

The Ultimate UNREAL

Beginner's Guide

TIPS, TRICKS AND TECHNIQUES TO MASTER UE5



Contents

Level Editor	1
Actors	2
Content Browser	3
Hotkeys	4
Material Editor	5
Static Mesh	6
Project Templates	7
Preferences	8
Marketplace	9
Searching	10
Plugins	11
Project Settings	12
Modes	13
C++	14
Blueprints	15
Tips & Tricks	16
Nanite	17
Lumen	18
Megascans	19
Quixel Mixer	20
Join Us!	21
Need Help?	22

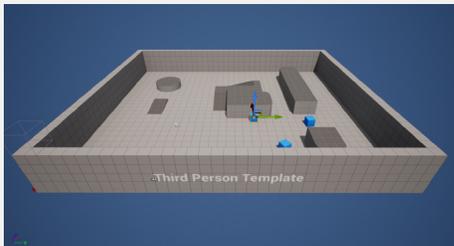


Level Editor

The **Level Editor** includes a range of tools for placing and manipulating objects, building environments and setting up game play mechanics. It is the main window of Unreal Engine.

Viewport

Main area of the interface where you can view and interact with your game world.



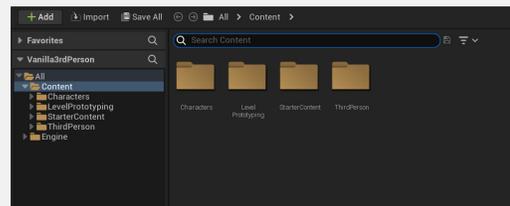
Toolbar

Located at the top of the interface and contains a variety of tools and options for working with the engine.



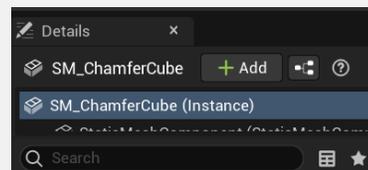
Details panel

Shows information about the currently selected object or element.



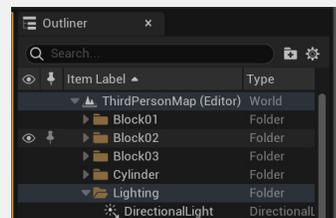
Content browser

Central location to manage assets in your project.



Outliner

Displays all the objects within the scene in a selectable hierarchy.

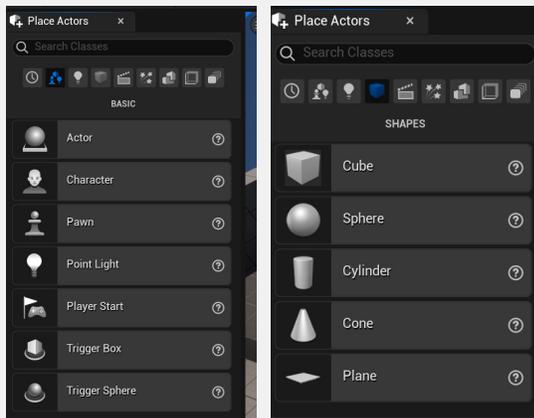


Actors

Actor is the term for anything that can be placed in a level. They can be static or dynamic. Actors support 3D transformations such as translation, rotation and scaling.

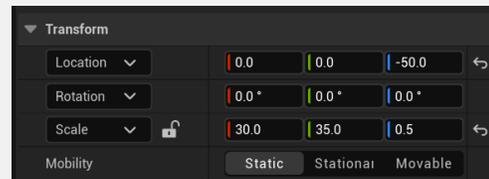
You can add a *Place Actors* panel from **Window > Place Actor**. The panel will show up on the left side of the editor window. Drag and drop the actors from the panel into the scene.

There are different panel types available in the different tabs along the top.



After adding an actor to the scene you can click on it and see the details about it.

In the **transform section** at the bottom you can choose which type of actor that object will be.



Static is for Actors that are not intended to move or update during game play.

Stationary means they can change during game play but not move.

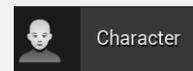
Movable means an actor can be added, removed or moved during game play.



An Actor is anything that can be placed in a level



A Pawn is an actor that can be controlled.



A Character is a humanoid pawn.

