# AIM ASSIST PLUGIN

Project Zero

# Contents

## Scope and Limitations

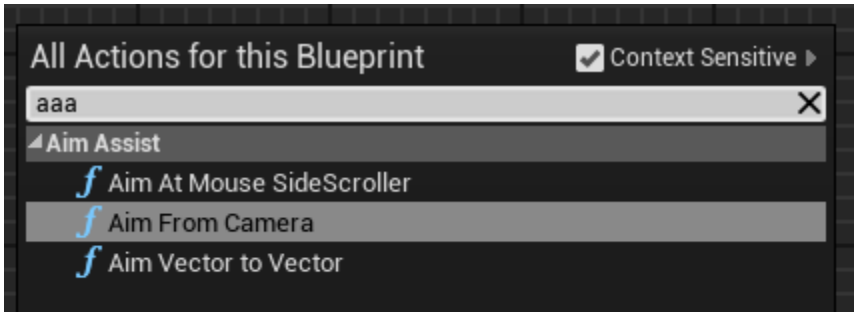
The main scope is to provide simple functions that can facilitate quick aiming procedures. In the first person view, aiming where the camera points is the norm. In the third person view, the same logic is applied. For sidescroller shooters, aiming where the mouse is pointing is the norm. Finally, some easier vector-to-vector aiming fills all other holes, as would be needed by NPCs and turrets or any other idiosyncratic aiming needs.

The plugin tries to fill the gaps in the first and third person view mode where the camera is the main source of aiming. Although there was a consideration for aiming at the mouse from the camera, there are a few issues that make this almost impossible to be done correctly, mainly the source of aiming and the depth. As such, those are out of scope.

There was also a consideration for accuracy computation. However, because it is 3 dimensional, the amount of work and time needed would basically make it the main feature rather than the aiming. As such, it is also out of scope.

## Features

There are 3 nodes that take inputs into consideration and output vectors and if needed rotators. All nodes are quick to search using “AAA” as a prefix.



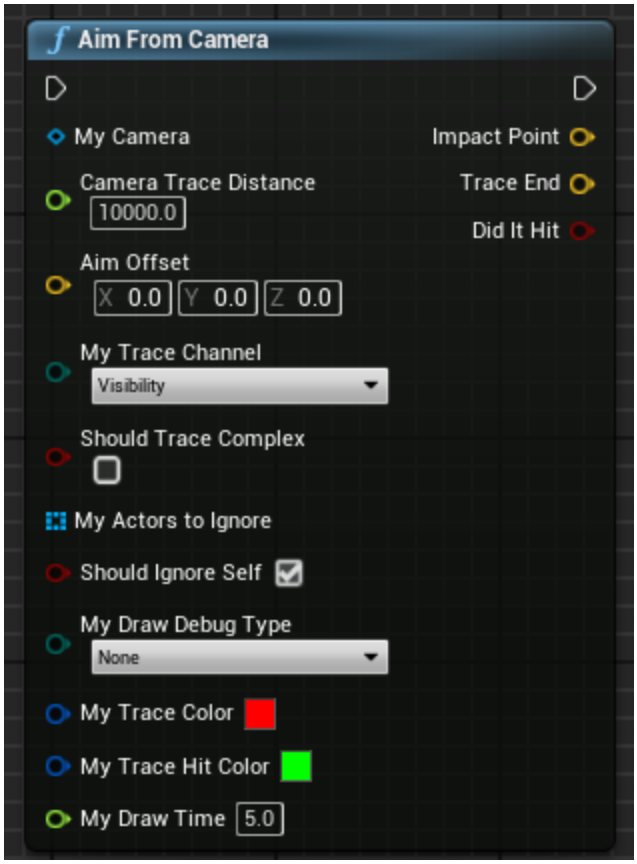

### Aim From Camera Node

The Aim from Camera uses the center of the camera to trace for an impact and use that impact to aim at. If nothing is hit, the trace end can be used instead with the Boolean. There is also an Aim Offset provided, which offsets at the source (the camera) in case you want to offset it away from the center of the screen or further away from the camera (for instance if you want to summon a portal that shoots fire).

