C# breakdown of scripts:

Key Classes

[AI Script: 2](#_Toc24923090)

[AI\_Movement: 2](#_Toc24923091)

[AI\_Behaviours: 2](#_Toc24923092)

[AI\_Hearing\_Detection: 3](#_Toc24923093)

[AI\_Shooting: 3](#_Toc24923094)

[AI\_Structure 4](#_Toc24923095)

[Player Controller: 5](#_Toc24923096)

[Player Movement: 5](#_Toc24923097)

[Player Shooting: 5](#_Toc24923098)

[Player Structure 6](#_Toc24923099)

[Environment Scripts 7](#_Toc24923100)

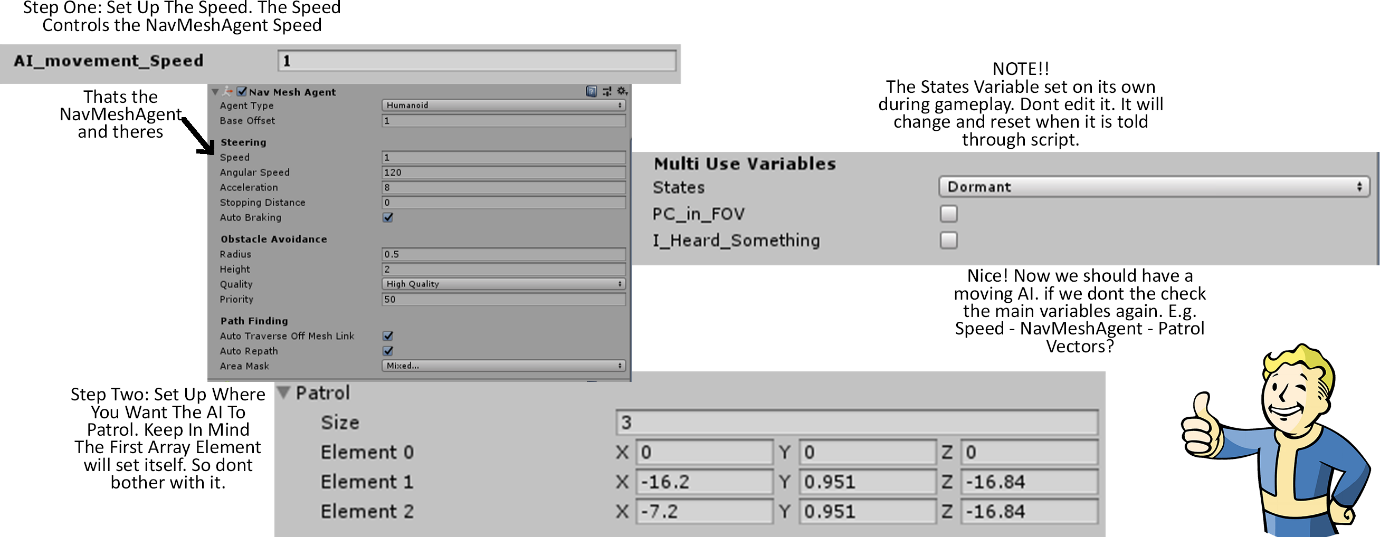
[Pulse Light: 7](#_Toc24923101)

# AI Script:

How the script works:

## AI\_Movement:

The AI can move in 3 separate ways – Patrolling / Searching for the player and chasing the player. The main movement we can manipulate is patrolling. There is a Array of Vector3 which tells the object to move towards.



