

Advanced Door Pack

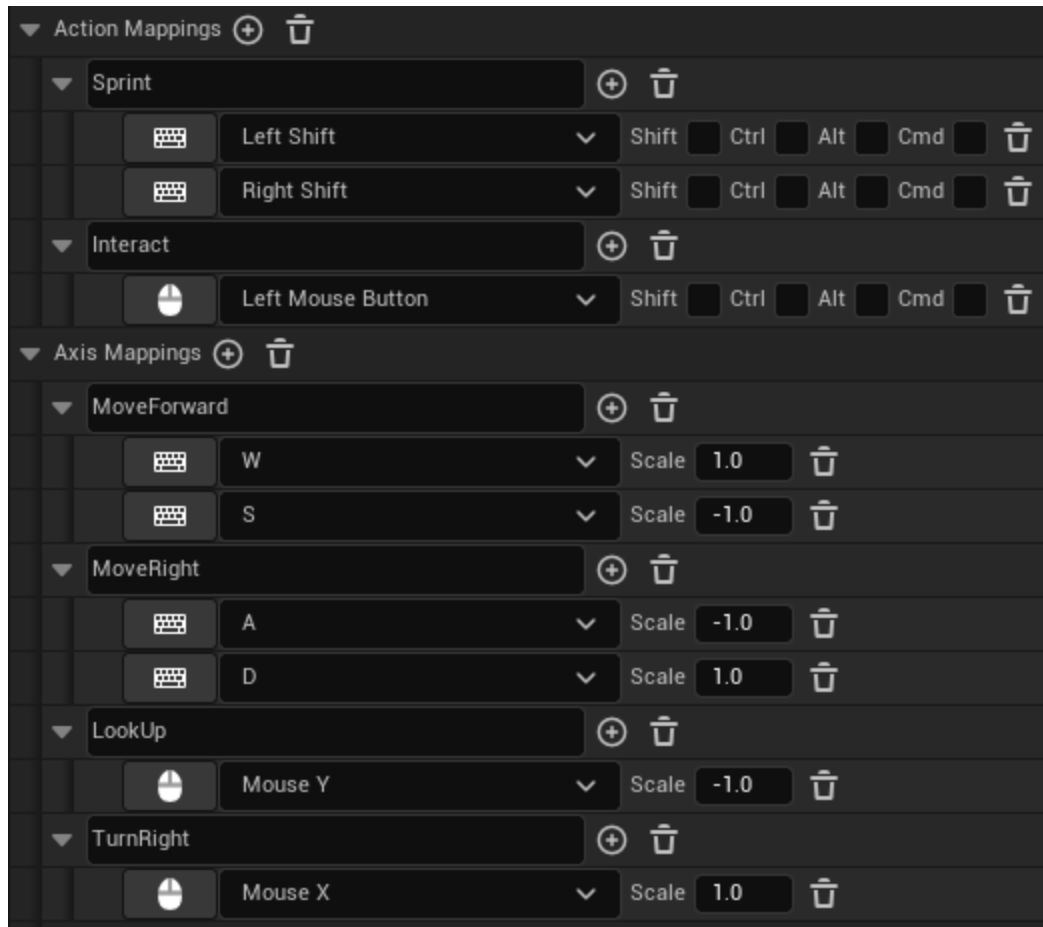
With Player Interaction and UI Indicator System

Content

These numbers exclude demo room assets

- 1 Showcase Map
- 55 Static Meshes, 106 Textures
- 36 Material Instances, 2 Master Materials
- 22 Blueprints, 2 Interfaces
- 5 Structs, 4 Data Tables
- 7 Sound Waves, 5 Sounds Cues, 2 Sound Attenuations

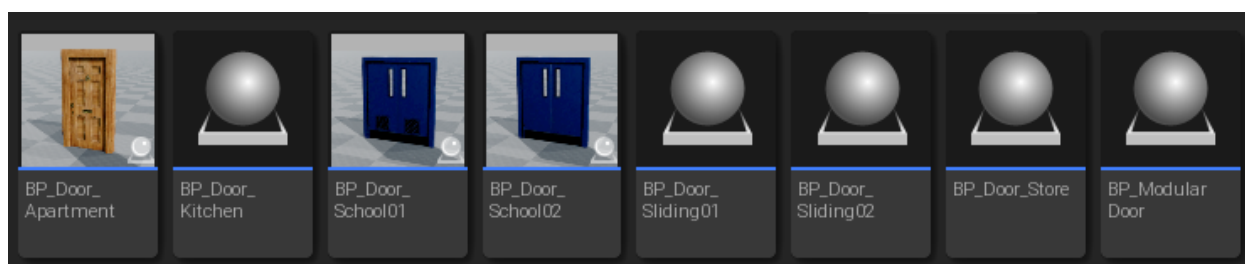
Note: You need the following Input mappings in order for the **Showcase Map** to work properly. Alternatively you can import input settings [from here](#). This will **overwrite** your input settings.



How To Use

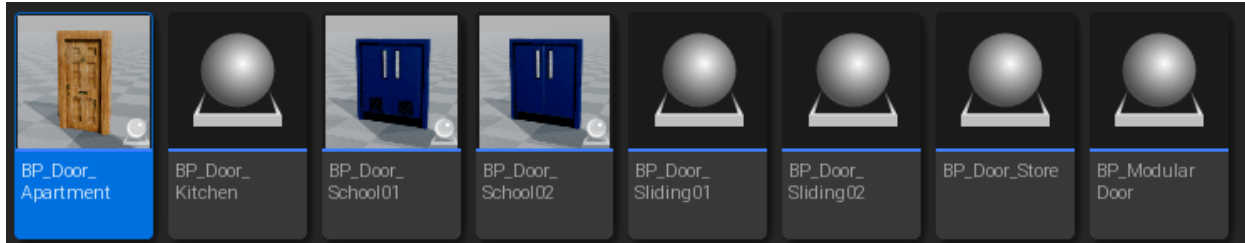
Door Blueprints are located in Blueprint/Doors. You can drag and drop any of the blueprints into the scene. Some blueprints have customizable options.

All doors are **interactable** with the player.



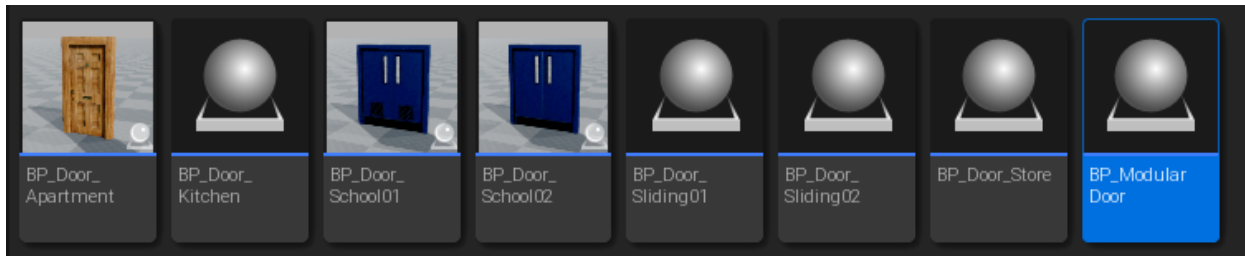
Simple Door

This is a basic apartment door example. It has a frame and a rotating door mesh.



Modular Door

Modular Door is a single blueprint that allows you to easily switch Door Mesh and choose a Material provided from various data tables.



