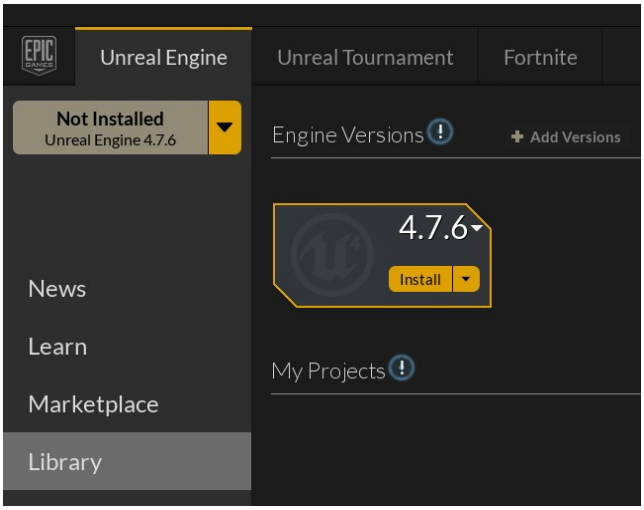
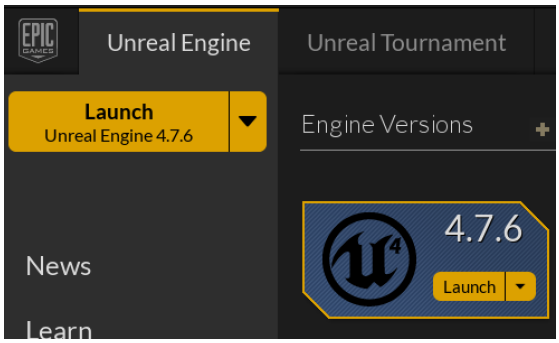
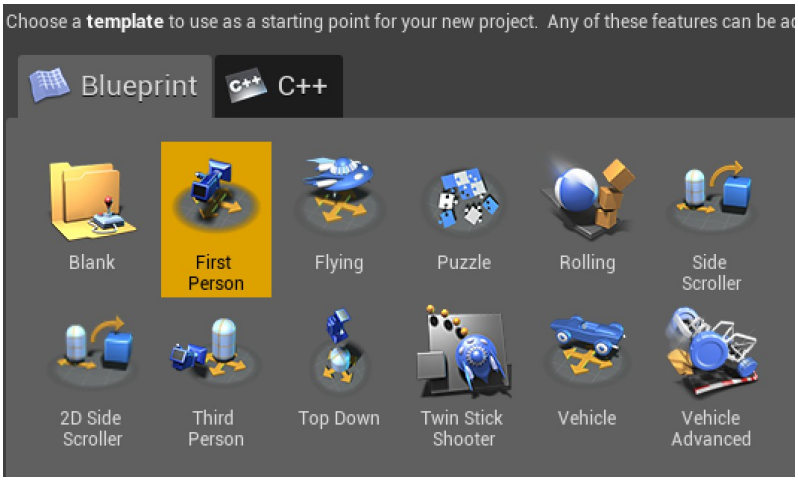
Chapter 1:Object Interaction with Blueprints

Creating a project and the first level

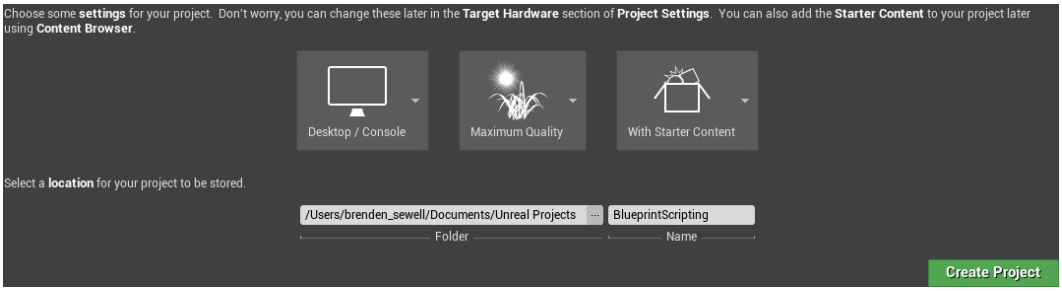




Setting a template for a new project

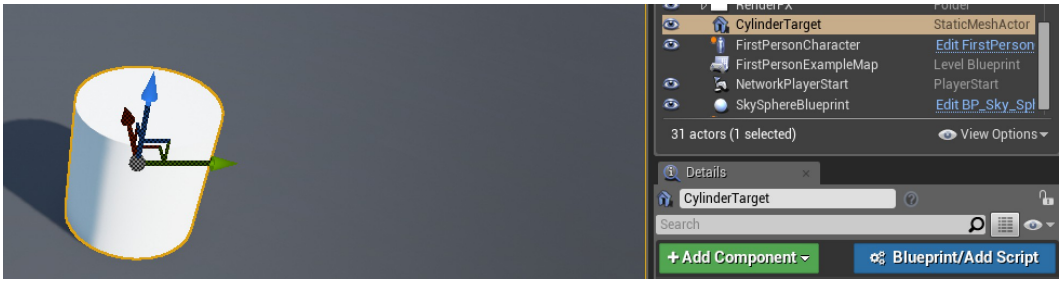


Making sense of the project settings



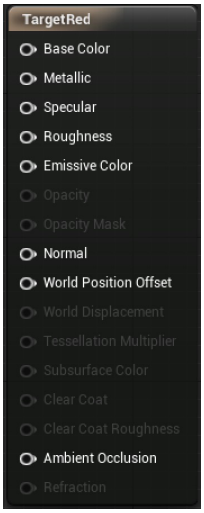
Creating the project

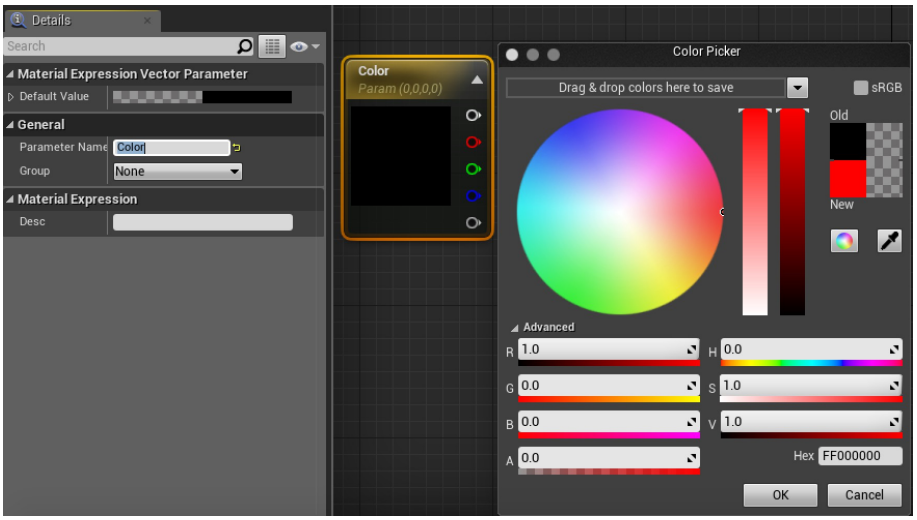
Adding objects to our level

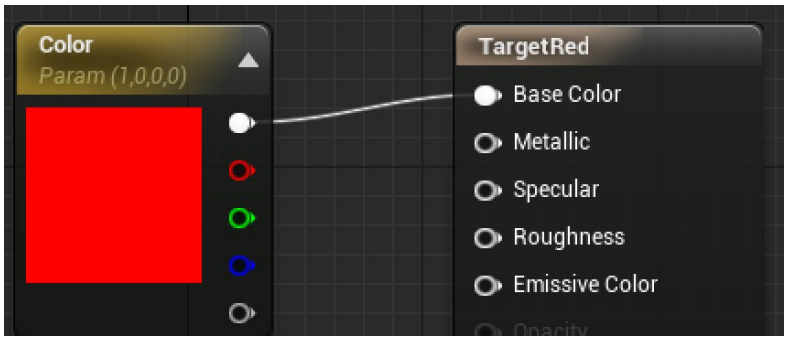


Exploring materials

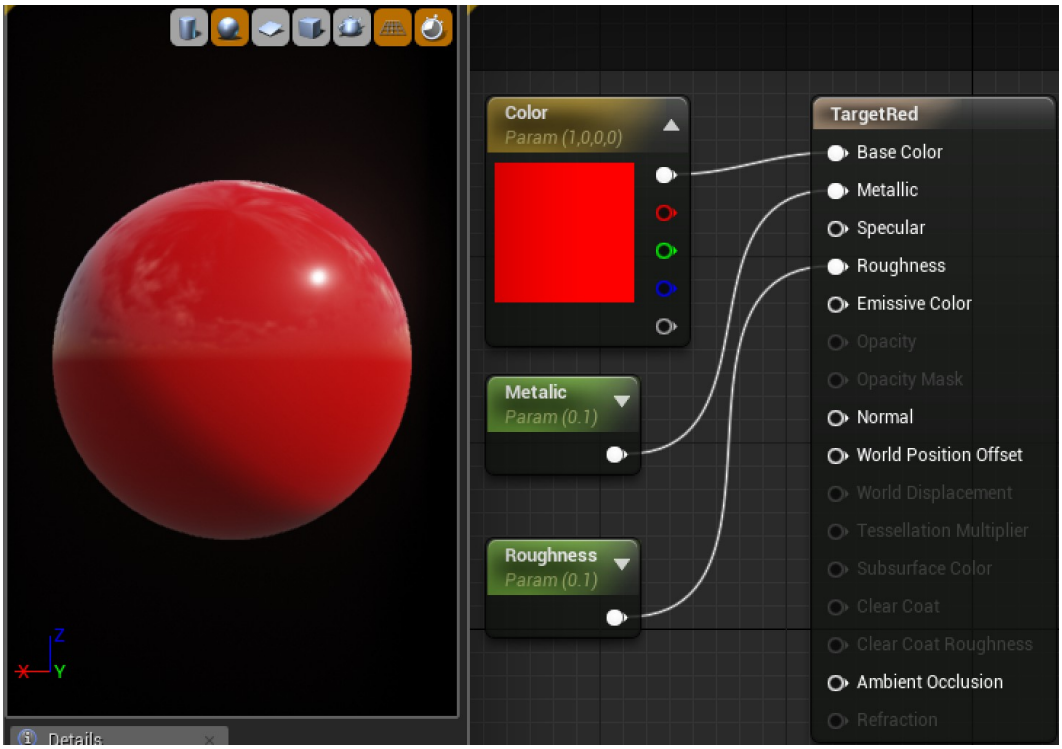
Material Properties and Blueprint Nodes



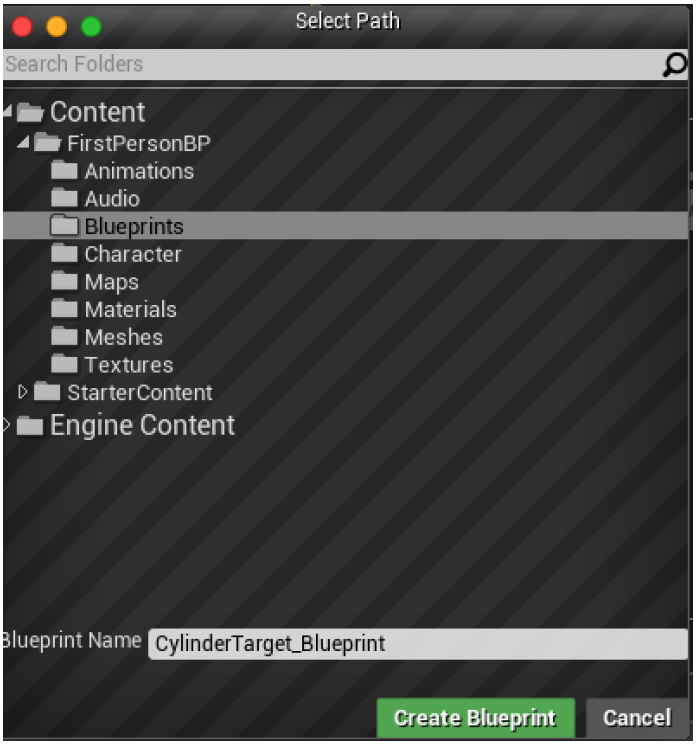

Details
Search
Material Expression Vector Parameter
Default Value
General
Parameter Name
Color
Group
None
Material Expression
Desc
Color
Param (0,0,0,0)
Color Picker
Drag & drop colors here to save
sRGB
Old
New
Advanced
R 1.0
G 0.0
B 0.0
A 0.0
H 0.0
S 1.0
V 1.0
Hex FF000000
OK
Cancel


Color
Param (1,0,0,0)
TargetRed
Base Color
Metallic
Specular
Roughness
Emissive Color

