

Level Up ToolKit Documentation

Tutorial for learning to use level up toolkit

Contents

Installing the software	1
1: Download Unreal 5.1	1
2: Download Level Up Toolkit	1
Recommended Resources for Learning UE5	2
UE5 Hotkeys	3
The Basics You Need To Know	4
Tools for Editor	6
Jump Range Visualization	6
Axial replication	7
Duplicate by Bounding Box	8
Duplicate by Custom Distance	9
Resources in Toolkits	11
Models	11
Measurement Visualization Features	11
Visual Guidance Features	11
Decorative Features	12
Animation	13
Levels	14
Making Interactive Experiences	15
Character Control	16
Moving Component	18
Rotating Component	19
Event Trigger Interface	21
Self Event Trigger	21
Pickup Event Trigger	22
Event Sender Component	22
Event Receiver Component	23
Event Sender Trigger	23
Event Receiver Trigger	24
Transformational Event Receiver Trigger	24
Pole	25
Door	25
Button	26
Remote Door	27
Hide or Show Object	28
Portal Enter	30
Portal Exit	31

Hanger	31
Conveying Hanger	32
Slide Down Pole	33
Climbable Objects	34
Health System	34
Dead Zone	34
Damage	35
Damage Once	35
Damage Loop	35
Spawn System	35
Respawn	35
Check Point	36
End Zone	36
Timer	36
Destructible	38
Conveyor Belt	41
Security Camera	42

Installing the software

STEP 1: Download Unreal 5.1

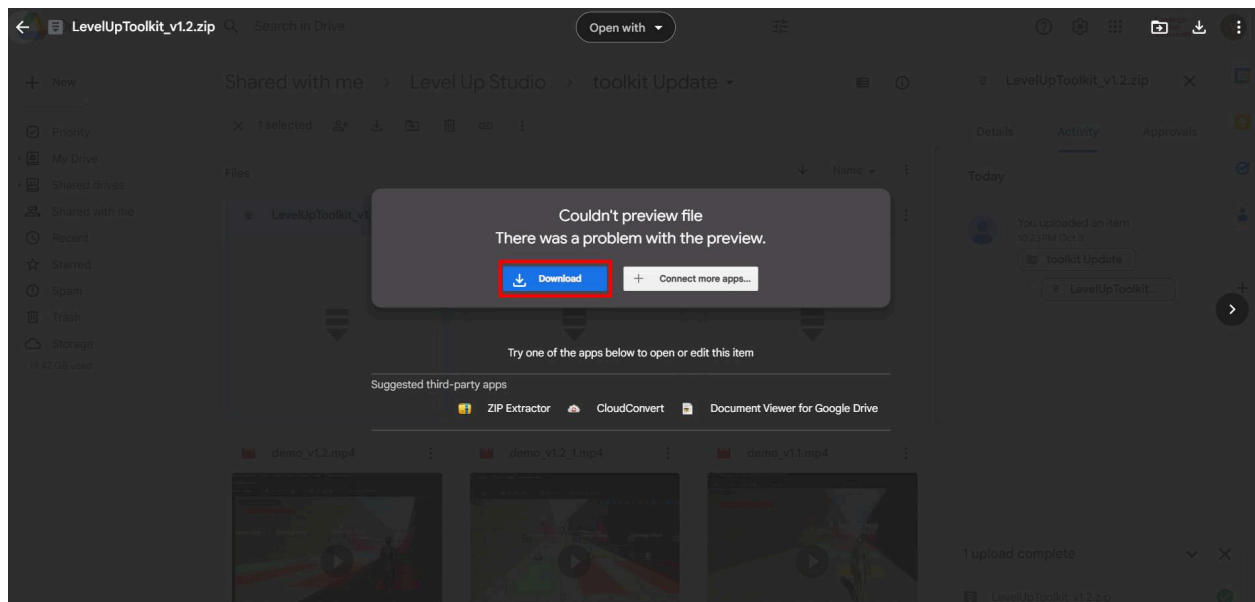
<https://www.unrealengine.com/en-US/download>

STEP 2: Download Level Up Toolkit

Click the link and download the latest version of level up toolkit plugin:

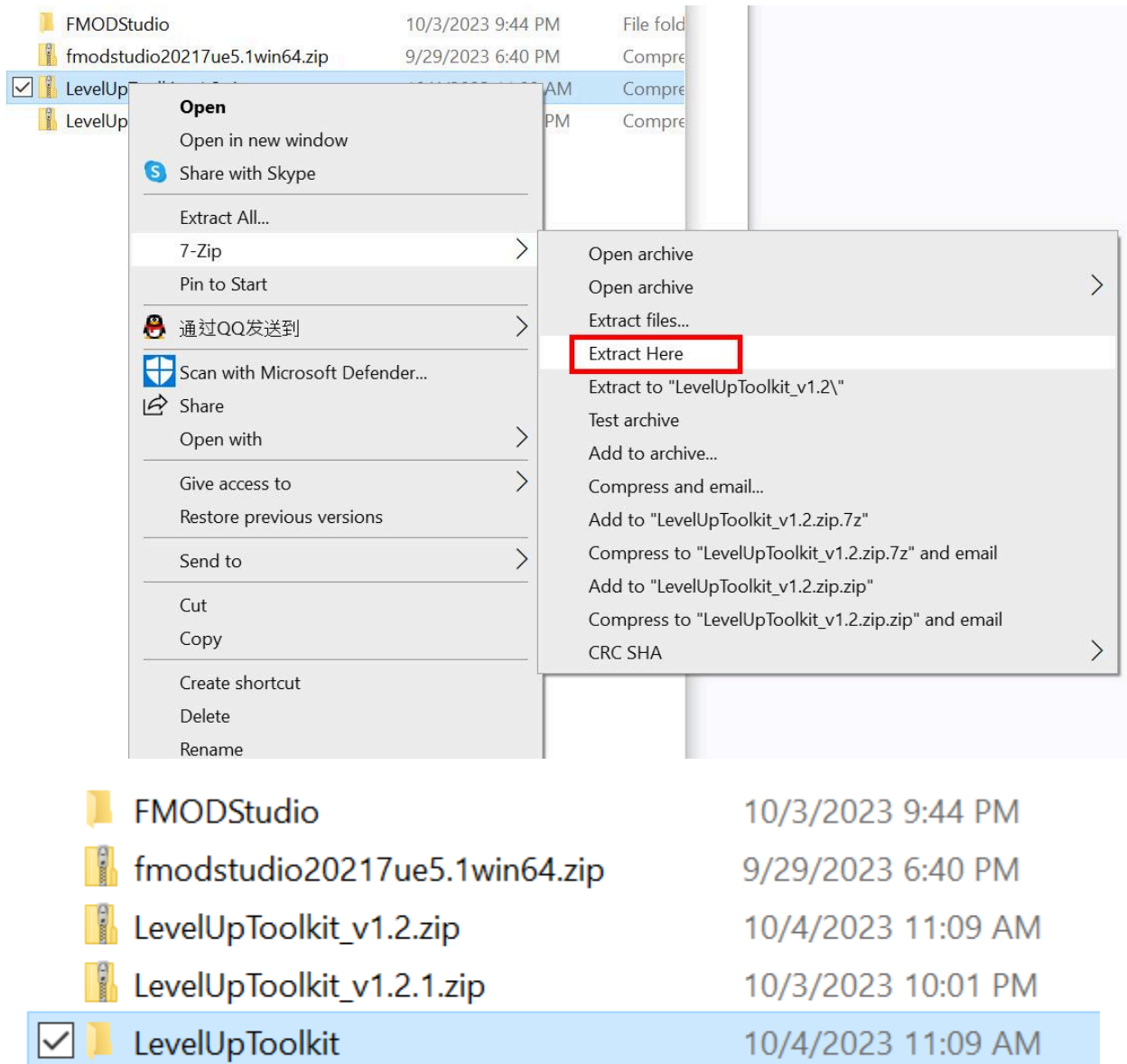
V1.6(latest)

https://drive.google.com/file/d/1HT2wFBbR6EFRTzw8D4UqsEaWP1xMmXR/view?usp=drive_link

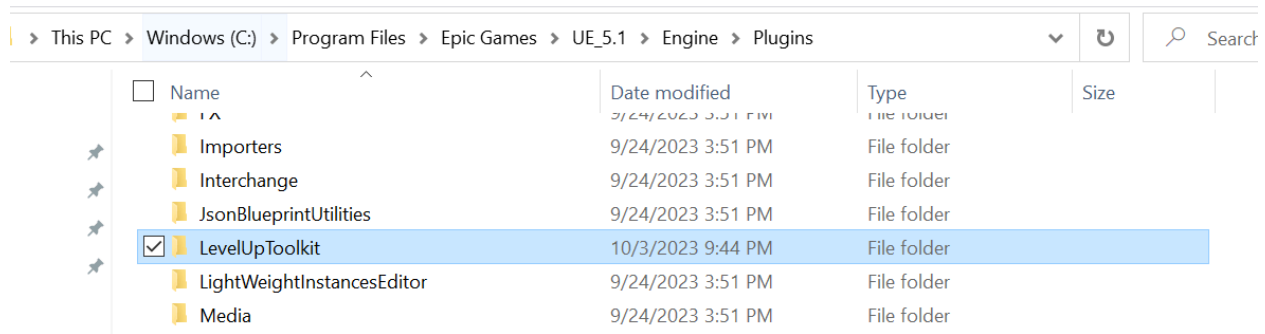


STEP 3: Install

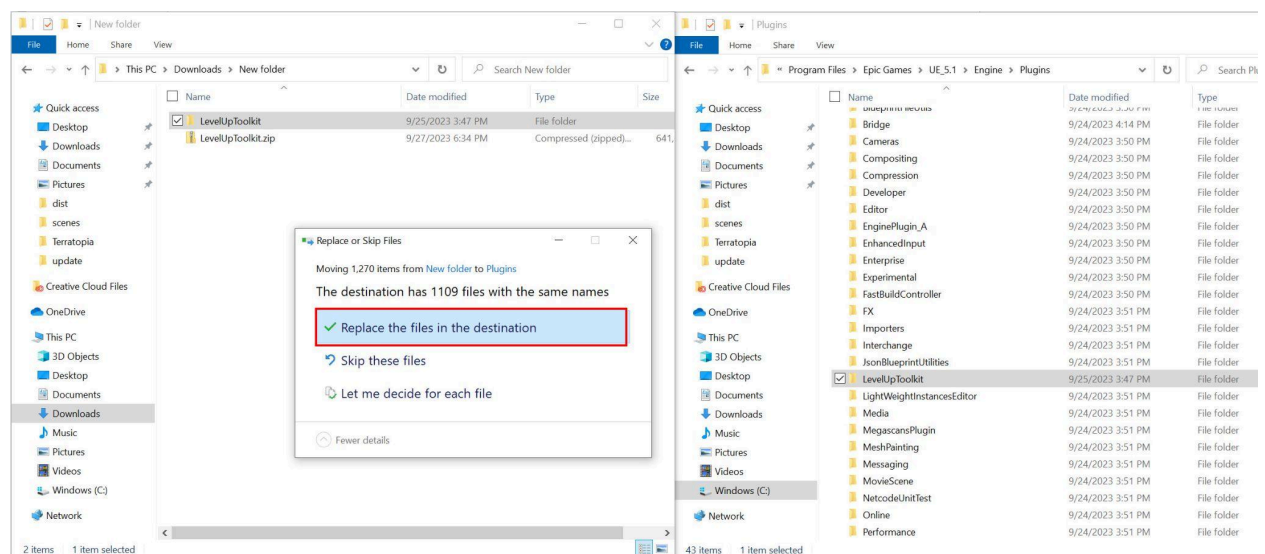
1. After downloading the zip file of our toolkit, extract the zip file. Then you will get a folder named 'LevelUpToolkit'.



2. Drag and drop the 'LevelUpToolkit' folder to your Unreal Engine plugin installation root directory. (Default directory is C:\Program Files\Epic Games\UE_5.1\Engine\Plugins)

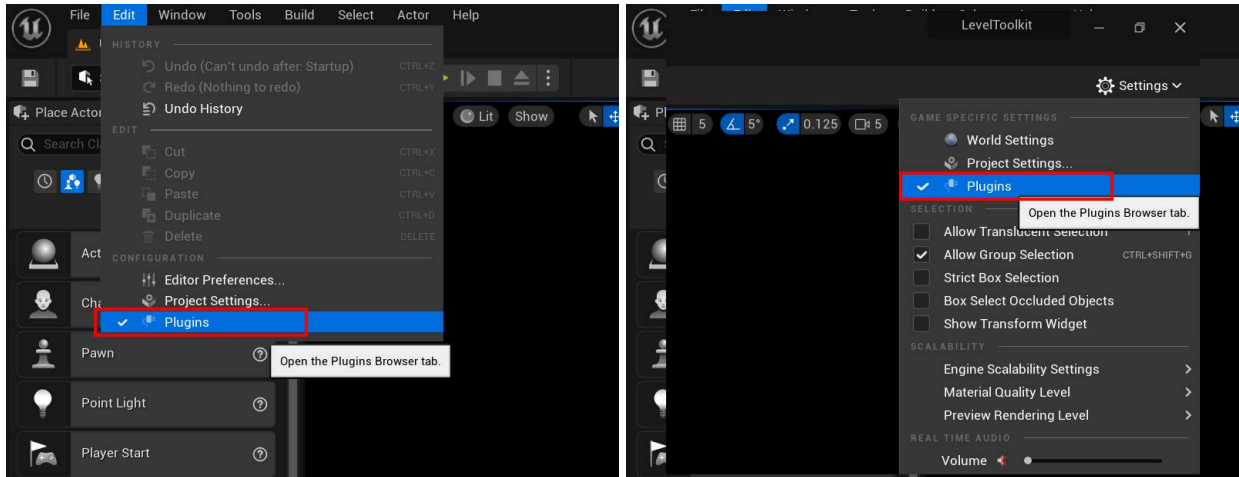


3.If there is already an old version toolkit in the plugin folder, just click the replace button when paste the new version to the plugin folder.

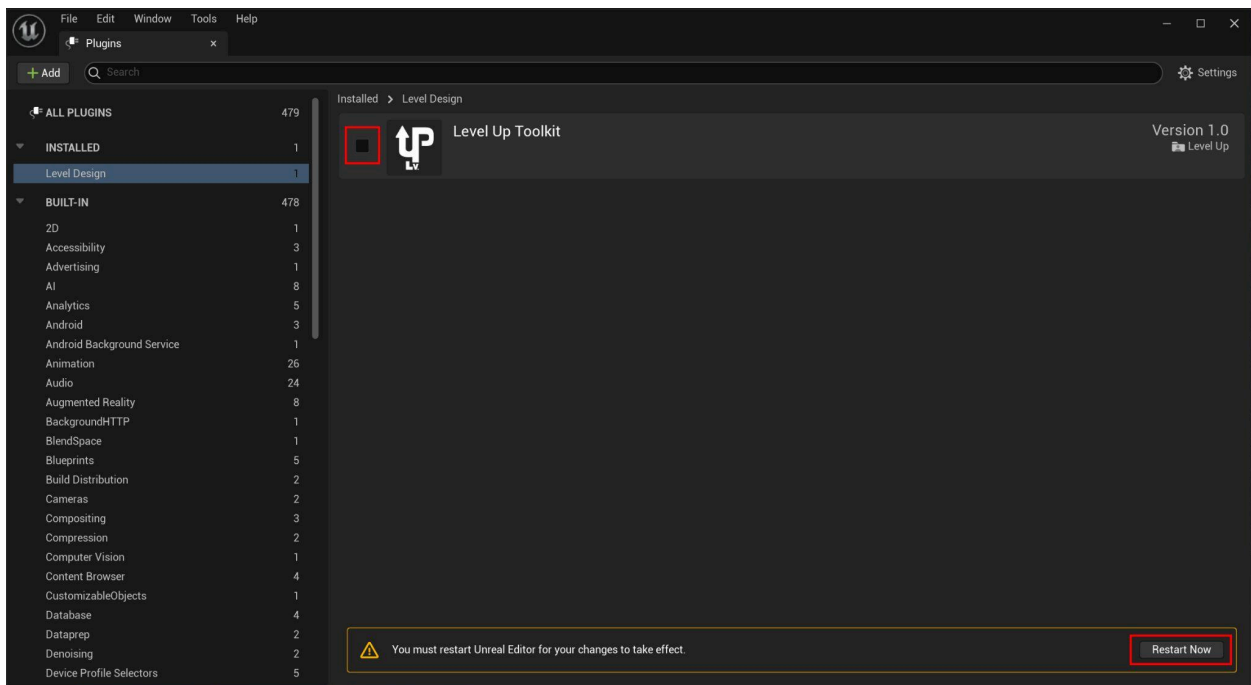


4. Enable the Level Up Toolkit

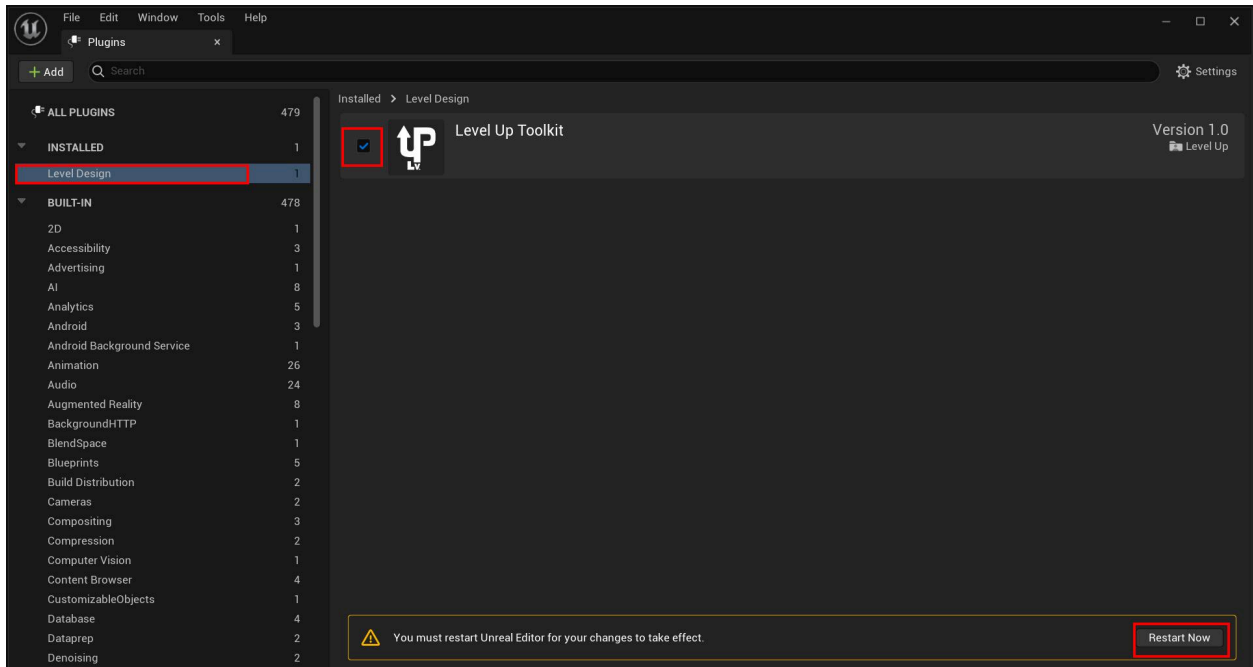
1. Click on Settings on the upper right or Edit on the upper left, then choose Plugins.



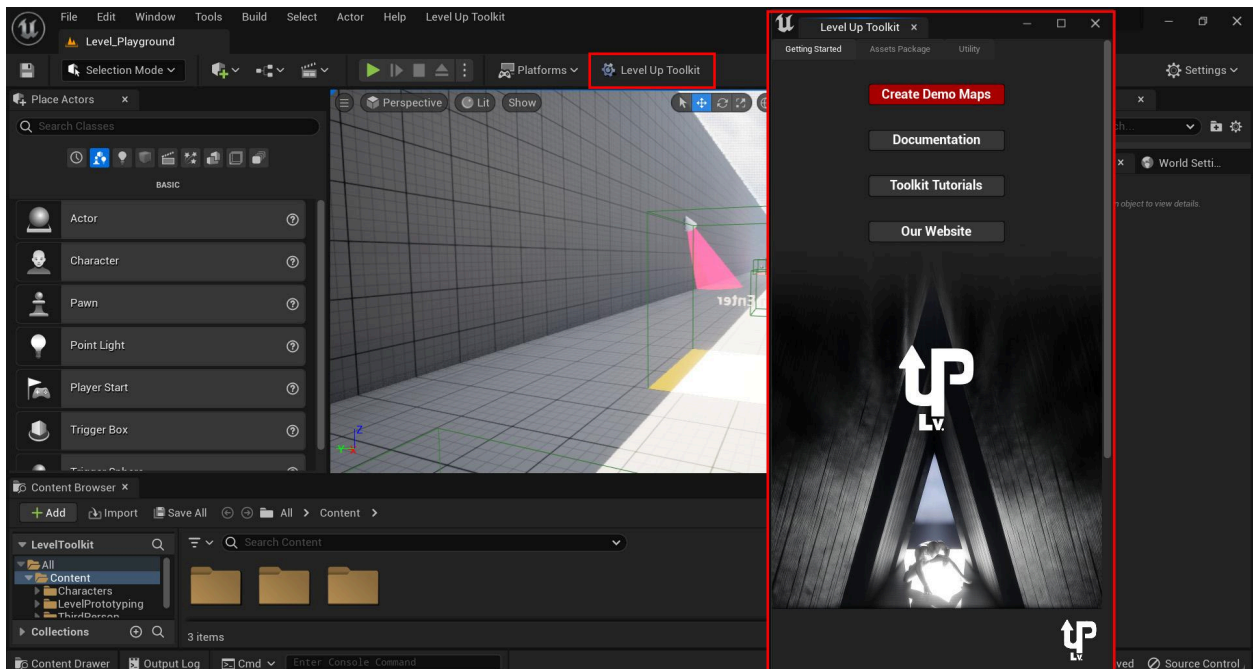
2. Under Installed, select Level Design. **(If you have already enabled the level up toolkit plugin,) first uncheck it and restart Unreal.**



3. Select the checkbox to the left of the toolkit, and restart Unreal when requested.

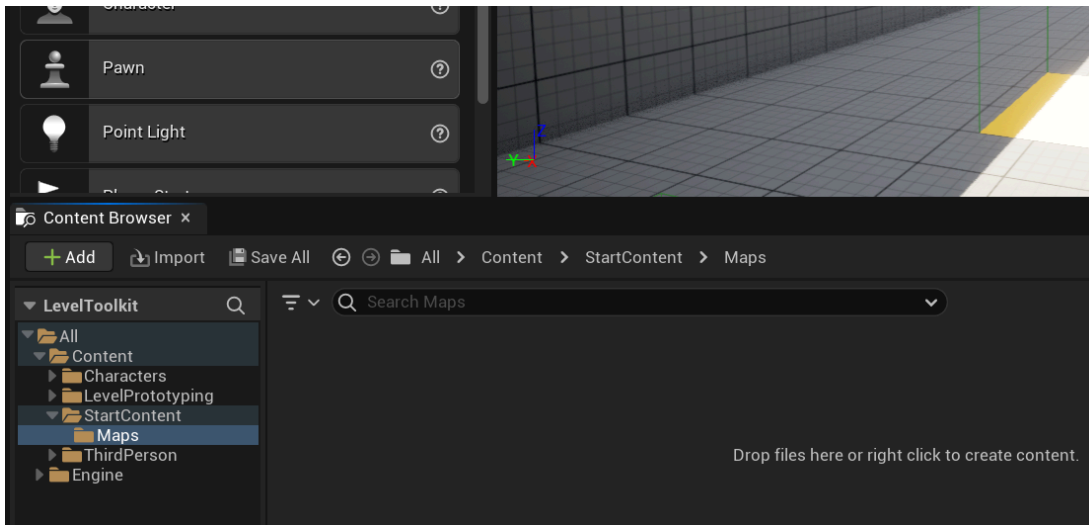


4. If you could see the level up toolkit button on the top, that means you installed successfully. You can click the button and use it in the level up toolkit panel.

