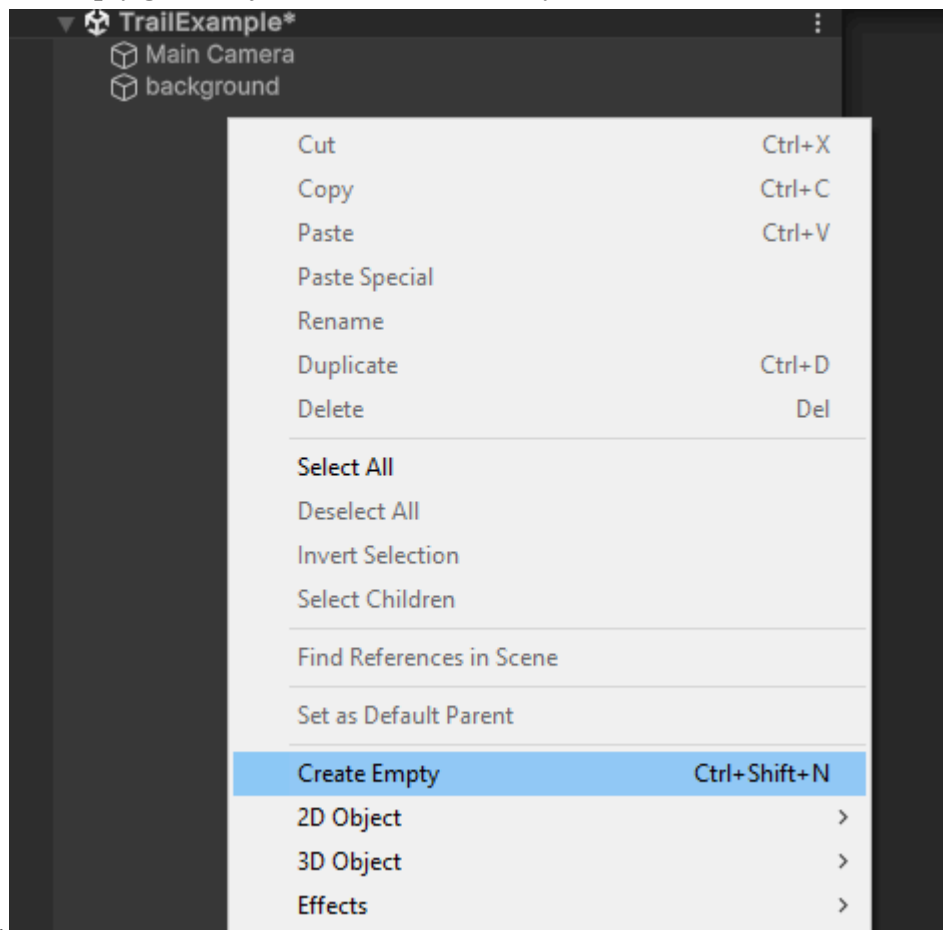


***Important: Please avoid touching anything in my Resources folder unless you are just moving them to another Resources folder, thank you <(_ _)>.**

