

# ④ An Introduction to Game Development

Mario Santos

Brief Overview of Unreal Engine

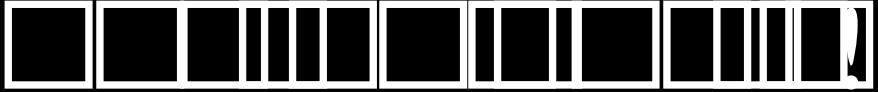
Basics of C++ in Unreal

Getting into the Game Development Industry

Demonstration: Building a Simple Game in Unreal using C++

Q&A Session





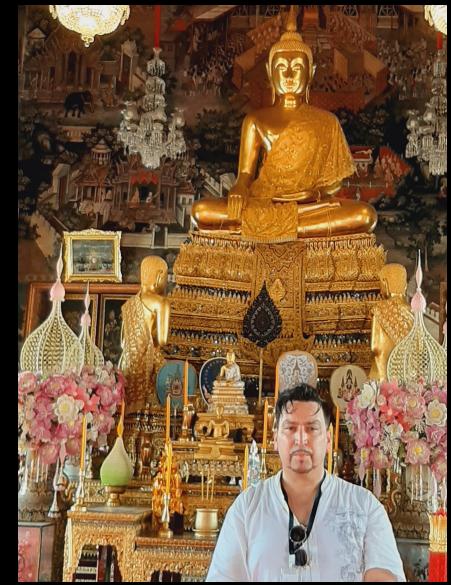
CHASE AND STATUS



ARCTIC  
MONKEYS

RED HOT CHILI PEPPERS

MUSE



Microsoft

NUANCE





# Introduction to Unreal Engine

Unreal Engine: Driving Next-Gen Gaming and Beyond.

## > HELLO WORLD!

Implementation after the  
PRIMERICA ENGINE



# Unreal Engine 5.3 Feature Highlights





FINAL FANTASY  
VII  
REMAKE

# Introduction to C++ for Unreal Engine

Bridging the Gap between Blueprints and C++

```
#pragma once

#include "CoreMinimal.h"
#include "UObject/Interface.h"
#include "Interactable.generated.h"

UINTERFACE(MinimalAPI)
class UInteractable : public UIInterface
{
    GENERATED_BODY()
};

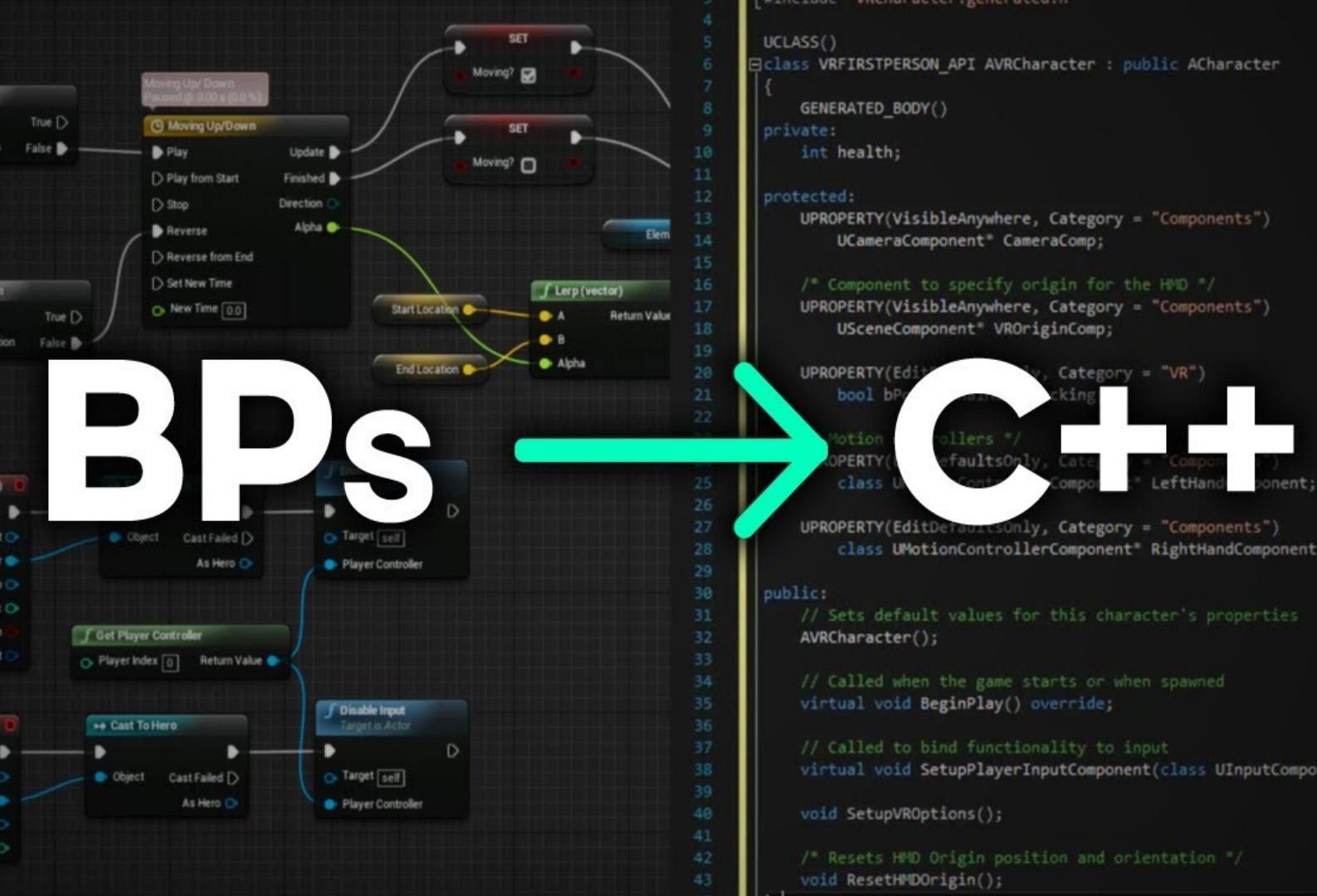
/**
 */
class HOWTO_API IInteractable
{
    GENERATED_BODY()

public:

    virtual FName GetName() = 0;

    virtual FName GetQuestID() = 0;

    UFUNCTION(BlueprintCallable, BlueprintNativeEvent, Category = "Interact")
    void OnInteract();
};
```



# Blueprints vs C++ in Unreal

- **Blueprints: The Intuitive Visual Language.**
- **C++ in Unreal: The Power Behind the Throne.**
- Blueprints have made game design more accessible, but C++ remains the bedrock, especially when performance is key.

# The Unique Nature of C++ in Unreal

## UClass Definition

### Contains:

- Properties: Variables exposed to the editor.
- Functions: Methods for in-game calls.
- Events: Handlers for specific game occurrences.

## UClass Compilation

### Process:

Transforms the UClass definition for game usage.

## Unreal Engine Interaction

### Components:

#### Blueprints:

- Graphical scripting interface.
- Accesses and uses "Properties", "Functions", and "Events" from the UClass.

#### C++ Gameplay Code:

- Implements game logic, AI, physics, etc.
- Directly utilizes "Properties", "Functions", and "Events" of the UClass.

## Game Output

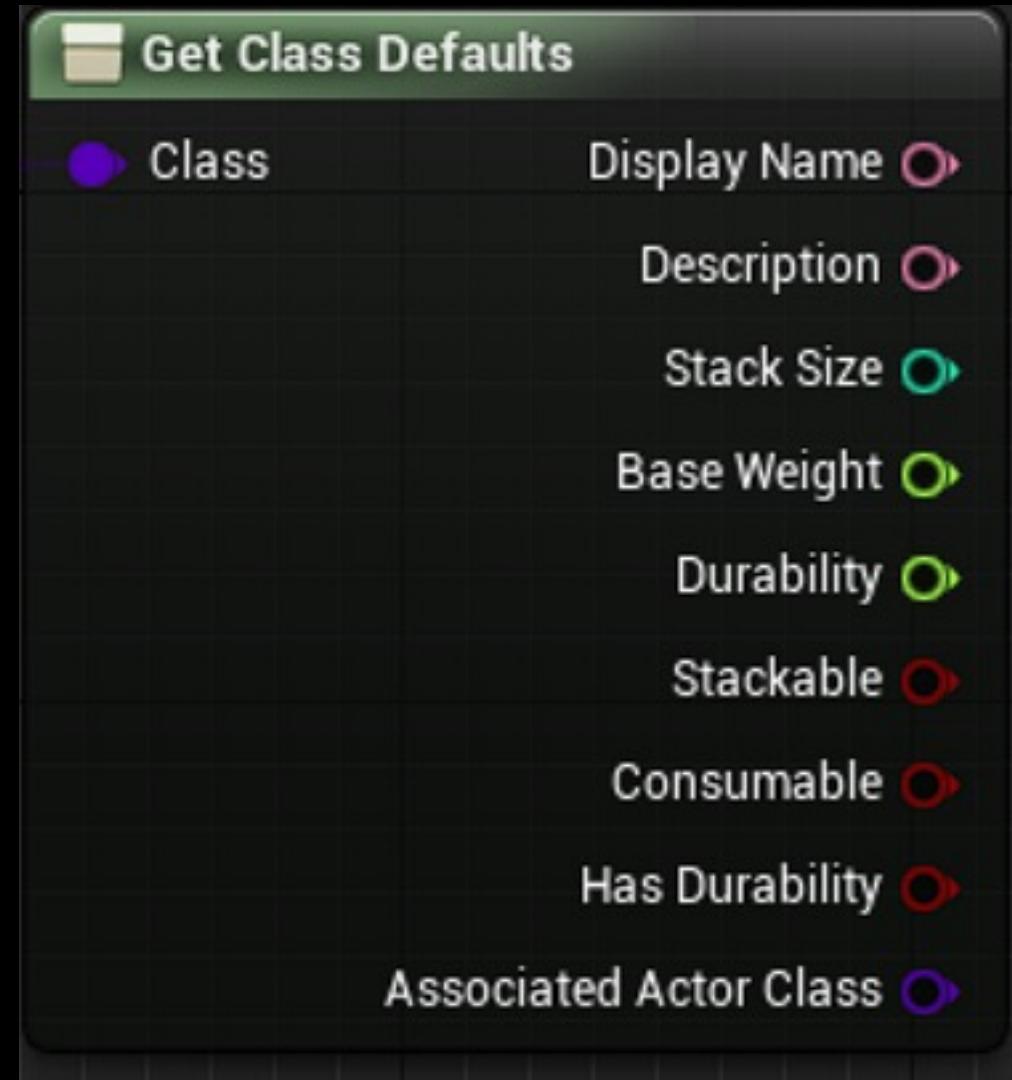
### Result:

Gameplay elements driven by UClass interaction, including characters, AI, and landscapes.

**UClasses:** allow the engine to seamlessly integrate with the expansive capabilities of C++

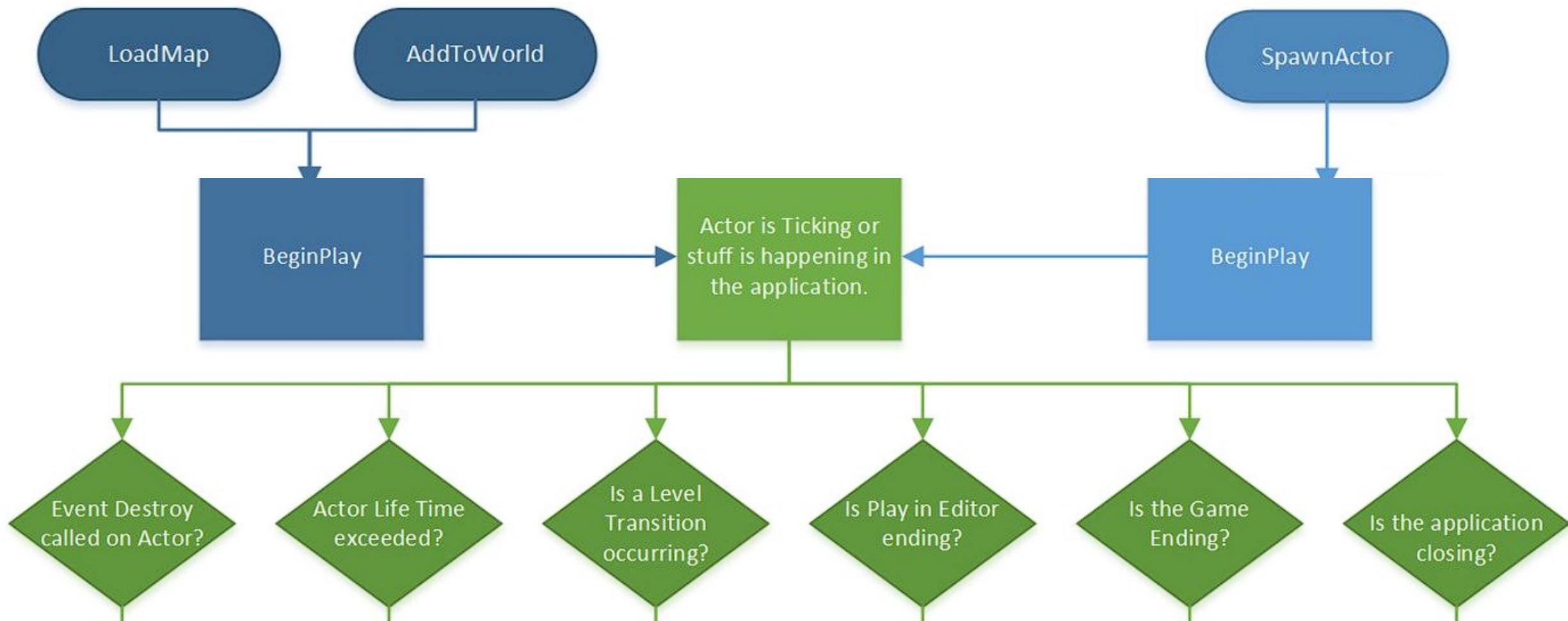
# Class Default Object (CDO) in Unreal

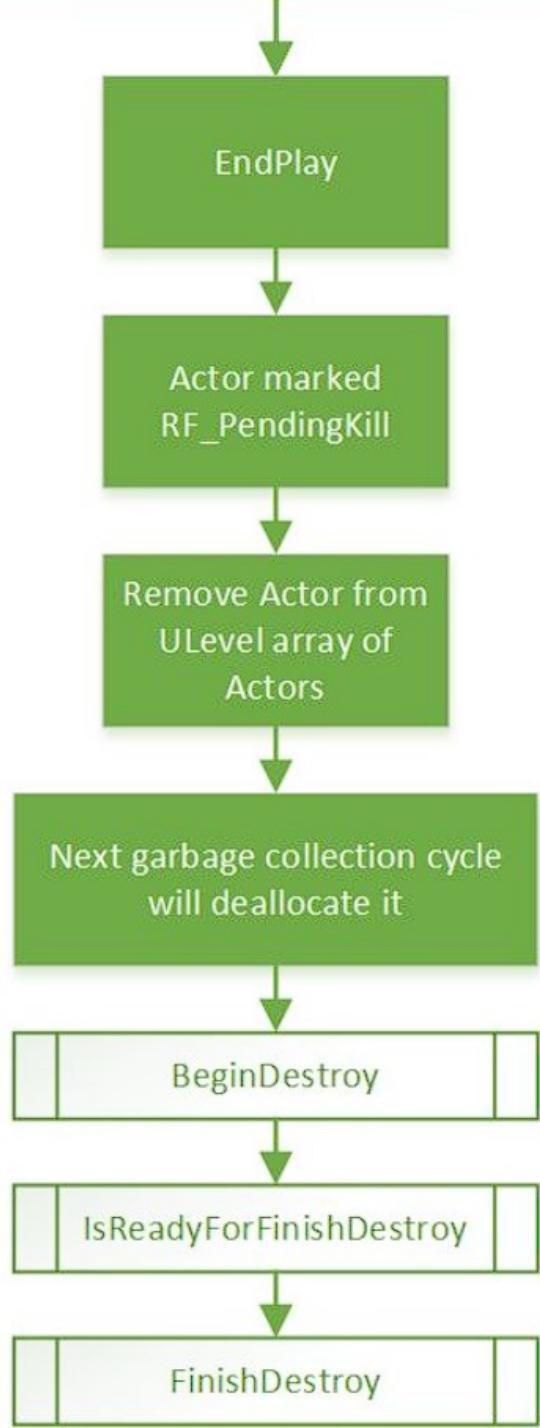
- Unreal's default template for each class.
- Ensures consistent starting values for new instances.
- Unreal uses CDOs while standard C++ relies on constructors for initial values.



