

# **Killer Drones**

TILAK MAHARASHTRA VIDYAPEETH, PUNE

(Dept. Bachelors of Computer Application)



A PROJECT REPORT ON

## **Killer Drones**

**Genre - First Person Shooter**

Submitted by

Shubham Awchare

Towards the partial fulfillment of the  
**Bachelor of Computer Application**

**[Game & Mobile Software  
Development]**

**Tilak Maharashtra Vidyapeeth, Pune**  
**[2019-2022]**

---

## CERTIFICATE

This is to certify that the project word entitled

**“Killer Drones”**

Has been satisfactorily completed and carried out by

**Shubham Sunil Awchare**

Towards the partial fulfillment of the  
“Bachelor's of Computer Application”

For the academic year [2019-20],  
Tilak Maharashtra Vidyapeeth, Pune.

This project is completed under the guidance of  
**Head of Department**

Project Guide

Examiner

Head of Department  
[TMV - 01 - PUNE]

---

## **Acknowledgment**

With immense pleasure, I am presenting the “*Killer Drones*” Project report as part of the curriculum of ‘*Bachelor’s of Computer Application*’. I wish to thank all the people who gave me unending support.

I express my profound thanks to our head of department, project guide and project in-charge Mr. Abid Shaikh and all those who have indirectly guided and helped me in preparation for this project.

**Shubham Sunil Awchare**

---

## **Abstract**

‘*Killer Drones*’ is a single-player First Person Shooter on the concept of some of the famous shooting game like Call of Duty, Battlefield adding a new twist to it

### **Minimum System Requirement -**

OS - Windows 7, Windows 8, Windows 8.1 and Windows 10

RAM - 4 GB

Graphics Card - DX9

---

## **Contents**

### **1. Introduction**

- 1.1 Project Overview
- 1.2 Idea Generation
- 1.3 High Concept
- 1.4 Game Synopsis

### **2. Pre-Production**

- 2.1 Concept/Idea
- 2.2 Game Mechanics
  - 2.2.1 Core Mechanics
  - 2.2.2 Secondary Mechanics
- 2.3 References
  - 2.3.1 Games
  - 2.3.2 Art
- 2.4 Tools and Software required
- 2.5 Unreal Engine Gameplay Framework
- 2.6 Project Timeline

### **3. Production**

- 3.1 Blueprints Created
- 3.2 Set Animation Blueprint (Fn)
  - 3.2.1 Construction Script
  - 3.2.2 Play Fire Montage
  - 3.2.3 Set HUD
  - 3.2.4 Set Health
  - 3.2.5 Fire Weapon
  - 3.2.6 Perform line trace
  - 3.2.7 Add Ammo
  - 3.2.8 Heal
  - 3.2.9 Variables
  - 3.2.10 Event Graph

### **4. Post-Production**

- 4.1 Testing
- 4.2 Optimization

### **5. Conclusion and Developer comment**

### **6. Reference and Bibliography**

---

## **Introduction**

### **1.1 Project Overview**

This project is a single-player First Person Shooter Game. To play this game you require a working computer/laptop. The game can be played using the W,A,S,D keys to move the player, left-click to shoot bullets. The player can combat the AI by shooting some bullets at it. The main feature of my game is the aggressive mode of the drones when we shoot them.

The game was developed using Unreal Engine 5.0.0 and using blueprints . First Person Shooter games are fun to play, these games will never get old, Looking back at old Doom Games and now looking at the new Doom games somethings never change it just gets upgraded.

While the game may seem simple from the user's perspective, it requires a complex and diverse blueprints knowledge and different types of logic which will drive the gameplay.

I was thrilled to apply the key mechanics of the core gameplay like movement, camera rotation, patrol points, character HUD, etc.

## 1.2 Idea Generation

As my game has a first person element I've looked upon a lot of games from Drone the game, Call of Duty War-zone,etc .

Patrolling, movement, and items are inspired by some old games and looking into the future drones will be mostly used in war, surveillance, delivery, etc. I tried to mix the vibes by using old and new ideas .

## 1.2 High Concept

The concept is inspired by Spiderman: Far from Home and The Bourne Legacy. Keeping in mind that drones can be very deadly

when fallen into wrong hands. With every wave the enemies' winning chances will increase as they will grow stronger. Some enemies may have more health than others and some may have more damage than others. User needs to choose their attack wisely.

## 1.2 Game Synopsis

The Player is in search of new element discovered by a vigorously dangerous company called Diamond Co-Op. The element is so strong that it can use the full energy of sun to create a Supermassive blackhole which might be the end of the human race. Leon who is the main protagonist of the game who has been equipped with the best gear on the planet has been sent to North Korea to put and end to this mass genocidal plan of the Diamond Co-Op. Will he be able to stop the Cooperations evil plans? Will he prevail to safe the human race?

---

## **Pre-Production**

Pre-production is the phase of further developing and planning prior to the process of production. Like all great projects, the success is in the planning. The section gives an overview of the different steps involved in the pre-production phase, it helps in creating a roadmap for the future of the game.

### **2.1 Concept/Idea**

The concept is inspired by Spiderman: Far from Home and The Bourne Legacy. Keeping in mind that drones can be very deadly when fallen into wrong hands. With every wave the enemies' winning chances will increase as they will grow stronger. Some enemies may have more health than others and some may have more damage than others. User needs to choose their attack wisely.

Mainly my attention was drawn towards the gameplay mechanics used by games when we have to stealth kill or games like Thief or Assassins Creed has to offer while killing the enemies they are unaware and when they see a player or hear a noise they detect the player and get alerted.

I wasn't sure how to combine the elements of the 2 games in a single game. I had to brainstorm and made a decision that I will add both the mechanics of the games and adding consumables so that user can gain stats by consuming the items which can be found in the entire map. As I was working on the prototype, I started adding extra features just to test whether it will work or not. Turned out, some of the features were actually going well with the game.

**For example** - We can distract a guard in Assassins Creed by throwing a rock at any place near him while hiding, same goes for the Thief the protagonist throws a bottle to distract the enemies.

---

## **2.2 Game Mechanics**

### **2.2.1 Core Mechanics**

Every game has a core mechanic. A core mechanic is the essential play activity players perform again and again in a game. Sometimes, the core mechanic of a game is a single action, in a footrace, for example, the core mechanic is running. In a trivia game, the core mechanic is answering questions.

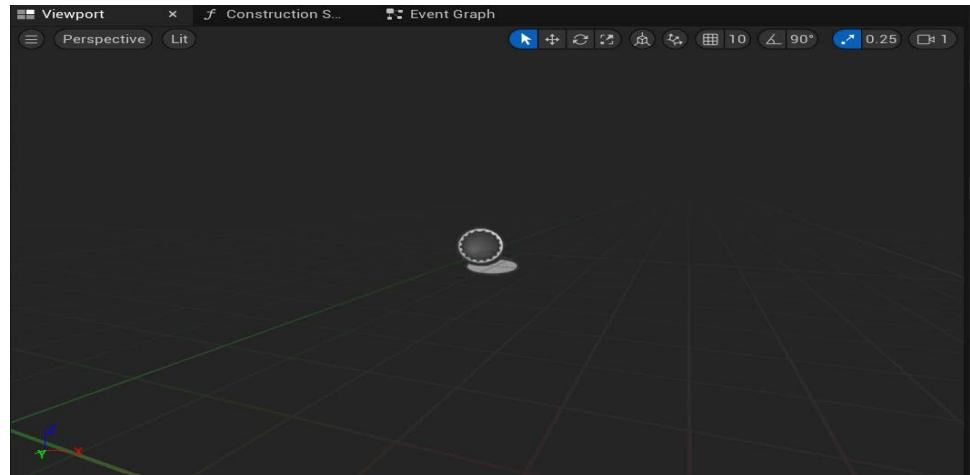
The core mechanic of my game is the Aggressive mode of the drones when they are being fired at.

### **2.2.1 Secondary Mechanics**

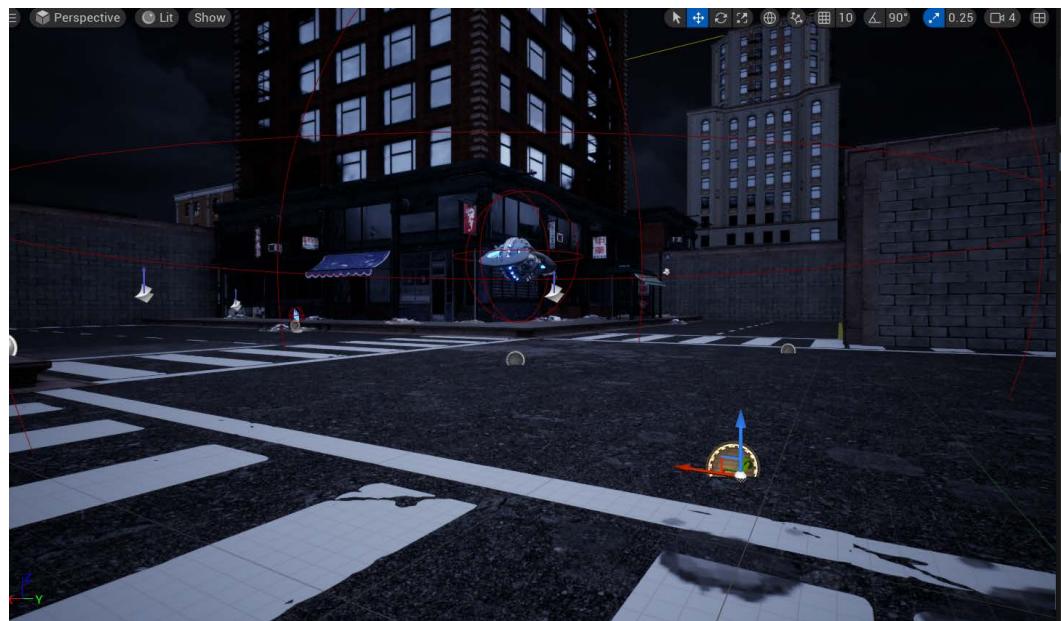
Secondary Mechanics are either available occasionally or require their combination with a core mechanic in order to be functional.

Secondary mechanics in my game are -

Patrolling - The drones are able to move from one location to another thanks to this patrolling points which have been spread throughout the map.



The Drones are constantly in motion so that the difficulty level will be maintained.



Aggro mode - The Light of the drones change to red when they are hit with a bullet.



---

## 2.1 References

### 2.2.1 Games

Some of the reference games which inspired me to go with my idea are -

- i) Drone the game



## ii) Call of Duty: Warzone



Drone the game, Warzone gave me the idea of the drone .  
Assassins Creed and Thief gave me the idea of the Aggro system.

## 2.2.2 Art

Some of the art references are from the following games

i) Spiderman



ii) Thief



Spiderman PS4 and Thief gave me the idea for the environment of the game. Combined both and came up with a little similar environment.



---

## **2.4 Tools and Software required**

The following tools and software were used for the development phase of the game -

- ➔ Unreal Engine 5.0.0
- ➔ Blueprints
- ➔ Assets from Unreal Engine Marketplace

System requirement for development.

- 16 GB RAM
- RTX 3060 Graphics
- AMD Ryzen 5 5600x Processor
- Windows 10

---

## 2.4 Unreal Engine Gameplay and Framework

Unreal Game Engine is one the few game engines that enables the developer to code their games using Blueprints and C++.

Unreal Engine (UE) is a 3D computer graphics game engine developed by Epic Games, first showcased in the 1998 first-person shooter game Unreal.

Initially developed for PC first-person shooters, it has since been used in a variety of genres of games and has seen adoption by other industries, most notably the film and television industry. Written in C++, the Unreal Engine features a high degree of portability, supporting a wide range of desktop, mobile, console and virtual reality platforms.

The developer community of the Unreal Game Engine is well connected with its user and often updates and adds new features with each release.

Understanding this framework is critical in being successful and efficient in building your own project.

Building games in Unreal Game Engine is made simpler because of the user-friendly interface and the pre-written classes which can be used to create almost any game mechanics.

The latest generation, Unreal Engine 5, was launched in April 2022. As its predecessor released in March 2014, its source code is available on GitHub after registering an account, and commercial use is granted based on a royalty model. Epic waives their royalties margin for games until developers have earned US\$1 million in revenue and the fee is waived if developers publish on the Epic Games Store. Epic has included features from acquired companies like Quixel in the engine, which is seen as helped by Fortnite's revenue.

We shall now go through some important blueprints that I used in the development of my project. I will be providing the snippets of my blueprints, and an overview of the script.

---

## 2.4 Project Timeline

A timeline is a chronological order of events. In most cases, it is a line with dates, events or actions. It is very important for project managers to develop skills to build an accurate timeline. It shows what phases are already in the past, what is in the progress now and what is supposed to be finished in future. It means it helps to be on projects' tracks.

*Of course, it depends on a project. But there are typical elements that timelines should include in any case.*

The key elements of a timeline:

Tasks that are to be accomplished in a project;

Dates of tasks;

Duration of tasks;

Dependencies between tasks.

As we discovered, every project should be based on a timeline. So, if you want to succeed with your project regardless of its purpose and deliverables, it should include an accurate timeline. All tasks there will have start and end dates, milestones and project deliverables.

But project management is not the only way to benefit from timelines. They are a good way for displaying historical events and depicting biographies.

In total this game took me around 1 month to complete. As I was juggling between my internship and final project. I only had weekends to work on my project.

---

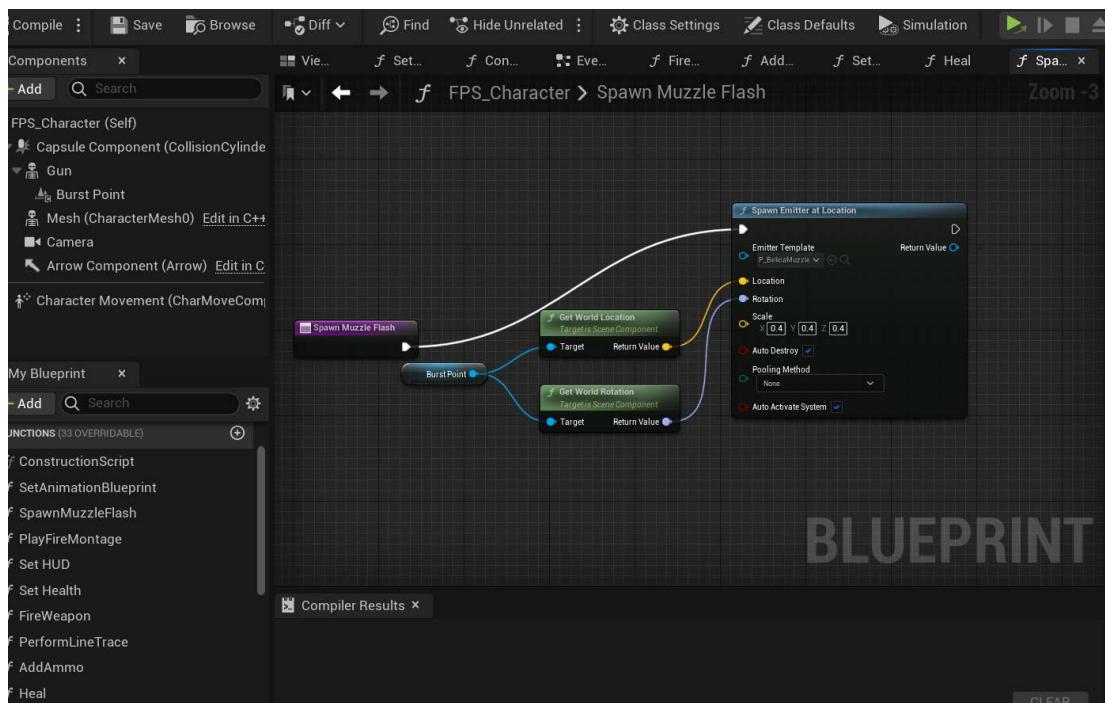
# Production

## 3.1 Blueprints Created:

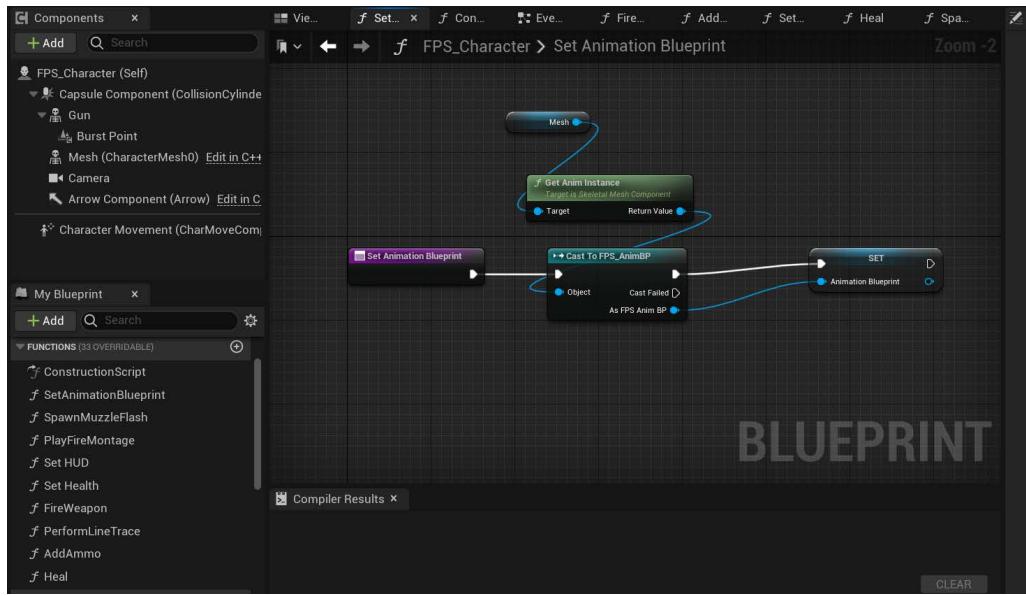
### ❖ FPS\_Character -

This is the utmost important blueprint of the project.

