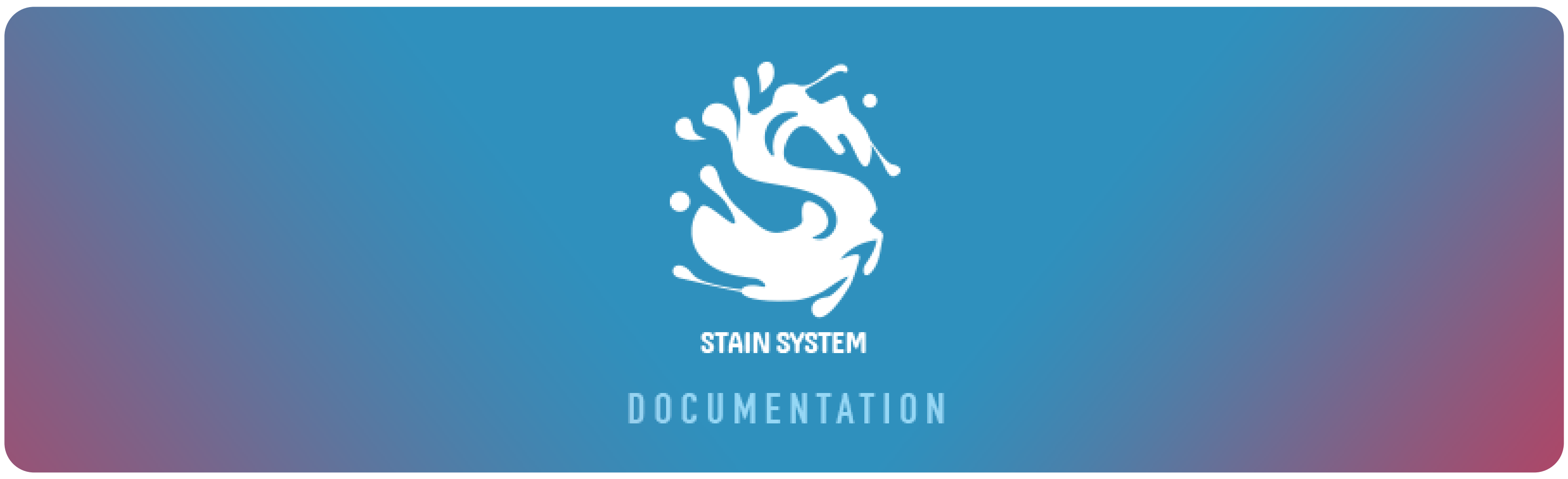
Stain System. Online Manual (https://dustyroom.com/stain-system-online-manual/)