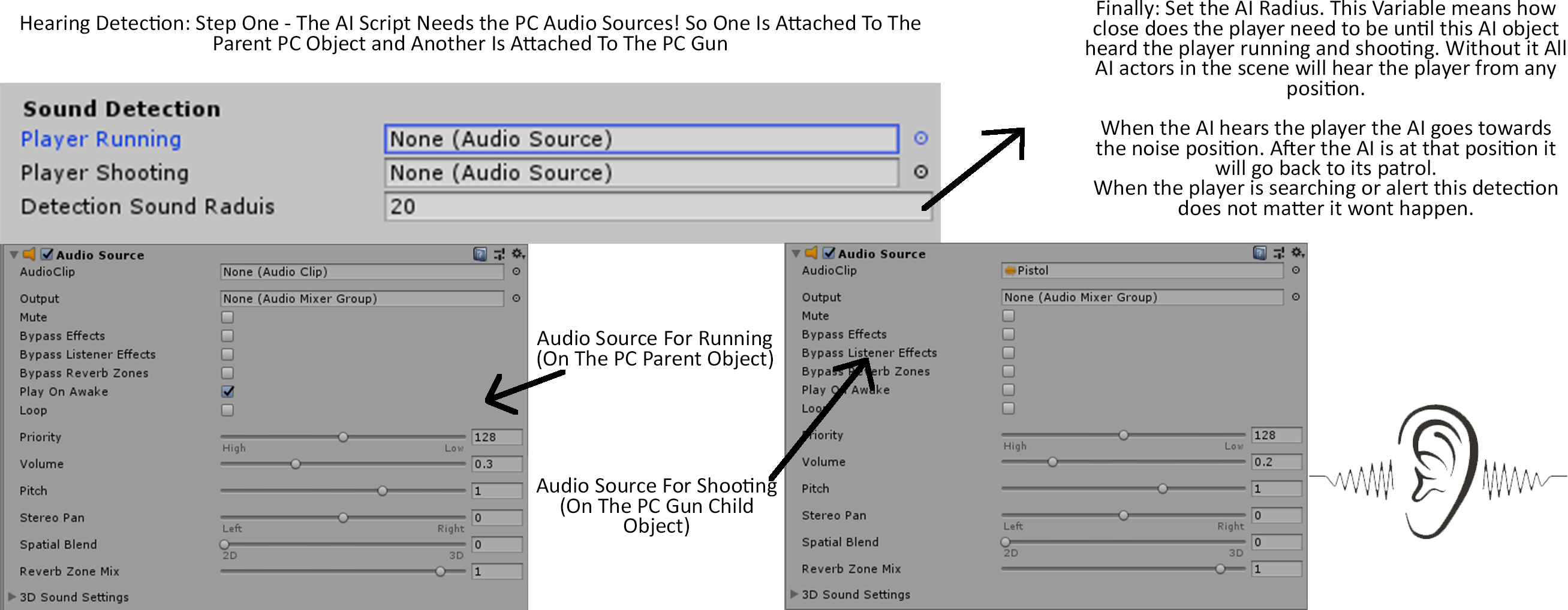
## AI\_Behaviours:

The Behaviour Logic in this script also known as Game States or Enum controls the behaviour of the AI. If it’s Dormant we are patrolling. If we are searching then we are trying to find the player and if we are Alert well then AI can start action with the player. The behaviours change depending on howlong the player is in sight (In The FOV “Field Of View”).



### AI\_Hearing\_Detection:

The Hearing detection allows the AI detect the player’s sound presence depending on a distance. If the player is close enough to the AI and runs or shoots the gun then the AI goes towards the player.

