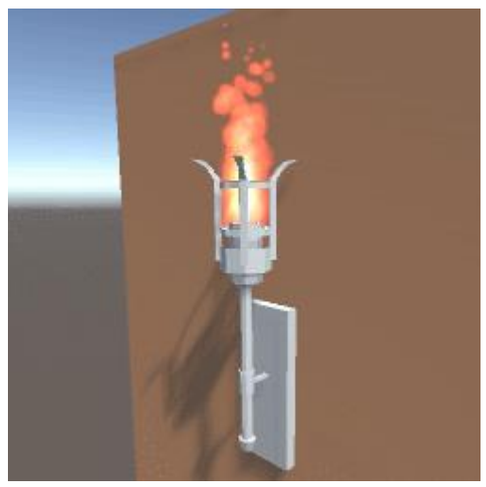
COMP 3064

Game Development

Week 11

Today:
Particles

- Particle System Component (Shuriken)
- MinMax Curves
- Animated Particles via Texture Sheets
- Scripting Particle Emissions



Particle System

- Any game object that has a **Particle System component** attached to it.
- *Game Object > Create Other … > Particle System*
- As a game object, it can **translate**, **rotate** and **scale**.

Particle System

- Specialized object used to create **particle-driven special effects**.
  - E.g. Rain, smoke, fire, sparks, magic and sci-fi effects, etc.
- Involves **many small elements** that come together in a larger system.
  - E.g. each drop of rain is an element of the rain effect.
- This results in a **swarm of individual effects** that add up to a particle effect.

Particle System

- Comprised of an **Emitter** and a **Particle**.

- *Emitter* — emits particles
  - Controls particle **lifespan**, their **emission rate** and their **overall change through time**.

- *Particle* — basic element of a particle system
  - Controlled by the emitter.
  - Often a **billboard** (2D texture plane or quad mesh) **facing the camera**, but it can also be a 3D object.
  - Particle lifespan lasts from when it enters the particle system until it leaves it.

Particle System

- Particle Systems are currently intended to be used in **3D space**.

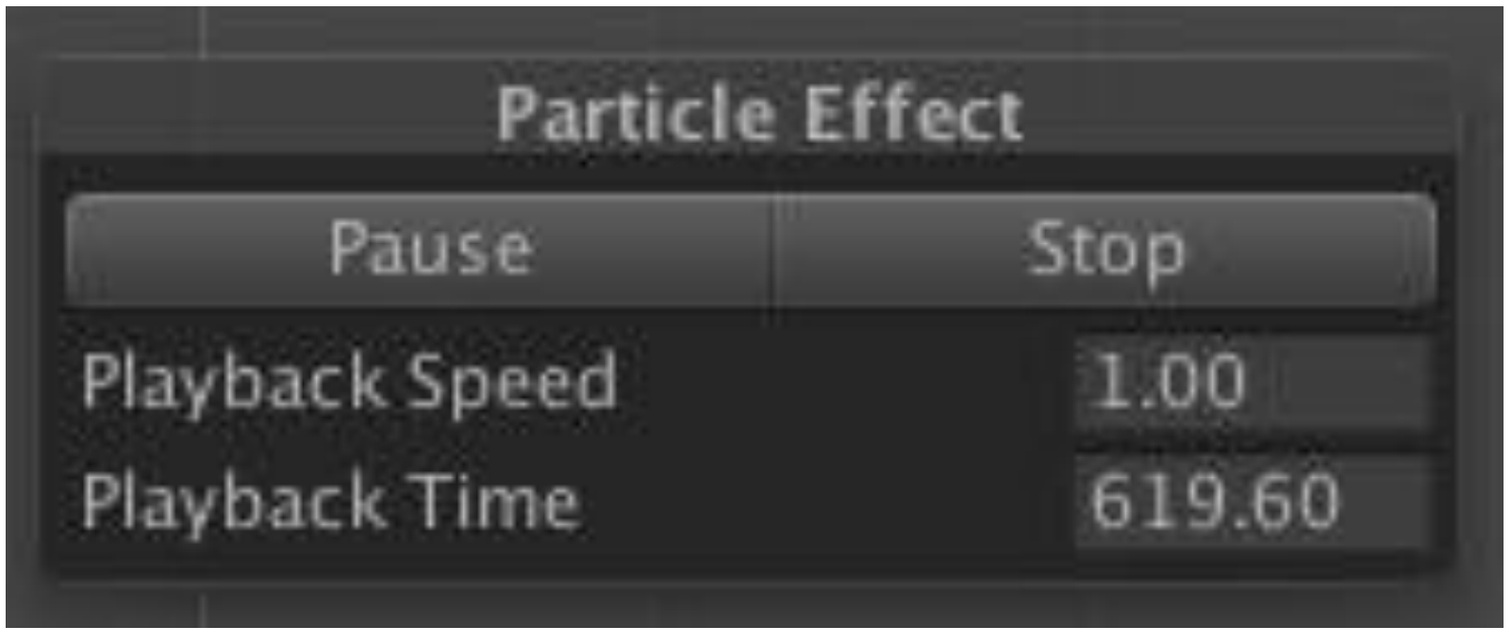
- They can be used in 2D, but they have **no concept of Sprites**.

- Sorting Layers cannot be set on a Particle System via the Inspector, but they can be set in your scripts.

- This is problematic for our 2D games.

Particle Effect Viewport

- Shown in the **Scene View**.
- Controls the **preview** of particle effect's playback.
- Changes made to the preview controls **do not affect** the actual particle effect during runtime.



Particle System Inspector (Shuriken)

- As a game object component, the Particle System shows up in the Inspector View for the currently selected particle system.

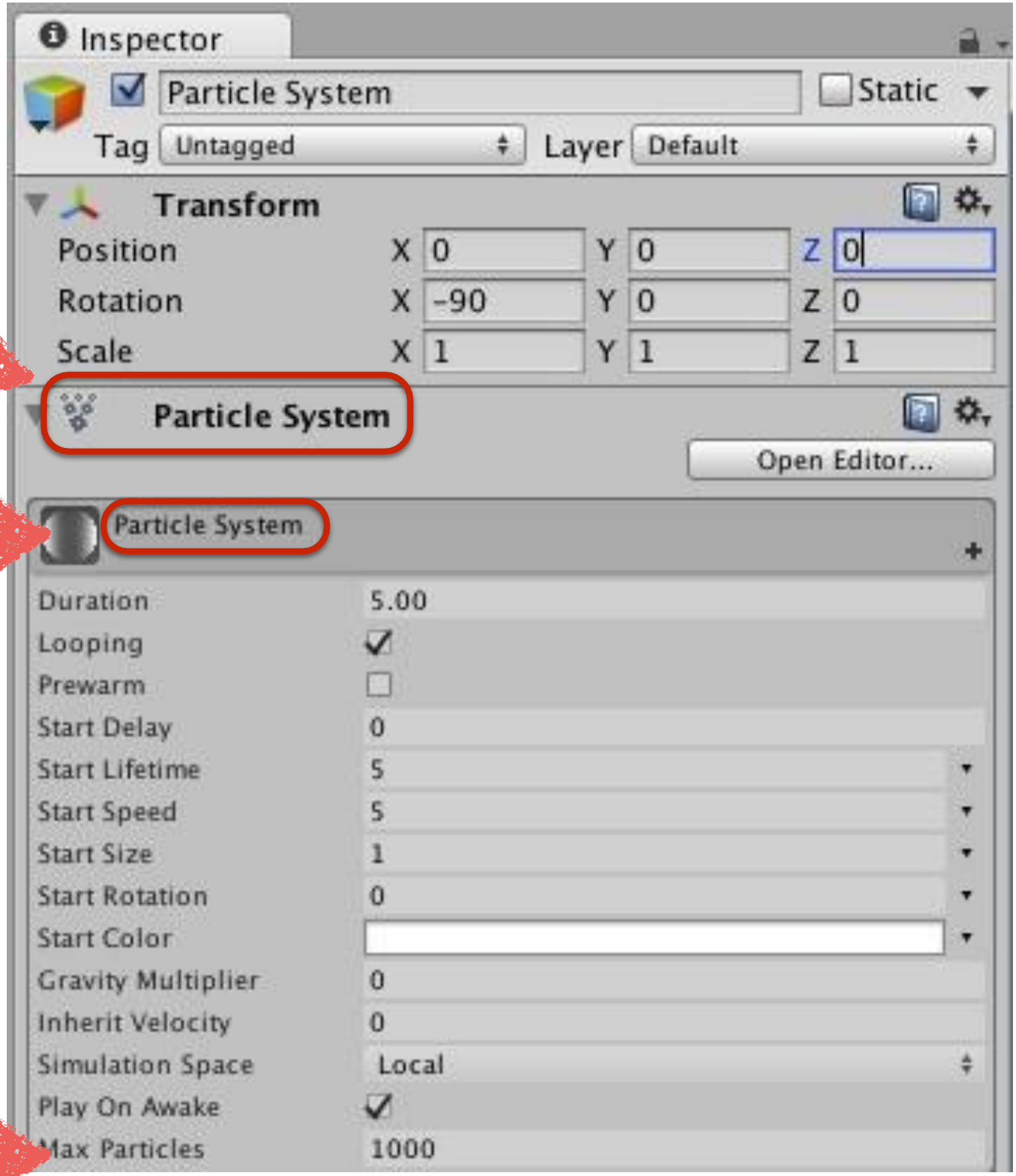
- It is also referred to as the **Shuriken editor**, because it uses Unity's Shuriken particle system to drive its behaviour.

- Contains multiple **modules** shown as tabs in the component view.

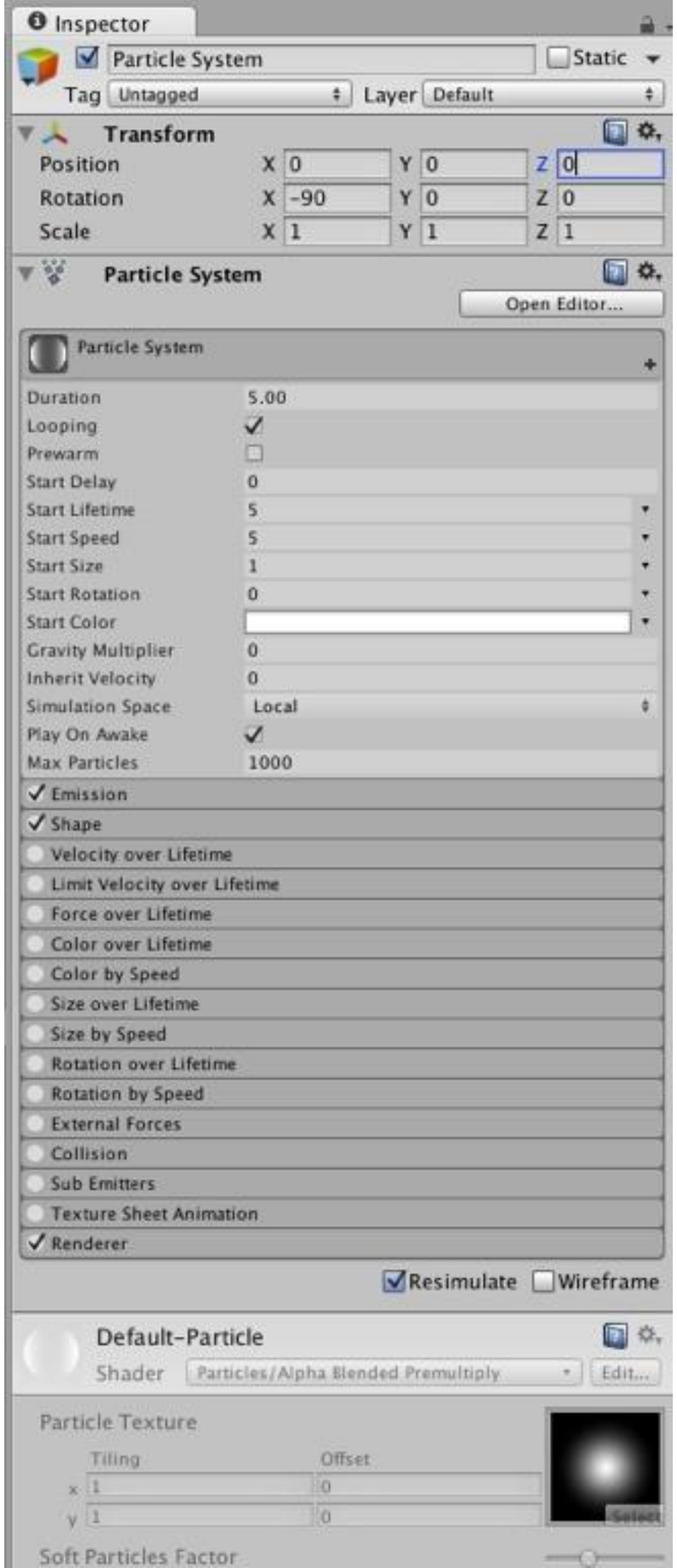
- The first module has the same name: **Particle System**.

