Developer
/ Documentation
/ Unreal Engine
/ Unreal Engine 5.4 Documentation
/ Programming and Scripting
/ Blueprints Visual Scripting
/ Anatomy of a Blueprint

Anatomy of a Blueprint

The User Guide is the go-to source to learn the different parts of Blueprints and nodes that are available to use within Blueprint graphs.

Blueprint Classes

A **Blueprint Class**, often shortened as **Blueprint**, is an asset that allows content creators to easily add functionality on top of existing gameplay classes. Blueprints are created inside of Unreal Editor visually, instead of by typing code, and saved as assets in a content package. These essentially define a new class or type of Actor which can then be placed into maps as instances that behave like any other type of Actor.

Components

Blueprints do not always need to contain scripted behavior. For example, a lightpost in your level may not be interactive, and would just need a mesh to represent the post and a light. Using **Components** to build up reusable Blueprints would accelerate your level design process. Of course, you could then work with those Components in a graph to allow your players to interact with the lights or have a time-of-day system adjust them accordingly.

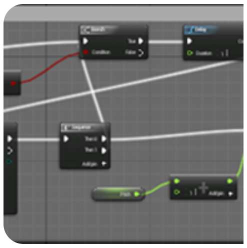
Graphs

Graphs contain the design-time and game-time behavior of your Blueprints. The **Construction Script** runs following the Components list when an instance of a Blueprint Class is created, enabling you to dynamically adjust the look and feel of the new Object or Actor.

The **EventGraph** of a Blueprint contains a node graph that uses events and function calls to perform actions in response to gameplay events associated with the Blueprint. This is used to add functionality that is common to all instances of a Blueprint. This is where interactivity and dynamic responses are setup. For example, a light Blueprint could respond to a damage event by turning off its `LightComponent` and changing the material used by its mesh. This would automatically provide this behavior to all instances of the light Blueprint.

EventGraph

Uses events and function calls to perform actions in response to Blueprint events.



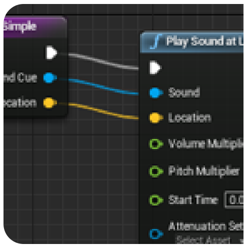
Graphs

Node graph that uses events and function calls to perform actions in response to events associated with the Blueprint.

Organizing and Reusing Script

As you create more script in your Blueprints, you may find there are certain sections of script you frequently reuse. **Functions** and **Macros** both enable you to reuse sections of script, although each method has different strengths and use cases. For more on the key differences between Functions and Macros, see the Blueprint Best Practices guide.

You can also collapse sections of your Graph into nested Graphs for organization.



Function Calls

Nodes that execute both code-defined and user-defined functions of a target Actor or Object.

Using Macro Libraries

Macros inside a Macro Library are used to increase the health and scale of an Actor.