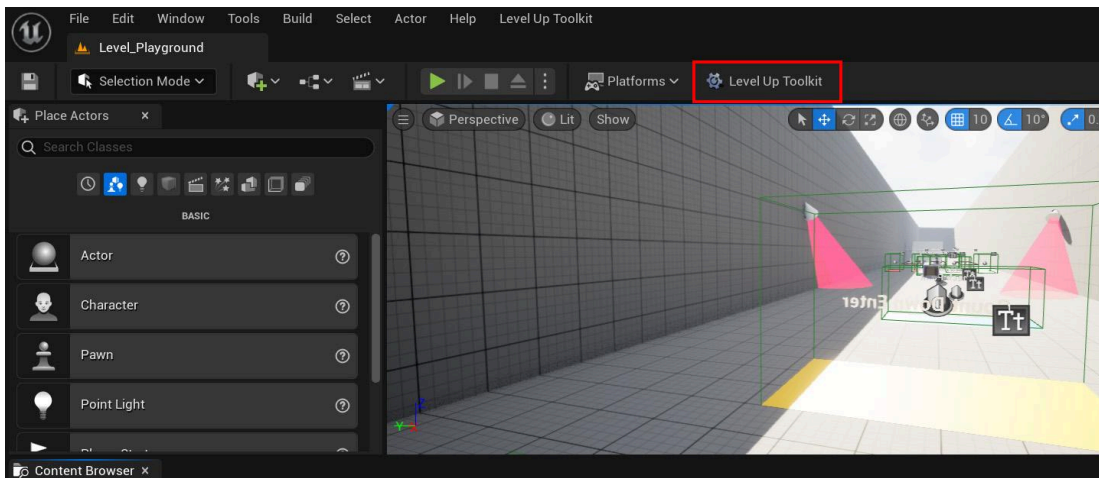


5. (optional) Create demo maps for player ability reference

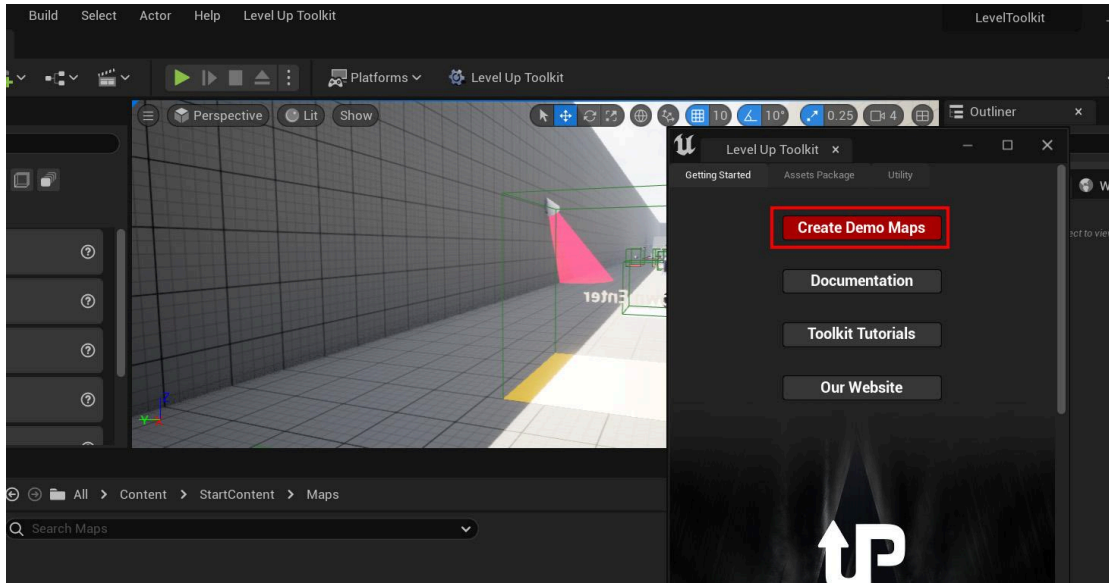
1. Open your project where LevelUp Toolkit is installed and enabled (by following the steps above)
2. Navigate in the Content Drawer to the location you want to put the Level Up demo maps, and select that folder in the Drawer by single clicking (for instance, All/Content/StarterContent/Maps)



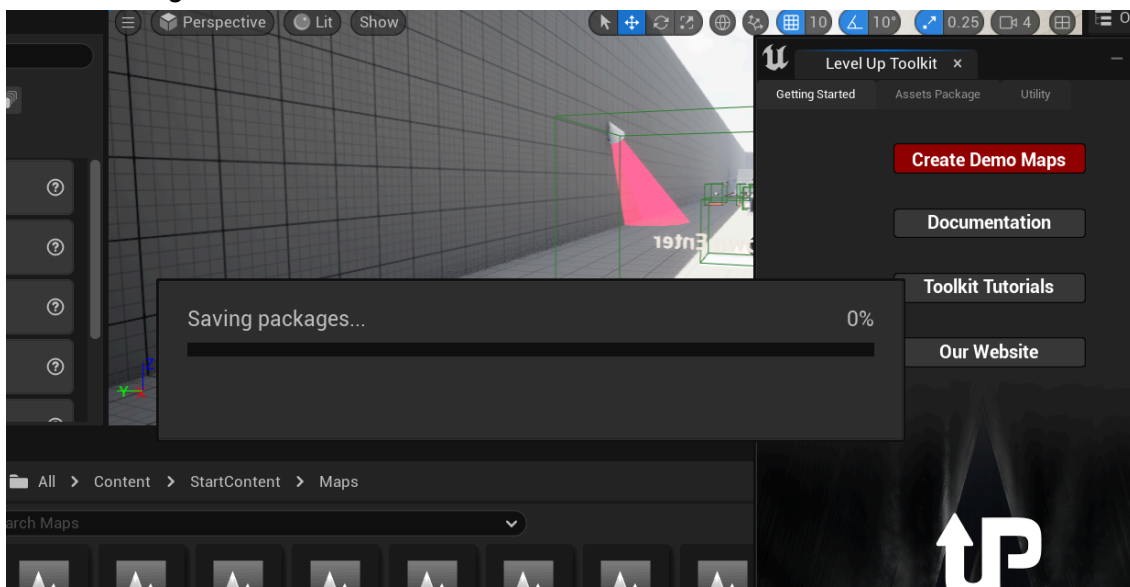
3. Click on Level Up Toolkit button in the header bar to bring up the menu



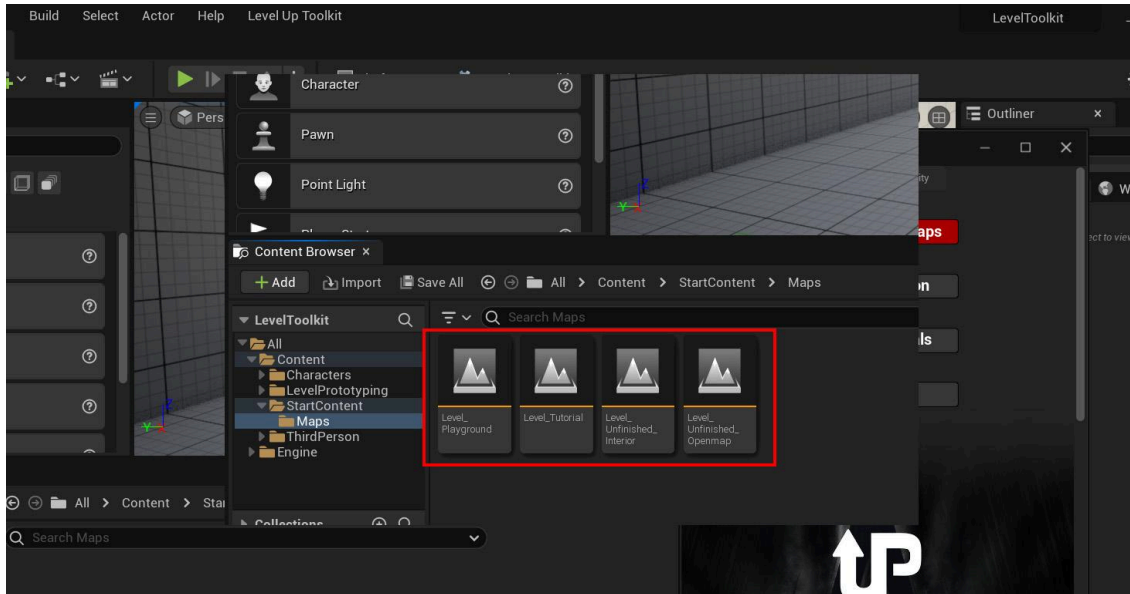
4. Click Create Demo Maps button in Level Up Toolkit menu



You should see some installations and file saving progress bars. If you don't, try later to click the button again.



After installation the demo maps will be located in the folder you selected in step 2.



Recommended Resources for Learning UE5

Unreal Engine Official Learning Resource

<https://dev.epicgames.com/community/unreal-engine/learning>

Your First Game in UE5 | Tech Talk | State of Unreal 2022

https://www.youtube.com/watch?v=Itd677YZi50&list=PLZlv_N0_O1gZmDqpSAEXEmATpv-HgQJYp&index=6

Recommend Youtube Channel

<https://www.youtube.com/@UnrealEngine>



Unreal Engine

@UnrealEngine 959K subscribers 2.2K videos

Welcome to the official Unreal Engine channel! >

<https://www.youtube.com/@UnrealSensei>



Unreal Sensei ✓

@UnrealSensei 354K subscribers 20 videos

Unreal Sensei creates content about Unreal Engine 5 that includes tutorial... >

UE5 Naming Convention (Style) Guide

<https://github.com/Allar/ue5-style-guide>

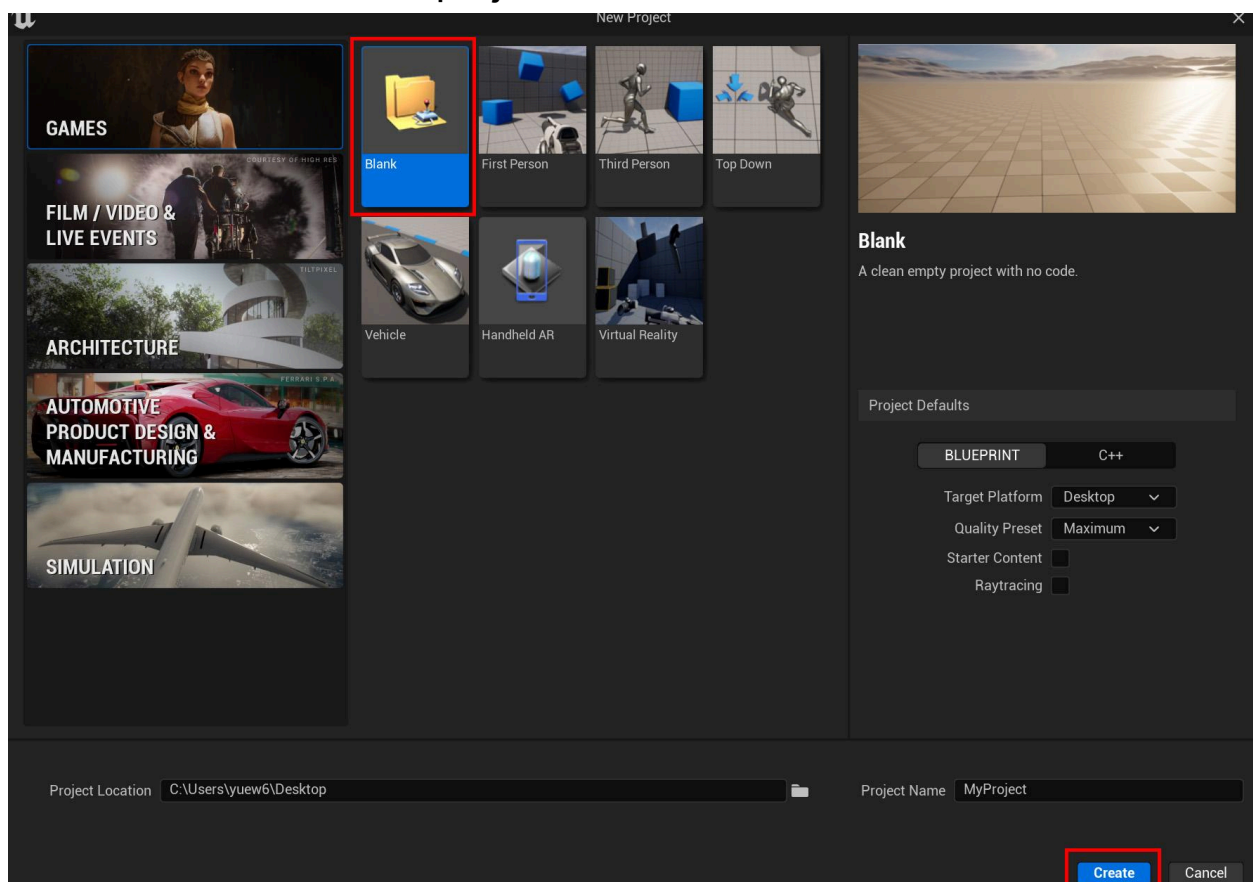
UE5 Hotkeys

https://docs.google.com/document/d/1OGqWogjhtuudNUVN3JjkTOvFRzPwNs_phn7RykhkHhs

The Basics You Need To Know

1. How to open the project?

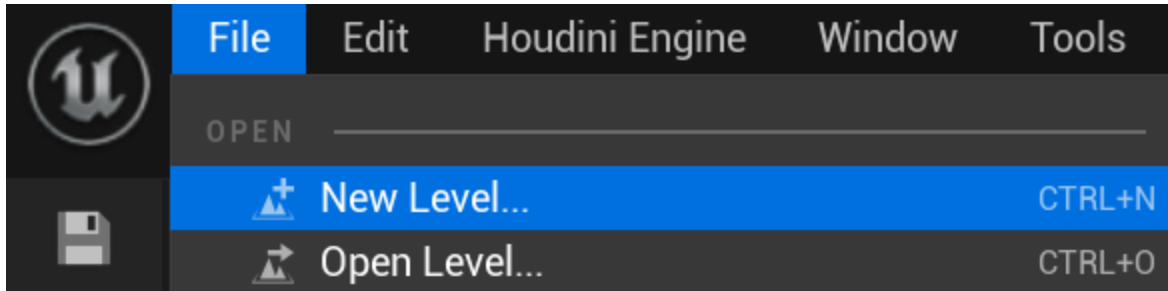
Install the UE 5.1 version, type your project name, then create a new project.



through the Unreal Engine Launcher or double click an existing .uproject file.

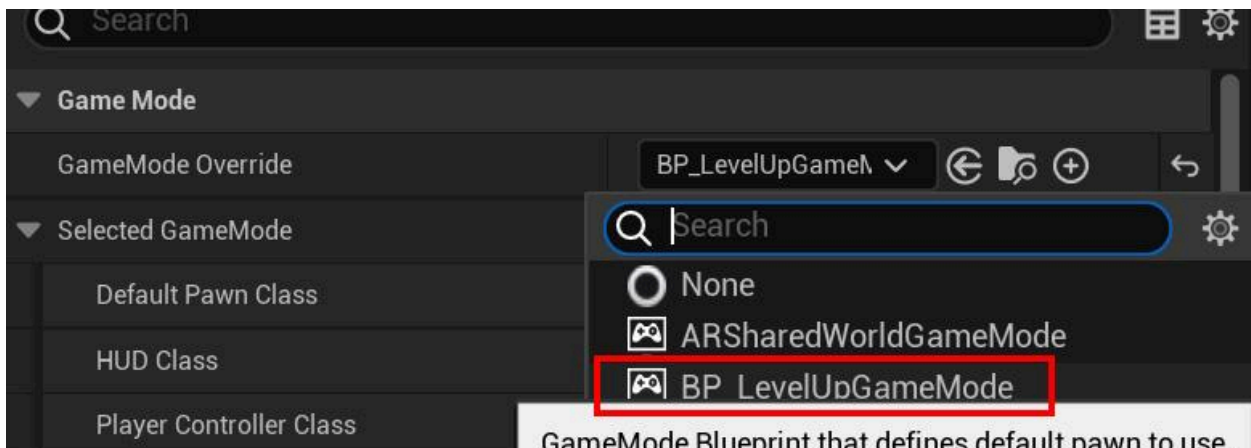
2. How to create a new level?

Go to File->New Level...->Open World/Basic/Empty Level.

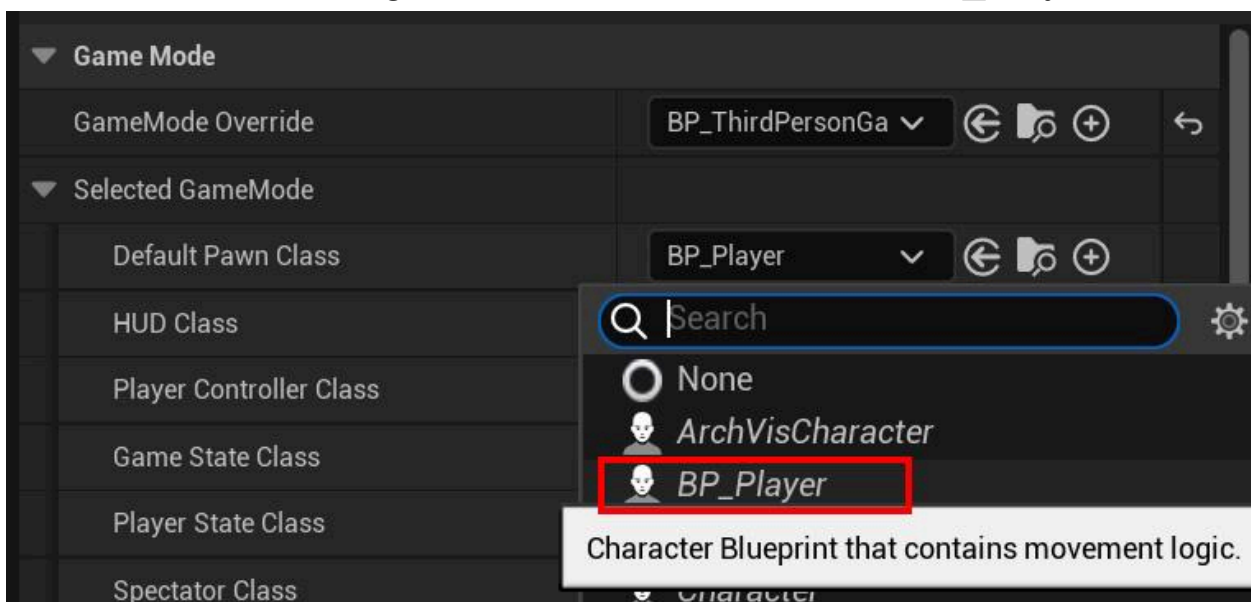


3. How to change Game Mode?

Click the Settings button on the top right corner, then click World Settings, change the GameMode Override to BP_LevelUpGameMode.

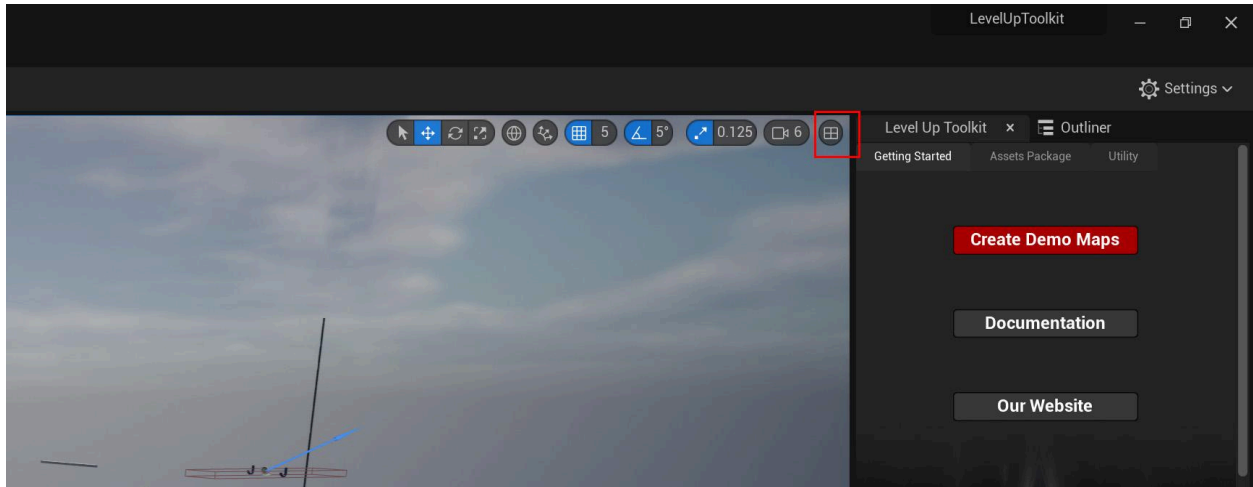


Then change the Default Pawn Class to BP_Player.

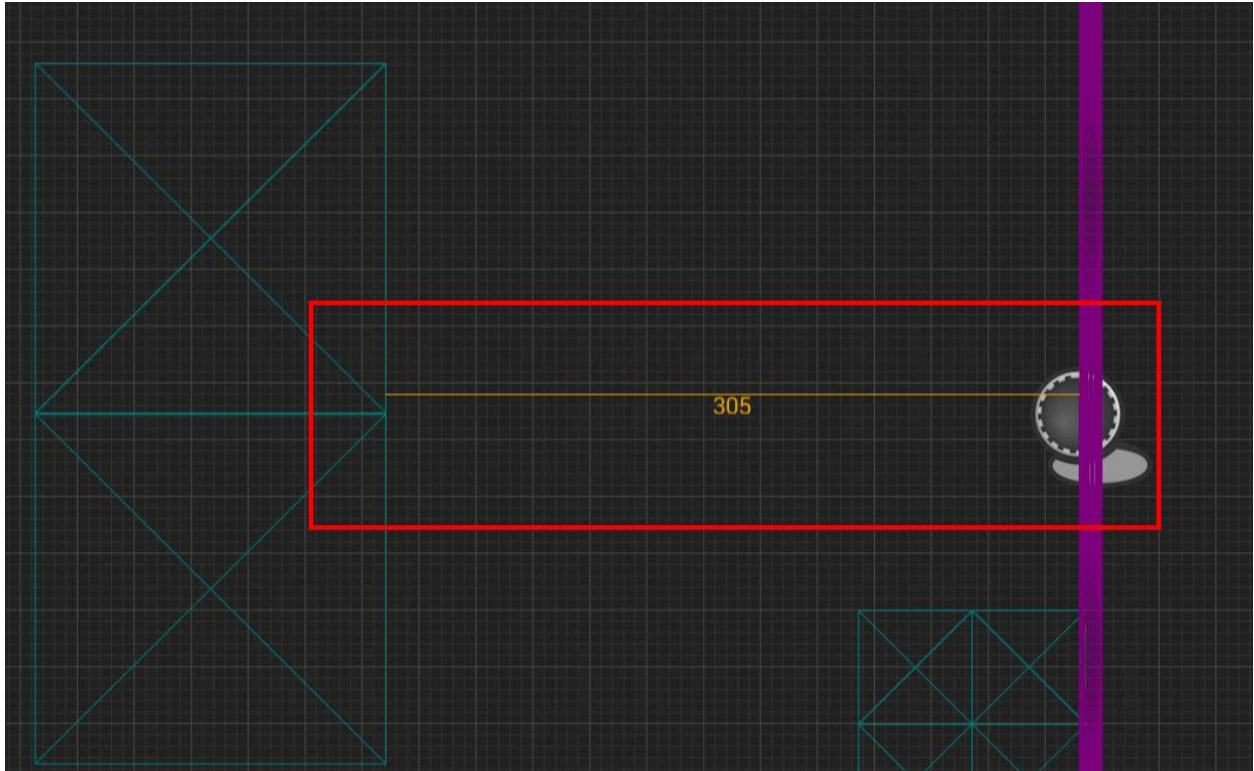


4. How to Measure the distance?

Click the square button on the top right corner of the view port.



In the Orthographic viewport, Hold middle mouse button on the start point you want to measure, and then move the mouse to the end point. When you hold the mouse, there will be a number to show the distance between the start point and the end point. The unit is centimeter.

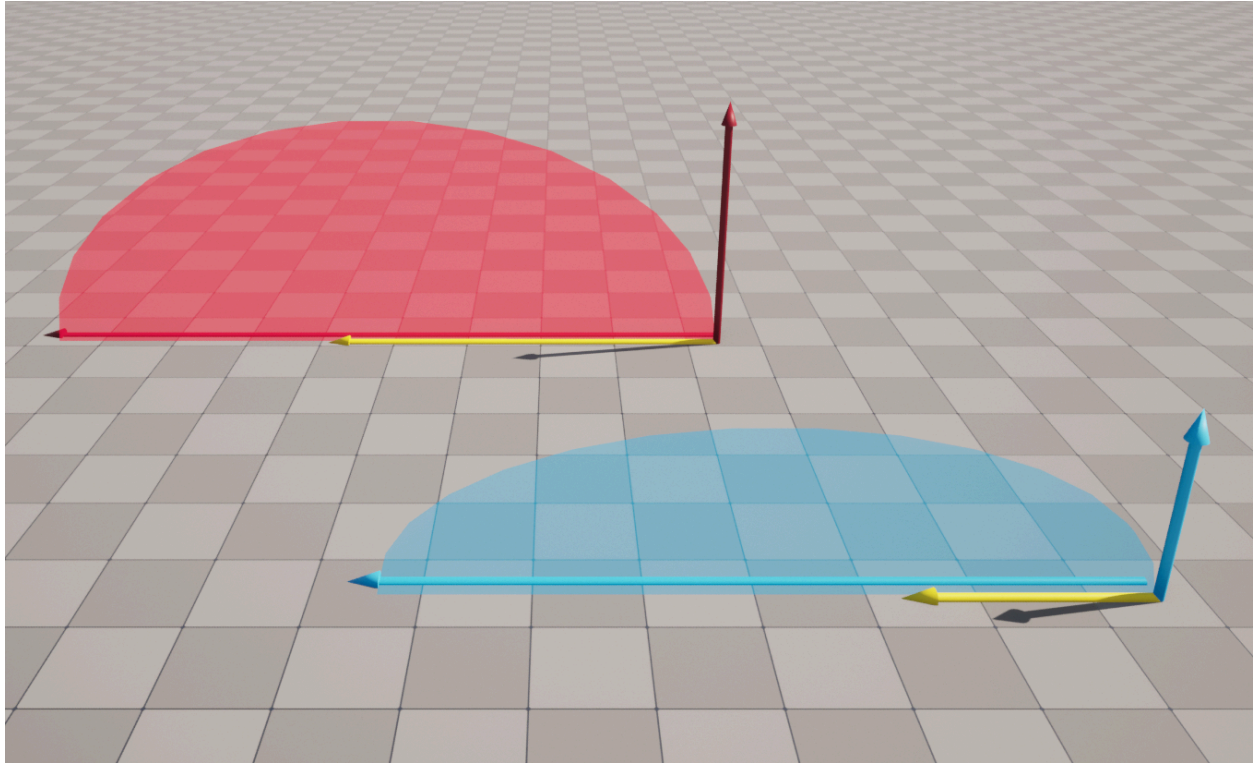


Click the top right square button again to hide other viewports.

