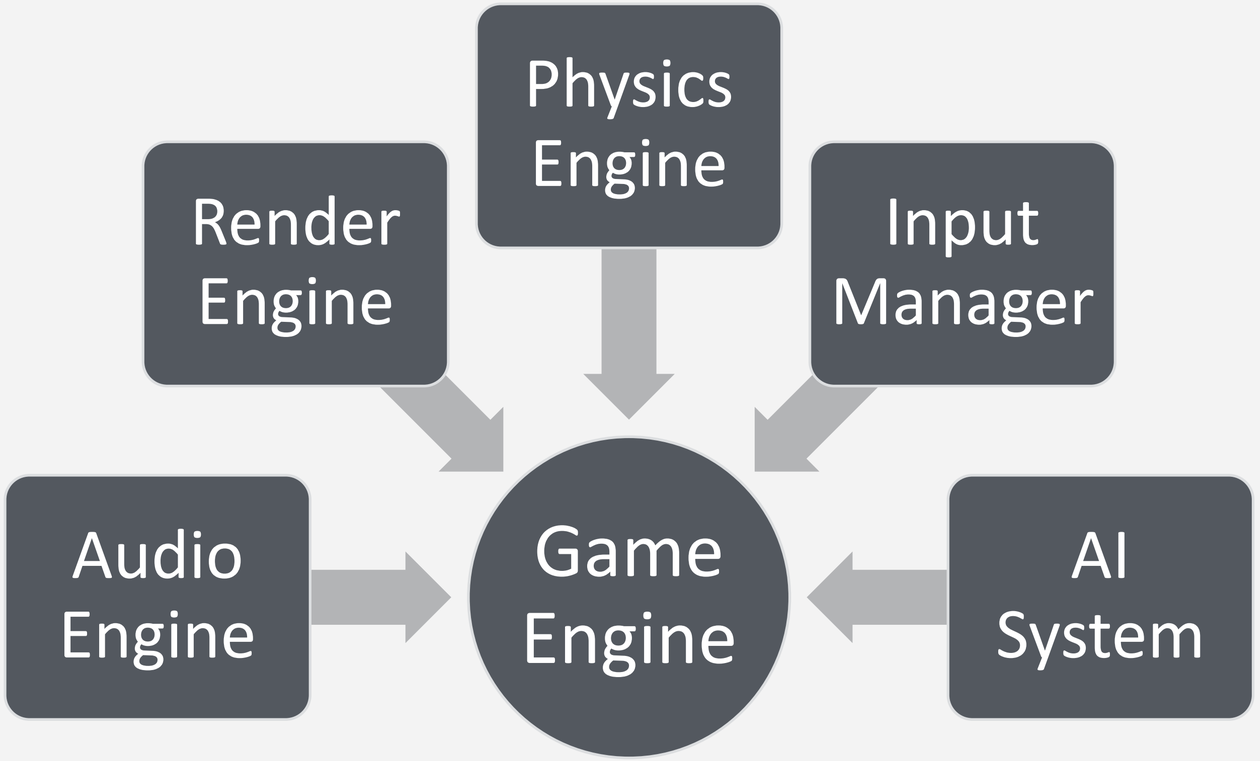
INTRO TO UNREAL ENGINE

An Overview of the Tools

Slides Courtesy of Chris Murphy of Epic Games

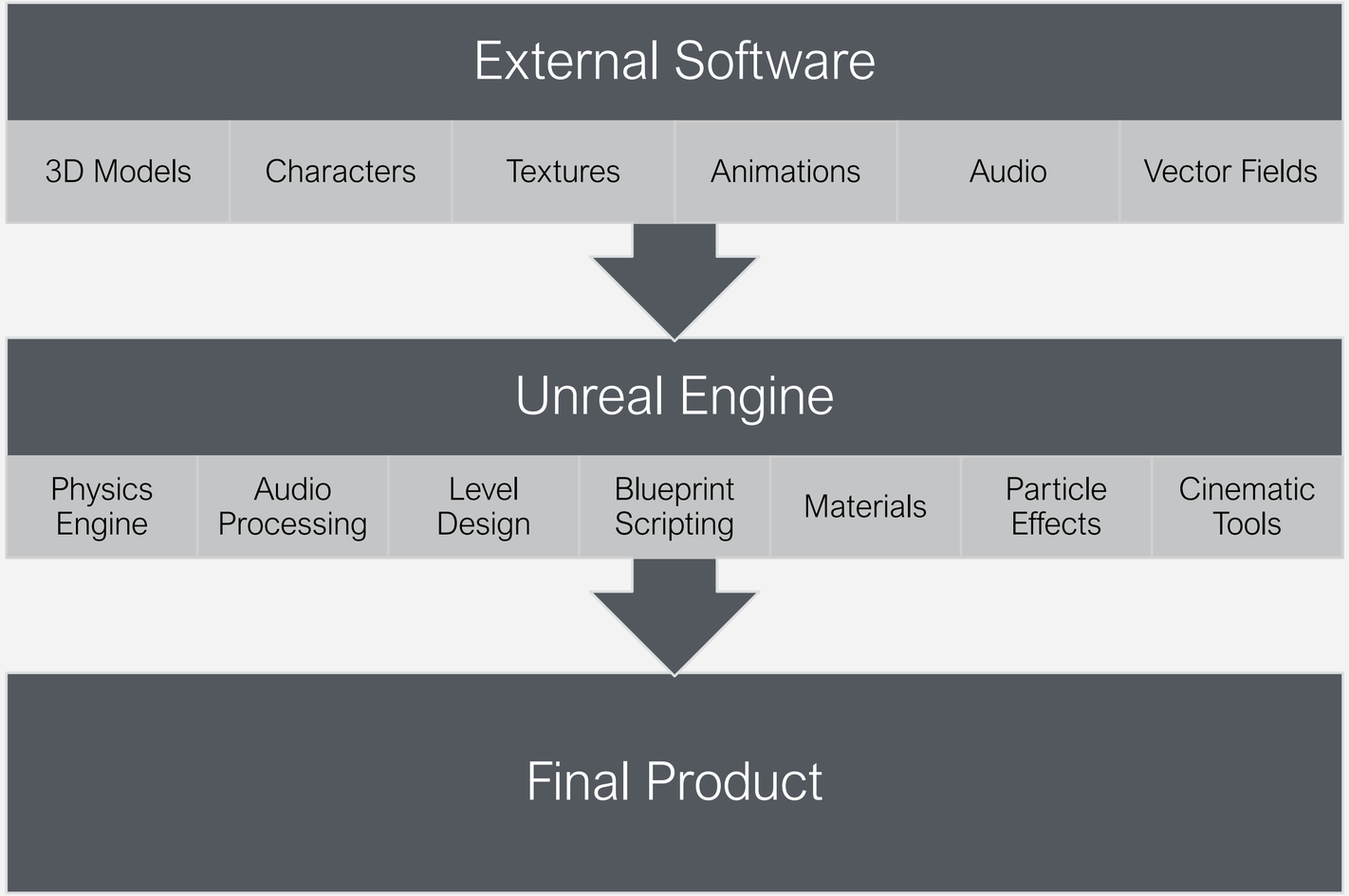
WHAT IS A GAME ENGINE?

A game engine is a framework and toolset that can be used to develop interactive experiences. While it’s referred to as a “game” engine, any interactive experience can be crafted upon this framework.



UNREAL ENGINE 4

Unreal Engine 4 (UE4) is a full suite of integrated tools for developers to use to design and build games, simulations, and visualizations. While many assets can be developed from within Unreal, most are created externally.



UNREAL ENGINE 4

Features and Technology

PHYSICALLY BASED RENDERING

Physically Based Rendering refers to a visual model that aims to realistically simulate surfaces. In Unreal Engine Materials are a a simplified format for storing complex rendering instructions developed through node networks in the Material Editor. When applied to a model in a scene materials change the appearance of the surface to match a variety of real world or fantasy materials.

PARTICLE SYSTEMS

Unreal Engine contains a powerful and robust particle system, allowing artists to create mind-blowing visual effects ranging from smoke, sparks, and fire to far more intricate and otherworldly examples.

Unreal’s Particle Systems are edited via Cascade, a fully integrated and modular particle effects editor. Cascade offers real-time feedback and modular effects editing, allowing fast and easy creation of even the most complex effects.

OPEN WORLDS

Creating massive terrain-based worlds for your projects is something that Unreal Engine 4 can do quite well. The Landscape system enables you to create terrain for your world—mountains, valleys, uneven or sloped ground, even openings for caves—and easily modify both its shape and its appearance by using a range of tools.