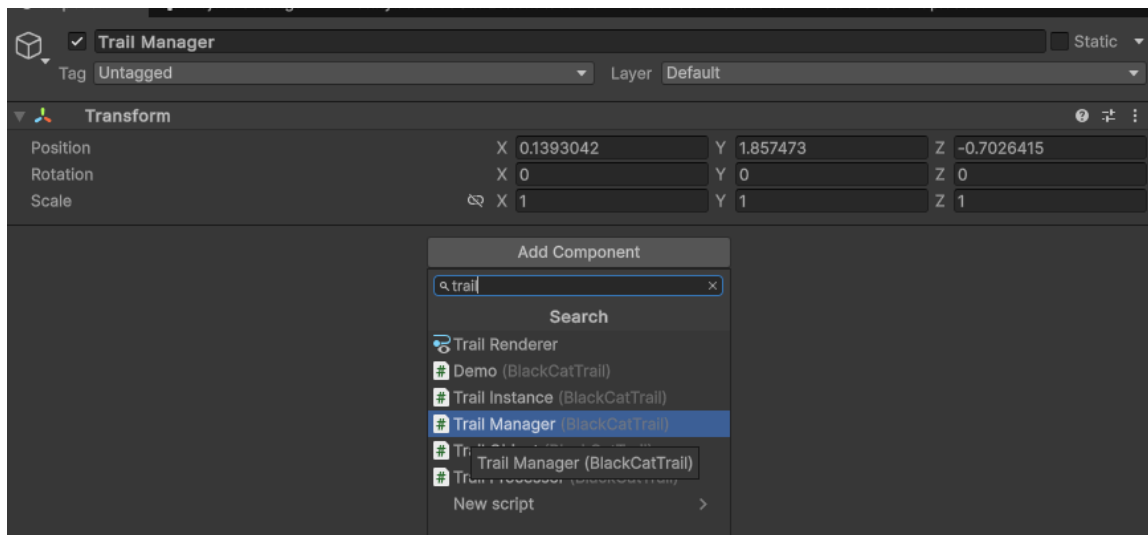
Set up

Before you can use the trails, you need to set some things up.

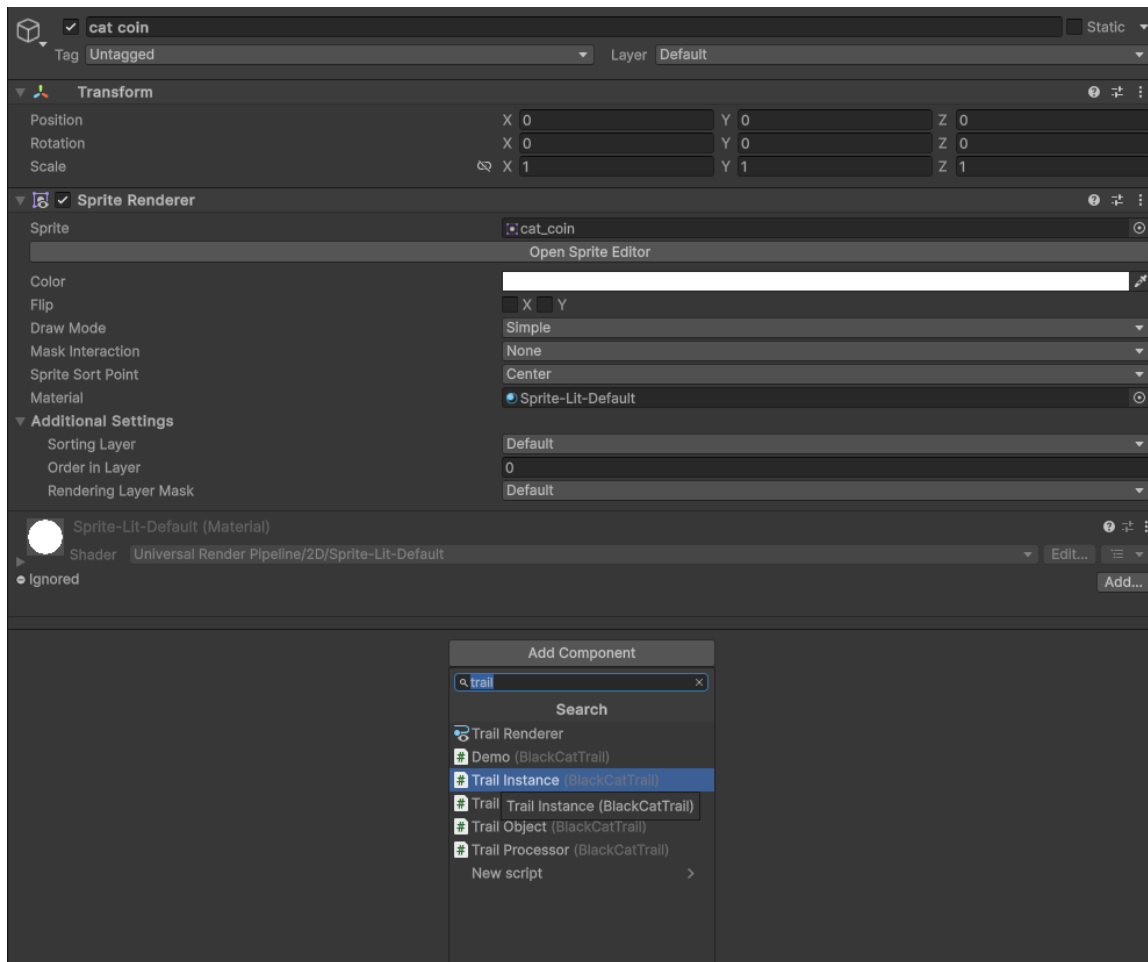
1. Create an empty game object, name it whatever you want, and attach the “Trail Manager”