Tools for Editor

Jump Range Visualization

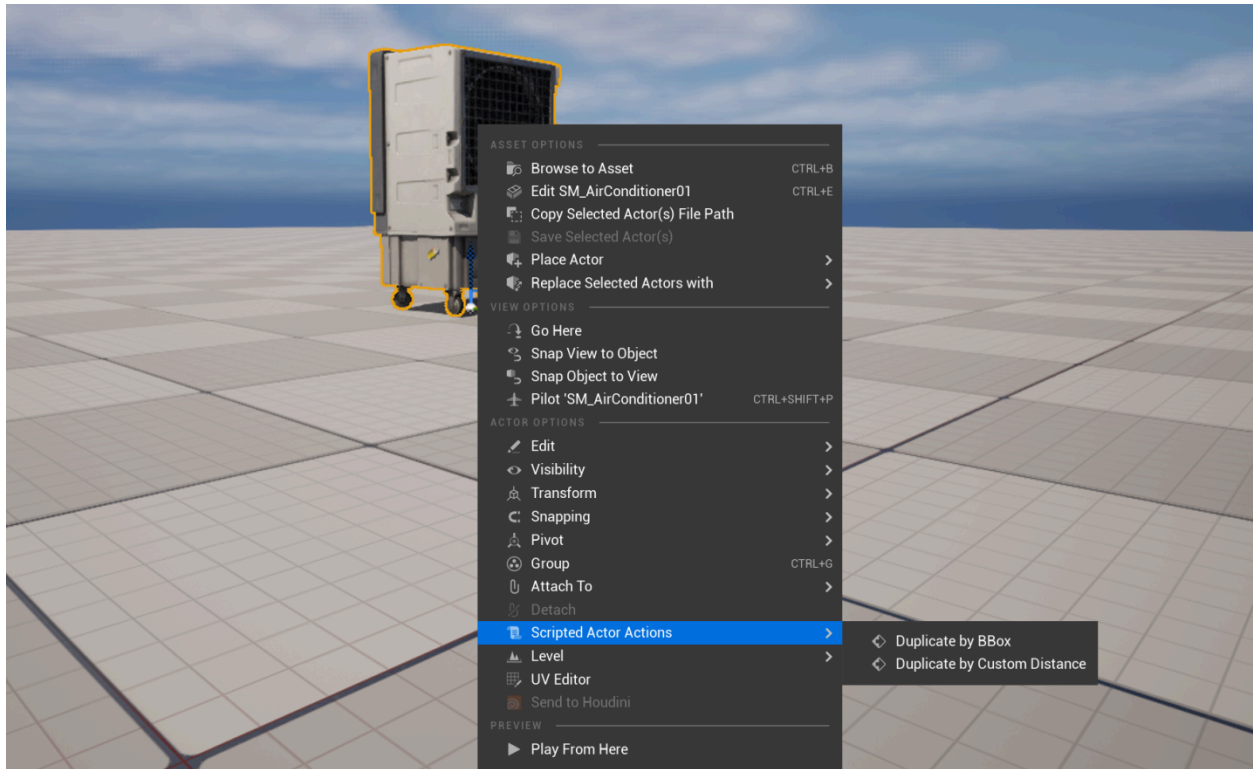
Our character has two jumping modes, normal jump and pole-vault. In order to visualize the maximum distance and maximum height of the two jumps, we made two models for you to use as a reference to design levels. The transparent arc can be understood as the player's jumping arc. Note that depending on whether the player is running or not, the jumping distance can be different.



Basically, the blue arc shows the furthest the player can jump with a normal jump, and the red arc shows the furthest the player can jump with a pole-vault.

Axial replication

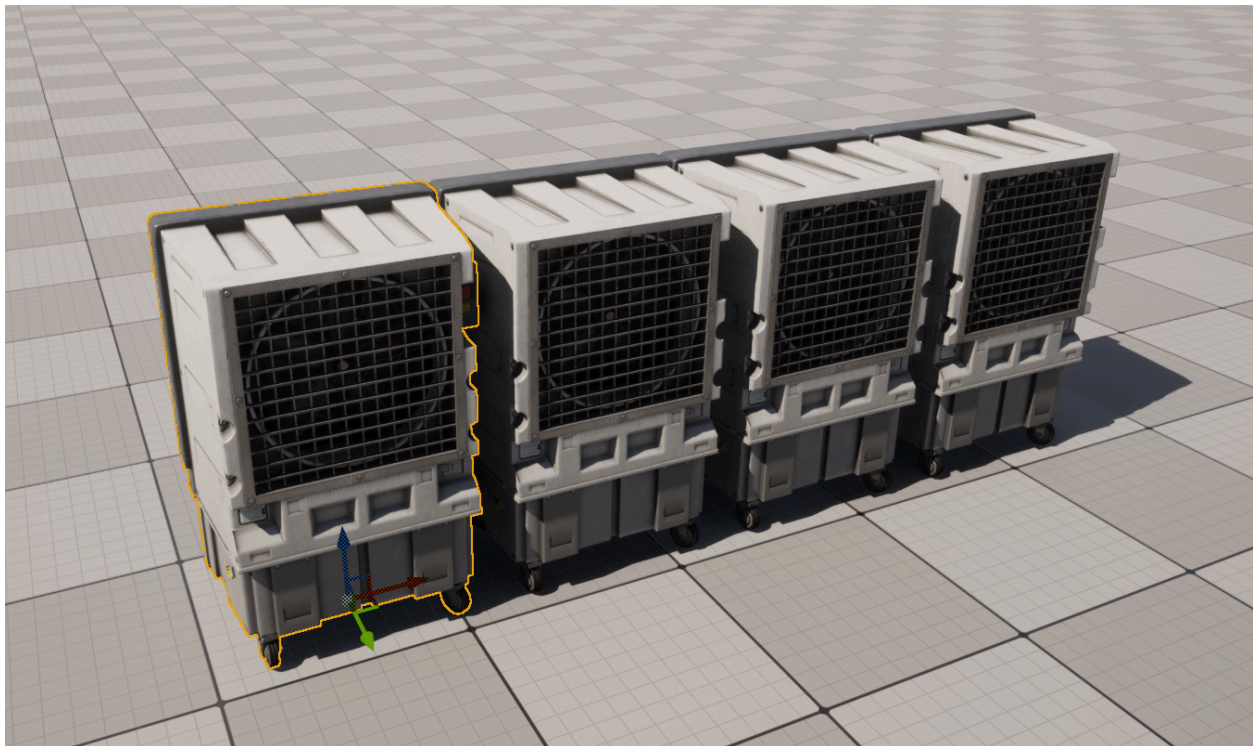
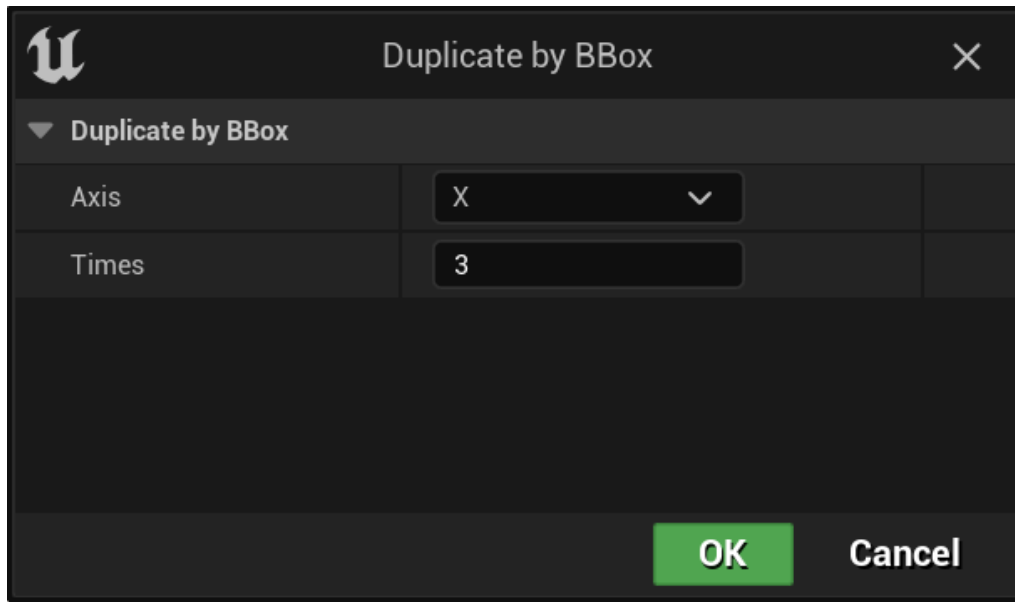
If you want to duplicate the actors by the same distance in the scene, you can select the actor and right click, choose Scripted Actor Actions, we provide you two options: Duplicate by BBox and Duplicate by Custom Distance.



PS: If you want to copy along a non-axis direction, you can copy along the axis first, then select all objects and rotate.

Duplicate by Bounding Box

Duplicate by Bounding Box will be based on the global axis X, Y, or Z and detect the actor's bounding box, this value will be set to the duplicate distance, you can change the duplicate times.



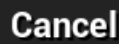
Duplicate by Custom Distance

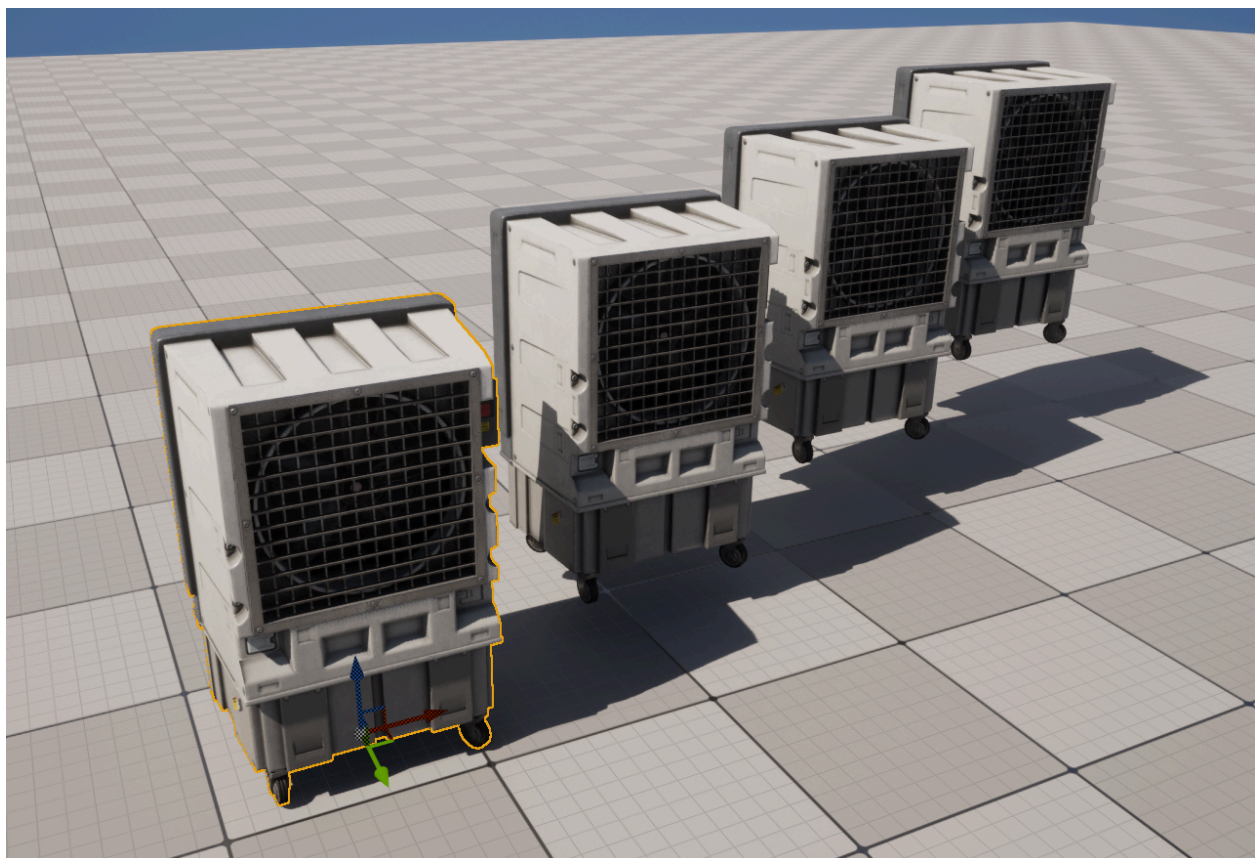
Duplicate by Custom Distance allow you to manually enter values.

 Duplicate by Custom Distance 

▼ Duplicate by Custom Distance

X	150.0	
Y	-50.0	
Z	20.0	
Times	3	

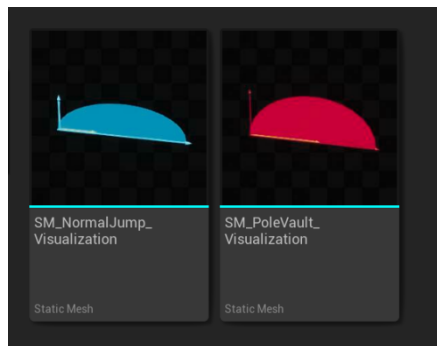


Resources in Toolkits

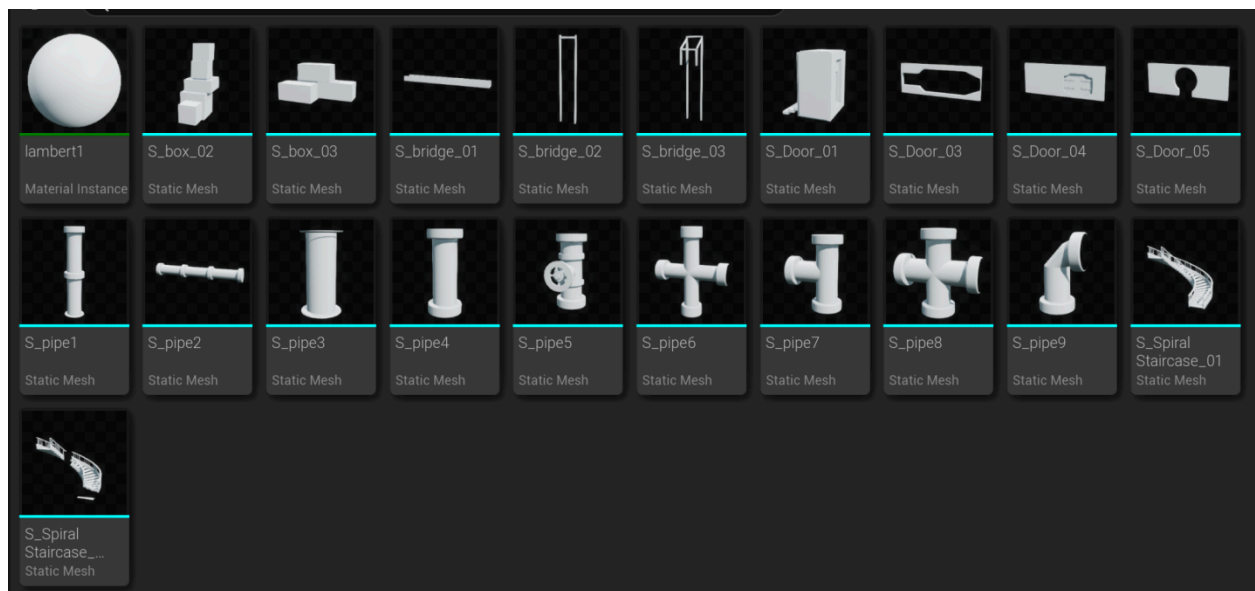
Models

Path: Content>Model

Measurement Visualization Features



Visual Guidance Features

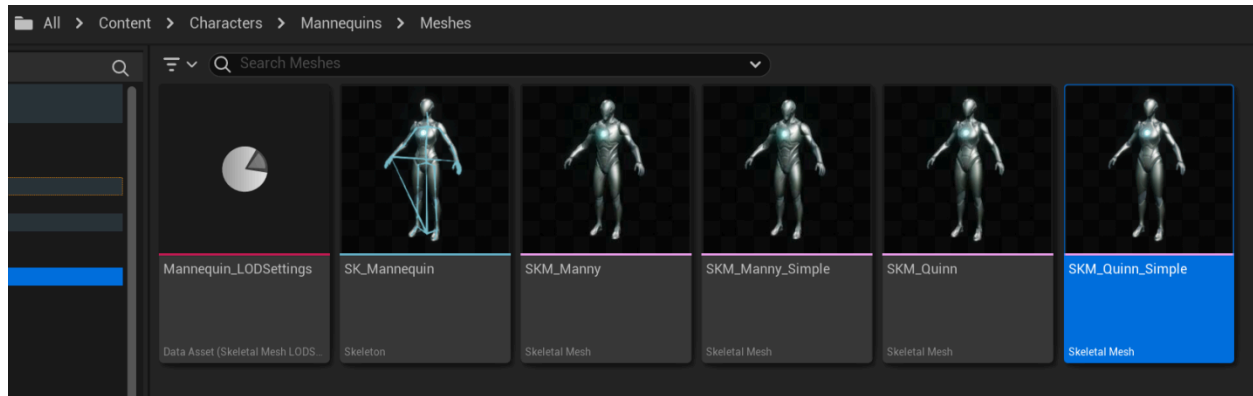


Decorative Features



Animation

The skeletal mesh of the player character is in the following path:
Content> Characters> Mannequins> Meshes.



The animation includes:

Animation	Speed/Height/Length	Description
Run	800cm/s	Normal movement.
Normal jump	Length: Min:2m, Max:7m Height: Max:1.6m (Based on feet)	Normal jump.
Sprint	300cm	The character can sprint.
Slide	500cm	The character can slide.
Fall		The character falls from a higher place.
Land		The character lands on the ground.
Climb Edge	Max height: Normal Jump: 3.3m Vault Jump: 5.35m	When a player falls on the edge of a specific wall, the character can climb the edge automatically.
Crouch	300cm/s	The character can crouch and move.
Grab		Grab pole.
Vault Jump	Length: Min: 5m, Max: 9m Height: Max: 3m (Based on feet)	After the player gets the pole, the character can do a vault jump, which can let the character jump higher and further.
Climb Wall		After the player gets the pole, the character can climb the wall by the pole.
Hang		When the player gets the pole, the character can use the pole to hang on the hooks.
Slide Down		When the player touches the specific rope or pole, the character will slide down the rope or

		pole.
Hit		After the player gets the pole, the character can use the pole to hit buttons.

Levels

In the toolkit, there are some finished levels and unfinished levels. The users can try the finished levels, or try to finish the unfinished levels, and also users can create their own levels.

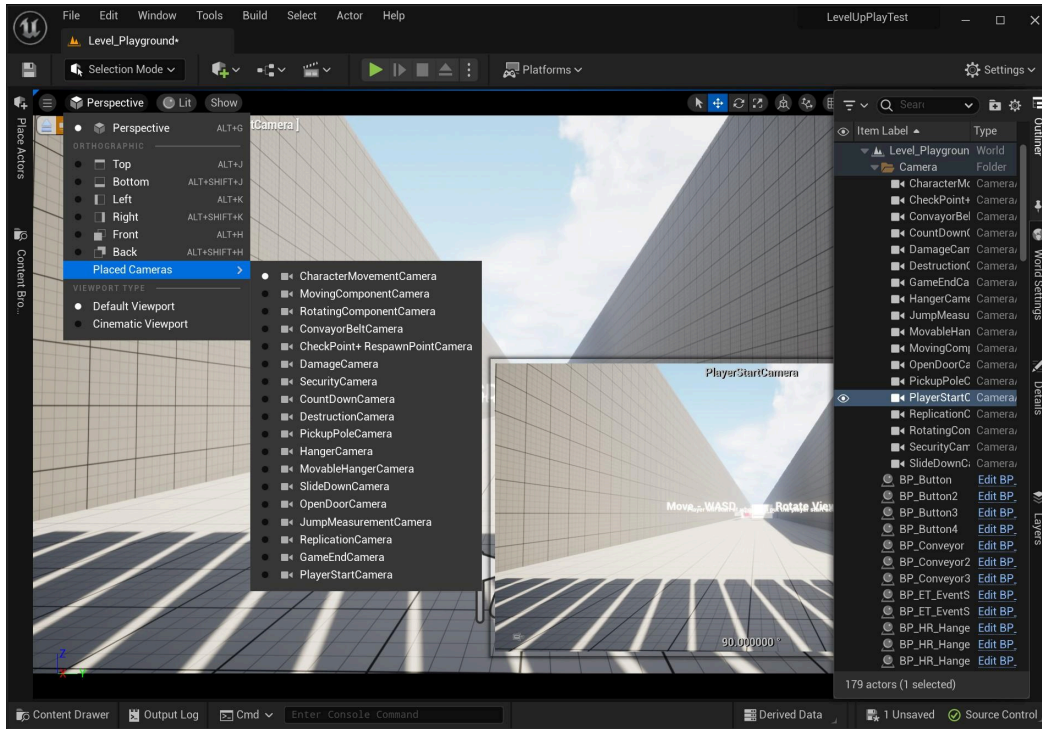
The levels are in the following path: Level Up Toolkit
Content>Maps

The levels include:

Finished Level	James_GameJam_2_23
Unfinished Level	Level_Unfinished_Interior
Playground	Level_Playground
	TestMap

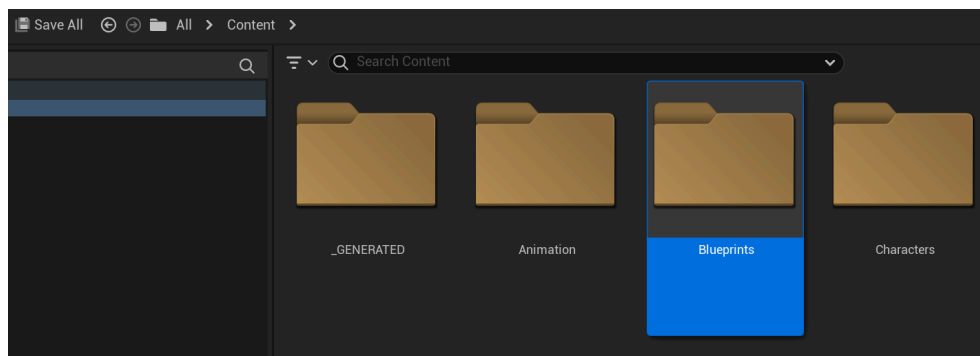
Playground Level

All toolkit functions are in Level_Playground. There are several cameras in this level. Use Perspective> Placed Cameras to jump to the specific camera view and see the corresponding function.



Making Interactive Experiences

The Interactable features are in the Blueprint Folder. The guides in this section will provide a reference for how to use these features and what's the result of these features.



Character Control

Name: BP_Player

Type: Character

Parent Class: Character

File Location:

Description: Control the character movement and interact with other objects.

Input:

1)Sprint:

Variable Name: Sprint Distance

Type: Float

Usage: DetermineDecide the distance of sprint.

2)Slide:

Variable Name: Slide Distance

Type: Float

Usage: Determine the distance of slide.

3)Crouch:

a. Variable Name:Max Walk Speed

Type: Float

Usage: Determine the maximum running speed.

b. Variable Name: Max Crouch Speed

Type: Float

Usage: Determine the maximum crouching speed.

4)Damage:

Variable Name: Spawn Pos Num Choice

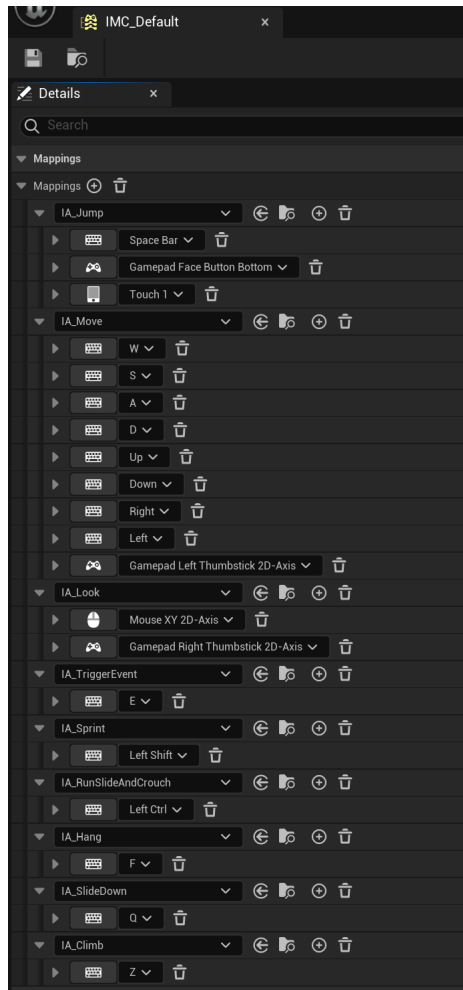
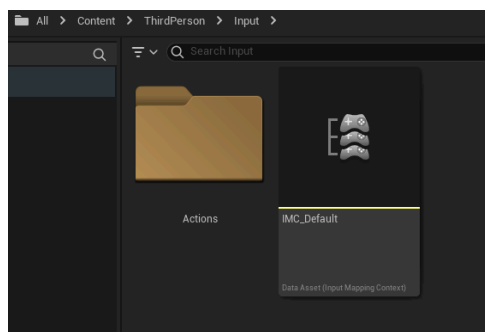
Type: Integer

Usage: Determine which respawn point to respawn.