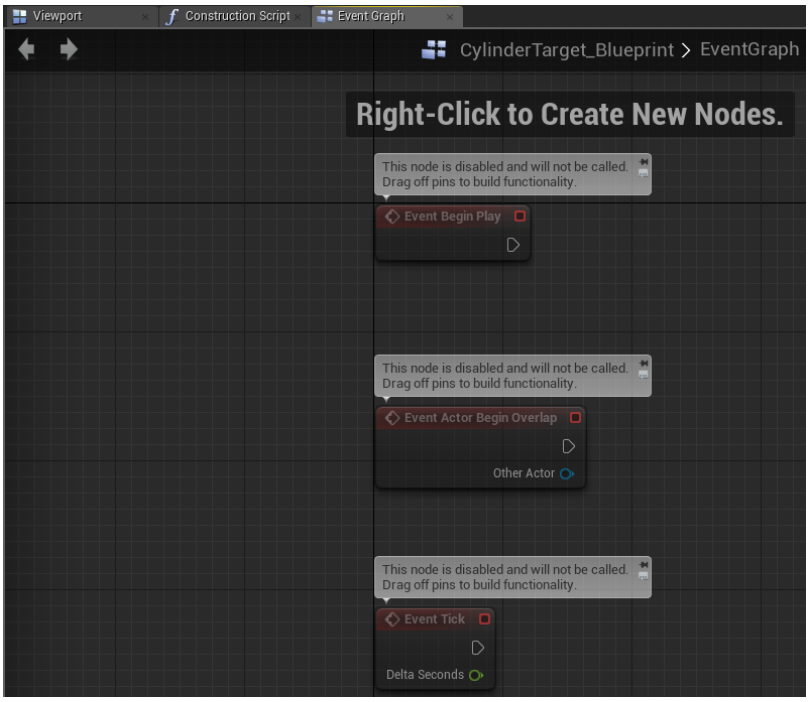
Adding substance to our material



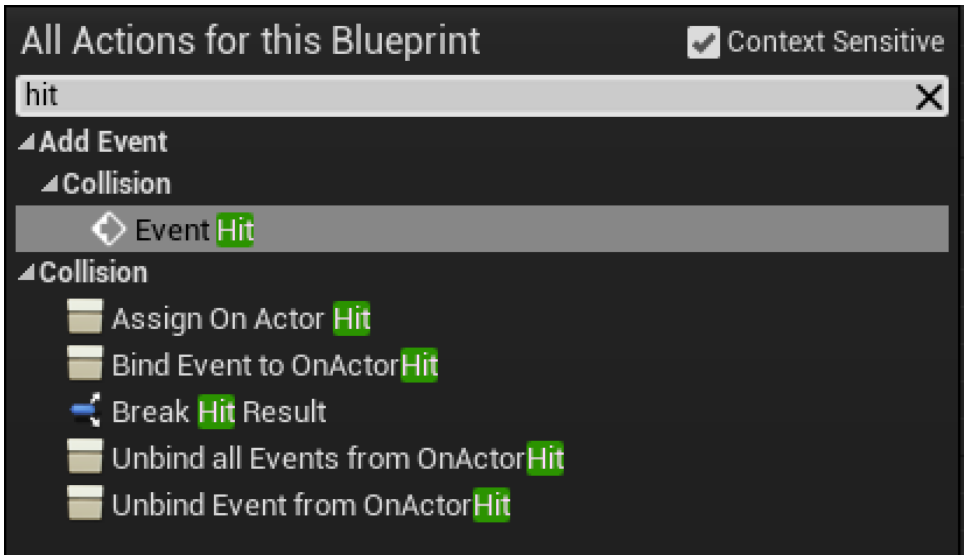
Creating our first Blueprint

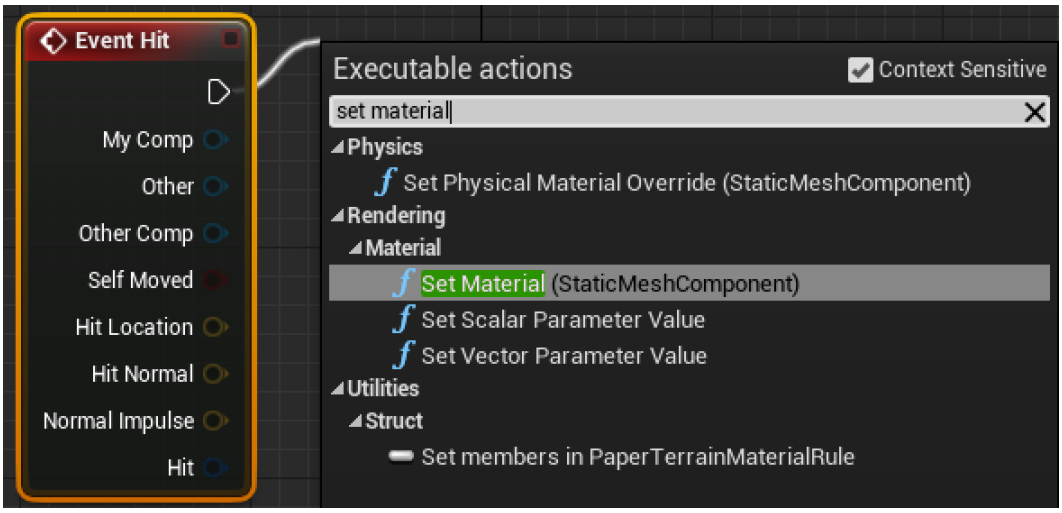


Exploring the Event Graph panel

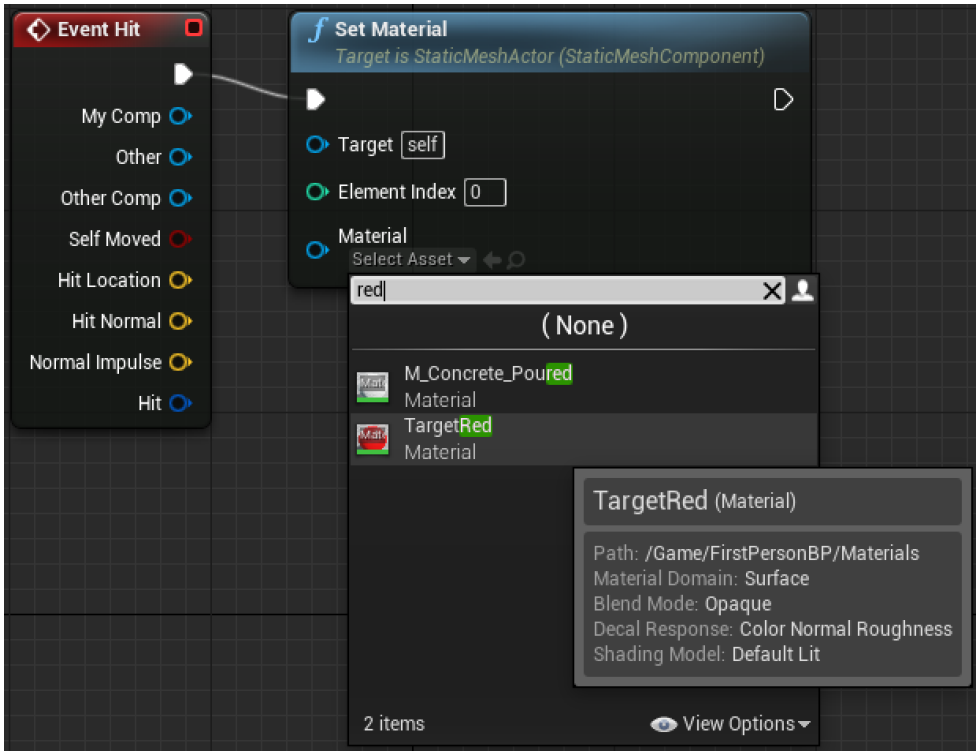


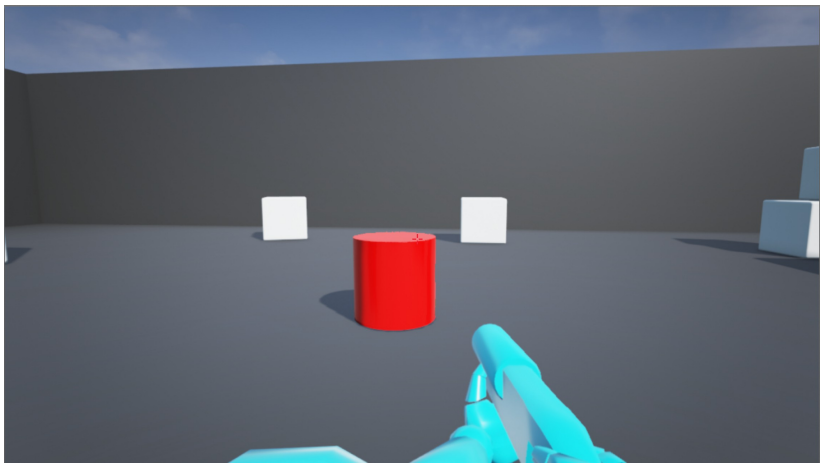
Detecting a hit



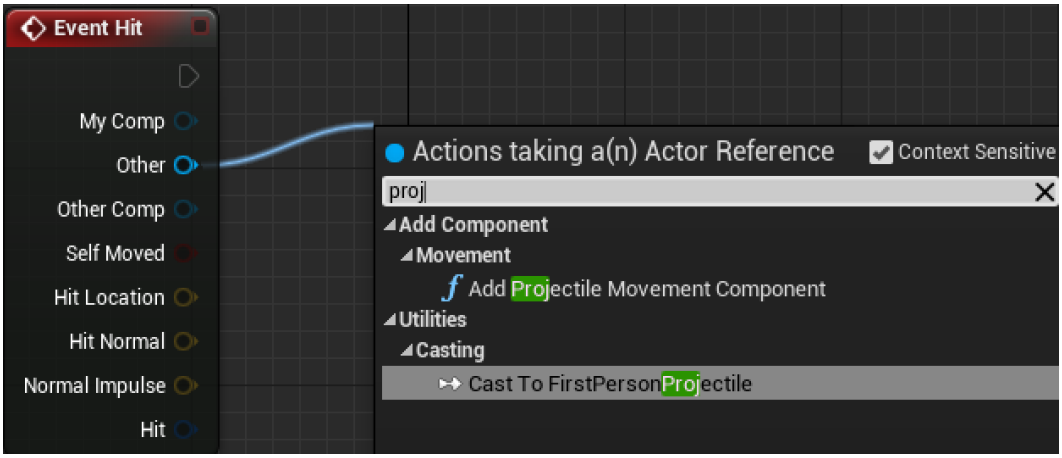


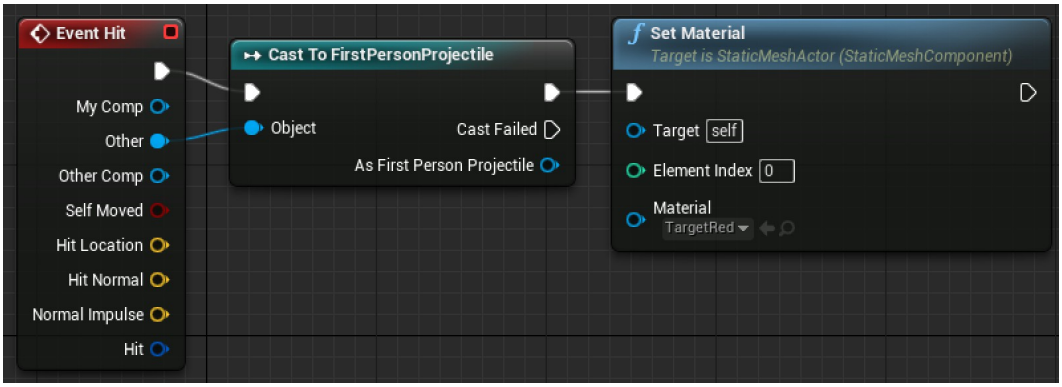
Swapping a material





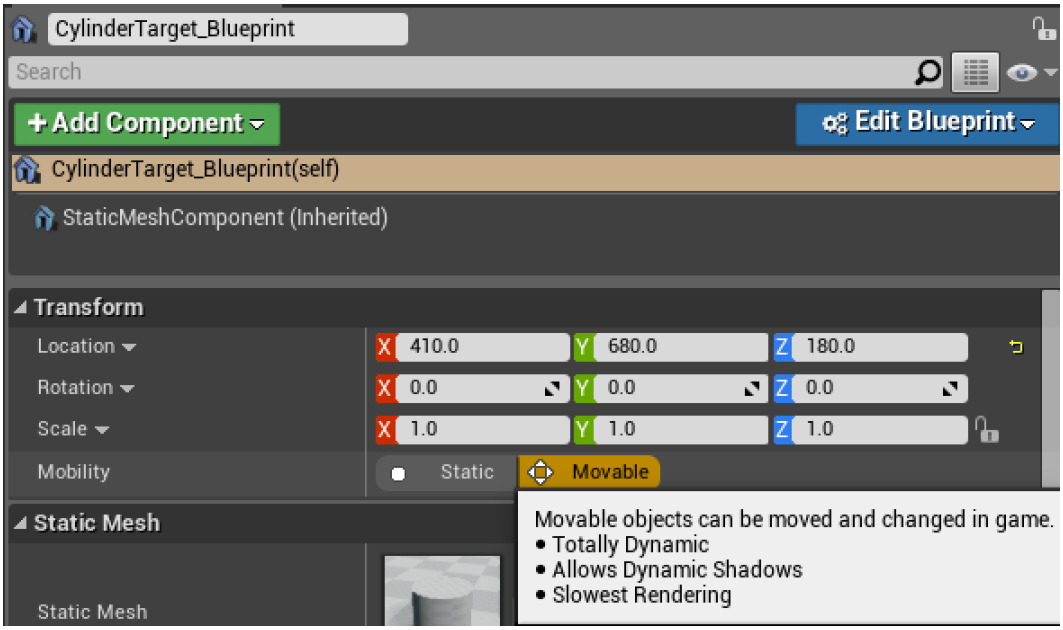
Improving the Blueprint

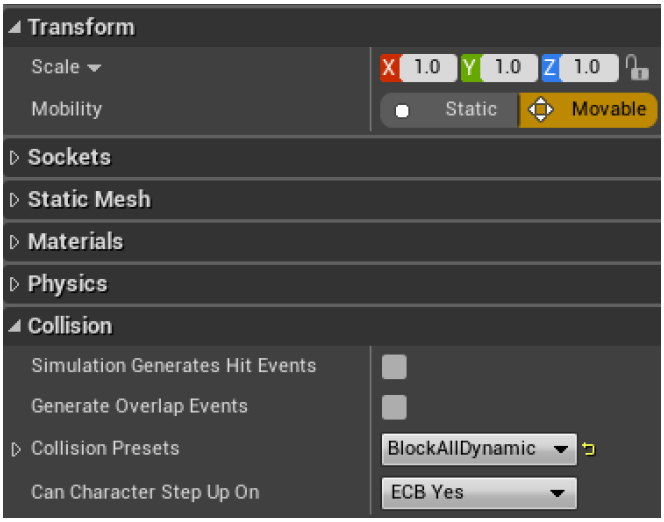




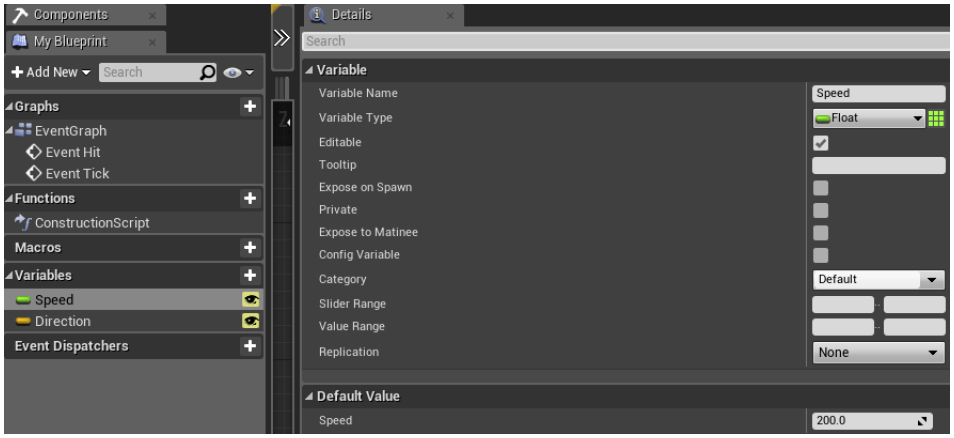
Adding movement

Changing actor mobility and collision

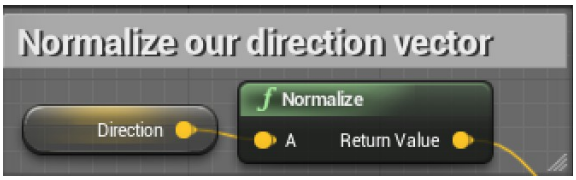




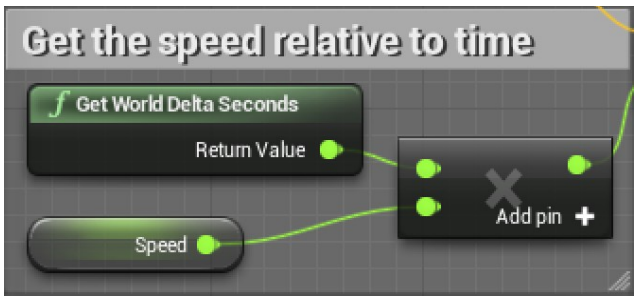
Storing data with variables



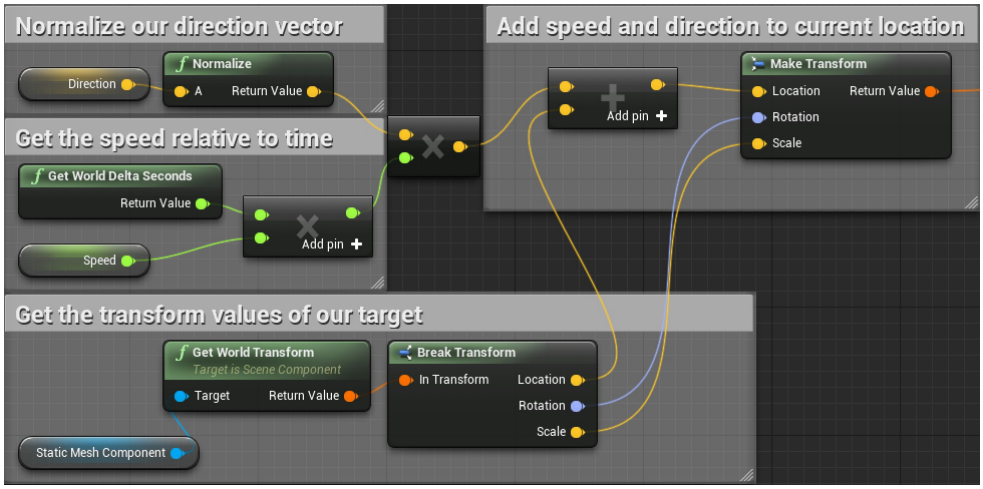
Readying direction for calculations



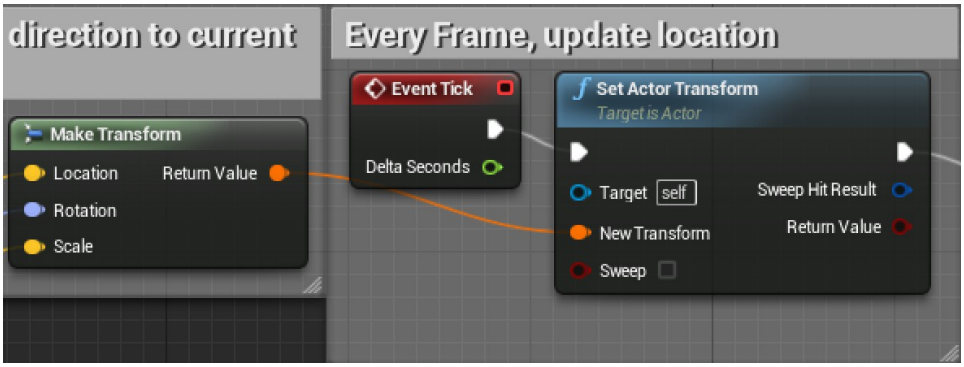
Getting relative speed using delta time



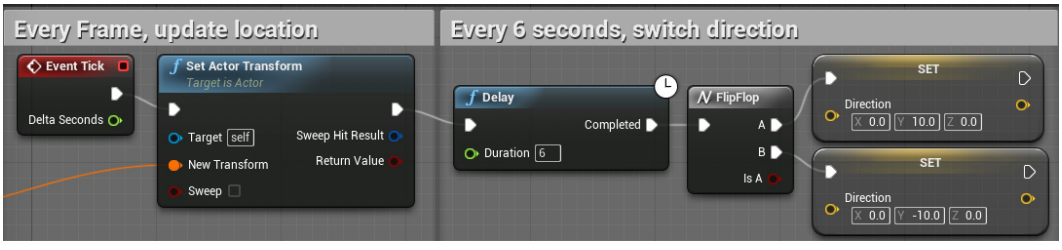
Translating existing location



Updating location



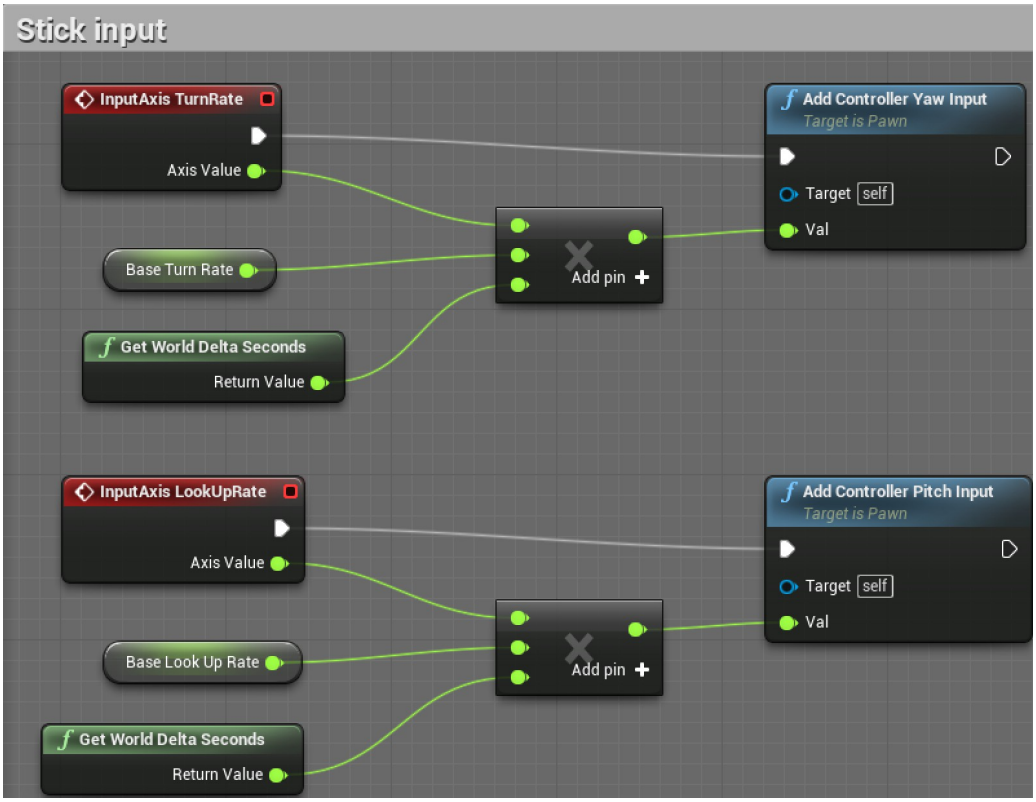
Changing direction

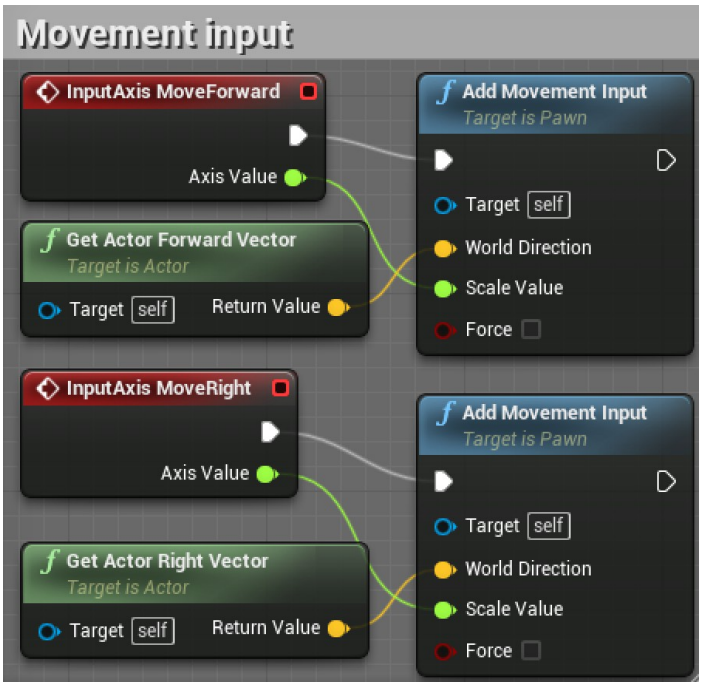


Chapter 2: Enhancing Player Abilities

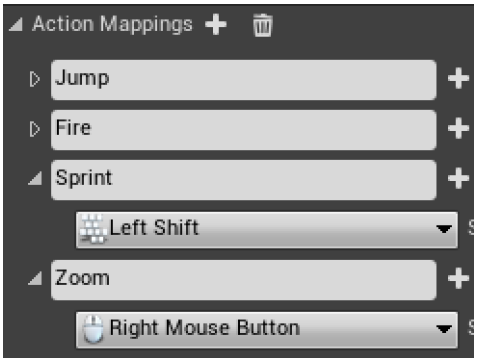
Adding the running functionality by extending a Blueprint