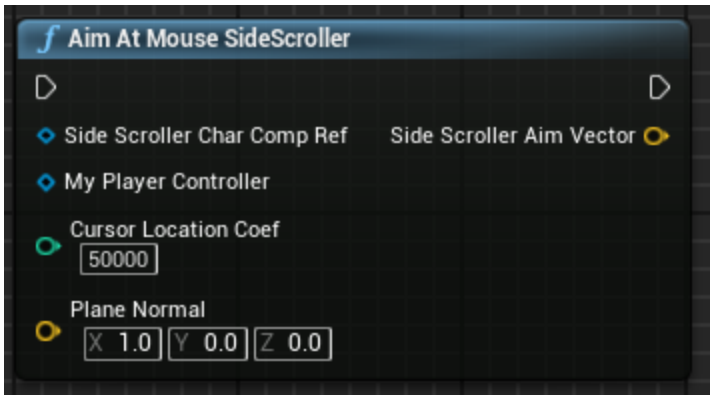
The rest are typical line trace parameters.

There are no rotators given as there are too many possible different sources to derive from. As such, only the 3D vector location is provided and a rotator can be derived using the “find look at rotation” function.

**NOTE: The Camera is a mandatory input.**




## Aim at Mouse SideScroller Node




The Aim at Mouse SideScroller will take in a character component reference to see where it is in the world, your player controller to determine what the mouse cursor is, a cursor location coefficient to determine the depth and the plane normal for the plane normal.

The cursor location coefficient has to be a high value. I found that 50 000 is most appropriate but it is up to you to determine it.

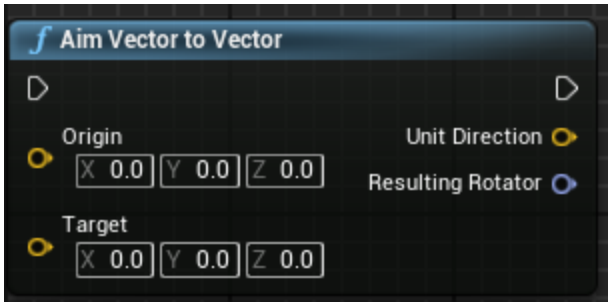
The plane normal uses the default plane normal from the sidescroller template.

As with the Aim From Camera node, there are no rotators given as there are too many possible different sources to derive from. As such, only the 3D vector location is provided and a rotator can be derived using the “find look at rotation” function.

**NOTE: The Player controller and the primitive component are mandatory inputs.**

### Aim Vector to Vector

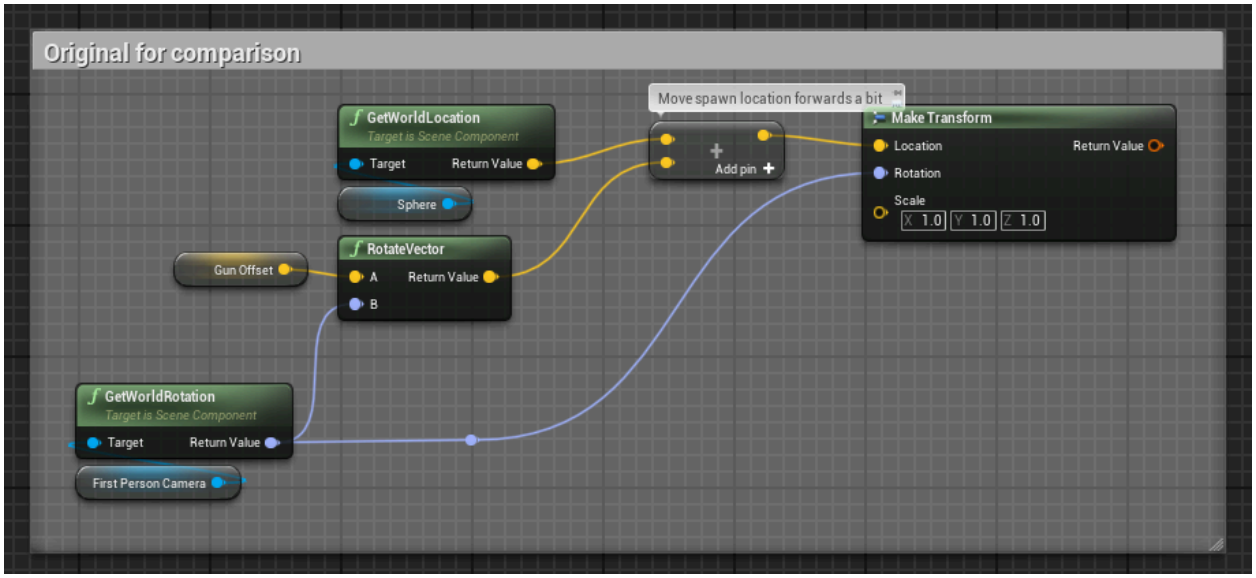
The Aim Vector to Vector is the most basic form of aiming and quickly offers both the unit direction vector and the resulting rotator. This is most suitable for non-character actors that require aiming such as turrets.

