

# Pigeon Coop Toolkit - Trails

---



Our trail effects were created because the built in trail renderer lacked some key features. With our trail renderers you can clear the trail, check if the trail has any visible segments (nice to have when you want to wait for the trail to fade away before destroying an object!), specify your own 'up' direction for the trail (so you can have skid marks that are flat against the ground rather than billboarded to the camera) and you can even extend the code to have trails with custom behavior. The trails will billboard to the main camera by default or billboard in the direction specified.

Included is a standard trail renderer which is identical to the built in one but it implements the helpful features outlined above. As well as that, we include 2 special trails - the smoke trail and smoke plume scripts.

Possible smoke trail uses: a trail of smoke behind an object, showing the path of a fast moving bullet (like a sniper rifle or machine gun), laser, etc

Possible smoke plume uses: A plume of smoke from a fire, "Diablo"-y flame eyes, gun barrel smoke after weapon fire, etc.

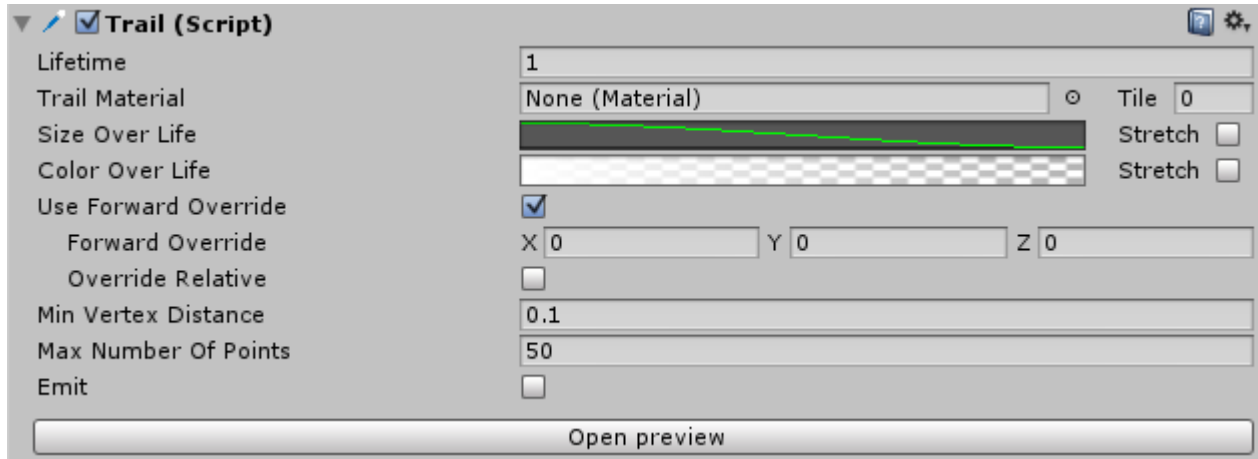
## Contents

Components.....	2
Trail Component .....	2
Smoke Trail Component.....	3
Smoke Plume Component .....	4
Smooth Trail Component.....	5
Preview Utility.....	6
Simple Input .....	7
Code Reference.....	8
Shared between all components .....	8

