Projectile Management System.

Creating Projectiles :

To create a new projectile make a PrefabVariant form the Projectile_Template.

Make your projectile mesh or effect a child of the Effects_Mesh transform.

Make the effects you want to play when the projectile hits a target a child of the Effects_Impact transform

There are options for :

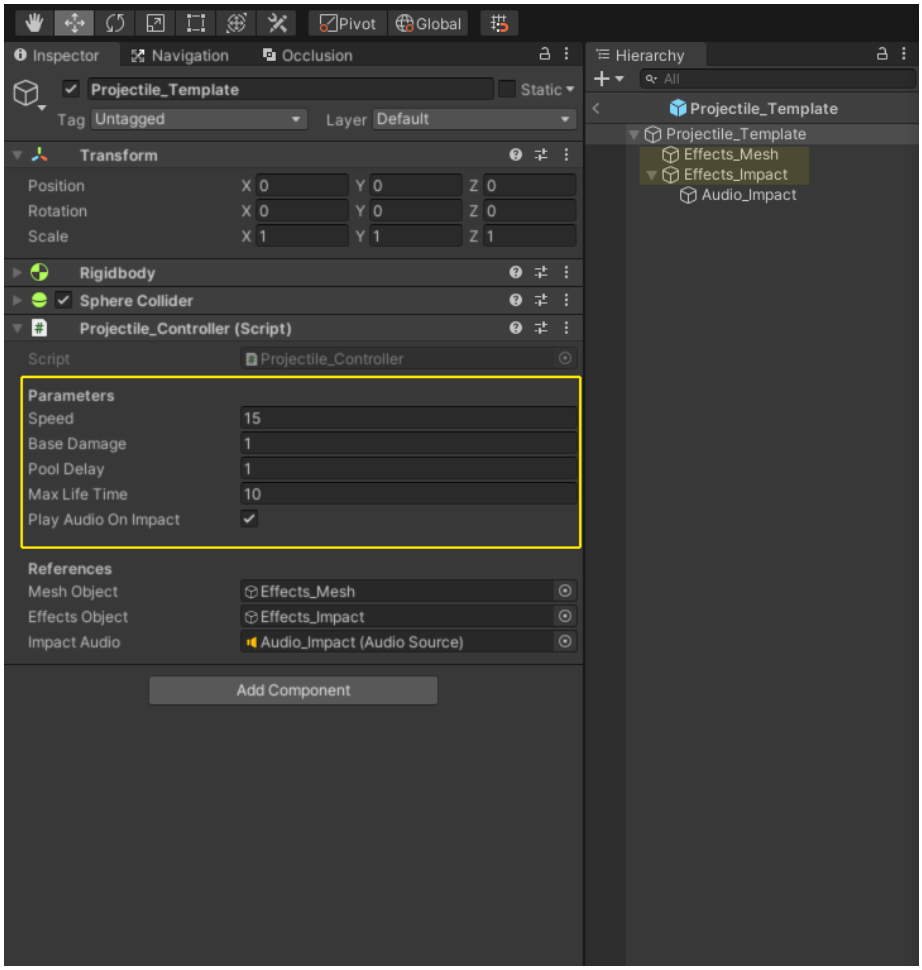
Speed : projectile movement speed,

Base Damage : can be multiplied when fired by a _damageMultiplier parameter,

Pool Delay : the time to wait for the explosion effects to play out before pooling,

Max Life Time : if nothing is hit how, long till the projectile is pooled,

Play audio on impact : yes or no.

