

Rising Star Mechanic Project Plan

Contents

Initial Drafting	2
Objective	2
Pre-Production	3
Mechanics and UI Development	4
Core Mechanic	4
Second Mechanic	8
Third Mechanic	10
Fourth Mechanic	12
User Interface	13
Mechanic Design Documentation	14
Assets Used	17
References	17

Nathan Cooper

Initial Drafting

My main idea for a mechanic was to have the player use a torch to deter enemies and have no weapons. This mechanic would force the player to use their only deterrent sparingly, and also find objects to keep their torch functioning.

During the conceptualisation of the mechanic, I had one other idea, which was a respawn mechanic in a single player game, where the player would possess another character instead of ending the game upon death. I decided to create the torch mechanic, as I believed there is more potential to this mechanic. Once I had drafted out how the mechanic would work, I would then start to gather research on how to create the mechanics and features for this project.

I conceptualised different methods and features of this mechanic and believe there is potential for this mechanic to be made into a video game. My main inspirations for the mechanic were taken from a real-life scenario from a previous job where I was working a night shift, the sight mechanic from *SCP – Containment Breach* (Undertow Games, 2012), and from scenes in *I Am Legend* (Warner Bros. Pictures, 2007).

Objective

My main objective from this project was to create a short experience which showed the four mechanics working together in their different forms. My idea for these mechanics is to be part of a game which is set in an office building, where the player is a security guard on a night shift. The enemies are in the players imagination and would be seen as ghosts or spirits. They would hover above the ground, and move quickly, in order to scare the player and create an intense experience. Whilst the objective of the game is to illuminate each room in the building in order to win.

The mechanic is to serve as a deterrent towards the enemies. The player would have no weapons and would only have a torch. The torch would either scare off the enemies or kill them, as these enemies are vulnerable to any source of light. Additionally, the torch has a battery life, which the player must pick up additional batteries, in order to keep the torch functioning. The player must also use the torch to charge dead lights, to create additional safe zones, and also kill or deter the enemies in these safe zones, as the enemies would not be able to travel past the lights.

Mechanics and UI Development

Core Mechanic

The core mechanic is the torch, which the player must shine on enemies to kill them. They will eventually die, and the player must find batteries in order to keep the torch alive. As the battery drains, the light of the torch will become less intense, which is an additional visual aid towards the player, alongside the UI bar on screen which shows how much life is left in the battery.

In order for this mechanic to function correctly, I needed to create a spotlight and static mesh component and attach them to the first person character in the First Person Character blueprint. The static mesh was changed to a cone to simulate the line of sight of the torch, and also act as a collision trigger. I then set up the collision overlap for the Sight Cone, in order to tell whether an enemy is in view or not. I included a short animation as a trigger, to know when the code has begun to work correctly. This then casts an interface message to the enemy blueprint to execute the destroy actor code. I also needed to create a Boolean variable, which checks if the torch is on or not, and is also set in other lines of code.

