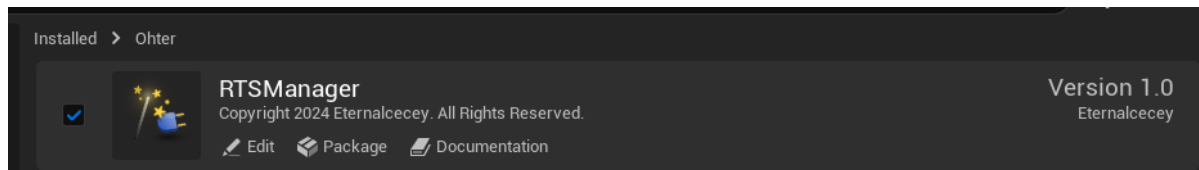
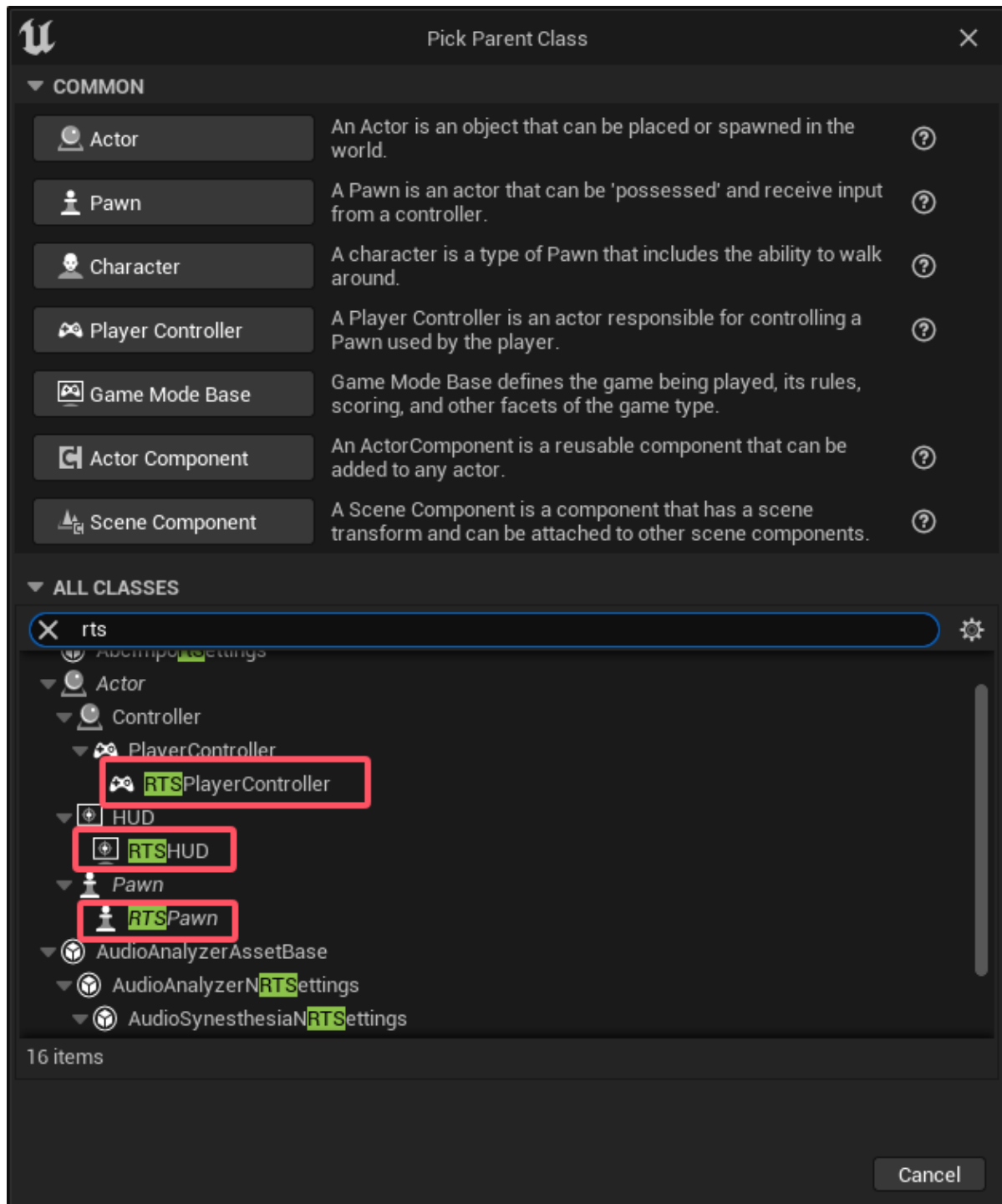