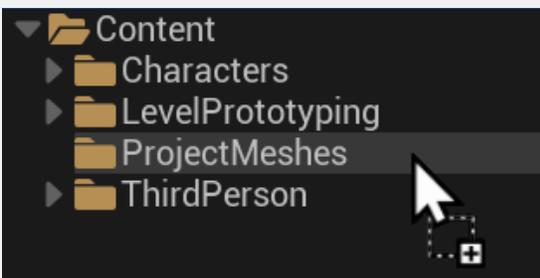


Content Browser

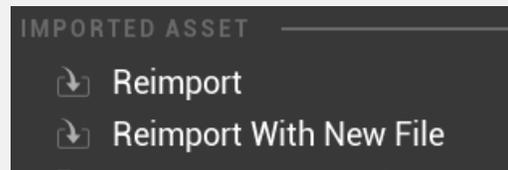
The **Content Browser** is where you manage the assets in your project. The **Content Drawer** button is located at the bottom toolbar in the editor. It opens a temporary Content Browser. If the Content Drawer is *Docked in Layout* it stays open.

Assets can be **dragged and dropped** into the Content Browser or imported with the Import button on the top left of the window.

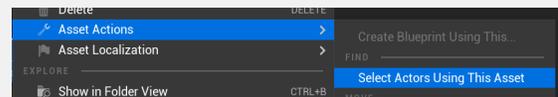
Any folders and items will import with the same folder structure into the folder highlighted by the mouse.



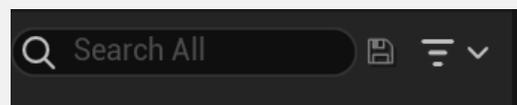
Inside the right click menu under *Imported Asset* you can **reimport your asset** if it has changed or also reimport with a new file to replace your asset in the scene.



Right click on an asset and choose **Asset Actions > Select Actors** and find all the places the asset is referenced in the scene.

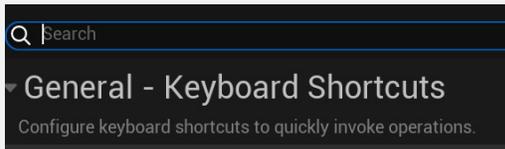


You can **search the Content Browser** with the search menu. Next to the search bar is a save icon so you can save your searches. You can also use the filter drop-down to filter your searches.



Hotkeys

Hotkeys are a way to speed up your workflow in Unreal Engine. If you want to add or change hotkeys you can go to **Edit > Editor Preferences > General > Keyboard Shortcuts**. Make sure to use the Search Bar to find the shortcuts you want to change.



Asset Management

F2 - Rename selected item.

Level Editor

Ctrl E - Open selected item in the Static Mesh Editor

H - Hide selected actors

Ctrl H - Unhide all actors

Ctrl Alt - Use a box selection

Viewport Hotkeys

Ctrl ` - Cycle between world and local coordinates

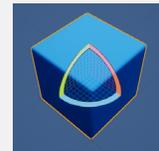
F - Focus on the selection

End - Snap to floor

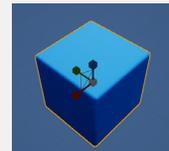
Space - Cycle between Translate, Rotate and Scale



W for Move



E for Rotate

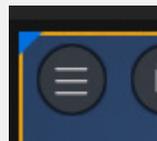


R for Scale

Bookmarks

You can bookmark up to **10 locations and orientations**. Set your view the way you want and hit the control key plus a number key.

To call back the view assigned to the number **hit the number used** to set up the bookmark.



A list of bookmarks are in the Viewport options menu on the top left.

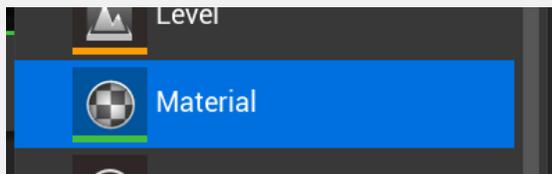


Material Editor

The **Material Editor** is a tool that creates and modifies the materials used on 3D models in Unreal. The Material Editor is node based and the nodes are connected to create anything from simple to complex materials.

Material: Is built with nodes in the material editor.

Material Instance: Is built from a parent material and can be modified without recompilation of the material.



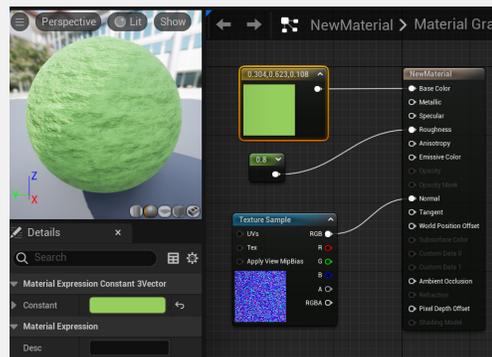
To **create a new material** right click in the content browser and choose Material. You can then double click the new Material to add nodes in the Material Editor.

Some Basic Nodes

Constant3Vector: This is a **basic color node**. Click the black area and pick a color.