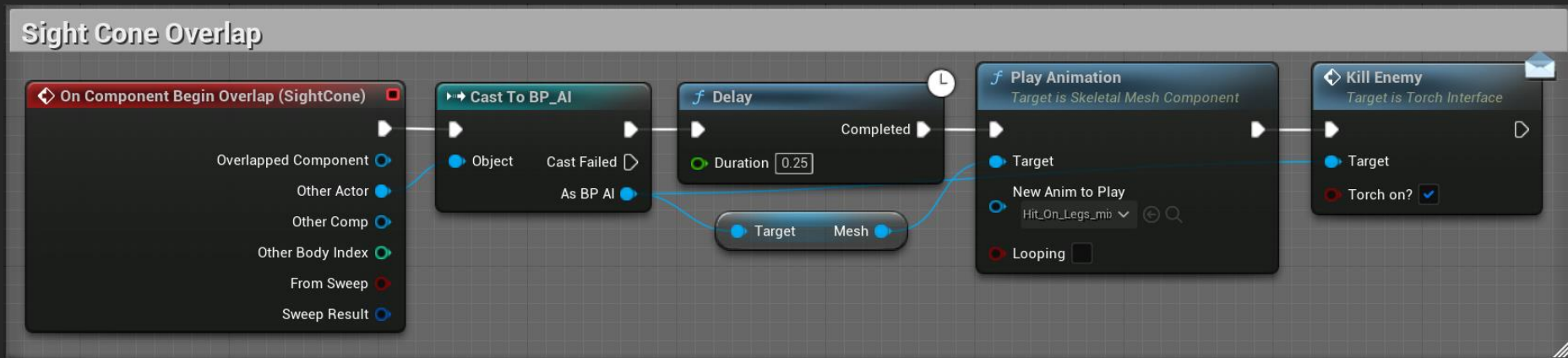


Figure 2 - Sight Cone Overlap

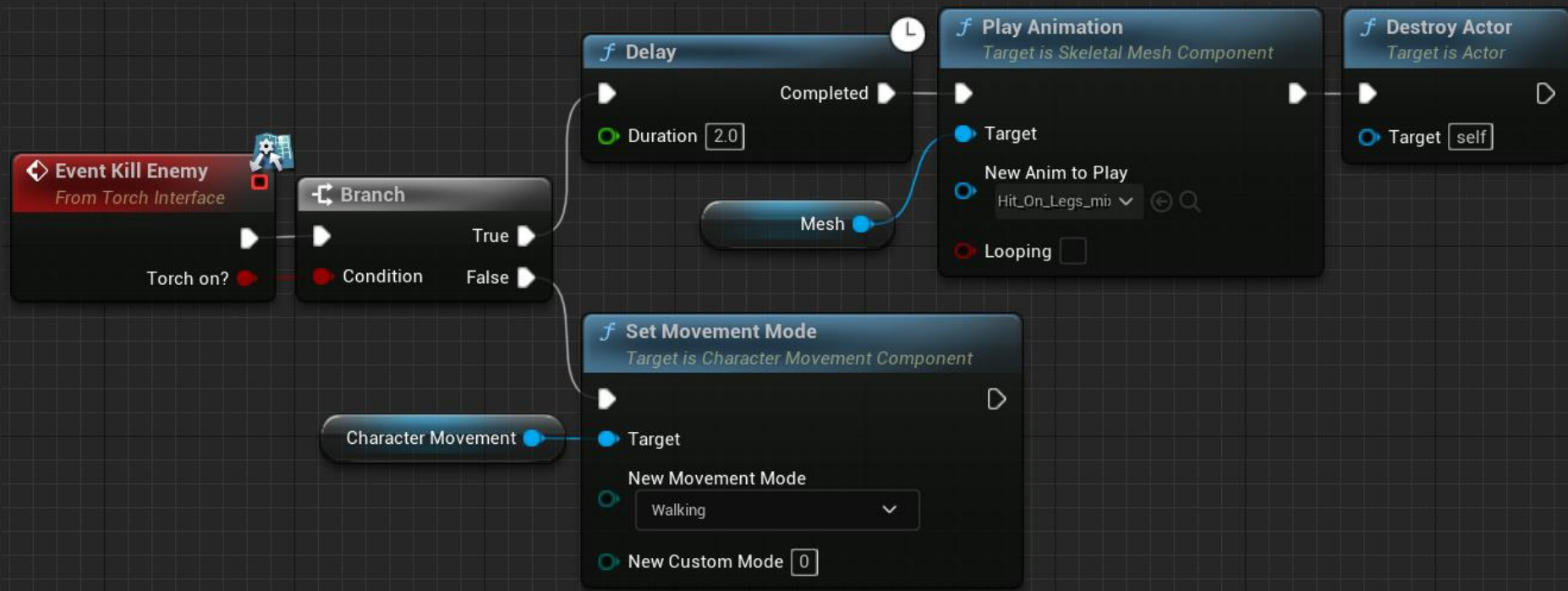


Figure 3 - AI Blueprint

The next implementation was to scale the Sight Cone with the wall, to make sure the Sight Cone did not pass through any walls. As this would cause the mechanic to incorrectly function. This feature was particularly difficult, as I needed to gather research in order to know how to implement this feature. The nodes below, scale the Sight Cone with any wall to make sure the Sight Cone does not pass through and kill any enemy beyond the players line of sight, by dividing the scale of the Sight Cone whenever it comes into contact with a wall, using the Line Trace By Channel node.

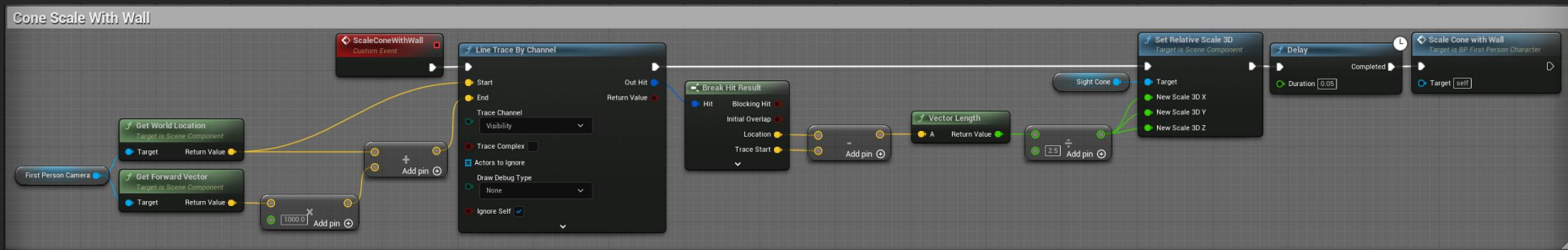


Figure 4 - Cone Scale Code

As the game begins, I needed to disable the collision of the Sight Cone and set the torch to hidden. As through play testing I found that the Sight Cone had automatically spawned even if the torch is off. As