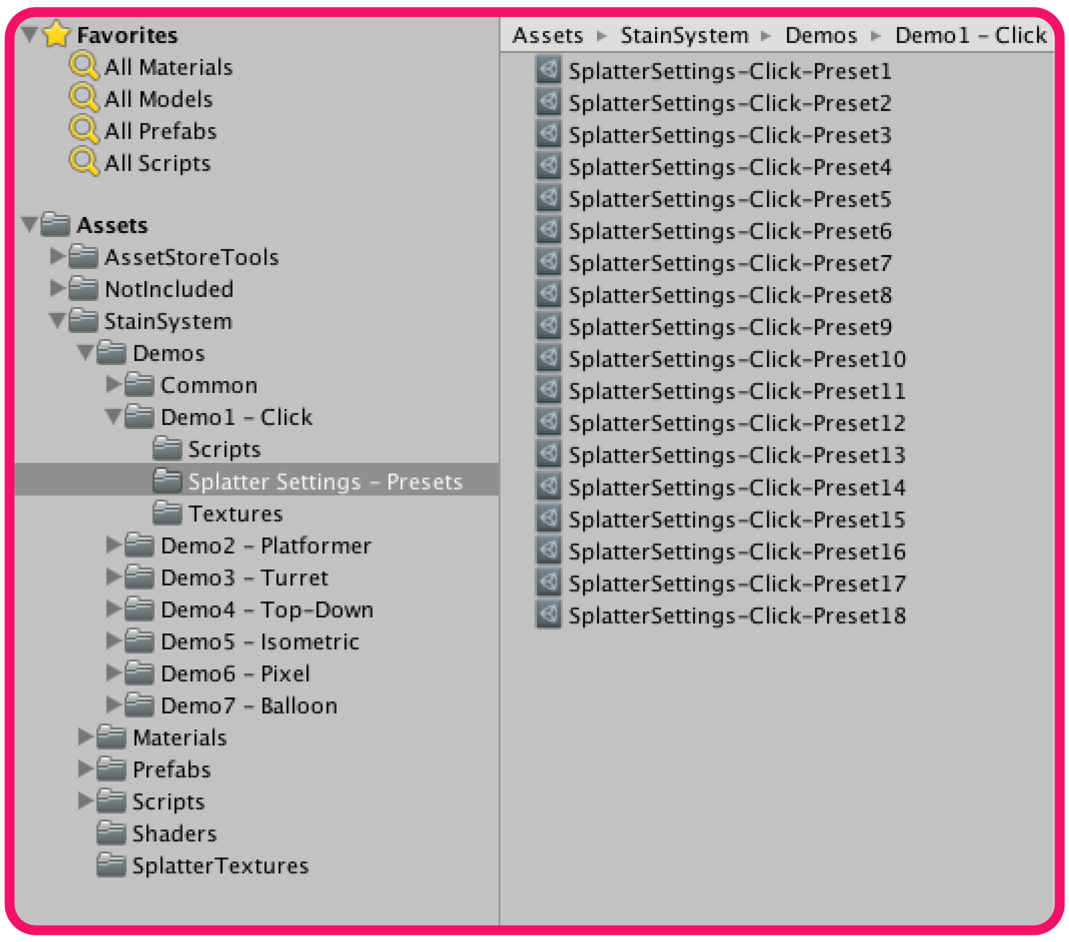


We strongly recommend using the online version of the manual, as it is always up-to-date:

dustyroom.com/stain-system-online-manual/ (https://dustyroom.com/stain-system-online-manual/)

Overview

Stain System is an asset that allows to add splatter effects like paint, dirt or blood to Unity games. The asset includes scripts, shaders, textures and prefabs as well as demo scenes and presets that show how to use Stain System.



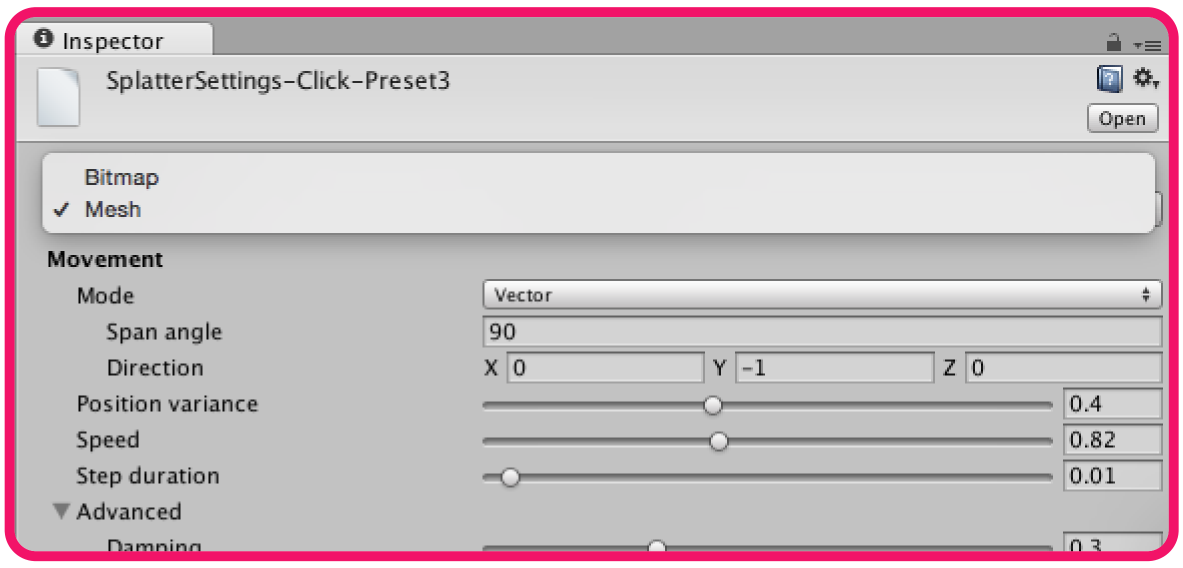
```
Favorites
  All Materials
  All Models
  All Prefabs
  All Scripts

Assets
  AssetStoreTools
  NotIncluded
  StainSystem
    Demos
      Common
      Demo1 - Click
        Scripts
        Splatter Settings - Presets
        Textures
      Demo2 - Platformer
      Demo3 - Turret
      Demo4 - Top-Down
      Demo5 - Isometric
      Demo6 - Pixel
      Demo7 - Balloon
    Materials
    Prefabs
    Scripts
    Shaders
    SplatterTextures
```