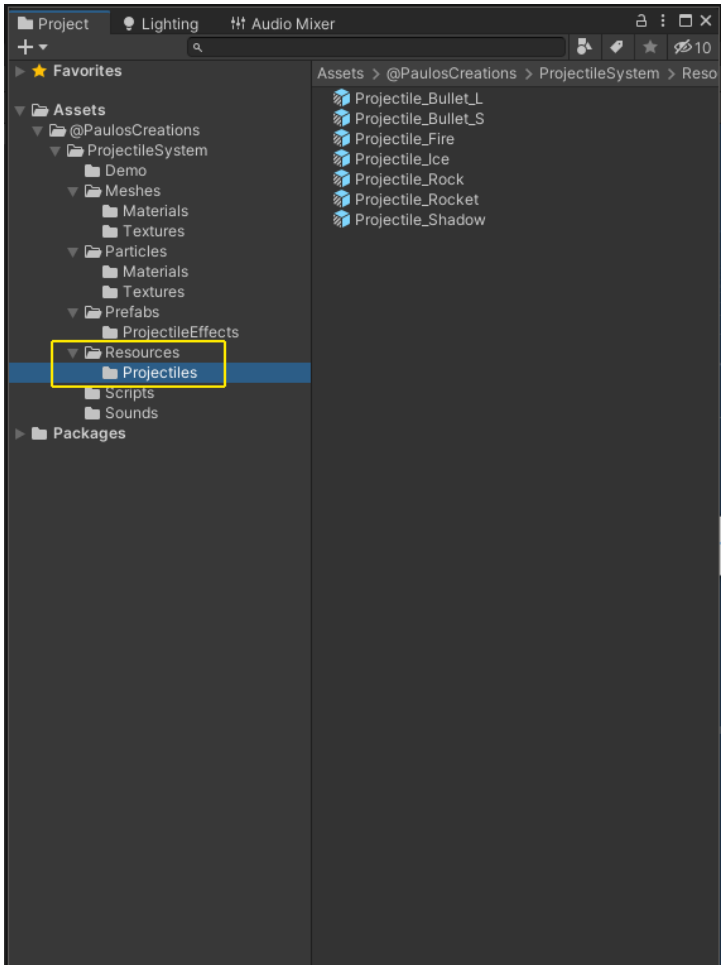


Storing Projectiles:

Created projectiles should be placed in a folder inside a Resource folder in your project.

The name of the folder is used to get the projectiles by the Projectile Manager.

You can make different folders for different levels if you want to.