Default Input:

Behavior	Key	Behavior	Key
Move	WASD	Interact	E
Normal jump	Click Space	Grab	E
Jump-Vault	Hold Space	Climb Edge	(Automatic)
Climb Wall	Space	Slide Down	(Automatic)
Sprint	Left Shift	Jump on the hanger	Space
Slide	Left Control (When moving)	Jump down from the hanger	F
Crouch	Left Control (When static)	Stow the Pole / Unstow	G
Die & Respawn	R	Hurt once	M
Extend the pole from one side	T	Attack	Left Mouse Button

Users can change the input by IMC_Default:



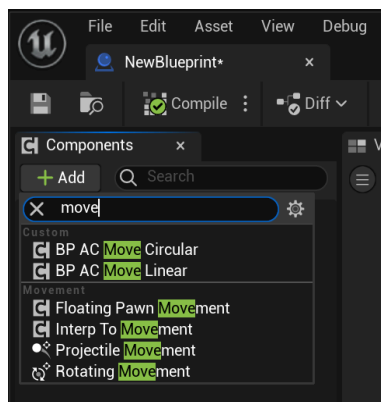
Moving Component

Name: BP_AC_MoveLinear

Type: Actor Component

Parent Class: Actor Component

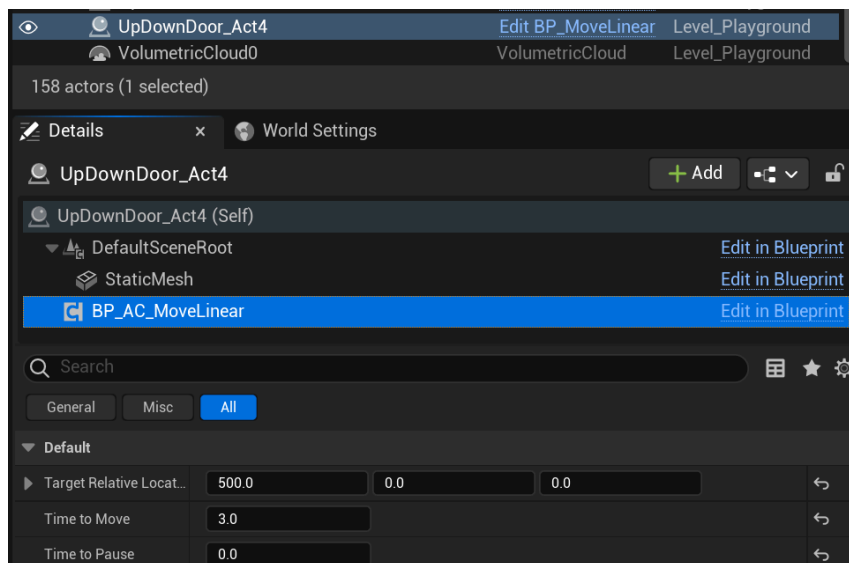
File Location: You can add it to any actor blueprint



Description: Move linear from init location to target location.

Input:

- 1): Target Relative Location
Type: Vector
Usage: The offset between init location and target location.
- 2): Time To Move
Type: Float
Usage: Time to move in each move.
- 3): Time To Pause
Type: Float
Usage: Time to pause after each move.



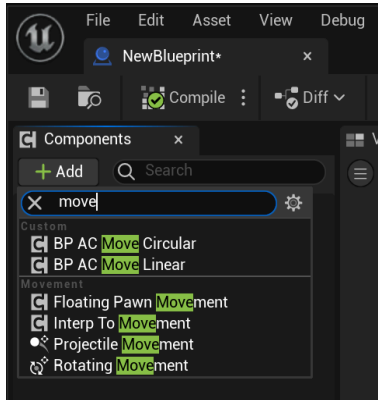
Rotating Component

Name: BP_AC_MoveCircular

Type: Actor Component

Parent Class: Actor Component

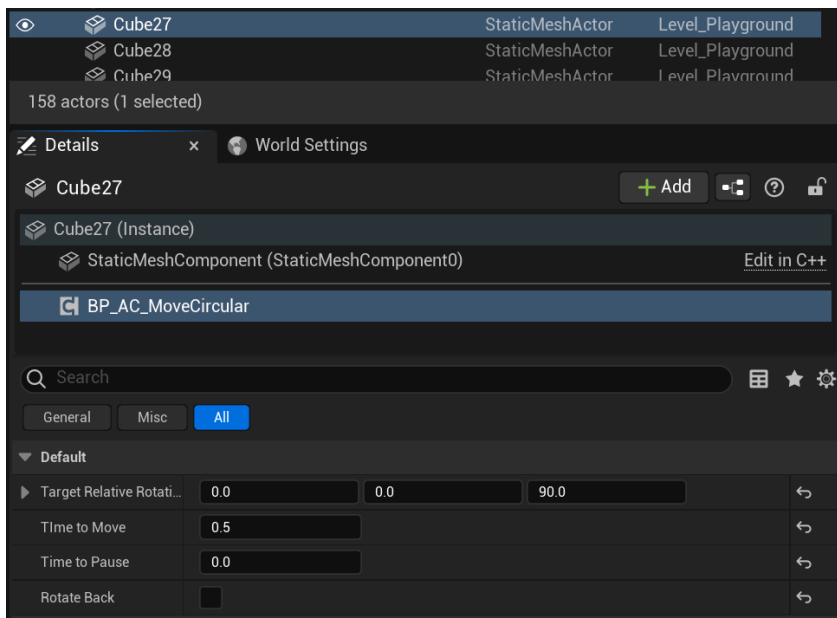
File Location: You can add it to any actor blueprint



Description: Move circular from init rotation to target rotation, keep rotating or rotate back.

Input:

- 1): Target Relative Rotation
Type: Vector
Usage: The offset between init rotation and target rotation.
- 2): Time To Move
Type: Float
Usage: Time to move in each move.
- 3): Time To Pause
Type: Float
Usage: Time to pause after each move.



Event Trigger Interface

Name: BP_ET_EventTrigger

Type: Actor

Parent Class: Actor

File Location:

Description: Base Interface of the event trigger. It is inherited by other classes.

Self Event Trigger

Name: BP_ET_SelfTrigger

Type: Actor

Parent Class: BP_ET_EventTrigger

File Location:

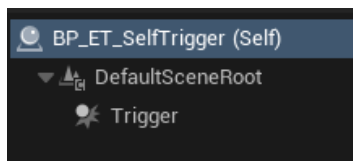
Description: An event trigger that can detect and call an event by itself. It is an interface that needs to be inherited by other classes to use.

Input:

1): Trigger

Type: Sphere Collision

Usage: When the avatar enters the trigger, the event calls.



2): Need Input

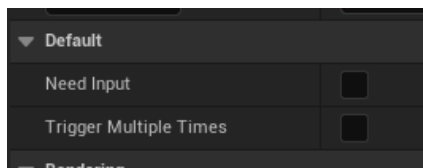
Type: Boolean

Usage: If true, the player needs to make an input action to start the event.

3): Trigger Multiple Times

Type: Boolean

Usage: If true, the event can be triggered multiple times.



Pickup Event Trigger

Name: BP_ET_PickupTrigger

Type: Actor

Parent Class: BP_ET_SelfTrigger

File Location:

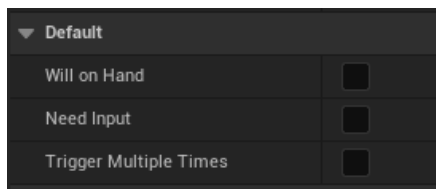
Description: Event Trigger for items that can be picked up by the avatar.

Input:

1): Will on hand:

Type: Boolean

Usage: If true, the actor will be on the avatar's hand after triggered.



Event Sender Component

Name: BP_ETC_EventSender

Type: Scene Component

Parent Class: Scene Component

File Location:

Description: A Component for calling the event and sending it to the receiver. This component is embedded in BP_ET_EventSender.

Input:

1): Need Input

Type: Boolean

Usage: If true, the player needs to make an input action to start the event.

2): Trigger Multiple Times

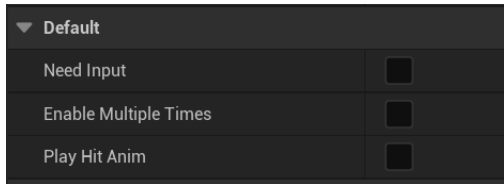
Type: Boolean

Usage: If true, the event can be triggered multiple times.

3): Play Hit Anim:

Type: Boolean

Usage: If true, the avatar plays the hit animation while triggering the event.



Event Receiver Component

Name: BP_ETC_EventReceiver

Type: Scene Component

Parent Class: Scene Component

File Location:

Description: A Component for receiving the event and calling the callback. This component is embedded in the BP_ET_EventReceiver.

Input:

1): Sender Ref:

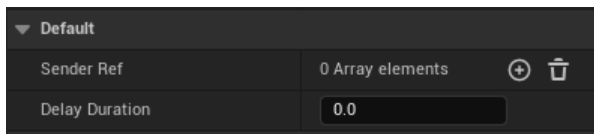
Type: Array of BP_ET_EventSender instances

Usage: The callback is executed if receive all senders' signal.

2): Delay Duration:

Type: Float

Usage: The delay time before the callback function is executed.



Event Sender Trigger

Name: BP_ET_EventSender

Type: Actor

Parent Class: Actor

File Location:

Description: Event Trigger for items that can call an event. This class should be inherited by other classes to become real assets like buttons.

Input:

1): Trigger

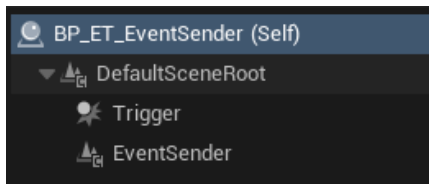
Type: Sphere Collision

Usage: When the avatar enters the trigger, the event calls.

2): Event Sender Component

Type: BP_ETC_EventSender

Usage: See “Event Sender Component.”



Event Receiver Trigger

Name: BP_ET_EventReceiver

Type: Actor

Parent Class: BP_ET_EventTrigger

File Location:

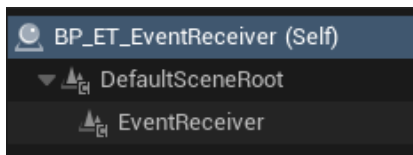
Description: Event Trigger for items that can receive an event and do callback. This class should be inherited by other classes to become real assets like doors.

Input:

1): Event Receiver Component

Type: BP_ETC_EventReceiver

Usage: See “Event Receiver Component.”



Transformational Event Receiver Trigger

Name: BP_ET_TransformReceiver

Type: Actor

Parent Class: BP_ET_EventReceiver

File Location:

Description: Event Trigger for items that can receive an event and do transform as a callback. Transform includes translation, rotation, and scale. This class should be inherited by other classes to become real assets.

Input:

1): Event Receiver Component

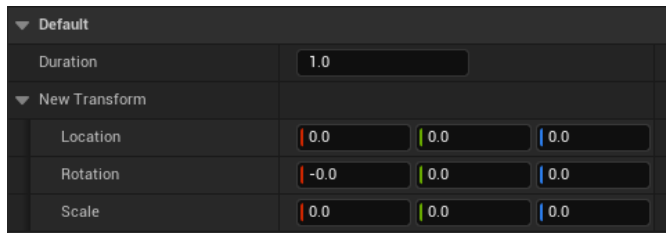
Type: BP_ETC_EventReceiver
Usage: See “Event Receiver Component.”

2): Duration

Type: Float
Usage: Transformation’s duration.

3): New Transform

Type: Transform
Usage: The delta transform applies to the actor.



Pole

Name: BP_Pole

Type: Actor

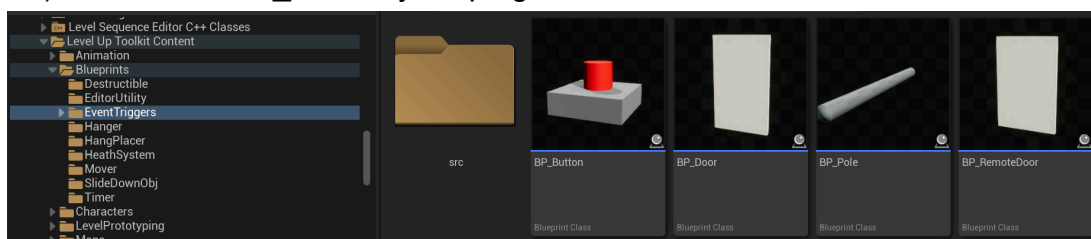
Parent Class: BP_ET_PickupTrigger

File Location:

Description: A functional pole asset that the avatar can grab and put it on his hand.

How to add the pole into your level:

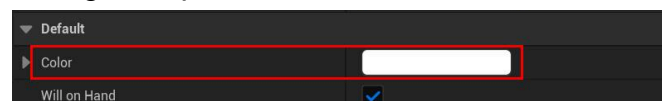
1): Locate the BP_Pole in your plugin menu or content drawer.



2): Drag and drop the pole into the level.

3): Optionally change the trigger’s radius to reset the pickup detect zone.

4): Optionally change the pole color.



Door

Name: BP_Door

Type: Actor

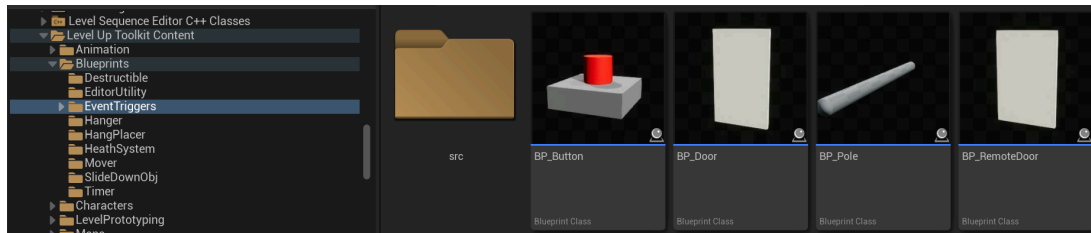
Parent Class: BP_ET_SelfTrigger

File Location:

Description: A door that the avatar can open by hitting it (press key E).

How to add the door into your level:

1): Locate the BP_Door in your plugin menu or content drawer.



2): Drag and drop the door into the level.

3): Optionally change the “Need Input” to false, then the door will open automatically when the avatar enters the trigger zone.

Button

Name: BP_Button

Type: Actor

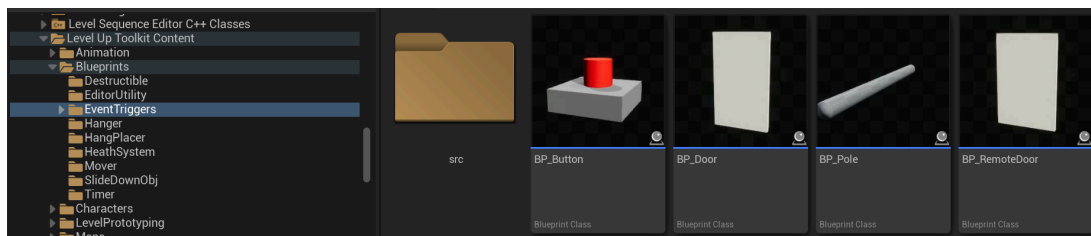
Parent Class: BP_ET_EventSender

File Location:

Description: A functional button asset that can call an event.

How to add the button into your level:

1): Locate the BP_Button in your content drawer.



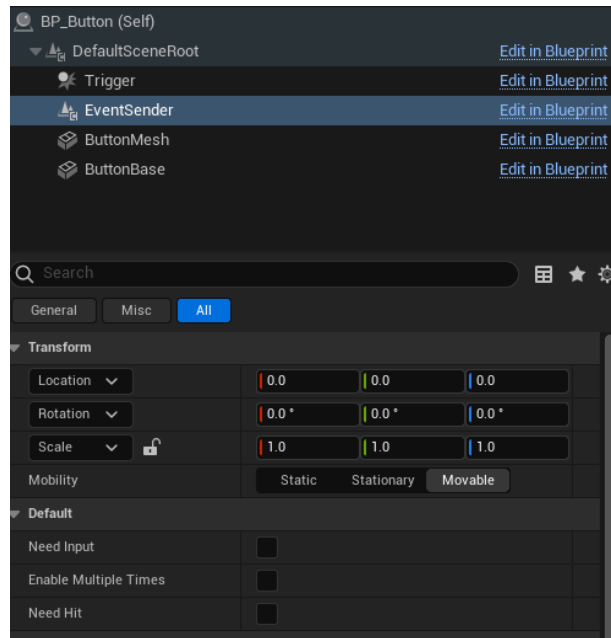
2): Drag and drop the button into the level.

3): Optionally change the trigger’s radius to reset the pickup detect zone.

4): Optionally change the “Need Input” to true, then the button will be pushed only when the avatar enters the trigger zone and press E.

5): Optionally change the “Need Hit” to true, then the avatar will play the hit animation when it pushes the button.

6): Optionally change the “Enable Multiple Times” to true, then the button can be pushed multiple times.



Remote Door

Name: BP_RemoteDoor

Type: Actor

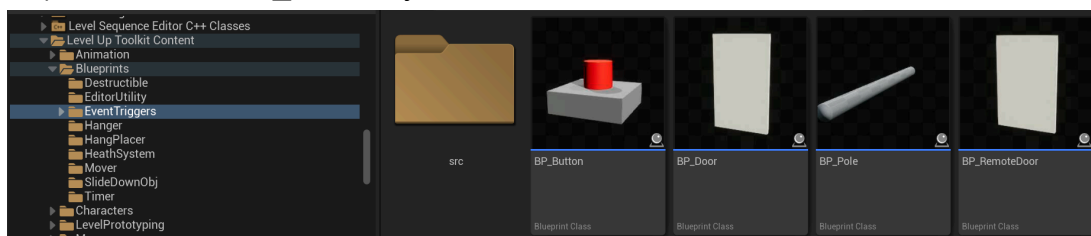
Parent Class: BP_ET_TransformReceiver

File Location:

Description: A door that the avatar can remotely open it by sending an event, e.g. Pushing a button.

How to add the door into your level:

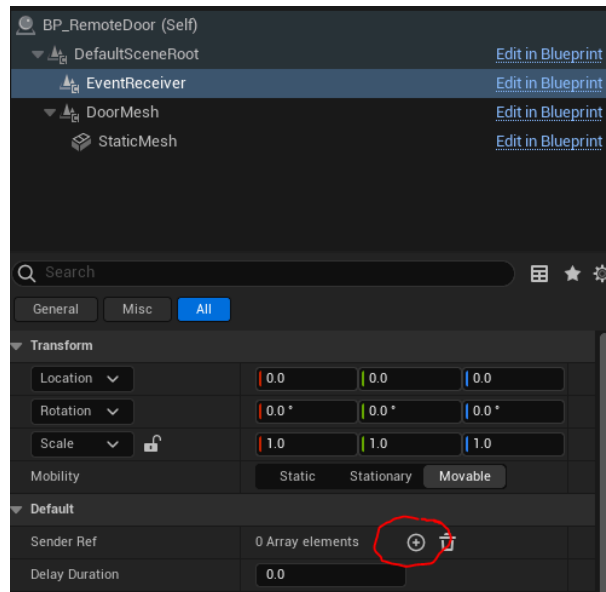
1): Locate the BP_Door in your content drawer.



2): Drag and drop the door into the level.

3): You also need an event sender actor to open the door, a fast approach could be the BP_Button.

4): Add the sender ref in the settings of BP_RemoteDoor's EventReceiver. Select the event sender you selected in the previous step.



5): Optionally change the “Duration” to reset the total time of opening the door.

6): Optionally change the “NewTransform” to change the direction of the opening.

7): Optionally change the “Delay Duration” to reset the delay of the callback that happens after the event is triggered.

Hide or Show Object

Name: BP_HideOrShow

Type: Actor

Parent Class: BP_ET_HideTrigger

File Location:

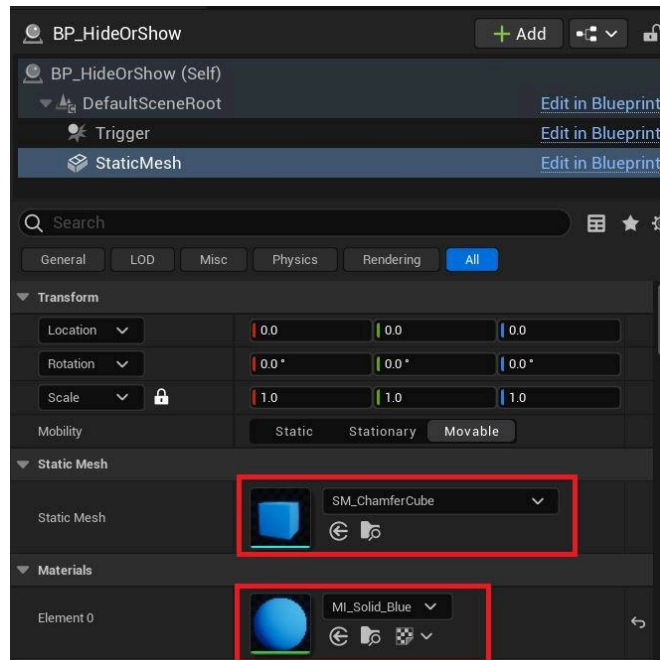
/Script/Engine.Blueprint'/LevelUpTest/Blueprints/EventTriggers/BP_HideOrShow.BP_HideOrShow'

Description: An actor that the avatar can interact with to hide or show an object.

Input:

1): Static Mesh & Material:

Change the hidden object model by choosing StaticMesh of the actor, and then change the Static Mesh and Materials in the detail panel.



2): Trigger to Hide:

Type: Boolean

Usage: Decide whether to hide the object when first trigger the event.

Checking the box means showing the object at first. The first time the player triggers the event, the object will hide.

3): Need Input:

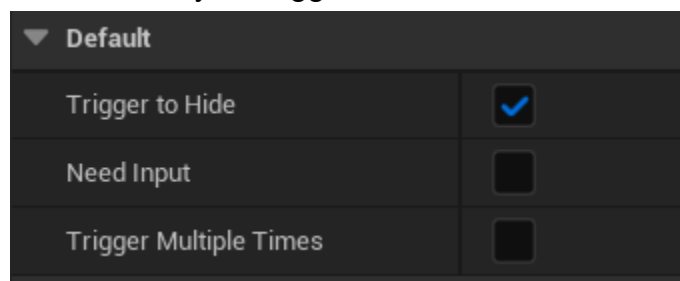
Type: Boolean

Usage: Decide whether the player needs to press a key to interact with the event. If check the box, the player needs to press E (the interact key) to trigger the event. If not check the box, when the player goes into the trigger area, the event will be triggered automatically.

4): Trigger Multiple Times:

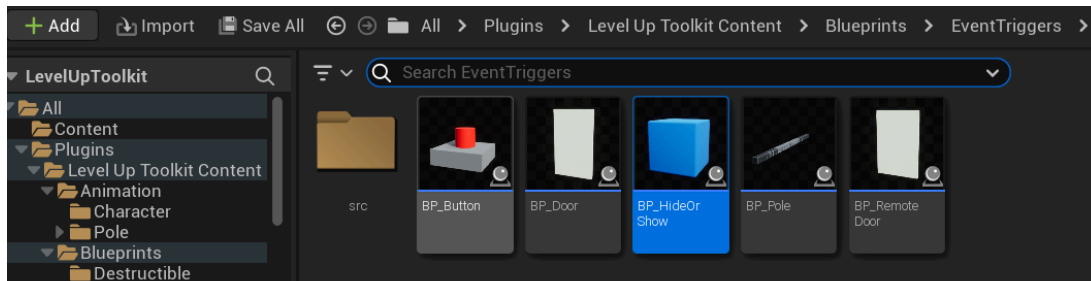
Type: Boolean

Usage: Decide whether to trigger the event multiple times. If not check the box, the hide/show event can only be triggered once.



How to add the actor into your level:

1): Locate the BP_HideOrShow in your plugin menu or content drawer.



2): Drag and drop the actor into the level.

3): Optionally change the trigger's transform to reset the interaction detect zone.

Portal Enter

Name: BP_Portal_Enter

Type: Actor

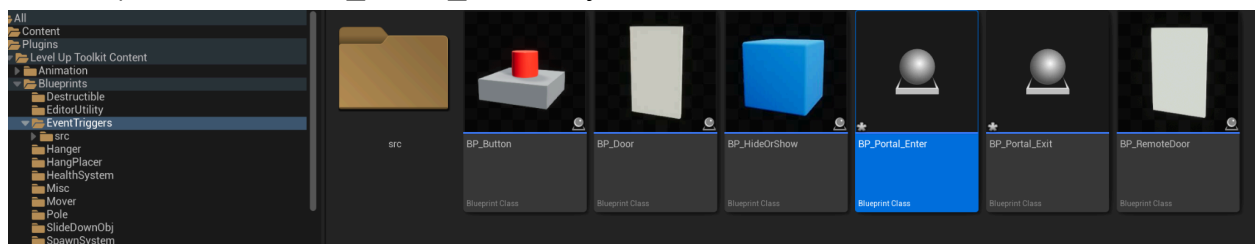
Parent Class: BP_ET_EventSender

File Location:

Description: A portal that will transmit the object it overlaps to another portal.

How to add a portal enter into your level:

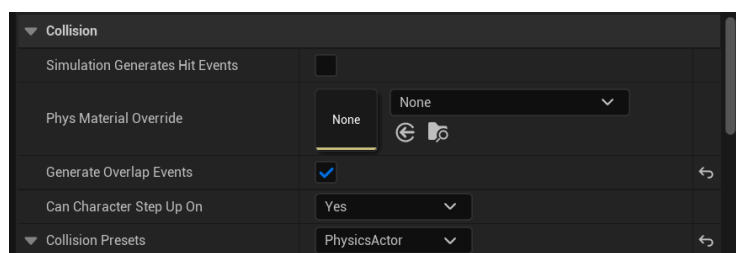
1): Locate the BP_Portal_Enter in your content drawer.



2): Drag and drop the Portal Enter into the level.

3): Optionally change trigger sphere's radius.

4): The item that you want to transmit should have "Generate Overlap Events" checked.



Portal Exit

Name: BP_Portal_Exit

Type: Actor

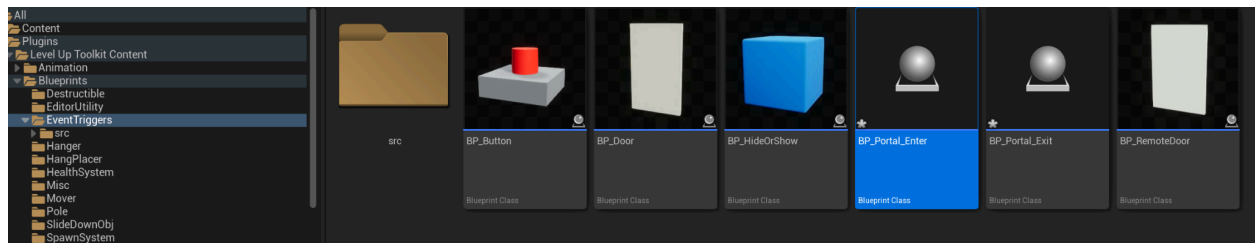
Parent Class: BP_ET_EventReceiver

File Location:

Description: A portal that can be set as the destination of your portal transmission.