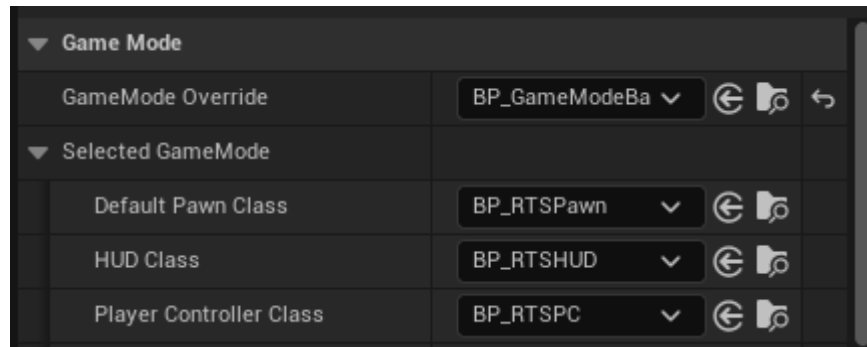
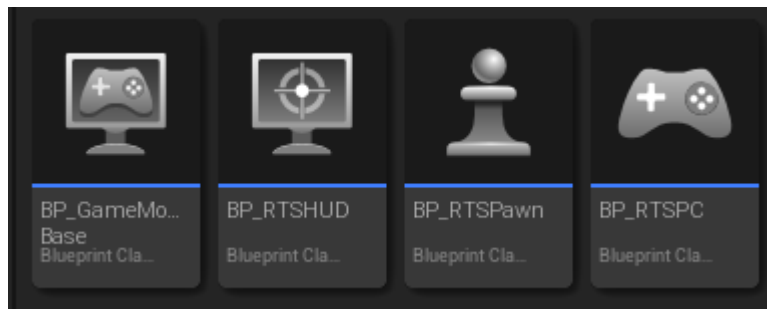
—Activate Plugin



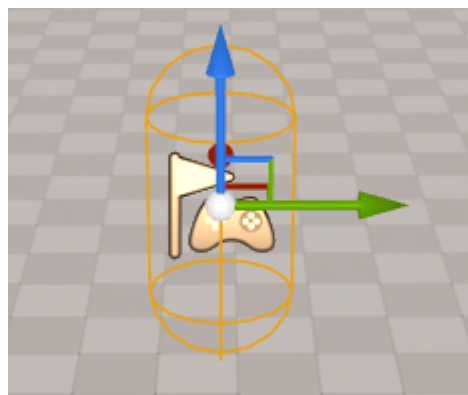
二.Use the plugin to create a basic framework

Create your blueprint classes, requiring inheritance from `RTSPlayerController`, `RTSHUD`, and `RTSPawn`. Note that for `GameMode`, you only need to inherit from the system default or your own `GameMode`; `RTSManger` does not provide a `GameMode`.





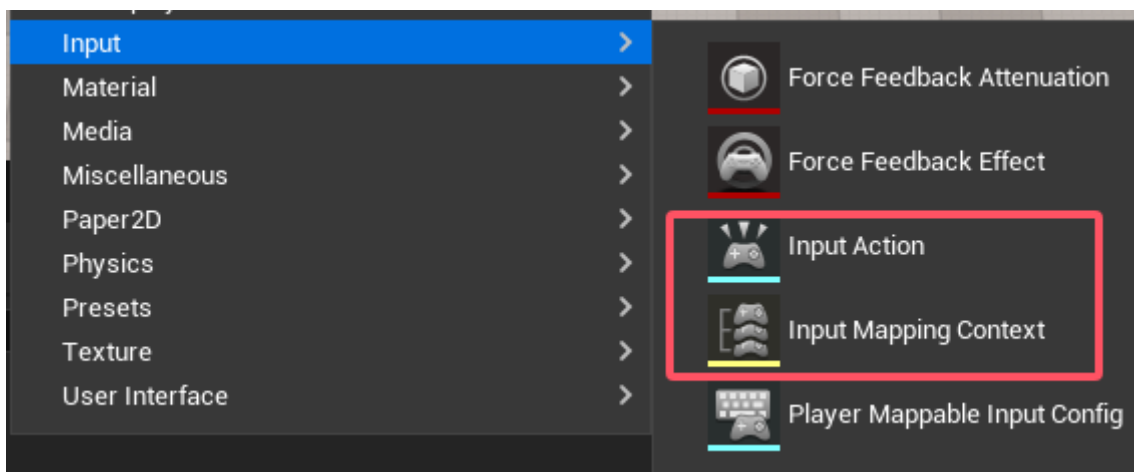
There must be a PlayerStart in your scene, otherwise RTSPawn may not function properly.



