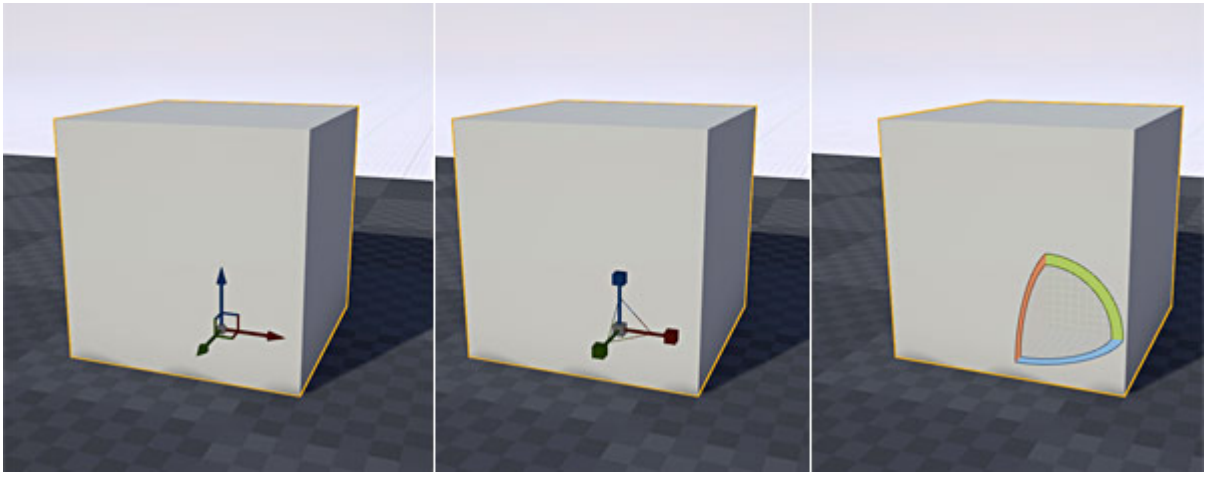
Selecting, Deselecting, Deleting:

Very basic functions for selecting, deselecting and removing objects:

- **Left Click** = Select the Object
- **Escape** = Deselect the Object
- **Delete** = Remove the Object
- **Hold Ctrl + LMB** = Add to or Remove from a Selection

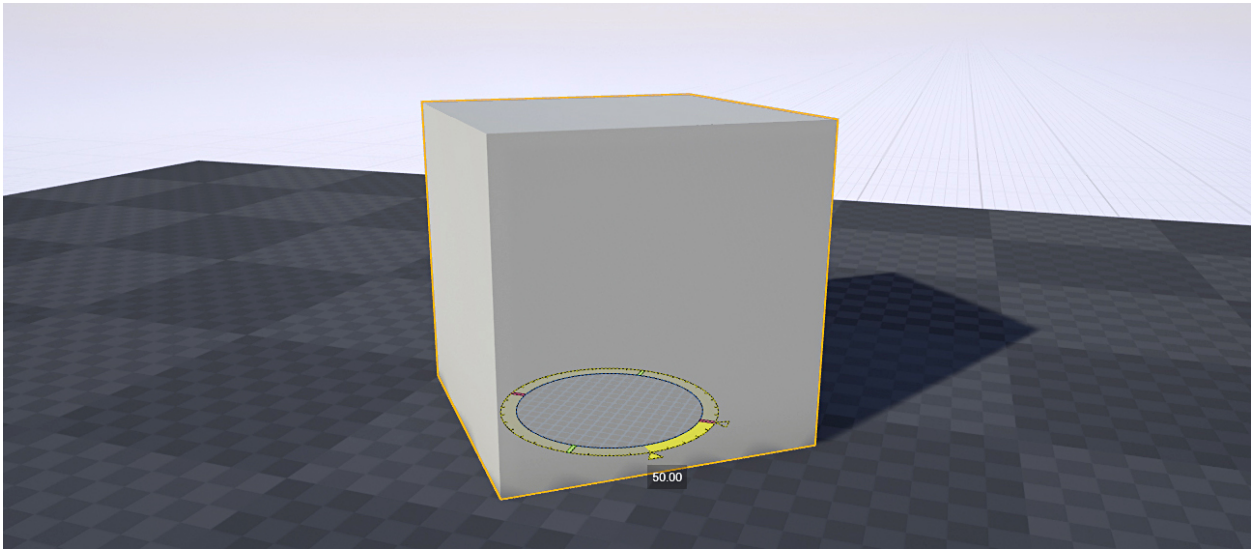
Move, Rotate, Scale:

After you select an object in the viewport you will see a move/rotate/scale gizmo appear:



- **W** = Move
- **E** = Rotate
- **R** = Scale
- **Spacebar** = Cycle Between Move/Rotate/Scale Transformation Gizmo

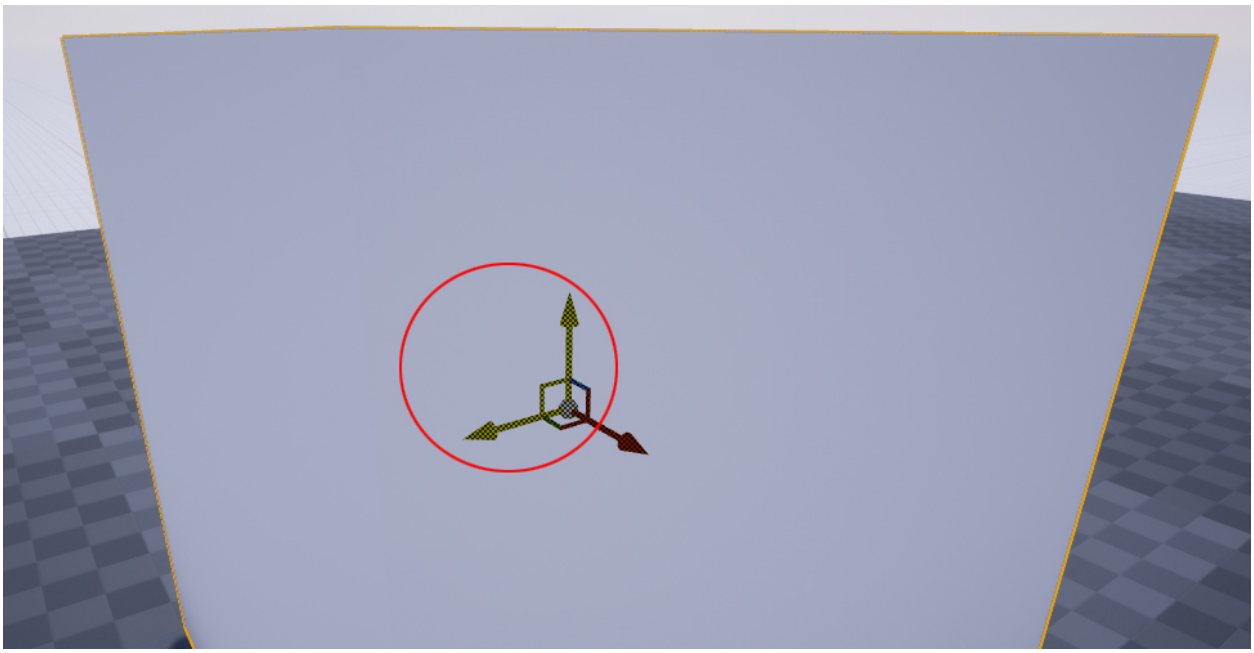
To move/rotate/scale the selected object, left-click hold and drag on the appropriate transformation handle within that gizmo, it will become highlighted yellow as you hover over it that gizmo:



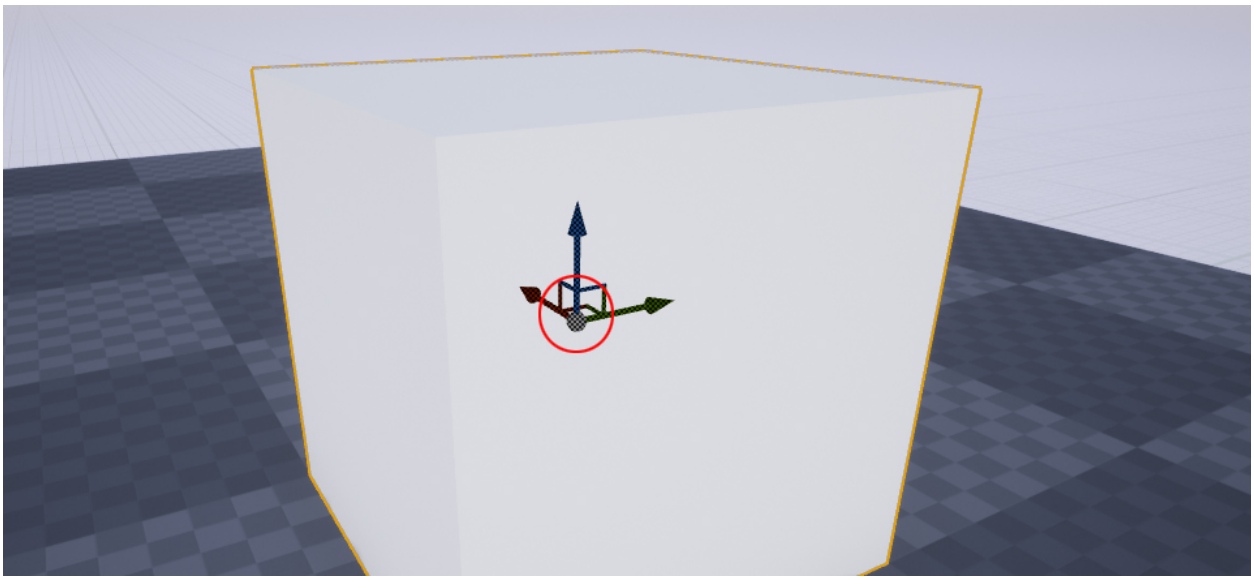
If you don't see the Move/Rotate/Scale gizmo, press **G** key to disable game mode.

You can move, rotate, scale the actors in one direction, on X, Y or Z.

Or, you can move objects constrained to 2-axis, XY, XZ or YZ. Press W for move tool and position the mouse over the gizmo. When you see two of them highlight in yellow, left-click hold and drag to move that object along 2-axis:



To freely move an object along XYZ in perspective viewport, left click hold and drag right in the middle of XYZ:



Moving Along with the Object:

You'll often want to move the view along with the object at the same time. To do this:

- **Hold Shift + LMB** = Move Along with the Object

Moving Objects Without Selecting Specific Axis:

You can move objects without having to select a specific direction axis. To do this, select the object first then:

- **Hold Ctrl + LMB** = Move Object Along X
- **Hold Ctrl + RMB** = Move Object Along Y
- **Hold Ctrl + LMB + RMB** = Move Object Along Z

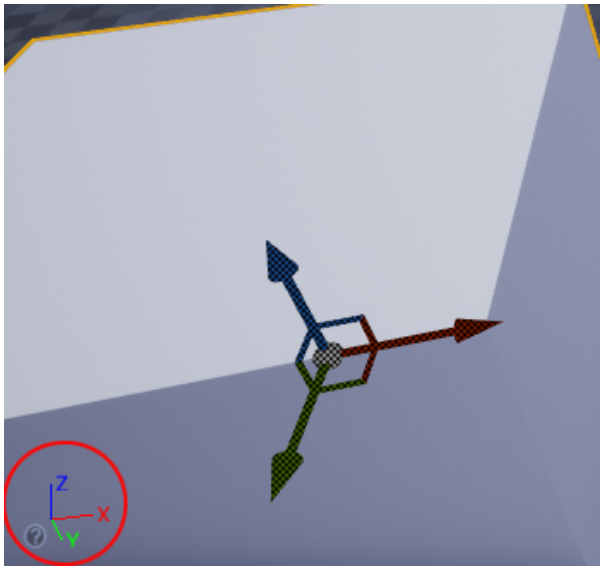
How to find what are X, Y or Z axis?

XYZ is the direction in 3d space.

In UE4:

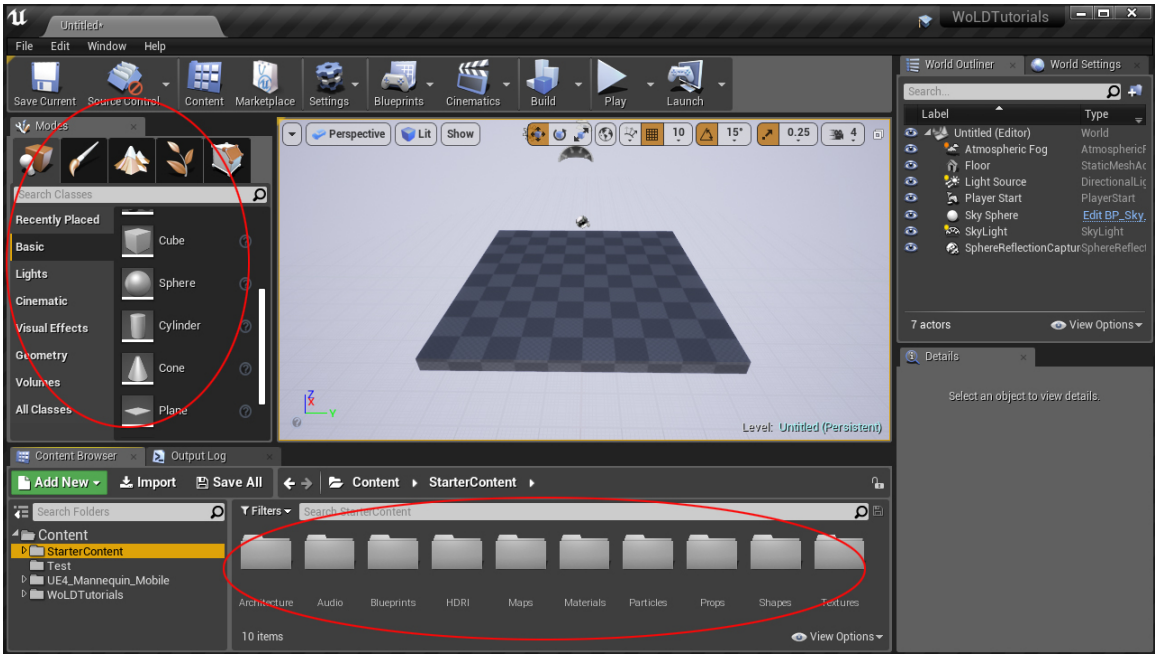
- **X** = forward/back (red color)
- **Y** = left/right or side to side (green color)
- **Z** = up/down (blue color)

On the bottom of the perspective viewport you'll see XYZ icon. This will tell you the XYZ directions and colors. Use this to look at the object's gizmo to determine which handle belongs to which axis:



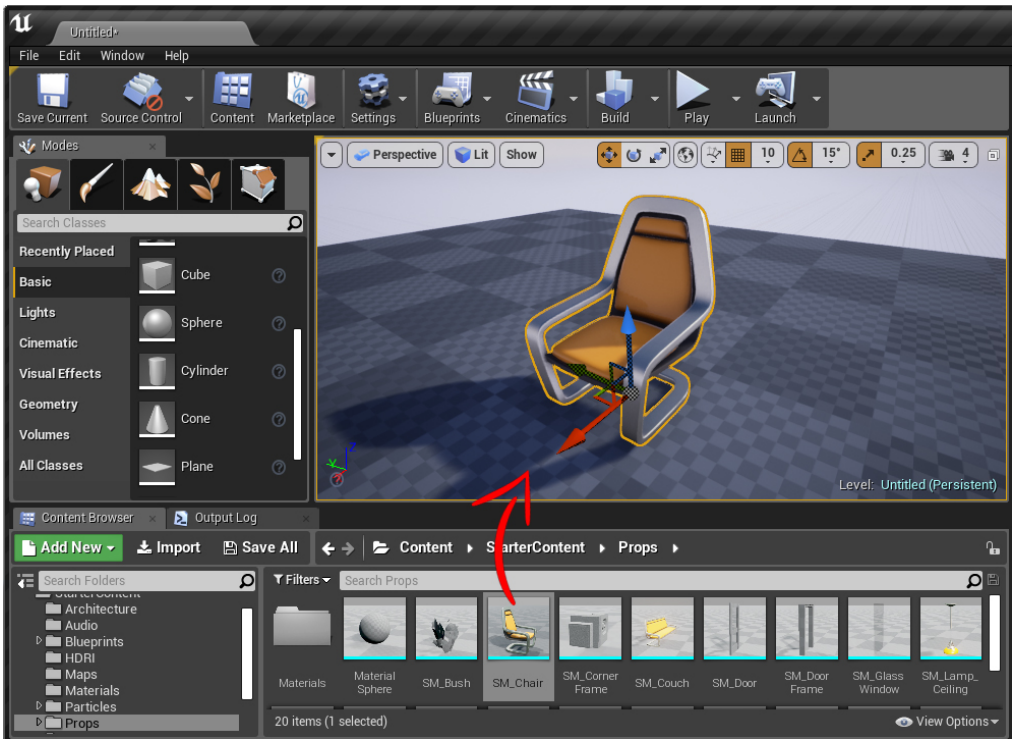
Inserting Objects:

There are two places to insert actors from - Content Browser and Modes Panel.

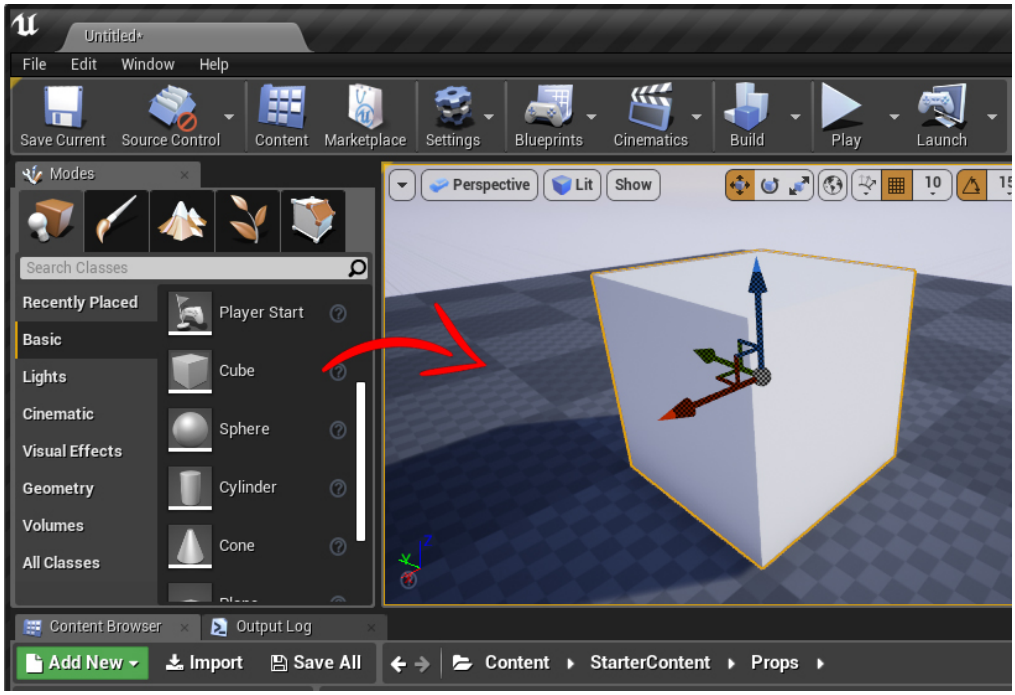


We'll go more into depth what each of them do but right now, here is how to insert objects into your level.

From **Content Browser** (located on the bottom of the interface) simply left-click and drag right from the Content Browser into perspective viewport:



From **Models Panel** (located on the left of the interface) do the same thing, left-click and drag right from the Models Panel into perspective viewport:



Duplicating Objects:

There are many ways to duplicate objects. Select any objects in perspective viewport and do one of the following:

- **Ctrl + W** = Duplicate
- **Hold Alt + Left Click Hold and Drag** = Duplicate
- **Ctrl + C** = Copy
- **Ctrl + V** = Paste

Centering View on Objects:

Select an object in the viewport and press **F** to center your view. Then hold Alt + RMB to rotate around the centered object or hold Alt + RMB + LMB to move in/out on that centered object.

- **F** = Center View on Selected Object
- **Hold Alt + RMB** = Rotate Around Centered Object
- **Hold Alt + RMB + LMB** = Move In/Out on Centered Object

Undo/Redo:

These are self-explanatory.

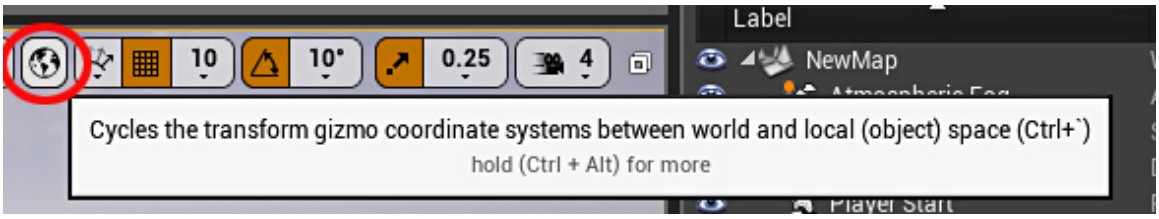
- **Ctrl + Z** = Undo
- **Ctrl + Y** = Redo

11. LOCAL AND WORLD COORDINATE SYSTEM

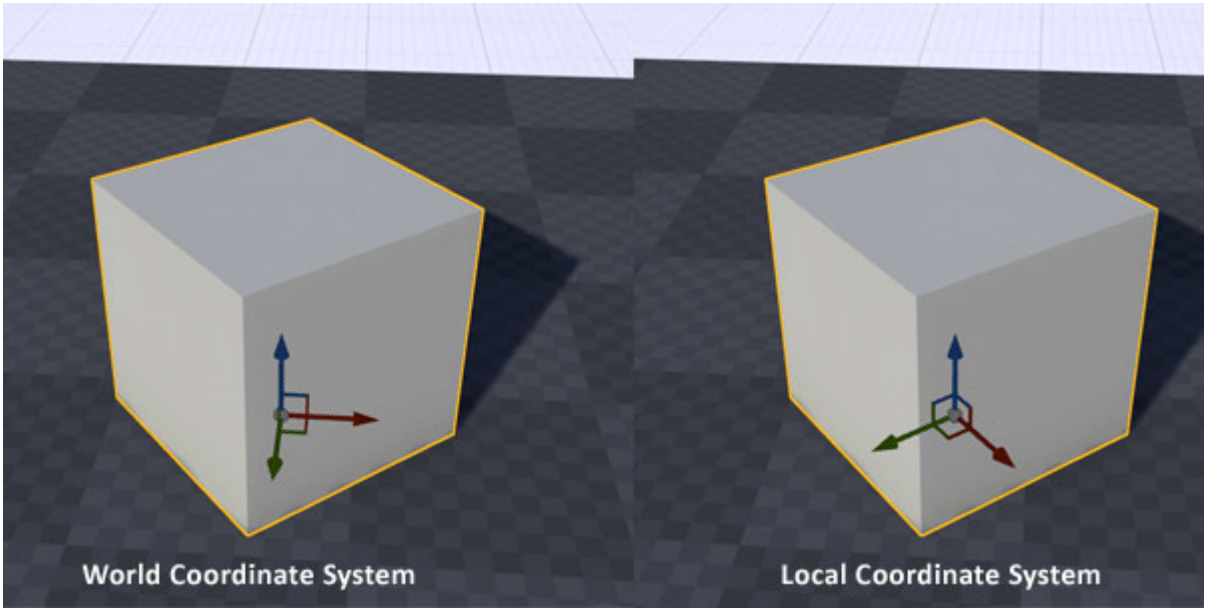
Each object in UE4 can be moved along 2 types of coordinate systems - world or local. Most of the time you will be working in **World Coordinate System** as you create your environment. But there will be time when you'll need to work with object's **Local Coordinate System**.

- **World Coordinate** applies to the world, XYZ direction does not change
- **Local Coordinate** applies to the individual object, ignores the universal XYZ position of the world

Click on the World/Local icon within the viewport to cycle World vs Local:



- **Ctrl + ~** = Switch Between Local and World Coordinate System



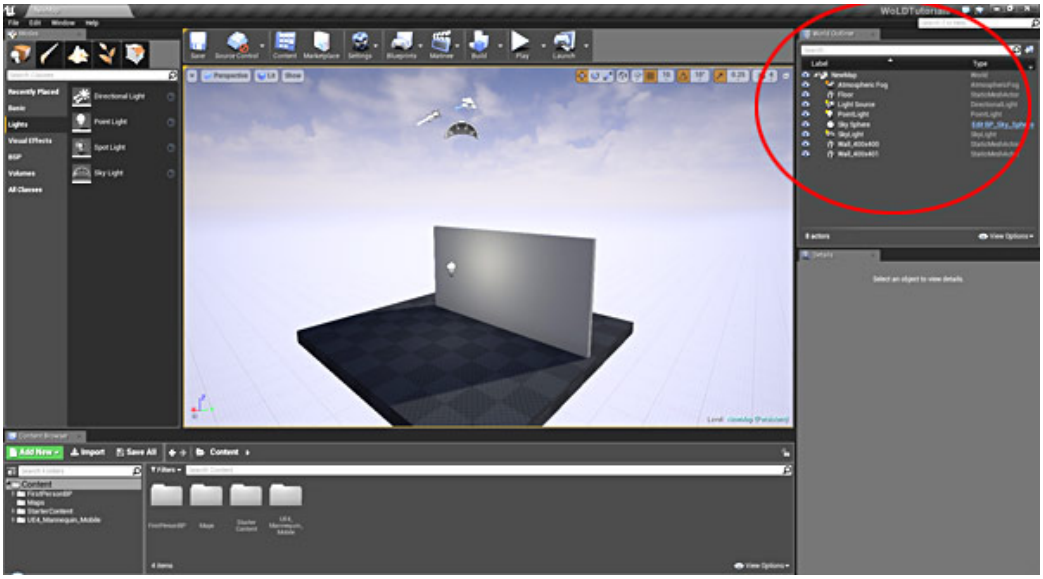
To see how this works follow these steps:

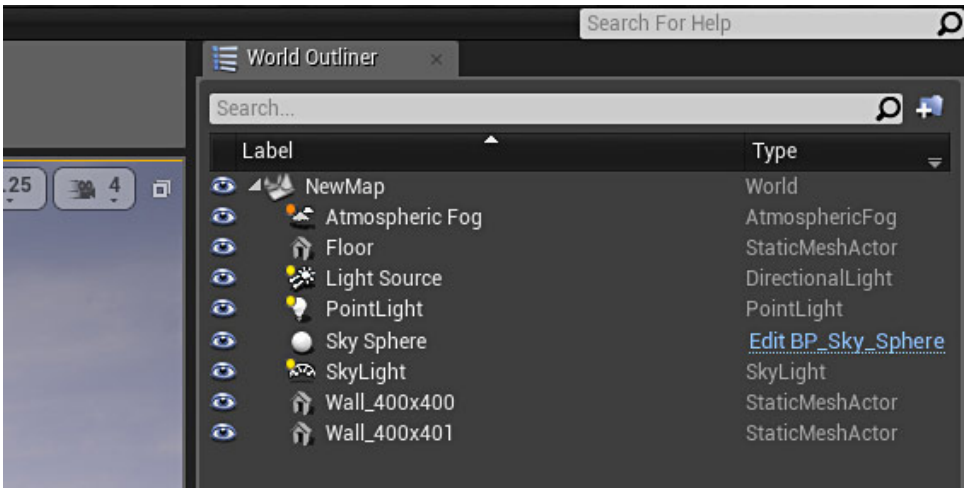
- Select any object inside the viewport (BSP or Static Mesh)
- Rotate it 45 degrees on Z axis
- Switch to Move tool (W)
- Move the box on X and on Y
- Switch from World to Local coordinate system (Ctrl + ~)
- Move the box again on X and on Y
- Notice the difference of how you are able to work along the World or Local coordinate system
- Switch back to World Coordinate System (Ctrl + ~)

12. WORLD OUTLINER

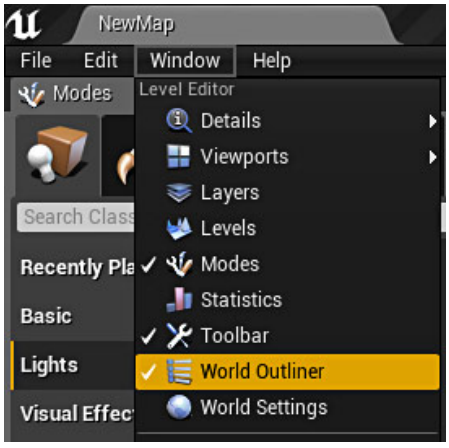
World Outliner will list all actors/objects within your level.

It is located on upper right hand side of the editor:



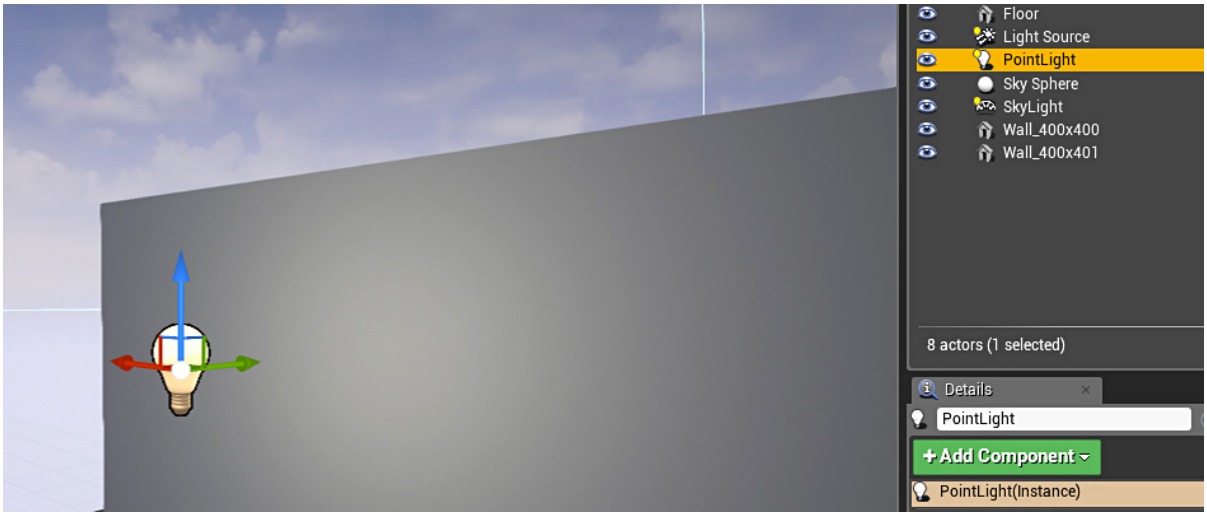


If you do not see it, go to **Window → World Outliner**:

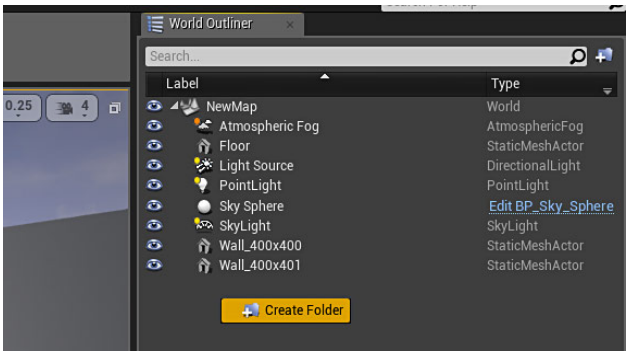


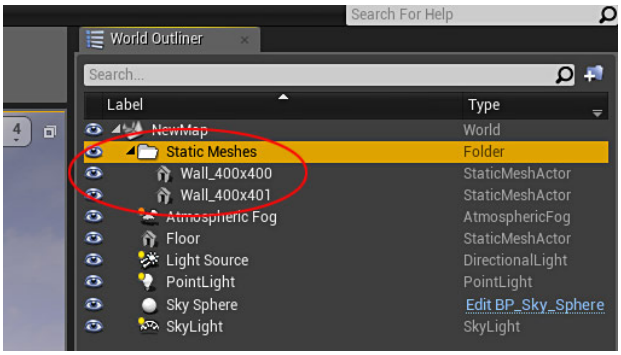
Using the World Outliner, you can select any actor in the level or search for one:

You can double click on any actor from World Outliner to center your view on the object:

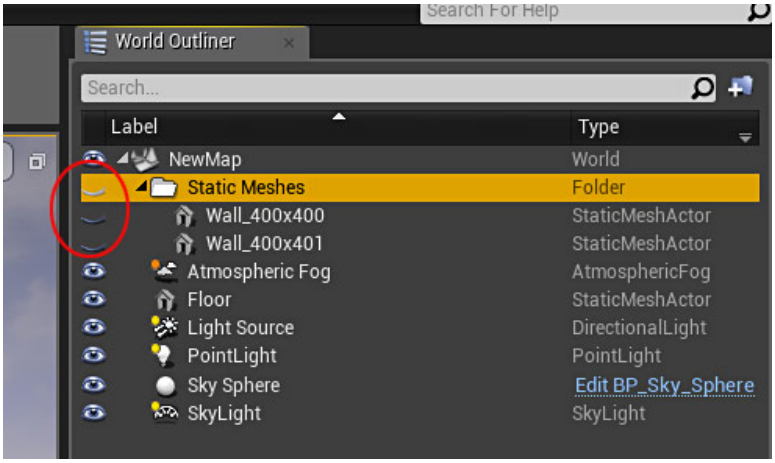


And, you can organize your scene better by using folders:



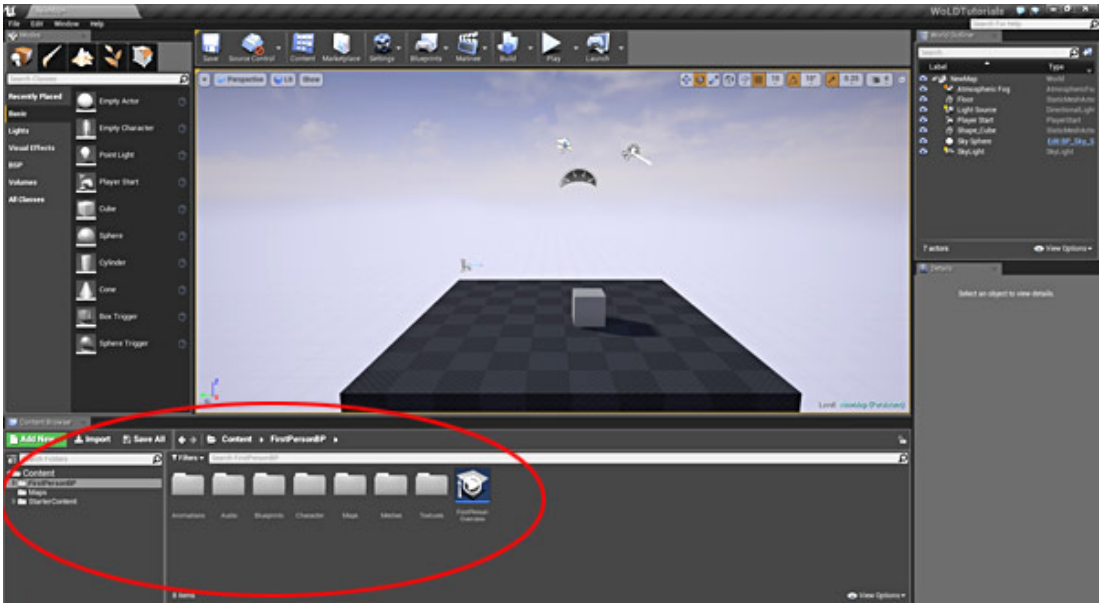


You can also enable/disable visibility of individual objects or folders:



13. CONTENT BROWSER

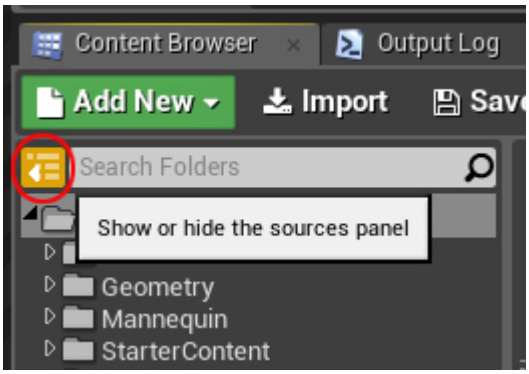
Content Browser is the content management system in UE4:



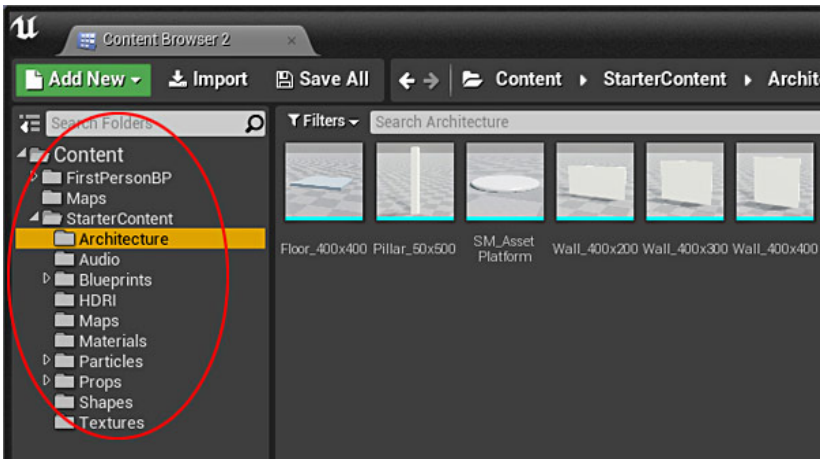
Through the Content Browser you will:

- You will look through available textures, materials, Static Meshes, animations, Blueprints, audio or particle effects to insert into your level
- You will import custom static meshes and textures
- You will create new blueprints, materials, particle effects, etc.
- You will organize, move and rename assets for your projects

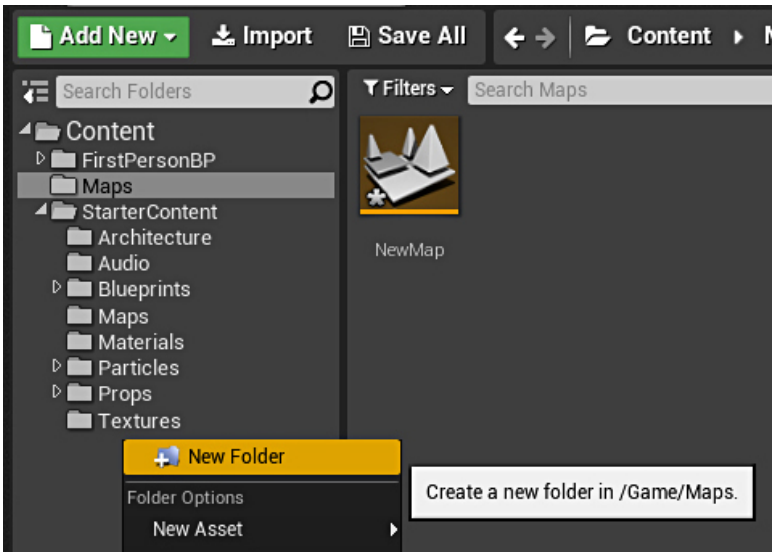
Click on **Extended View** icon to **Open Source Panel**. It makes it easier to select and browse through available folders for your project:



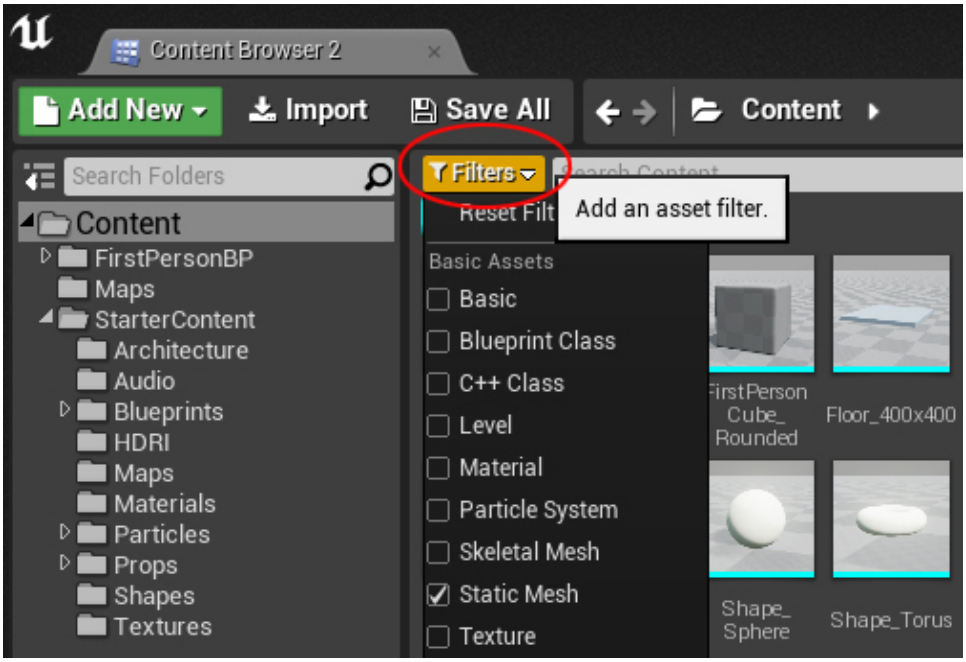
Use the folder structure to look into any folder and find the assets that are available to use in your levels:



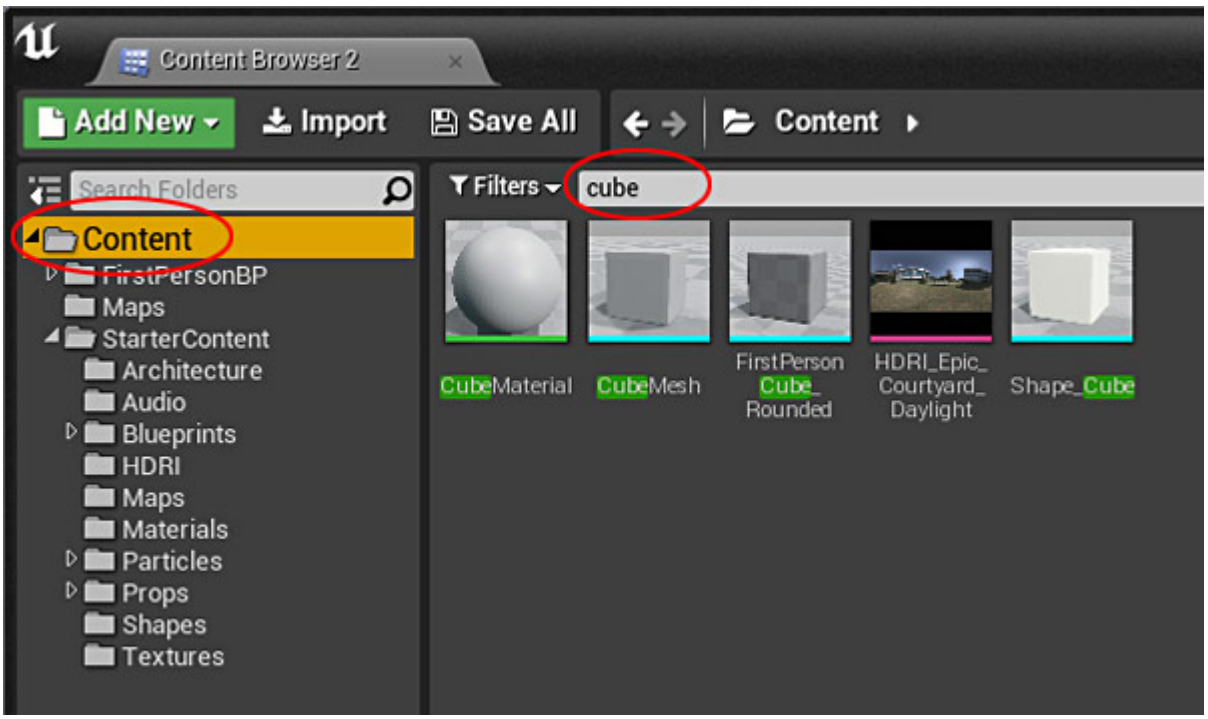
You can create new folders to better organize your assets:



Look for specific asset type by using the **Filters** drop down menu:

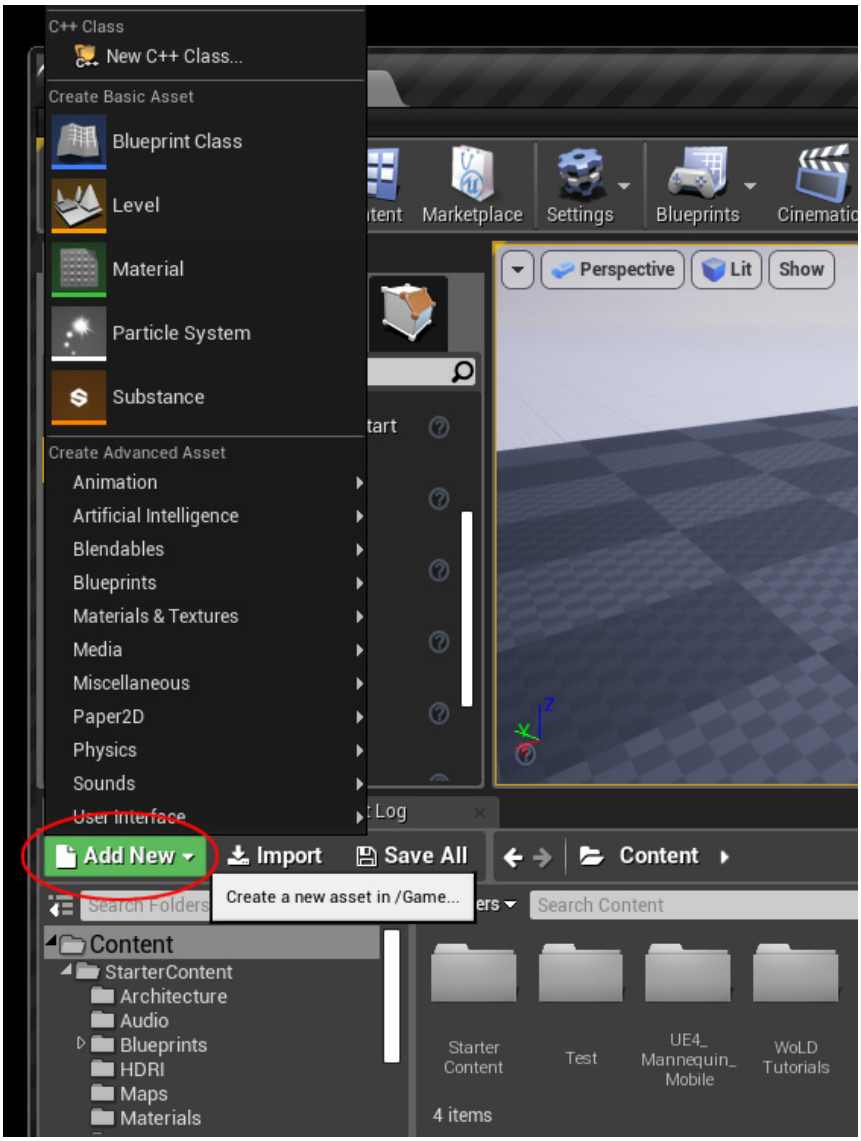


Use the **Search** to look for a name of the asset:

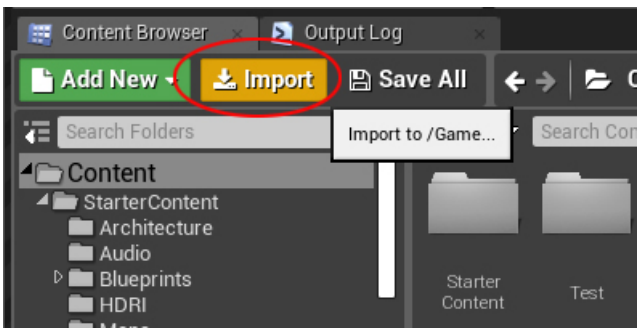


To search and to filter the entire directory of your project, make sure to select the **Content** folder, this way you are searching everything within the Content folder.

Click on **Add New**, you will be able to create new asset types such Material, Particle System or Blueprint Class:

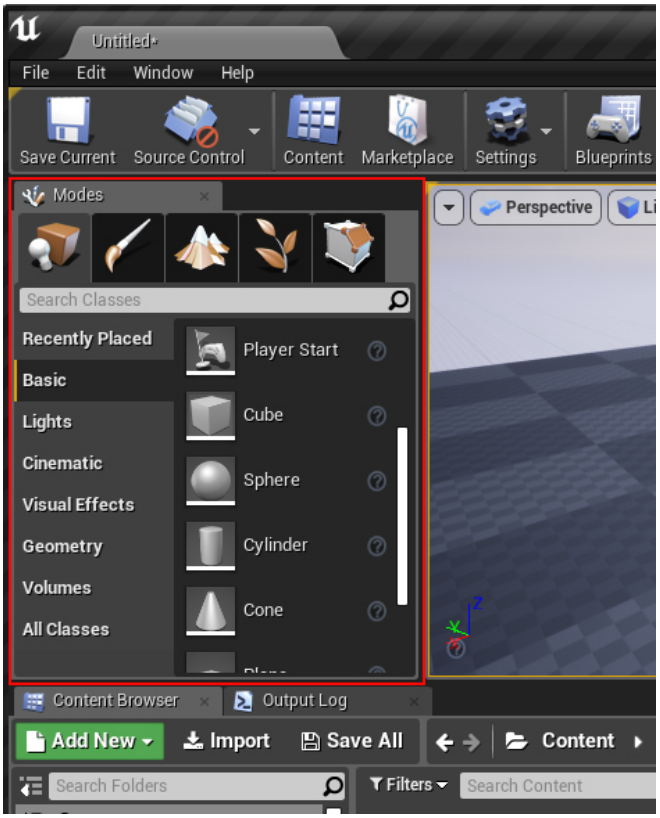


And if you click **Import**, you can import assets you created in an external software such as textures or Static Meshes:



14. MODES PANEL

Modes Panel offers various editor entities that can be inserted into your level such as lights, volumes or Geometry (BSP), as well as editing modes such as Landscape, Foliage, Vertex Painting and Geometry (BSP) Editing.



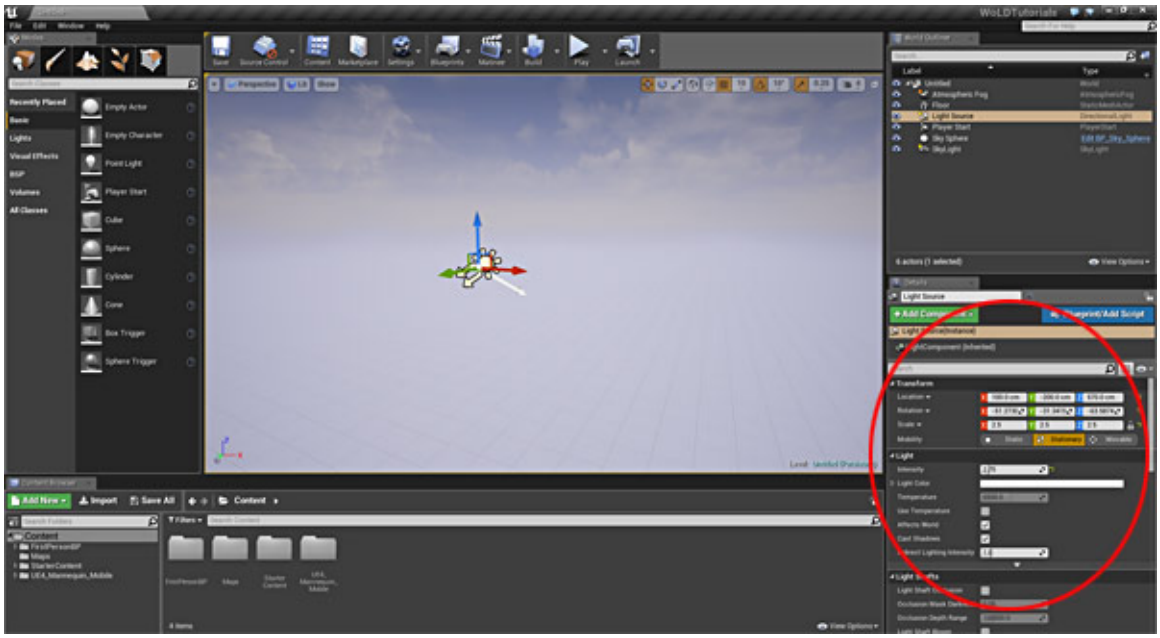
- **Shift + 1** = Place (level entities)
- **Shift + 2** = Paint (vertex painting)
- **Shift + 3** = Landscape (terrain system)
- **Shift + 4** = Foliage (mesh painting)
- **Shift + 5** = Geometry Editing (BSP brush editing)

You will use Models Panel often.

