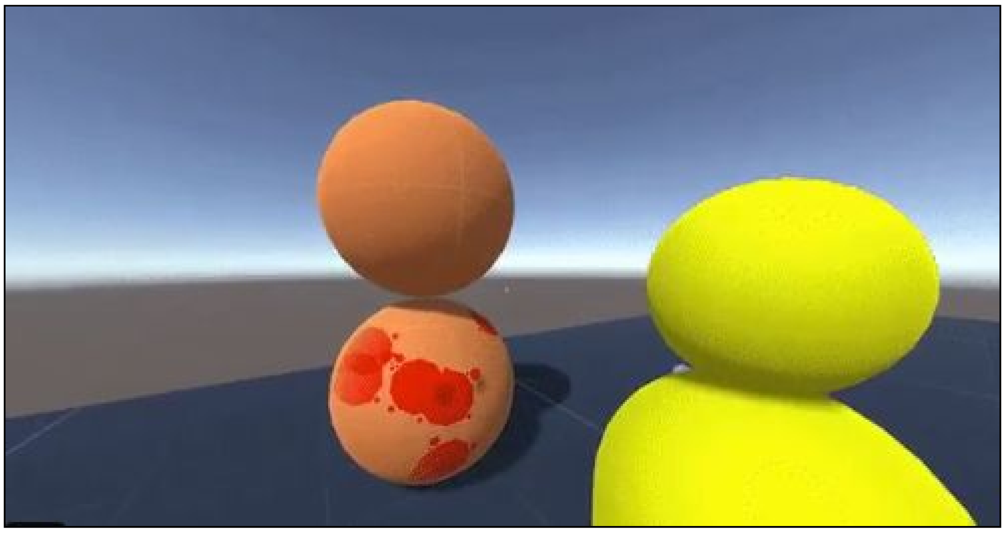
FPS Decal System User Guide

Welcome to the user guide for the FPS Decal System, this guide will cover how to use the system and provide screenshots on how the UI works. This is included in both the Unity asset store version of the system and the open source Github repository.

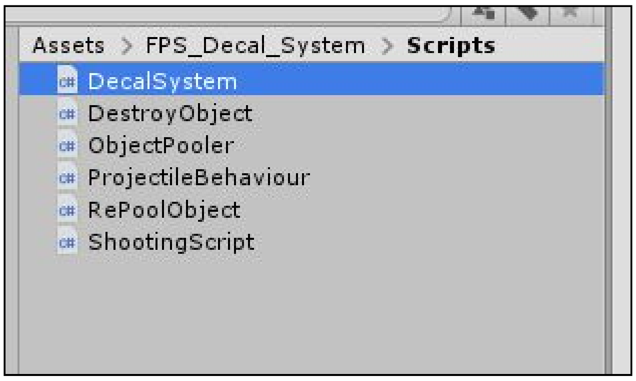


What is the FPS Decal System?

Before any information about how to use the system is given it is important to explain what it is and what features it has. The decal system uses a projector based method of creating runtime decals with a focus on first person shooters and the feedback that is generally expected from a modern FPS game. Within this package is a series of scripts, materials and an example scene that will allow the developer using this to improve their first person shooter games with bullet hole decals, splattering decals and effects for bleeding/blood splattering. These features come packaged with a character controller and basic shooting script that offers both raycast and ballistic based shooting logic, along with bullet penetration, exit decals and tag-based overrides.

The scripts provided

The main meat of the system is the scripts provided within the package. These are variety of uses from the utility of the pool manager to the main decal code of the DecalSystem class. These class provide the base experience well but can be expanded upon by the developer to include greater numbers of use cases or to make the system better suited to the experience they want to craft. The scripts provided are as follows:


Assets > FPS_Decal_System > Scripts

- DecalSystem
- DestroyObject
- ObjectPooler
- ProjectileBehaviour
- RePoolObject
- ShootingScript