# Components

## Trail Component

Find it under Component → Pigeon Coop Toolkit → Effects → *Trail*

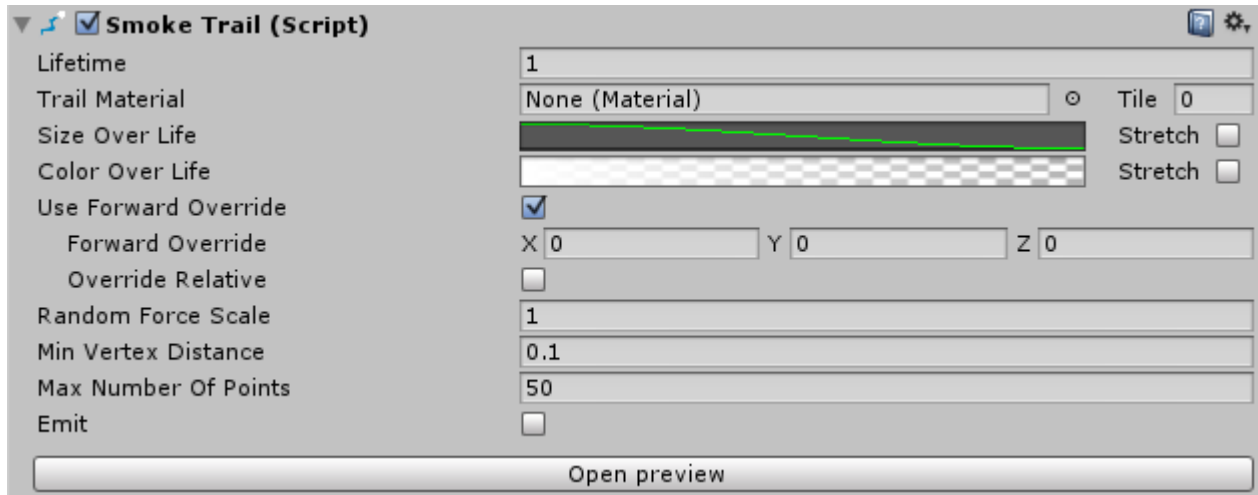


### Properties

<b>Lifetime</b>	Life of the trail, measured in seconds.
<b>Trail Material</b>	A material used for rendering the trail. Particle shaders work the best for trails.
<b>Tile</b>	Adjusts how often the materials texture tiles along the trail. If set to 0, the materials texture will stretch from the start of the trail to the end.
<b>Size Over Lifetime</b>	A curve representing the trails size over its lifetime.
<b>Color Over Lifetime</b>	A gradient representing the trails color over its lifetime.
<b>Stretch</b>	Stretches the color/size from the start of the trail to the end.
<b>Use Forward Override</b>	Instead of using the cameras 'forward' to calculate billboarding, you can specify your own vector, for example, specify "up" for your trail to billboard against the ground regardless of the viewing angle.
<b>Forward Override</b>	The override vector.
<b>Override Relative</b>	If this is ticked, forward override vector is treated as a local vector and is transformed into world space. Check the sword scene for a good example. (Any of the sword trails)
<b>Min Vertex Distance</b>	The minimum distance between anchor points of the trail.
<b>Max Number Of Points</b>	Maximum number of points in a trail. Any new points after this will override the oldest.
<b>Emit</b>	Whether to emit or not.

## Smoke Trail Component

Find it under Component → Pigeon Coop Toolkit → Effects → *Smoke Trail*



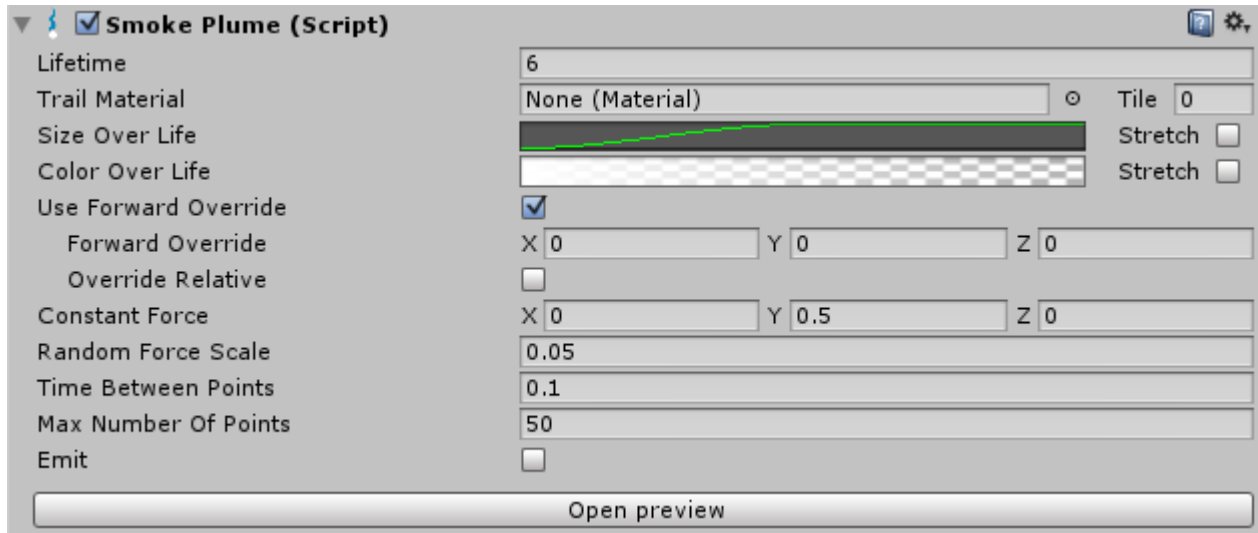
### Unique Properties

#### Random Force Scale

Each point along the trail will start drifting along a random direction to give the trail a 'smokey' feel. This value controls the scale of that effect is.