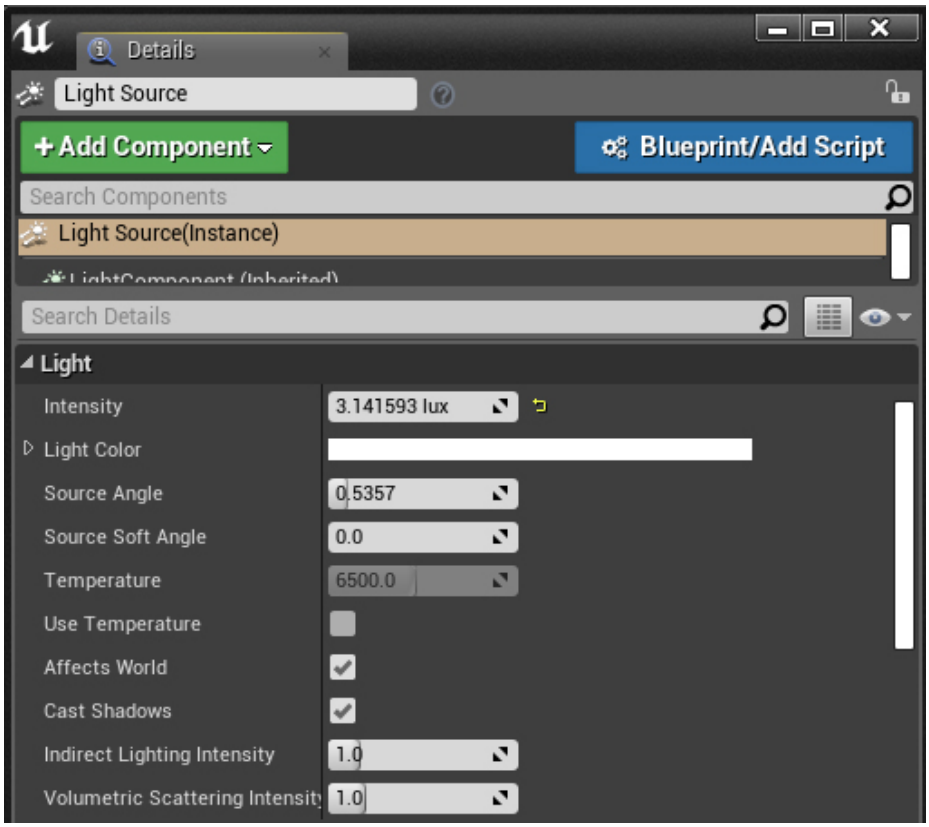
15. DETAILS PANEL / OBJECT PROPERTIES

Each object/actor placed in the level will contain a set of properties. These settings will **ONLY** apply to objects inserted into the level.

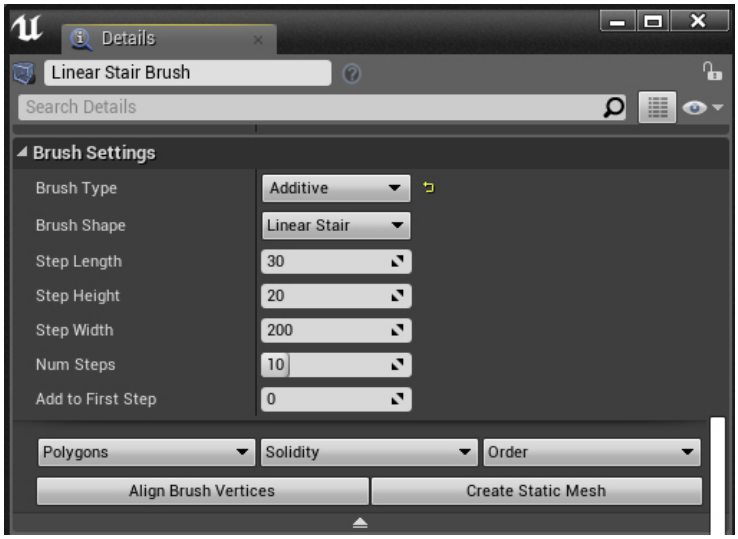
Select an object inside your level and on the bottom right of the editor you'll see Details panel. These are selected object's properties.



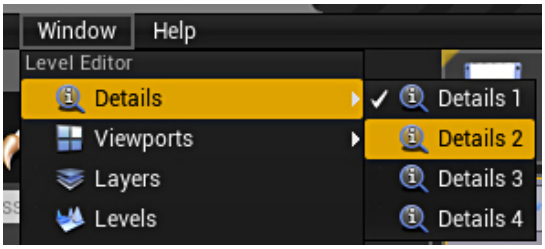
Depending on the type of actor you select, settings will be vary**.** For example if you select a Light, you can change color, light intensity and radius:



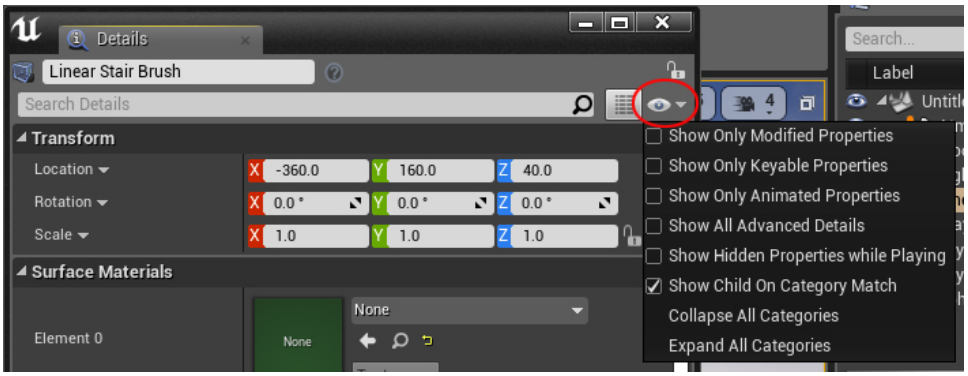
But if you select a BSP brush, you can change its dimension:



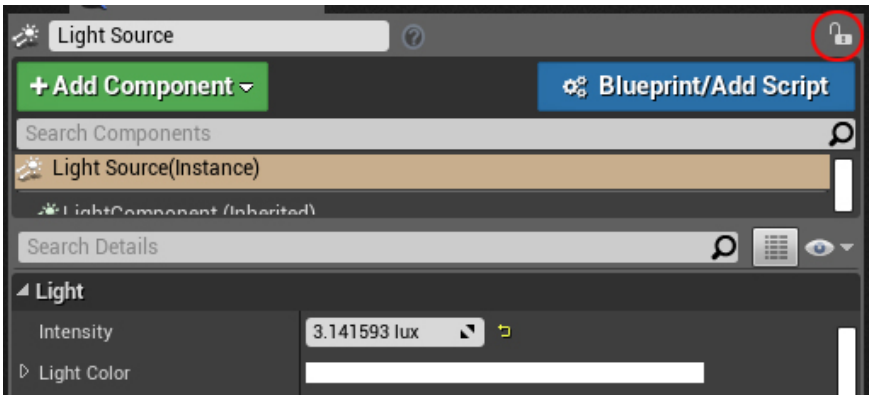
You can open additional Details panels by going to **Window → Details → Details 2, 3 or 4**:



Each Detail panel will have an option to enable additional settings such as **All Advanced Details** or **Only Modified Properties**. Click on this icon for drop down menu:



The content of each Detail window will change depending on the object you select, but you can lock this window to always show selected actor's settings by enabling this lock icon at the top right of the panel:

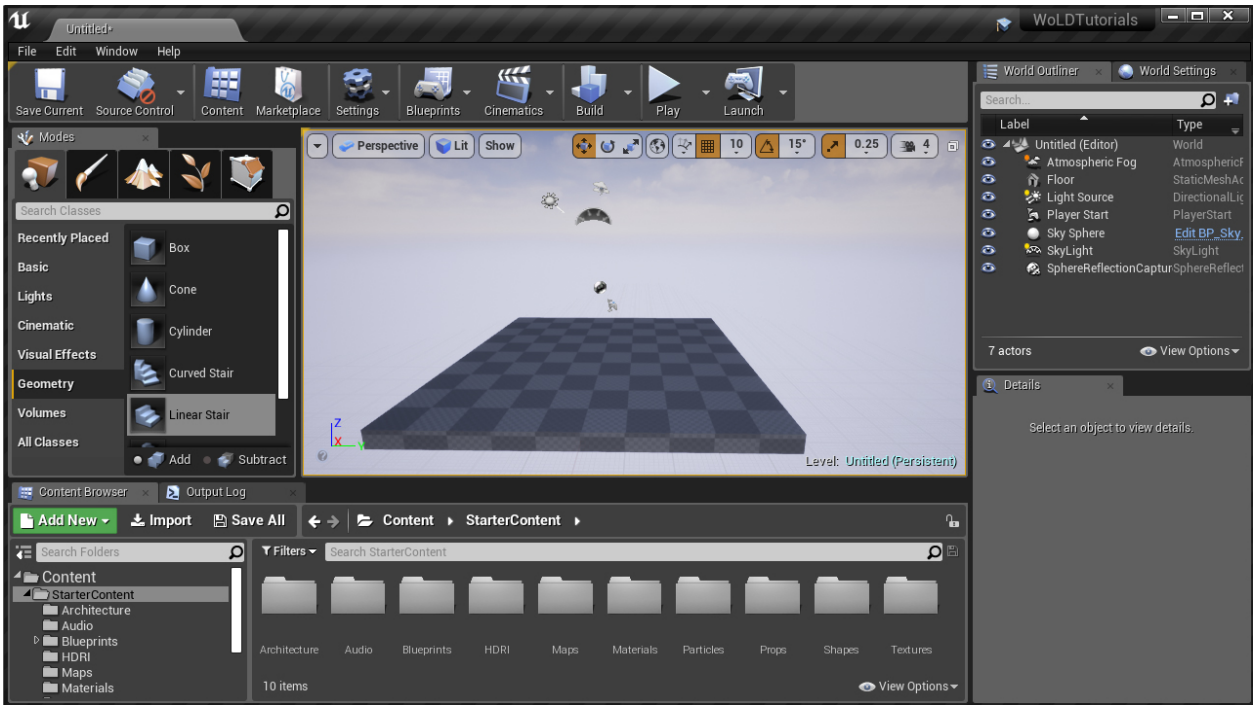


Disable the lock when you want to select and view other actor's properties.

16. 20 MOST COMMONLY USED EDITORS & TOOLS WITHIN UE4

UE4 editor itself can be looked at as a collection of different editors within it.

Level Editor or the world editor is where you create your environments in. You will insert BSP geometry, Static Meshes, lights, integrate gameplay and use other necessary level actors to complete the environment.

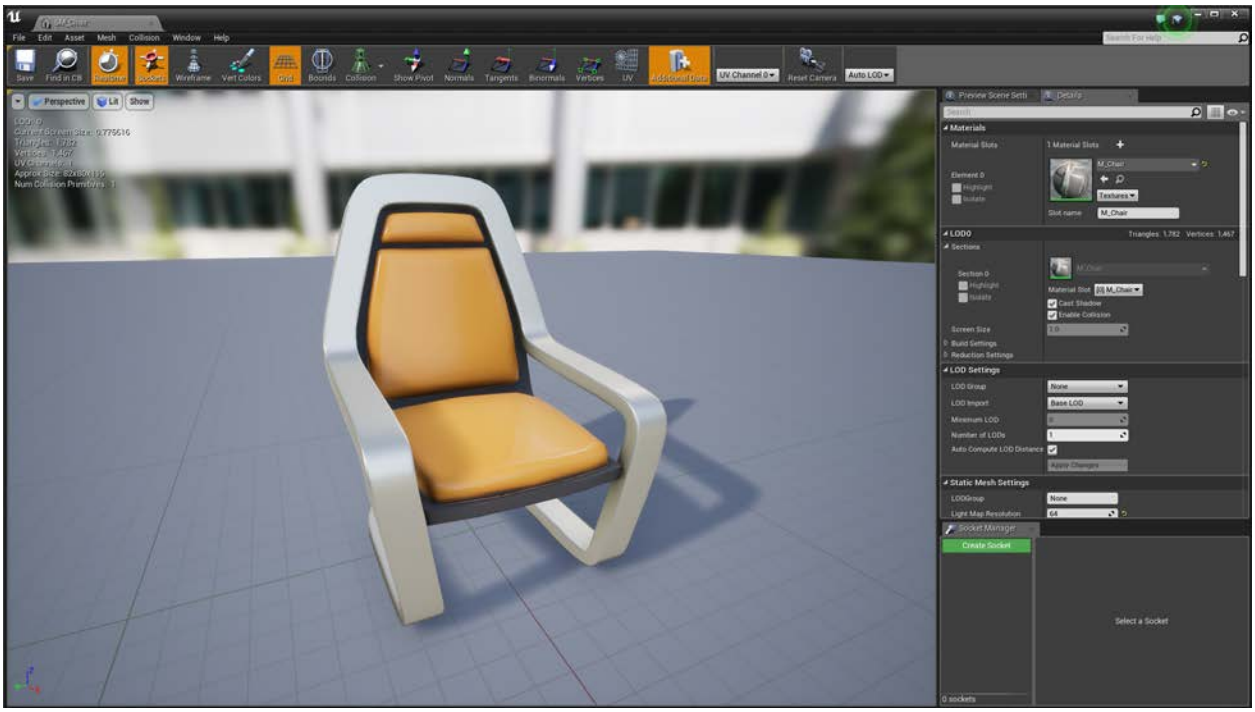


But then within UE4 you have a subset of asset editors, such as:

- Static Mesh Editor
- Material Editor
- Texture Editor
- Blueprint Editor
- Cascade Editor (Particles)
- Sequencer Editor (Movie/Cinematics)
- Matinee Editor
- Persona Editor (Animations)
- UMG UI Editor (User Interface)
- Behavior Tree Editor (AI)
- Sound Cue Editor
- Paper2D Sprite and Flipbook Editor
- Physics Asset Tool Editor
- Media Player Editor

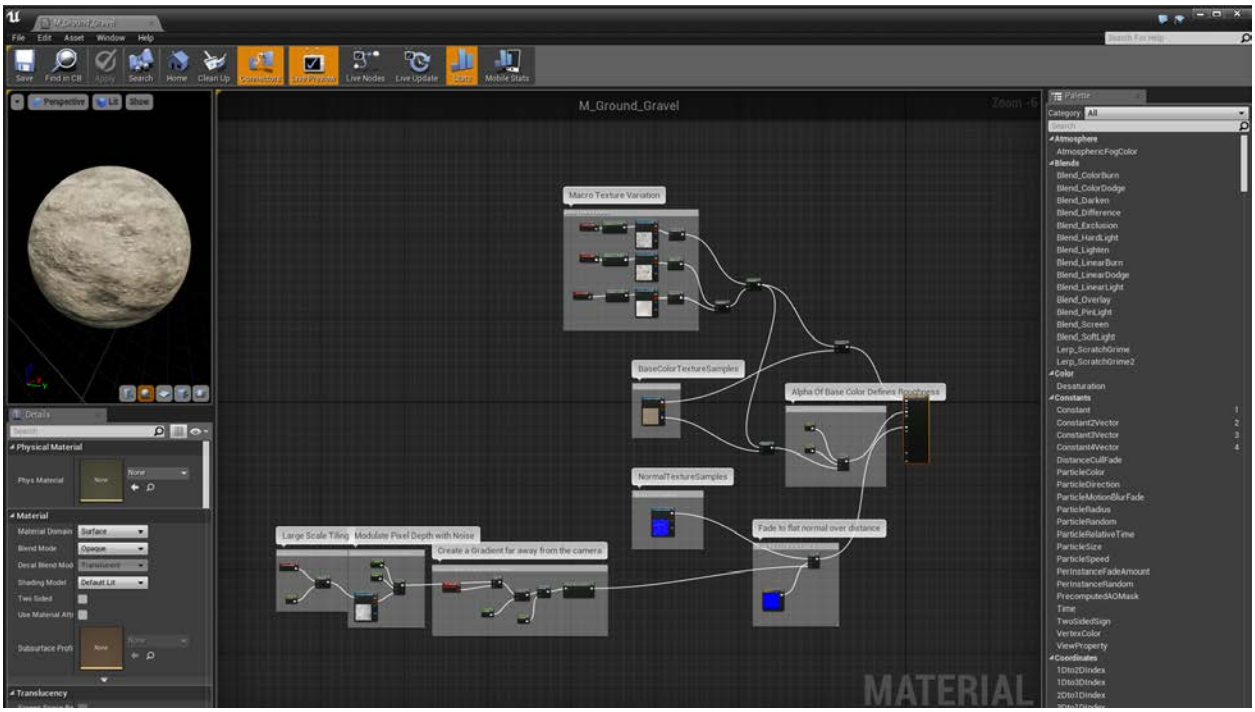
More about each editor and how to use them in-detail can be found here: https://docs.unrealengine.com/en-us/GettingStarted/SubEditors

Static Mesh Editor is used to preview and change properties of imported 3d models such as materials, rendering, lightmaps, collision, UVs and LODs.

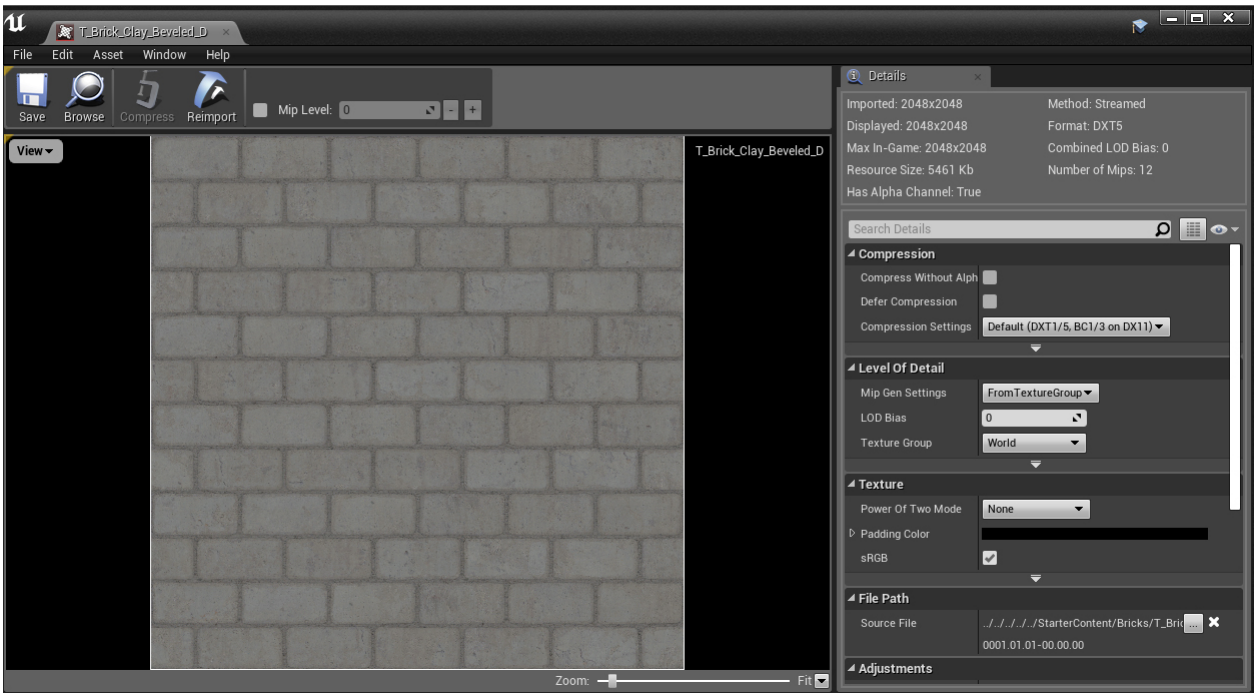


Material Editor is where you create and edit properties of how a material looks and functions. Materials are what you apply onto Static Meshes, BSP

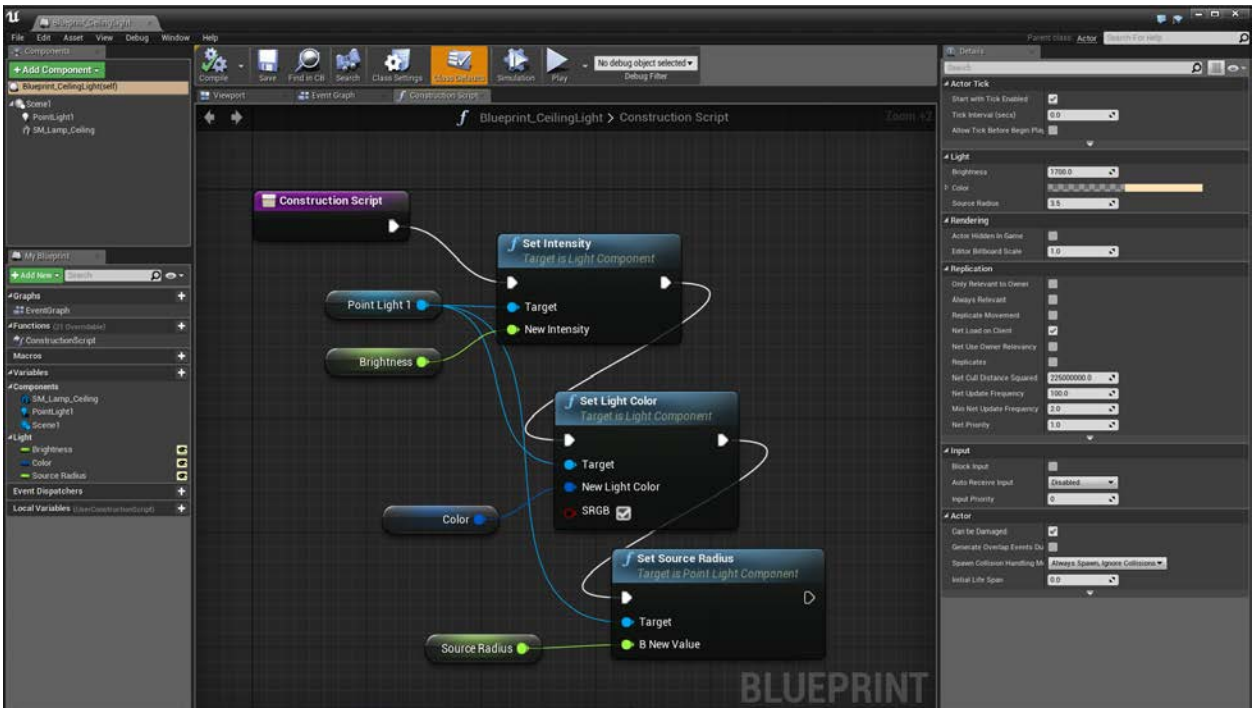
brushes and landscape to create what they look like. Think of Materials as the textures you see on the objects in the level. Although know that Materials and Textures are technically different from each other.



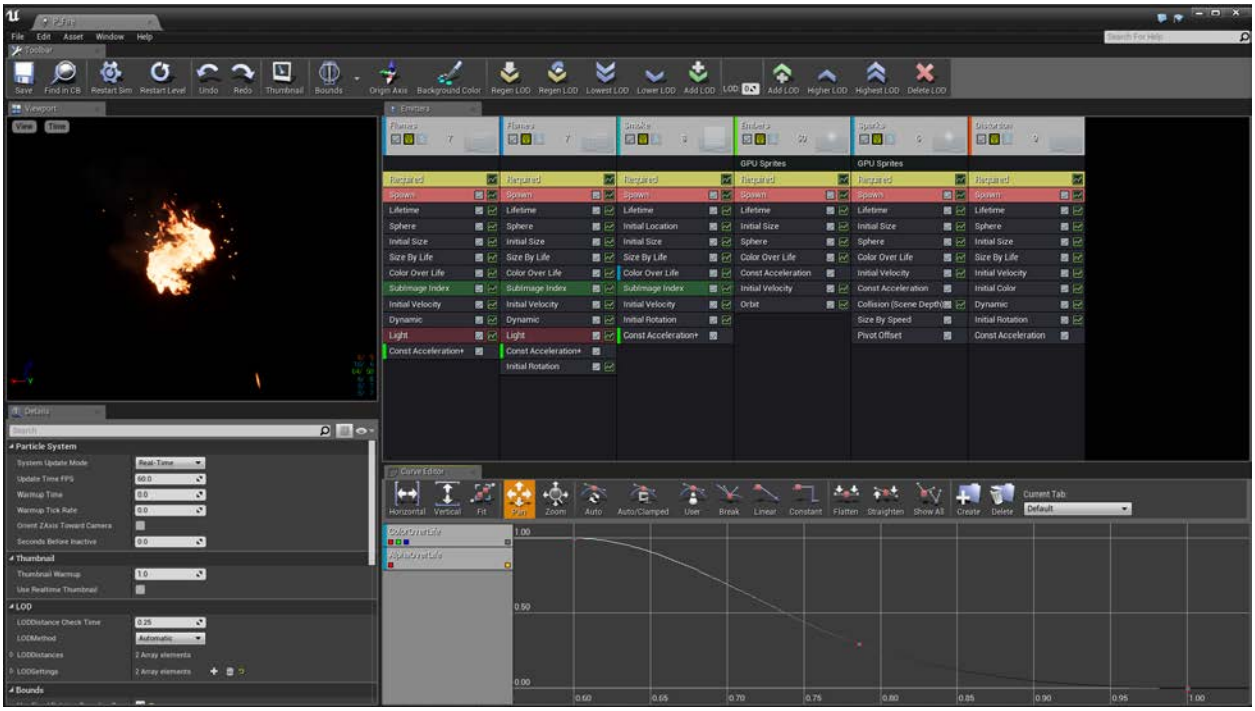
Texture Editor is where you edit the properties of UE4 textures.



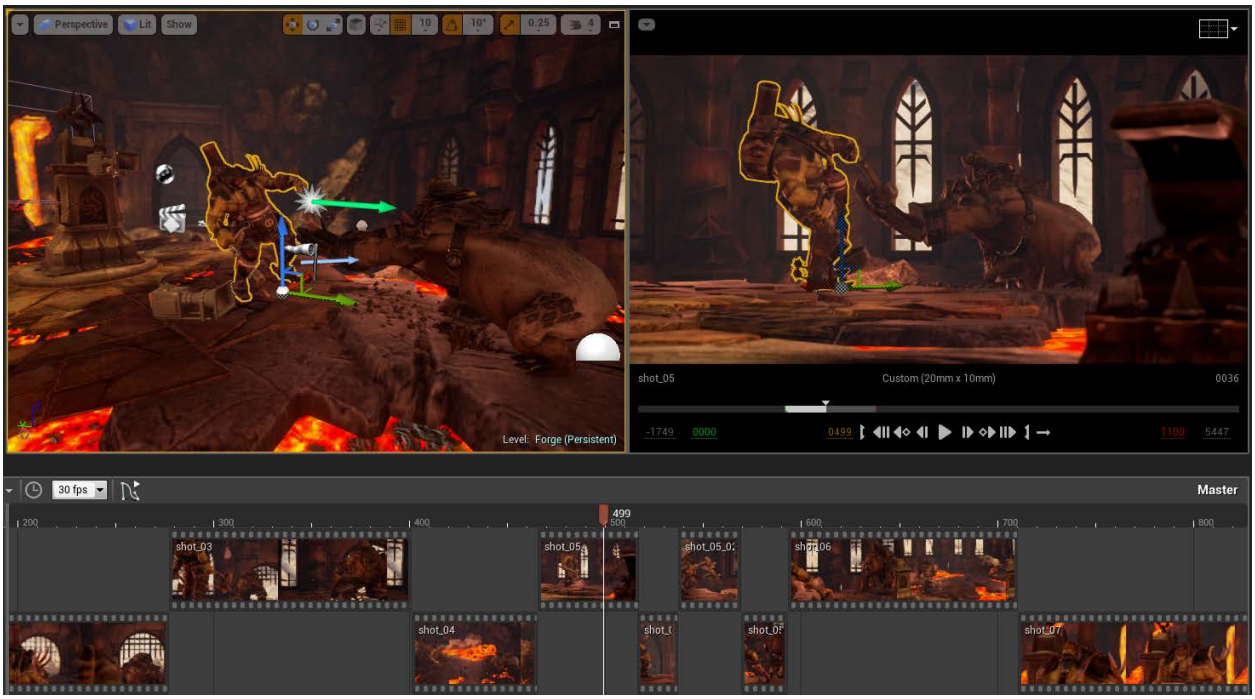
Blueprint Editor is where you edit and work with Blueprints. Blueprint is the visual scripting language in UE4 that can be used right inside the editor without any C++ programming knowledge.



Cascade is a particle editor where you create and edit particle effects to be used in your levels.

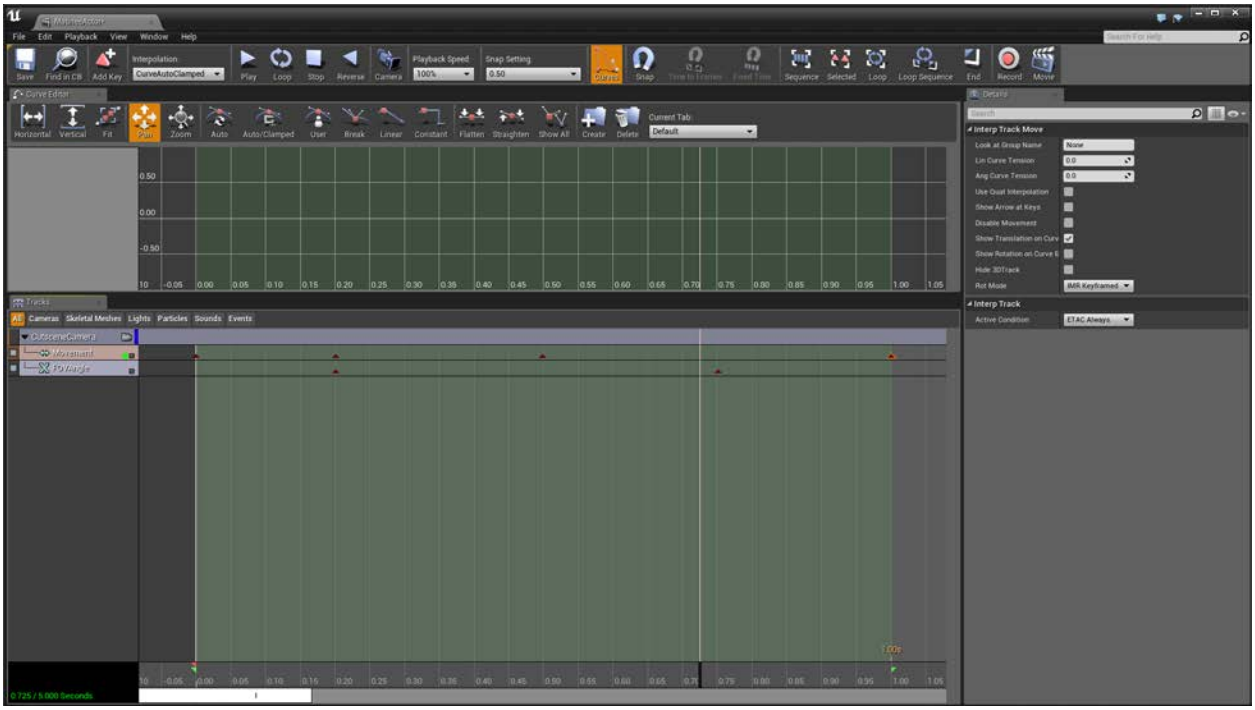


Sequencer Editor is used for creating and editing in-game cinematics. Sequencer is replacing Matinee Editor.

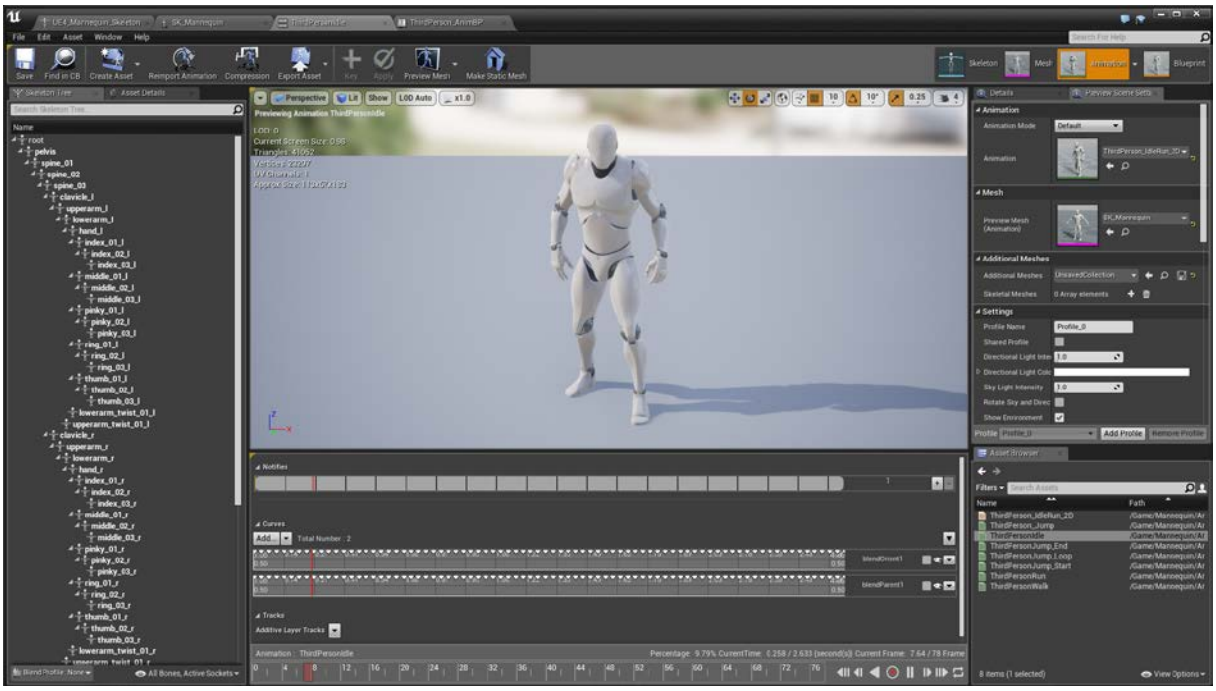


Example of Sequence Editor with Epic Games content

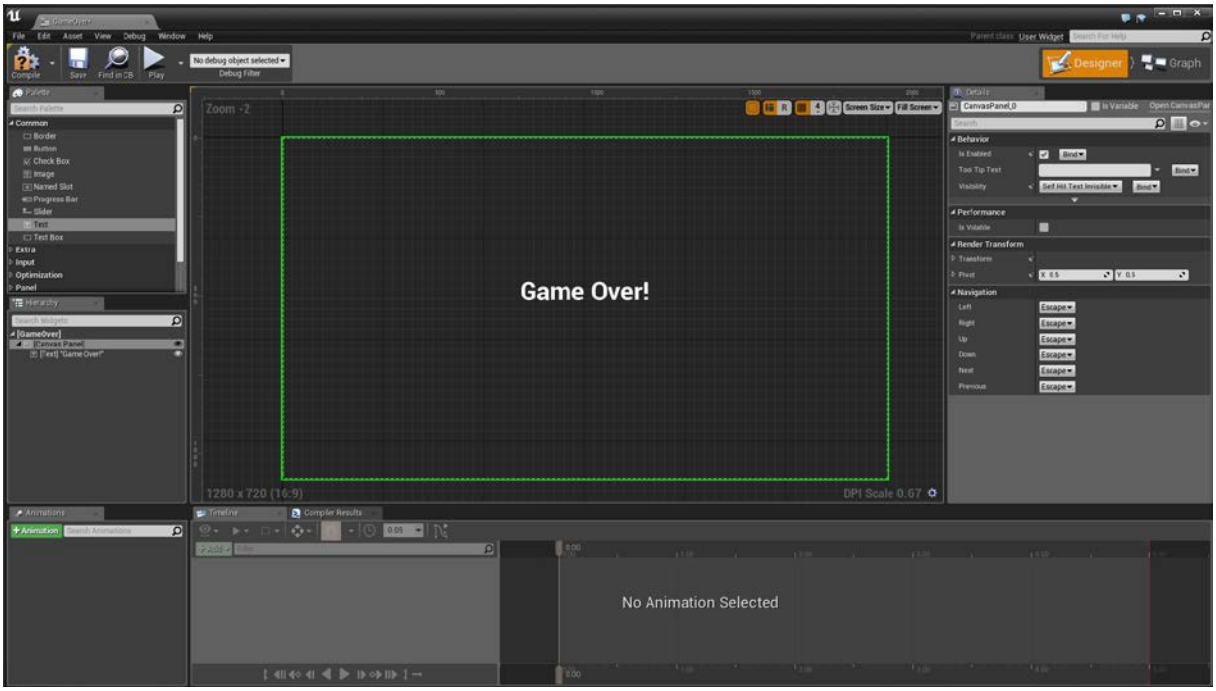
Matinee Editor was used for creating and editing in-game cinematics. It has now been replaced with Sequencer Editor.



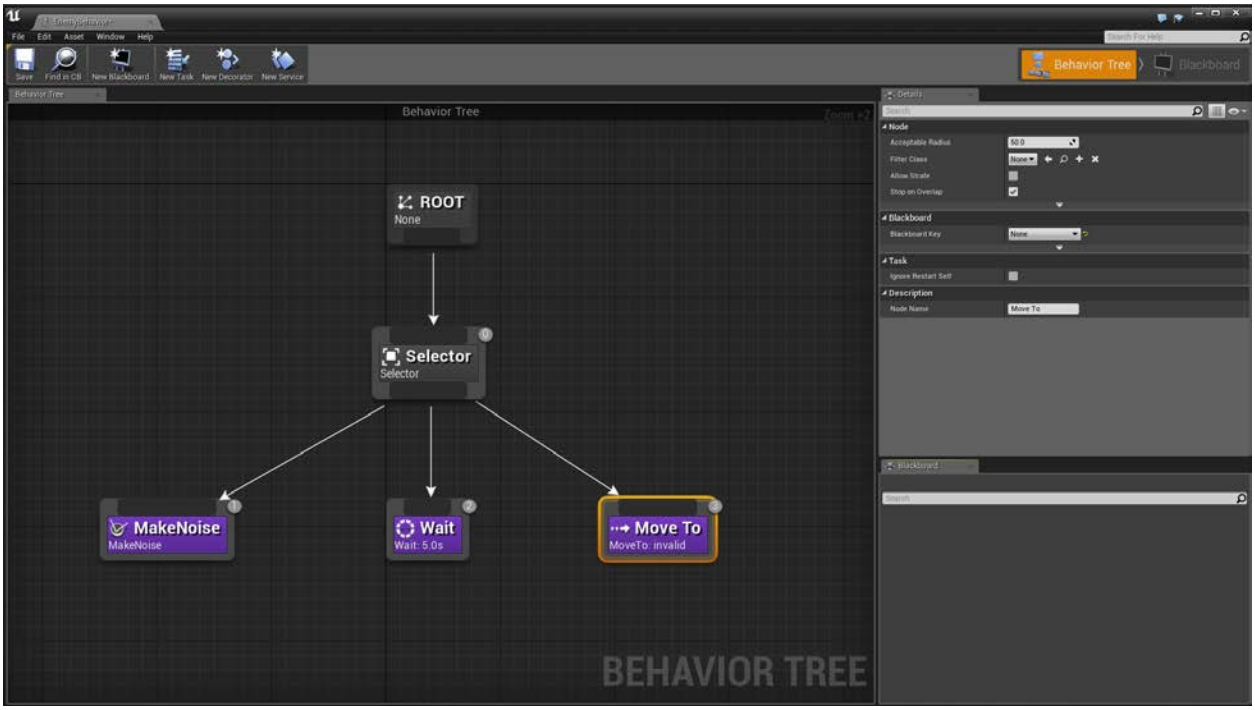
Persona Editor is used for animation editing in UE4 such as editing skeletal meshes, skeletal assets and animation blueprints. Almost all animating editing work in UE4 is done through this editor.



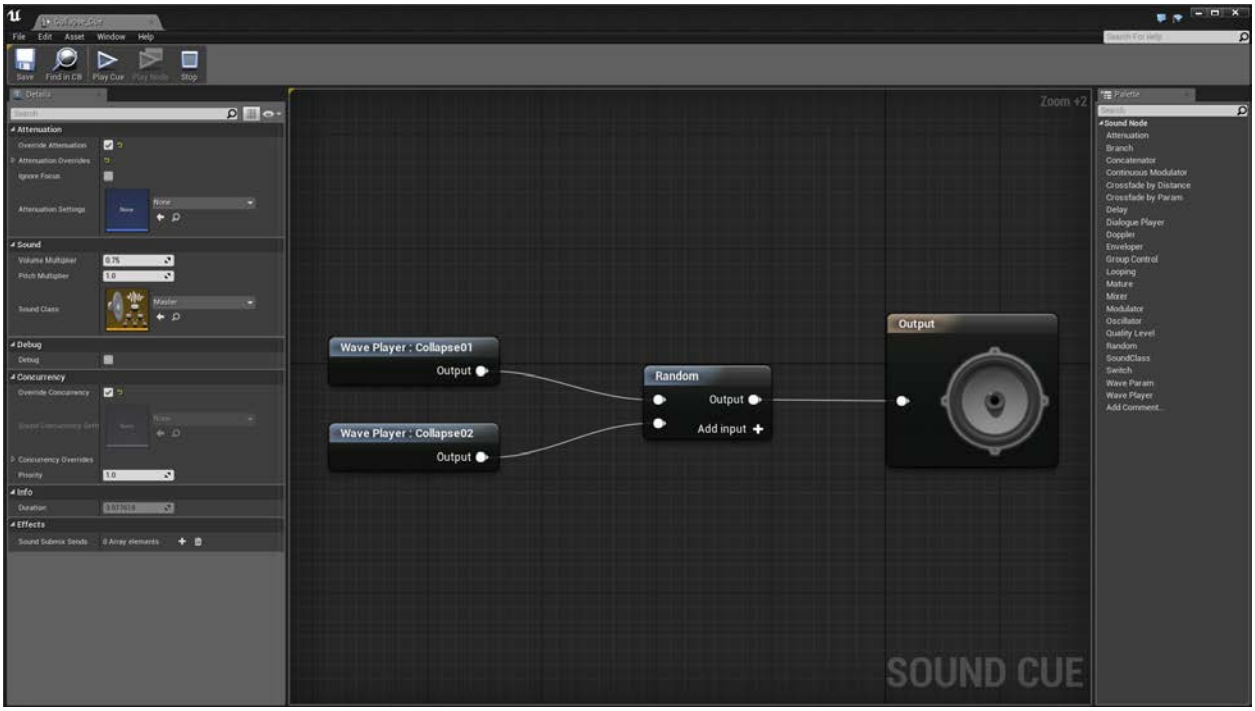
UMG UI Editor is UI (user interface) editor where you create and edit menus and graphics for what the player/user will see and interact with in your game or application.



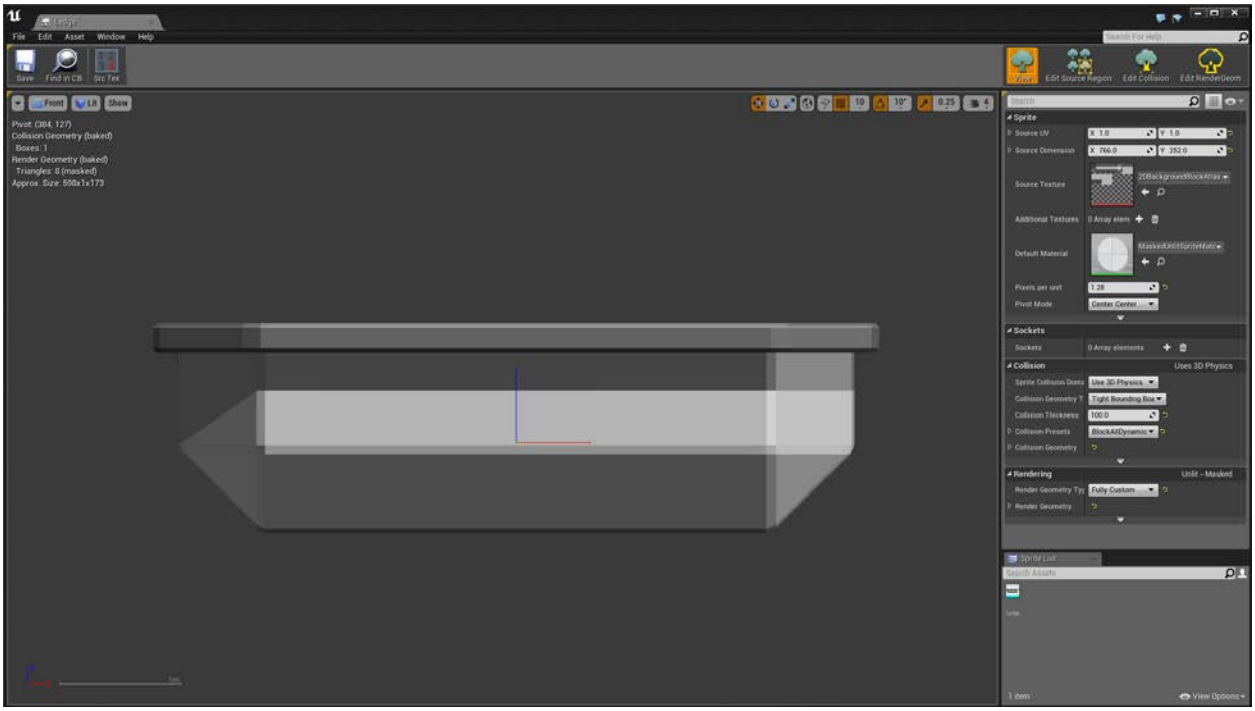
Behavior Tree Editor (AI) is where you edit and script AI behaviors.



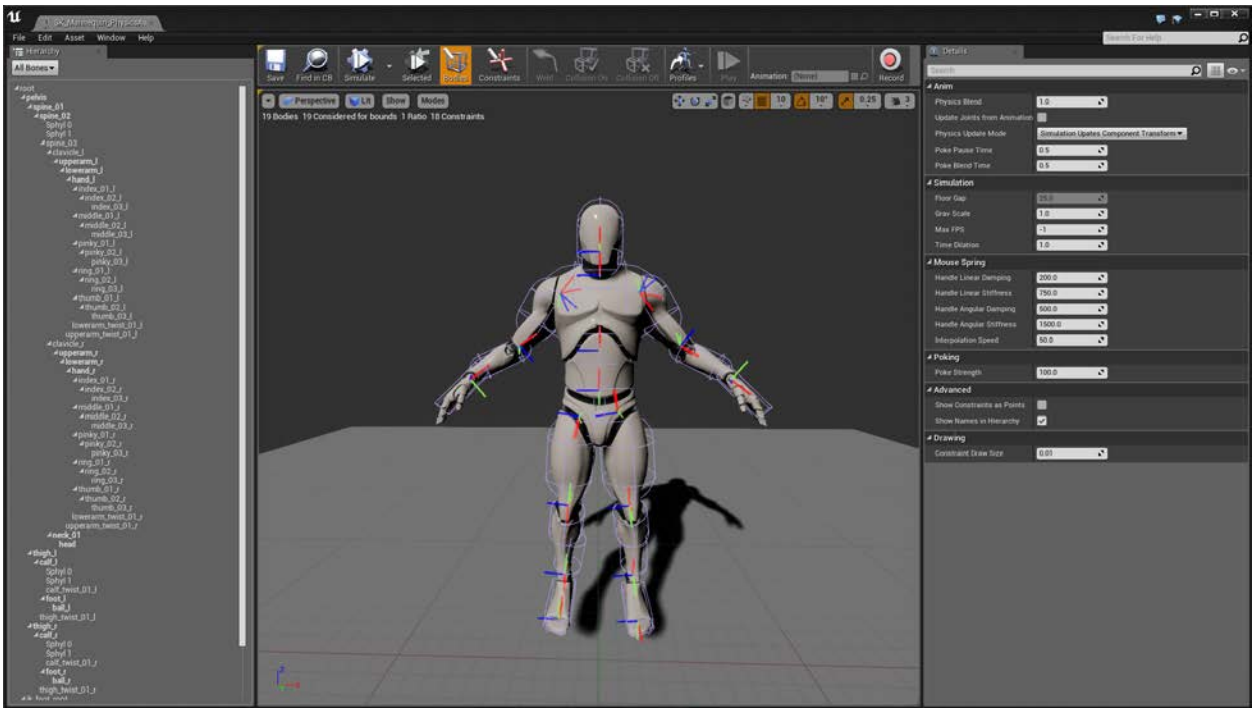
Sound Cue Editor is the sound editor and mixer in UE4; it's where you edit all your audio to be used in your environments.



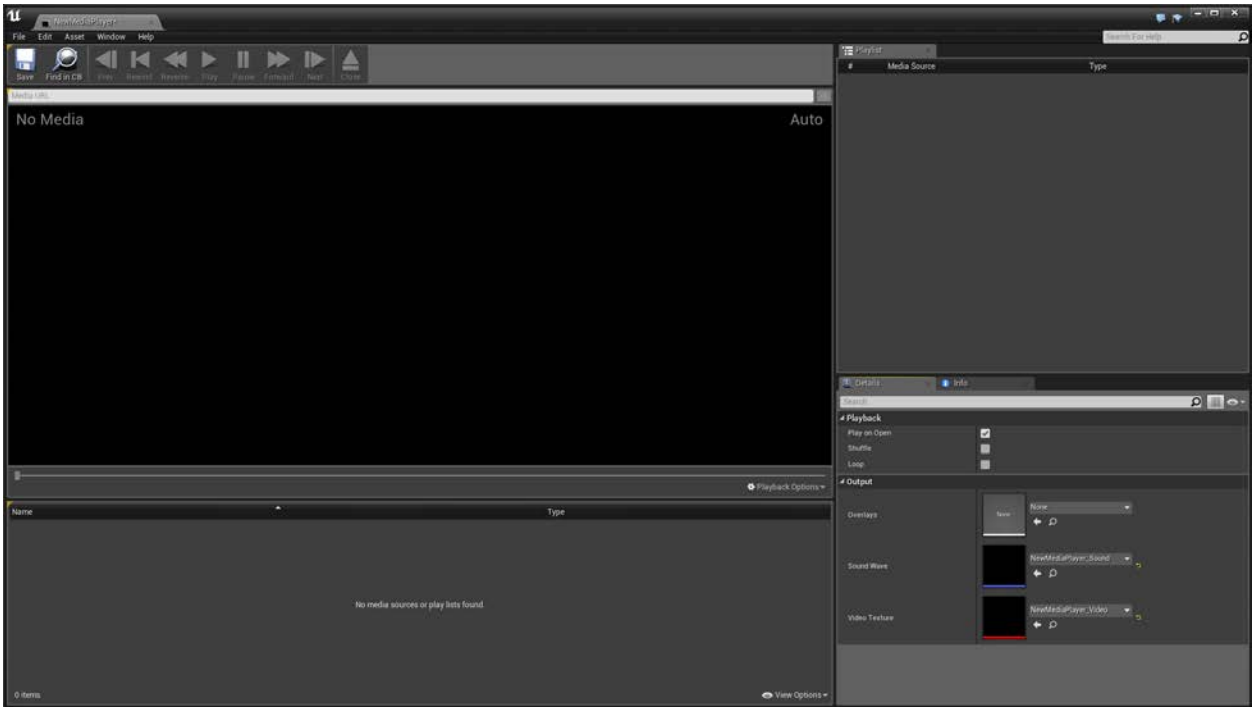
Paper2D Sprite and Flipbook Editor are two editors that allow you to create and edit 2d images to be used for sprites and 2d animations.



Physics Asset Tool Editor is used to create physics, from basic set of physic bodies and constraints to complete ragdolls.

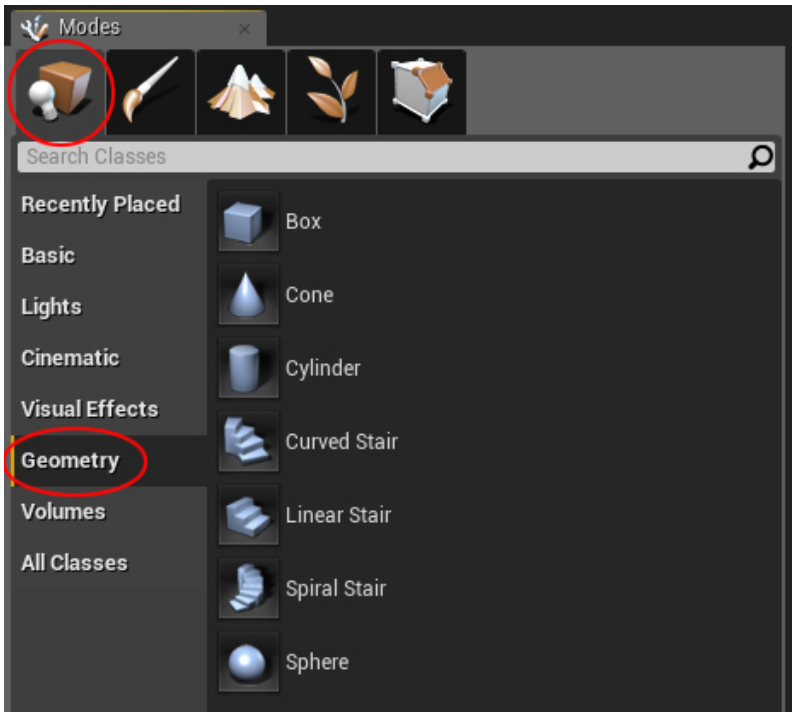


Media Player Editor which allows you to define imported media files to be used for playback in UE4.