Breaking down the Blueprint character movement

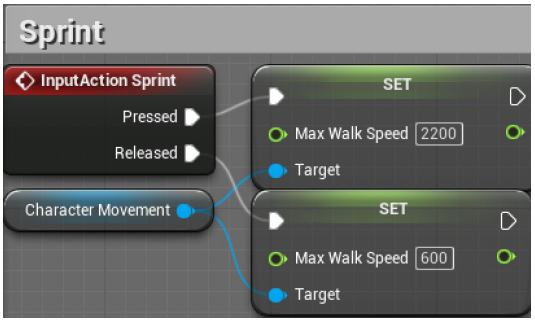


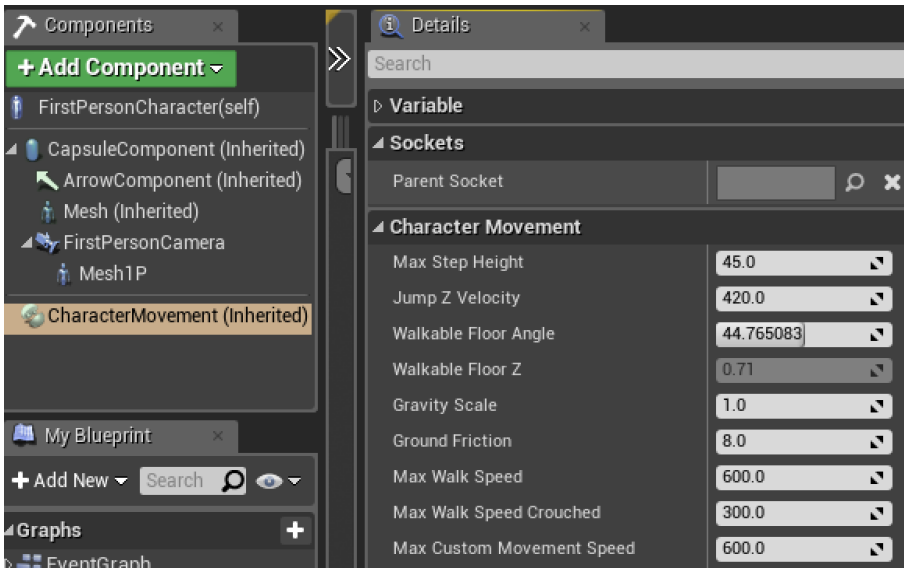


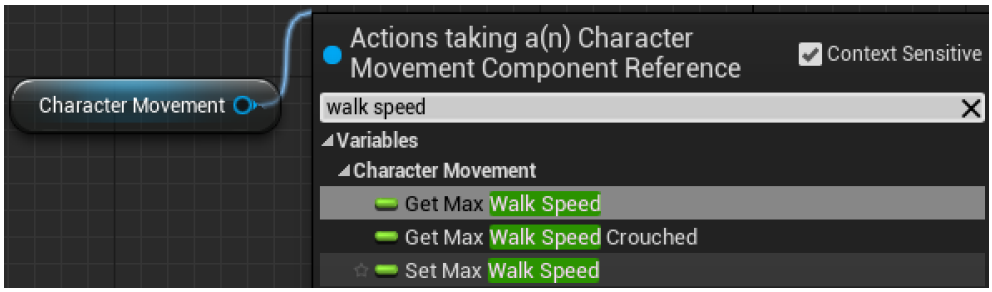
Customizing control inputs



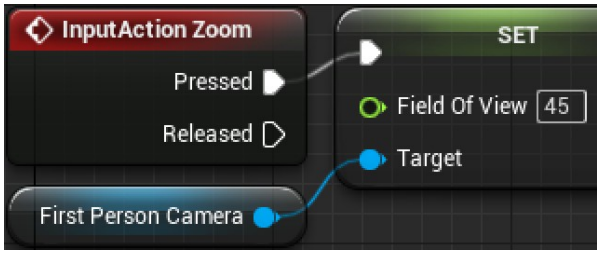
Adding a sprint ability



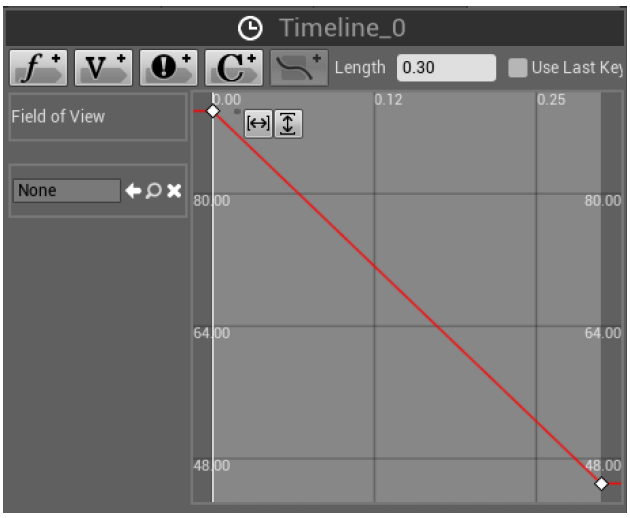




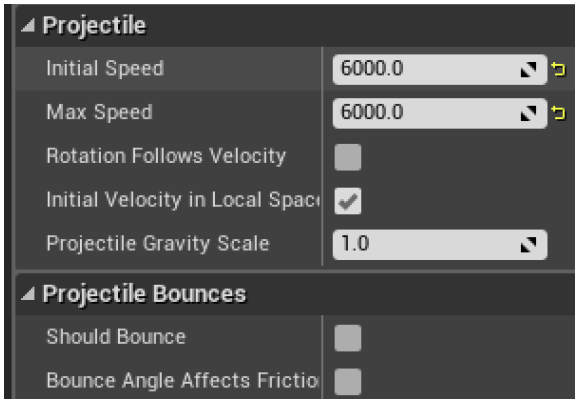
Animating a zoom view



Using a timeline to smooth transitions

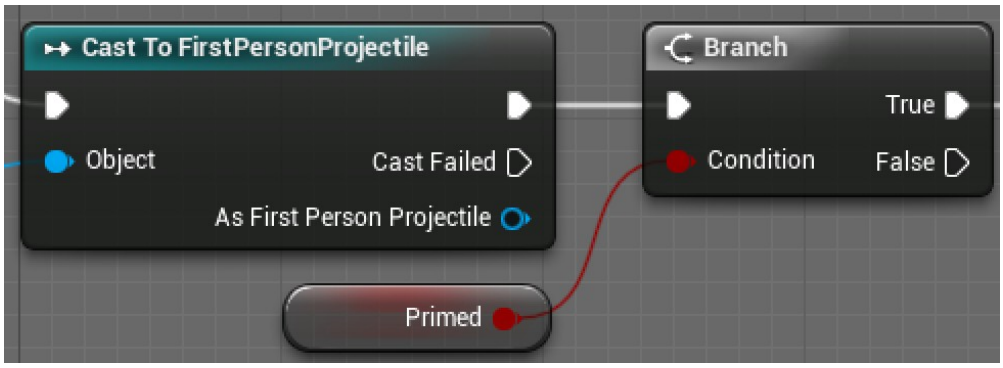


Increasing the projectile's speed

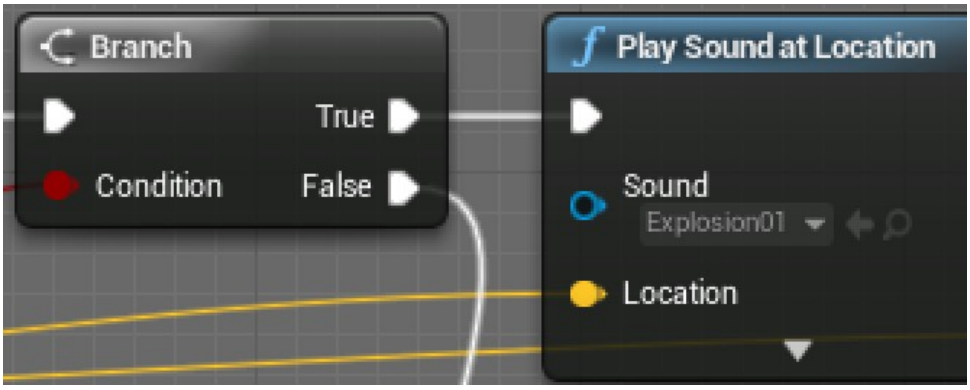


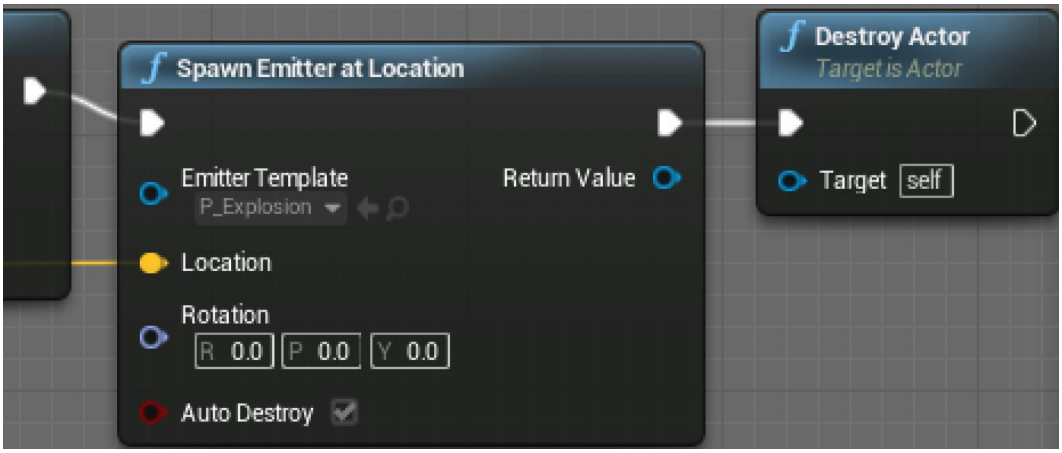
Adding sound and particle effects

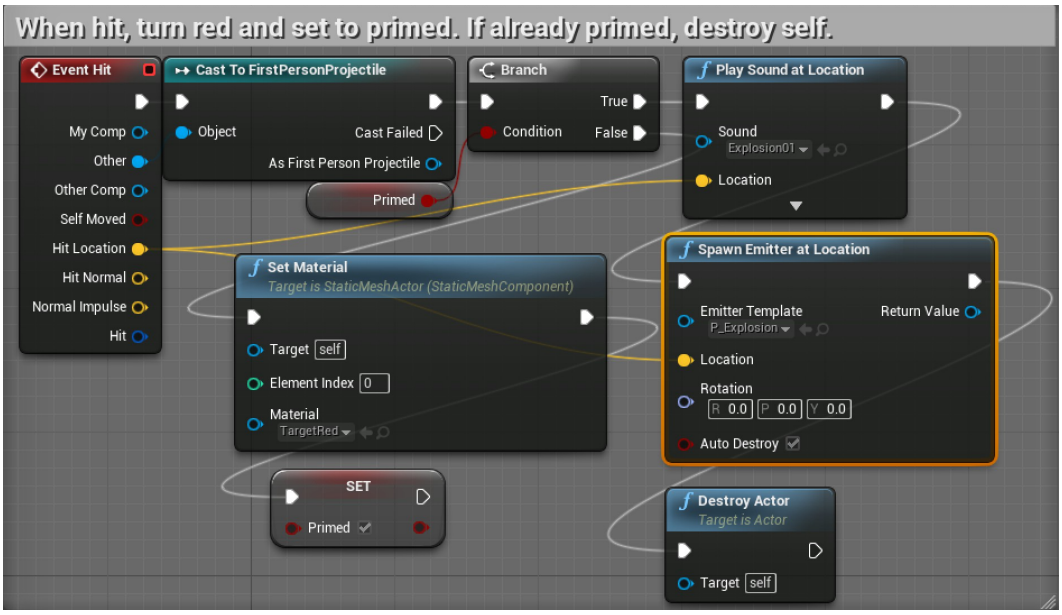
Giving our targets state with branches



Triggering sound effects, explosions, and destruction



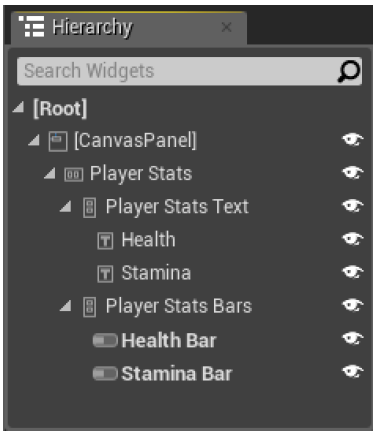


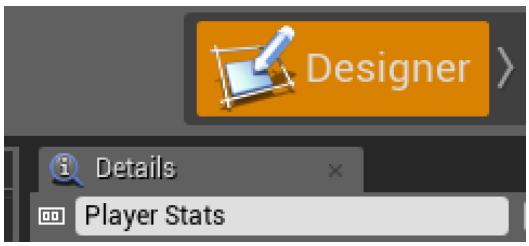

When hit, turn red and set to primed. If already primed, destroy self.
Event Hit
My Comp
Other
Other Comp
Self Moved
Hit Location
Hit Normal
Normal Impulse
Hit
Cast To FirstPersonProjectile
Object
Cast Failed
As First Person Projectile
Primed
Branch
True
Condition
False
Play Sound at Location
Sound
Explosion01
Location
Set Material
Target is StaticMeshActor (StaticMeshComponent)
Target
self
Element Index
0
Material
TargetRed
SET
Primed
Spawn Emitter at Location
Emitter Template
P_Explosion
Return Value
Location
Rotation
R 0.0
P 0.0
Y 0.0
Auto Destroy
Destroy Actor
Target is Actor
Target
self

Chapter 3: Creating Screen UI Elements

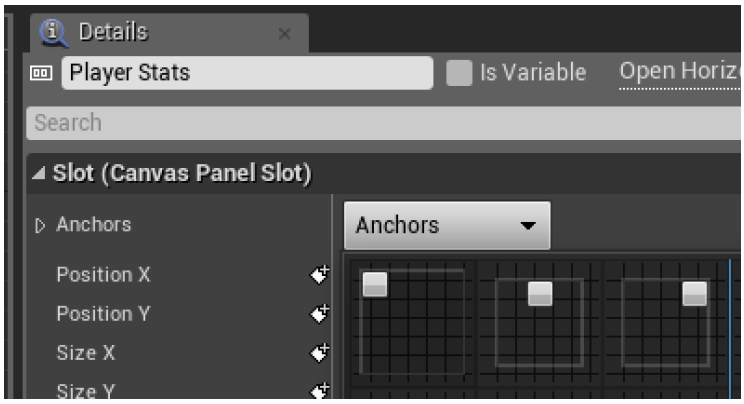
Creating simple UI meters with UMG

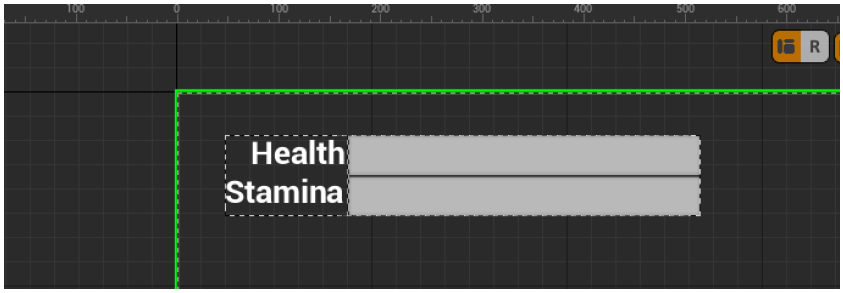
Drawing shapes with widget Blueprints



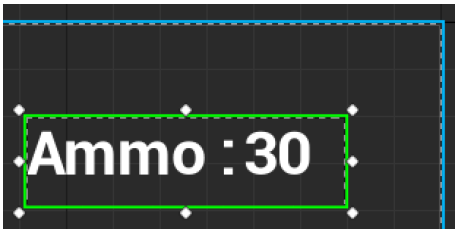


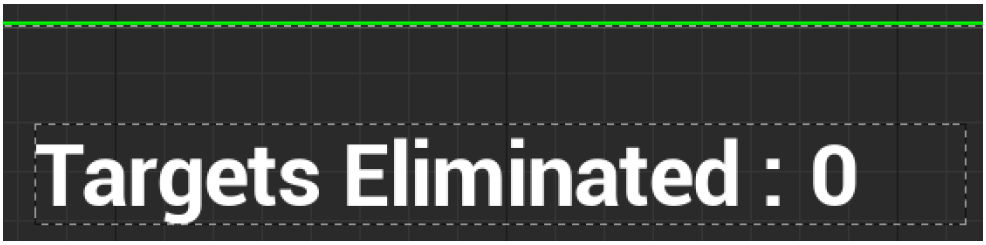
Customizing the meter's appearance



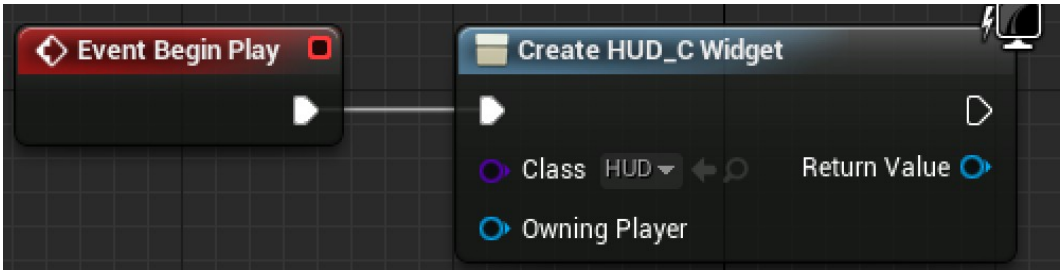


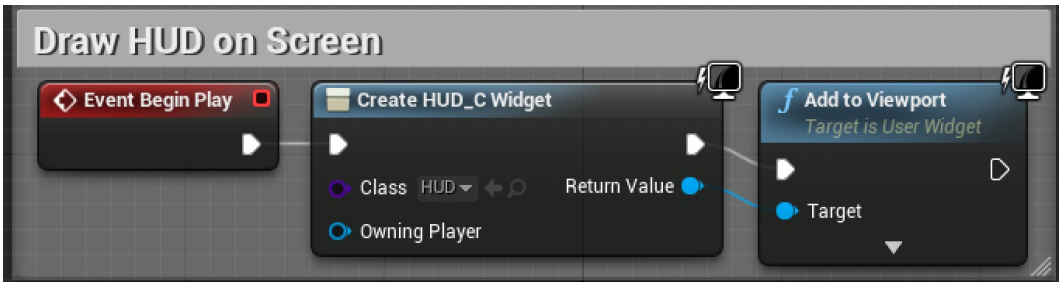
Creating ammo and enemy counters





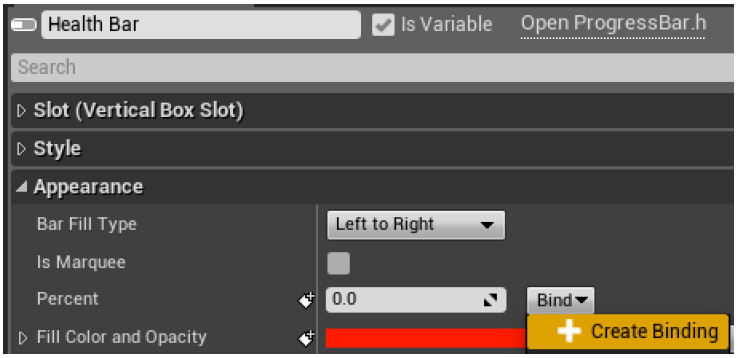
Displaying the HUD

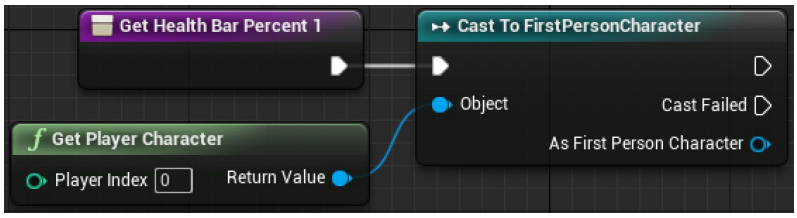


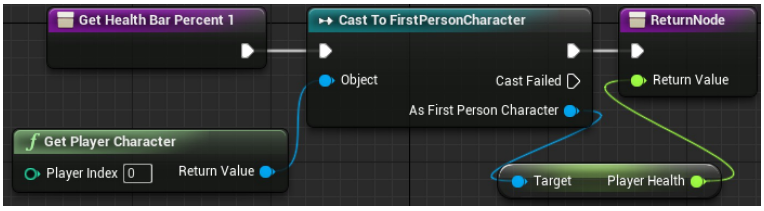


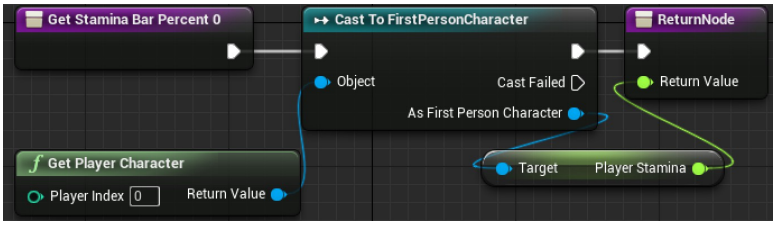
Connecting UI values to player variables

