# Experiment-6

Learning the Blueprint Scripting System
in Unreal Engine

Nidhi Kumari
Monisha Govindaraj
3 October 2023

Blueprints in Unreal Engine are a visual scripting system that allows developers and artists to create gameplay mechanics, interactions, and logic without writing code. With Unreal Engine 5, many features and functionalities have been introduced or improved upon, enabling creators to develop content for various applications, including gaming.

Blueprints, we can keep track of health, energy, score, and anything else we need to keep track of. And we can specify game logic, like what are the requirements for completing a puzzle, what happens when that puzzle is complete,
what happens when I shoot an enemy, what happens if a 2-ton piano falls on me

There are two main types of Blueprints - The Level Blueprint and Blueprint Classes.

A Level Blueprint is used to hold data and instructions for a particular Level. So it
might hold data such as the time remaining to complete the Level, or the number of keys you've
collected in that Level, and so on. It would also be used to store instructions that pertain only
to that Level. For example, let's say there was a spot in a Level where there was a bridge and when
the player crosses that bridge, a meteor will fly across the sky. If that's a one-time unique
occurrence just for that spot in that Level, it would make sense to store those instructions in the Level Blueprint for that Level. Blueprint Classes are a way to turn any
Actor or asset into a Blueprint. This allows us to create objects with custom traits and behaviors.
So let's say that I was building a haunted house. And let's say I wanted to have a chair that
floated up and down, as if it were possessed or something. Let's also say that I want my character
to be able to shoot the chair and if I hit the chair enough with my projectiles, it will destroy
the chair. I can achieve all of this by creating a Blueprint Class out of this chair mesh.

So within the Blueprint, I could specify that the chair should move straight up and down,
over and over again, starting from wherever it is placed in the Level. I could also specify
that the Chair should contain a variable called Health with a default value of 100.
I could also say that any time the Chair was hit by a projectile, that 10 should be subtracted from
its health. And finally, I could specify that if the Health of the chair ever gets to 0 or below,
that the Chair should be destroyed. So one of the great things about Blueprint Classes
is, you can use them to create as many copies, or instances, of your creation as you want.

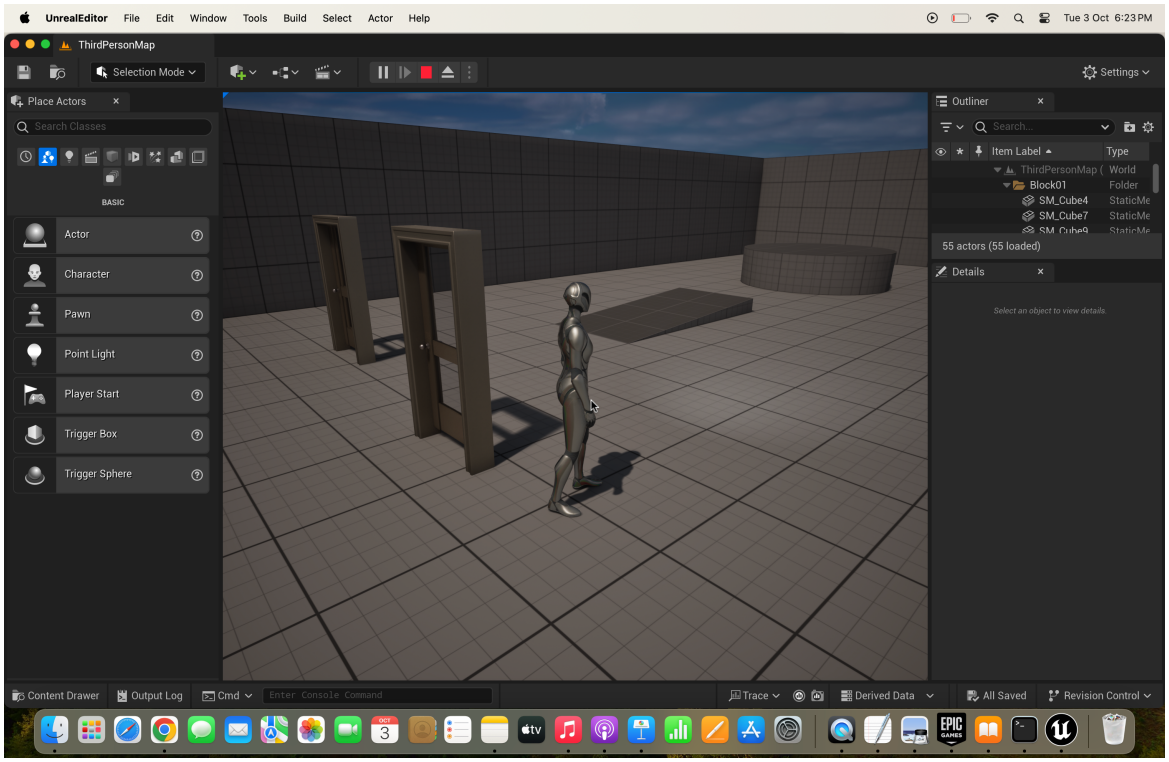
Here are some of the key commands and functionalities of Blueprints in Unreal Engine 5 and how they enable content creation for gaming and other applications:

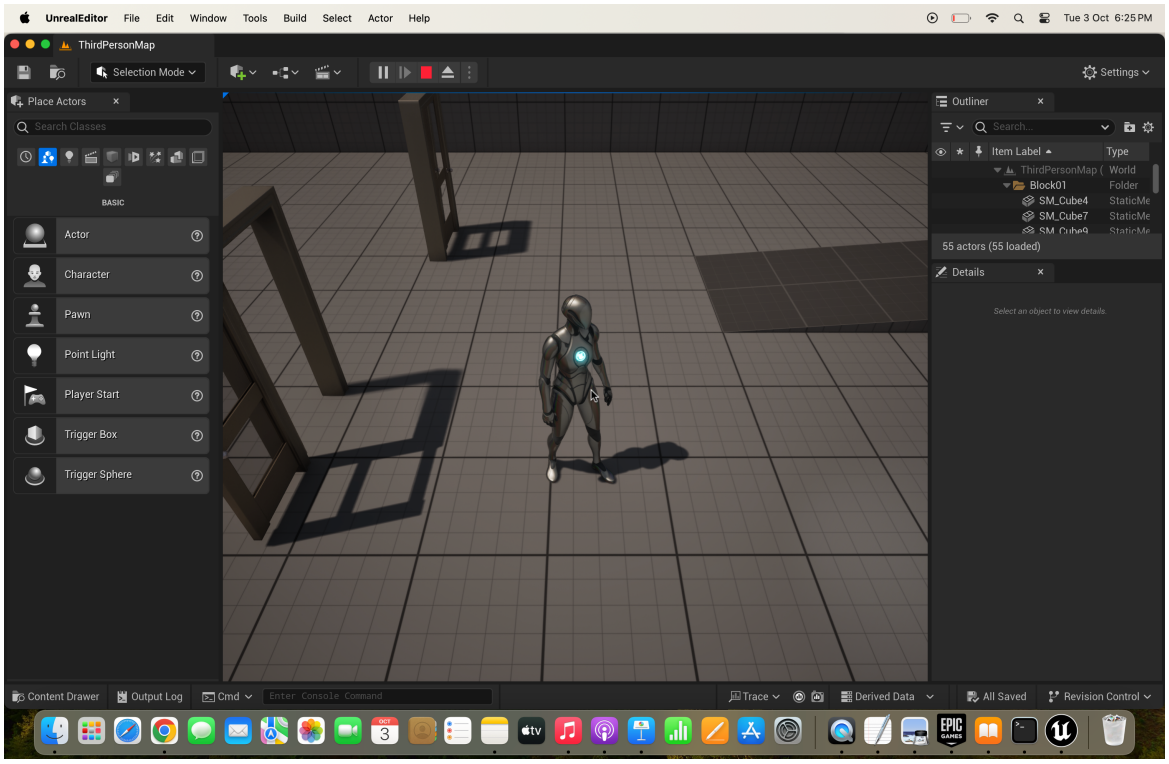
- **Visual Scripting**: Blueprints provide a node-based visual scripting interface that allows you to create and manipulate game logic by connecting nodes together. This visual approach simplifies the process of creating complex behaviors and interactions in your game.
- **Event Graph**: The Event Graph is the primary workspace in Blueprints where you define the logic for your game objects' behaviors. You can respond to events such as player input, collisions, and timers, and trigger actions in response to these events.
- **Variables**: You can create and manage variables within Blueprints to store and manipulate data. This is essential for maintaining game state, tracking scores, and managing character attributes.