This Blueprint controls the character and its behavior towards the environment like firing the gun, healing the player, adding ammo, etc.

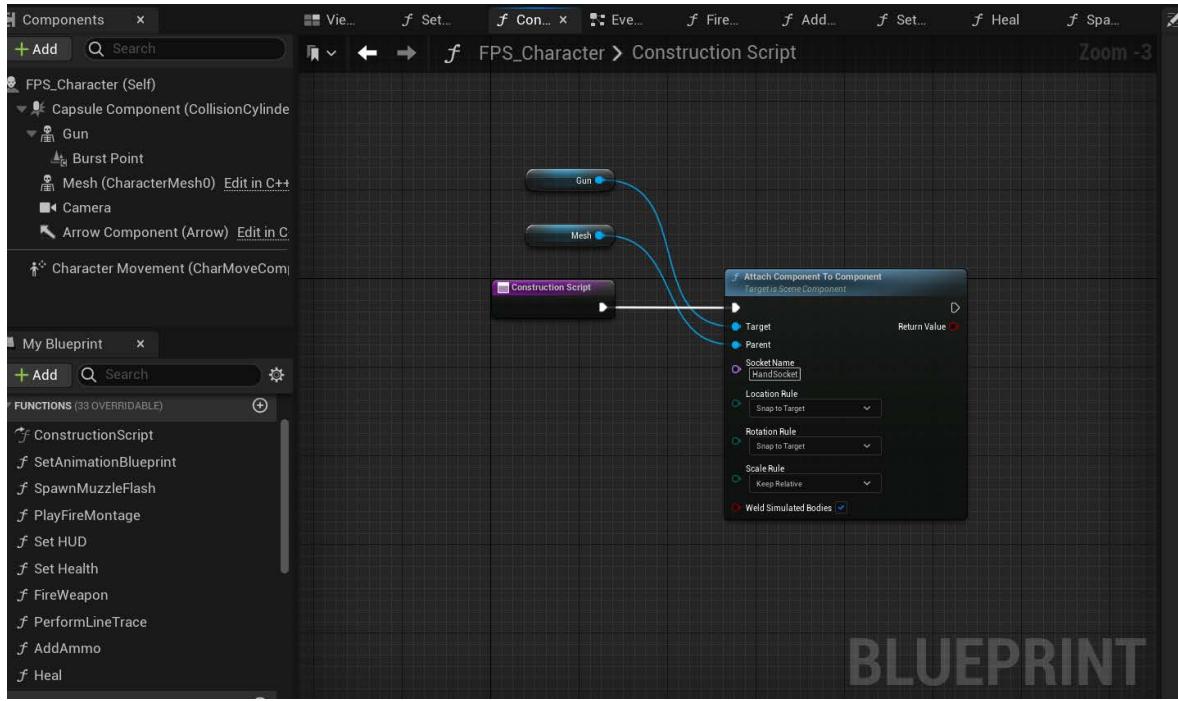


**SpawnMuzzleFlash(Function)** - By using this function when the bullet is fired from the gun we can see the muzzle flash displayed on the front of the gun.



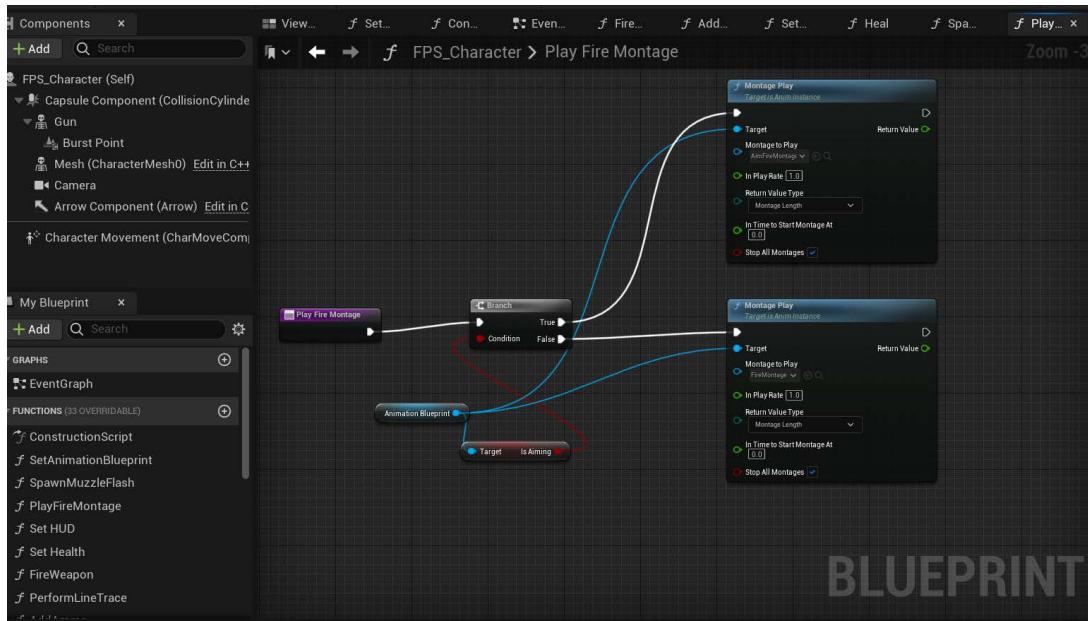
## ❖ Set Animation Blueprint (Function)-

This function is used to set a certain animation and cast it to the FPS\_Character Blueprint.



## Construction Script:

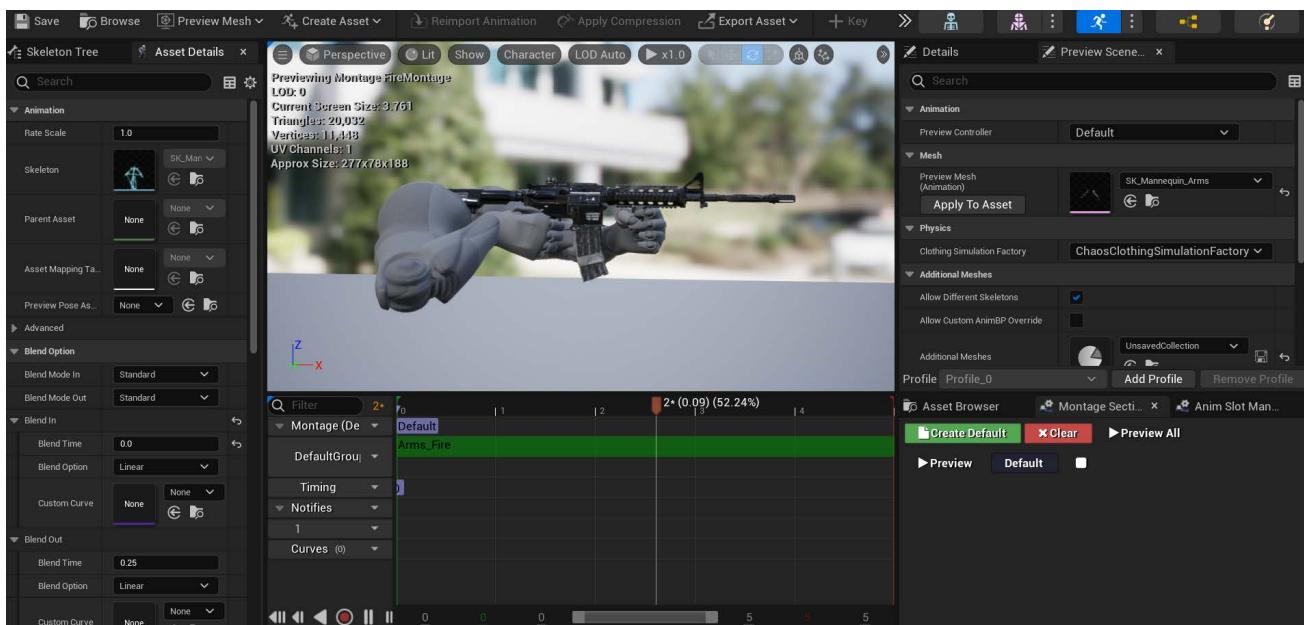
This function is used to keep the gun in the hand socket of the player by using the created Get gun variable and Get mesh then attaching it to component.

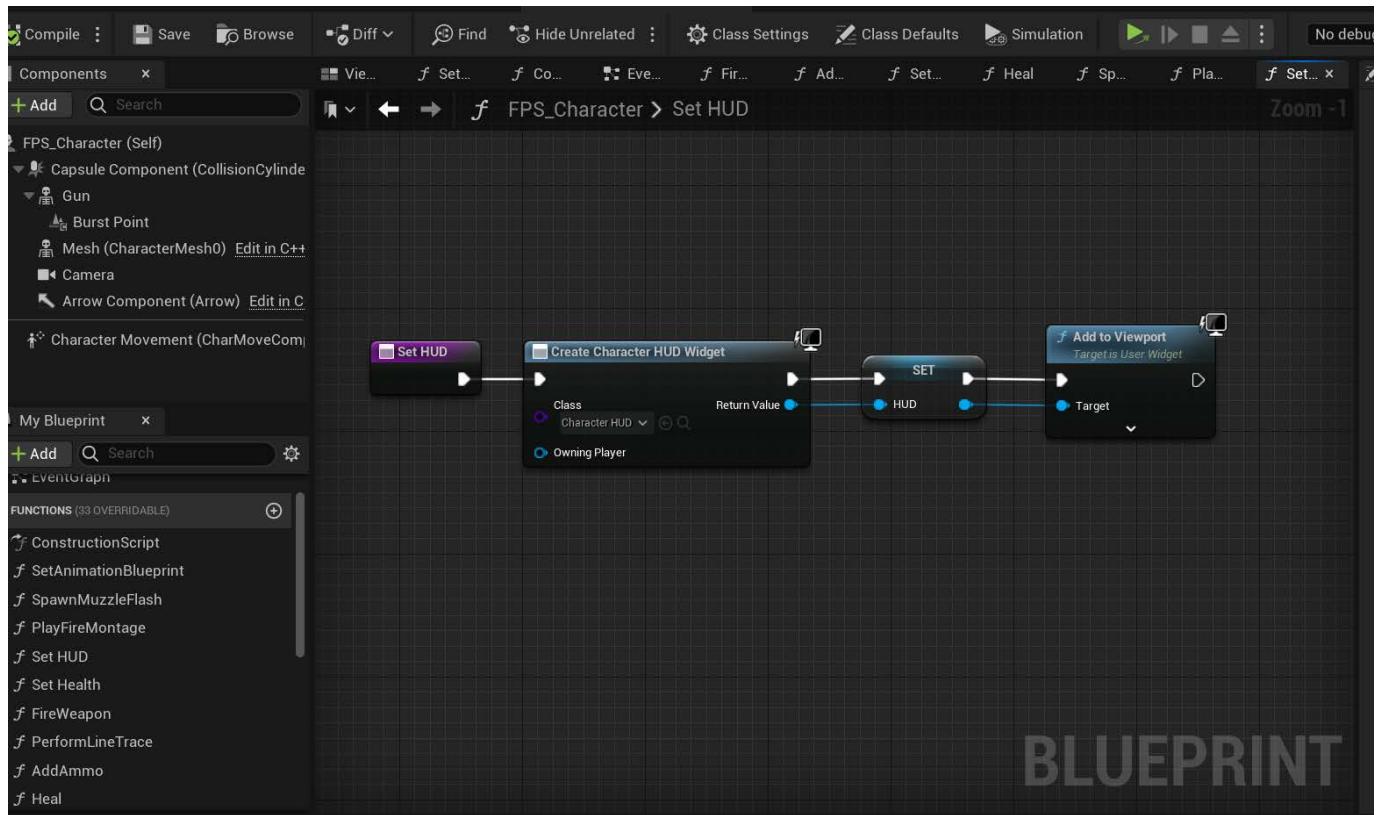


**BLUEPRINT**

### Play Fire Montage:

This function is used play the animation which is will play when firing the bullets.