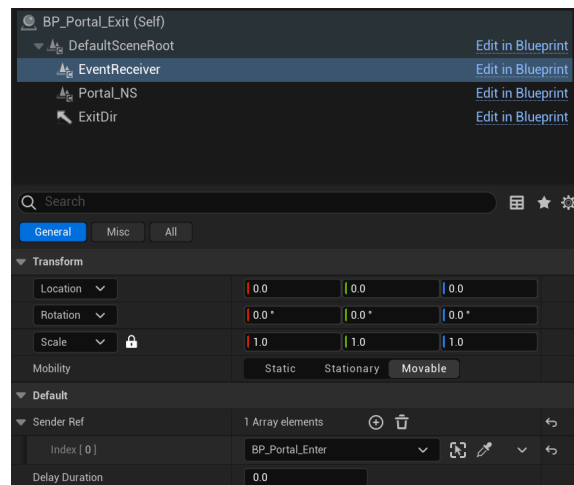
How to add the Portal Exit into your level:

1): Locate the BP_Portal_Exit in your content drawer.



2): Drag and drop the Portal Exit into the level.

3): Within its EventReceiver component, link its Portal Enters.



4): Optionally change the “Delay Duration” to reset the delay of the callback that happens after the event is triggered.

Hanger

Name: BP_HR_Hanger

Type: Actor

Parent Class: Actor

File Location:

Description: An actor that the avatar can hang on it with the pole.

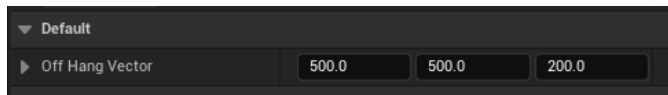
#Note: The avatar can only hang on it if it has the pole. When the avatar is on the hanger, he will jump off it by pressing the “Jump” key or the “s” key.

Input:

1): Off Hang Vector

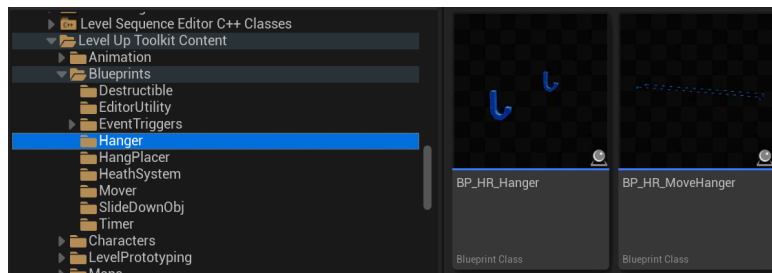
Type: Vector

Usage: Offhang direction and magnitude.



How to add the hanger into your level:

1): Locate the BP_HR_Hanger in your content drawer.



2): Drag and drop the hanger into the level.

3): Optionally change the “Off Hang Vector” to [0,0,500] if you want the avatar to jump up vertically after hanging.

Conveying Hanger

Name: BP_HR_ConveyingHanger

Type: Actor

Parent Class: BP_HR_Hanger

File Location:

Description: Conveying hanger that will bring the hanging avatar from start hanger to end hanger.

#Note: Designer should move StartHanger and EndHanger when customizing the two endpoints.

Input:

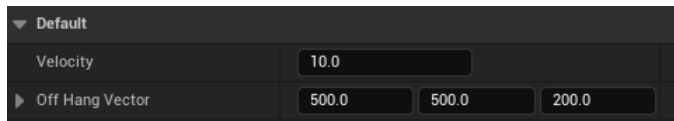
1): Velocity

Type: Float

2): Off Hang Vector

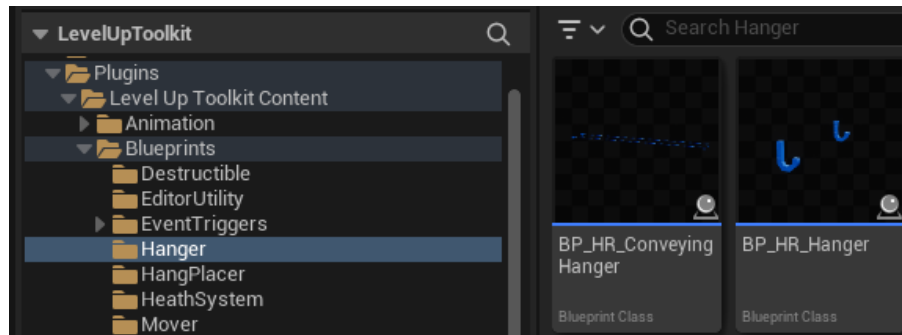
Type: Vector

Usage: Offhang direction and magnitude.



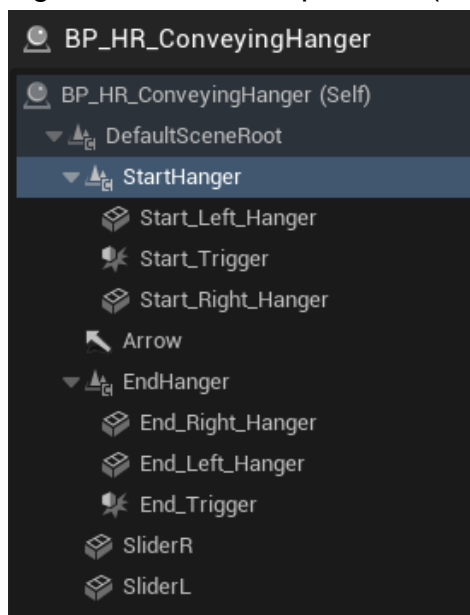
How to add the moveable hanger into your level:

1): Locate the BP_HR_ConveyingHanger in your content drawer.



2): Drag and drop the conveying hanger into the level.

3): If you want to customize the start position and the end position, select the StartHanger and the EndHanger and reset their position. (See the note above.)



4): Optionally change the “Velocity” to adjust the speed of sliding.

5): Optionally change the “Off Hang Vector” to [0,0,500] if you want the avatar to jump up vertically after hanging.

Slide Down Pole

Name: BP_SP_SlidePole

Type: Actor

Parent Class: Actor

File Location:

Description: The avatar can slide down along the pole.

Climbable Objects

Tag: Climbable

Target Type:

Description: All the actors with “Climbable” actor tag can be climbed by the avatar. The character can jump to hang on the edge of the object. Then, he can choose either climb up (using jump button) or drop down (using s key).

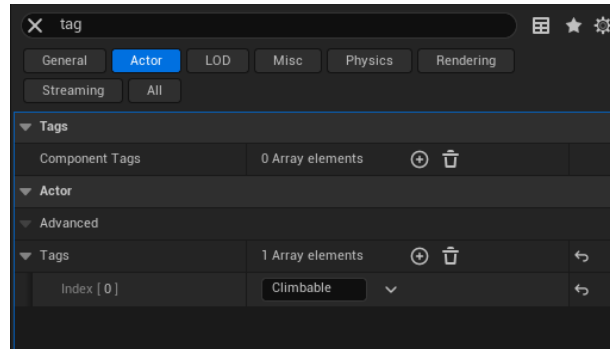
Note: For better experience, the edges are better to be straight.

Note: The character can ONLY climb on an actor that has a “static mesh component.”

Note: The character can ONLY climb on an actor that is more than 180 cm(character's height) away from the ceiling(if there is a ceiling).

How to add a climbable object into your level:

- 1): Select an object in the scene.
- 2): Add an actor tag called "Climbable" to the object.



Health System

Dead Zone

Name: BP_TB_DeadZone

Type: Actor

Parent Class: Trigger Box

File Location:

Description: a dead zone trigger area

Damage

Damage Once

Name: BP_TB_DamageOnce

Type: Actor

Parent Class: Trigger Box

File Location:

Description: Cause damage once when the player enters the trigger area.

Input:

- 1): Damage Per Overlap

Type: Float

Usage: The amount of damage caused each time.

Damage Loop

Name: BP_TB_DamageLoop

Type: Actor

Parent Class: Trigger Box

File Location:

Description: Keep causing damage when the player is in the trigger area.

Input:

1): PerXSecond

Type: Float

Usage: Cause damage per x second.

2): DamagePerXSecond

Type: Float

Usage: The amount of damage caused each time. To heal, make damage negative.

Spawn System

Respawn

Name: BP_SpawnPos

Type: Actor

Parent Class: Actor

File Location:

Description: Determine the respawn position according to the check point.

Input:

1): Spawn Pos Sum

Type: Integer

Usage: match with the number of check points. [important]

Check Point

Name: TB_CheckPoint

Type: Actor

Parent Class: Trigger Box

File Location:

Description: Save player's progress and respawn player at corresponding spawn point.

Input:

1): Spawn Pos Sum To

Type: Integer

Usage: match with the number of spawn point. [important]

End Zone

Name: TB_EndZone

Type: Actor

Parent Class: Trigger Box

File Location:

Description: The player's destination to win the game.

Timer

CountDownEnter

Name: BP_TB_CountDownEnter

Type: Actor

Parent Class: Trigger Box

File Location:

Description: Set a countdown timer when the player enters the trigger box, if counts to zero, the player dies.

Input:

1): Minutes

Type: Integer

Usage: Minutes of the timer.

2): Seconds

Type: Integer

Usage: Seconds of the timer.

CountDownExit

Name: BP_TB_CountDownEnter

Type: Actor

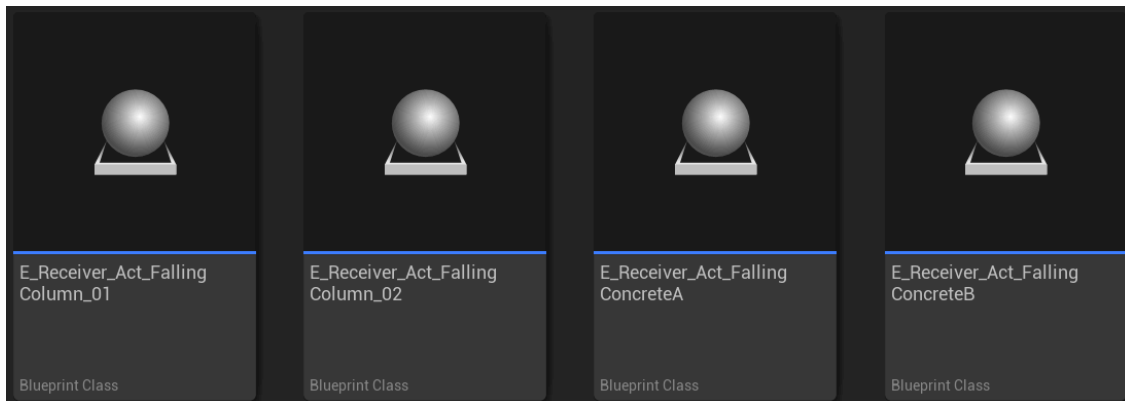
Parent Class: Trigger Box

File Location:

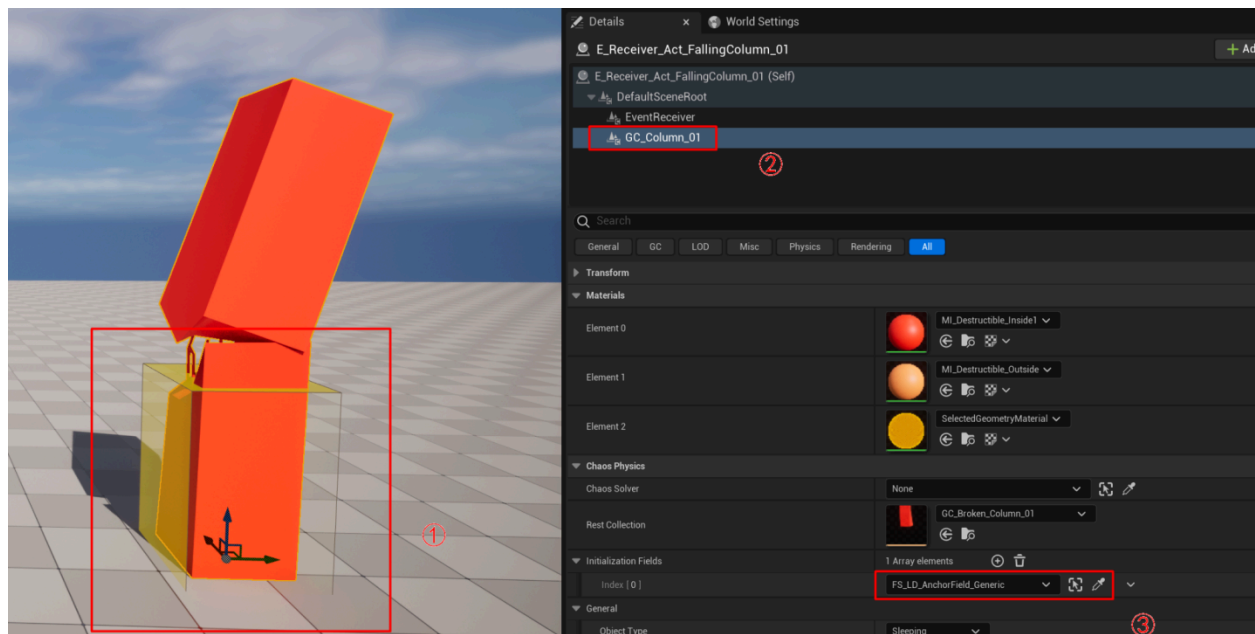
Description: Cancel a countdown timer when the player enters the trigger box

Destructible

All of these Blueprints starting with the `E_Receiver_Act` prefix are the destructible objects. These are pre-made breakable assets. As an Event Receiver, they need an Event Sender Trigger to be activated.



After you trigger them, the Geometry Collection Component in these blueprints will set their Object Type from Sleeping to Dynamic, so they start falling. Sometimes you want to keep part of the geometry static, you can use `FS_LD_AnchorField` (the yellow cube) to implement this.

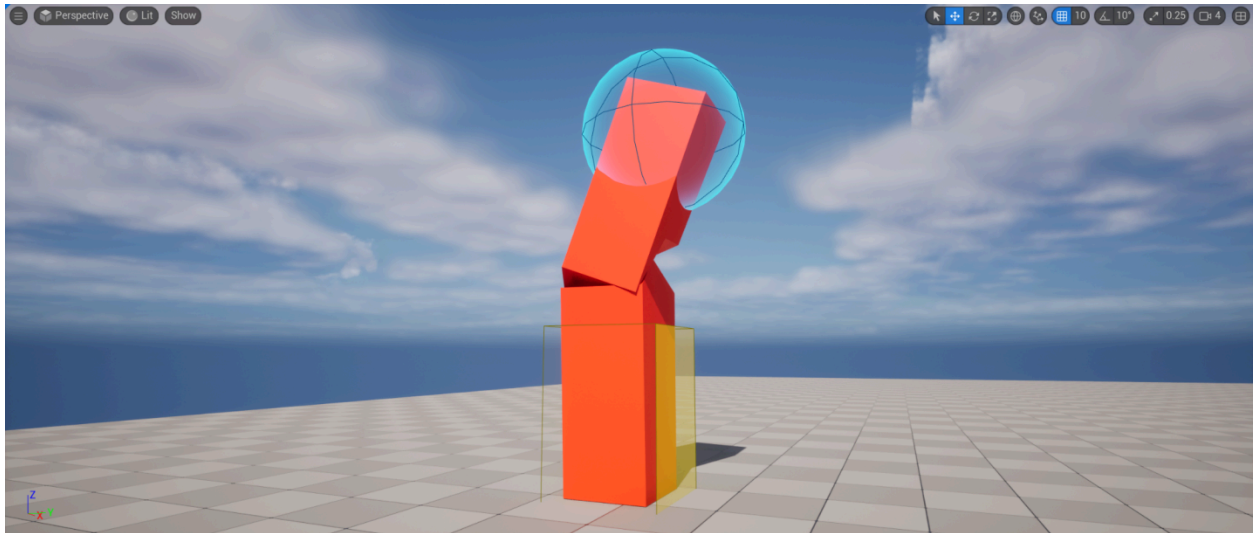


- ① Move the FS_LD_AnchorField to the right place and scale it.
- ② Select the geometry collection component in the E_Receiver_Act
- ③ Add an array element using the “+” button and select your field.



The difference between using and not using the FS_LD_AnchorField

If you want to manually control the destruction position and destruction time, you can use the FS_LD_MasterField (the blue sphere).



In the details panel, you can select the Activation Type, for example the Delay mode, you can set up the Delay Amount to control how many seconds after begin play you destructible actor will explode.

▼ Default	
Field Active	☒
Debug	☐
Activation Type	Delay ▼
Field Falloff Shape	Sphere ▼
Delay Amount	1.0

To learn more about the destruction in UE5, please look at the official documentation.

<https://docs.unrealengine.com/5.0/en-US/chaos-destruction-in-unreal-engine/>

Conveyor Belt

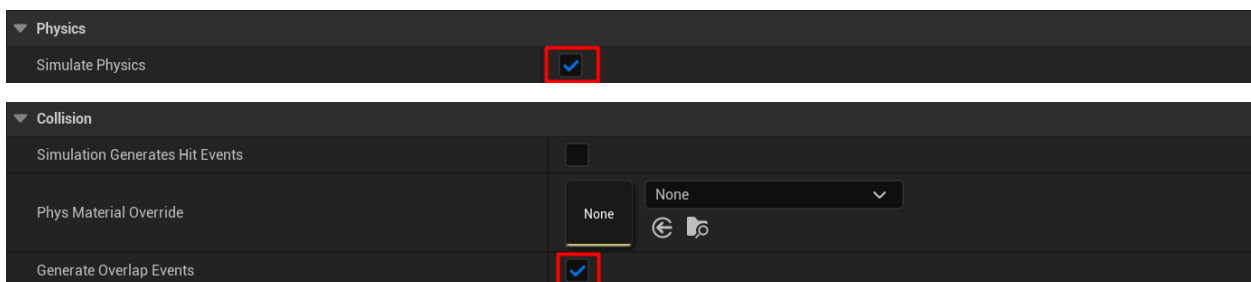
Conveyor belts can be used to speed up or slow down players at specific locations. The arrows show the direction of the conveyor belt.



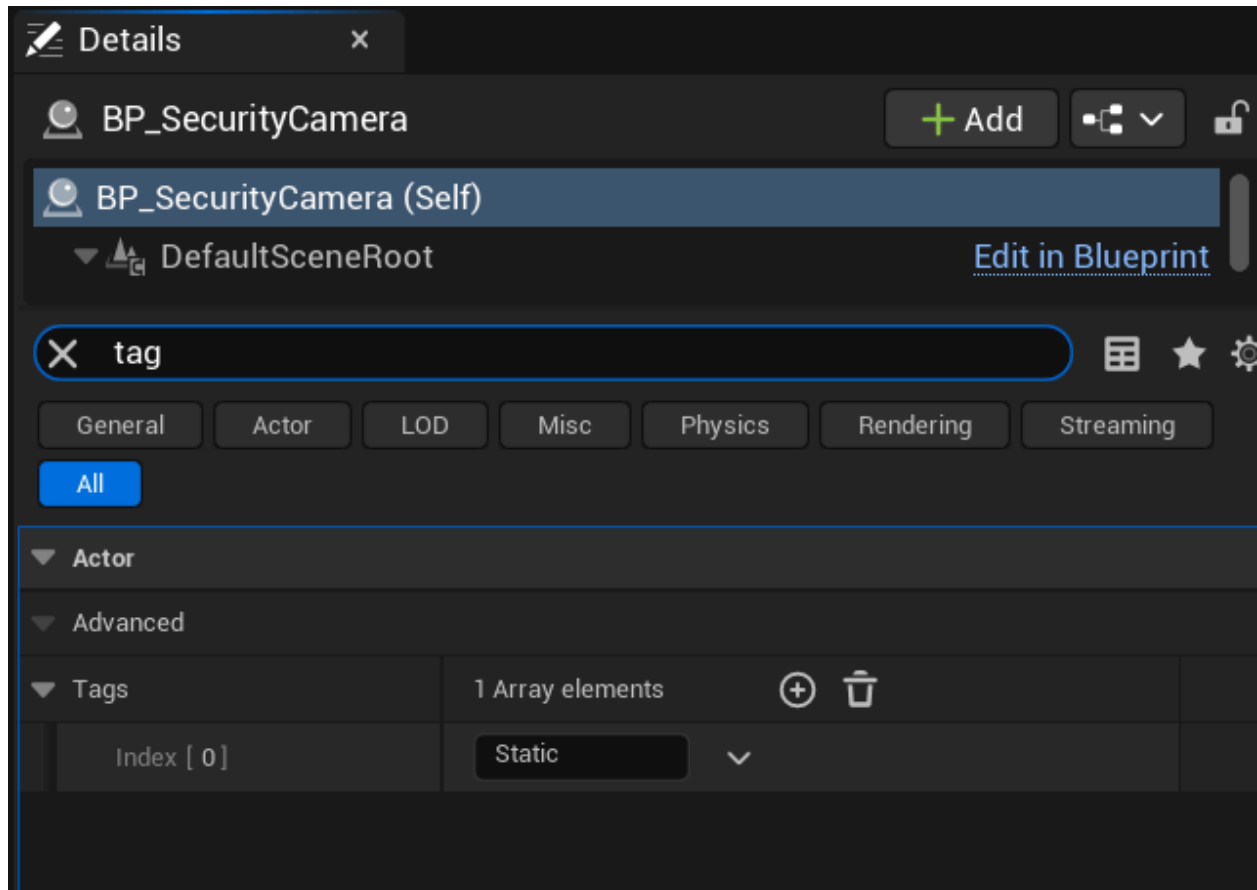
There's a speed value in the details panel to adjust the speed. 700 is basically the value that is equal to the player's running speed.



If you want an actor to also move on the conveyor belt, select actor and check **Simulate Physics** and **Generate Overlap Events** in its details panel.



If you want a blueprint actor that doesn't move on the conveyor belt, add an actor tag called "Static" to the object.

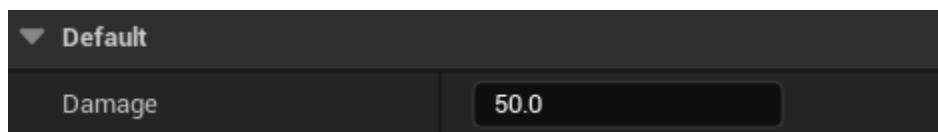


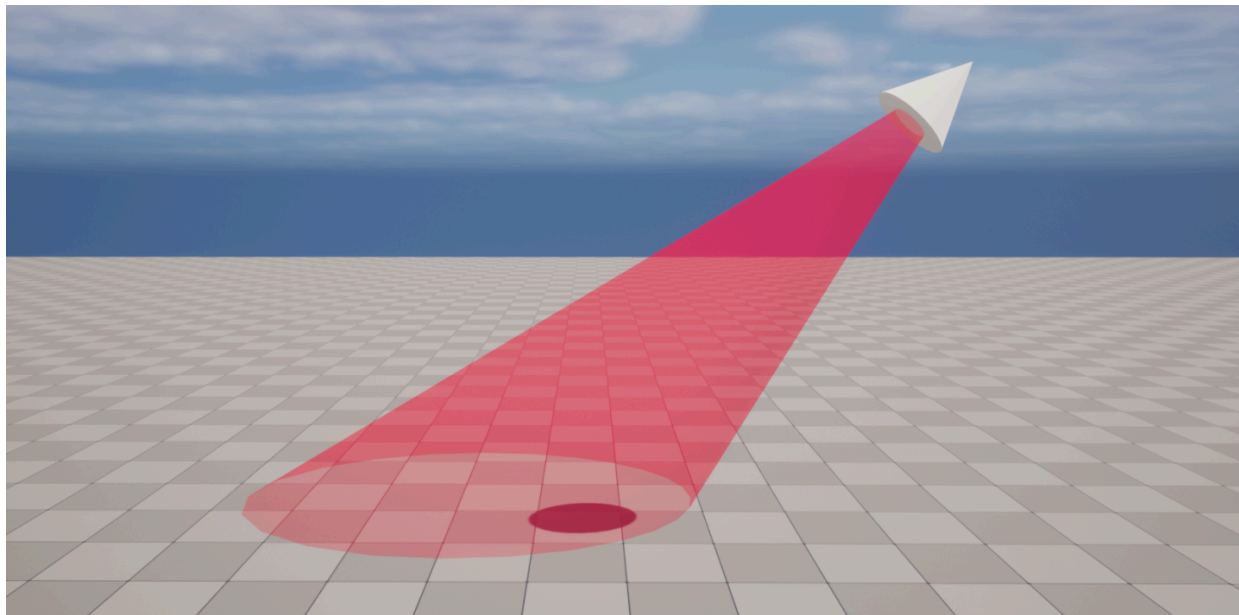
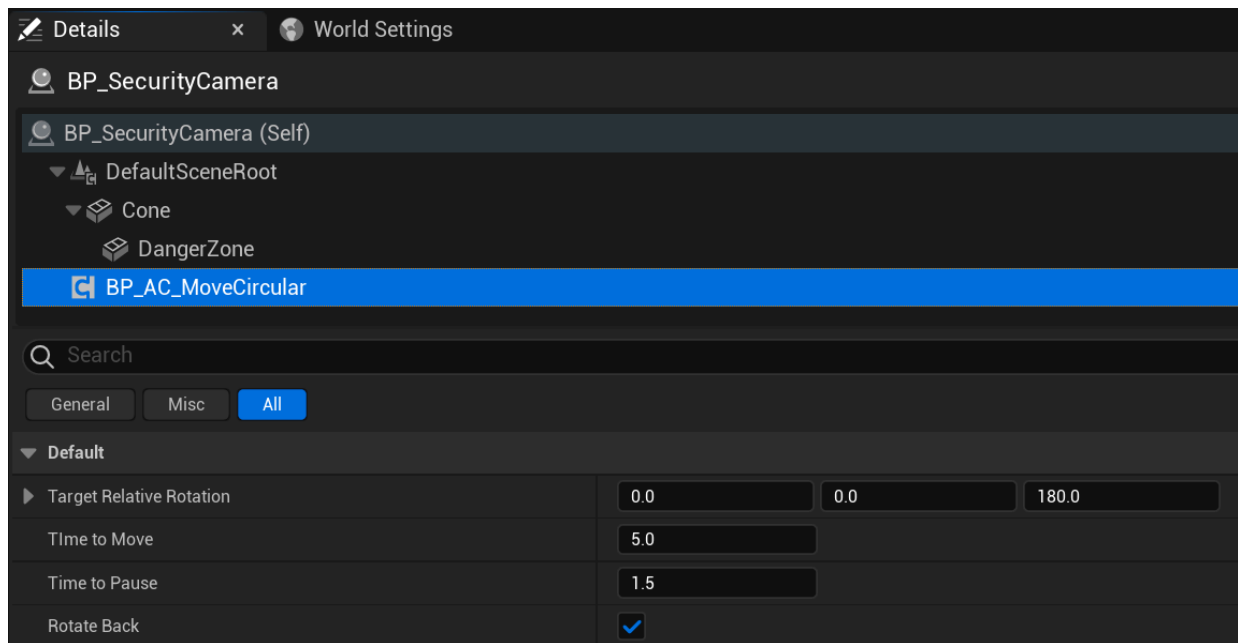
Security Camera

The security camera is a moving target that the player needs to avoid. It contains a BP_AC_MoveCircular component to control the rotation angle and speed of the camera.