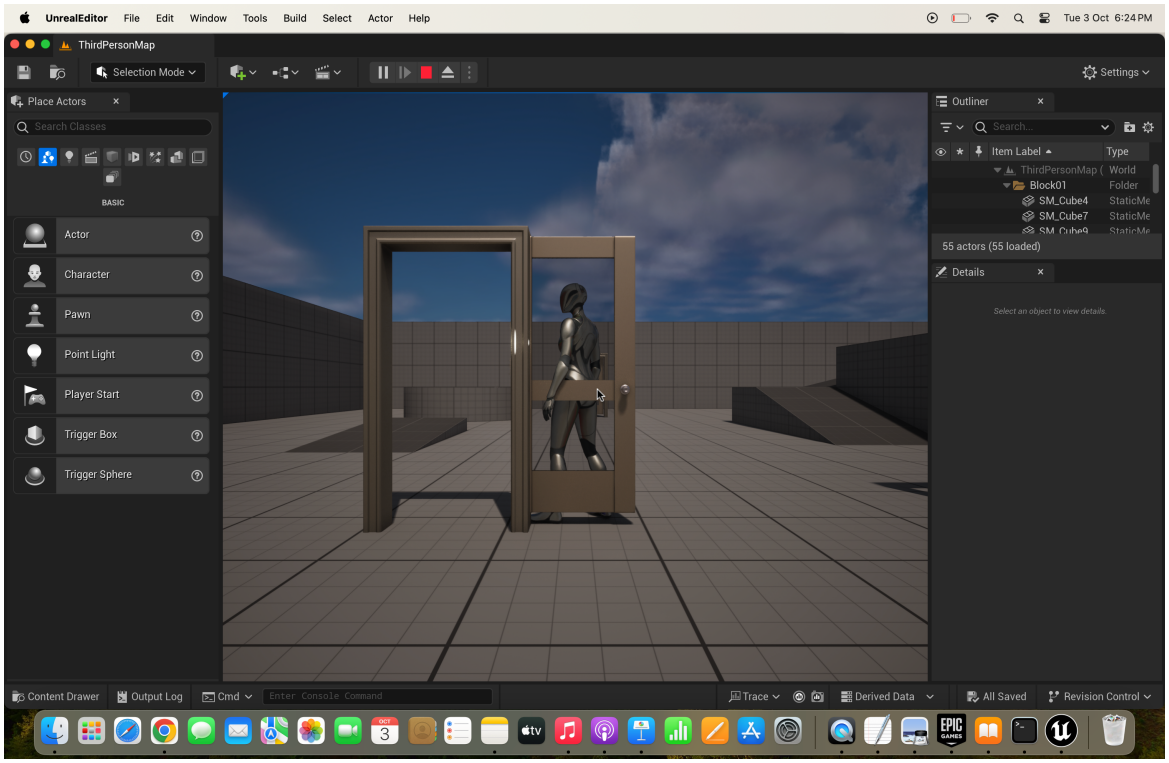
- **Functions and Macros**: Blueprints allow you to encapsulate sets of nodes into functions and macros, which can be reused throughout your project. This modularity simplifies code organization and encourages best practices.
- **Flow Control**: Blueprints support various flow control nodes like conditionals (if/else), loops (for/while), and switches to control the execution flow of your logic.
- **Communication with C++**: Unreal Engine provides the ability to bridge between Blueprints and C++ code. You can create C++ functions and expose them to Blueprints, allowing for seamless integration of custom code when needed.
- **Animation and Physics**: Blueprints enable you to create and control animations and physics simulations for characters and objects. You can blend animations, set up ragdoll physics, and control the behavior of characters.
- **User Interface (UI) Design**: You can design UI elements, menus, and HUDs using Blueprints. This is crucial for creating user interfaces in games and applications.
- **Level Blueprint**: Each level in Unreal Engine can have its own Blueprint, known as the Level Blueprint. This allows you to set up level-specific logic and interactions.
- **Blueprint Interfaces**: You can define custom interfaces in Blueprints, allowing different actors to communicate and interact with each other in a standardized way.
- **Debugging Tools**: Unreal Engine provides debugging tools like breakpoints, watch variables, and execution traces to help you identify and fix issues in your Blueprints.
- **VR and AR Development**: Blueprints support virtual reality (VR) and augmented reality (AR) development, making it easier to create immersive experiences for these platforms.
- **Optimization**: Unreal Engine 5 includes profiling and optimization tools to help you identify performance bottlenecks in your Blueprints and improve the overall performance of your game or application.