Assets ▸ StainSystem ▸ Demos ▸ Demo1 - Click

```
SplatterSettings-Click-Preset1
SplatterSettings-Click-Preset2
SplatterSettings-Click-Preset3
SplatterSettings-Click-Preset4
SplatterSettings-Click-Preset5
SplatterSettings-Click-Preset6
SplatterSettings-Click-Preset7
SplatterSettings-Click-Preset8
SplatterSettings-Click-Preset9
SplatterSettings-Click-Preset10
SplatterSettings-Click-Preset11
SplatterSettings-Click-Preset12
SplatterSettings-Click-Preset13
SplatterSettings-Click-Preset14
SplatterSettings-Click-Preset15
SplatterSettings-Click-Preset16
SplatterSettings-Click-Preset17
SplatterSettings-Click-Preset18
```

When developing Stain System we tried more *than 10 different ways* to perform a simple task – render as many as possible particles on a surface without overflowing into neighbour spaces. We have come up with a robust and flexible solution which is a combination of two techniques – using **bitmap textures** or a **single mesh** with custom shaders to draw splatters. While this allows to adapt the system to most applications and assures optimal performance, Stain System is not a simple drag-and-drop asset. Using it requires a selecting one of the two rendering methods and in some cases using a provided material on your game assets.

Stain System includes **two major** components that perform the same task of generating and rendering splatters in different ways:



SplatterSettings-Click-Preset3

Bitmap
✓ Mesh

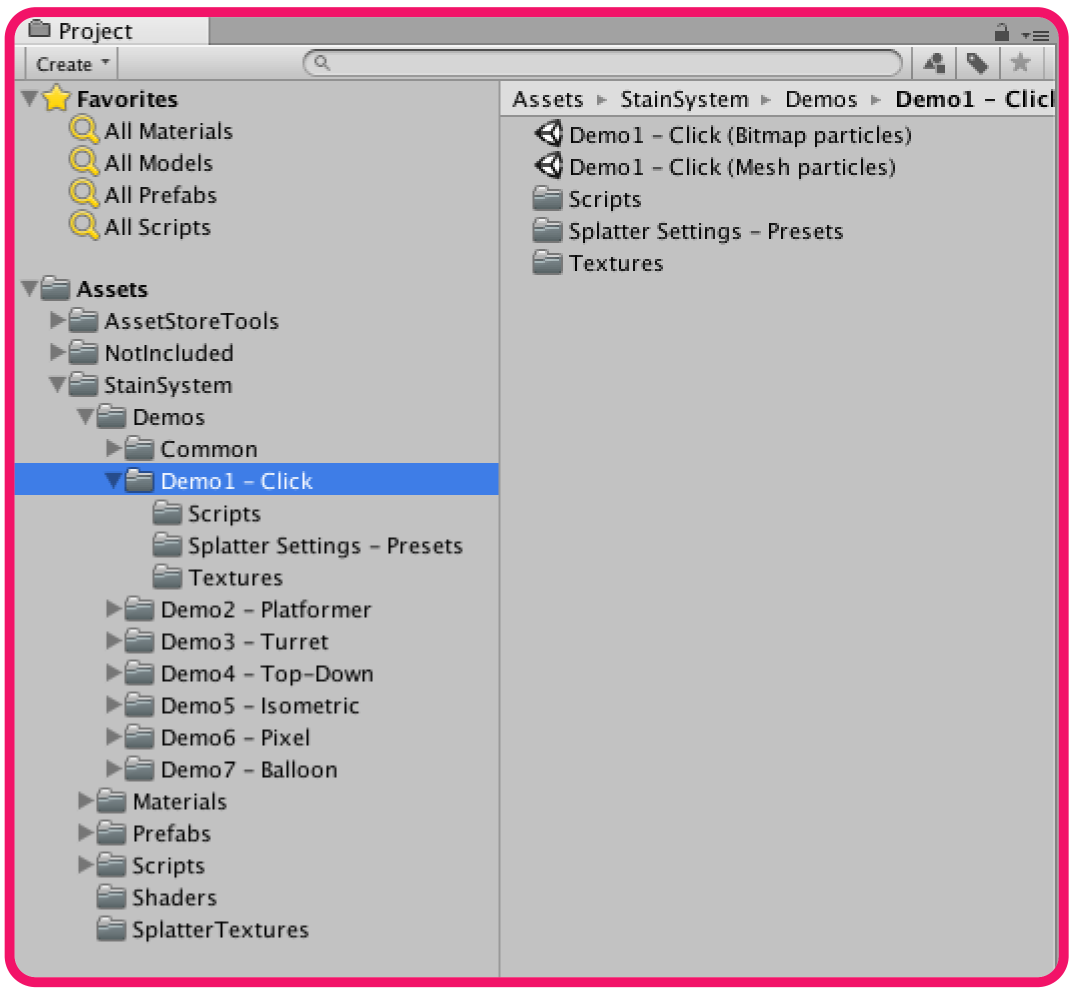
Movement

Mode	Vector
Span angle	90
Direction	X 0 Y -1 Z 0
Position variance	0.4
Speed	0.82
Step duration	0.01
▼ Advanced	
Damping	0.3

1. **Bitmap splatters** – use textures to render particles. This approach allows to generate infinite number of splatters without ever slowing down, but it allocates some texture memory in the areas where splatters may be spawned. Here are advantages and disadvantages of this approach:
    - + Infinite number of particles on any platform.
    - + Can be used on moving or static objects.
    - – Currently, only works in 2D.
    - – Uses texture memory to generate splatter areas. Amount of memory is discussed in a section below.
    - – Paint is not shared among the pre-positioned splatter-receiving areas touch or overlap.
    - – Once splatters are spawned, they cannot be individually removed.
2. **Mesh splatters** – use a single dynamic mesh to render particles. This approach allows to easily generate lots of particles, but the number of particles is limited. That said, this limit is way higher than most (even mobile) games require. Internally, this approach uses Unity's build-in particle system to generate the mesh because it's the only way to access and modify vertices with zero memory allocation. Here are pros and cons of this approach:
    - + Works in 2D, 3D and isometry with a limitation of being planar (doesn't simulate physical behaviour on corners).
    - + In addition to the normal alpha blending, it's possible to use multiply blending.
    - + Anti-aliasing is possible and does not require any additional set-up.
    - + Splatters can be tweened out after a time-out.
    - – Number of particles is bound by GPU. Normally, it handles well up to 10K particles on old mobiles and more than 20K on desktop.
    - – Splatters cannot be moved with dynamic objects.

Once you have decided on the approach that best suits your game, you can follow the instructions in one of the sections below. Note that bitmap and mesh splatters can be mixed together in the same scene.

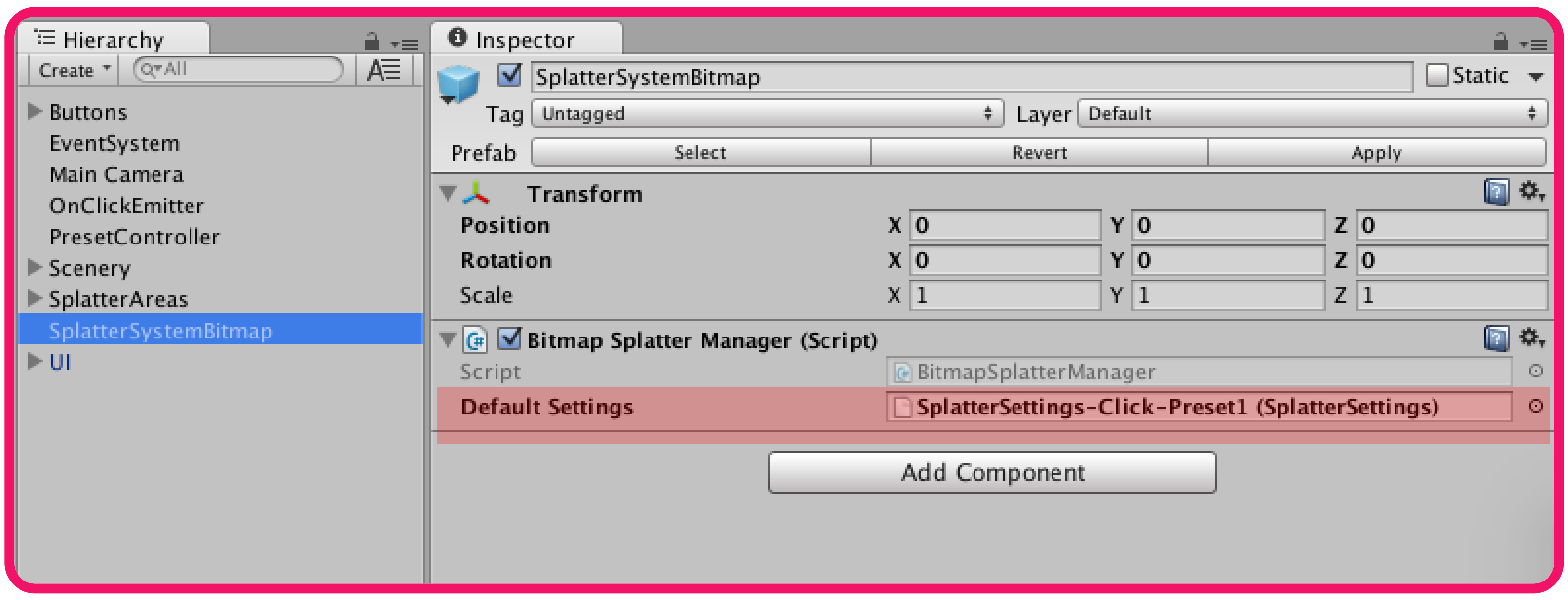
Note. The Demo Scenes included in the Asset package are represented in both Bitmap and Mesh modes.