Creating bindings for health and stamina

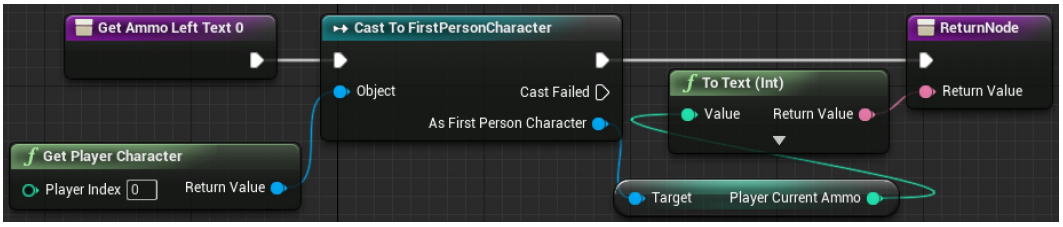


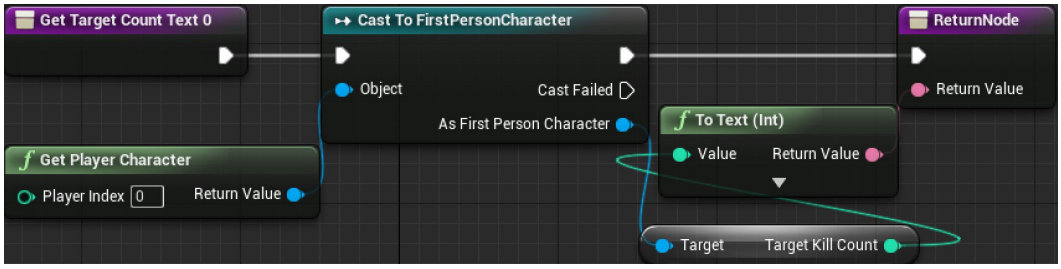






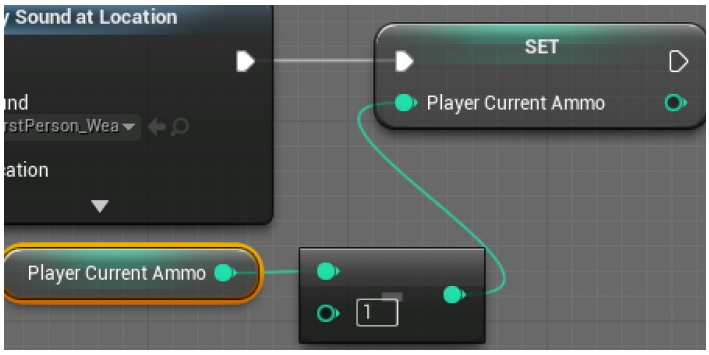
Making text bindings



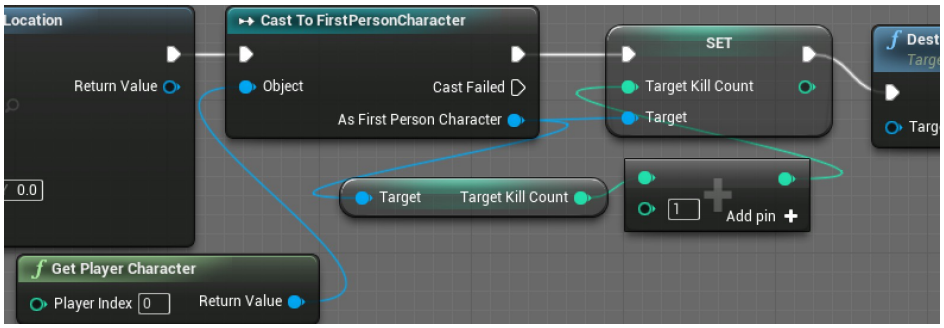


Tracking the ammo and eliminated targets

Reducing the ammo counter



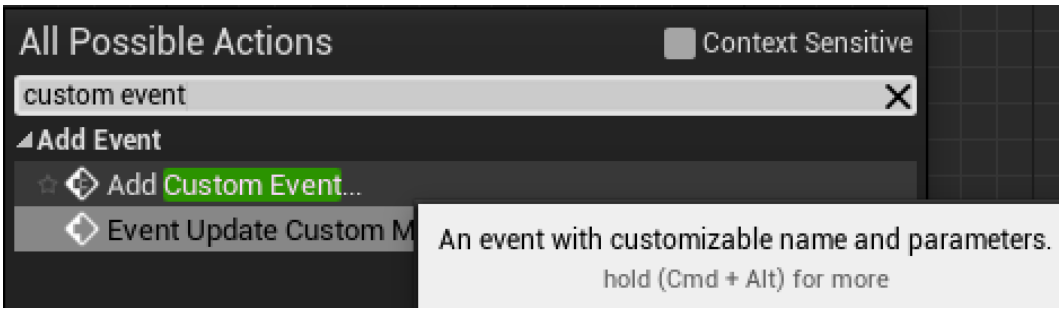
Increasing the targets eliminated counter

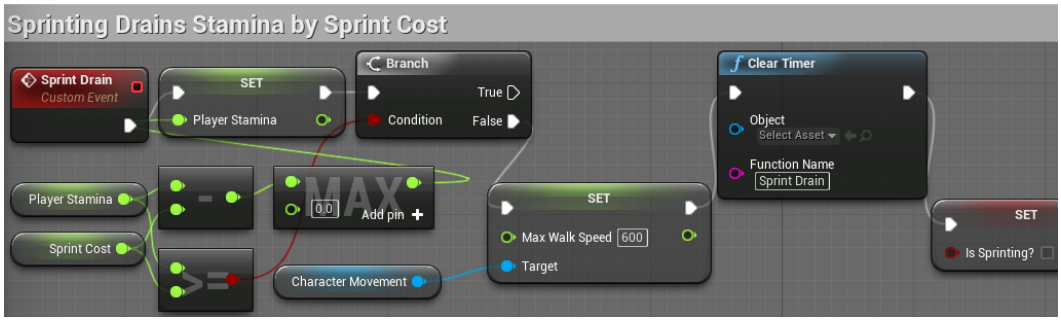


Chapter 4: Creating Constraints and Gameplay Objectives

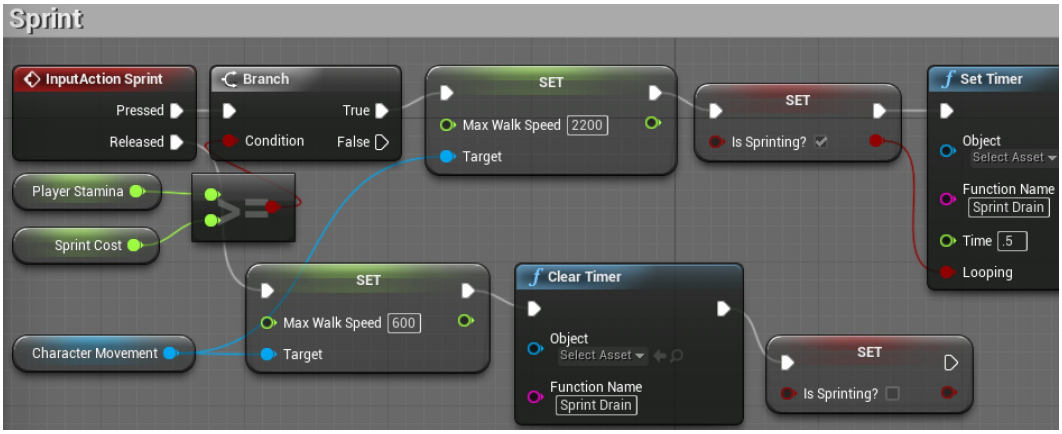
Constraining player actions

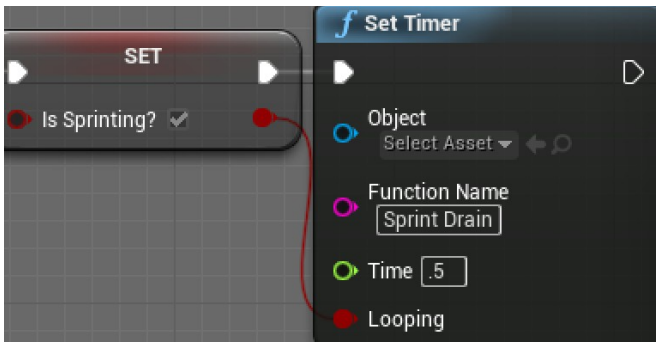
Draining stamina while sprinting

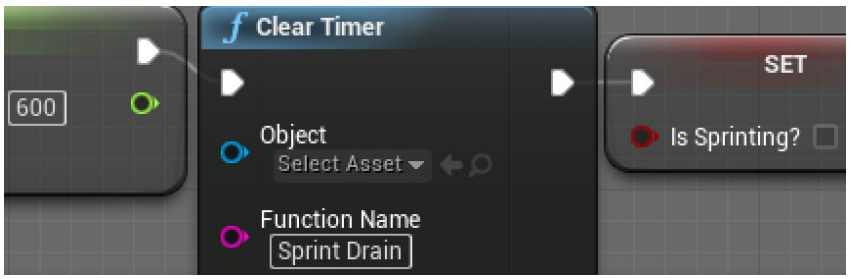




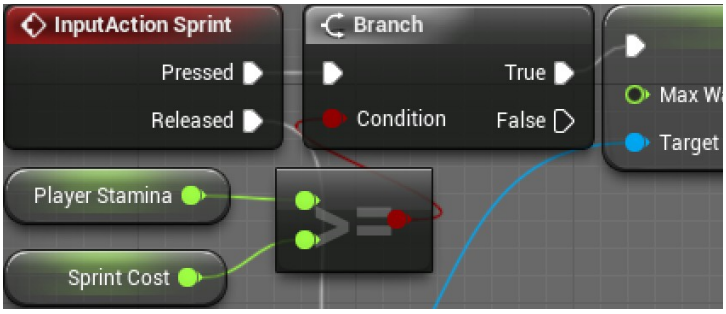
Using looping timers to repeat actions



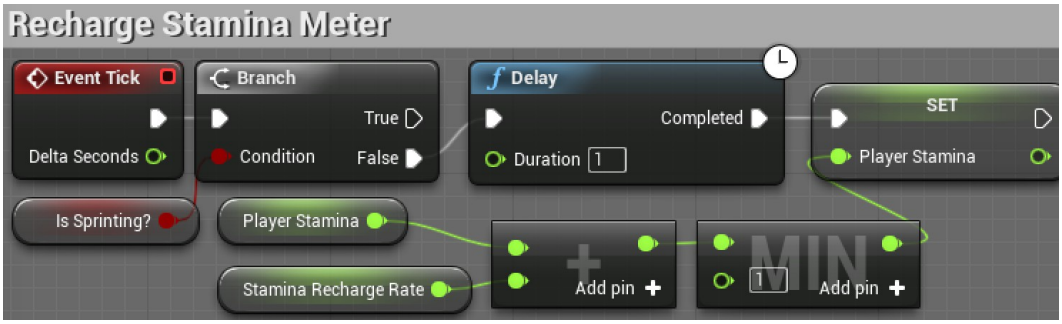




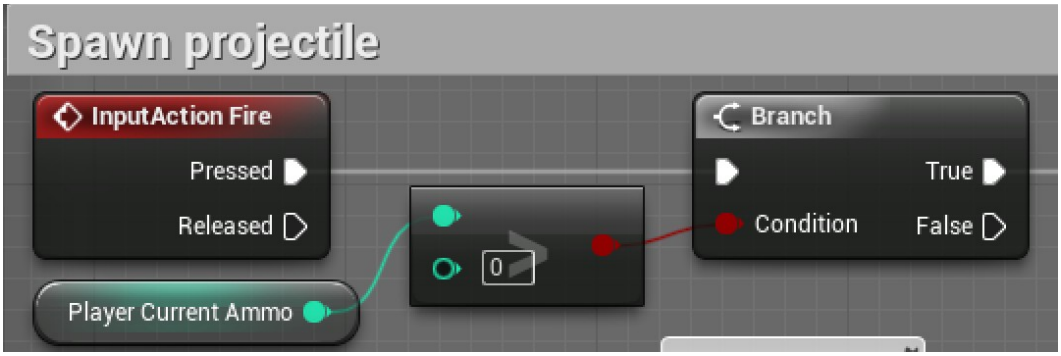
Blocking actions with branch



Regenerating stamina

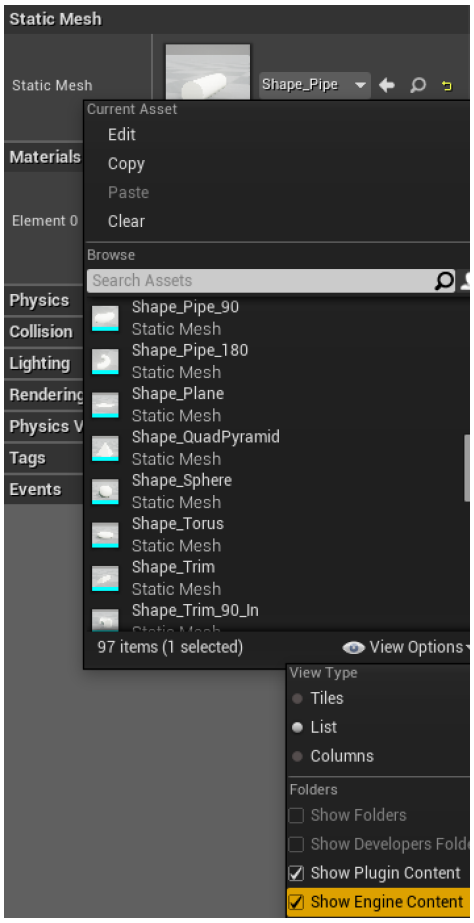


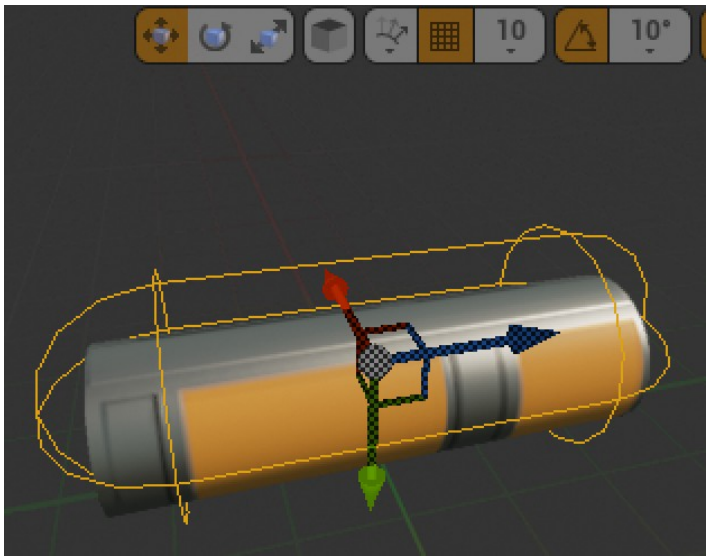
Preventing firing actions when out of ammo

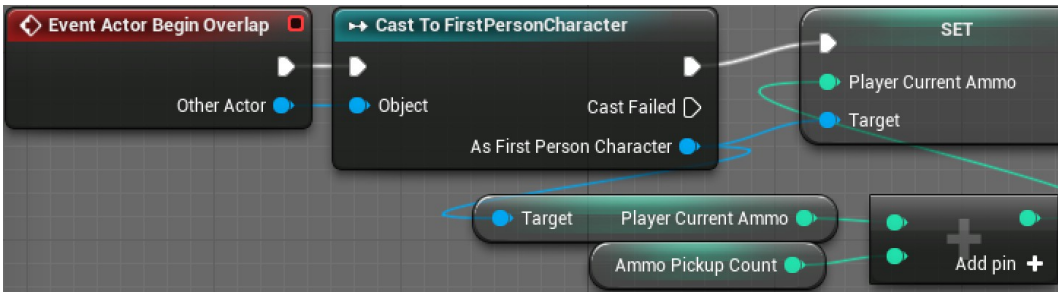


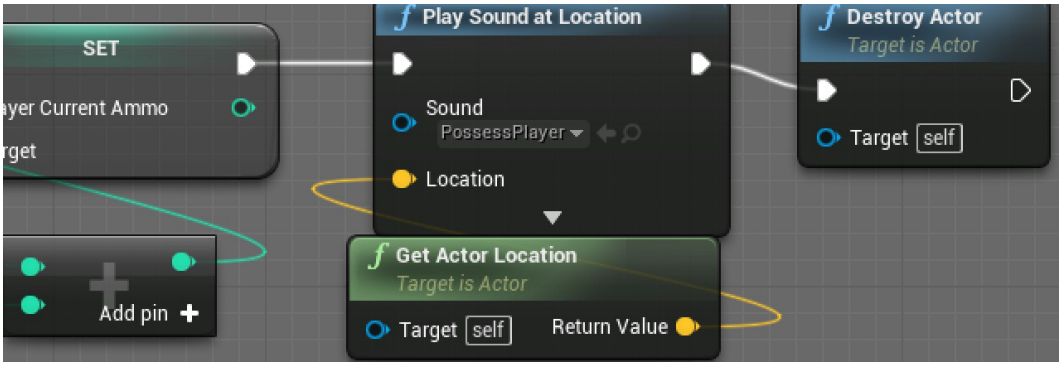
Creating collectable objects

Setting up collection logic




10
10°

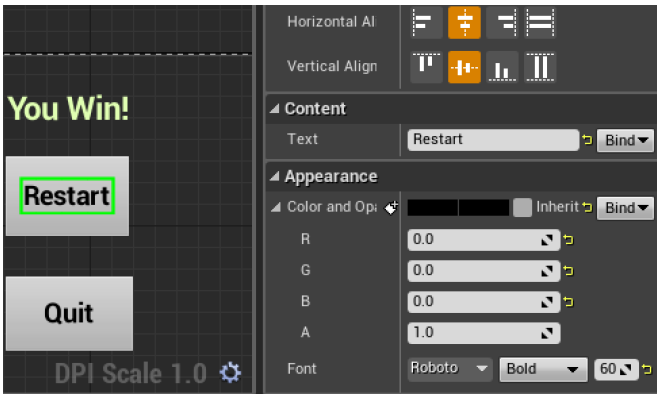

Event Actor Begin Overlap
Other Actor
Cast To FirstPersonCharacter
Object
Cast Failed
As First Person Character
SET
Player Current Ammo
Target
Target
Player Current Ammo
Ammo Pickup Count
Add pin