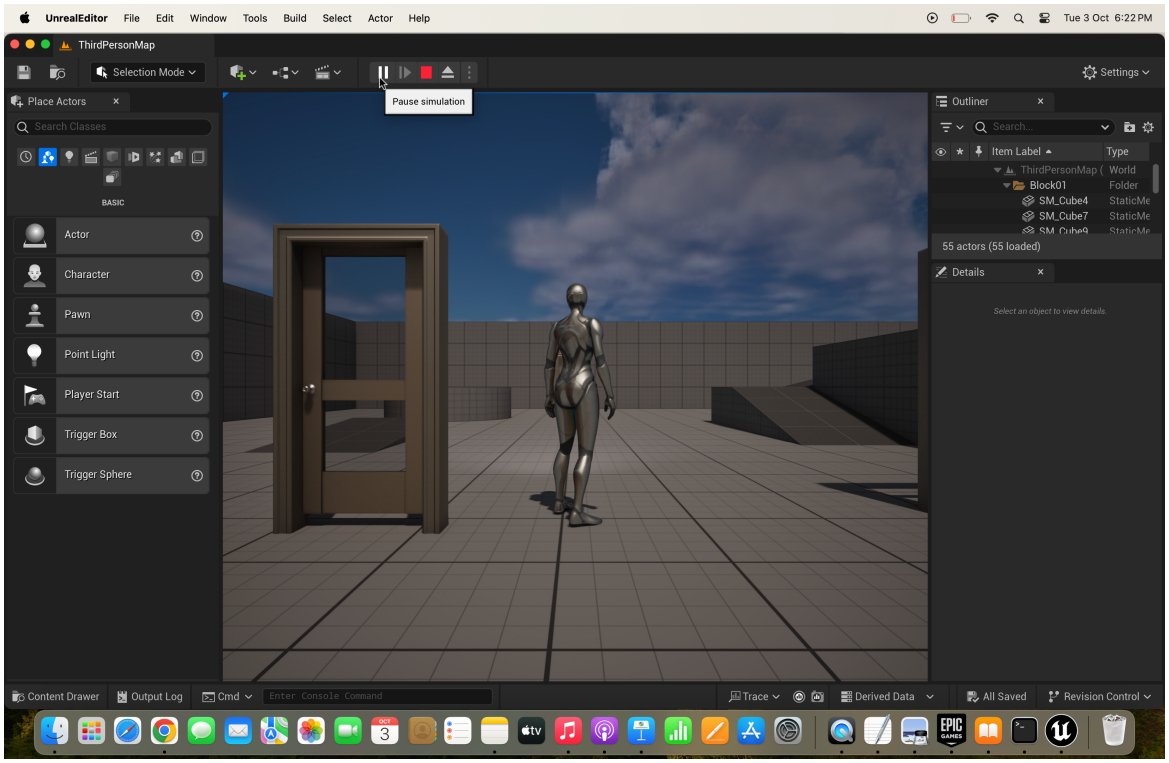
By using these Blueprint tools and functionalities, creators can efficiently design, prototype, and develop game mechanics, interactive experiences, and simulations. Unreal Engine 5's Blueprints empower a wide range of users, including artists, designers, and programmers, to collaborate and bring their creative visions to life, ultimately resulting in engaging and immersive gaming and interactive applications.