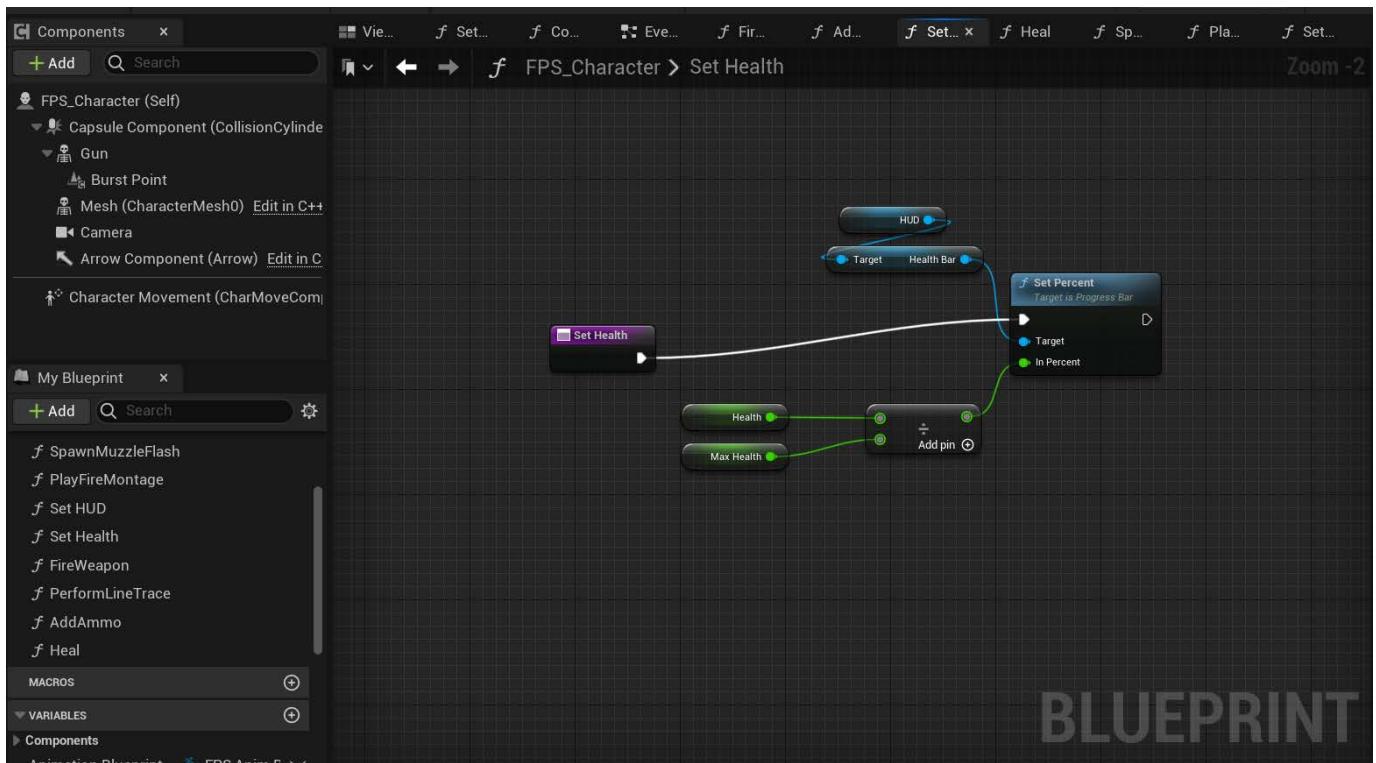


## Set HUD:

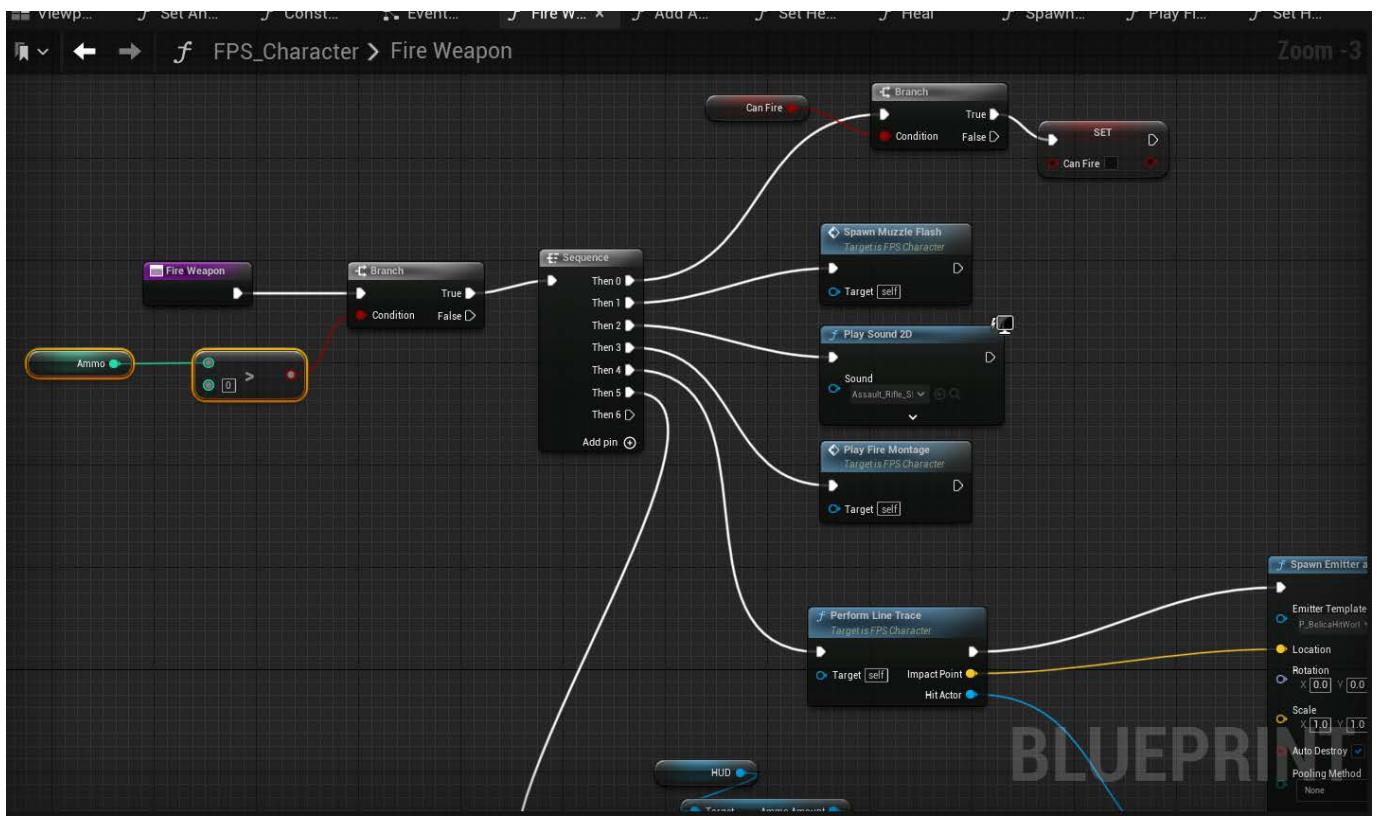
This function has been used to apply the crosshair, time remaining and drone count HUD.

## ◆ Set Health -

This function has been used to create the health of the player which is stored in a progress bar it can get decreased when hit by drone or can be increased with the help of pickup.

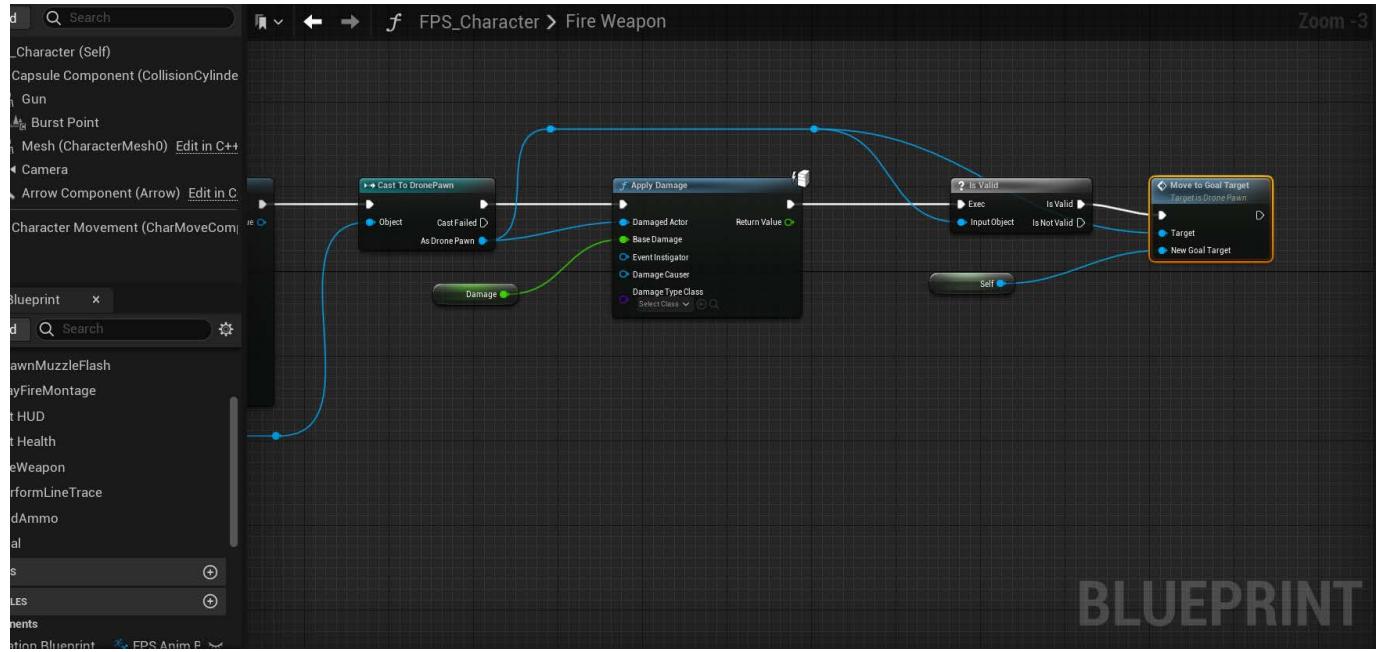
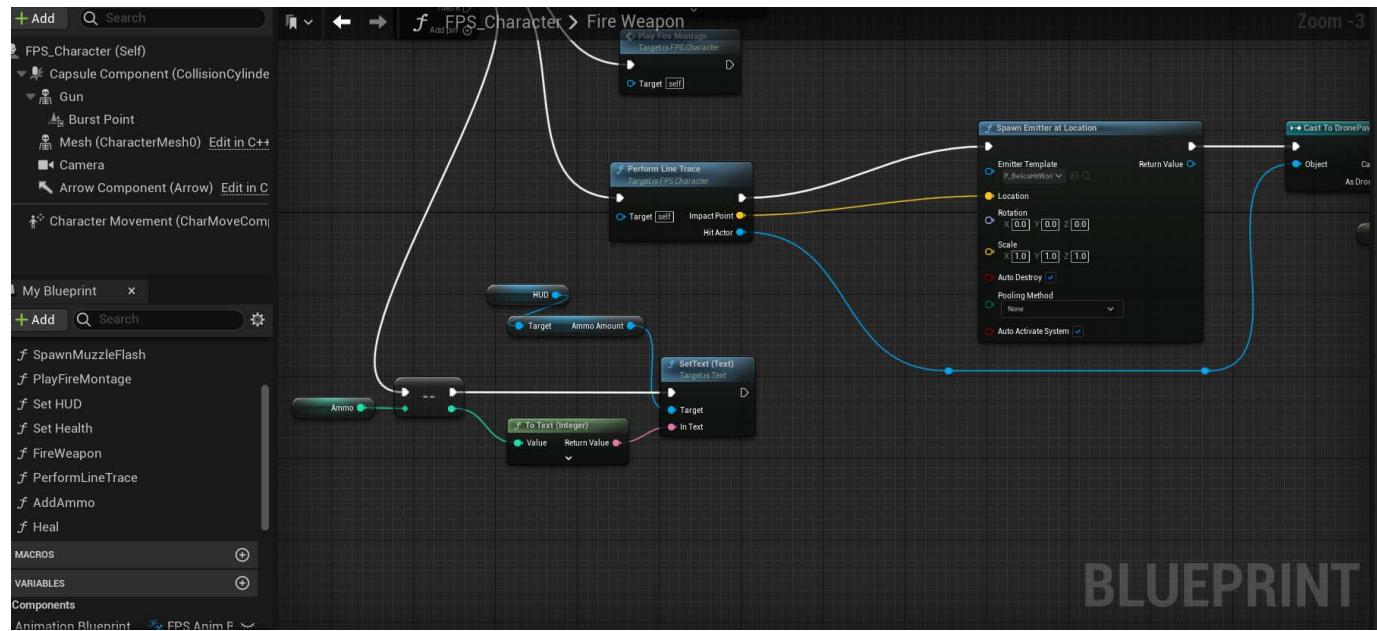


BLUEPRINT



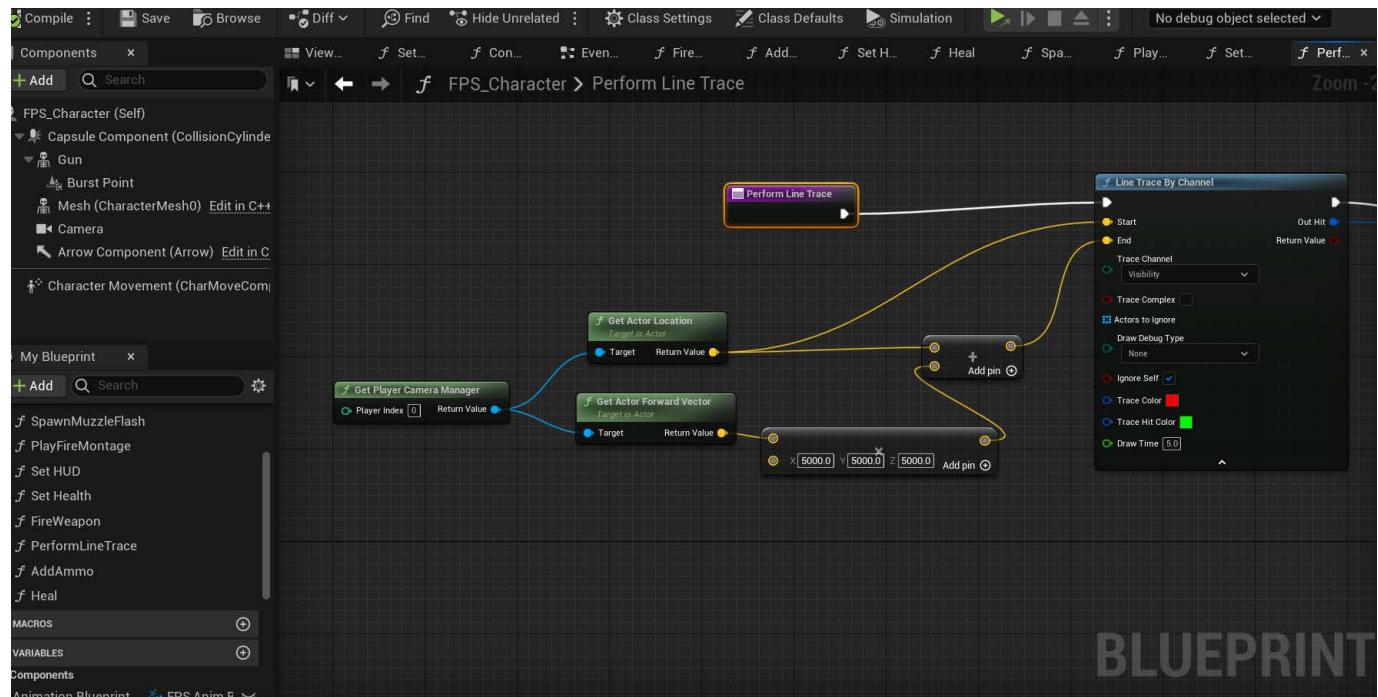
## ◆Fire Weapon -

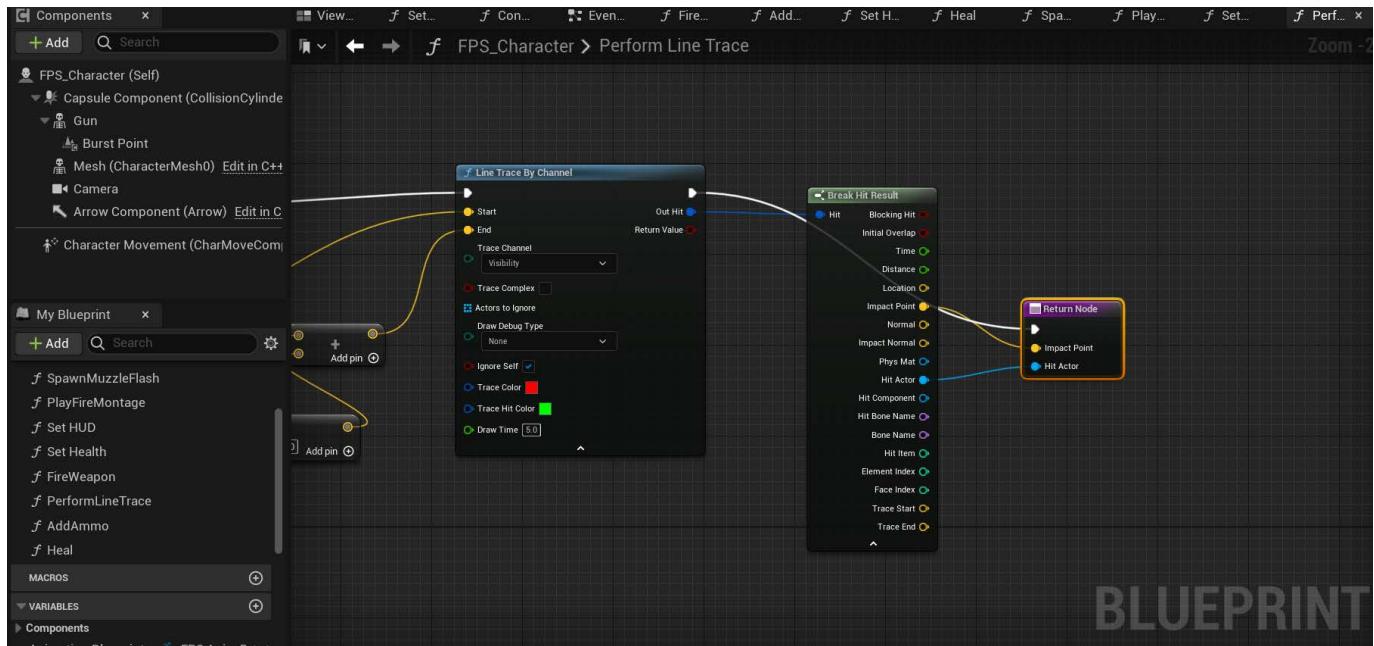
This function is very important cause this almost contains every single thing which will be used to fire the gun. It contains the muzzle flash, muzzle flash sound, fire montage, bullet impact, applying damage, etc



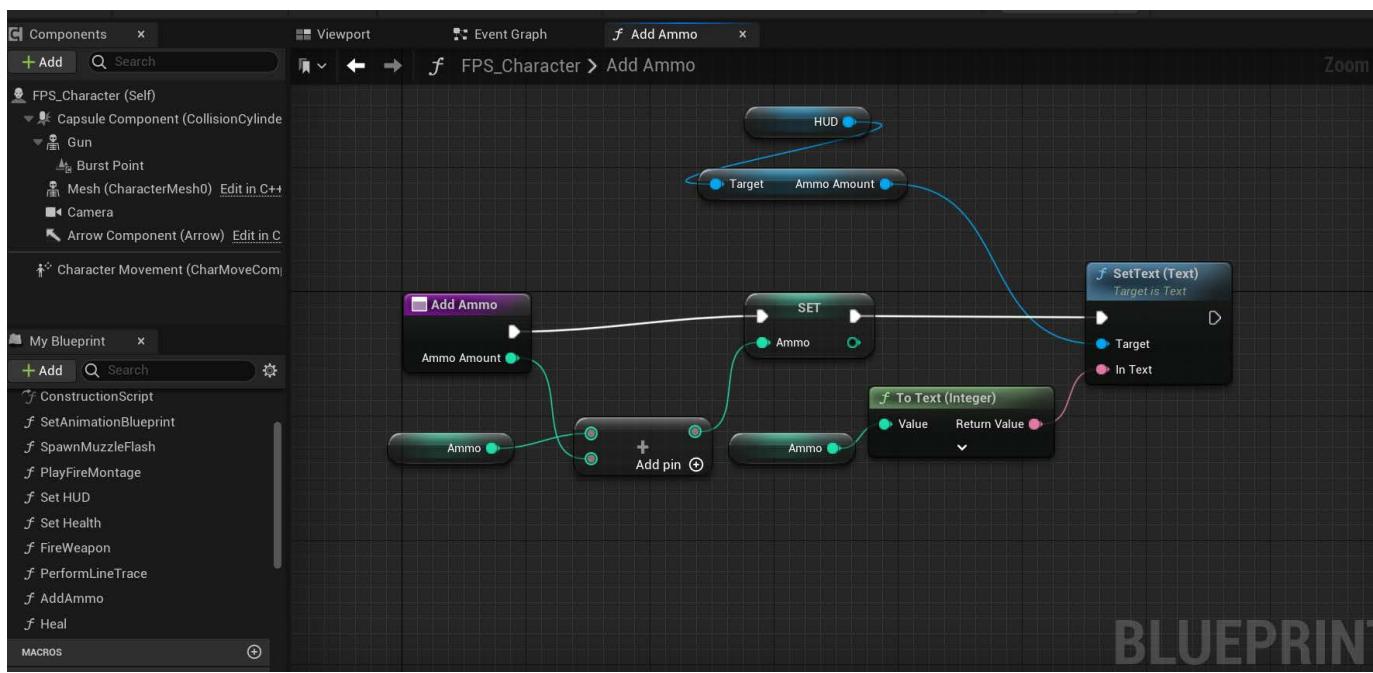
## ❖ Perform line trace -

This function is used to follow the location of actor and attack the player.