POST PROCESSING

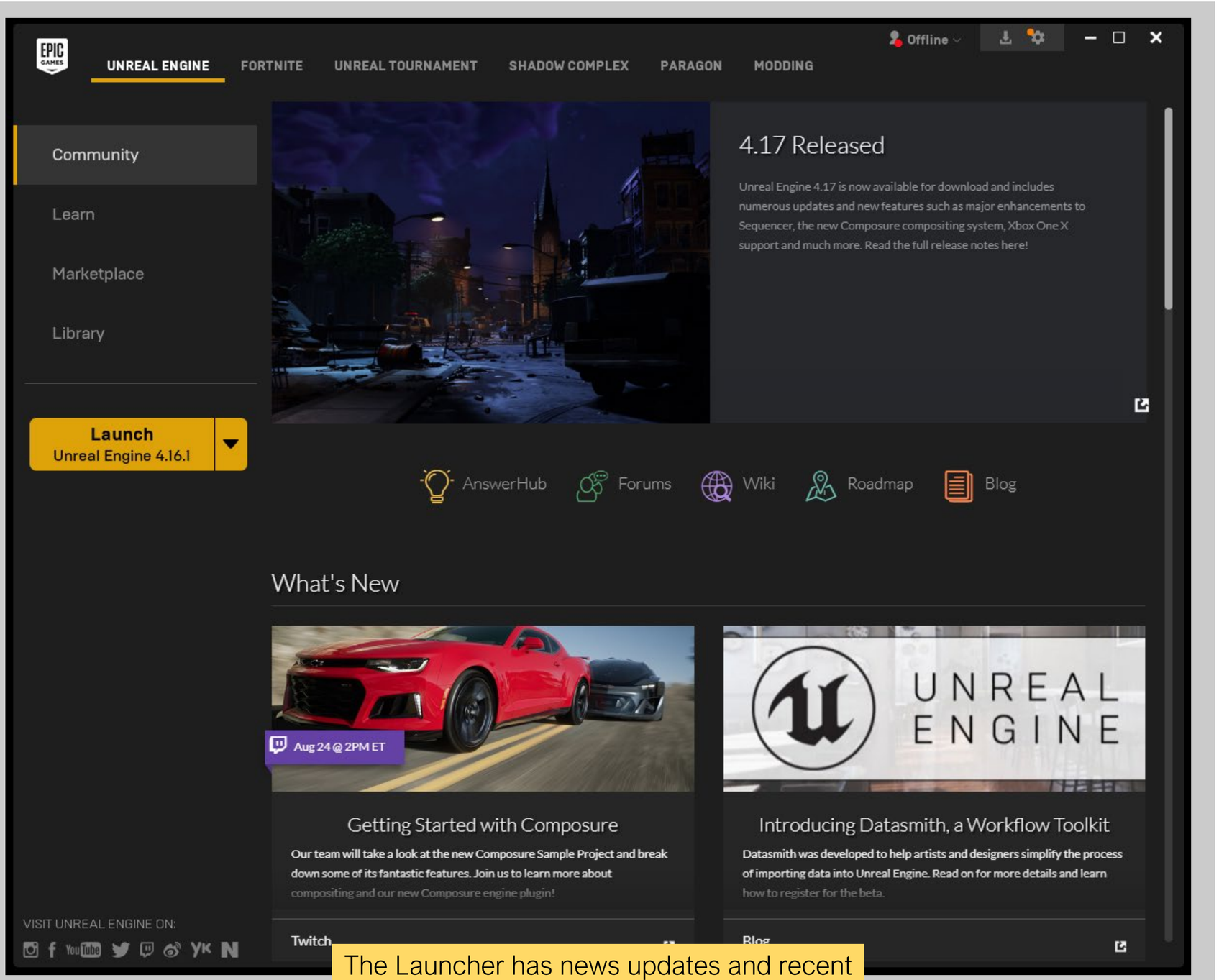
Post Process Effects enable artists and designers to tweak the overall look and feel of the scene by applying effects such as bloom (HDR blooming effect on bright objects), ambient occlusion, and tone mapping to the rendered scene before displaying it in the viewport.

PHYSICS ENGINE

Unreal Engine 4 uses the PhysX 3.3 physics engine to drive its physical simulation calculations and perform all collision calculations. PhysX provides the ability to perform accurate collision detection as well as simulate physical interactions between objects within the world. Having physics in your game will help improve the immersion value of every scene, as it helps players believe that they are interacting with the scene and that the scene is responding back in some way or another.

UNREAL EDITOR

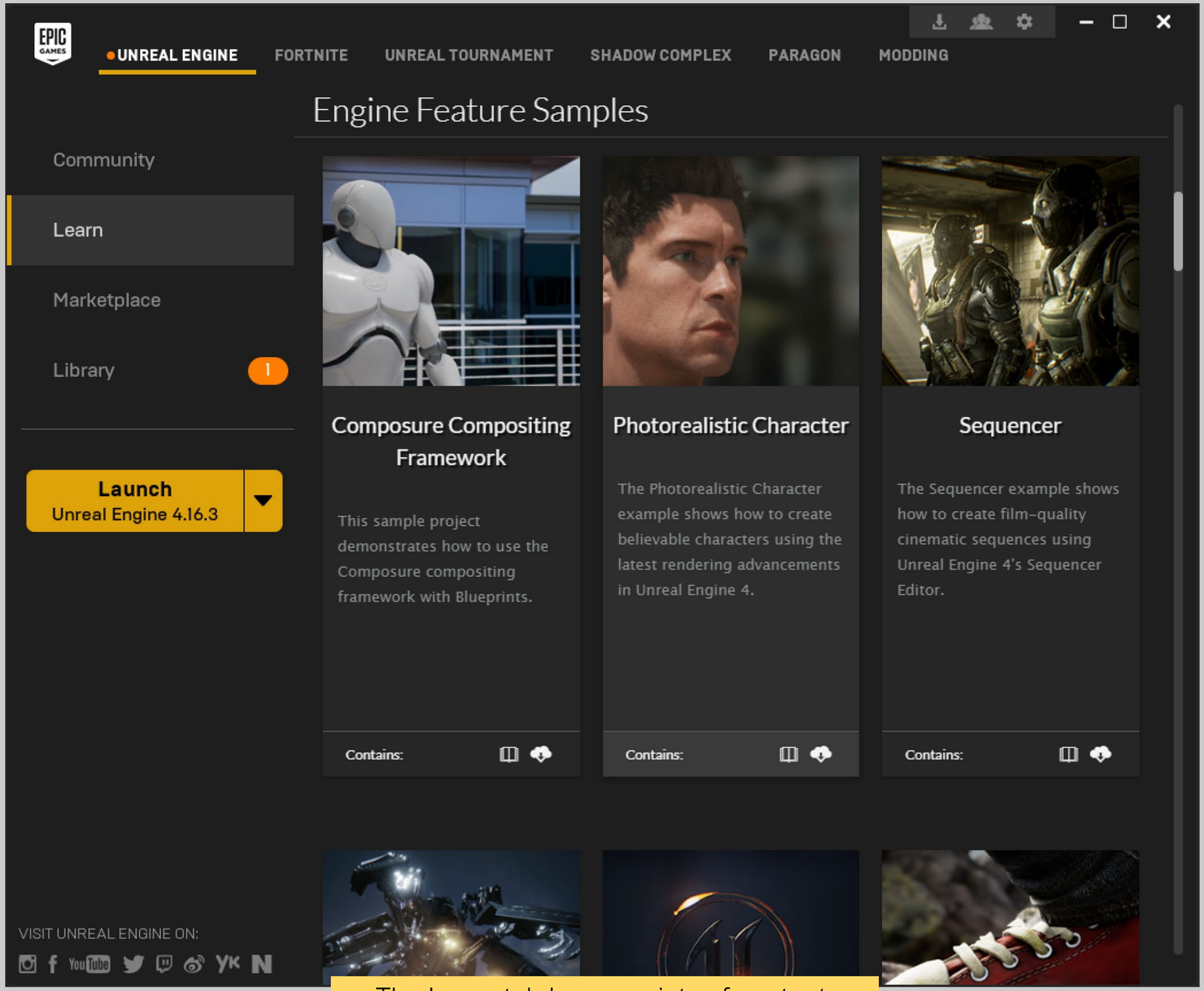
The Interface and Toolsets



The Launcher has news updates and recent releases on the first page.

THE LAUNCHER

The Epic Games Launcher is the portal to download Unreal Engine 4 builds, access Learning tools, and link through to community and learning resources for UE4.

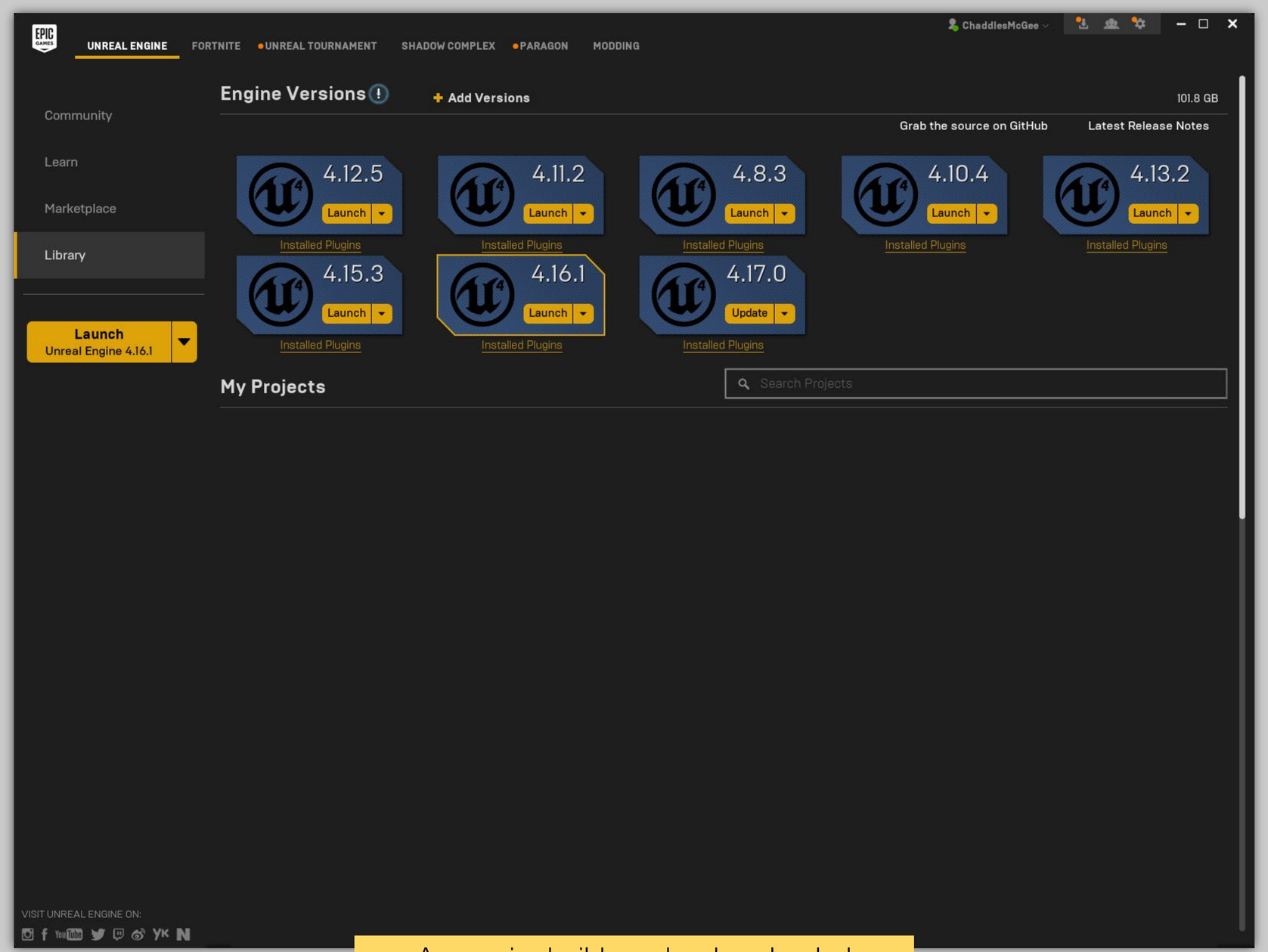


The Learn tab has a variety of content.