Component

Module Tab



Other Modules



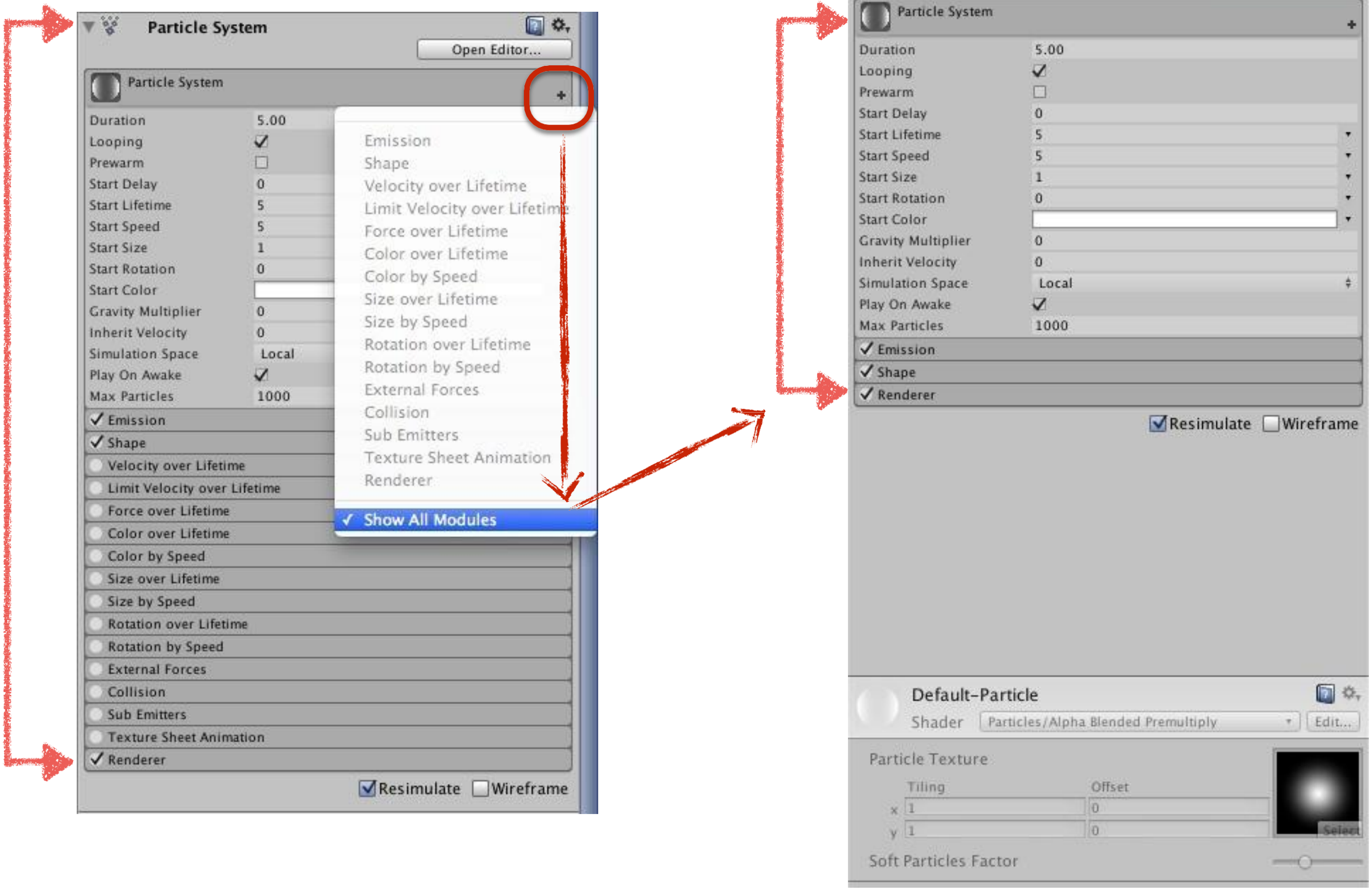
Modules

- When enabled, a module adds **specialized behaviour** to the particle system.

- Initially, there are only a few modules enabled.

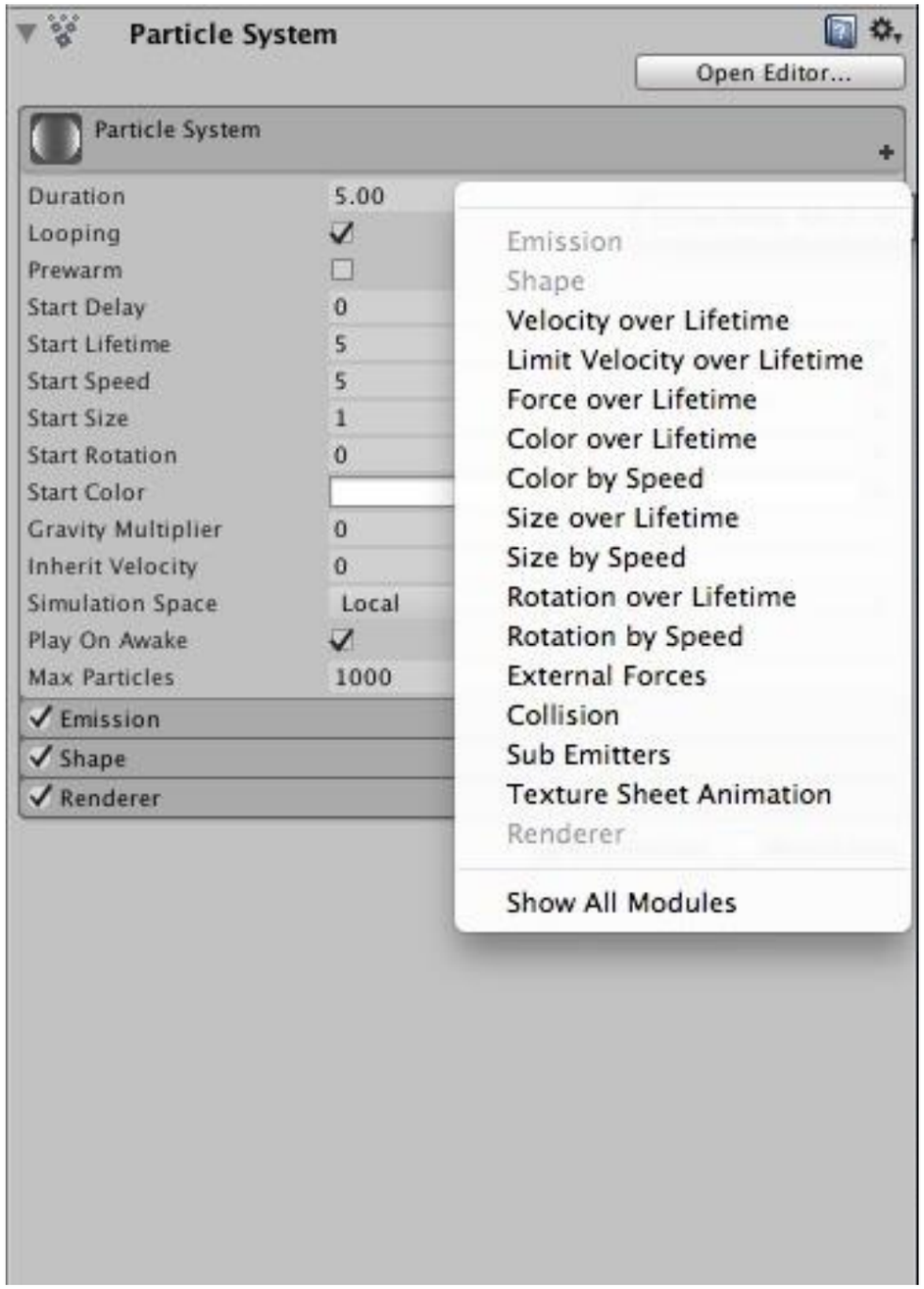
- By default, **you can see all available modules** right under the main Particle System module.

- This also shows any modules that are not used. You can **hide and show disabled modules** via the **(+) button** in the top right of the **Particle Emitter module tab**. This opens up a module selection list from which you can select which modules to show.


Particle System
Open Editor...
Particle System
Duration 5.00
Looping
Prewarm
Start Delay 0
Start Lifetime 5
Start Speed 5
Start Size 1
Start Rotation 0
Start Color
Gravity Multiplier 0
Inherit Velocity 0
Simulation Space Local
Play On Awake
Max Particles 1000
Emission
Shape
Velocity over Lifetime
Limit Velocity over Lifetime
Force over Lifetime
Color over Lifetime
Color by Speed
Size over Lifetime
Size by Speed
Rotation over Lifetime
Rotation by Speed
External Forces
Collision
Sub Emitters
Texture Sheet Animation
Renderer
Show All Modules
Resimulate
Wireframe
Default-Particle
Shader
Particles/Alpha Blended Premultiply
Edit...
Particle Texture
Tiling
Offset
Select
Soft Particles Factor

- Deselecting “Show All Modules” **condenses the list** to expose only those modules that are currently enabled.

- At this point, if you then wish to **enable a module**, you can select it individually from the selection list.



Particle System

- Can be viewed in its own window by pressing **Open Editor** button in the Inspector view.
- This brings up the **Particle Effect** editor window.
- The Particle Effect editor window **contains all features** of Unity’s Shuriken particle system.

Inspector

Particle System

Static

Tag Untagged

Layer Default

Transform

Position	X	0	Y	0	Z 0
Rotation	X	-90	Y	0	Z 0
Scale	X	1	Y	1	Z 1

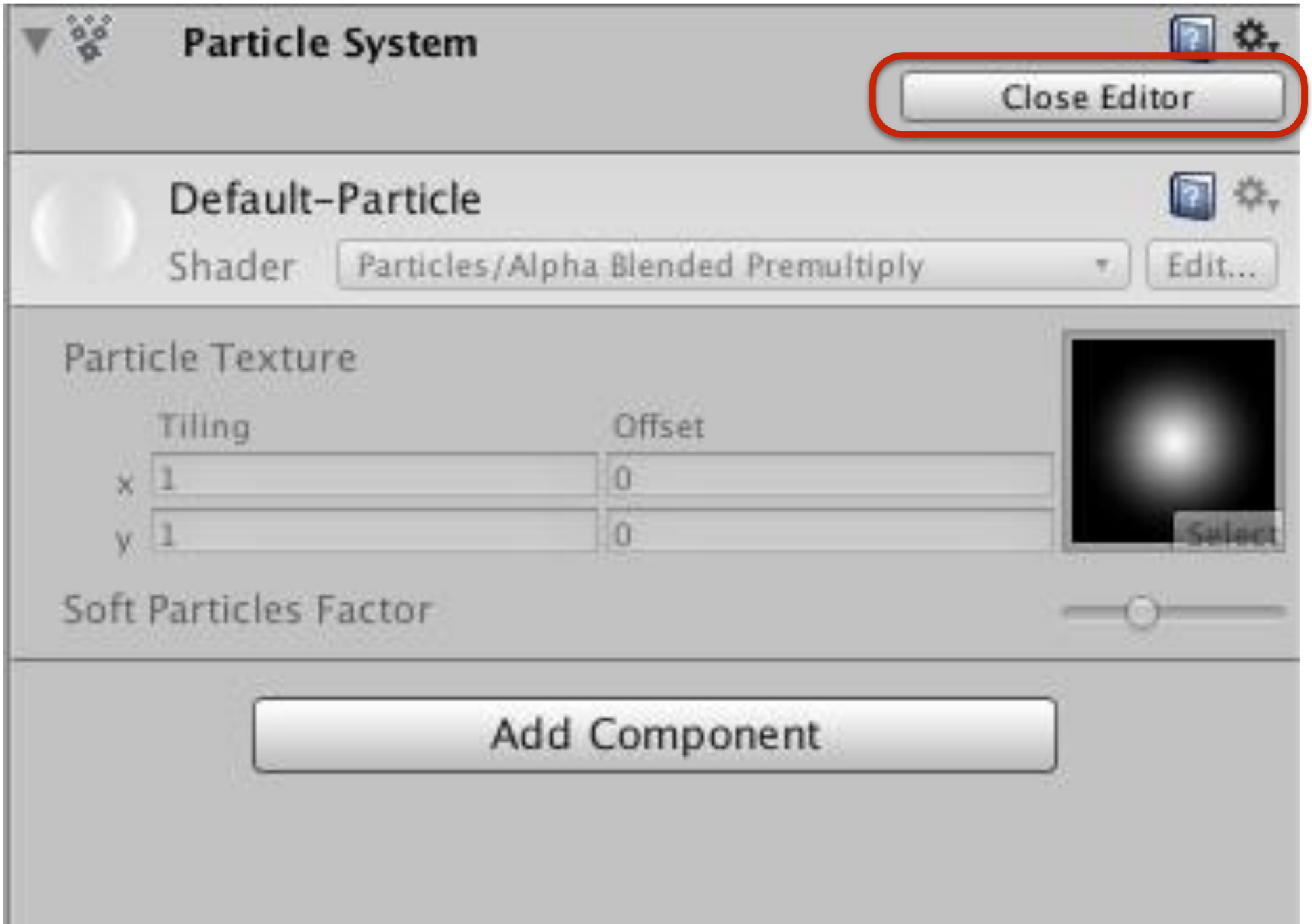
Particle System

Open Editor...