SET
Play Sound at Location
Sound
PossessPlayer
Location
Destroy Actor
Target is Actor
Target
self
Add pin
Get Actor Location
Target is Actor
Target
self
Return Value

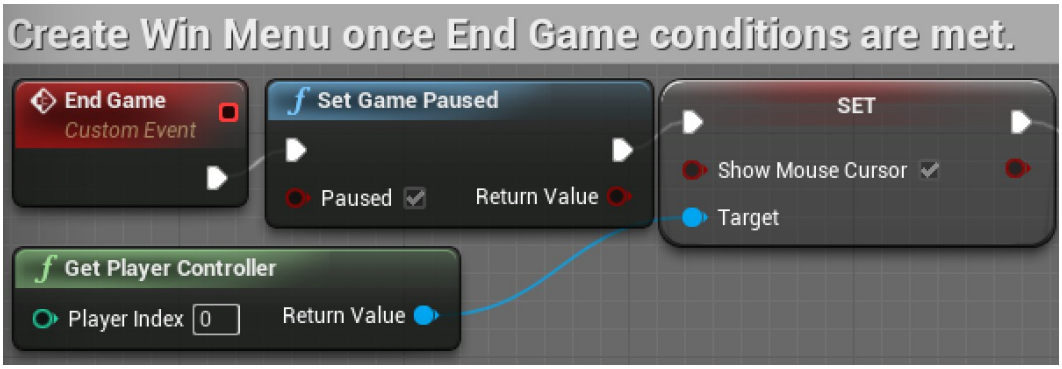
Setting a gameplay win condition

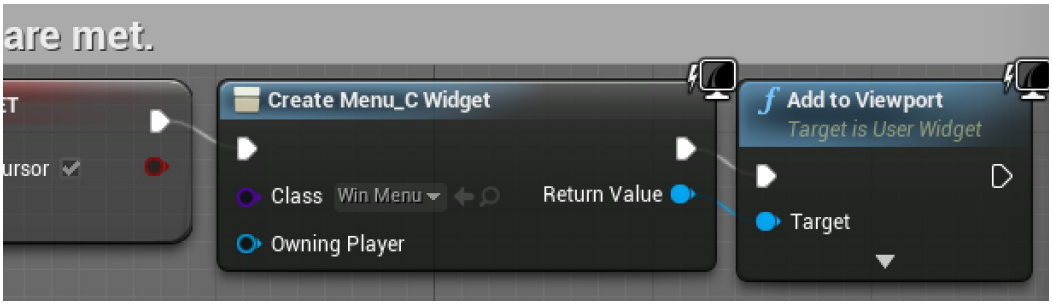
Displaying a target goal in the HUD

Creating a WinMenu

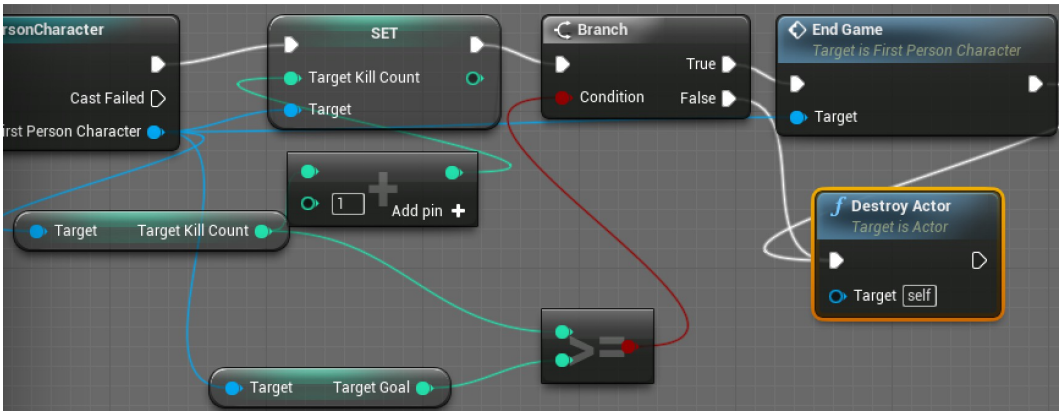


Displaying the menu





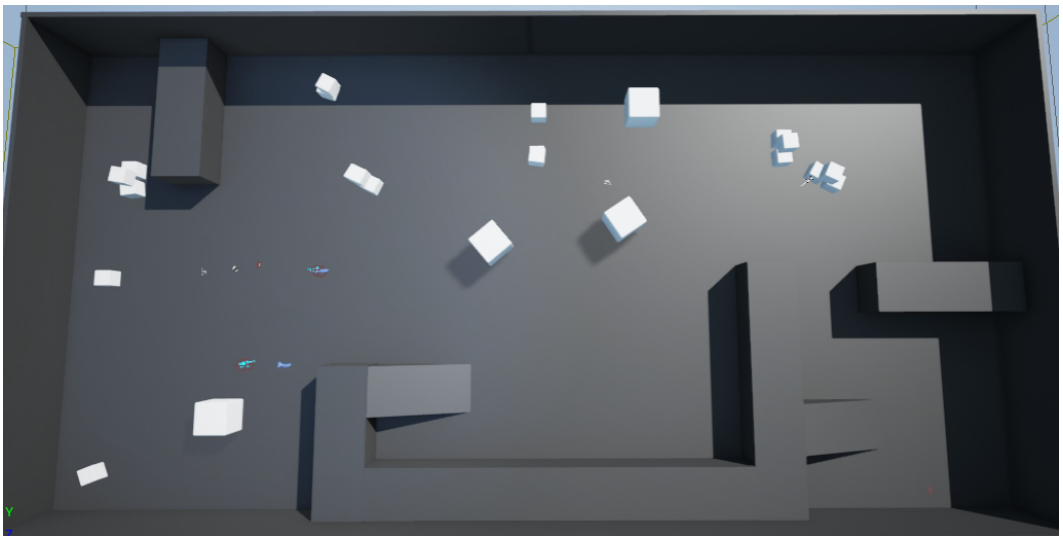
Triggering a win



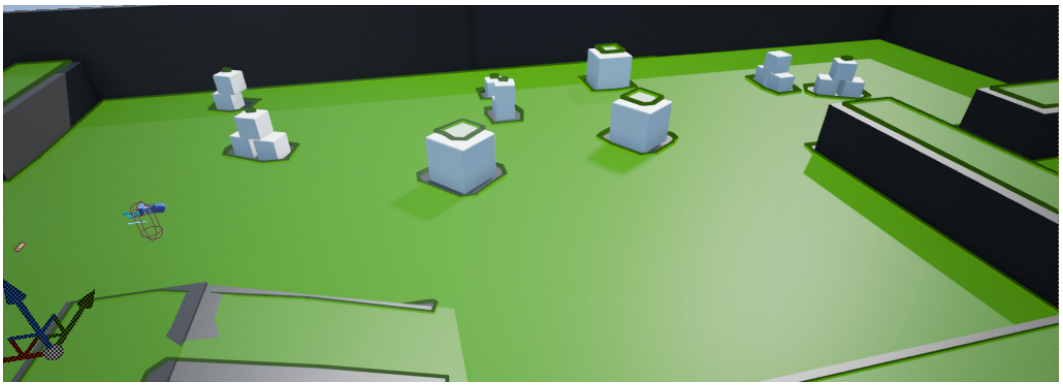
Chapter 5: Making Moving Enemies with AI

Setting up the enemy actor to navigate

Expanding the play area



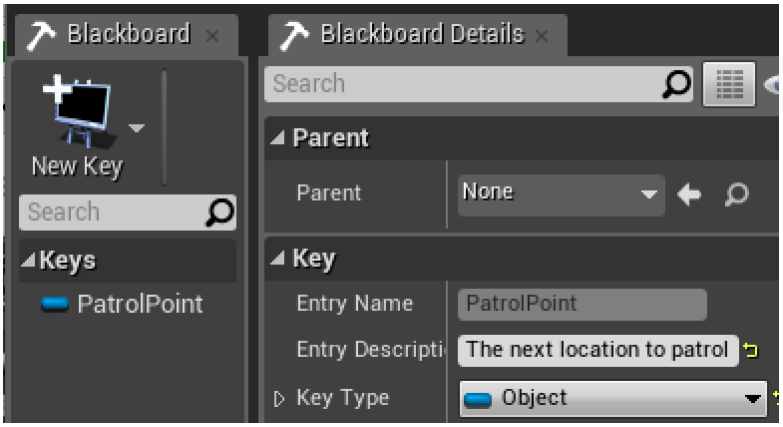
Making the level traversable with a Nav Mesh

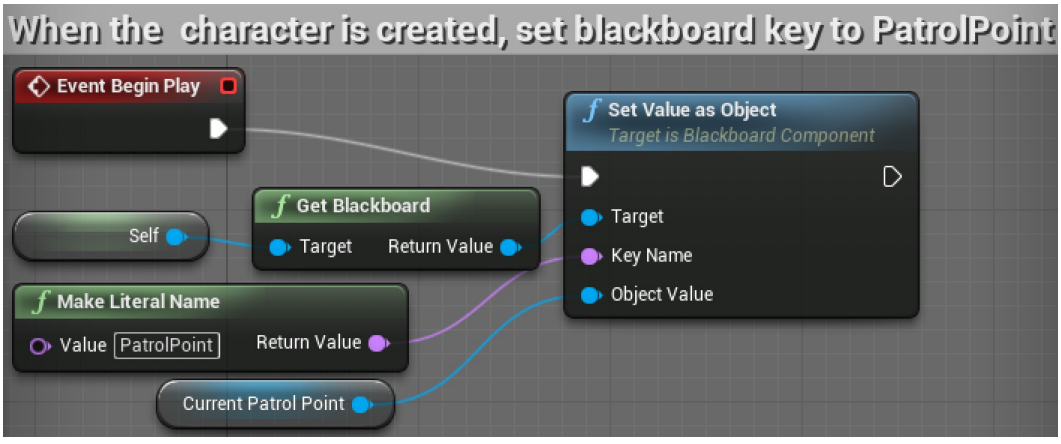


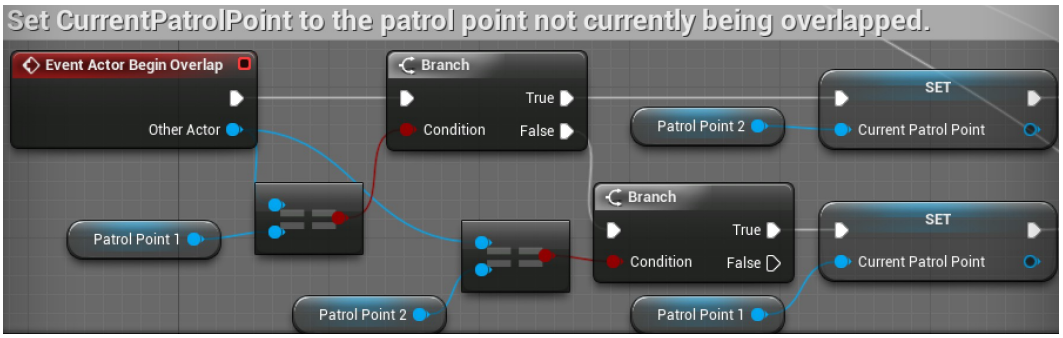
Creating navigation behavior

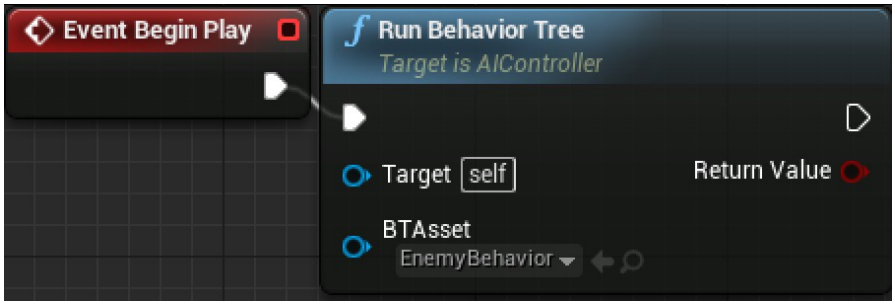
Setting up patrol points

Enabling communication between assets

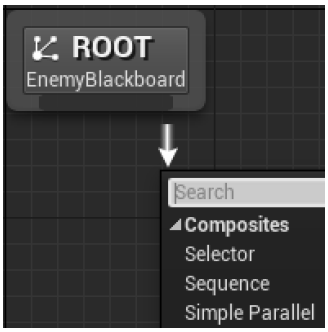


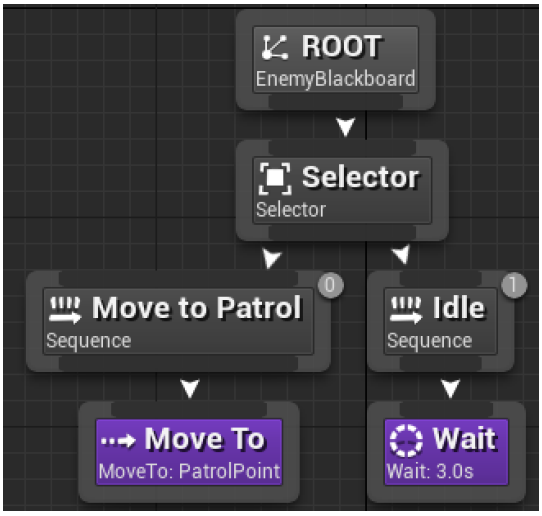


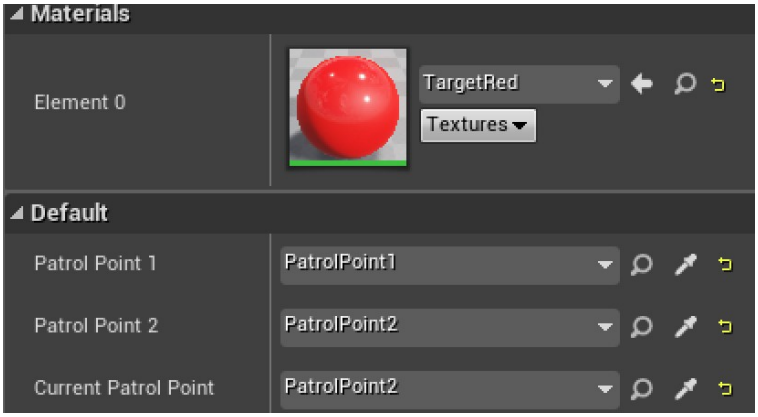

Set CurrentPatrolPoint to the patrol point not currently being overlapped.
Event Actor Begin Overlap
Other Actor
Branch
True
Condition
False
Patrol Point 2
SET
Current Patrol Point
Patrol Point 1
Branch
True
Condition
False
SET
Current Patrol Point
Patrol Point 2
Patrol Point 1


Event Begin Play
Run Behavior Tree
Target is AIController
Target
self
Return Value
BTAsset
EnemyBehavior

Teaching our AI to walk with the Behavior Tree

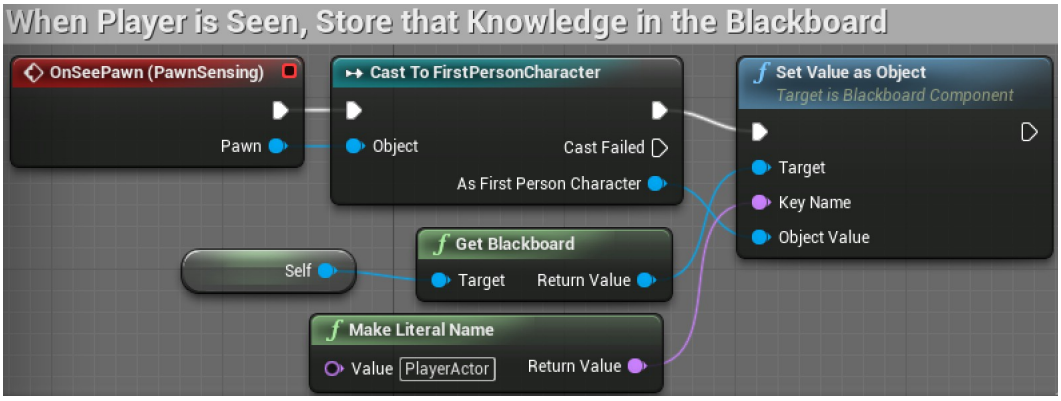




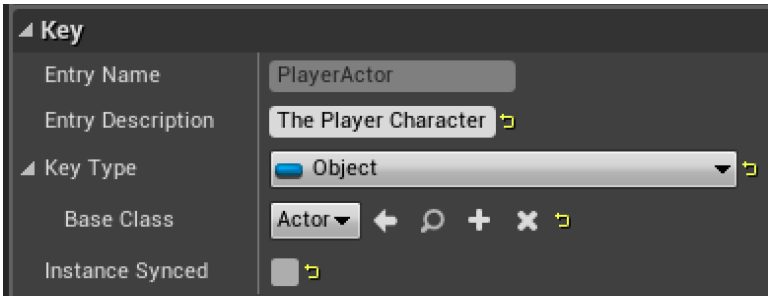


Making the AI chase the player

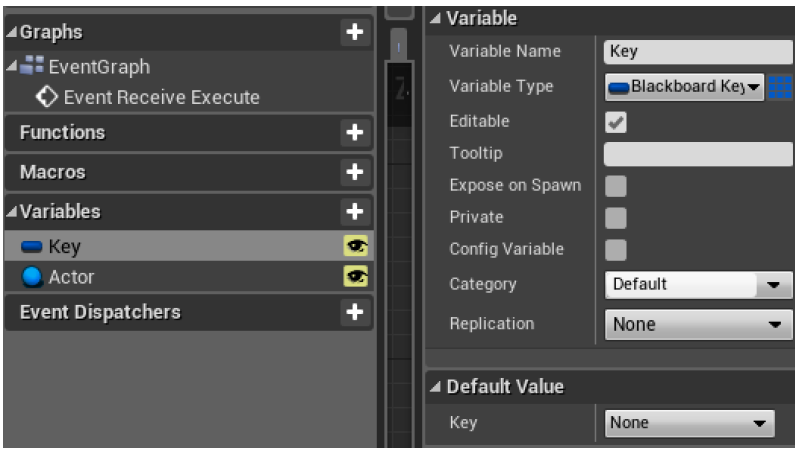
Giving the enemy sight with Pawn Sensing