THE LEARN TAB

The Learn tab in the launcher contains many powerful examples of content implementation that can be drawn from and utilized for your own projects.

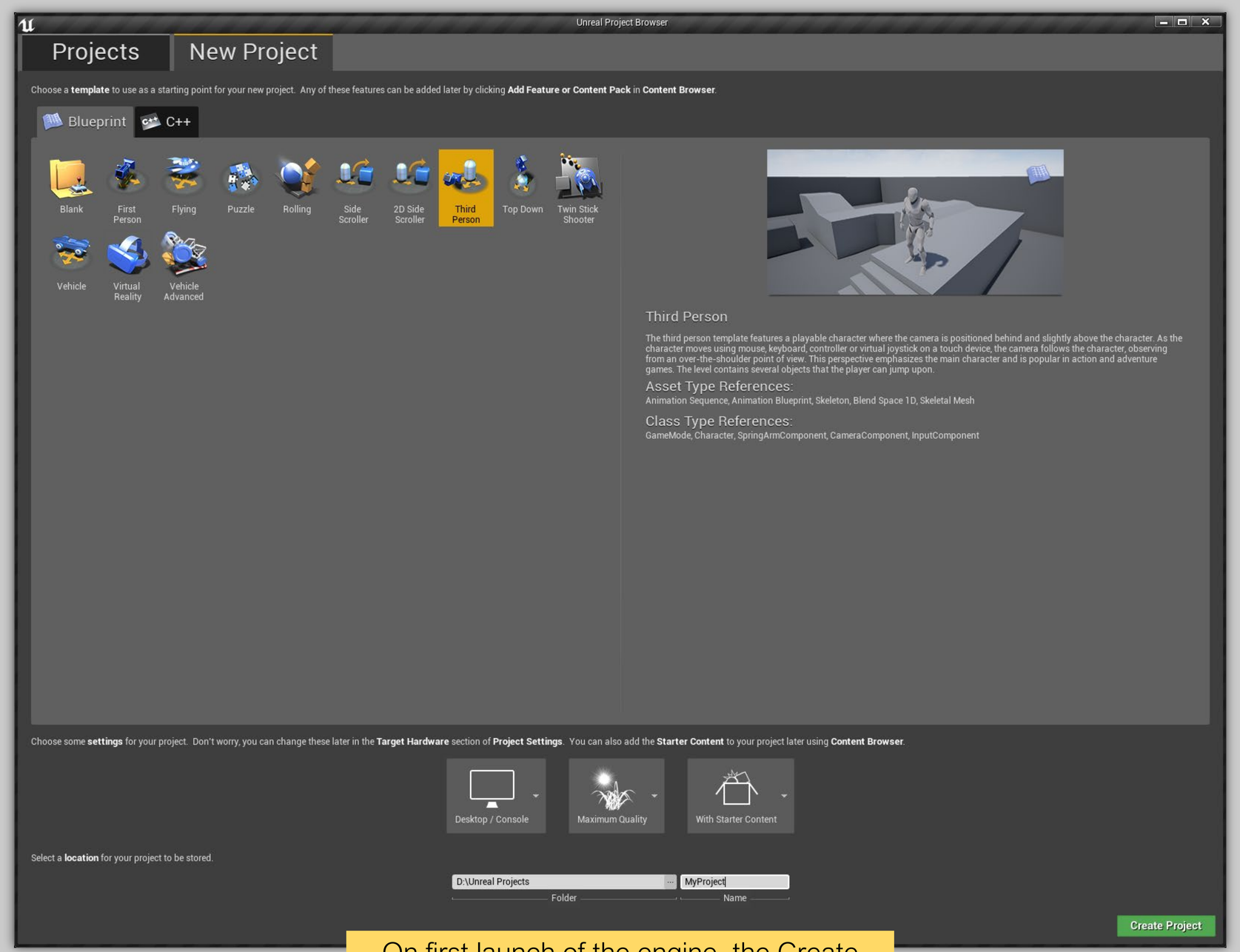
Of particular note is the Content Examples project, which provides samples of many engine features isolated into simple demonstrations for a better understanding of their use and implementation.



Any major build can be downloaded from the Library tab.

THE LIBRARY TAB

You can download the latest build of the engine from the Library tab of the Unreal Engine page in the launcher. You can also download previous versions of the engine, which is useful when working with projects that have not been updated.



On first launch of the engine, the Create Project screen will be visible.

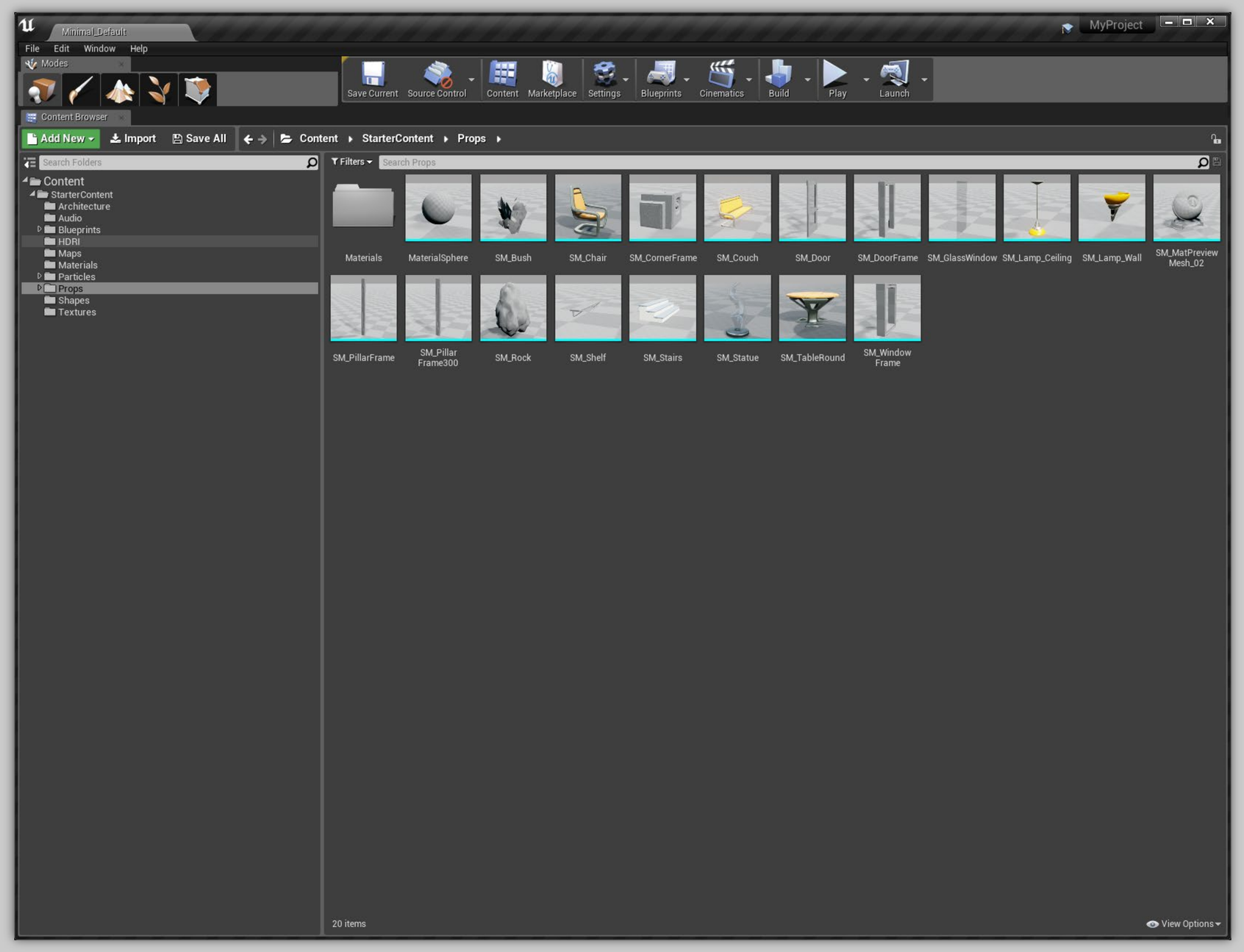
CREATING PROJECTS

Projects can be created from blank slates or derived from existing templates. The templates provide basic asset setups to build projects from. While these are powerful for prototyping, developers should strongly consider how appropriate a template is for long-term development of a project. The Project Browser allows you to create new projects or open existing ones.

The Starter Content features Tables, Chairs, and material examples.

LEVEL EDITOR

The **Level Editor** is the entry point for the engine. From within it, you can develop environments, add 3D meshes, and develop your projects. The view can be switched, or split, between perspective and orthographic cameras to help in design and mesh placement. Any selected actor can be moved in relation to the world (World Space) or in relation to itself (Local Space). The view can be toggled in the top right of the viewport.