# Memory Management in Unreal

- Garbage Collection: Unreal's Cleaning Crew.





## Blueprint Function Description

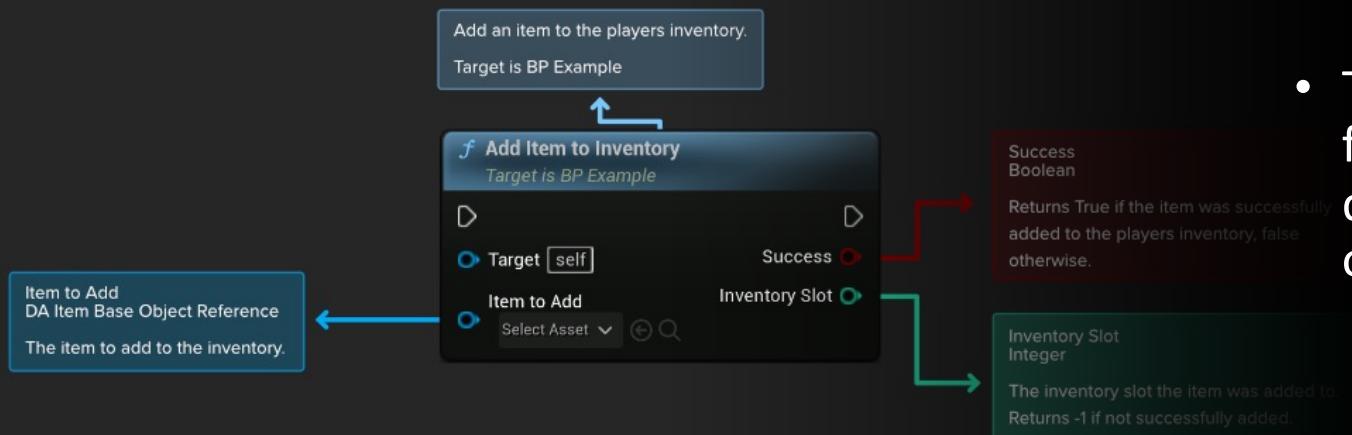
### Description

Add an item to the players inventory.  
@param ItemToAdd The item to add to the inventory.  
@param bSuccess Returns True if the item was successfully added to the players inventory.  
@param InventorySlot The inventory slot the item was added to. Returns -1 if not successful.

### C++ Function Comment

```
/**  
 * Add an item to the players inventory.  
 * @param ItemToAdd The item to add to the inventory.  
 * @param bSuccess Returns True if the item was successfully added to the players inventory.  
 * @param InventorySlot The inventory slot the item was added to. Returns -1 if not successful.  
 */  
UFUNCTION(BlueprintCallable, Category = "Inventory|Modifier")  
void AddItemToInventory(UDataAsset* ItemToAdd, bool& bSuccess, int32& InventorySlot);
```

### Blueprint Result



# Directing Gameplay with Macros

- UFUNCTION() & UPROPERTY(): The Directors of Gameplay.
- These macros are essential tools, allowing developers to expose functions and variables to Blueprints and the Unreal Editor.

- This is how complex interactions, from character movements to dynamic environments, are orchestrated in Unreal games.

# Unreal Framework Classes: Building Blocks of the Engine

## **UObject:**

- Base for all Unreal objects, enabling garbage collection and reflection.

## **AActor:**

- Base for interactive world objects, enhanced by components.

## **APlayerController:**

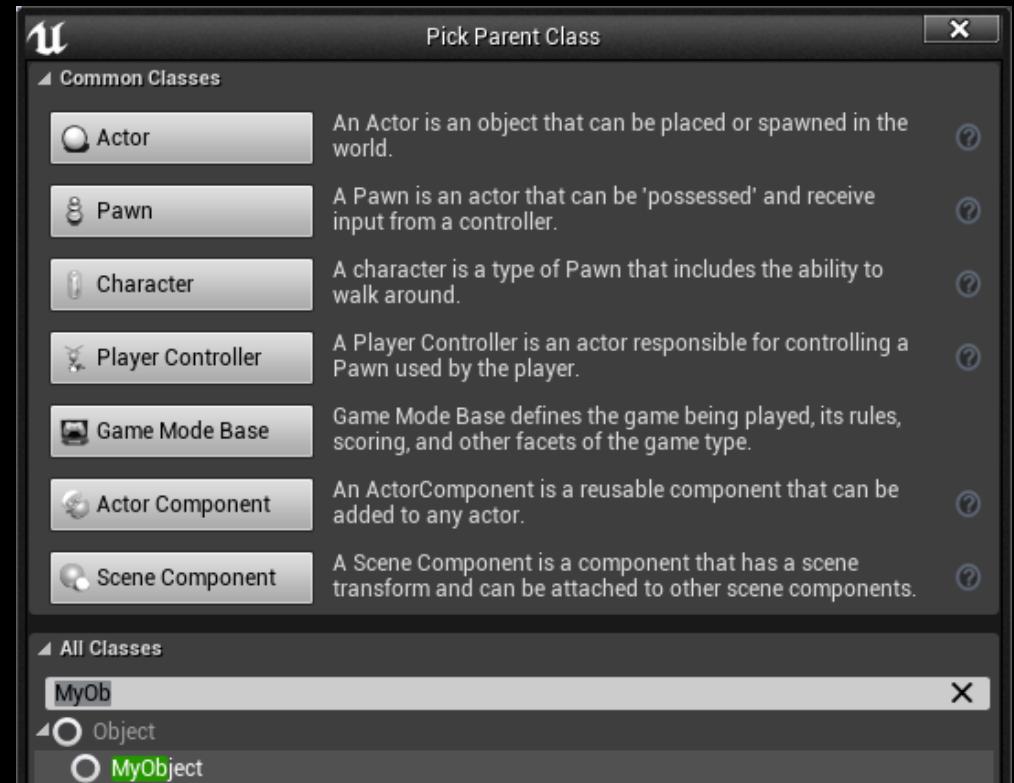
- Represents players, managing their input and game interactions.

## **UComponent:**

- Base for Actor components, allowing functionality extensions.

## **APawn:**

- AActor subclass for characters, forming playable entities or NPCs.



## C++ in Unreal: Key Takeaways:

1. C++ offers deep control in Unreal.
2. Unreal's CDO is unique and foundational.
3. Blueprints & C++ can coexist & complement each other.
4. Continuous learning is key in game development.



**What's your primary motivation for pursuing a career in computing and game development?**

- ⓘ Start presenting to display the poll results on this slide.



# Entering Game Development

## Essential Skills:

- Programming (C++, Blueprints, C#, Lua, etc.)
- Game Design
- Graphics & Animation.

PR AND SUPPORT

File 3 point Line

# Entering Game Development



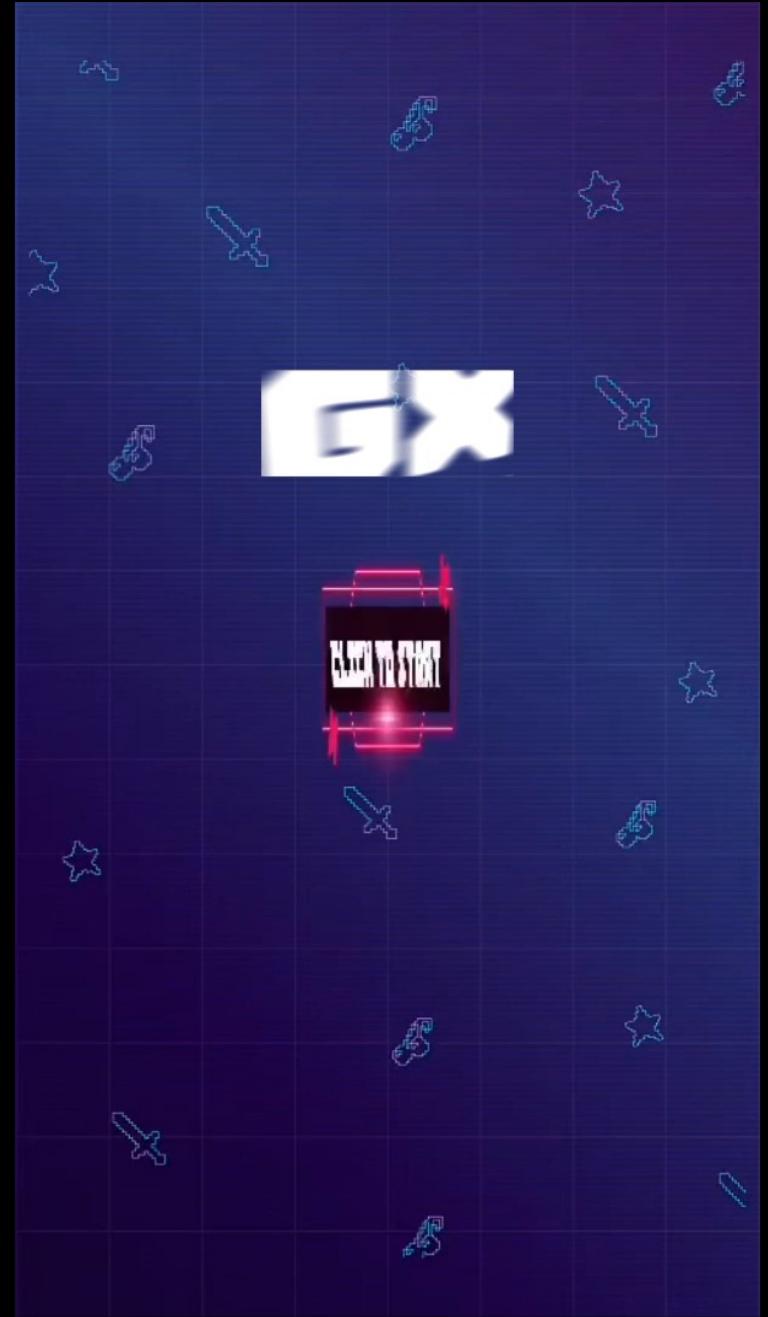
WHILE TECHNICAL SKILLS GET YOU STARTED, SOFT SKILLS LIKE COMMUNICATION, TEAMWORK, AND ADAPTABILITY ARE INVALUABLE.



BEING PROACTIVE IN SEEKING FEEDBACK AND ITERATING ON YOUR PROJECTS IS CRUCIAL.  
SELF-MOTIVATION IS KEY!

# Entering Game Development

- Networking:  
Attend game jams,  
conventions,  
and join online forums &  
communities.

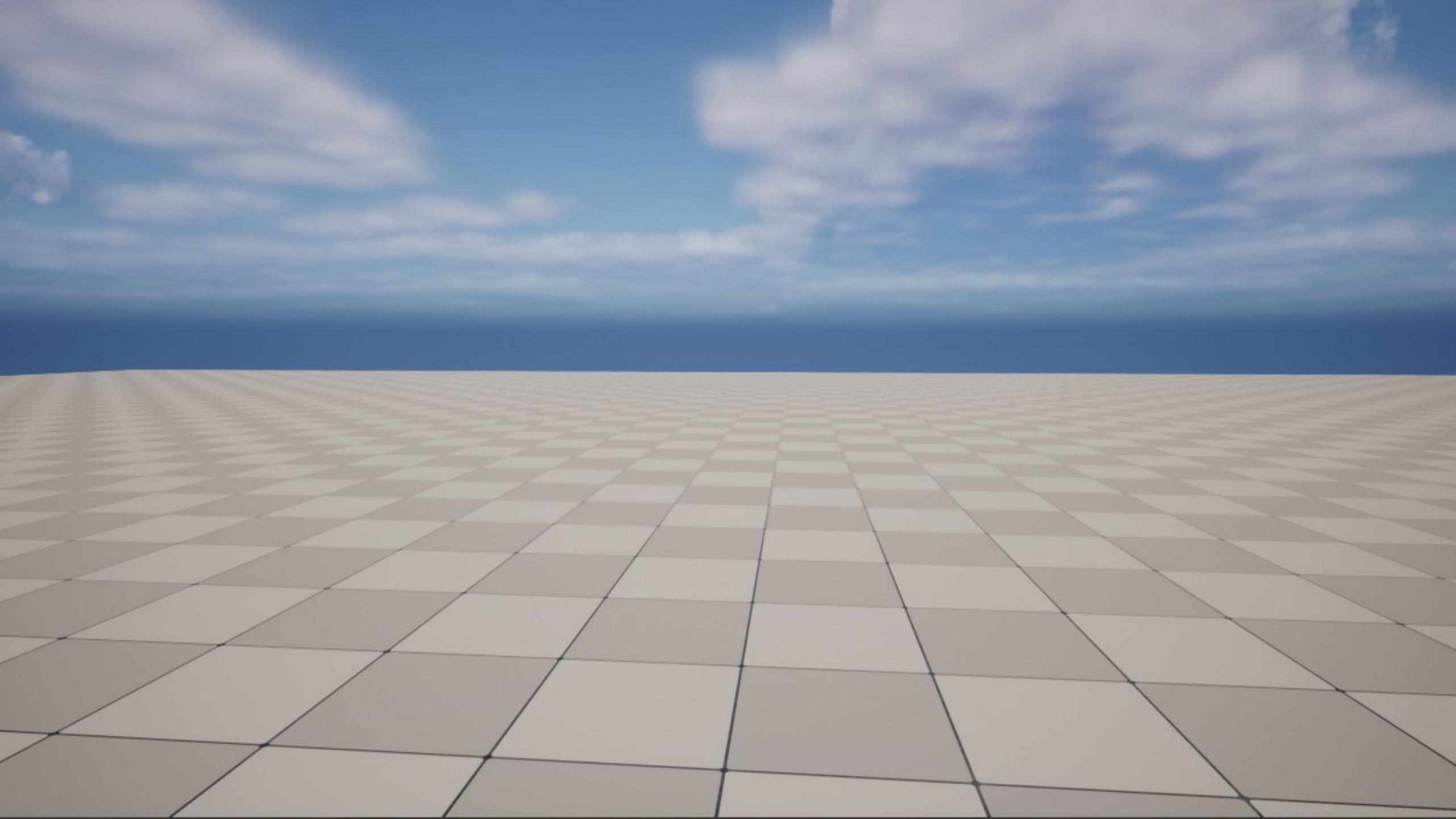


# Demonstration Overview

Next>>>

We'll delve into a step-by-step demonstration of creating a basic game scenario in Unreal using C++ and Blueprints.

As we walk through the demonstration, pay attention to how the tools and code come together to bring the game to life.







Microsoft

Visual Studio

Developer Tools ▾

Downloads

Buy ▾

Subscriptions ▾

Free Visual Studio



# Thank you for downloading Visual Studio

Your download will start shortly. If your download does not begin, [click here to retry](#)

## Visual Studio Installer

Getting the Visual Studio Installer ready.

Downloaded

Installing

Download ▾

Community 2022

Professional 2022

Enterprise 2022

# Visual Studio Installer

Installing — Visual Studio Community 2022 — 17.7.5

[Workloads](#)   [Individual components](#)   [Language packs](#)[Desktop & mobile \(5\)](#)

## Installation locations

**.NET Multi-platform App UI development**   
Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.

**Desktop development with C++**   
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

**Mobile development with C++**   
Build cross-platform applications for iOS, Android or Windows using C++.

## Gaming (2)

**Game development with Unity**   
Create 2D and 3D games with Unity, a powerful cross-platform development environment.

**Game development with C++**   
Use the full power of C++ to build professional games powered by DirectX, Unreal, or Cocos2d.