BLUEPRINT

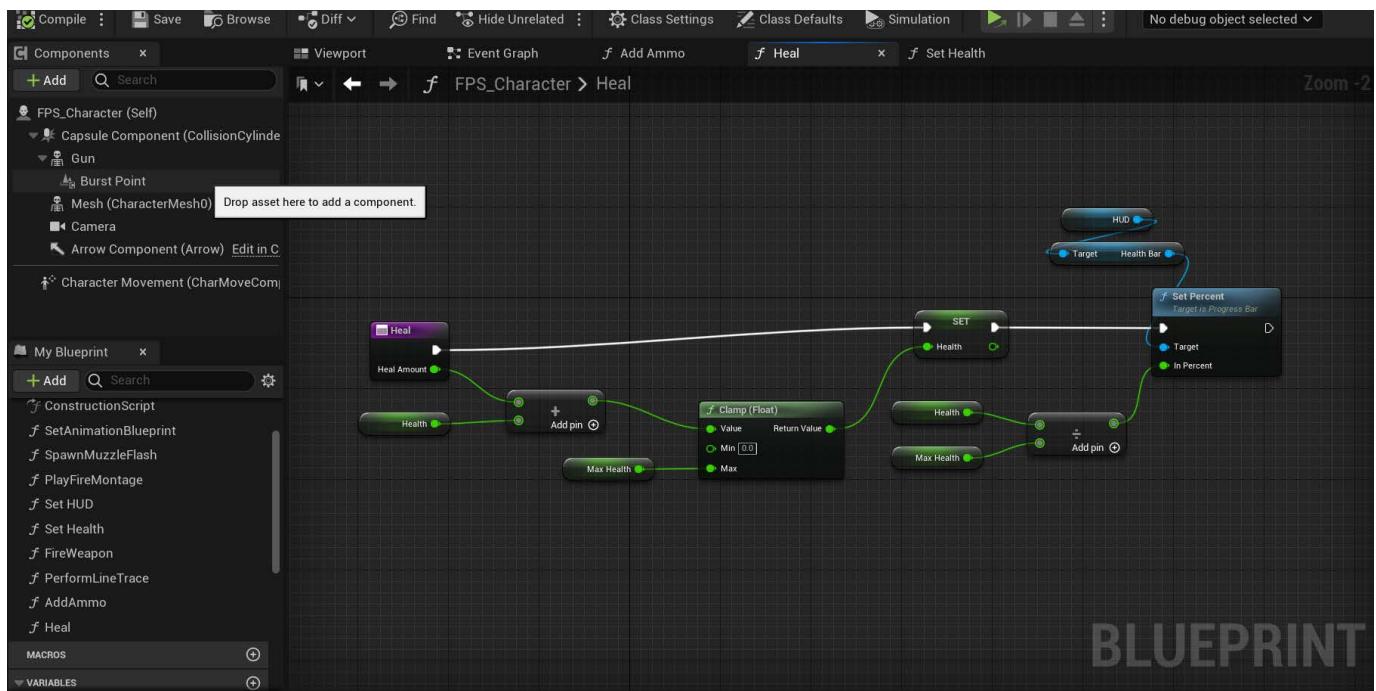


## Add Ammo :

This function has been used to pickup the ammo when the player interacts with the ammo pickup.

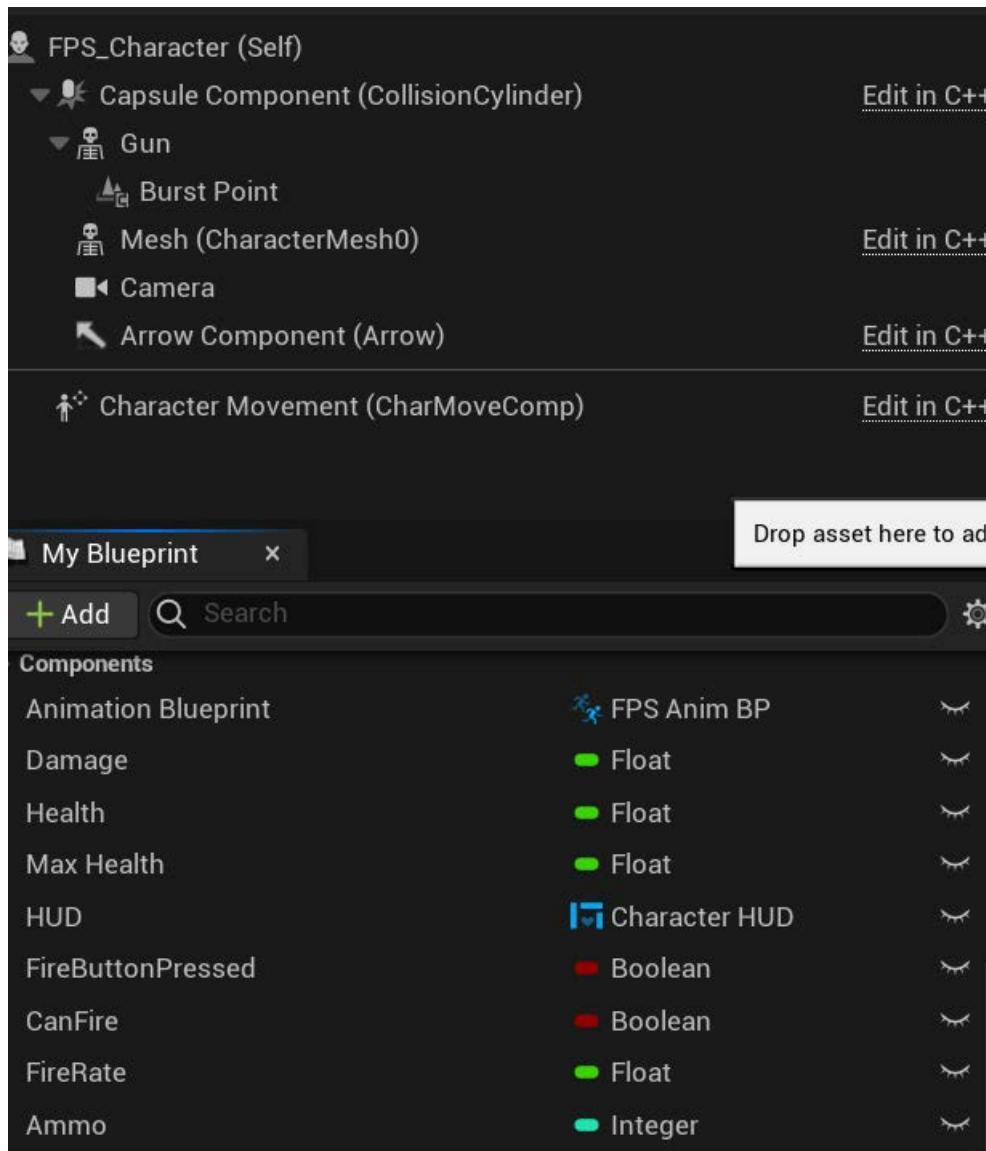
## ◆ Heal -

This function is used to heal the player when the player interacts with the health pickup in the map

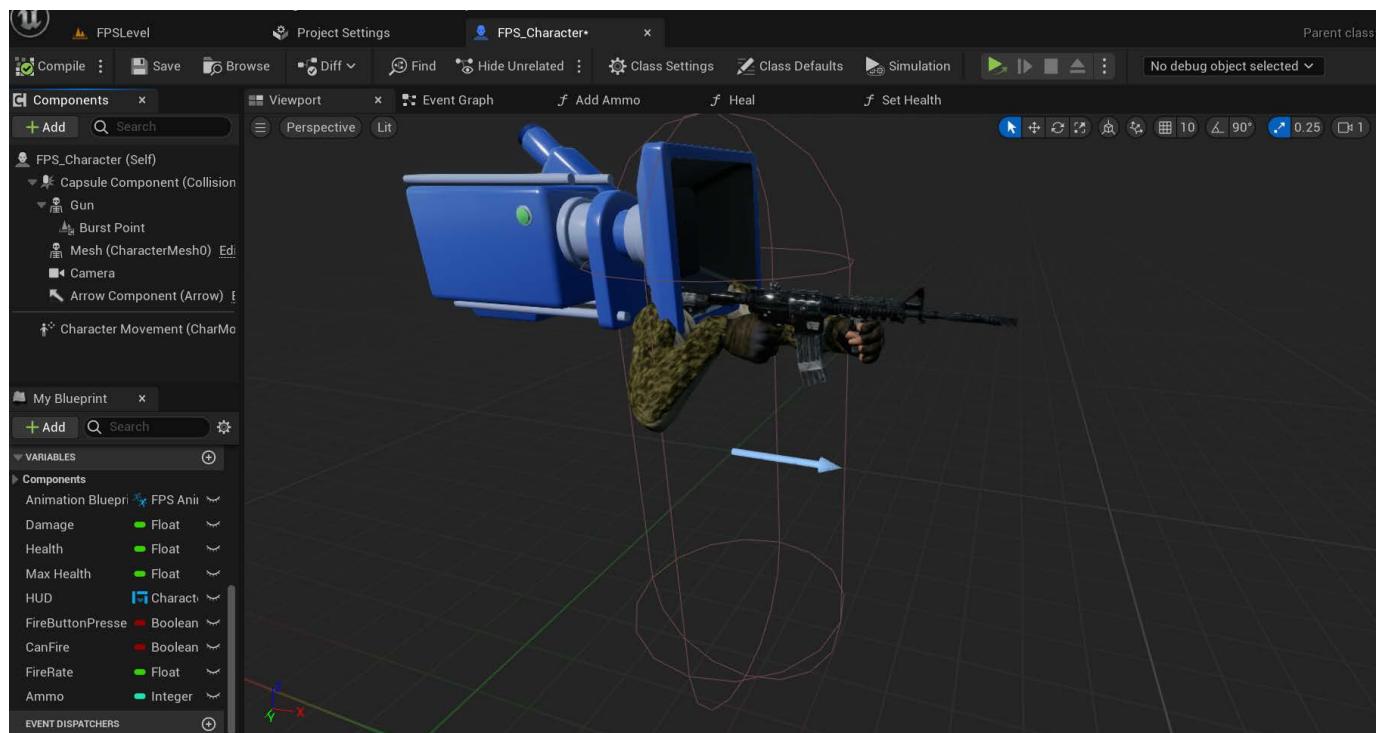


## ◆ Variables :

Below mentioned variables are responsible for the blueprint to compile successfully and it has the ability to interact with the viewport.