The Content Browser

CONTENT BROWSER

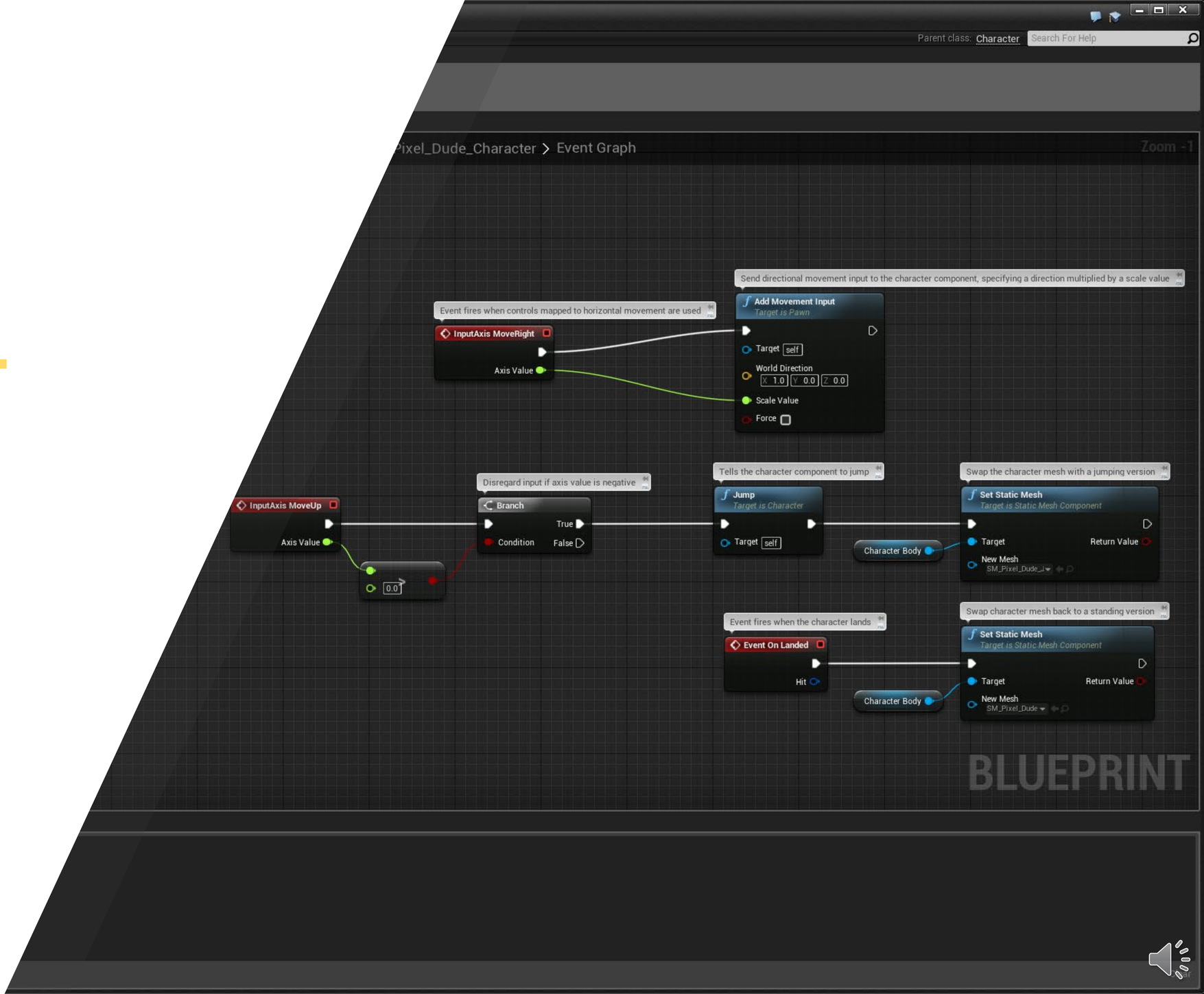
The **Content Browser** is the primary area of the Unreal Editor for creating, importing, organizing, viewing, and modifying content assets within Unreal Editor.

It also provides the ability to manage content folders and perform other useful operations on assets, such as renaming, moving, copying, and viewing references.

The Content Browser can search for and interact with all assets in the game and makes managing your assets and game content easy and intuitive.

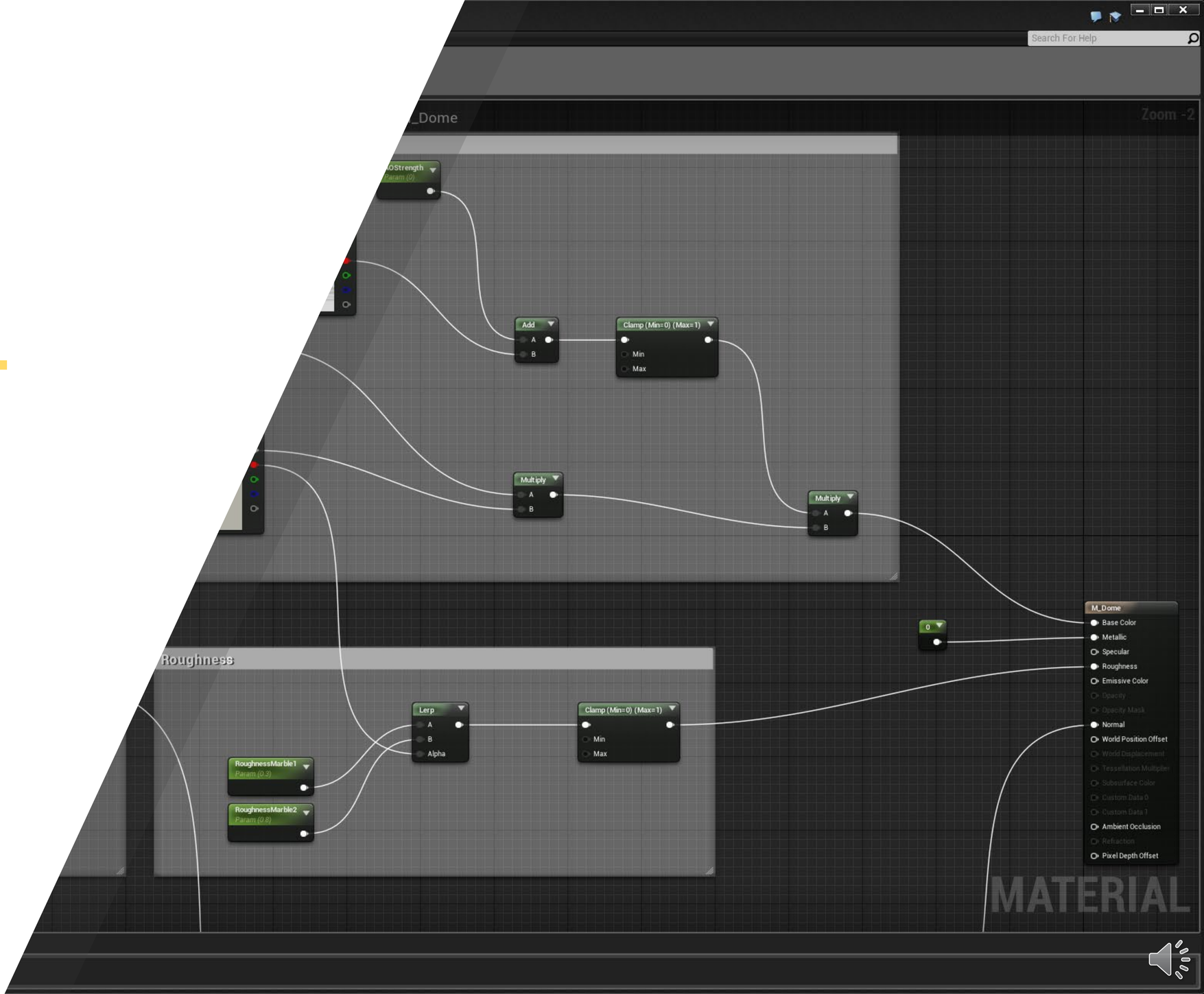
BLUEPRINT

Blueprint is an intuitive node-based programming language used to create new types of Actors and script level events, giving designers and gameplay programmers the tools to quickly create and iterate gameplay from within Unreal Editor without ever needing to write a line of C++ code.