When the player enters the red zone, they will get hurt.

There is a damage value in the detail panel to adjust the security camera damage.





Spawn Enemy

Name: BP_SpawnEnemy

Type: Actor

Parent Class: BP_ET_SelfTrigger

File Location: /Content/Blueprints/Enemy/BP_SpawnEnemy.uasset

Description: A trigger that can spawn certain types and numbers of enemies in certain areas. After the player has a pole, press the left mouse button to attack the enemy. The enemy randomly walks in a certain range, after it finds the player, it will chase and attack the player in the range.

How to add the spawn enemy into your level:

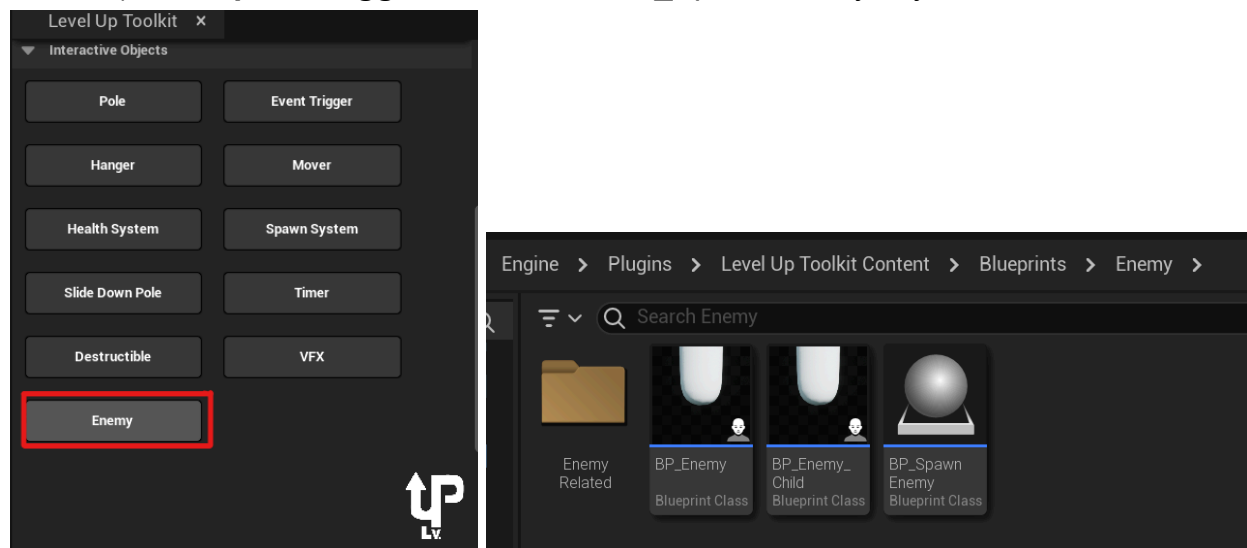
Video Instruction:

https://drive.google.com/file/d/1bmCUTcd8VpT36mpBp0MjGLEKYCqgogur/view?usp=drive_link

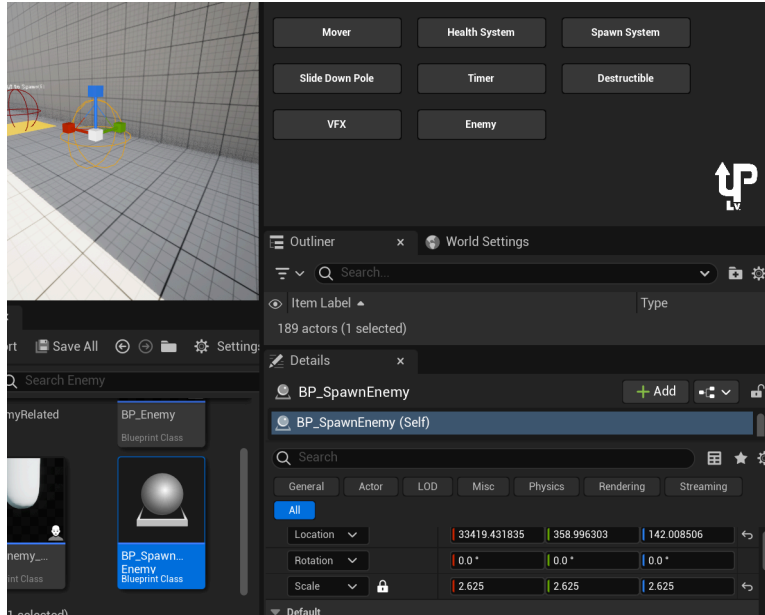
Patrol Route Instruction:

https://drive.google.com/file/d/178EeJC_72TyqaRB7Wd2Bsu57lhuDv5TK/view?usp=sharing

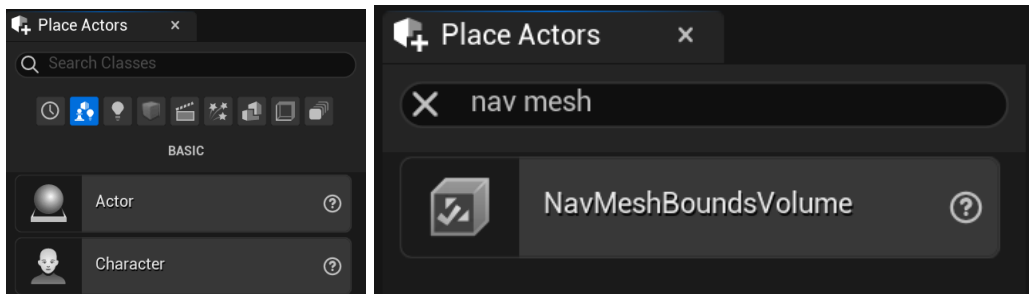
1): **Find spawn trigger:** Locate the BP_SpawnEnemy in your content drawer.



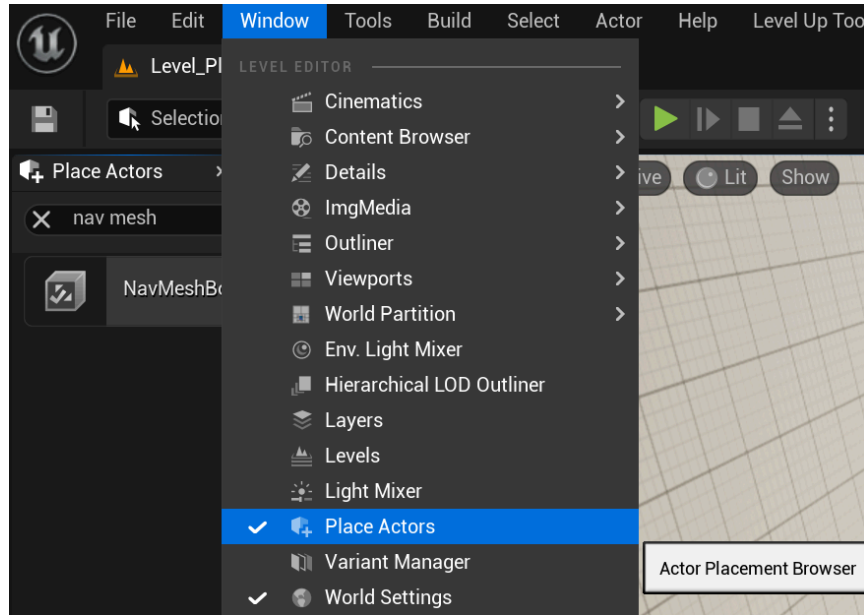
2): **Place spawn trigger:** Drag and drop the BP_SpawnEnemy into the level. The player must collide with the BP_SpawnEnemy area in order to trigger the spawning, so adjust the size and placement of BP_SpawnEnemy to ensure the player enters that space (or alternatively use step 7 for other trigger methods).



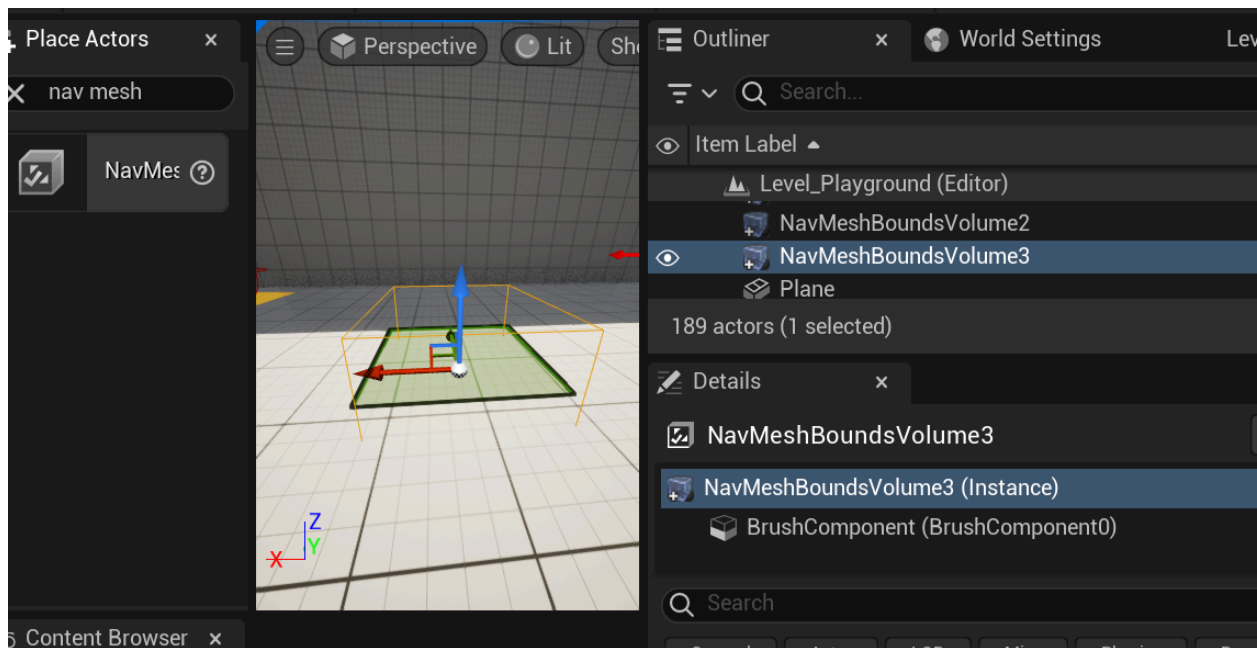
3): **Find nav mesh:** Find Nav Mesh Bounds Volume in the Place Actors panel on the left side.



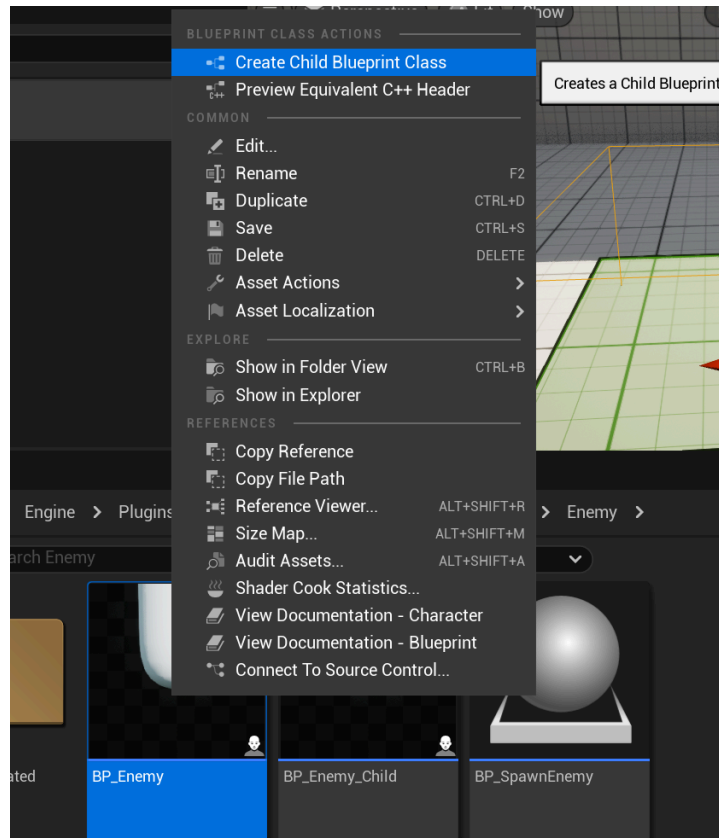
If you can't find the Place Actors panel, choose Window>Place Actors on the top left toolbar.



4):**Place nav mesh:** Drag and drop the Nav Mesh Bounds Volume in the scene, press P to show the navigable area (green area), this area is the enemy's walkable range.

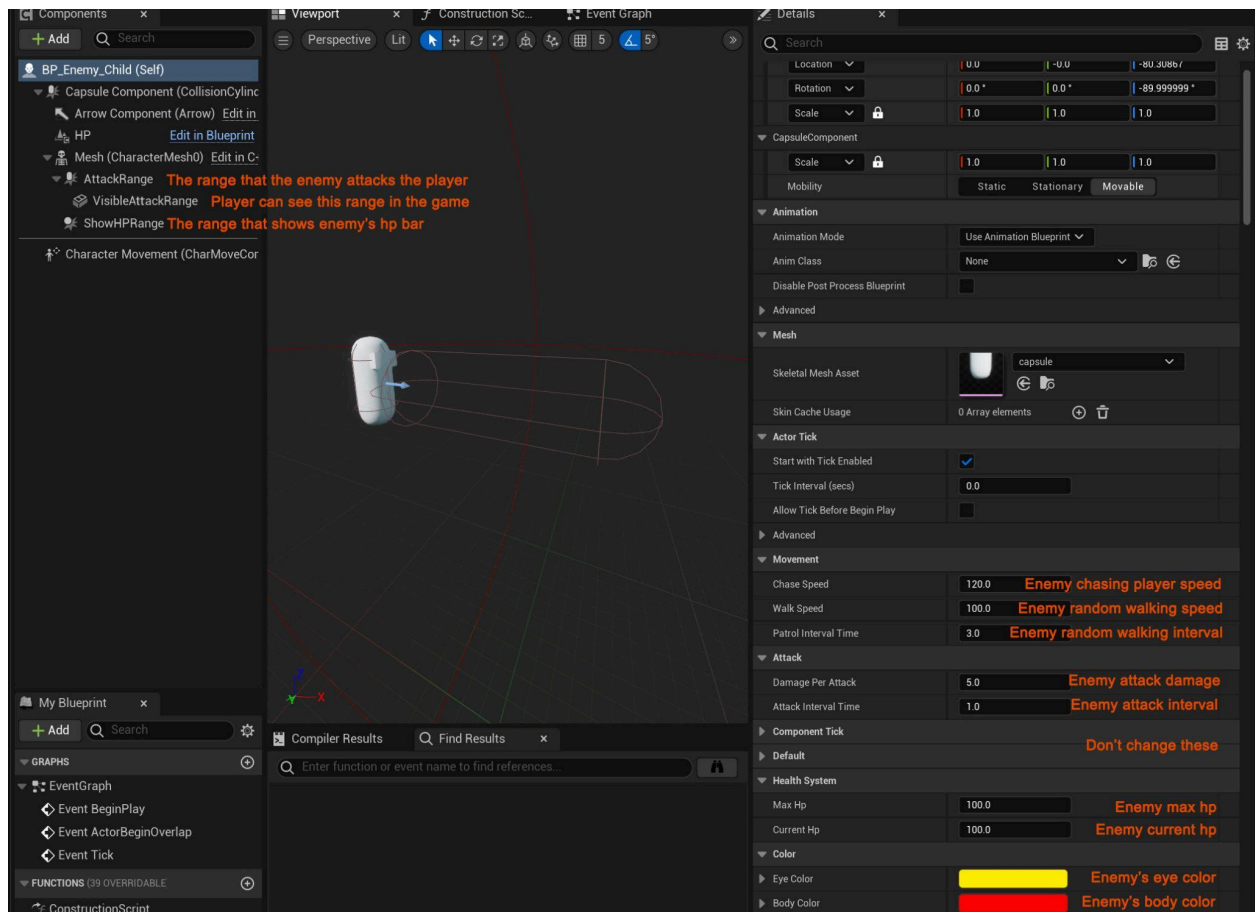


5): **Create new enemy:** Right click on the BP_Energy, choose create child blueprint class. Optionally change the name of the new blueprint.

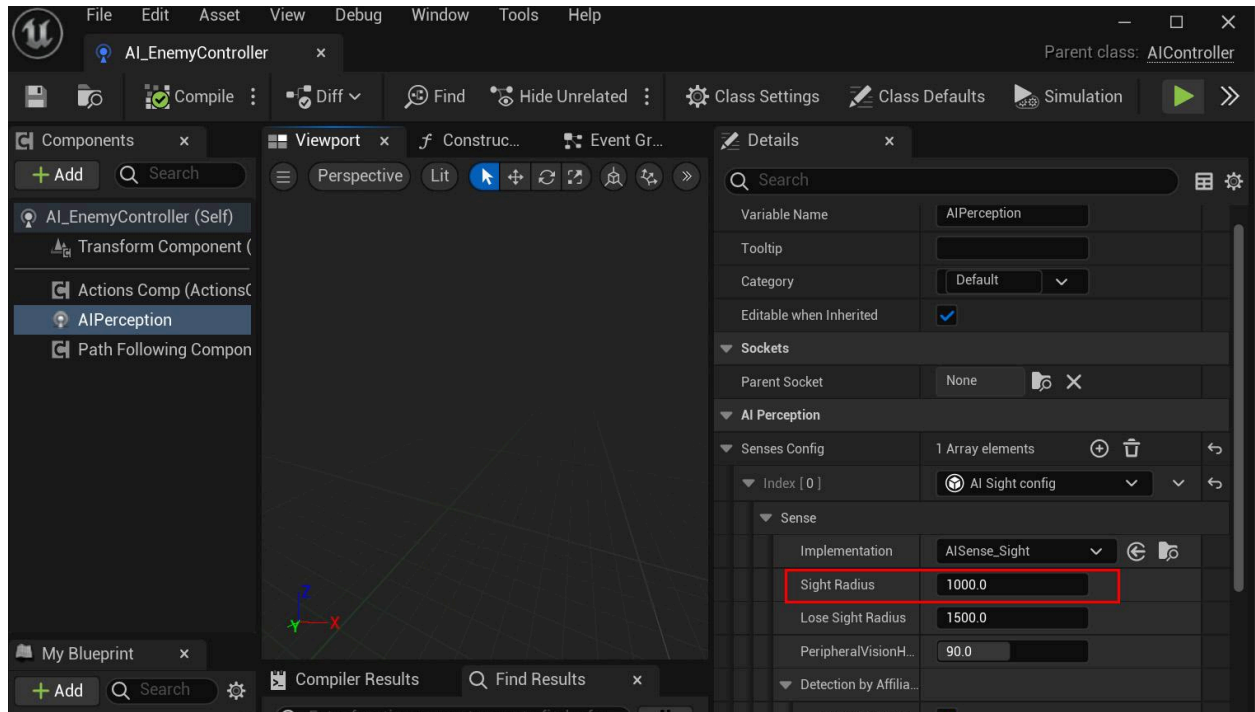


Double click on the new blueprint and enter the blueprint window. In the detail panel on the right side and the components panel on the left side, there are some values you can change to customize the enemy.

The player attack damage is 50, so you can change the max hp to adjust the enemy's difficulty. Hp range is 0 to infinite.



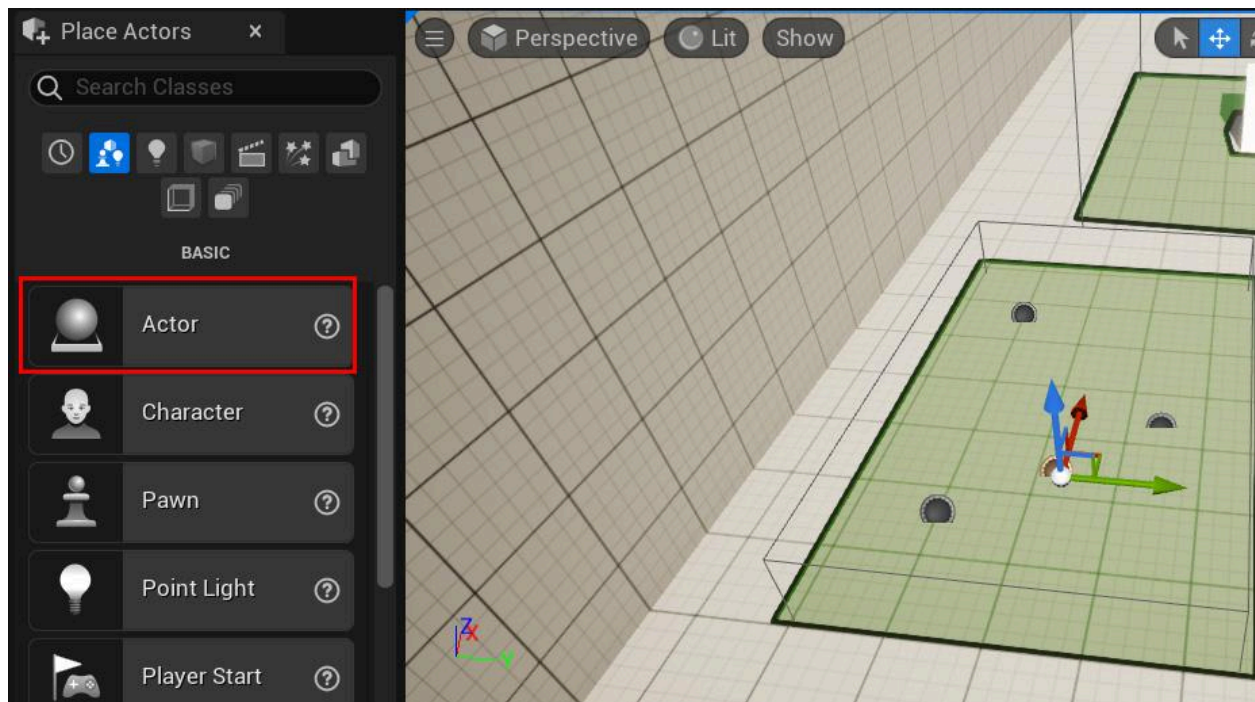
Double click on the **AI_EnergyController** blueprint in the enemy folder, choose **AI Perception** in the components panel on the left side, you can change the seeing player range by changing the sight radius on the right side. All enemies share the same sight radius.



Patrol Route

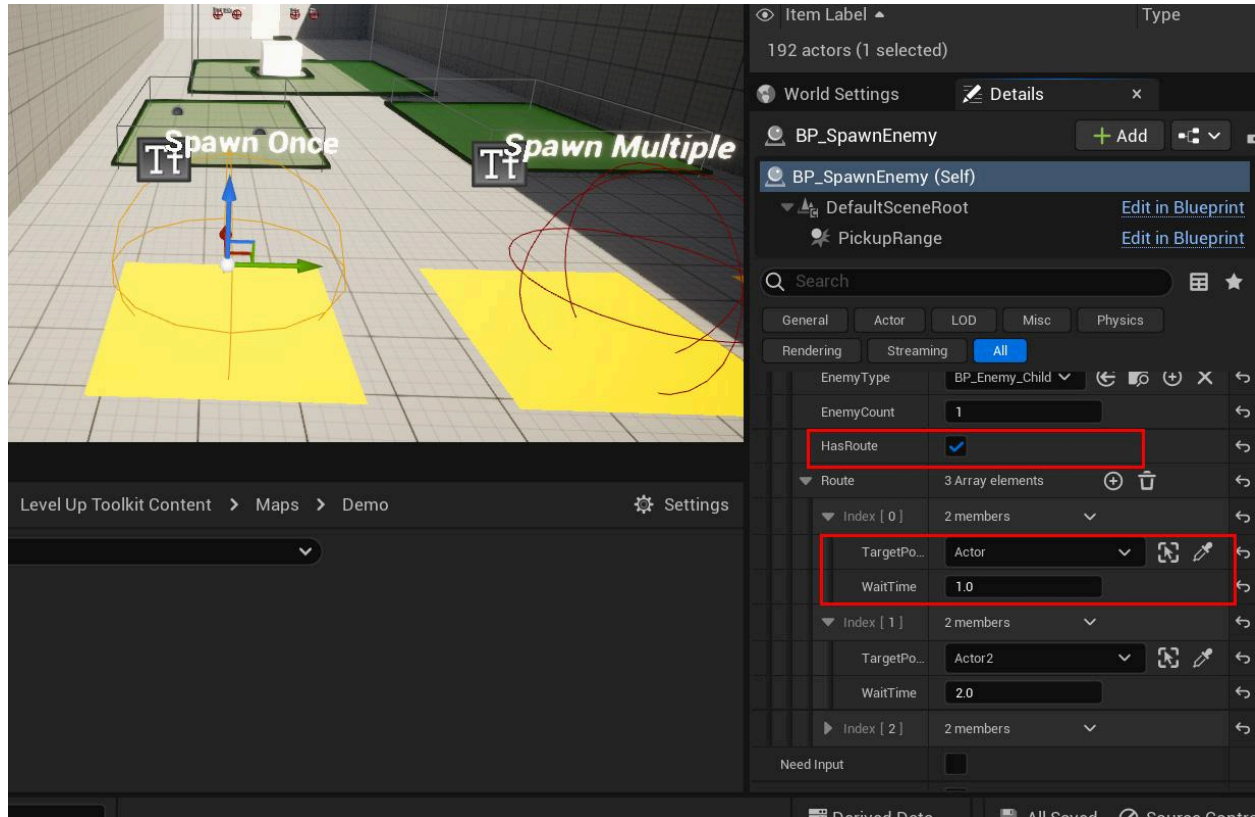
Enemies can also patrol according to established routes.