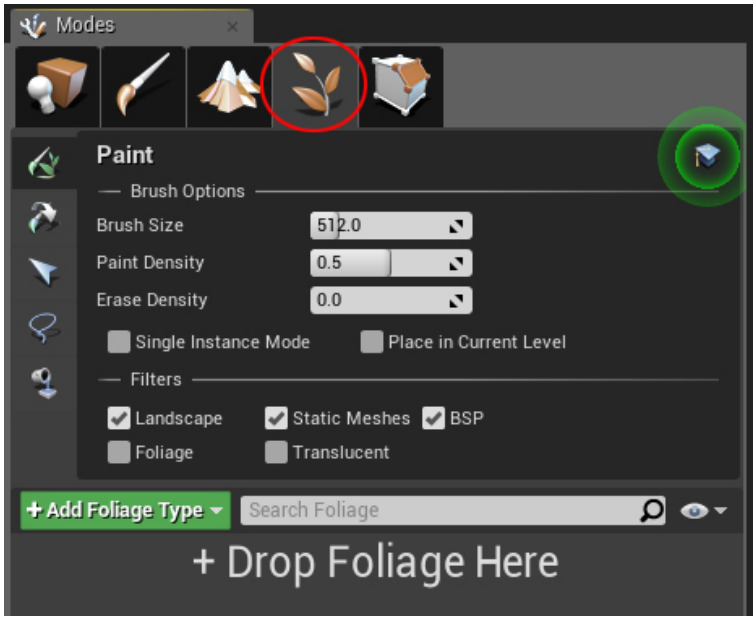
Then you have additional Tools available through the **Modes Panel**.

Geometry Tool is where you create BSP brushes:

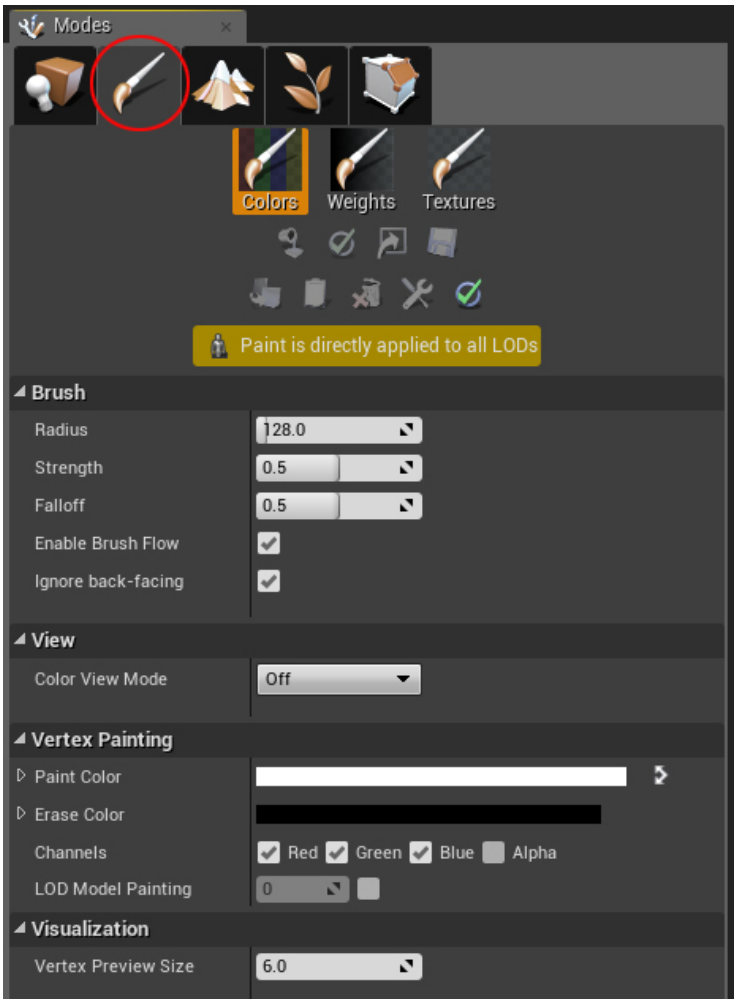


Landscape Tool is for creating and editing terrain:

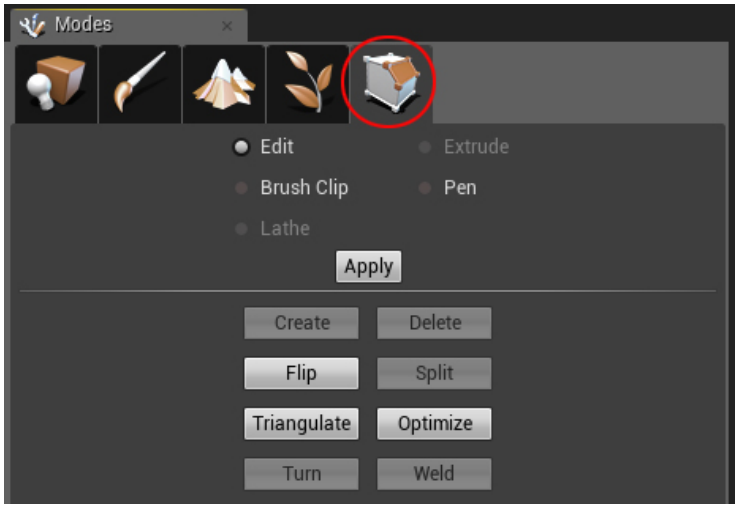
Foliage Tool is for painting/placing meshes on existing level geometry:



Vertex Painting Tool is for painting and modifying color and blending materials on Static Meshes within your level:



Geometry Editing Tool is for modifying and editing placed BSP brushes:

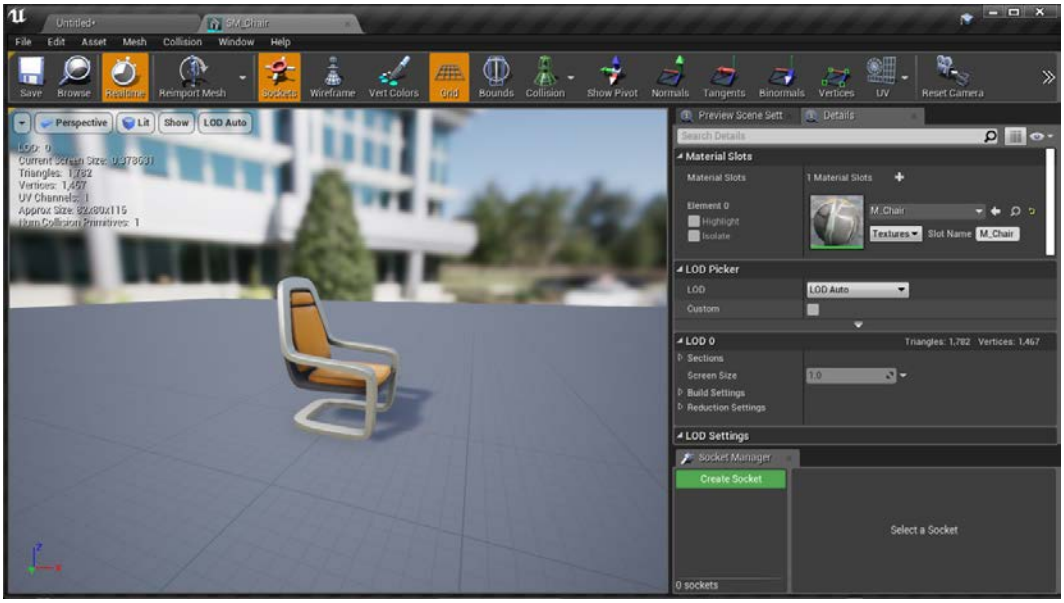


17. GLOBAL VS LOCAL STATIC MESH SETTINGS

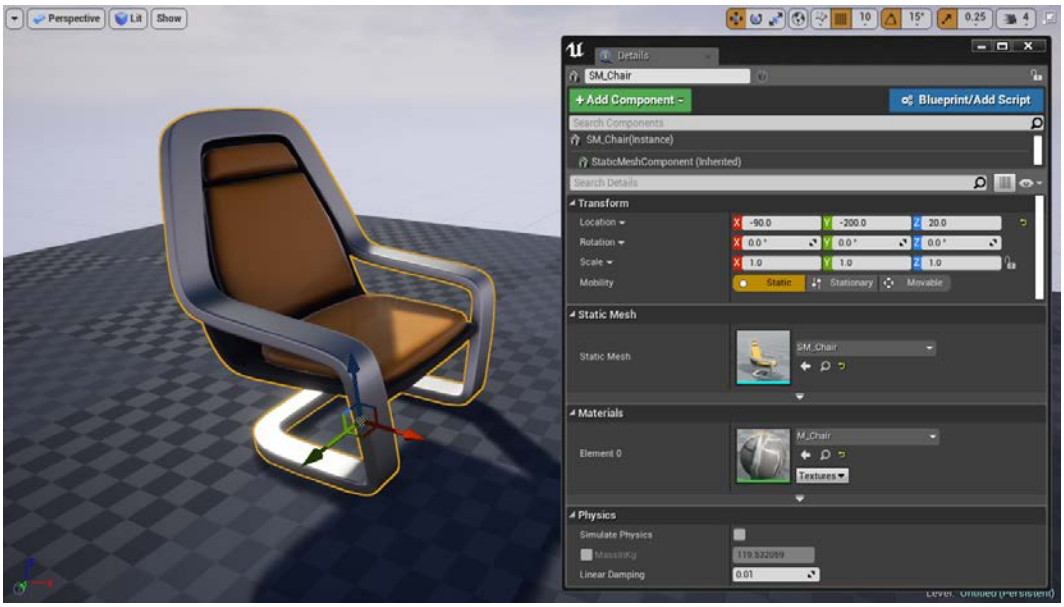
*"When you do change **Static Mesh Editor settings** and when do you change **Static Mesh settings through the Details panel**?"*

There are two panels available to modify Static Mesh properties.

One is through Static Mesh Editor (Global):



Another is through Details panel (Local):



- **Static Mesh Editor is the global or universal setting**. It applies to every instance of the object placed in all of your maps or any object that will be placed.
- **Detail panel is the local setting**. Applies ONLY to the selected object inside your level. Overrides global settings.

Let me give you a basic workflow of using Static Mesh Editor and Static Mesh Details panel:

- If you want to update a setting to be applied to every instance of a Static Mesh across every single level that uses that Static Mesh - use Static Mesh Editor.
- When you import a Static mesh into UE4 for the first time you will use Static Mesh Editor to adjust all initial properties or its global/universal settings.
- If you want to change a specific properties for one Static Mesh placed inside your level then select that Static Mesh in the level and update properties through the Details panel. These will override the settings in the Static Mesh editor.

---

18. STAYING SNAPPED ON THE GRID

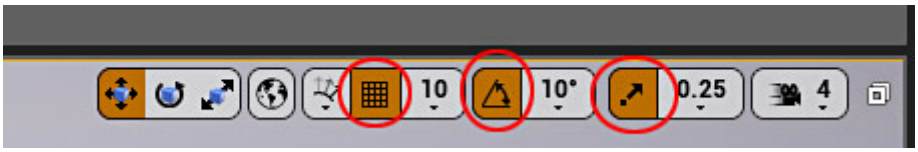
Grid snapping is one of the most important aspects of constructing environments. You want every BSP brush and Static Mesh you place inside your level to snap together like a set of Legos.

IMPORTANT!
Keep Everything You Work On In Your Level <u>On The Grid</u>

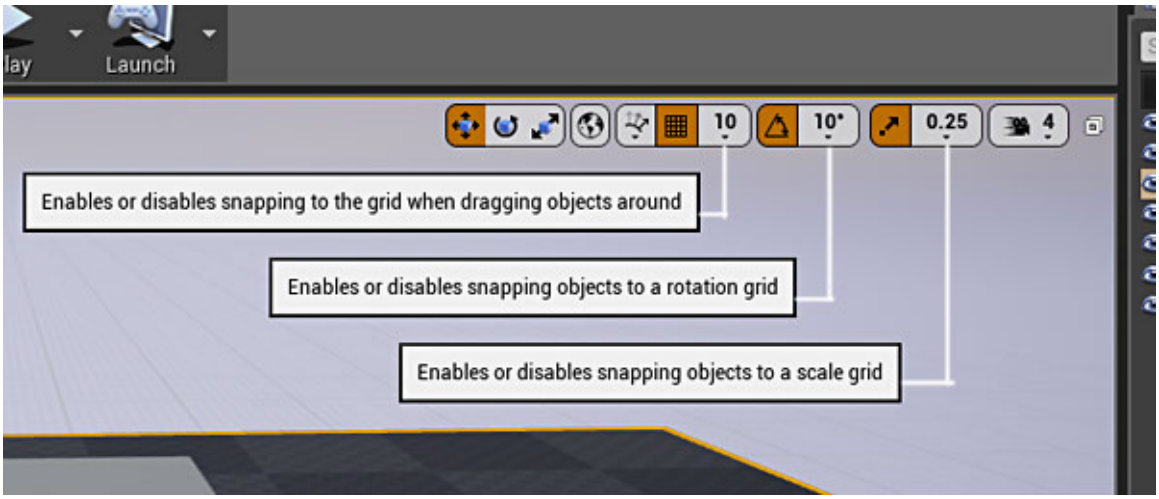
Grid snapping is enabled by default.

To enable/disable grid snaps left-click on the move, rotate and scale snaps icon within the viewport.

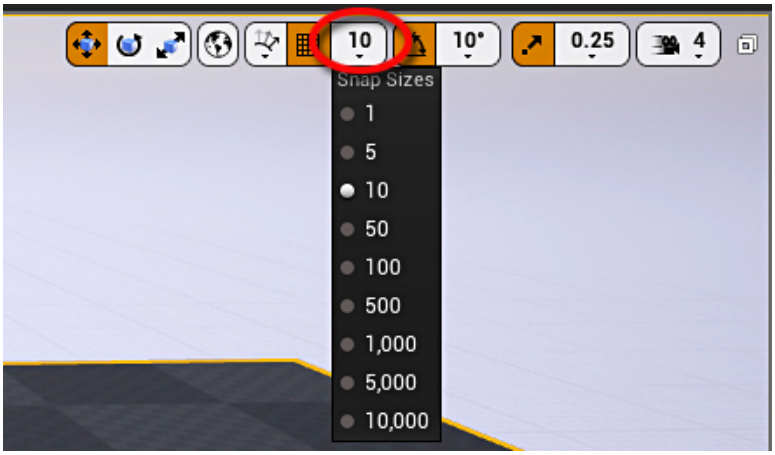
You have 3 type of grid snapping options - move, rotate and scale.



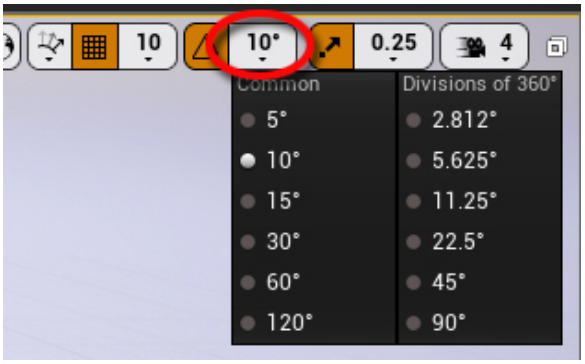
Use the drop down menu to set the size of each grid type and at what values you want them to snap to.



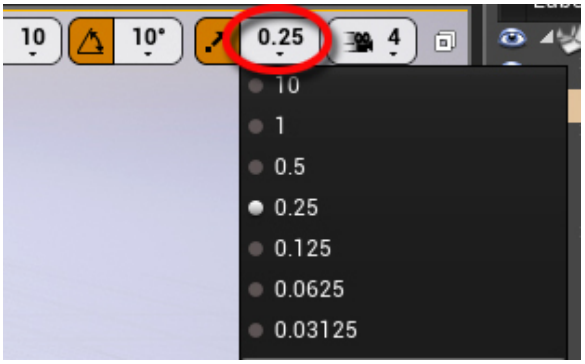
Move snap grid option will be one of the most important ones. Move grid spacing starts at 1 and continues 5, 10, 50, 100, 500, 1000, 5000, 10000:



Rotate snap degrees:



Scale snap values:



Shortcut key to changing move grid spacing are the bracket keys [].

- **[ ]** = Decrease/Increase Grid Size

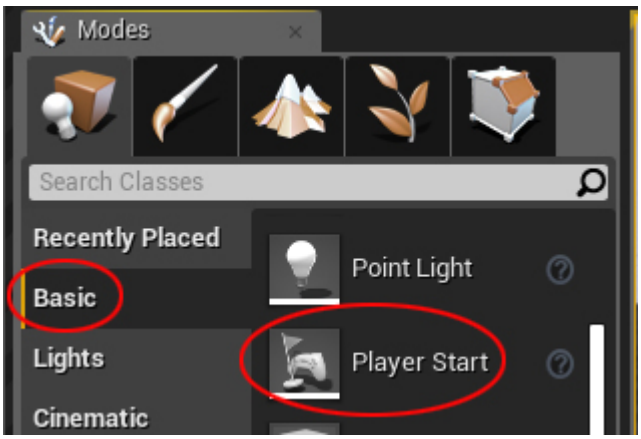
Begin with grid spacing of 10, 50 or 100 as you construct the level. Switch to lower grid snaps of 1 and 5 for more precise detail work.

19. INSERTING A PLAYER START

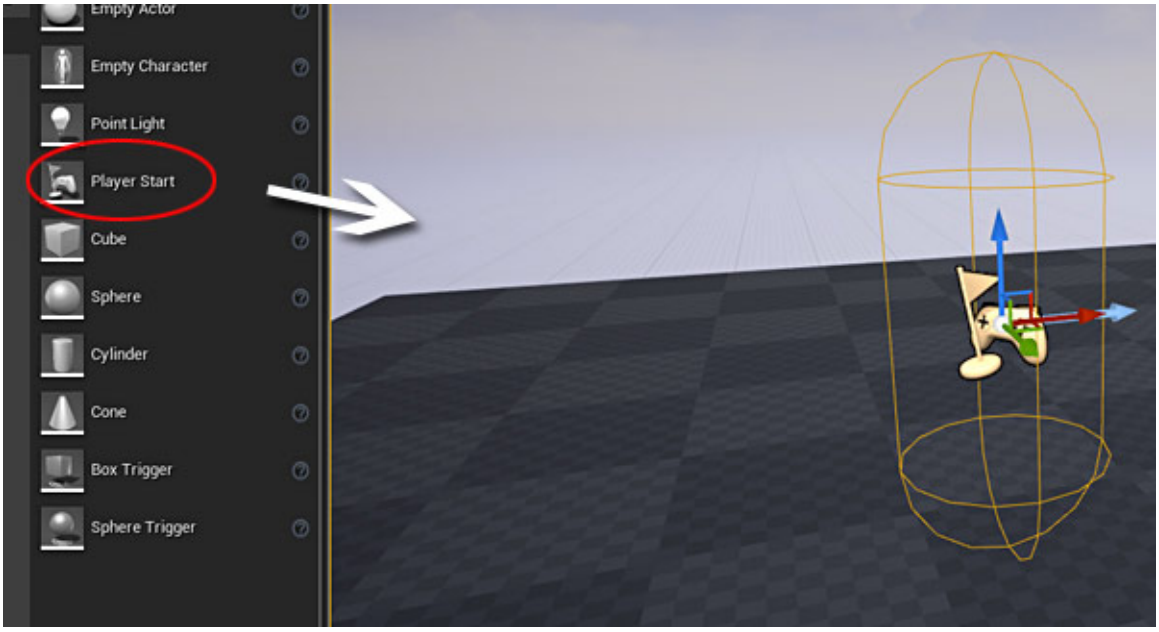
Player Start defines the location a player character will spawn from inside the level. Every environment you create should have at least one player start.

If you are creating a stand-alone game environment that doesn't require player participation, you probably won't need one. Although, it's a good habit to still insert one.

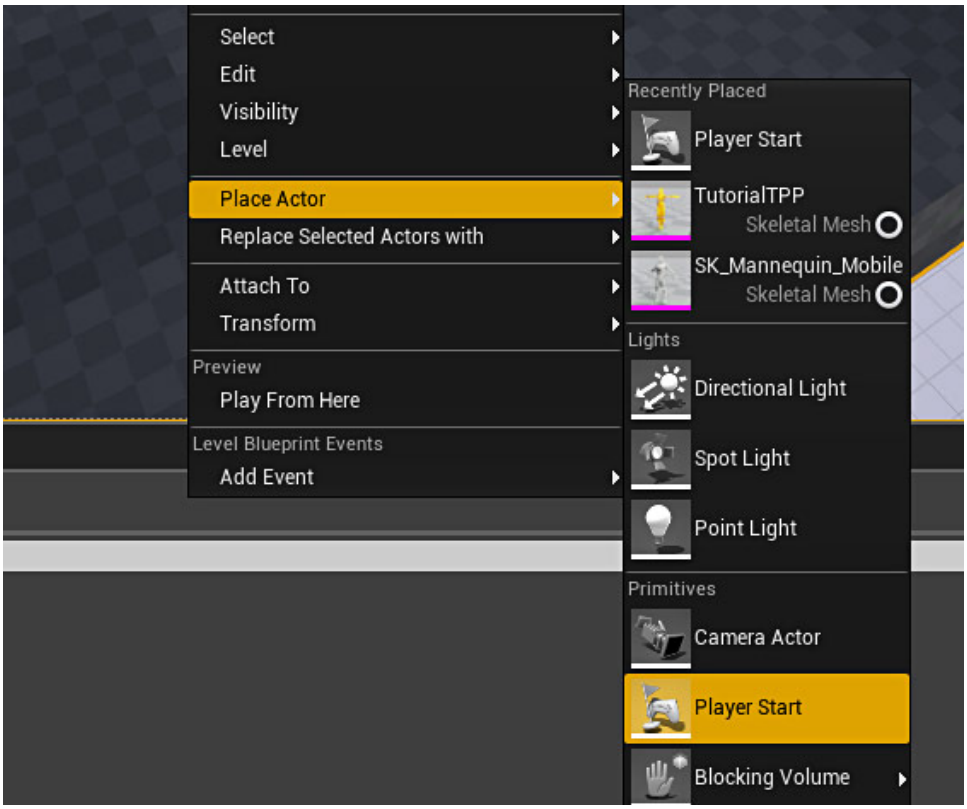
To insert a Player Start, go to Place Mode (Shift+1) and Basic:



Left Click and drag the Player Start from the menu into your level:



Another common way to insert is Right Click inside perspective viewport, choose **Place Actor → Player Start**:



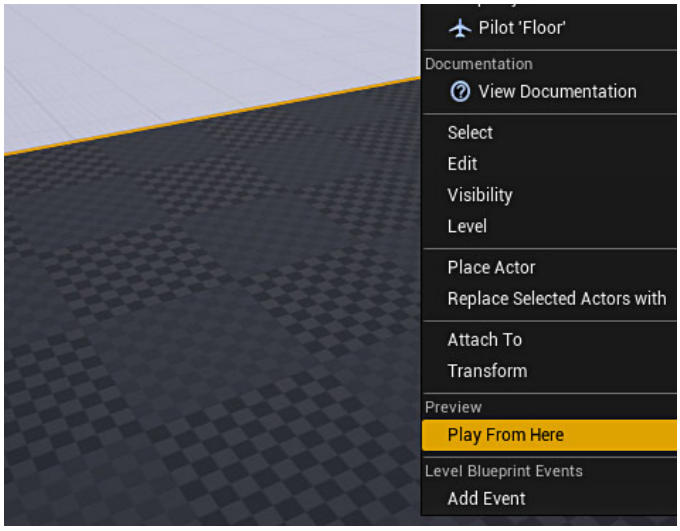
20. PLAY TESTING YOUR LEVEL

You'll use the perspective viewport as main source for how the level looks. Remember two useful shortcuts to see your level as it would appear in-game:

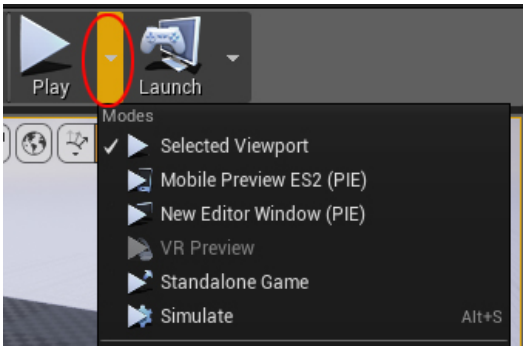
- **G** = Game Mode
- **Ctrl + R** = Real Time

One of the most powerful features of UE4 is the ability to spawn inside the level and playtest with a single button press.

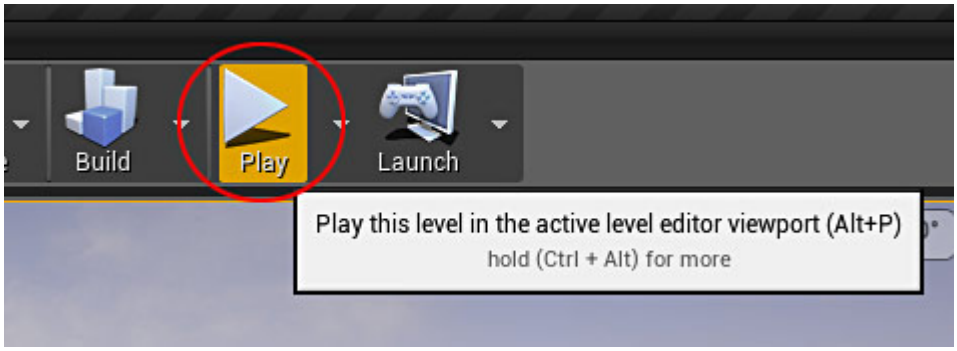
In perspective viewport, Right Click where you would like to spawn from and choose **Play From Here**. It will spawn you right where you Right Clicked and chose to Play From:



For additional options use the drop down menu and choose how you want to Play Test your level; such as Selected Viewport, Mobile Preview, Standalone Game or Simulate:



To use a **Player Start** as a spawning location, choose **Selected Viewport**.



Note: whatever the last option you chose will be the one set as an icon at the top toolbar:

I use **Play From Here** as well as **Selected Viewport** most of the time.

If the level doesn't contain a Player Start then you'll spawn right where your camera view is. So it's recommended that you have at least one Player Start in your level.

Blank Projects will spawn you as a free floating camera at player start or from camera view position.

- **Selected Viewport**: will play in the active viewport
- **New Editor Window**: start play in editor in a new window
- **Simulate**: simulate the game inside the active viewport, good for testing physics and Blueprints

Press Escape to exit play testing and simulate.

- **Esc** = Exit Play In Editor

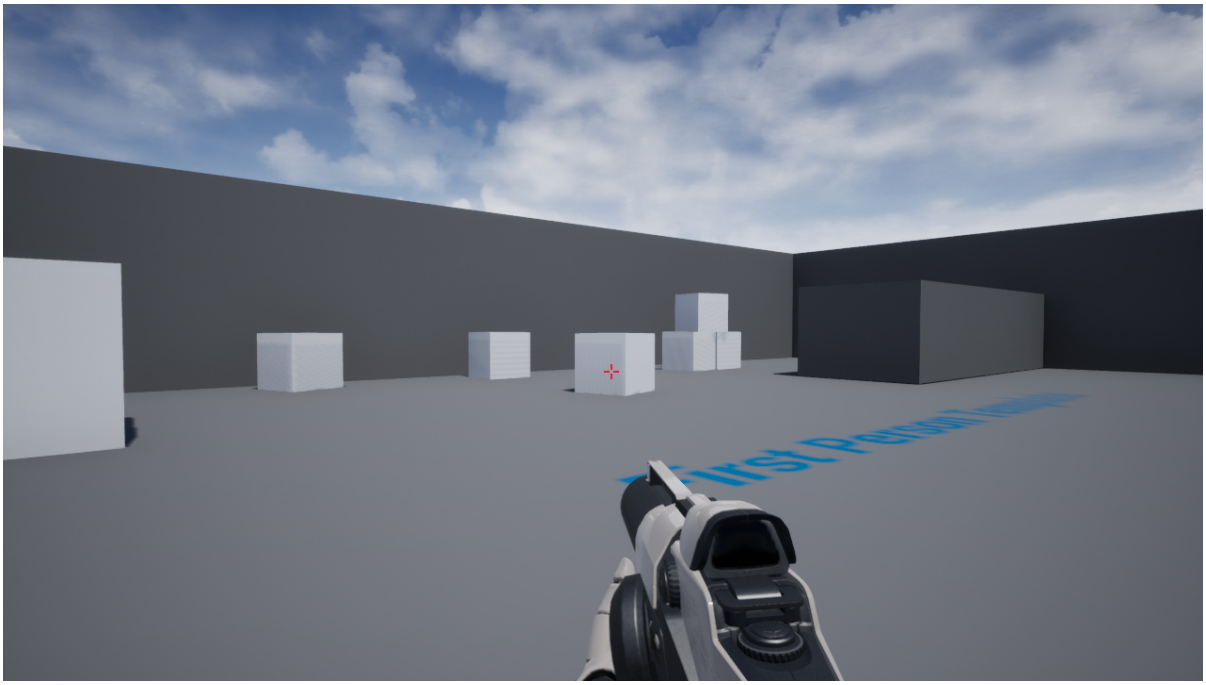
Also, you are able to play test your level the same way as how the player will see it through the game menu. So if you had setup UI (User Interface) to start your map, it would appear just as the player would see. Choose **Standalone Game** or **Mobile Preview** as the launch option.

- **Standalone Game**: starts the game in its own window and opens the level through the game menu; it is how the end-user will see this level being accessed and played in-game
- **Mobile Preview**: starts the game in its own window and opens the level as Mobile Preview for how the end-user will see it being accessed and played in-game on a targeted Mobile device.

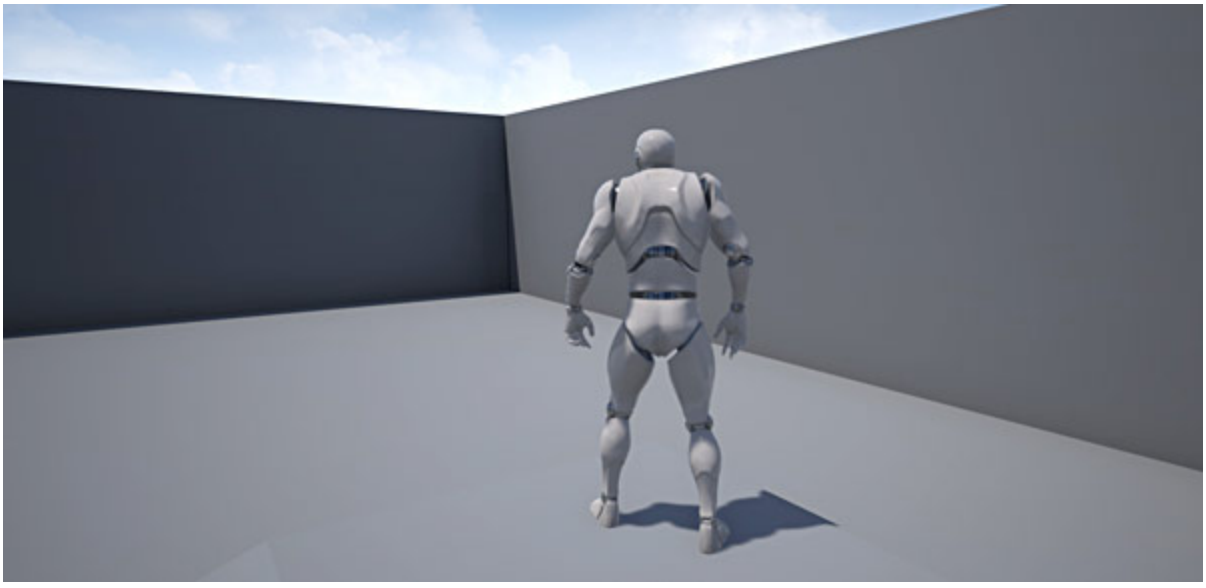
Press Alt + F4 to exit Standalone Game and Mobile Preview testing.

- **Alt + F4** = Exit Standalone Game or Mobile Preview

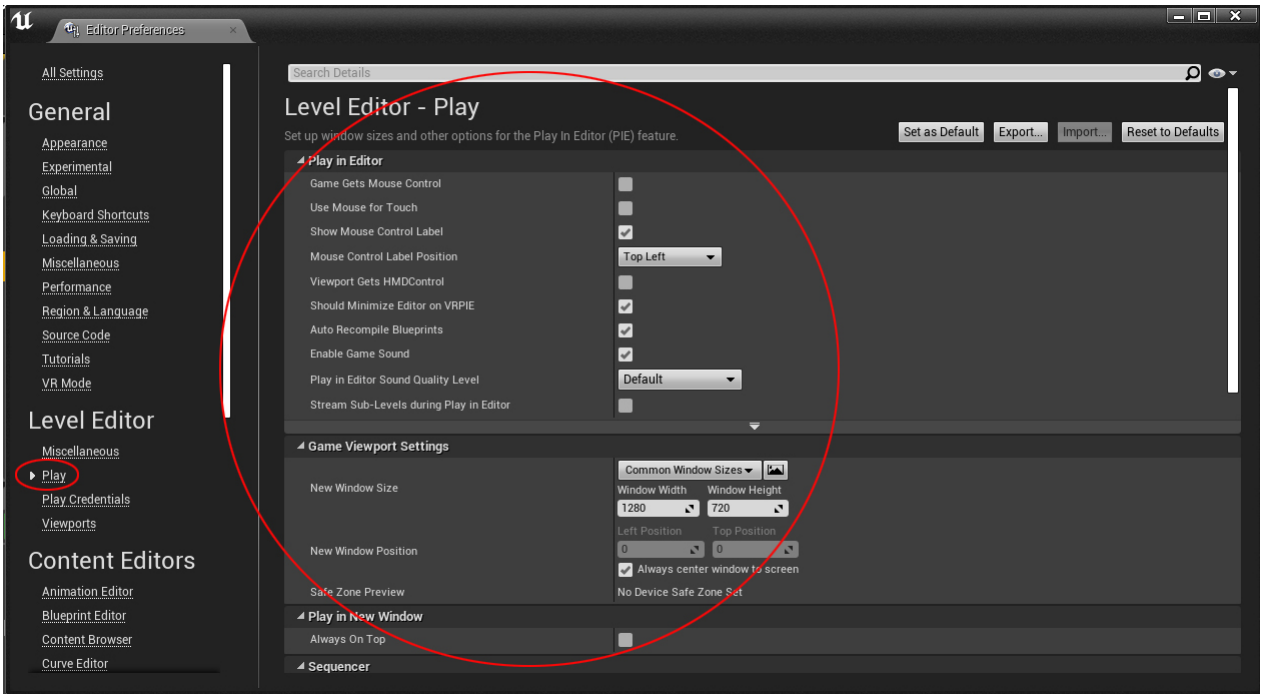
The way you spawn into the level will vary depending on the game template you chose for the project. For example spawning inside the level with FPS Shooter Template:



Spawning inside the level with Third Person Template:

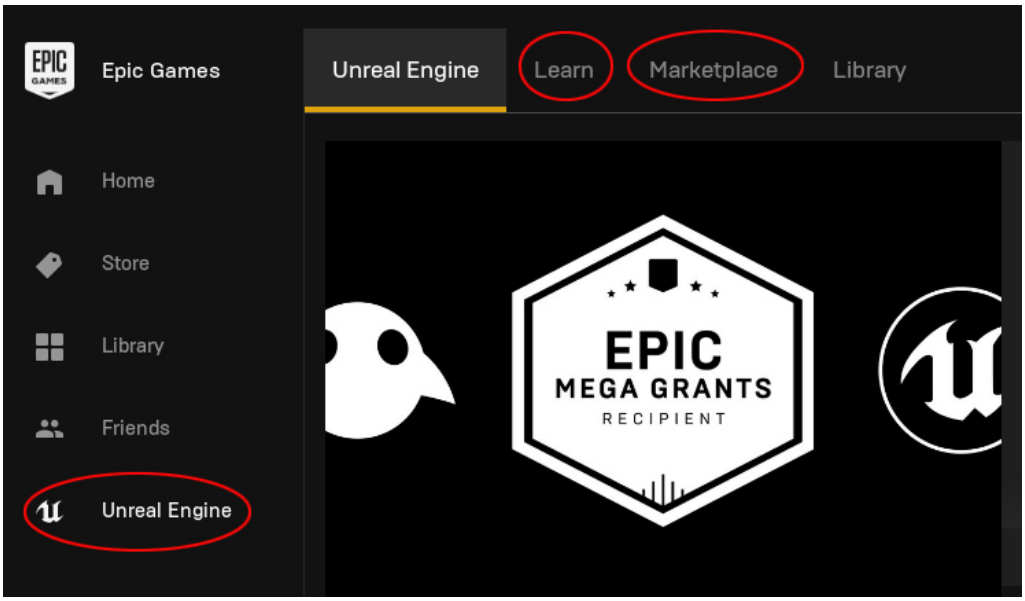


Few additional settings you can change for Play in Editor function is under **Edit → Editor Preferences** then **Level Editor: Play**



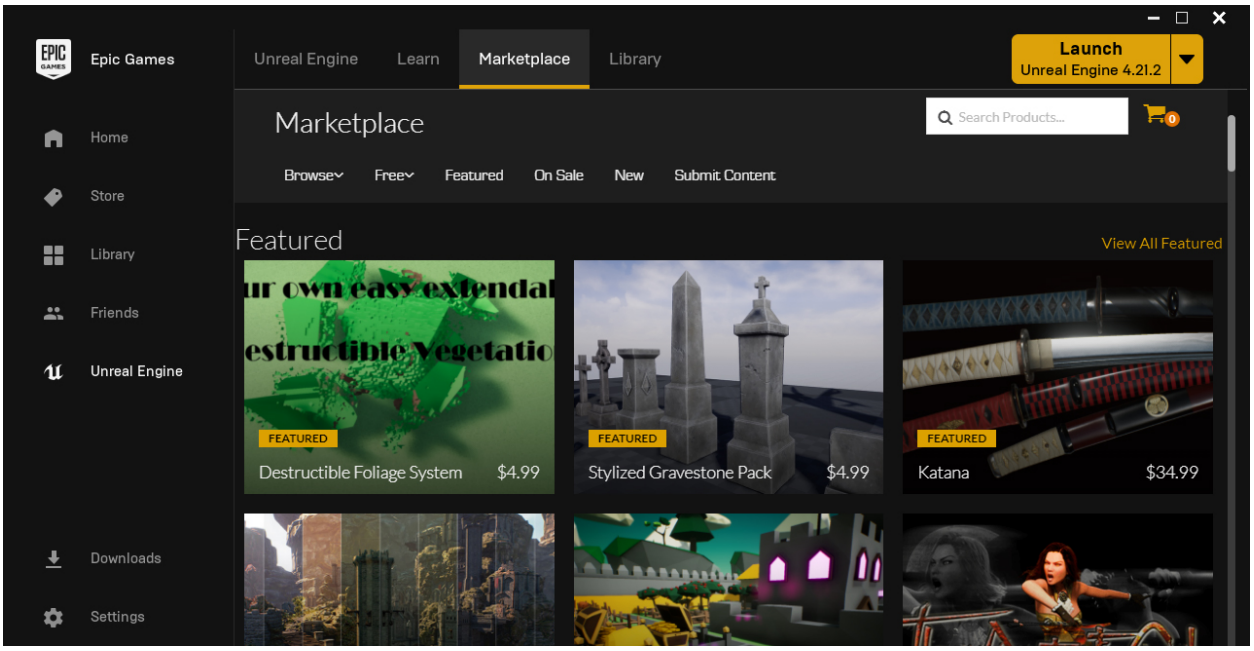
21. MARKETPLACE CONTENT AND LEARN SECTION

Marketplace and Learn sections can be found inside Epic Games Launcher in Unreal Engine section:

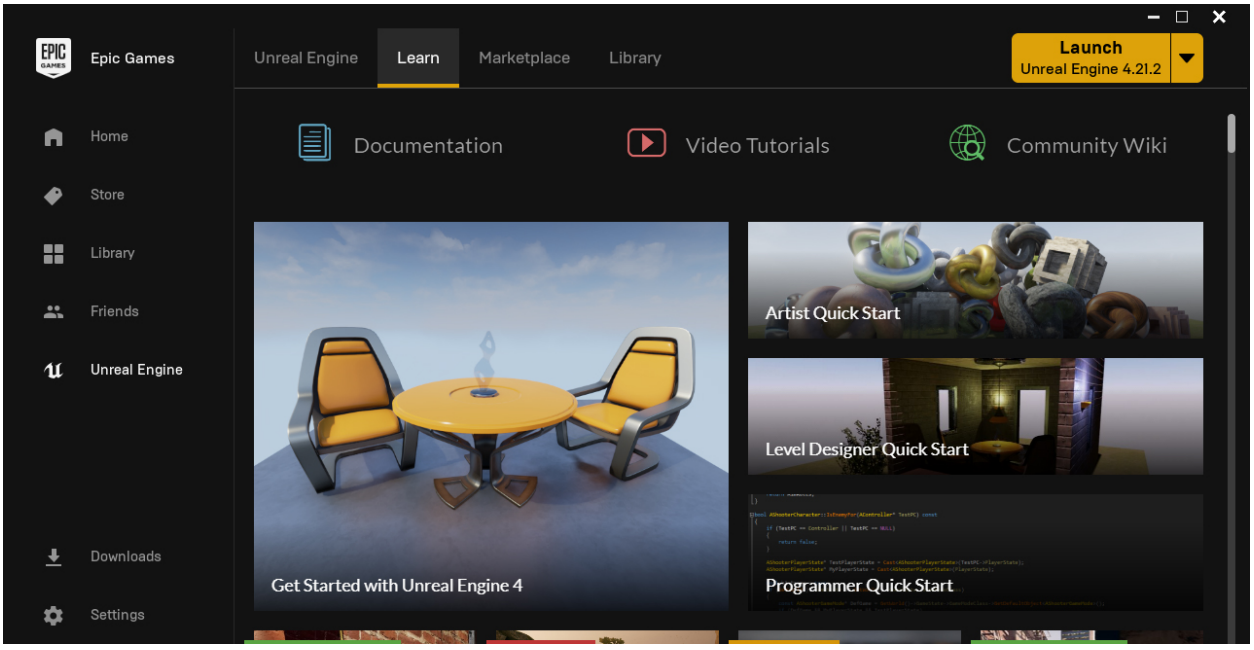


Marketplace allows you to download free and paid content such as Static Meshes, characters, animations, sounds, particle effects, materials etc.

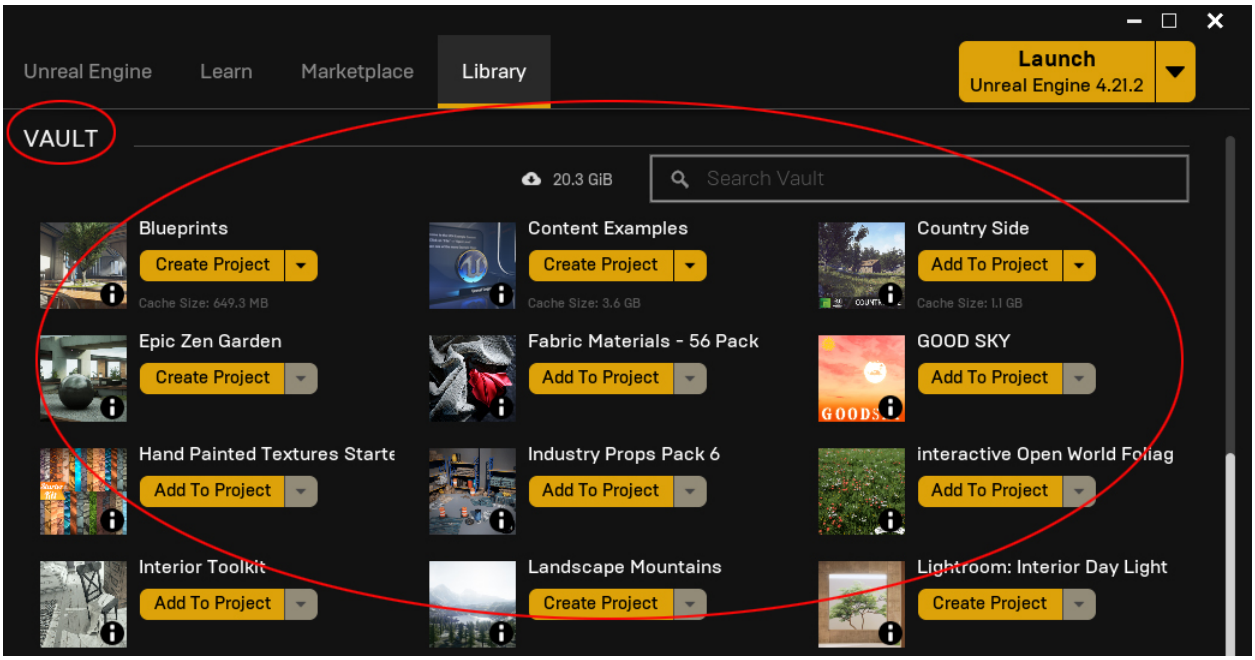
All of these can be used to help you create game environments without having to create these assets on your own:



Learn section allows you to download free game engine samples and game examples. You can use these to [reverse engineer] to learn from.



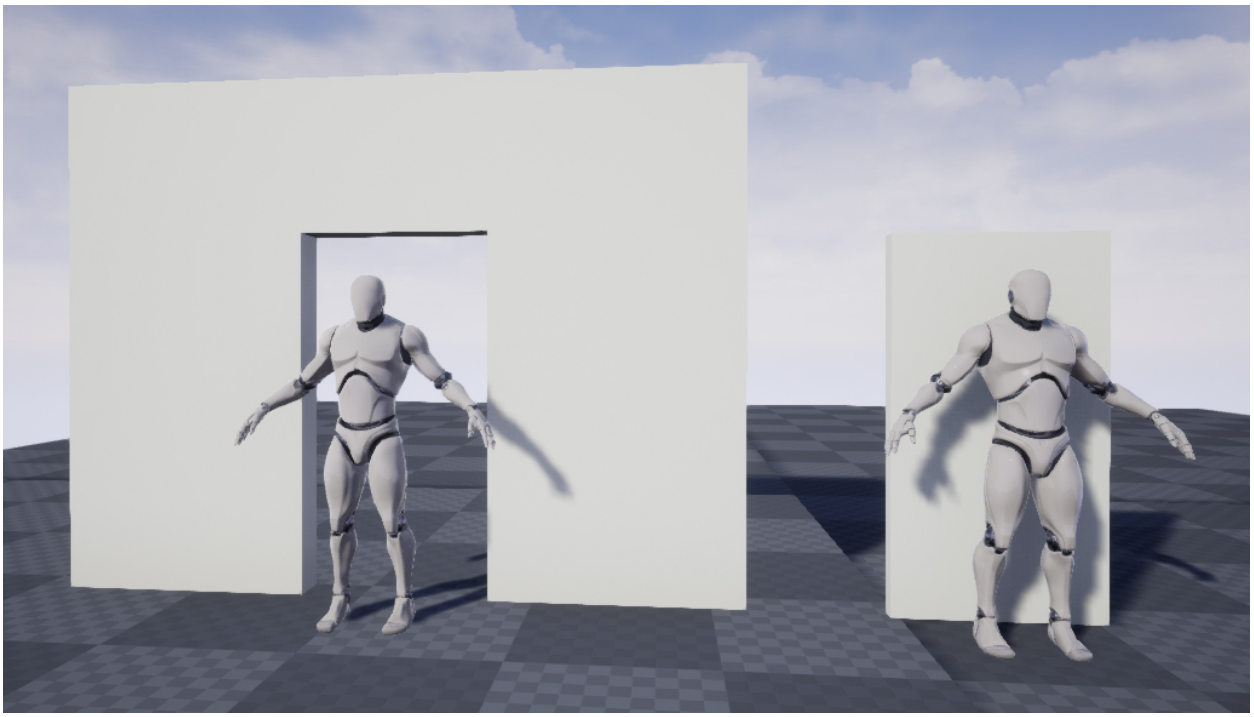
Any project you add from the Learn Section or UE4 Marketplace will appear in the **Vault section of the Library**:



Next section is extremely important!

It covers world scale dimensions and how to keep everything you create to correct proportions.

SECTION 2:
WORLD SCALE DIMENSIONS & PROPORTIONS



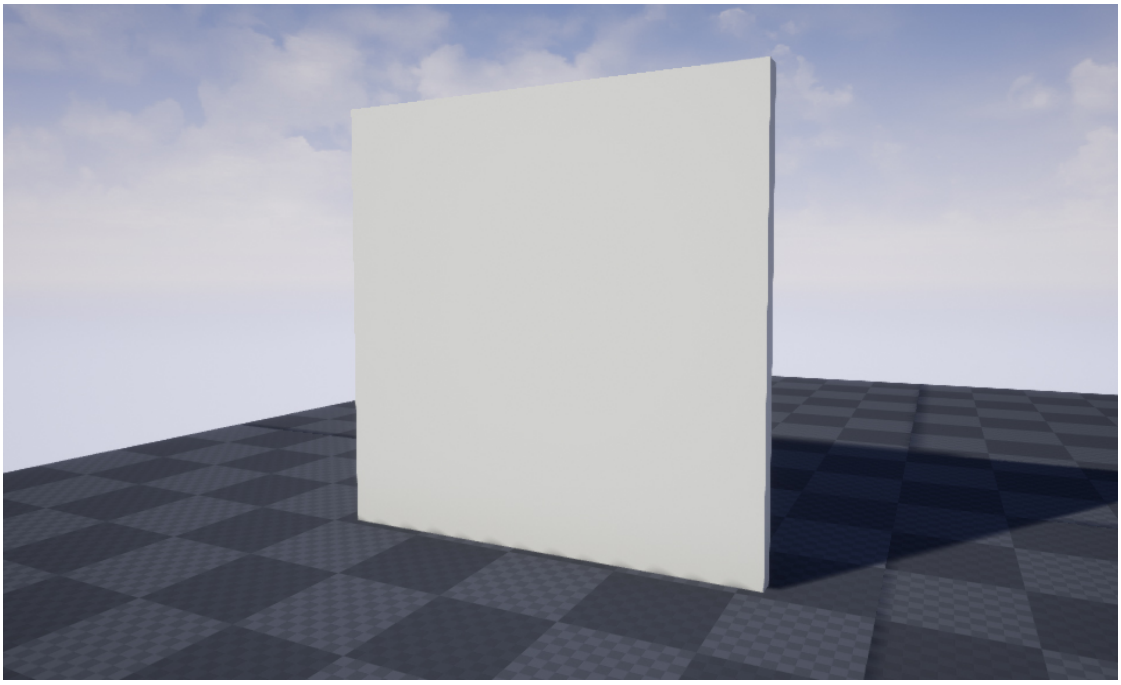
How do you create your environments to correct scale dimensions and proportions?

Scale and proportion are probably one of the most important elements when building levels in UE4.

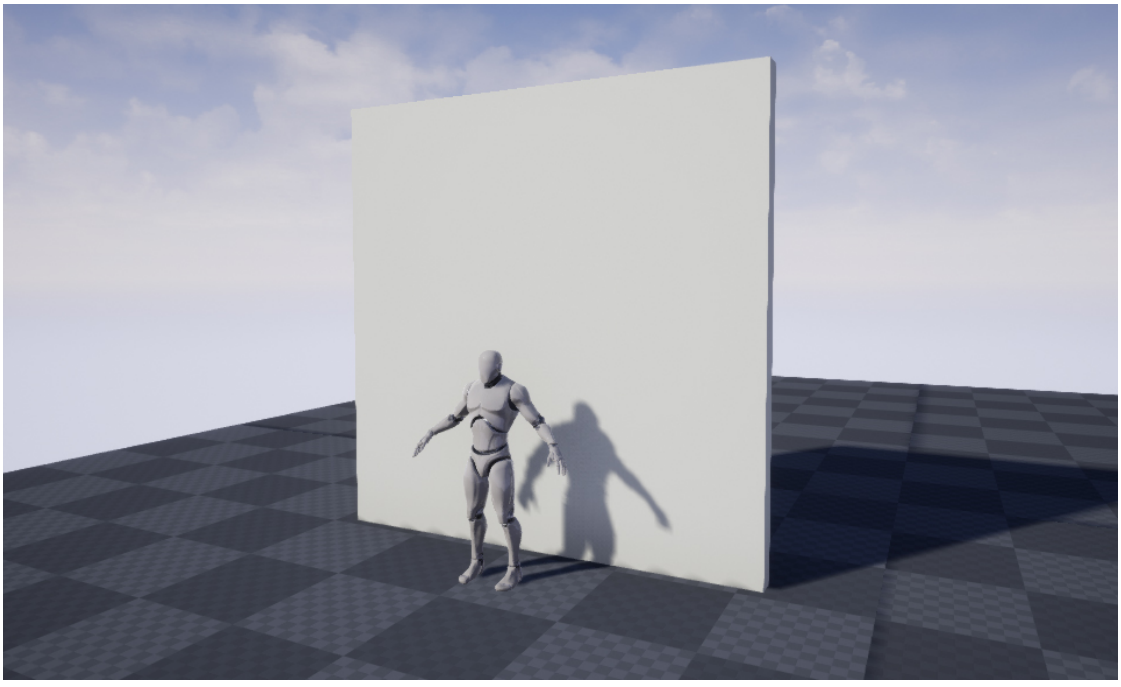
Nothing destroys the illusion of your level faster than disproportionate game world.

4 KEYS TO BUILDING EVERYTHING TO CORRECT SCALE AND PROPORTION

If you have a wall, it's difficult to tell how big that wall actually is:



Until you have a human reference scale next to it:



You should always insert some sort of a human reference scale into your levels to judge proportions.