Once you have completed the above steps, run the game. The viewport will already be able to move when the mouse touches the edge of the viewport.

≡.Configure key bindings.

The RTSManager plugin's key configuration is based on EnhancedInput. You only need to configure the InputMappingContext and InputAction.



Enter your PlayerController blueprint class (here referred to as BP_RTSPC).

▼ RTSManager		
▼ Input		
RTSMapping Context	None	None
▼ Selected		
Selected Input	None	None
Multi Selected Hotkey	None	None
▼ View		
View Rotate Input	None	None
View Zoom in Out Input	None	None
View Drag with Keys Input	None	None
▼ Move		
Move Input	None	None

Instructions:

- RTSMapping Context: Create your IMC file using InputMappingContext.
- Selected Input: Select and box-select units using (recommended to use the left mouse button).
- Multi Selected Hotkey: Use in conjunction with Selected Input to achieve incremental selection.
- View Rotate Input: Implement pitch and yaw rotation of the camera.
- View Zoom in Out Input: Implement zoom in and out of the camera.
- View Drag with keys Input: Implement drag of the camera.
- Move Input: Implement moving the boxed-selected units by clicking on the ground or other units.

四.Operation parameter description.

In BP_RTSPC, the operation parameters have been subdivided. You can adjust them according to your actual situation.

ViewSettings

Move

Zone Boundary Movement Rate1400.0

Drag

Drag Screen Enable☒

Drag Screen Speed20.0

Drag Reverse☐

Object for Drag Types0 Array elements + -

Rotate

View Init Rotate0.0300.00.0

X0.0

Y300.0

Z0.0

Yaw

View Yaw Rotation Enable☒

View Yaw Rotation Speed50.0

Pitch

View Pitch Rotation Enable☒

View Pitch Rotation Speed10.0

Clamp Pitch Min285.0

Clamp Pitch Max330.0

ZoomInOrOut

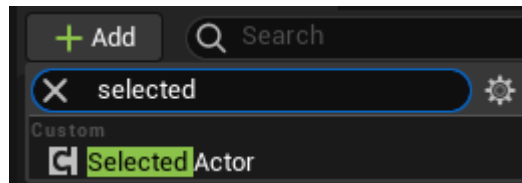
Turn on Zoom in Out☒

Zoom in Out Speed200.0

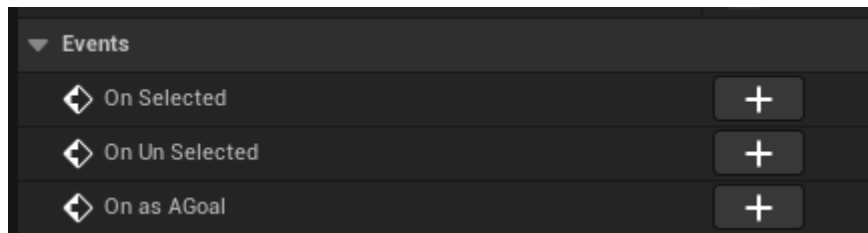
- **Zone Boundary Movement Rate:** The speed at which the viewport moves when the mouse is moved to the edge of the viewport.
- **Drag Screen Enable:** Whether to enable viewport dragging.
- **Drag Screen Speed:** The speed of viewport dragging.
- **Drag Reverse:** Reverse dragging.
- **Object for Drag Types:** The mouse drag window is based on the current mouse recognition of the terrain. If dragging does not work, add the terrain's collision category; otherwise, no need to add.
- **View Init Rotate:** Keep the default unless you need to change the initial camera rotation.
- **View Yaw Rotation Enable:** Whether to enable Yaw rotation of the camera.
- **View Yaw Rotation Speed:** The speed of Yaw rotation of the camera.
- **View Pitch Rotation Enable:** Whether to enable Pitch rotation of the camera.
- **View Pitch Rotation Speed:** The speed of Pitch rotation of the camera.
- **Clamp Pitch Min:** The minimum value for Pitch rotation restriction.
- **Clamp Pitch Max:** The maximum value for Pitch rotation restriction.
- **Turn on Zoom in Out:** Whether to enable zooming in and out of the camera.
- **Zoom in Out Speed:** The speed of zooming in and out of the camera.