You can select door mesh (with frame), handle mesh and materials separately.

▼ Setup

▼ Door / Frame

Has Door Frame	☒
▼ Select Door	
Data Table	DT_Door ▼ ↶ 📁
Row Name	Apartment_1 ▼

▼ Handle

Has Front Handle	☒
▼ Select Front Handle	
Data Table	DT_Handle ▼ ↶ 📁
Row Name	Handle_1 ▼
Has Back Handle	☒
Use Separate Back Handle	☐
▶ Select Back Handle	

▼ Lock Plate / Strike Plate

Has Lock Plate	☒
Has Strike Plate	☒
▶ Select Lock Plate	

▼ Sockets

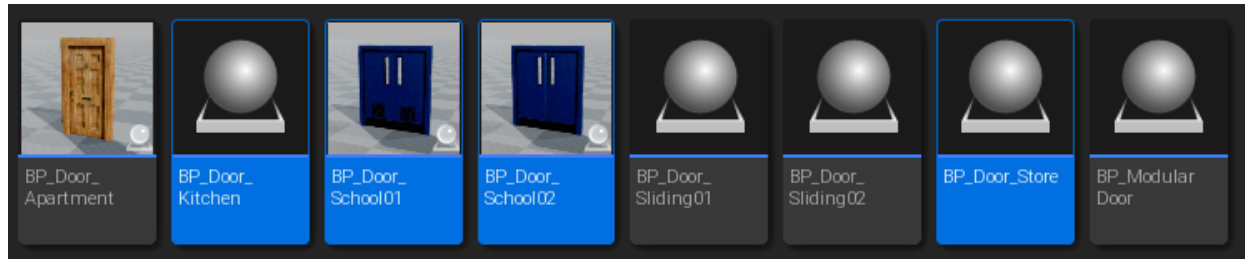
Door Socket	Door
Handle Socket	Handle
Front Handle Socket	FrontHandle
Back Handle Socket	BackHandle
Lock Plate Socket	LockPlate
Strike Plate Socket	StrikePlate

▼ Materials

Set Materials from Data Table	☒
▶ Material Selections	3 Map elements + 🗑

Double Door

Double Doors contain a single frame with two Doors. These blueprints inherit from **BP_DoubleDoorBase** for common functionality.

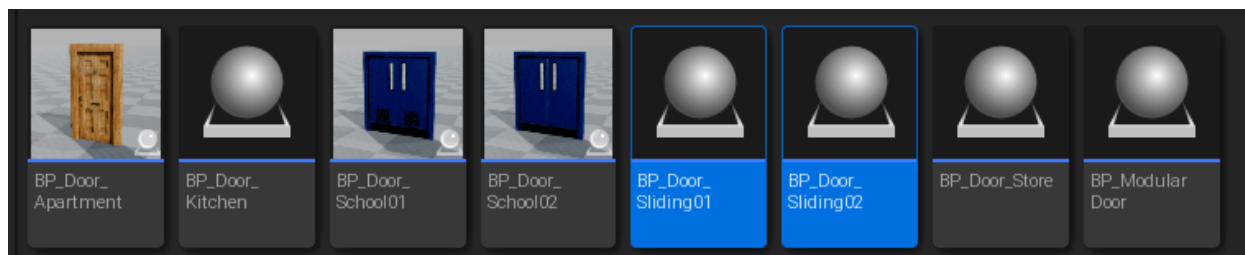


Currently there are **4** double doors and each door has **2** sets of materials that can be selected from the details panel.

Sliding Door

Sliding Doors move horizontally. These blueprints inherit from **BP_SlidingDoorBase** for common functionality.

Note: Currently sliding doors are not physical.



Currently there are **2** sliding doors and each door has **2** sets of materials that can be selected from the details panel.

Material Selections

Some blueprints provide **Material Selections** option.

This map allows you to select a material for a specific material slot name. only **Selected Material** is editable.

These materials are queried from a data table (Currently from Blueprint/Core/DataTable/DT_MaterialSets)

Material Selections		3 Map elements		
▼	Door_Frame_Apartment	3 members	▼	
	Select Material	0		
	Number of Materials	2		
	Selected Material	 MI_Door_Apartment_V1		
▼	Door_Apartment	3 members	▼	
	Select Material	0		
	Number of Materials	2		
	Selected Material	 MI_Door_Apartment_V1		
▼	Door_Handle	3 members	▼	
	Select Material	0		
	Number of Materials	4		
	Selected Material	 MI_Door_Handles_Black		

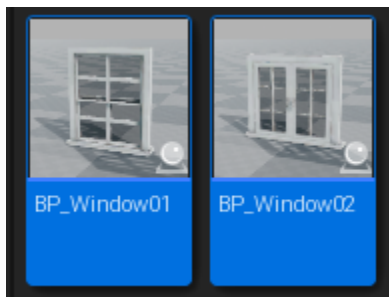
Select Material: Must be equal or greater than zero and less than the **Number of Materials**.

Number of Materials: Number of materials available to select. This is read-only and determined by the source DataTable.

Selected Material: Selected material soft reference. This is read-only and determined by the source DataTable.

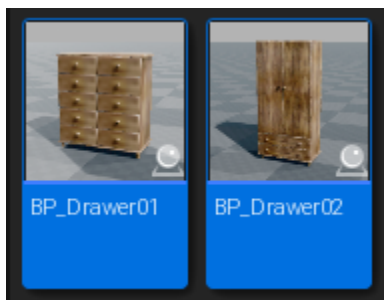
Windows

This pack includes additional blueprints for Rotating and Sliding windows. Both blueprints are interactable.



Drawers

This pack includes additional blueprints for Drawers. Both blueprints are interactable.



Components

Door Component

Door Component Inherits from Static Mesh Component. It rotates a mesh from its pivot and has basic functionality of opening or closing the door and handles basic mouse input and sound effects.

Physics simulation is enabled when the door is enabled.

All options have tooltips.

Setup

General

Closed Rotation Z	0.0
Opened Rotation Z	90.0
Near Close	1.5
Rotations Are Relative To Current Rotation	☐
Can be Closed	☒

Interaction

Mesh Rotation Offset	-90.0
Rotations Are Clockwise	☒

Sound

Closed Sound	 SC_Door_Close  
Opened Sound	 SC_Door_Open  
Grabbed Sound	 SC_DoorHandle_Grab  
Released Sound	 SC_DoorHandle_Release  
Hard Close Sound Switch	Hard
Hard Close Sound Threshold	0.8
Sound Attenuation Settings	 SA_Door  

