

Enemy Artificial Intelligence

13th Week, 2021

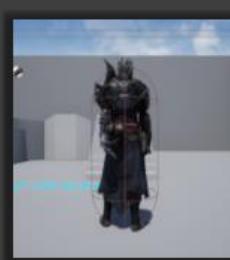
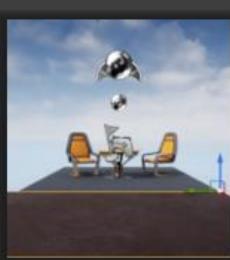
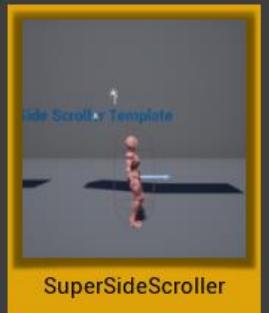


UNREAL
ENGINE



Select or Create New Project

Recent Projects



More

New Project Categories



Games

Start your game development journey with one of our key classes, levels, and examples.



Film, Television, and Live Events

Choose from templates and examples for nDisplay, VR Scouting, and virtual production workflows.



Architecture, Engineering, and Construction

Select a starting point for multi-user design reviews, photorealistic architectural design visualizations, sunlight studies, or stylized renderings.

Open Project

Cancel



Enemy AI (1)

- › **AI** (Artificial Intelligence)
 - An entry that is aware of its environment and performs choices that will help optimally achieve its intended purpose
 - Uses **finite state machines** to switch between more than one state based on the input it receives from the user or its environment
- › Interesting and fun AI is crucial to any game, and depending on the game you are making, this can mean a very complex or very simplistic AI.



Enemy AI (2)

- › In **SuperSideScroller** game, how the enemy will behave:
 - A very simple enemy that has a basic back and forth movement pattern and will not support any attacks; only by colliding with the player character will they be able to inflict any damage.
 - Need to set the locations to move between for the enemy AI
 - Decide whether the AI should change locations, should constantly move between locations, or should there be a pause in between selecting a new location to move to?



AI Controller

- › A **Player Controller** and an **AI Controller**
 - Both of these actors derive from the base **Controller** class
 - › A Controller is used to take control of a **Pawn** or **Character**
- › The main difference between a Player Controller and an AI Controller
 - A Player Controller relies on the input of an actual player
 - An AI Controller applies AI to the characters and responds to the environment based on the rules set forth by the AI.
- › Multiple instances of the same AI pawn can share the same AI Controller, and the same AI Controller can be used across different AI pawn classes.



Auto Possess AI (1)

- › The AI Controller must possess a *pawn*

- To possess a pawn:

```
void AController::Possess(APawn* InPawn)
```

- To unpossess a pawn:

```
void AController::UnPossess()
```

- Also `void AController::OnPossess(APawn* InPawn)` and `void AController::OnUnPossess()` functions are called whenever the `Possess()` and `UnPossess()` functions are called, respectively.



Auto Possess AI (2)

- › In Unreal Engine 4, there are two methods in which AI Pawns or Characters can be possessed by an AI Controller.
 - **Placed in World:** How you will be handling AI
 - › You will manually place these enemy actors into your game world, and the AI will take care of the rest once the game begins.
 - **Spawned:** A little more complicated
 - › Requires an explicit function call, either C++ or Blueprint, to **Spawn** an instance of a specified class
 - **Placed in World or Spawned:** A safe option if you are unsure of which method you want to use
 - › Both methods are supported



Exercise 13.01: Implementing AI Controllers

The screenshot shows the Unreal Engine Content Browser interface. On the left, the Content Browser tree view is open, showing a hierarchy of assets. The 'Content' folder is selected and highlighted with a red box. Under 'Content', there is an 'Enemy' folder, which is also highlighted with a red box. To the right of the tree view is a search bar with the placeholder text 'Search Enemy'. A context menu is open over the 'Enemy' folder, listing options such as 'New Folder', 'Import Asset', 'Create Basic Asset' (with sub-options like 'Blueprint Class', 'Level', 'Material', and 'Particle System'), and 'Create Advanced Asset' (with sub-options like 'Animation', 'Artificial Intelligence', 'Blendables', 'Blueprints', etc.). At the bottom of the Content Browser, it says '2 items'. In the center of the screen, the 3D Editor Viewport displays a character model with a bounding box and a blue arrow indicating movement or rotation. The title 'Side Scroller Template' is visible in the background. On the far right, a vertical panel lists various assets: 'Ledge1' through 'Ledge9', 'Floor', and 'StaticMeshActor' instances. Below this list, it says '26 actors'. At the bottom right of the screen, there are 'Details' and 'World Settings' buttons, and a message 'Select an object to view details.'



This screenshot shows the Content Browser panel from the Unreal Engine 4 Editor. The top bar includes "Add/Import", "Save All", and navigation buttons. The Content Browser lists several asset categories: Content, Enemy, Geometry, MainCharacter, Mannequin, SideScrollerCPP, StarterContent, and C++ Classes. Under the "Content" category, there are three folders: "AI" (highlighted with a yellow background and a red arrow pointing to it), "Blueprints", and "Materials". The status bar at the bottom indicates "3 items (1 selected)".

SideScrollerExampleMap DDC SuperSideScroller

File Edit Window Help

Place Actors

Search Classes

Recently Placed

Basic

Lights

Cinematic

Visual Effects

Geometry

Volumes

All Classes

Content Browser

Add/Import Save All

Output Log

Content

Enemy

AI

Blueprints

Materials

Geometry

MainCharacter

Animation

Blueprints

Mesh

Mannequin

SideScrollerCPP

0 items

Save Current Source Control Modes Content Marketplace Settings Blueprints Cinematics Build Compile

Perspective Lit Show

New Folder Import Asset Create Basic Asset Blueprint Class Level Material Particle System

Blueprints are special assets that provide an intuitive, node-based interface that can be used to create new types of Actors and script level events; giving designers and gameplay programmers the tools to quickly create and iterate gameplay from within Unreal Editor without ever needing to write a line of code.

hold (Ctrl + Alt) for more

Drop files here or right click to create content.