the Sight Cone should only spawn when the torch is on. Additionally, as the game begins, the Sight Cone custom event begins, as well as the widget and first person mesh nodes being executed.

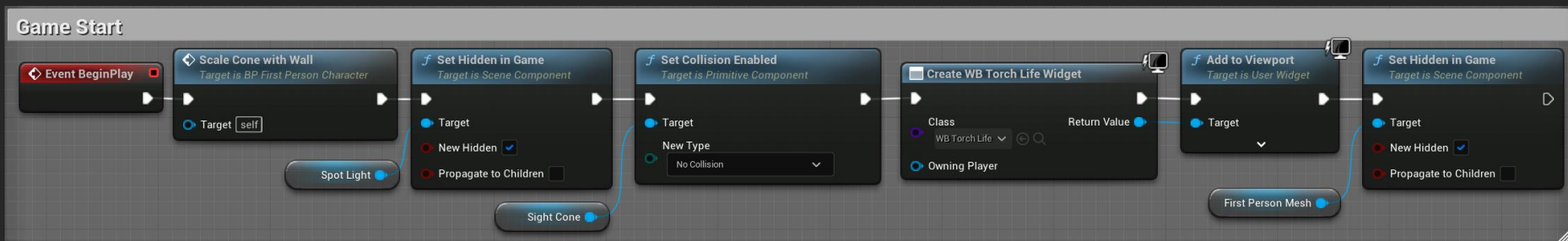


Figure 5 - Game Start Code

Since this mechanic relies on the player using a torch, the player would need to be able to access the use of this element and be able to toggle the torch on and off. This was implemented by firstly using the Flip Flop node in order to know if the player had pressed the F key, which turns the torch on and off. Then checking if the battery of the torch was higher than 0 using a Branch node. If the battery was not higher than 0 then the player would not be able to turn the torch on. With the B line of the Flip Flop then being executed. However, if there was still battery life then the A line of the Flip Flop node would run, and the Drain Battery event would also continue. Including the Boolean torch check being set on each line.