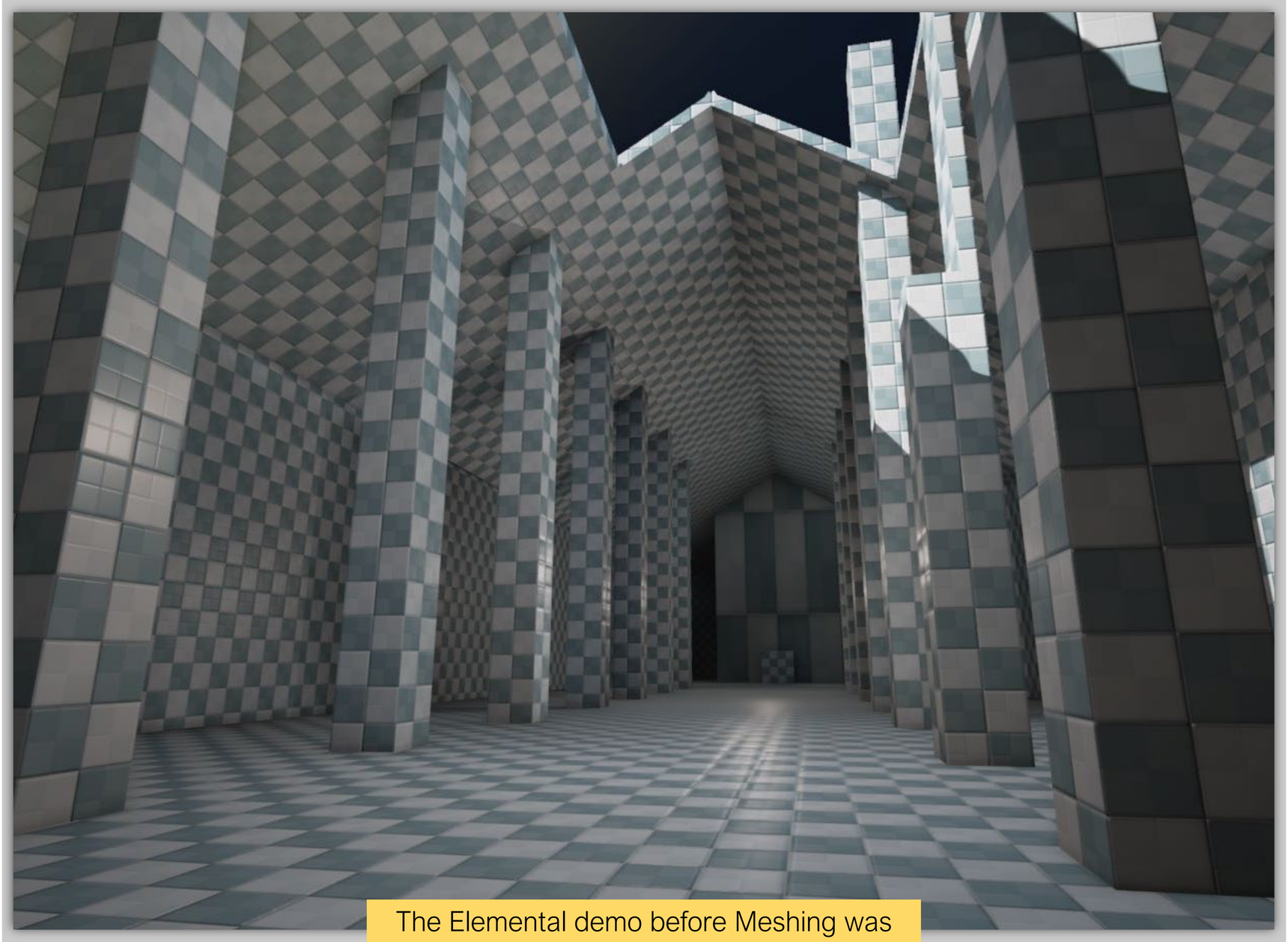
MATERIAL EDITOR

The **Material Editor** is a node-based graph interface that enables you to create Materials that can be applied to your geometry, such as Static and Skeletal Meshes, or that can be used with other systems such as Cascade to create interesting materials.



UNREAL EDITOR

3D Geometry



The Elemental demo before Meshing was just Geometry Brushes.

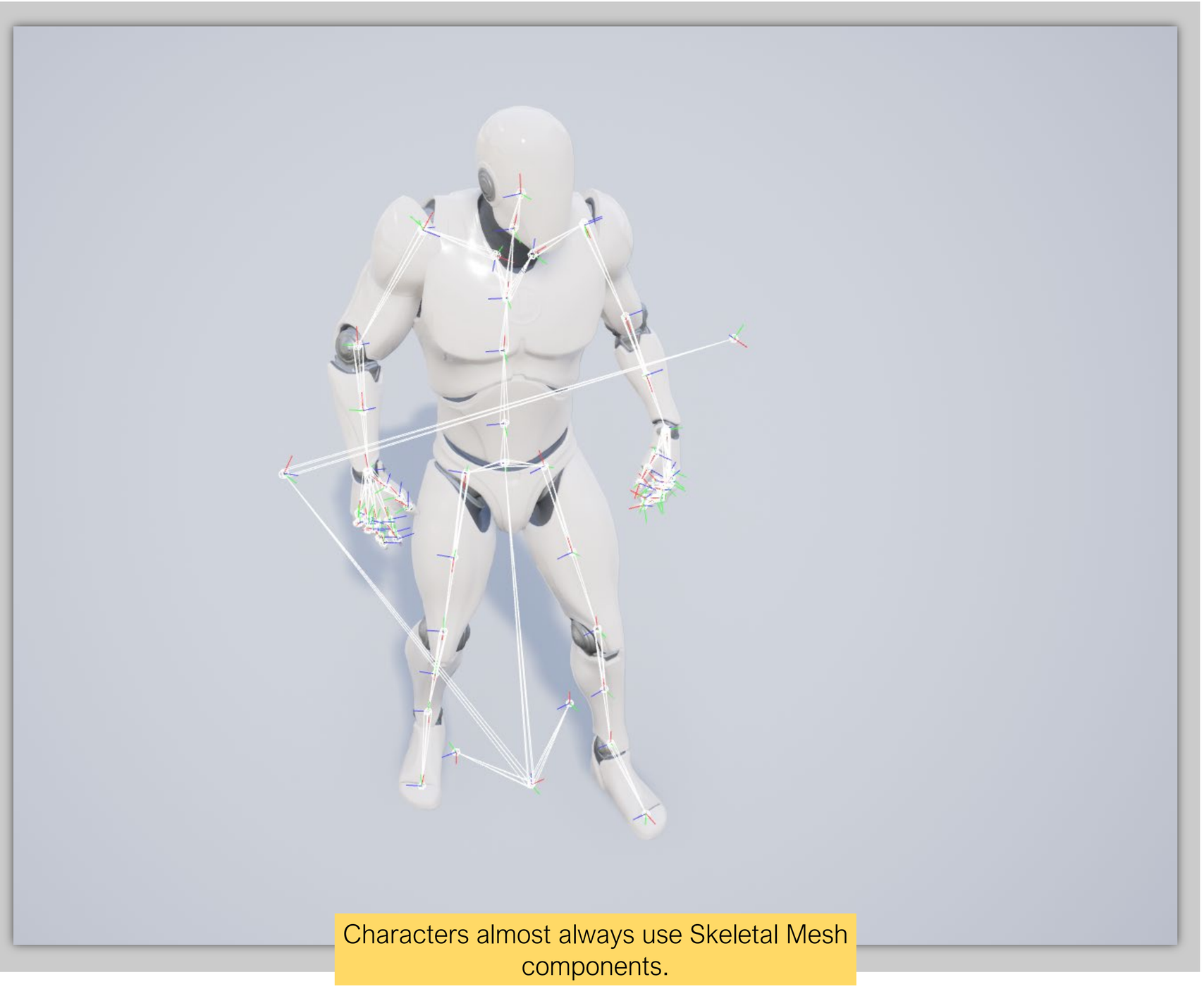
GEOMETRY BRUSHES

Geometry Brushes are the most basic tool for level construction in Unreal. Conceptually, it is best to think of a Geometry Brush as filling in and carving out volumes of space in your level. Geometry Brushes are typically used in the early stages of a project for rapid prototyping of levels and objects, as well as for level construction by those who do not have access to 3D modeling tools.

This chest can be found among the freely available *Infinity Blade: Fire Lands* assets.

STATIC MESHES

A Static Mesh is a piece of geometry that consists of a static set of polygons and is the basic class used to create world geometry for levels in Unreal Engine 4.



Characters almost always use Skeletal Mesh components.

SKELETAL MESHES

Skeletal Meshes are made up of two parts: a set of polygons composed to make up the surface of the Skeletal Mesh, and a hierarchical set of interconnected bones that can be used to animate the vertices of the polygons.

Content Examples: Landscapes

LANDSCAPES

To streamline the development pipeline and allow for dynamic optimizations, large outdoor environments are developed within Unreal Engine itself using “Landscapes”.

This system allows artists to dynamically paint complicated environmental geometry that automatically increases and decreases it’s level of detail according to the distance of the surface from the camera.

UNREAL EDITOR

Rendering and Graphics

These are available in the free Automotive Material pack on the Marketplace.