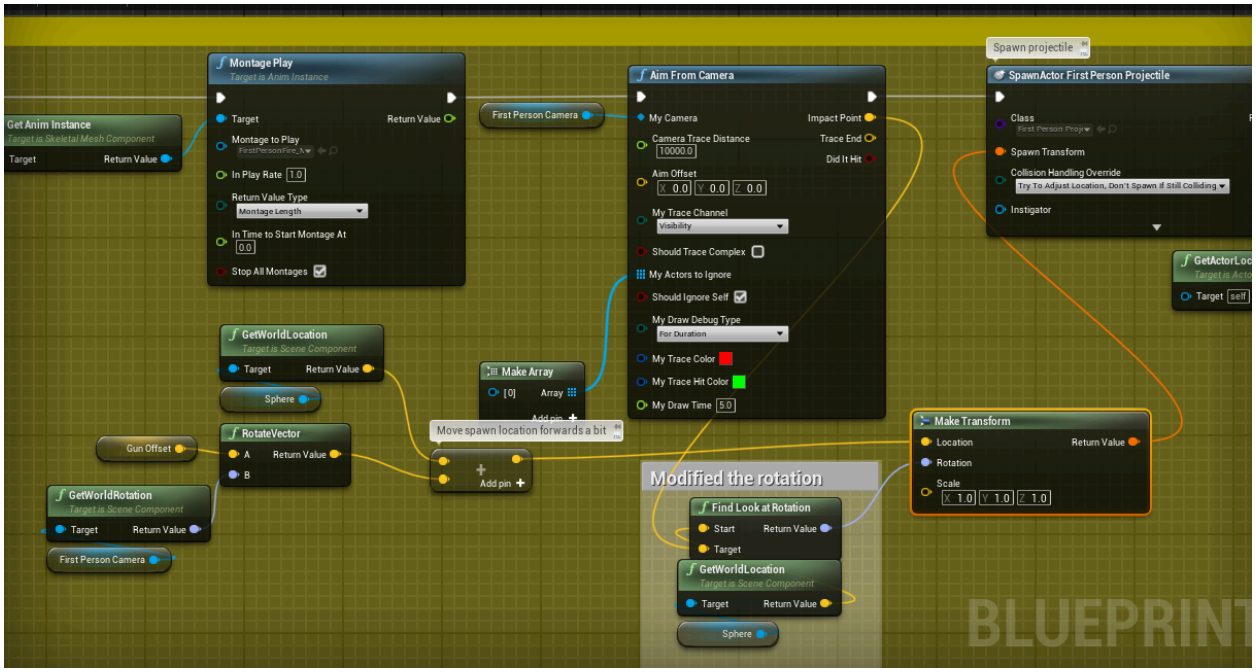



## Setup Guide

### First Person Shooter

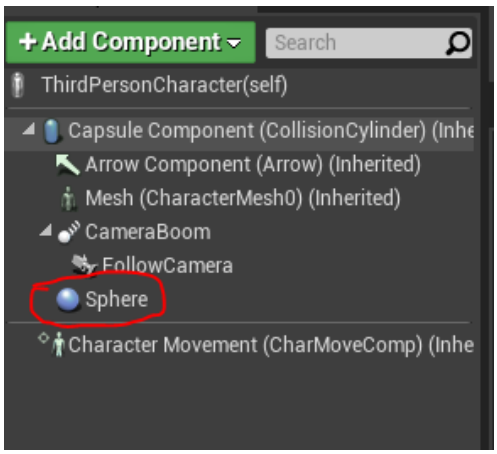
For the first person shooter, using the “shooting” part of the blueprints, you can set up the blueprint nodes as shown below. The original “aiming” is also shown for comparison. Originally, all it does is shoot somewhere around the center of the screen. The Aim from Camera node will however accurately find the center, shoot at it and correct the angle to shoot based on the distance.