**Location**  
C:\Program Files\Microsoft Visual Studio\2022\Community [Change...](#)[Remove out-of-support components](#)

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 26.34 GB

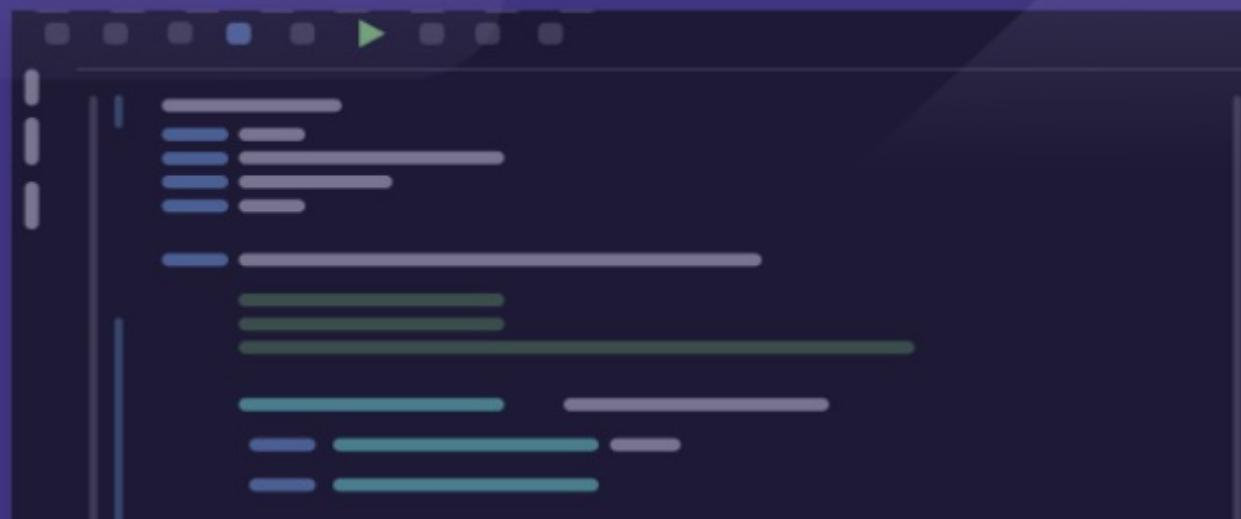
[Install while downloading](#) [Install](#)

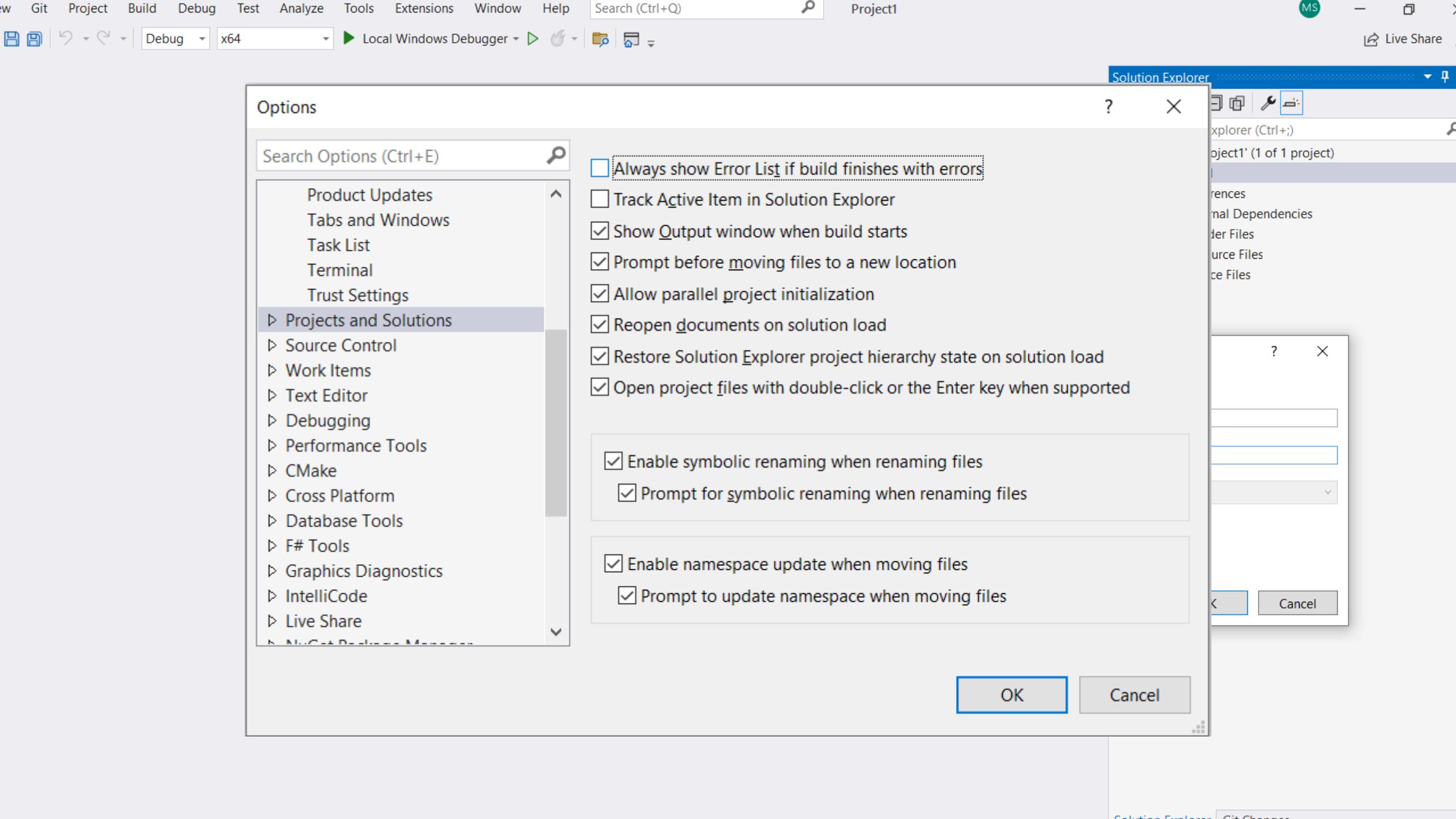
## Installation details

- [Windows Universal Runtime](#)
- [C++ 2022 Redistributable Update](#)
- Optional**
  - [MSVC v143 - VS 2022 C++ x64/x86 build t...](#)
  - [C++ profiling tools](#)
  - [C++ Build Insights](#)
  - [C++ AddressSanitizer](#)
  - [vcpkg package manager](#)
  - [Windows 11 SDK \(10.0.22621.0\)](#)
  - [IntelliCode](#)
  - [IDE support for Unreal Engine](#)
  - [HLSL Tools](#)
  - [Windows 11 SDK \(10.0.22000.0\)](#)
  - [Windows 10 SDK \(10.0.20348.0\)](#)
  - [Windows 10 SDK \(10.0.19041.0\)](#)
  - [Windows 10 SDK \(10.0.18362.0\)](#)
  - [Incredibuild - Build Acceleration](#)
  - [Cocos](#)
  - [Unreal Engine installer](#)
  - [Android IDE support for Unreal Engine](#)

# Sign in to Visual Studio

Sync settings across devices, collaborate in real time, and integrate seamlessly with Azure Services.

[Sign in](#)[Create an account](#)[Skip this for now.](#)





Epic Games

News Samples Marketplace Library Twinmoc



Store

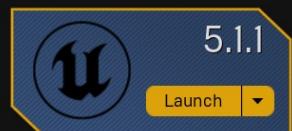


Library



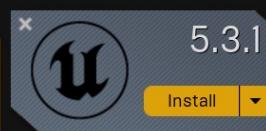
Unreal Engine

ENGINE VERSIONS +



5.1.1

Launch ▾



5.3.1

Install ▾

Installed Plugins



## Unreal Editor

Unreal Editor 5.3.1

75% - Initializing....



Copyright © Epic Games, Inc. All rights reserved.



RECENT PROJECTS

GAMES

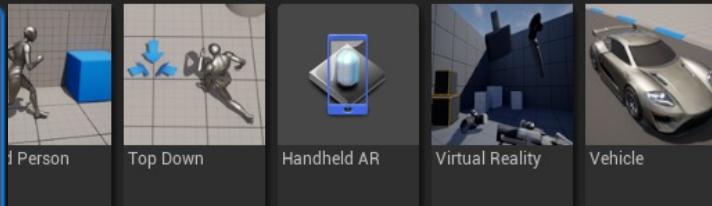
FILM / VIDEO &  
LIVE EVENTS

ARCHITECTURE

AUTOMOTIVE  
PRODUCT DESIGN &  
MANUFACTURING

SIMULATION

Project Location E:\Unity Projects\



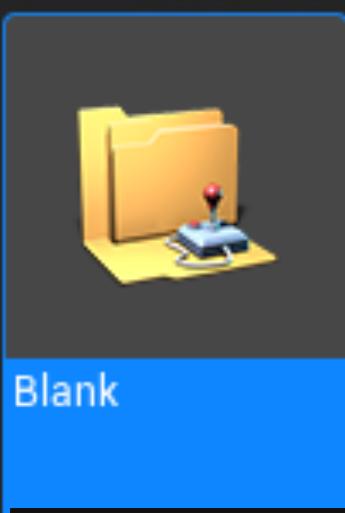
1 Person

Top Down

Handheld AR

Virtual Reality

Vehicle



Blank



Blank

A clean empty project with no code.

## Project Defaults

88%

Creating project...

INT

C++

Platform

Desktop

Preset

Maximum

Starter Content

Raytracing

Generating project files...

Project Location E:\Unity Projects\

Project Name MyCPlusPlusProject

Create



## CONTENT DETAIL

[Home](#) [Browse](#) [Industries](#) [Free](#) [On Sale](#) [Vault](#) [Help](#)

 Search Products.. Search


## Visual Studio Integration Tool

Microsoft Corp - Code Plugins - Oct 14, 2022

★★★★★ 50 reviews written | 49 of 50 questions answered

Visual Studio Integration Tool is a plugin that enables deep integration with Visual Studio 2022.

Free

OR

 Add to Cart Heart
**Supported Platforms**

**Supported Engine Versions**

4.27, 5.0 - 5.3

**Download Type**

Engine Plugin

# Project - Maps & Modes

Default maps, game modes and other map related settings.

Export...

Import...

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

## Default Modes

Default GameMode

Selected GameMode

Advanced

## Default Maps

Editor Startup Map

Editor Template Map Overrides

Game Default Map

Advanced

## Local Multiplayer

Use Splitscreen

Two Player Splitscreen Layout

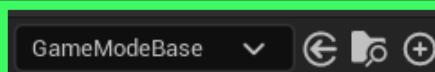
Three Player Splitscreen Layout

Four Player Splitscreen Layout

Skip Assigning Gamepad to Player 1

## Game Instance

Game Instance Class



OpenWorld

CURRENT ASSET

Edit

Copy

Paste

Clear

BROWSE

Search Assets

Advanced\_Lighting

Level

FPSMap

Level

Minimal\_Default

Level

StarterMap

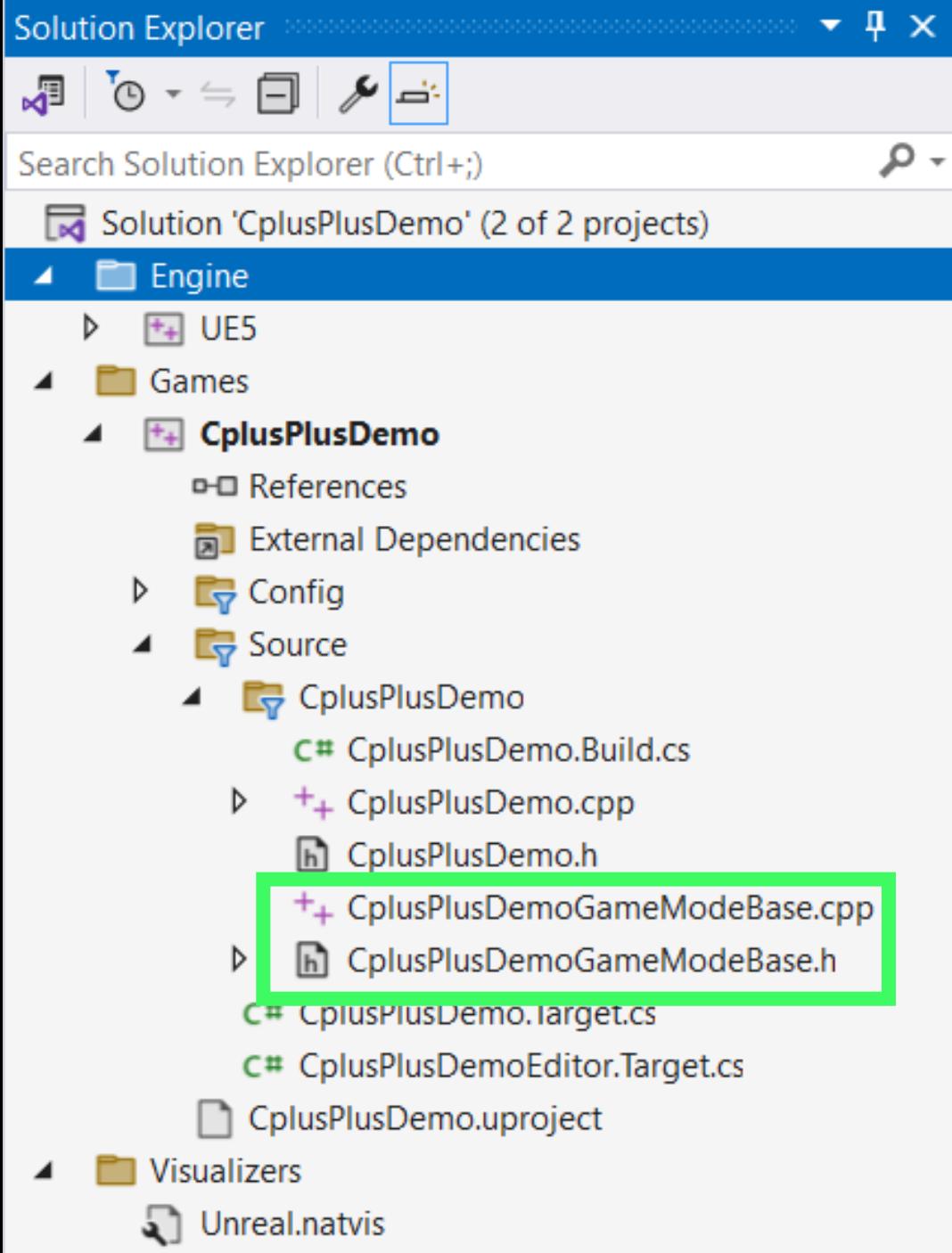
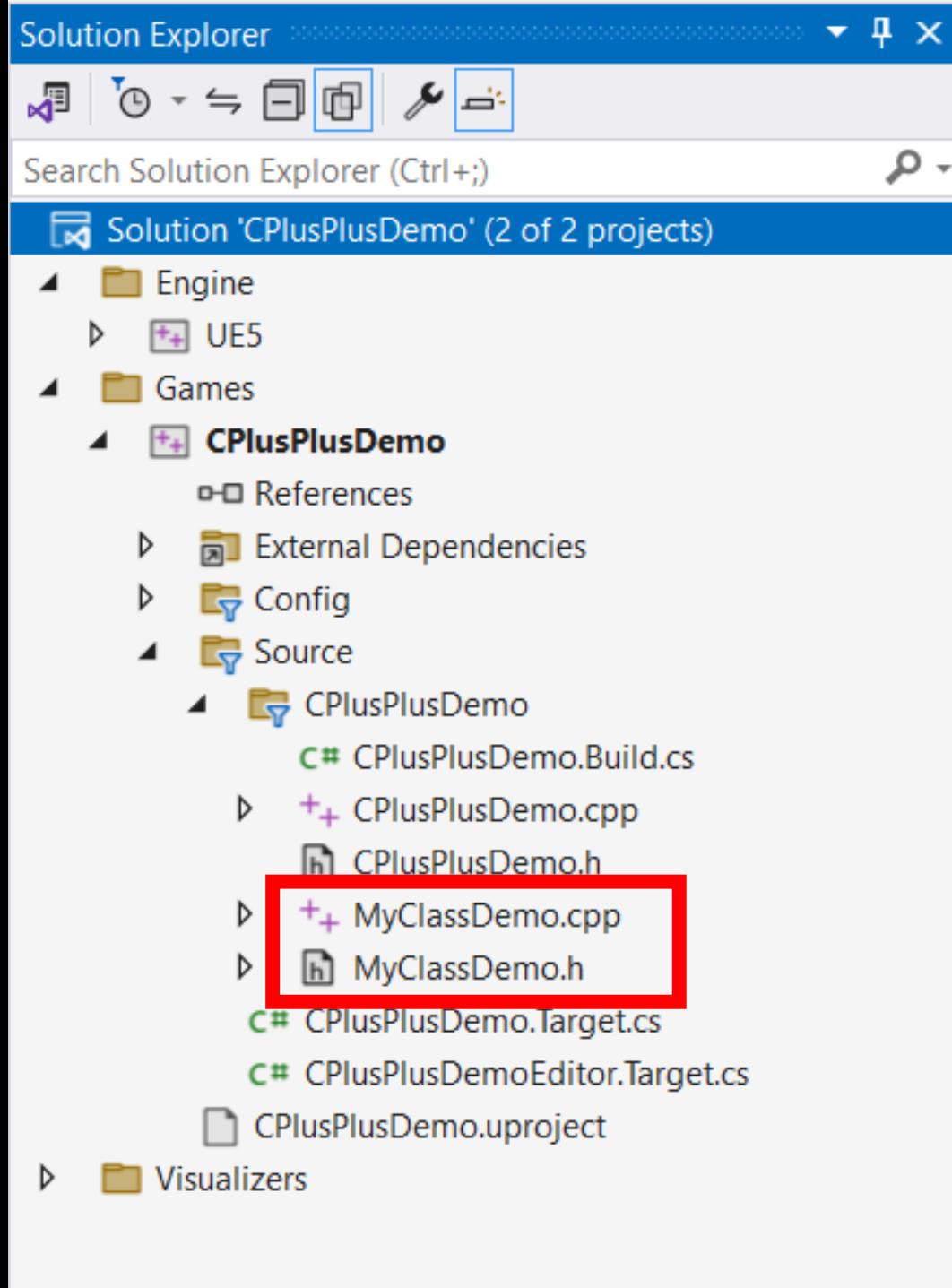
Level

