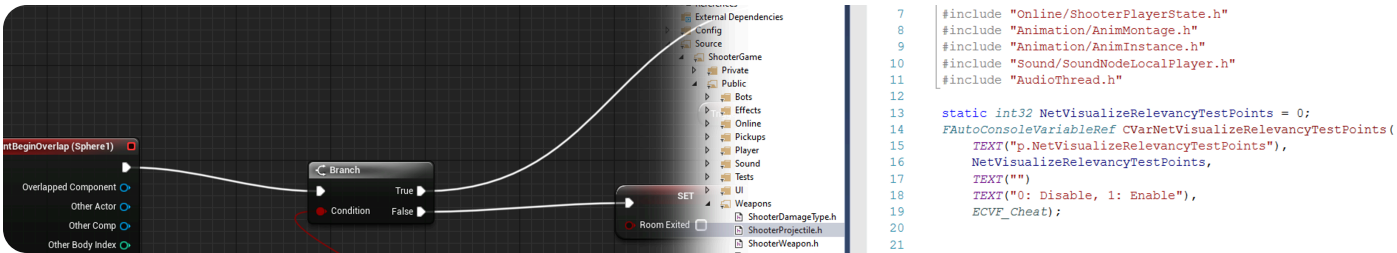


# Programming and Scripting

How to use the programming and scripting languages and tools for controlling Unreal engine programmatically at runtime.



Programming and Scripting contains information on how to use the following **Unreal Engine (UE)** programming features:

- [Programming with C++](#) in UE is similar to standard C++, using **Classes**, **Functions**, and **Variables**. These are defined using standard C++ syntax. Each class defines a template for a new **Object** or **Actor** that can be further encapsulated with the [Unreal Engine Reflection System](#).
- [Blueprints Visual Scripting](#) is a visual scripting programming tool that creates classes, functions, and variables in the Unreal Editor. These classes can then be executed by connecting various nodes together. C++ classes are used as a base for **Blueprint** classes, programmers can set up fundamental gameplay classes that are then sub-classed and iterated on by Designers.
- [Python Scripting](#) is a high-level scripting language used for extending UE's production pipeline with editor utilities, such as asset management scripts. Python scripts also have specific applications in toolsets throughout UE, such as the [Movie Render Queue](#). Python is intended only for editor tooling and cannot be used at runtime.
- [Online Subsystems and Services](#) provides a common way to access the functionality of various online services such as **Playstation Network**, **Xbox Live**, **Epic Online Services**, and **Steam**. Developers can use these tools when working on a game that ships on multiple platforms or supports multiple online services that are configured individually for each supported service.
- [Unreal Architecture](#) contains UE's Modules. These encapsulate specific editor tools, runtime features, libraries, or other functionality in standalone units of code. All projects and plugins have their own primary module by default, however, you can define additional modules to organize your code.
- [Development Setup](#) contains guides on how to set up your development environment, how to download and build the UE source code, and instructions on the use of supplemental programming tools.
- [Slate UI Programming](#) is a custom and platform-agnostic user interface framework designed to make efficient user interfaces for tools and applications such as the Unreal Editor or in-game user interfaces. It combines a declarative syntax with the ability to

easily design, layout, and style components that allows for easily creating and iterating on UIs.

# Section Directory



## Programming with C++

Information for programmers developing with Unreal Engine.



## Blueprints Visual Scripting

Overview of using the Blueprint visual scripting system for gameplay.



## Class Creation Basics

Examples showing how to create classes with Blueprints alone, C++ alone, and a combination of C++ and Blueprints.



## Online Subsystems and Services

Learn how to use online subsystems and services in Unreal Engine, including Epic Online Services.



## Unreal Architecture

Programming in the Unreal Architecture



## Development Setup

How to set up your development environment for programming with C++ in Unreal Engine.