## Smoke Plume Component

Find it under Component → Pigeon Coop Toolkit → Effects → *Smoke Plume*



### Unique Properties

#### Constant Force

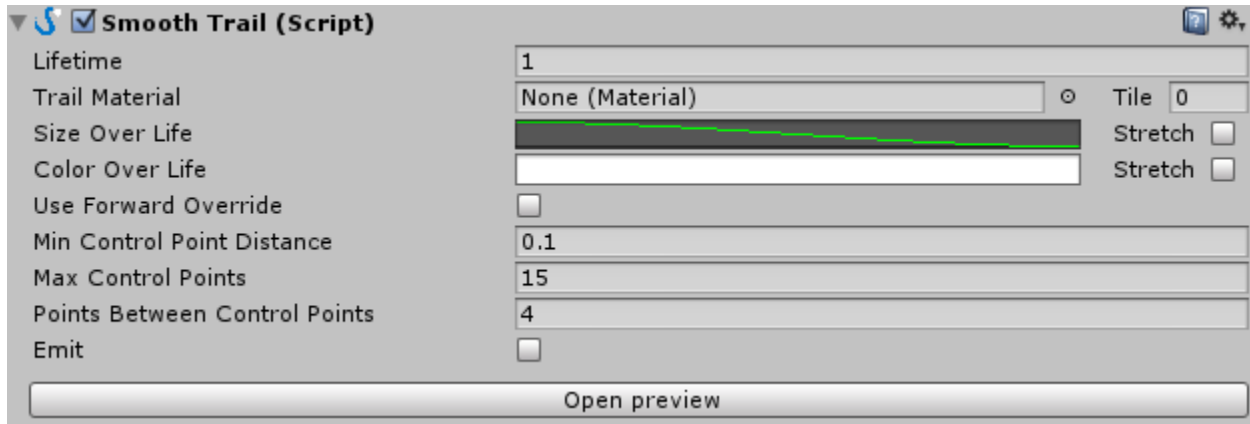
A constant force to apply to all points in the trail. The default value (0, 0.5, 0) gives the impression of a rising smoke column.

#### Time Between Points

Instead of Min Vertex Distance, this type of trail adds a new point to the trail every so often. This value specifies the time between new points.