Important to Know:

- 1 unreal unit = 1 centimeter

In UE4, base character dimensions are:

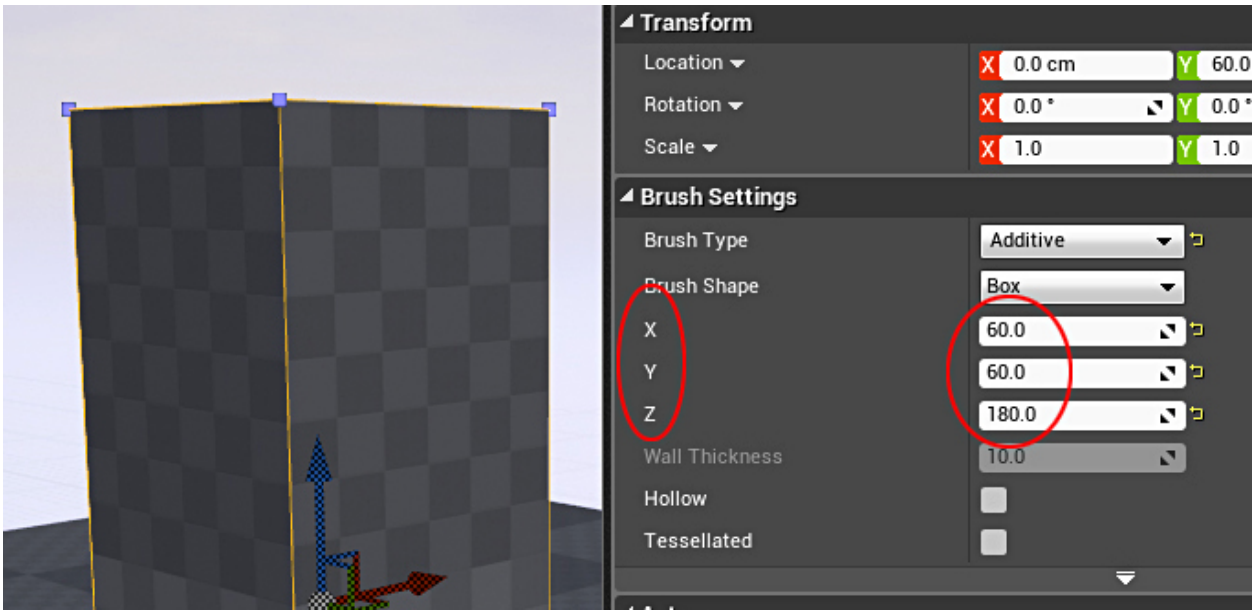
- Height: 180uu/cm
- Width/Length: 60uu/cm

Important: in-game scale and proportion should always be consistent to the game you are working on. Use the default, 180 unreal unit height as the starting point but know this may vary depending on the style of the environment you are creating and dimensions of an actual characters in-game.

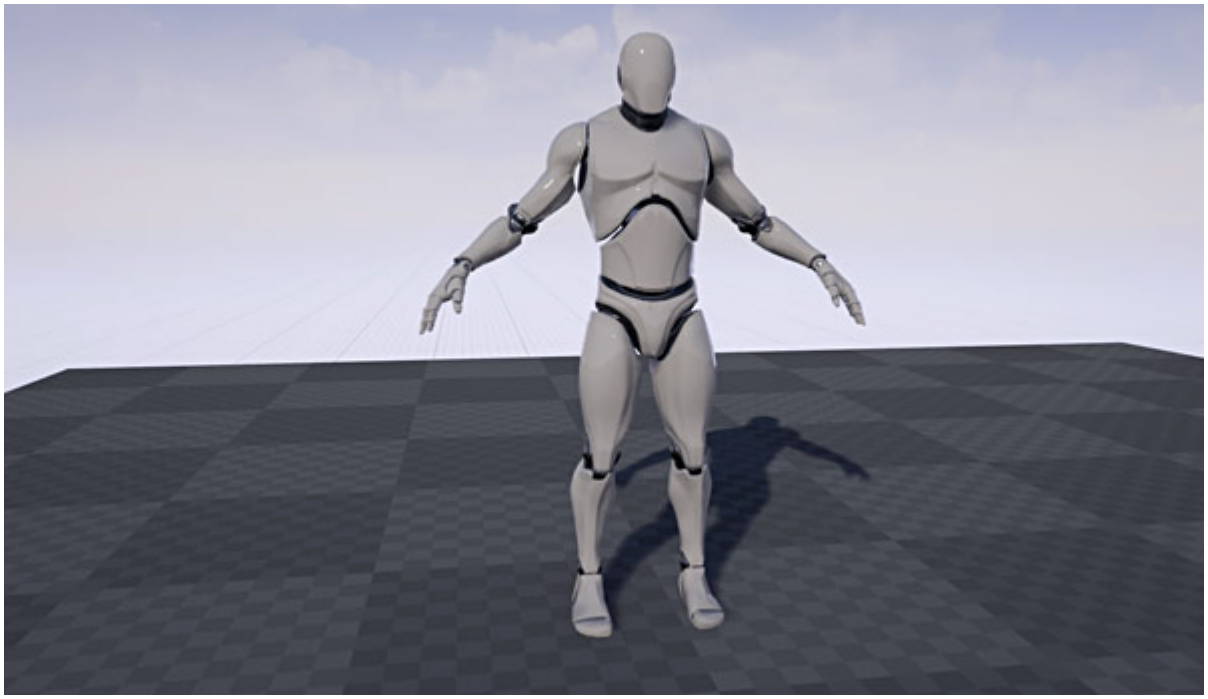
There are **4 options** for using human scale reference.

Option 1: Create a BSP brush with the following dimensions:

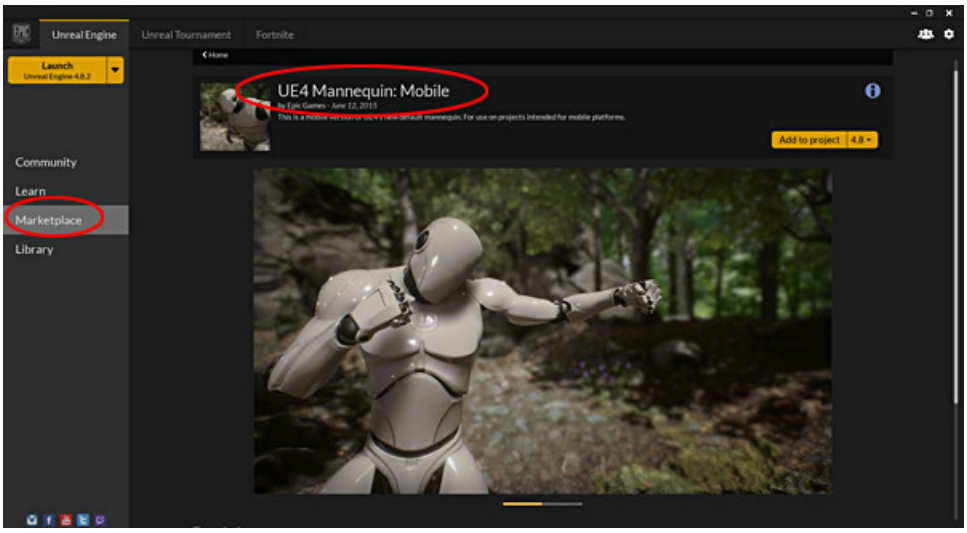
- Height: 180uu/cm
- Width/Length: 60uu/cm



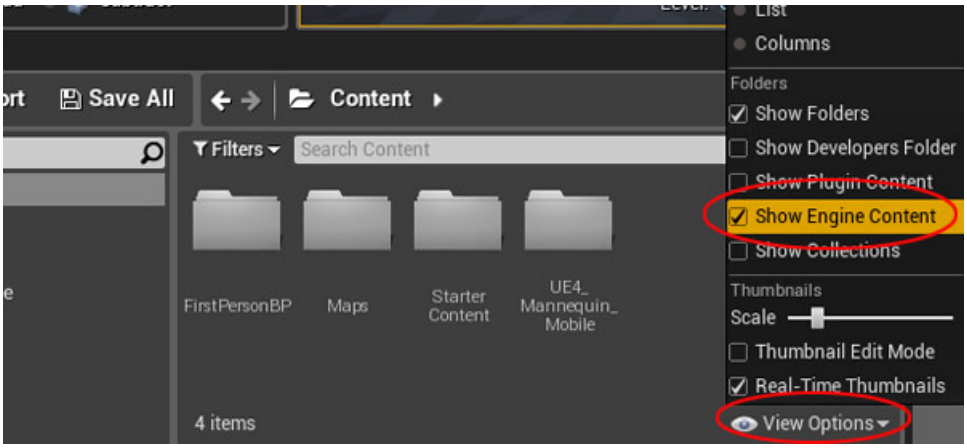
Option 2 (What I Use): Use **Mannequin Mobile** skeletal mesh available from Marketplace for free.



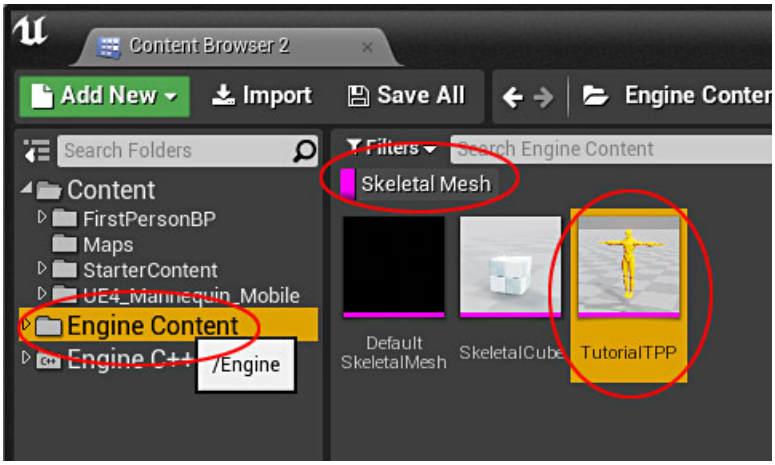
See this tutorial on how to download it and add to your projects: https://www.youtube.com/watch?v=tL9ba98H2_Q



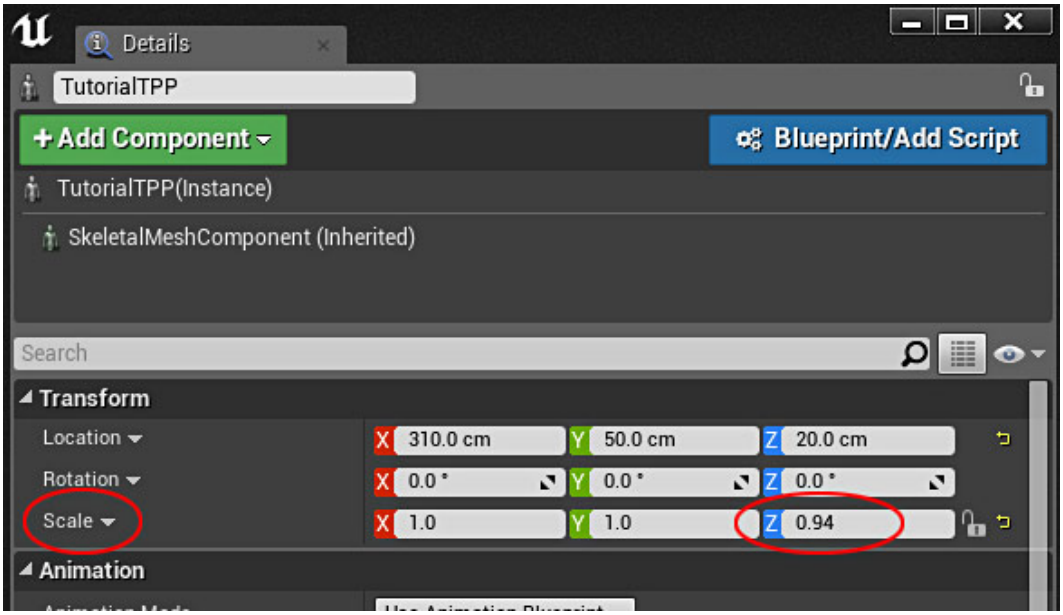
Option 3: Insert skeletal mesh reference from the Epic's Content Folder. Go to Content Browser, under **View Options** enable **Show Engine Content**:



Select Engine Content folder and filter by Skeletal Mesh:



Use the **TutorialTPP skeletal mesh character** as a starting point for your scale. This character is slightly taller than 180 units in height, so scale this down using the Details panel on Z axis down to .94:



Option 4: If you creating or working on a game, check to see what scale and dimension the game uses. Then use the provided character mesh as a reference.

Use one of these options to insert human reference scale into your level to judge proportion as you build. Once you have everything blocked out and it looks proportionately correct, you can delete these human references from your level. If you are creating a stand-alone game environment without any gameplay, use 180uu/cm height as the character guide.

4 keys to building everything to correct scale and proportion:

1. Use human/character scale model that is are the same dimensions as a player in game
2. Establish base dimensions for all of world's architecture (walls, doors, doorways, stair, windows etc.)
3. Constantly reference your level to player character scale and architecture surrounding it
4. Use Play in Editor as often as possible to see what everything will actually look like in-game

ESSENTIAL ARCHITECTURE DIMENSIONS

Once you know the base character dimensions you'll use, decide on dimensions for world's architecture.

Know the following common architecture dimensions of your environment:

- Average wall height, width, depth and length
- Door, doorway height and width
- Step height and depth
- Average window height and width

For playable levels you also need to know:

- Character jump height and jump distance
- Crouch height
- Smallest crouch space character can crawl through

You may need to know additional dimensions depending on gameplay mechanics of your game (if any).

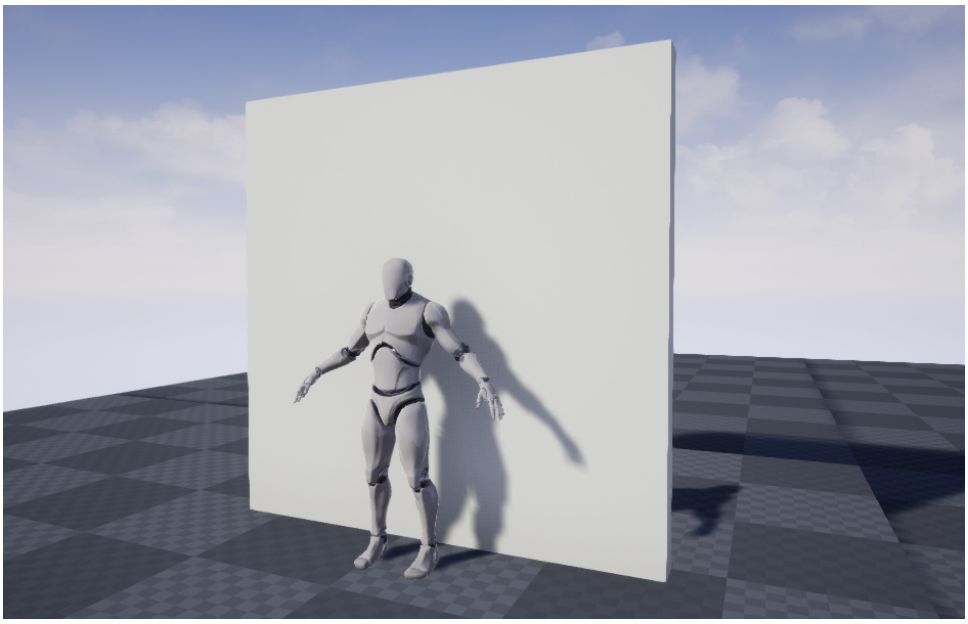
I understand that architecture styles vary so adjust these dimensions to fit your environment style.

Common Architecture Dimensions are listed below. These dimensions are based on default character height of 180uu (about 6ft). These dimensions should get you started for most of your projects.

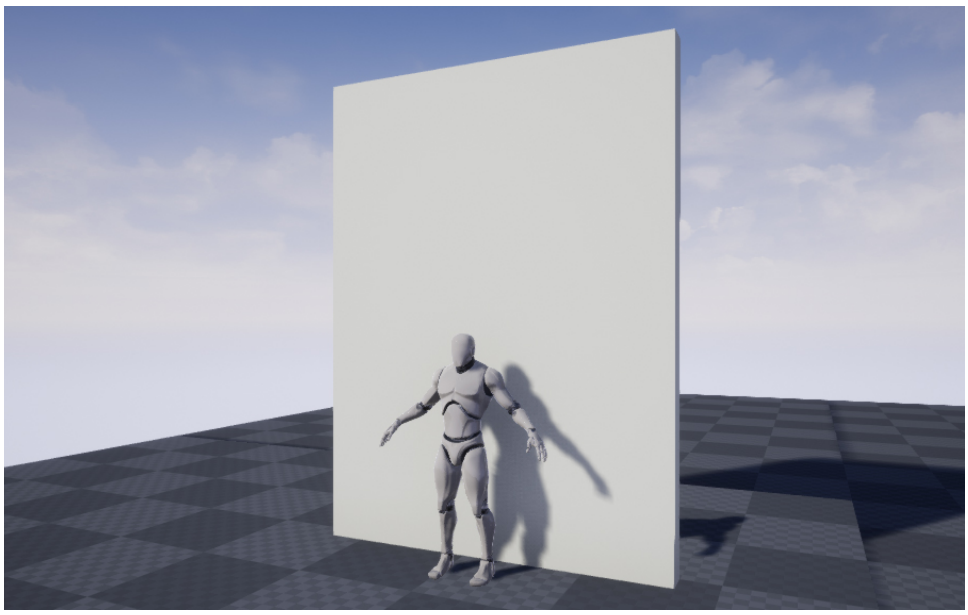
- **Character Height**: 180uu

Walls

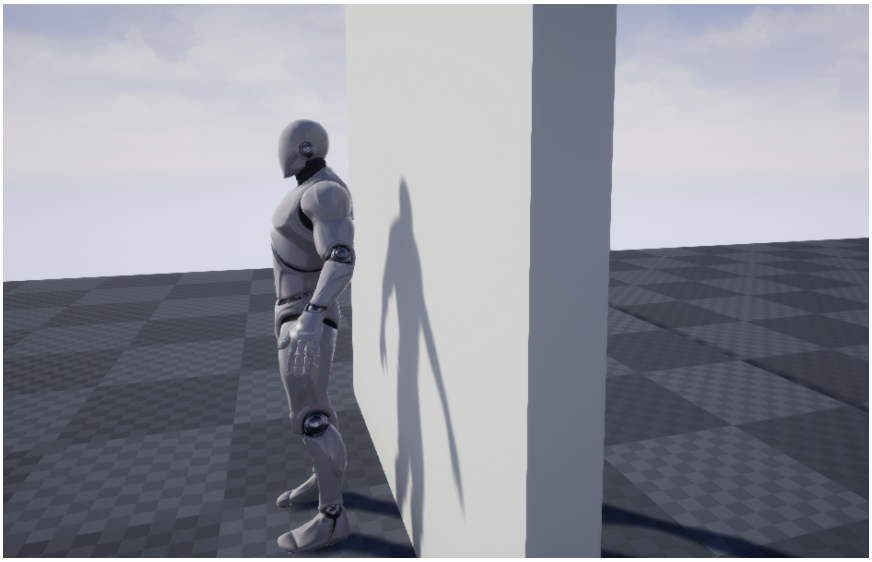
Average wall height is 300uu (9.8ft):



Wall height of 400uu (13.1ft) will be slightly larger and may work better:



Good wall thickness is 20uu:



- Average Wall Height: 300-400uu
- Wall Depth (thickness): 10-20uu

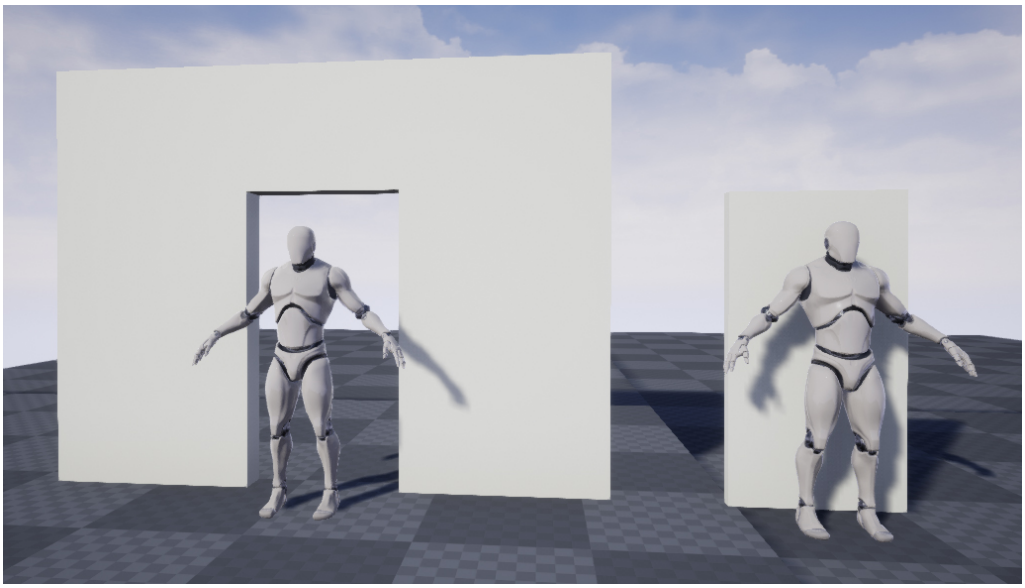
Doorways and Doors

Average doorway and door are 210-240uu height and 110-140uu width.

The dimensions of the doorway below are height at 210uu and width at 110uu:

- Doors/Doorways Height: 210-240uu
- Doors/Doorways Width: 110-140uu

Doors will have the same dimensions as the doorway opening you are using with possible 10-20uu for a door frame that may be included for aesthetics:



Always test in-game to see if you can walk through it due to player collision.

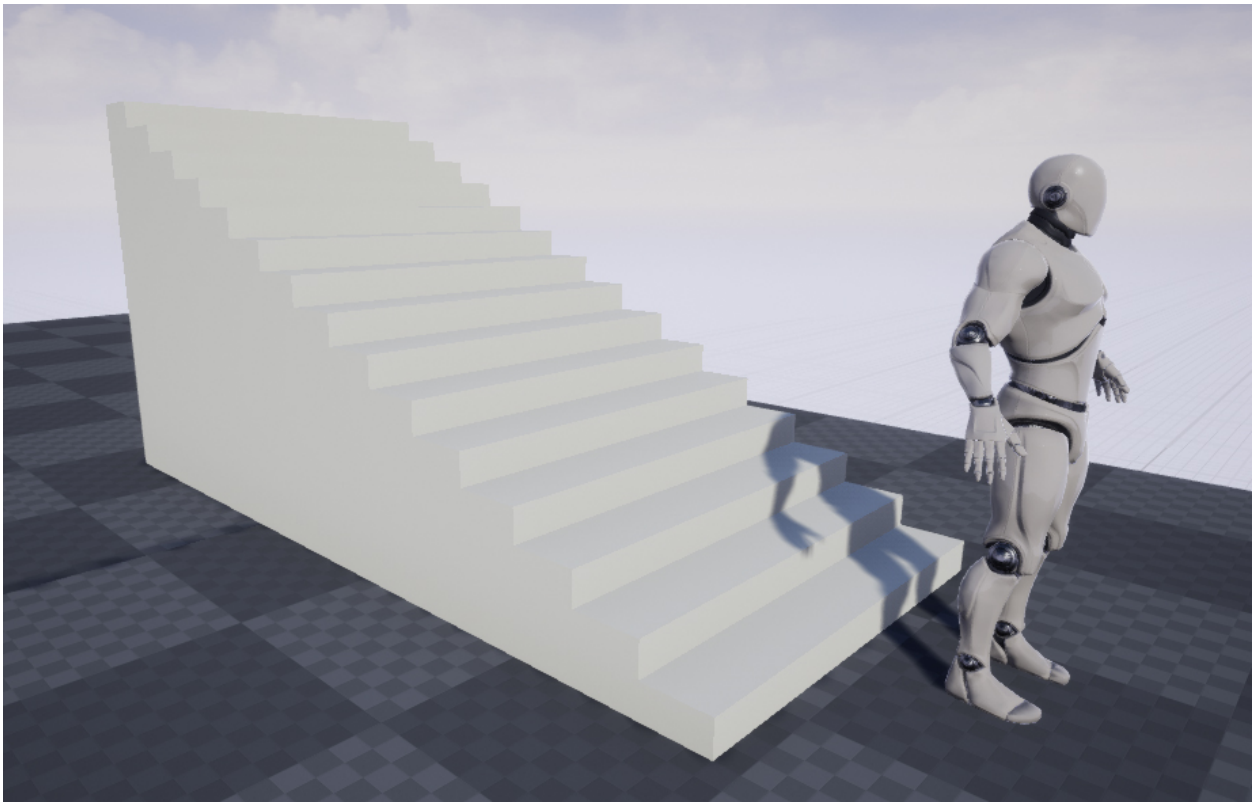
Dimensions that look good for Arch Viz are 210uu height and 110uu width:

- Door Height: 210uu
- Door Width: 110uu

Stairs

Average stair height and depth that looks the best are:

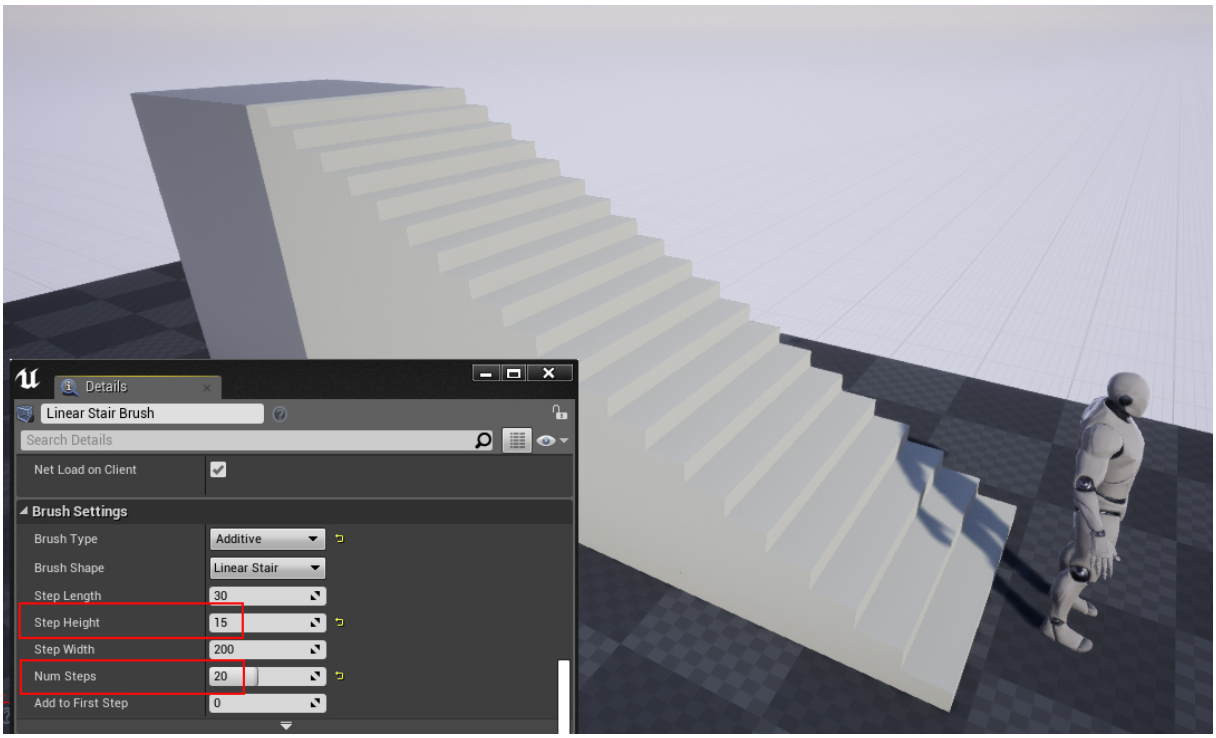
- Step Height: 15uu
- Step Length/Depth: 30uu
- Step Width: varies depending how wide you want the stairs to be



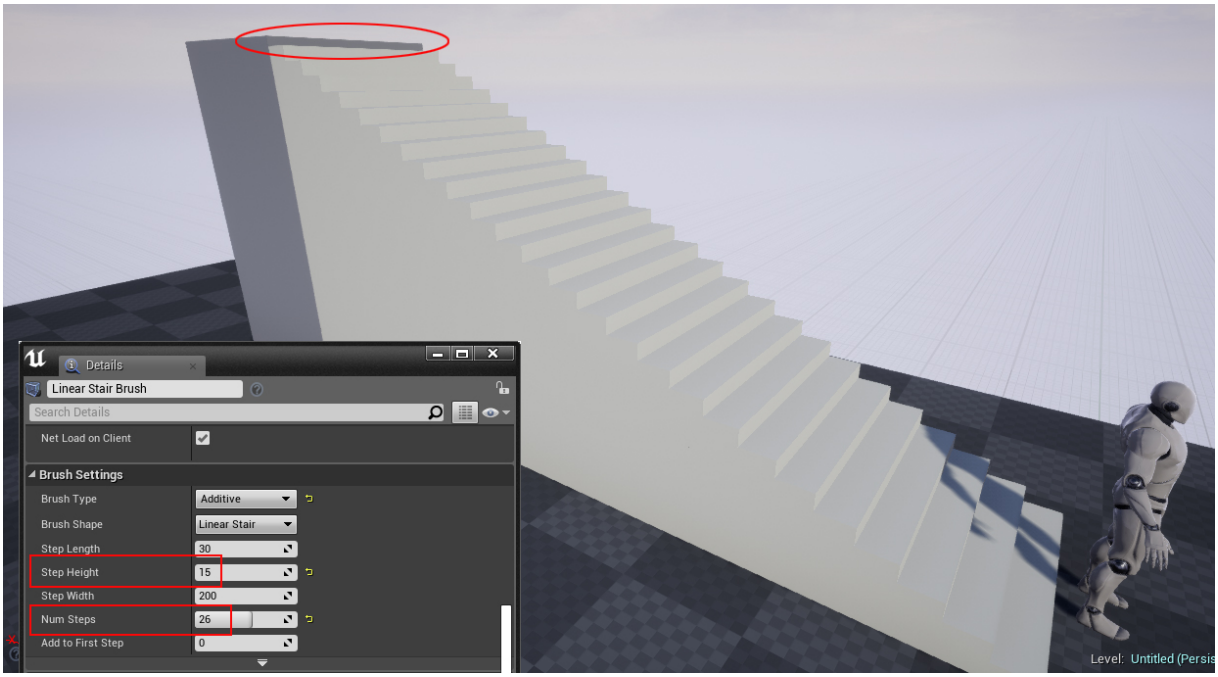
One thing to consider when you are creating stairs is the height of the entire stair set.

If you have a wall that is 300uu high then you would need 20 continuous steps to reach that height.

- 20 steps x 15uu each step = 300uu



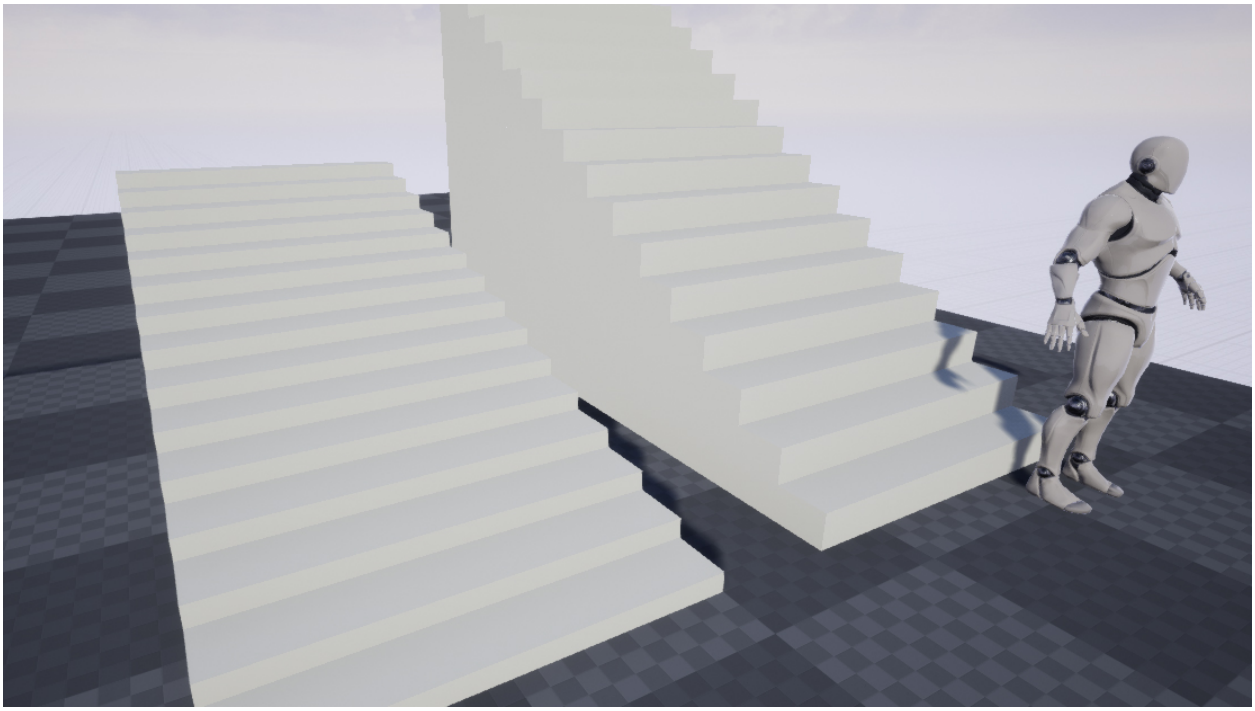
If you have a wall that is 400uu high then you would need 26 steps to reach that height and 27 steps would be too high. This leaves you with a tiny gap at the top of the stairs and second floor:



So this is something to keep in mind as you create stairs for different floor heights.

If you use stair height of 10uu or 20uu and depth of 30uu, these will work but the step height of 10uu (left) is too small and 20uu (right) is larger than I would like it to be.

These don't tend to look right as steps with 15uu in height:



Again, you will have to play around with these values to find what you are happy with.

Windows

Windows are going to be relative to the wall and often will vary in size.

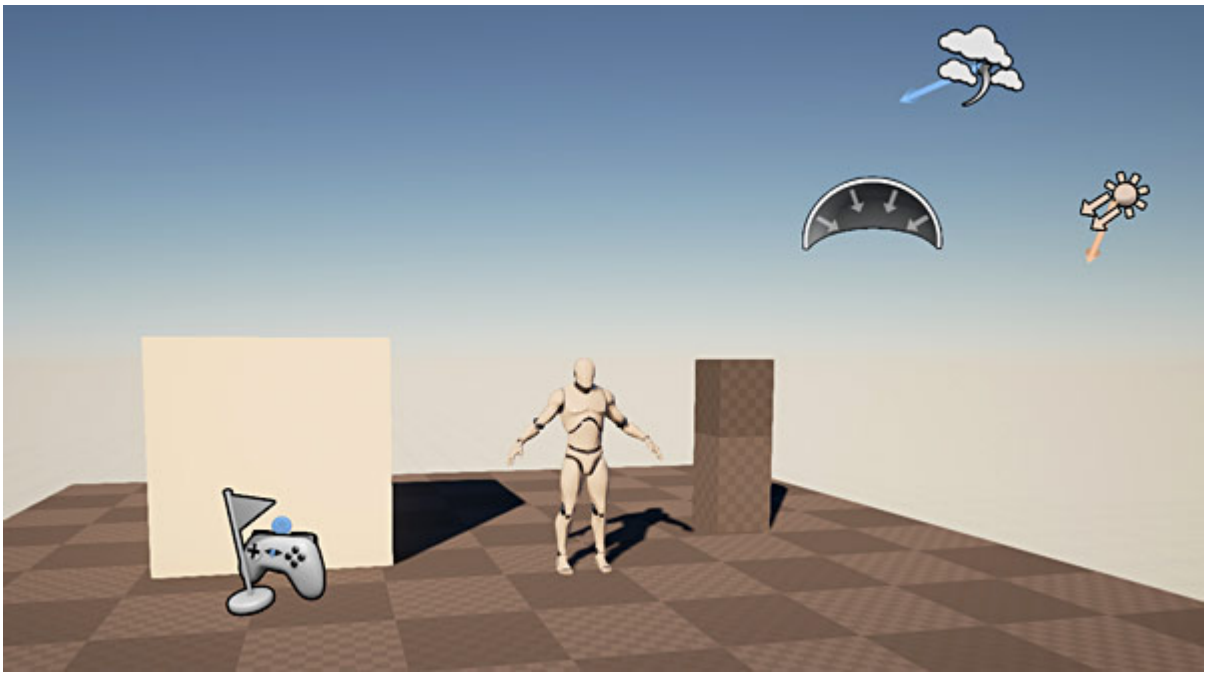
As long as you get the wall size correct, you will be able to carve out a window based on the architecture style you are after.

Here is a window opening with height of 160uu and width of 110uu:

Remember: few other things to about proportions to consider is various field of vision in-game or speed at which the character moves, so you may have to adjust these dimensions to fit your need.

Always test!

SECTION 3:
BEGINNER'S STEP-BY-STEP TO FIRST LEVEL



How do you create your first playable level inside Unreal Engine 4?

As a beginner, you want to get started with a game engine/level editor as quickly as possible and build something you can play in.

We going to re-create the Default Level, the same one you see when you go to File → New Level and select Default.

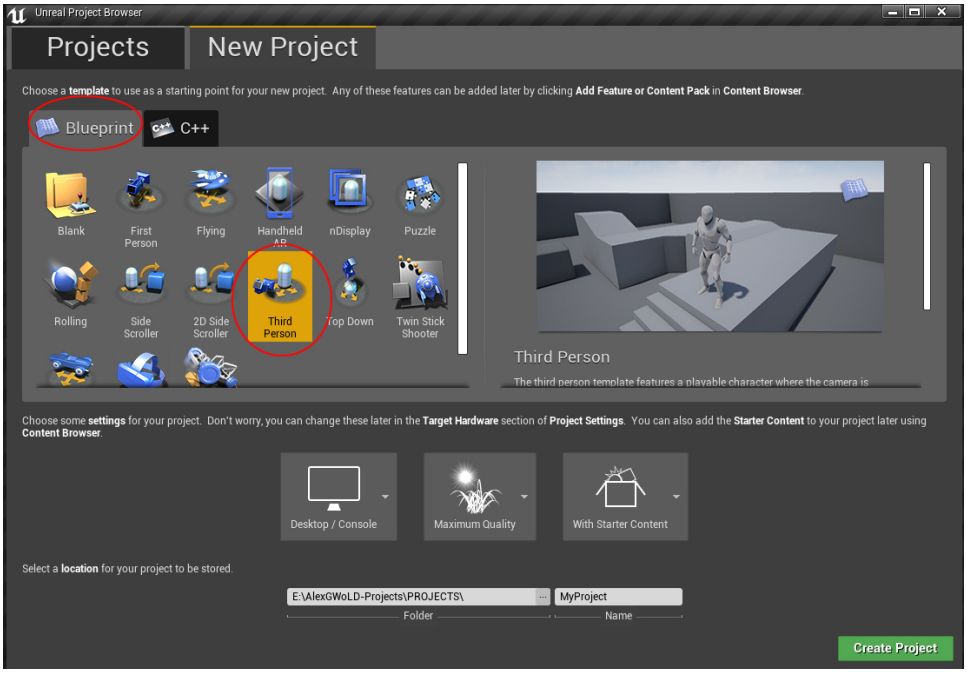
The default level shows you the essential actors you need to have in a working environment. There are only 7 actors here and it is all you need to have a basic functioning level.

Follow this section for a step-by-step for how to create your first playable level.

Let's get started.

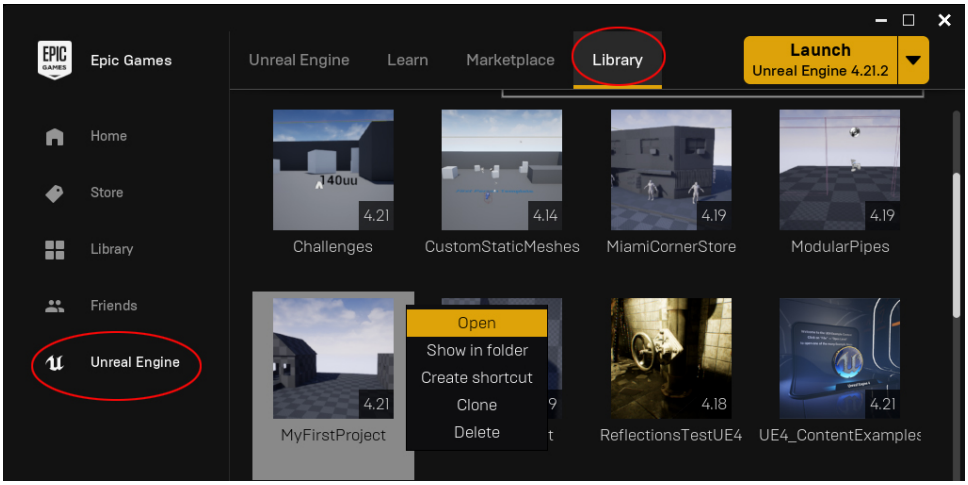
STEP 1: SETTING UP & LAUNCHING THE EDITOR

You should have a project created using any of the available game templates and with added Starter Content. I am using Blueprint Third Person game template and Starter Content but you can use any game template you want.



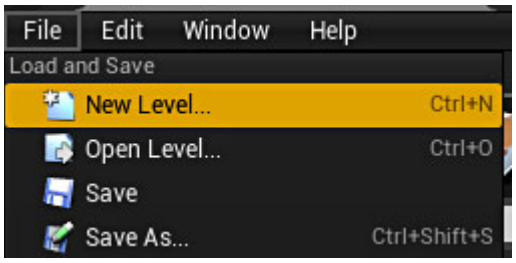
Open your project to follow along.

Inside the **Epic Games Launcher: Unreal Engine**, switch over to **Library** tab and double click on your project image thumbnail or right click and choose open to launch the editor:

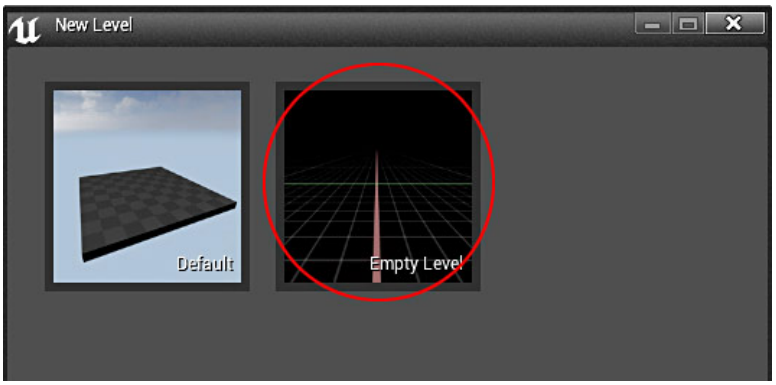


STEP 2: STARTING A NEW BLANK LEVEL

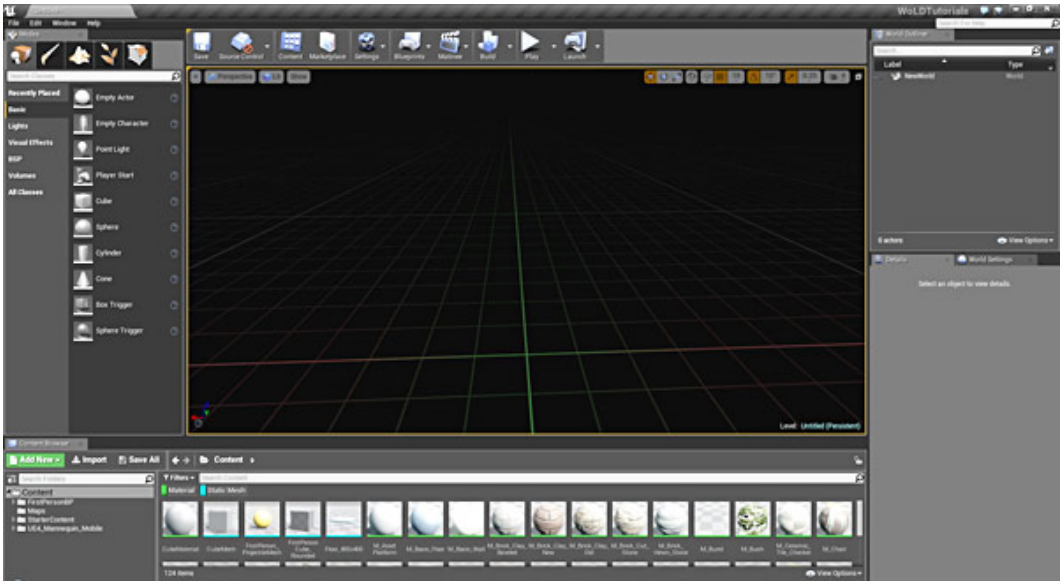
With the editor open, create an empty level. Go to **File → New Level**:



Choose Empty Level:



We are going to start with a blank, empty level and insert all needed actors into it:

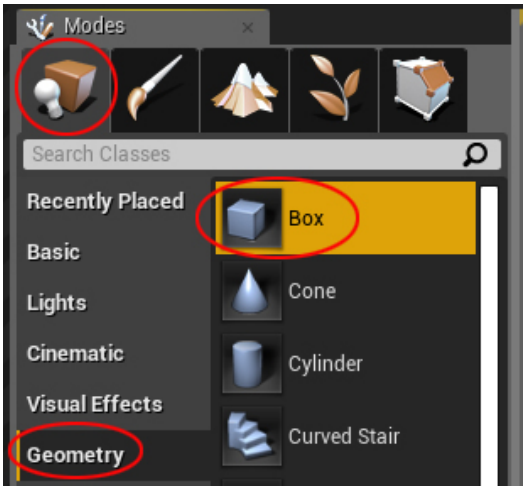


STEP 3: ESTABLISHING A GROUND PLANE

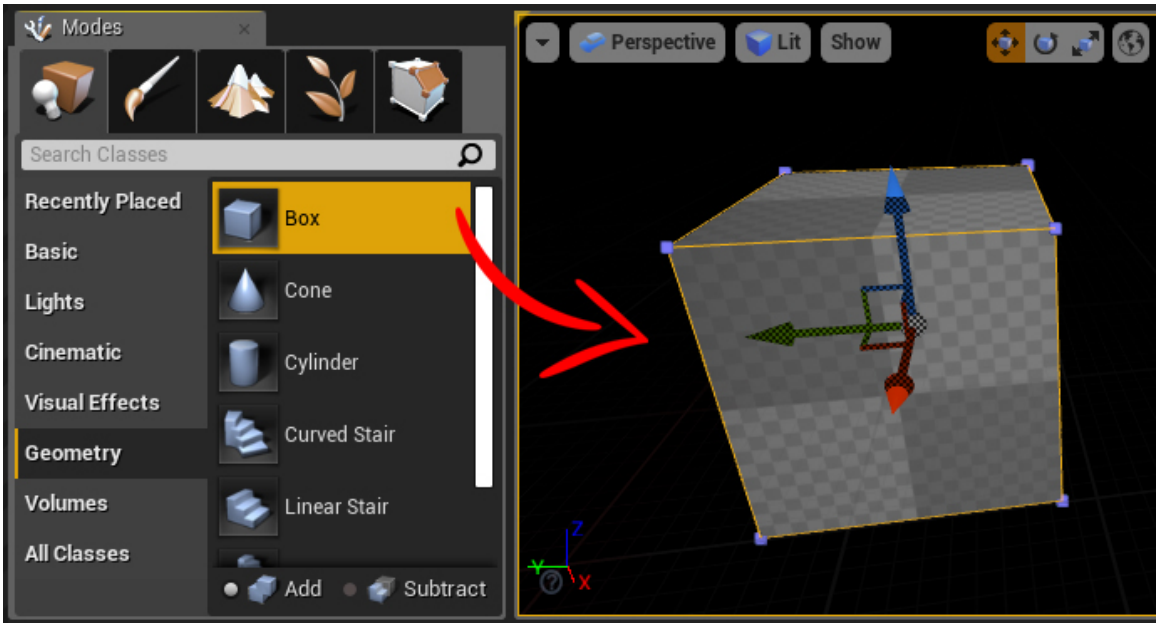
We need a ground plane, a floor for the player to stand and walk on. For ground plane you can use one of the following:

- BSP Brush
- Static Mesh (3d model)

We'll use a BSP brush. Go to **Modes, Place** tab **(Shift +1)** and switch to **Geometry**:



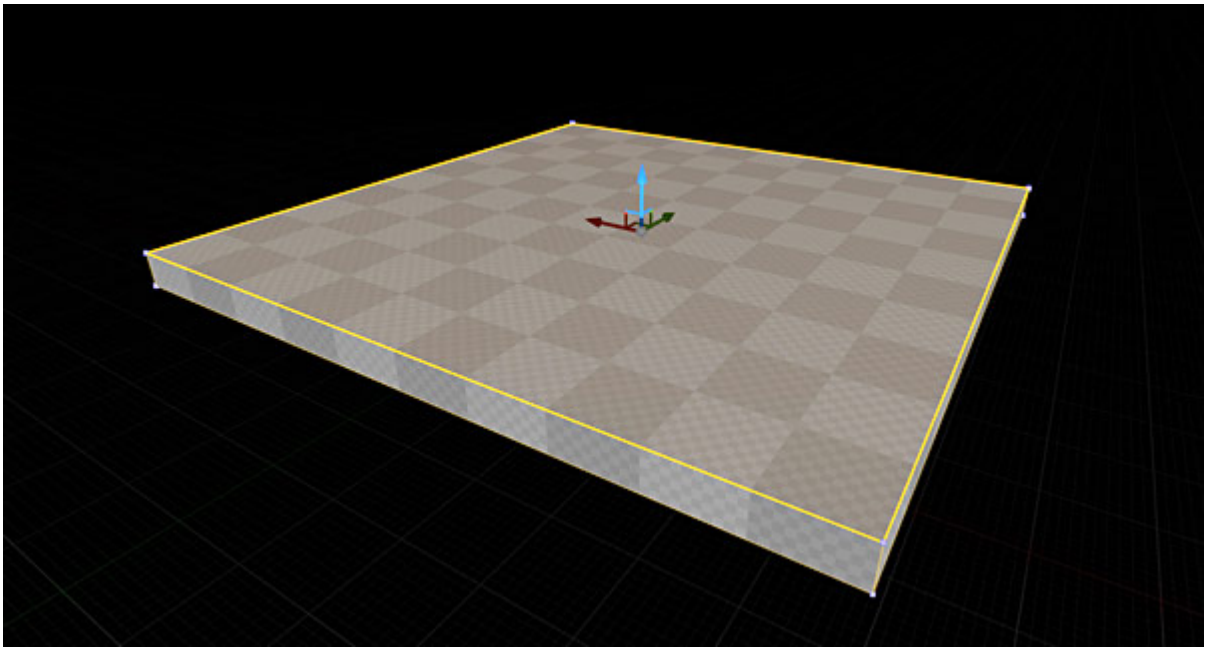
Left Click and drag a BSP Box into perspective viewport:



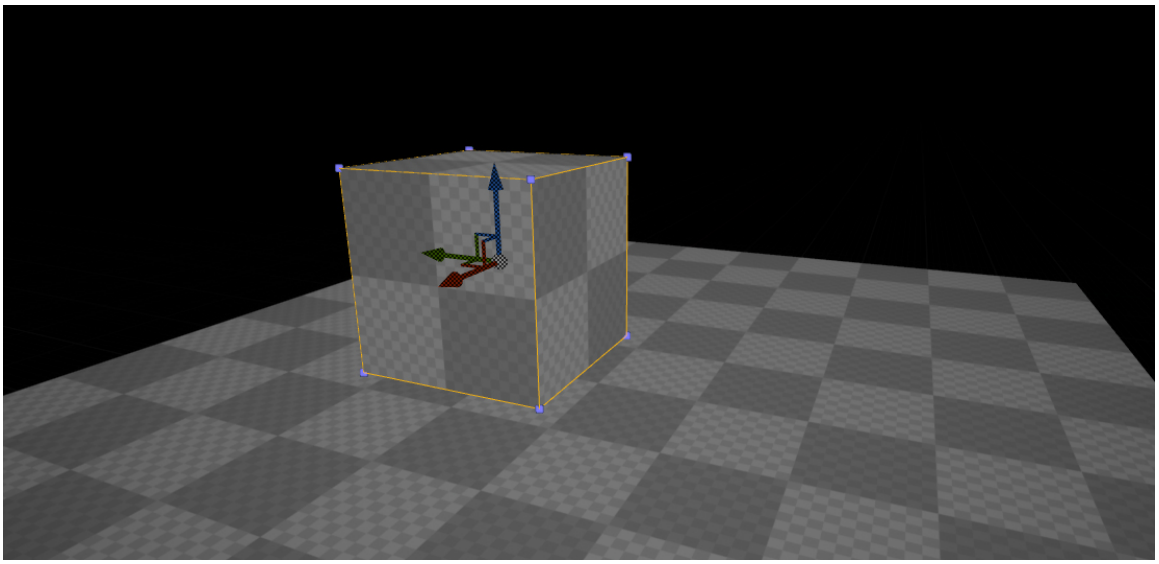
With BSP brush selected, go to **Details** panel and resize the brush to following values:

- X = 1000
- Y = 1000
- Z = 50

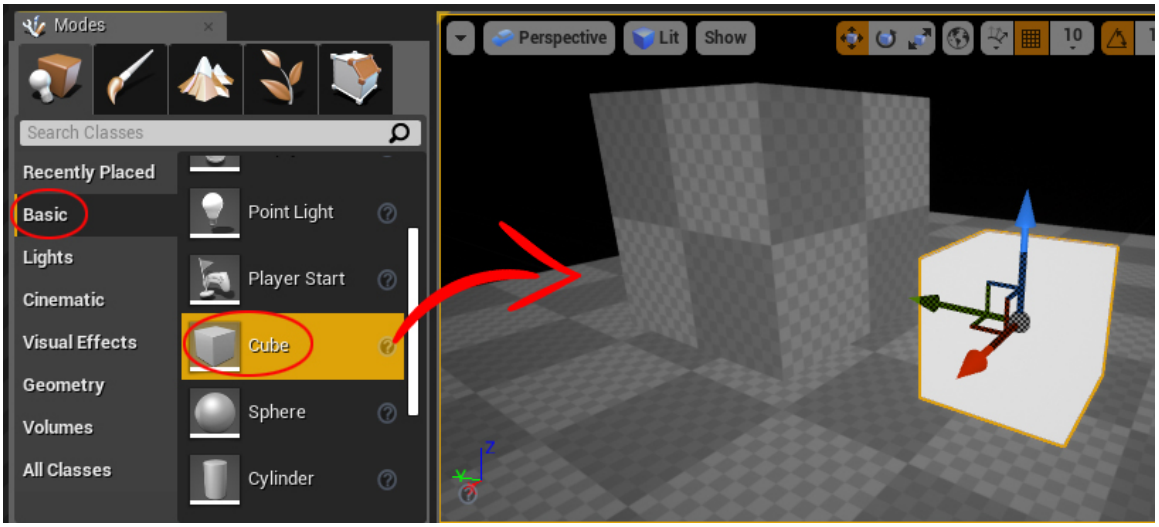
We now have a ground plane:



Let's also insert a BSP box and Static mesh on top of the ground plane. This way we have some objects inside our level. Left click and drag another BSP Box into perspective viewport and position it on top of the ground plane:



Then let's also add a simple Static Mesh Cube. In **Modes, Place** tab, switch to **Basic** and insert a **Cube** Static mesh into the level:



STEP 4: CHARACTER REFERENCE SCALE

You want to build your worlds to correct scale and proportion. No object or architecture should be too large or too small otherwise it will ruin the illusion of your environment very quickly.