FPSMap (Level)

Path: /Game

Asset Filepath Length: 47 / 210

Cooking Filepath Length: 98 / 260



File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search CplusPlusDemo

Live Share

CplusPlusDem...ModeBase.cpp CplusPlusDe...eModeBase.h\* ACplusPlusDemoGameModeBase

CplusPlusDemo

```
1 // Copyright Epic Games, Inc. All Rights Reserved.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/GameModeBase.h"
7 #include "CplusPlusDemoGameModeBase.generated.h"
8
9 /**
10 *
11 */
12 UCLASS()
13 class CPLUSPLUSDEMO_API ACplusPlusDemoGameModeBase : public AGameModeBase
14 {
15     GENERATED_BODY()
16
17     virtual void StartPlay() override; //This function declaration allows you to override
18 //StartPlay so that you can print a log message
19 // to the screen when gameplay begins.
20
21 };
22
```

Solution Explorer

Solution 'CplusPlusDemo' (2 of 2 projects)

- Engine
- UE5
- Games
- CplusPlusDemo
  - References
  - External Dependencies
  - Config
  - Source
    - CplusPlusDemo
      - CplusDemo.Build
      - CplusDemo.cpp
      - CplusDemo.h
      - CplusDemoGame
    - CplusDemoGame
    - CplusDemo.Target.cs
    - CplusDemoEditor.Target.cs
  - CplusPlusDemo.uproject
- Visualizers
- Unreal.nativs

Solution Explorer Git Changes

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help | Search | CplusPlusDemo | MS -

Development Editor Win64 Local Windows Debugger | Live Share

CplusPlusDe...odeBase.cpp\* CplusPlusDe...meModeBase.h

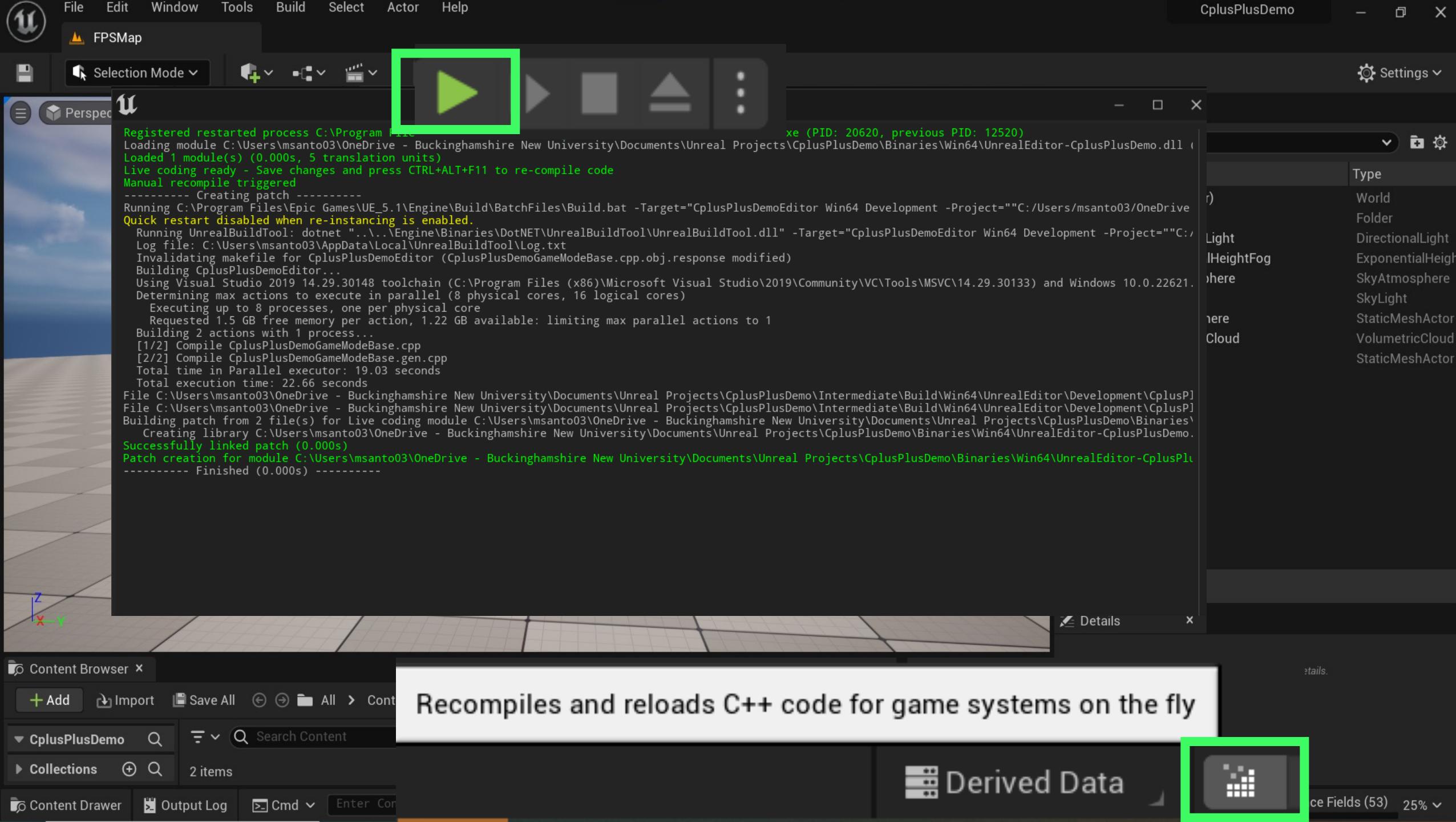
CplusPlusDemo ACplusPlusDemoGameModeBase StartPlay()

```
1 // Copyright Epic Games, Inc. All Rights Reserved.
2
3
4 #include "CplusPlusDemoGameModeBase.h"
5
6 void ACplusPlusDemoGameModeBase::StartPlay()
7 {
8     Super::StartPlay(); //StartPlay() will print a new debug message
9         //("Hello World, this is FPSGameModeBase!")
10        // to the screen in yellow text for five seconds
11        // when gameplay begins.
12
13     check(GEngine != nullptr);
14
15     // Display a debug message for five seconds.
16     // The -1 "Key" value argument prevents the message from being updated or refreshed.
17     GEngine->AddOnScreenDebugMessage(-1, 5.0f, FColor::Yellow, TEXT("Hello World, this is FPSGameMode!"));
18 }
19 }
```

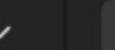
Command: cmd.exe /C ""C:\Program Files\Epic Games\UE\_5.1\Engine\Build\BatchFiles\Build.bat" CplusPlusDemoEditor Win64 Development -Project="C:\Users\msanto03\OneDrive - Buckinghamshire County Council\Documents\GitHub\CplusPlusDemo\Project\Project.uproject"

Solution Explorer Git Changes

Ready Add to Source Control Select Repository



Selection Mode ▾



**You might wonder why our log message isn't showing on screen in PIE mode.**

**It's because the Editor is using the default Game Mode at this development stage.**

Selection Mode ▾



Add Blueprint Class

X

## NAME YOUR NEW CPLUS PLUS DEMO GAME MODE BASE

Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.

When you click the "Create" button below, a new Blueprint class will be created.

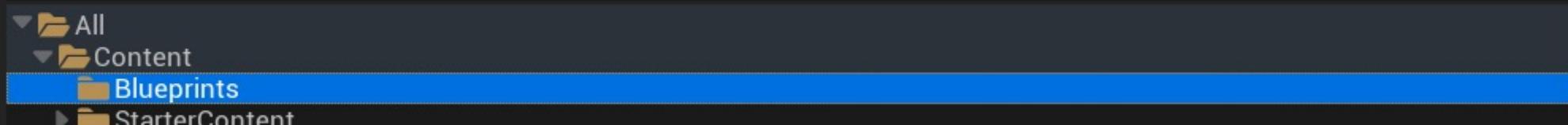
Class Type

Name

MyCplusPlusDemoGameModeBase

Path

Search Folders



Create Blueprint Class

Cancel

Collections



1 item (1 selected)

MyCplusPlusDemoGa... x Parent class: Cplus Plus Demo Game Mode Base

Hide Unrelated Class Settings Class Defaults Simulation No debug object selected

Event Graph x

Saves this asset (Ctrl+S)

DefaultS...

My Blu... x

+ Add

GRAPHS

EventGraph

Event BeginPlay

This node is disabled and will not be called. Drag off pins to build functionality.

Executable actions Context Sensitive

Search

Select a Component to see available Events & Functions

Accessibility

Actor Tick

Start with Tick Enabled

Tick Interval (secs) 0.0

Allow Tick Before Begin Play

Advanced

Classes

Game Session Class	GameSession
Game State Class	GameStateBase
Player Controller Class	PlayerController
Player State Class	PlayerState
HUD Class	HUD
Default Pawn Class	DefaultPawn

Content Browser x

+ Add Import Save All

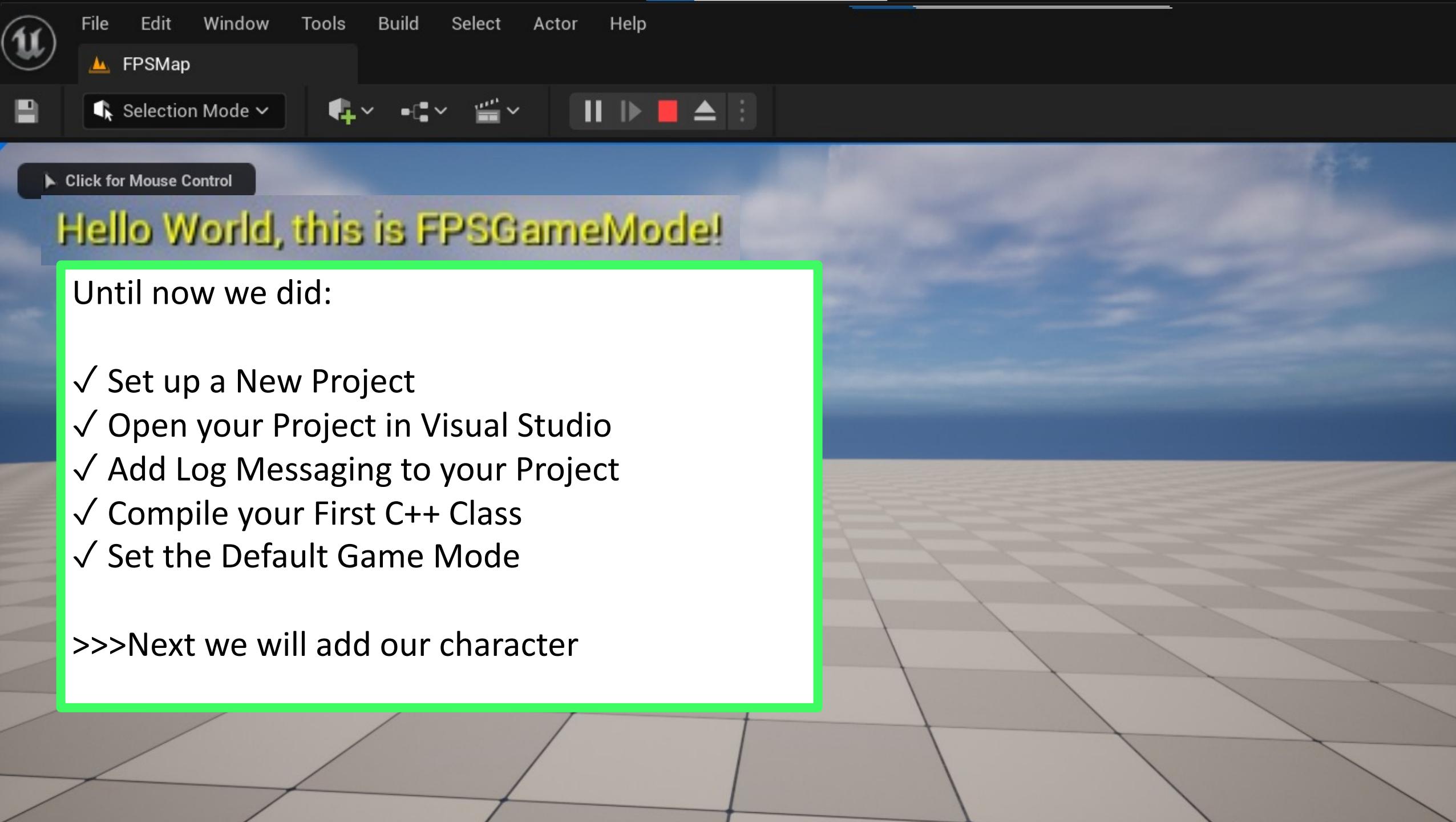
All Content Blueprints

Settings

CplusPlusDemo

All Content Blueprints StarterContent C++ Classes CplusPlusDemo

MyCplusPlus DemoGame ModeBase



File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search CplusPlusDemo Live Share ADMIN

FPSCharacter.cpp FPSCharacter.h CplusPlusDem...ModeBase.cpp CplusPlusDe...meModeBase.h

CplusPlusDemo (Global Scope)

// Fill out your copyright notice in the Description page of Project Settings.

```
#include "FPSCharacter.h"

// Sets default values
AFPSCharacter::AFPSCharacter()
{
    // Set this character to call Tick() every frame
    PrimaryActorTick.bCanEverTick = true;

    Super::BeginPlay();

    // Called every frame
    void AFPSCharacter::Tick(float DeltaTime)
    {
        Super::Tick(DeltaTime);
    }
}
```

ve performance if you don't need it.