Texture Sample: This node contains an **image texture**. If you drag an image into the Material Editor it will be in a texture sample node.

Constant: Single **numerical value**.



The material node setup above contains:

Constant3Vector for color

Constant for roughness

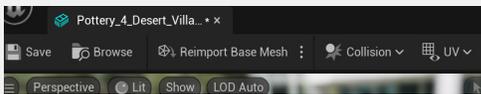
TextureSample for normal map



Static Mesh

The **Static Mesh Editor** is a tool to preview and manipulate the properties of static meshes. A static mesh is a **model used for world geometry**. The editor allows you to modify the look, collision and UVs of the static meshes in Unreal.

To open the editor **double click a static mesh** inside the Content Browser/ Content Drawer.



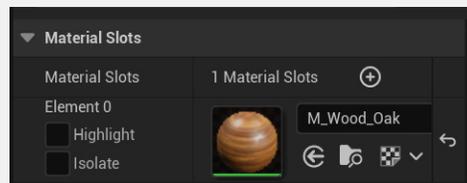
Save: Save the static mesh.

Browse: Open the content browser and find another static object to modify.

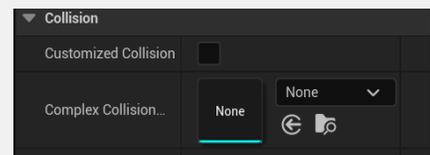
Reimport Base Mesh: Allows you to re-import the mesh to update it if it has been modified outside of Unreal.

Collision and UV: Allows you to modify collision and UV.

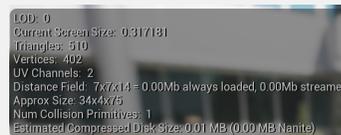
To **change the material** in the static mesh editor find the Material Slots in the top right of the window. Select the material name and in the drop down pick the new material.



To **change the collision** settings you can click "Customize Collision" and pick a static mesh you want to use for a collision.



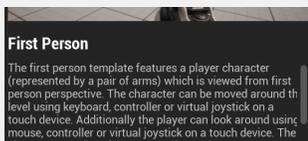
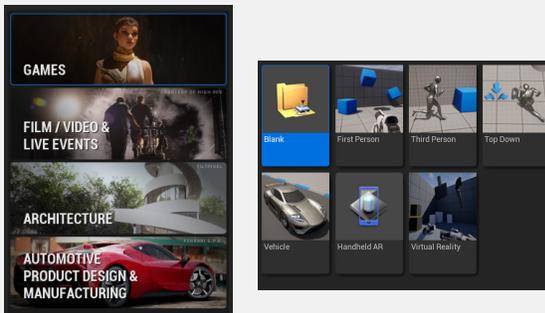
You can see the details of your mesh in the upper left corner of the Static Mesh Editor viewport.



Project Templates

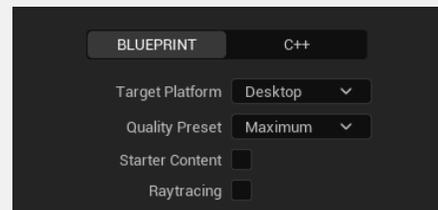
Unreal templates are available when you set up a new project. Click the different tabs to see the many options that are available. In the templates many parts of the project will already be set up. Use these templates to get a jump start on your project.

There are four ready-made template sections you can choose



Each template has a description and options for your game.

Most of the options for projects can be modified later in **Edit > Project Settings**.

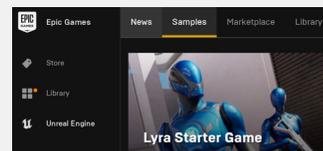


If you aren't sure if you want **Blueprints or C++** you can pick one and add the other one later.

Target platform selects if you are planning to publish to Desktop or Mobile. Quality should be maximum for PC or Console and Scalable for mobile.

Starter content includes extra content to use for designing your project.

Ray tracing will be enabled or disabled.



There are also many **Sample projects** available in the Epic Game Launcher.

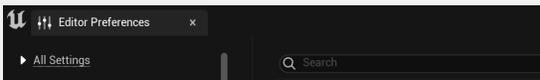


Preferences

To change your **Preferences** in Unreal Engine go to **Edit > Editor Preferences**. These changes will affect your preferences in the Unreal Editor. It is also where you can configure plugins that are in your project.

Editor Preference Categories:

- General
- Level Editor
- Content Editors
- Plugins
- Privacy
- Advanced



Editor Preferences has a search bar and is often the best way to find specific preferences.