World Outliner

Label Type

SideScrollerExampleMap (World) Folder

ArenaGeometry Folder

Ledges StaticMeshActor

Ledge1 StaticMeshActor

Ledge2 StaticMeshActor

Ledge3 StaticMeshActor

Ledge4 StaticMeshActor

Ledge5 StaticMeshActor

Ledge6 StaticMeshActor

Ledge7 StaticMeshActor

Ledge8 StaticMeshActor

Ledge9 StaticMeshActor

Floor StaticMeshActor

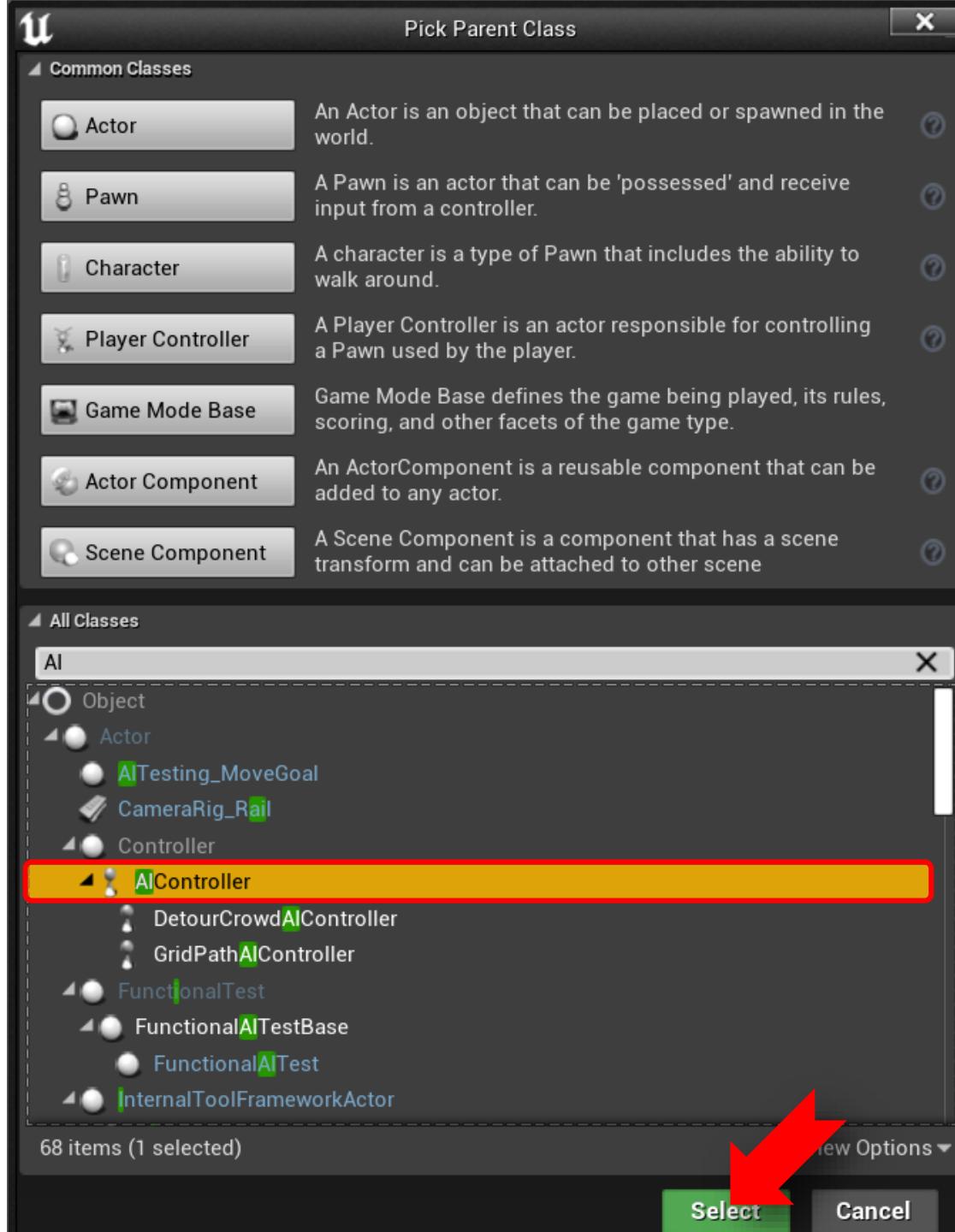
26 actors View Options

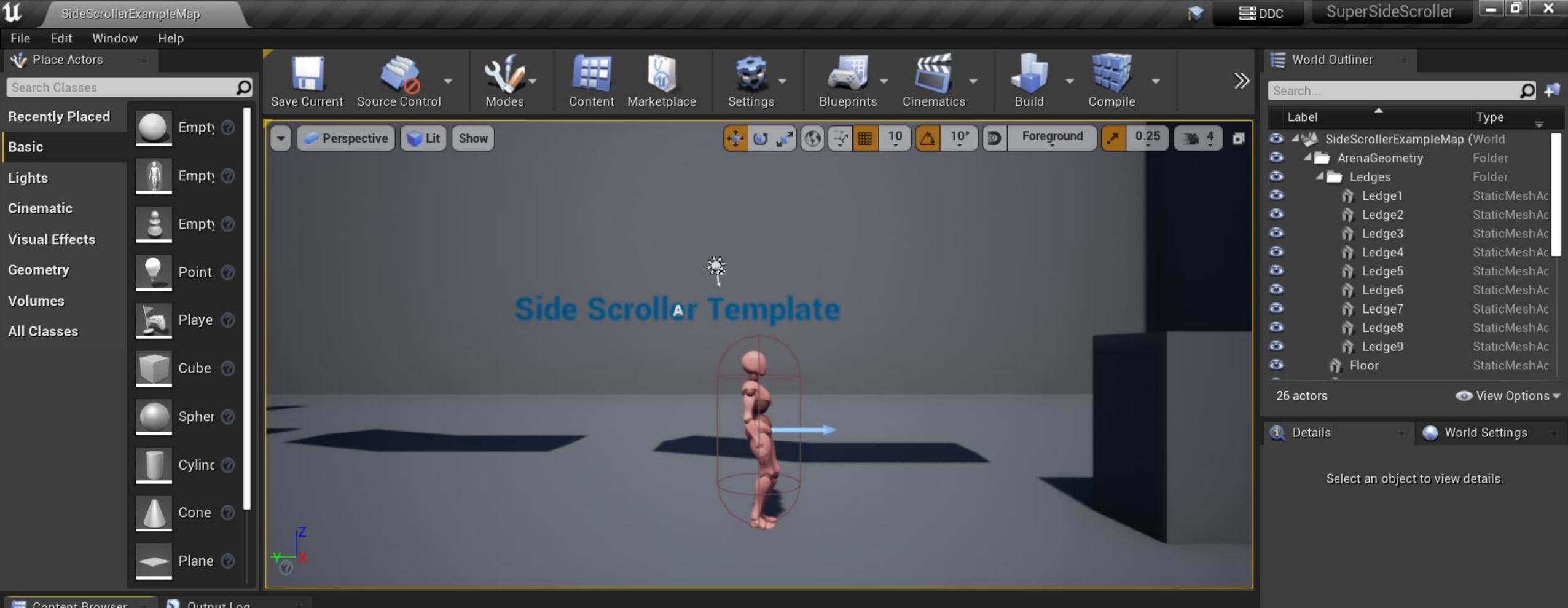
Details World Settings

Select an object to view details.