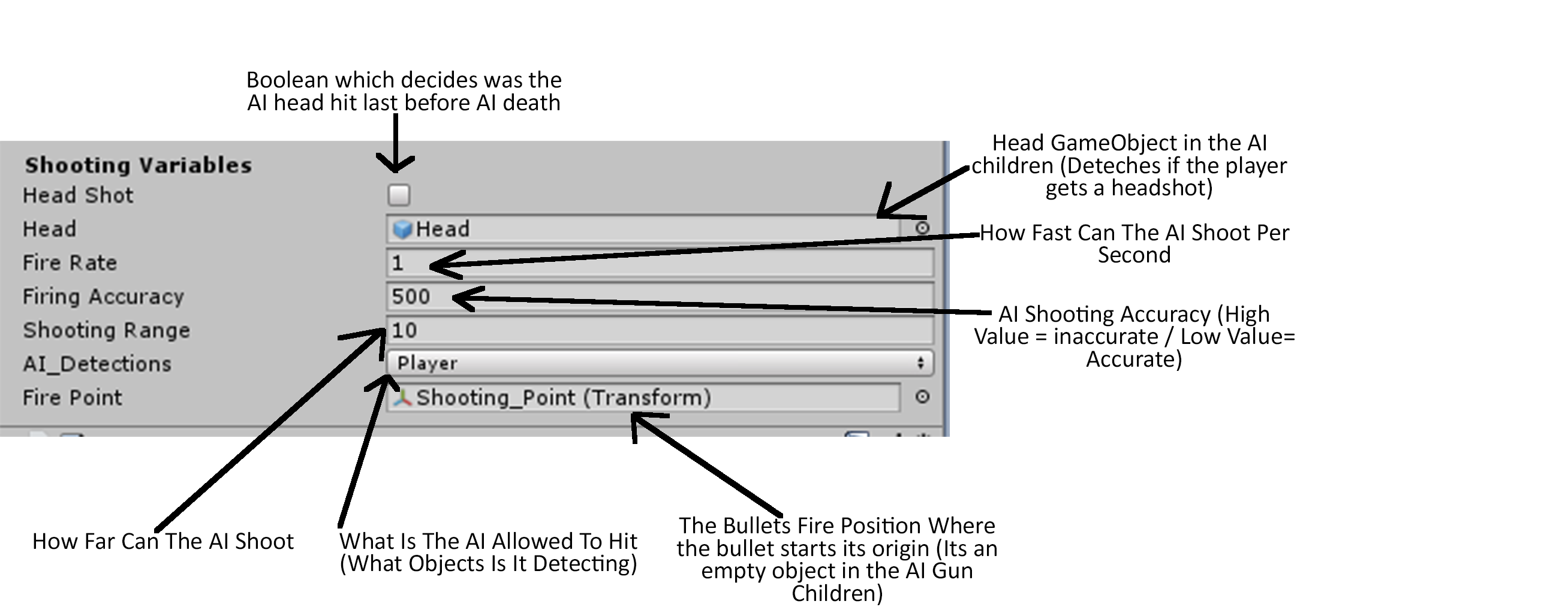


## AI\_Shooting:

The AI shooting logic is executed when the Searching and Alert states are active.

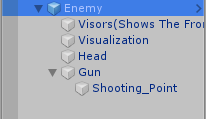
Searching allows for less accurate shots giving the implementation of AI fear and random shooting whereas

Alert Shooting is more accurate and more aimed towards the player.



## AI\_Structure

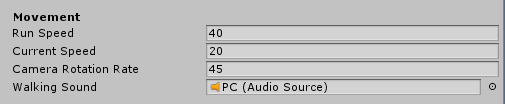
The AI Structure is just how the Object prefab is set up in the hierarchy. The Visors are just a indeicator on where the simple object is facing. (This is the simple concept breakdown). This will change when there are 3D models to use with animations.



# Player Controller:

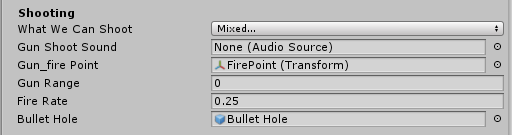
The Player Controller is the script which holds all the relevant logic for our characters actions (“Moving, Shooting, Camera Movement etc”).

## Player Movement:



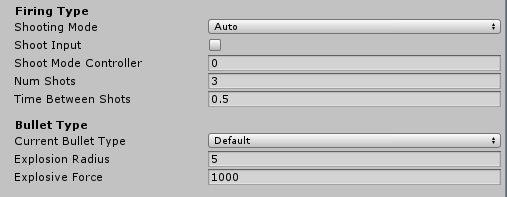
For Player movement the player can move with a set speed for Current Speed. However when holding down “Shift” The PC will run this value can also be set ad changed to whatever speed you wish to go. The camera rotation is a value which allows you to change how sensitive the player camera movement is. The camera moves with mouse movement on the X and Y. So (Up, Down, Left and right). The Up and Down rotation is clamped so the player can only rotate so high and low.

## Player Shooting:



This is the shooting logic at its core:

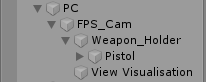
There is a layer mask “What We Can Shoot” which can be filled with layers so we know what the player can shoot. Walls or ground AI actors etc. There is a sound effect which player’s every time that the player shoots with the left mouse button and can aim with the right mouse button. The Ray that gets shot starts at the Fire point which is childed to the gun object which you can see in the object structure. The Gun Range allows the bullet to be shot from whatever value is given and the fire rate is how fast we can shoot a second. The Bullet whole prefab is just something that spawns at the end of the raycast.



The Players have the power to change Firing modes on the gun as well as bullet types. Currently we have shooting modes {Auto (Fully automatic like a machine gun) Semi (Simple one bullet at a time) Burst/Fly (3+ Bullets fire all at once)}. To Do This Press Q on the keyboard to change firing types and to change bullet types press B. We have default and explosive bullet types.

## Player Structure

The Player Prefab holds 2 key and important scripts which make it function: Firstly there is the Player controller. Which controls the player movement and camera functionality this will always be on the parent. However the Pistol holds all the shooting logic. Which allows players to shoot what is detected change fire modes and bullet types.