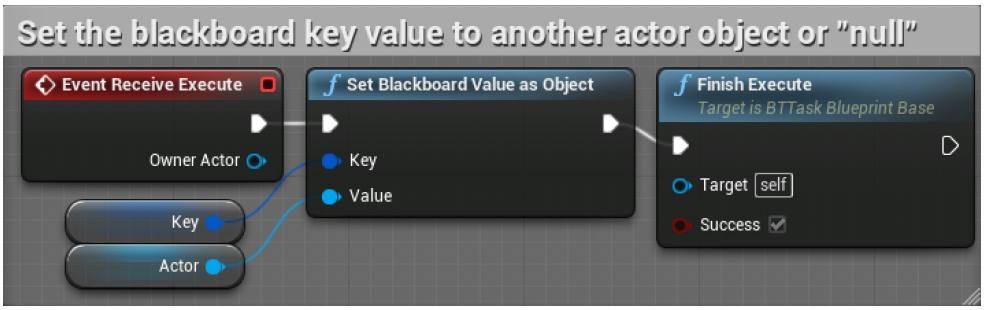


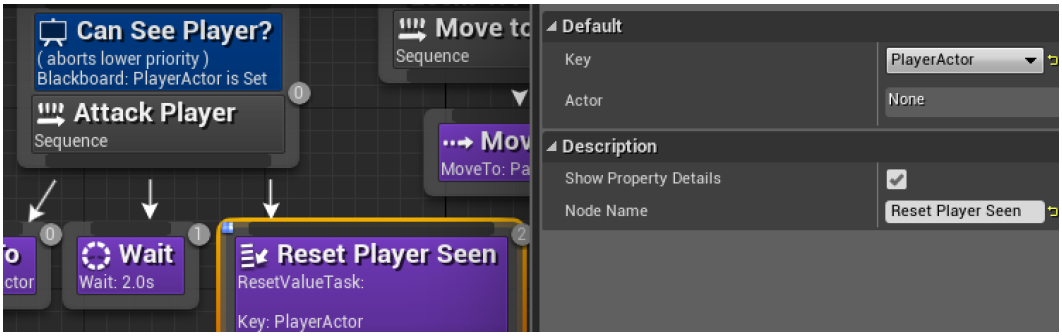
Adding conditions to the Behavior Tree



Creating chasing behavior



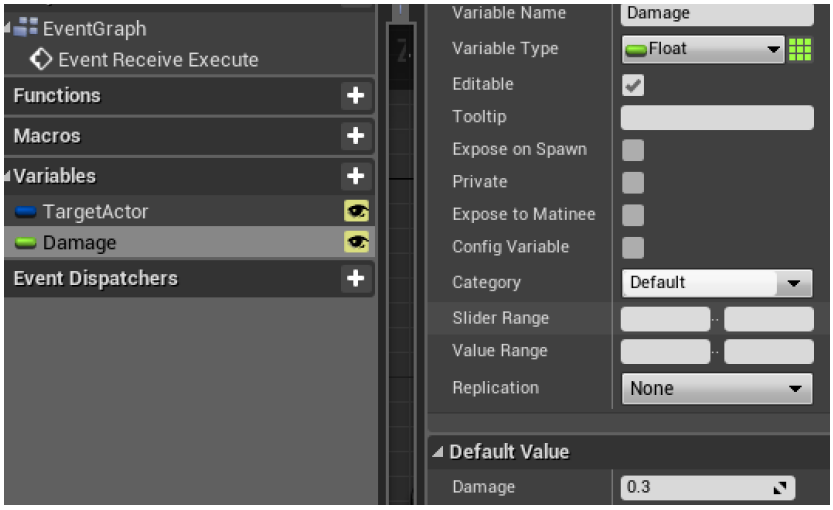


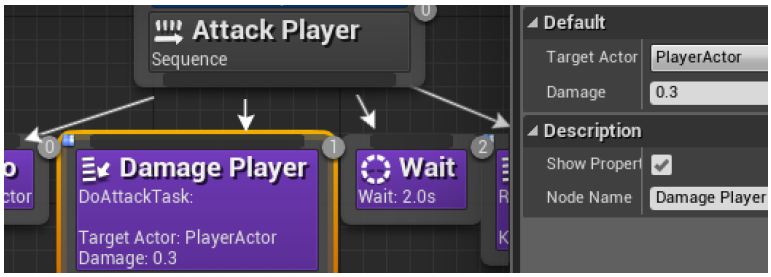


Chapter 6: Upgrading the AI Enemies

Creating an enemy attack

Making an attack task



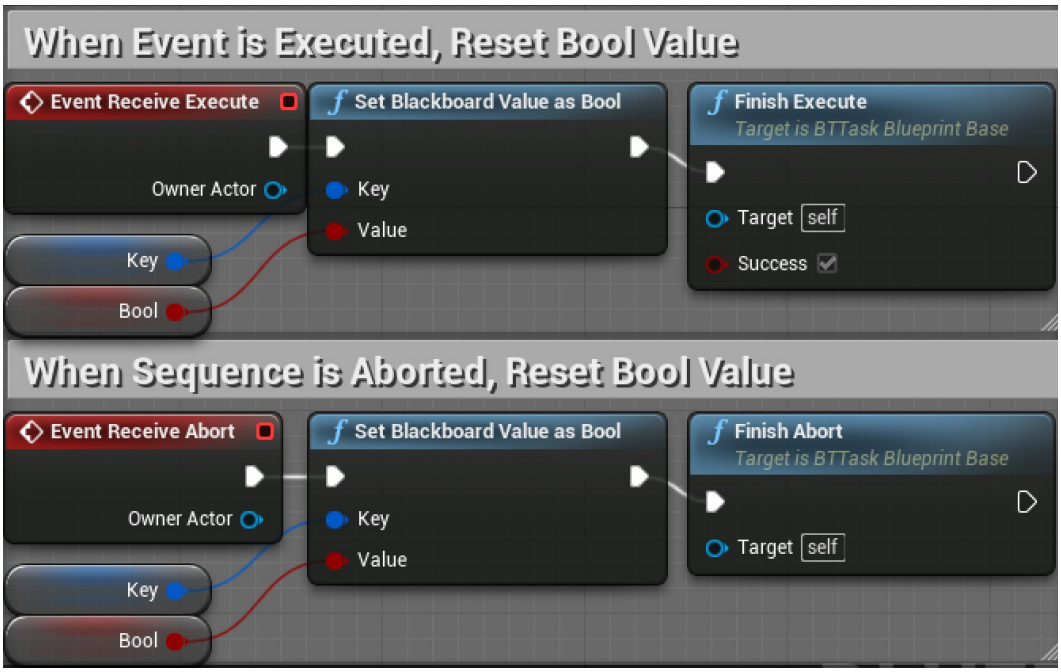


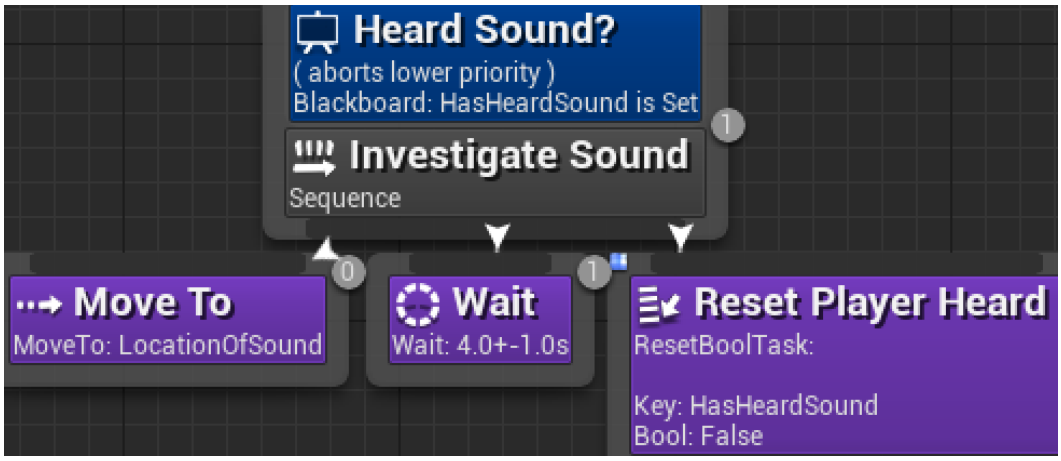
Updating the health meter

Making enemies listen to and investigate sounds

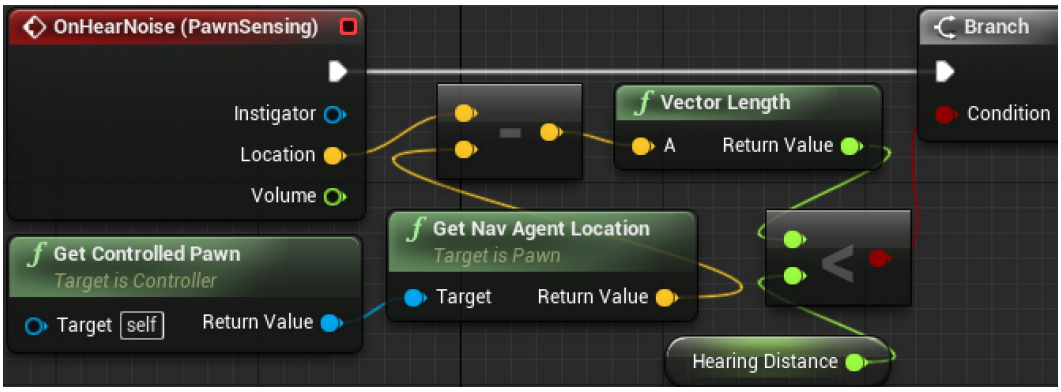
Adding hearing to the Behavior Tree

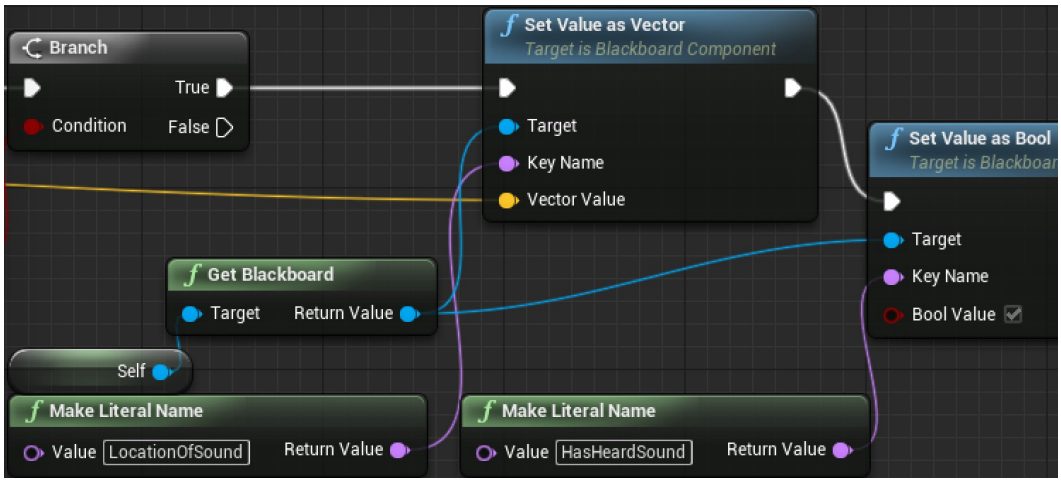
Seeding the investigating tasks



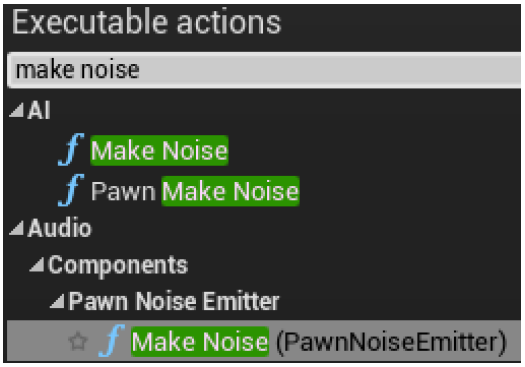


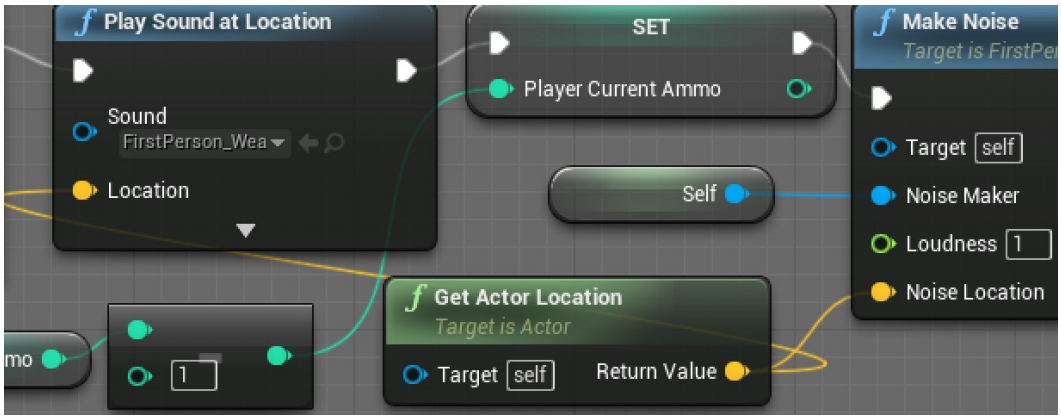
Interpreting and storing the noise event data





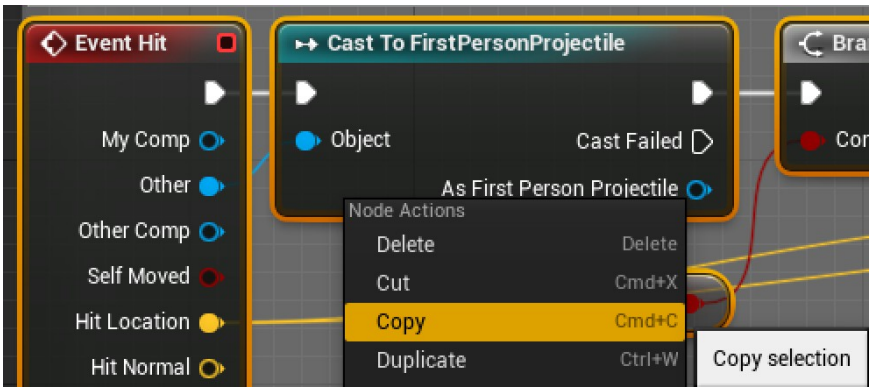
Adding noise to the player's actions

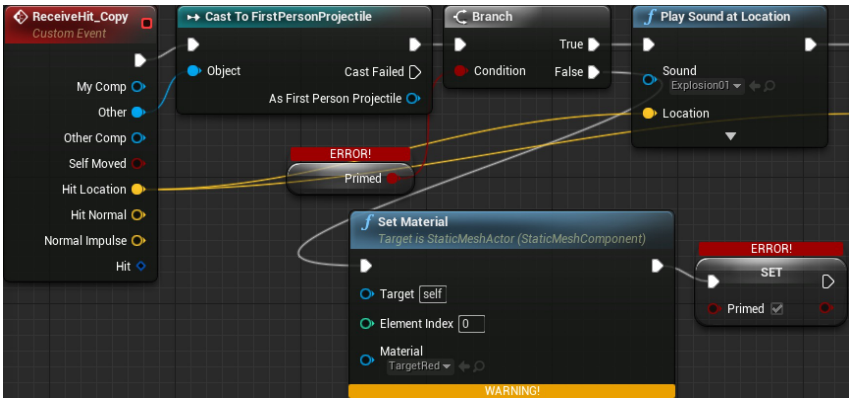


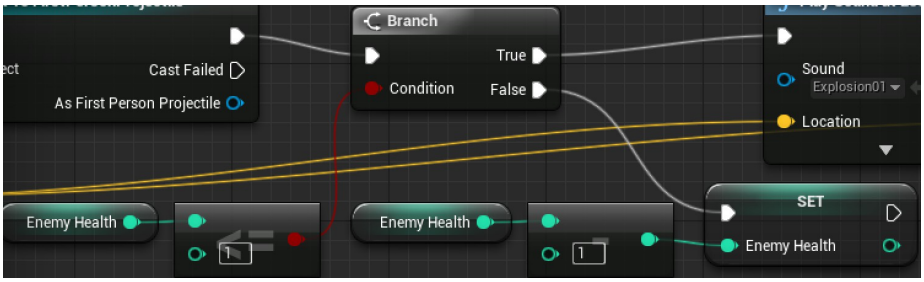


Making the enemies destructible

Saving time by reusing existing Blueprint content

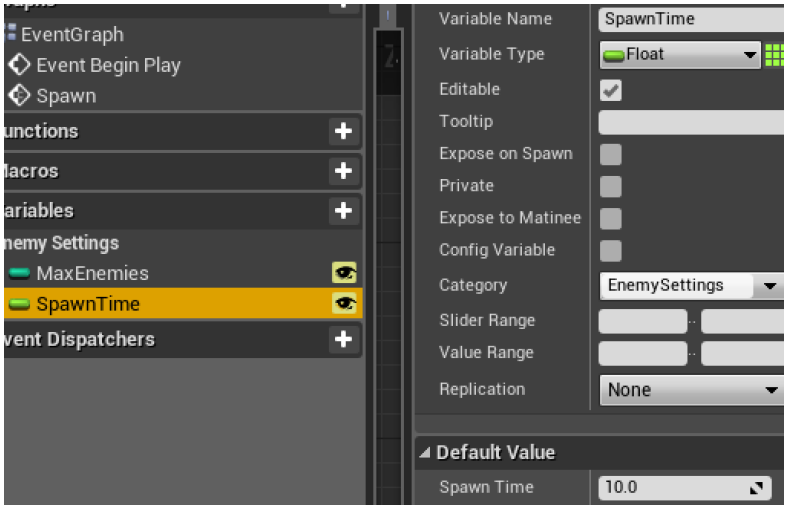


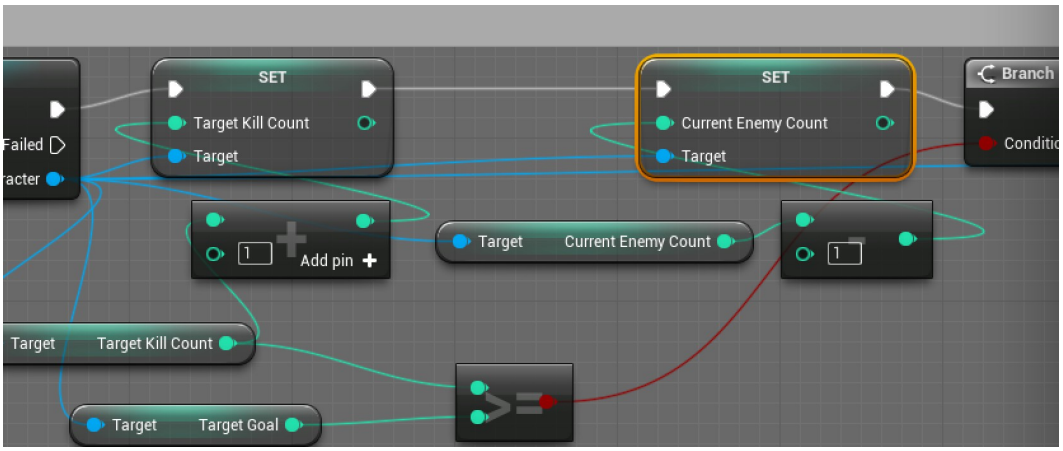

ReceiveHit_Copy
Custom Event
My Comp
Other
Other Comp
Self Moved
Hit Location
Hit Normal
Normal Impulse
Hit
Cast To FirstPersonProjectile
Object
Cast Failed
As First Person Projectile
Branch
True
Condition
False
Play Sound at Location
Sound
Explosion01
Location
ERROR!
Primed
Set Material
Target is StaticMeshActor (StaticMeshComponent)
Target
self
Element Index
0
Material
TargetRed
WARNING!
ERROR!
SET
Primed


Cast Failed
As First Person Projectile
Branch
True
Condition
False
Sound
Explosion01
Location
Enemy Health
1
Enemy Health
1
SET
Enemy Health

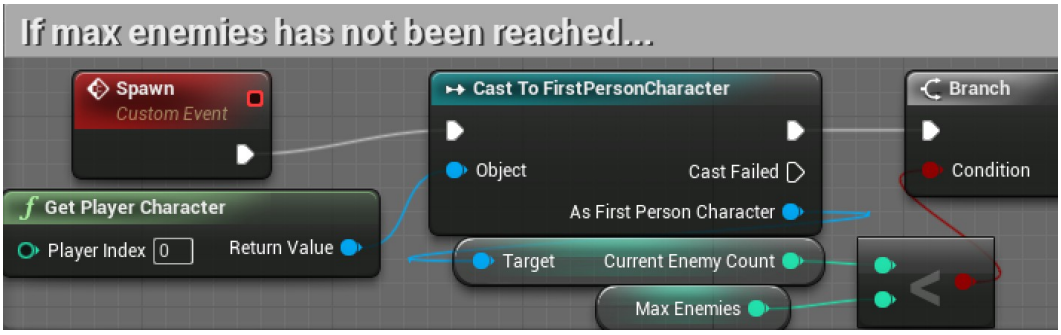
Spawning more enemies while the game is playing