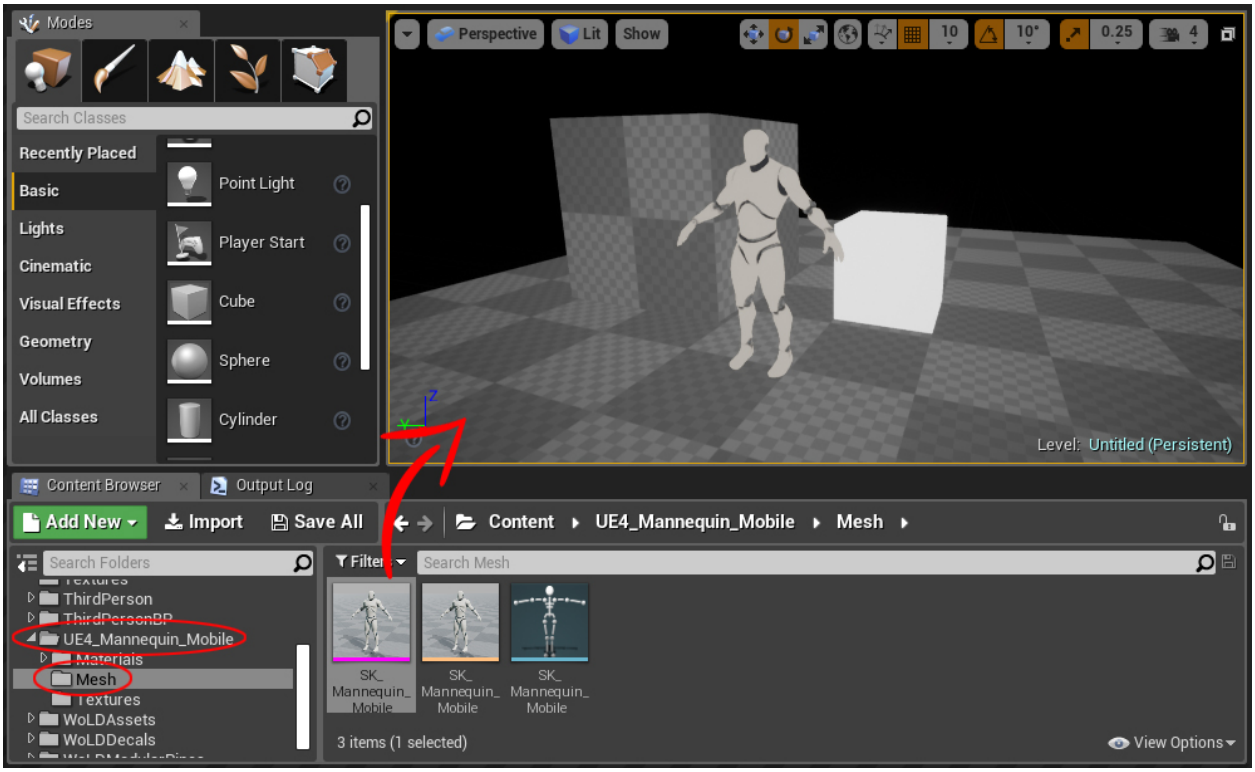
To do this we'll insert a human reference with the same dimensions as a player model in game. It will be used to judge scale as you build.

Since we aren't creating this level for any specific game that contains a specific collision model for the player character, we'll use UE4 base dimensions.

Base Character Dimensions in UE4 are:

- Height = 180 cm/uu
- Width/Depth = 60 cm/uu

Let's insert the Mannequin Mobile as the human reference scale. Left click and drag the **SK_Mannequin_Mobile** from the **UE4_Mannequin_Mobile/Mesh** folder into the level's ground plane.

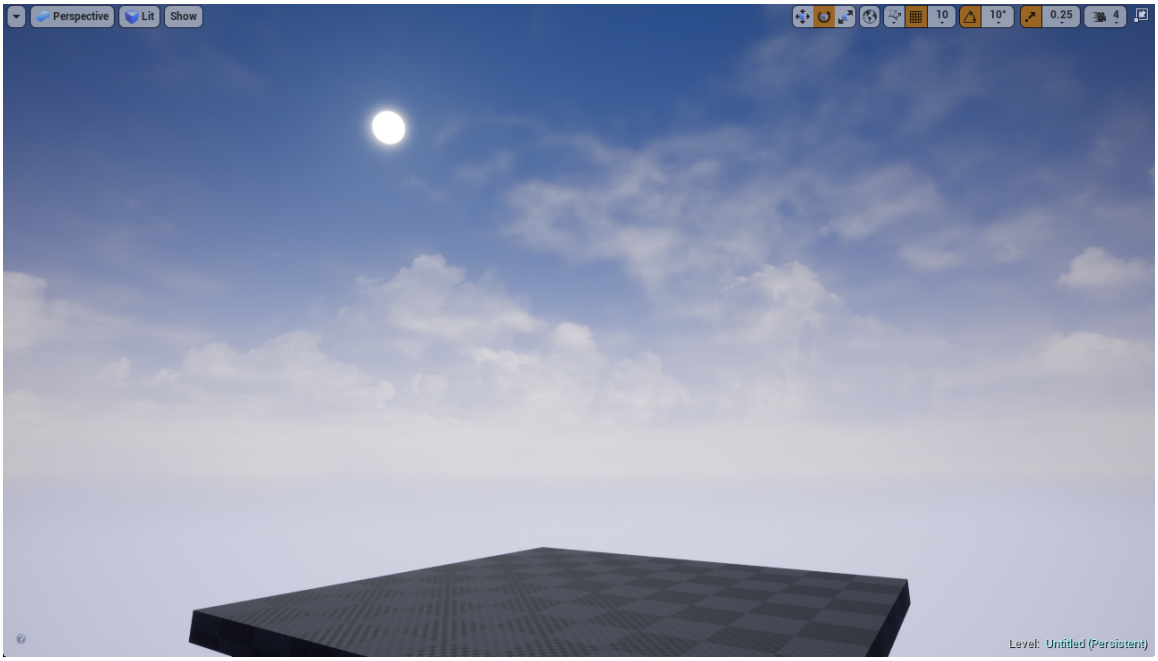


Downloading and adding the Mannequin Mobile is shown here in this tutorial: https://www.youtube.com/watch?v=tL9ba98H2_Q

Scale Reference Options: for human reference scale you can use other options such as BSP cube, Skeletal Mesh provided with UE4 in Engine Content folder, Mannequin Mobile (what we are using) or a custom character mesh.

STEP 5: BP_SKY_SPHERE

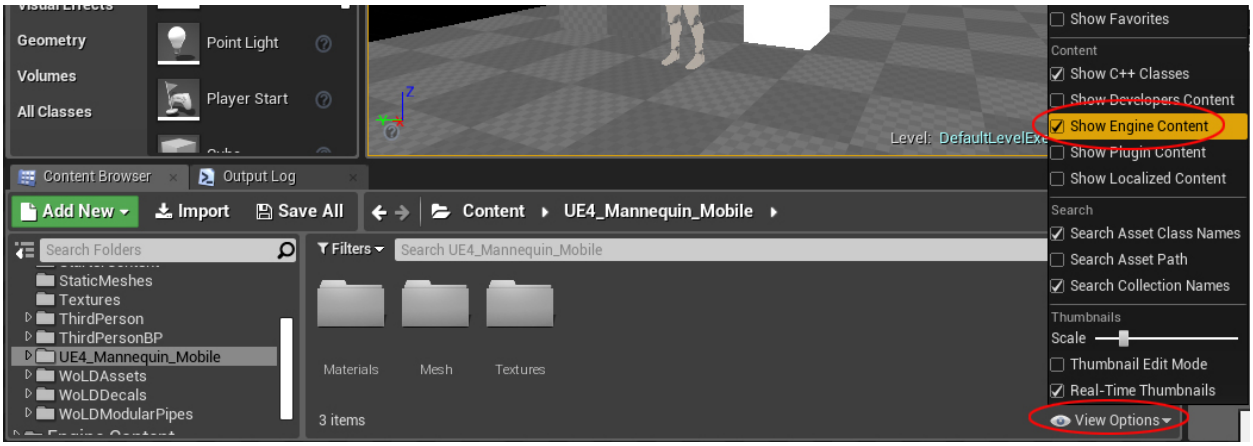
BP_Sky_Sphere is a blueprint asset that was created by Epic Games and included with every project you create. It's the same sky you see in the Default level:



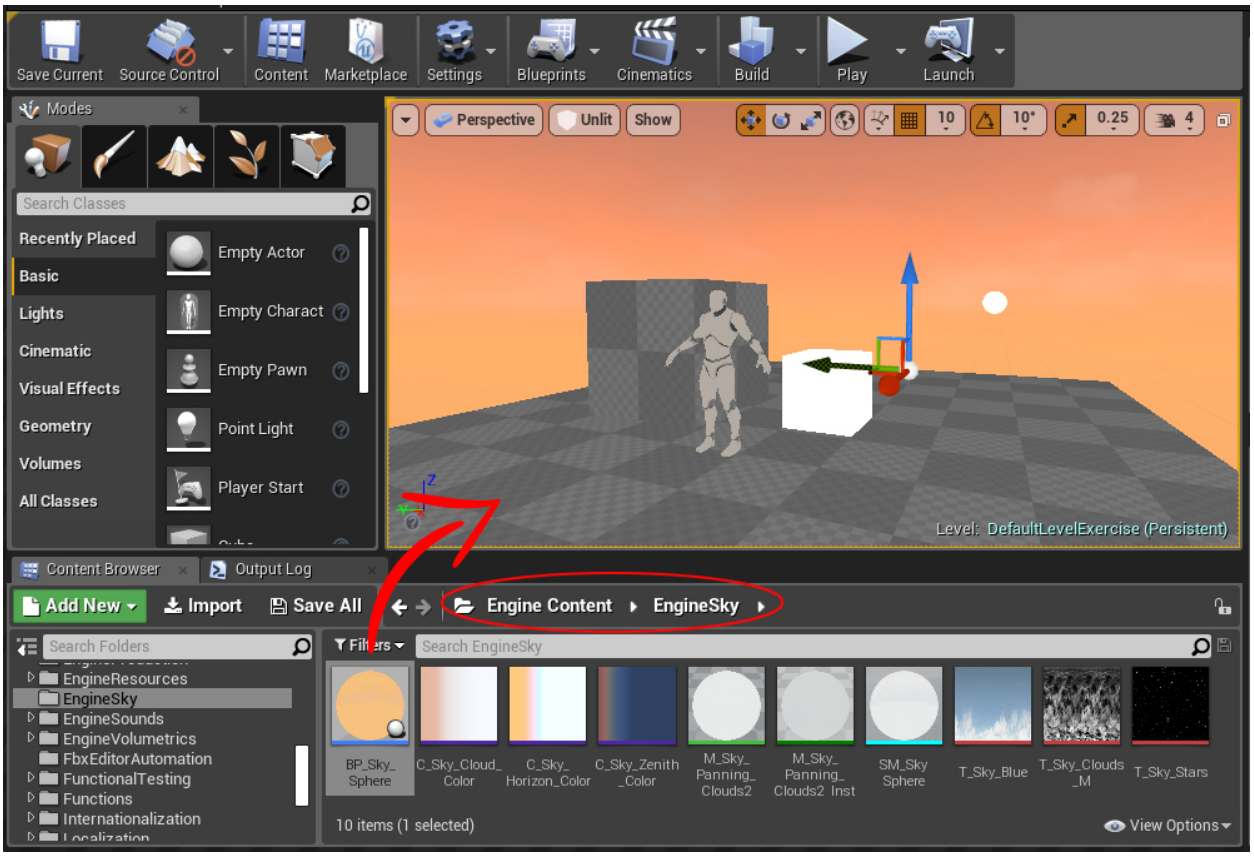
BP_Sky_Sphere is great to use for any project due to how flexible it is.

The sky contains moving clouds, works with Directional Light to change color of the sky based on time of day and has additional properties to tweak the sky texture.

In **Content Browser**, go to **View Options,** enable **Show Engine Content**:



Then go into **Engine Content → EngineSky** folder and insert **BP_Sky_Sphere** into your level:



It won't look like it does in the default level yet. But we'll update it in a bit.

You can disable **Show Engine Content** from the Content Browser.

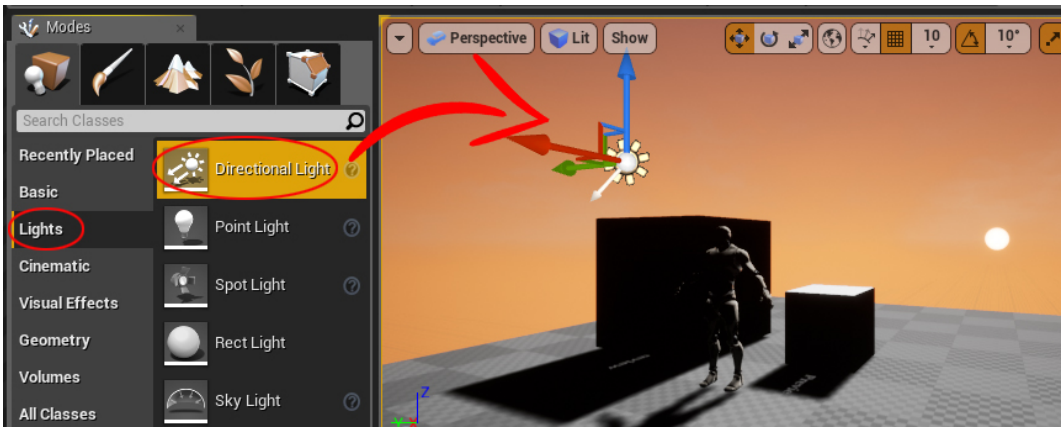
STEP 6: DIRECTIONAL LIGHT & BP_SKY_SPHERE

Directional Light is the sun light that will illuminate the entire environment.

We need to insert a Directional Light and connect BP_Sky_Sphere with it.

Go to **Modes Panel: Place** tab (Shift+1) and switch to **Lights**.

Left click and drag the Directional Light actor into the perspective viewport. You'll see Directional Light take effect:

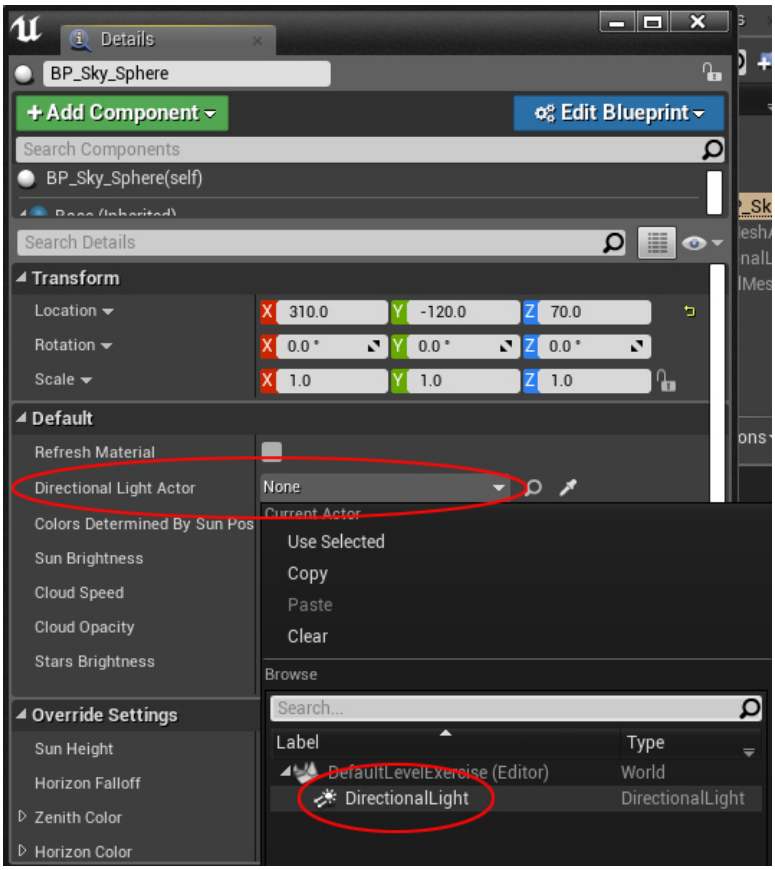


If you do not, switch to Lit Mode (Alt+4).

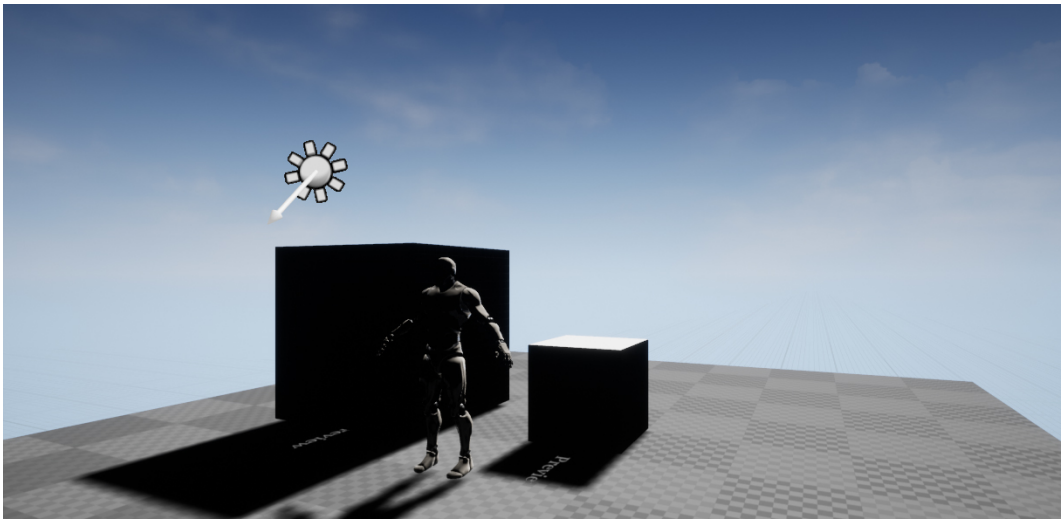
- **Alt + 4** = Lit Mode

We need to connect the Directional Light to work with the BP_Sky_Sphere. This functionality has been scripted into the BP_Sky_Sphere blueprint.

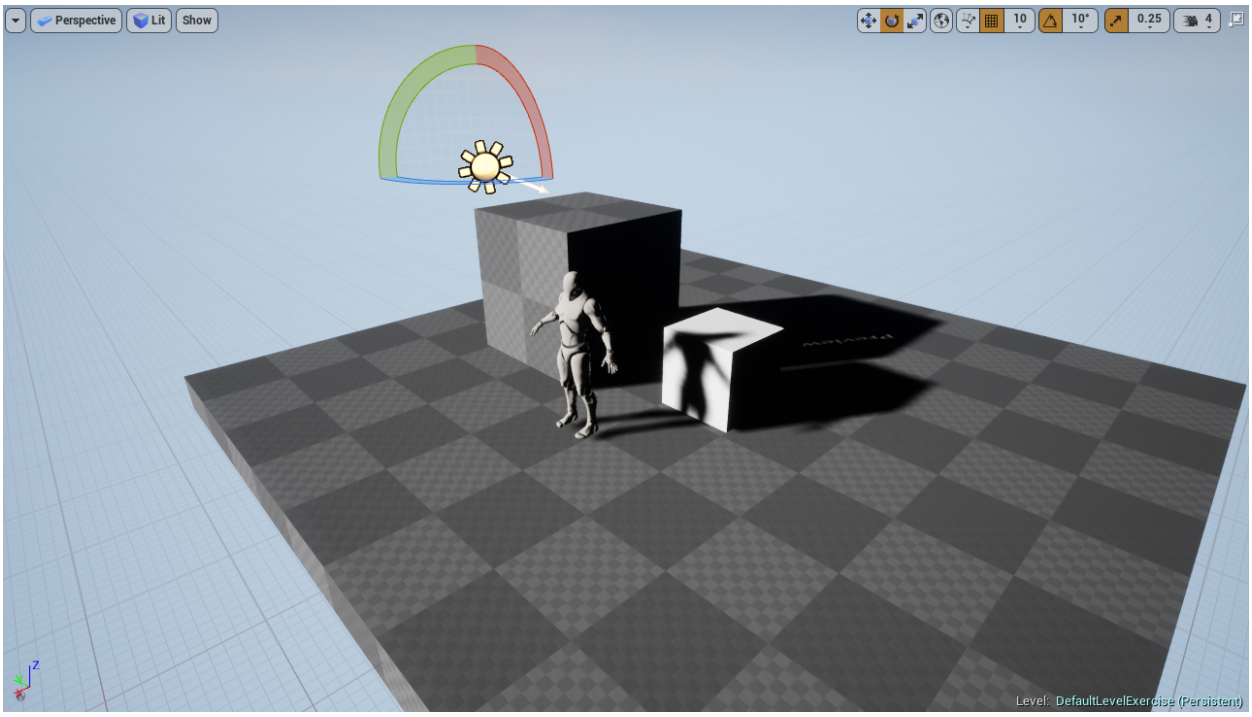
Select the BP_Sky_Sphere and go to Details panel. Under Directional Light Actor use the drop down menu to select the Directional Light:



The BP_Sky_Sphere texture will now update:

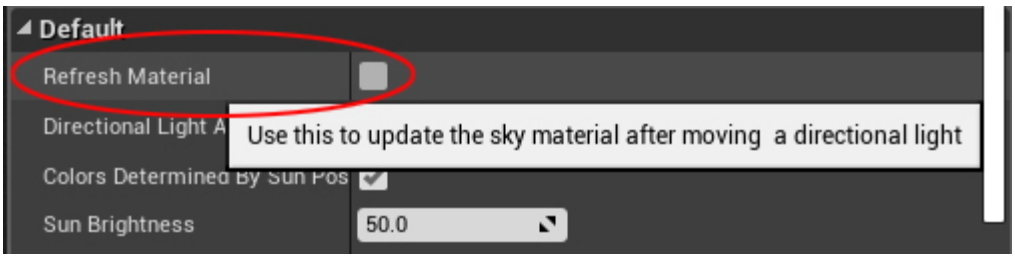


Select the Directional Light and rotate (E) to point the light towards the ground plane for any angle you want:



Depending on the angel of Directional Light, texture of BP_Sky_Sphere will update. You can set the sky texture for early morning, afternoon, evening and even night.

To see this update you'll need to select BP_Sky_Sphere and click on **Refresh Material** in Details panel:



Select the Directional Light and in Details panel focus on adjusting any of the following properties:

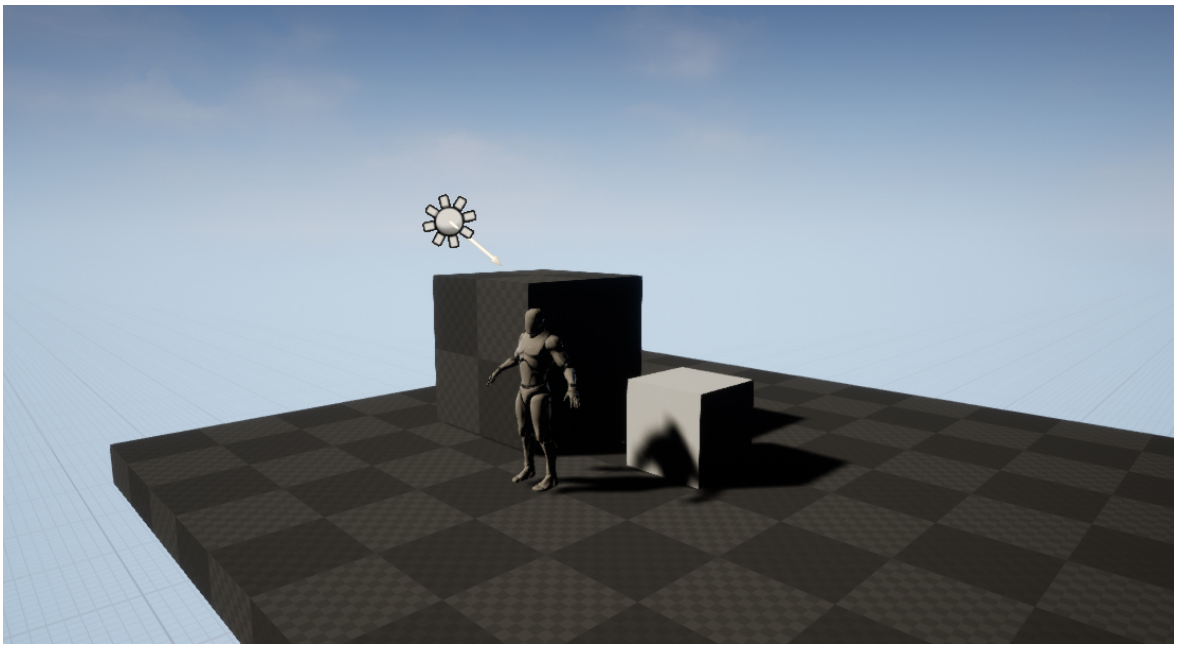
- **Intensity**: strength of the light
- **Light Color**: sets the color of your light

Recommended value for correct **Directional Light intensity** is **3.14**:

Choose any **Light Color** you want by clicking on the color bar to open the Color Picker:

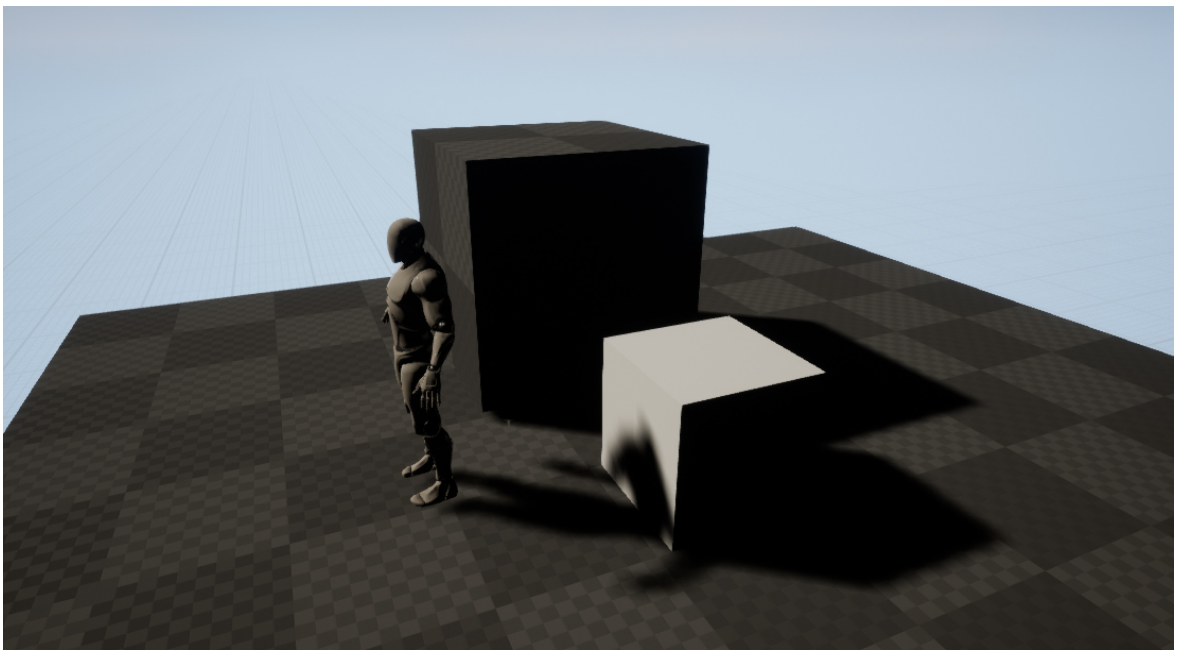
Color you choose should reflect the time of day you want and should match the sky. Once you've chosen the color, click OK.

Here is the updated scene with a lighter orange/peach light color and 3.14 light intensity:



STEP 7: SKY LIGHT

The sunlight is working but you'll notice the shadows on the ground and in the back of the boxes are very dark:

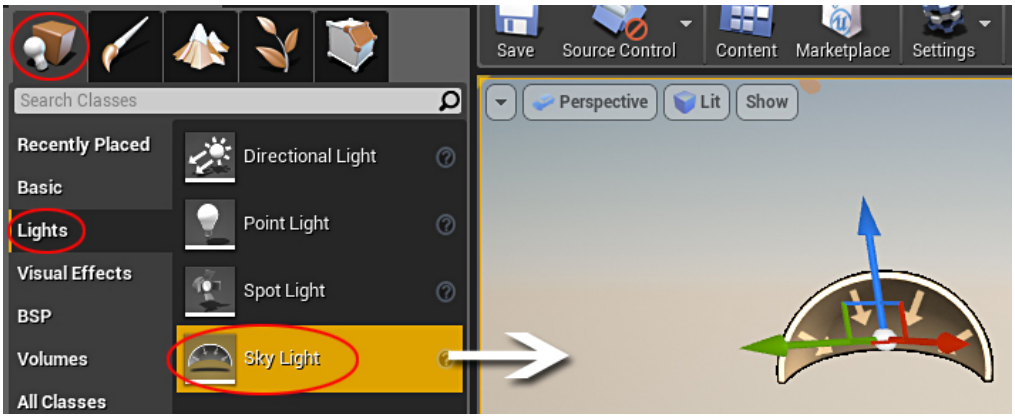


In real world you would have bounced light coming from the sky and from the ground and it would never be this dark. To fix this, we need a Sky Light.

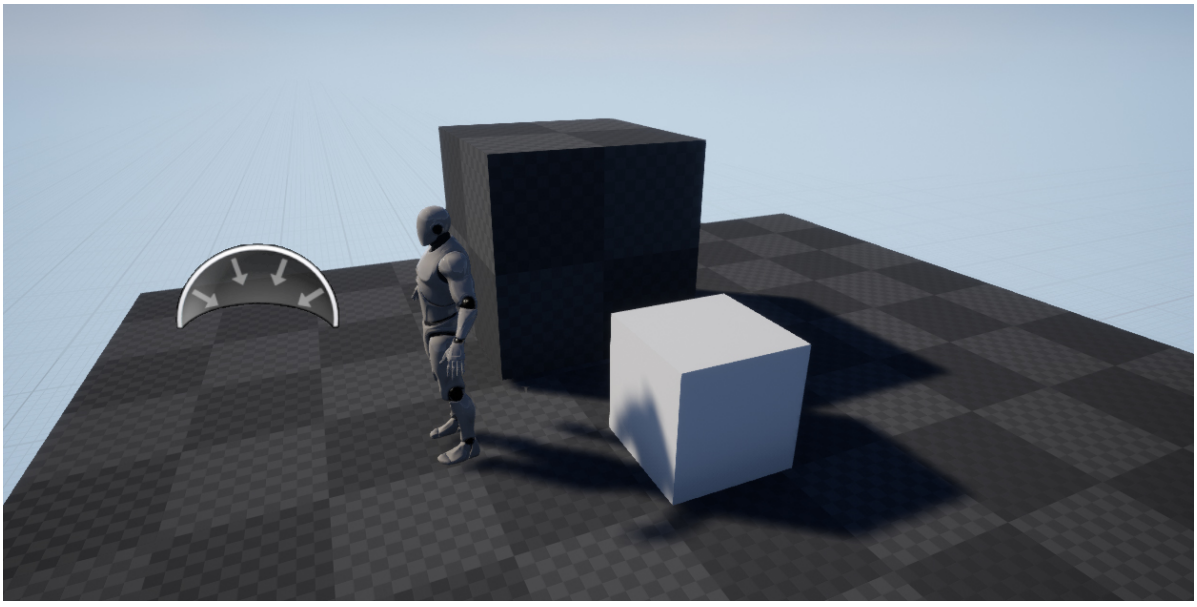
Sky Light is an additional light actor that illuminates indirectly lit areas; areas that aren't being directly lit by Directional Light.

Important: Sky Light will work only if you have a Sky Sphere. This can be the BP_Sky_Sphere we inserted earlier, it can be a static sky sphere using a simple Static Mesh with an unlit material or another option without a Sky Sphere is to use Atmospheric Fog.

Go to **Modes Panel: Place** tab (Shift+1) and switch to **Lights**. Left click and drag the Sky Light into your level:

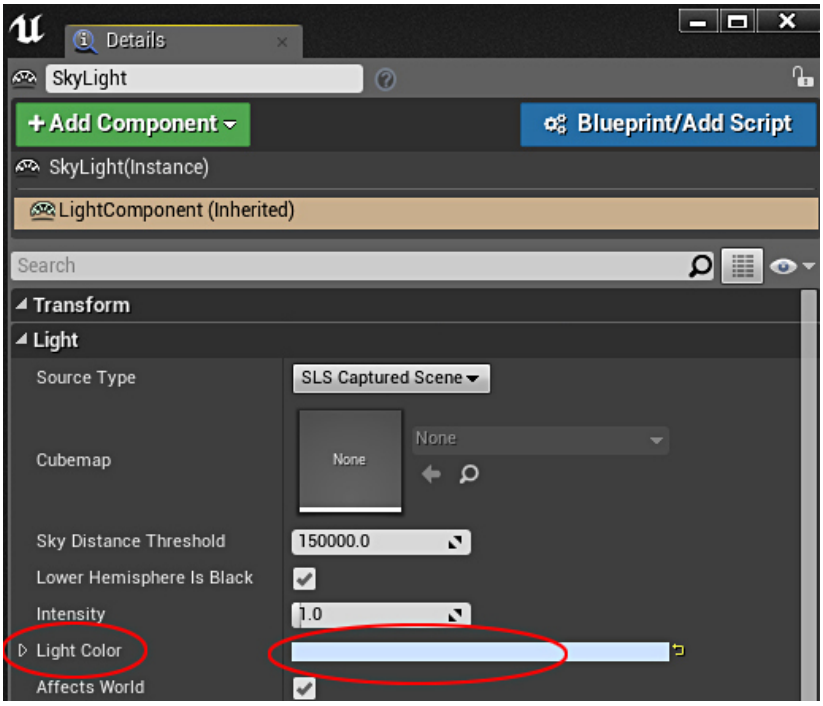


The areas in shadow are no longer black and it looks more realistic with indirectly lit areas receiving a blue hue from the Sky Light:



Remember: for a Sky Light to work you need to have either a Sky Sphere or Atmospheric Fog inserted into your level.

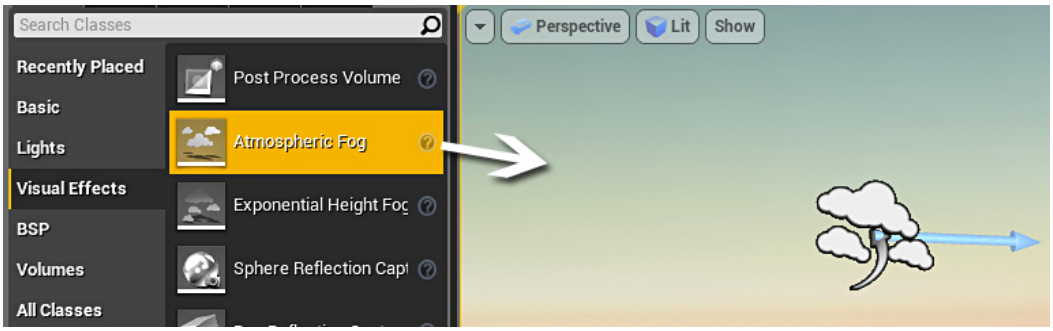
You can change Sky Light's indirect lit areas with different color or adjust Intensity by selecting the **Sky Light** and in Details panel adjust **Light Color** or **Intensity**:



STEP 8: INSERTING ATMOSPHERIC FOG

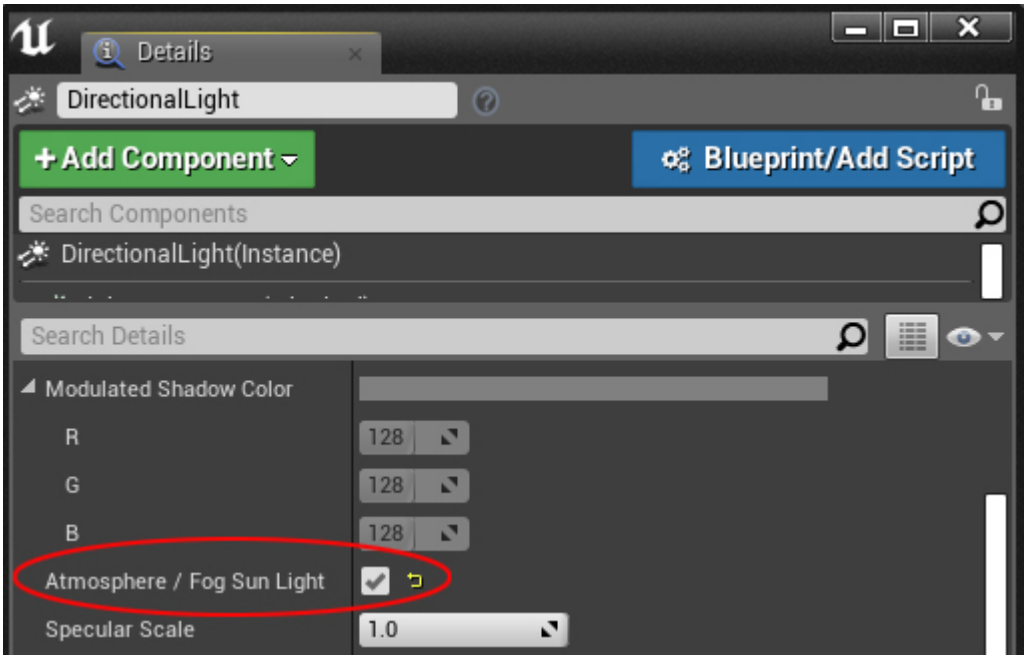
Let's insert Atmospheric Fog. It will give more realistic atmosphere in the level.

Go to **Modes, Place** tab and under **Visual Effects**, left click and drag **Atmospheric Fog** into perspective viewport:

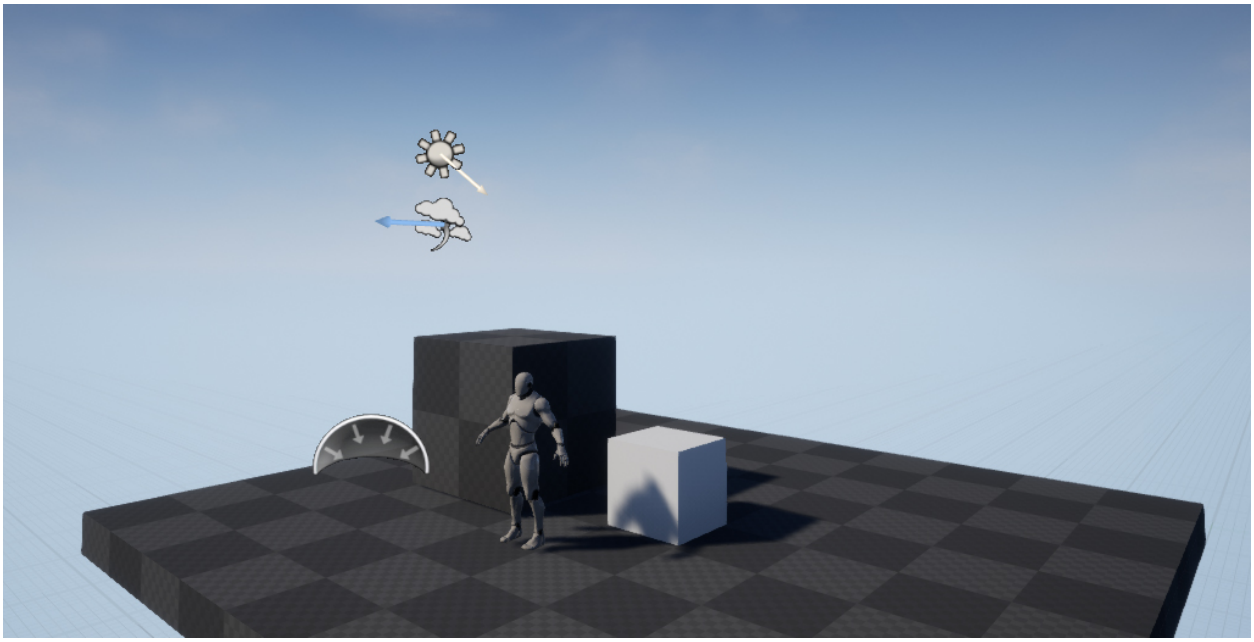


We aren't going to change any properties for Atmospheric Fog but we do need to link it with the Directional Light.

Select the Directional Light and in Details panel, enable **Atmospheric / Fog Sun Light**:



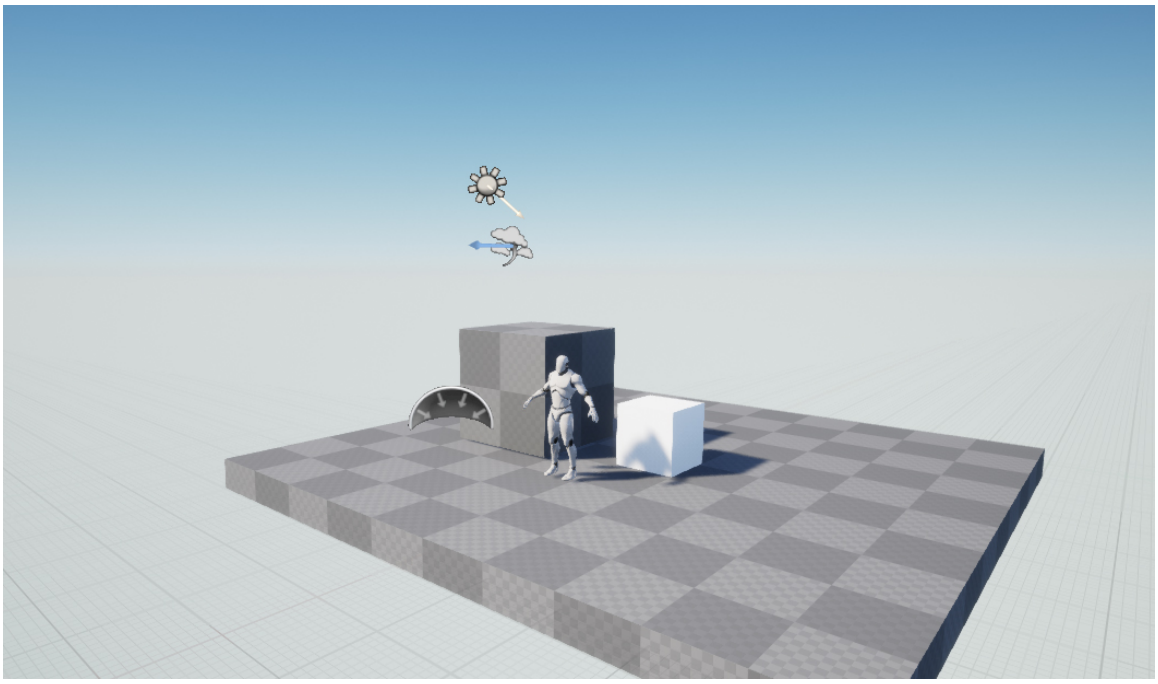
We now have an Atmospheric Fog and the Directional Light working together:



If you don't see any fog in your level, make sure to enable its visibility using **Show** drop down menu:

- **Alt + F** = Toggle Fog On/Off

No BP_Sky_Sphere: If you didn't want to use BP_Sky_Sphere, you can use Atmospheric Fog to give you an illusion of a skybox as seen below (image below contains no Sky Sphere):

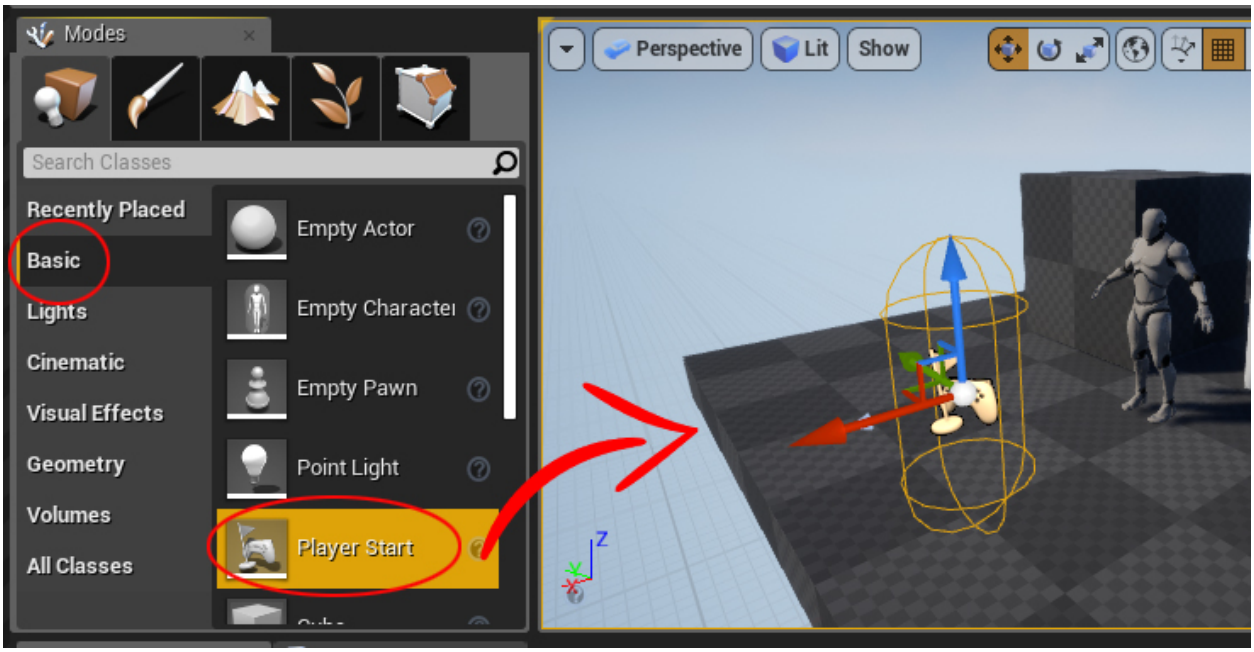


STEP 9: INSERTING A PLAYER START

Player Start defines a spawning location for the player inside the level.

Every level should have at least one player start.

Go to **Modes, Place** tab and under **Basic**, left click and drag **Player Start** into perspective viewport, onto the ground plane:



Now when play testing using various **Play in Editor** options, you'll spawn from this player start location.

STEP 10: INSERTING REFLECTION CAPTURE ACTOR

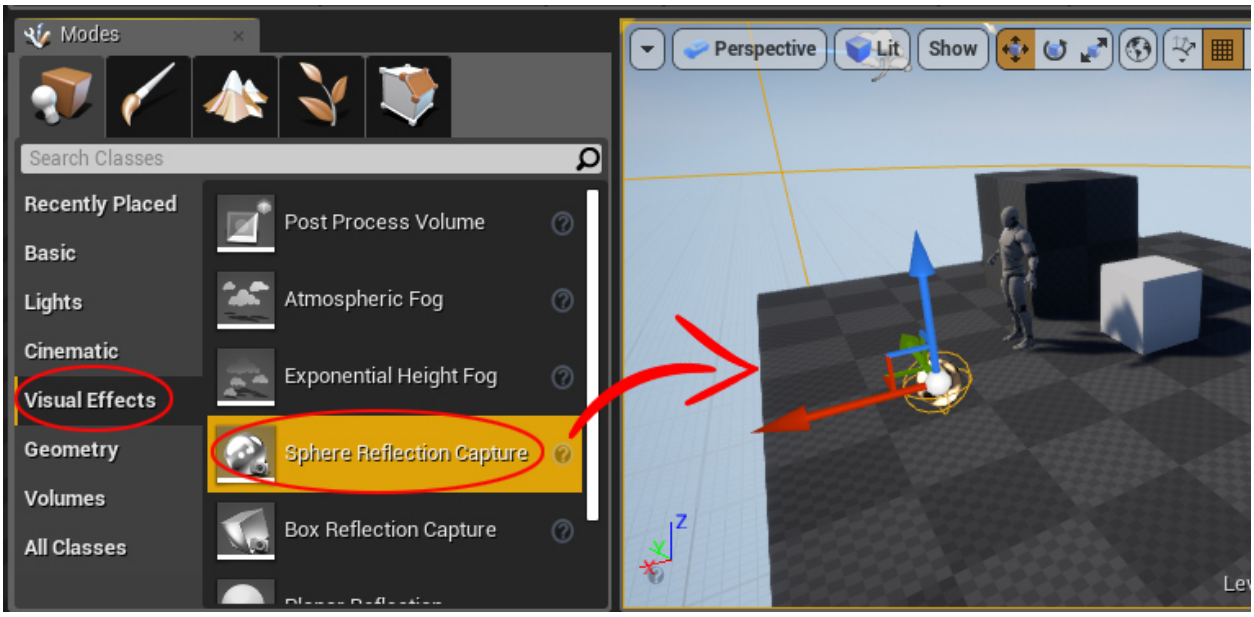
Every level needs at least one reflection capture actor to reflect the environment around it. This helps certain surfaces to accurately display the world such as water, glass, metals or other shiny materials. There are **3 reflection capture actors** you can use:

1. Box
2. Planar
3. Sphere

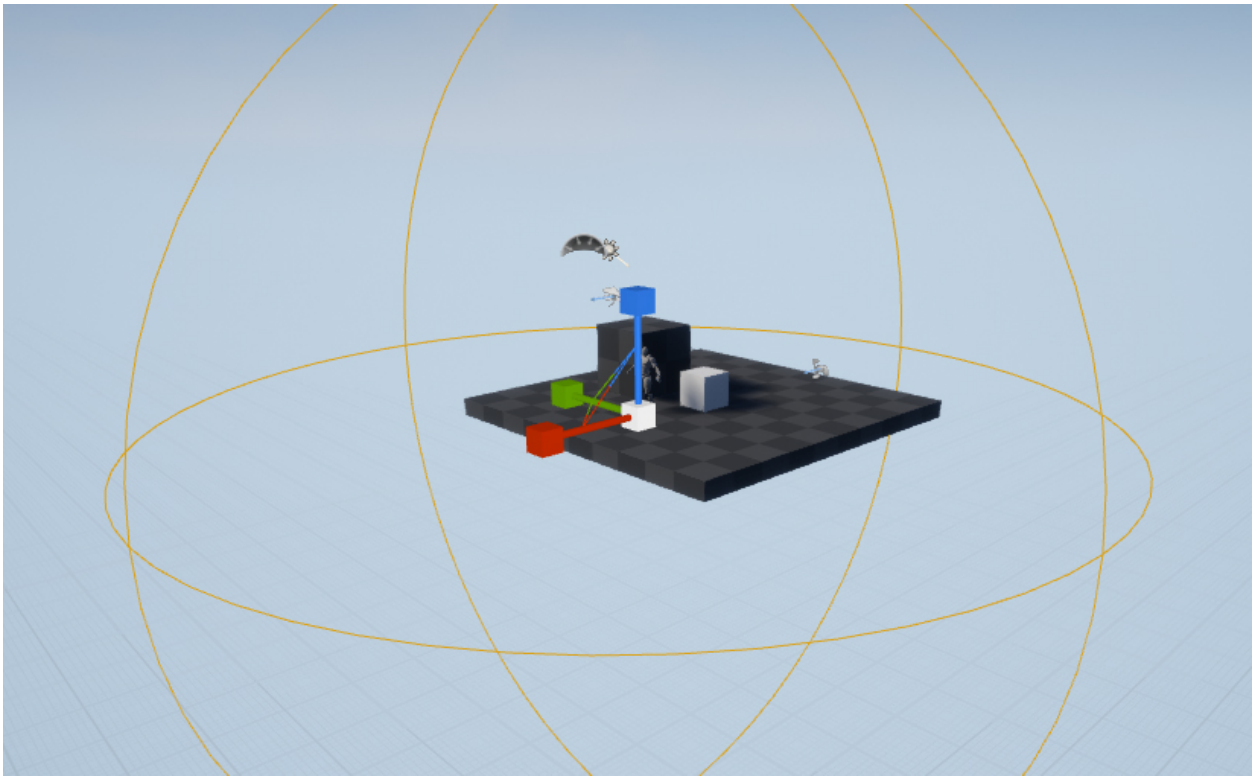
Sphere Reflection Capture is the most flexible and primarily used actor for capturing reflections.