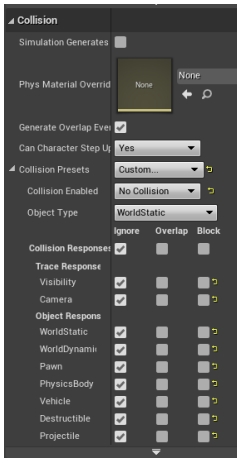


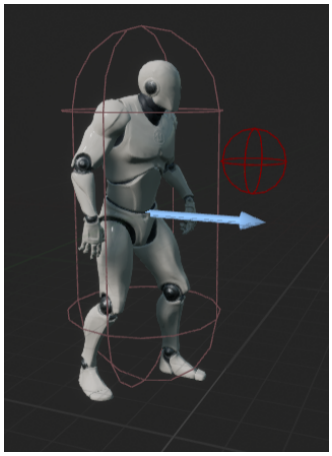



## Third Person Shooter

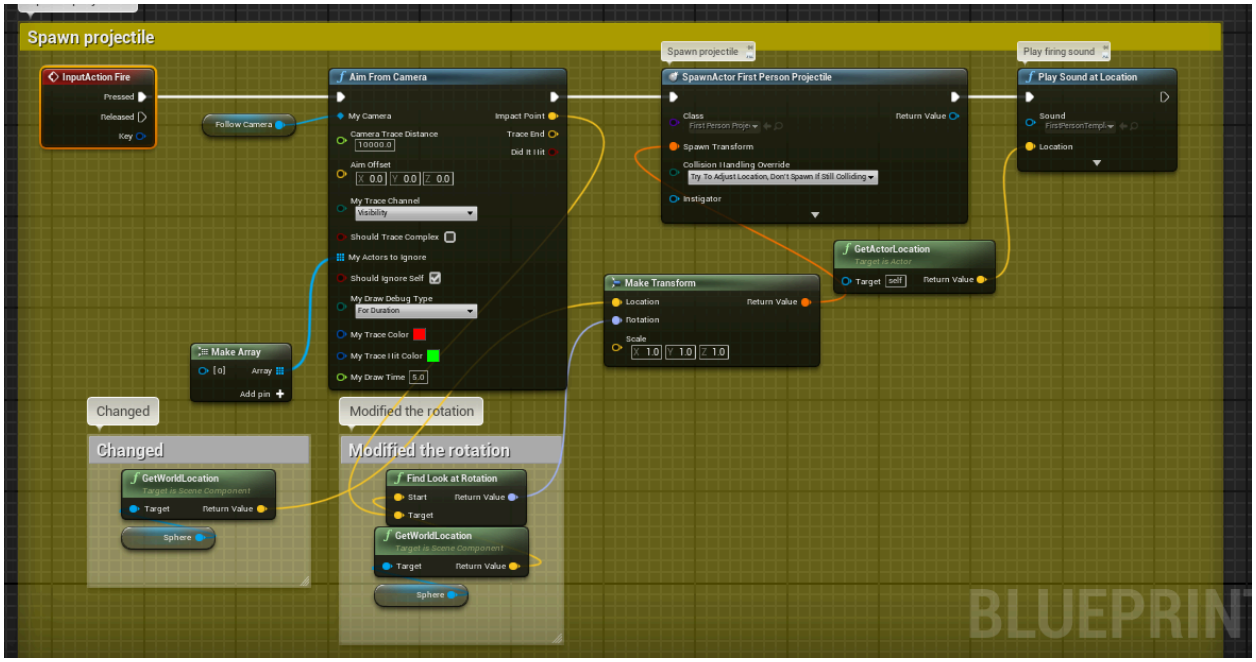
For the Third person, we start off with the third person template and add the shooting function from the first person. What we need to define is the source of the projectile. In this case, we added a sphere collider with no collision to shoot from in front of the character:



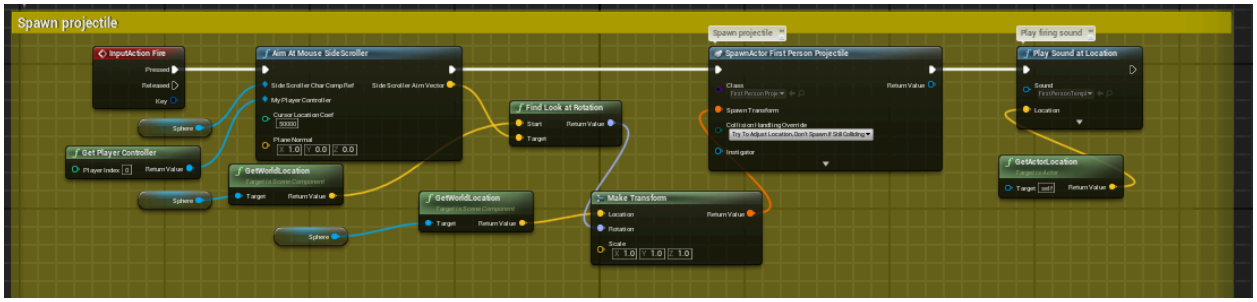
Afterwards, we can copy and paste the projectil function and remove the need to play a montage. Finally, we just need to define the source of the projectile spawn.



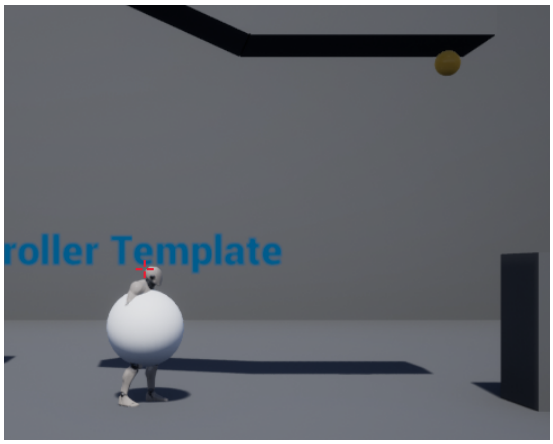

## SideScroller Shooter

As with the third person character, we will add a component with no collision to shoot from. In this case, it's a sphere mesh. It can be a collider or a gun socket or whatever you want.

Then we add the shooting blueprint function and modify it as such:



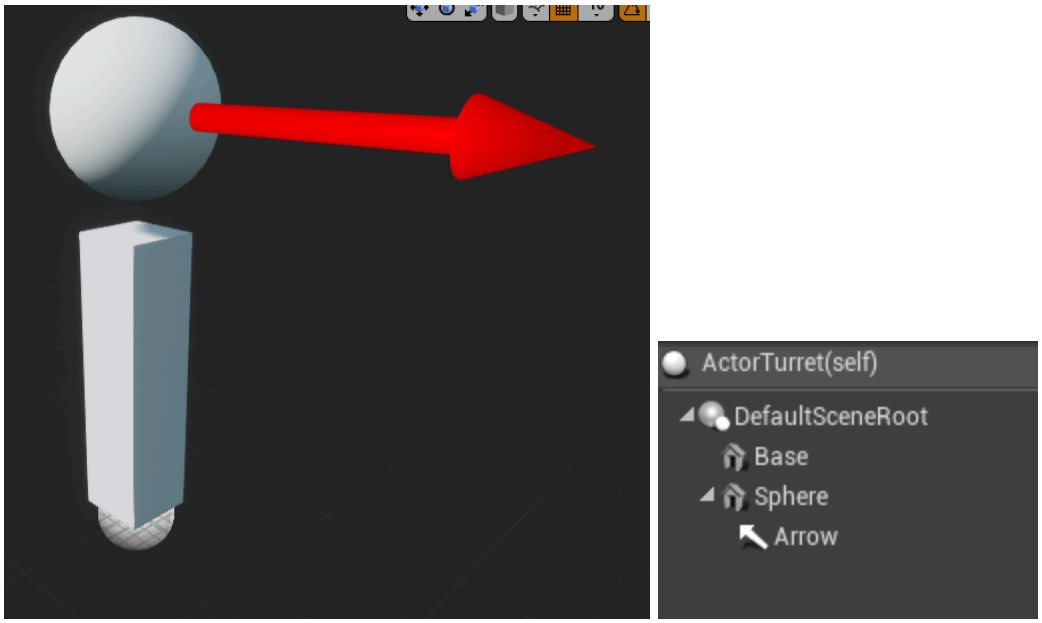

Here is the result:



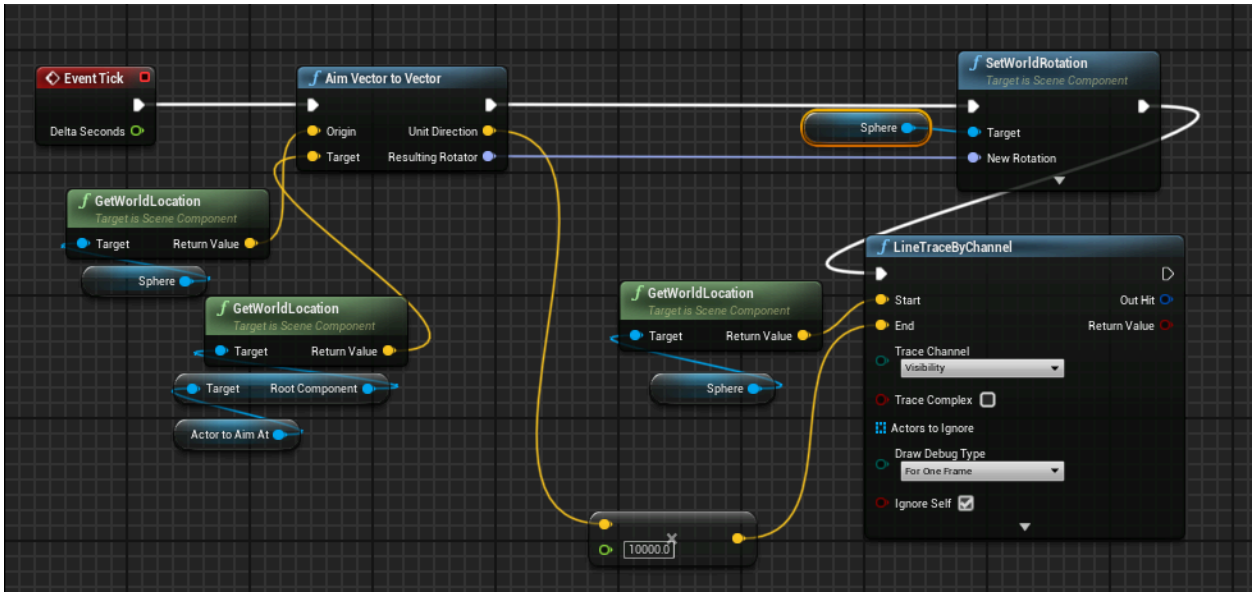

## Vector to Vector Aiming

This is best used for undefined or idiosyncratic needs for aiming. For instance a turret or an eyeball.