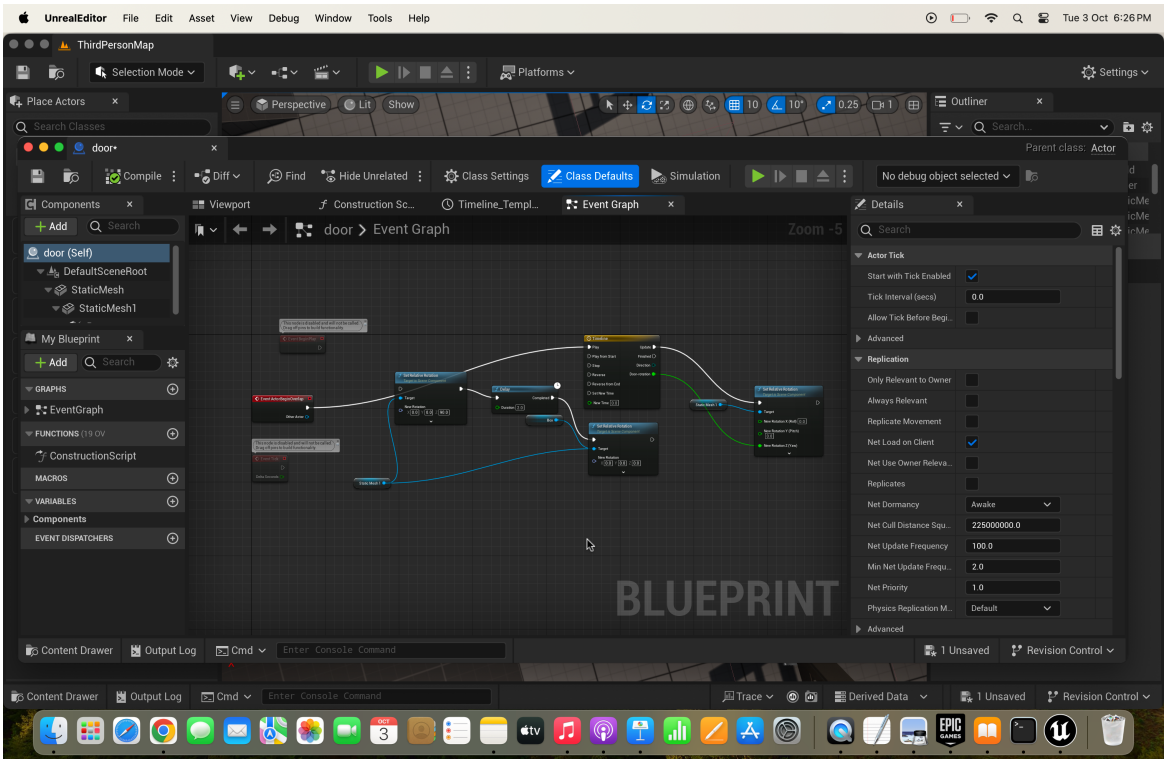
Example: I added character which can run to every door. When character reaches to door it gets open, when it moves away door gets closed. I added light, when character moves/runs we can see light focus on wall.