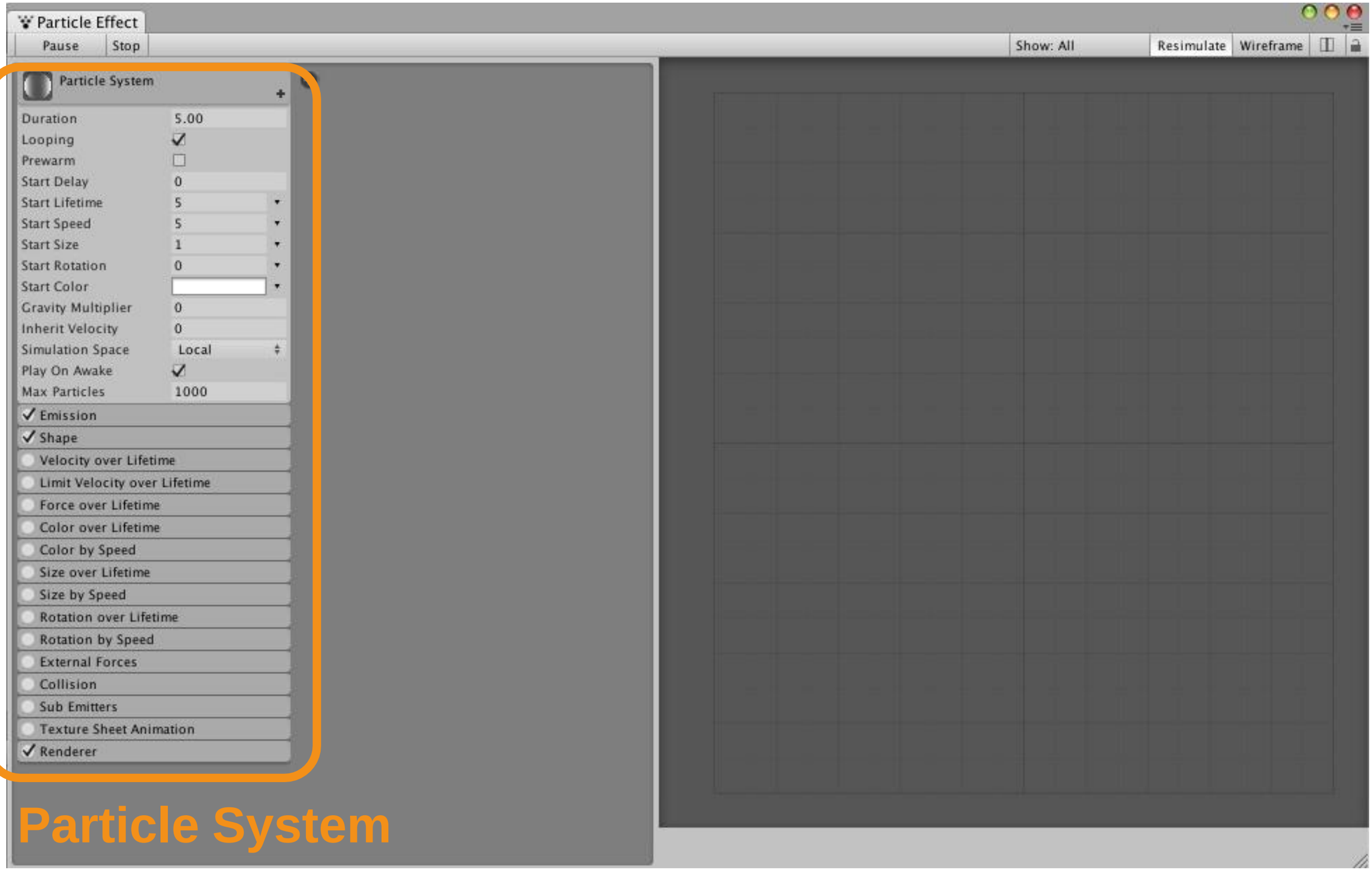
Particle System

Duration	5.00
Looping	✓
Prewarm	☐
Start Delay	0
Start Lifetime	5
Start Speed	5
Start Size	1
Start Rotation	0
Start Color	
Gravity Multiplier	0
Inherit Velocity	0
Simulation Space	Local
Play On Awake	✓
Max Particles	1000

While the Particle Effect editor window is open, the Inspector View shrinks to display only the "Close Editor" button.



This is the Particle Effect editor window.



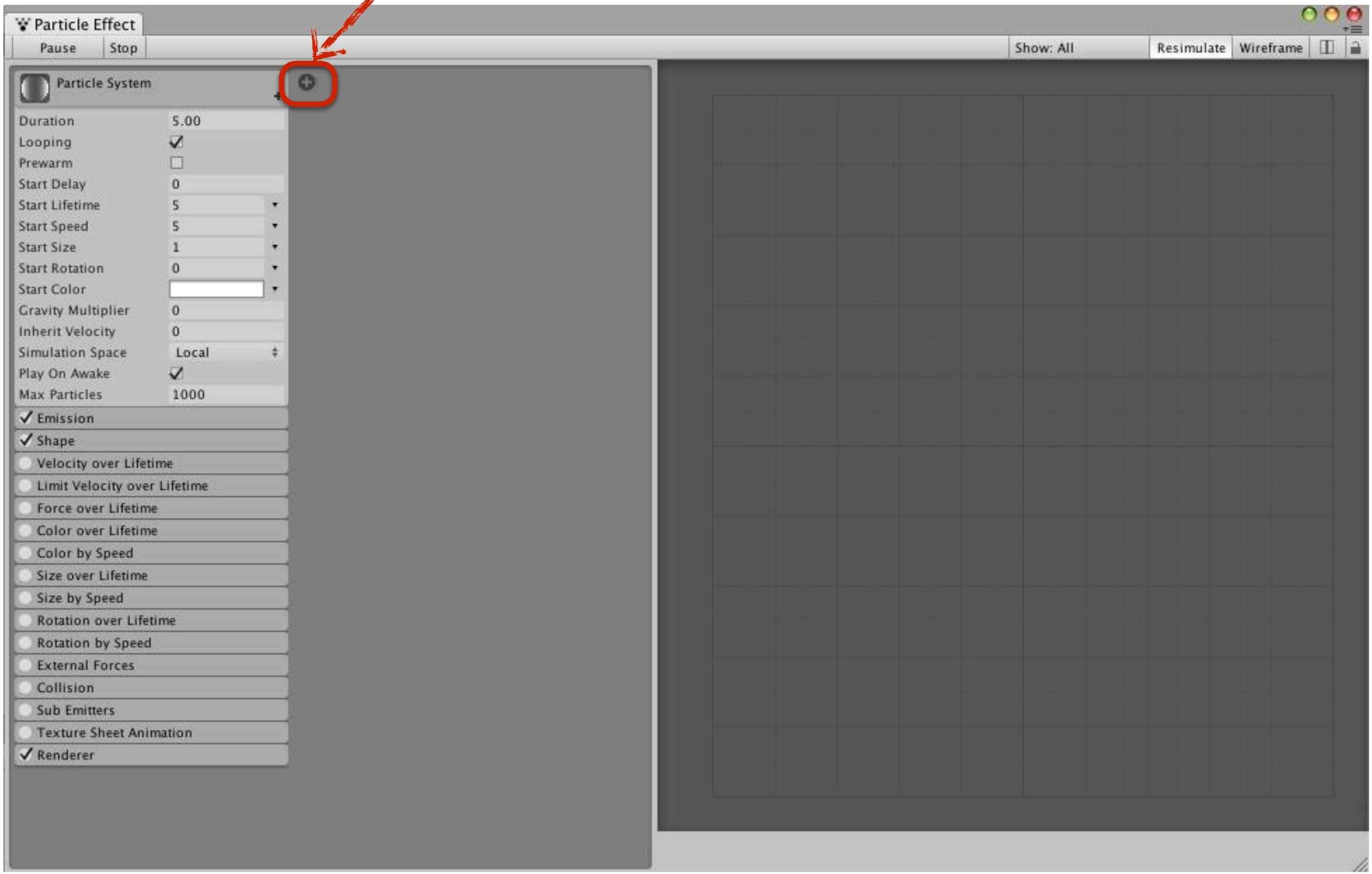
Particle System vs Particle Effect

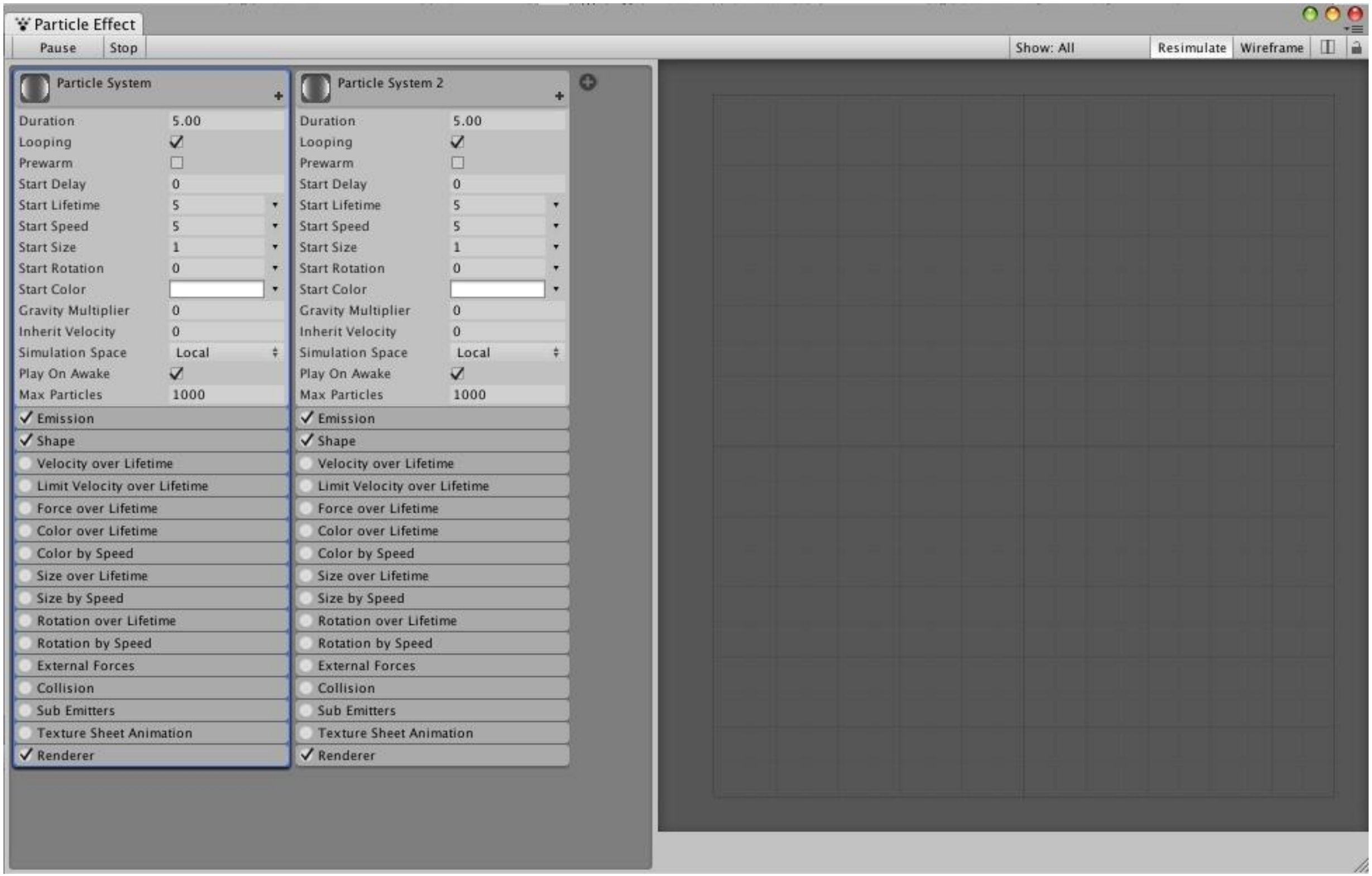
- Particle Effects contain **one or more Particle System** components parented to the same root.