Output

Show output from: Unreal Engine Integration Logging

Command: cmd.exe /C ""C:\Program Files\Epic Games\UE\_5.1\Engine\Build\BatchFiles\Build.bat" CplusPlusDemoEditor Win64 Development -Project="C:\Users\msanto03\OneDrive - Buckinghamshire New University\Documents\UNREAL PROJECTS\CPLUSPLUSDEMO\Source\CPLUSPLUSDEMO\CPLUSPLUSDEMOGAMEMODEBASE.uproject"

Running UnrealHeaderTool for "C:\USERS\MSANTO03\ONEDRIVE - BUCKINGHAMSHIRE NEW UNIVERSITY\DOCUMENTS\UNREAL PROJECTS\CPLUSPLUSDEMO\Source\CPLUSPLUSDEMO\CPLUSPLUSDEMOGAMEMODEBASE.h"

Command: cmd.exe /C ""C:\Program Files\Epic Games\UE\_5.1\Engine\Build\BatchFiles\Build.bat" CplusPlusDemoEditor Win64 Development -Project="C:\Users\msanto03\OneDrive - Buckinghamshire New University\Documents\UNREAL PROJECTS\CPLUSPLUSDEMO\Source\CPLUSPLUSDEMO\CPLUSPLUSDEMOGAMEMODEBASE.uproject"

Running UnrealHeaderTool for "C:\USERS\MSANTO03\ONEDRIVE - BUCKINGHAMSHIRE NEW UNIVERSITY\DOCUMENTS\UNREAL PROJECTS\CPLUSPLUSDEMO\Source\CPLUSPLUSDEMO\CPLUSPLUSDEMOGAMEMODEBASE.h"

Command: cmd.exe /C ""C:\Program Files\Epic Games\UE\_5.1\Engine\Build\BatchFiles\Build.bat" CplusPlusDemoEditor Win64 Development -Project="C:\Users\msanto03\OneDrive - Buckinghamshire New University\Documents\UNREAL PROJECTS\CPLUSPLUSDEMO\Source\CPLUSPLUSDEMO\CPLUSPLUSDEMOGAMEMODEBASE.uproject"

Analyzing Blueprints - Project: CplusPlusDemo.uproject

"C:\Program Files\Epic Games\UE\_5.1\Engine\Binaries\Win64\UnrealEditor-Cmd.exe" "C:\Users\msanto03\OneDrive - Buckinghamshire New University\Documents\Unreal Projects\CplusPlusDemo"

Solution Explorer

Search Solution Explorer (Ctrl+Shift+F)

Solution 'CplusPlusDemo' (2 of 2 projects)

- Engine
- UE5
- Games
- CplusPlusDemo
  - References
  - External Dependencies
  - Config
  - Platforms
  - Source
    - CplusPlusDemo
      - CplusPlusDemo.Build.cs
      - CplusPlusDemo.cpp
      - CplusPlusDemo.h
      - CplusPlusDemoGameModeBase.cpp
      - CplusPlusDemoGameModeBase.h
      - FPSCharacter.cpp
      - FPSCharacter.h
      - CplusPlusDemo.Target.cs

+ FPSCharacter.cpp

H FPSCharacter.h

Ready Add to Source Control Select Repository



## Build Solution

Ctrl+Shift+B

```
6     // Sets default values
7     AFPSCharacter::AFPSCharacter()
8
9
14     // Called when the game starts or when spawned
15     void AFPSCharacter::BeginPlay()
16     {
17         Super::BeginPlay();
18         check(GEngine != nullptr);
19
20         // Display a debug message for five seconds.
21         // The -1 "Key" value argument prevents the message from being updated on refreshes.
```

### Content Browser x

+ Add

Import

Save All



All > C++ Classes > CplusPlusDemo

Settings

### CplusPlusDemo

Search CplusPlusDemo



CplusPlusDemoGame  
ModeBase

FPSSCharacter

2 items



## Add Blueprint Class

### NAME YOUR NEW FPSCHARACTER

Enter a name for your new class. Class names may only contain alphanumeric characters, and may not contain a space.

When you click the "Create" button below, a new Blueprint class will be created.

#### Class Type

Name

BP\_FPSCharacter

Path

Search Folders

### Content Browser

+ Add

Import

Save All

↶

↷

↶

↷

All

Content > Blueprints

Settings

#### CplusPlusDemo

Q

Blueprints

Content

All

Starter Content

C++ Classes

CplusPlusDemo

C++

BP\_FPSCharacter

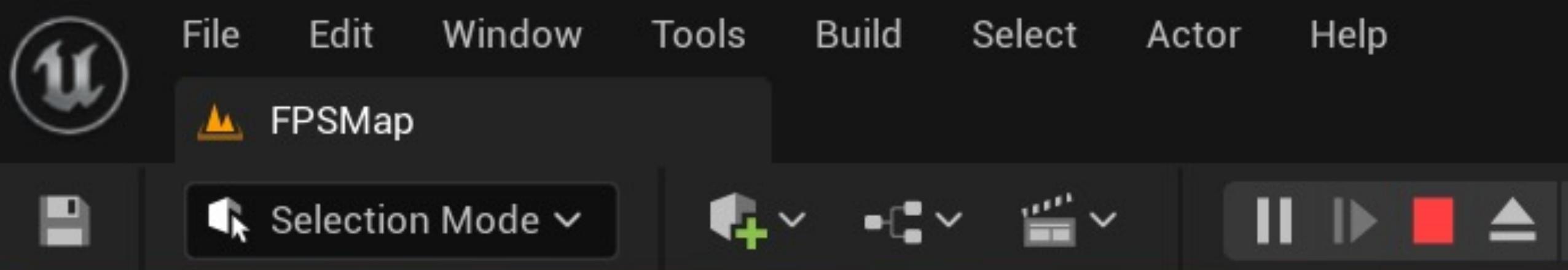
Search Blueprints



>>> Setting the Default Pawn Class

After extending the Game Mode to Blueprints, we set BP\_FPSCharacter as the default Pawn for our project.

2 items (1 selected)



▶ Click for Mouse Control

Hello World, this is FPSGameMode!

We are using FPSCharacter.

Next>>> Setting up Axis Mapping:  
Axis Mappings link keyboard, mouse, and controller inputs to game actions for smooth movements.  
In this step, we set up movement using the W, A, S, and D keys.

## Engine

AI System

Animation

Animation Modifiers

Audio

Chaos Solver

Cinematic Camera

Collision

Console

Control Rig

Cooker

Crowd Manager

Data Driven CVars

Debug Camera Controller

Enhanced Input

Enhanced Input (Editor Only)

Gameplay Debugger

Garbage Collection

General Settings

Hierarchical LOD

## ▶ Input

Interchange

Landscape

Level Sequence

MetaSounds

Navigation Mesh

Navigation System

Network

Search



## ▼ Engine - Input

Input settings, including default

These settings are saved in DefaultInput.ini, which is currently unavailable.

Export...

Import...

## ▼ Bindings



Axis and Action mappings are now deprecated, please use Enhanced Input Actions and Input Mapping Contexts instead.



## ▼ Axis Mappings

## MoveForward



W



Scale

1.0



S



Scale

-1.0

## MoveRight



D



Scale

1.0



A



Scale

-1.0

Enable Legacy Input Scales

Enable Motion Controls

Filter Input by Platform User

Should Flush Pressed Keys on Viewport Focus Lost

Enable Dynamic Component Input Binding

Android Volume Down

## ▼ HTC Vive

Vive (L) Trackpad Down

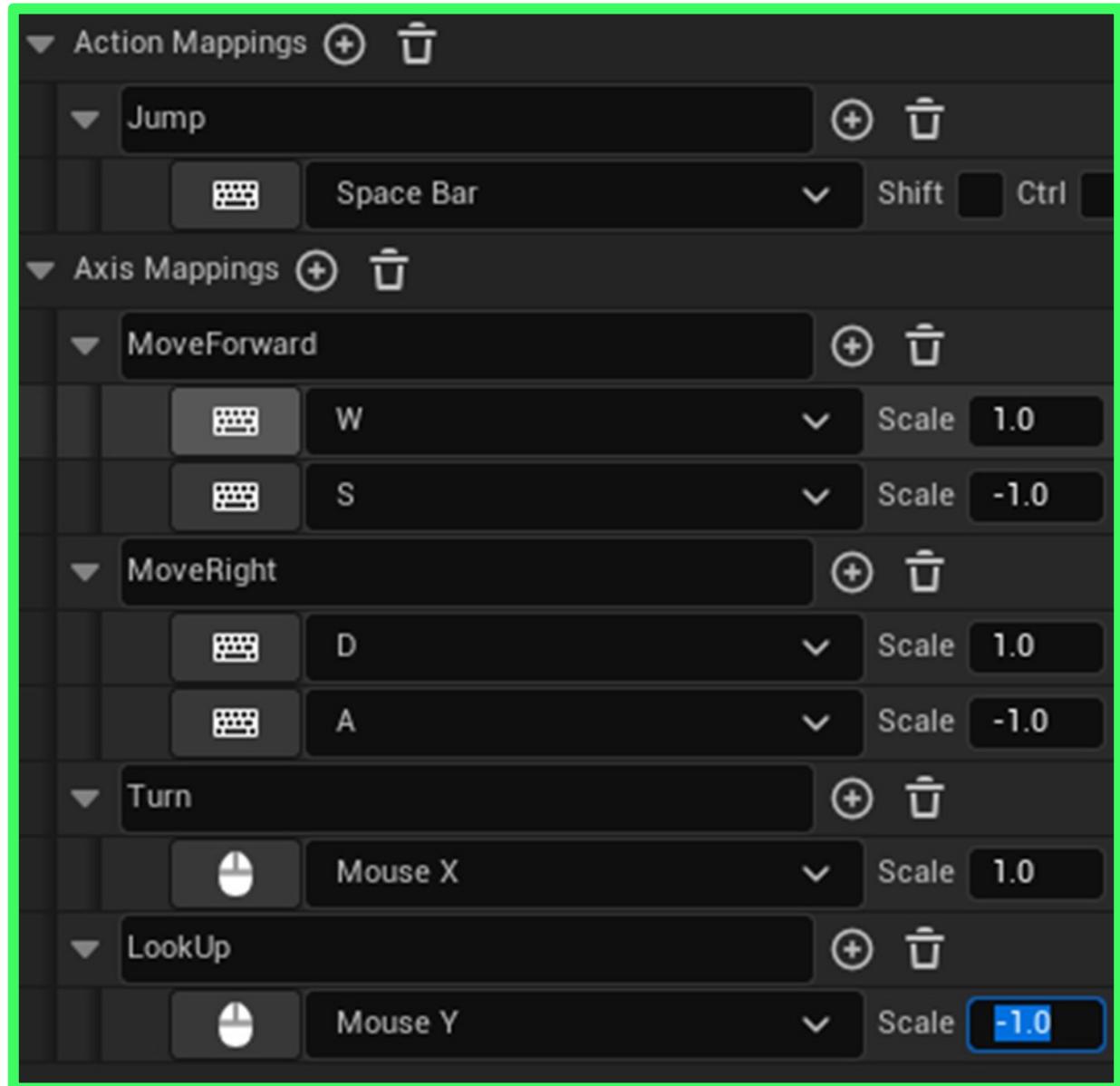
Vive (R) Trackpad Down

## ▼ Windows Mixed Reality

Mixed Reality (L) Thumbstick Down

Mixed Reality (L) Trackpad Down

Mixed Reality (R) Thumbstick Down



```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/Character.h"
7 #include "FPSCharacter.generated.h"
8
9 UCLASS()
10 class CPLUSPLUSDEMO_API AFPSCharacter : public ACharacter
11 {
12     GENERATED_BODY()
13
14 public:
15     // Sets default values for this character's properties
16     AFPSCharacter();
17
18 protected:
19     // Called when the game starts or when spawned
20     virtual void BeginPlay() override;
21
22     // Called to bind functionality to input
23     virtual void SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent) override;
24
25     // Handles input for moving forward and backward.
26     UFUNCTION()
27     void MoveForward(float Value);
28
29     // Handles input for moving right and left.
30     UFUNCTION()
31     void MoveRight(float Value);
```

An InputComponent handles input data and can be connected to an actor to let it receive input.

FPSCharacter.cpp   X   FPSCharacter.h\*   CplusPlusDem...ModeBase.cpp   CplusPlusDe...meModeBase.h

+ CplusPlusDemo

↓ AFPSCharacter

BeginPlay()

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "FPSCharacter.h"
5
6 // Sets default values
7 AFPSCharacter::AFPSCharacter()
8 {
9     // Set this character to call Tick() every frame. You can turn this off to improve performance if you don't need it.
10    PrimaryActorTick.bCanEverTick = true;
11 }
12
13
14 // Called when the game starts or when spawned
15 void AFPSCharacter::BeginPlay()
16 {
17     Super::BeginPlay();
18     check(GEngine != nullptr);
19
20     // Display a debug message for five seconds.
21     // The -1 "Key" value argument prevents the message from being updated or refreshed.
22     GEngine->AddOnScreenDebugMessage(-1, 5.0f, FColor::Red, TEXT("We are using FPSCharacter."));
23 }
24
25 // Called every frame
26 void AFPSCharacter::Tick(float DeltaTime)
27 {
28     Super::Tick(DeltaTime);
29 }
30
31
32 // Called to bind functionality to input
33 void AFPSCharacter::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
34 {
35     Super::SetupPlayerInputComponent(PlayerInputComponent);
36
37     // Set up "movement" bindings.
38     PlayerInputComponent->BindAxis("MoveForward", this, &AFPSCharacter::MoveForward);
39     PlayerInputComponent->BindAxis("MoveRight", this, &AFPSCharacter::MoveRight);
}
```

```
// Called to bind functionality to input
void AFPSCharacter::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
{
    Super::SetupPlayerInputComponent(PlayerInputComponent);

    // Set up "movement" bindings.
    PlayerInputComponent->BindAxis("MoveForward", this, &AFPSCharacter::MoveForward);
    PlayerInputComponent->BindAxis("MoveRight", this, &AFPSCharacter::MoveRight);
}
```

```
// Sets jump flag when key is pressed.
```

```
UFUNCTION()
```

```
void StartJump();
```

```
// Clears jump flag when key is released.
```

```
UFUNCTION()
```

```
void StopJump();
```

**To test character movement:**

Save the FPSCharacter header and C++ files.

Find FPSProject in Solution Explorer.

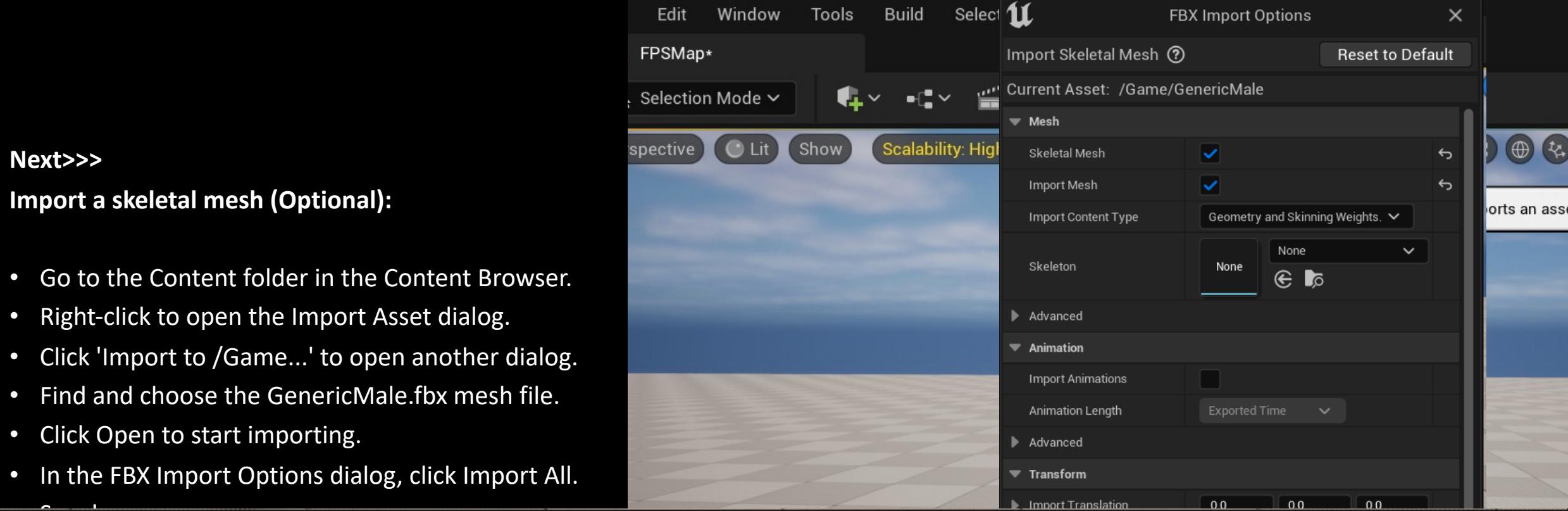
Right-click FPSProject and choose Build to compile.