Bitmap Splatters

Here is how to use bitmap splatters:

1. Drop **Prefabs/BitmapSplatterArea** in your scene and place it over an area where splatters may be spawned. You can resize this object as you like. Duplicate it to cover all such areas, but try to keep them tight since each area is occupying texture memory equal to its size. If the areas where splatters should be visible are not rectangular, the effect can be rendered based on objects behind the area. In this case you should use the **BitmapSplatterAreaSelective** prefab instead of **BitmapSplatterArea**. This also requires setting material of all sprites that are behind the area and can receive splatters to the included **Materials/SpriteSplatterSurface** material. Check out the **Demo2 – Platformer (Bitmap particles)** for an example of this case.
2. Drop the **Prefabs/SplatterSystemBitmap** into the scene. This object manages all the areas we added in the previous step.
3. Create new **SplatterSettings** object or select an existing one and assign it to the **Default Settings** field of the **Splatter Manager** component of **SplatterSystemMesh** Game Object (see screenshot below). Splatter Settings objects are described in detail in a section below.
4. To spawn a splatter you need to write code (or use assets such as Playmaker) that calls the **Spawn(Vector3 position)** method of the **Splatter Manager** component. You need to pass the position of the desired splat center in world coordinates. Additionally, you may pass a Splatter Settings object, a color, or a direction to override the default settings.



SplatterSystemBitmap
Tag Untagged
Layer Default
Prefab

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Bitmap Splatter Manager (Script)
Script BitmapSplatterManager
Default Settings SplatterSettings-Click-Preset1 (SplatterSettings)

Mesh Splatters

Here is how to use mesh splatters:

1. To see stains on a mesh you need to do either of:
2. Use the included **Materials/SpriteSplatterSurface** or **Materials/StandardSplatterSurface** on the painted sprite/mesh. You should set material of all sprites/objects that can receive splatters to one of those two materials. Check out the **Demo2 – Platformer (Mesh particles)** for an example of this case. Here is a quick explanation:
    - **Materials/SpriteSplatterSurface** is the same material as **Sprite-Default**, but can display splatter effect. You can use it instead of the standard sprite material in 2D games.
    - **Materials/StandardSplatterSurface** is the same material as **Standard**, but can display splatter effect. You can use it instead of the standard material in 3D/isometric games. 2.Set stencil buffer in your own shader to the same value as in the stain material (default is 128). This is done by adding the following code into the Pass section of the shader code:

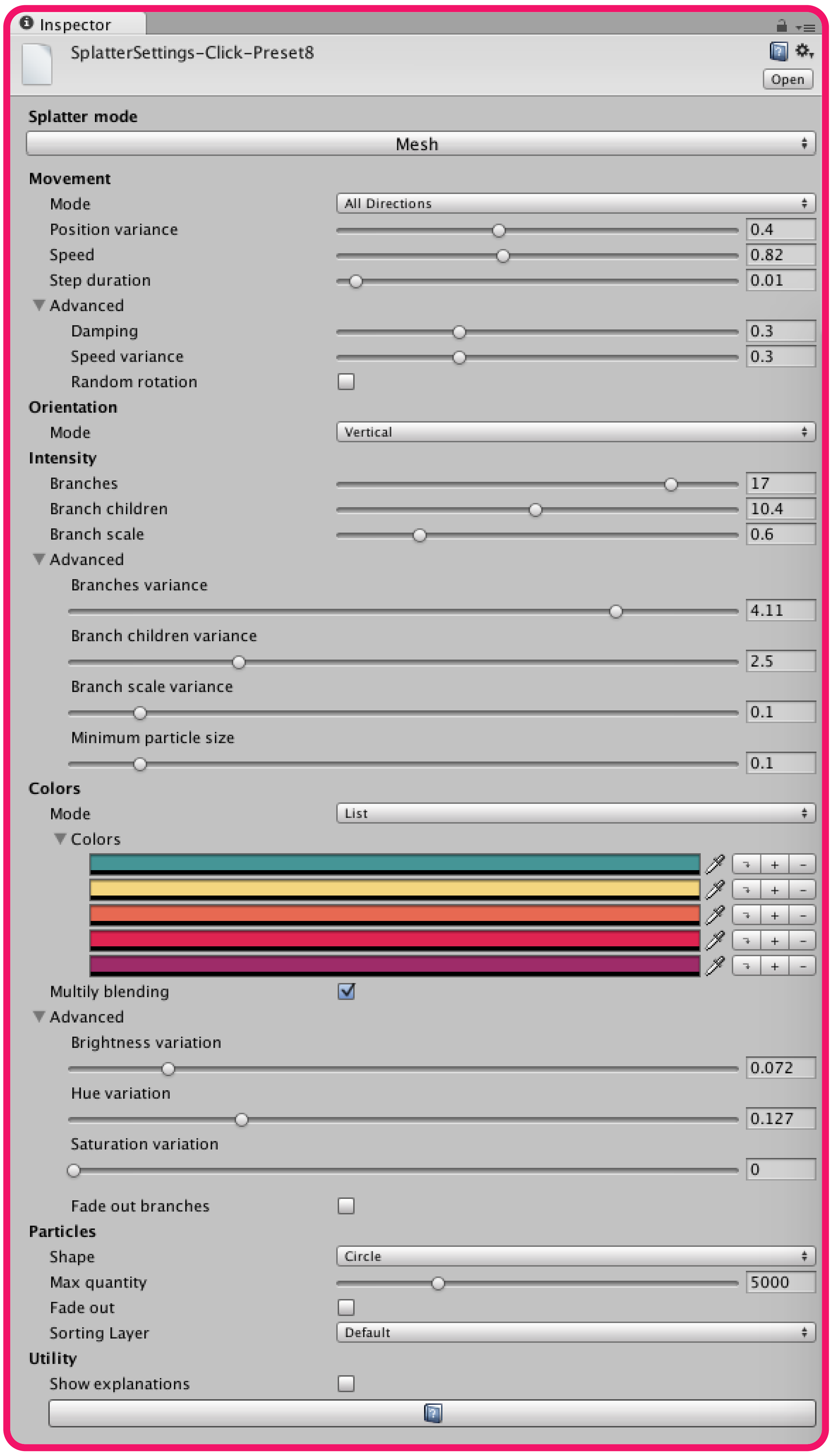
```
Stencil {
Ref [_StencilRef]
ReadMask 128
WriteMask 128
Comp Equal
}
```

1. The effect is always rendered only on object that use one of the provided shaders. For convenience, there are also materials that already have the shaders assigned.
2. Drop the **Prefabs/SplatterSystemMesh** into the scene. This object manages all the splatter particles in the scene.
3. Create new **SplatterSettings** object or select an existing one and assign it to the **Default Settings** field of the **Splatter Manager** component of **SplatterSystemBitmap** Game Object. Splatter Settings objects are described in detail in a section below.
4. To spawn a splatter you need to write code (or use assets such as Playmaker) that calls the **Spawn(Vector3 position)** method of the **Splatter Manager** component. You need to pass the desired position of splat center in world coordinates. Additionally, you may pass a Splatter Settings object, a color, or a direction to override the default settings.

Hint. If you want more splatters at lower resources cost – use the complex texture that already contain the variety of "splatter debris". Use either the ones that come with the Stain System asset, or prepare custom splatter textures. See Example – Click Scene – Presets 2, 8.