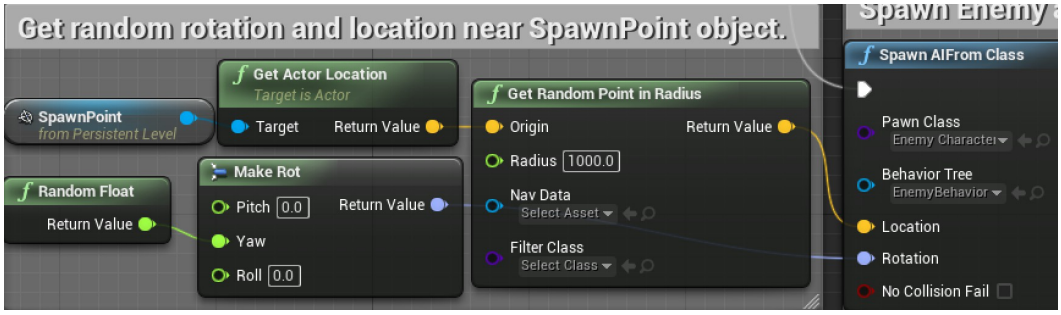
Managing spawn rates and limits with variables

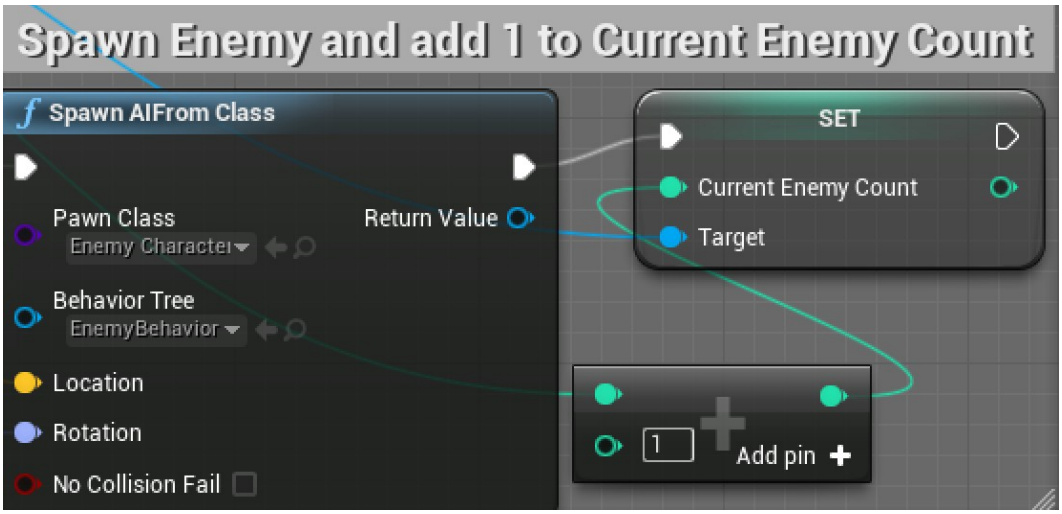




Spawning new enemies in the Level Blueprint



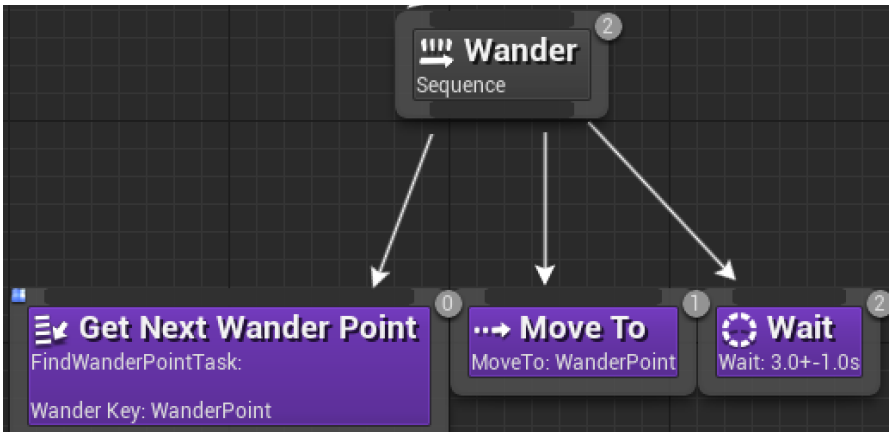



Spawn Enemy and add 1 to Current Enemy Count
Spawn AIFrom Class
Pawn Class
Enemy Character
Return Value
Behavior Tree
EnemyBehavior
Location
Rotation
No Collision Fail
SET
Current Enemy Count
Target
1
Add pin

Creating wandering behavior for the enemies

Identifying a wander point with a custom task

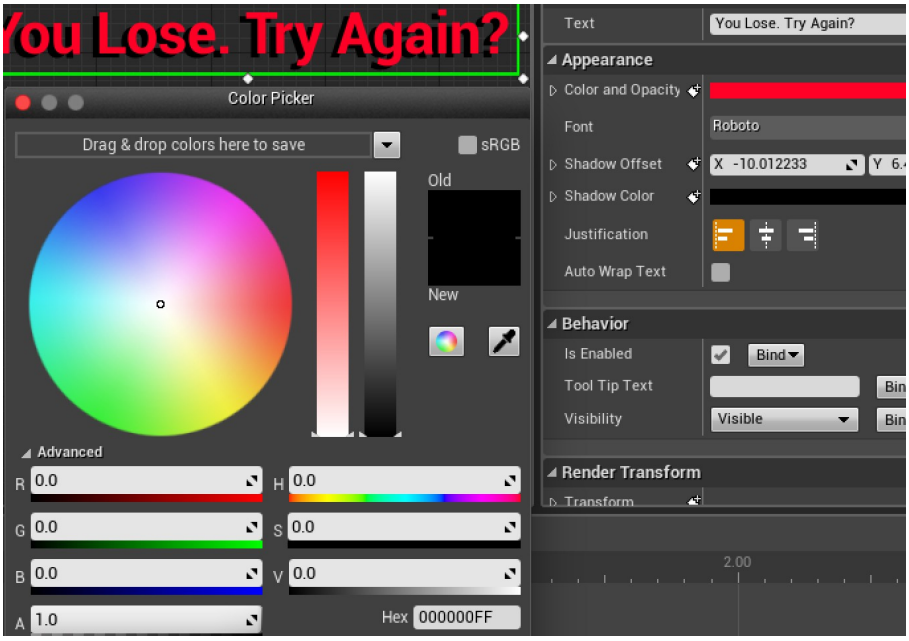
Adding wandering to the Behavior Tree

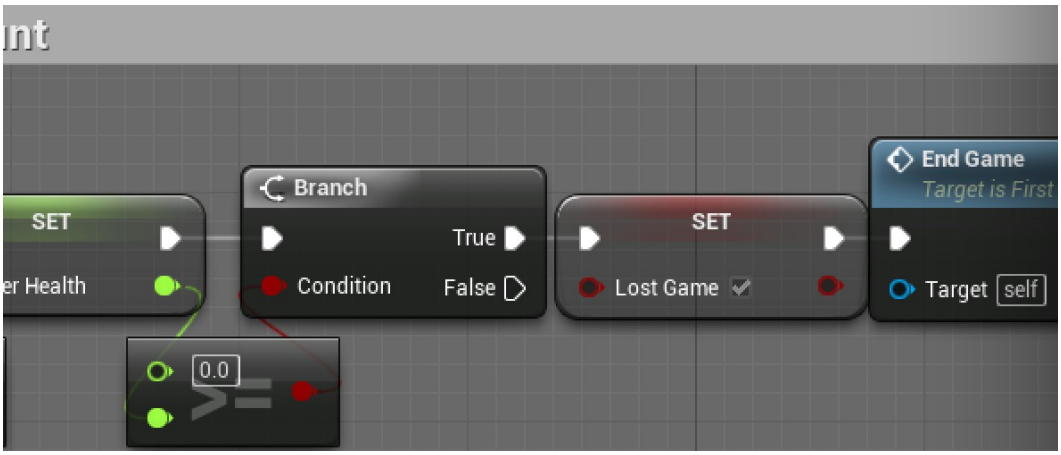



FirstPersonCharacter(self)
CapsuleComponent (Inherited)
Mesh (Inherited)
Player Health
1.0
Player Stamina
1.0
Player Current Ammo
30
Target Kill Count
0
Sprint Cost
0.1
Stamina Recharge Rate
0.05
Target Goal
20
Current Enemy Count
0

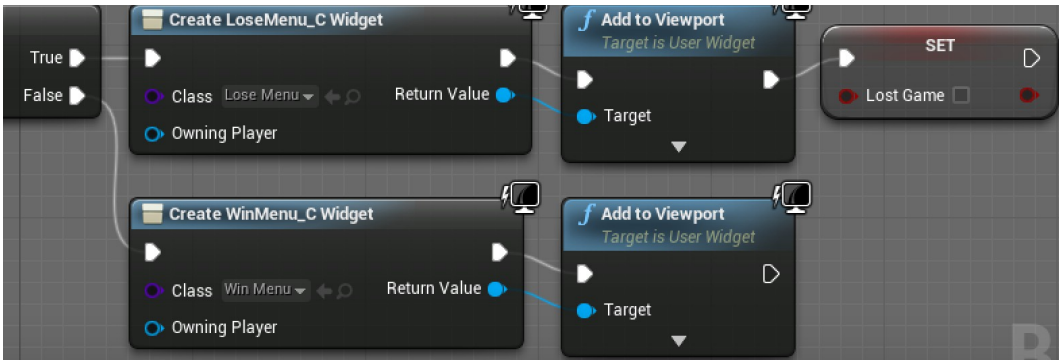
Chapter 7: Tracking Game States and Finishing Touches

Setting up a lose screen



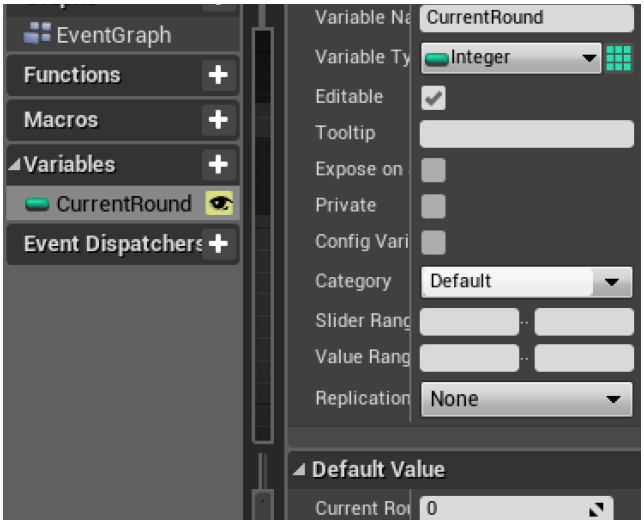

SET
er Health
Branch
True
Condition
False
SET
Lost Game
End Game
Target is First
Target
self
0.0


Create Win or Loss Menu once End Game conditions are met.
End Game
Custom Event
Set Game Paused
Paused
Return Value
SET
Show Mouse Cursor
Target
Branch
Condition
Get Player Controller
Player Index
0
Return Value
Lost Game


True
False
Create LoseMenu_C Widget
Class
Lose Menu
Return Value
Owning Player
Add to Viewport
Target is User Widget
Target
SET
Lost Game
Create WinMenu_C Widget
Class
Win Menu
Return Value
Owning Player
Add to Viewport
Target is User Widget
Target

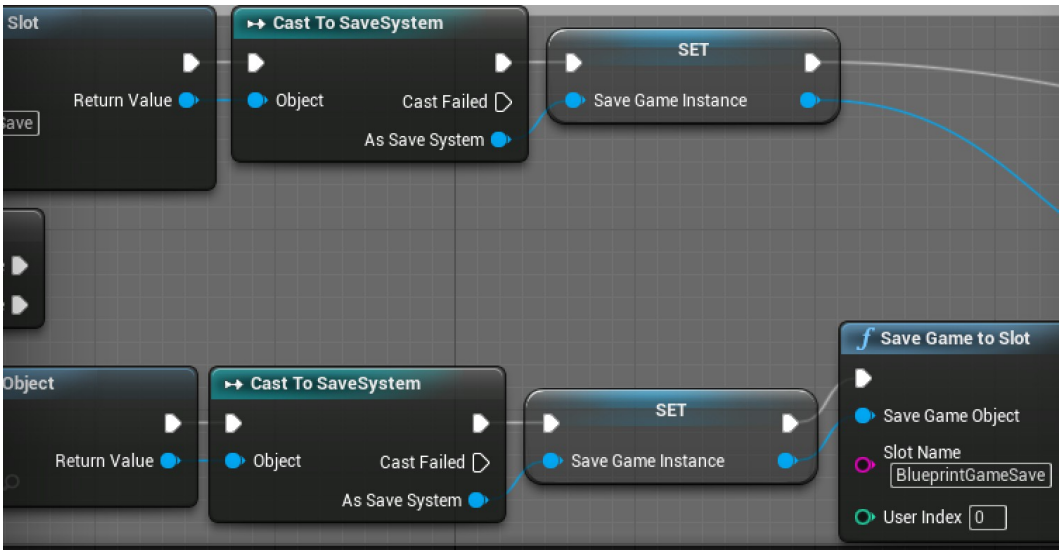
Creating round-based scaling with Save Games

Storing game information using a SaveGame object

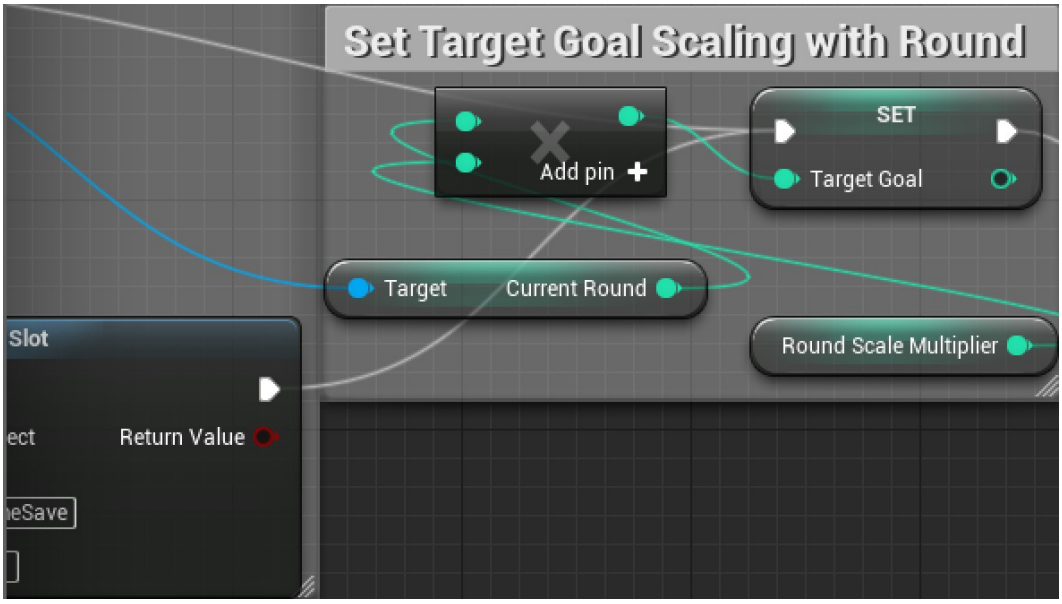


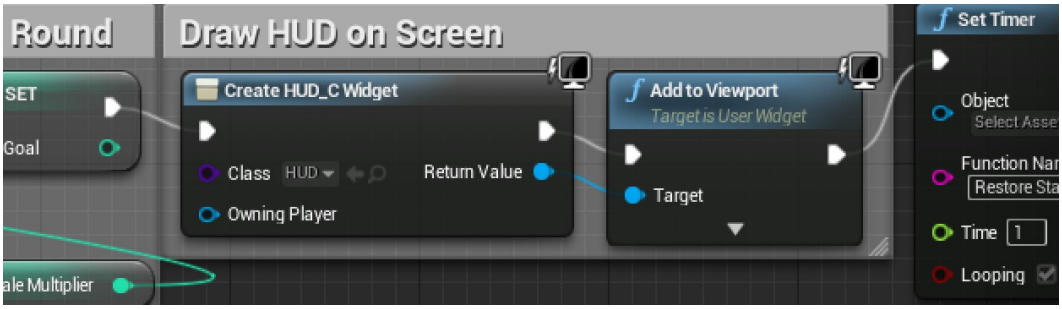
Storing and loading the saved data when starting the game


Create a save, or if one exists, load it
Print String
In String
Save Found
Load Game from Slot
Slot Name
BlueprintGameSave
User Index
0
Event Begin Play
Does Save Game Exist
Slot Name
BlueprintGameSave
Return Value
User Index
0
Branch
True
Condition
False
Print String
In String
Creating Save
Create Save Game Object
Save Game Class
Save System

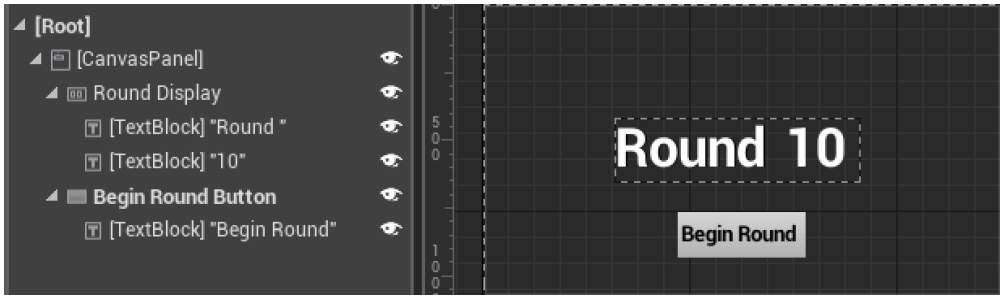

Slot
Return Value
Cast To SaveSystem
Object
Cast Failed
As Save System
SET
Save Game Instance
Object
Return Value
Cast To SaveSystem
Object
Cast Failed
As Save System
SET
Save Game Instance
Save Game to Slot
Save Game Object
Slot Name
BlueprintGameSave
User Index
0