五.Specify Actor response to box selection functionality

In addition to specifying the box selection hotkeys in the PlayerController, we also need to specify which Actors respond to box selection. To respond to the box selection functionality, you need to add the Selected Actor component to the corresponding Actors.



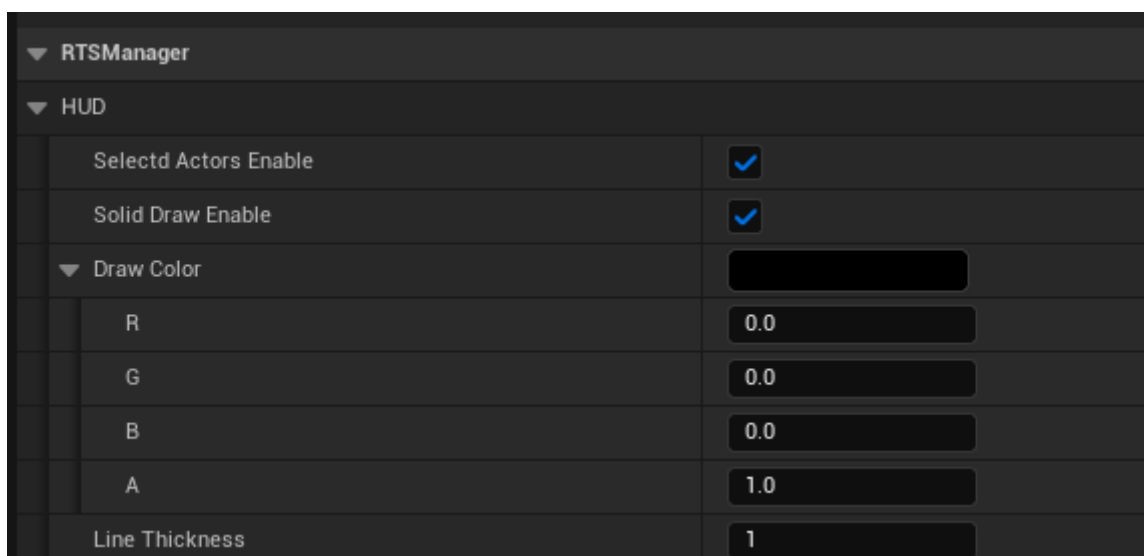
After adding this component, we need to implement three delegate events for this component to accept the corresponding responses.



- **On Selected**: When this unit is selected, this event is triggered.
- **On Un Selected**: When this unit is deselected, this event is triggered.
- **On as AGoal**: When another unit targets this unit, this event on this unit is triggered and the other unit is passed.

5.1 Additional notes on box selection

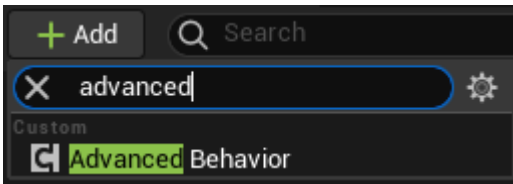
In the HUD, under the RTSManager category, there are additional settings for box selection.



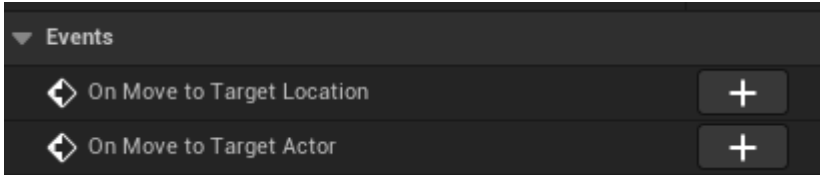
- **Selected Actors Enable**: Whether to enable box selection.
- **Solid Draw Enable**: True for a solid box, false for an empty box.
- **Draw Color**: Box selection color (note that Alpha cannot be 0, otherwise it will be transparent and invisible).
- **Line Thickness**: Box selection line thickness.

六. Formation

RTSManager provides the Advanced Behavior component. When multiple units move, this component assigns a coordinate in the formation to each unit.