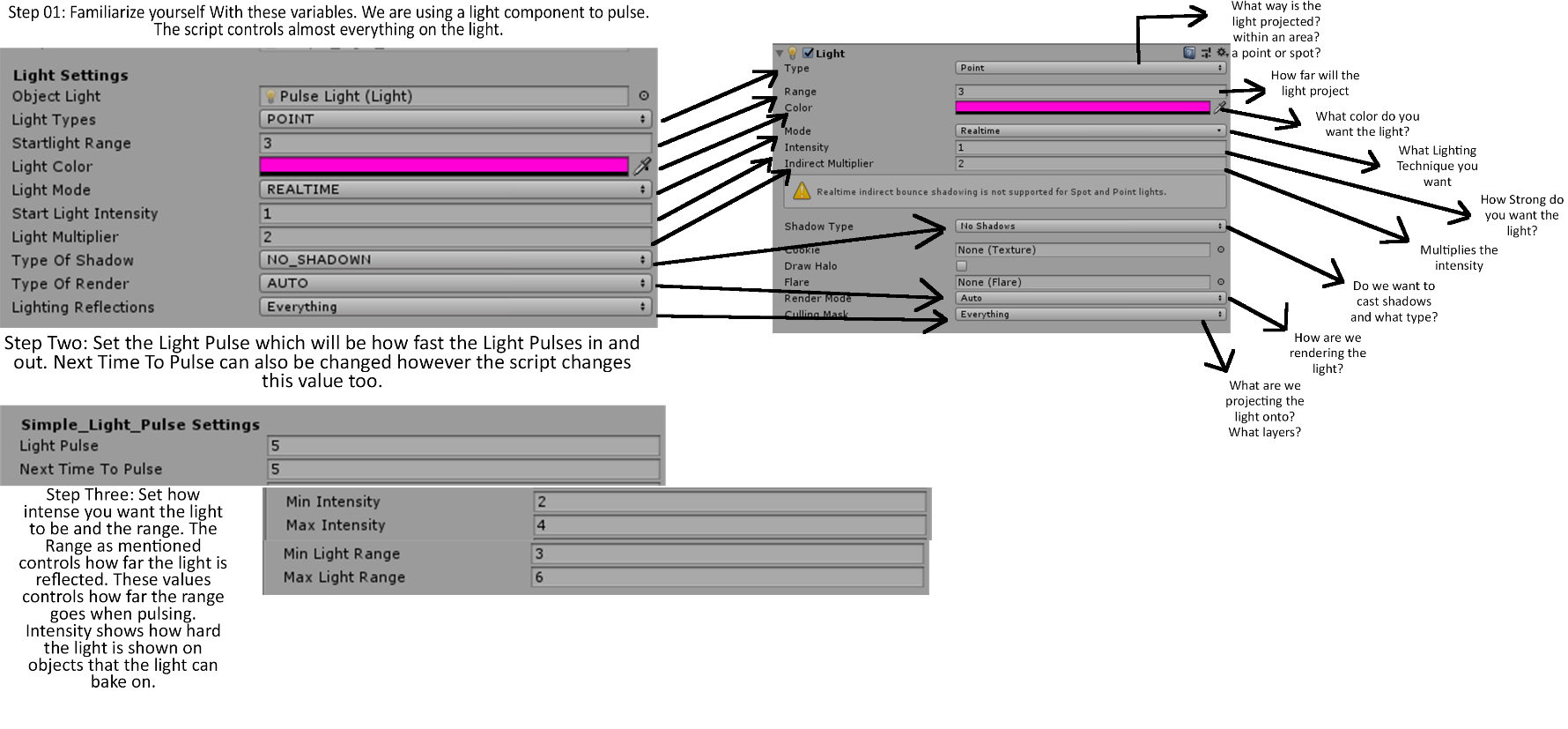


The Weapon Holder is also for scrolling though weapons if there is several we can go through them with the scroll wheel.