DoorCloser

1 - Go To Opened Rotation

2 - Go To Closed Rotation

3 - This Is Closed Rotation

4 - This Is Opened Rotation

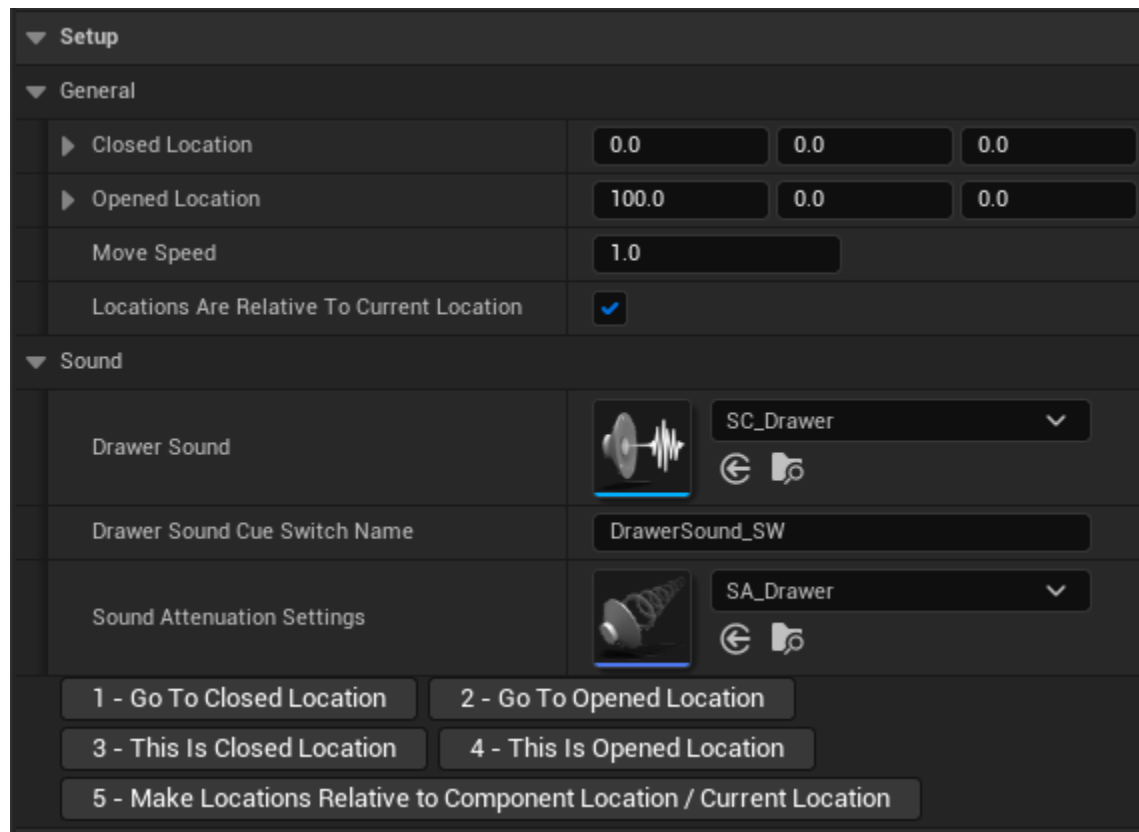
5 - Make Rotations Relative to Component Rotation / Current Rotation

Drawer Component

Drawer Component Inherits from Static Mesh Component. It moves in a straight line and has basic functionality of a drawer. It handles basic mouse input and sound effects

Currently drawers don't have physics simulation.

All options have tooltips.



Interaction System

We also provide a ready to use Interaction system which is used for but not limited to Doors and Drawers.

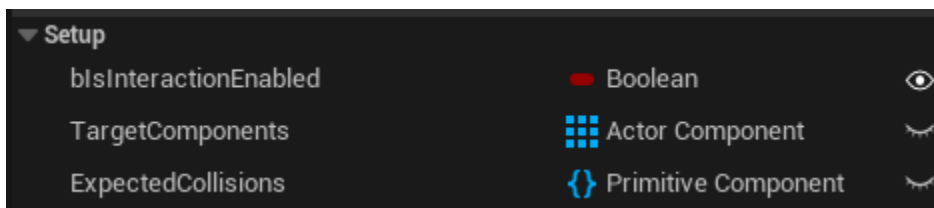
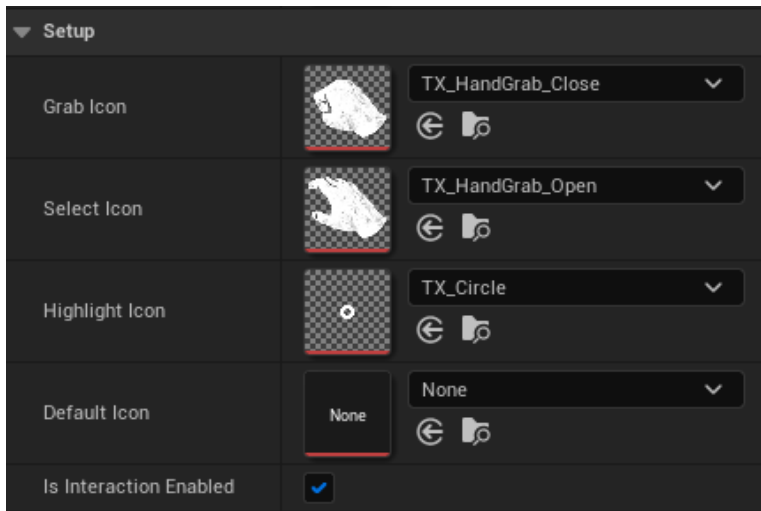
Interaction is achieved by four components.

- **Interaction Component** on the Player Character to manage interactions.
- **Indicator Component** on the Actors that can be Interacted to indicate possible interactions.

- **Sphere Collision Component** on the Player Character to detect nearby interactable actors.
- Components implementing **I_Interact** interface to handle the actual interaction.

Indicator Component

Indicator component is a simple screen space Icon to indicate player interaction with the environment.



Grab Icon: when the player holds Interaction. (for example holding the left mouse button down)

Select Icon: when the player is looking at the Indicator and is close enough to begin interaction.

Highlight Icon: when Indicator is close but not enough for interaction. The closest indicator to the player view is highlighted.

Default Icon: when Indicator is close but not enough for interaction. This is for indicators that are not highlighted.

Is Interaction Enabled: setting to false will prevent the Indicator from showing up and interacting.

This can be changed at any time.

Note: when an indicator is not close, it will be **deactivated** and disappear.

TargetComponents: These components should implement the **I_Interact** interface.

This is the list of components that will be interacted when the player uses this indicator for interaction.

Avoid adding duplicate items to this list.

ExpectedCollisions: Set of components that are acceptable hit when doing **Sphere Trace** to find accessible Indicators.

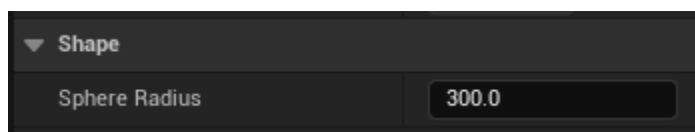
If the sphere trace hits any component in this set, The indicator can be shown, but if the sphere trace hits a component that is not in this set, the indicator remains hidden.