- Particle Effects are edited in the **Particle Effect editor window.**

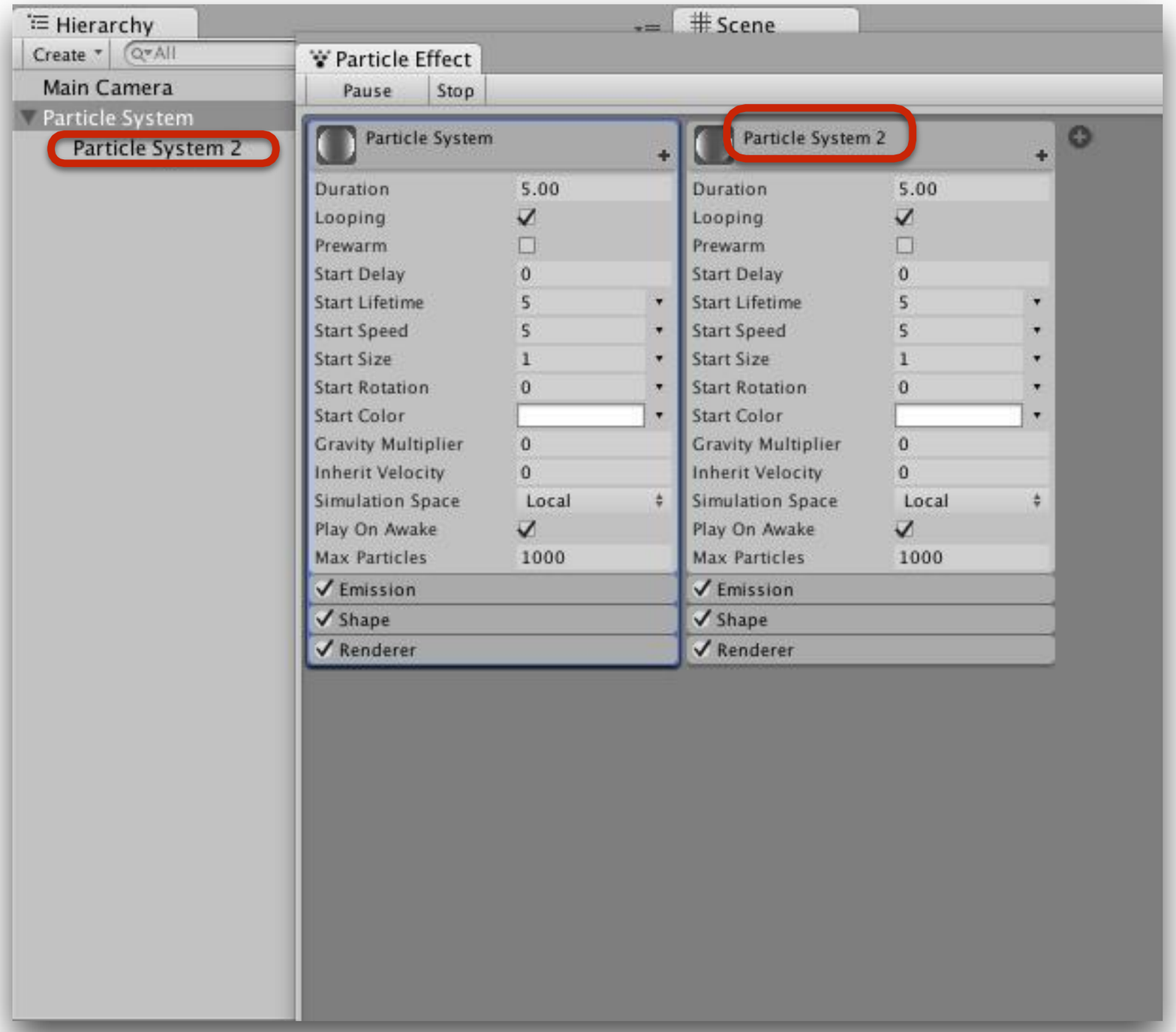
- Individual Particle System components are edited in both the Inspector View and in the Particle Effect window.

Add an additional Particle System to this Particle Effect

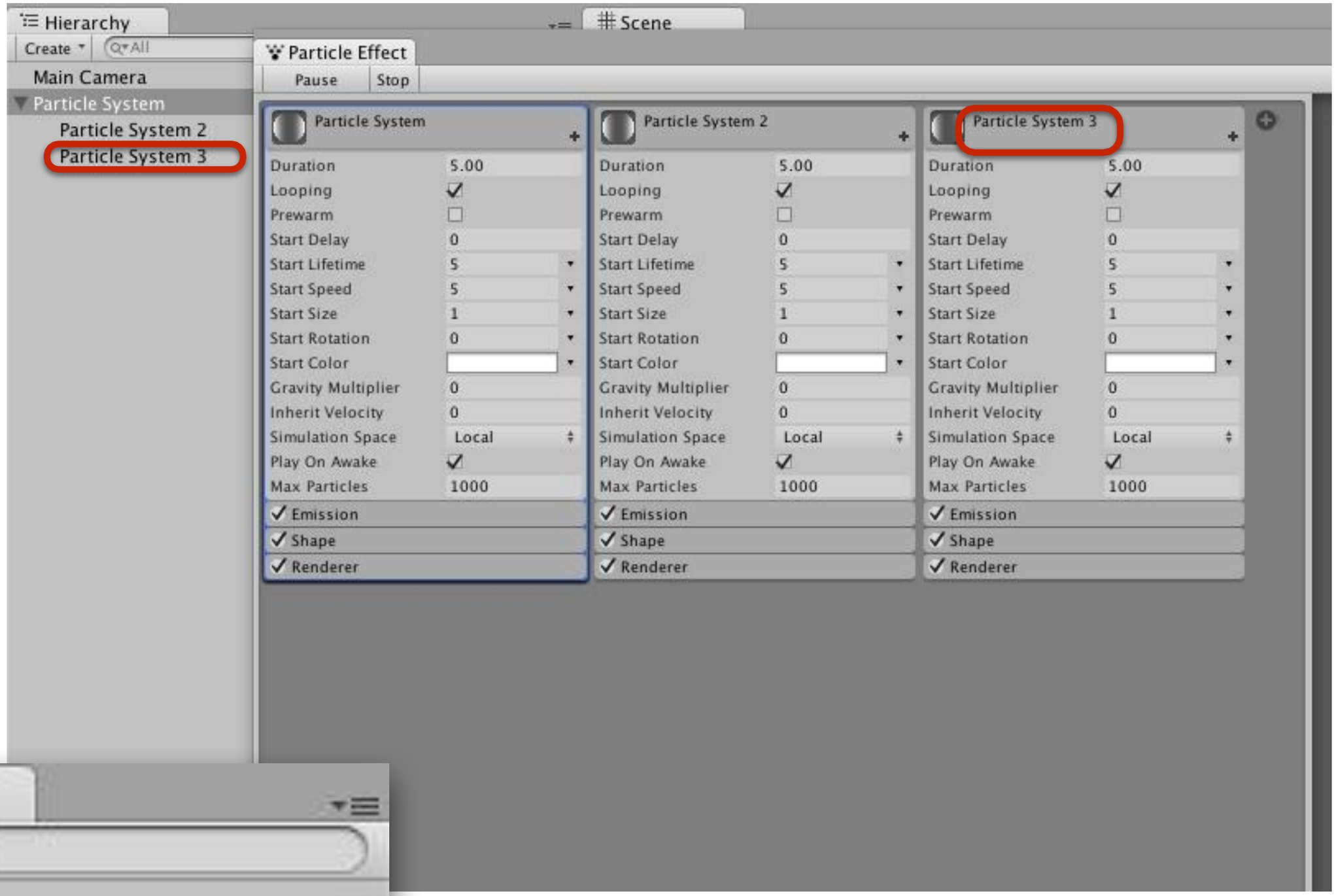


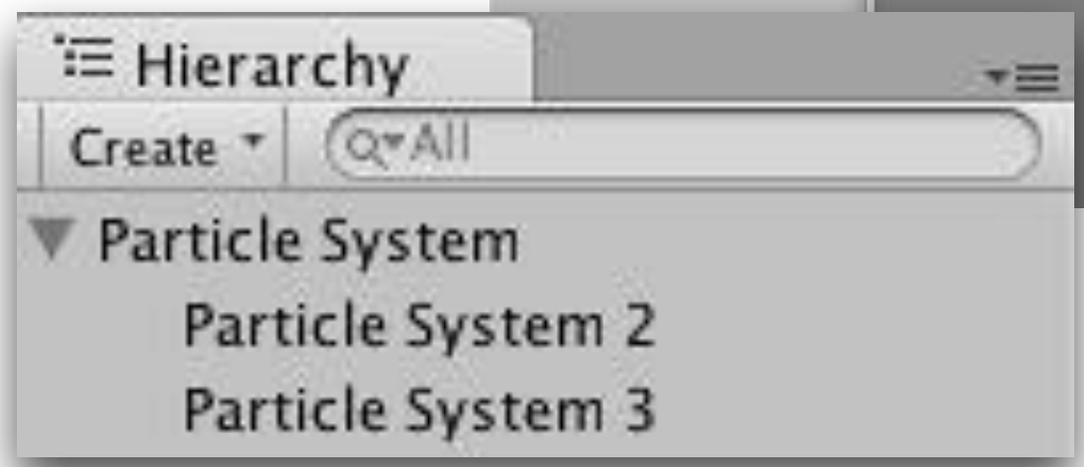

Particle Effect
Pause
Stop
Show: All
Resimulate
Wireframe
Particle System
Duration 5.00
Looping
Prewarm
Start Delay 0
Start Lifetime 5
Start Speed 5
Start Size 1
Start Rotation 0
Start Color
Gravity Multiplier 0
Inherit Velocity 0
Simulation Space Local
Play On Awake
Max Particles 1000
Emission
Shape
Velocity over Lifetime
Limit Velocity over Lifetime
Force over Lifetime
Color over Lifetime
Color by Speed
Size over Lifetime
Size by Speed
Rotation over Lifetime
Rotation by Speed
External Forces
Collision
Sub Emitters
Texture Sheet Animation
Renderer
Particle System 2
Duration 5.00
Looping
Prewarm
Start Delay 0
Start Lifetime 5
Start Speed 5
Start Size 1
Start Rotation 0
Start Color
Gravity Multiplier 0
Inherit Velocity 0
Simulation Space Local
Play On Awake
Max Particles 1000
Emission
Shape
Velocity over Lifetime
Limit Velocity over Lifetime
Force over Lifetime
Color over Lifetime
Color by Speed
Size over Lifetime
Size by Speed
Rotation over Lifetime
Rotation by Speed
External Forces
Collision
Sub Emitters
Texture Sheet Animation
Renderer

This will also add a new Particle System as a child game object.



This will also add a new Particle System as a child game object.





Removing the child particle systems will <u>also</u> remove them from the Particle Effect editor.

Particle System
MinMax Curves

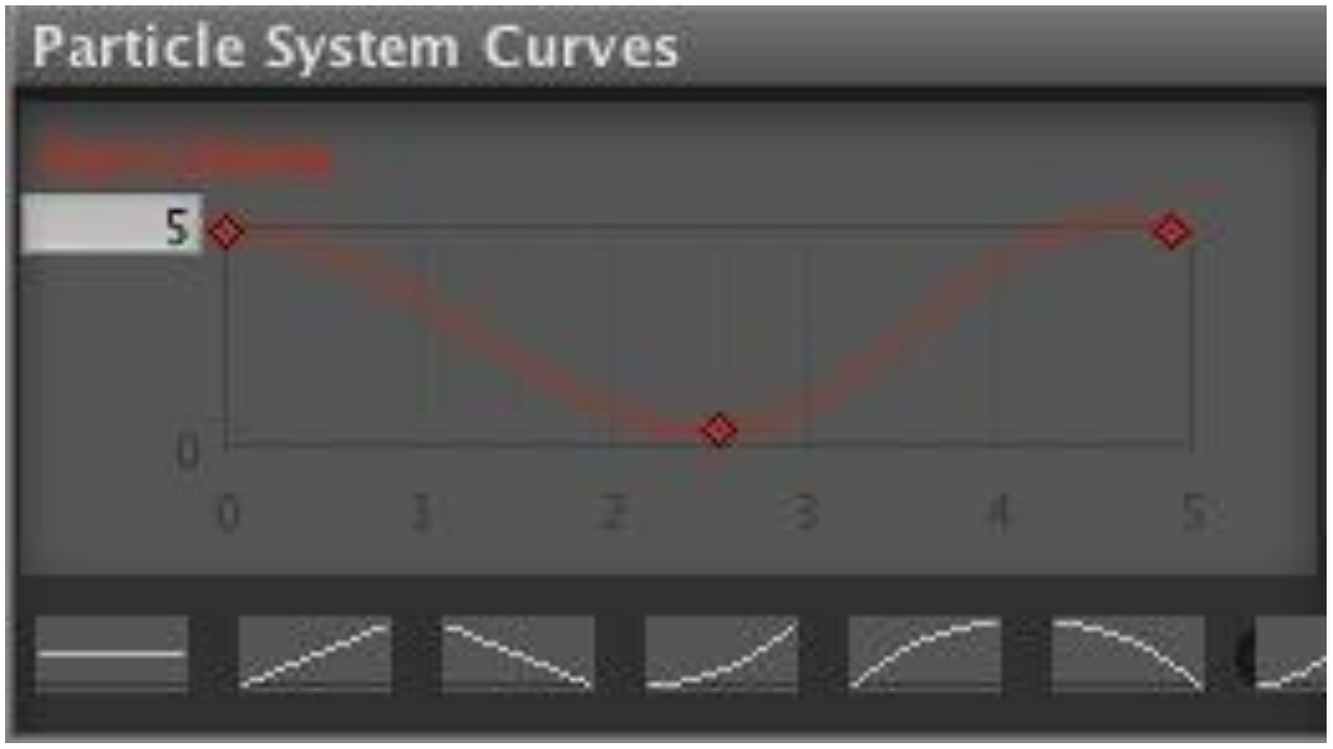
- Some module properties define a **change of values through time** using MinMax Curves.

