Tools and Editors

An overview of the different types of Editors contained within Unreal Engine 5.

Unreal Engine 5 provides a combination of **tools**, **editors**, and **systems** you can use to create your game or application.

This page uses the following terms:

- A **tool** is something you use to perform a specific task, like placing Actors inside a level, or painting terrain.
- An **editor** is a collection of tools you use to achieve something more complex. For example, the **Level Editor** enables you to build your game's levels, or you can change the look and feel of Materials inside the **Materials Editor**.
- A **system** is a large collection of features that work together to produce some aspect of the game or application. For example, **Blueprint** is a system used to visually script gameplay elements.

> Sometimes, systems and editors can have similar names. For example, the Material Editor is used to edit Material assets, while the Material system provides the underlying support for using Materials in Unreal Engine.

Some of these tools and editors in Unreal Engine are built-in, while others come in the form of optional **plugins** that you can enable or disable depending on your project needs. To learn more about plugins, refer to the Working with Plugins page.

This page gives an overview of the major tools and editors you will be working with inside Unreal Engine 5. The use of various Unreal Engine tools is covered in detail in feature-specific documentation.

Whether you use the **Blueprint Editor** to script behaviors for the Actors in your level, or create particle effects with the **Niagara Editor**, a good understanding of what each editor can do and how to navigate each one can improve your workflow and help prevent stumbling blocks during development.

Level Editor

Gameplay Levels

The **Level Editor** is the primary editor where you construct your gameplay levels. This is where you define the play space for your game by adding different types of Actors and Geometry, Blueprints Visual Scripting, Niagara, and so on. By default, when you create or open a project, Unreal Engine 5 will open the Level Editor.

For more information, see Level Editor.

Static Mesh Editor

Static Meshes

You can use the **Static Mesh Editor** to preview the look, collision, and UV mapping, as well as set and manipulate the properties of Static Meshes. Inside the Static Mesh Editor, you can also set up LODs (or Level of Detail settings) for your Static Mesh assets to control how simple or detailed they appear based on how and where your game is running.

For more information, see Static Mesh Editor UI.

Material Editor

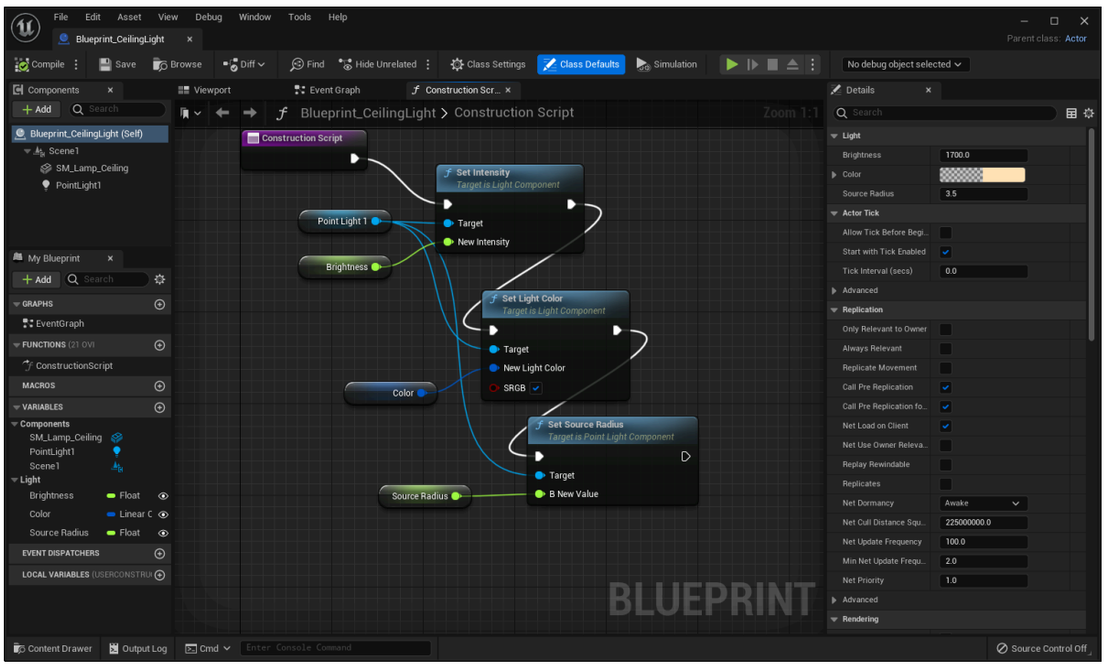
Materials

The **Material Editor** is where you create and edit Materials. Materials are assets that can be applied to a mesh to control its visual look. For example, you can create a dirt Material and apply it to floors in your level to create a surface that looks like it is covered in dirt.

