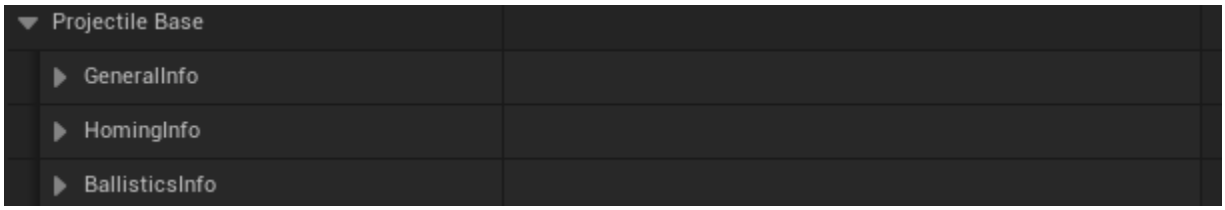
-Last Updated 1/31/2025.
Video Demo: https://www.youtube.com/watch?v=DhAvApan7Ao

Overgrown Projectile Base Class Documentation

The Projectile base is a configurable Actor Blueprint used to streamline the creation and iteration of projectiles. It can be found in Dynamic, Blueprints, Enemies, Projectiles.

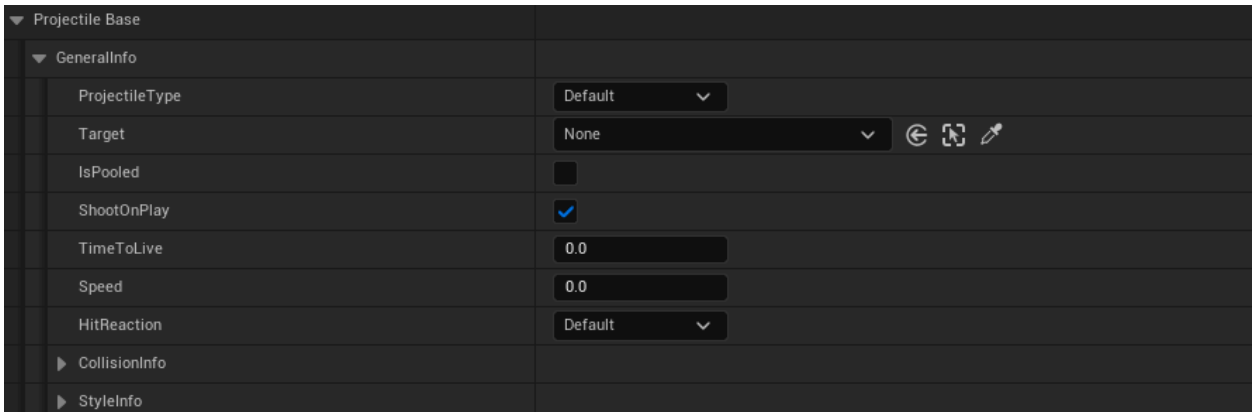
There are three main types of projectile: **Default, Homing, and Ballistics.**

A Default projectile will shoot in a straight line. It can optionally aim at a target. A Homing projectile will track to a specified target. A Ballistics projectile will shoot in an arc to hit a specified target. A target input is mandatory for both Homing and Ballistics projectiles.



The projectile has three main input sections: **General Info**, **Homing Info**, and **Ballistics Info.** Homing Info and Ballistics Info will only be relevant if the projectile is set to either Homing or Ballistics.