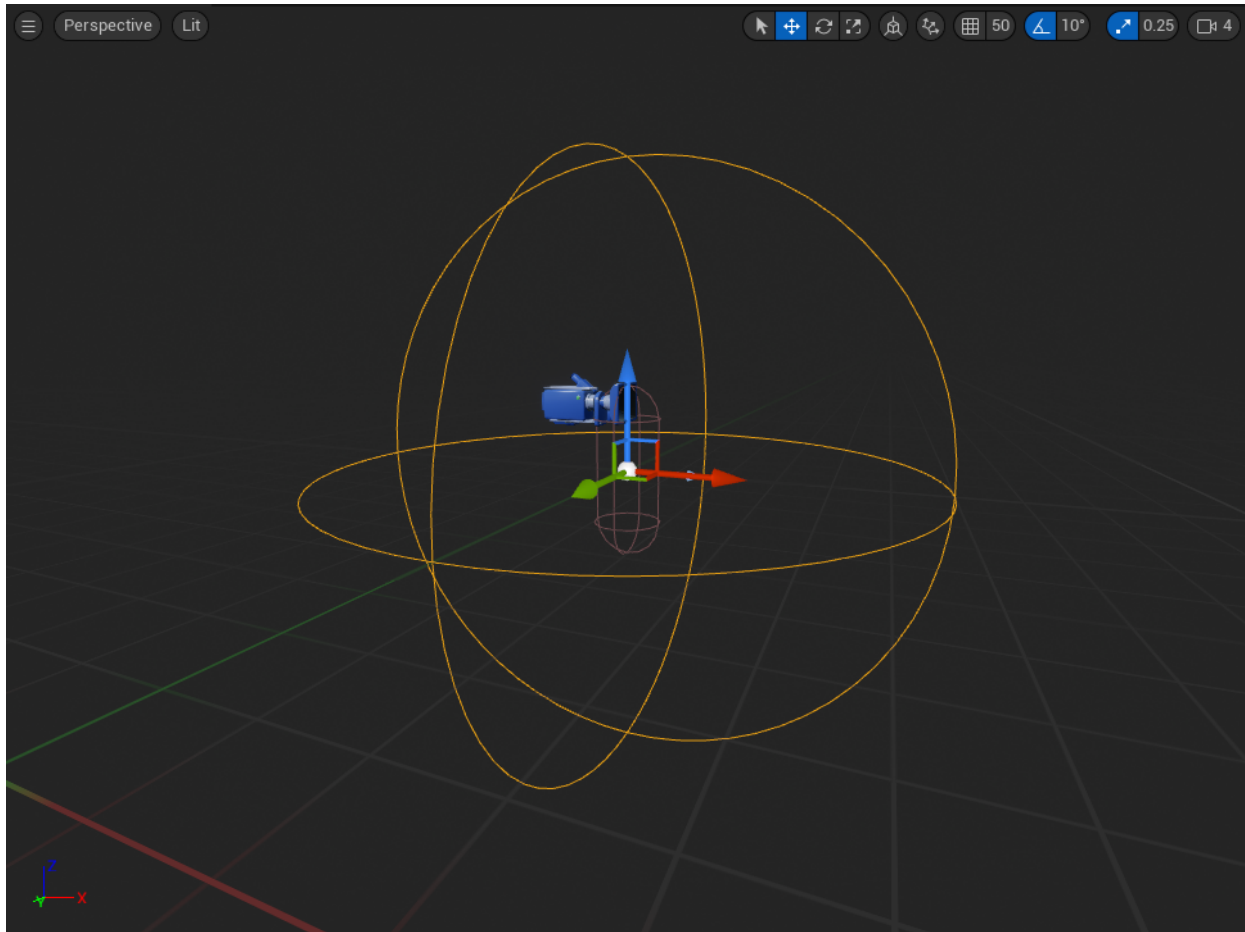
If the sphere trace hits nothing, the Indicator can still be shown.

Sphere Collision Component

A simple Sphere Collision is used to detect nearby actors that can be interacted with. Radius of the sphere will determine at which distance Indicators are detected for interaction.

Sphere Collision is placed on the player character.





Interaction Component

The **Interaction Component** will manage player interaction with other actors via **Indicator Components** placed on other actors (Such as Doors).

This component is placed on the player character.

▼ Setup	
▼ General	
Interaction Range Ratio	0.5
Keep Interaction Range Ratio	0.7
Interaction Max Angle	15.0
Highlight Max Angle	45.0
Update Interval	0.05
▼ SphereTrace	
Sphere Trace	☒
Trace Channel	Visibility ▼
Trace Complex	☐
Trace Radius	0.5
Hit Location Max Distance	10.0

Interaction Range Ratio: This ratio is multiplied by the radius of the **Sphere Collision** and determines the player interaction range.

Interaction Range = Ratio * Sphere Radius

For example let Ratio be 0.5 and Sphere Radius be 300. Player Interaction Range will be 150.

Interaction Range = $0.5 * 300 = 150 \text{ cm}$

Which means only Indicators closer than this distance can be interacted. Calculated **distance is in 2D**.

Keep Interaction Range Ratio: Calculated the same way as Interaction Range Ratio.

When the player holds interaction, Indicators can get this far before interaction is canceled. This should be larger than the Interaction Range Ratio.

Set to 0 to disable this feature, which means once the player starts an Interaction, it can be held at any distance.

Interaction Max Angle: The Indicator can be interacted if the angle between the player view and the Indicator is smaller than this angle.

Highlight Max Angle: The Indicator can be highlighted if the angle between the player view and the Indicator is smaller than this angle.

This value should be larger than Interaction Max Angle

Update Interval: Interval for updating nearby indicators. Higher intervals can improve performance while lower intervals reduce response time.

Sphere Trace: when enabled, every interval, Sphere Trace is performed between the player view location and all nearby indicators. Hit result is used to determine if indicator is reachable or not.

Indicators that are blocked by other objects are not shown and not interactable.

Trace Channel: Trace channel to use for the Sphere Trace. You can use custom channels for better control but takes more time to set up.

Trace Complex: This is more expensive and not recommended.

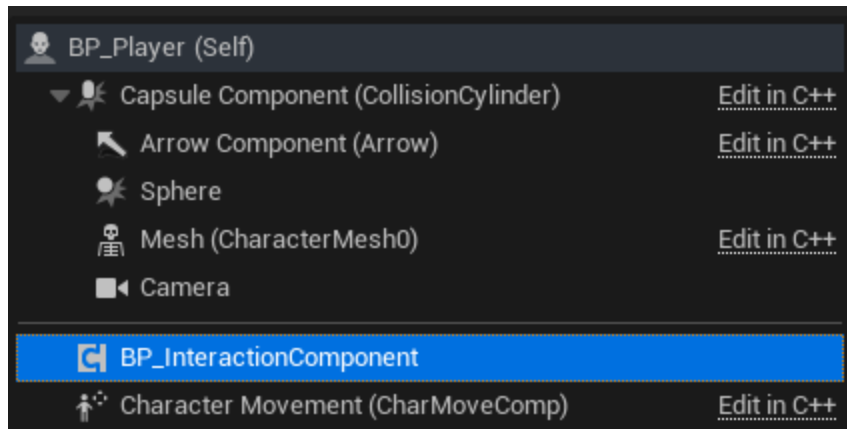
Trace Radius: Specify the radius for sphere traces in centimeters.

Hit Location Max Distance: when sphere trace hits an expected collision, distance between the Indicator and the hit location should not be longer than this value. This is helpful if there are multiple indicators for a single mesh. (for example two indicators on opposite sides of a door).