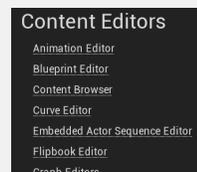
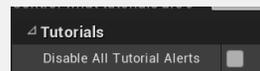
Commonly changed preferences

Level Editor > Keyboard Shortcuts - Change keyboard shortcuts.

General > Loading & Saving - here you can choose to load the project default or the last loaded level on startup.

General > Tutorials - To turn off the new tutorial alert.

Engine > General Settings - Scroll down to the Screenshots category and choose your default screenshot directory.



Content Editors - Configure the look and feel of the editors in Unreal.

Plugins - Configure the Plugins you have activated.

General > Appearance > Graphs You can customize the graphs you see in the Material and Blueprint editor.

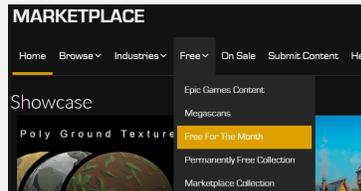
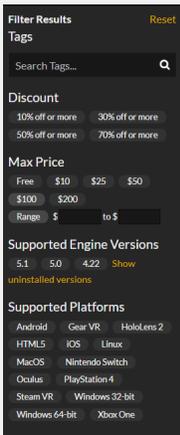


Marketplace

The **Marketplace** is an online store where content creators connect with developers and offer content and code. It contains both free and paid content and is available as a tab on the Unreal Engine part of the Epic Launcher.

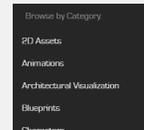
The Marketplace has **monthly free content** you can find under Free > Free For The Month.

There are many other free items you can find in the Free tab



When you **search in the Marketplace** there are many search options to help you find what you need.

To **find content** you can browse the different sections of the Marketplace.



To **purchase an asset** add it to your cart and checkout. Once you checkout you will find the asset in your **Unreal Engine Library**.



If you purchased a project you will have a **Create Project** button.

If you can add it to your project you will have an **Add To Project** button.



When you select Create Project it will create a new project.

When you select Add To Project it will ask you which project you want to add it to. You can select a currently open project.

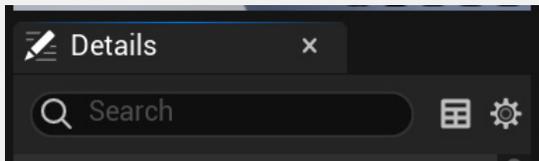
After processing, your project will have the **purchase in the content browser**.



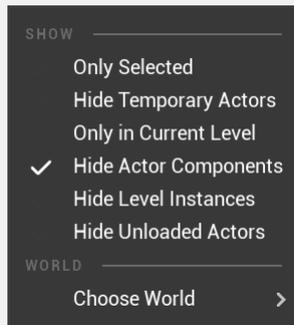
Searching

Searching in Unreal is a very powerful tool. It allows you to find and locate specific assets, code snippets or other resources within the engine by using keywords, phrases and filters to narrow down your search results.

Most windows in Unreal Engine have **search bars**, including all settings windows and editors.



You can use the **setting gear** to narrow your search.



Unreal **supports sophisticated searches** and allows searches of asset metadata.

Operators are based on normal search syntax using standard arithmetic operators.

Strings can be quoted or not quoted.

Special Keys are also available for specifically Unreal functions.

Name

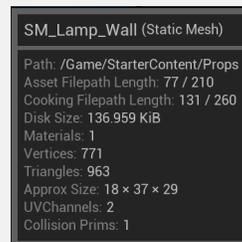
Path

Class

Collection

Hovering over an asset brings up the *Metadata*. **Metadata names** can be used for search arguments using the syntax

[Metadata Name] [Operator] [String or Numeric Value]



In this case **UVChannels=2** would bring up this asset.

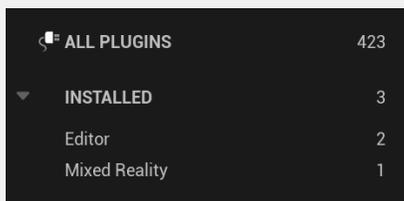


Plugins

Plugins are modular packages that extend the functionality of the Unreal Engine.

There are many built-in plugins created by Epic Games but they are not installed by default. For performance, only install the plugins you actually need for your project.

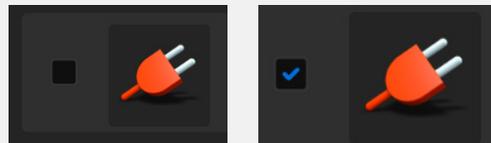
Find the Plugin menu with **Edit > Plugins**. All the installed plugins and the built in plugins will be listed there.