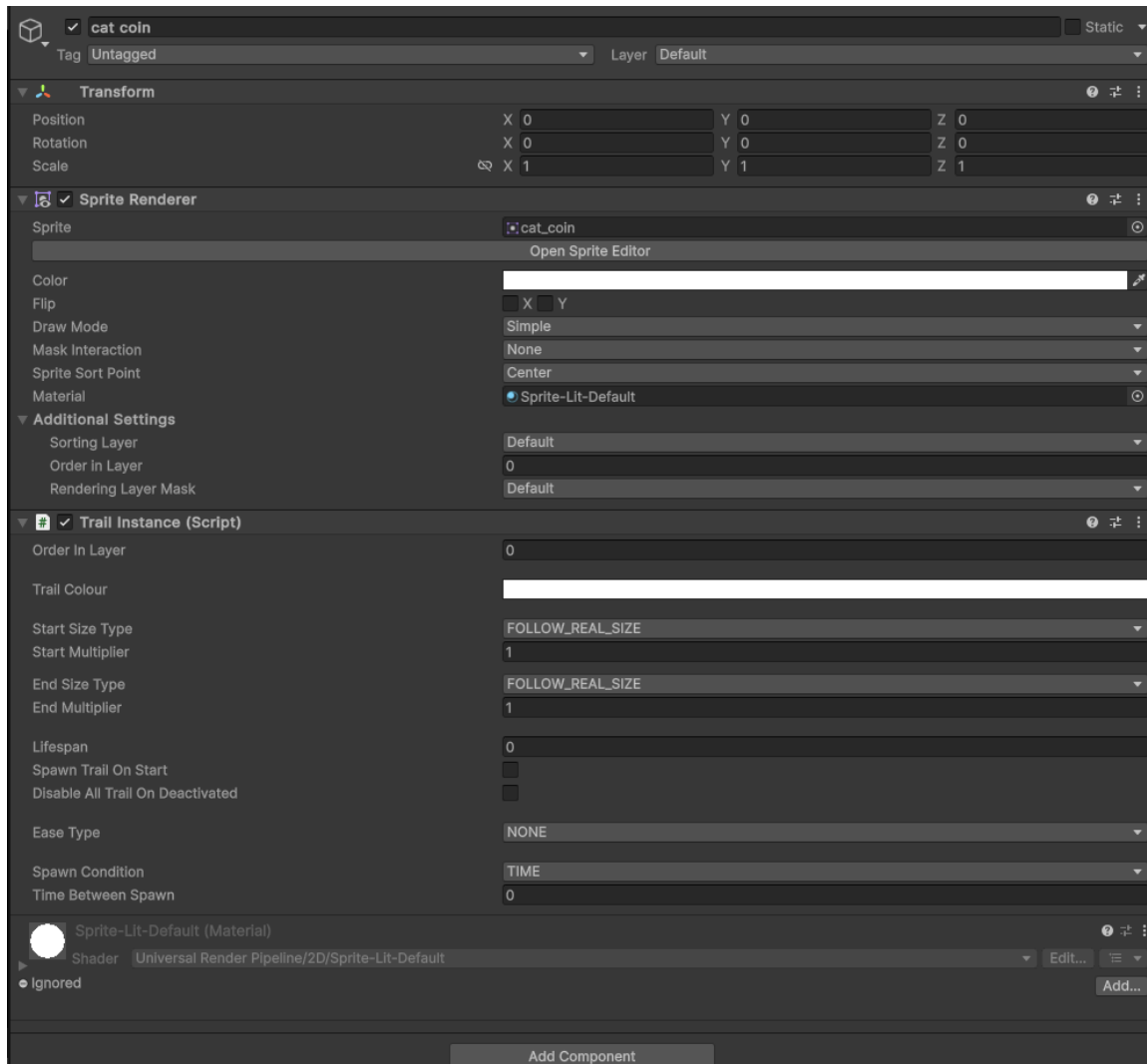


script.



2. In the game object that you want to spawn trails of, attach the “Trail Instance” script.





3. Configure the Trail Instance of your desire. You should be able to read the tooltips if you hover to each field, but if you want more descriptions, see **Field Explain-Trail Instance**.