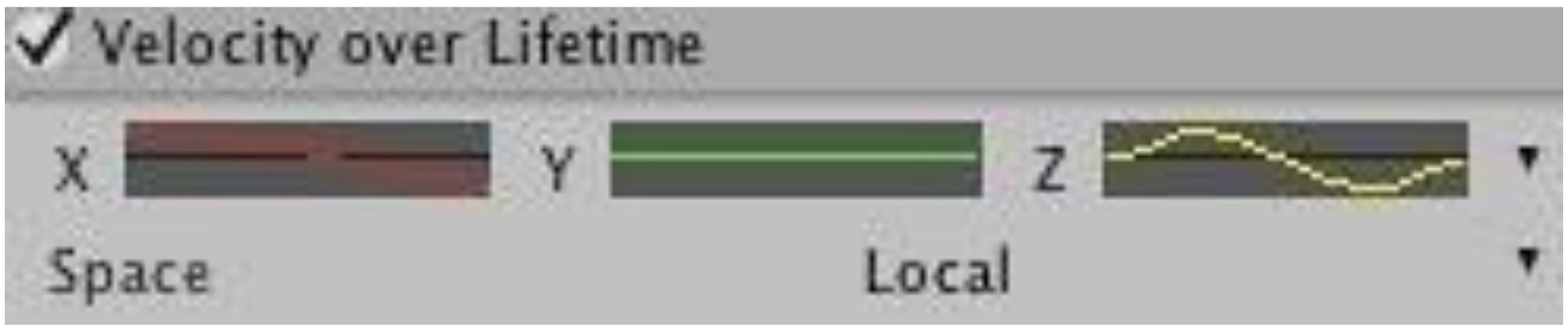
- MinMax Curves interpolate a curve between two points. This can be a line as well.

- **Interpolation** is a process of filling in values **between two points**.

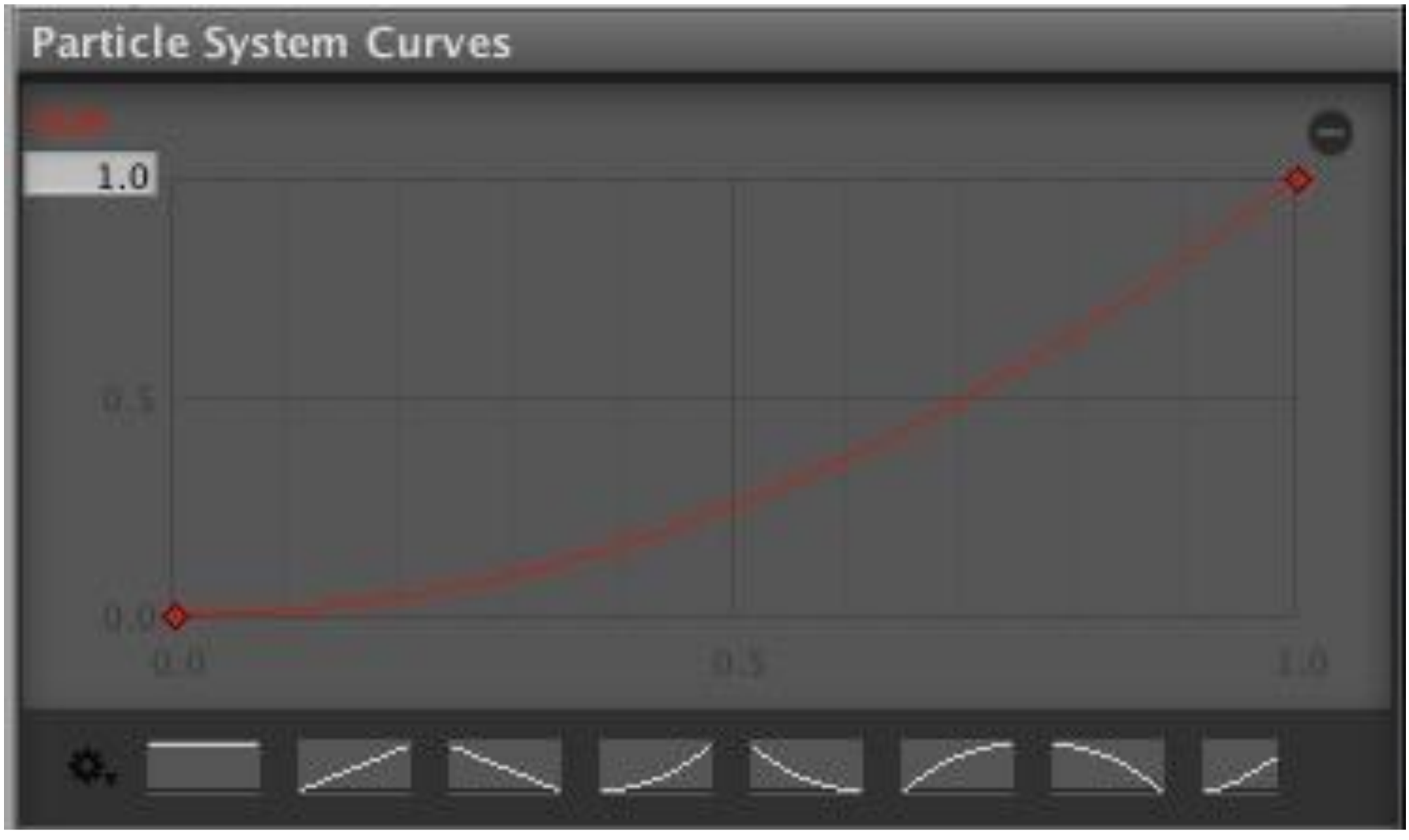
- You can add **additional points** to the MinMax curve to define more complex changes through time.
  - Double click on the curve to add an additional point.