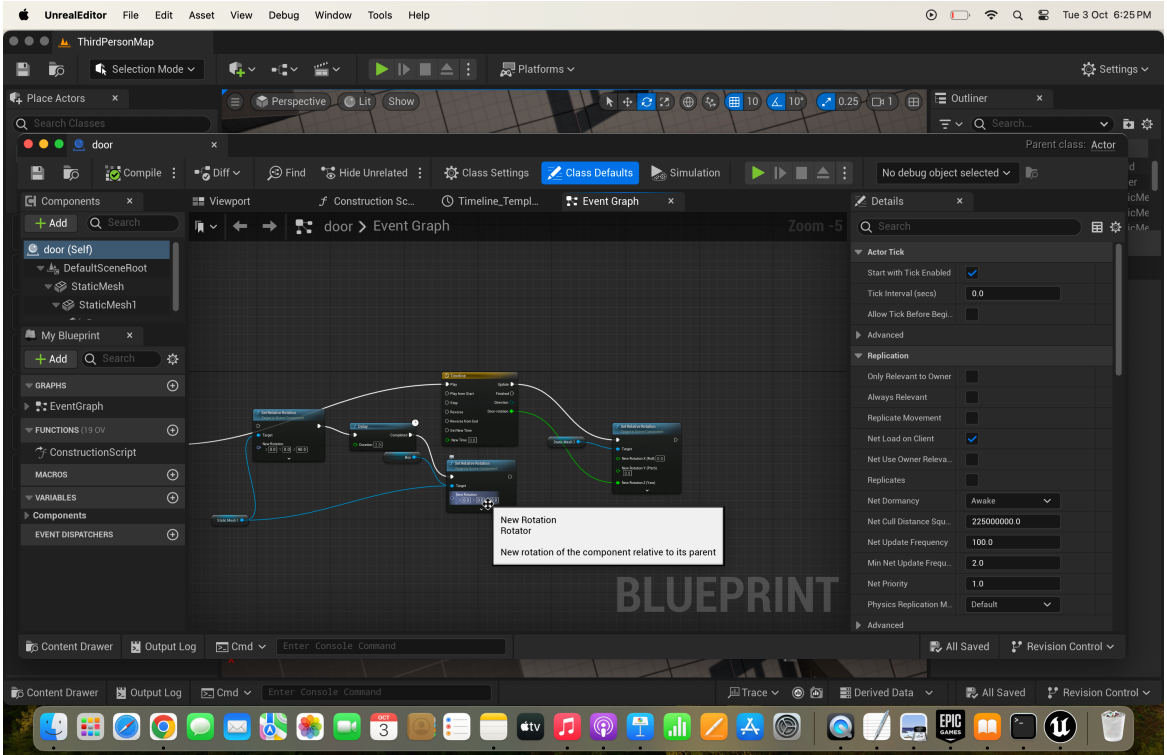



UnrealEditor File Edit Window Tools Build Select Actor Help
Tue 3 Oct 6:24 PM
ThirdPersonMap
Selection Mode
Settings
Place Actors
Search Classes
BASIC
Actor
Character
Pawn
Point Light
Player Start
Trigger Box
Trigger Sphere
Outliner
Item Label
Type
ThirdPersonMap ( World
Block01 Folder
SM_Cube4 StaticMe
SM_Cube7 StaticMe
55 actors (55 loaded)
Details
Select an object to view details.
Content Drawer
Output Log
Cmd
Enter Console Command
Trace
Derived Data
All Saved
Revision Control


UnrealEditor File Edit Window Tools Build Select Actor Help
Tue 3 Oct 6:22 PM
ThirdPersonMap
Selection Mode
Pause simulation
Settings
Place Actors
Search Classes
BASIC
Actor
Character
Pawn
Point Light
Player Start
Trigger Box
Trigger Sphere
Outliner
Item Label
Type
ThirdPersonMap ( World
Block01 Folder
SM_Cube4 StaticMe
SM_Cube7 StaticMe
55 actors (55 loaded)
Details
Select an object to view details.
Content Drawer
Output Log
Cmd
Enter Console Command
Trace
Derived Data
All Saved
Revision Control


UnrealEditor File Edit Asset View Debug Window Tools Help
Tue 3 Oct 6:26 PM
ThirdPersonMap
door
Compile Diff Find Hide Unrelated Class Settings Class Defaults Simulation
No debug object selected
Components
door (Self)
DefaultSceneRoot
StaticMesh
StaticMesh1
My Blueprint
GRAPHS
EventGraph
FUNCTIONS
ConstructionScript
MACROS
VARIABLES
EVENT DISPATCHERS
Viewport
Construction Sc...
Timeline_Templ...
Event Graph
door > Event Graph
Details
Actor Tick
Start with Tick Enabled
Tick Interval (secs)
Allow Tick Before Begi...
Advanced
Replication
Only Relevant to Owner
Always Relevant
Replicate Movement
Net Load on Client
Net Use Owner Releva...
Replicates
Net Dormancy
Awake
Net Cull Distance Squ...
225000000.0
Net Update Frequency
100.0
Min Net Update Frequ...
2.0
Net Priority
1.0
Physics Replication M...
Default
BLUEPRINT
Content Drawer
Output Log
Cmd
Enter Console Command
1 Unsaved
Revision Control
Trace
Derived Data


UnrealEditor File Edit Asset View Debug Window Tools Help
Tue 3 Oct 6:25 PM
ThirdPersonMap
door
Compile Diff Find Hide Unrelated Class Settings Class Defaults Simulation
No debug object selected
Components
door (Self)
DefaultSceneRoot
StaticMesh
StaticMesh1
My Blueprint
GRAPHS
EventGraph
FUNCTIONS
ConstructionScript
MACROS
VARIABLES
EVENT DISPATCHERS
Viewport
Construction Sc...
Timeline_Templ...
Event Graph
door > Event Graph
Details
Actor Tick
Start with Tick Enabled
Tick Interval (secs)
Allow Tick Before Begi...
Advanced
Replication
Only Relevant to Owner
Always Relevant
Replicate Movement
Net Load on Client
Net Use Owner Releva...
Replicates
Net Dormancy
Awake
Net Cull Distance Squ...
225000000.0
Net Update Frequency
100.0
Min Net Update Frequ...
2.0
Net Priority
1.0
Physics Replication M...
Default
New Rotation
Rotator
New rotation of the component relative to its parent
BLUEPRINT
Content Drawer
Output Log
Cmd
Enter Console Command
All Saved
Revision Control
Trace
Derived Data