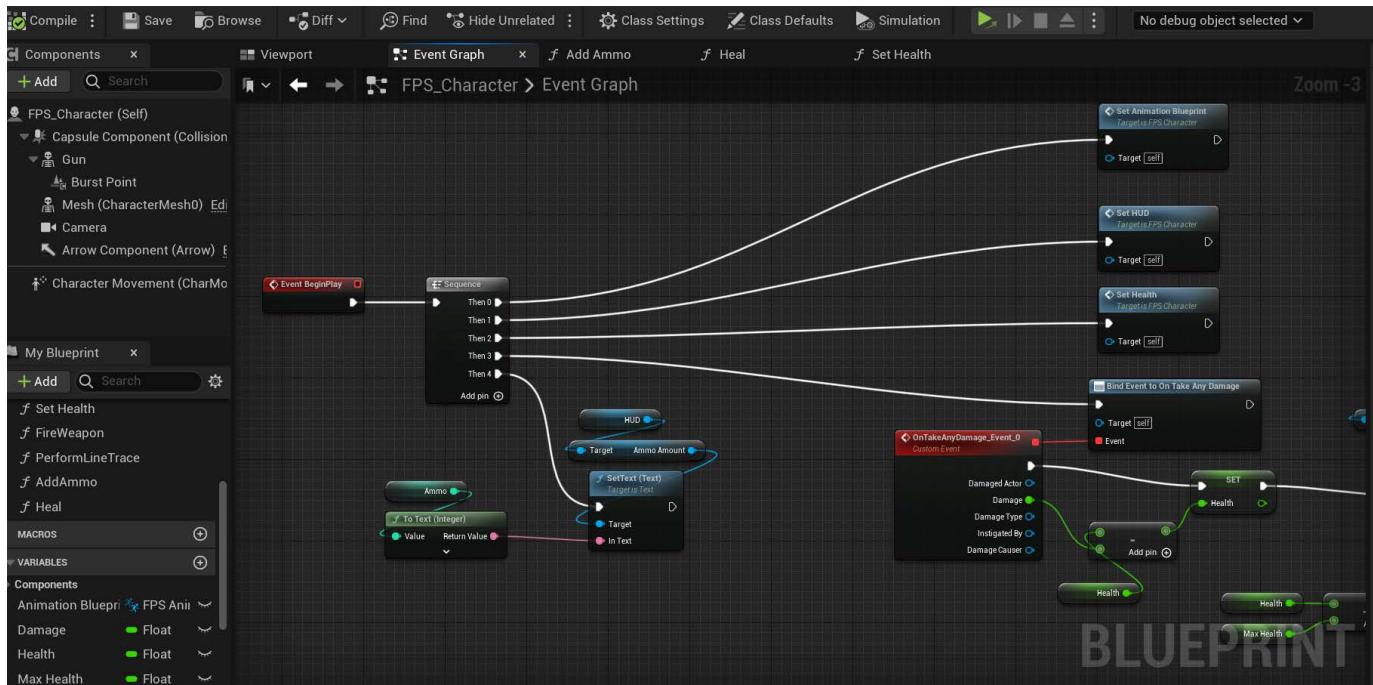


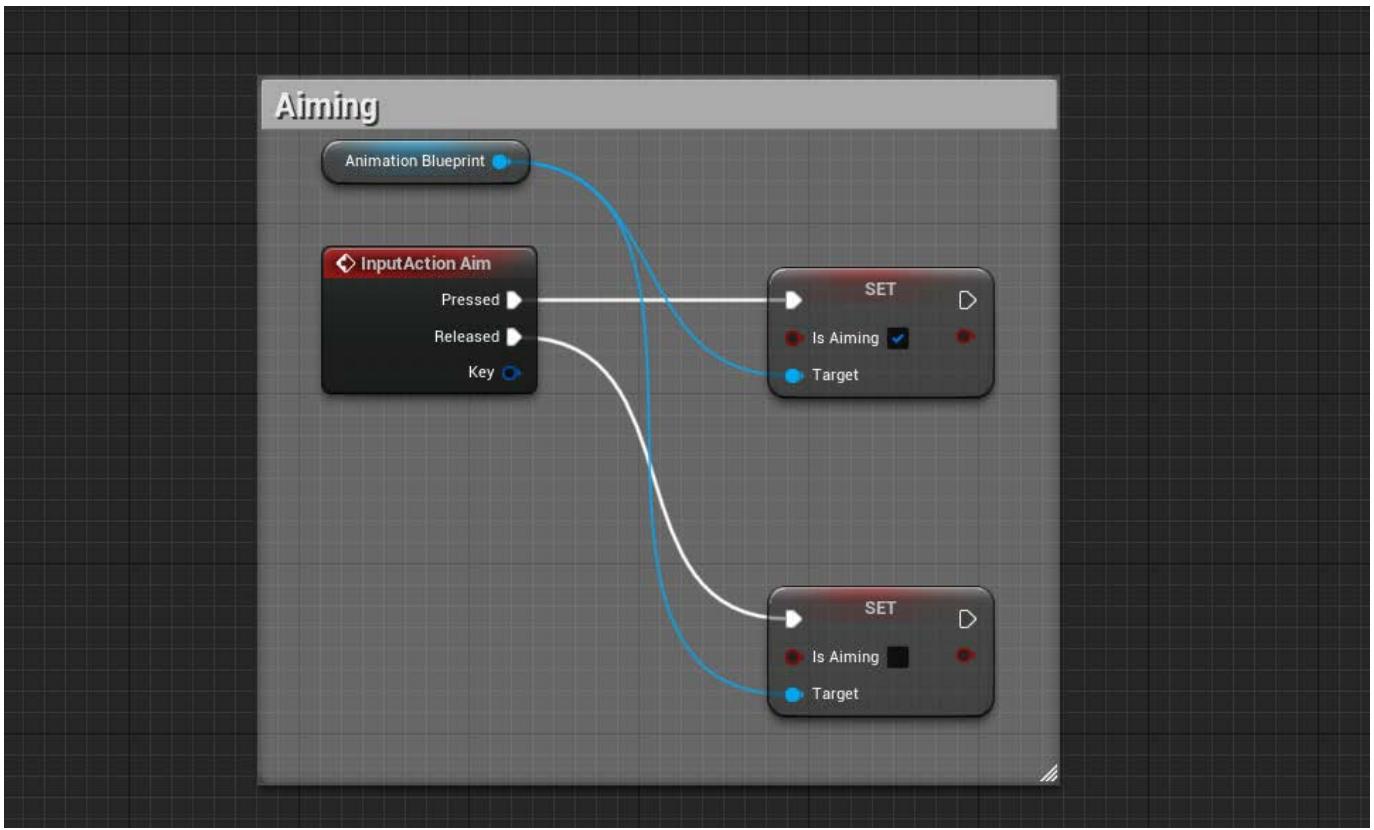
## This is the Fps Character Blueprint Class:



## ❖ Event Graph -

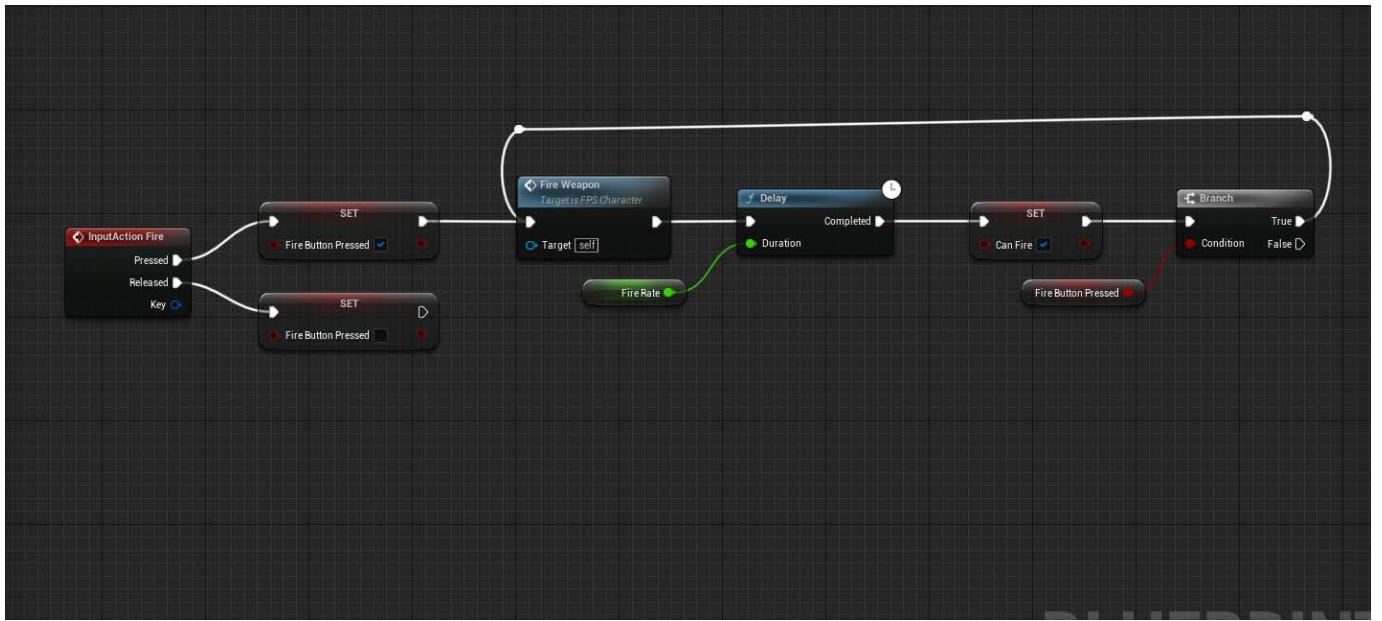
Event Graph can be said like the main hub of any actor which we want to use as a character. The event graph contains of the amount of ammo collected using pickups, no. of health increased using health pickups, the health bar increases when the player collects the health pickup, take damages from drone, and everything as mentioned above.





## ❖ Aiming the gun -

Input action component and isAiming variable is used here



### **Input Action Fire :**

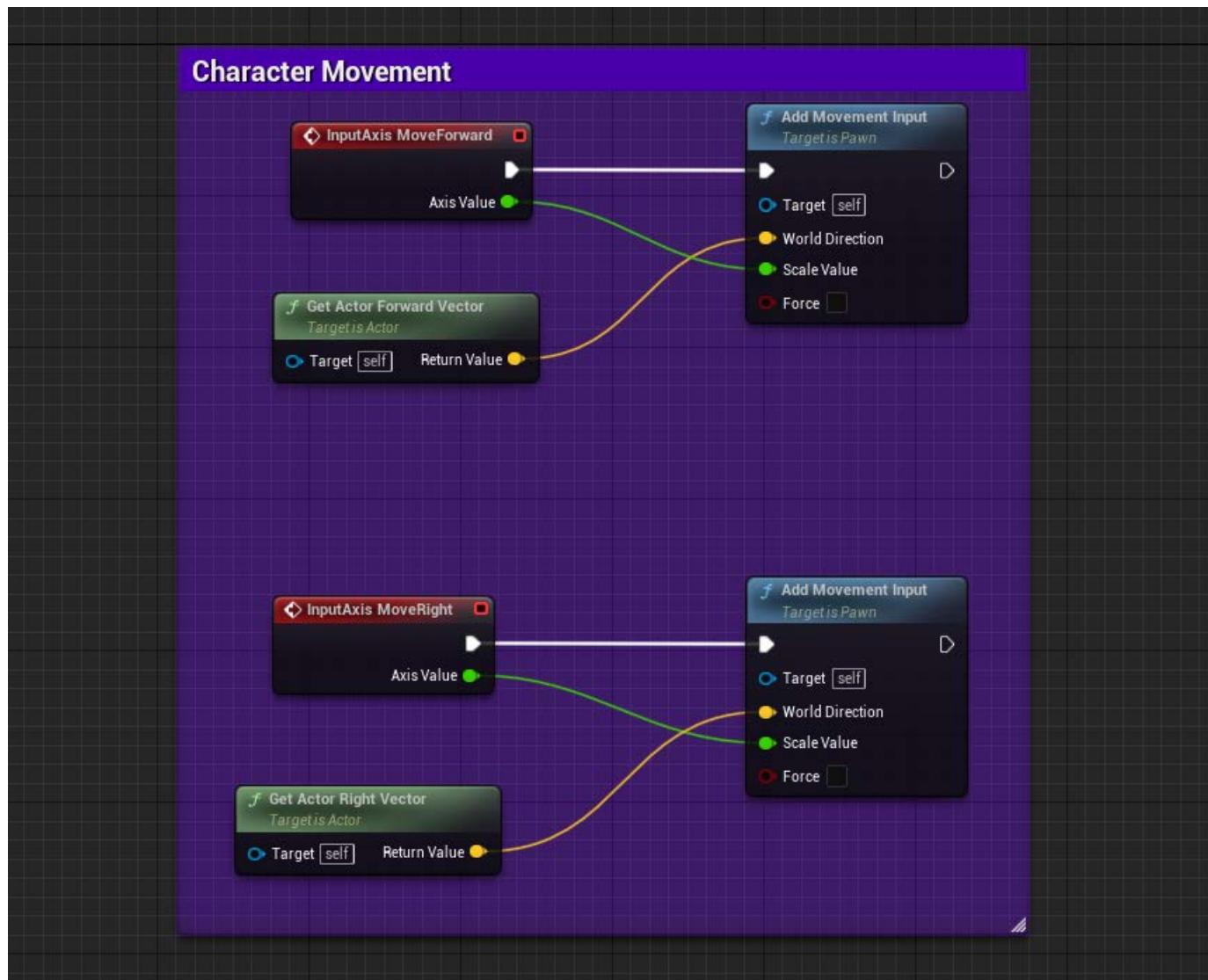
Here Input Action Fire event is used to fire the bullet from the gun.

## ❖ Character Movement -

Here Input Axis Move

Forward event and Get Actor Forward vector has been linked to Add Movement Input to move the character forward and backwards.

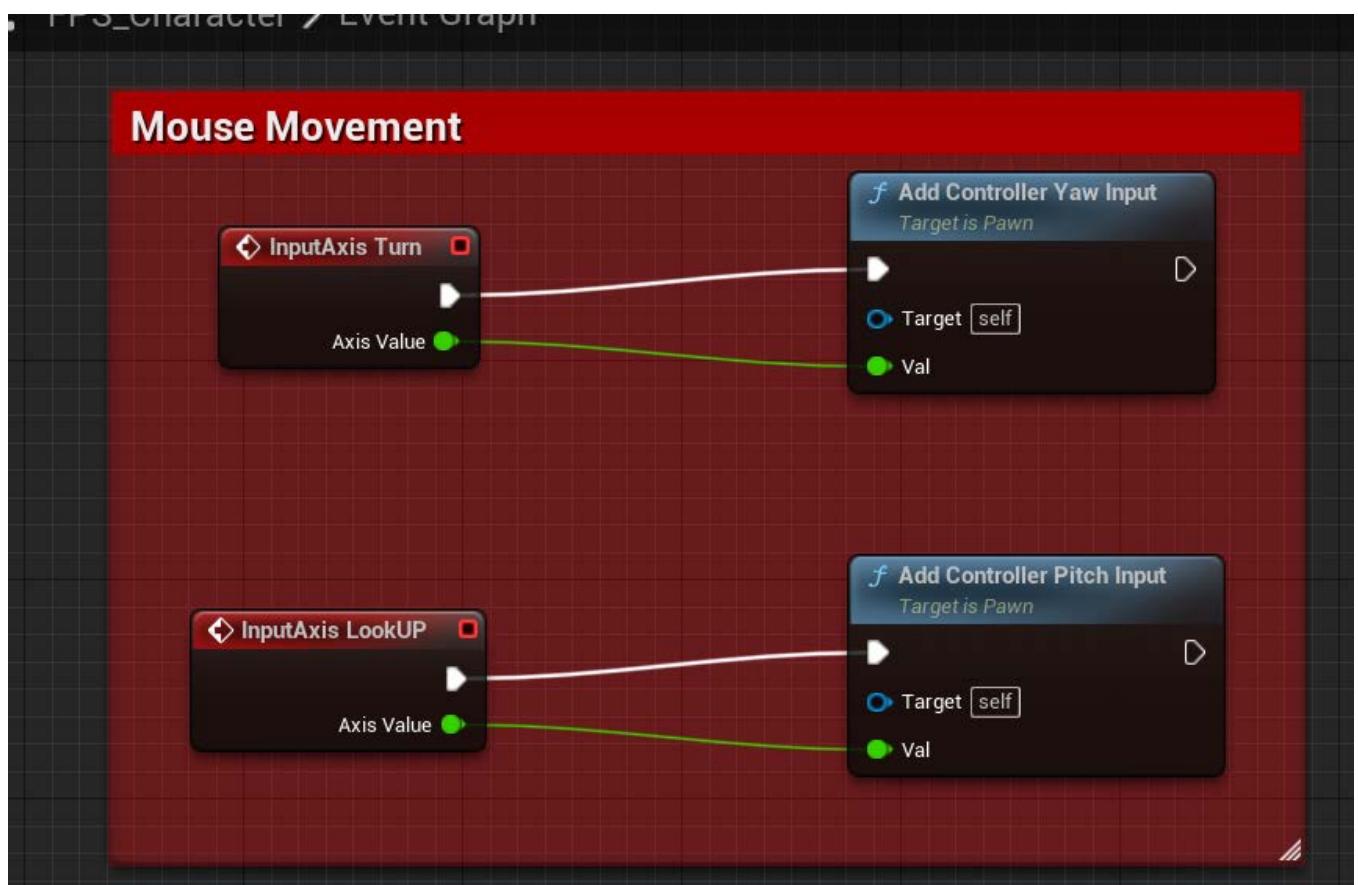
While Axis Move Right event and Get Actor Right Vector has been linked to Add Movement Input to move the character to the right and left.



## Mouse Movement :

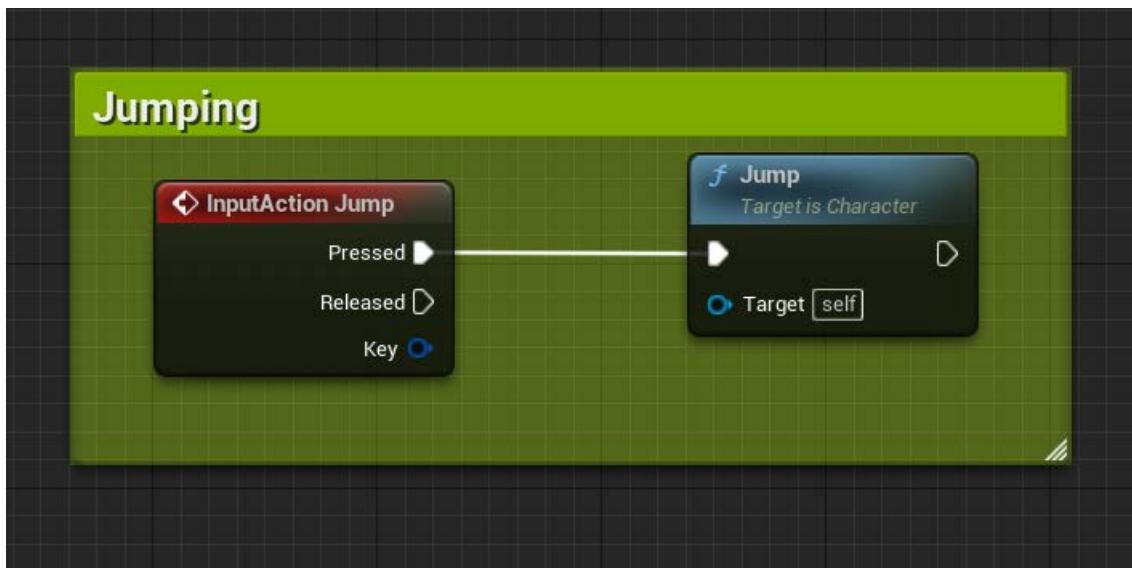
Here Input Axis Turn event has been linked with Add Yaw Input to be able to move the mouse and look left and right.

While Input Axis LookUP event has been linked with Add Controller Pitch Input to be able to move the mouse and look up and down.