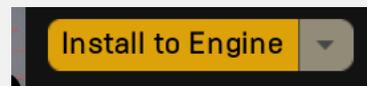
When installed, plugins are in the **Plugins subfolder** in your project's directory.

If you do not see them, select View Options and check Show Engine Content and Show Plugin Content.

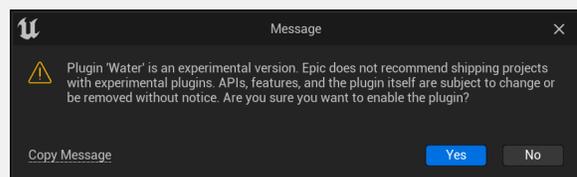
To **install a built-in plugin** just check the box in front of the plugin. Unreal will confirm you want to enable the plugin. After a plugin is enable you may be asked to restart Unreal Editor.



To **install plugins from the Unreal Marketplace** check out (from your cart) the plugin. It will then be in your Library in the Epic Games Launcher. You press "Install to Engine" and your plugin will be available to be enabled in that Unreal build.



Some plugins are Experimental and Unreal does not suggest games are shipped with these plugins enabled.



Project Settings

Project Settings are a set of configuration options that are applied to the entire project. There are options that affect how the engine behaves when running your project and how it should behave on different platforms. The Project Settings options are accessed through **Edit > Project Settings**.

There are 6 categories in the Projects Setting options.

Project: Describes how the project behaves.

Game: Asset management and tools.

Engine: Specific system and sections of the engine.

Editor: How the editor should look and feel on a per-project basis.

Platforms: Options for the platforms your project can run on.

Plugins: Options for select plugins.

In **Project - Description** you can set your project description, name and version. You can also set the thumbnail to be shown in the Epic Games Launcher under Project Thumbnail.

▼ Project - Description

Descriptions and other information about your project.

In **Project - Maps & Modes** you can set the editor startup map and game default map.

▼ Project - Maps & Modes

Default maps, game modes and other map related settings.

In **Platforms** are the settings for the specific platforms. This contains unique settings for those platforms.

▼ Platforms - Android

Project settings for Android apps

Under the name of the section it tells you where these **settings are stored**. The settings can be locked so that they can not be changed.

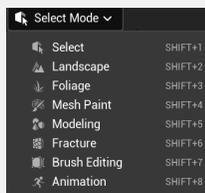
🔒 These settings are saved in DefaultEngine.ini, which is currently writable.



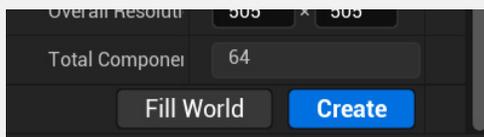
Modes

Level Editor Modes are powerful editing modes for specialized interfaces. They are used for different tasks related to creating and editing levels. The toolbar at the top of the Viewport has a Select Mode with a drop down menu where you can pick a Mode to work in.

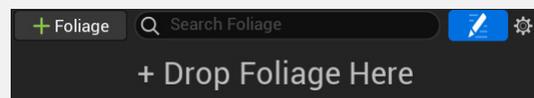
Select Mode is the mode for selecting Actors in the scene. This is the default mode.



Landscape Mode is for Landscape terrains and you will get options for creating and modifying a Landscape. The create button is at the bottom of the window.

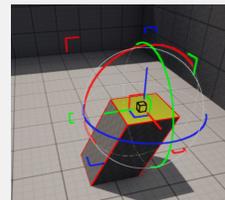


Foliage Mode is for painting foliage. You will have to add foliage static meshes to your project to use them. You can find some in the starter content and free and paid foliage in the Marketplace.



Mesh Paint is to paint vertex colors and textures on Static Mesh Actors in the viewport.

Modeling is used for Model editing mode. You can do some editing of models in Unreal.



Fracture is for creating destructible objects and environments.

Brush Editing allows you to modify Geometry Brushes.

Animation is for the Animation editing mode.



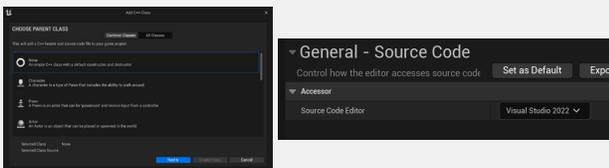
C++

C++ is the coding language for Unreal Engine. You can use both C++ and Blueprints in any project. Whichever default you picked Unreal Engine will update your project if you choose to use the other at any time in your project.

You can **create a new script** by using **Tools > New C++ Class**. Then choose which parent class you will use.

The script will then open in your IDE (integrated development environment) such as Visual Studio.