Right-Click

View Options

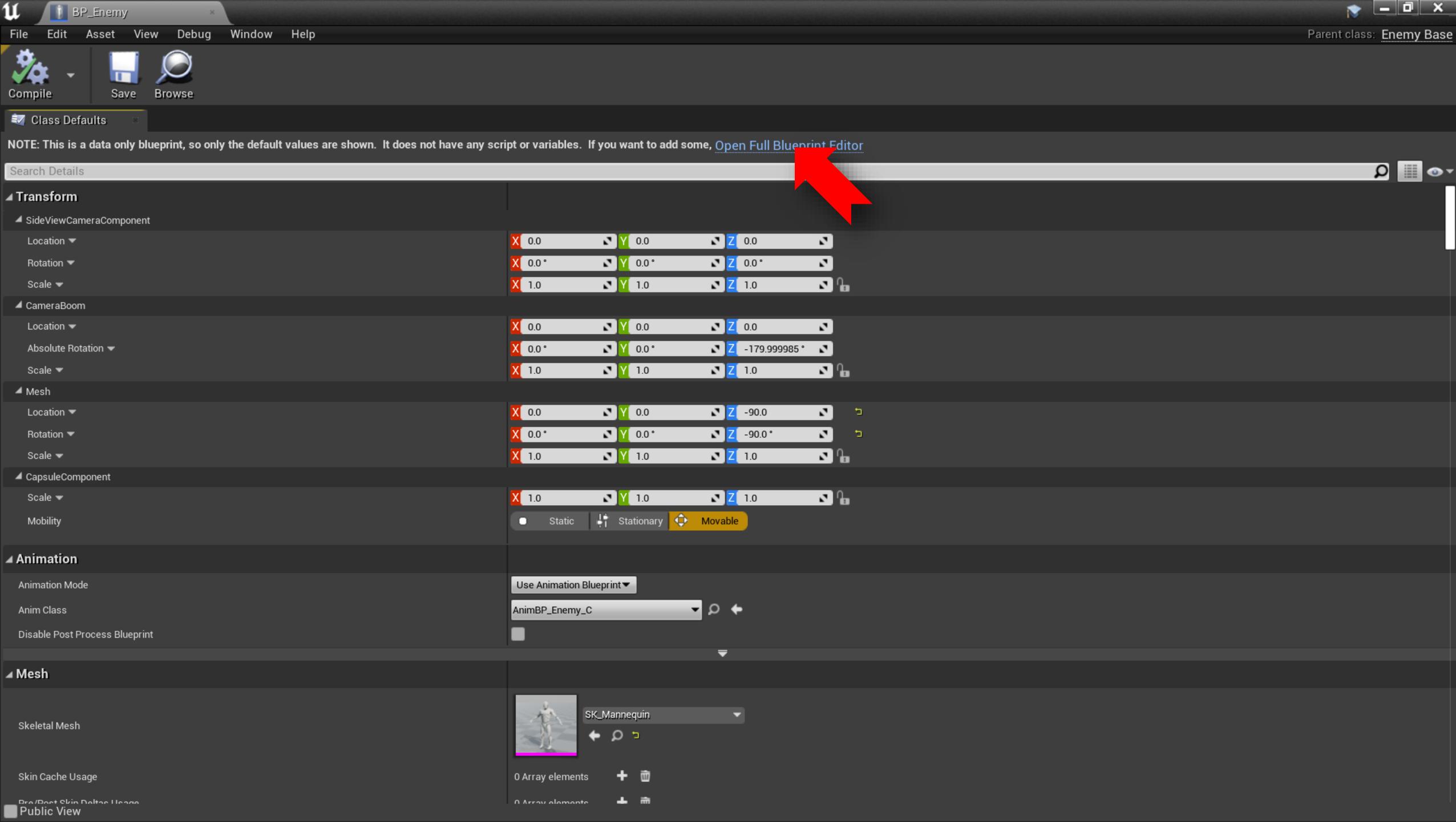


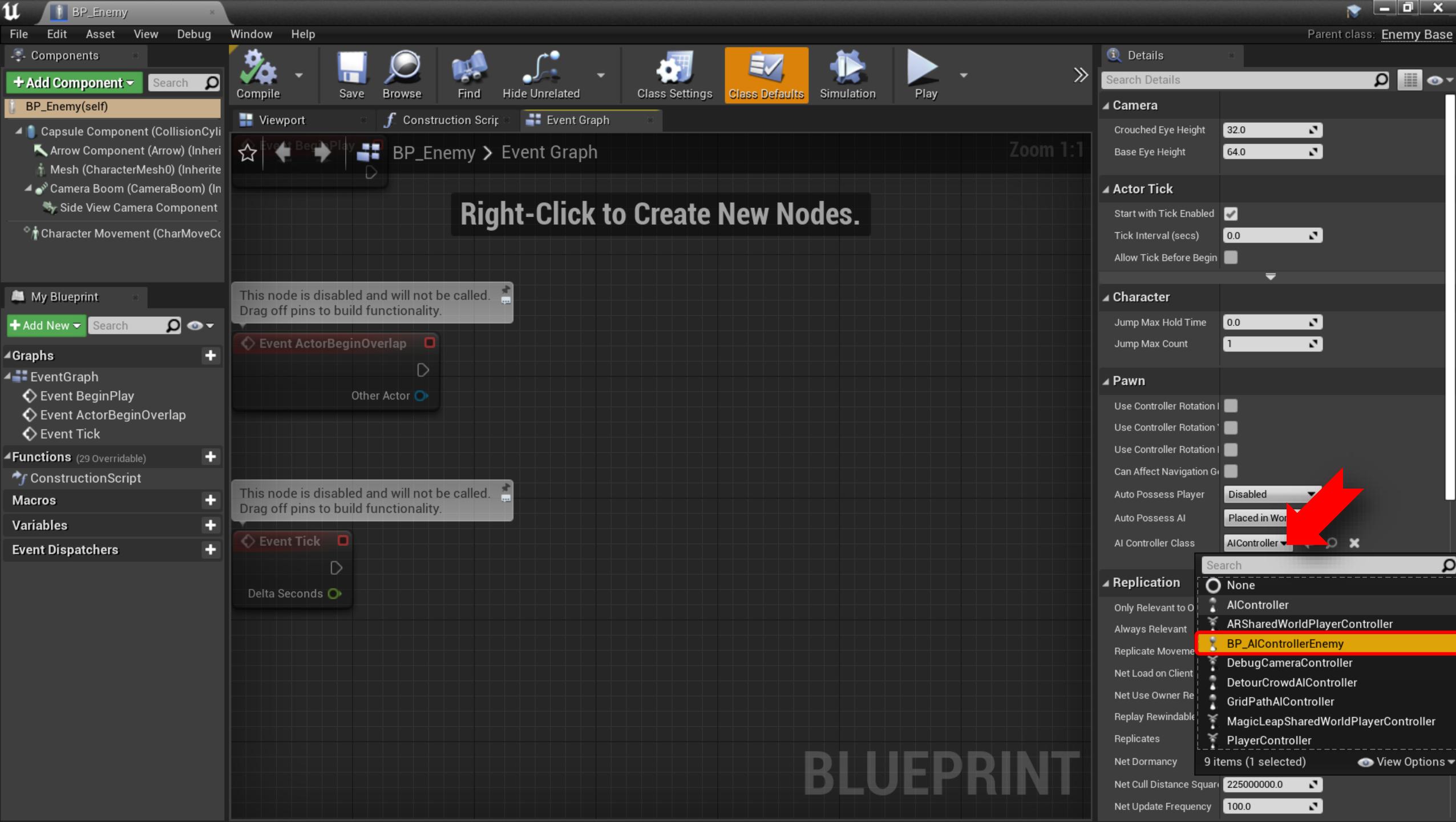


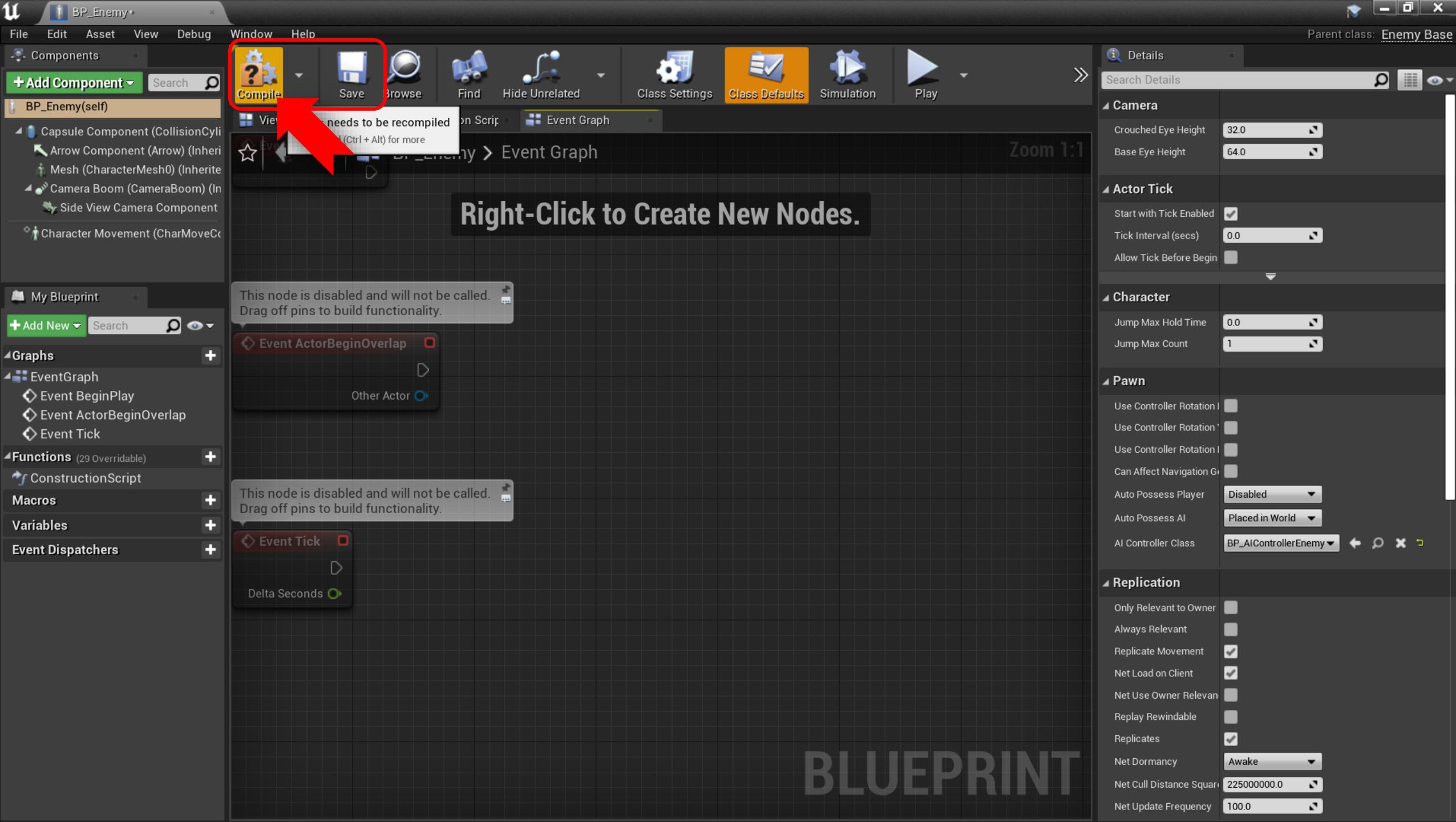
This screenshot shows the Content Browser panel of the Unreal Engine 4 Editor. The left sidebar displays the file tree under the "Content" folder, with the "Enemy" and "AI" subfolders expanded. The "BP_AIController" item is highlighted with a yellow border. The main area shows the "Filters" dropdown and a search bar with the placeholder "Search AI". Below the search bar, there is a message stating "1 item (1 selected)".



The screenshot shows the Content Browser panel. The left sidebar lists Content, Enemy (selected), AI, Blueprints, Materials, Geometry, MainCharacter, Animation, Blueprints, Mesh, Mannequin, and SideScrollerCPP. The main area displays two items: AnimBP_Enemy and BP_Enemy. A red arrow points to the BP_Enemy thumbnail, which is highlighted with a yellow border. The status bar at the bottom indicates "2 items (1 selected)".





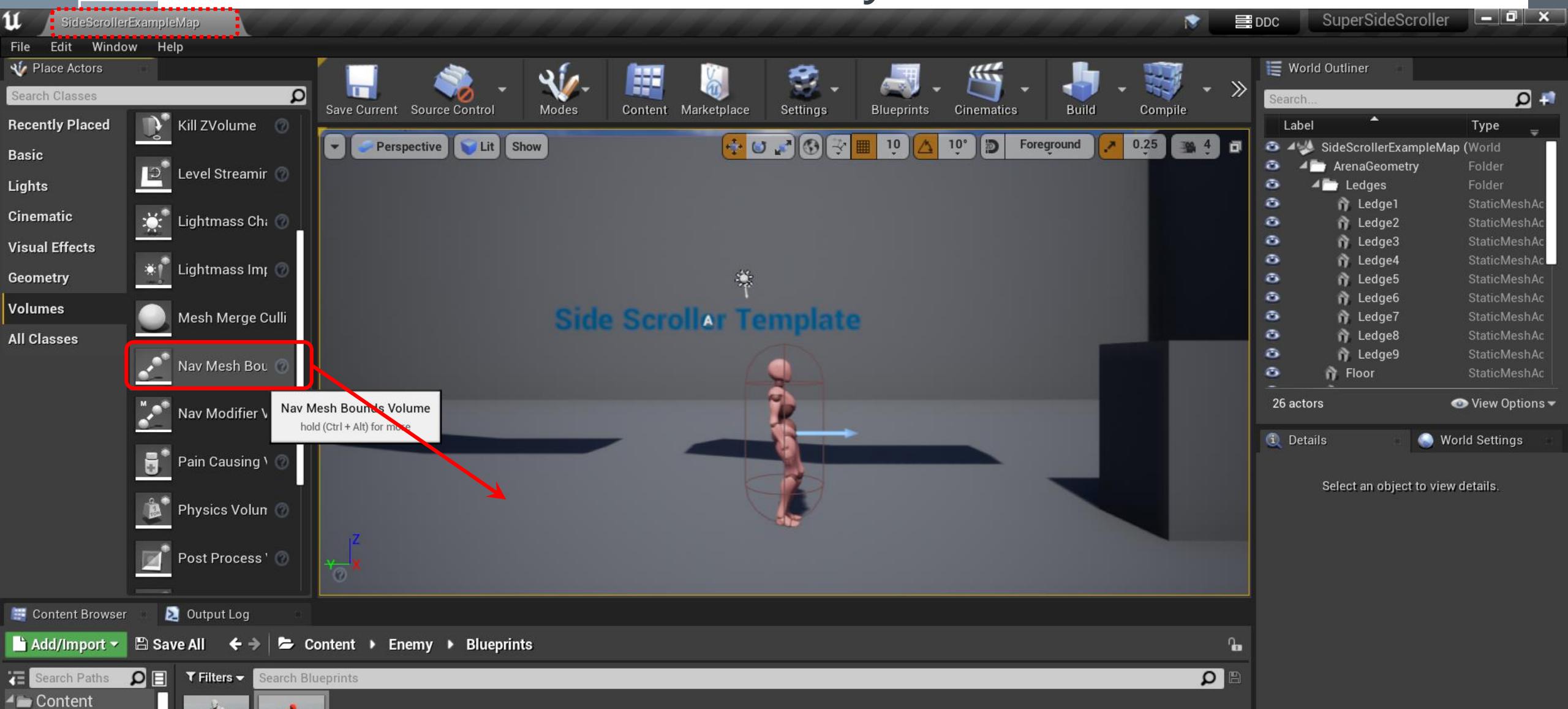


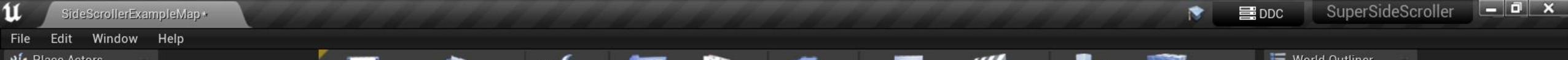


Navigation Mesh

- › One of the most crucial aspects of any AI, especially in video games, is the ability to navigate the environment in a sophisticated manner.
- › We will need a **Navigation Mesh** or **Nav Mesh** in our world, so that our AI can effectively navigate the playable bounds of the game world.
- › Unreal Engine 4 also support a **Dynamic Navigation Mesh**, which allows the Nav Mesh to update in real-time as dynamic objects move around the environment.