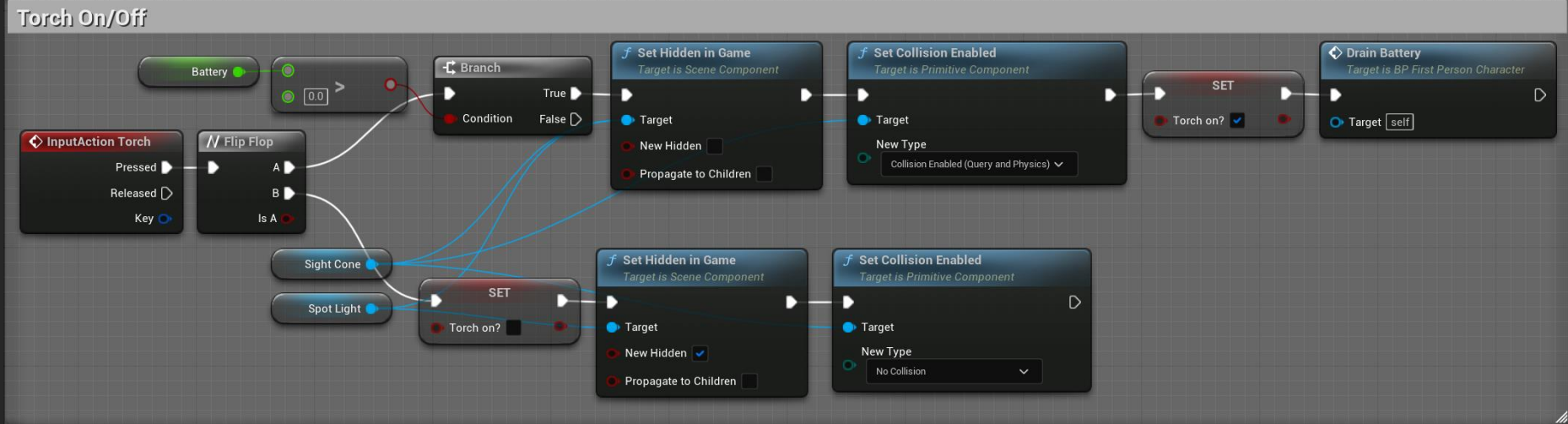


Figure 6 - Torch On/Off

Second Mechanic

The second mechanic are the batteries, which the player would have to collect, in order to keep the torch alive and kill the enemies. For this mechanic I needed to create two float variables. Battery, and Max battery. The Battery variable would check for the current state of the battery and has a default value of 0.8. While the Max Battery variable indicates the max life of the battery and has a default value of 1. For the battery mechanic to work I needed to create two custom events which drained the battery percent and restores the battery life once a battery has been picked up.

The drain battery event checks the Torch On variable and then drains the battery life in one second increments of 0.05 as the torch stays on. I also lowered the intensity of the torch as an additional visual aid for the player when the battery life reaches 0.4 and becomes less intense over eight seconds and then dies. This intensity is triggered using a Timeline node. Once the battery life has drained then the spotlight and Sight Cone are no longer called upon in game, and the torch intensity is reset once the battery has died. Furthermore, the Drain Battery event is looped to make sure the event is continuously called upon, as the battery must continue to drain when it is on.