Splatter Settings

Splatter Settings is a reusable configuration file that describes how splatters behave and look. To create such file right-click anywhere in the Project window and select **Create ▶ SplatterSettings**. A new file will appear in your project that can be dropped into the **Default Settings** field of **SplatterSystemBitmap** or **SplatterSystemMesh** or when calling the **Spawn** method on either of these components. Explanation of each parameter is included in the Inspector view of the settings, to view them, check the **Show explanation** box at the bottom of the settings inspector.



SplatterSettings-Click-Preset8

Splatter mode: Mesh

Movement
- Mode: All Directions
- Position variance: 0.4
- Speed: 0.82
- Step duration: 0.01
- Advanced
  - Damping: 0.3
  - Speed variance: 0.3
  - Random rotation: ☐

Orientation
- Mode: Vertical

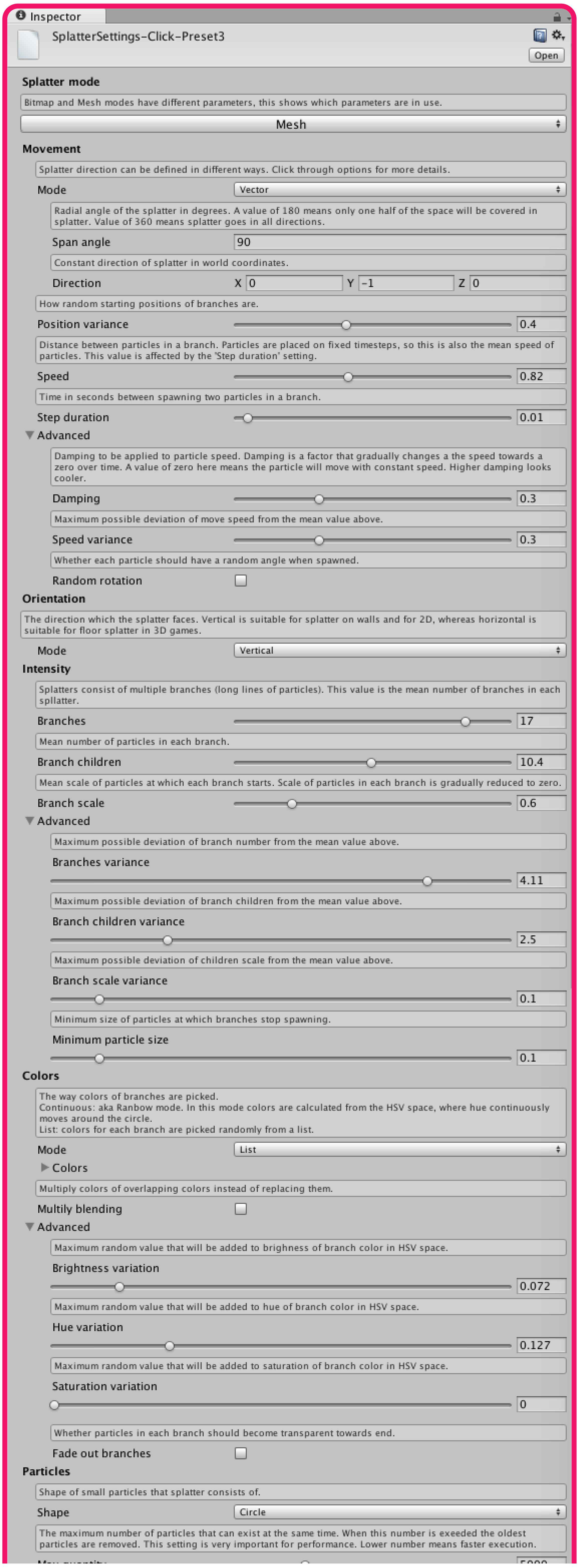
Intensity
- Branches: 17
- Branch children: 10.4
- Branch scale: 0.6
- Advanced
  - Branches variance: 4.11
  - Branch children variance: 2.5
  - Branch scale variance: 0.1
  - Minimum particle size: 0.1

Colors
- Mode: List
- Colors
  - 🟦
  - 🟨
  - 🟧
  - 🟥
  - 🟪
- Multily blending: ✓
- Advanced
  - Brightness variation: 0.072
  - Hue variation: 0.127
  - Saturation variation: 0
  - Fade out branches: ☐

Particles
- Shape: Circle
- Max quantity: 5000
- Fade out: ☐
- Sorting Layer: Default

Utility
- Show explanations: ☐

Below is the screenshot of the **Splatter Manager** Inspector Panel with **Show Explanations** engaged.