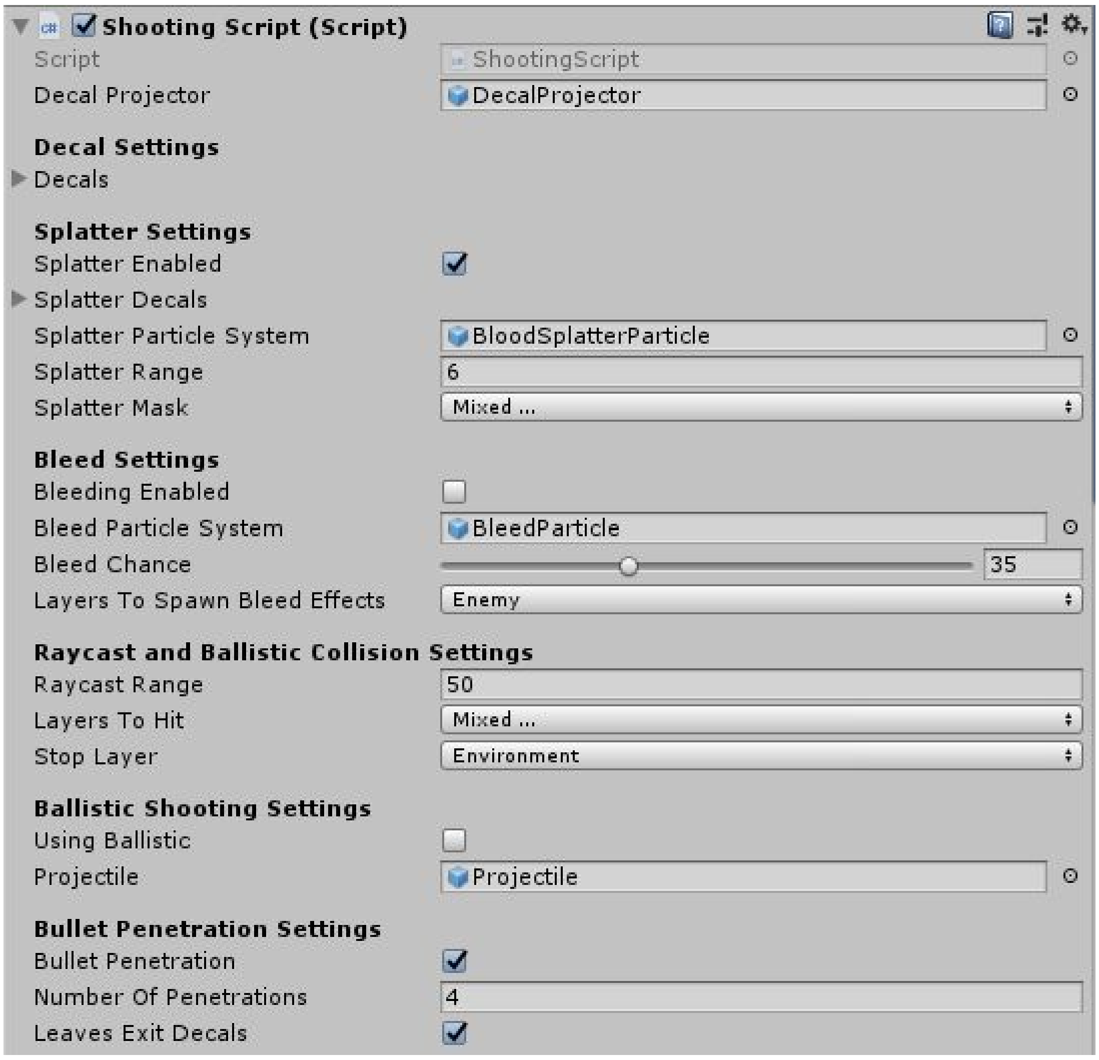
The DecalSystem class:

This script is the main script which is responsible for the creation of decals, splatter decals and the particle effects. It inherits from the monobehaviour class which means any script that want to use the methods from this script will only have to inherit from DecalSystem and adjust the inspector settings to their desire. This is designed in a way in which there is a single method, the LeaveDecal method that must be called in order to place a decal down. This takes a few arguments such as position (vector 3), rotation (quaternion), hit (raycasthit) and an isEntry (bool), apart from this the rest is handled based on the settings in the inspector.