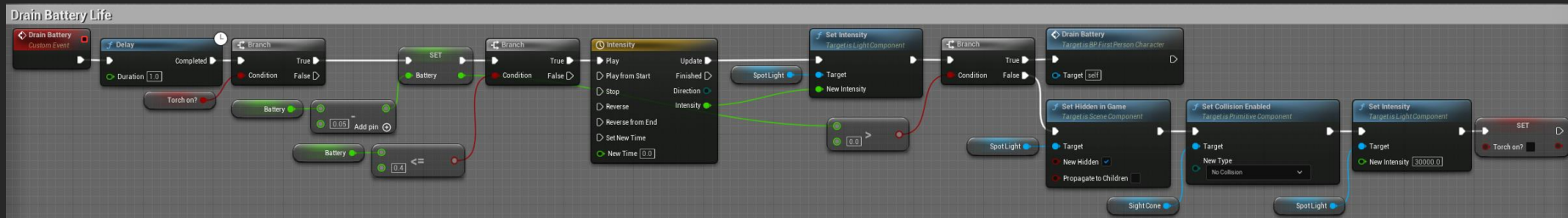


Figure 7 - Drain Battery Life

The Battery Pickup custom event restores the battery life and will be called upon once the player has collected a battery. This event was simply created by setting the battery variable back to 1 whenever the player overlaps with a battery. This custom event is called upon in the battery actor blueprint.

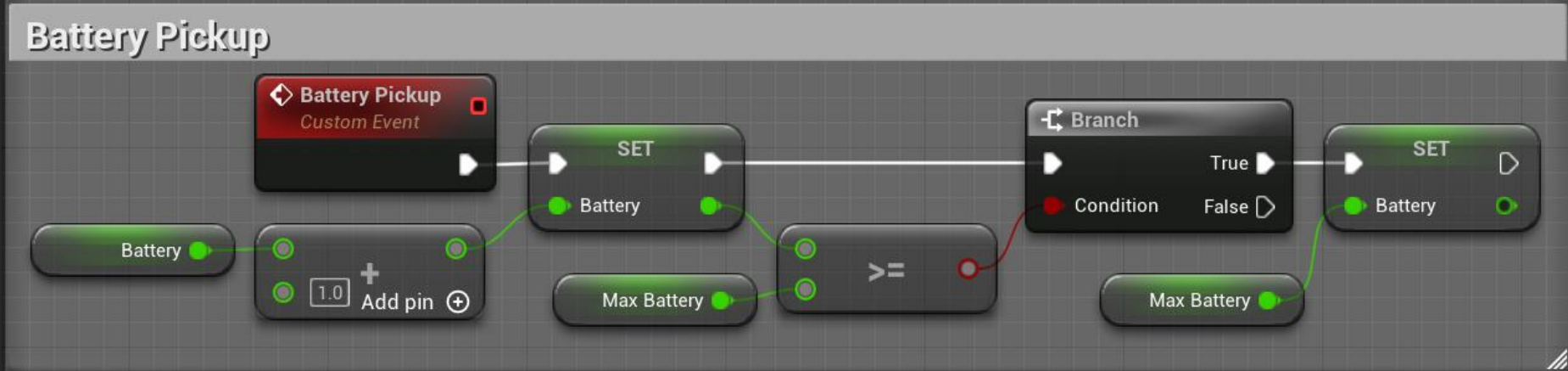


Figure 8 - Battery Pickup

Below is the collision box overlap for the battery actor. Which casts to the first person character and then calls the Battery Pickup event. Once the player overlaps with the battery, the battery actor will be destroyed, and the torch will return to full battery life.