Many of these can be placed around your level with almost no performance cost because they are calculated before run time.

Go to **Modes** panel, **Place** tab and under **Visual Effects** left click and drag **Sphere Reflection Capture** actor into perspective viewport:



Position it so the sphere of influence surrounds the entire ground plane:



If you need to increase or decrease this influence, use the scale tool (R).

STEP 11: AUTO-EXPOSURE/EYE ADAPTATION

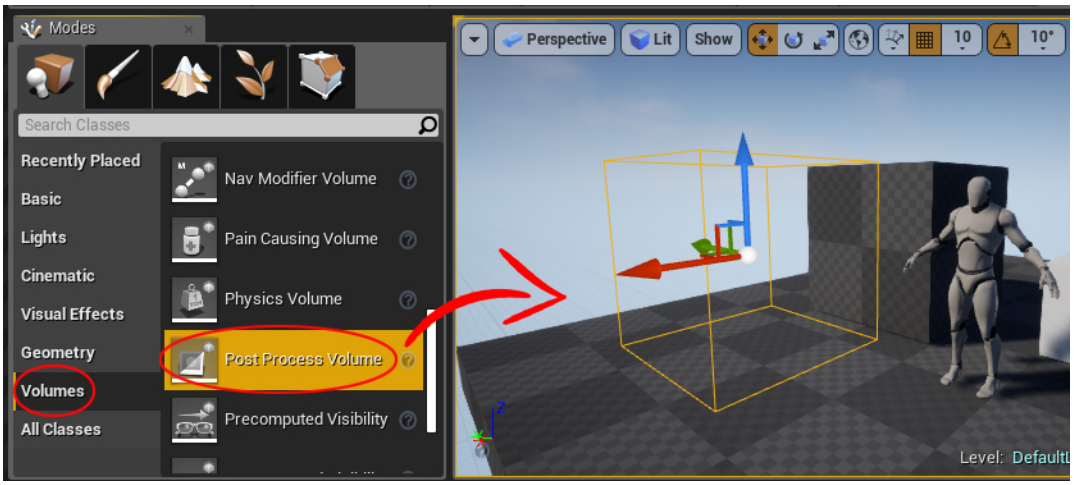
Auto Exposure or Eye Adaptation is how your eyes naturally adjust when you walk from bright into dark or from dark into bright environments.

In UE4 this happens automatically in all levels unless you control it.

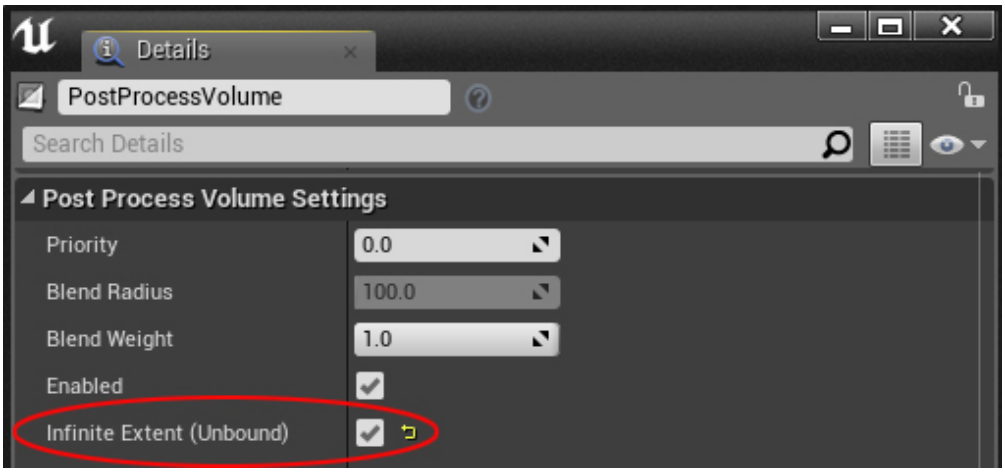
Auto-Exposure is a great feature but it's very distracting when you are just beginning to create the environment and aren't focused on lighting.

I disable default exposure (eye adaptation) in the beginning of all projects.

Go to **Modes** panel, **Place** tab and under **Volumes**, insert **Post Process Volume** into your level:



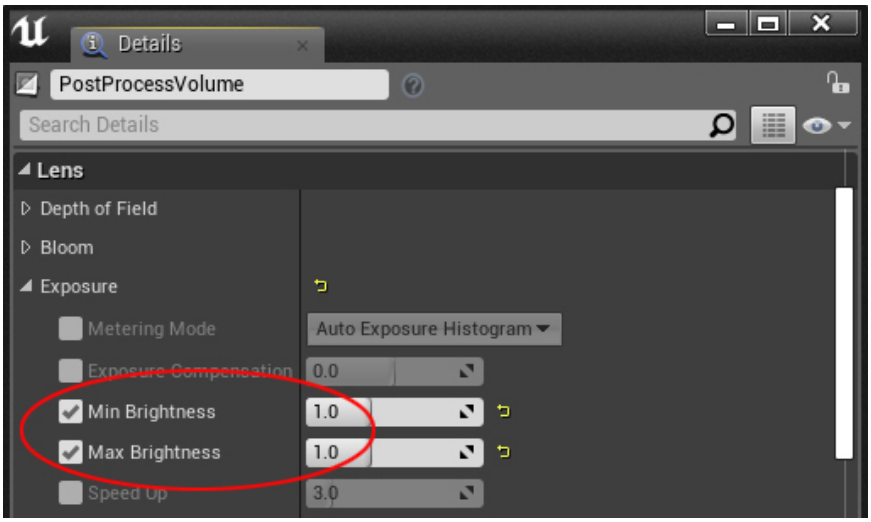
In Details panel for **Post Process**, search for **unbound** and enable **Infinite Extent (Unbound)**:



This will make Post Process Volume universal, so you don't have to be inside this volume for it to take effect.

Under **Lens: Exposure**, set **Min / Max Brightness** to **1**:

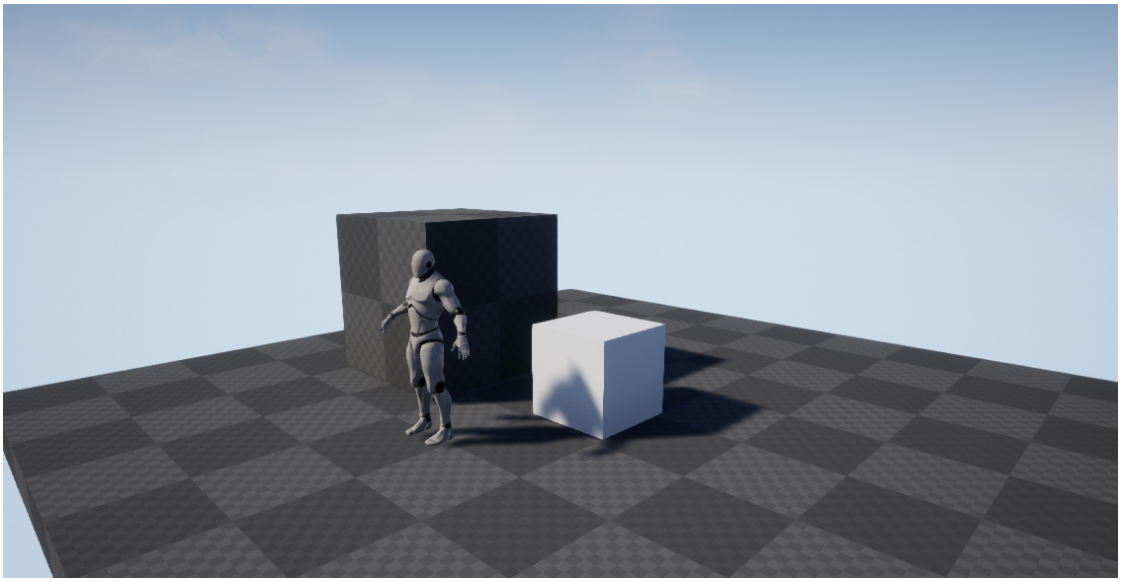
- **Min Exposure**: 1.0
- **Max Exposure**: 1.0



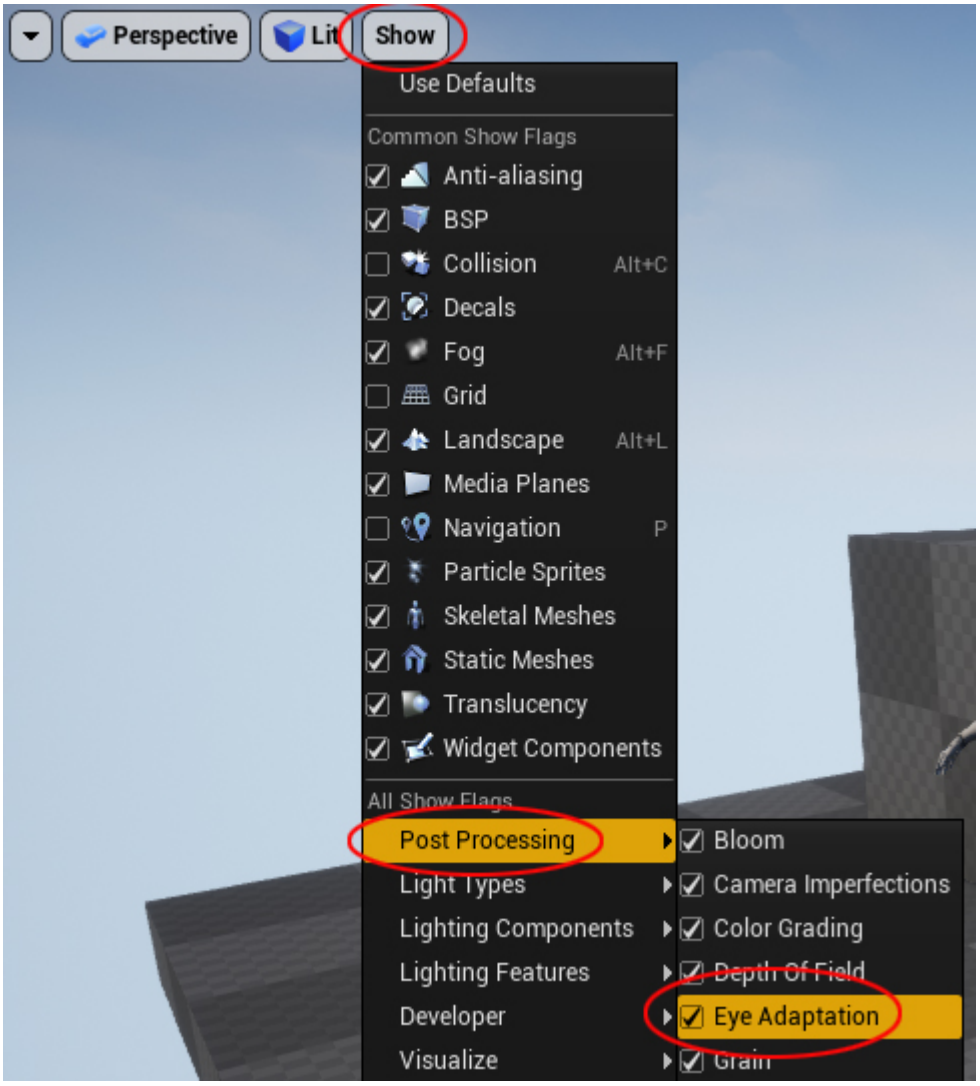
This will disable **Exposure** both in-game and in editor.

Once you begin lighting the level and work on visuals, you'll want to adjust Min/Max Exposure values to fit your environment art direction.

Here is what we have so far:



You can disable **Eye Adaptation** in editor viewport only but not in-game. In perspective viewport, go to **Show → Post Process → Eye Adaptation** and disable it:

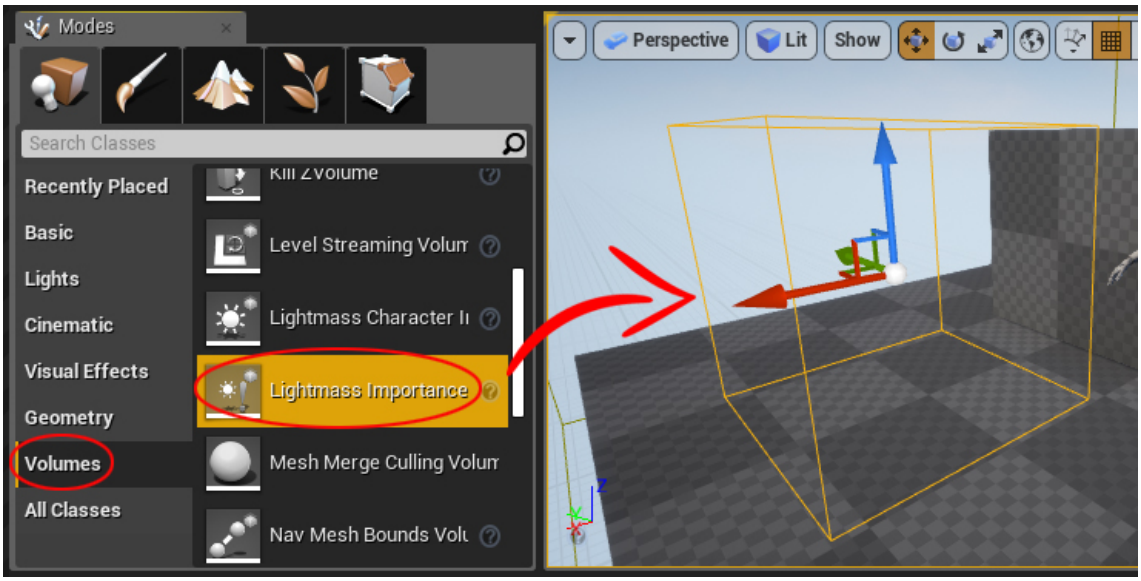


More on Auto-Exposure: https://docs.unrealengine.com/en-us/Engine/Rendering/PostProcessEffects/AutomaticExposure

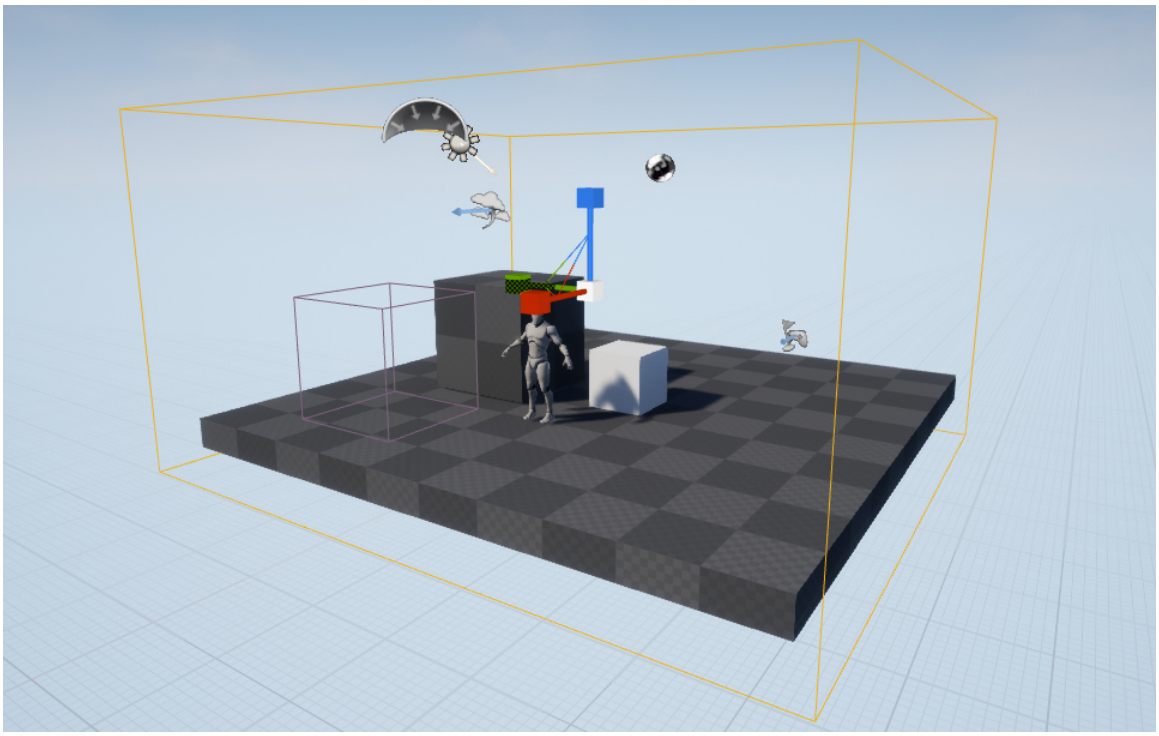
STEP 12: LIGHTMASS IMPORTANCE VOLUME

Lightmass Importance Volume tells Unreal Engine the areas to focus its lighting calculations in. If you have a large level but only a small part of it is playable, surround this area with Lightmass Importance Volume.

Go to **Modes** panel, **Place** tab and under **Volumes**, insert **Lightmass Importance Volume** into your level:



Use Geometry Editing Mode (Shift+5) or scale (R) tool to resize the volume:

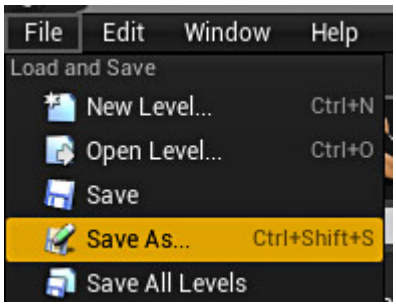


Surround important (playable) areas of your level inside this volume. In the beginning it will probably be just a small section.

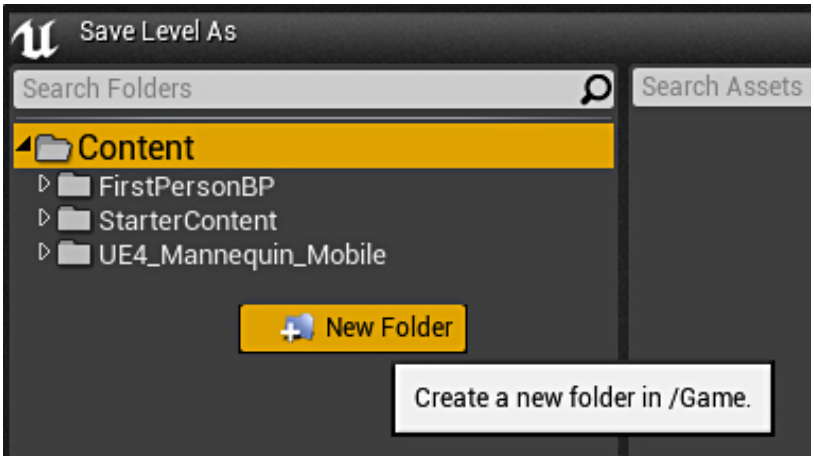
Note: you can have more than one Lightmass Importance Volume.

STEP 13: SAVING YOUR LEVEL

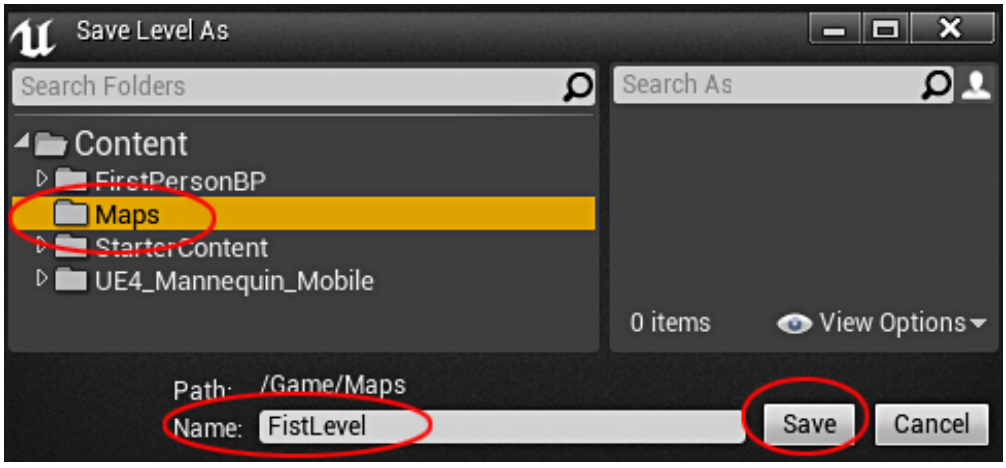
Let's save before we lose any work. Go to **File → Save As**:



For better organization, create a new folder to save all the maps into. Right click and choose New Folder and name it **Maps**:



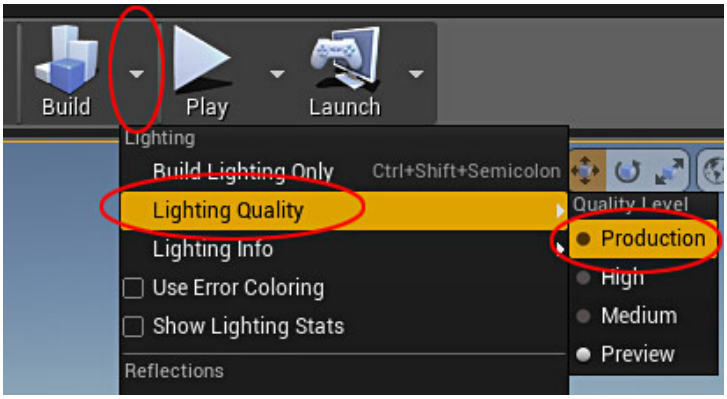
Select the Maps folder, name your level and click Save:



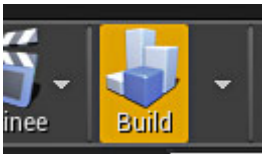
STEP 14: BUILDING LIGHTS AND BUILD ALL

Build option calculates and renders lighting, geometry and navigation. It shows how the environment really looks. Up until now, the level showed you lighting approximation or preview lighting.

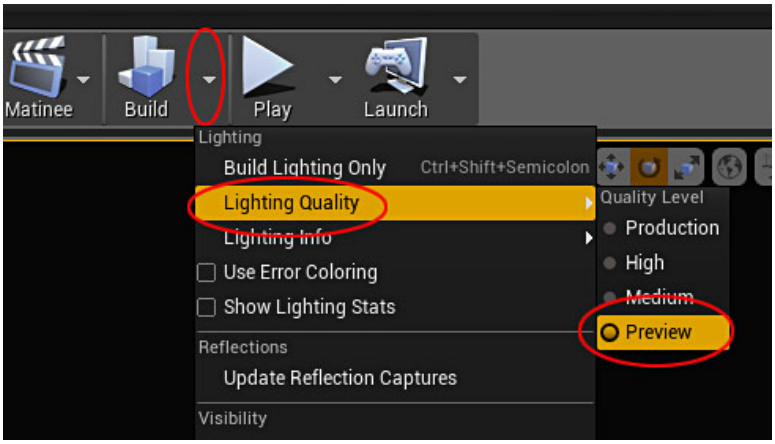
Let's **Build** the level on **Production**. Go up to Main Toolbar and use the drop down menu to set **Lighting Quality to Production**. This will build the lights on highest quality but this will also take the longest.



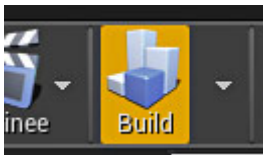
After setting to Production, click **Build**:



Preview Lighting Quality: throughout creation of your level, you'll want to test your lights using **Lighting Quality set to Preview**. It will allow you to continue working without having to wait for long build times. Eventually, at the very end you'll build using Production for best results. Use the drop down **Build option** and set **Lighting Quality to Preview**:

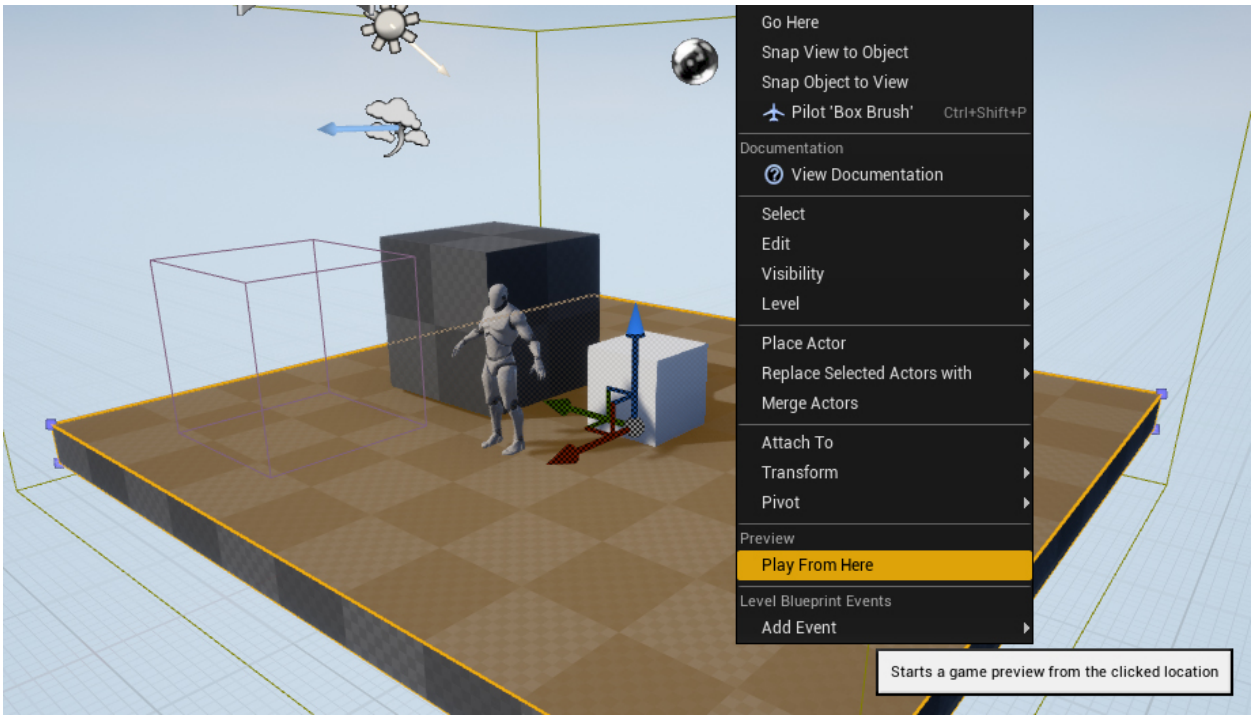


Then, you would click **Build**:

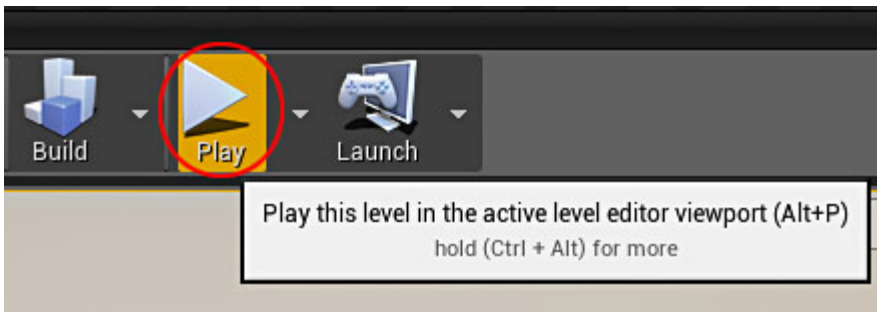


STEP 15: TESTING YOUR LEVEL

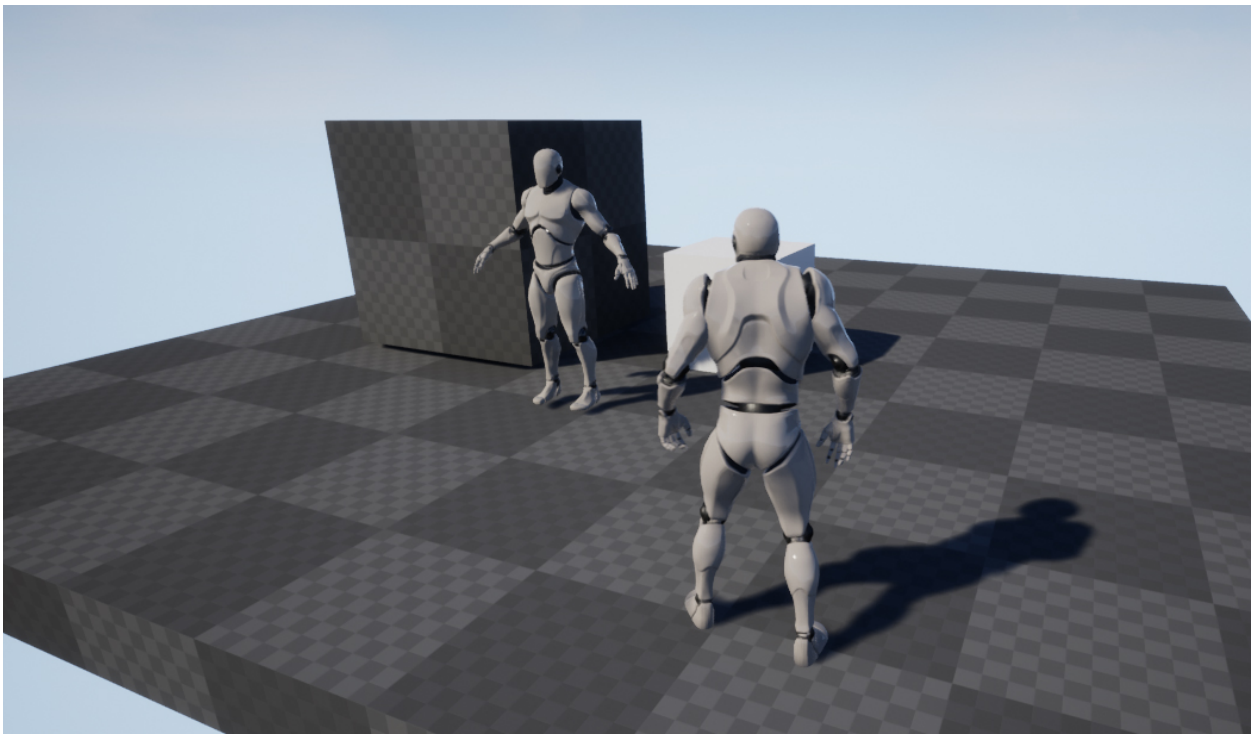
You can play test your level right from the editor. In perspective viewport, right click and choose **Play From Here**:



If you press on the **Play** icon at the Top Toolbar, it will spawn you from the Player Start actor inserted earlier:



You will spawn inside the level and as a player character depending on which game template you chose at the beginning of project set-up:



Press **Esc** to return to editor mode.

WHAT'S NEXT?

You have now created a very simple level complete with a player start, exterior lights, Static Mesh, BSP brush and fog. You controlled eye-adaptation/auto-exposure, built the level on production quality and saved the level.

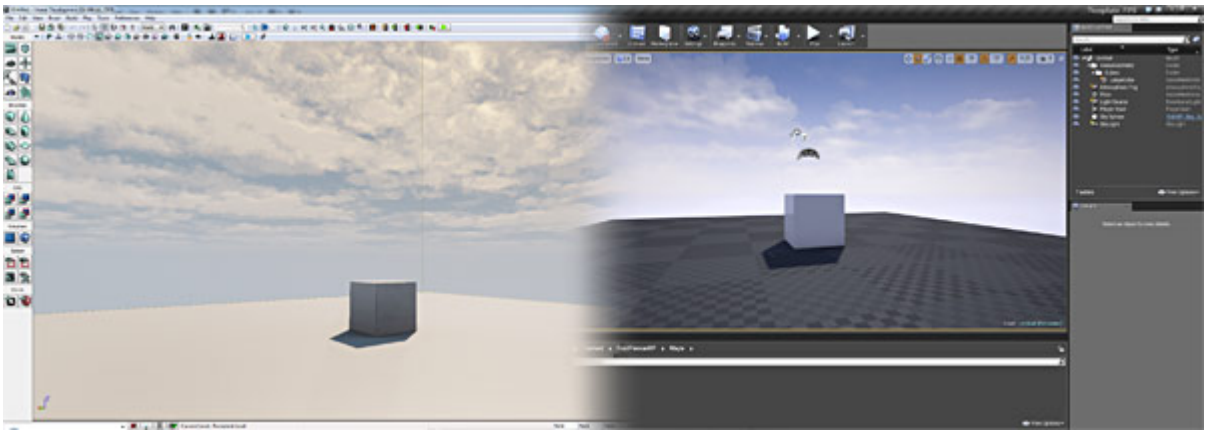
Of course there is a lot more to creating environments than what we just covered such as:

- Building with BSP brushes
- Building with Static Meshes
- Lighting your level using Directional Light, Sky Light, Point and Spot Lights
- Using and working with Post Process to define visual style and look of your level
- Adding audio

- Adding particle effects
- And so much more...

I recommend the following tutorial series that covers all these topics. You can get it here: **“UE4 Fundamentals Vol. 1: The Essential Beginner's Guide to Getting Started with Unreal® Engine 4”**.

SECTION 4: TRANSITIONING FROM UDK TO UE4



How do you transition from UDK to UE4?

If you have been working with UDK (UE3) for a few weeks, months or years it's probably a good time to switch to Unreal Engine 4. The updated tools, easier workflow and new improved tech are too appealing to pass up. And it is best time as any to make this transition.

It is now free (no more subscription fee) and constantly being updated, while UDK is no longer receiving updates. In fact, it's hard to even find any links to download it.

With the experience and knowledge you have from UDK, it will be easier to make the jump to UE4. Most of the functions, principles and workflows haven't changed, they just received an upgrade.

In this section you will learn how to make the quick transition and how to get started with UE4 when you are coming from UDK (UE3).

UE4 is very easy to get into, even for a complete beginner.

Here is a broad overview of just a few improvements and upgrades:

- Overall appearance of the editor has been changed and updated; the system interface is completely fluid and you can customize it to exactly how you want it
- Scripting/Coding is now in C++, no more Unreal Script

- Kismet visual scripting language has been replaced with a much more powerful tool called Blueprints
- Material creation is different than what you are used to in UDK - UE4 now uses PBR (Physically Based Rendering) workflow
- Lighting has been improved
- Bringing assets into UE4 is a simpler process
- Various game templates to get your game started
- Ability to create and manage your projects

And so much more that it would take an entire book just to outline all the new and improved features

Now, let's get started with how to quickly transition from UDK to UE4.

1. FREE UNREAL ENGINE 4

Unreal Engine 4 is now free. It started off as a monthly subscription service, but Epic has let that go, making it free for everyone. So just like UDK, you can download UE4 for free.

Tutorial: UE4 - How to Download and Install Unreal Engine 4

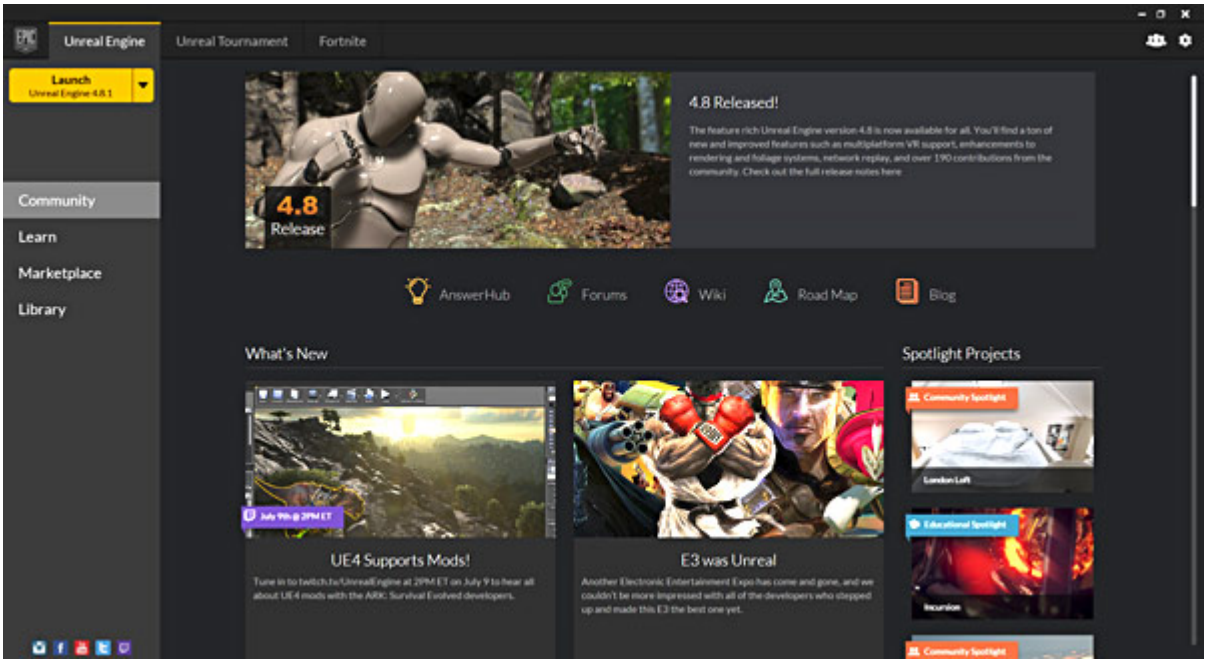
As for royalties and releasing your games with UE4, visit here for all the details.

2. UNREAL ENGINE/EPIC GAMES LAUNCHER

In UDK, you would download a specific version of the engine as an executable to install.

With **UE4**, the download and install works differently.

First, you download a small installation file that will set up Epic Games Launcher. This is a portal through which you download and install any version of the engine, old and new. Any future updates will be applied and downloaded through this Launcher.



So you no longer have to check if there is a new Unreal Engine 4 version out, the Launcher will keep everything up to date and notify you what's new.

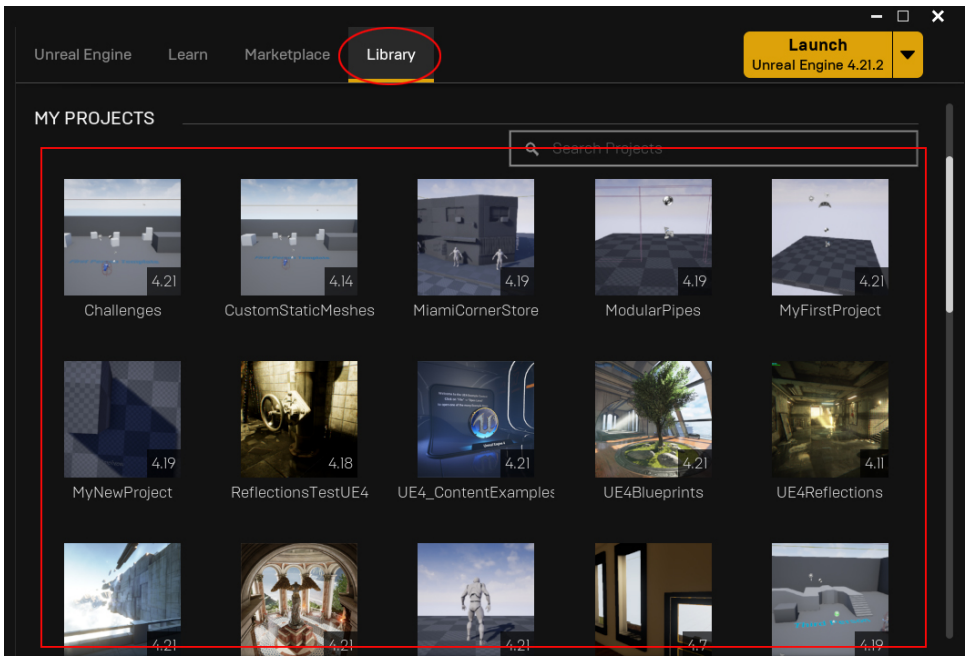
Through the Launcher you will also have the ability to create and manage your projects, purchase Marketplace content for your projects, download examples and game engine content.

3. PROJECT MANAGEMENT

In UDK, to create a project you would launch the editor and start working on your project. There was no way to keep various projects separate other than setting up folders ahead of time and saving all the maps, packages and content into it.

In UE4, you have to create a project you will be working on in order to open the editor. To work on another project, you have to launch another instance of the editor for that project.

All of your project will be under the **Library** tab:

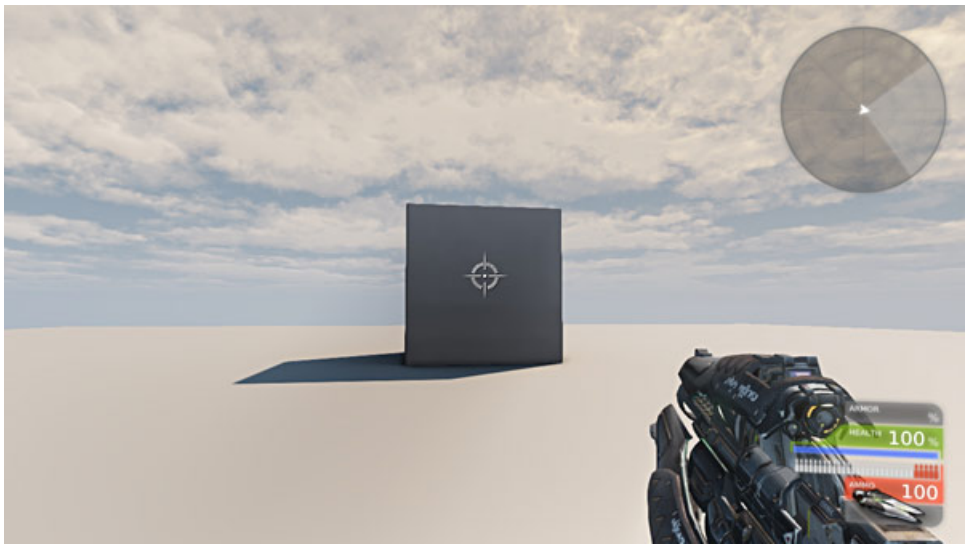


Before starting UE4 editor you would define a project, then launch the UE4 editor with all the files associated with it for that project.

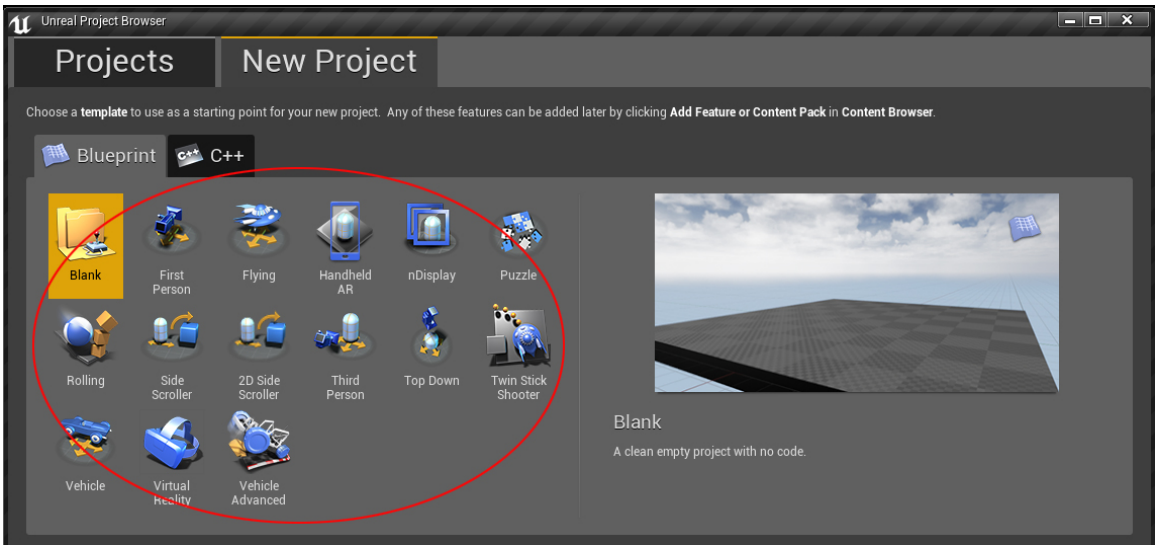
Tutorial: [UE4 - How to Create, Start and Open Your First New Project]

4. GAME TEMPLATES

In UDK, you would start with first person game mode. If you wanted third-person, top down or side-scroller game, you had to set this up yourself. There was no choice for game templates that gives you basic functionality of a game you wanted to create.



In UE4, you get to choose to start any project with a game template. You can choose to have C++ or Blueprint template.



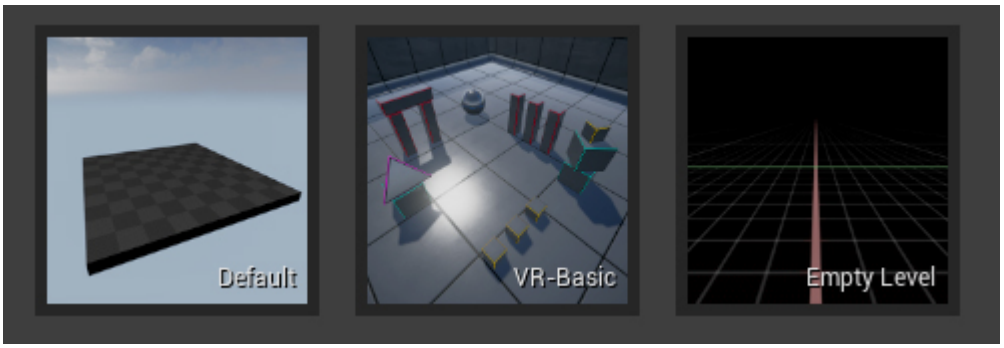
Then, you choose first-person shooter, third-person, side-scroller or any other available game template which gives you a starting point for basic functionality to build and expand upon for your own game.

5. NEW LEVEL TEMPLATES

Level template gives you few necessary actors to begin creating the game world without having to insert these actors yourself. It is a very quick and efficient way to get a level started.

In UDK, if you go to **File → New Level**, a menu pops up to choose between various map templates. You had 4 templates for different time of day and one blank map template:

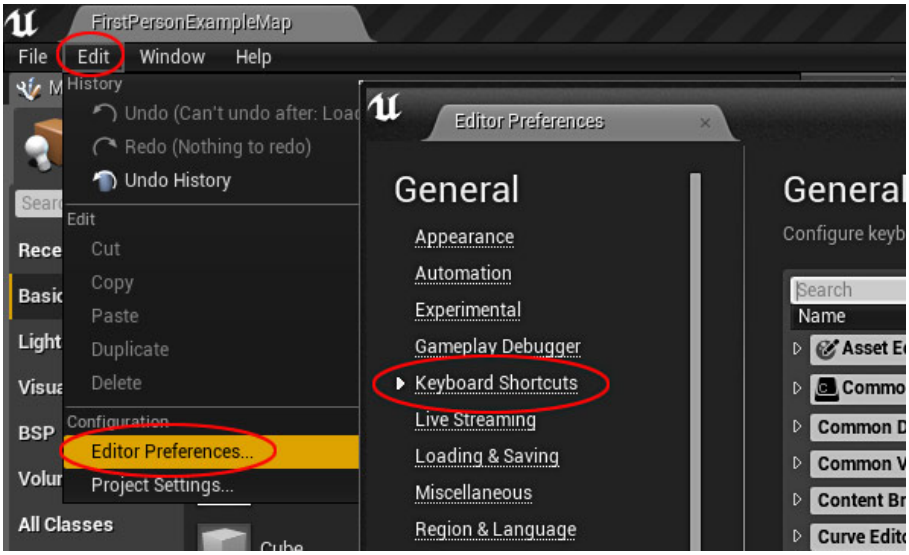
In UE4, you still have new level templates, but as of right now there are only 3 templates to choose from. Default, VR-Basic and Blank Map:



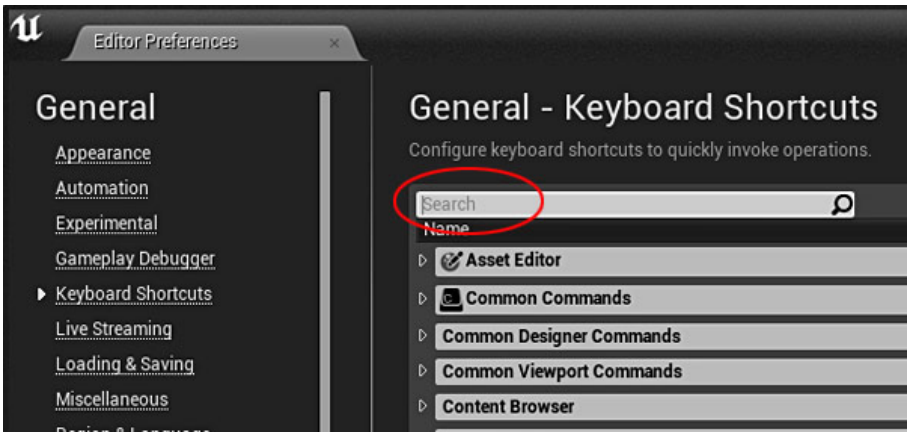
The current single template in UE4 is enough to get started. Perhaps later we'll have more additional time of day templates.

6. KEYBOARD SHORTCUTS

Keyboard shortcuts are almost identical between many common functions from UDK to UE4, but not all. In **UE4**, it is extremely simple to add/update shortcuts by going to **Edit → Editor Preferences → Keyboard Shortcuts**:

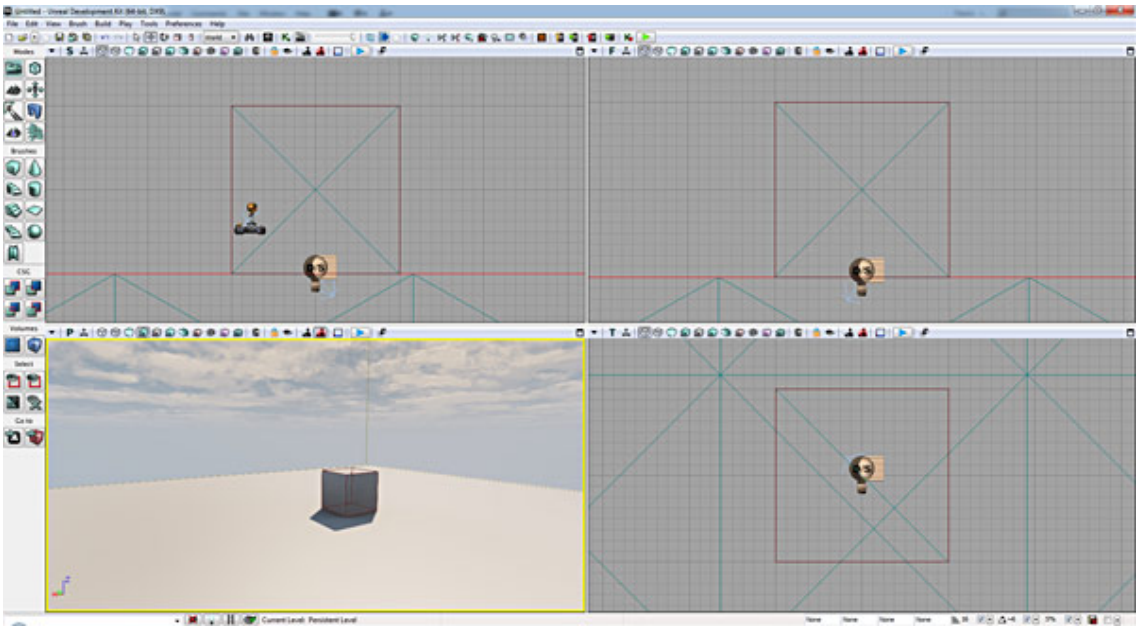


Search for a shortcut and enter the keys for that shortcut to use:

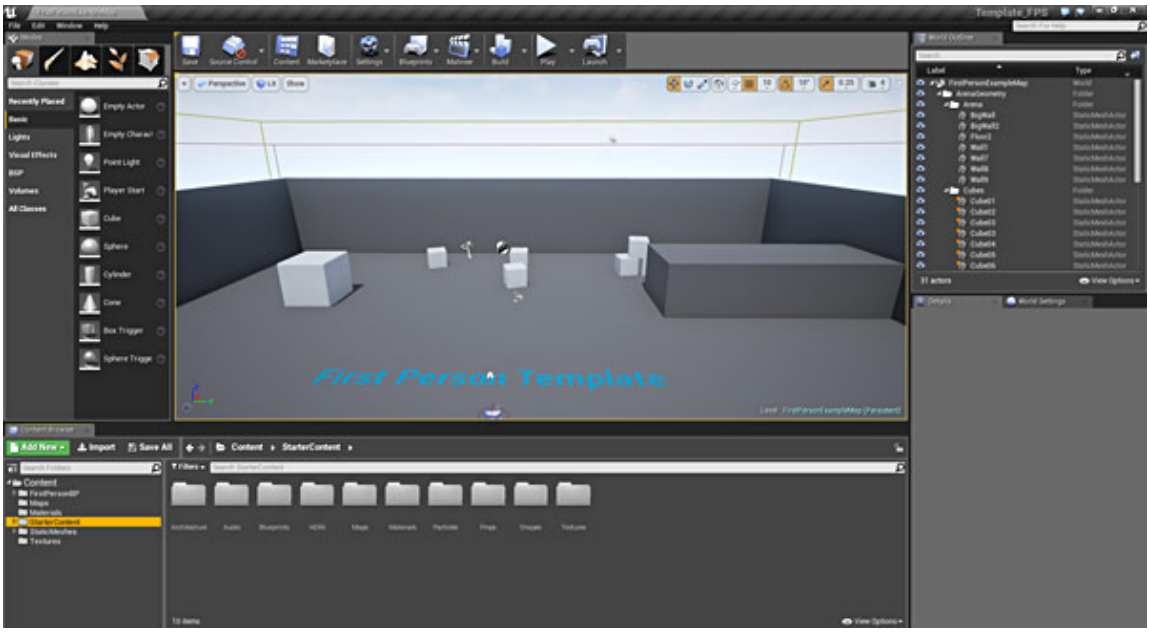


7. TABS AND FLUID INTERFACE

UDK interface contained a lot of buttons, icons and menus. As a complete beginner, it was overwhelming to make sense of it all.