The DecalSystem script can be used without the provided shooting scripts as a standalone piece of code by calling the LeaveDecal method, this allows the use of custom shooting scripts if a more complex solution is required. It can also be used without any shooting logic if that is not required.

The ShootingScript class:

This script provides shooting logic to a gameobject that it is placed upon. This script inherits from the decal system class and therefore takes all the settings from that script, creating a single location in the inspector to change settings. This script can be used out of the box or modified for more complex use. There are no callable methods in this script and the LeaveDecal method is called automatically. In order to use the ballistic settings a projectile prefab must be placed in the appropriate field, this prefab must have a projectile behaviour script on it in order to work correctly.

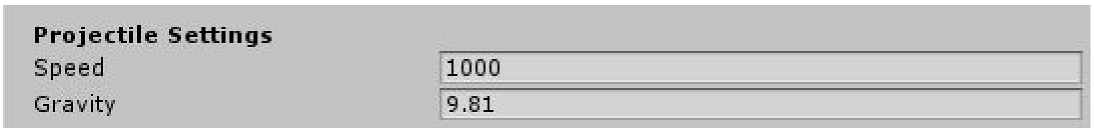


Shooting Script (Script)

Script	ShootingScript
Decal Projector	DecalProjector
Decal Settings	
Decals	
Splatter Settings	
Splatter Enabled	✓
Splatter Decals	
Splatter Particle System	BloodSplatterParticle
Splatter Range	6
Splatter Mask	Mixed ...
Bleed Settings	
Bleeding Enabled	☐
Bleed Particle System	BleedParticle
Bleed Chance	35
Layers To Spawn Bleed Effects	Enemy
Raycast and Ballistic Collision Settings	
Raycast Range	50
Layers To Hit	Mixed ...
Stop Layer	Environment
Ballistic Shooting Settings	
Using Ballistic	☐
Projectile	Projectile
Bullet Penetration Settings	
Bullet Penetration	✓
Number Of Penetrations	4
Leaves Exit Decals	✓

The ProjectileBehaviour class:

The projectile behaviour script, like the shooting script, inherits from the DecalSystem class, giving it all the same settings that the shooting script has. This script also has a method that is called from the shooting script that assigns all of its settings over the projectile when it is created, meaning you do not need to enter the same settings twice. This script should be placed on the projectile prefab you want to use for ballistic shooting. The script also has two projectile specific settings, speed and gravity. These change how fast a projectile will travel and how affected by gravity they are.



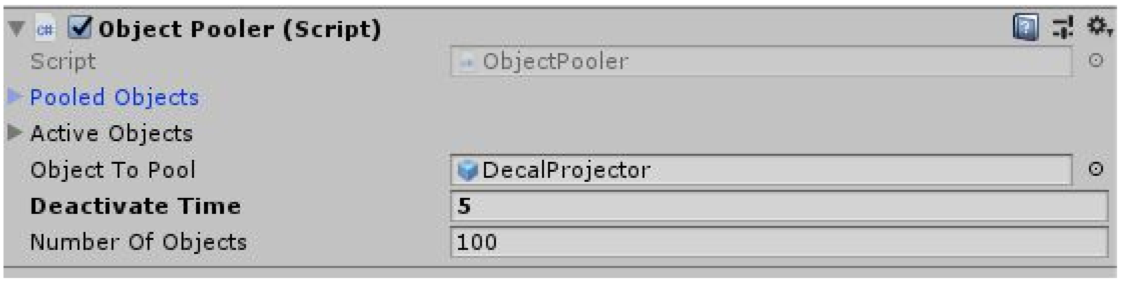
Projectile Settings	
Speed	1000
Gravity	9.81

The DestroyObject and RePoolObject classes:

These two classes are similar in their functions, mainly they exist to clear the game of decals, projectiles and particle systems once they have done what they need to do or a certain amount of time has elapsed. The DestroyObject class does so without use of the pooling system whereas the RePoolObject uses the pooling system. Currently only the decals use the pooling system so other spawned objects in the scene will instead use the DestroyObject class.