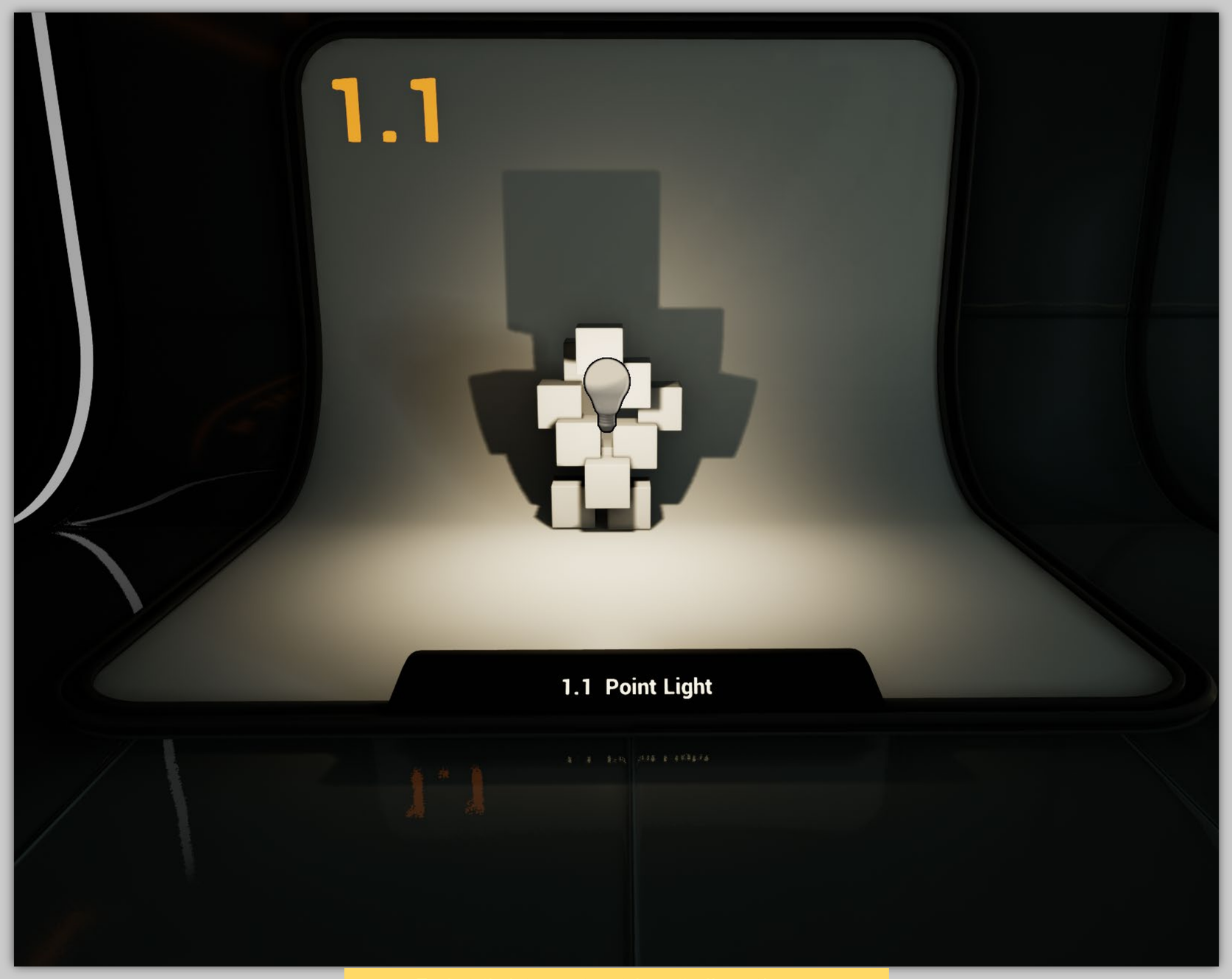
MATERIALS

A Material is an asset that can be applied to a mesh to control the visual look of the scene. At a high level, it is easiest to think of a Material as the “paint” that’s applied to an object. But even that can be a bit misleading, since a Material literally defines the type of surface from which your object appears to be made. You can define its color, roughness, opacity, and much more.

Decals can add easy visual variation to repetitive surfaces.

DECALS

Decals are Materials that are projected onto meshes in your level, including Static Meshes and Skeletal Meshes. Decals are typically used to provide slight variation to a world or add singular effects like scorch marks or liquid splatters.



Point lights output in all directions.

POINT LIGHTS

Point Lights work much like a real-world lightbulb, emitting light in all directions. Attenuation, falloff and measurement units can be changed individually along with whether or not they cast shadows.

Spot Lights output only in a given range.

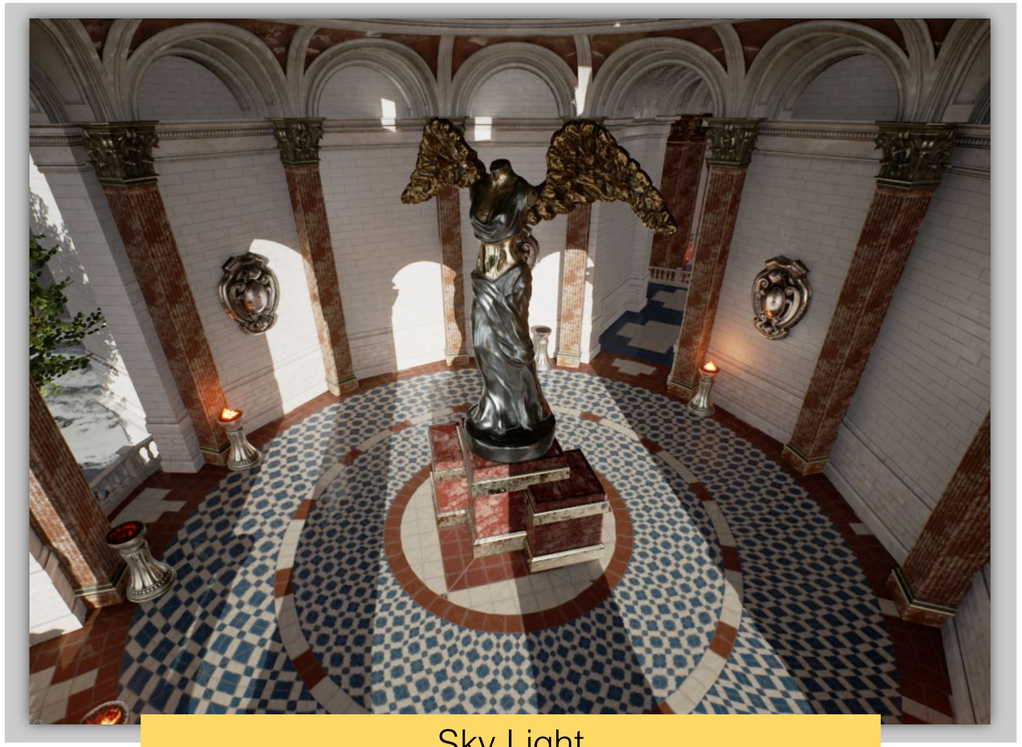
SPOT LIGHTS

A Spot Light emits light from a single point in a cone shape. Users are given two cones to shape the light—the Inner Cone Angle and Outer Cone Angle. Within the Inner Cone Angle, the light achieves full brightness.

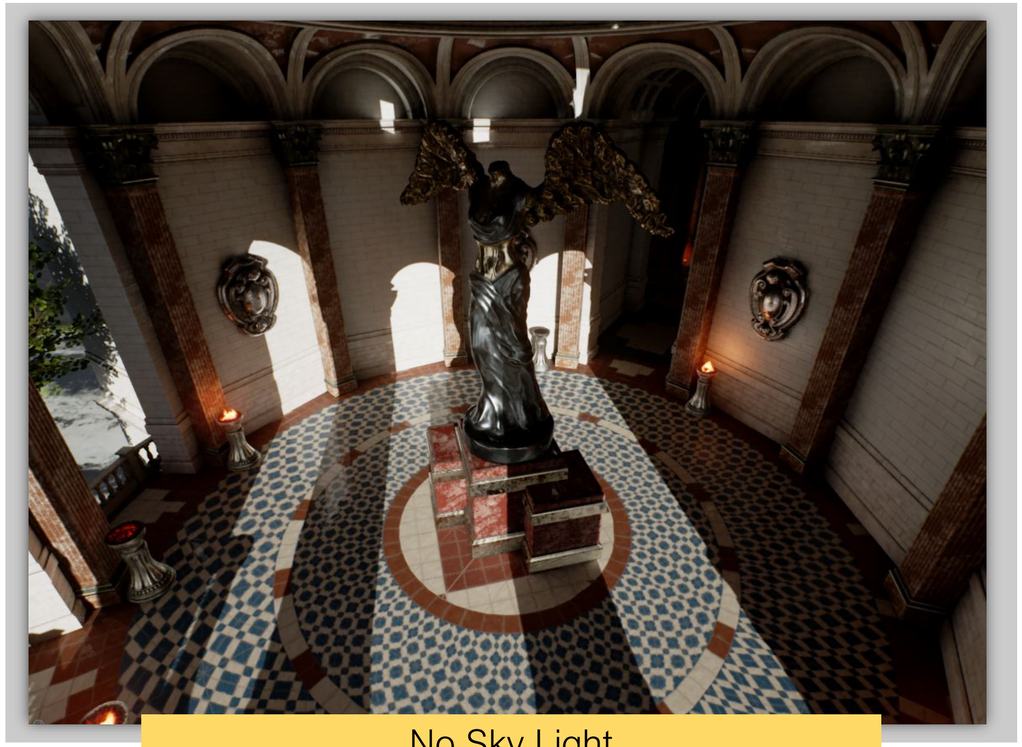
Directional Lights shine into a scene from a point calculated to be infinitely far away.

DIRECTIONAL LIGHTS

The Directional Light simulates light that is being emitted from a source that is infinitely far away. This means that all shadows cast by this light will be parallel, making it the ideal choice for simulating sunlight.