Exercise 13.02: Implementing a Nav Mesh Volume for the AI Enemy



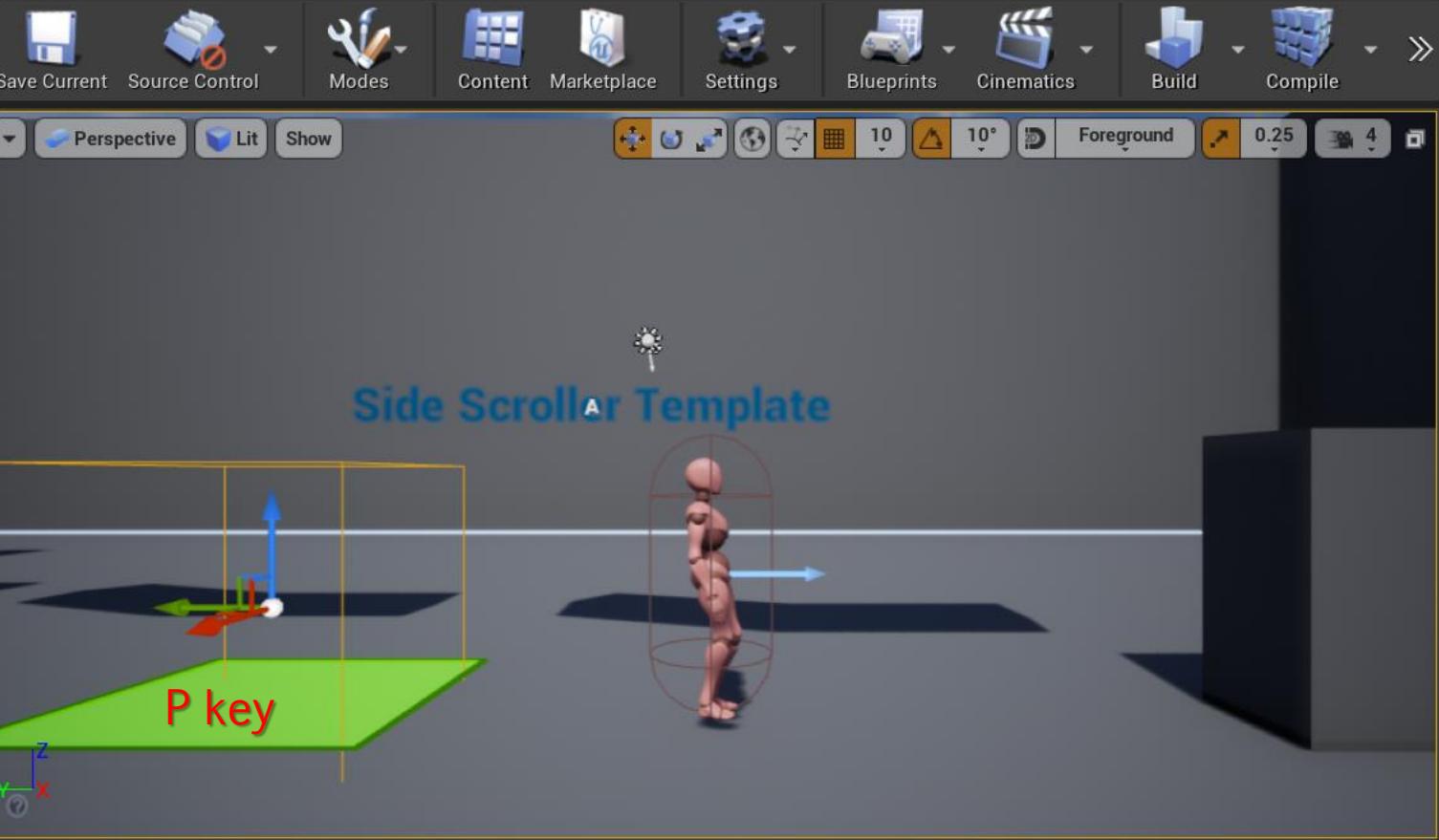


File Edit Window Help

Place Actors

Search Classes

Recently Placed	Kill ZVolume
Basic	Level Streamir
Lights	Lightmass Ch:
Cinematic	Lightmass Imp
Visual Effects	Mesh Merge Culli
Geometry	Nav Mesh Bou
Volumes	Nav Modifier V
All Classes	Pain Causing \
	Physics Volun
	Post Process'



World Outliner		
Search...		
Label	Type	
SideScrollerExampleMap (EdWorld)	Folder	
ArenaGeometry	Folder	
Lighting	Folder	
RenderFX	Folder	
AtmosphericFog	AtmosphericFog	
SphereReflectionCaptur	SphereReflect	
NavMeshBoundsVolume	NavMeshBound	
NetworkPlayerStart	PlayerStart	
RecastNavMesh-Default	RecastNavMe	
SideScrollerCharacter	Edit BP_Sup	
SkySphereBlueprint	Edit BP_Sky	
TemplateLabel	TextRenderAct	

28 actors (1 selected) View Options

Details World Settings

NavMeshBoundsVolume

Search Details

Transform

Location	1200.0	-210.0	180.0
Rotation	0.0 °	0.0 °	0.0 °
Scale	1.0	1.0	1.0
Mobility	Stat	Stat	Mov

Navigation

Supported Agents all

Fill Collision Under

Collision

Simulation Genera

Phys Material Ove

None

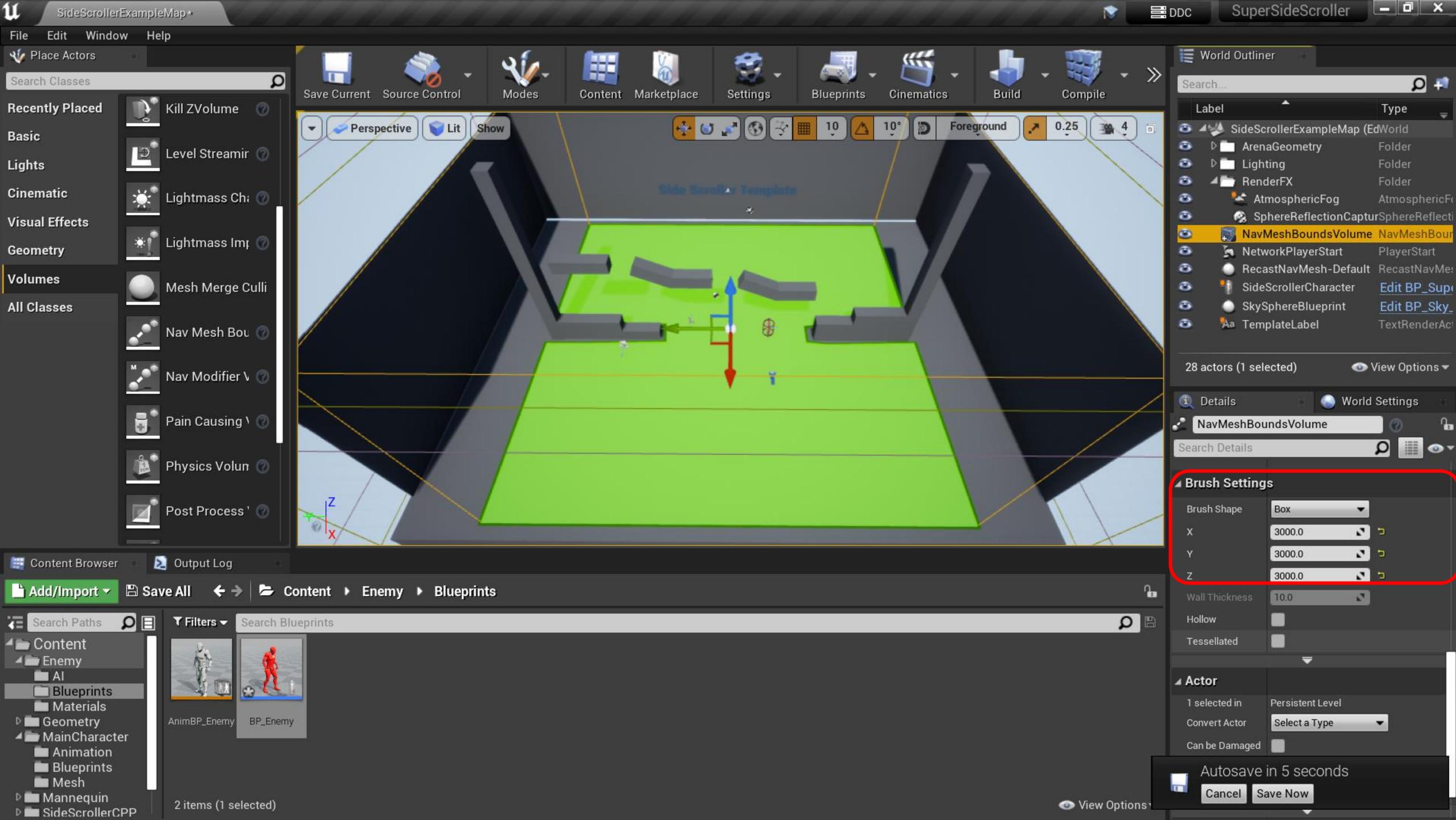
Generate Overlap

Add/Import Save All Content > Enemy > Blueprints

Content	Filters	Search Blueprints
Content		
Enemy		
AI		
Blueprints		
Materials		
Geometry		
MainCharacter		
Animation		
Blueprints		
Mesh		
Mannequin		
SideScrollerCPP		

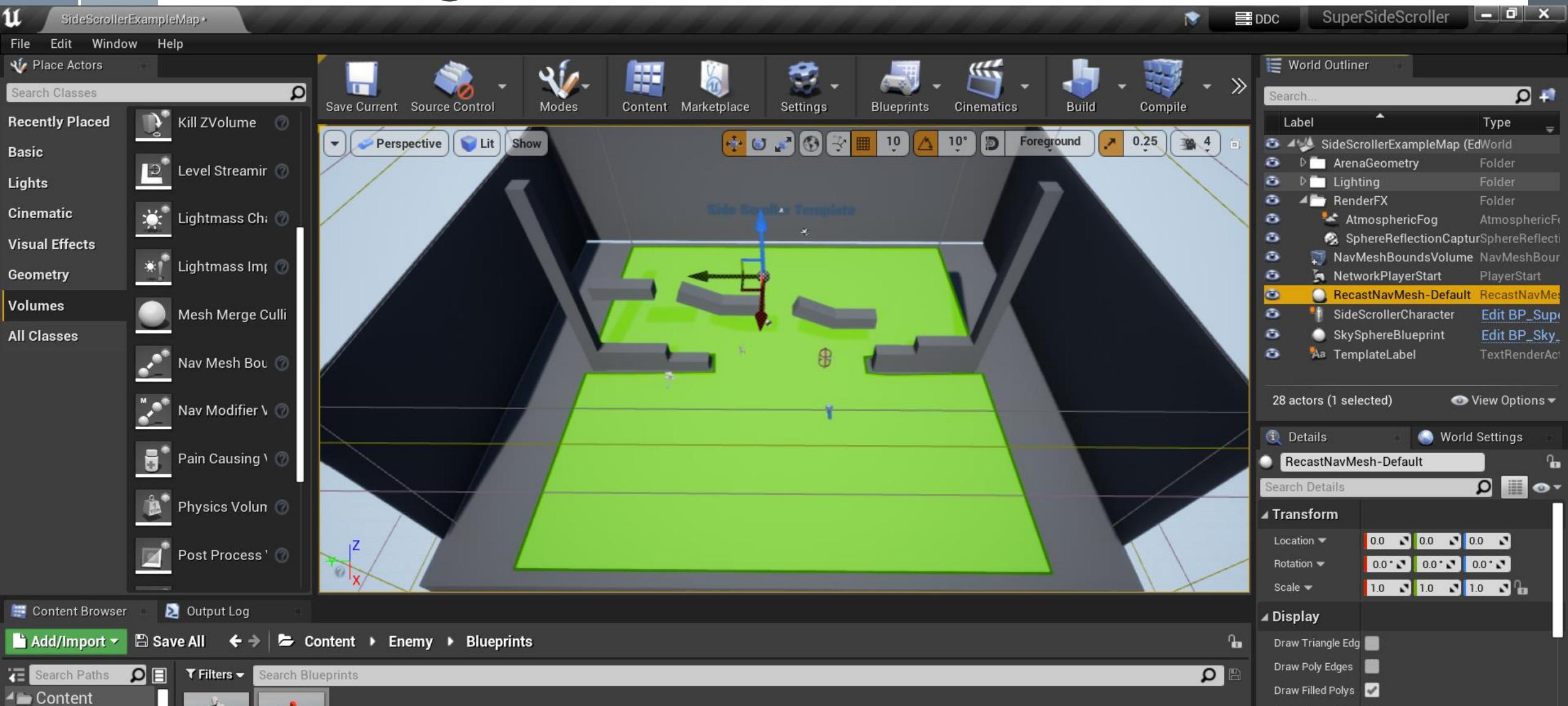
2 items (1 selected)