The ObjectPooler class:

This is a basic pool manager and pools the bullet hole decals and splatter decals. This script is included in the pool manager prefab which must be in the scene in order for the system to work. The settings for this script are the object to pool, in this case the projector prefab you want to use for the decals, a maximum number to spawn at once, and a time limit for how long they will stay in the world. If more decals than the maximum are spawned then the oldest decals will be re-pooled and used as a new decal. In addition an array of pooled objects and active objects can be seen, this is useful for debugging purposes but doesn't need to be manipulated manually.



Object Pooler (Script)

Script	ObjectPooler
Pooled Objects	
Active Objects	
Object To Pool	DecalProjector
Deactivate Time	5
Number Of Objects	100

The materials provided

The FPS decal system comes with a number of example materials provided. The decals system uses a projector based system which means a certain projector shader must be used in order for the system to work properly. Two shaders are provided by Unity for the use with projectors, a light and a multiply shader. All of the provided materials make use of the multiply shader. There are some requirements in order to get the materials to be projected properly by the system:

The 'Cookie Texture':

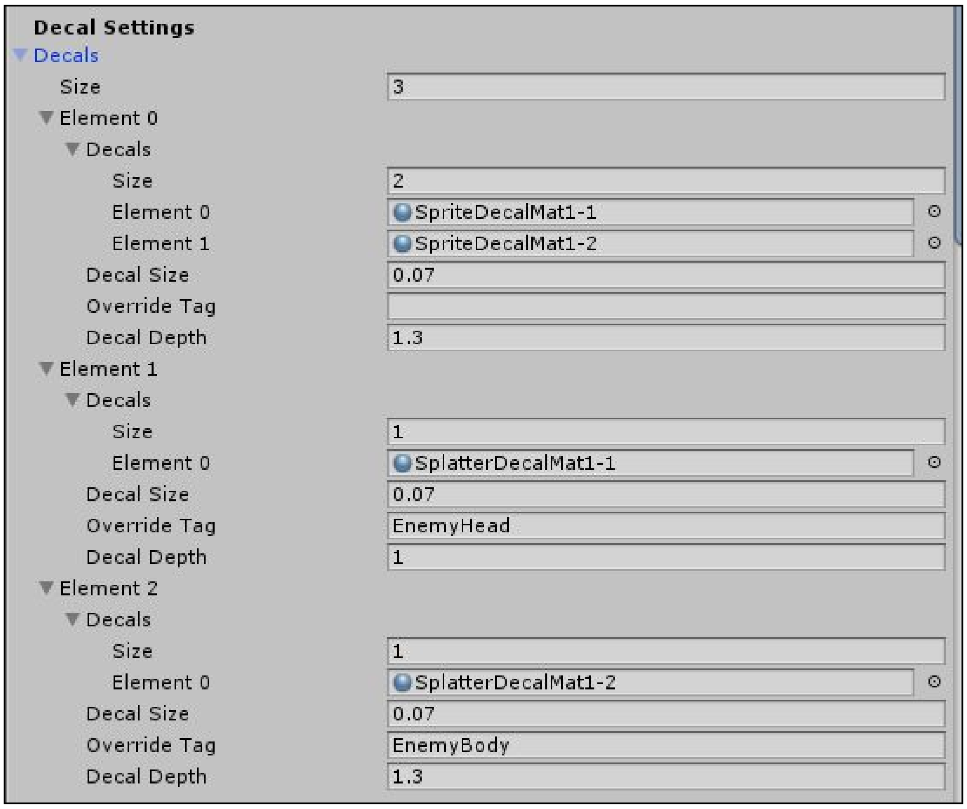
- Make sure texture mode is set to "Clamp"
- Turn on the "Border Mipmaps" options in the texture import settings
- Use an uncompressed texture format
- Set the alpha channel to "Alpha from Grayscale"
- Make sure there is white space around each texture and that no texture flows over the edge.
- White space is used for transparency.

A falloff texture is provided within the system that must be placed into the Falloff texture area in the material. This will ensure that the projector displays properly. A custom falloff texture can be made if you want a different gradient of falloff, with the leftmost column being white for transparency. Once again this texture should be set to Clamp.

Setting up decals:

Decals can be set up in the inspector from the DecalSystem class or any class that inherits from it.