## Smooth Trail Component

Find it under Component → Pigeon Coop Toolkit → Effects → *Smooth Trail*

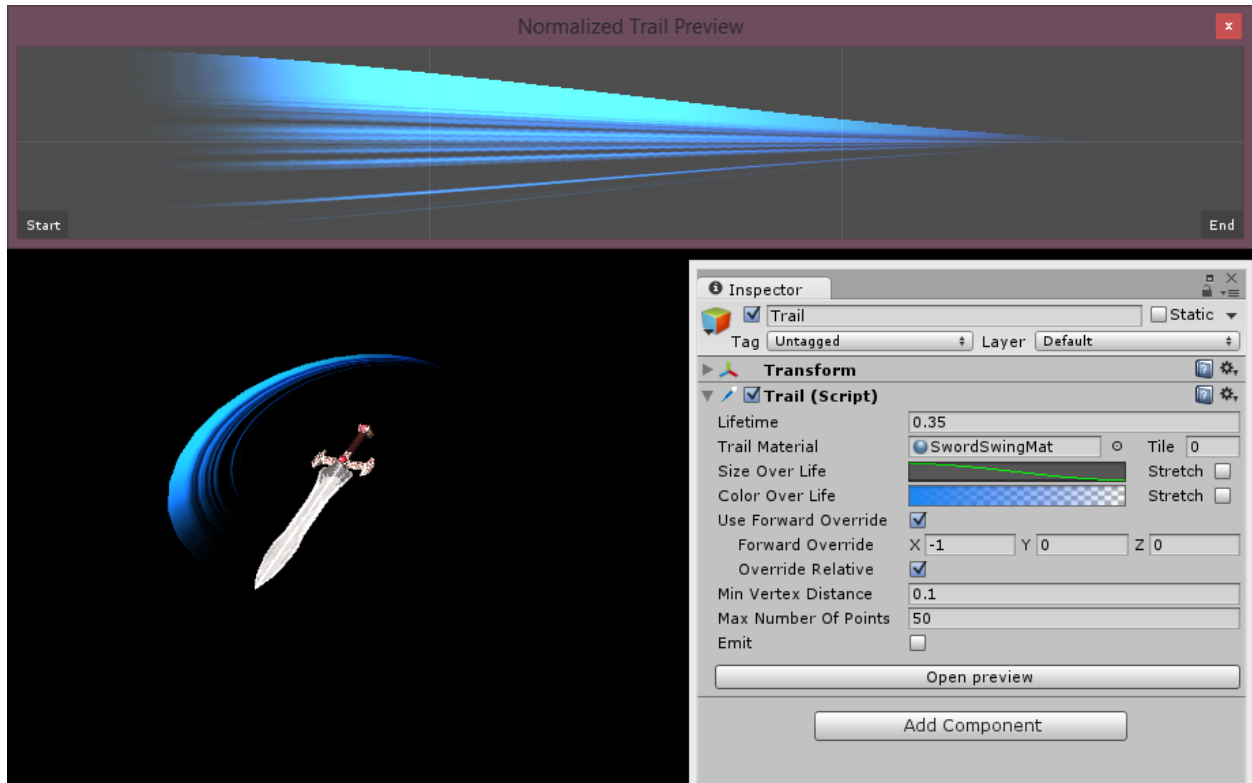


### Unique Properties

<b>Min Control Point Distance</b>	The minimum distance between anchor points of the trail.
<b>Max Control Points</b>	Works the same as “max points” for a regular trail.
<b>Points Between Control Points</b>	Effects ‘smoothness’ of the trail. This value tells you how many ‘segments’ there will be between your control points. The more points you have between control points, the smoother the result can be. However, this will greatly increase the complexity of your trail. Try to start with small values and work your way up.

## Preview Utility

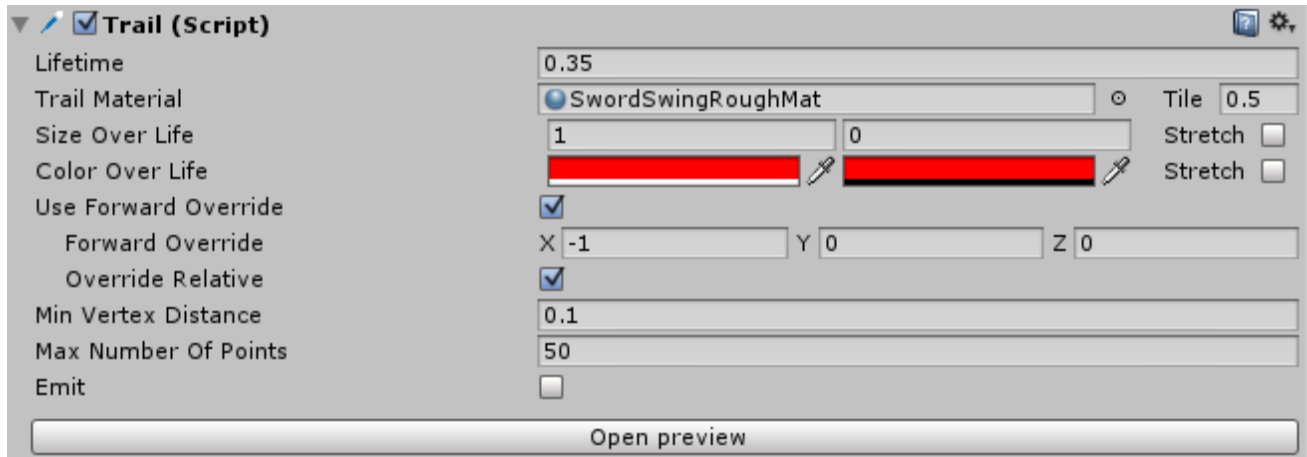
Access it by clicking on *Open Preview* on any of the trail component inspectors.



The preview utility is intended to provide you with some visual feedback as you set up your trails. The previewer shows you a general preview of what your trail looks like, though your trail might look very different in game as it moves around the scene.

## Simple Input

Access it by right click on the Trail inspector and click on the on **Toggle** options in the inspector.



With simple input, it is quicker to specify the look and feel of your trail. Size over life time is reduced to a start size and an end size instead of a curve. Similarly, the color over life time is reduced to 2 color fields instead of a gradient.

## Code Reference

### Shared between all components

`void CreateTrail(Vector3 from, Vector3 to, float distanceBetweenPoints)`

Insert a trail into this renderer.

`void ClearSystem(bool emitState)`

Clears all the particles and sets new emit state.

`void Translate(Vector3 t)`

Moved (translates) the entire active trail by 't'. Useful for teleporting objects.

`int NumSegments()`

Returns the number of trail segments currently active in this renderer. This is a useful function for cleaning up – once you stop emitting the trail, you check the number of segments. If it is not 0, it means that the trail is still fading out. You can wait for this to return 0 before you destroy the trail so the trail doesn't 'pop out of existence' when you destroy it.