View Options





Recasting The Nav Mesh (1)





Recasting The Nav Mesh (2)

- › The **RecastNavMesh** acts as the “brain” of the Nav Mesh because it contains the parameters needed to adjust the Nav Mesh that directly influences how the AI navigates the given area.
 - **Display** (visual debug display of generated navigable area)
 - **Generation**
 - › **Cell Size**
 - › **Agent Radius**
 - › **Agent Height** (the Half Height of the collision component)
 - › **Agent Max Slope** (by default, 44 degrees)
 - › **Agent Max Step Height**



Exercise 13.03: Recasting Nav Mesh Volume Parameters

The screenshot displays the Unreal Engine Editor interface, specifically the 3D level editor. A green navigation mesh is visible, representing the pathfinding grid. A specific area of the mesh is highlighted with a red dashed box, indicating a region of interest or a problem area. The 'Content Browser' at the bottom left shows two enemy blueprints: 'AnimBP_Enemy' and 'BP_Enemy'. The 'Details' panel on the right shows parameters for 'RecastNavMesh-Default', with 'Cell Size' set to 5.0 and 'Agent Height' set to 192.0, both of which are highlighted with red boxes.

Content Browser:

- Add/Import
- Save All
- Content > Enemy > Blueprints

Content Browser Details:

- Content
- Enemy
- AI
- Blueprints
- Materials
- Geometry
- MainCharacter
- Animation
- Blueprints
- Mesh
- Mannequin
- SideScrollerCPP

Selected Items: 2 items (1 selected)

World Settings:

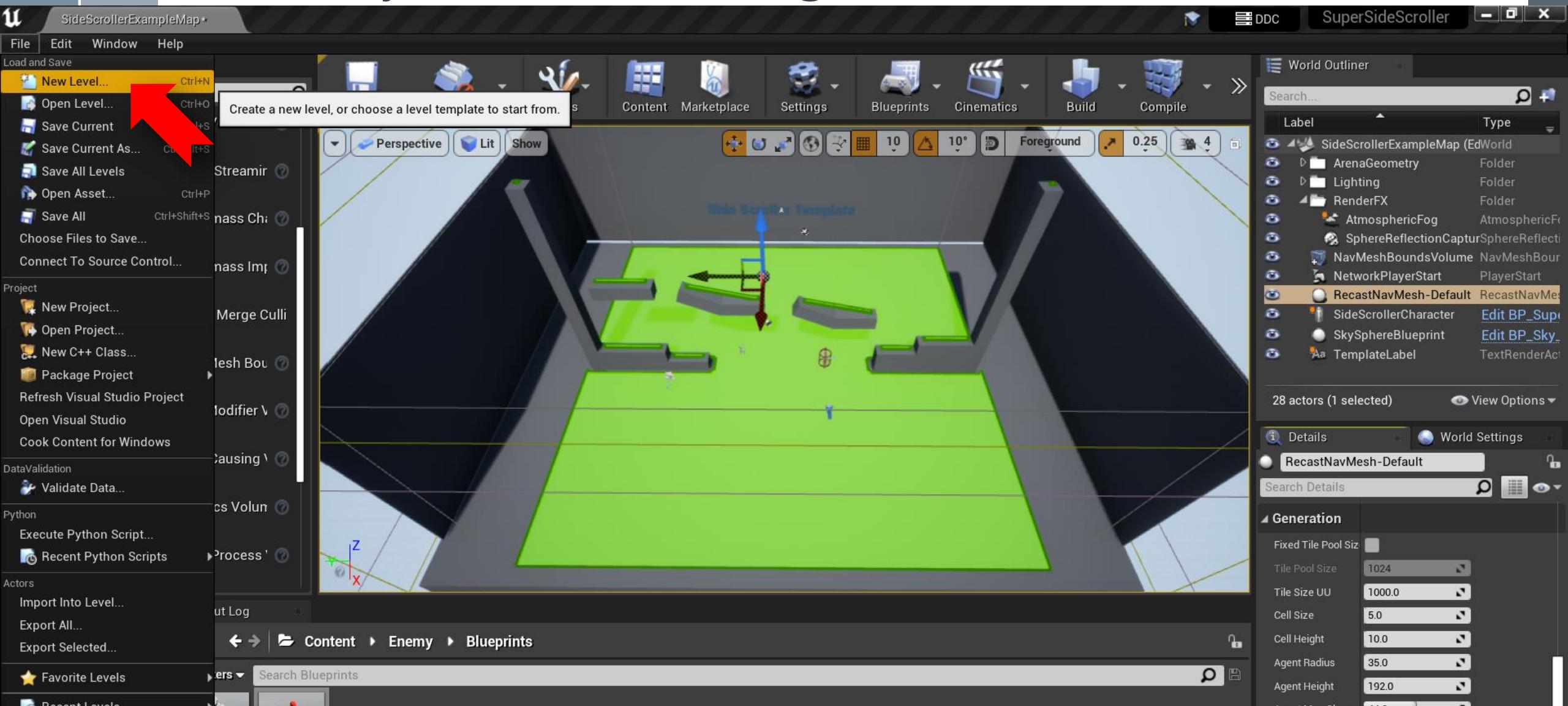
- RecastNavMesh-Default
- SideScrollerCharacter
- SkySphereBlueprint
- TemplateLabel

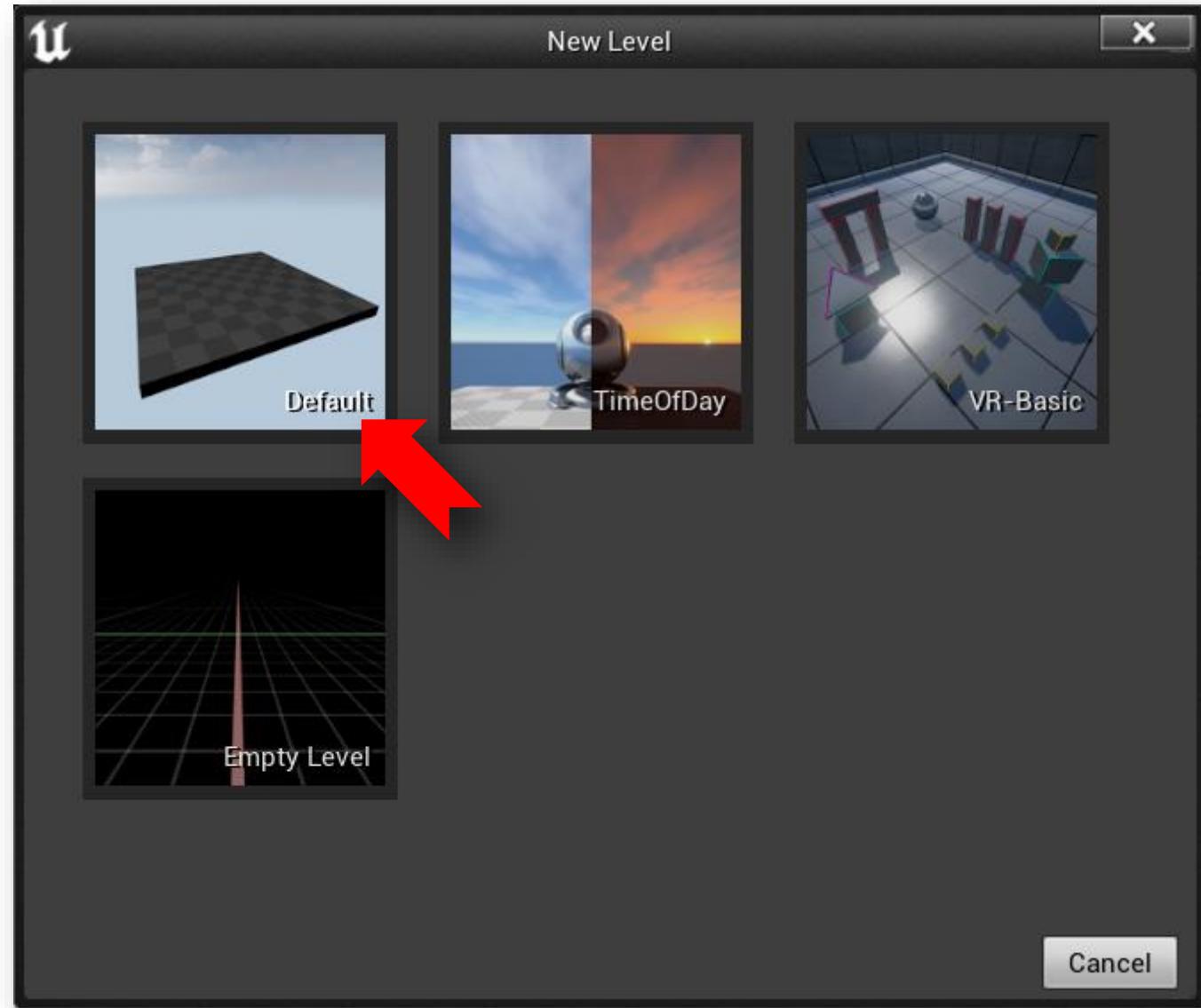
Details Panel (RecastNavMesh-Default):