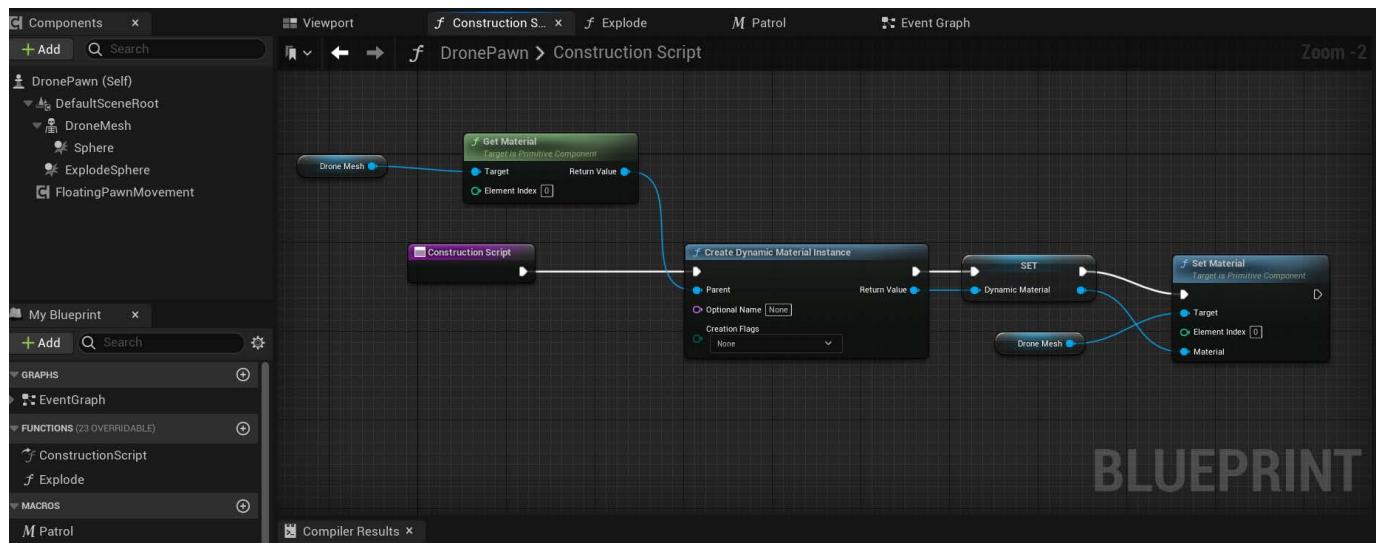
## ◆ Jumping -

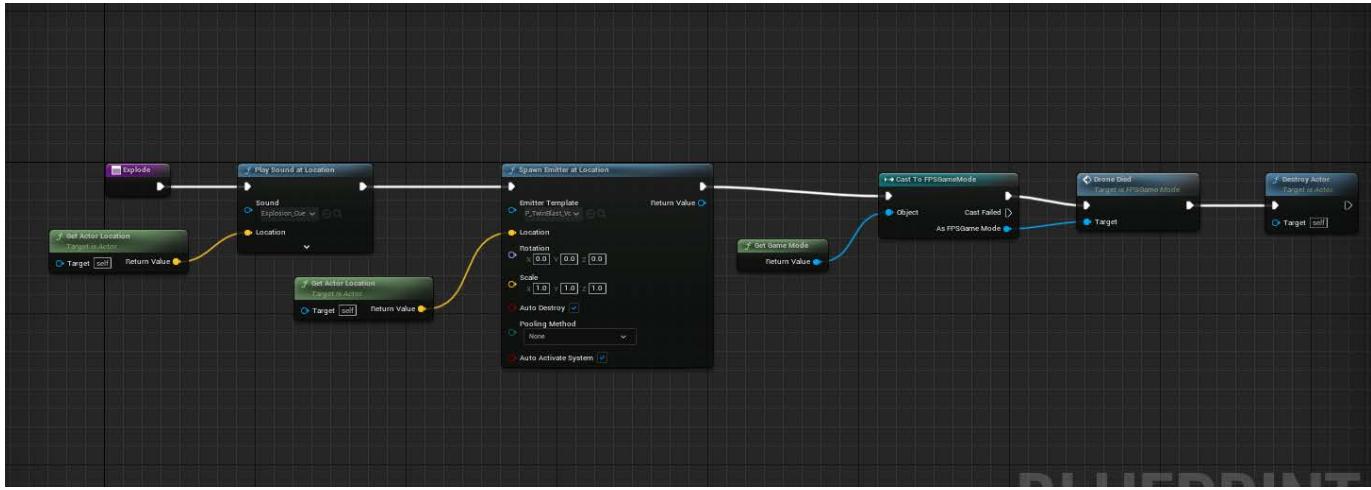
Here Input Action Jump event is used to jump function when pressed so that the player can jump.



## ❖ Drone Pawn (Construction Script) -

In this blueprint inside the construction script we get material component, create dynamic material instance , set material being used to apply the material on the drone.



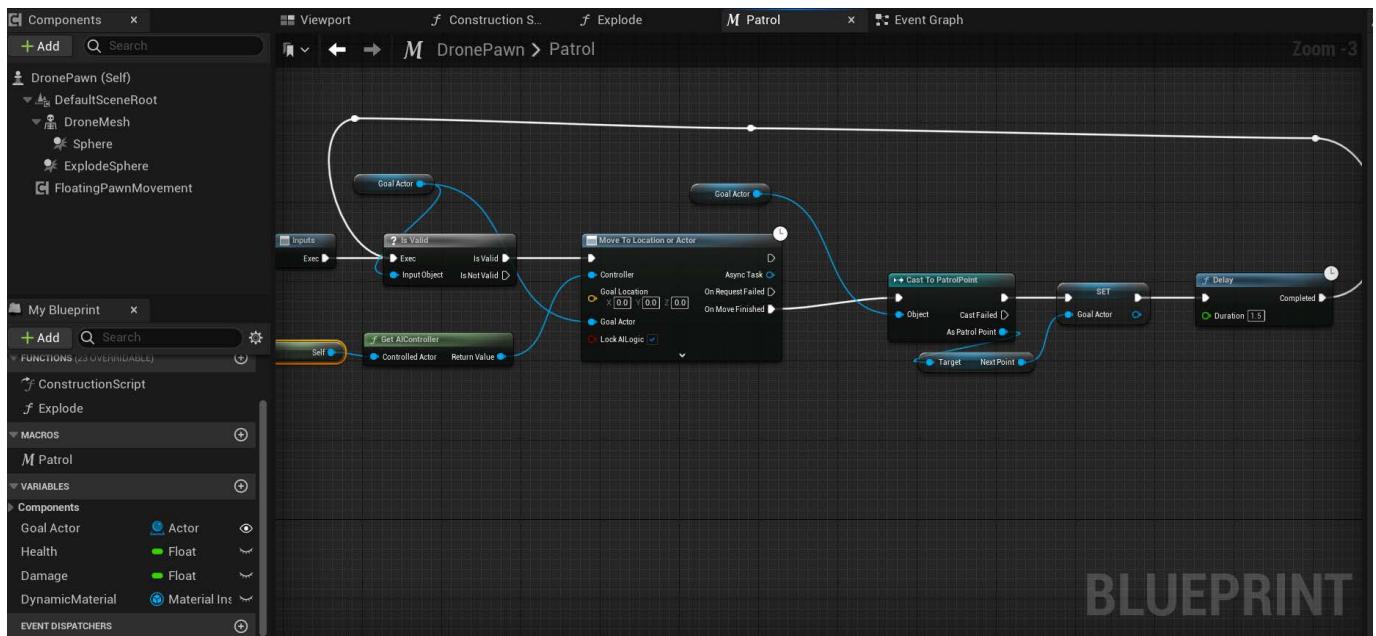


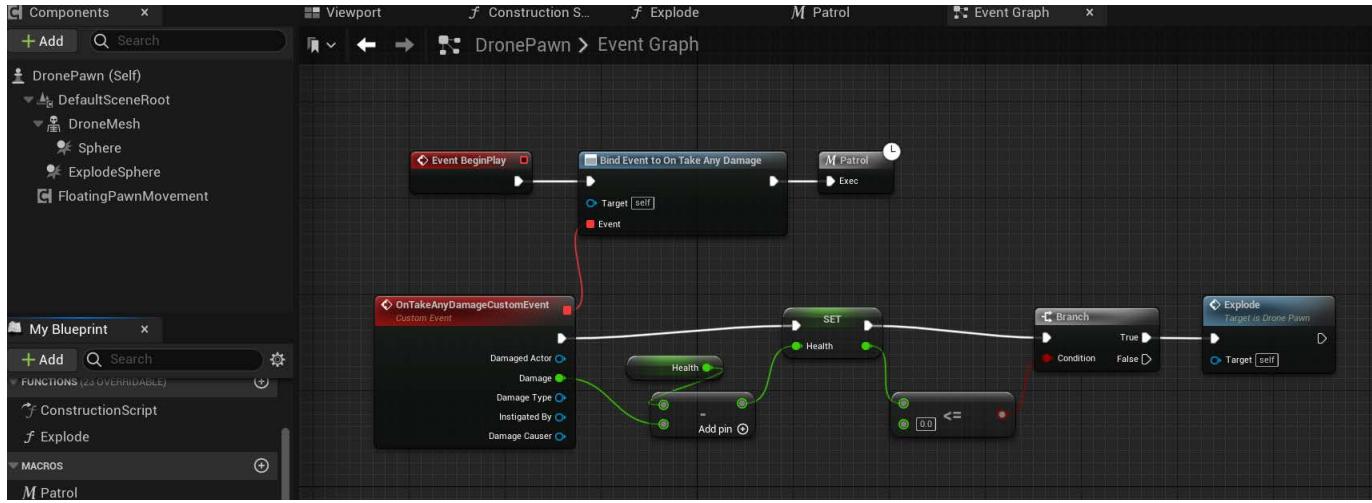
### Explode(Function):

In this function we have used the Play Sound at Location, Spawn Emitter at Location, Destroy Actor,etc. In this function the drone pawn will explode when it gets fired at and will play explosion sound, then the drone will get destroyed.

## ◆ Patrol -

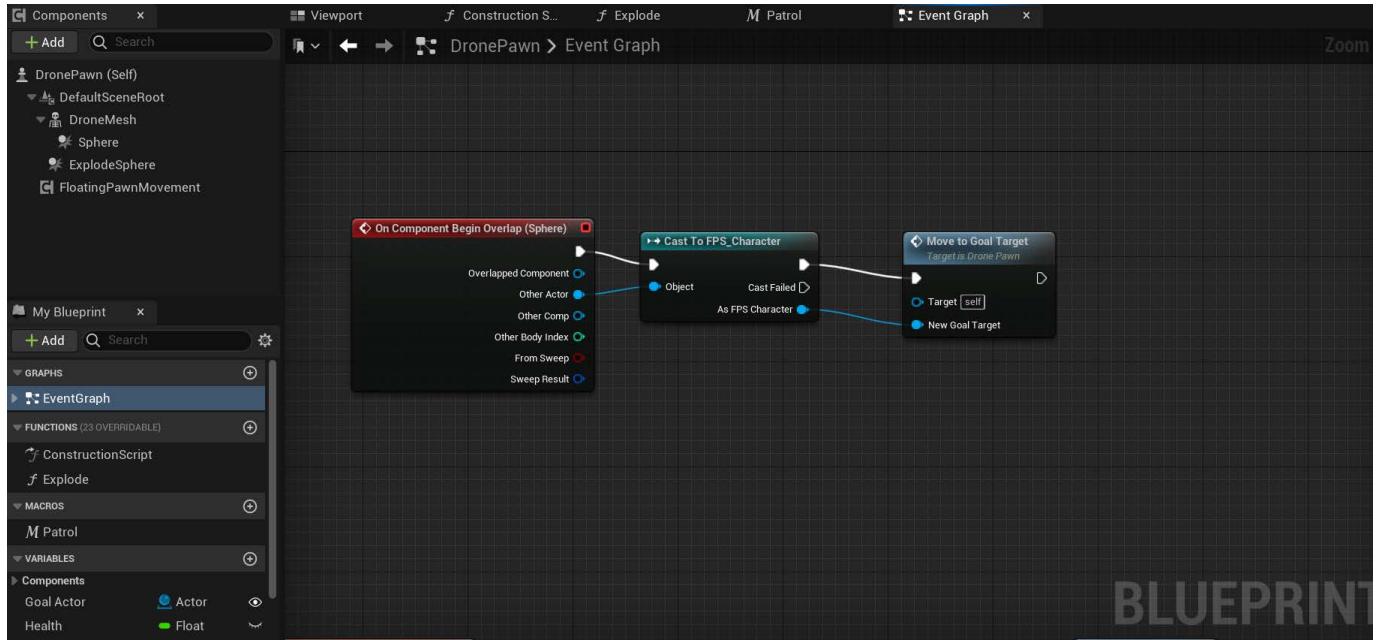
This is one of the main features of this project. The components used in this Macro are Move to Location Actor, Cast to Patrol Point, Get AI Controller. The drones patrol from one point to another point seamlessly.





## Event Graph :

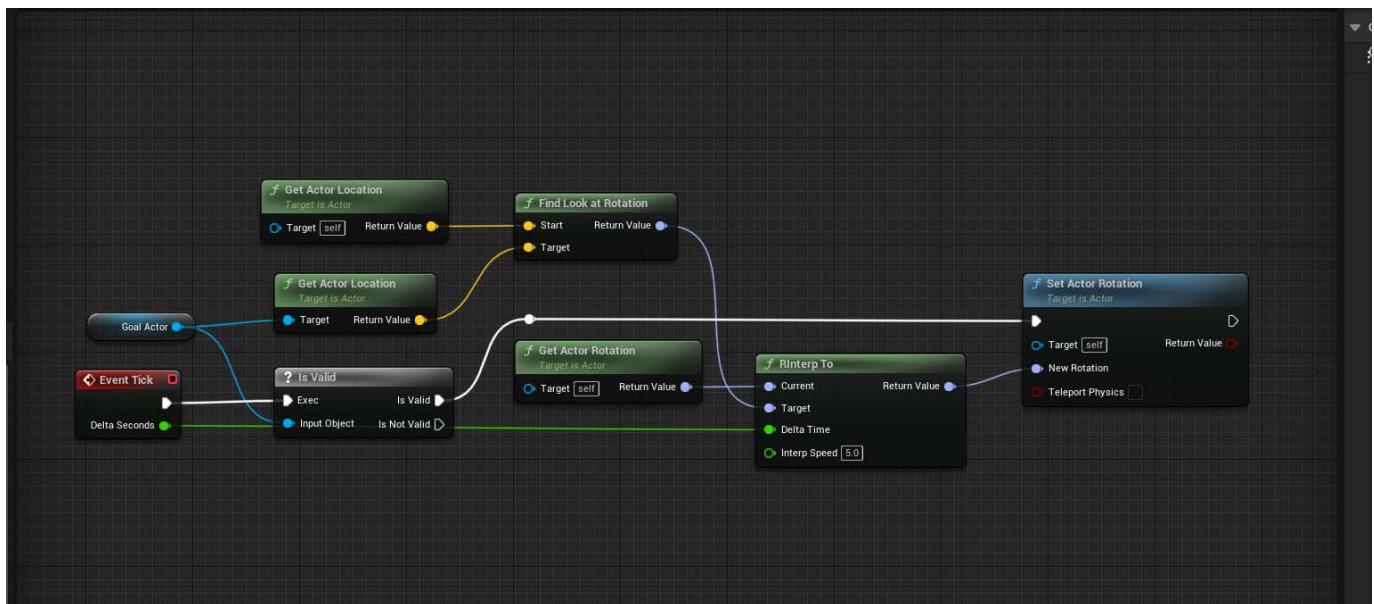
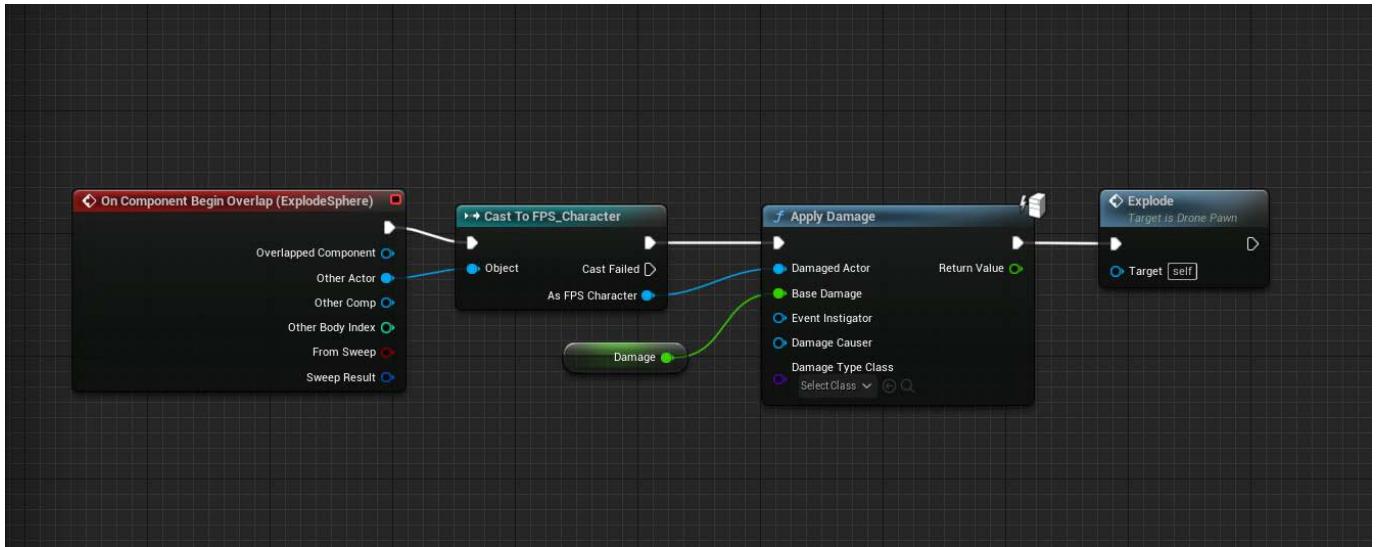
In the Event Graph we have used the components Bind Event to On Take Any Damage, OnTakeAnyDamageCustomEvent. The image mentioned above states that the drone can take the damage and if the damage reaches a certain point the drone explodes.