General Info:

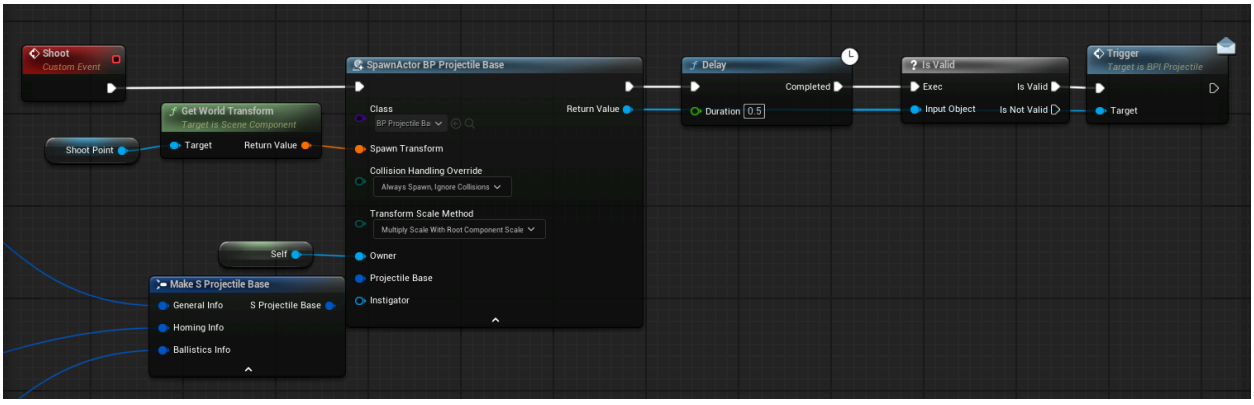


Projectile Type- This enumerator determines whether the projectile will have default, homing, or ballistics behavior. If you fill out ballistics information but set the projectile to default, none of the default information will be factored in.

Target- Sets a target the projectile will shoot towards. If no target is specified, the projectile will shoot in a straight line at an angle specified at spawn.

Is Pooled- This is important if your projectile is part of an object pool. If this is set to true, the projectile will spawn in when its owning actor does in a “deactivated” state. On impact, it will “deactivate” and return to the pool so that the spawning actor can shoot it again. This helps with memory, because it allows an enemy to spawn and destroy many projectiles quickly without risk of lag spikes and memory fragmentation. Pooling the projectiles also requires adding the object pool component to the shooting actor itself. Read more about this later.

Shoot on Play- This means the projectile will shoot as soon as it is spawned in (or called out of an object pool). If you set this to *false,* you must *trigger* the projectile some time after you spawn it in. *(Example below)*

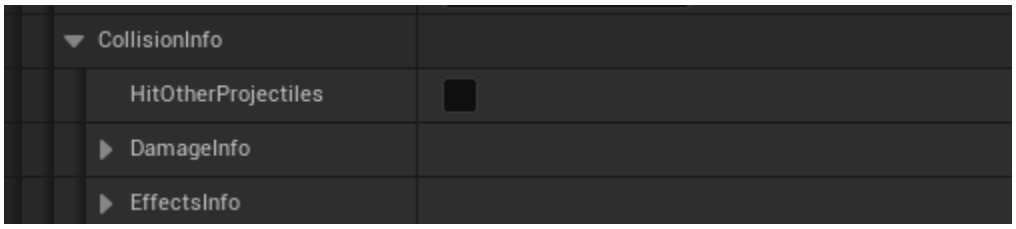


Time to Live- The time after a projectile is spawned or set-in-use out of the object pool. It does *not* start on trigger. If you plan to spawn in the projectile and wait 2 seconds before triggering, you should factor this into its time to live. This will not despawn a pooled projectile, only deactivate it.

Speed- Sets the speed of a projectile.