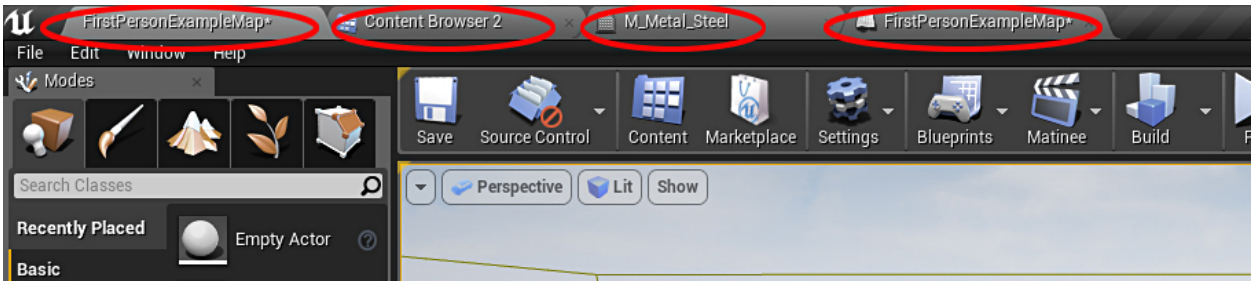


The entire UE4 interface has received a complete overhaul - from colors, to panels, to functionality. Everything is now fluid. You can change and customize the entire interface to exactly how you like it. Resize or drag and drop any panel within the editor to modify the interface.



Another useful function I like is the ability to have tabs - just like a web browser. You can have multiple tabs running across the editor. Dock these tabs anywhere you want so you can have Blueprint, Matinee and Material Editor all in their own tabs to avoid having multiple floatable windows open.

You can still have menus as float windows, but tabbing adds additional organizational option.



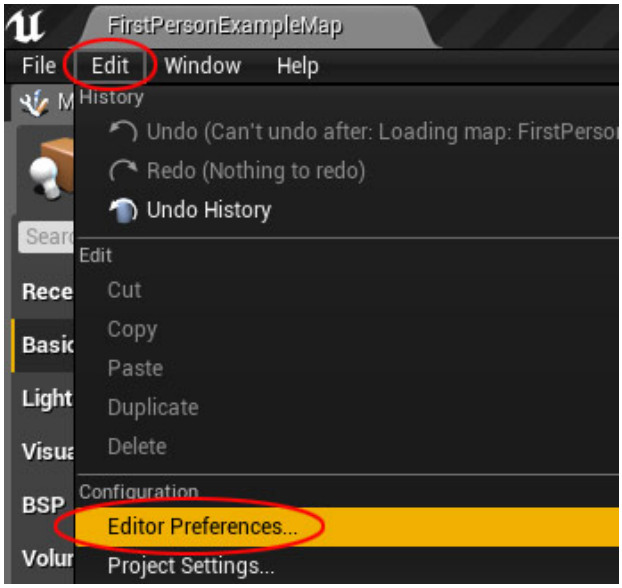
8. VIEWPORT NAVIGATION IN UE4

Viewport navigation from UDK to UE4 is almost identical with a few slight changes.

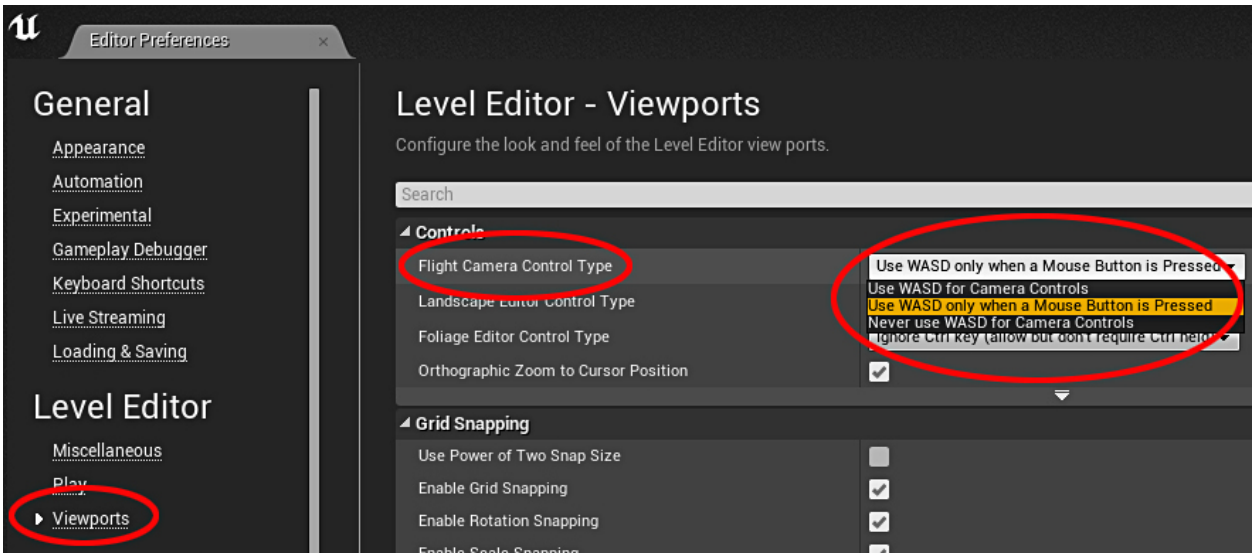
Perspective Viewport Navigation: In UDK, you use WASD keys to move viewport camera in front/back/side directions.

In **UE4** you have to press and **hold Right Mouse Button while using WASD**.

You can disable having to press/hold right mouse button for viewport navigation by going to **Edit → Editor Preferences**:



Under **Viewport** and **Controls** section, change **Flight Camera Control Type** to **Use WASD for Camera Controls**:



The rest of the viewport navigation is the same from UDK. Such as holding the Right or the Left mouse button to look around and move forward and back; and if you hold both Left + Right Mouse buttons you move up/down/side-to-side.

In UDK, if you liked Maya style navigation you had to press L + left or right mouse button in to use it.

- **Hold Alt + Left or Right Mouse** = Maya LT/Maya Style Navigation

Also in UE4, press F to center view on selected object and then press and hold Alt while holding and moving the left mouse button. This will rotate around the selected object. Very useful for looking at a single selected object inside the scene.

- **F** = Center View on Selected Object

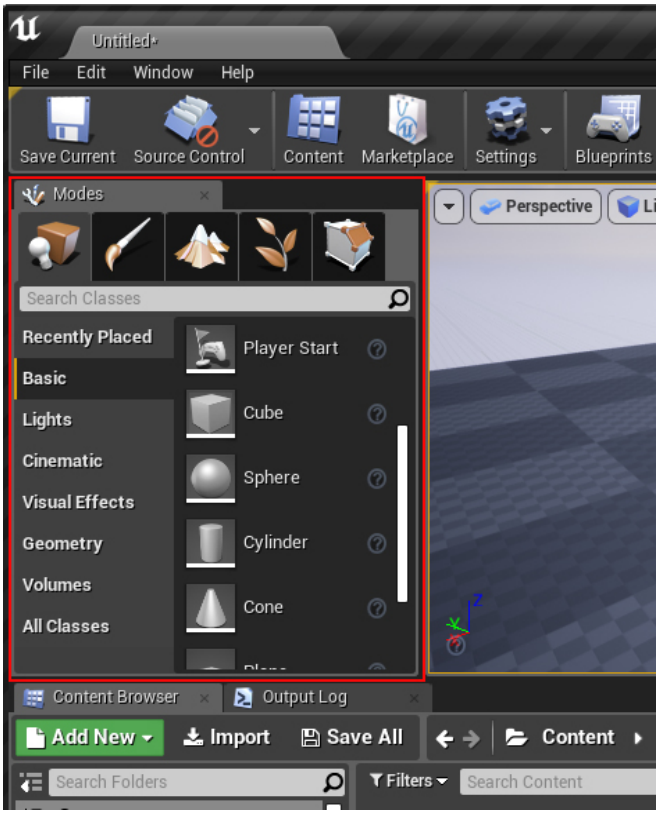
Orthographic Viewport Navigation: In UDK, if you hold and move either the right or the left mouse button both enabled panning inside the viewport.

Now, in UE4 Orthographic Viewport:

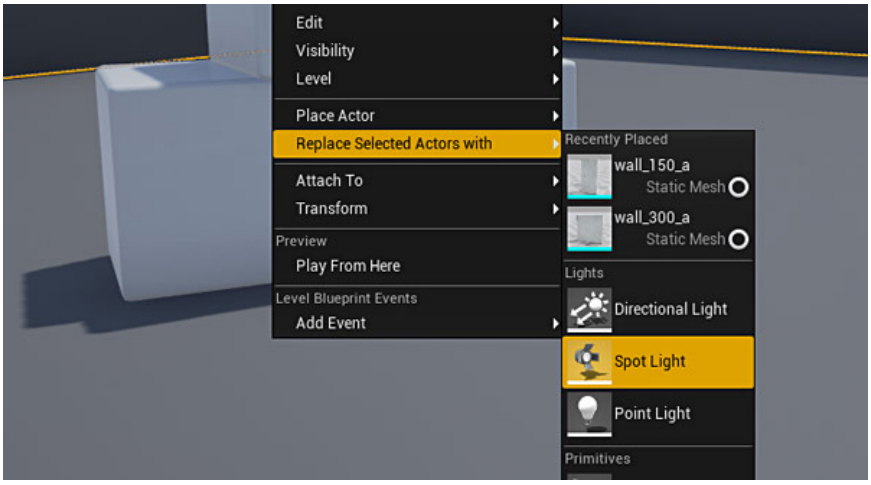
- **Left-Click Hold and Drag** = Marquee Selection
- **Right-Click Hold and Drag** = Pan Within Orthographic Viewport
- **Hold Right + Left Mouse Button and Drag** = Zoom In and Out

9. MODES PANEL

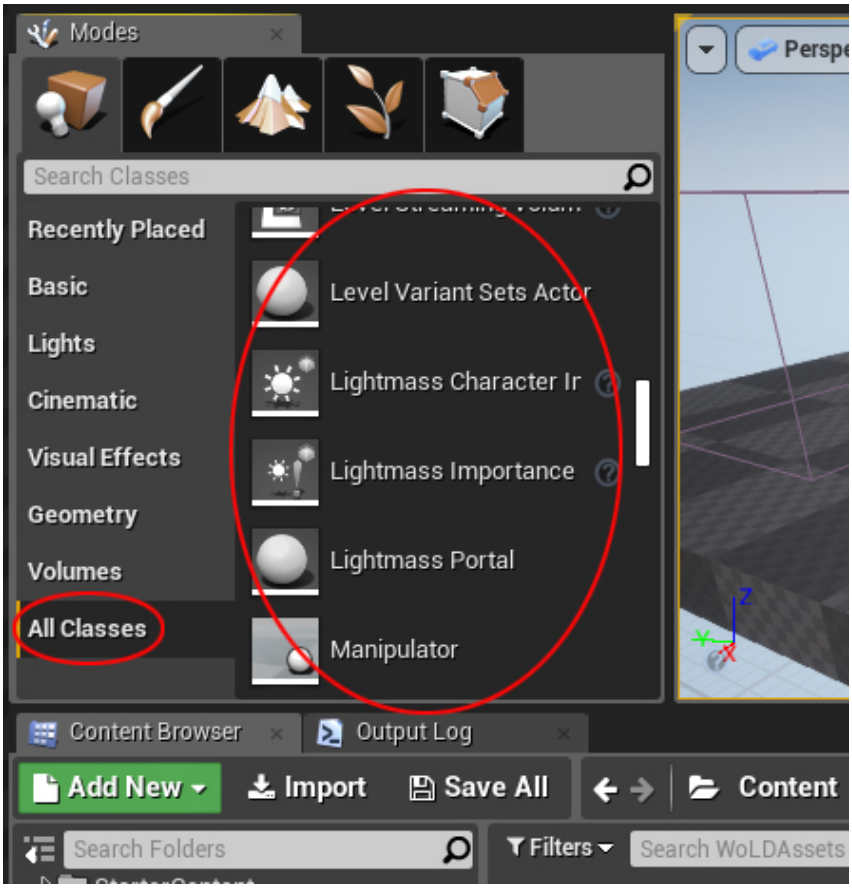
In UE4, placing objects is a lot easier with the addition of the Modes panel (Shift+1). Here you'll find all the actors you need for your environment creation.



Left click and drag right from the Models panel into the viewport or you can right click inside the perspective viewport to insert:



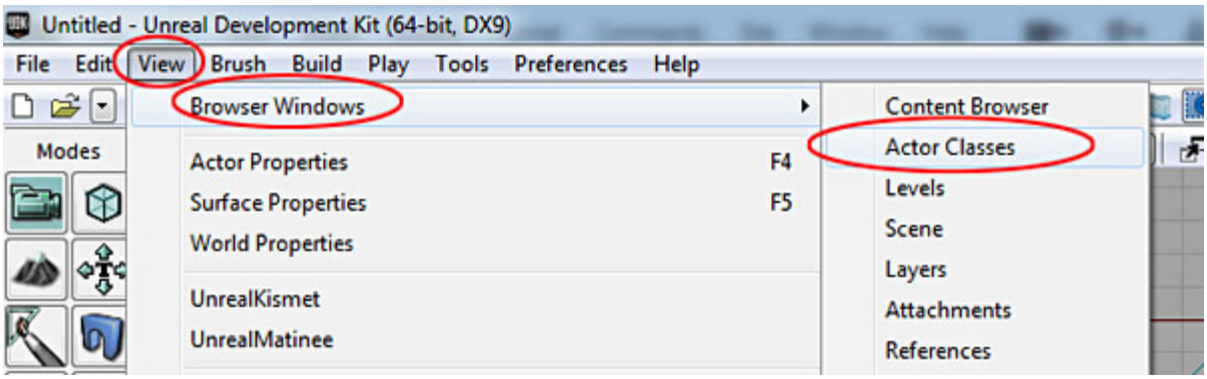
For all actors inside UE4 you can use **All Classes** in the Modes panel:

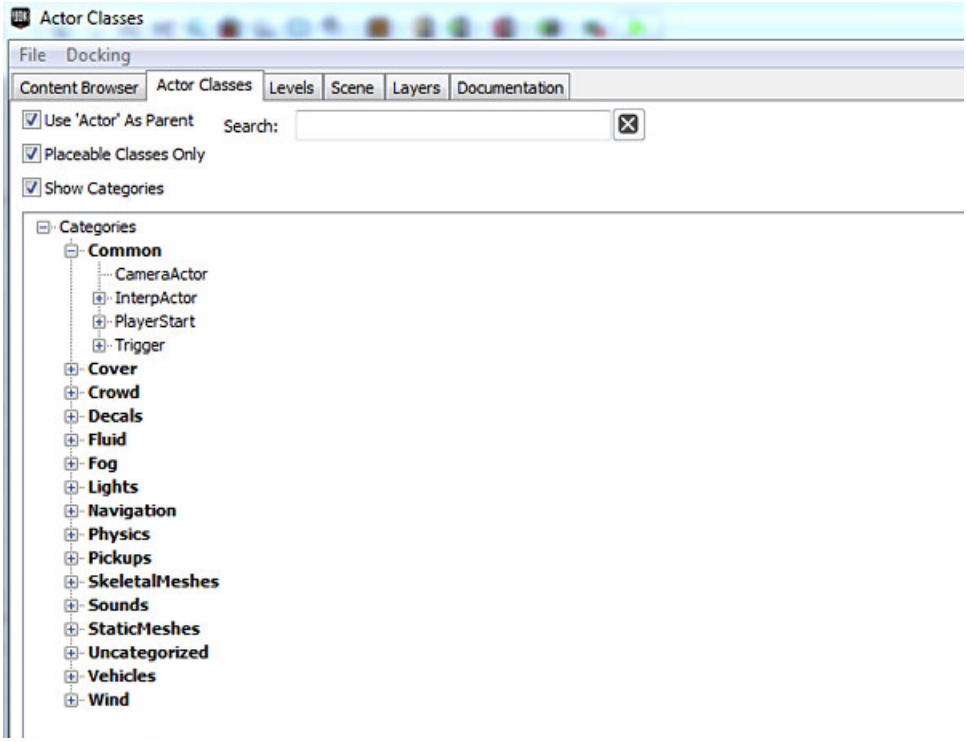


10. CLASS VIEWER

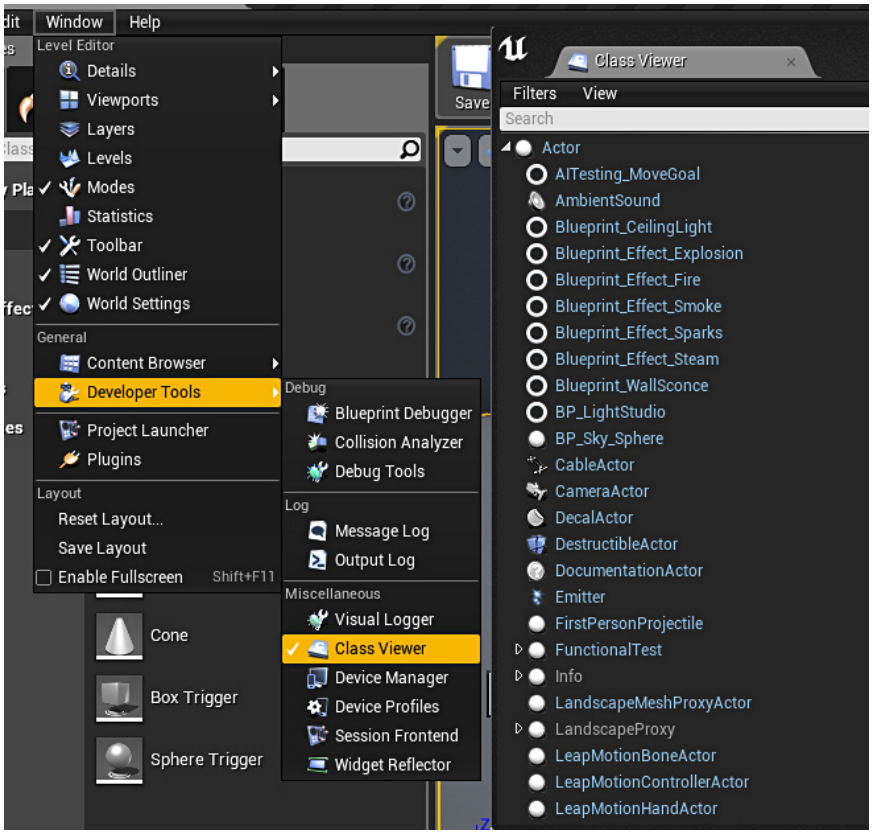
In UDK, to view all actors you had to access them through Actor Classes menu.

View → Browser Window → Actor Classes:



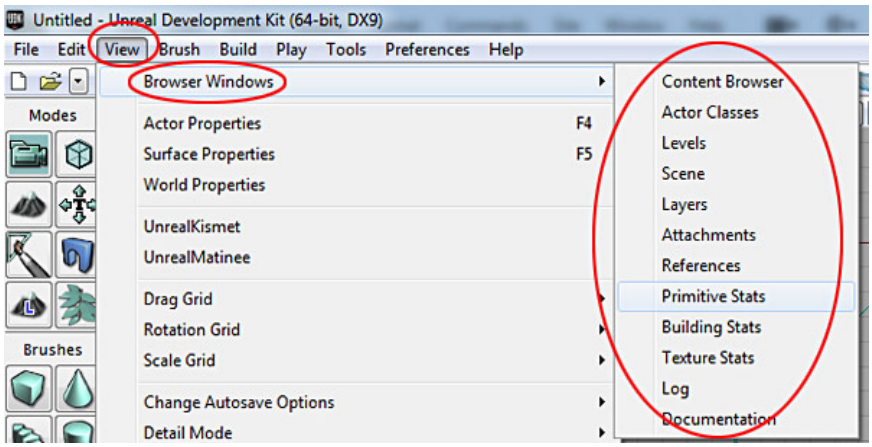


In UE4, Actor Classes are now called Class Viewer. Go to **Window → Developer Tools → Class Viewer**:

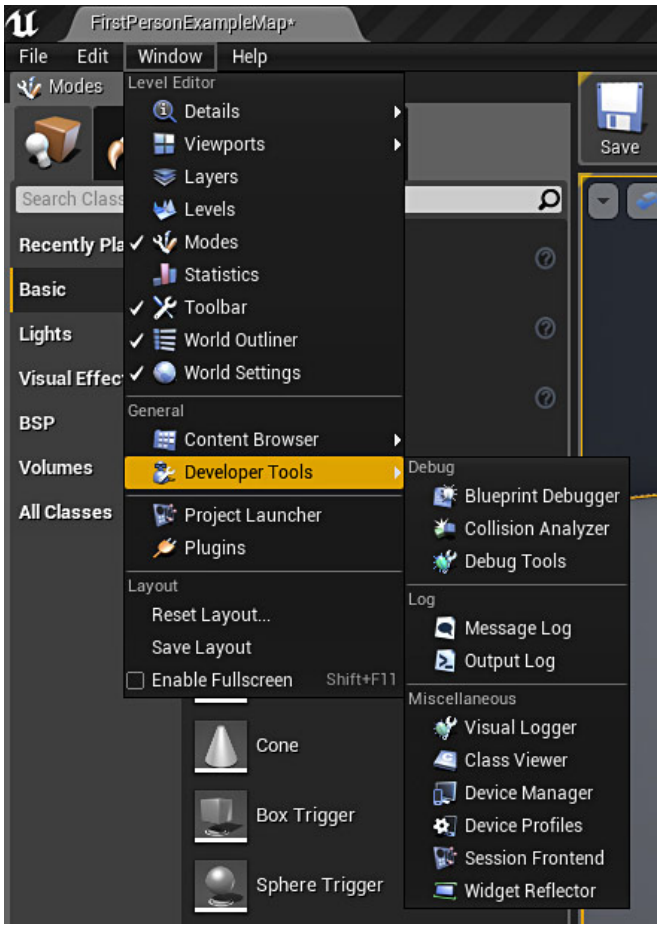


11. DEVELOPER TOOLS

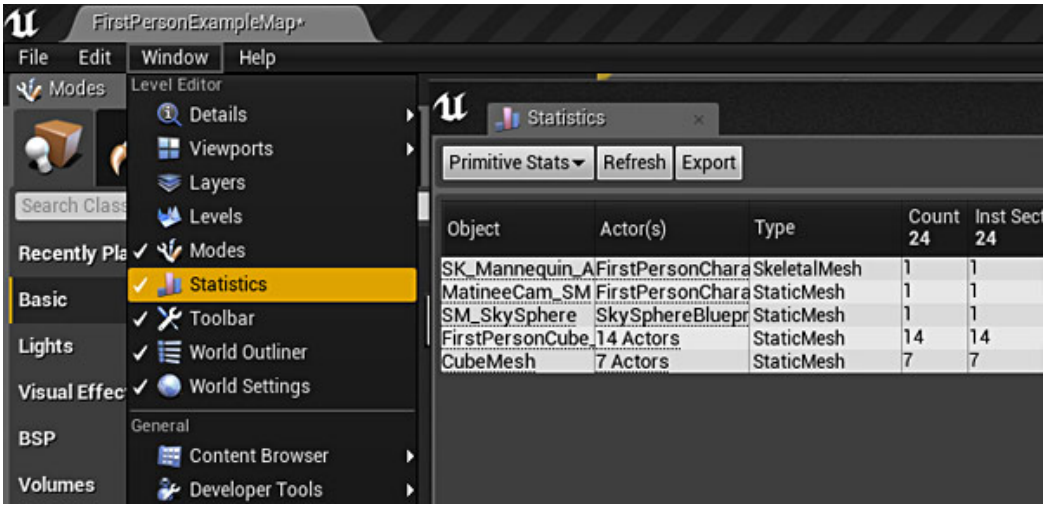
In UDK, you had **View → Browser Window**, which gave you access to various developer tools such as Log, Texture Stats and Primitive Stats:



In UE4, these developer tools are under **Window → Developer Tools**:



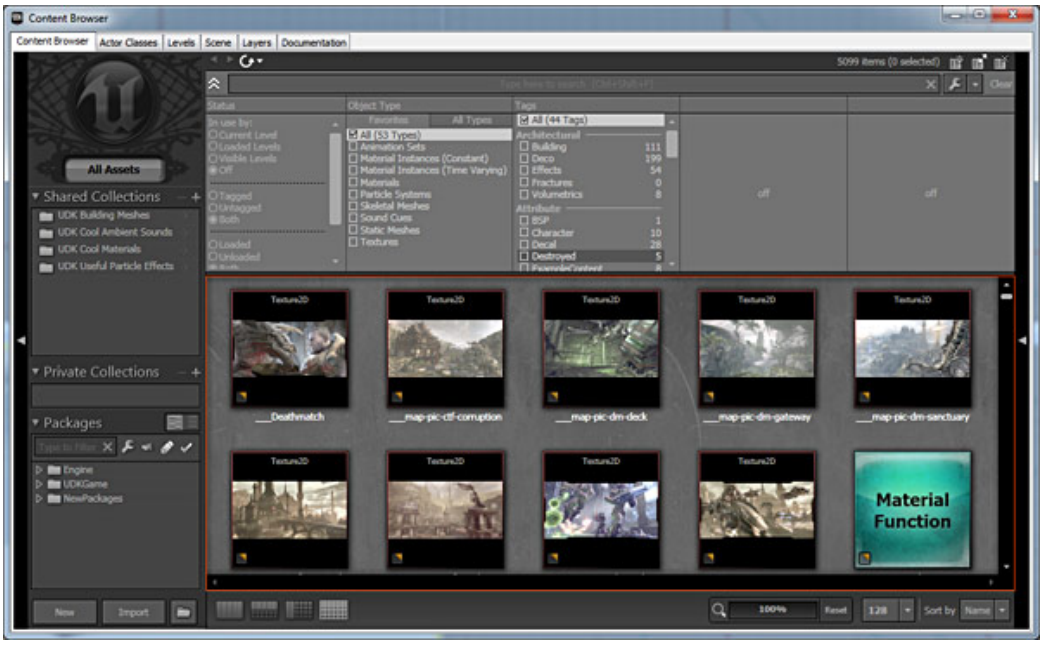
To access Statistics for primitive objects, go to **Window → Statistics**:



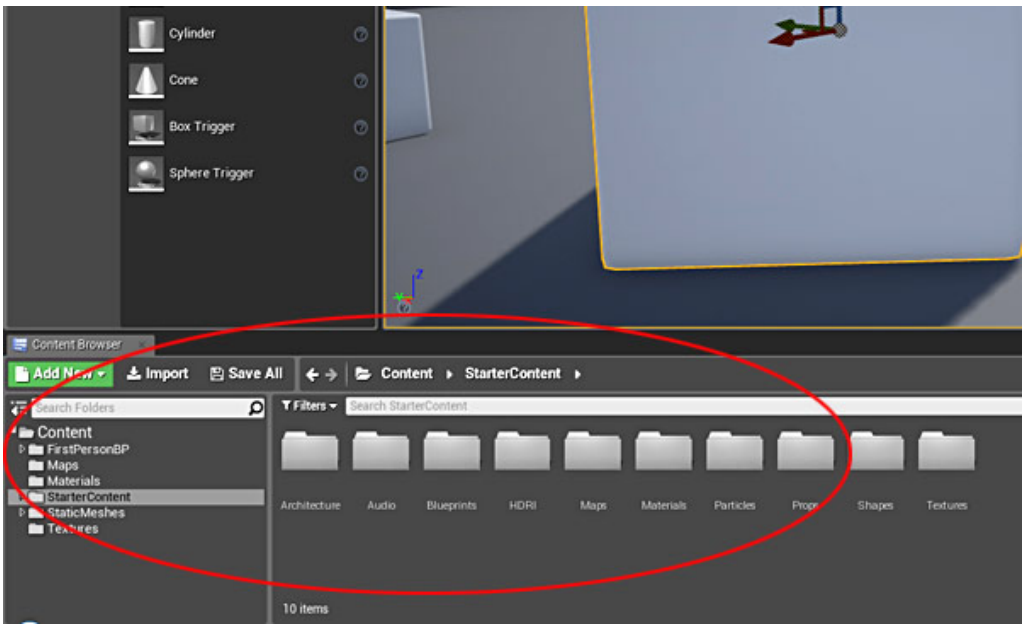
12. CONTENT BROWSER

Content Browser is the content management system in both UDK and UE4.

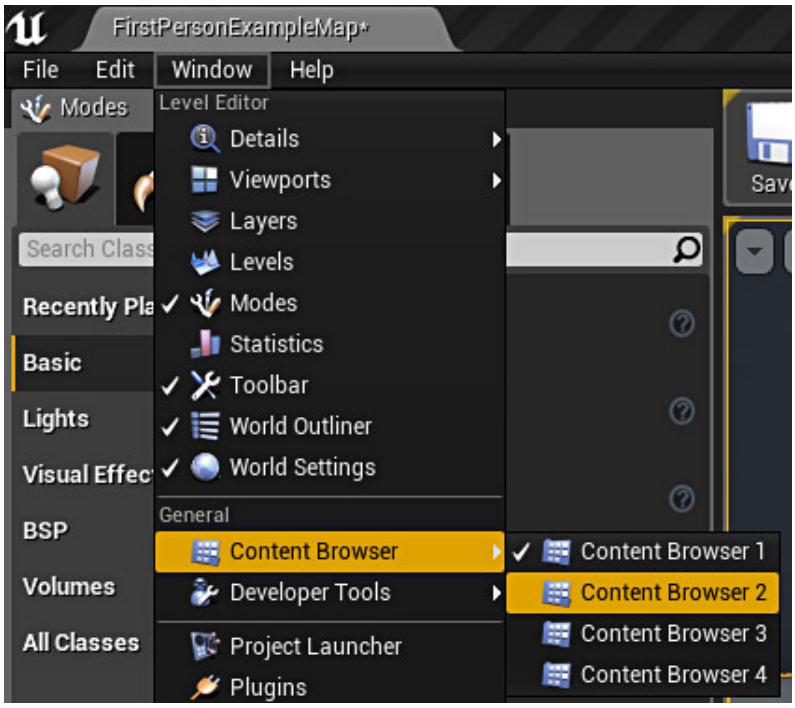
In UDK, Content Browser was a floating window and could be accessed via top toolbar icon or Ctrl+Shift+F shortcut:



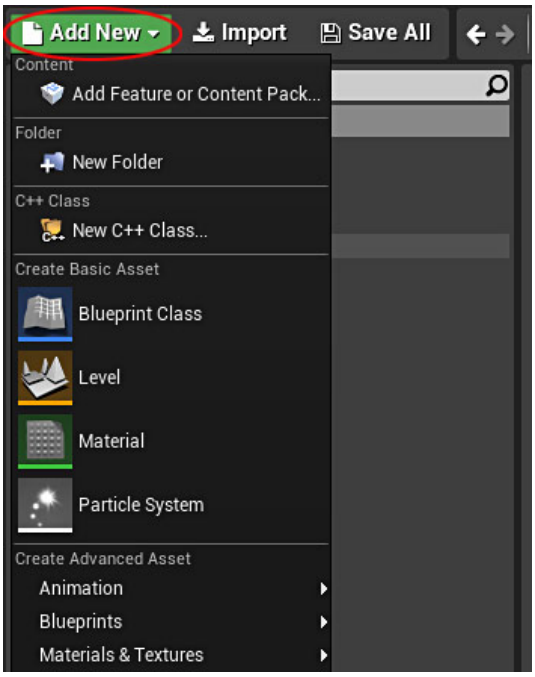
In UE4 Content Browser is now part of the interface, by default it's on the bottom panel of the editor (or on the left for Engine versions 4.7 or older):



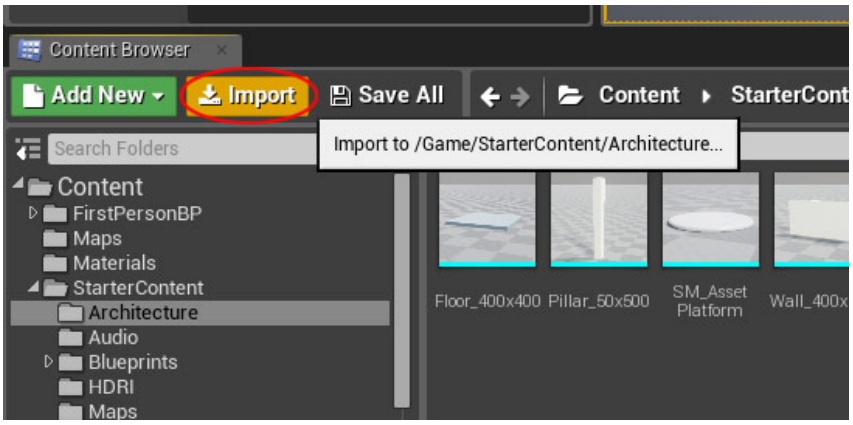
You can open up additional Content Browser windows:



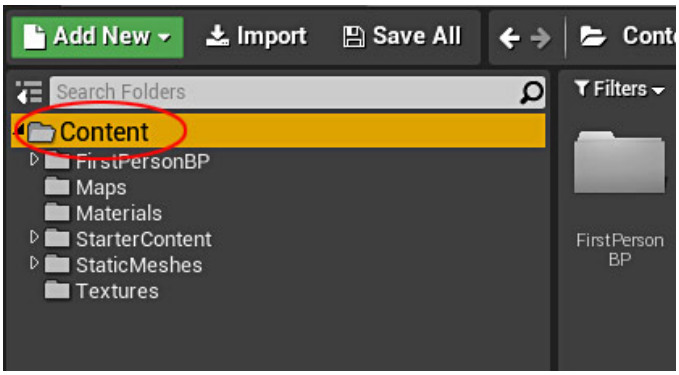
Using the Content Browser in UE4 is more intuitive. You can quickly create any asset by clicking on **Add New**:



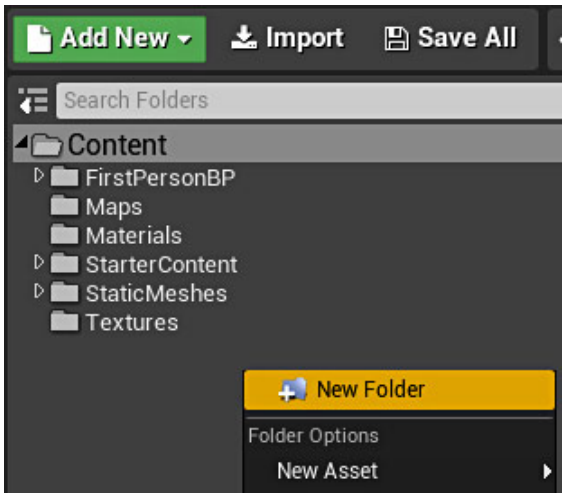
Or quickly **Import** assets:



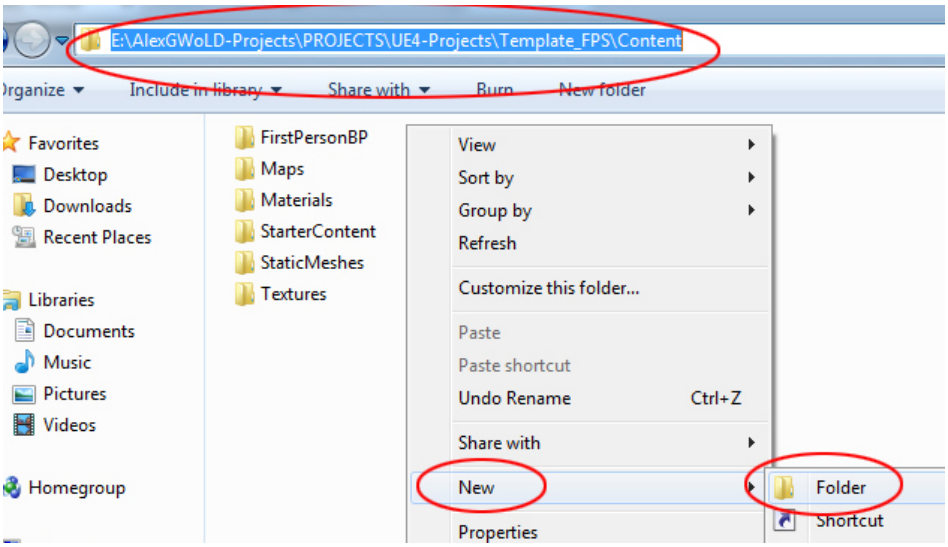
Everything is organized by folders and all of your project assets will be found inside the main Content folder:



You can create folders to organize your content inside the Content Browser itself. **Right click** and choose **New Folder**:



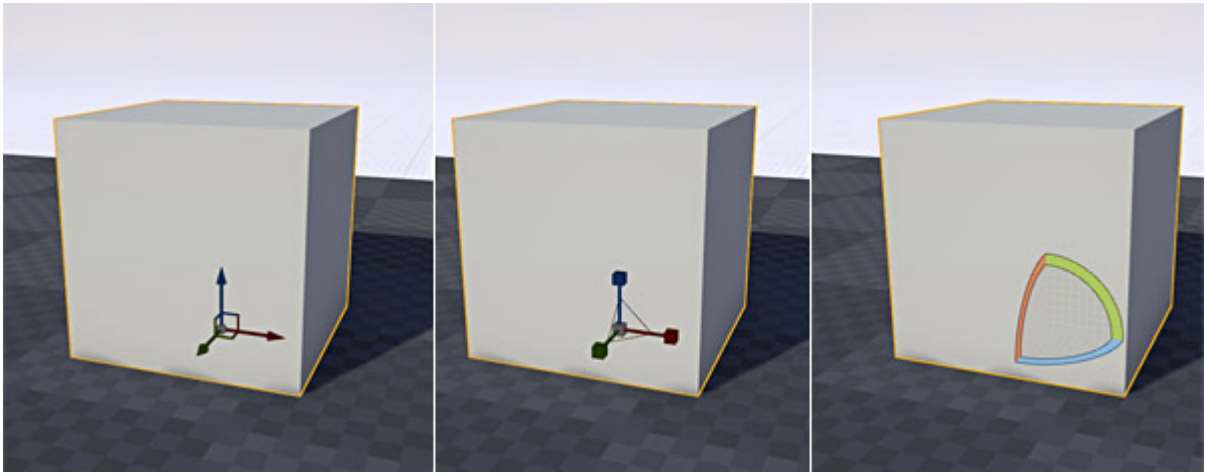
You can also create folder by navigating to a Project Folder directory, within Content folder. This folder will then show up inside Content Browser:



13. WORKING WITH OBJECTS

Working with Actors (objects inside the editor) is similar to UDK. Placing, selecting, deleting, resizing, moving and rotating are the same.

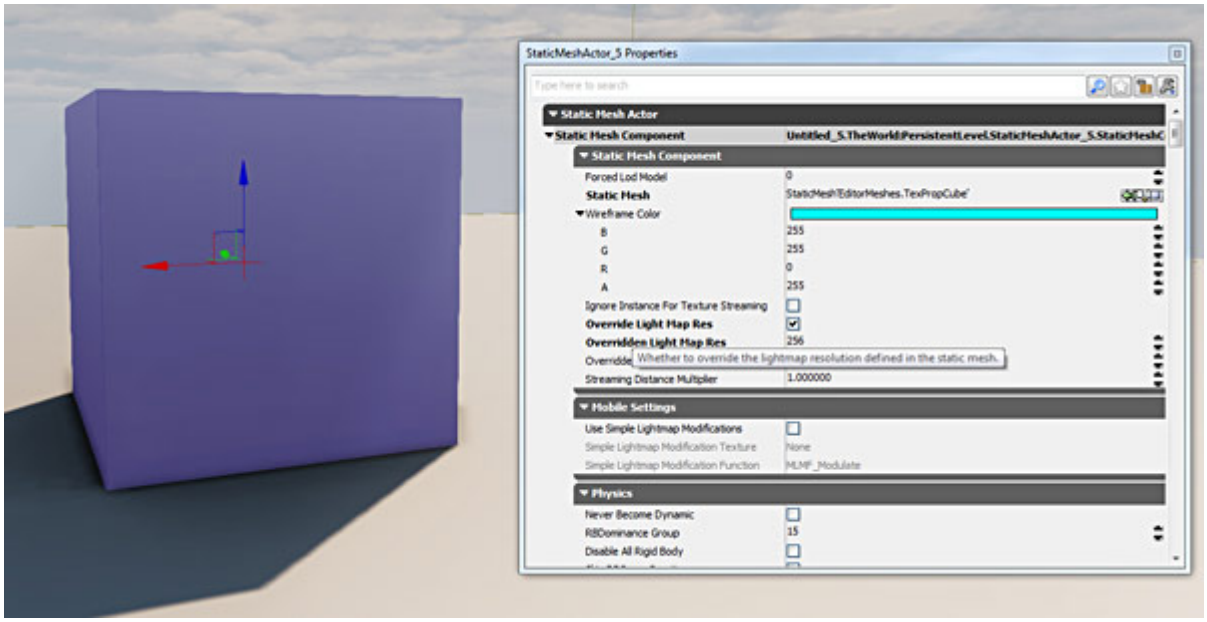
Such as drag and drop right from the Content Browser and using the Spacebar to cycle between move/rotate/scale gizmo.



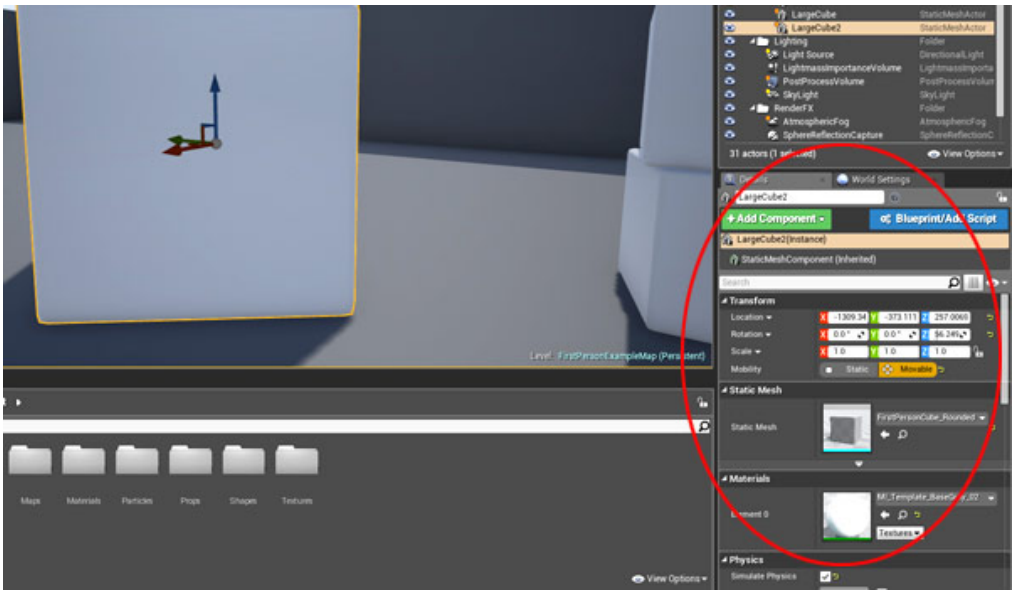
World Outliner window gives you additional control over your entire scene and selecting objects inside your level but more on this later.

14. DETAILS PANEL

In UDK, you have to double click to access the object properties. This would always be a floating window.

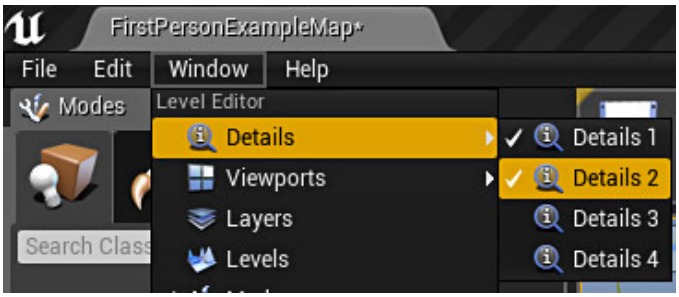


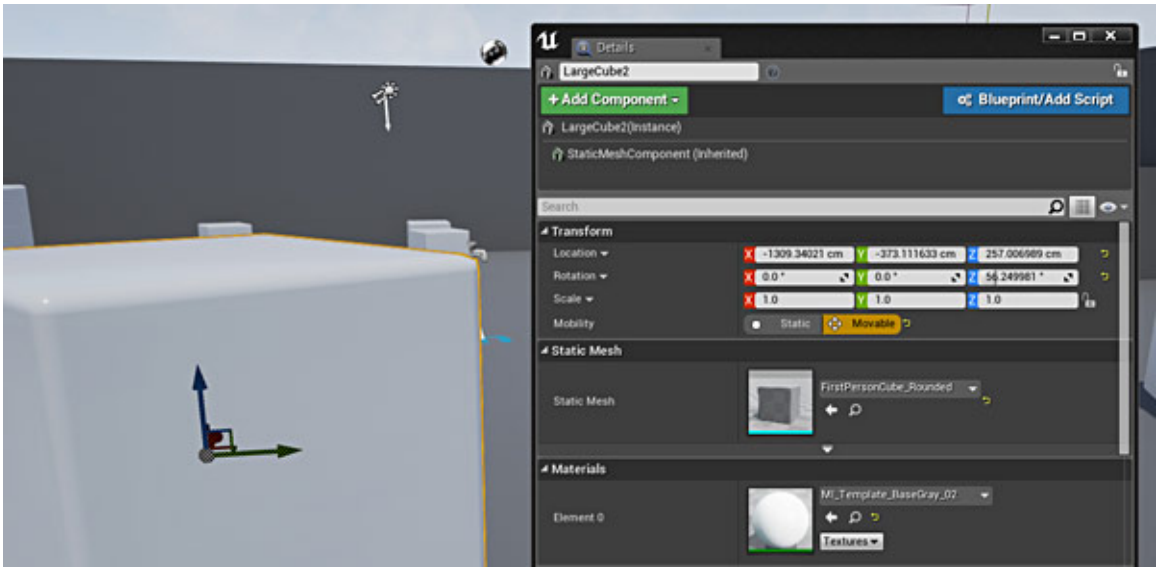
In UE4, object properties are now called **Details**. This menu is part of the editor and you'll find it on the lower right hand side:



Every time you select an actor/object in UE4, the detail panel will change and show the properties for that selected object.

Of course you could also have Details menu as floating window:

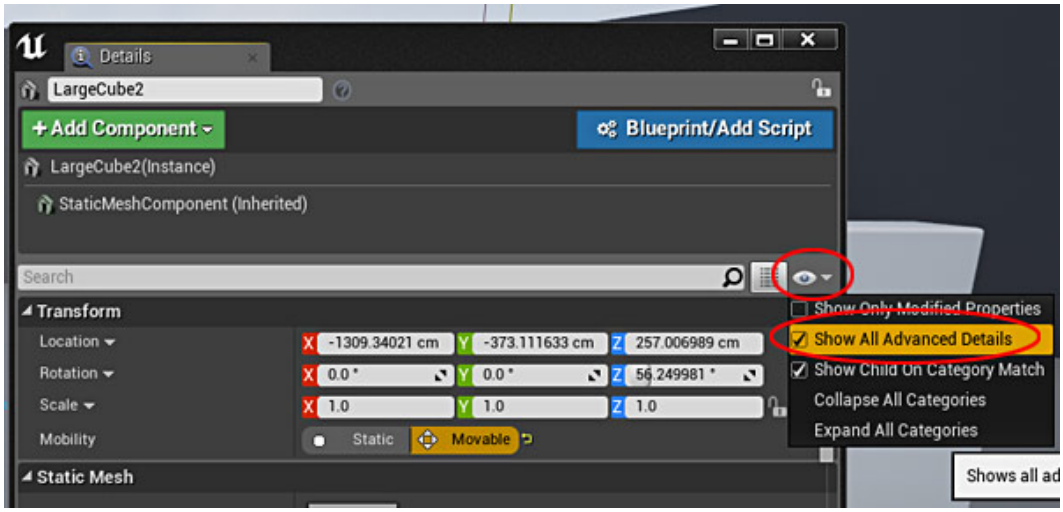




15. SHOW ALL ADVANCED DETAIL PROPERTY

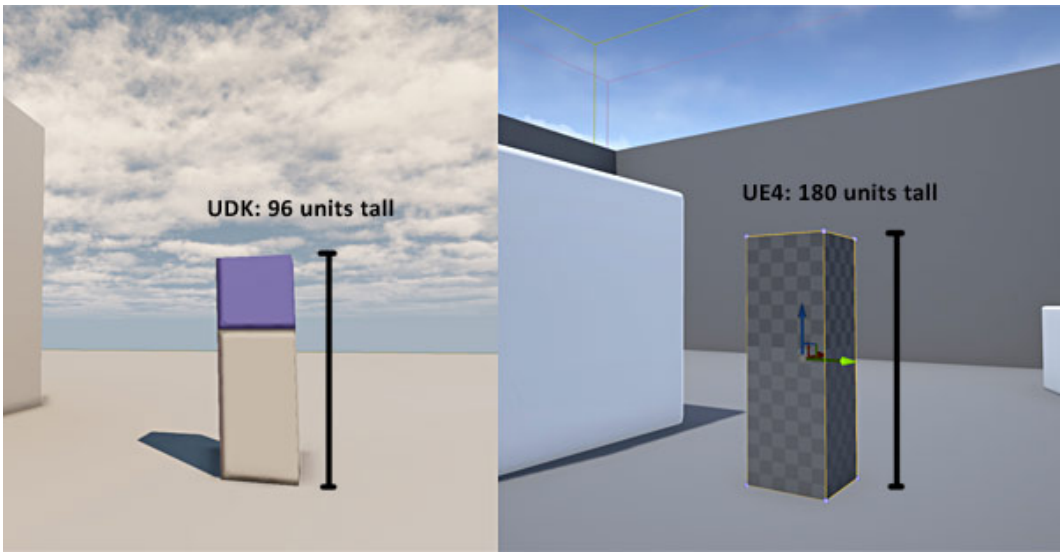
While looking at Object Properties via Details Panel in UE4, you will not see all properties available for that object.

You have to enable to show all advanced settings by left clicking on the eye icon, then turning on **Show All Advanced Details**:



16. SCALE

Scale is different in UE4 from UDK.



- UDK: 1uu = 2cm
- UE4: 1uu = 1cm

Important: make sure to change working units inside your modeling application to centimeters.

Tutorial: UE4/Maya: How to Set Up Grid in MayaLT/Maya to Match Unreal Engine 4

UE4 Character Scale Dimensions are:

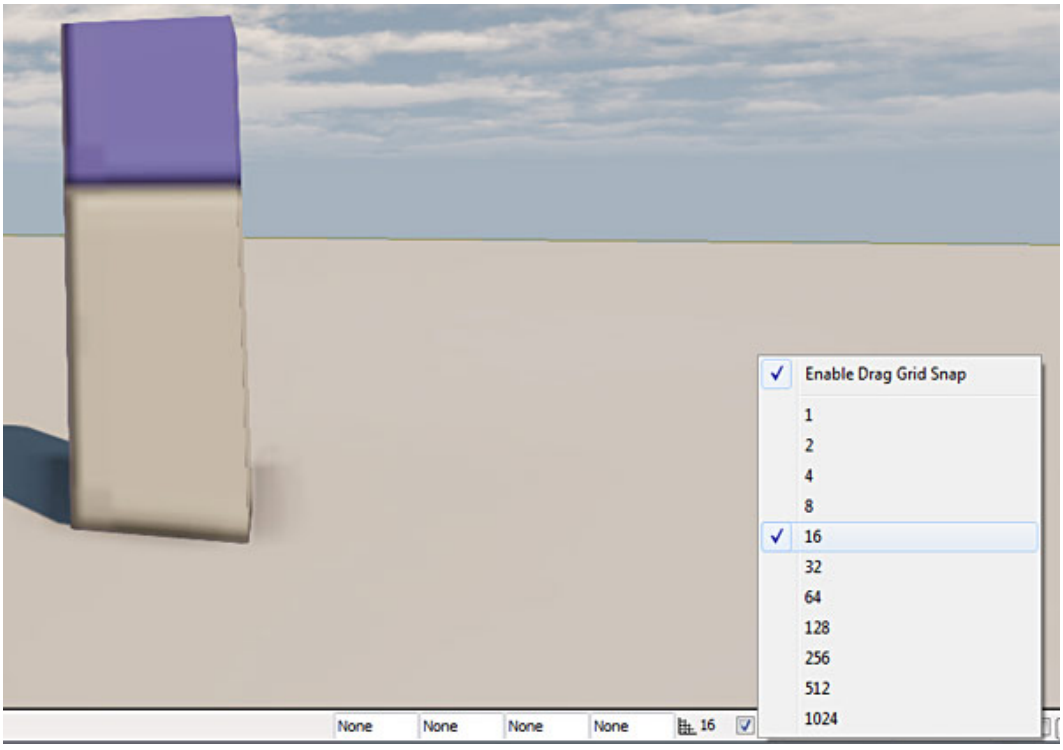
- Height: 180uu/cm

See **World Scale Dimensions and Proportions** section of this guide for more detail.