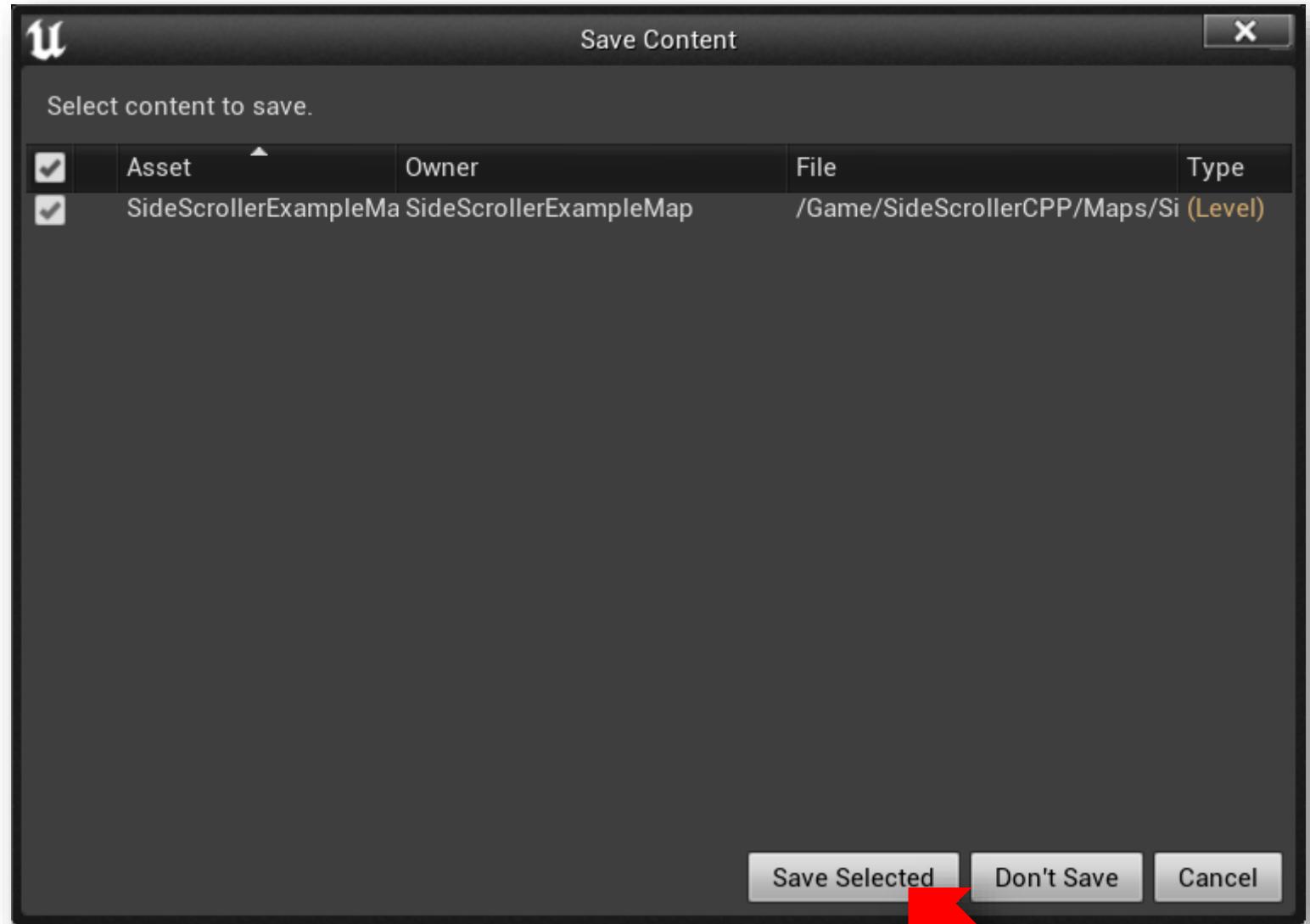
Parameter	Value
Cell Size	5.0
Cell Height	10.0
Agent Radius	35.0
Agent Height	192.0
Agent Max Slope	44.0
Agent Max Step He	35.0
Min Region Area	0.0
Merge Region Size	400.0
Max Simplification	1.3
Sort Navigation Ar	