# Environment Scripts

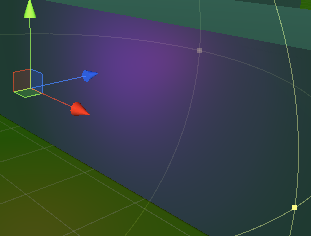
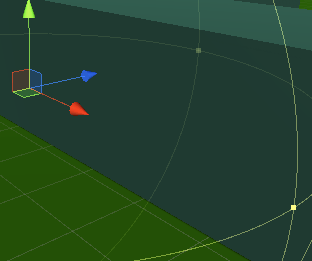
### Pulse Light:

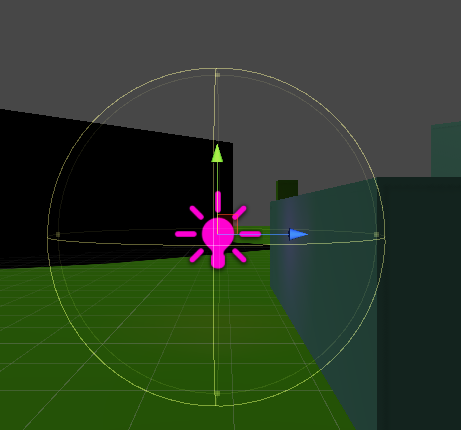
The Pulse light is a light in which pulsates in an out. It is an effect archives through code. However the reason this class is important is because this class gives us full access to the Light Components variables without needing it to exist before runtime. What I mean by this is you can make an empty GameObject attached the script and the code will work.



The Layer mask allows us to decide what will be rendered depending on a layer and the intensity allows us to show how hard we want the light to project.

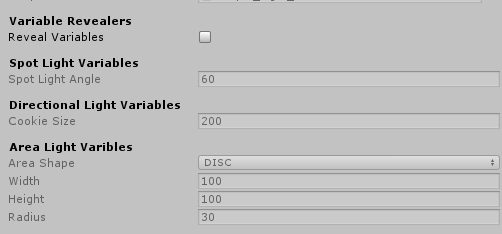
*Light Light no longer projects onto the cube object because we took the layer off*



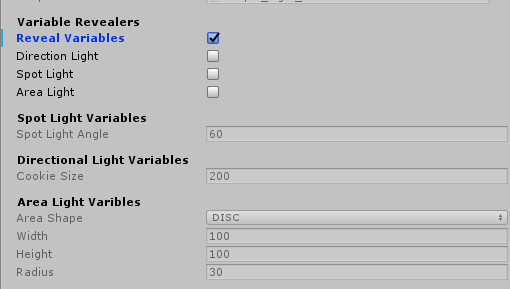


We can also edit the range of the light no matter what light type we are using (light types are Spot, Point, Direct and Area).

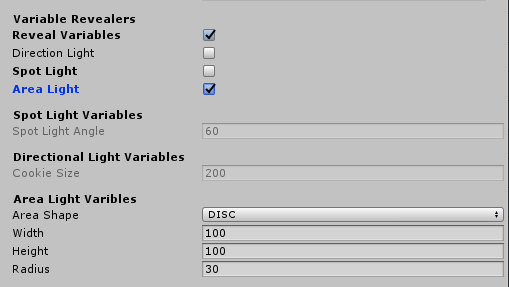
The script allows us to change the Light Type. The Mode of lighting Colour of the light if we project shadows etc. We make the light and it obtains the values that we set up. You also may be wondering what about editing the spot light angle? Or the area light shape being rectangle or a disk or even maybe you wonder what about the Directional light Cookie size variables. All useful properties but I found a way around this without cluttering the key logic.



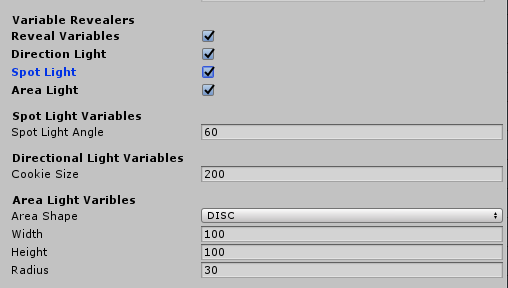
We have variables on this script which you may find you can’t edit all of them. As in you click on them in the inspector and you can’t type what you want. Well there is a reason for this



This class is controlled. A custom editor has been placed inside this code through another script which in short allows us to hide things or make that variable inactive. This works without having to play the game as we are setting up an object to work within the game. Which means all this is edited before play.



The reveal Variables allows us to unhide the Booleans in which control the variables inactive state. So when clicking on the Boolean Area Light to true. We simple turn on the Variables we want to edit



And this goes for all the variables we hide or are inactive. Remember the pulse was meant for Point lights however works with all lights. You can edit the script as much as you want in different ways to achieve whatever pulse effect you see as the best.