```
FPSCharacter.cpp  X  FPSCharacter.h*  CplusPlusDem...ModeBase.cpp  CplusPlusDe...meModeBase.h
+ CplusPlusDemo
    ↓ AFPSCharacter
    ↓ BeginPlay()
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3
4 #include "FPSCharacter.h"
5
6 // Sets default values
7 AFPSCharacter::AFPSCharacter()
8 {
9     // Set this character to call Tick() every frame. You can turn this off to improve performance if you don't need it.
10
11     Super::Initialize();
12
13     // ...
14
15     // Called to bind functionality to input
16     void AFPSCharacter::SetupPlayerInputComponent(UInputComponent* PlayerInputComponent)
17     {
18         Super::SetupPlayerInputComponent(PlayerInputComponent);
19
20         // Set up "movement" bindings.
21         PlayerInputComponent->BindAxis("MoveForward", this, &AFPSCharacter::MoveForward);
22         PlayerInputComponent->BindAxis("MoveRight", this, &AFPSCharacter::MoveRight);
23
24         // Set up "look" bindings.
25         PlayerInputComponent->BindAxis("Turn", this, &AFPSCharacter::AddControllerYawInput);
26         PlayerInputComponent->BindAxis("LookUp", this, &AFPSCharacter::AddControllerPitchInput);
27
28         // Set up "action" bindings.
29         PlayerInputComponent->BindAction("Jump", IE_Pressed, this, &AFPSCharacter::StartJump);
30         PlayerInputComponent->BindAction("Jump", IE_Released, this, &AFPSCharacter::StopJump);
31     }
}
```

The screenshot shows a Microsoft Visual Studio interface with the following components:

- Code Editor:** The main window displays the `CplusPlusDemo.cpp` file. The code defines a class `AFPSCharacter` derived from `ACharacter`. It includes methods for `BeginPlay`, `Tick`, `SetupPlayerInputComponent`, `MoveForward`, `MoveRight`, `StartJump`, and `StopJump`. The `MoveRight` method is highlighted.
- Solution Explorer:** Located on the right side, it shows the project structure for 'CplusPlusDemo' (2 of 2 projects). The solution includes 'Engine', 'UE5', 'Games', and 'CplusPlusDemo' with files like `CplusPlusDemo.cpp`, `CplusPlusDemo.h`, `CplusPlusDemoGameModeBase.h`, `FPSCharacter.cpp`, `FPSCharacter.h`, and `CplusPlusDemo.Target.cs`.
- Output Window:** At the bottom, the output window shows the build results:

```
Show output from: Build
Building solution: CplusPlusDemo.sln
Build started at 15:29 and took 01.215 seconds
Build succeeded.
0 succeeded, 0 failed, 0 up-to-date, 0 skipped
```
- Status Bar:** The status bar at the bottom indicates the current view is 'Development Editor'.



Next>>>

### Import a skeletal mesh (Optional):

- Go to the Content folder in the Content Browser.
- Right-click to open the Import Asset dialog.
- Click 'Import to /Game...' to open another dialog.
- Find and choose the GenericMale.fbx mesh file.
- Click Open to start importing.
- In the FBX Import Options dialog, click Import All.

>>>Setting up a Third Person Mesh  
Double-click the BP\_FPSCharacter Blueprint Class icon in  
Content > Blueprints to open it in the Blueprint Editor.

Click the Mesh component in the Components tab.



BP\_FPSCharacter\*

Parent class: FPSCharacter

Components

+ Add Search

BP\_FPSCharacter (Self)

Capsule Component (CollisionCylinder) Edit in C++

Arrow Component (Arrow) Edit in C++

Mesh (CharacterMesh0) Edit in C++

My Blueprint

+ Add Search

GRAPHS

EventGraph

- Event BeginPlay
- Event ActorBeginOverlap
- Event Tick

FUNCTIONS (32 OVERRIDABLE)

ConstructionScript

MACROS

VARIABLES

EVENT DISPATCHERS

Viewports

Perspective Lit

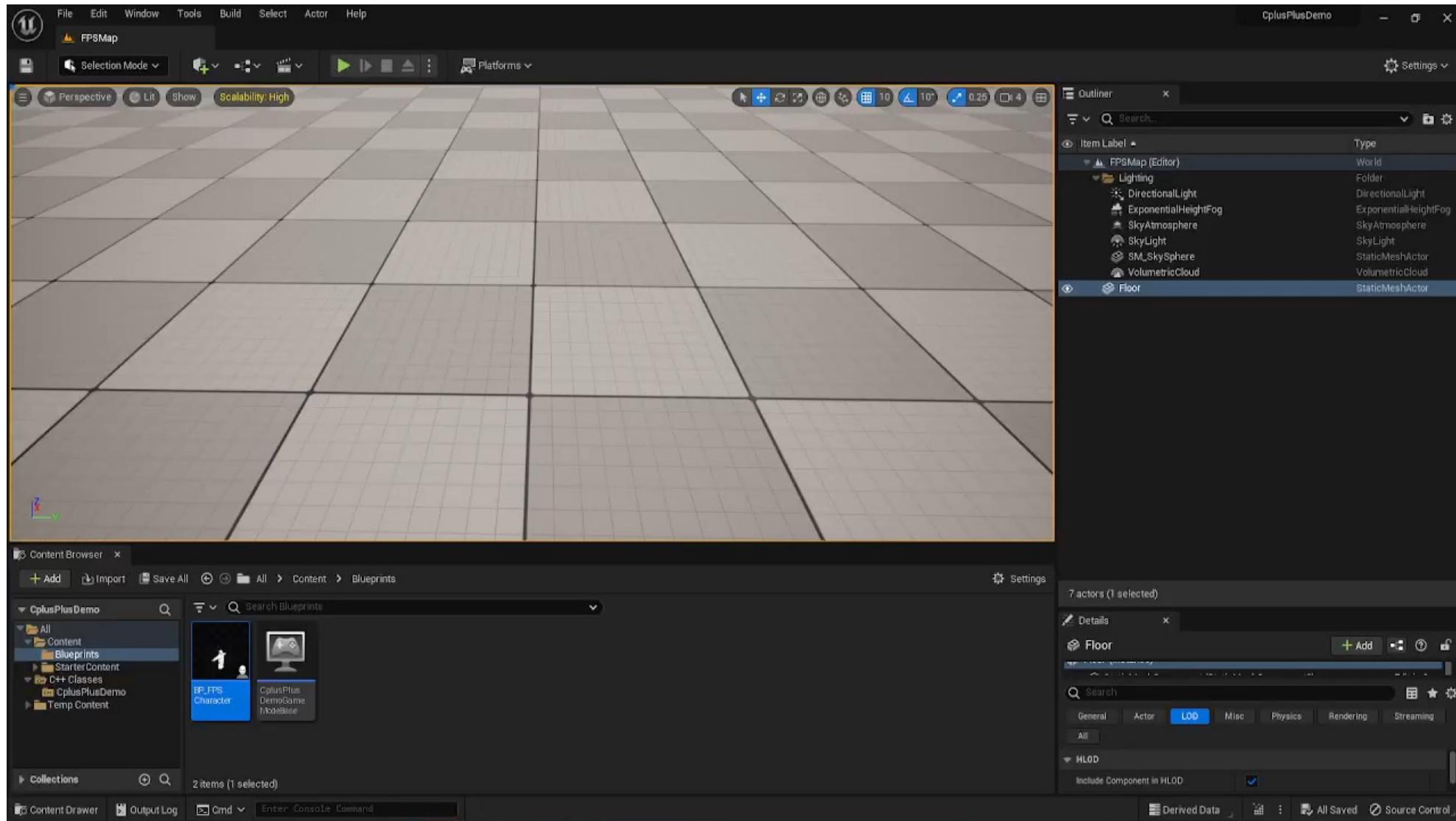
Event Graph

Details

Viewport Construction S... Event Graph Details

10 10° 0.25 4

Fn+F8 key to eject yourself from your pawn



The camera is inside the mesh's neck. Next, To fix this, we add more files to FPSCharacter.h for additional camera functions.

FPSCharacter.cpp

FPSCharacter.h

CplusPlusDem...ModeBase.cpp

CplusPlusDe...meModeBase.h

CplusPlusDemo

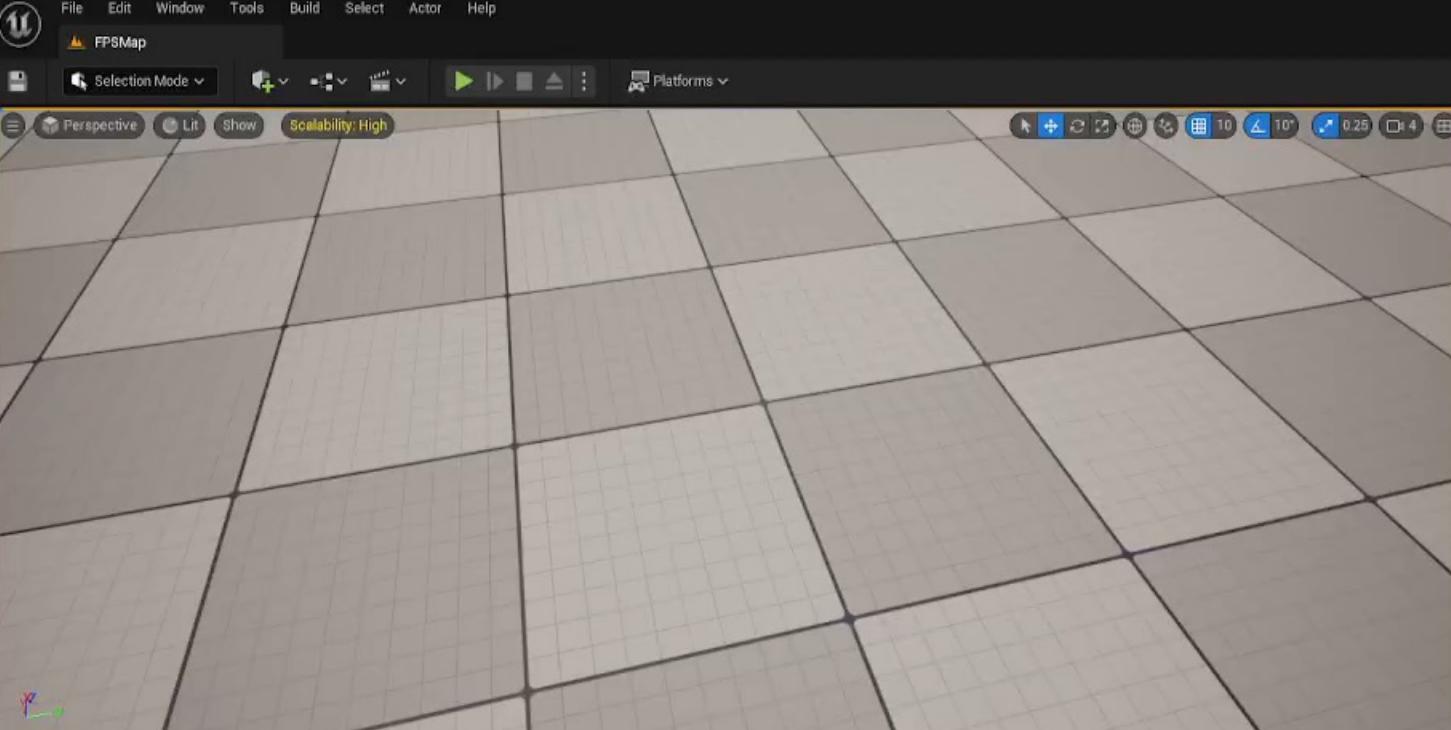
AFPSCharacter

```
1 // Fill out your copyright notice in the Description page of Project Settings.
2
3 #pragma once
4
5 #include "CoreMinimal.h"
6 #include "GameFramework/Character.h"
7 #include "FPSCharacter.generated.h"
8
9 #include "Camera/CameraComponent.h"
10 #include "Components/CapsuleComponent.h"...
11 ...
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
```

FPSCharacter.cpp X FPSCharacter.h CplusPlusDem...ModeBase.cpp CplusPlusDe...meModeBase.h

+ CplusPlusDemo - AFPSCharacter()