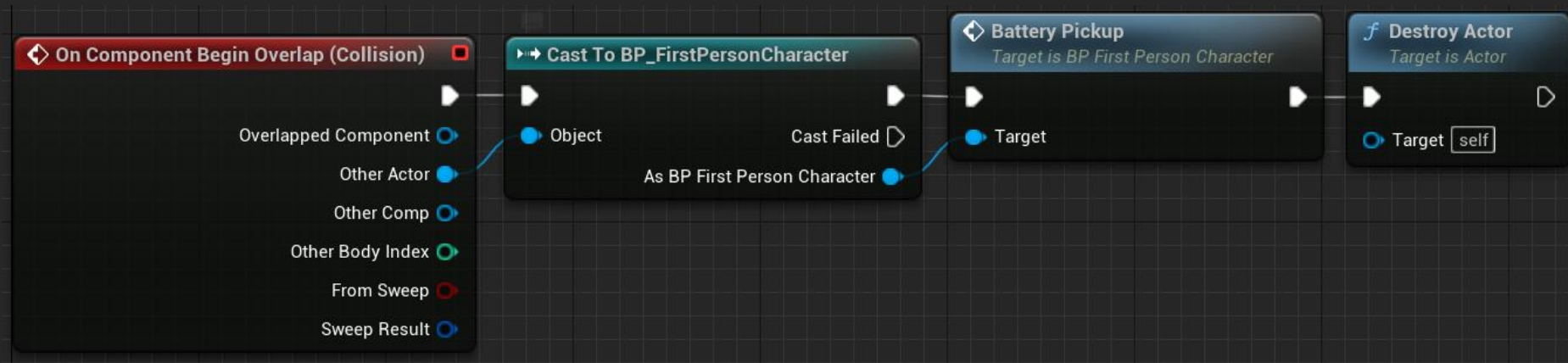


Figure 9 - Battery Actor Blueprint

Third Mechanic

The third mechanic is the ability to recharge dead lights with the torch. This mechanic was created by adding a Sphere Collision and placing it over the Point Light of the ceiling light. Then checking if the Sight Cone of the player is overlapping with the Sphere Collision of the ceiling light. If so, then the intensity of the ceiling light will increase over five seconds. With an ambient sound being emitted from the recharged light. Once the light is fully emitting then the code for the fourth mechanic will begin to execute.

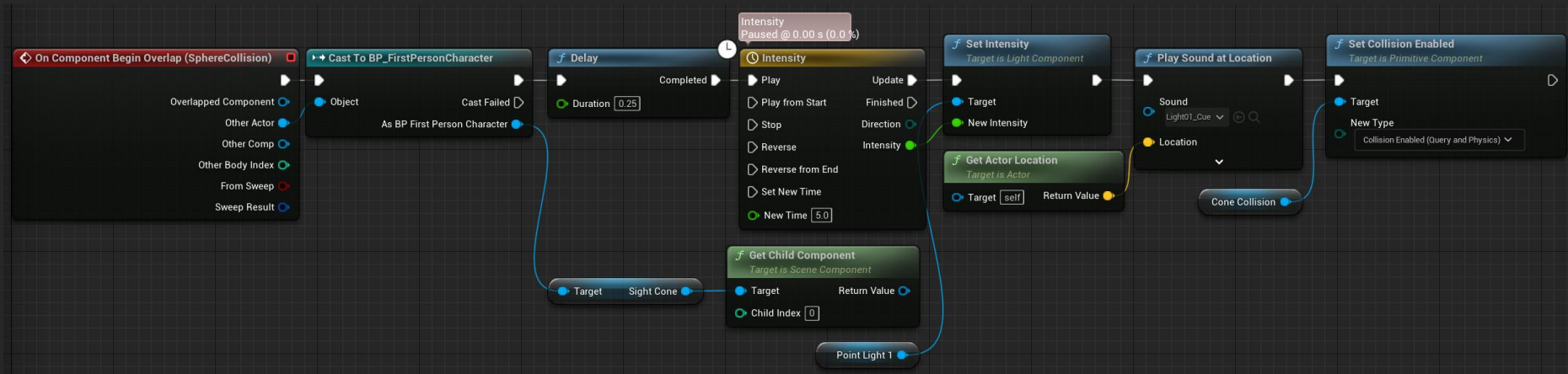


Figure 10 - Ceiling Light Blueprint

Similar to the torch, the components of the ceiling light are also disabled once the game begins. As these components can only be activated by the player.