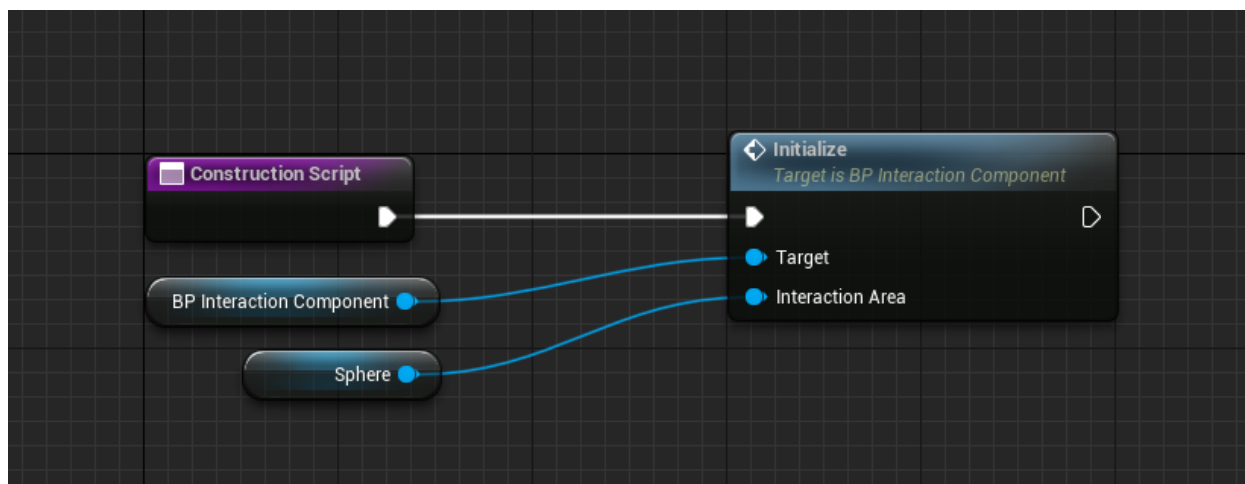
Set to 0 to disable this feature which means indicators can be at any distance from the hit location and still be acceptable for interaction.

Player Blueprint Setup

The **Interaction Component** must be placed on the **Player Character**.

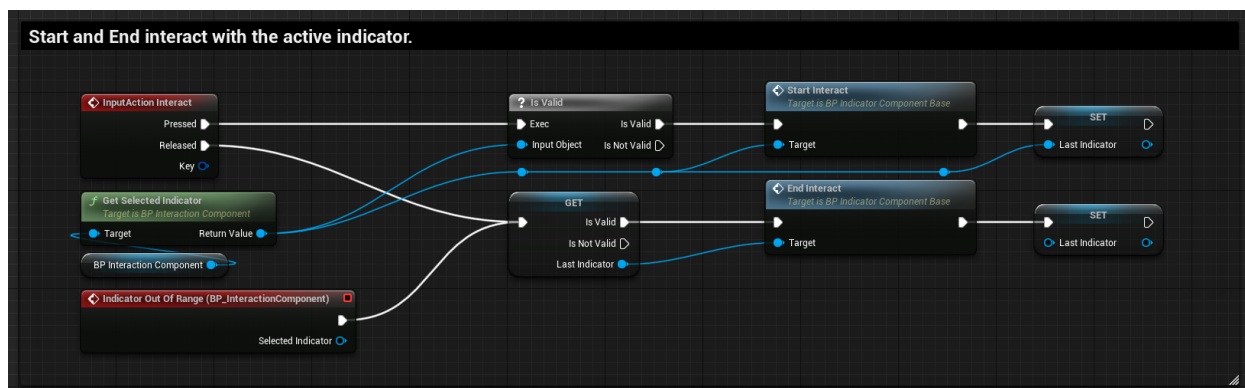


In the **Construction Script** of the Player Character, setup Interaction Component.



And in the **Event Graph** handle Player input for interaction.

Note: **Selected Indicator** is the chosen indicator from all nearby indicators by the **Interaction Component** based on the player view and distance.



Your player is now ready to interact with doors and drawers.

How To Extend

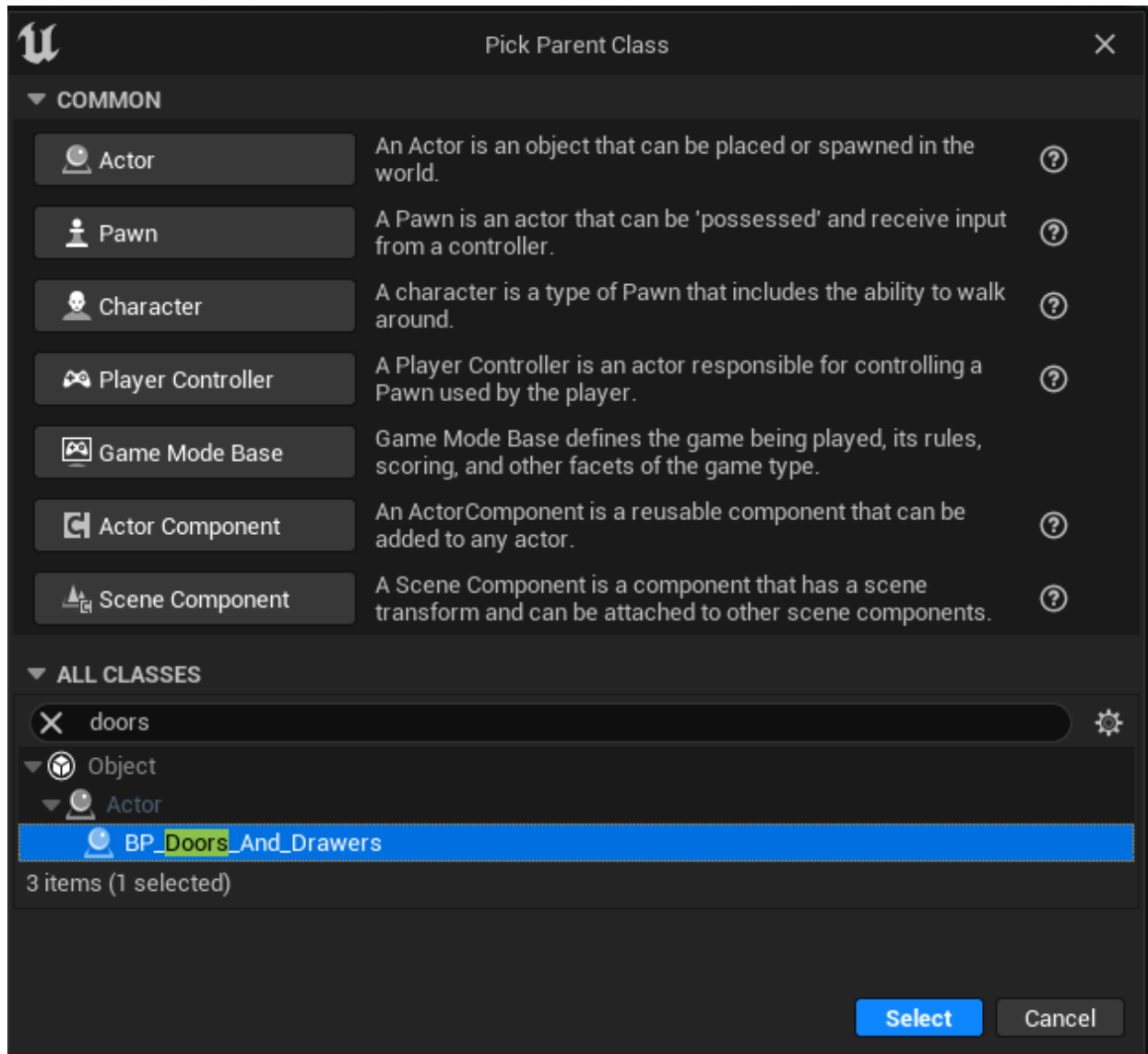
In this section we will show you ways to extend from the content.

Simple Blueprint

You can easily create more doors by creating a child blueprint from **BP_DoorsAndDrawers**.