BLUEPRINT

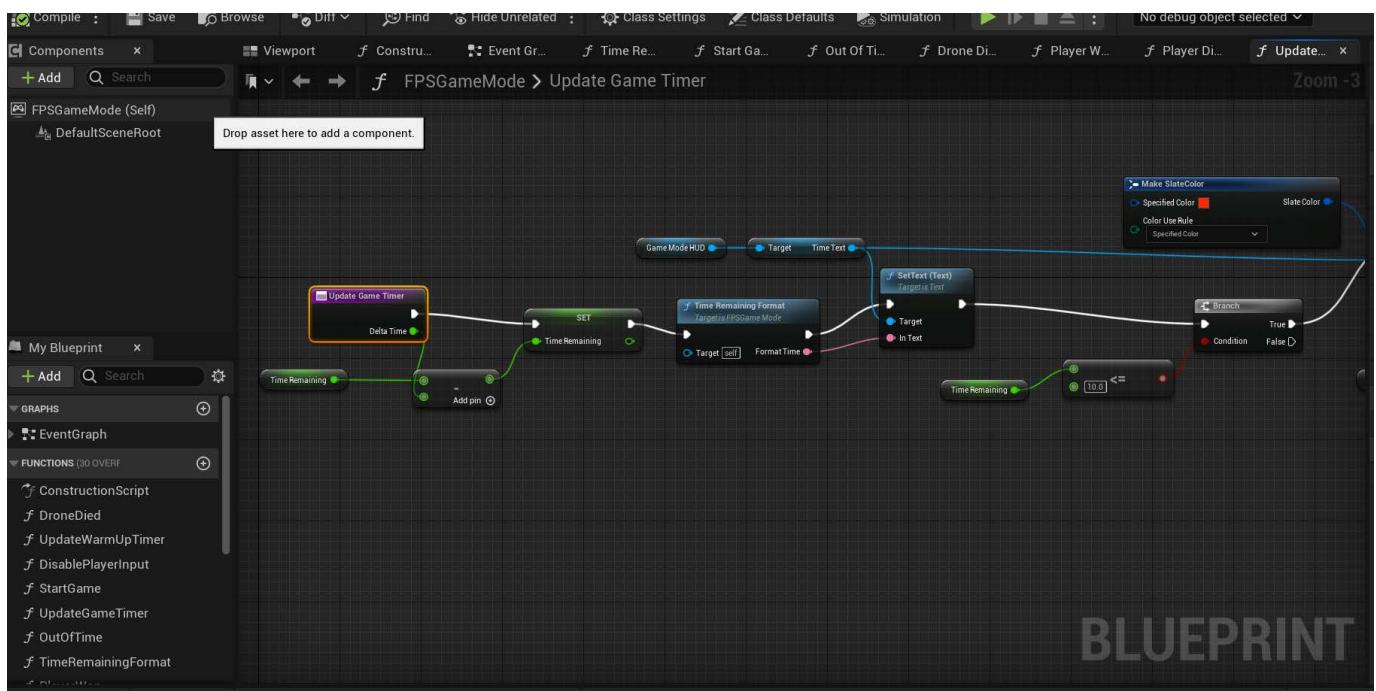
## Change the Drone Color when fired a bullet -

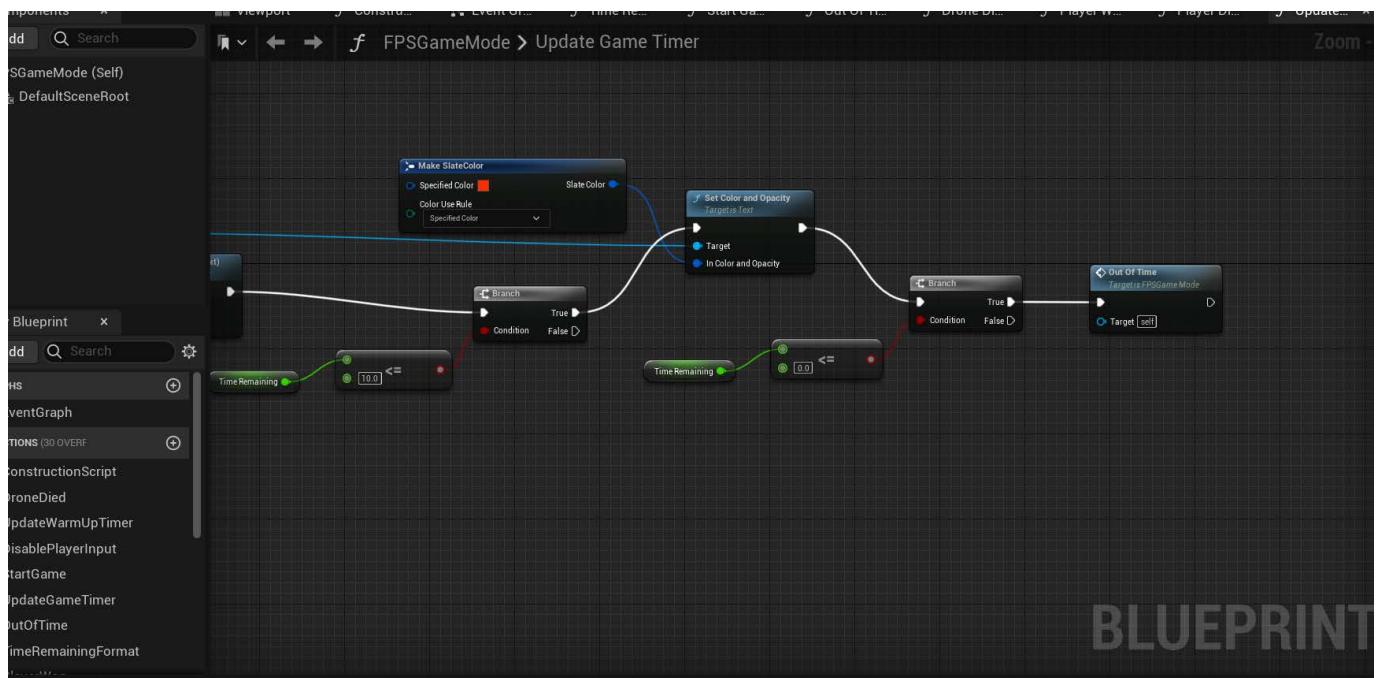
In the above mentioned ss the components are AI Move To, Move to Goal Target event, Set Vector Parameter Value. These components will change the color of the drone when we hit a bullet at them.



## ❖ FPS Game Mode Base -

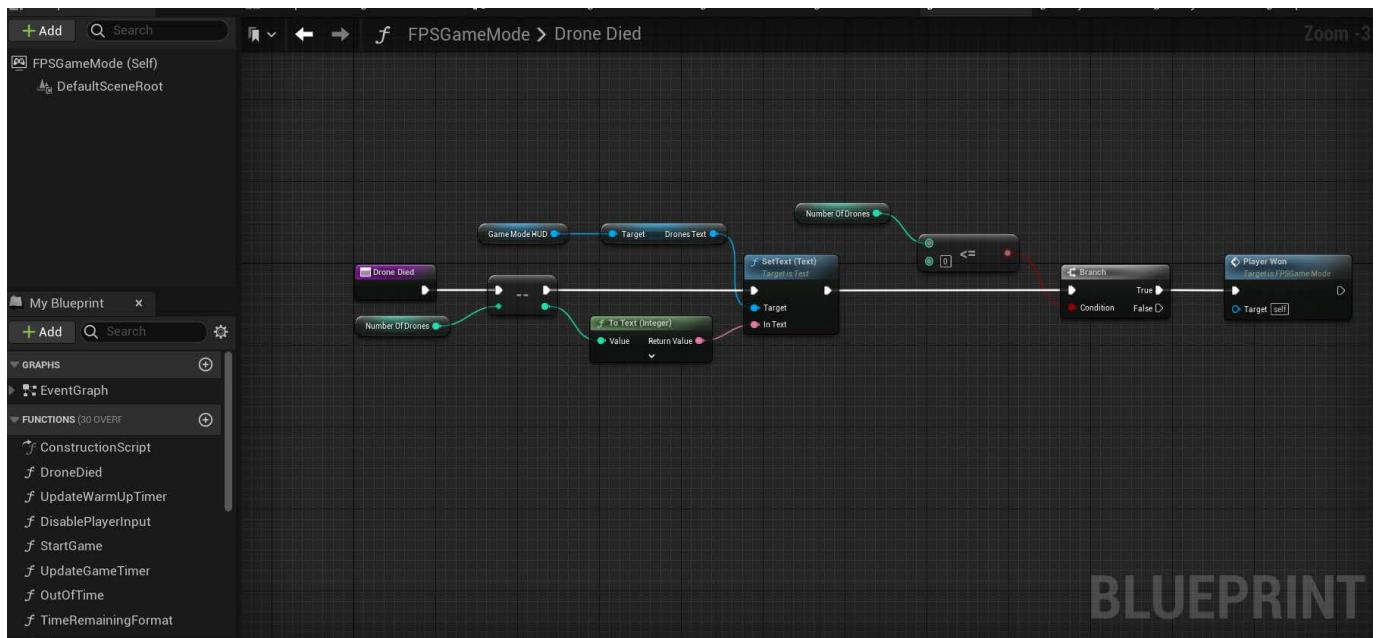
In this event graph we have set a text where the text denotes the time remaining to finish the level and when the time goes below 10 sec the text changes to red color.





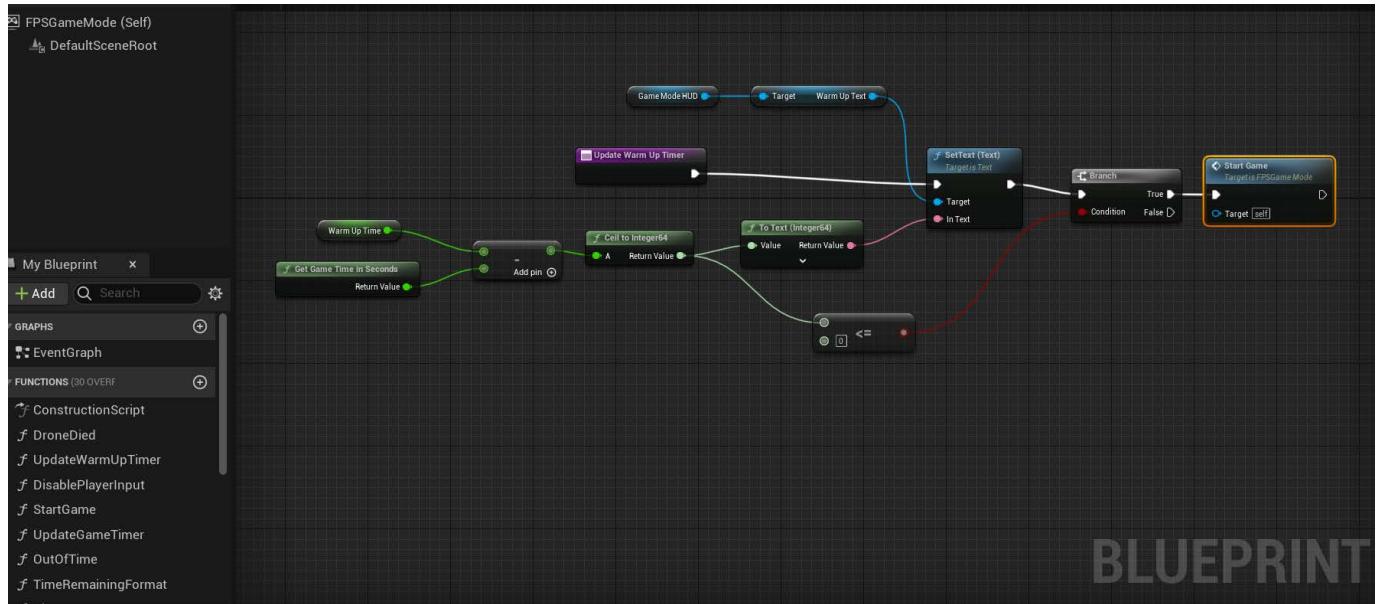
## ❖ Drone Died -

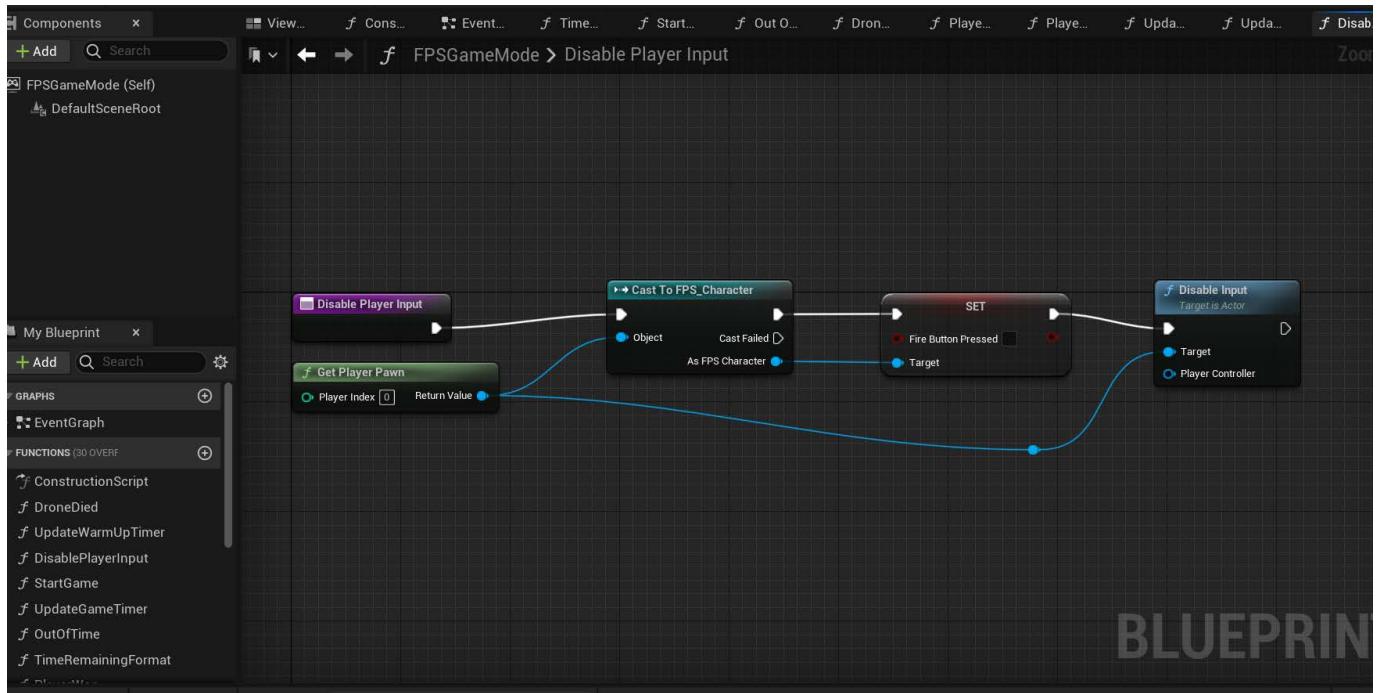
In this function the game starts and when the players kills all the drones the game ends and the text is displayed You Win!.