Particle System MinMax Curves



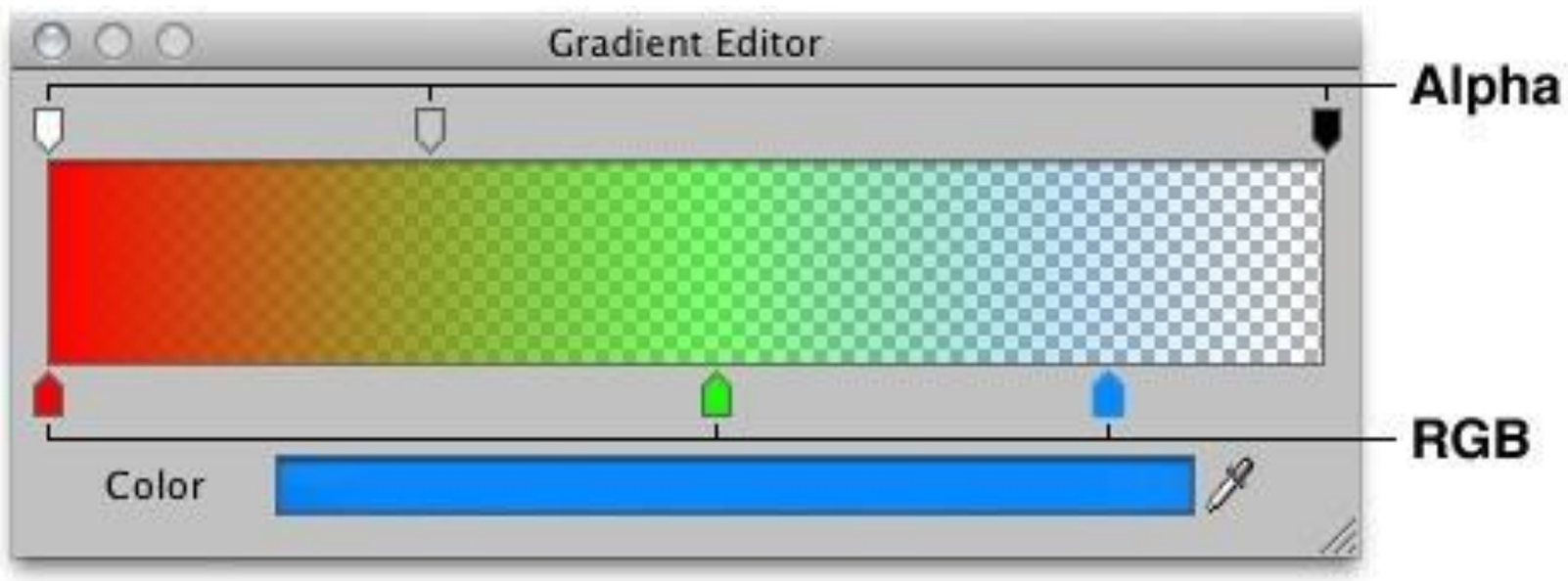


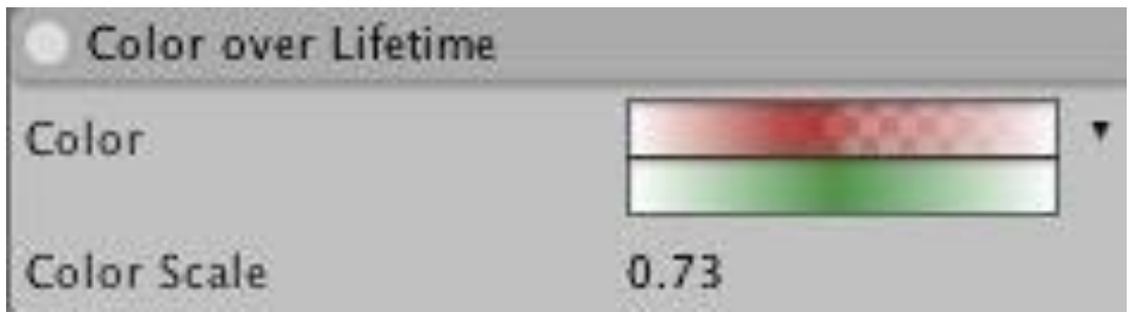
Particle System MinMax Curves



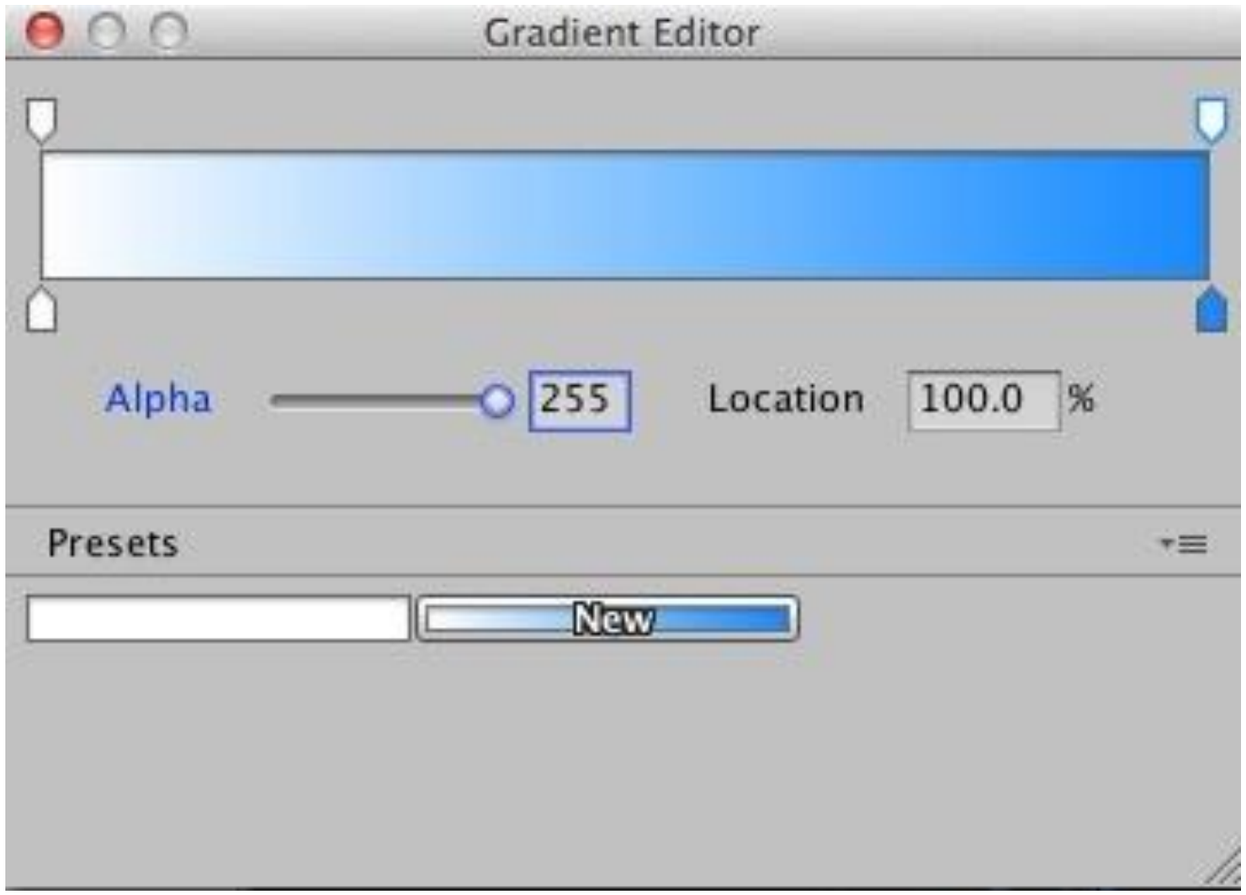
Colours and Gradients

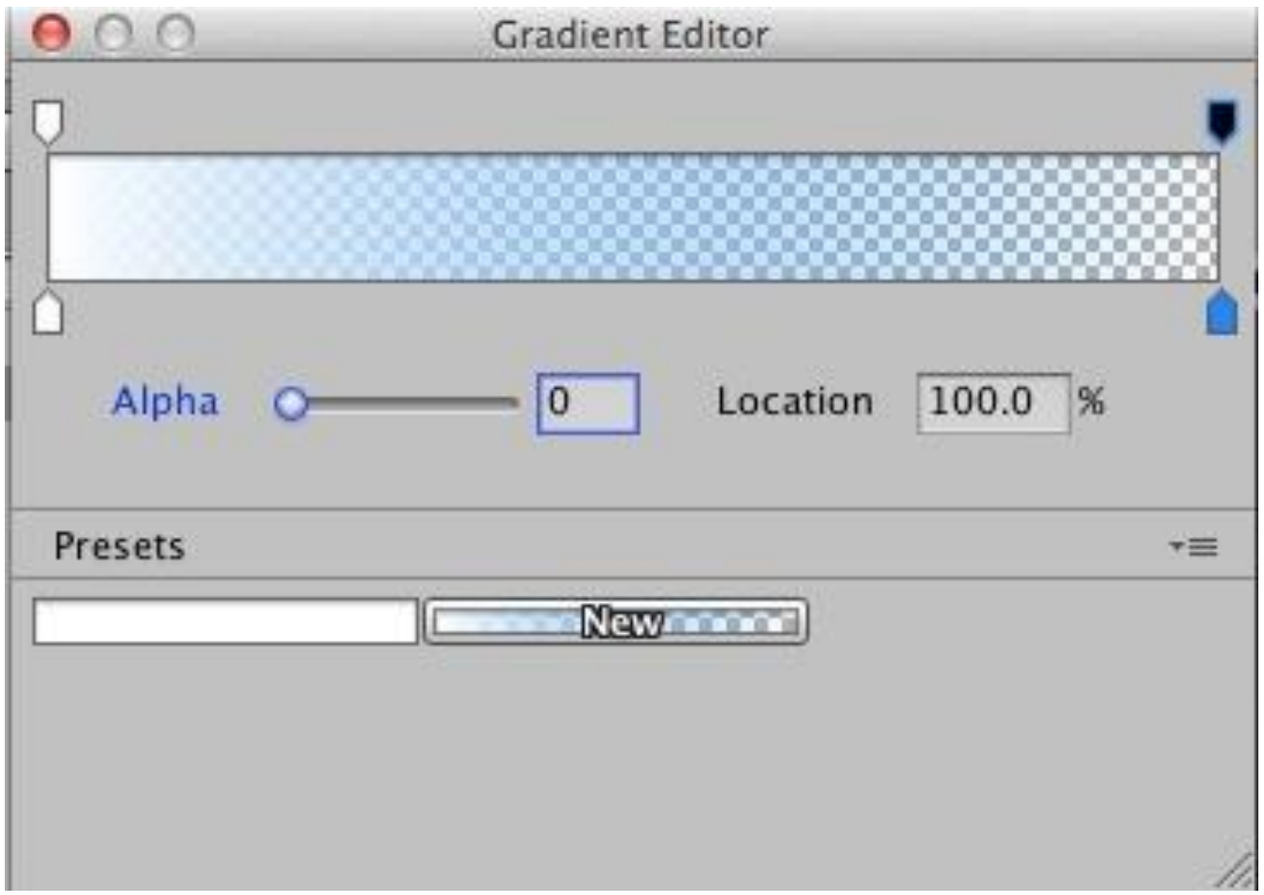
- Module properties that deal with colour use **Colour and Gradient Editor**, which works similar to the Curve Editor.





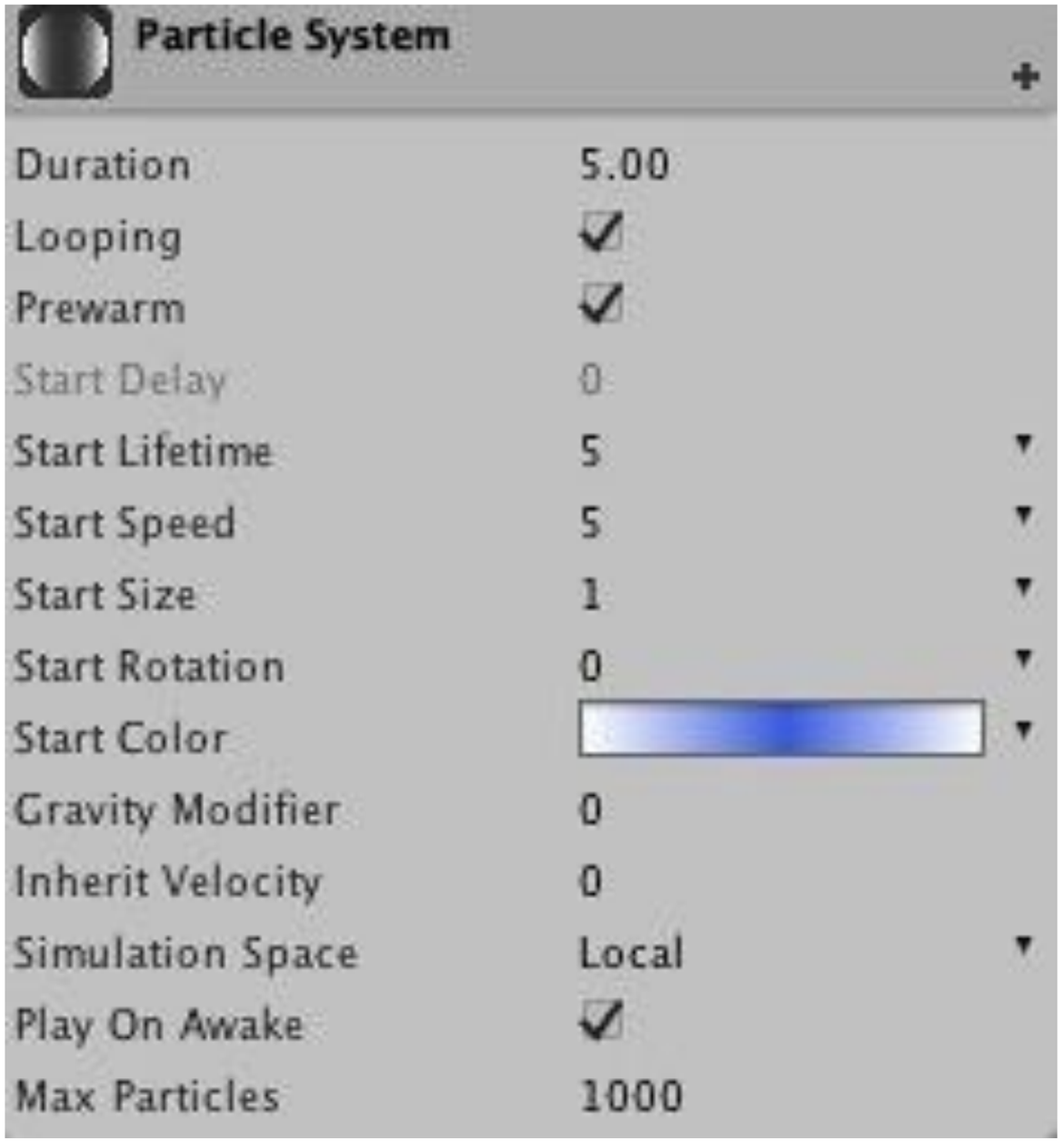
Colours and Gradients





Particle System Module

- Defines overall behaviour of the particle system.
- Always present.
- It cannot be removed or disabled.



Particle System Module

- **Duration** — defines the duration of the particle system.
- **Looping** — whether the system occurs once or loops forever.
- **Prewarm** — looping systems will start as if they have emitted particles for one cycle.
- **Gravity Modifier** — gravity effect on particles.
  (1 = normal effect, 2 = double, 0.5 = half)

Particle System Module

- **Inherit Velocity** — used for moving particles, specifies the amount of velocity to inherit from its parent Transform.

- **Simulation Space** — Local or World space simulation.

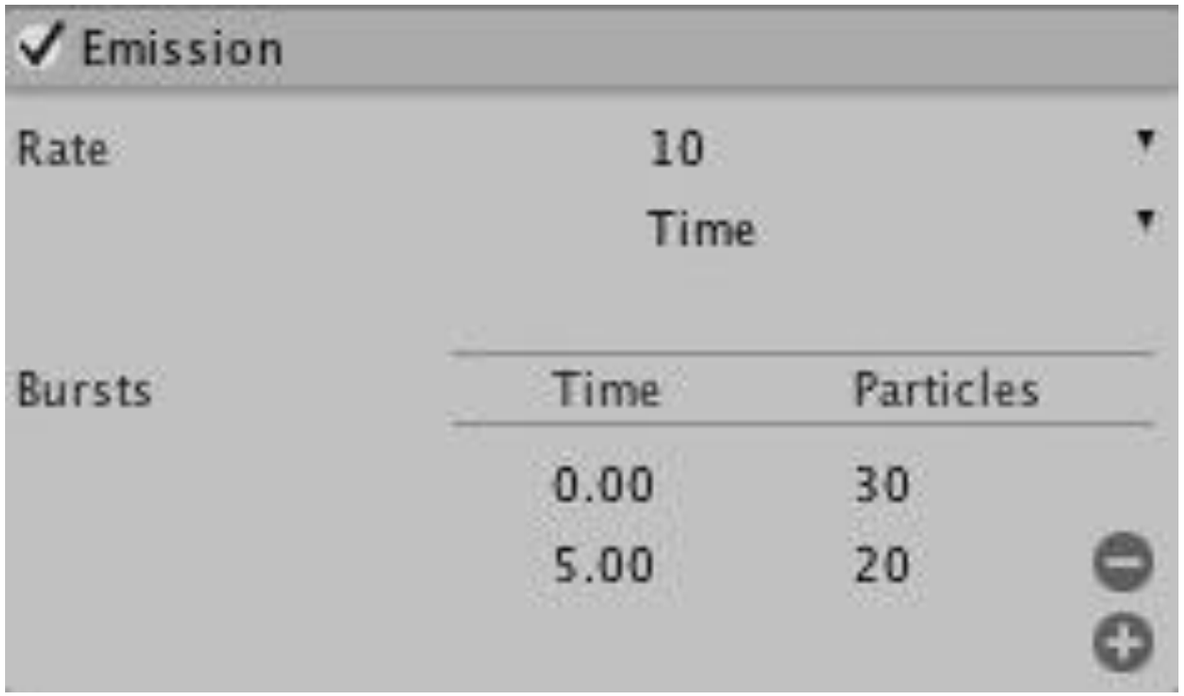
- **Play on Awake** — whether to automatically play the Particle System when it is created.

- **Max Particles** — the maximum number of particles that can be present in the scene at any one time for this Particle System.

Particle System Demo

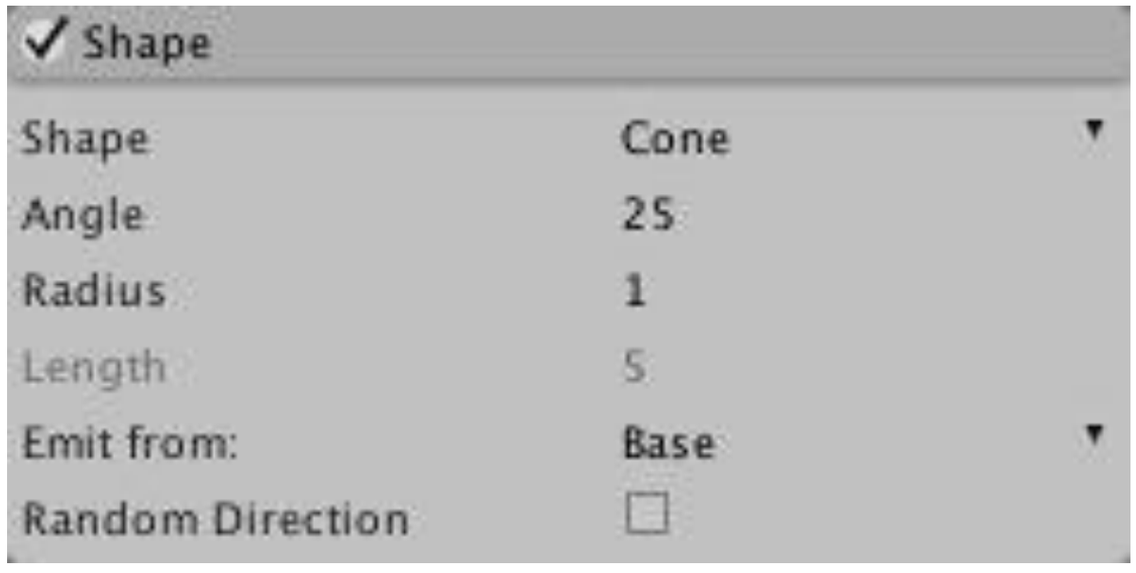
Emission Module

- Controls particle **emission rate**.
- Also allows **bursts** of particles at certain moments within the Particle System duration time.
- Bursts are useful when a bunch of particles need to be created at once.
  - E.g. explosions.