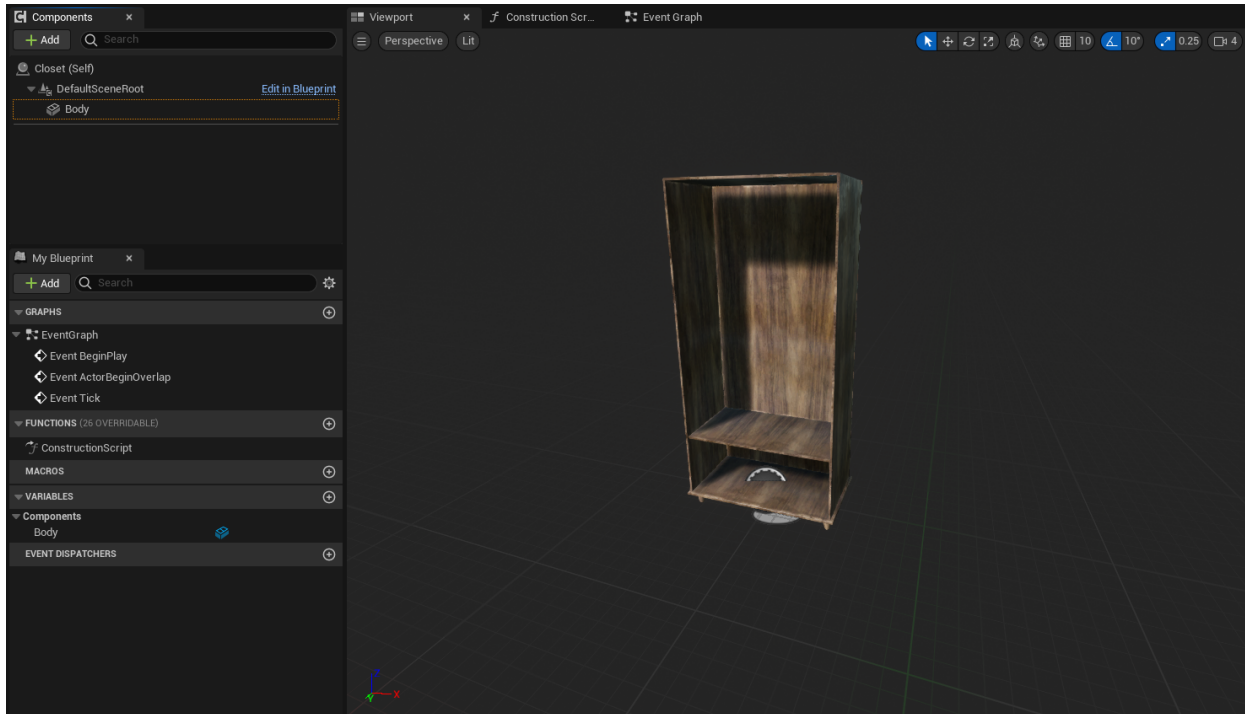
This actor will automatically take care of creating Physics Constraint Components for Doors and Sound Components for Drawers and setting them up.

In this example we will create a blueprint for a Closet with Drawers. Create a new blueprint and select **BP_Doors_And_Drawers** as the parent.

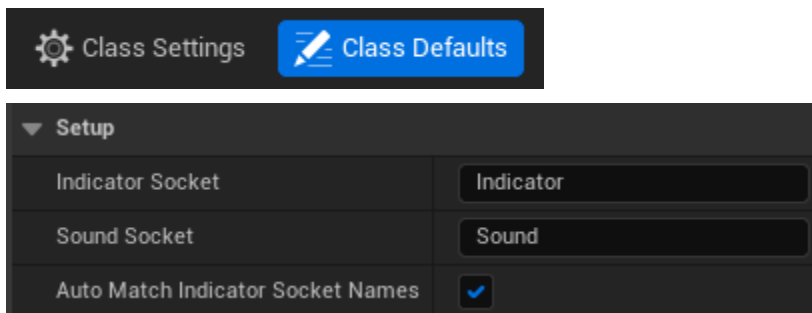


Add one Static Mesh Component for the Closet Body.

Static Mesh: Meshes/Drawers/SM_Body_Drawer02



In the class defaults you have these settings.



Indicator Socket: an indicator will be automatically created for sockets on Doors/Drawers with this name.

Sound Socket: spawned sounds will be attached to this socket on Doors/Drawers.

Auto Match Indicator Socket Names: if true, multiple indicators will be created for Doors/Drawers that have multiple sockets starting with specified **Indicator Socket** name.

For example Indicators will be created for socket Indicator_2, IndicatorB, etc...

Make sure to have the specified sockets on Door/Drawer meshes.

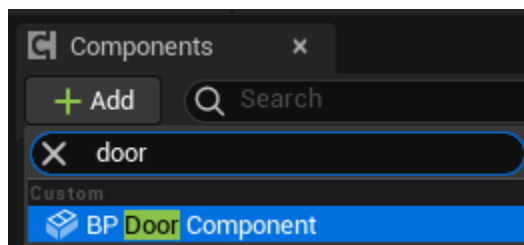


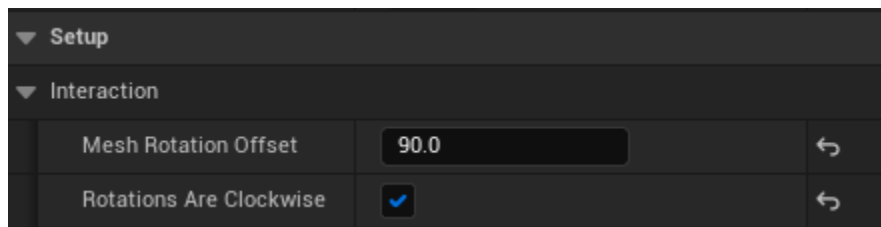
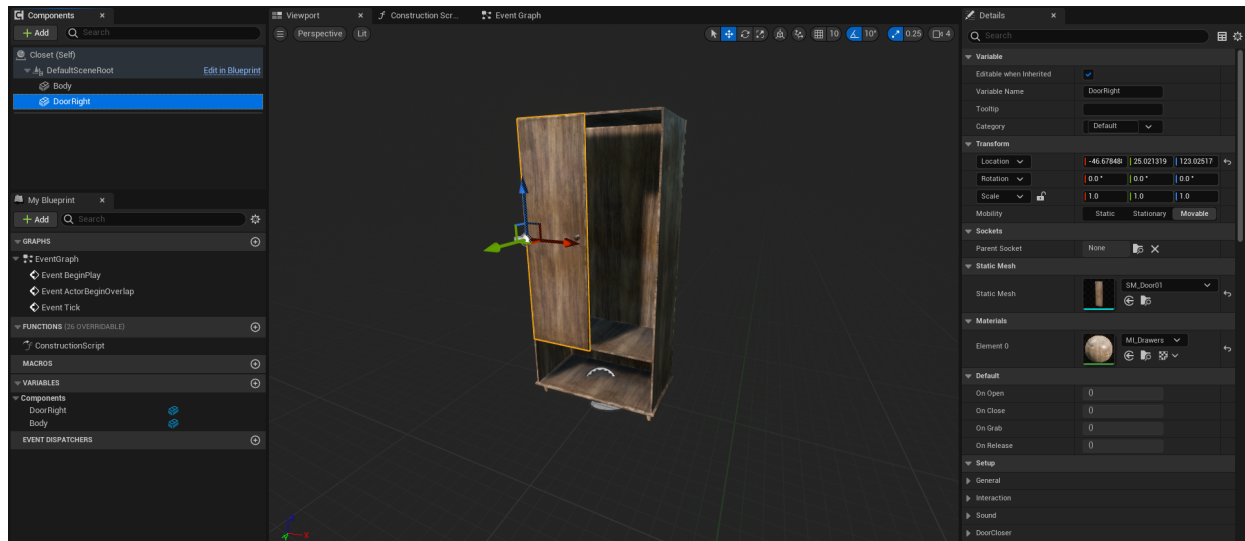


Creating Doors

Add a **BP Door Component** and select the **Static Mesh** for the door in the details panel.