## ❖ UpdateWarmUpTimer -

In this Function before starting the game we get a 3 second delay which says Clear Zone and gets removed after 3 seconds.





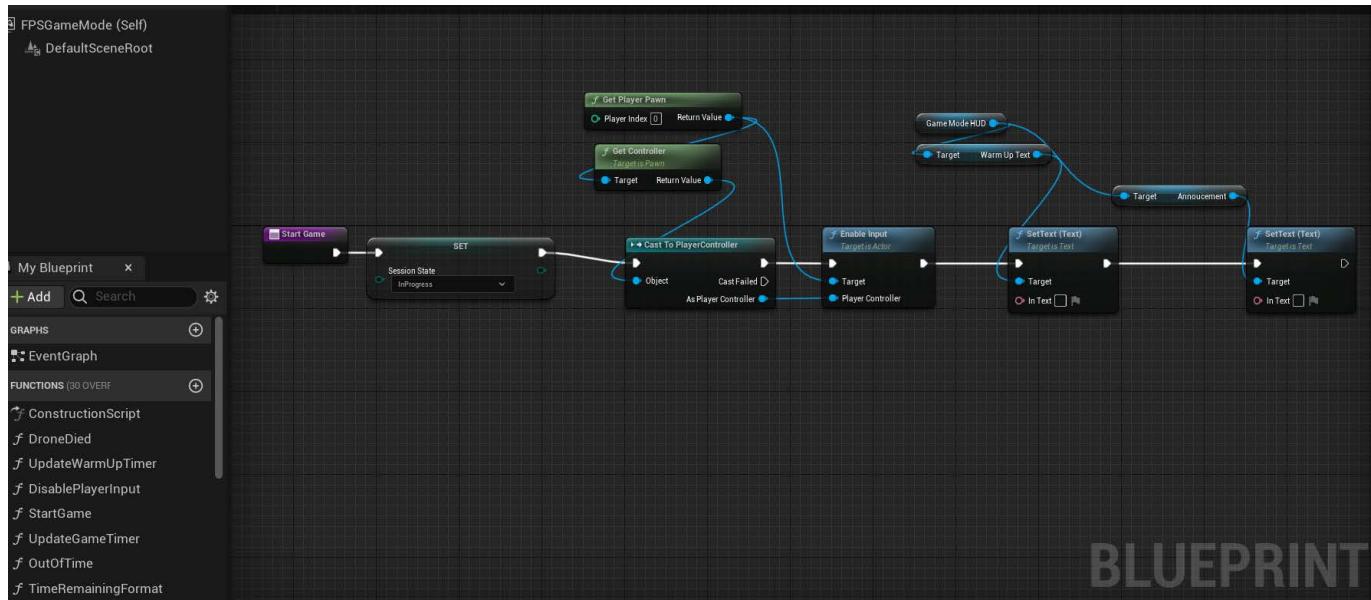
BLUEPRINT

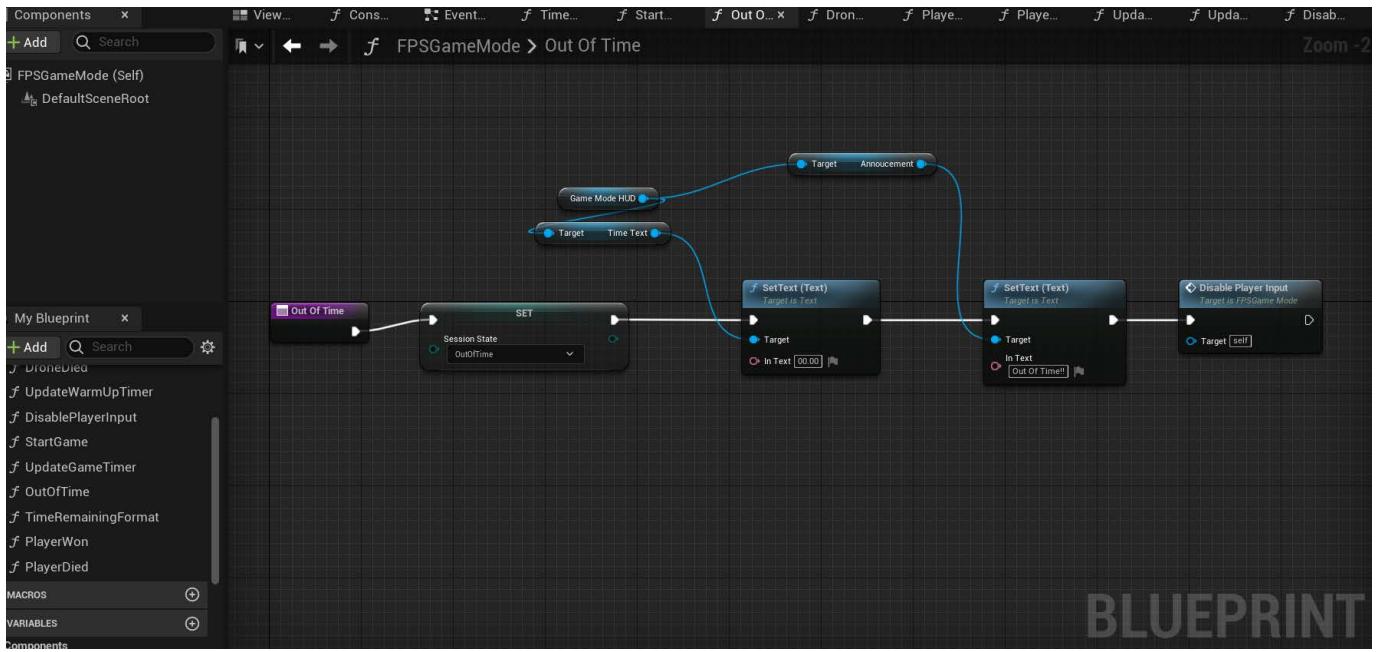
## ❖ Disable Player Input -

In this function the Input from the player is disabled for the warmup screen and when the game ends.

## Start Game :

This function is used to start the game.





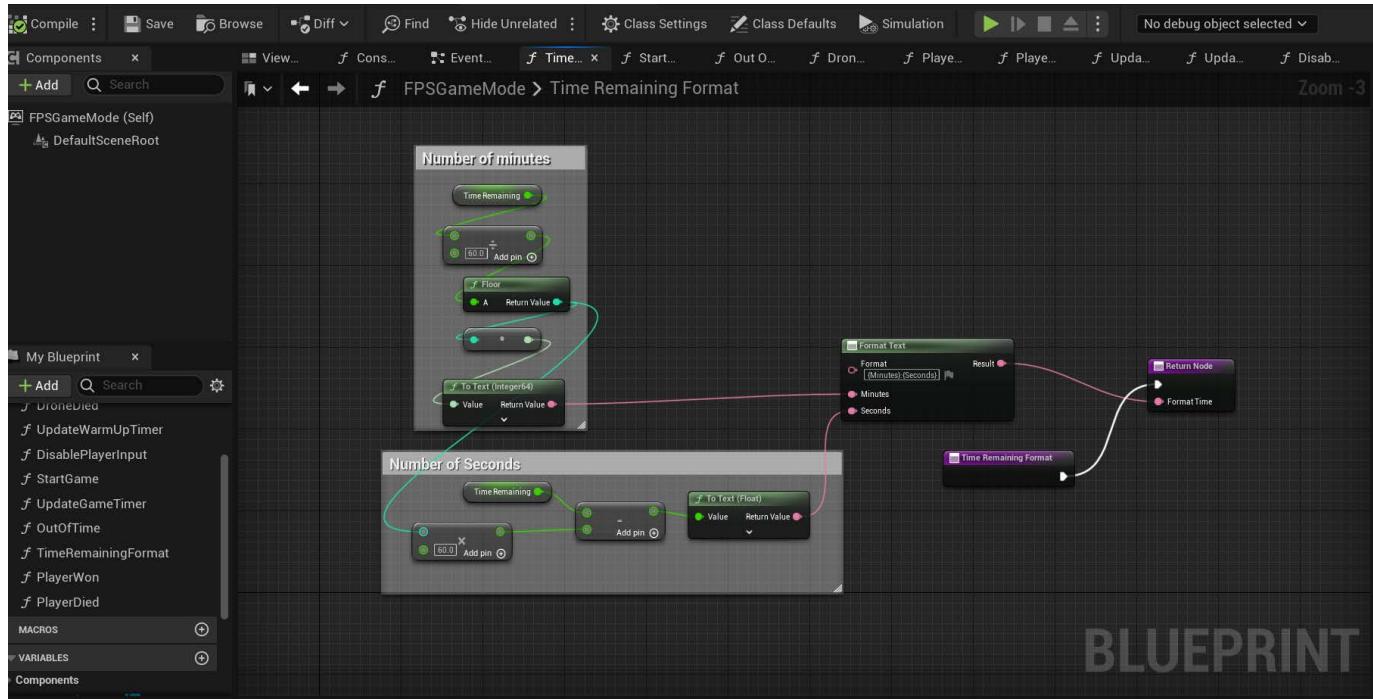
**BLUEPRINT**

## Out of time :

This Function is used to display the message that the player has ran out of time and wont be able to play the game.

## ◆ Time Remaining Format:

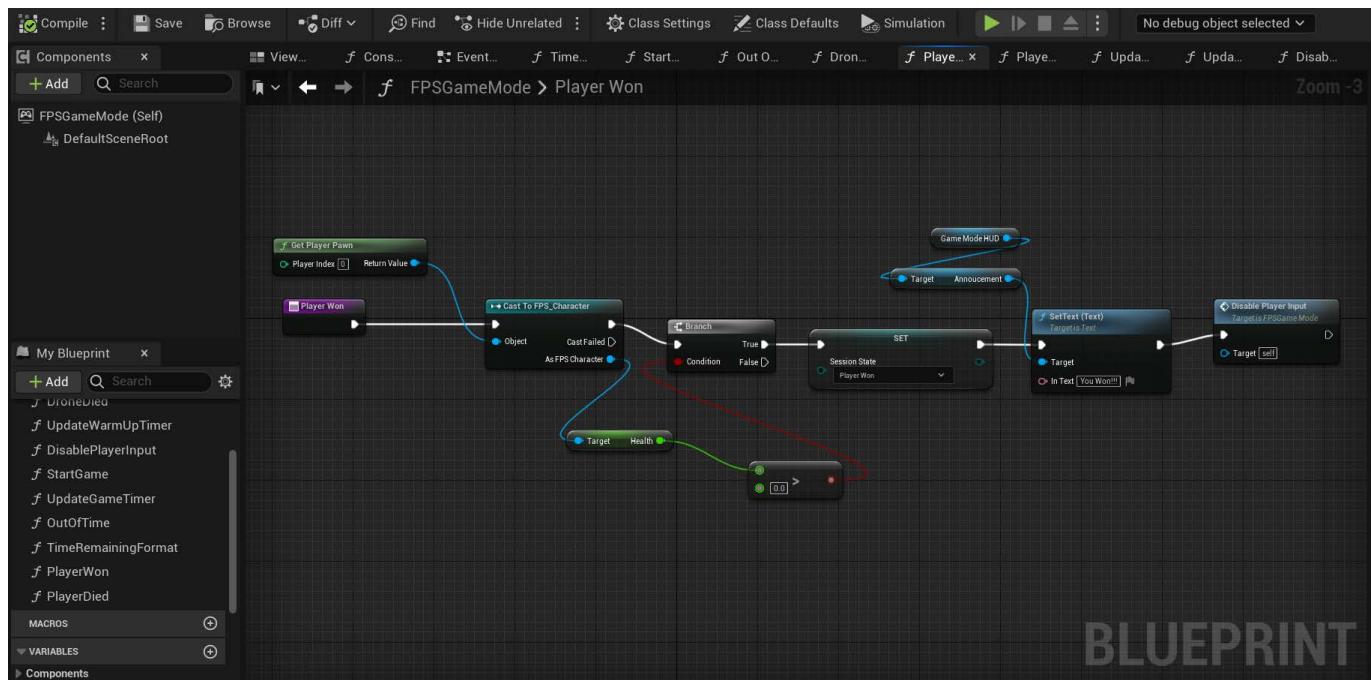
This function is used to display the time remaining to destroy the drones and win the format used is in minutes and seconds.



BLUEPRINT

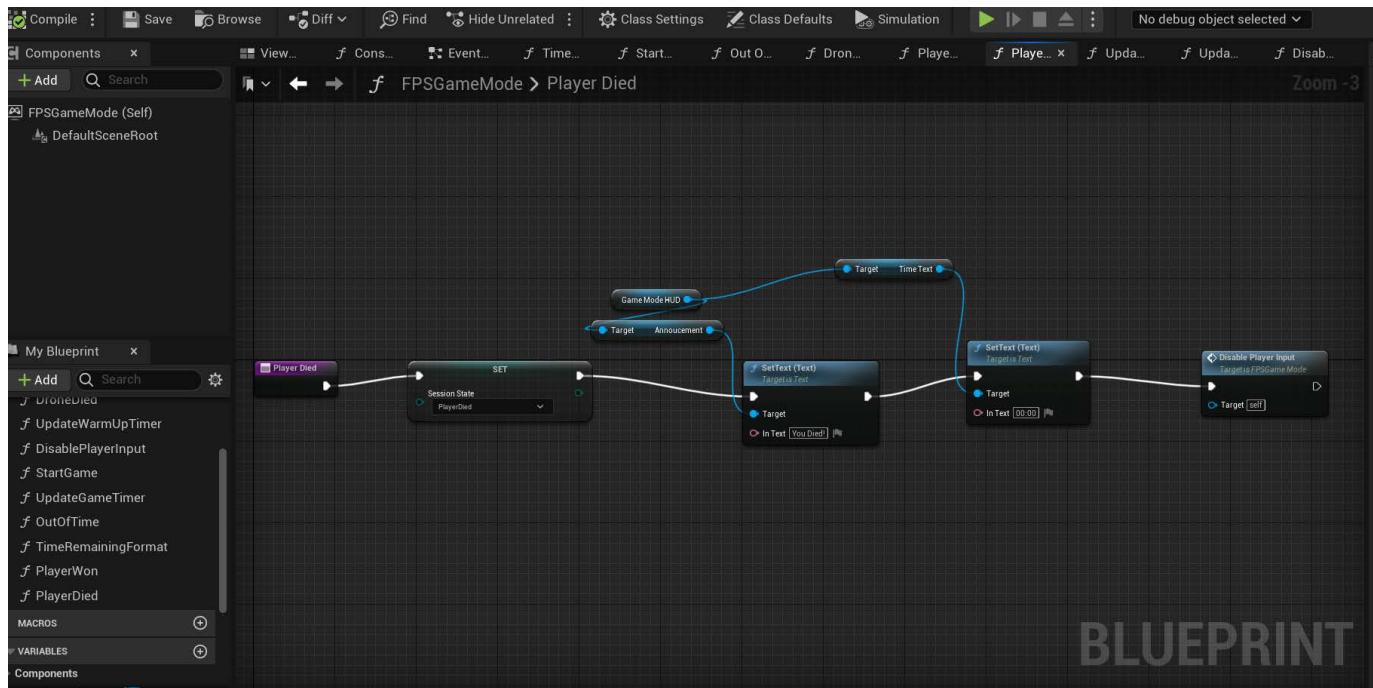
## ◆ PlayerWon -

This function is called when the player kills all the drone. You Win text is displayed .



## Player Died:

This function is called when the player dies.



---

## **Post-Production**

### **4.1 Testing**

Game Testing is a phase of development where the game is repeatedly played with various approaches with the intention of finding and reporting bugs and glitches.

*Killer Drones* was deployed and tested several times by friends. A couple of bugs were found which was instantly fixed.

### **4.2 Optimization**

---

Optimization is a crucial part of game development as this directly affects the performance of the game. If the optimization is poor, the game cannot end on a low end device thus reducing the market share and giving a poor gaming experience.

---

## **Conclusion and Developer comment**

*Killer Drones* is a first person shooter mainly famous for its aggressive drone combat system.

I had a lot of fun developing this game and the game is to mostly demonstrate my programming skills. This game consists of some unique combinations like blueprints. It was very difficult to manage both internship and project. But in the end, I overcame the challenges that I faced and completed the game.

Developing this game while working as an intern in All Star Games Deftouch Interactive Art Pvt. Ltd. was like a quest for me. As time management was the key element to complete my day to day tasks that I assigned to myself to complete the project on time. After completing my project I learned two important things, first was, time management was important and the second was, being confident. It was an inspiring quest as I feel more confident about developing more complex and challenging games.

---

## **Reference and Bibliography**

Unreal Engine API reference : <https://docs.unrealengine.com/5.0/en-US/>

Youtube, Udemy, Google and Unreal Forum.