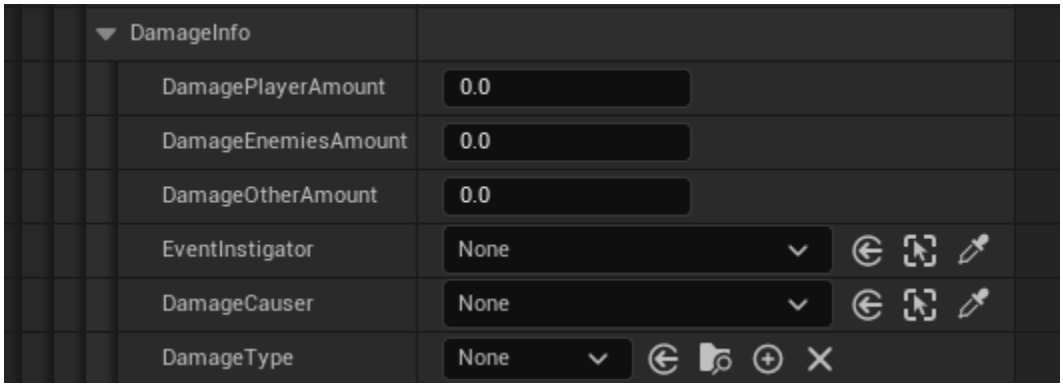
Hit Reaction- Choose between Default, Parryable, and Killable. A default projectile will ignore a player hitting it. If the player hits a parryable projectile, it will fly back in the direction the player faces and set to damage/knockback enemies instead of the player (if it was set to damage the player in the first place).

Collision Info:



Hit Other Projectiles- Should it hit other projectiles?

Damage Info:



This is identical to the AOE Damage info

Damage Player Amount-
The amount of damage that will be done to the player on sphere overlap. Keep this low for over-time AOE's, since it will apply at every update interval.

Damage Enemies Amount-

The amount of damage that will be done to enemies on sphere overlap.

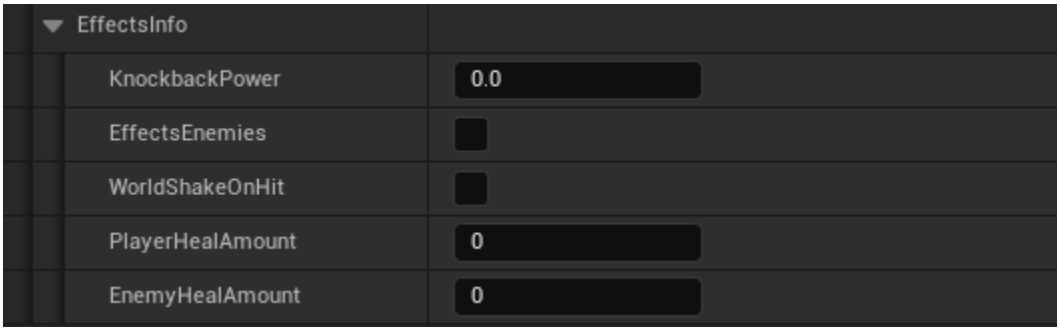
Damage Other Amount-
The amount of damage that will be done to other actors with the damageable interface

Event Instigator-
refers to the actor controller. Usually unimportant

DamageCauser-
Typically a reference to self, if the AOE is spawned by an enemy.

Damage Type-
This may be important if we ever do anything with damage types.

Effects Info:



This is identical to the AOE Effects info
Knockback Power-
If this is set to 0, no knockback will happen. A small knockback is around 500.

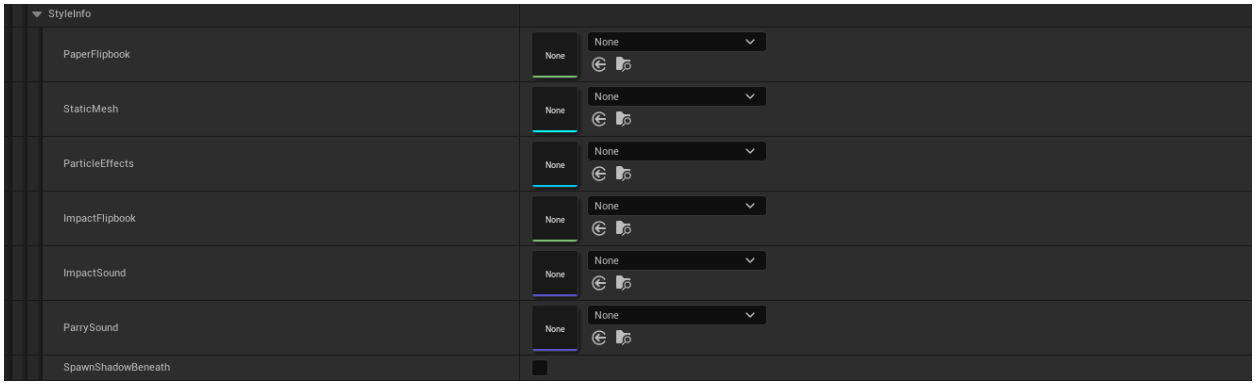
Effects Enemies-
Toggles enemies getting knocked back along with the player.