Static Mesh: Meshes/Drawers/SM_Door01



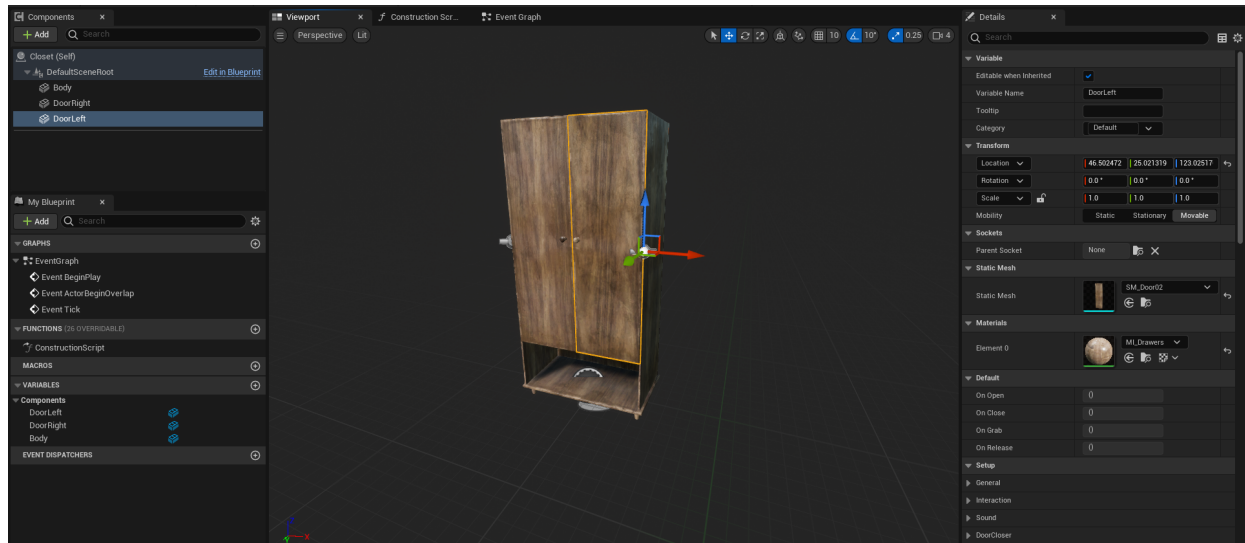


In the **Setup** category,
 Set Mesh Rotation Offset to **90**
 Set Rotations Are Clockwise to **true**

Note: You may need to set different settings if the orientation of the mesh is different.

Add another **BP Door Component** for the left door.

Static Mesh: Meshes/Drawers/SM_Door02



In the **Setup** category,
 Set Opened Rotation Z to **-90**
 Set Mesh Rotation Offset to **90**

Note: You may need to set different settings if the orientation of the mesh is different.

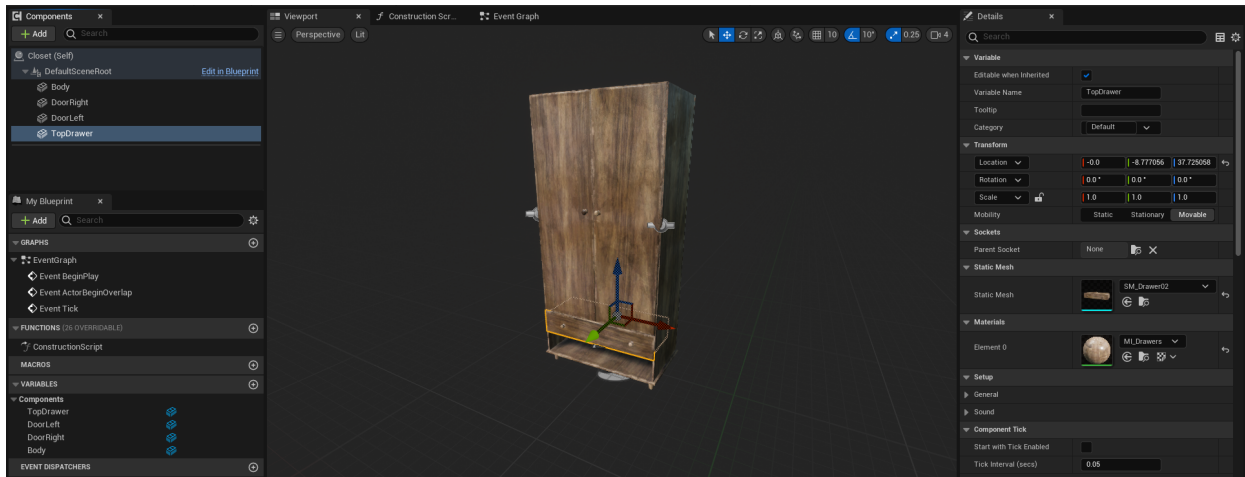
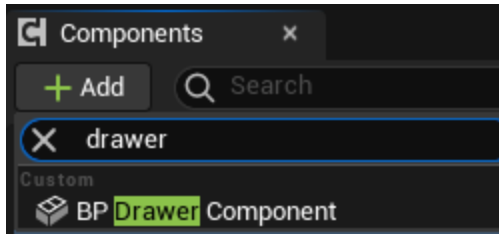


Closet doors can now be interacted by the player.

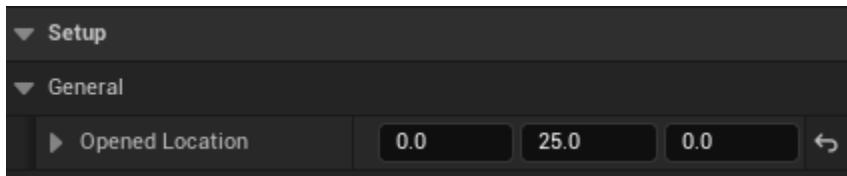
Creating Drawers

Add a **BP Drawer Component** and select the **Static Mesh** for the Drawer in the details panel.