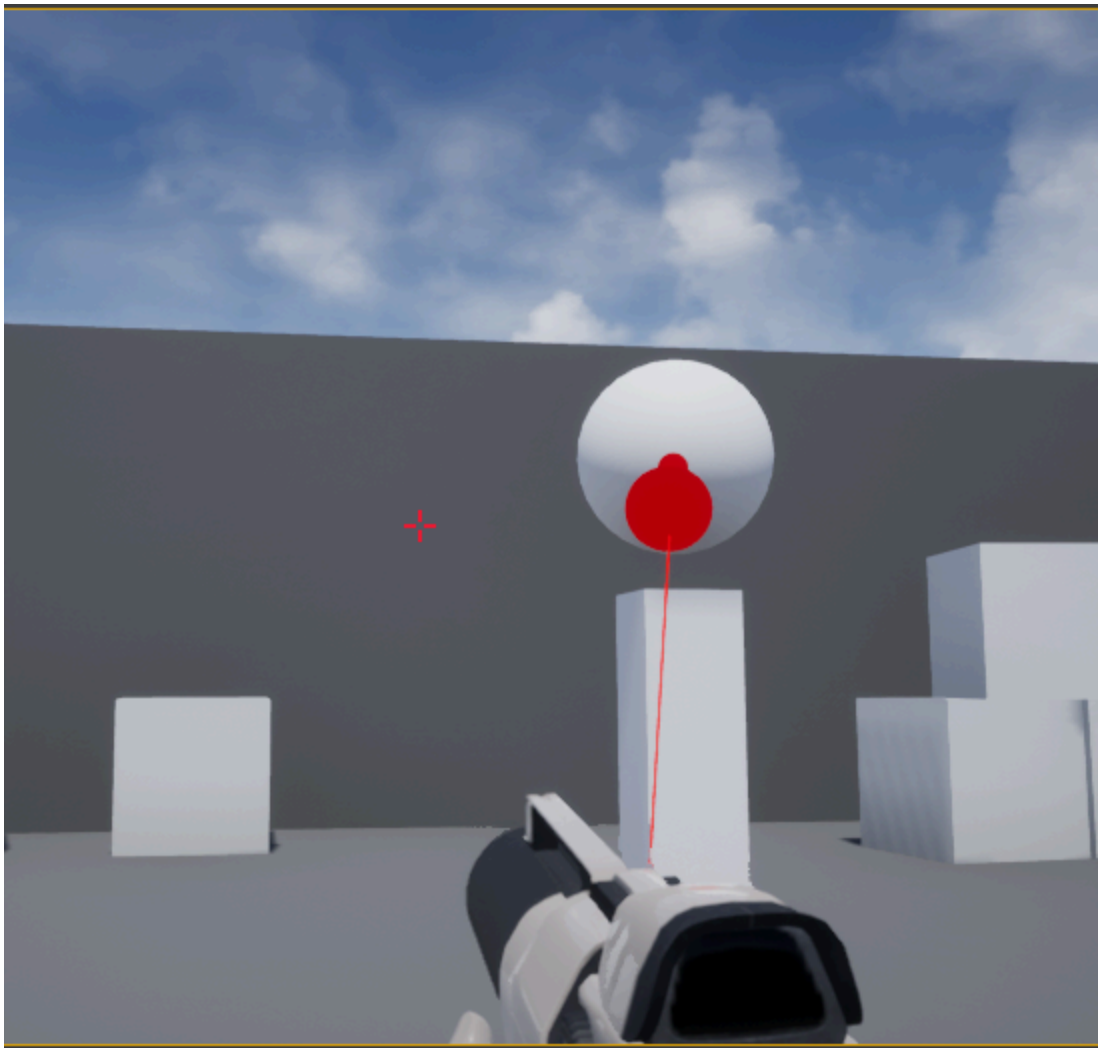
In this example, a turret is set up.




What we want to do is make the head (sphere) turn towards an actor. We added a public variable for another actor and in our tests we used the player pawn as the target.




Here is the result:



The turret is aiming at me and tracing at me.

## Future Developments

If there is a demand, a topdown aim assist could be provided although currently the top down template offers quite a lot to do it and the vector-to-vector aiming does fulfill that niche.