For more information, see Material Editor Guide.

Blueprint Editor

Blueprints



Blueprint Editor inside Unreal Engine 5. Click for full view.

The **Blueprint Editor** is where you can work with and modify Blueprints. These are special Assets that you can use to create gameplay elements (for example, controlling an Actor or scripting an event), modify Materials, or implement other Unreal Engine features without the need to write any C++ code.

For more information, see Blueprint Editor Reference.

Physics Asset Editor

Physics

You can use the **Physics Asset Editor** to create Physics Assets for use with Skeletal Meshes. In practice, this is how you implement physics features like deformations and collisions. You can start from nothing and build to a full ragdoll setup, or use the automation tools to create a basic set of Physics Bodies and Physics Constraints.

For more information, see Physics Asset Editor.

Behavior Tree Editor

AI Behavior

The **Behavior Tree Editor** is where you can script Artificial Intelligence (AI) through a visual node-based system (similar to Blueprints) for Actors in your levels. You can create any number of different behaviors for enemies, non-playing characters (NPCs), vehicles, and so on.

For more information, see Behavior Tree User Guide.

Niagara Editor

Particle Effects

The **Niagara Editor** is for creating special effects by leveraging a fully modular particle effects system composed of separate particle emitters for each effect. Emitters can be saved in the Content Browser for future use, and serve as the basis of new emitters in your current or future projects.

For more information, see Niagara Key Concepts.

UMG UI Editor

User Interface

The **UMG (Unreal Motion Graphics) UI Editor** is a visual UI authoring tool that you can use to create UI elements, such as in-game head's up displays, menus or other interface-related graphics.

For more information, see UMG UI Designer Quick Start Guide.