Shape Module

- Defines the shape of the particle emitter: Sphere, Hemisphere, Cone, Box and Mesh.



- Each shape has its own set of properties.

- You can choose whether to emit from the **shell** (edge) of the shape or within the **volume** of the shape itself.

Velocity Modules

Velocity Over Lifetime

- Enables some **animation control** over a particle, by altering its **speed** and **direction over time**.

Limit Velocity Over Lifetime

- Limits velocity to a set **maximum velocity threshold.**
- Threshold can be **velocity vector** or an **individual axis** (x, y, z).

Force Over Lifetime

- Same as Velocity Over Lifetime, except it applies a force to the particle.

Color Modules

Color Over Lifetime

- Controls the **colour change** of each particle during its lifetime.
- Shorter particle lifetime results in **faster colour change**.
- The colour acts as a tint on the particle texture, so **white colour** results in **no colour change**.

Color By Speed

- Same as Color Over Lifetime, except it changes the colour **based on the particle’s speed** rather than its lifetime.

Size Modules

Size Over Lifetime

- Controls **change in particle size** over its lifetime.
- Shorter particle lifespan results in **faster size change**.

Size By Speed

- Same as Size Over Lifetime, except it changes the particle size **based on the particle's speed** rather than its lifetime.

Rotation Modules

Rotation Over Lifetime

- Controls **particle rotation speed** in degrees per second over its lifetime.
- Particle lifespan does not affect rotation speed.

Rotation By Speed

- Same as Rotation Over Lifetime, except it changes the particle rotation speed **based on the particle’s movement speed**.

Collision Module

Enables collisions with a **flat plane** or **scene Colliders**.

Flat Plane Collision (Planar Collision)

- Very efficient.
- Planes are defined by a game object Transform.

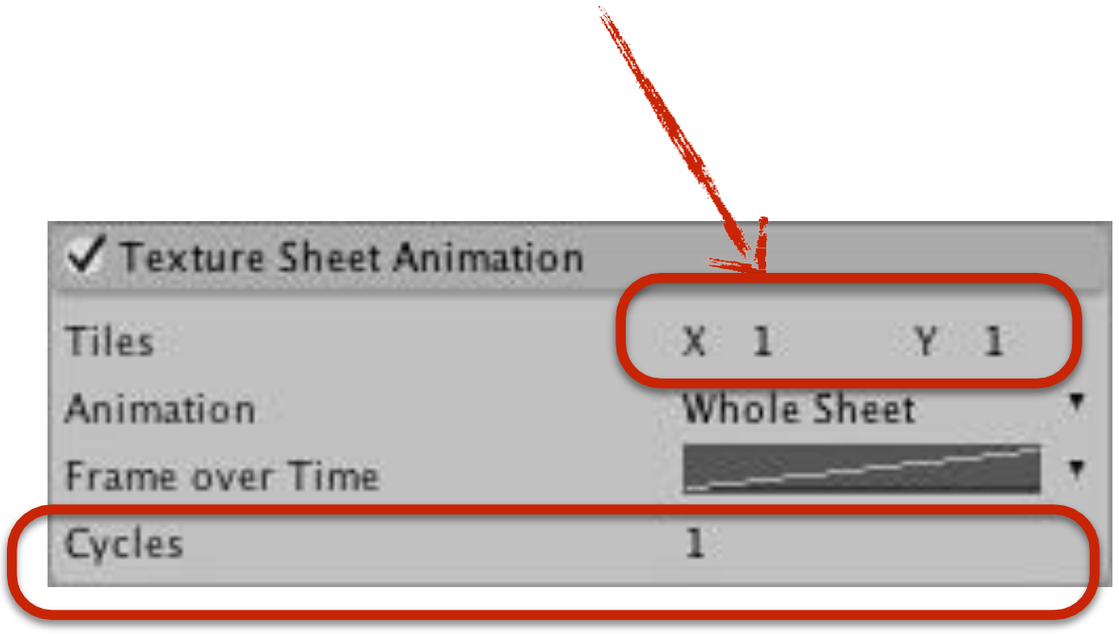
Scene Collider Collision (World Collision)

- Uses ray casts to determine collisions with a Collider shape.
- Can be performance-intensive, depending on collision quality.

Texture Sheet Animation Module

- Enables a particle to have an **animated texture**.
- Concept similar to how animated sprite sheets work.
- Texture sheet is **divided into a grid of tiles**.

Tile grid dimensions

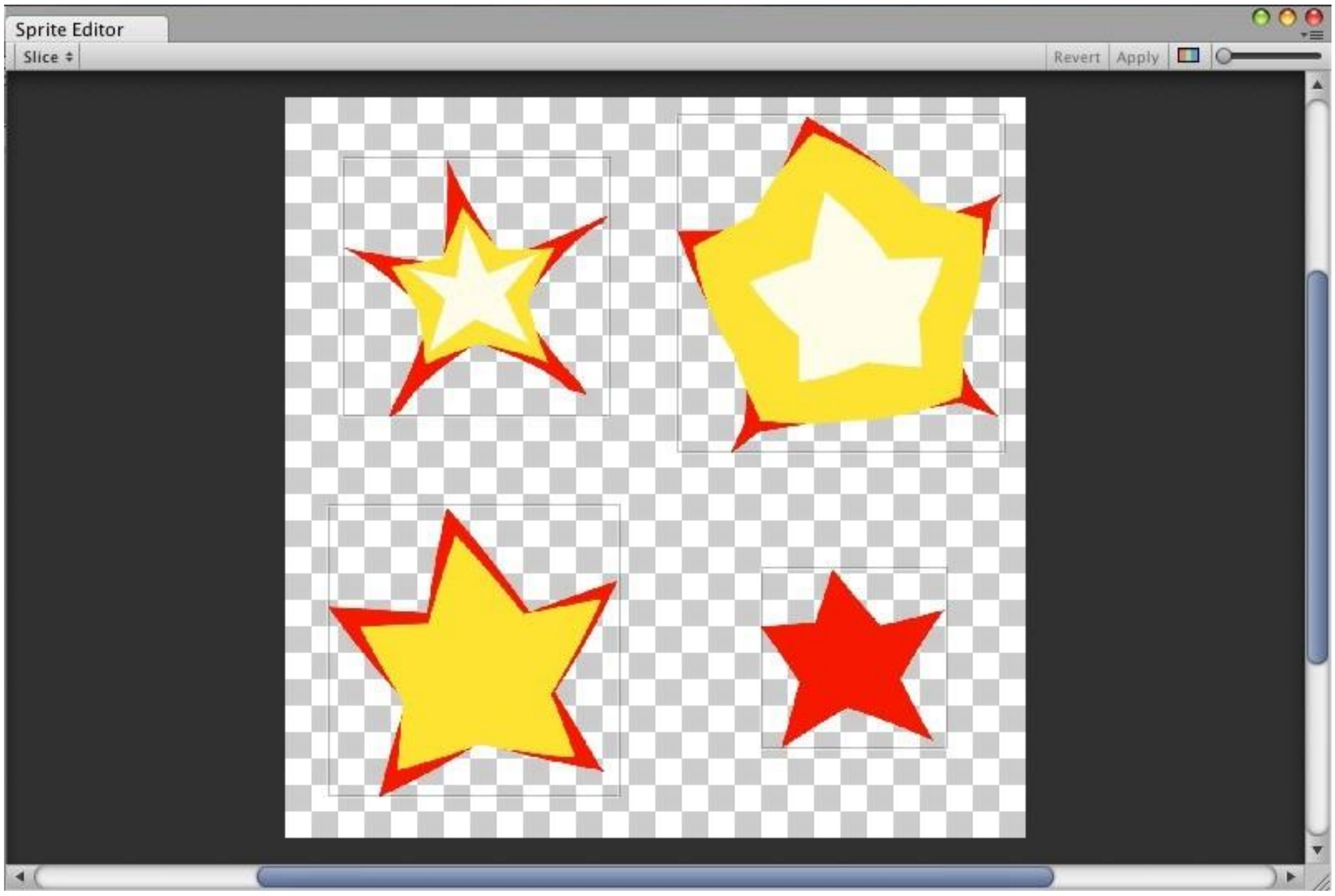


Number of cycles through the texture sheet during a particle's lifespan.

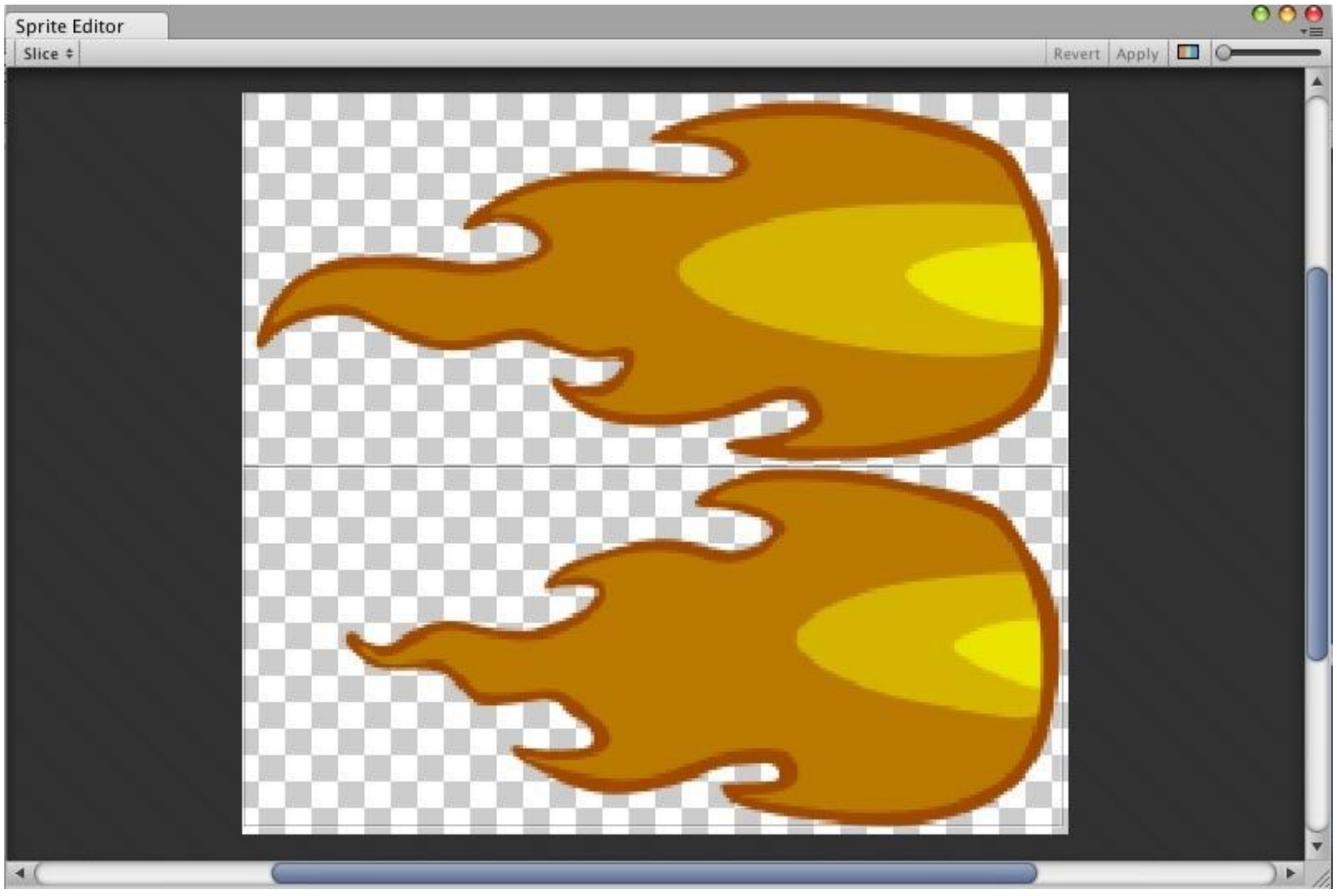
Texture Sheet Animation Module

- **Cycles** property defines the **animation speed**.
- More cycles means faster animation.
  - E.g. if Cycles = 4 then the texture sheet animation will play four times during a particle's lifespan.
- **Frame over Time** property alters the animation speed through a particle's lifespan.