World Shake On Hit-
Toggles whether the world shake function will play if the player character is hit. This is currently non-functional.

Player Heal Amount-
How much the player will heal on AOE overlap. If 0, the heal function will not play.

Enemy Heal Amount-
How much the enemy will heal on AOE overlap. If 0, the heal function will not play.

Style Info:



Paper Flipbook- 2D sprite for the projectile. This can be animated or static. It will always face the camera.

Static Mesh- Just in case you need a 3D projectile. Lean away from this.

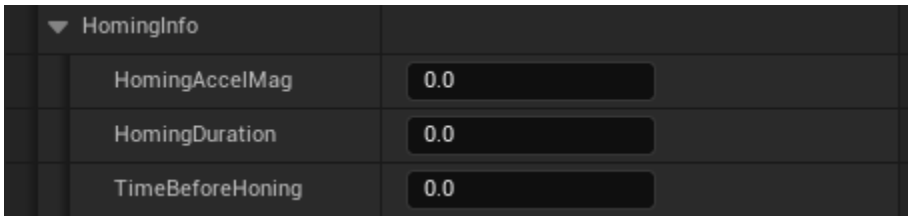
Particle Effects- Niagara system effects that will spawn on projectile impact.

Impact Flipbook- A flipbook that will play on projectile impact.

Impact Sound- Sound that will play on projectile impact.

Spawn Shadow Beneath- Will spawn a small circular shadow on the ground beneath the projectile

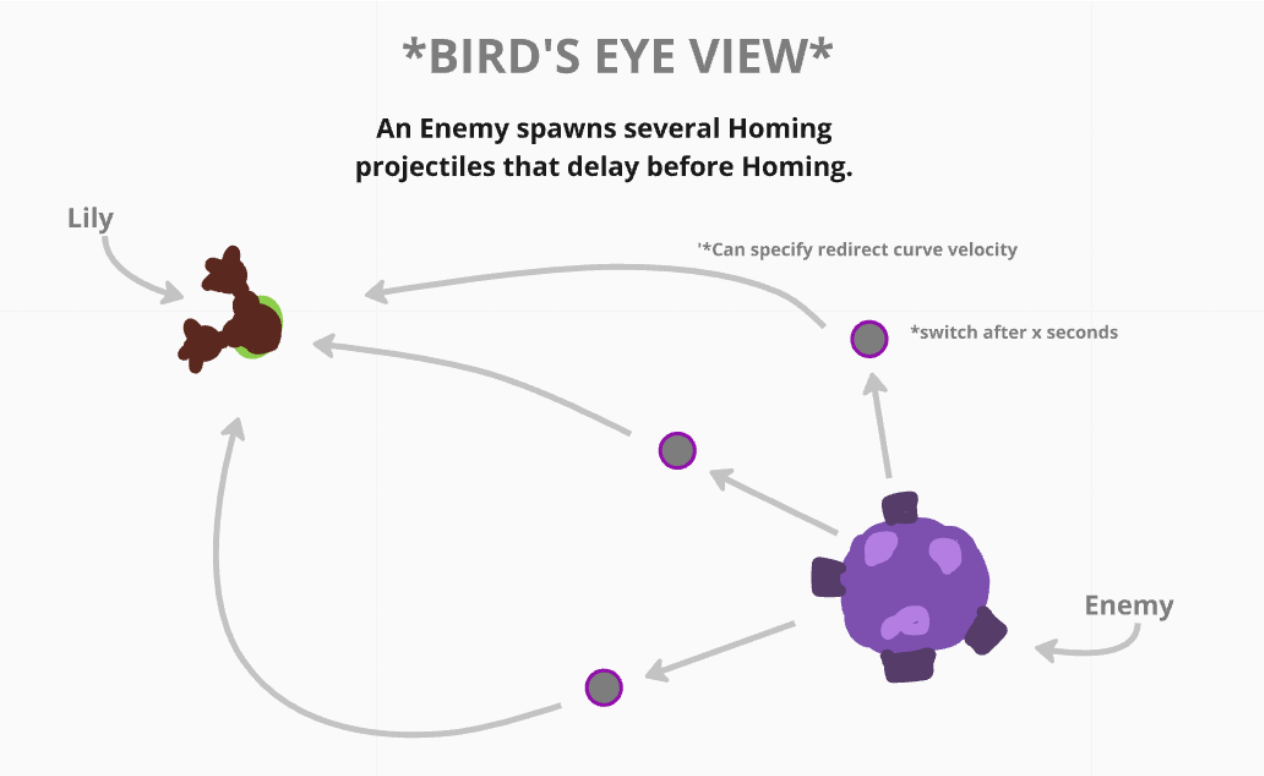
Homing Info:



Homing Accel Mag- How quickly should a homing projectile redirect towards its target? Higher numbers mean quicker redirects. This is limited by projectile speed.

Homing Duration- How long should the projectile home for before going in a straight line?

Time Before Homing- How long should the projectile shoot in a straight line before it starts homing to its target?



Ballistics Info:



Time To Reach Target- How long should the projectile take after activating to reach the target? This means that when the target is farther, the projectiles will be quicker.

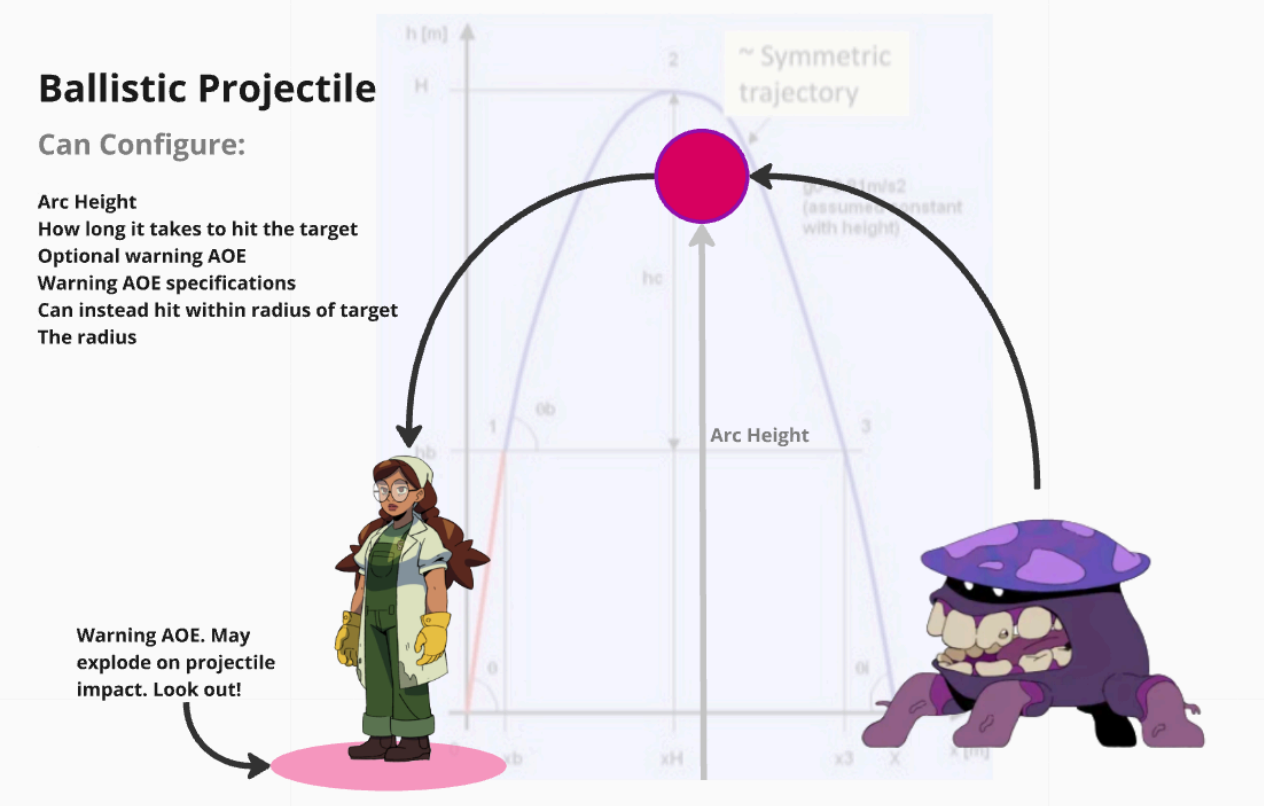
Arc Height- How tall should the top of the arc be above the shooter?