First, drag and drop some actors from the Place Actors panel to the map as the target destination.



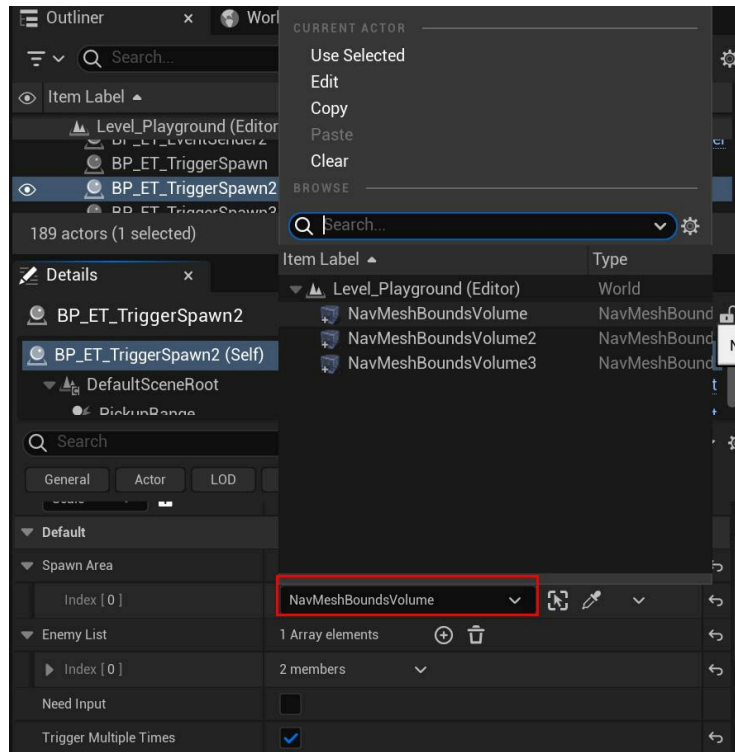
Then, check the **HasRoute** checkbox in the BP_SpawnEnemy. If you uncheck this checkbox, the enemy will walk randomly. Click the plus button on the right of Route,

and click the Eyedropper tool on the right of the **TargetPosition**, choose the target position you put in the map, then the target destination will be in the route. You can also change the **Wait Time** after the enemy reaches each target point. Add every target point in the route. Then the enemy will follow the route.

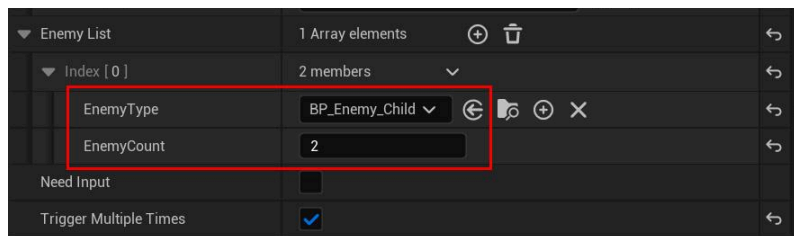


6): **Set values in spawn trigger:** choose the spawn trigger in the map.

In the Spawn Area, choose the NavMeshBoundsVolume where you want the enemy spawn .



In the enemy List, choose the enemy type and the quantity you want to spawn.



7): Optionally change the “Need Input” and “Trigger Multiple Times” to decide how to trigger this event, for more details, see “Event Sender Component” chapter.

Extra plugins –FMOD

Installation

During project creation, you must select the **C++** Project Default, although you can add Blueprints to your **C++** projects later.

Resources:

Visual Studio download link: <https://visualstudio.microsoft.com/downloads/>

FMOD download link: <https://www.fmod.com/download#fmodstudio>

LevelUpToolkit download link:

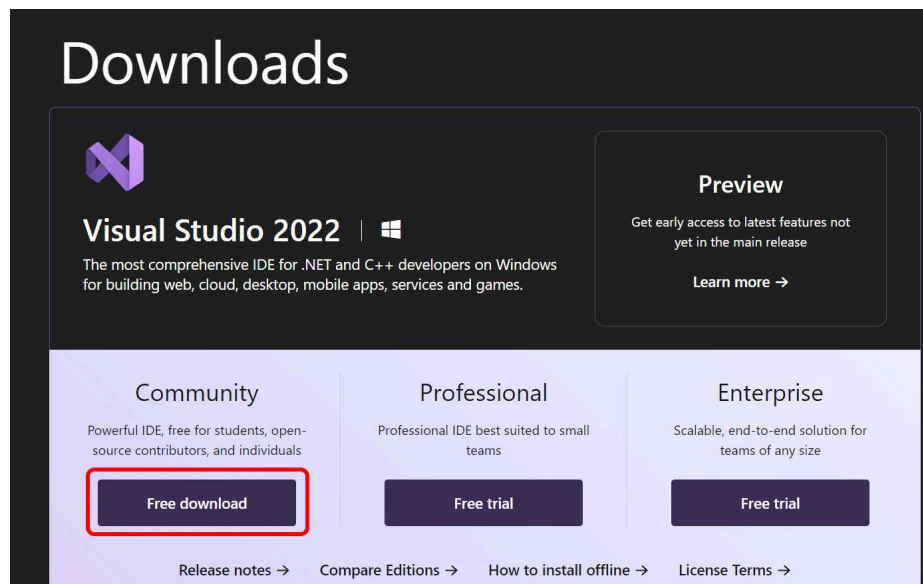
https://drive.google.com/file/d/105uBboS6Vw6ntMwTTPQjy7bK5UeJBKU2/view?usp=drive_link

STEP 1: Install Visual Studio

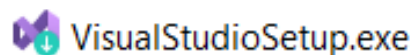
If you've already installed visual studio, please skip this step.

Visual Studio is commonly used for C++ development in Unreal Engine, and it can be used to set up the development environment for projects that involve FMOD for audio.

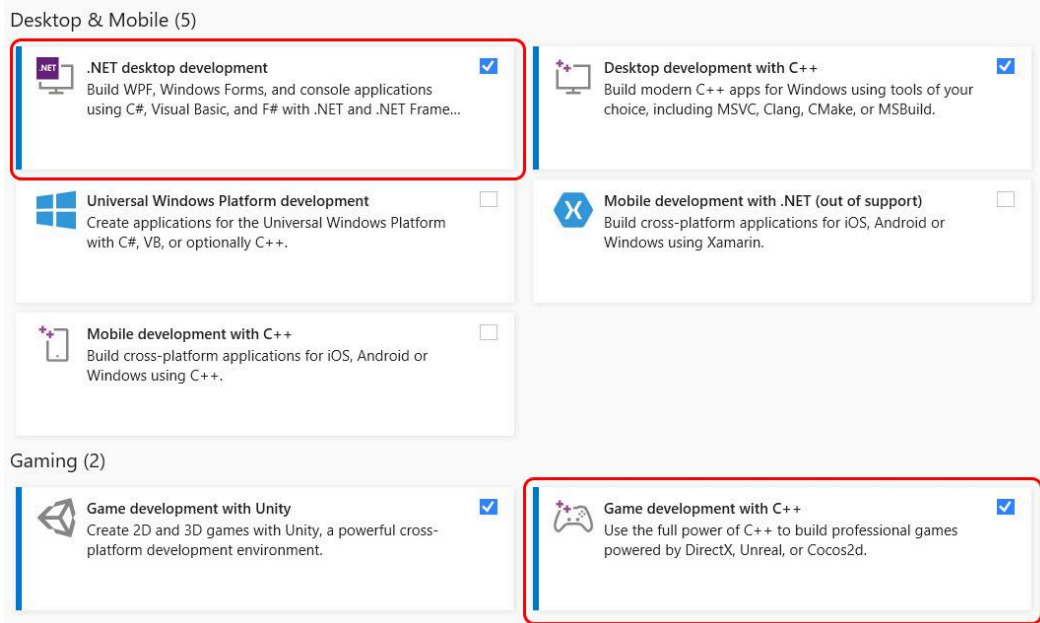
Download link: <https://visualstudio.microsoft.com/downloads/>



After downloading the installer, double click **VisualStudioSetup.exe** to install.



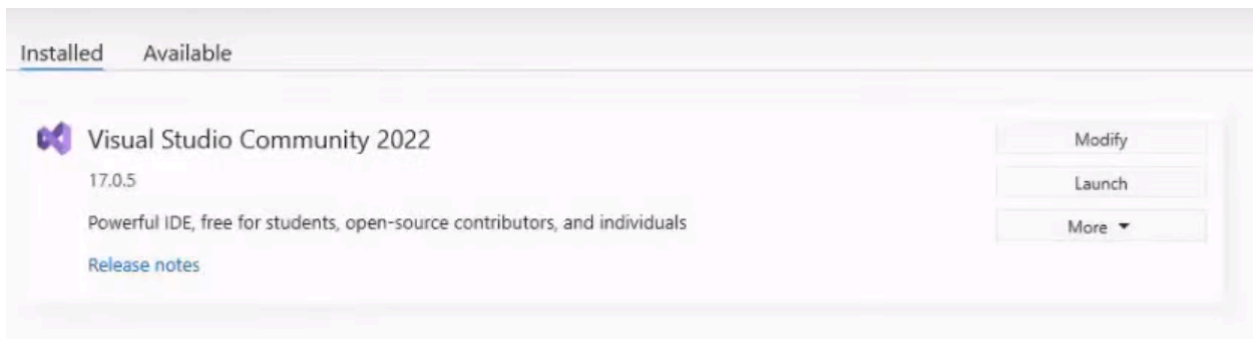
Choose the following options in the installer: **.NET desktop development, Game development with C++**



Press **Install** to start the installation.



If you see this panel, it means the installation is complete.

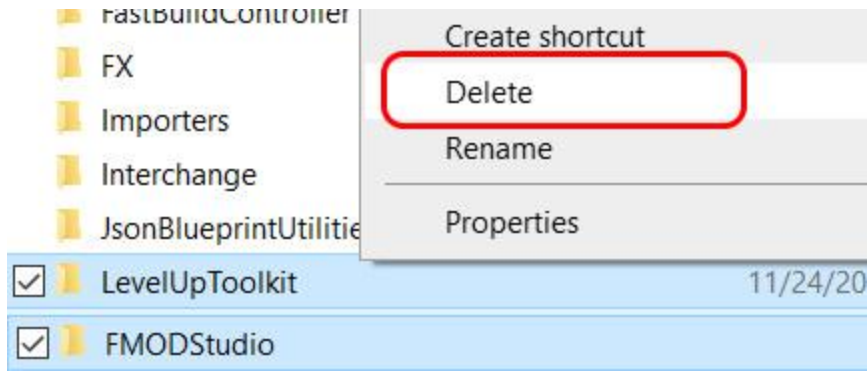


STEP 2: Remove Level Up Toolkit/FMOD From Engine Folder

If you didn't install the level up toolkit or FMOD plugin before, please skip this step.

Open **the Unreal Engine plugin installation root directory**. (Default directory is C:\Program Files\Epic Games\UE_5.1\Engine\Plugins)

Find **LevelUpToolkit** and **FMODStudio** folder , **delete** the folders.



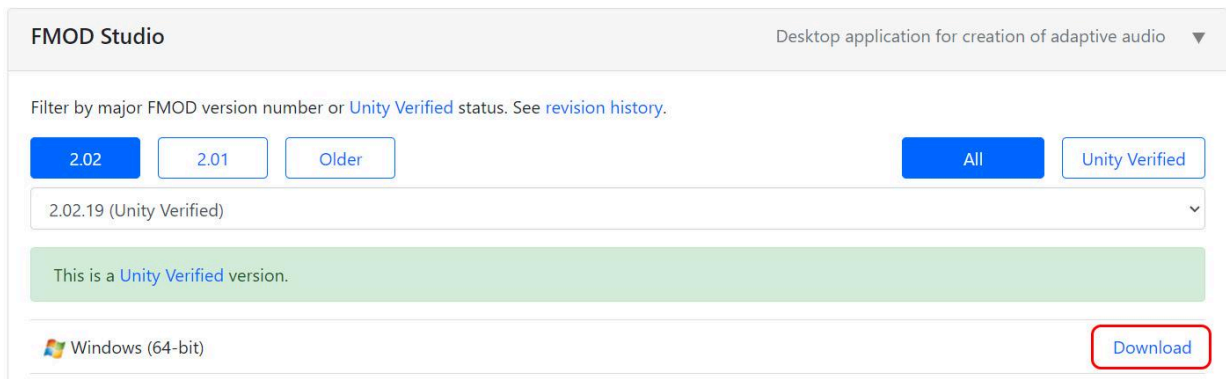
STEP 3: Download and Install FMOD

link: <https://www.fmod.com/download#fmodstudio>

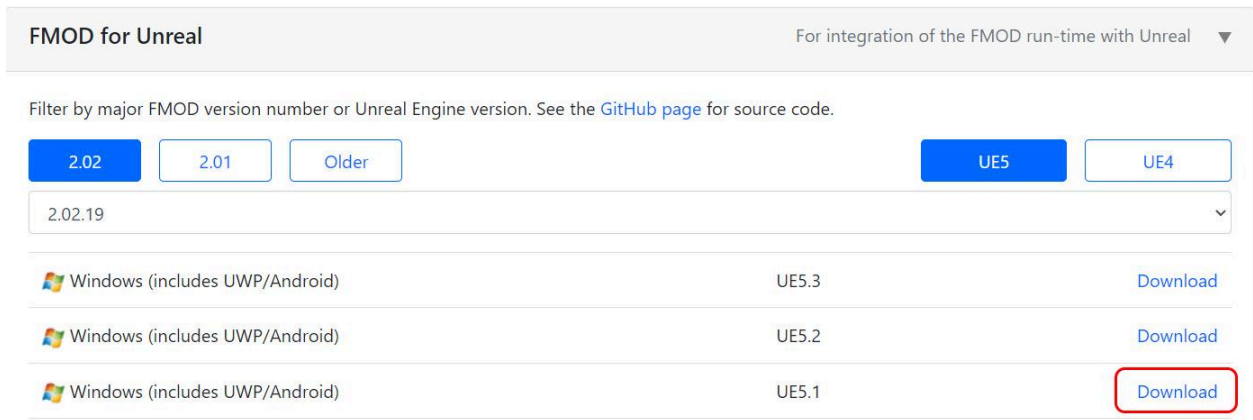
First, **sign in or register** an account on the FMOD website.

Please **sign in** to view downloads.

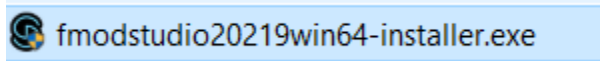
After signing into the website, locate **FMOD Studio**, **download** the latest version of FMOD studio.



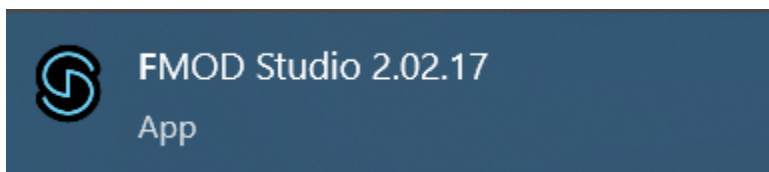
Scroll down the website and locate **FMOD for Unreal**. Select the version compatible with UE5.1, and click the **download** button.



After downloading, double click the **fmodstudio20219win64-installer.exe**. Press **Next** until it starts to install.

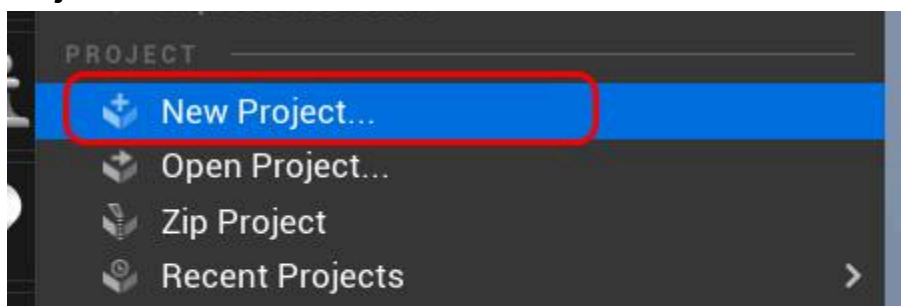


After installation, if you can find FMOD Studio in the Windows search bar, that means the installation was successful.

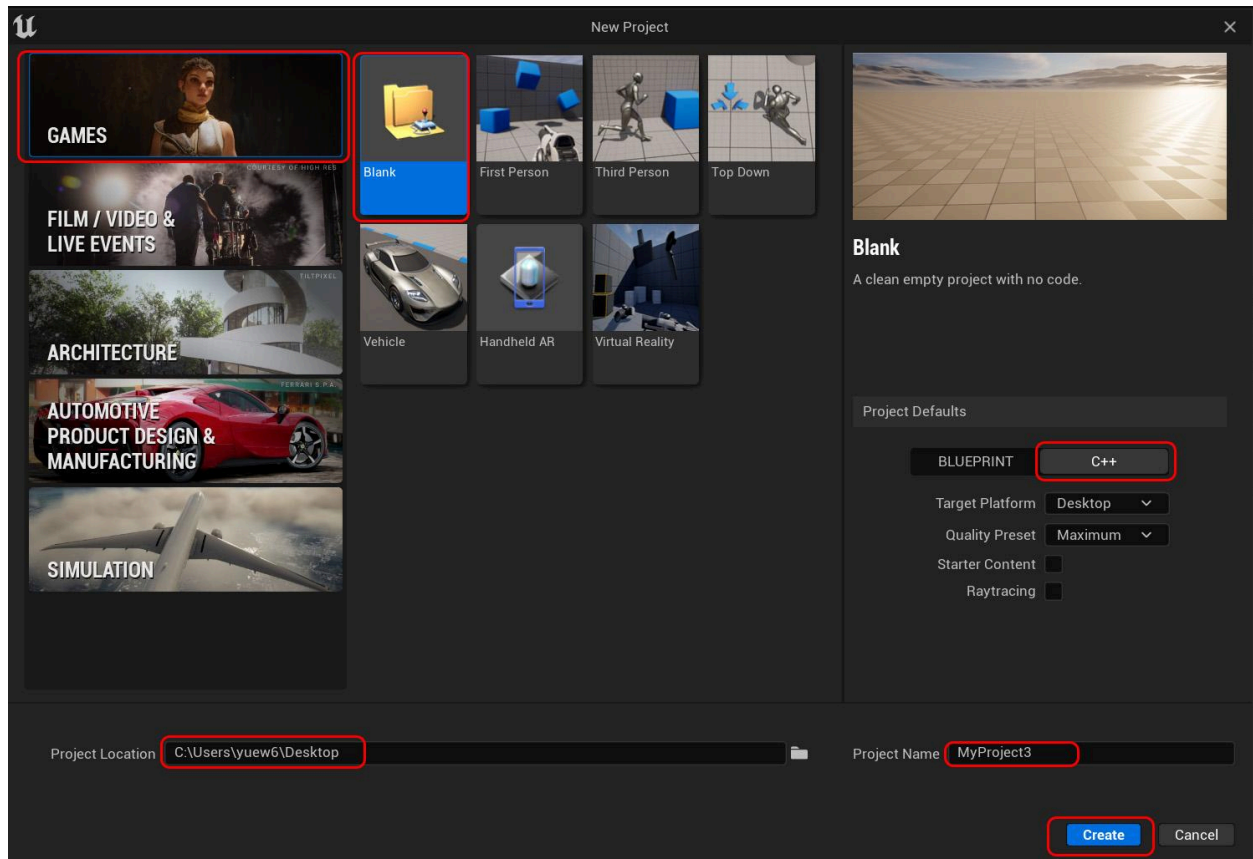


STEP 4: Create New C++ Project

Open Unreal5.1, if you've already created projects before, choose **File> New Project**.



Choose **GAMES**, **Blank**, **C++**. Choose your **Project Location** and **Project Name**. And then click **Create**.



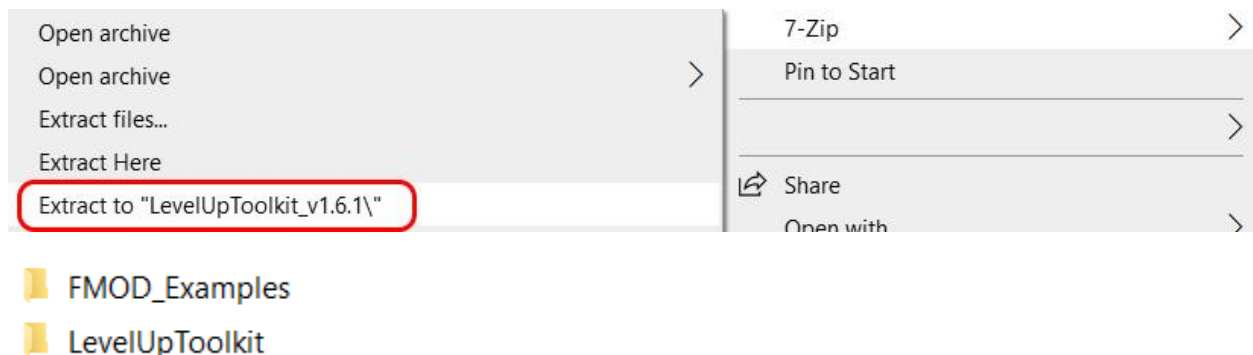
After creation is complete, **close** the project.

STEP 5: Install Level Up Toolkit Plugin

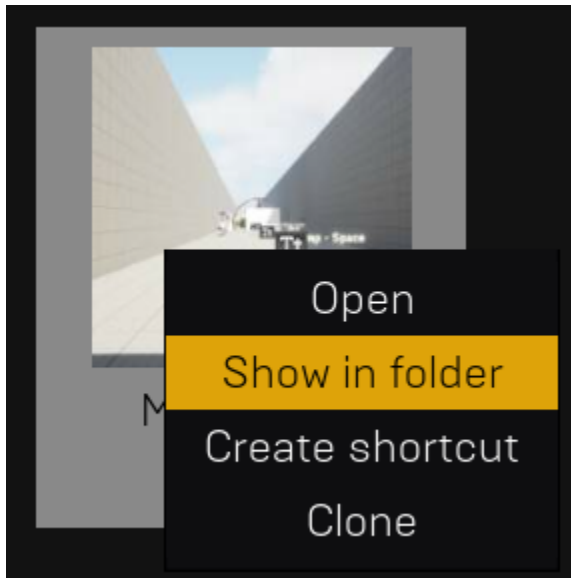
1. Download Level Up Toolkit package.

link: https://drive.google.com/file/d/1oMhHXoVFzdmWj68nBrOos_9E2J_xzzRe/view?usp=drive_link

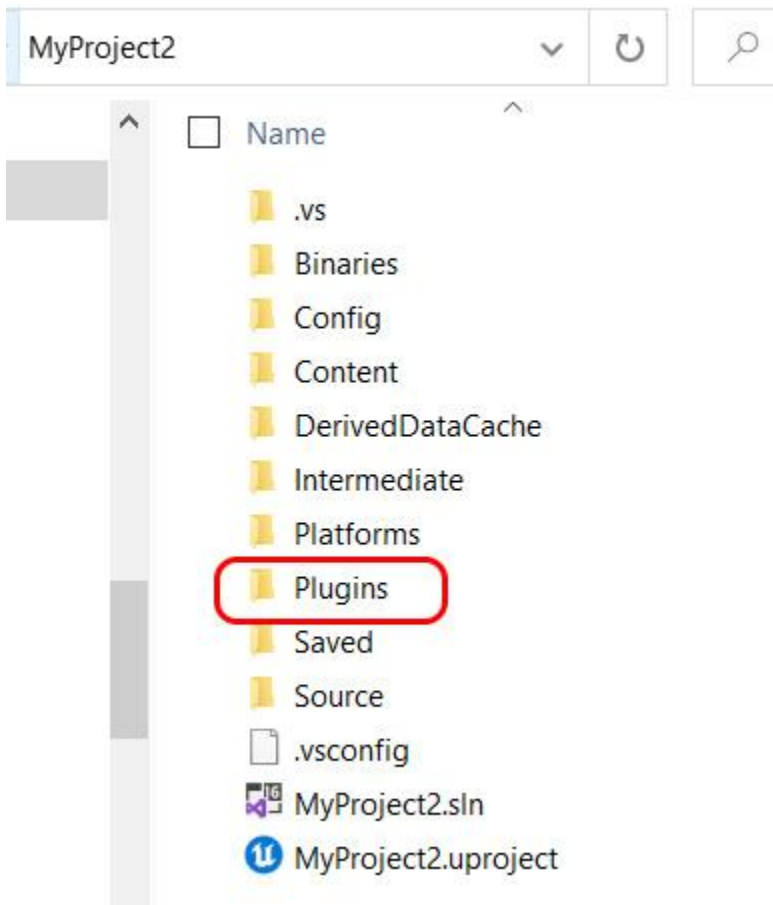
2. After downloading the zip file of our toolkit, right click the zip pack, choose **7-Zip>Extract to “LevelUpToolkit_v1.6.1\”** to extract the zip file. Then you will get two folders named ‘LevelUpToolkit’ and “FMOD_Example”.