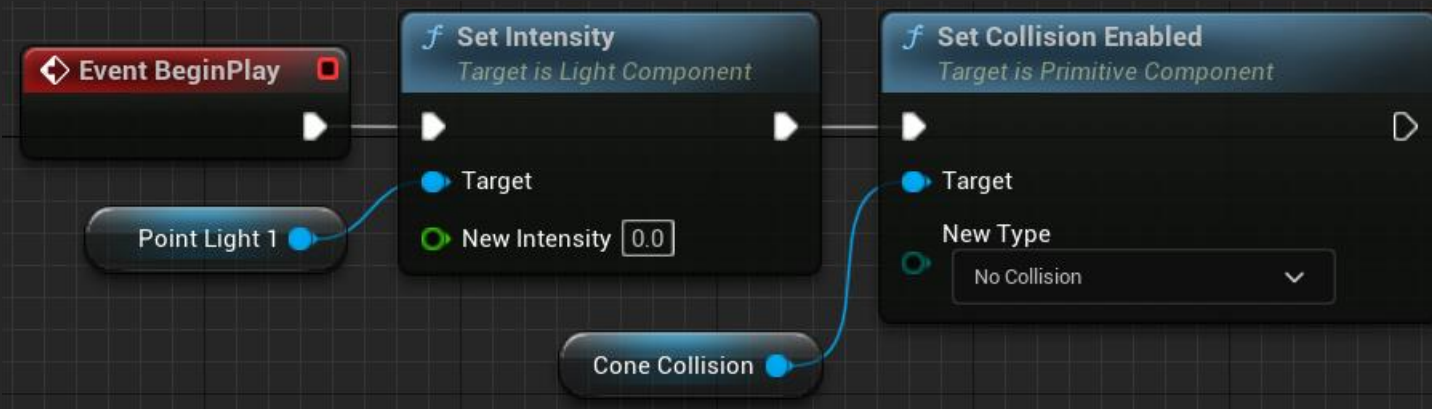


Figure 11 - Ceiling Light Begin Play

Fourth Mechanic

The fourth mechanic is once the dead lights are recharged these lights will act as a safe zone for the player as the enemies will not be able to pass through, and they will be killed by this newly created lighting. Once the player has recharged a light then a cone collision will spawn which covers the area of light being emitted by the ceiling light. Furthermore, once an enemy overlaps with this cone then the Kill Enemy interface message will trigger the death animation of the enemy.

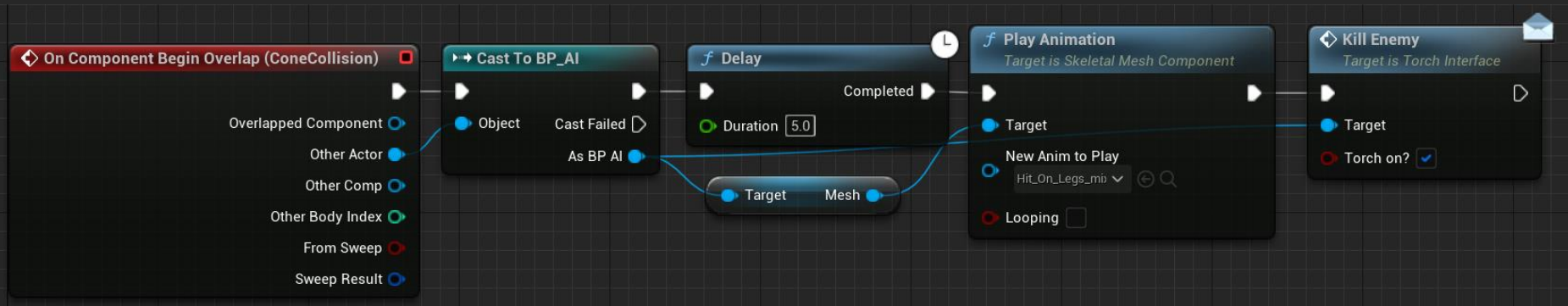


Figure 12 - Ceiling Light Cone Collision

User Interface

I wanted to create a simple visual aid for the player, in order for them to see how much battery life is left. I created a Widget Blueprint, and then inserted a progress bar. This progress bar was binded to the Drain Battery event in the First Person Character blueprint, which then drained as the torch stayed on.