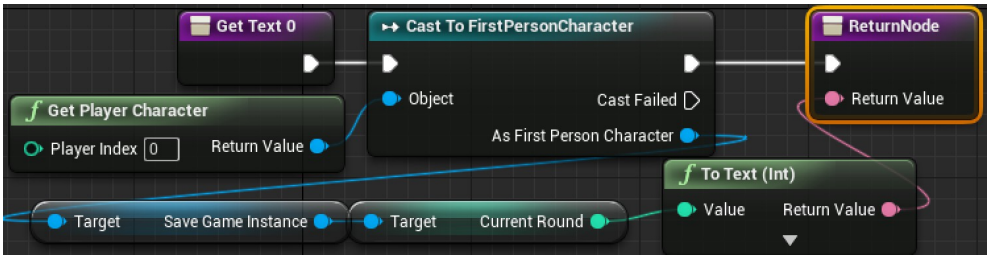
Increasing the enemy target goal



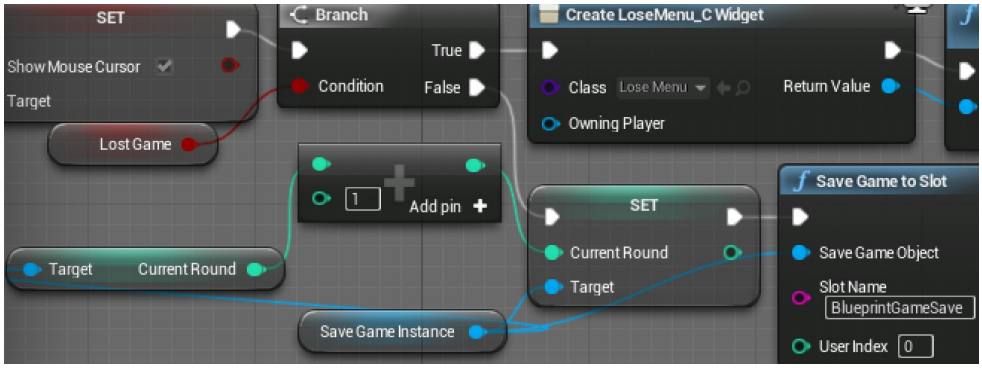


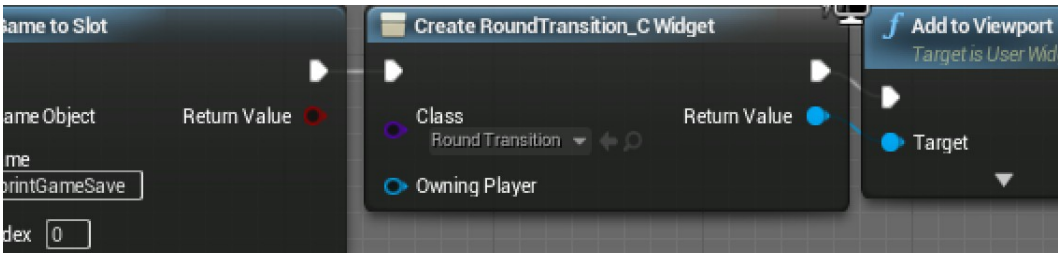
Create a transition screen to show between rounds





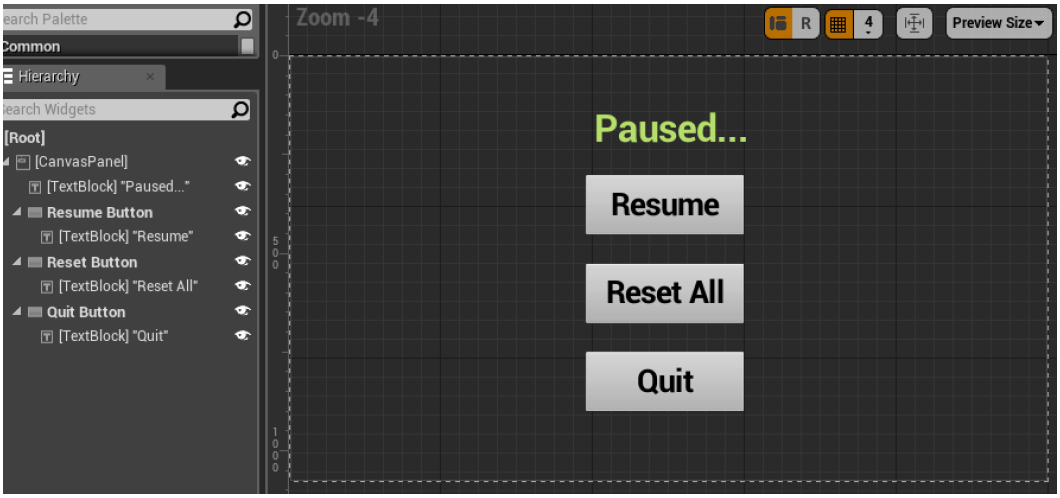
Transitioning to new round when current round is won



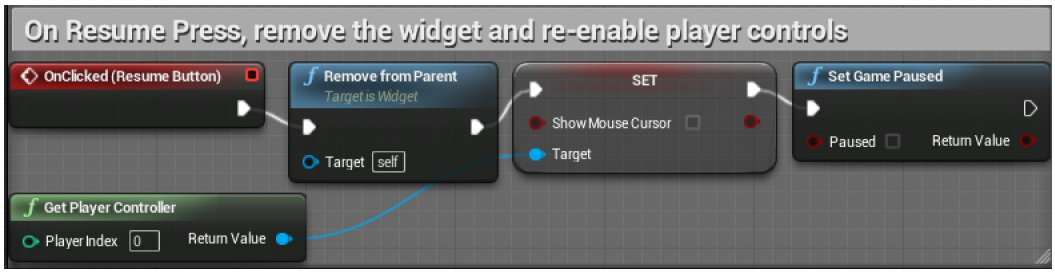


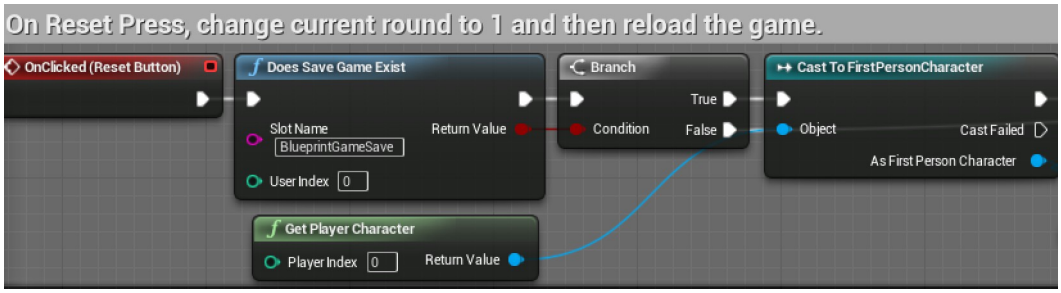
Pausing the game and resetting the save file

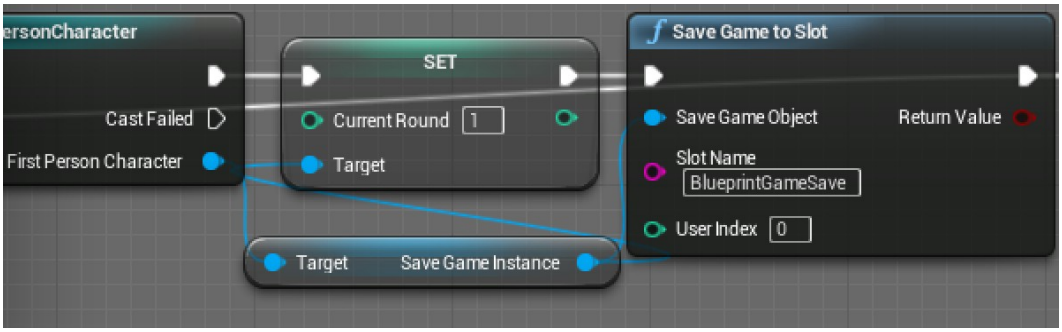
Creating a pause menu

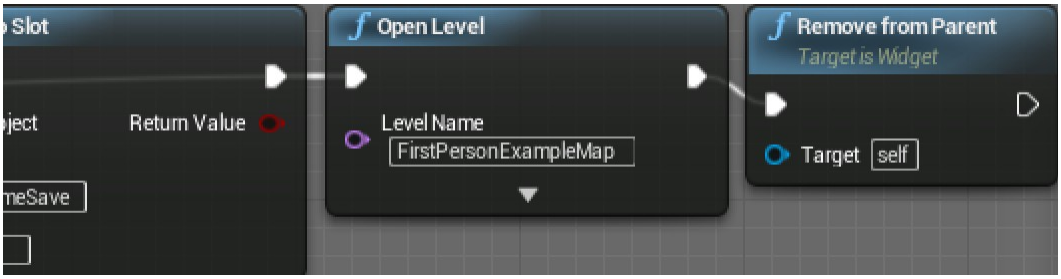


Resuming and resetting the save

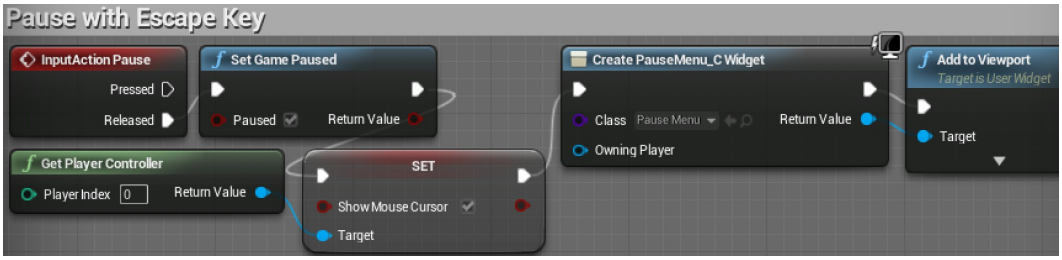


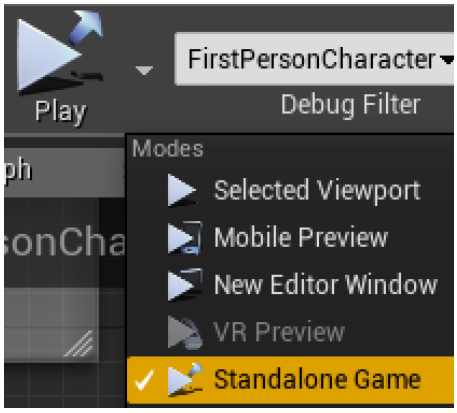






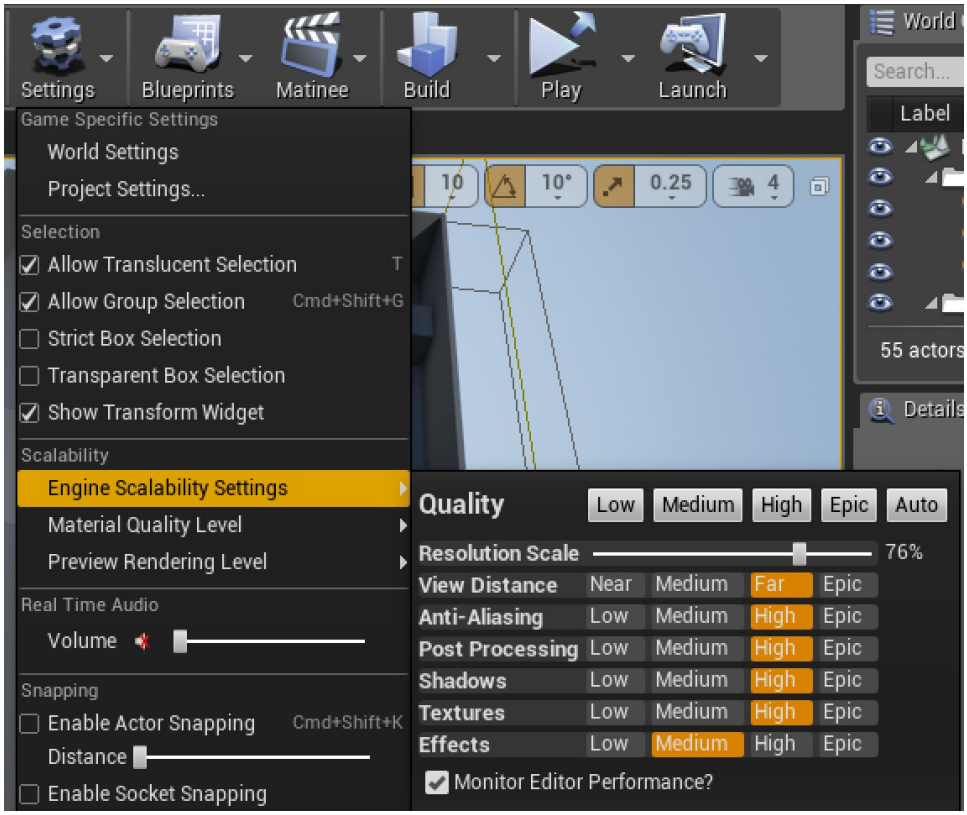
Triggering the pause menu



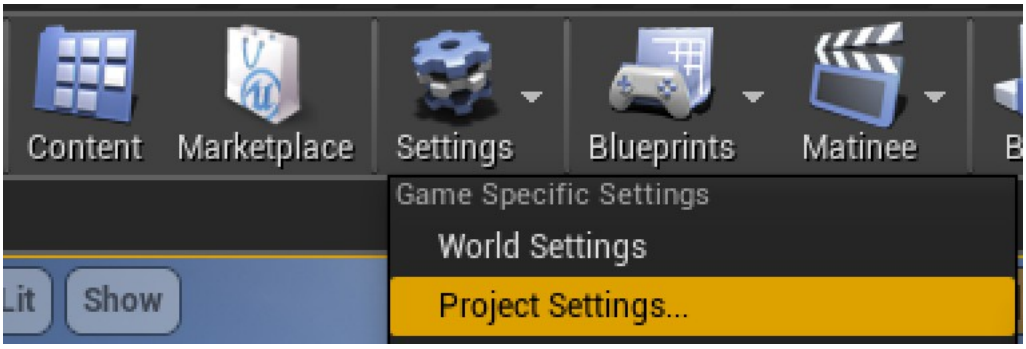


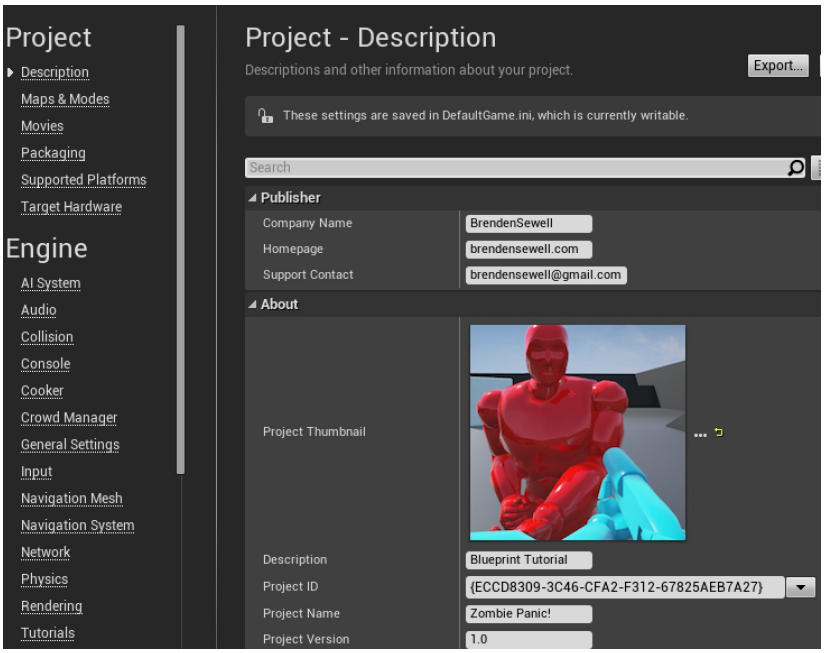
Chapter 8: Building and Publishing

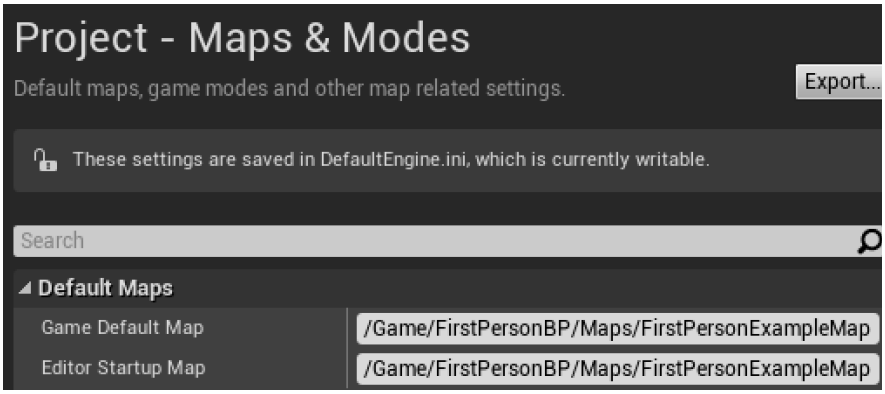
Optimizing your graphics settings

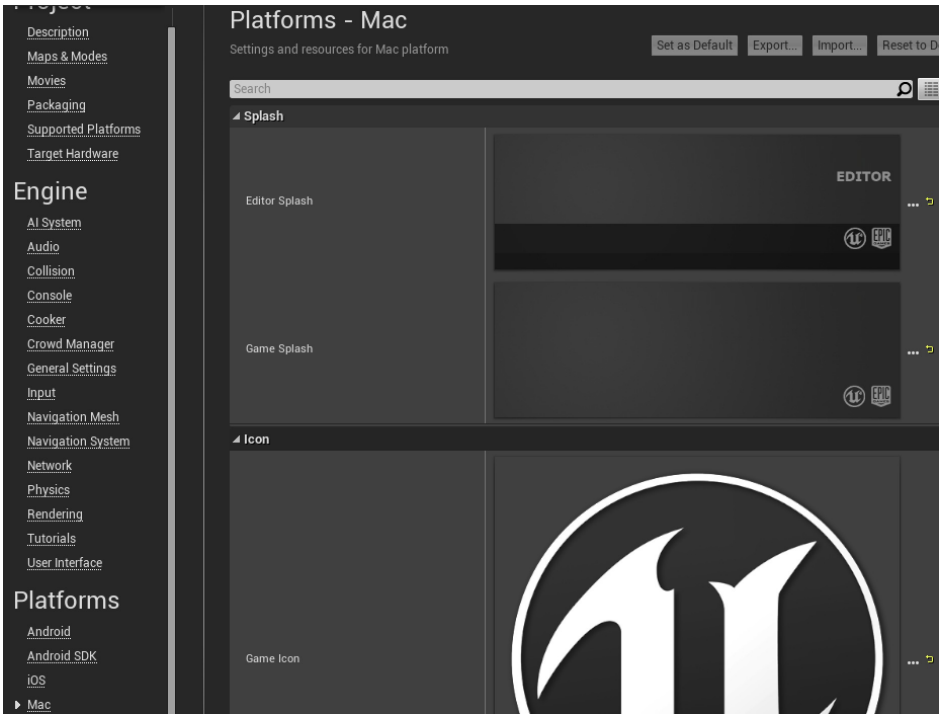


Setting up our game to be played by others




Project
Description
Maps & Modes
Movies
Packaging
Supported Platforms
Target Hardware
Engine
AI System
Audio
Collision
Console
Cooker
Crowd Manager
General Settings
Input
Navigation Mesh
Navigation System
Network
Physics
Rendering
Tutorials
Project - Description
Descriptions and other information about your project.
Export...
These settings are saved in DefaultGame.ini, which is currently writable.
Search
Publisher
Company Name
BrendenSewell
Homepage
brendensewell.com
Support Contact
brendensewell@gmail.com
About
Project Thumbnail
Description
Blueprint Tutorial
Project ID
{ECCD8309-3C46-CFA2-F312-67825AEB7A27}
Project Name
Zombie Panic!
Project Version
1.0


Project - Maps & Modes
Default maps, game modes and other map related settings.
Export...
These settings are saved in DefaultEngine.ini, which is currently writable.
Search
Default Maps
Game Default Map
/Game/FirstPersonBP/Maps/FirstPersonExampleMap
Editor Startup Map
/Game/FirstPersonBP/Maps/FirstPersonExampleMap


Description
Maps & Modes
Movies
Packaging
Supported Platforms
Target Hardware
Engine
AI System
Audio
Collision
Console
Cooker
Crowd Manager
General Settings
Input
Navigation Mesh
Navigation System
Network
Physics
Rendering
Tutorials
User Interface
Platforms
Android
Android SDK
iOS
Mac
Platforms - Mac
Settings and resources for Mac platform
Set as Default
Export...
Import...
Search
Splash
Editor Splash
EDITOR
Game Splash
Icon
Game Icon

Packaging the game into a build