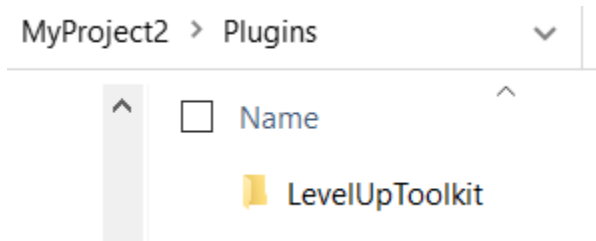
3. Open Epic Games Launcher, go to Unreal Engine> Library, find the project you just created in the **My Projects**. Right click the project, choose **Show in folder**.



4. In the project folder, create a new folder. Name it as **Plugins**.

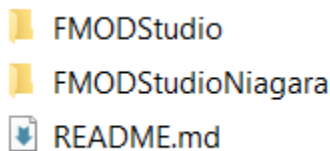


5. Drag and drop the '**LevelUpToolkit**' folder from the downloaded package to your project plugins folder.

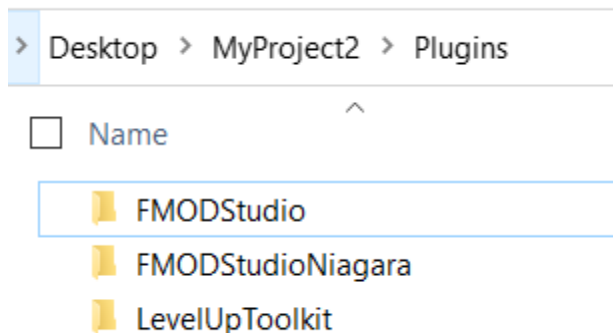


STEP 6: Install FMOD Plugin

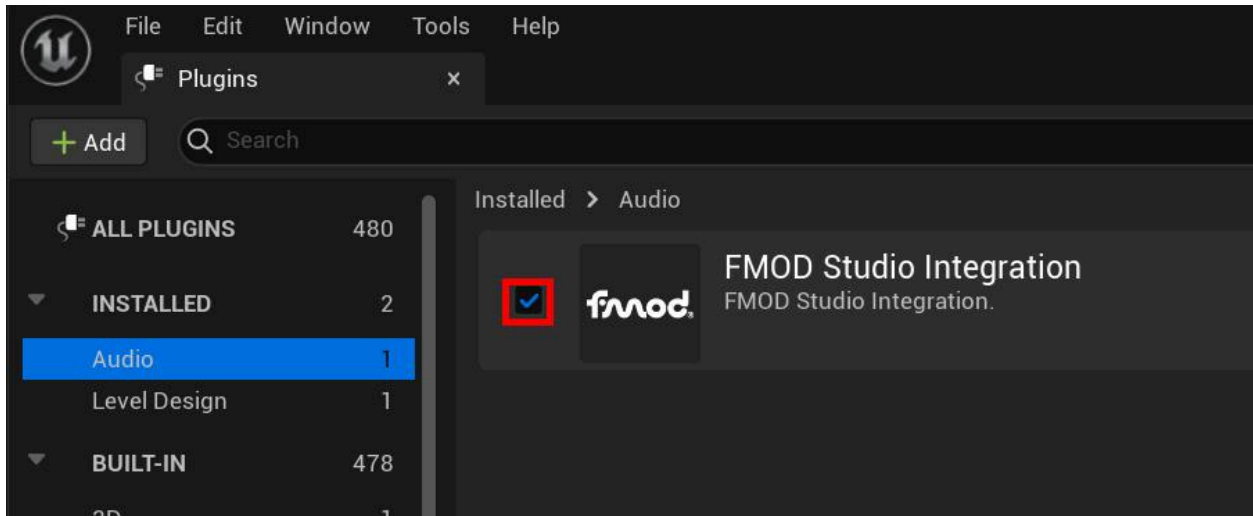
1. Extract the **fmodstudio20219ue5.1win64.zip** package, which downloaded from FMOD official website in Step3. After extracting, you will see two folders and a README note.



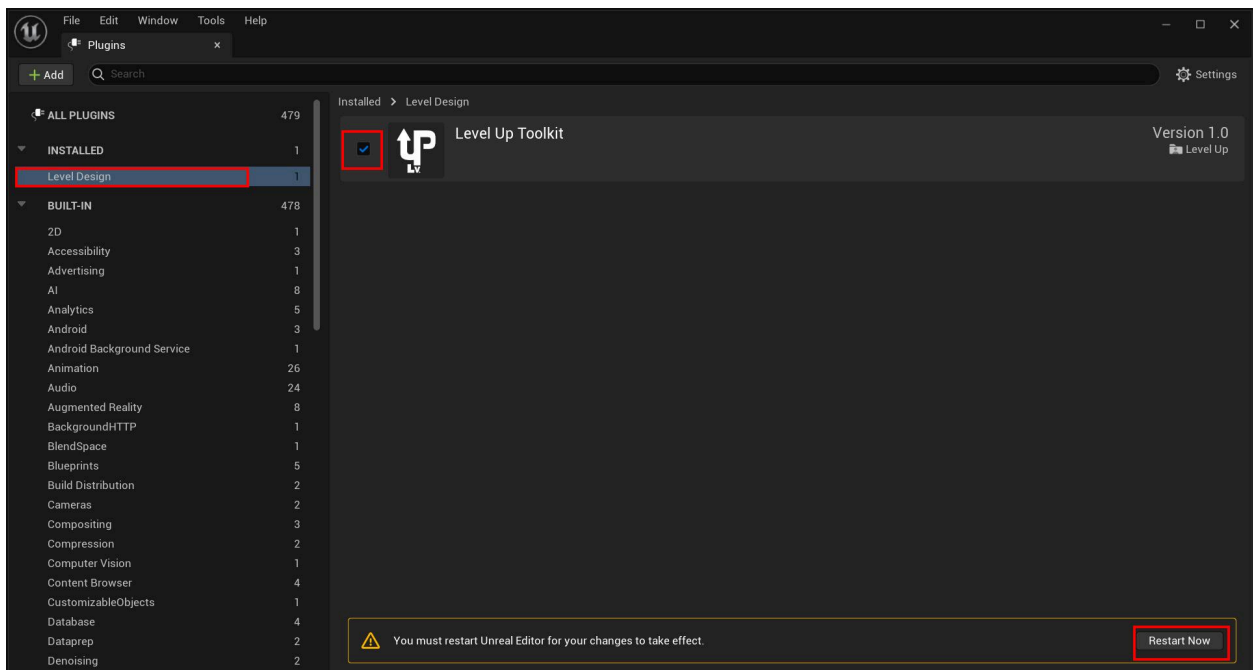
2. Drag and drop both the '**FMODStudio**' folders and the '**FMOD Studio Niagara**' folders to your project **Plugins** folder.



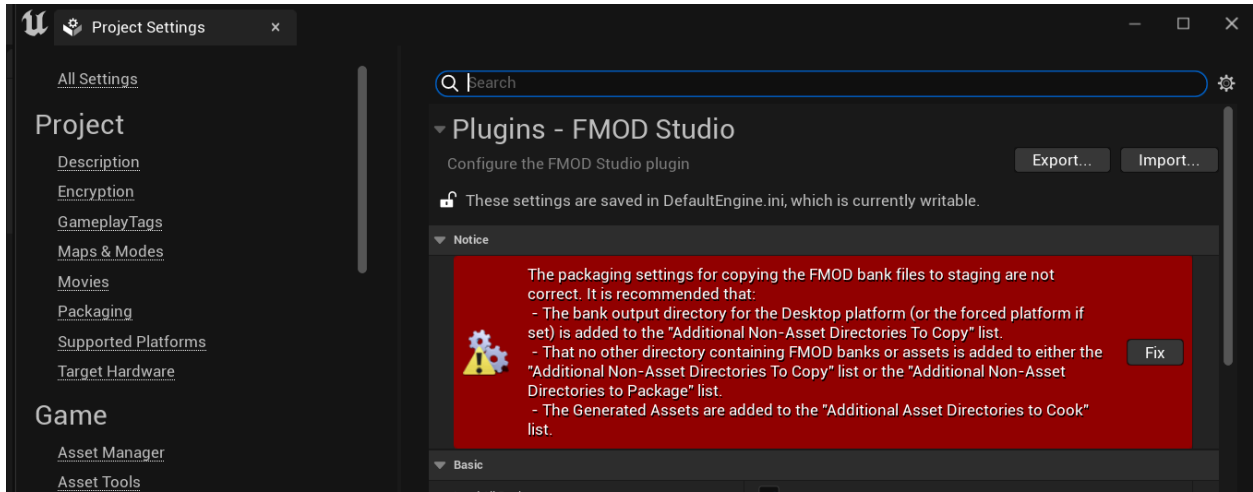
3. Open the UE5 project again. The FMOD and LevelUpToolkit plugins should be automatically checked in the plugin panel. If not, please check the **FMOD** plugin and restart the project.



Enable the level up toolkit plugin, and **restart** the project.

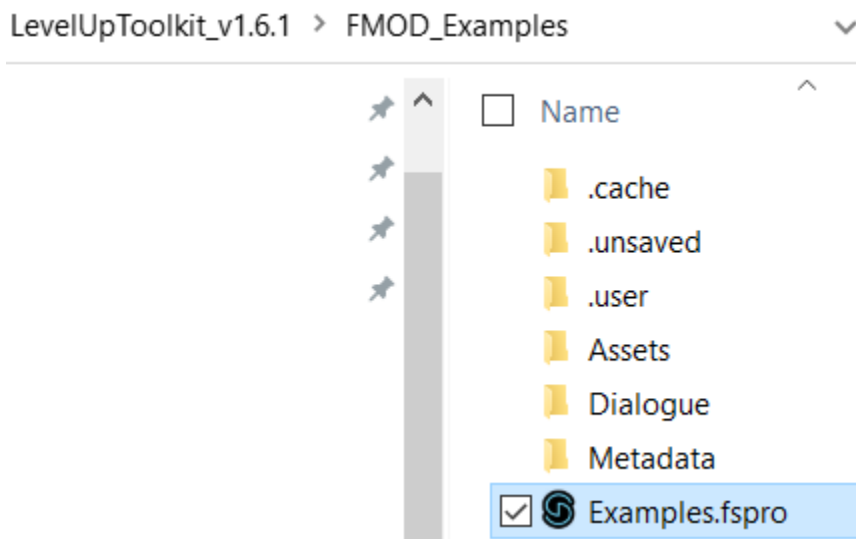


4. Open **Edit>Project Settings>Plugins>FMOD Studio**, press **Fix** button in the Notice.

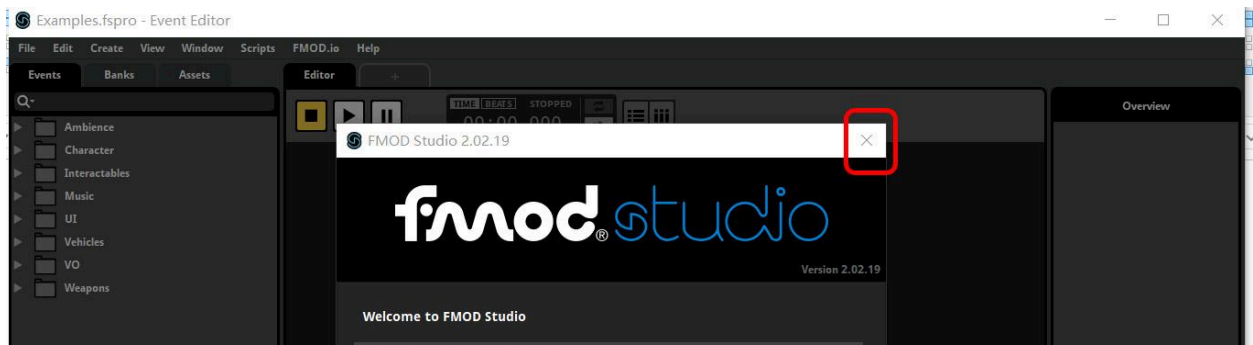


STEP 7: Bind FMOD Project to UE Project

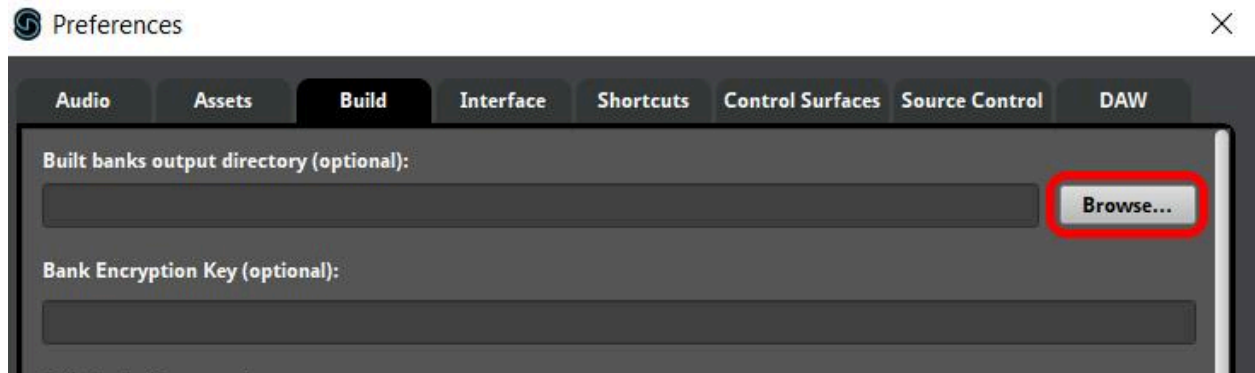
1. Open the **LevelUpToolkit_v1.6.1** folder, which is downloaded in step 5. Open **FMOD_Demo>Examples**. Open **Examples.fspro**.



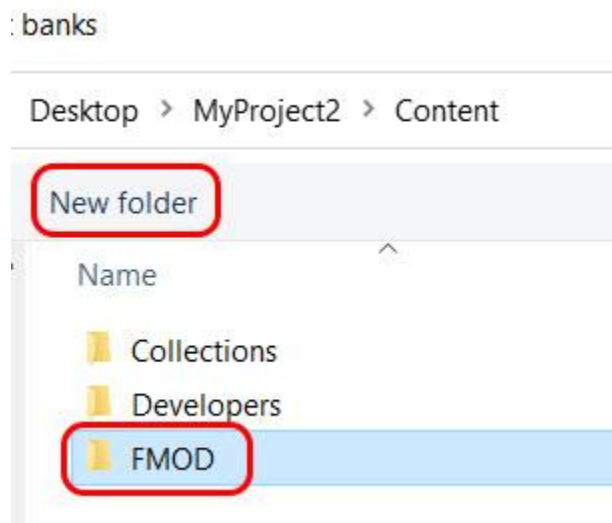
2. After opening the Fmod project, there is a welcome panel. **Close** the panel.



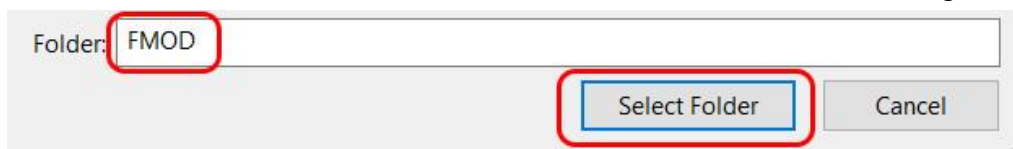
3. Choose **Edit>Preferences>Build**. In the Build banks output directory(optional), press **Browse**.



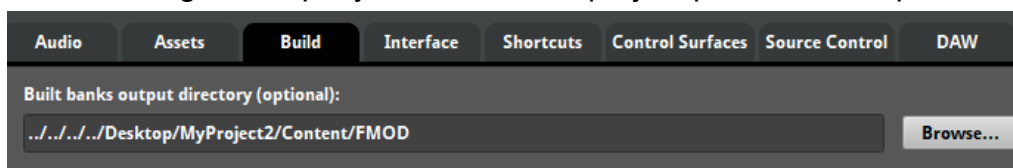
4. Choose your UE project path location, open the **Content** folder. Create a new folder named **FMOD** (the name can't be changed).



Click the **FMOD** folder, and choose **Select Folder** on the bottom right.



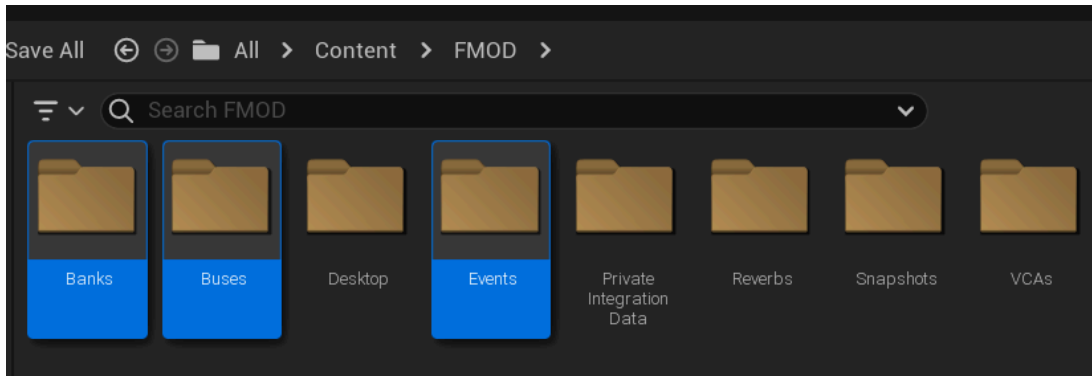
After finishing the steps, you can see the project path in the output directory.



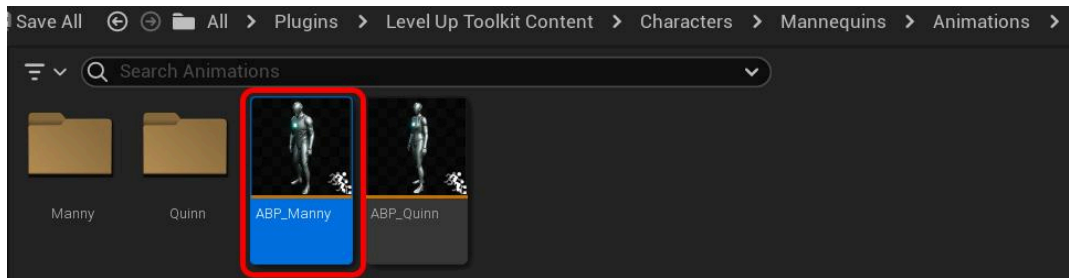
5. Close the Preferences window. In the FMOD project, choose **File>Build**. FMOD will automatically build the sound bank and the sound events into the UE project.

STEP 8: Apply Sound in UE Project

1. Open the UE project. If you can see the **Banks, Buses, Events** folders in All>Content>FMOD folder, that means the FMOD successfully binds with UE project.

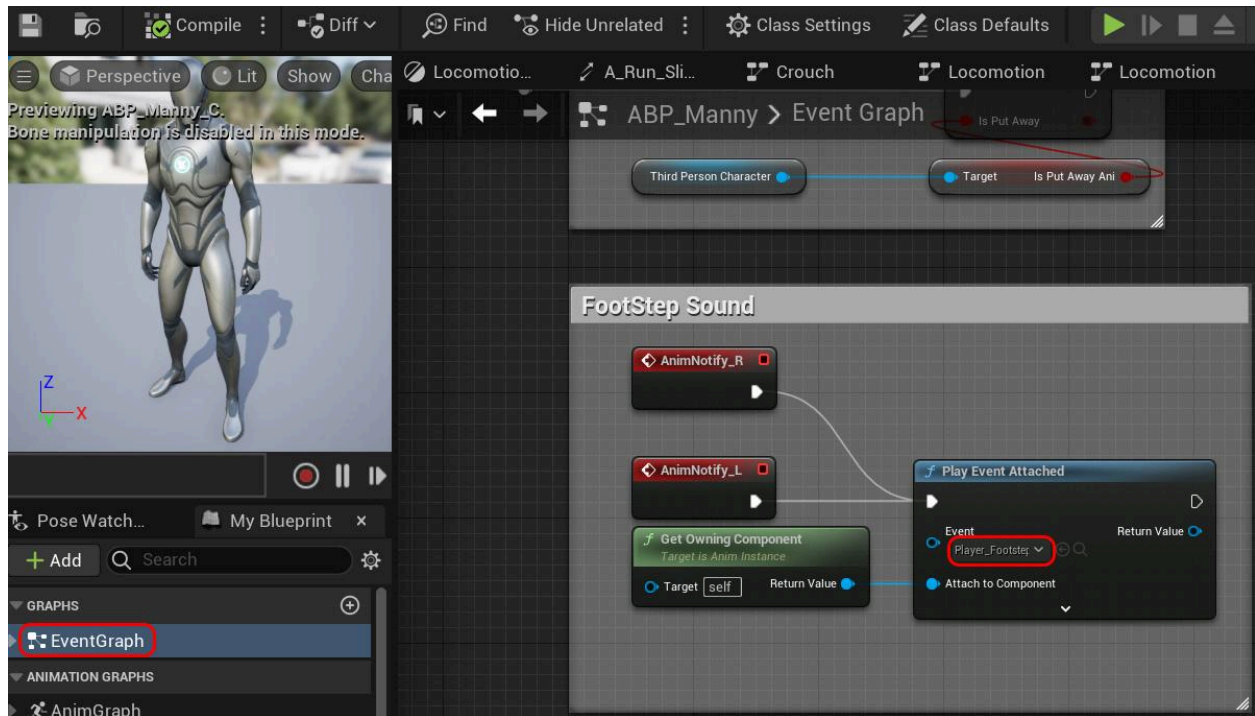


2. Locate All > Plugins > LevelUpToolkit > Content > Characters > Mannequins > Animations > ABP_Manny, double click **ABP_Manny** to open it.

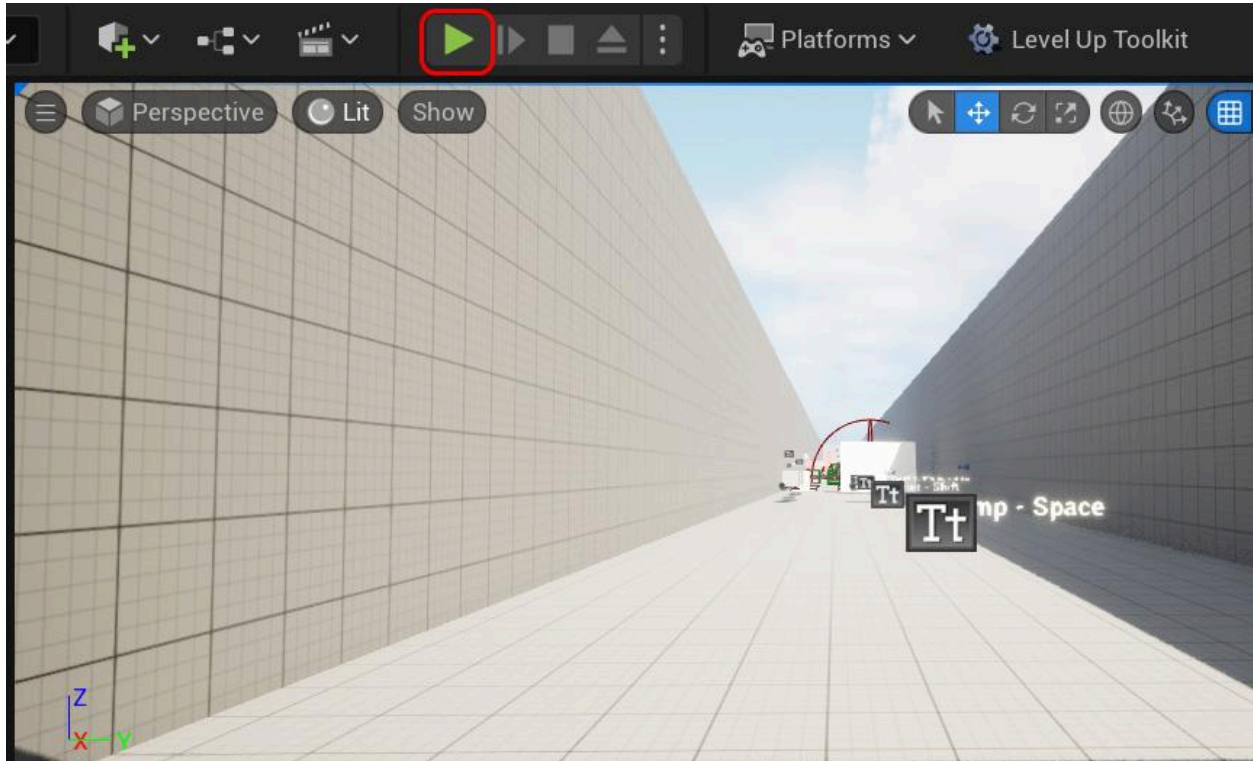


3. In the ABP_Manny animation blueprint, if you are not in the event graph page, double click the **EventGraph** on the left side.

In the Event Graph, find Footstep Sound, and then change the Event to **Player_Footsteps** in the Play Event Attached node.



4. Close the ABP_Manny animation blueprint. Navigate to **All > Plugins > LevelUpToolkit > Content > Maps > Demo**, open the **Level_Playground map**, and press the play button. If you can hear the character's footsteps when the character moves, the FMOD event has been successfully applied in the UE project. Now, you can create your own sound events.



Recommended References

FMOD Official User Guide:

<https://www.fmod.com/docs/2.02/unreal/user-guide.html>.

FMOD Studio for UE4 Video:

<https://www.youtube.com/watch?v=K64sGI9cKEg&list=PLp4vT3ssm5SW-DTHFHlotQ1IHFahcyoHJ&index=1>

FMOD vs Wwise (Part 2) | How to integrate FMOD with Unreal Engine 4

<https://www.youtube.com/watch?v=smiSKOffLn8&list=PLtjXM85CCHHoUo6g3yjh1MmJtoLVgljfo&index=2>