```
1 // Copyright Epic Games, Inc. All Rights Reserved.
2
3 #include "FPSCharacter.h"
4
5 // Sets default values
6 AFPSCharacter::AFPSCharacter()
7 {
8     // Set this character to call Tick() every frame. You can turn this off to improve performance if you don't need it.
9     PrimaryActorTick.bCanEverTick = true;
10
11     // Create a first person camera component.
12     FPSCameraComponent = CreateDefaultSubobject<UCameraComponent>(TEXT("FirstPersonCamera"));
13     check(FPSCameraComponent != nullptr);
14
15     // Attach the camera component to our capsule component.
16     FPSCameraComponent->SetupAttachment(CastChecked<USceneComponent>(GetCapsuleComponent()));
17 }
```



The screenshot shows the Unreal Engine 4 Editor interface. The main view is a 3D perspective editor window displaying a large, light-colored floor with a grid pattern. The Outliner panel on the right lists several game objects under the 'FPSMap (Editor)' folder, including 'Lighting' (DirectionalLight, ExponentialHeightFog, SkyAtmosphere, SkyLight, SM\_SkySphere, VolumetricCloud) and 'Floor' (StaticMeshActor). The Details panel at the bottom is open for the selected 'Floor' actor, showing tabs for General, Actor, LOD, Misc, Physics, Rendering, and Streaming. The LOD tab is currently selected.

For FPS MP games, use two character meshes:

- a full-body mesh for third-person view
- and a "weapon and hands" mesh for first-person view.

>>> Next, we will replace the mesh with a gun for easier demo.



1 of 11

## AK47 3D model

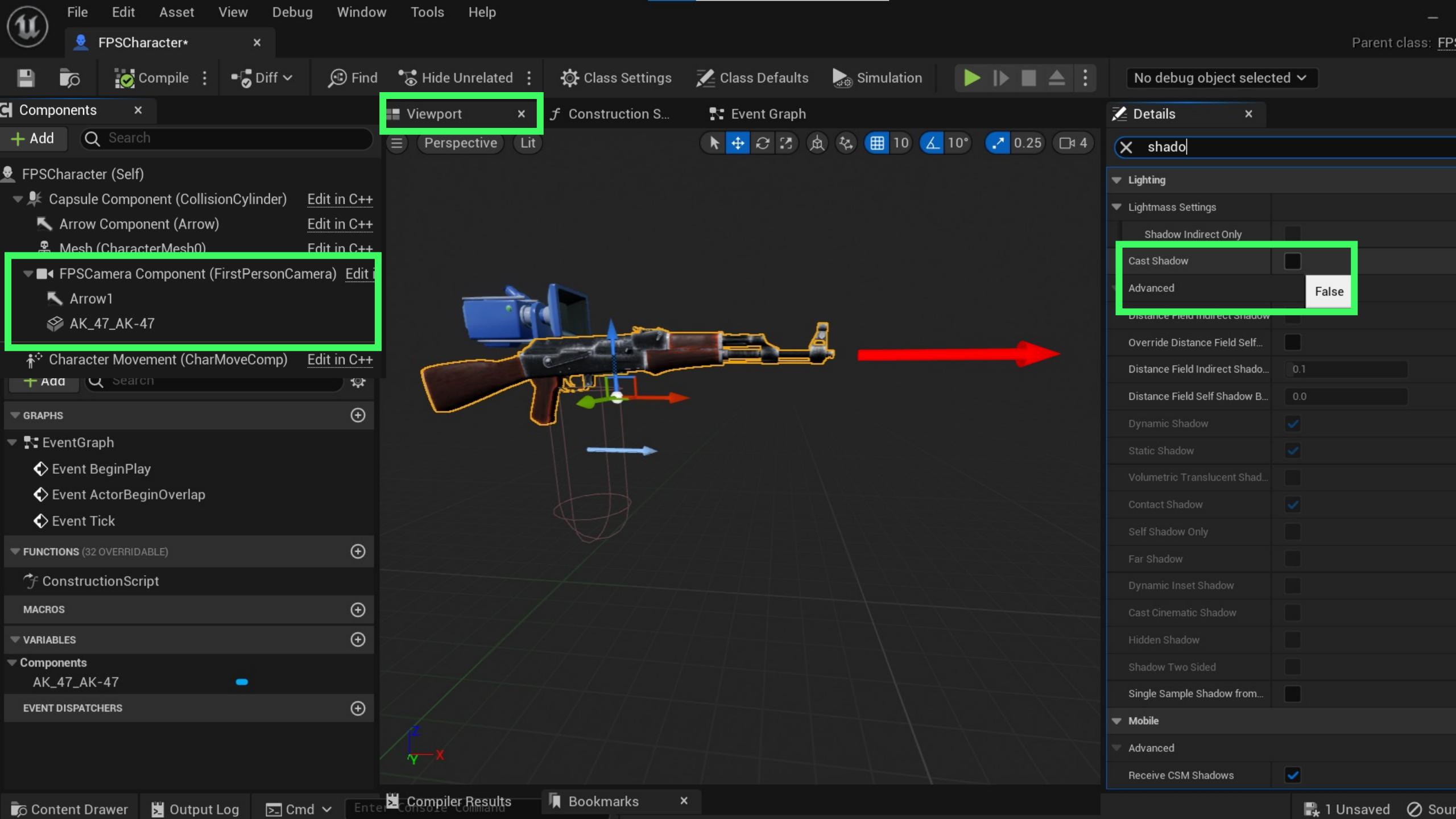
by CG VIEWPORT

Free

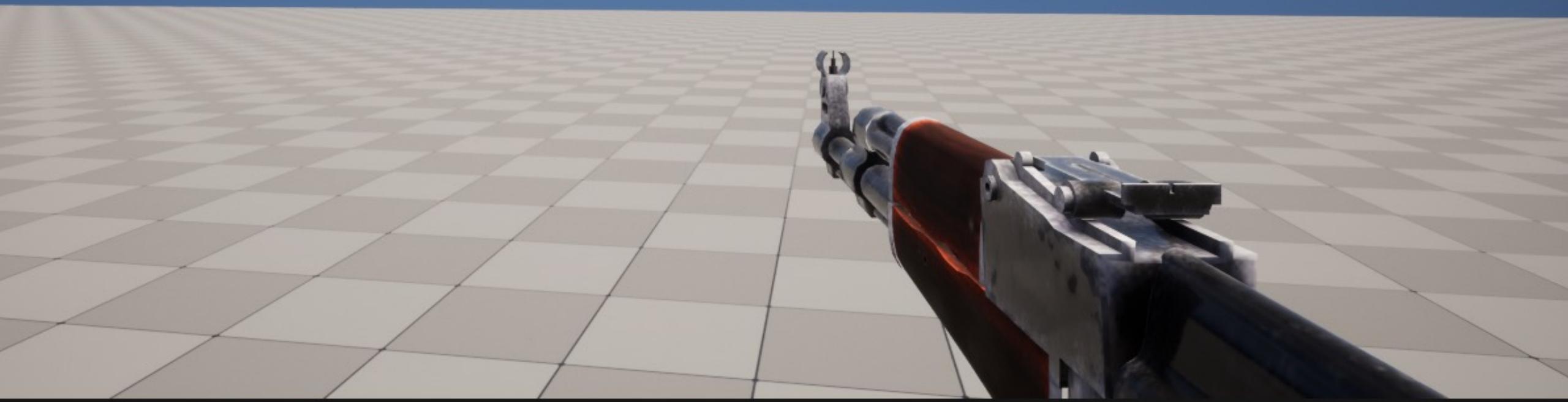
**Download**

3D Model License: Standard

Editorial Uses Only



in Mode



Viewport

f Construction Sc...

Event Graph

x



## BP\_Projectile &gt; Event Graph

This node is disabled and will not be called.  
Drag off pins to build functionality.

Event Tick



Delta Seconds



On Component Begin Overlap (Sphere)



Overlapped Component



Other Actor



Other Comp



Other Body Index



From Sweep



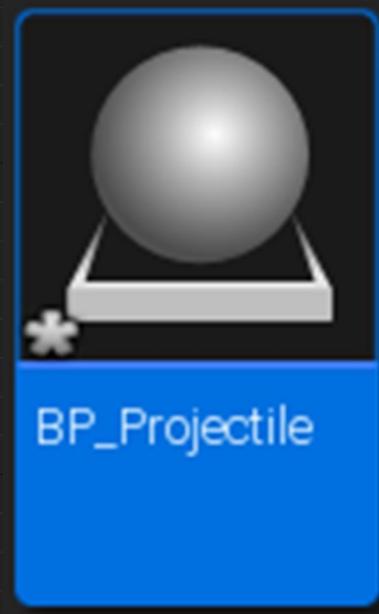
Sweep Result

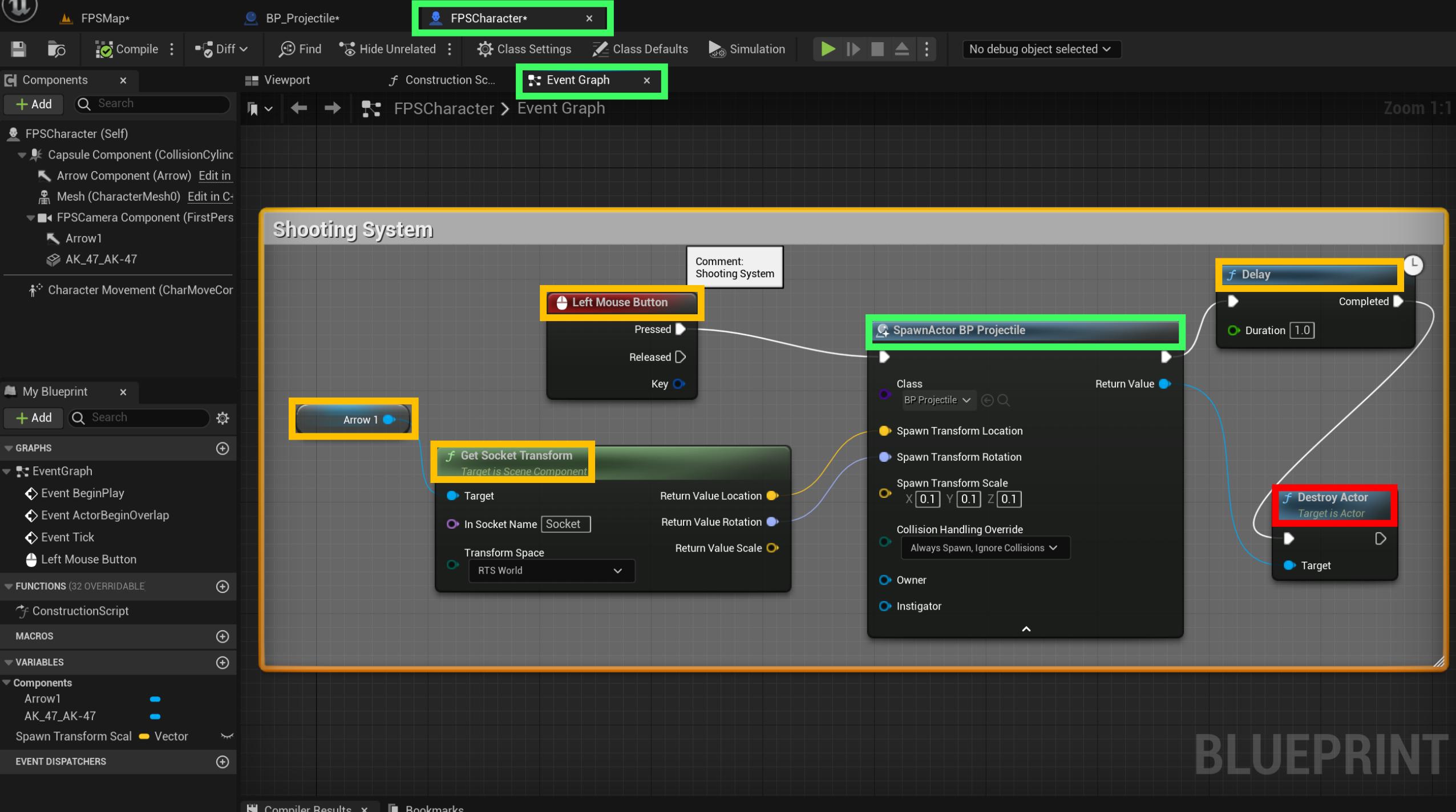
Destroy Actor

Target is Actor



Target [self]





# Conversion completed!

Download your converted file



ak-47-89833.wav

FINISHED

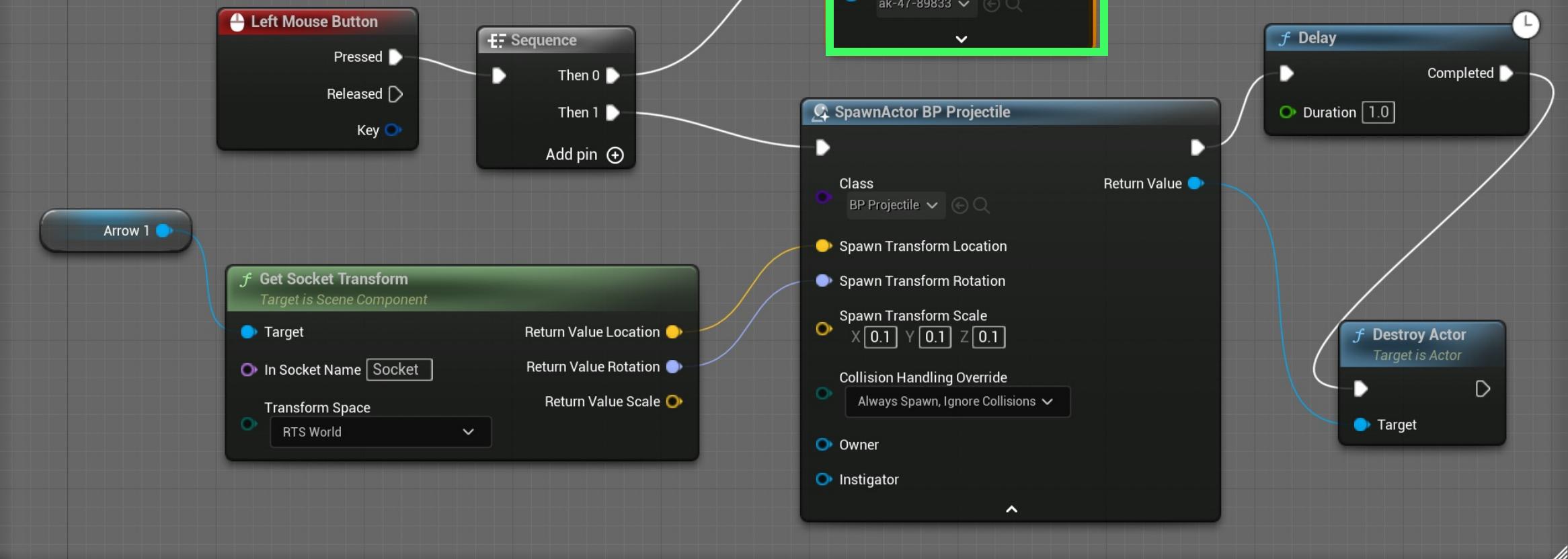
WAV / 194 KB

Download

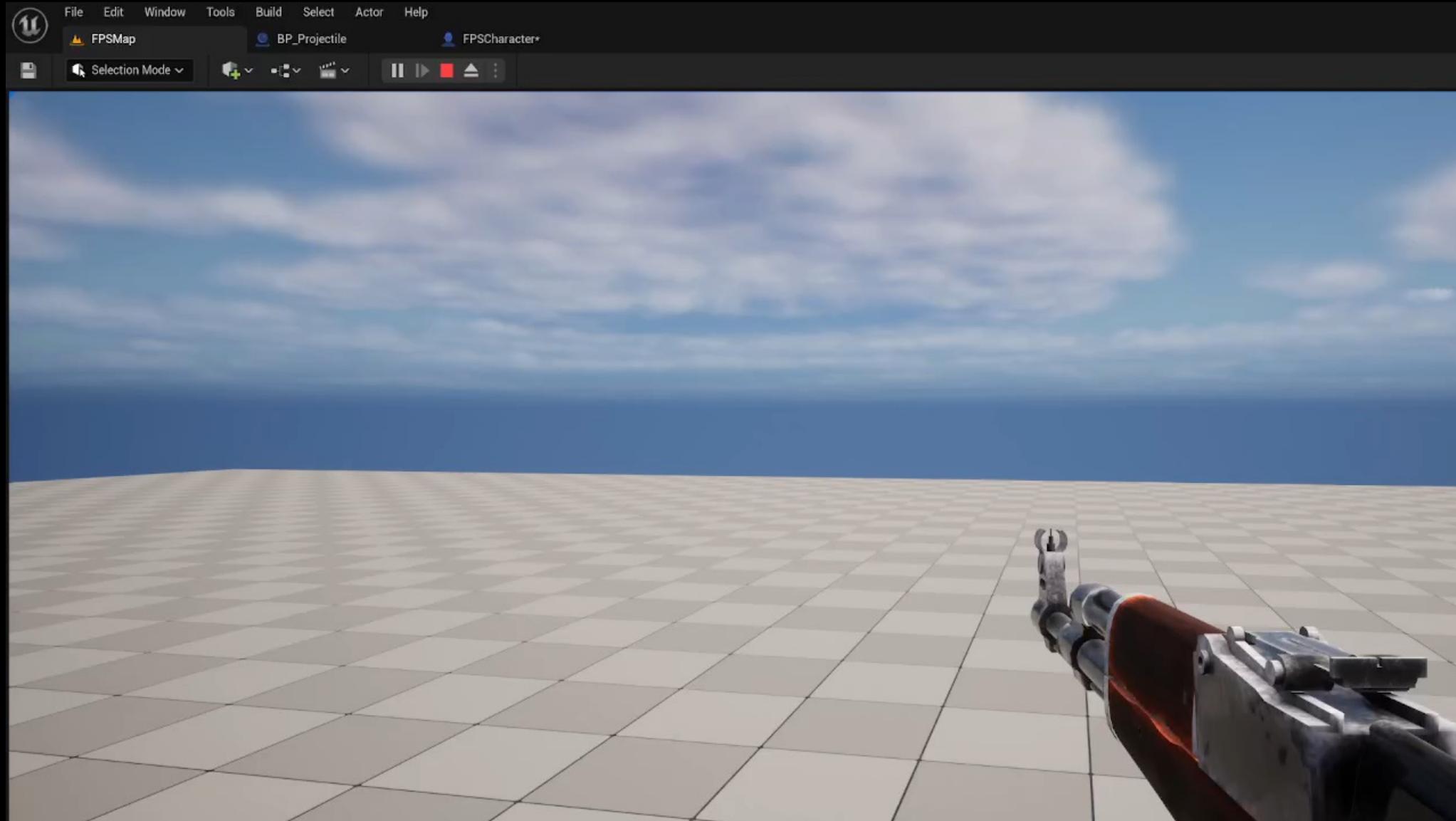