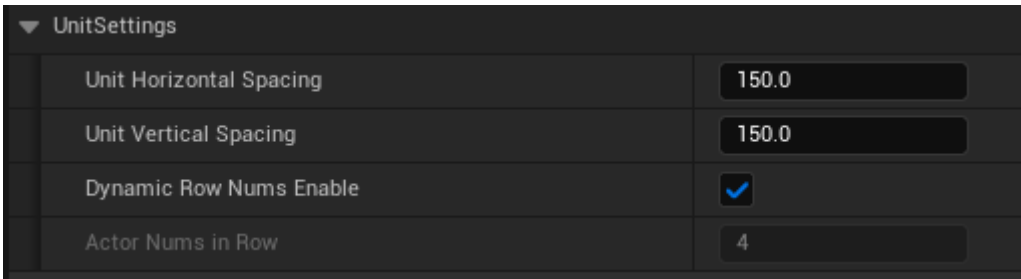
After adding this component, when the click-to-move action is triggered, it will automatically accept the assigned coordinate or target.



- **On Move to Target Location**: Coordinate point to move to.
- **On Move to Target Actor**: Actor to move to.

This component only assigns formation coordinates and does not implement movement. To achieve movement, you can use Unreal Engine's built-in navigation system and AI controller, or any custom or third-party navigation or pathfinding module.

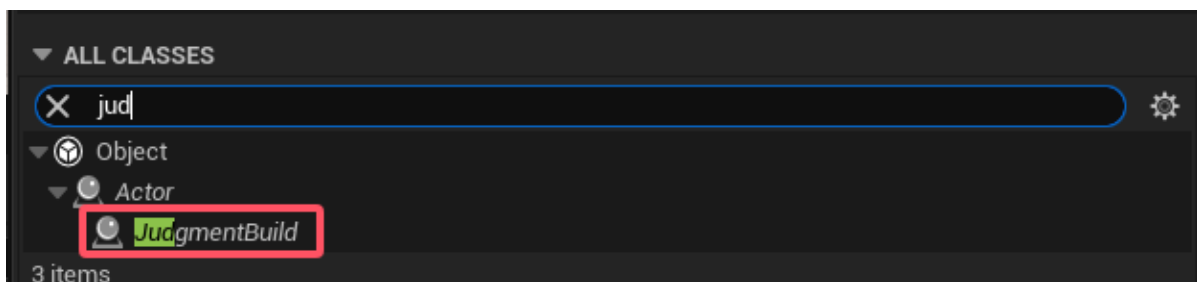
The main function of this component is to implement array functionality for the movement of multiple boxed-selected targets. The sorting or modification of the array can be done in the PlayerController options.



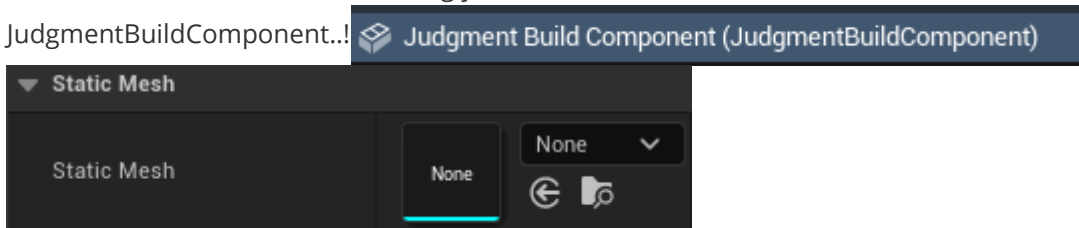
- **Unit Horizontal Spacing**: The horizontal spacing between each unit in the formation.
- **Unit Vertical Spacing**: The vertical spacing between each unit in the formation.
- **Dynamic Row Nums Enable**: Whether to enable dynamic formation (recommended to enable).
- **Actor Nums in Row**: The maximum number of units per row in a static formation (only takes effect when Dynamic Row Nums Enable=false).

七. Construction

The RTSManager plugin provides a JudgmentBuild class for pre-construction judgment.