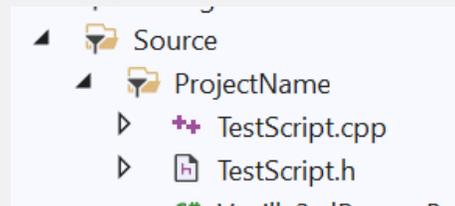
You can **make sure your IDE is connected to Unreal** in Edit > Editor Preferences under General - Source Code



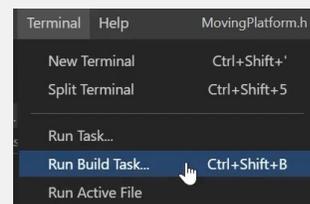
A .cpp and a .h script will open in your **script editor**.



They will be located where you designated when setting up the script.



The **.h file is a header file** where you declare variables and functions. The **.css file is the implementation file** and contains function definitions and object definitions.



In order to use C++ in your game you have to **recompile your game** when you change your scripts.

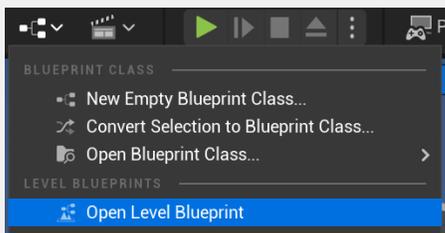


Blueprints

Blueprints is a visual scripting system in Unreal Engine. It is node based workflow so you don't have to write any code. Blueprints are designed to be a complete system and allow you to create powerful gameplay elements.

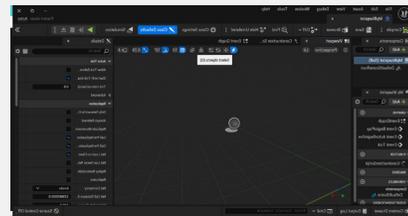
The two most common Blueprint types are **Level Blueprints** and **Blueprint Classes**.

Level Blueprints are a specialized type of Blueprint that is specific to the level. You can find the level blueprint in the toolbar at the top of the Level Editor.

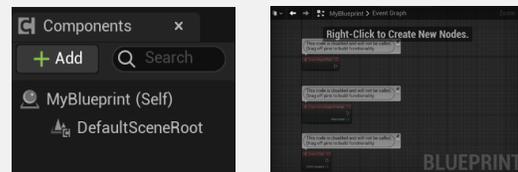


Blueprint Classes are used to add functionality on top of existing gameplay classes.

When you open the **Blueprint Editor** by double clicking on a Blueprint class it defaults to the Viewport which shows any objects that belong to the class.



On the left side of the Blueprint Editor is the **Components tab** which tells you what is inherited in the Blueprint you selected.



Select the tab **Event Graph** to set up your Blueprint.

Event BeginPlay is triggered when the game is started

Event ActorBeginOverlap happens when Unreal sees the actor as colliding with another actor that has Generate Overlap Events set to true.

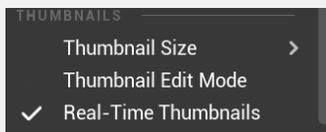
Event Tick happens every frame.



Tips & Tricks

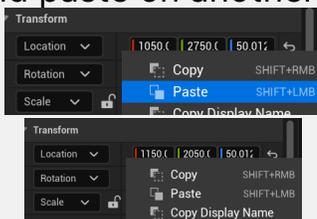
Edit Content Browser Thumbnails

At the bottom of the Content Browser Settings you can go into Thumbnail Edit mode and change your thumbnails.



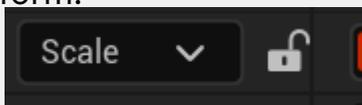
Copy Fields

Instead of copying the X, Y and Z values individually you can right click on a field and copy all the values within that field. Then right click and paste on another field.



Uniform Scaling

With your object(s) selected it is simple to scale evenly. Toggle the lock icon next to the Scale Transform.



Using Macros

On a Blueprint you can select multiple nodes and collapse them to a macro by right clicking and selecting Collapse to Macro. This is a great way to reduce duplicate logic and tidy up your blueprints.



Performance and FPS

To see your computer's performance and current FPS go to Edit > Editor Preferences and under 'Performance' enable 'Show Frame Rate and Memory'. It will then appear in the top right hand corner of your window.

FPS: 116.2 / 8.6 ms Mem: 1,936.07 mb Objs: 34,196 Stalls: 1

Camera Follow

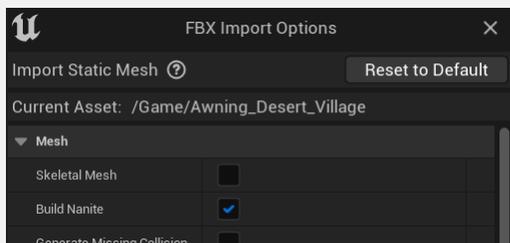
Holding shift while moving an object will allow the camera to follow this object throughout the scene wherever you move.