Files will be stored for 24 hours. Go to [My Files](#) to delete them manually.

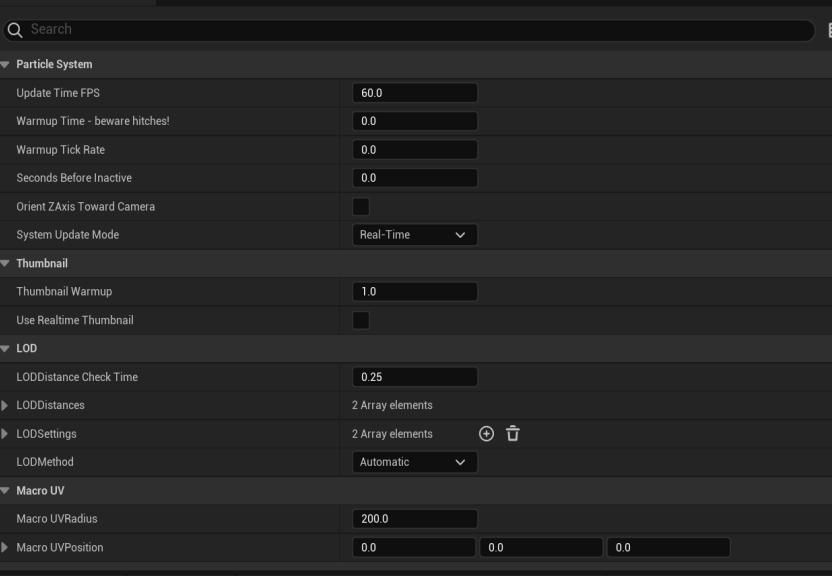
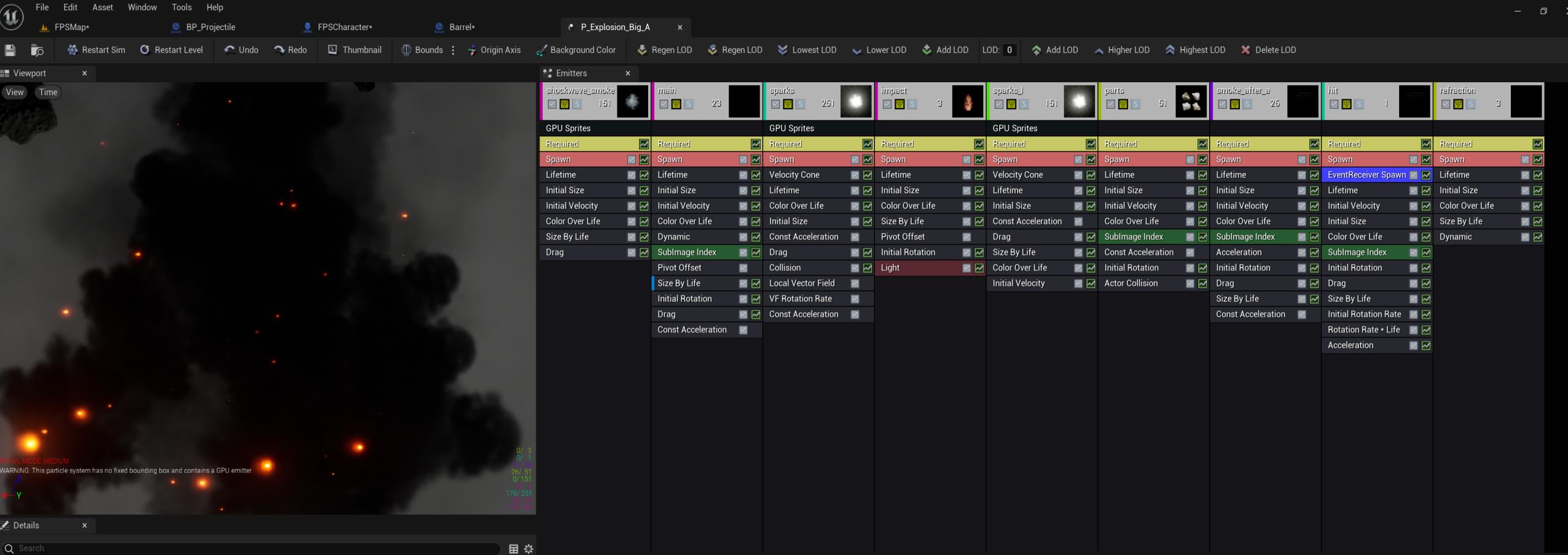
Convert more files

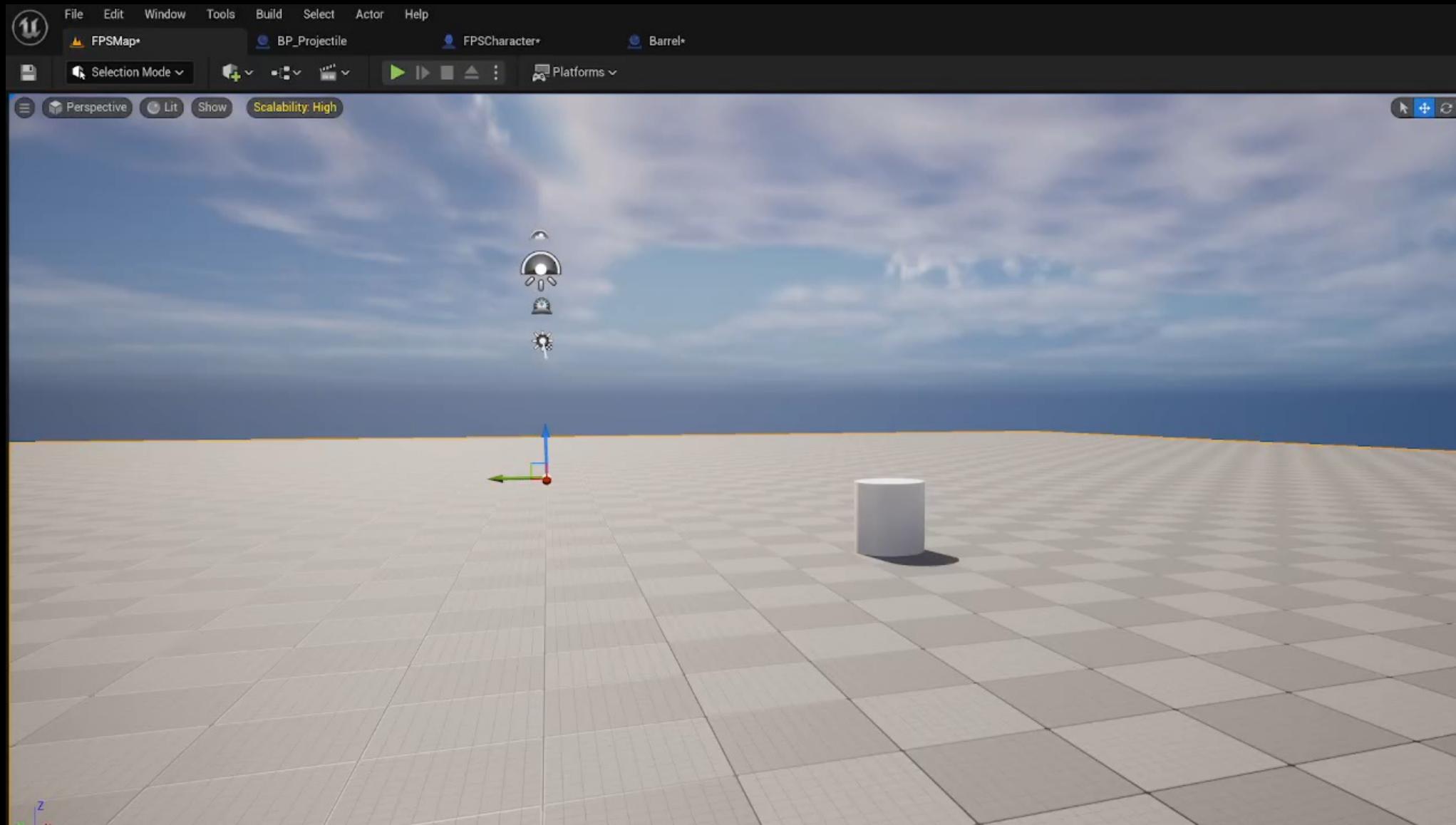
## Shooting System

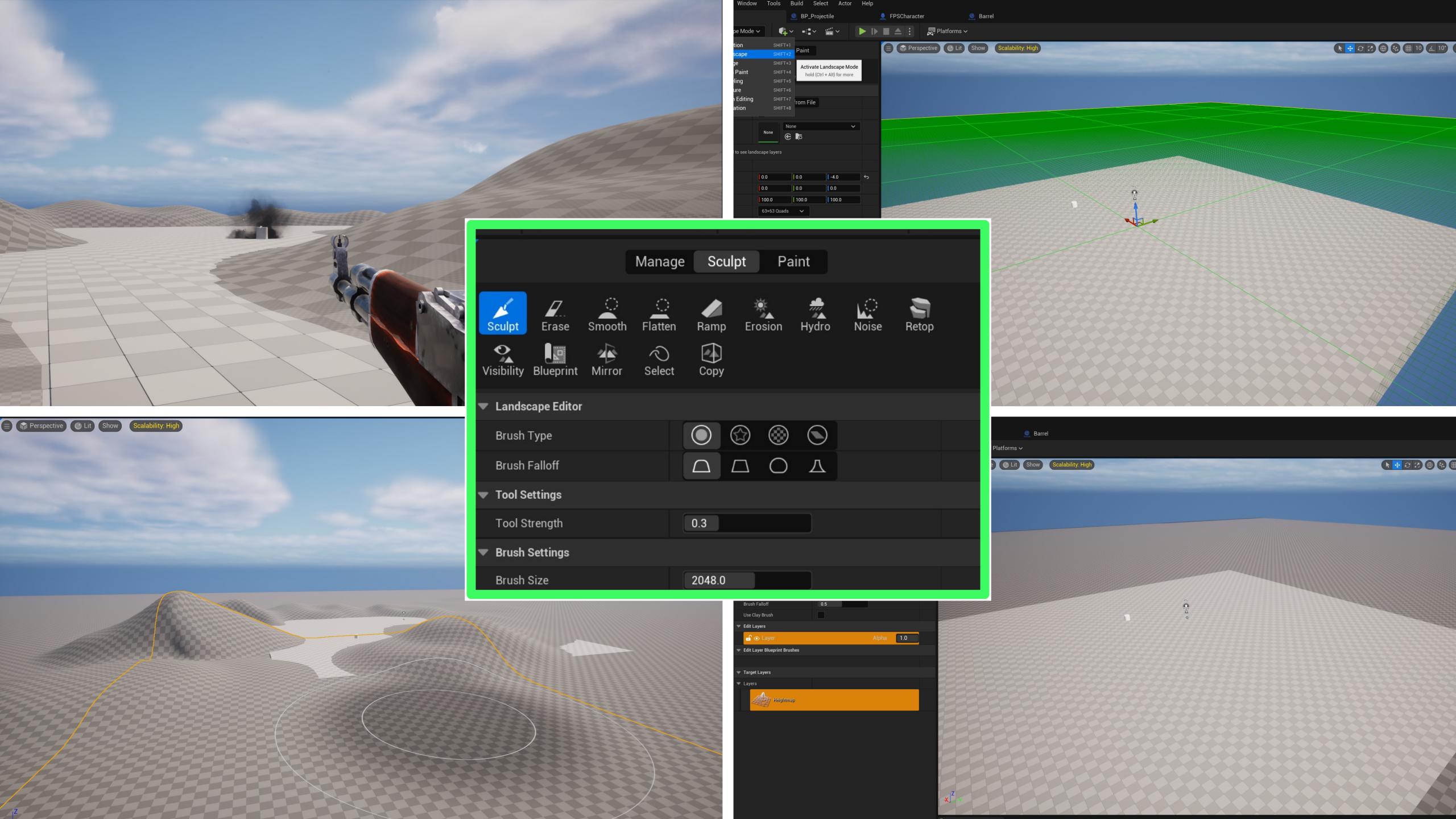


BLUEPRINT











Store



Library



Unreal Engine

&lt;Back

## CONTENT DETAIL

Home

Browse ▾

Industries ▾

Free ▾

On Sale

Submit Content Help

Search Products..



Free

OR

## Supported Platforms

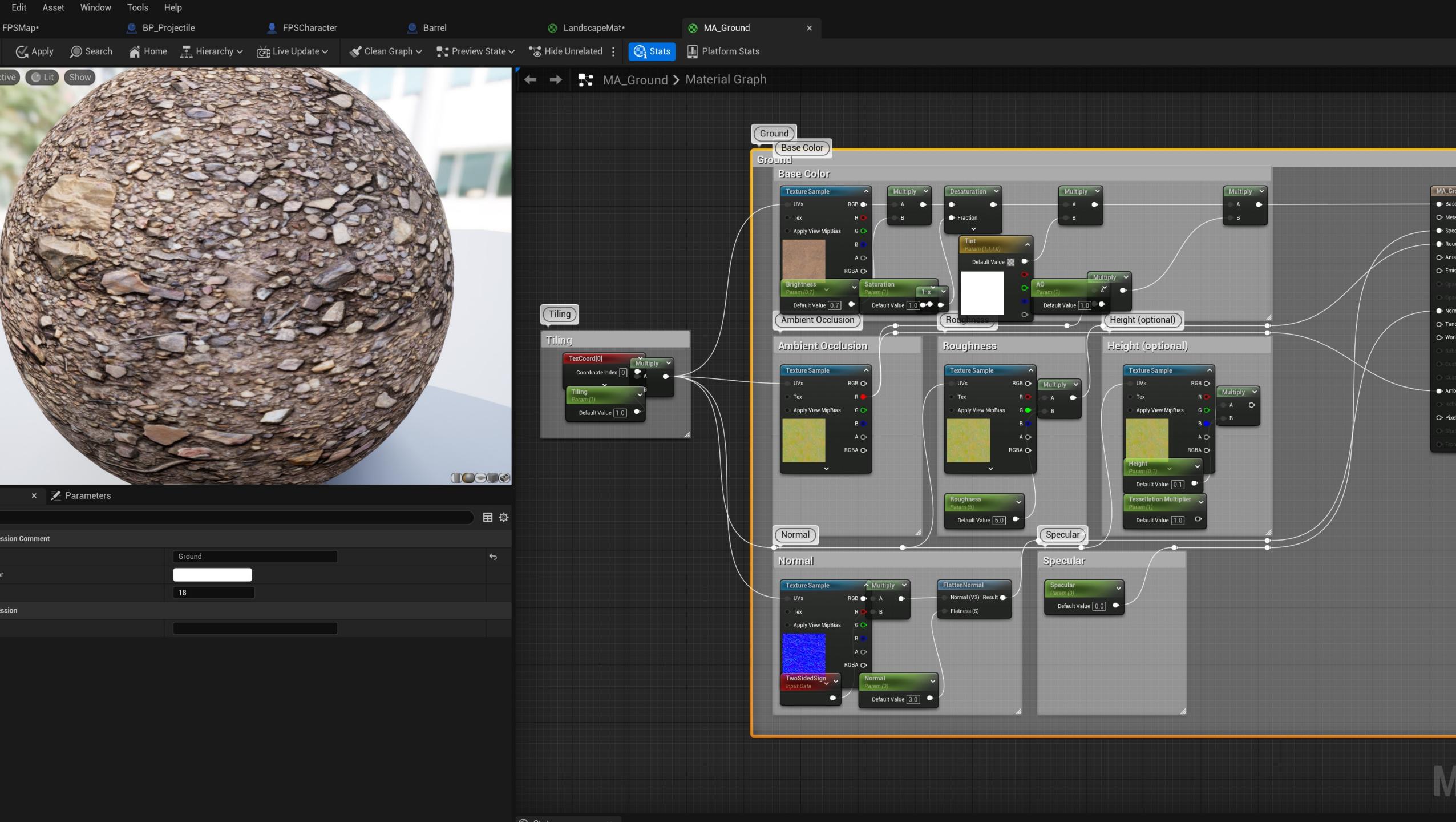


## Supported Engine Versions

PN\_tropicalGroundPlants &gt; Meshes

Search Meshes

tropicalPlant\_01\_01  
Static MeshtropicalPlant\_01\_02  
Static MeshtropicalPlant\_01\_03  
Static MeshtropicalPlant\_01\_04  
Static MeshtropicalPlant\_02\_01  
Static MeshtropicalPlant\_02\_02  
Static MeshtropicalPlant\_02\_03  
Static MeshtropicalPlant\_02\_04  
Static MeshtropicalPlant\_03\_01  
Static MeshtropicalPlant\_03\_02  
Static MeshtropicalPlant\_03\_03  
Static Mesh





FPSMap\*

BP\_Projectile

FPSCharacter

Barrel

MA\_Ground



Foliage Mode ▾



Platforms ▾



Reapply



## Paint

### Brush Options

Brush Size Paint Density Erase Density  Single Instance Mode: All Selected ▾ Place in Current Level

### Filters

 Landscape  Static Meshes  BSP Foliage  Translucent

Search Foliage



Perspective

Lit

Show

Scalability: High

Content Browser x

+ Add

Import

Save All

All

Content

PN\_tropicalGroundPlants

Meshe

FPSPProject

All

Content

AK\_47

Blueprints

Maps

PN\_tropicalGroundPlants

Blueprints

Map

Materials

Meshe

Textures

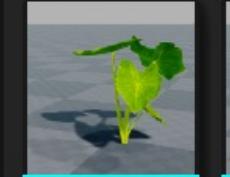
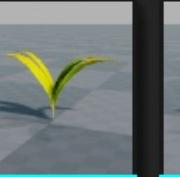
UE4\_ThirdPerson

Realistic\_Starter\_VFX\_Pack\_Vol2

C++ Classes

FPSPProject

Search Meshe

tropicalPlant\_01\_01  
Static MeshtropicalPlant\_01\_02  
Static MeshtropicalPlant\_01\_03  
Static MeshtropicalPlant\_01\_04  
Static MeshtropicalPlant\_02\_01  
Static MeshtropicalPlant\_03\_03  
Static MeshtropicalPlant\_03\_04  
Static MeshtropicalPlant\_04\_01  
Static MeshtropicalPlant\_04\_02  
Static MeshtropicalPlant\_04\_03  
Static Mesh





Content Browser x

# Online Resources & Further Learning



- Unreal Engine Documentation
- Unreal YouTube Channel
- Epic Games' Unreal Online Learning
- Forums like Unreal Engine Community and Stack Overflow

# Conclusion & Q&A

Thank you for your attention! I'm now open to any questions you might have.