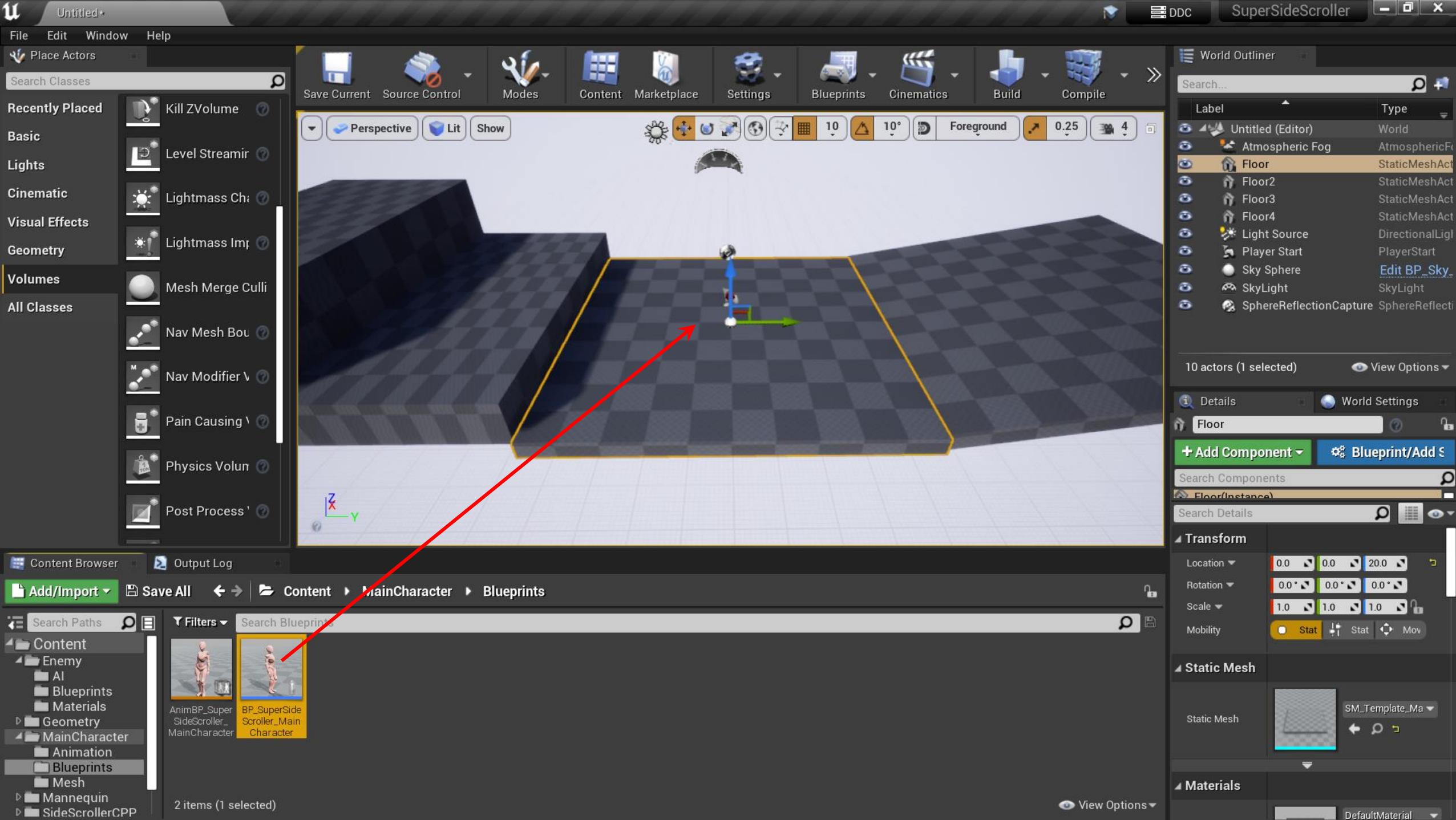


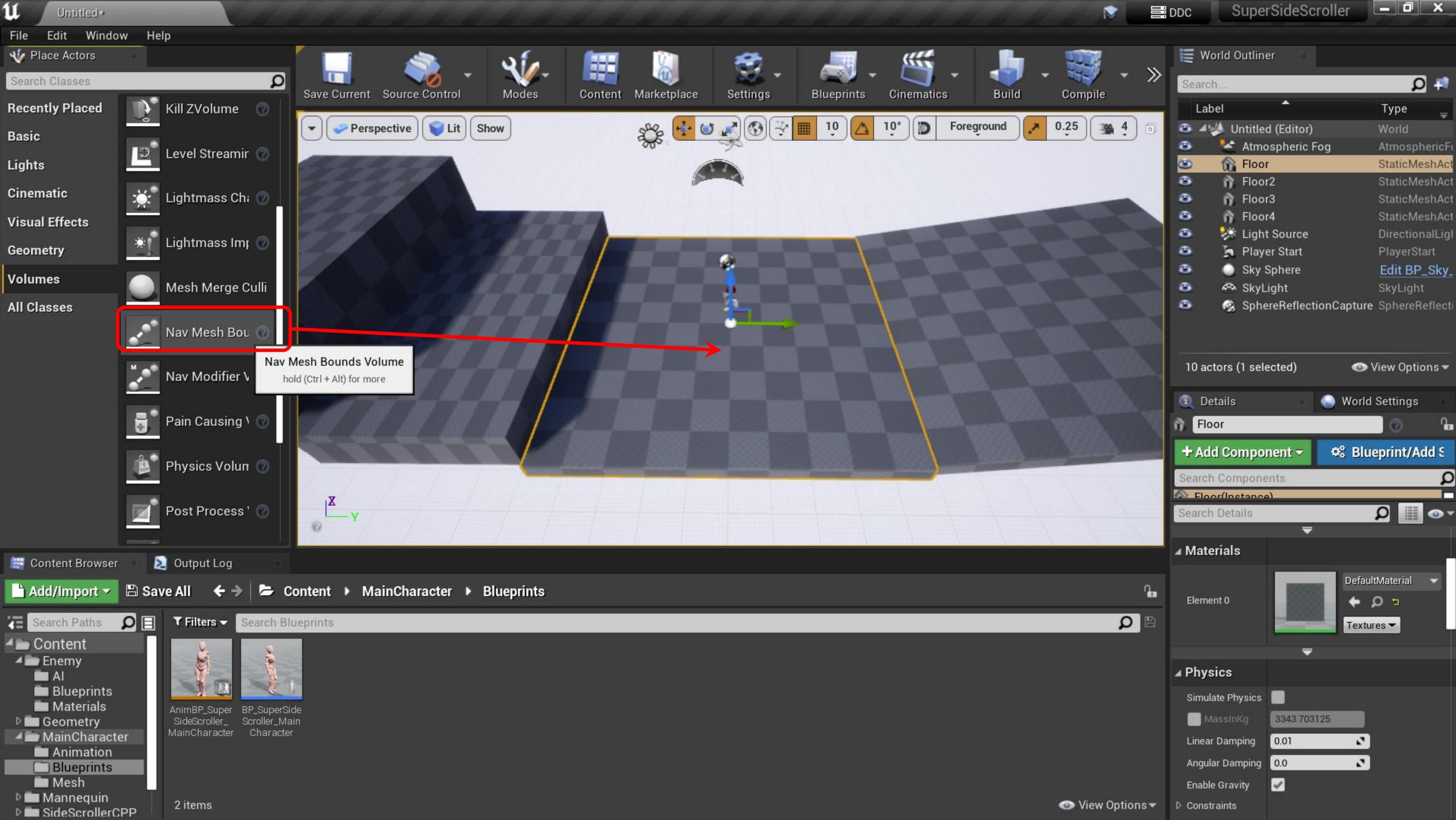
Activity 13.01: Creating a New Level

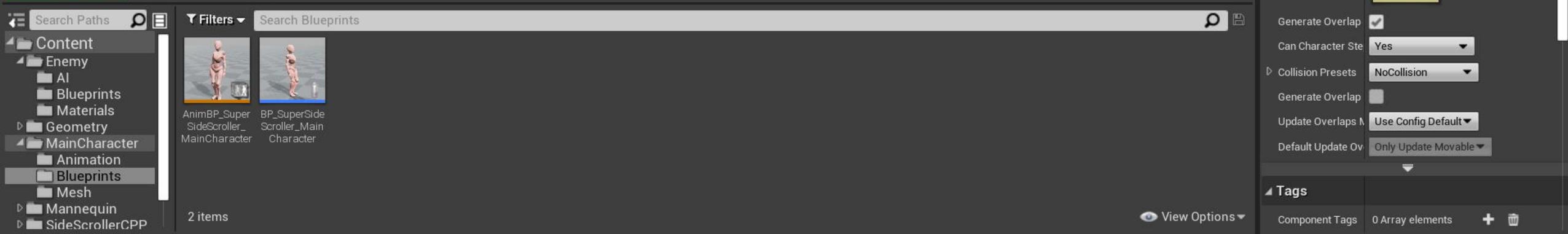
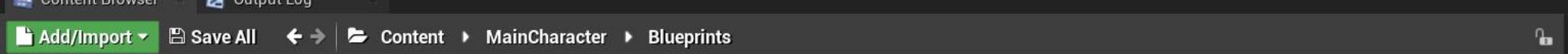
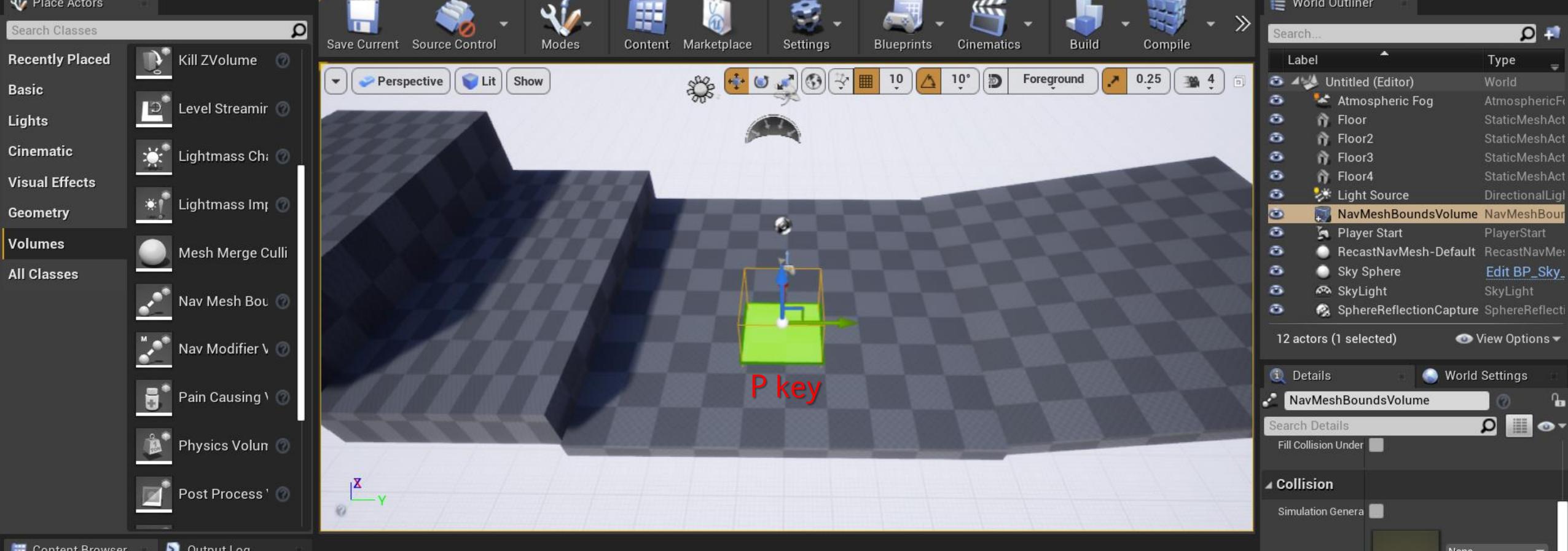
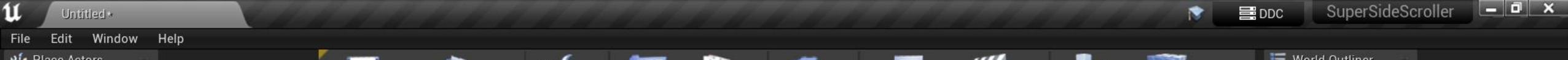