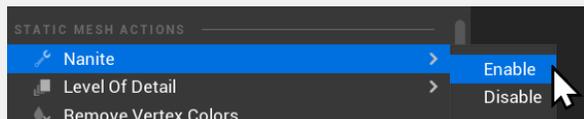
Nanite

Nanite is a virtualized geometry system that was released with Unreal 5 to optimize the geometry of meshes. When applied to meshes it shows only the detail that can be perceived. Unreal suggests Nanite is used for most meshes.

Nanite can be **enabled** when the mesh is imported by checking Build Nanite under the Mesh drop-down menu.

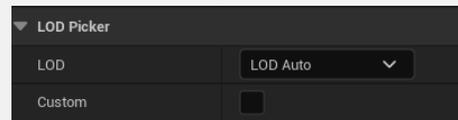


It can also be **enabled after import** by right clicking the mesh in the Content Browser and selecting Nanite > Enable.

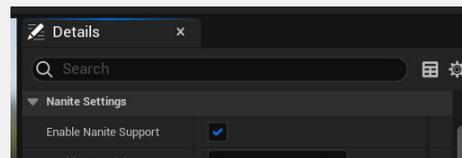


An LOD is an acronym for **Level of Detail**. It is used to simplify meshes that are farther away from the camera to help with performance.

Traditionally, meshes are imported with multiple resolutions created by the developer. **With Nanite the LODs are calculated automatically** rather than manually set.



There are many settings that can be changed for Nanite. You can find many of them in the **Tools > Nanite menu** and in the Static Mesh Editor Details Panel under the Nanite Settings section.



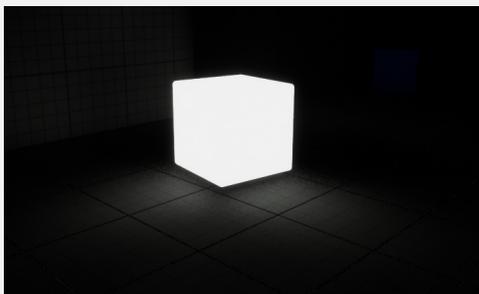
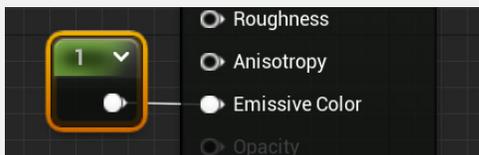
Nanite is a new technology and it is a good idea to continue to check the documentation on it with each new Unreal build.



Lumen

Lumen is the new illumination and reflection system created for Unreal 5. It allows for the creation of realistic lighting environments with dynamic lighting, soft shadows and reflections. Lumen is automatically enabled when you start a new Unreal 5 project.

With Lumen, objects set with an Emissive Color material node glow. This did not happen in Unreal 4.



One thing to remember about Lumen is that Unreal suggests you **use simple interiors**. Walls, floors, ceilings and furniture should all be separate meshes.

Also remember to **use Nanite with Lumen** to keep the frame rate of your project as high as possible.



Image from Valley of Ancients Sample Project

Lumen is set up to be usable out of the box and **you don't need to tweak any settings** to get amazing results.

▼ Engine - Rendering

Rendering settings.

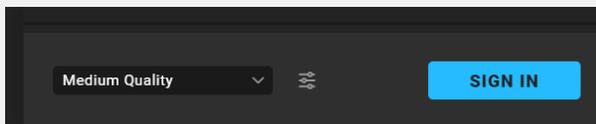
Edit > Project Settings > Engine - Rendering is where you will find the Lumen settings.



Megascans

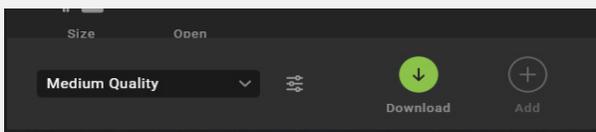
Megascans is a library provided by Unreal of high-resolution surface, vegetation and 3D scans. They are easily imported into projects along with all the textures needed. They are built into Unreal and are available at **Window > Quixel Bridge**.

The first time you open Quixel Bridge you will have to **Login**.



After logging in you get the option of **downloading any assets** you want from the library.

Once downloaded the asset is available to add to your scene.

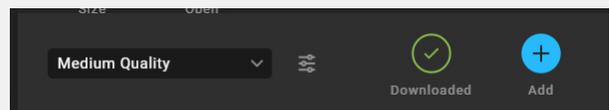


Quixel Bridge will open in a different Unreal window. You can maximize this window and **search for scans**.