Inspector

SplatterSettings-Click-Preset3

Open

Splatter mode

Bitmap and Mesh modes have different parameters, this shows which parameters are in use.

Mesh

Movement

Splatter direction can be defined in different ways. Click through options for more details.

Mode: Vector

Radial angle of the splatter in degrees. A value of 180 means only one half of the space will be covered in splatter. Value of 360 means splatter goes in all directions.

Span angle: 90

Constant direction of splatter in world coordinates.

Direction: X 0 Y -1 Z 0

How random starting positions of branches are.

Position variance: 0.4

Distance between particles in a branch. Particles are placed on fixed timesteps, so this is also the mean speed of particles. This value is affected by the 'Step duration' setting.

Speed: 0.82

Time in seconds between spawning two particles in a branch.

Step duration: 0.01

▼ Advanced

Damping to be applied to particle speed. Damping is a factor that gradually changes a the speed towards a zero over time. A value of zero here means the particle will move with constant speed. Higher damping looks cooler.

Damping: 0.3

Maximum possible deviation of move speed from the mean value above.

Speed variance: 0.3

Whether each particle should have a random angle when spawned.

Random rotation: ☐

Orientation

The direction which the splatter faces. Vertical is suitable for splatter on walls and for 2D, whereas horizontal is suitable for floor splatter in 3D games.

Mode: Vertical

Intensity

Splatters consist of multiple branches (long lines of particles). This value is the mean number of branches in each spllatter.

Branches: 17

Mean number of particles in each branch.

Branch children: 10.4

Mean scale of particles at which each branch starts. Scale of particles in each branch is gradually reduced to zero.

Branch scale: 0.6

▼ Advanced

Maximum possible deviation of branch number from the mean value above.

Branches variance: 4.11

Maximum possible deviation of branch children from the mean value above.

Branch children variance: 2.5

Maximum possible deviation of children scale from the mean value above.

Branch scale variance: 0.1

Minimum size of particles at which branches stop spawning.

Minimum particle size: 0.1

Colors

The way colors of branches are picked.
Continuous: aka Ranbow mode. In this mode colors are calculated from the HSV space, where hue continuously moves around the circle.
List: colors for each branch are picked randomly from a list.

Mode: List

▶ Colors

Multiply colors of overlapping colors instead of replacing them.

Multily blending: ☐

▼ Advanced

Maximum random value that will be added to brighness of branch color in HSV space.

Brightness variation: 0.072

Maximum random value that will be added to hue of branch color in HSV space.

Hue variation: 0.127

Maximum random value that will be added to saturation of branch color in HSV space.

Saturation variation: 0

Whether particles in each branch should become transparent towards end.

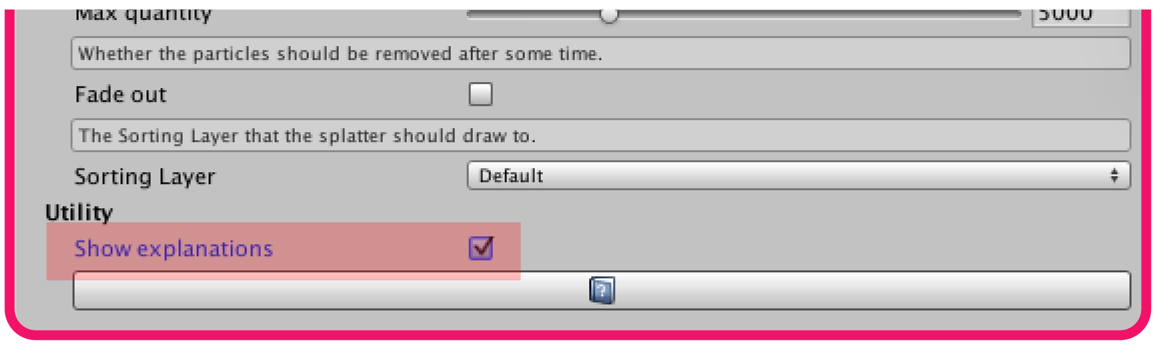
Fade out branches: ☐

Particles

Shape of small particles that splatter consists of.

Shape: Circle

The maximum number of particles that can exist at the same time. When this number is exeeded the oldest particles are removed. This setting is very important for performance. Lower number means faster execution.



Max quantity 5000

Whether the particles should be removed after some time.

Fade out ☐

The Sorting Layer that the splatter should draw to.

Sorting Layer: Default

Utility

Show explanations ✓