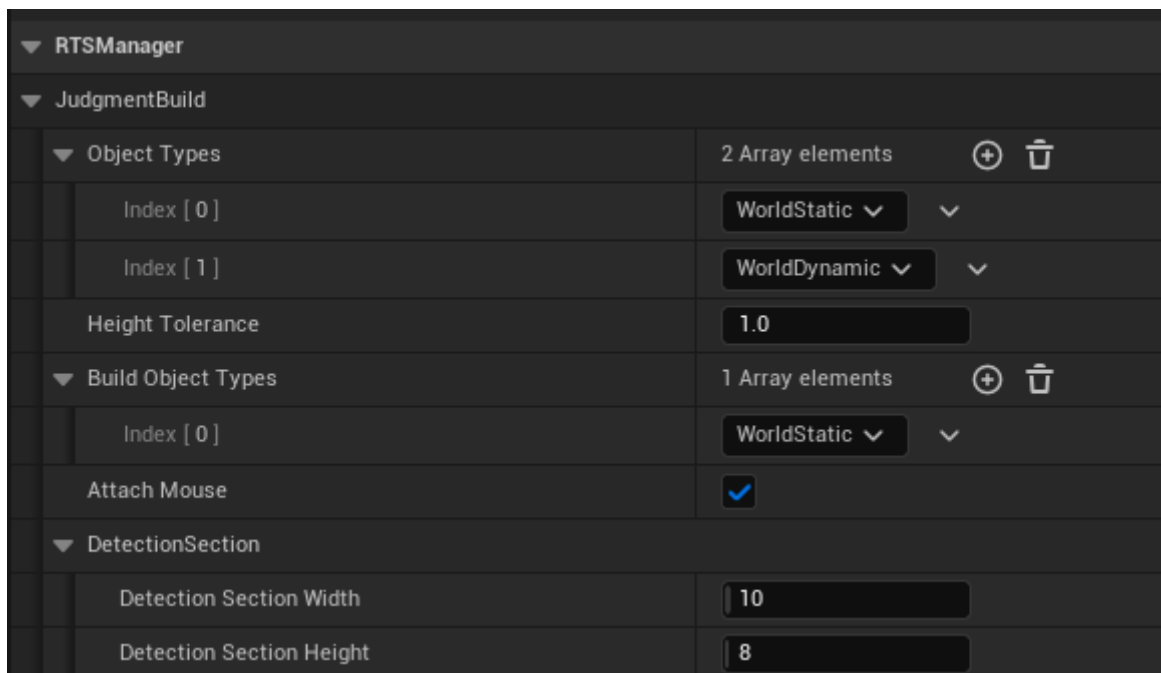


you only need to create a blueprint class that inherits from JudgmentBuild. For example, create a blueprint class that inherits from it and name it BP_JudgmentBuild. Enter this blueprint and you need to set the model of the building you want to construct to the StaticMesh of JudgmentBuildComponent...



Note that please ensure the center point of the model is at the bottom center of the model.

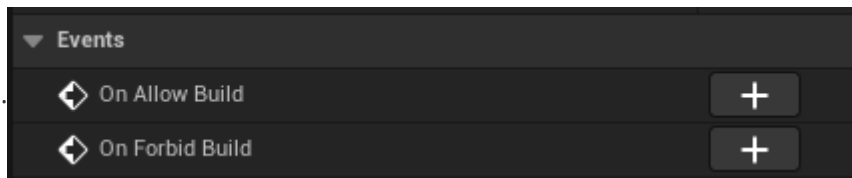
Under the RTSManager category, there are the following options.



- **Object Types:** Which collision types will interfere with the construction of the building.
- **Height Tolerance:** The height tolerance for construction (the height difference of the terrain).
- **Build Object Types:** On which collision types of terrain the construction judgment can be performed.
- **Attach Mouse:** When enabled, this blueprint will attach to the mouse.
- **Detection Section Width:** The fineness of terrain detection (horizontal).
- **Detection Section Height:** The fineness of terrain detection (vertical).

Note that JudgmentBuild is not the building you are constructing; it simulates your current building to judge constructibility and returns the result to your actual building Actor to complete the creation.

Bind these two delegates.



- **On Allow Build**: Triggered when creation is allowed.
- **On Forbid Build**: Triggered when creation is prohibited.

Additionally, do not use the Actor's own rotation function to rotate JudgmentBuild. Instead, use the following function, which allows for more precise detection during rotation.



Finally, use the construction function to perform the construction.



When construction is possible, the Return Value returns true and provides Location and Rotation for use in creating the actual Actor. Below is a simple usage example. BP_RealBuild is the blueprint of the building we want to display in the scene.