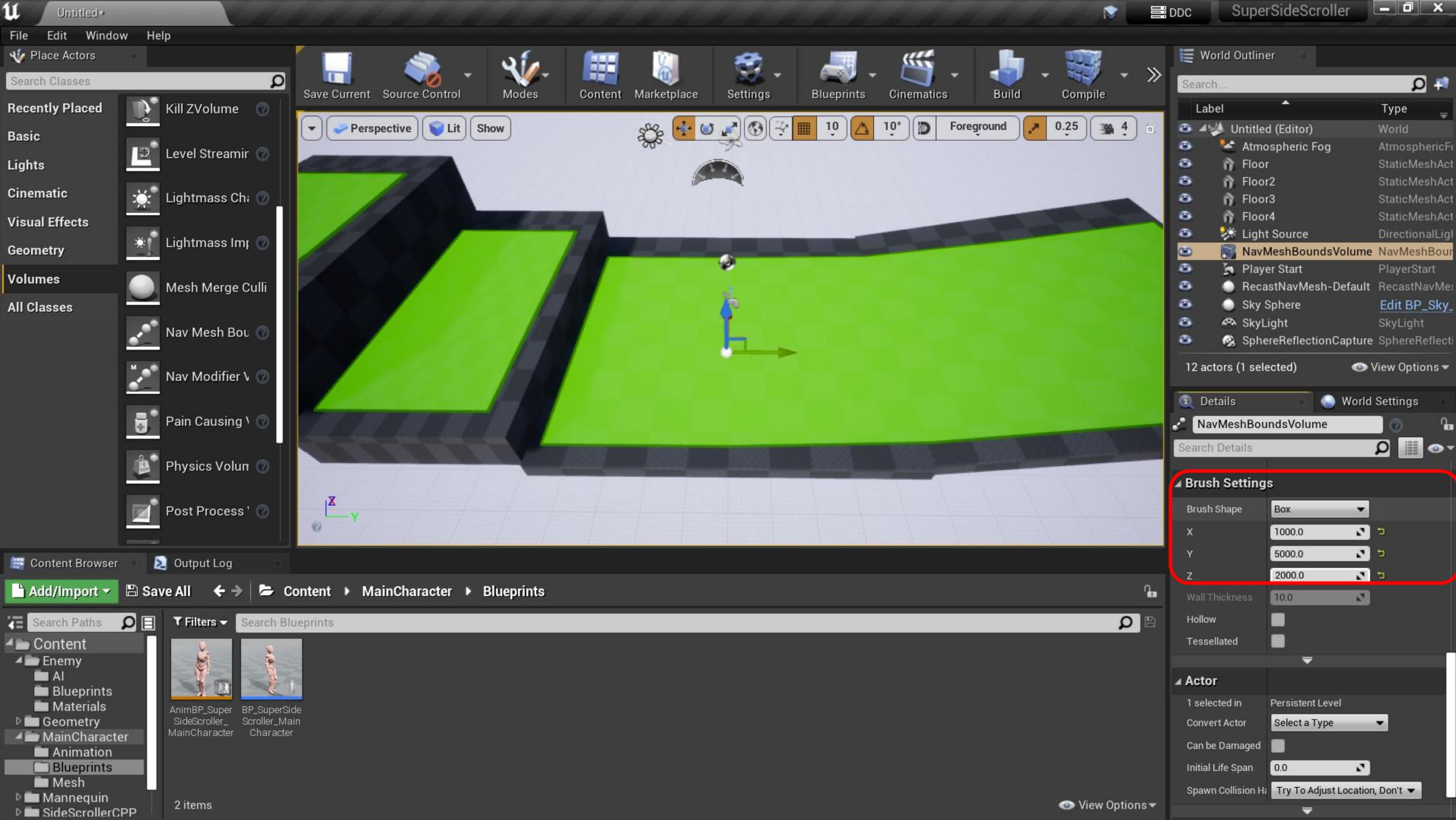


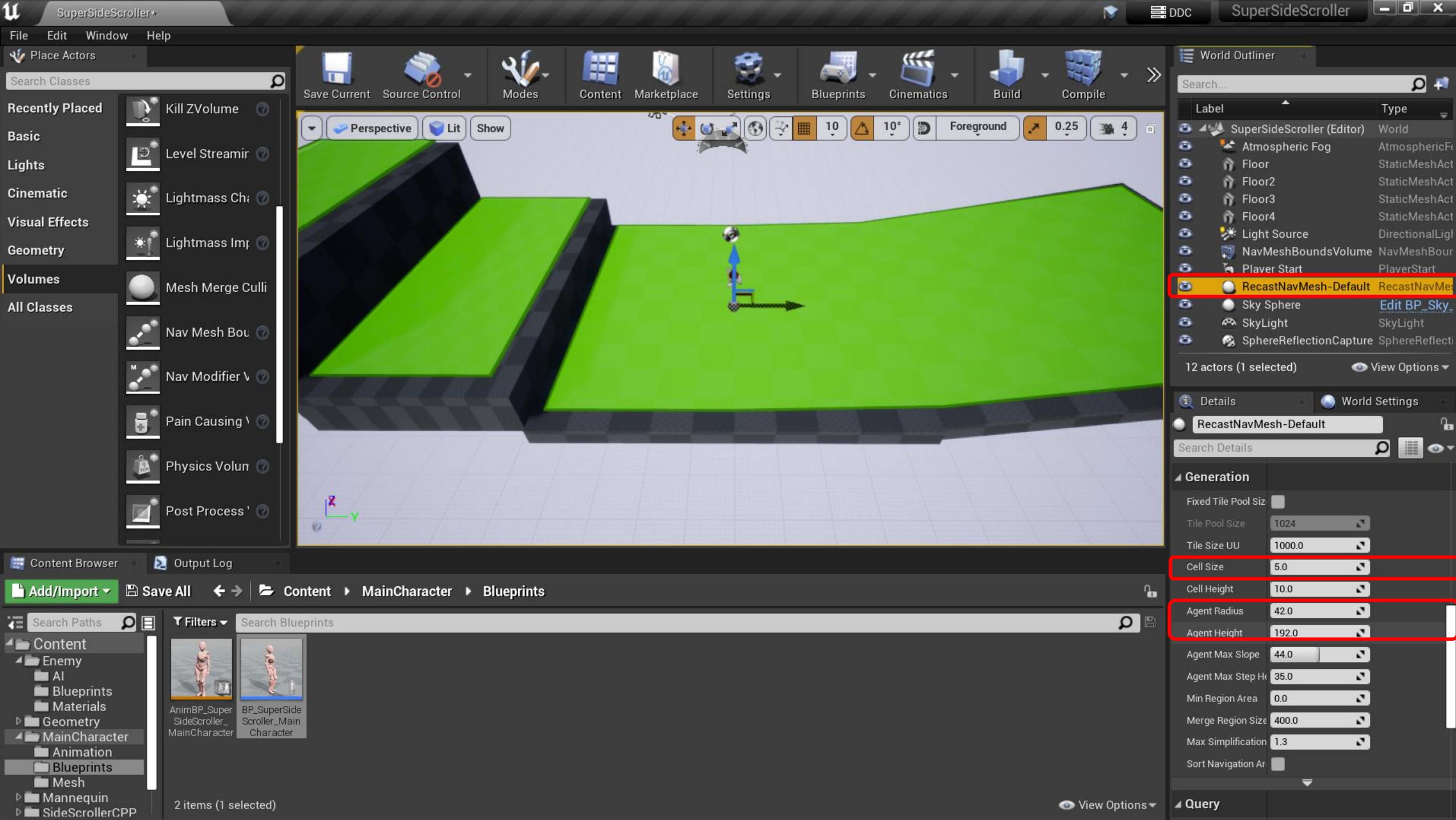


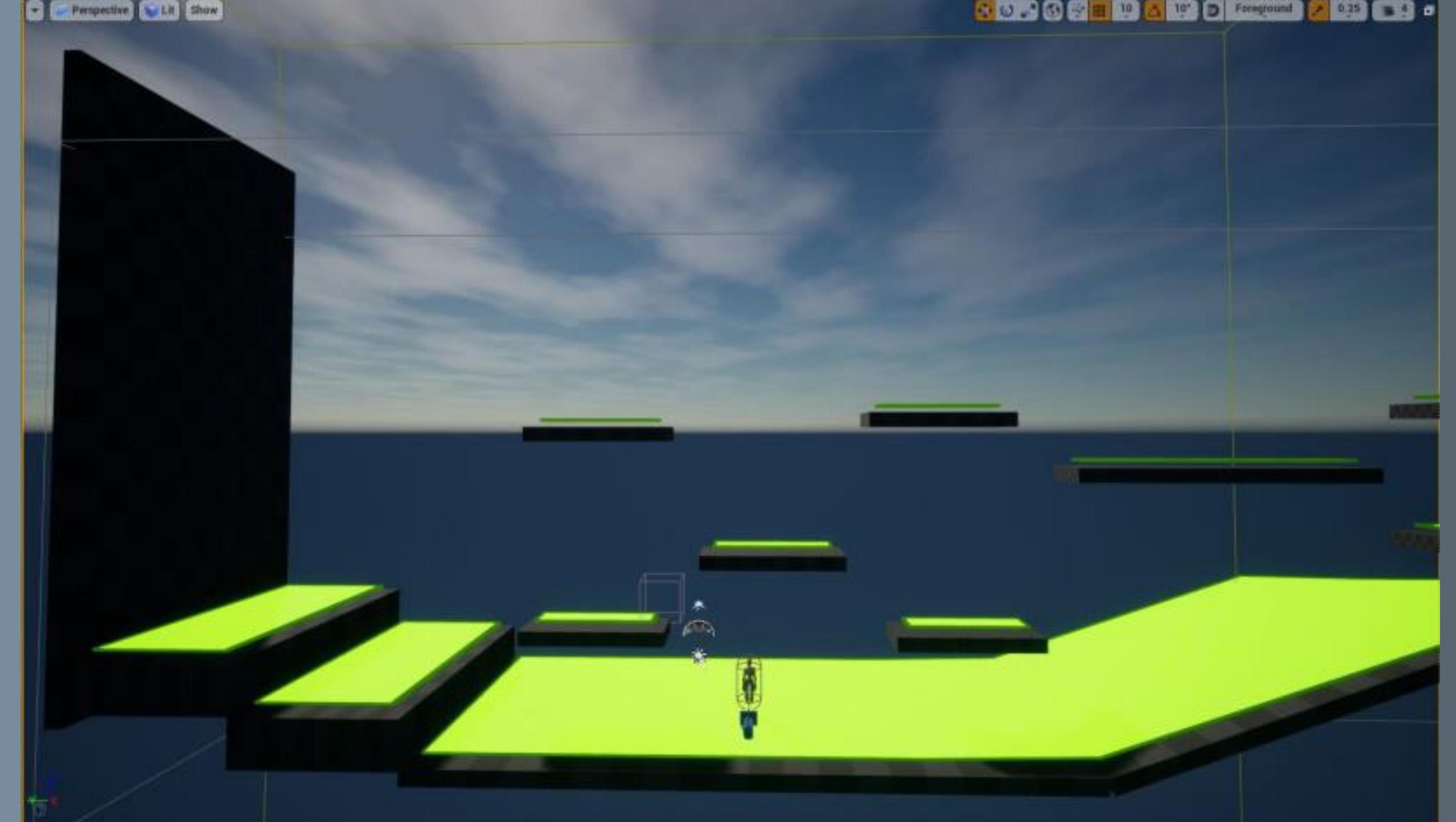












File Edit Window Help

Load and Save

- New Level... Ctrl+N
- Open Level... Ctrl+O
- Save Current Ctrl+S
- Save Current As... Ctrl+Alt+S
- Save All Levels
- Open Asset...
- Save All Ctrl+Shift+S

Choose Files to Save...

Connect To Source Control...

Project

- New Project...
- Open Project...
- New C++ Class...
- Package Project

Refresh Visual Studio Project

Open Visual Studio

Cook Content for Windows

DataValidation

- Validate Data...

Python

- Execute Python Script...
- Recent Python Scripts

Actors

- Import Into Level...
- Export All...
- Export Selected...

Favorite Levels

Recent Levels

Recent Projects

Exit

Geometry

- MainCharacter
- Animation
- Blueprints
- Mesh

- Mannequin
- SideScrollerCPP

Save Current Source Control Modes Content Marketplace Settings Blueprints Cinematics Build Compile

Perspective Lit Show

Volume

Save the current level as...

10 10° Foreground 0.25 4

Content MainCharacter Blueprints

Search Blueprints

?_Super_SideScroller_MainCharacter BP_SuperSideScroller_MainCharacter

2 items (1 selected)

View Options

World Outliner

Search...

Label	Type
SuperSideScroller (Editor)	World
Atmospheric Fog	AtmosphericFog
Floor	StaticMeshActor
Floor2	StaticMeshActor
Floor3	StaticMeshActor
Floor4	StaticMeshActor
Light Source	DirectionalLight
NavMeshBoundsVolume	NavMeshBoundary
Player Start	PlayerStart
RecastNavMesh-Default	RecastNavMesh
Sky Sphere	Edit BP_Sky
SkyLight	SkyLight
SphereReflectionCapture	SphereReflection

12 actors (1 selected) View Options

Details World Settings

RecastNavMesh-Default

Search Details

Generation

Fixed Tile Pool Size	1024
Tile Pool Size	1024
Tile Size UU	1000.0
Cell Size	5.0
Cell Height	10.0
Agent Radius	42.0
Agent Height	192.0
Agent Max Slope	44.0
Agent Max Step He	35.0
Min Region Area	0.0
Merge Region Size	400.0
Max Simplification	1.3
Sort Navigation Ar	

Query



Save Level As



Search Folders



Search Assets

Side
Scroller
Example

Content

- ▷ Enemy
- ▷ Geometry
- ▷ MainCharacter
- ▷ Mannequin

SideScrollerCPP

- Blueprints
- Maps

StarterContent



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Maps



Behavior Trees and Blackboards (1)

- › **Behavior trees** and **Blackboards** work together to allow our AI to follow different logical paths and make decisions based on a variety of conditions and variables.
- › A **behavior tree** (BT)
 - A visual scripting tool that allows you to tell a pawn what to do based on certain factors and parameters
- › **Blackboards**
 - Where you define the set of variables that are required in order to have the behavior tree perform actions and use those values for decision-making
 - **Keys, Tasks, Services, and Decorators**



Behavior Trees and Blackboards (2)