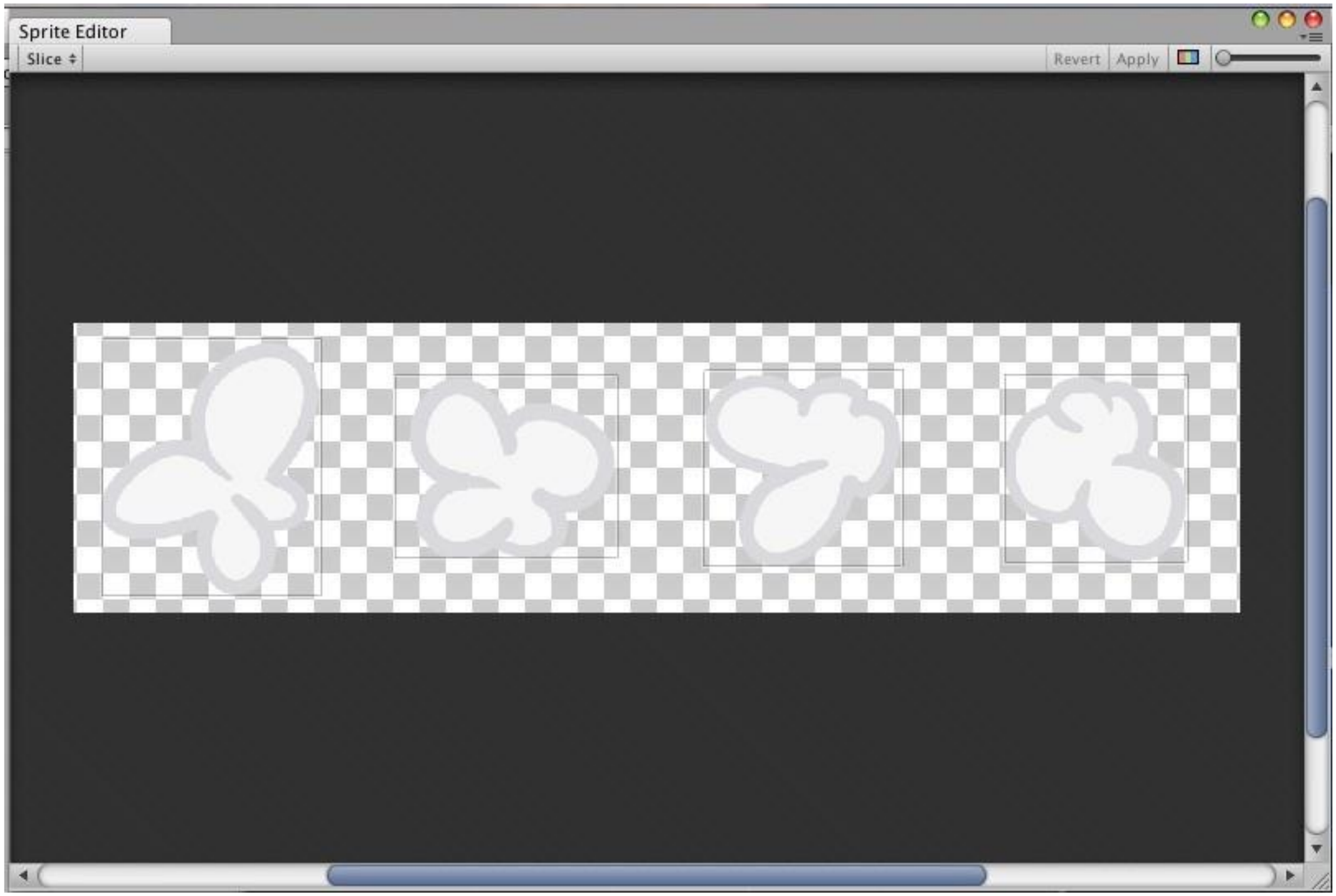
2x2 Texture Sheet



1x2 Texture Sheet

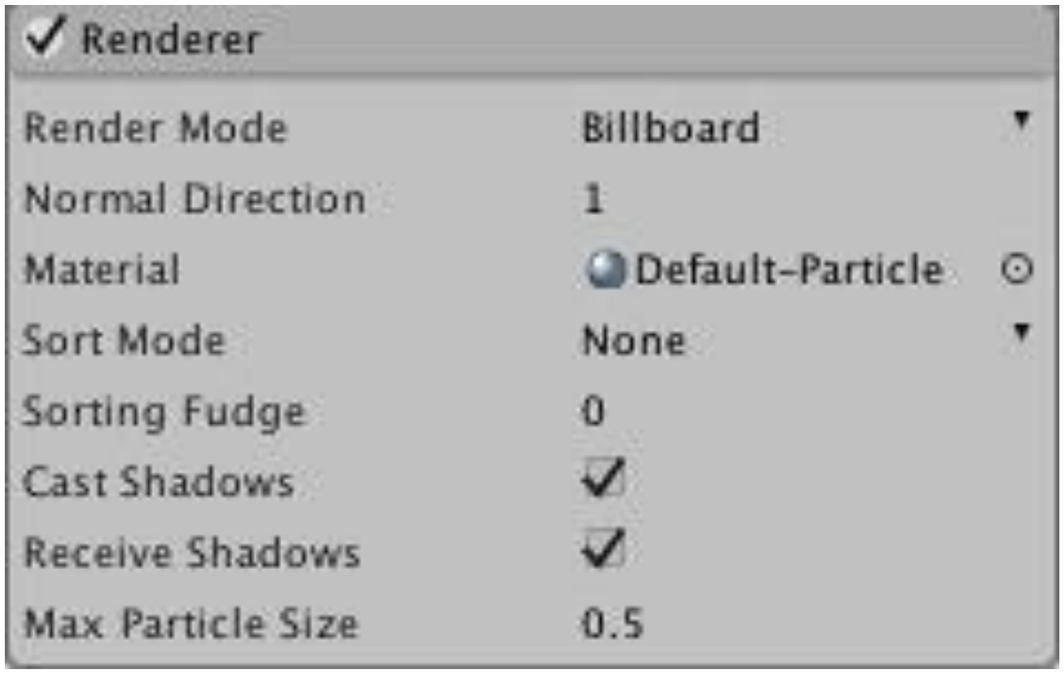


4x1 Texture Sheet



Renderer Module

- Controls how a particle is drawn on the screen.
- **Render Mode** specifies the shape of the particle.
  - **Billboard** makes the particle into a flat plane (quad) that always faces the camera.
  - **Mesh** uses a custom 3D object instead of a quad.
- You can specify the particle's **Material** here.



Particle Material

- The same principle applies for particles as it does for 3D meshes: it **requires a Material** to be able to display a texture on it.

- Recall: a **Material** has a **Shader** which takes at least one **Texture** as input.

Particle Material

- A particle's Renderer module therefore **requires a material** in order to draw the particle.

- This means you must have a defined **Material asset** in your project.

- This also means that you **cannot use a Sprite** asset directly to define a particle.

- You also **cannot use a Sprite Sheet** to play an animated texture on the particle.

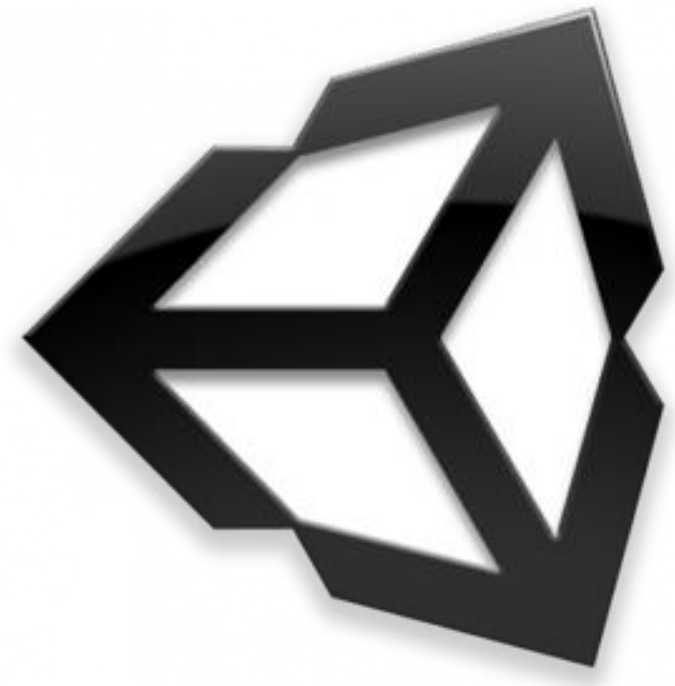
Particle Material

1. Define a **Material** asset first.
2. Specify which **Shader** it uses.
3. Provide a **Texture** as input to the Material’s Shader.
4. **Assign** the Material to the Particle System.

Particle Material

- Therefore, in order to play an animated sprite sheet on a particle:
    - First create a Material asset whose Texture is the Sprite Sheet (i.e. a Sprite asset containing multiple sprites).
    - Then apply that Material to the Particle System.
- In all cases, you are **applying a Material** to the Particle System.

Demo



Scripting Particle Systems

- Your script must have a reference to a `ParticleSystem` component.

- If the Particle Effect uses multiple Particle Systems, then often this reference will be the **root** Particle System.

- You can then make calls on the component:
  - e.g. `Play(), Stop()` — plays and stops the particle system and all of its child particle systems.

Scripting Particle Systems

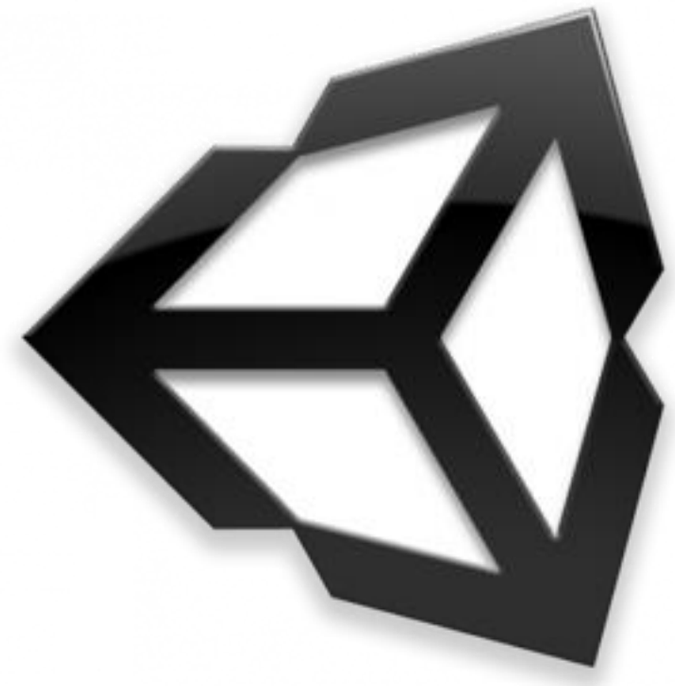
Variables	
duration	The duration of the particle system in seconds (Read Only).
emissionRate	The rate of emission.
enableEmission	When set to false, the particle system will not emit particles.
gravityModifier	Scale being applied to the gravity defined by Physics.gravity.
isPaused	Is the particle system paused right now ?
isPlaying	Is the particle system playing right now ?
isStopped	Is the particle system stopped right now ?
loop	Is the particle system looping?
maxParticles	The maximum number of particles to emit.
particleCount	The current number of particles (Read Only).
playbackSpeed	The playback speed of the particle system. 1 is normal playback speed.
playOnAwake	If set to true, the particle system will automatically start playing on startup.
randomSeed	Random seed used for the particle system emission. If set to 0, it will be assigned a random value on awake.
safeCollisionEventSize	Safe array size for use with ParticleSystem.GetCollisionEvents.
simulationSpace	This selects the space in which to simulate particles. It can be either world or local space.
startColor	The initial color of particles when emitted.
startDelay	Start delay in seconds.
startLifetime	The total lifetime in seconds that particles will have when emitted. When using curves, this values acts as a scale on the curve. This value is set in the particle when it is create by the particle system.
startRotation	The initial rotation of particles when emitted. When using curves, this values acts as a scale on the curve.
startSize	The initial size of particles when emitted. When using curves, this values acts as a scale on the curve.
startSpeed	The initial speed of particles when emitted. When using curves, this values acts as a scale on the curve.
time	Playback position in seconds.

Scripting Particle Systems

Functions

Clear	Remove all particles in the particle system.
Emit	Emit count particles immediately.
GetCollisionEvents	Get the particle collision events for a GameObject. Returns the number of events written to the array.
GetParticles	Get the particles of this particle system. Returns the number of particles written to the input particle array.
IsAlive	Does the system have any live particles (or will produce more)?
Pause	Pauses playing the particle system.
Play	Plays the particle system.
SetParticles	Set the particles of this particle system. size is the number of particles that is set.
Simulate	Fastforwards the particle system by simulating particles over given period of time, then pauses it.
Stop	Stops playing the particle system.

Demo



Lab Assignment