Use

Start A Trail

At the place of a script where you want to start a trail, add the following line:

```
BlackCatTrail.TrailManager.Instance.StartTrail(gameObject);
```

“gameObject” is literally the game object your script is attached to. If you want to start the trail on another game object, simply pass that game object to the bracket as the parameter instead.

Note that starting a new trail on a game object that is already spawning trails will immediately stop that trail and start the new trail.

Stop A Trail

At the place of a script where you want to stop a trail, add the following line:

```
BlackCatTrail.TrailManager.Instance.StopTrail(gameObject);
```

Just like StartTrail, “gameObject” is the game object your script is attached to. If you want to stop the trail of another game object, simply pass that game object to the bracket as the parameter instead.

Passing a game object that is not spawning any trails will not give you any errors.

Check If A Game Object Is Spawning A Trail

If you want to check if a game object is spawning a trail, use this line:

```
BlackCatTrail.TrailManager.Instance.IsObjectSpawningTrails(gameObject);
```

Destroy Some Inactive Trail Objects

This trail manager uses object pooling, which might cause a lot (and I mean A LOT!) of inactive trail objects being pooled in the scene.

If you find it too disturbing, the manager can automatically destroy some of them for you if too much time has passed since the last trail ended, but you can also call this function to destroy them manually:

```
BlackCatTrail.TrailManager.Instance.DestroyTrail(int);
```

“int” is an integer that represents the number of inactive trail objects you wish to destroy.

Fields Explain

Trail Manager

- **Pf_blank Trail:** The prefab of trail objects. If this is empty, go to “TheBlackCat/2D Trail/Scripts/Trail” and attach the "trail_object" prefab here.
- **Auto Clean Up:** Whether to allow the manager to automatically destroy inactive trail objects if too much time has passed since the last trail ended.
- **Clean Up After Seconds:** The time you want the manager to automatically clean up inactive trail objects since the last trail ended in seconds.
- **Number Of Trail Remains:** The minimum number of inactive trail objects you wish to remain after auto clean up.

Trail Instance

- **Order In Layer:** This sets the Order In Layer of the trails.
- **Trail Colour:** Or in American English, Trail Color. The colour gradient of the trails over time. This supports HDR colour (and is default using HDR colour).
- **Affected By Easing:** Whether the colour over time will be affected by the easing equation.
- **Start/End Size Type:** The size type of the starting/ending size.
 - - If "FOLLOW_REAL_SIZE" is selected, the scale will be set to be the world scale of the game object that is spawning the trail.
 - - if "FIXED_SIZE" is selected, the scale will be set to the fixed scale you provided.
- **Start/End Size:** The fixed scales. Only visible when the size type is set to “FIXED_SIZE”.
- **Start/End Multiplier:** Scale multipliers of the starting/ending sizes. Only visible when the size type is set to “FOLLOW_REAL_SIZE”.
- **EaseType:** The type of easing of colour and scale changes over time.
- **Power:** The exponent of the power easing types (quadratic, cubic, quartic, quintic). Only visible when the ease type is set to “EXPONENTIAL_EASE_OUT”,

“EXPONENTIAL_EASE_IN” or “EXPONENTIAL_EASE_IN_OUT”. This has a range of 2~5.

- **LifeSpan:** The amount of time a trail object will live in seconds.
- **Spawn Trail On Start:** Whether to spawn a trail object immediately after the trail starts.
- **Disable All Trail On Deactivated:** Whether to disable all trail objects of the game object immediately when the game object is disabled.
- **Spawn Condition:** The condition of spawning each trail object. You can choose to spawn over time or spawn over distance.
- **Distance/Time Between Spawn:** The distance or time between each trail object spawn. Note that spawning over distance does not mean it spawns after the game object has been moved over a total distance. It means to spawn as soon as the position of the game object and the last position spawning a trail object has exceeded a certain distance.

Possible Problems

- Due to inconsistency between each frame which causes the delta time to be imprecise, inconsistent gaps between each trail object are possible. I implemented some calculations to calculate the spawn positions as precisely as possible so that you will likely not notice it, but I can't guarantee the inconsistent gaps will never appear.
- This tool can automatically stop trails if the game objects are destroyed or disabled. However, I cannot guarantee that it always stops before object destruction. If you get into cases where it doesn't, simply stop the trails before destroying the game objects. But in general, it should work 99.9% of the time.