- The 'Decal Settings' header contains all the settings for placing a decal in the world
- The size field is how many different decals you want to have available, this should be set to at least one.
- Inside each of these collections is some additional per decal settings.
- The decals collection is an array of materials that will be randomly selected from on decal placement, this can be any number of kept at 1 if you don't want multiple materials being used.
- The decal size is the orthographic size of the projector, this effect how large the decal is in the world.
- The override tag is used to place decals differently depending on the tag of the object that is hit. If left blank this will be the default decal to place. If multiple fields are left blank the first one will be the default and if none are left blank the first one will also be used. This allows you to have different decal bullet holes on different tagged objects.
- The decal depth changes the far clipping plane of the projector, this can be adjusted based on how thick the surface hit is. A low depth will have decals fading away quicker and a high depth will make them fade slower. This is useful in case your decals can be seen on the other side of objects or they are fading away too soon on very curvy objects.



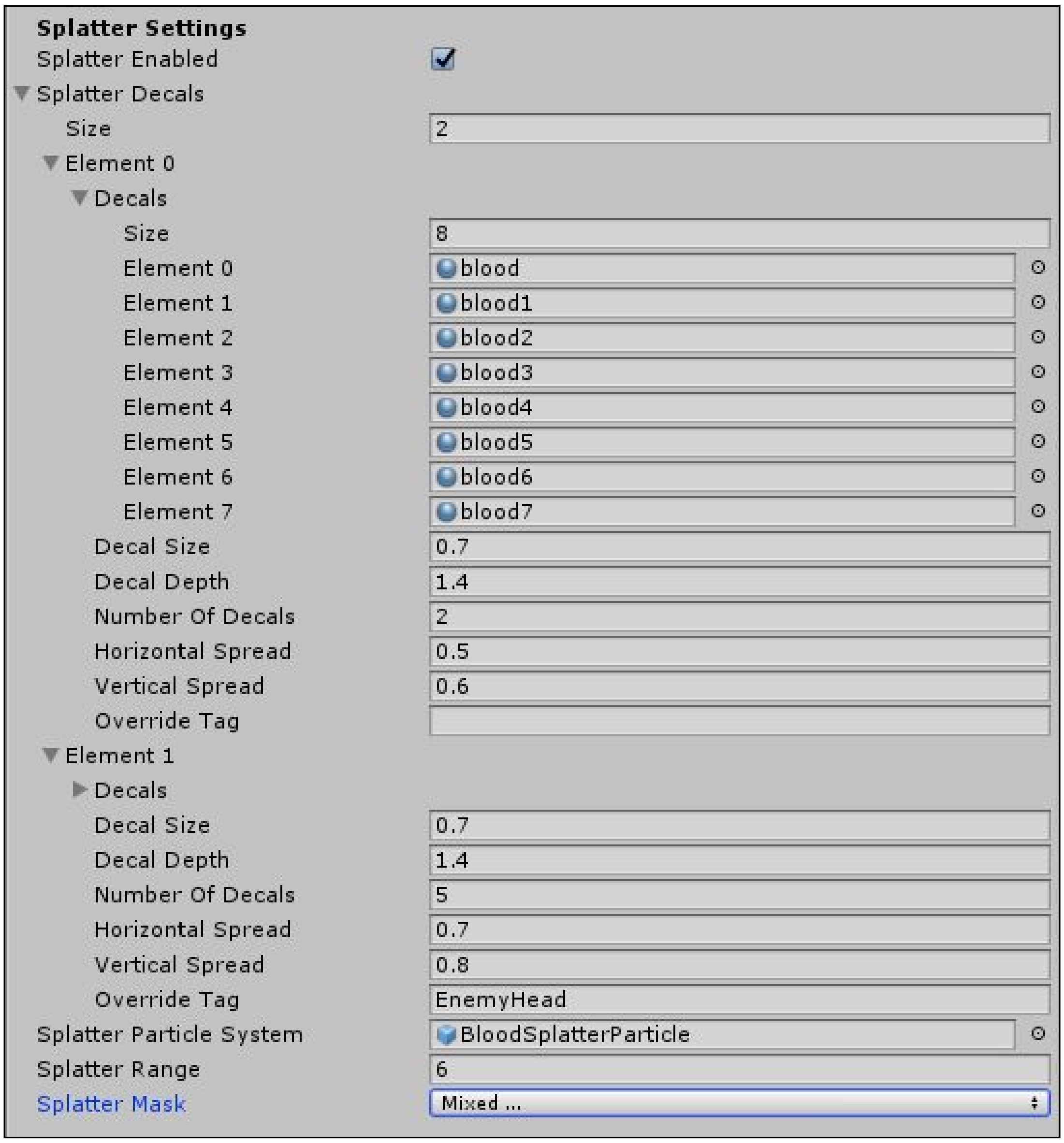
Decal Settings

▼ Decals	
Size	3
▼ Element 0	
▼ Decals	
Size	2
Element 0	SpriteDecalMat1-1
Element 1	SpriteDecalMat1-2
Decal Size	0.07
Override Tag	
Decal Depth	1.3
▼ Element 1	
▼ Decals	
Size	1
Element 0	SplatterDecalMat1-1
Decal Size	0.07
Override Tag	EnemyHead
Decal Depth	1
▼ Element 2	
▼ Decals	
Size	1
Element 0	SplatterDecalMat1-2
Decal Size	0.07
Override Tag	EnemyBody
Decal Depth	1.3

Setting up splatter decals:

Splatter decals can be set up in the inspector from the DecalSystem class or any class that inherits from it.

- Splatter decals work in very much the same way as the normal decals but there is a few more settings to customise.
- The number of decals is how many splatter decals appear with each event. This can make the amount of splatter seem more or less without having to change the sprites themselves.
- The vertical and horizontal spread is how much random range for the distance between splatter decals. This can be changed to make decals take up more space or be more dense.
- Whether splatter decals are on can also be toggled.
- This is where a splattering particle system is placed if you are using one.



Splatter Settings	
Splatter Enabled	✓
▼ Splatter Decals	
Size	2
▼ Element 0	