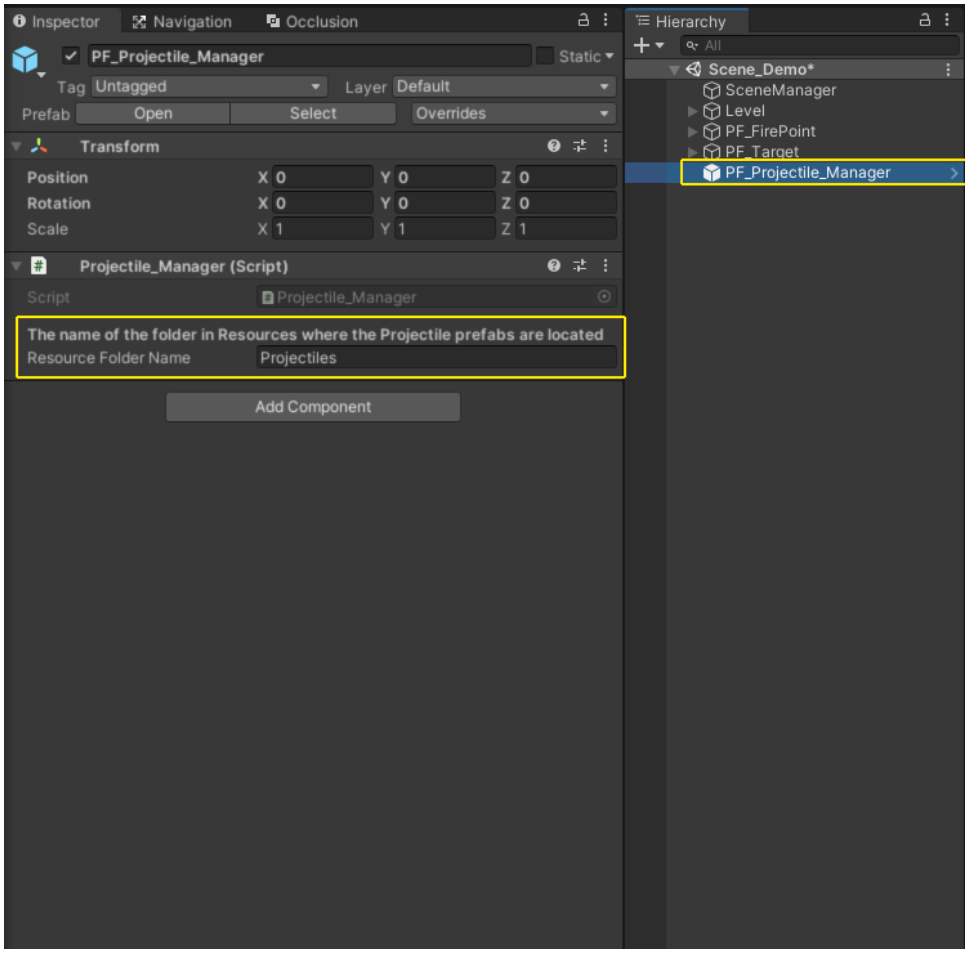


Projectile Manager:

Drag the PF_Projectile_Manager into each scene you want to be able to fire projectiles in.

The Resourse Folder Name should be the name of the folder you placed the projectile prefabs in.

This can be a different folder for each level, but can not be changed at runtime.



Fire Projectile:

From any script in the level you can fire a Projectile by calling one of these Functions.

To acces the manager include the NameSpace : using Paulos.Projectiles

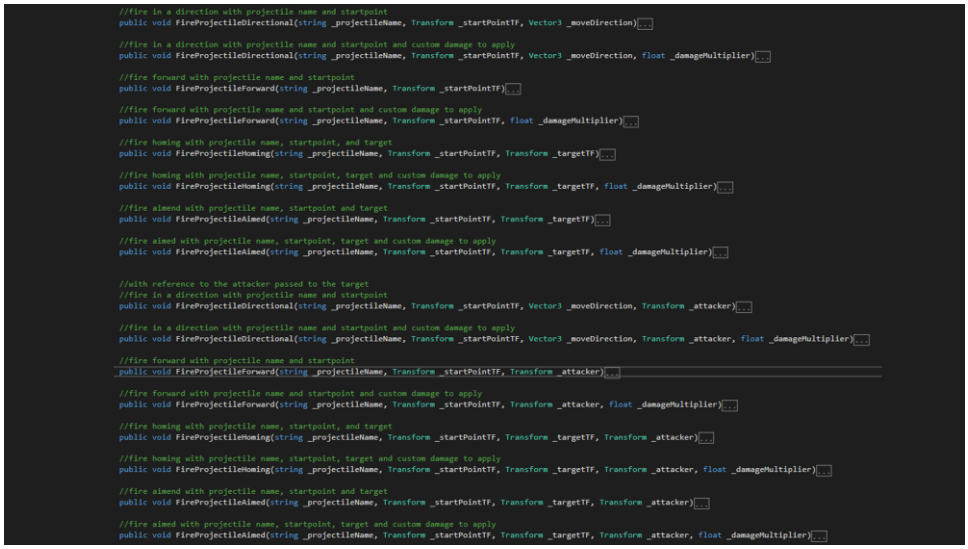
To call the functions that fires a projectile :

Projectile_Manager._Instance.FireProjectileForward(ProjectileName, the Transform you fire from);

There are 4 ways of projectile movement : straight forward, aimed at target, directional and homing.

All with several parameter options.

(including a damage multiplier and a reference to the transform that fired the projectile.)



Hitting the target:

Place the Projectile_Impact script on the targets you want to be effected/damaged by the projectiles.