Figure 13 - UI Code

Mechanic Design Documentation

Contents

Overview	15
Key Features & USPs	15
Genre	15
Audience	15
Platform	15
Mode	15
Engine	15
Gameplay	15
Game Objective	15
Background	15
Mechanics	16
Ideal Features	16
Project Milestones	16

Overview

Key Features & USPs

- Ability to use the light from the torch as a deterrent
- Play as a security guard attempting to illuminate the whole building
- Ability to recharge dead lights to use as additional weapons against the enemies

Genre

The game will be a first person horror game.

Audience

The target audience of the game will be ages 18 and over.

Platform

The game will be released on PC.

Mode

Single Player

Engine

Unreal Engine 5.0.3

Gameplay

Game Objective

The objective of the game is that the player must illuminate the office building they are in, while working the night shift. The player would only be armed with a torch and have to recharge dead lights around the building to complete the game and eradicate all of the enemies.

Background

When I was working the night shift in my previous job, I got the idea for this game and the mechanics within it, by having the lights off in the room I was in, and thinking I kept seeing something moving across from me. Whilst I knew that nothing was there, I could not help but think how a game with just a torch and ghosts would be a fun experience, whilst also giving the player a sense of unease. Furthermore, I was also inspired by the sight mechanic in *SCP – Containment Breach* (Undertow Games, 2012). Where the game would know when the player is looking at an enemy.

Mechanics

The game has four main mechanics. The first mechanic is that the torch can kill enemies when the player is looking at them. Leaving the torch as the players only weapon. The second mechanic are the batteries, which the player must collect in order to keep the torch from shining and deterring the enemies. The third mechanic is the ability to recharge dead lights with the torch. The fourth mechanic is these rechargeable lights can kill the enemies and create additional safe zones, in which they would not be able to pass through.

Ideal Features

Since this project is a prototype showcasing the four mechanics working together, I have a list of ideal features, if this project was made into a full game. Some features including:

- Enemies being ghosts or spirits
- Enemy animations
- A full explorable office building
- VFX
- SFX
- Main Menu
- 3D Assets
- More visual aids for the player
- Textures and many more features

Project Milestones

Task	Completion Date	Additional Notes
Formulate Original Mechanic Ideas	30/12/2022	Thinking of an original creative mechanic
Research Creation and Implementation	22/01/2023	
Development of Mechanics	26/01/2023	
Bug Fixing	26/01/2023	Fixing issues from development
Playtesting	26/01/2023	
Playable Prototype	26/01/2023	Mechanics work as intended
Documentation Creation	27/01/2023	
Submission	27/01/2023	

Assets Used

Animation – Mixamo (n.d.) *Hit On Legs* [3D Animation]. Description: Hit On Back Of Legs With An Object While Running. Available online:

<https://www.mixamo.com/#/?page=1&query=hit+on+legs&type=Motion%2CMotionPack> [Accessed 23/1/2023].

Battery – unrealengine432 (2019) *Battery Low Poly* [3D Asset]. Available online:

<https://sketchfab.com/3d-models/battery-low-poly-c726f8e634f14a7b9bcde2b96b3f7ba4> [Accessed 23/1/2023].

Character – Mixamo (n.d.) *Maynard* [3D Character]. Available online:

<https://www.mixamo.com/#/?page=1&type=Character> [Accessed 23/1/2023].

References

Aliens (1986) Directed by James Cameron [VHS]. 20th Century Fox.

I Am Legend (2007) Directed by Francis Lawrence [DVD]. Warner Bros. Pictures.

Undertow Games (2012) *SCP – Containment Breach* [Video game]. Available online:

https://store.steampowered.com/app/2090230/SCP_Containment_Breach_Remastered/ [Accessed 24/01/2023].