▼ Decals	
Size	8
Element 0	blood
Element 1	blood1
Element 2	blood2
Element 3	blood3
Element 4	blood4
Element 5	blood5
Element 6	blood6
Element 7	blood7
Decal Size	0.7
Decal Depth	1.4
Number Of Decals	2
Horizontal Spread	0.5
Vertical Spread	0.6
Override Tag	
▼ Element 1	
▶ Decals	
Decal Size	0.7
Decal Depth	1.4
Number Of Decals	5
Horizontal Spread	0.7
Vertical Spread	0.8
Override Tag	EnemyHead
Splatter Particle System	BloodSplatterParticle
Splatter Range	6
Splatter Mask	Mixed ...

Setting up bleed effects:

- The bleed effect can be toggled on and the layers that are able to spawn a particle can be selected.
- A bleed particle effect should be placed in the available field. A default one is provided for you in the prefabs folder.
- A bleed chance slider will change the percentage chance of a bleed particle spawning on hit.

Using the FPS functionality:

- The system also comes with some basic FPS functionality.
- The ShootingScript has code for both raycast and ballistic based shooting. Simply place this script on your player.
- The ballistic settings are found on the projectile behaviour script which should be place on any bullet projectiles you want to use.
- The projectile behaviour script has speed and gravity settings as well as all the settings from the decal system (it inherits from DecalSystem). These settings will be transferred over automatically from the shooting script providing the projectile prefab is placed in the ballistic shooting settings on the shooting script.

Setting up the Pool Manager:

- The system makes use of a basic pool manager.
- This is a prefab and should be placed in the scene before anything else can be done.
- Inside the pool manager in the the inspector, place the DecalProjector prefab (or a custom projector) and set the time each decal will remain in the scene and the max number that can exist in the scene at once.
- This system is rather basic and could be upgraded further if needs be.

The future of this project:

Currently this project exists as a free Unity store asset and an open source project on Github. Going forward there are plans to improve the current systems and add new ones to make the experience of using the asset easier for developers. The asset will remain free to use both price wise and commercially as the goal of this project was not to make money but to share an expanding system with other developers.

If you would like to contribute to this project you can find the open source version here: https://github.com/LKempton/ProjectorBasedDecalSystemInUnity