Static Mesh: Meshes/Drawers/SM_Drawer02

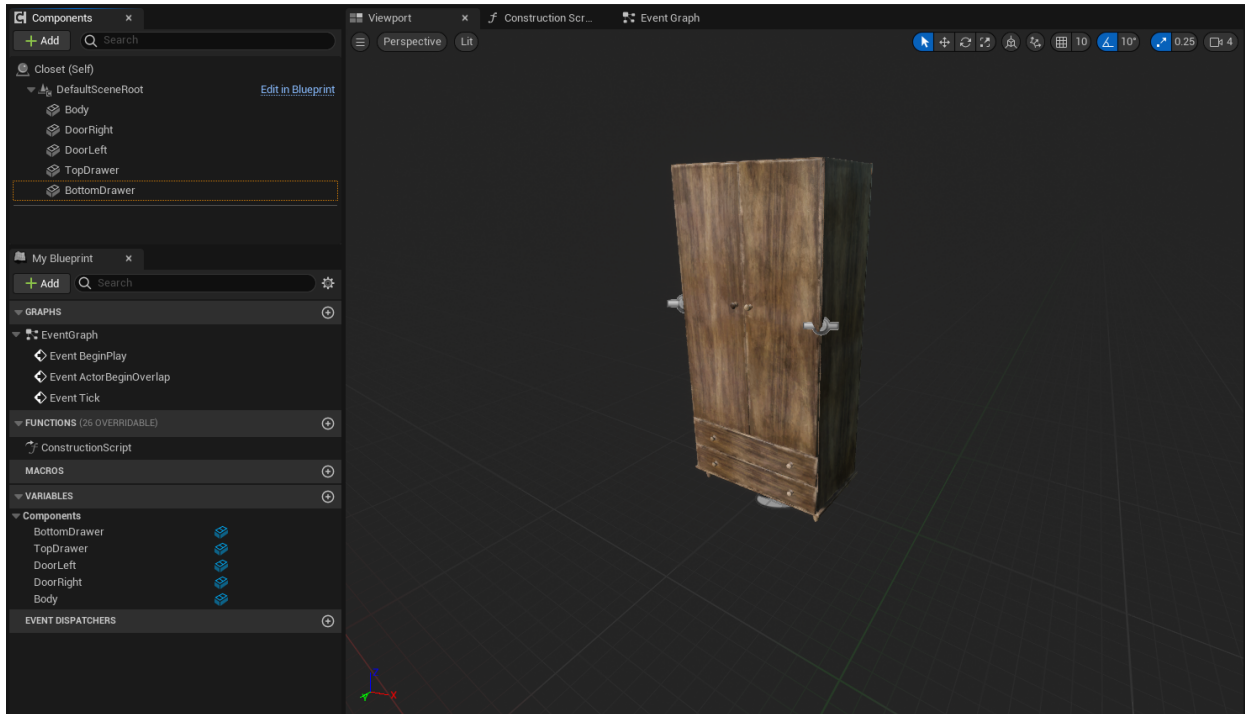


In the **Setup** category,
Set Opened Location to **(0, 25, 0)**

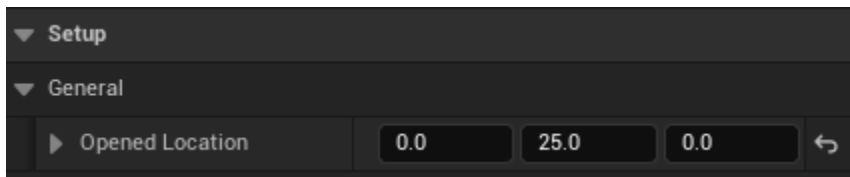


Add another **BP Drawer Component** for the bottom drawer.

Static Mesh: Meshes/Drawers/SM_Drawer02



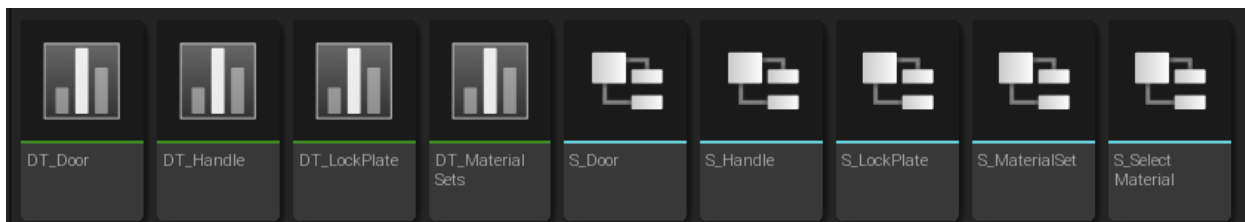
In the **Setup** category,
Set Opened Location to **(0, 25, 0)**



The Closet blueprint is now ready and the player can interact with all the Doors and Drawers in this blueprint.

Extending Data Tables

Data tables are located at Blueprint/Core/DataTable



An example of pairing Apartment Door Mesh with Apartment Door Frame Mesh.

Data Table		Data Table Details	
Row Name	Frame	Frame Transform	Door
1 Apartment_1	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_Frame_Apartment SM	("Rotation": ("X": 0, "Y": 0, "Z": 0, "W": 1), "Translation": ("X": 0, "Y": 0, "Z": 0))	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_Apartment SM Door
2 Bedroom_1	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_Frame_Bedroom SM	("Rotation": ("X": 0, "Y": 0, "Z": 0, "W": 1), "Translation": ("X": 0, "Y": 0, "Z": 0))	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_Bedroom SM Door
3 House_1	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_Frame_House SM	("Rotation": ("X": 0, "Y": 0, "Z": 0, "W": 1), "Translation": ("X": 0, "Y": 0, "Z": 0))	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_House SM Door
4 Office_1	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_Frame_Office01 SM	("Rotation": ("X": 0, "Y": 0, "Z": 0, "W": 1), "Translation": ("X": 0, "Y": 0, "Z": 0))	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_Office01 SM Door
5 Office_2	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_Frame_Office02 SM	("Rotation": ("X": 0, "Y": 0, "Z": 0, "W": 1), "Translation": ("X": 0, "Y": 0, "Z": 0))	/Game/PhysicalDoorPack/Meshes/Doors/SM_Door_Office02 SM Door

Row Editor

Apartment_1

Apartment_1

Frame

SM_Door_Frame_Apartment

Frame Transform

Door

SM_Door_Apartment

Door Transform

Modular Doors

These data tables can be extended for modular doors.

 DT_Door	 DT_Handle	 DT_LockPlate	 DT_MaterialSets	 S_Door	 S_Handle	 S_LockPlate	 S_MaterialSet	 S_SelectMaterial
--	--	---	--	---	---	---	--	---

Material Sets

More material sets can be specified in **DT_MaterialSets**.

Row names must match with the Material Slot name. Each static mesh has one or more Material Slots that can be addressed by their name.

Material Slots	
Material Slots	2 Material Slots
Element 0	Highlight Isolate  MI_Door_Store_V1 Slot Name Door_Store
Element 1	Highlight Isolate  MI_Glass_Doors Slot Name Door_Glass

1	Door_Glass	("Game/PhysicalDoorPack/Materials/Instances/MI_Glass_Doors.MI_Glas
2	Door_Handle	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Handles_Black
3	Door_Apartment	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Apartment_V1
4	Door_Frame_Apartment	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Apartment_V1
5	Door_Bedroom	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Bedroom_V1.M
6	Door_Frame_Bedroom	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Bedroom_V1.M
7	Door_House	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_House_V1.MI_I
8	Door_Frame_House	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_House_V1.MI_I
9	Door_Office01	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Office01_V1.M
10	Door_Frame_Office01	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Frame_Office0
11	Door_Office02	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Office02_V1.M
12	Door_Frame_Office02	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Frame_Office0
13	Door_School01	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_School01_V1.M
14	Door_School02	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_School02_V1.M
15	Door_Frame_School	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Frame_Schooli
16	Door_Kitchen	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Kitchen_V1.MI
17	Door_Frame_Kitchen	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Kitchen_V1.MI
18	Door_Sliding	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Sliding_V1.MI
19	Door_Frame_Sliding	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Sliding_V1.MI
20	Door_Store	("Game/PhysicalDoorPack/Materials/Instances/MI_Door_Store_V1.MI_D

Sockets

Static Mesh Sockets are defined for various purposes, including sockets for attachment of Doors, Handles, Indicators, Sounds, etc.