Font Editor

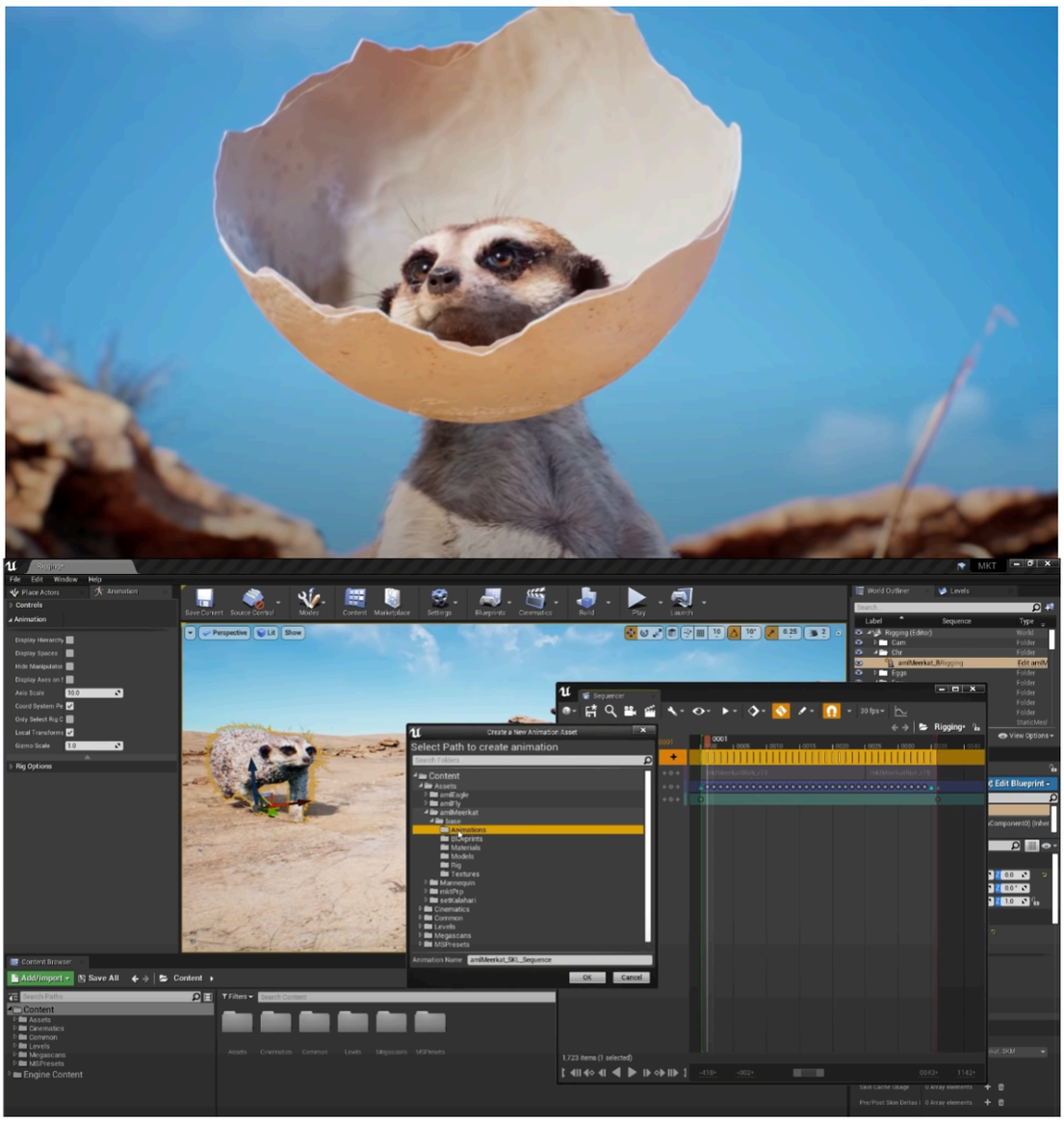
Fonts

Use the **Font Editor** to add, organize and preview Font Assets. You can also define font parameters, such as Font Asset layout and hinting policies (*font hinting* is a mathematical method that ensures your text will be readable at any display size).

For more information, see Font Asset and Editor.

Sequencer Editor

Cinematics and Dynamic Events



Sequencer was used in the production of the Weta Digital animated short, Meerkat. Click for full view.

The **Sequencer Editor** gives you the ability to create in-game cinematics with a specialized multi-track editor. By creating **Level Sequences** and adding **Tracks**, you can define the makeup of each Track, which will determine the content for the scene. Tracks can consist of things like Animations (for animating a character), Transformations (moving things around in the scene), Audio (for including music or sound effects), and so on.

For more information, see Sequencer Overview.

Animation Editor

Animation

The **Animation Editor** within Unreal Engine 5 is used for editing Skeleton Assets, Skeletal Meshes, Animation Blueprints, and various other animation assets.

For more information, see Animation Editors.

Control Rig Editor

Animation

Control Rig is a suite of animation tools for you to rig and animate characters directly in-engine. Using Control Rig, you can bypass the need to rig and animate in external tools, and instead animate in Unreal Editor directly. With this system, you can create and rig custom controls on a character, animate in Sequencer, and use a variety of other animation tools to aid your animating process.

For more information, see Control Rig.

Sound Cue Editor

Sound Cues

The behavior of audio playback in Unreal Engine 5 is defined within Sound Cues, which can be edited using the **Sound Cue Editor**. Inside this editor, you can combine and mix several sound assets to produce a single mixed output saved as a Sound Cue.

For more information, see Sound Cue Editor.

Media Editor

External Media Playback

Use the **Media Editor** to define media files or URLs to use as source media for playback inside Unreal Engine 5.

You can define settings for how your source media will play back, such as auto-play, play rate, and looping, but you can't edit media directly.

For more information, see Media Editor Reference.

nDisplay 3D Config Editor

Virtual Production and Live Events

nDisplay renders your Unreal Engine scene on multiple synchronized display devices, such as powerwalls, domes, and curved screens. With the **nDisplay Configuration Editor**, you can create your nDisplay setup and visualize how content will be rendered across all the display devices.

For more information, see nDisplay 3D Config Editor.

DMX Library Editor

Live Events

DMX in action. This screenshot is from a sample project by Moment Factory. Click for full view.

DMX (Digital Multiplex) is a standard for digital communication used throughout the live-events industry to control various devices, such as lighting fixtures, lasers, smoke machines, mechanical devices, and electronic billboards. In the **DMX Library Editor**, you can customize these devices and their commands.

For more information, see DMX.