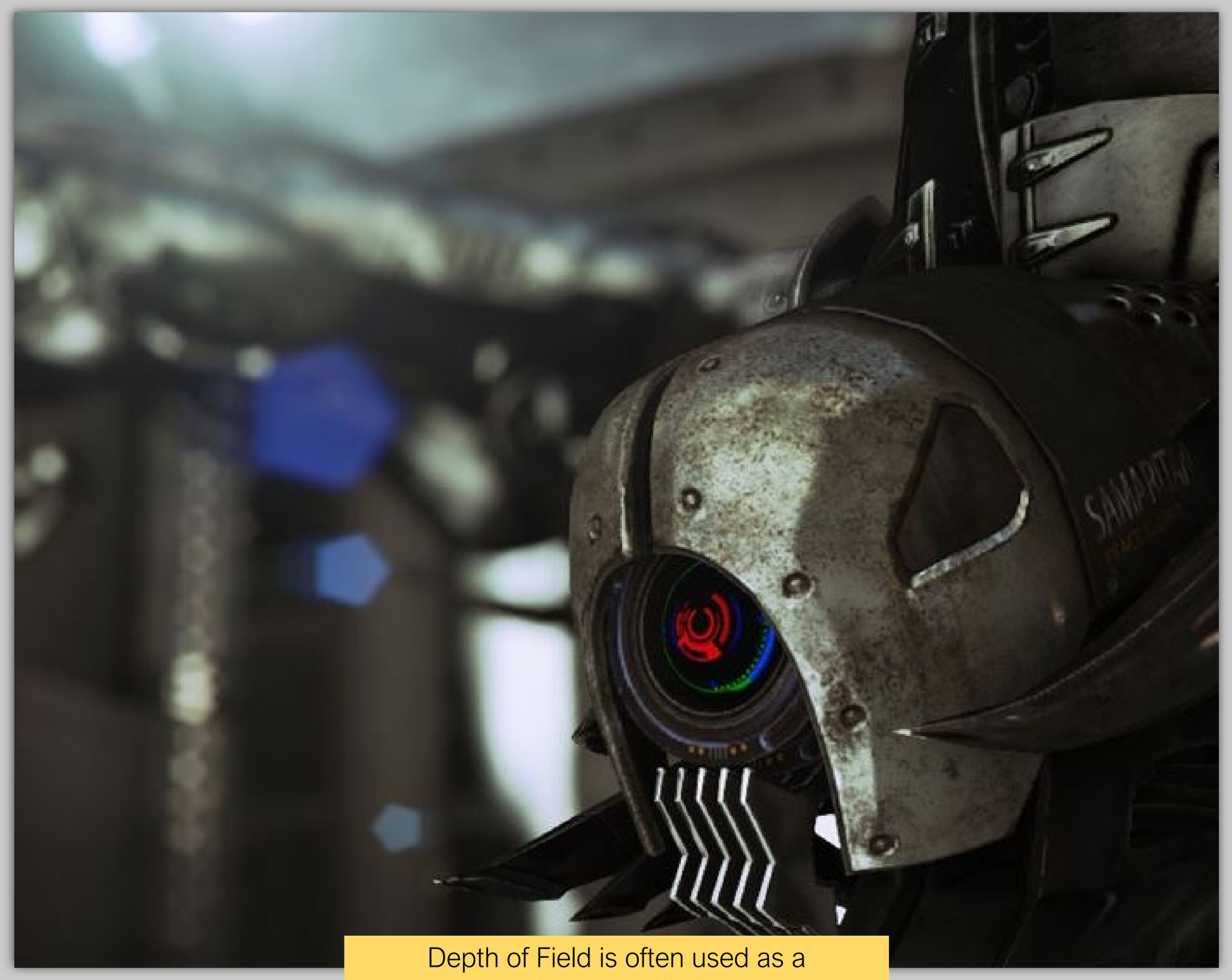
Sky Light



No Sky Light

SKY LIGHT

The Sky Light captures the distant parts of your level (everything farther than SkyDistanceThreshold) and applies that to the scene as a light. That means the sky's appearance and its lighting/reflections will match, even if your sky is coming from atmosphere, or layered clouds on top of a skybox, or distant mountains.



Depth of Field is often used as a cinematic effect.

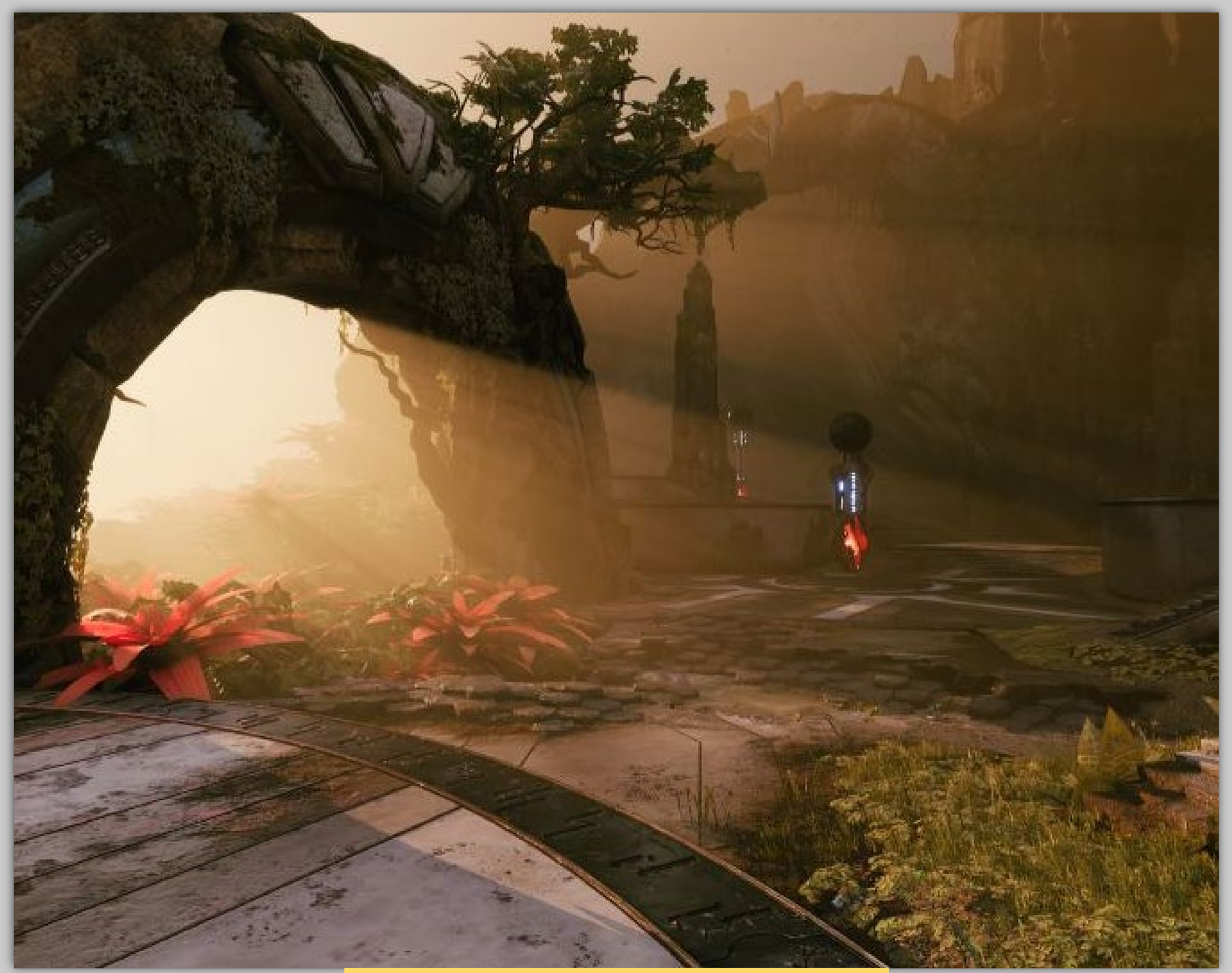
DEPTH OF FIELD

Depth of Field applies a blur to the scene based on distance in front of or behind a focal point.

Height fog can be used to create subtle atmosphere in a scene.

EXPONENTIAL HEIGHT FOG

Exponential Height Fog creates more density in low places of a map and less density in high places. The transition is smooth so you never get a hard cutoff as you increase altitude. Exponential Height Fog also provides two fog colors, one for the hemisphere facing the dominant directional light (or upward, along the *z* axis, if none exists), and another color for the opposite hemisphere.



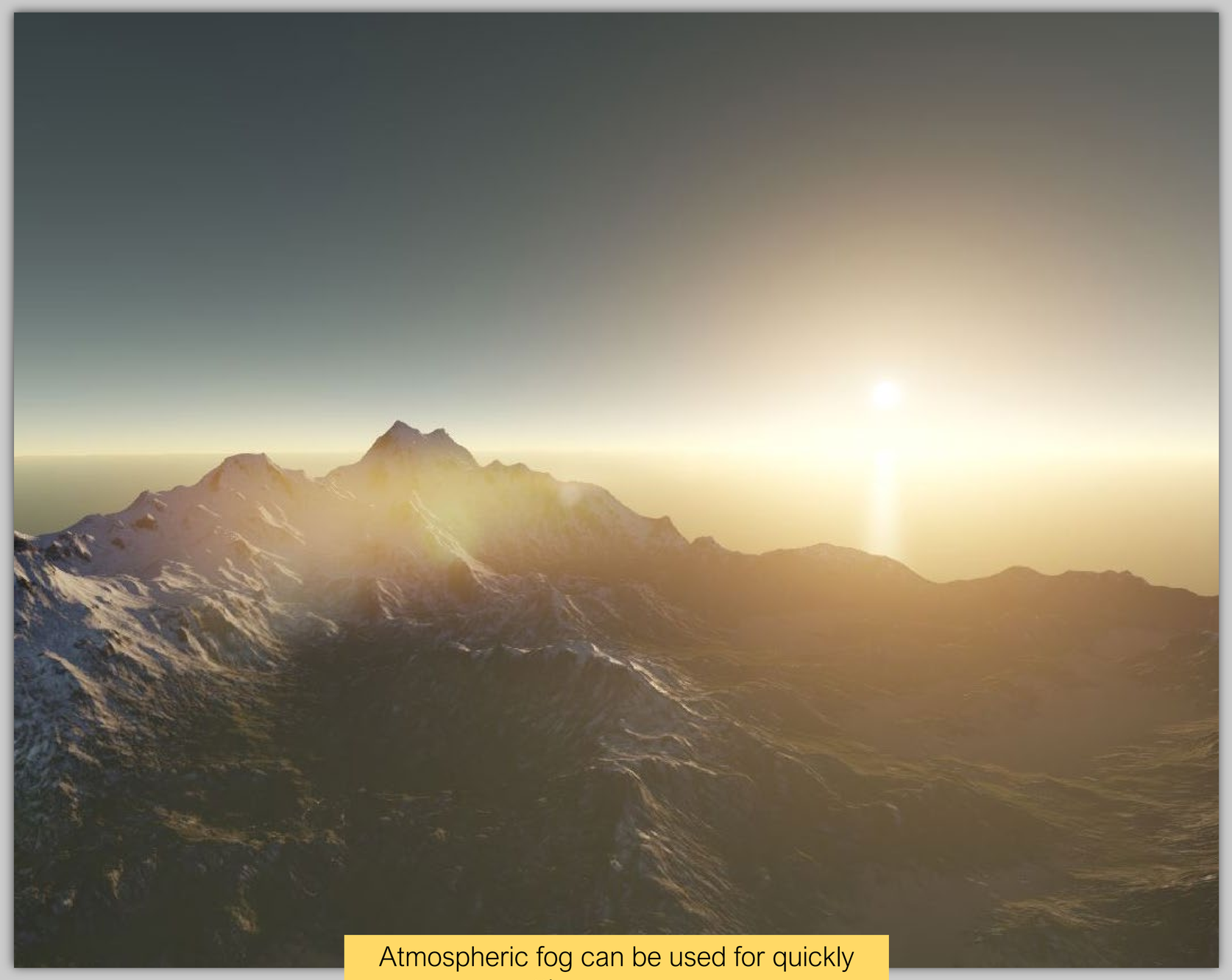
Volumetric Fog effects

VOLUMETRIC FOG

Volumetric Fog is created using a method that computes participating media density and lighting at every point in the camera frustum, which allows support for varying densities and any number of lights affecting the fog.