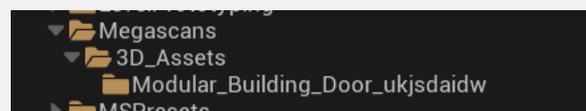
There are five ways to find assets.

- Home (all assets)
- Collections
- Metahumans
- Favorites
- Local (downloaded)

Once you download an asset you can easily add it to your project with the **plus button**.



You will find the asset inside the Content Browser in the Megascans folder. You can **drag and drop** your scans into your project scene.



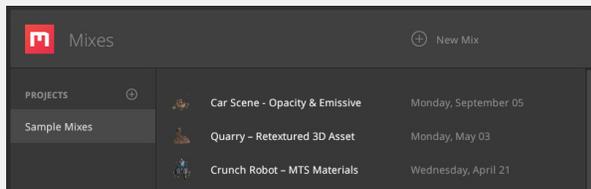
Quixel Mixer

Quixel Mixer is a standalone program that is available to Unreal users. It is a very powerful program that can paint and texture 3d models. When paired with Megascans it can create stunning assets for Unreal projects.

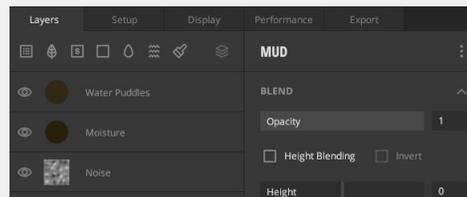
The Quixel website (quixel.com/mixer) has the download.

After installation you can sign in with your Epic login. An initial dialogue will popup and ask you to pick a Sample Mix.

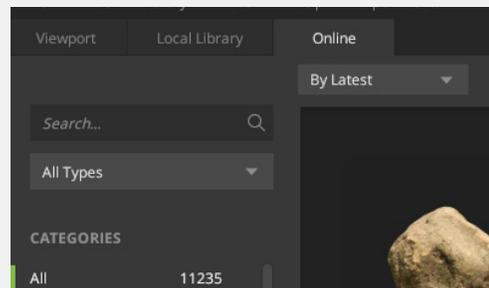
These are projects that are set up for learning. You can also click the **+New button** at the top right to start a new project.



When you click on the **layers tab** you can see each layer and layer group on the left. Each layer has its own information with sliders



At the top left in the Online tab you can find all the **Megascan models and textures** available for modification.



Quixel Mixer is a very powerful way to modify textures and use Megascans. In the **Help Menu > View Manual** you can access the manual and get detailed instructions on making your first mix.



Join Us!

Ready to make epic games in Unreal?

Unreal Engine is a fantastic platform which enables you to make AAA-quality games. With these highly-rated courses, some created in collaboration with Epic Games, **you'll make your first game within hours!**

[Make an awesome first person shooter game!](#)

Weapons, combat effects, hostile enemy drones with pathfinding, HUD with countdown timers, health bars, announcements, game states, scores AND it wouldn't be a FPS game without health and ammo pickups. Your game will have it all!

[Learn how to code your first video game in under 2 hours!](#)

In our Unreal Blueprints Course, learn how to make games without writing traditional code by using Unreal Engine's powerful visual coding system, Blueprints. By the end of the first section, you'll have gone from not even having Unreal installed to making your first game: Marble Run!

[Learn C++ and make epic games](#), including a FPS and tank game, in the game development engine used by AAA studios and indie developers worldwide.

[Join us and start your Unreal journey!](#)



Need Help?

If you're not already part of our amazing community, why not come and say hi!



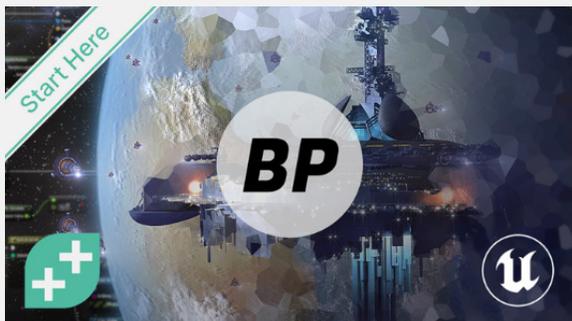
Some of our amazing Unreal courses!



[Unreal 5.0 C++ Developer: Learn C++ and Make Video Games](#)



[Unreal Engine 5 Blueprints: First Person Shooter \(FPS\)](#)



[Unreal Engine Blueprint Game Developer Online Course](#)



[Unreal Engine 5 Environment Design](#)