AOE Where Land- Should the projectile also spawn an AOE at its determined landing spot? This AOE will be destroyed once the projectile is destroyed or deactivated.

AOE Info- Struct with all information for the AOE that the projectile may spawn.

Hit Around Target- If checked, instead of aiming directly at a specified target, it will aim somewhere within a radius around the specified target.

Around Target Radius- If Hit Around Target is selected, this is the radius around the target that the projectile could land in.



Spawning Your Projectiles From Pools:

Creating a projectile pool is only recommended if you plan to have a kind of ridiculous number of projectiles spawning and despawning in rapid succession.

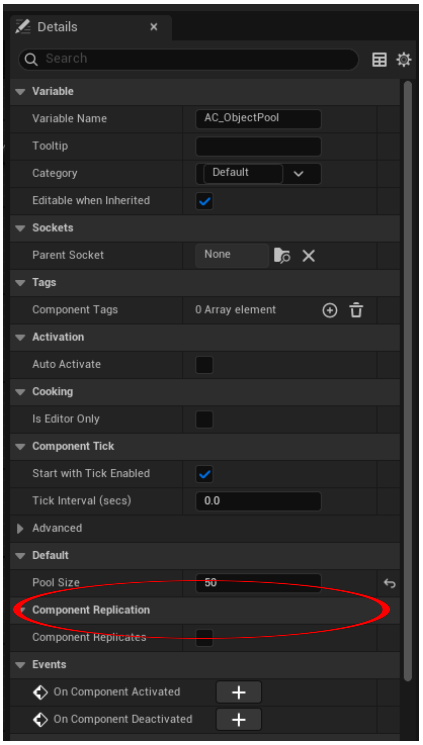


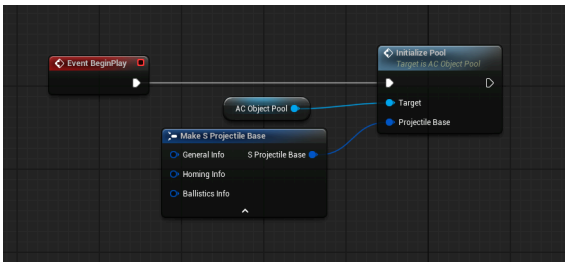


If you select “Pooled Projectile” in general info, but don’t set up the Object Pool actor component in the spawning actor, then nothing is going to happen. This is because the projectile will spawn in a “deactivated” state where its collisions, visibility, and ticking are turned off.



1. Add the Actor Component AC_ObjectPool to the actor that will be shooting the projectiles
2. Go to the variables and set the Pool Size variable to however many projectiles you want in the back-burner. This is the max number of pooled projectiles you can have activated at once.
3. On BeginPlay, add the AC_ObjectPool event “Initialize Pool”. Plug in Desired AOE Info.





4. When you are ready to shoot, drag out from the AC_ObjectPool component and call the event “Spawn from Pool”. **You can only have 1 type of pooled projectile per spawning actor.** The actor can still spawn any number of unique unpooled projectiles.