In this scene, Volumetric Fog is coming from the directional light source off screen through the arch and surrounding area to create shadowed fog.

This effect can be achieved through either a volumetric particle system or an option within Exponential Height Fog.



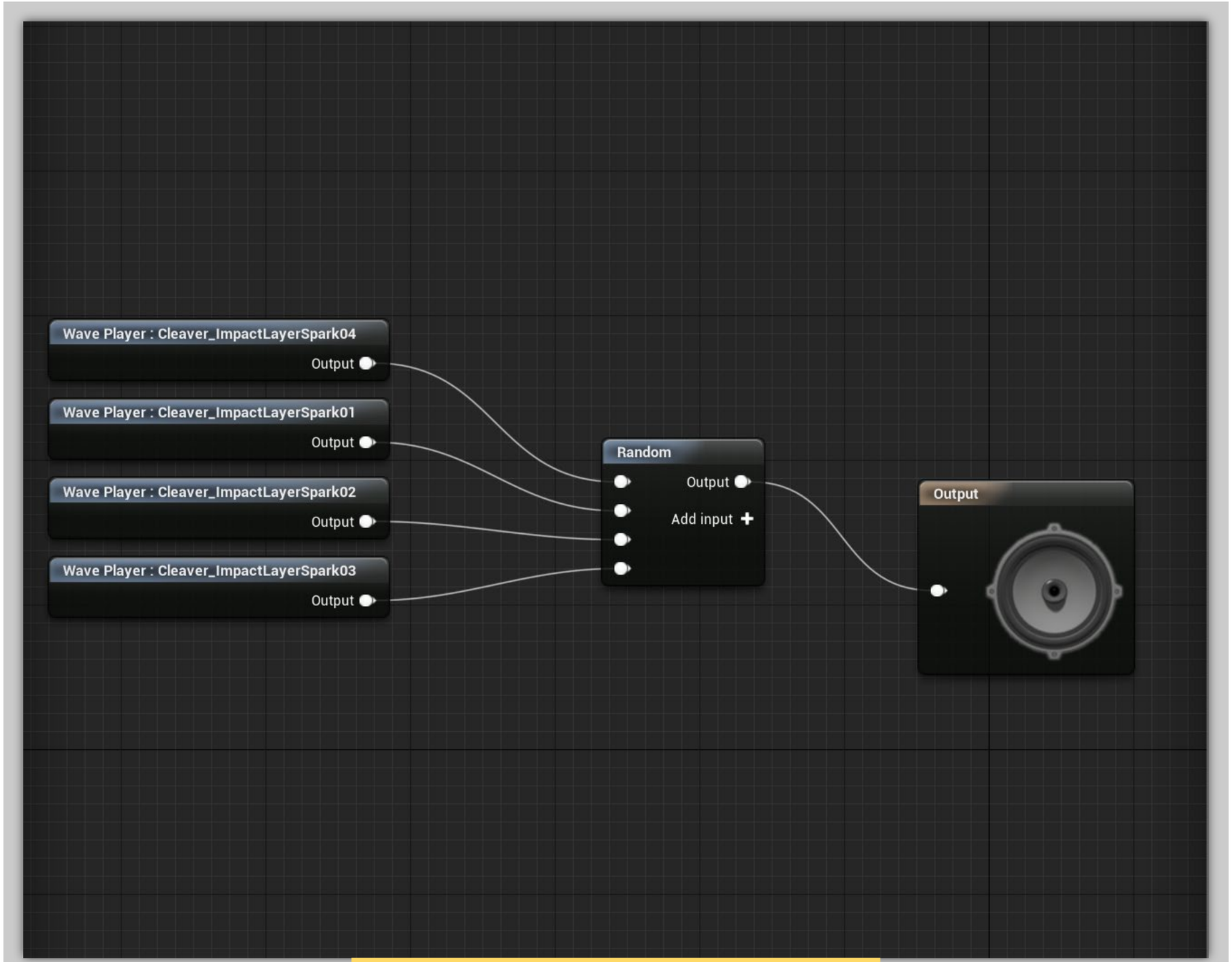
Atmospheric fog can be used for quickly setting up a scene.

ATMOSPHERIC FOG

Atmospheric Fog gives an approximation of light scattering through a planetary atmosphere. It can give your outdoor levels a much more realistic look. Details of atmospherics can be matched to a directional light by enabling it as the Sun in the actor details panel.

UNREAL EDITOR

Audio and Sound

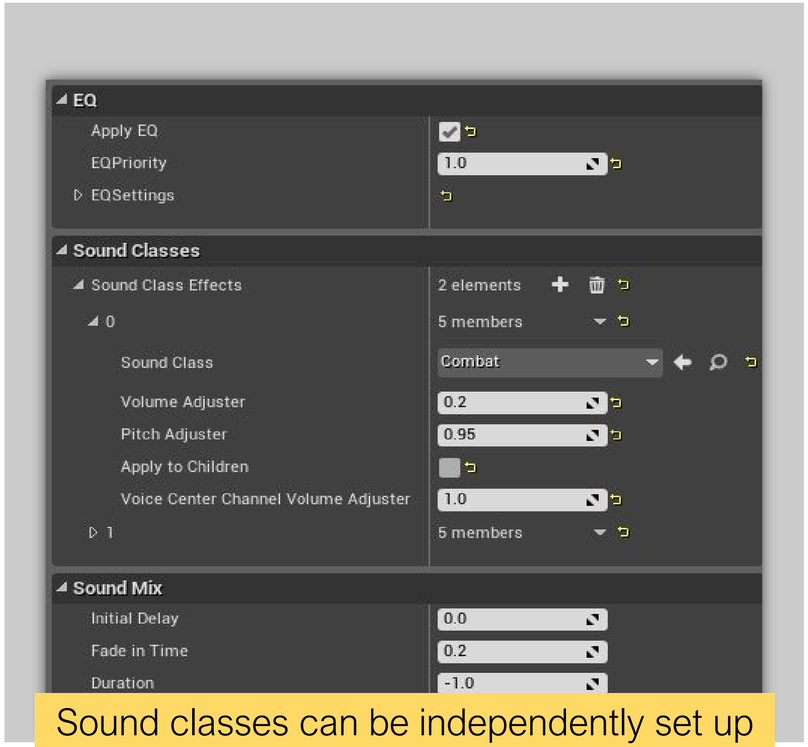


Sound Cue Editor

SOUND CUES

The audio system in Unreal Engine 4 is made up of several components, each working together to produce the audio experience for players. When you import an audio file into the engine and drop it in a level, you will have several options such as the basic Volume or Pitch levels to adjust, as well as other fine-tuning settings such as Sound Attenuation.

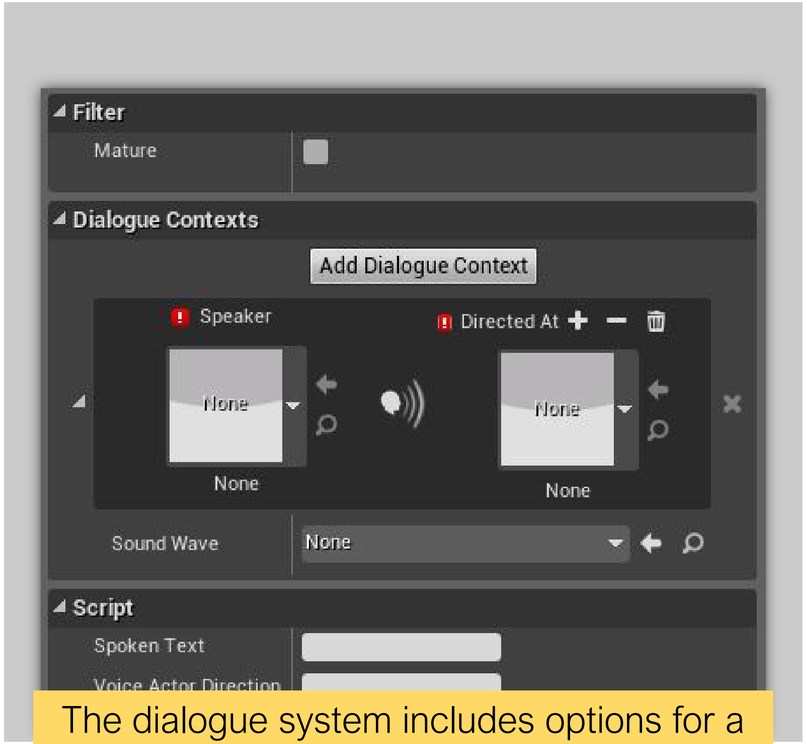
Unreal Engine 4 allows for building composite sounds in the form of Sound Cues and the Sound Cue Editor, which enable you to combine sounds as well as apply modifiers called Sound Nodes to alter the final output.



Sound classes can be independently set up for different types of effects.

Sound Mixes

Sound Mixes allow you to set the **EQ Settings** (Equalizer Settings) and modify **Volume** and **Pitch** properties of Sound Classes. These mixes can be dynamically modified in Blueprint at runtime to create the effect of slow motion or a post-explosion recovery.



The dialogue system includes options for a wide variety of speakers.

Dialogue

The Dialogue Voice and Dialogue Wave assets are used for generating in-game dialogue events, crafting subtitles, and supplementing localization efforts. When editing a newly created Dialogue Voice asset, you can define the **Gender**, **Plurality**, or other qualities of a voice actor.

C O N C L U S I O N

Questions?