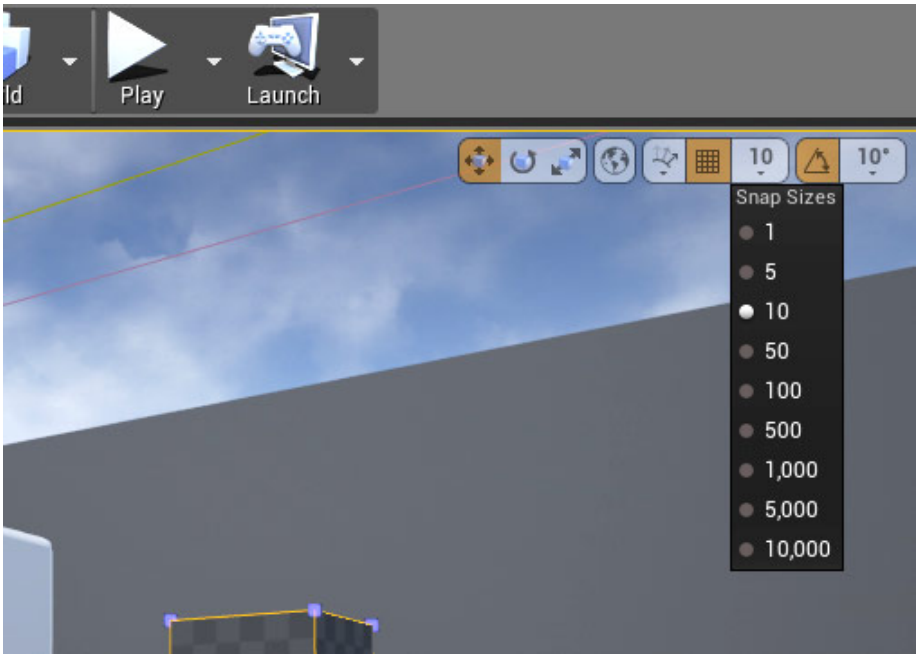
Tutorial: UE4 - Guide to Player Scale and World/Architecture Dimensions

17. GRID SETTING SIZE

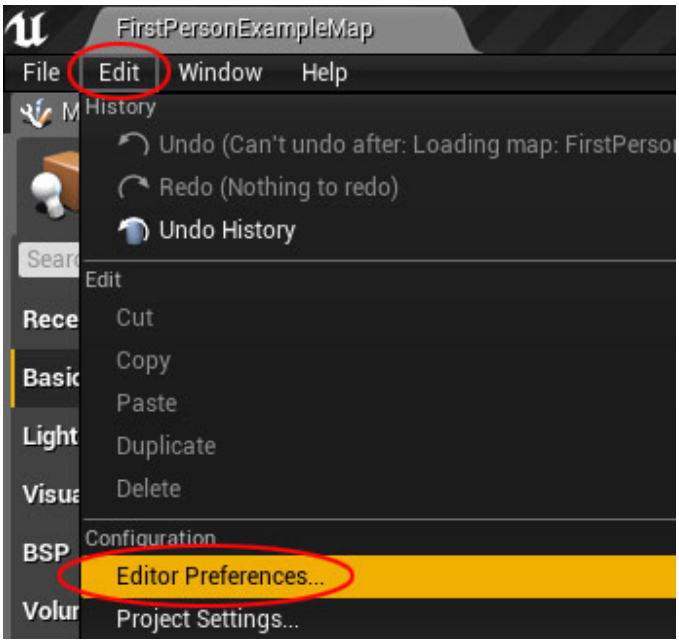
In UDK, grid size follows the power of 2. This means when you increase grid size it goes from 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048. 4096.



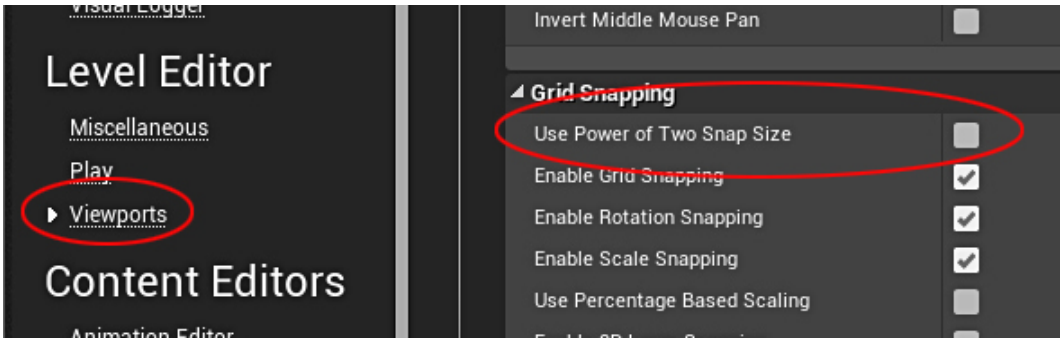
UE4 follows the standard decimal grid size system. Values are 1, 5, 10, 50, 100, 500, 1,000, 5,000 and 10,000. You have to set up 3d modeling application to match the grid size in Unreal Engine 4.



If you still want to use power of 2 grid in UE4, you can switch settings by going to **Edit → Editor Preferences**:



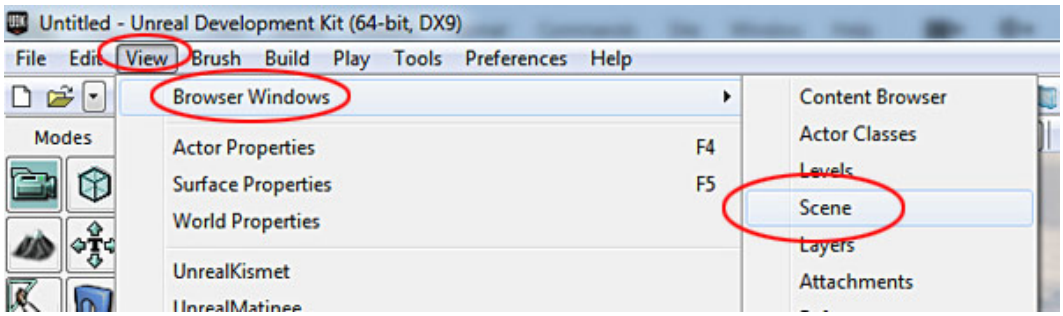
Then in **Level Editor → Viewports** and under **Grid Snapping** option, you will see a check box to **Use Power of Two Snap Size**:

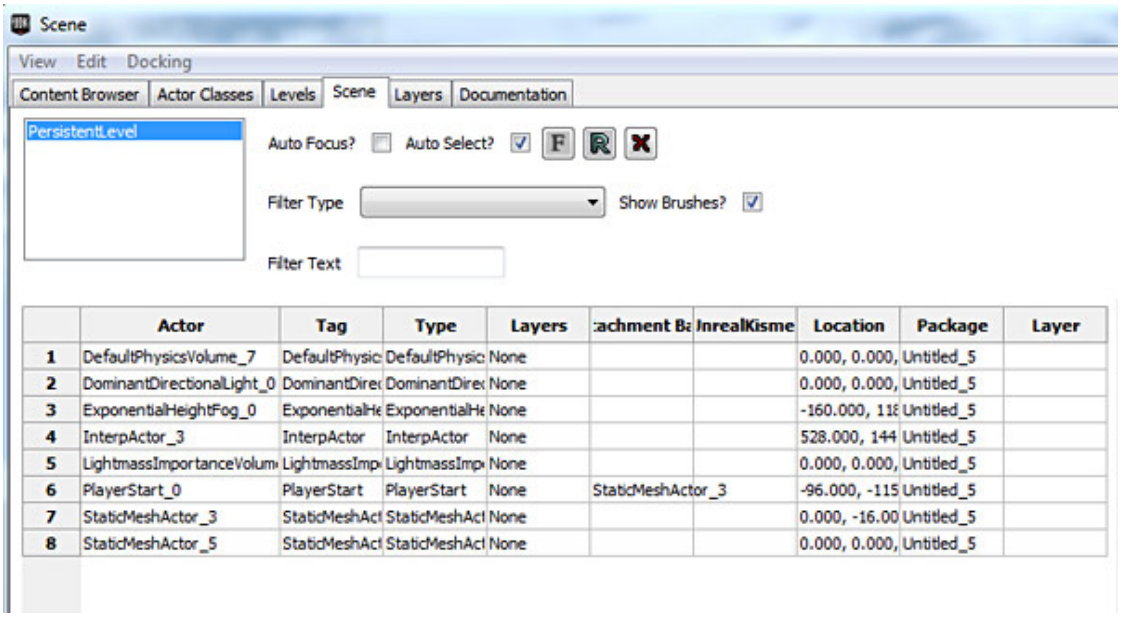


I recommend to stick with the new decimal grid system.

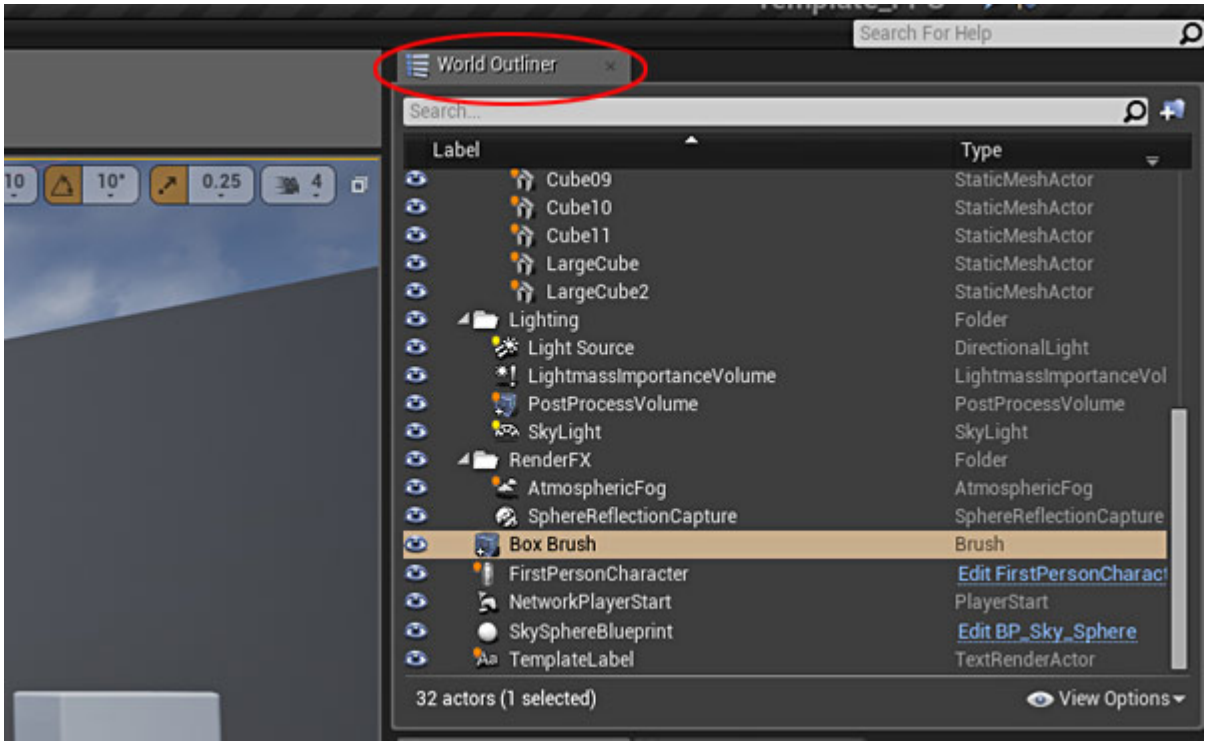
18. WORLD OUTLINER

In UDK, to see every single object placed inside your level you have to go to **View → Browser Window → Scene**:

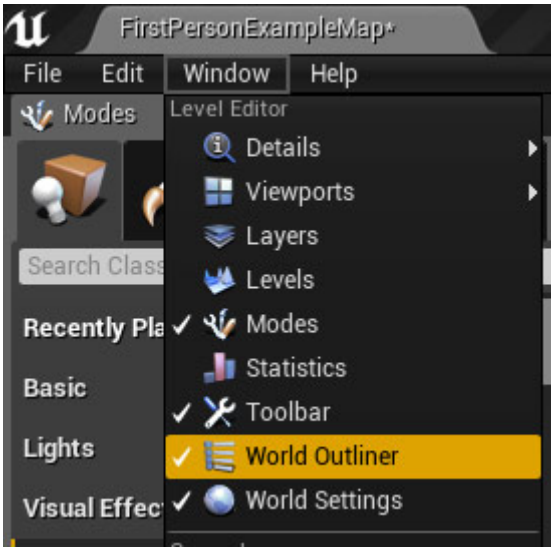




In UE4, this is now contained within its own editor panel called **World Outliner** and you will find this menu on upper right-hand side. It will list all objects inside the scene.



If you don't see the **World Outliner** go to **Window → World Outliner**:

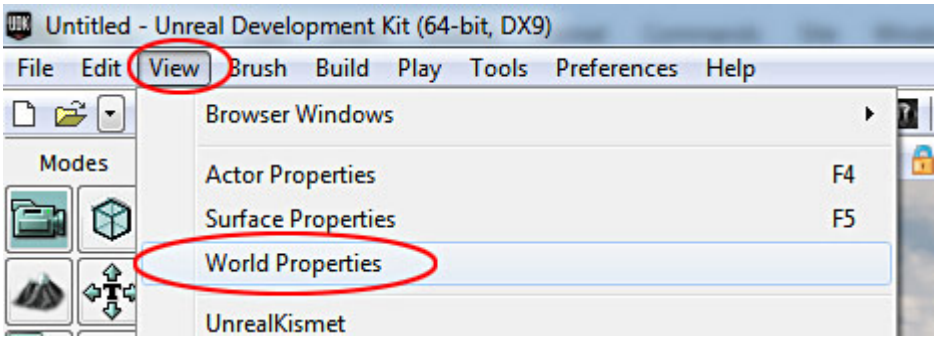


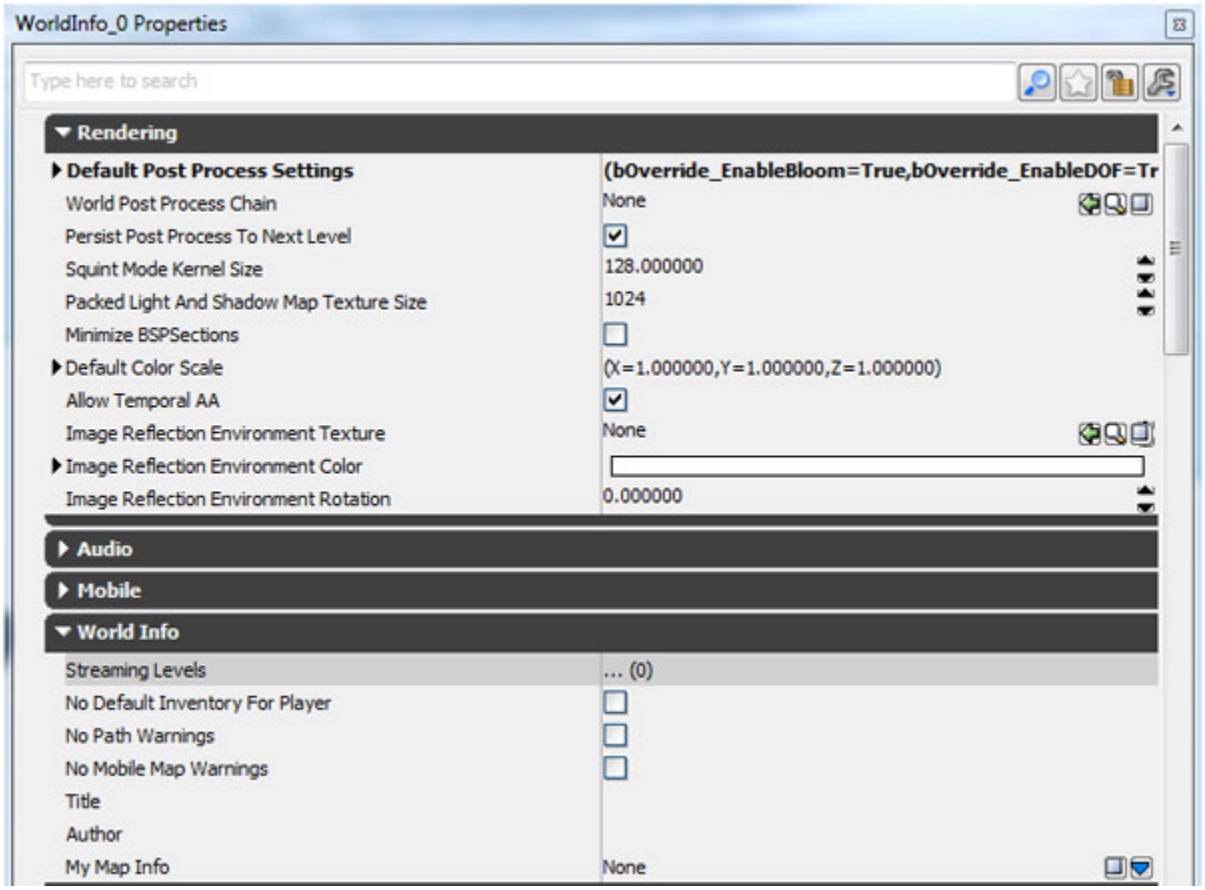
Through the World Outliner, you can select, search and organize all of the actors currently placed within your environment.

19. WORLD SETTINGS

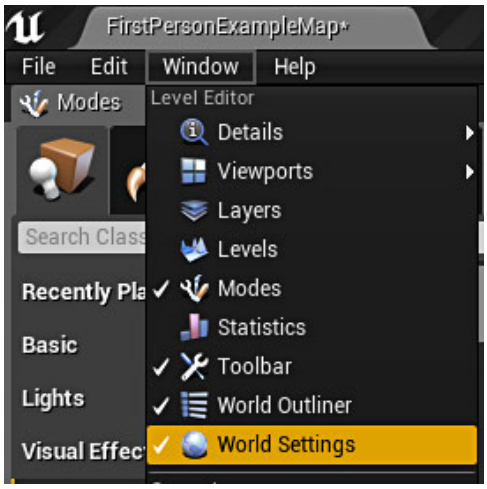
World Settings are universal properties for the entire environment.

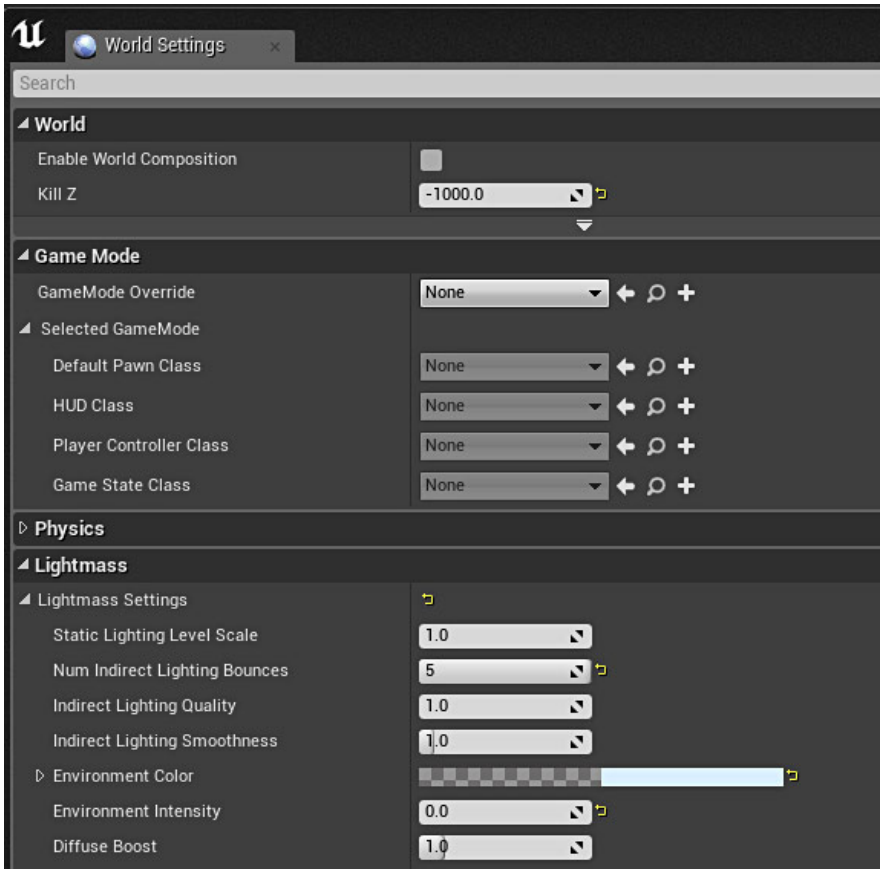
In UDK, to access World Properties, go to **View → World Properties**:





In UE4, World Properties are now called **World Settings** and you can access this menu by going to **Window → World Settings**:





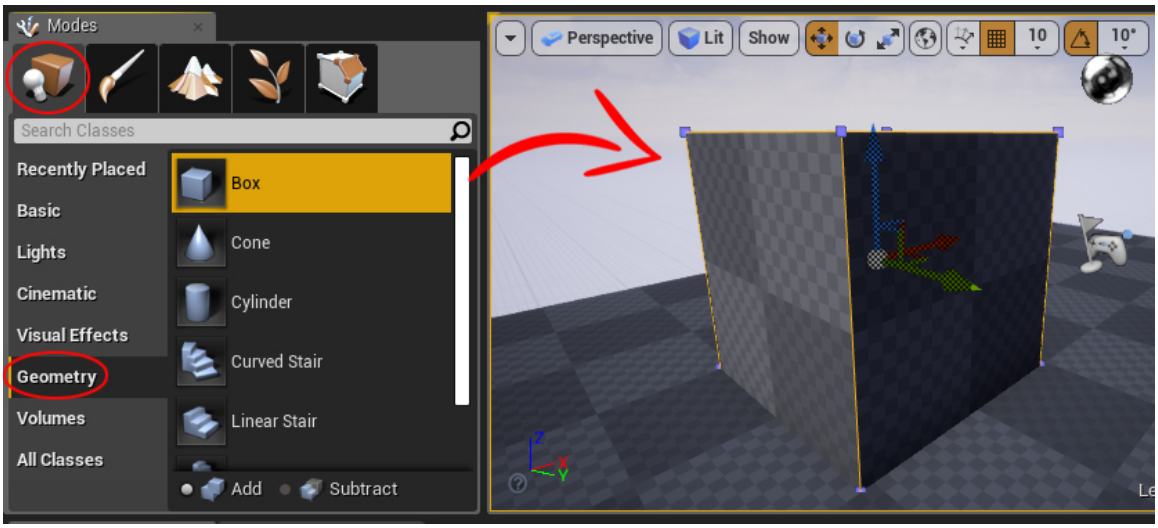
20. BSP BRUSHES

BSP brushes are extremely useful for blocking-in and prototyping a level. You can build a map with BSP brushes and have a layout to test very quickly.

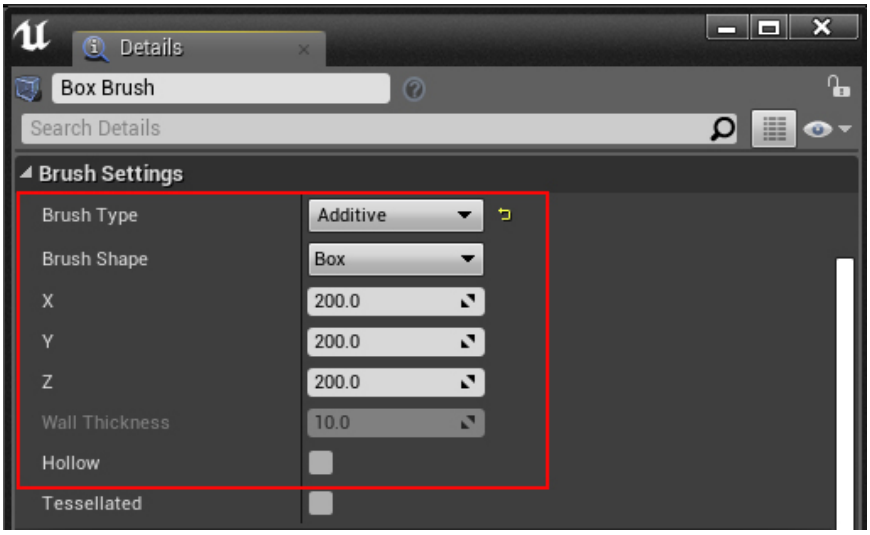
In UDK, BSP process was slow and cumbersome. You had to build geometry every time you modified any BSP brush to see changes. There is a setting under Preferences to update this option to automatic BSP visualization, but it would build geometry every time you move a brush and slowed down your workflow.

In UE4, BSP brushes process has been improved. You no longer need to be rebuilt geometry if brushes have been modified. The entire BSP process is now much faster. It's still not as streamlined as some other BSP based game engines.

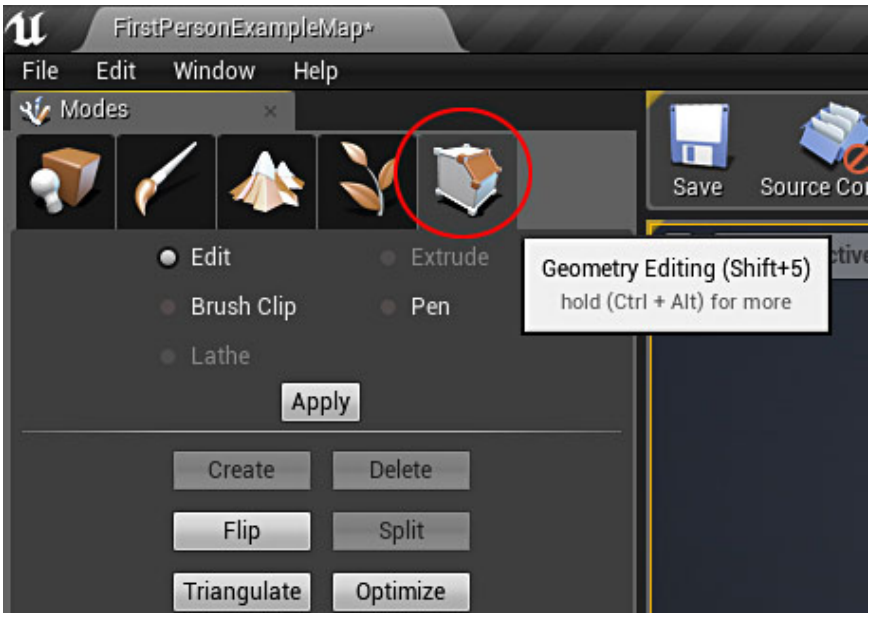
You'll find BSP brushes in the **Modes** panel, **Place** tab and Geometry. Left click and drag BSP brushes from this menu right into the viewport:

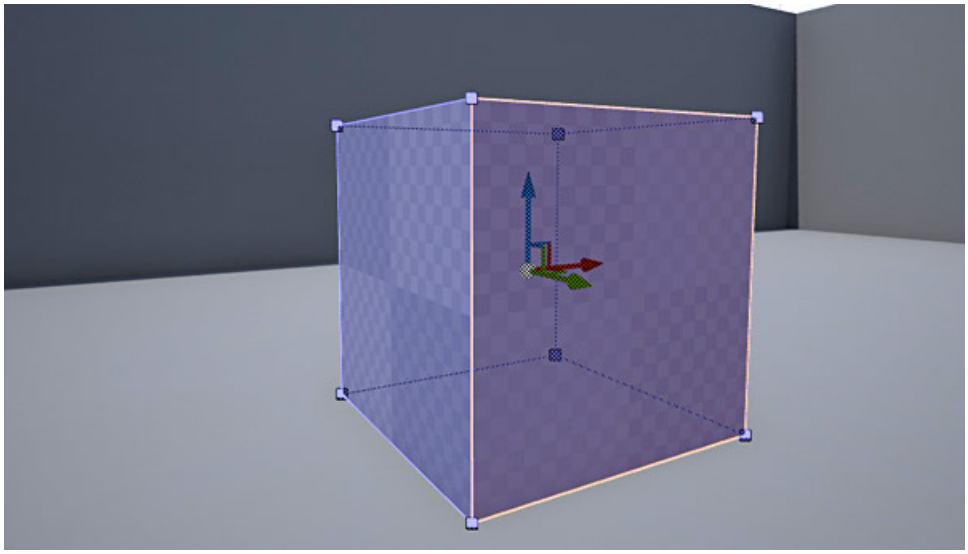


You can use Details panel to input specific dimensions for the BSP brush:



Or, you can use Geometry Editing Mode (**Shift+5**) to resize the brush using vertices, edges and faces:

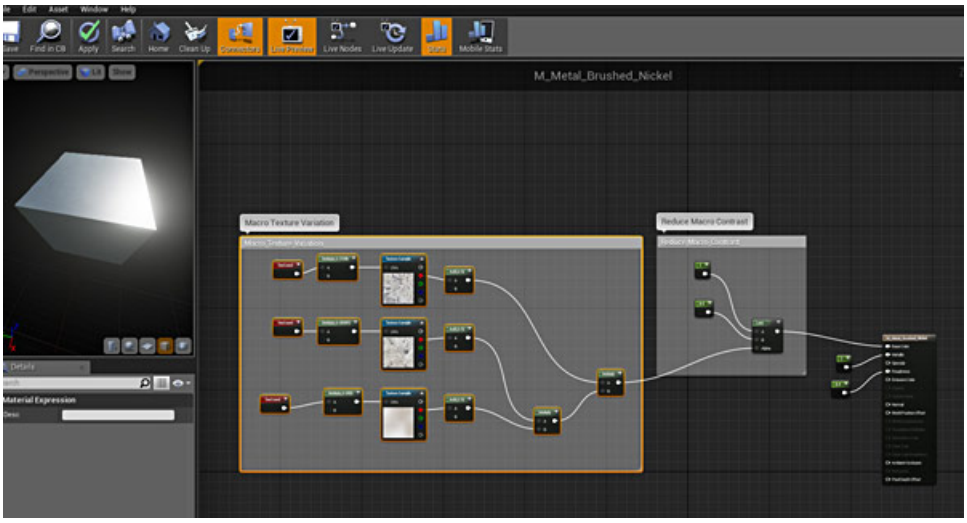




21. PBR MATERIAL CREATION

Material creation in UE4 is very different from UDK.

The material editor is similar to UDK but methodology and approach of creating materials has changed. UE4 now uses PBR or Physically Based Rendering, which offers more realistic way of rendering these materials.



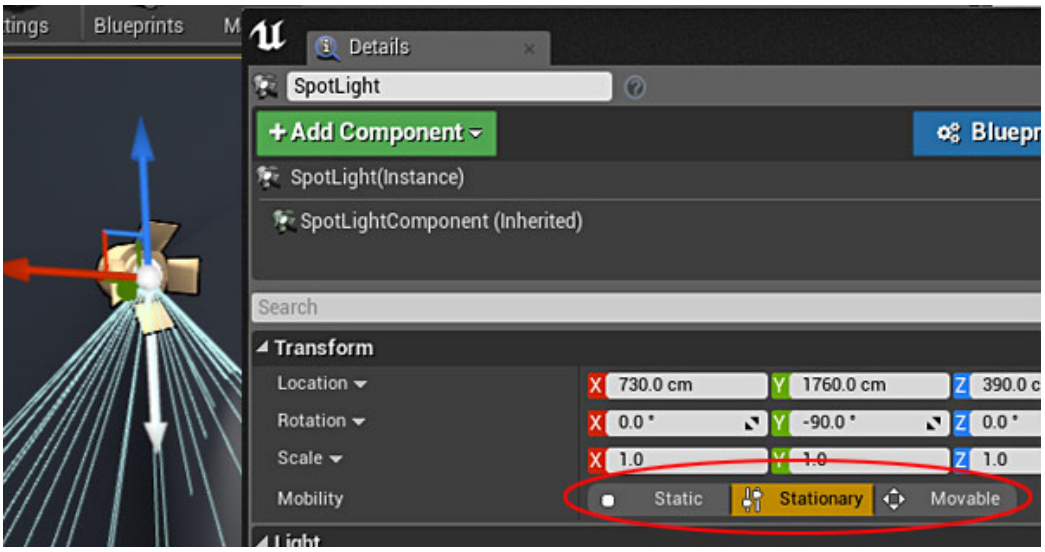
At first using PBR workflow may seem confusing. But once you learn how to create few PBR materials, it's actually easier than the previous workflow.

More information on UE4 PBR: https://docs.unrealengine.com/en-us/Engine/Rendering/Materials/PhysicallyBased

22. LIGHTING

Each light you place in UE4 will contain 3 different mobility types: Static, Stationary and Moveable.

Once you insert a light, using the Details panel you can change that light to be Static, Stationary or Moveable:

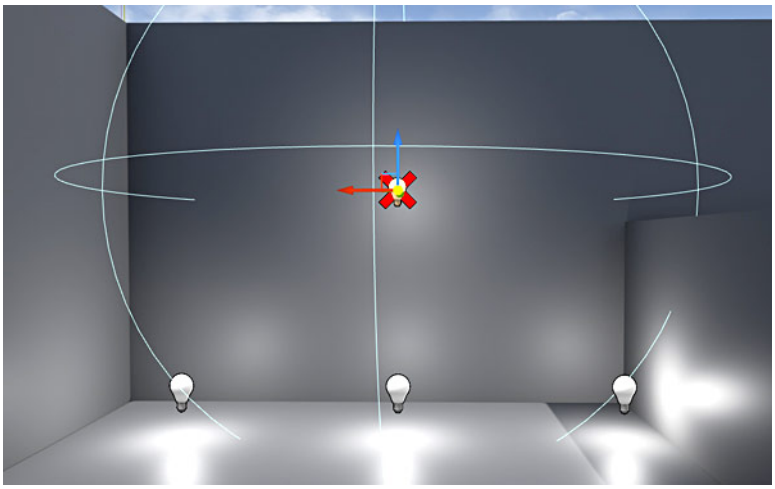


Static vs Stationary vs Movable:

- **Static**: completely static light that has no overhead during the game
- **Stationary**: can change its color and brightness at runtime (via Blueprint) but cannot move, rotate or change influence size
- **Movable**: fully dynamic light capable of changing all of its properties during runtime

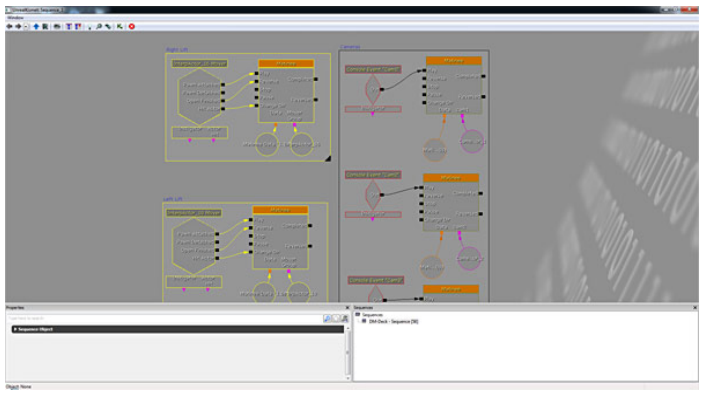
For important lights use Stationary (default). For fill-in lights, small area lights and far away lights use Static. For fully dynamic lights use Moveable.

Important: **you cannot have more than 3 overlapping Stationary lights** in an area. You can have more than 3 Stationary lights but the radius of influence for 3 of these lights cannot overlap at the same time:

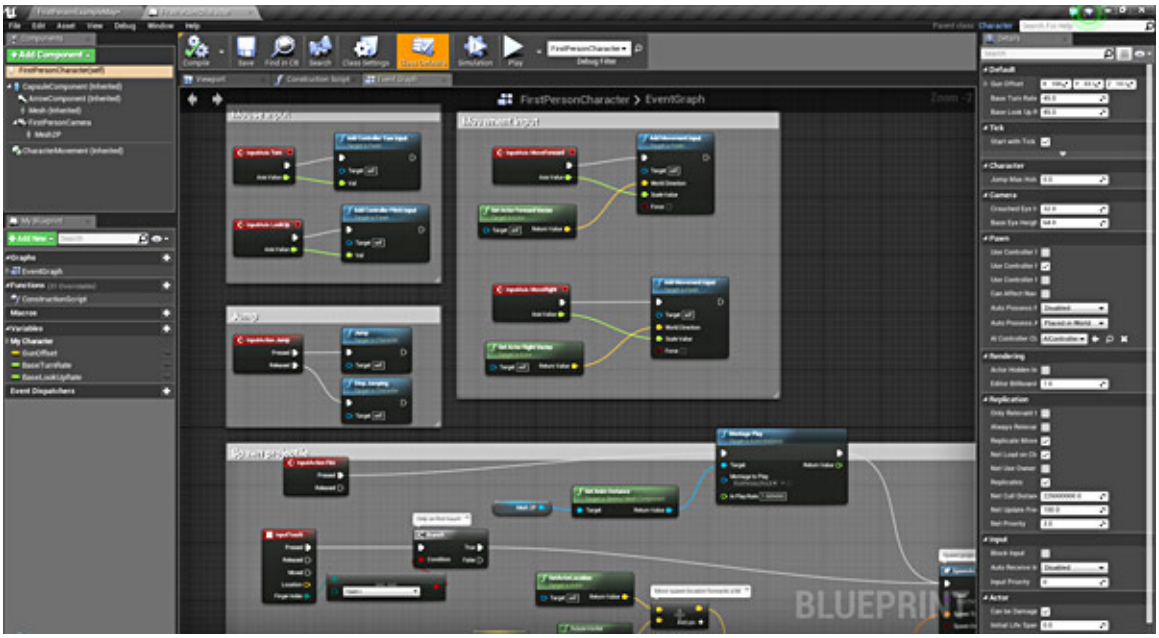


23. BLUEPRINT REPLACES KISMET

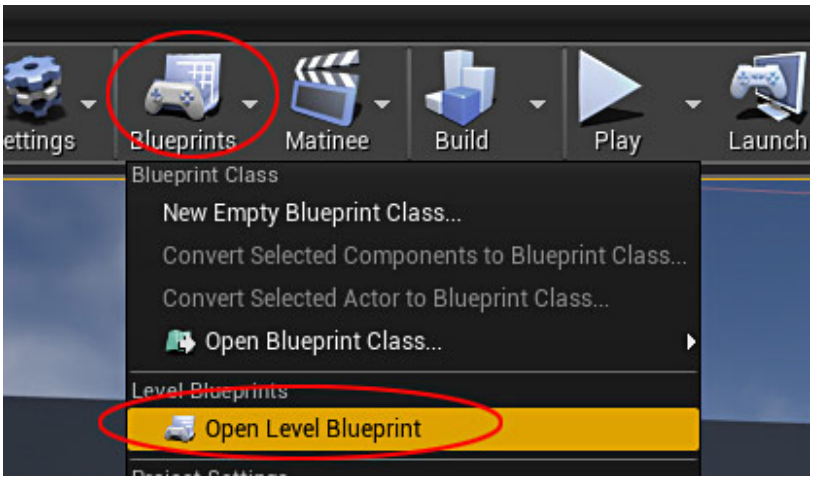
In UDK, you had Kismet, which is the visual scripting language in UDK:



In UE4, you now have Blueprint, which is a more powerful and improved system than Kismet.



To access Level Blueprint is through the main toolbar:

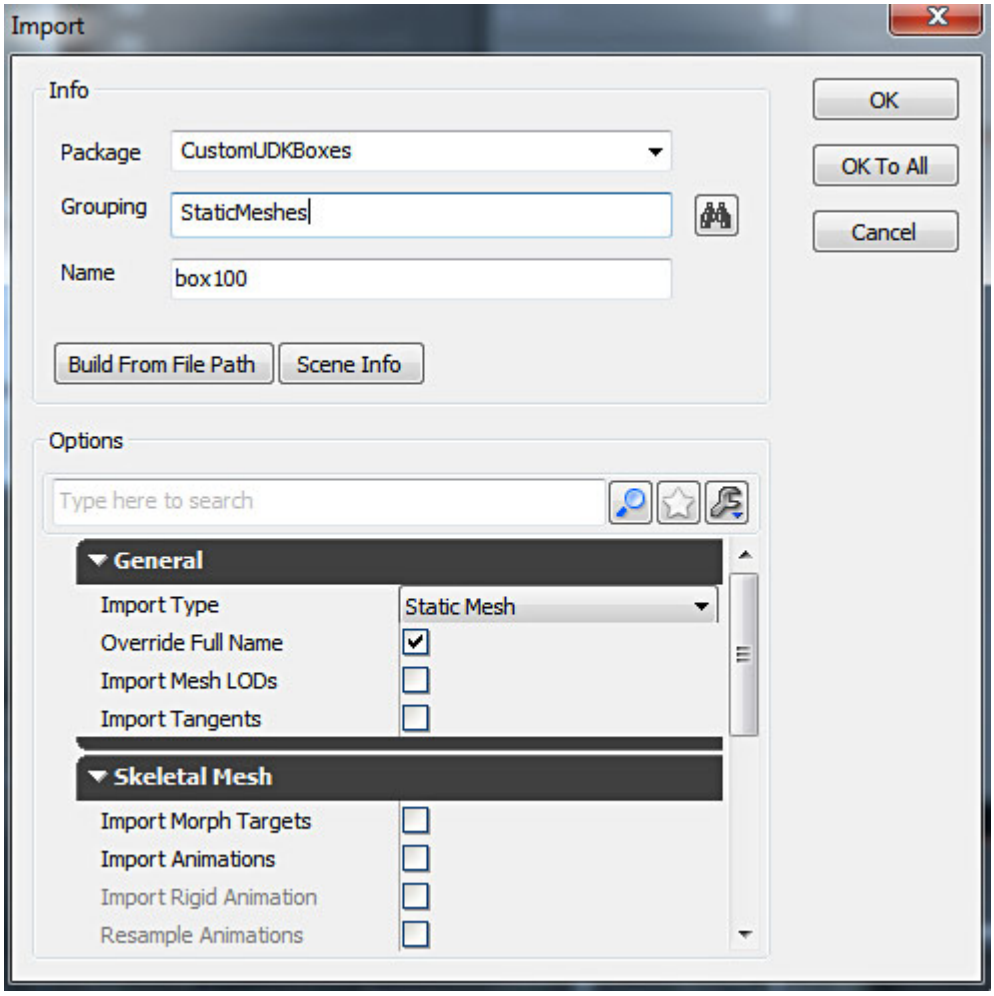


Blueprint has many improvements and functionalities. One of them is the ability to create **Class Blueprints**. These are self-contained items you create to have Blueprint functionality. You would then place Class Blueprints throughout the level without having to recreate it over again. Using Blueprint you can create simple actors, components, behaviors and gameplay mechanics without having to code.

Blueprint Documentation: https://docs.unrealengine.com/en-us/Engine/Blueprints

24. IMPORTING AND STORING CUSTOM ASSETS

In UDK, when you import any asset (Static Meshes, textures, materials, audio, animations etc.) you had to create a package where all of the imported content would go into.



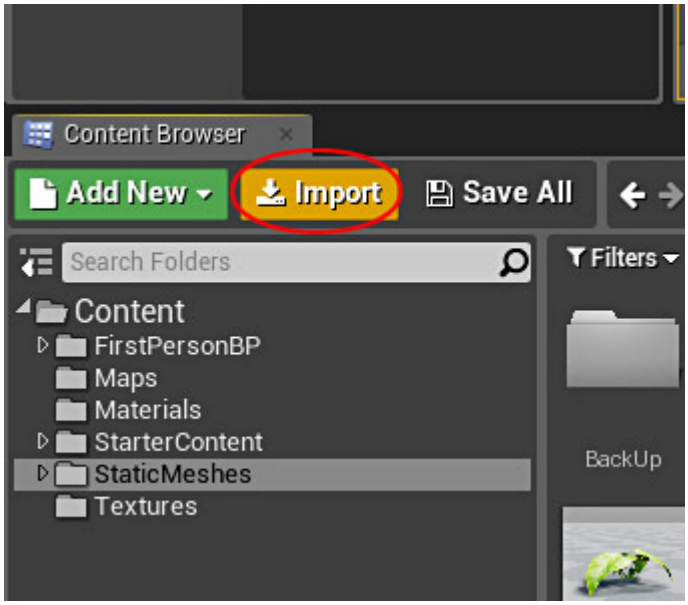
There were 2 most common file types .UDK and .UPK

- .udk = map file, this is your level that you see inside UDK
- .upk = package file, this file type contains 3d models (Static Meshes), textures, materials, sounds, animations etc that you use to place into your level

This system worked ok but I always wanted a better and more intuitive way to organize my imported content; such as a way to grab a single asset and

move it to another project without having to create a separate package to move it into.

In UE4, there are no more packages. Importing assets is very simple process. Hit import or just left-click and drag right into the Content Browser.



You now have .umap and .uasset:

- **.umap** = map or level file
- **.uasset** = individual asset file, such as a texture, a Static Mesh, a material etc.

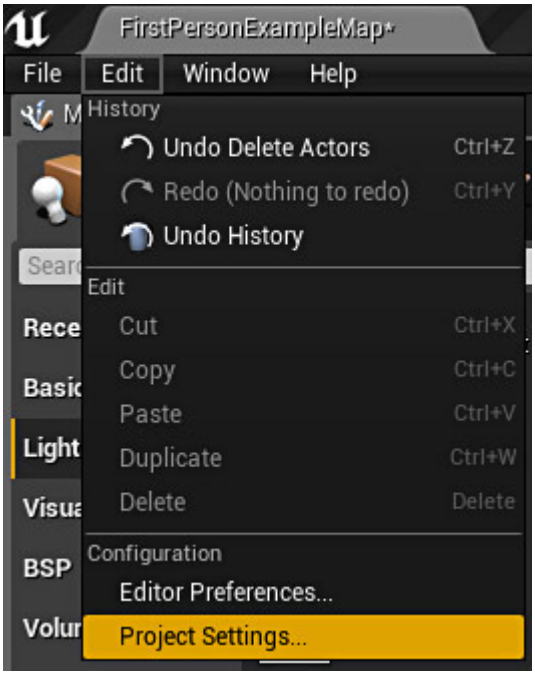
Each project has a Content folder. This is where your game assets are stored in. You can see the Content folder when you are inside the editor looking at the Content Browser. Anything you import for a project needs to be placed inside this Content folder. You can create sub-folders within the Content folder for better organization.

25. PROJECT SETTINGS

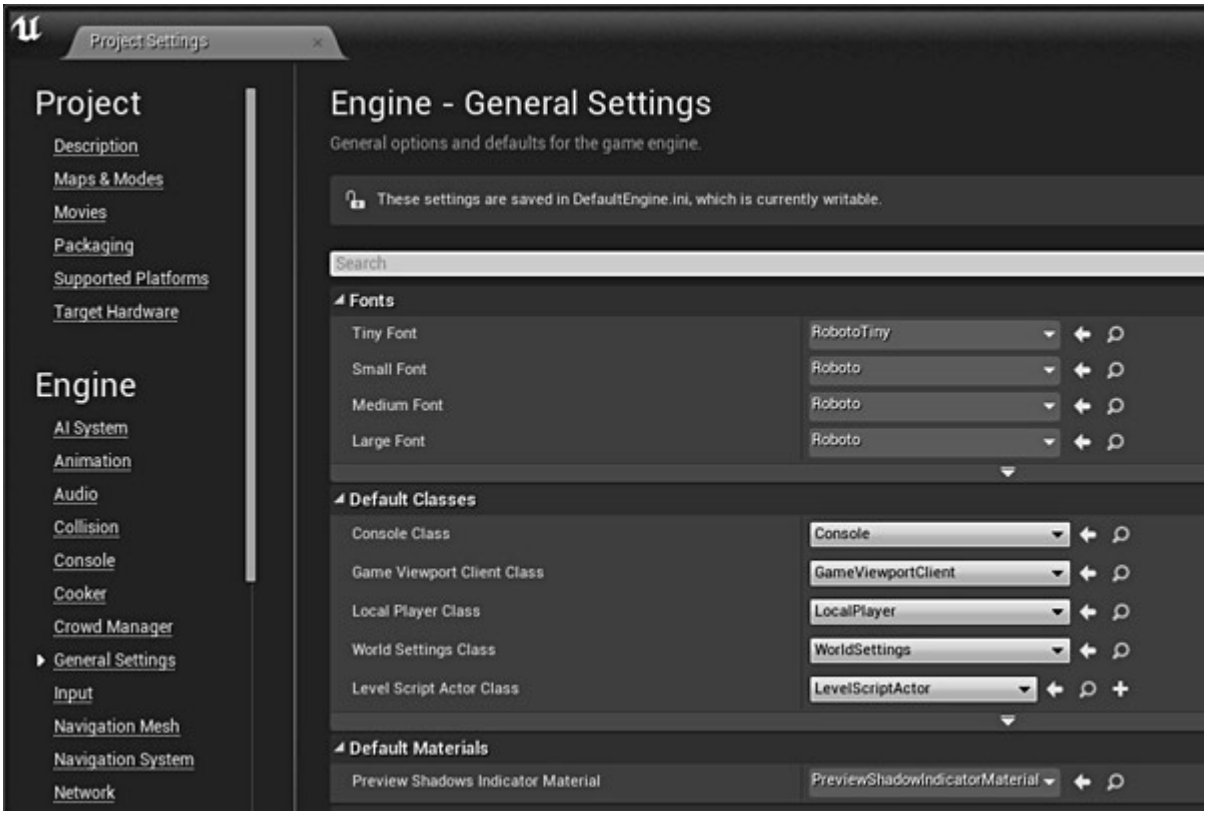
In UDK, game and level configurations were stored inside .ini configuration file. You would have to create and update this file with any changes.

In UE4, you can change your configurations inside Project Settings and you no longer need to create .ini files.

To access Project Settings go to **Edit → Project Settings**:



Through here you can change settings for Project, Engine and Editor:



26. CURRENT PROJECT IN UDK TO UE4

If you have a UDK project that you want to bring over to UE4. Here are some tips of what you can or cannot do:

- You can't open .upk or .udk files inside UE4
- Static Meshes and textures have to be re-imported. This should be simple if you kept your texture files and 3d modeling files (.fbx, .obj)
- Due to scale and different size use from UDK to UE4, Static Meshes used in UDK and if imported into UE4 will be smaller. You would have to scale them up in 3d modeling app or in UE4
- Materials have to be recreated following PBR workflow
- Particles Effects/Systems have to be recreated
- Kismet set-ups have to be recreated in Blueprint
- Any code has to be re-written from Unreal Script to C++
- Landscape heightmaps can be exported from UDK and imported into UE4

RECOMMENDED EXTERNAL LINKS

Epic Games - Transitioning From UE3 to UE4

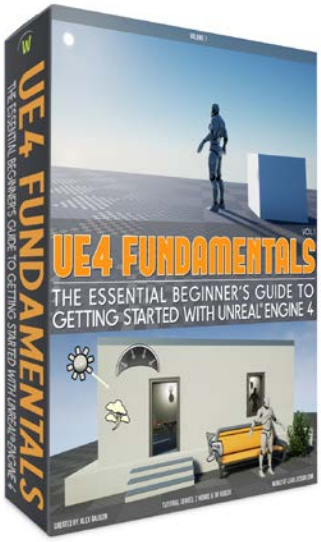
Unreal Engine Wiki UE3 to UE4 Transition Guide

From Unity3D to UE4

PREMIUM TUTORIAL GUIDES FOR UE4

This “UE4 Beginner’s Quick Start Guide” will get you started with Unreal Engine 4. But you’ll need more. I’ve created complete tutorial series below for deep dive into UE4 game engine and how to use it.

“UE4 FUNDAMENTALS VOL.1”



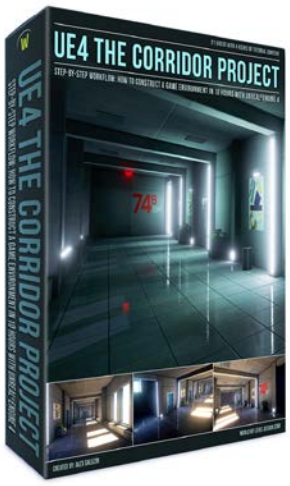
The essential complete beginner’s guide to learning and using Unreal® Engine 4 without any prior knowledge or experience.

Tutorial Series Includes:

- All New! Updated and revised
- 38 videos (7 hours)
- Instant Digital Download

Get the UE4 Fundamentals Vol. 1...

“UE4 THE CORRIDOR PROJECT”



UE4 The Corridor Project is an intermediate tutorial guide focused on constructing a game environment with provided custom Static Meshes. It’s in-depth guide for putting together an environment from start-to-finish.

Tutorial Series Includes:

- 21 videos (4 hours)
- Instant Digital Download

Get the UE4 Corridor Project...

Premium Tutorials come with WoLD 30-Day Money Back Guarantee!