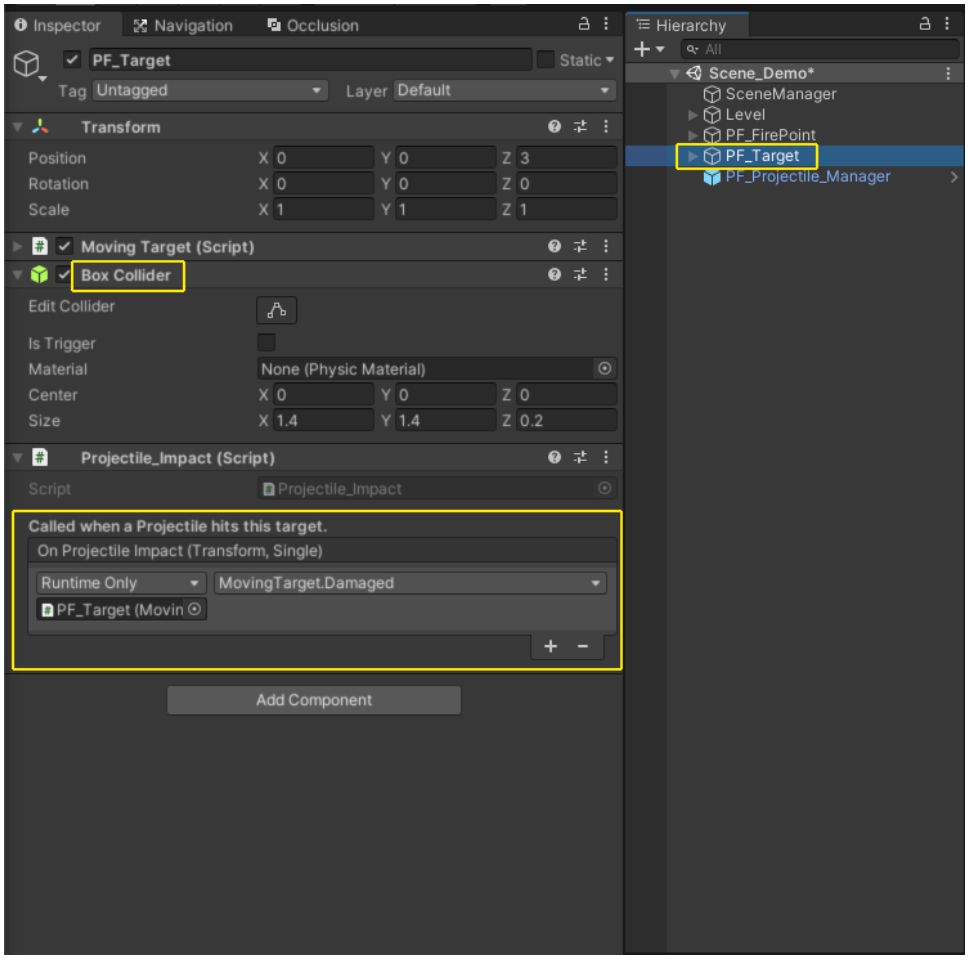
If the target has a RigidBody component attached to it,

the script should be placed on the Transform that RigidBody component is on.

Otherwise the script should be placed on the Transform that the Collision is on.

On a projectile hitting a target the OnProjectileImpact Event will be called passing the damage and optional transform that fired the projectile.

From here you can execute the functions from any script you want to be called when the target is hit.


Inspector
Navigation
Occlusion
PF_Target
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 3
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Moving Target (Script)
Box Collider
Edit Collider
Is Trigger
Material None (Physic Material)
Center X 0 Y 0 Z 0
Size X 1.4 Y 1.4 Z 0.2
Projectile_Impact (Script)
Script Projectile_Impact
Called when a Projectile hits this target.
On Projectile Impact (Transform, Single)
Runtime Only
MovingTarget.Damaged
PF_Target (Movin
Add Component
Hierarchy
All
Scene_Demo*
SceneManager
Level
PF_FirePoint
PF_Target
PF_Projectile_Manager