Particle Systems

Smoke, rain, explosions, and other effects

Programming – Game Development Foundations

Last modified 18/01/19 by Michael Kalis

aie
SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX

Contents

- What is a particle system?
- Particle systems in Unity
- Particle system performance tips
- Summary
- References

What is a particle system?

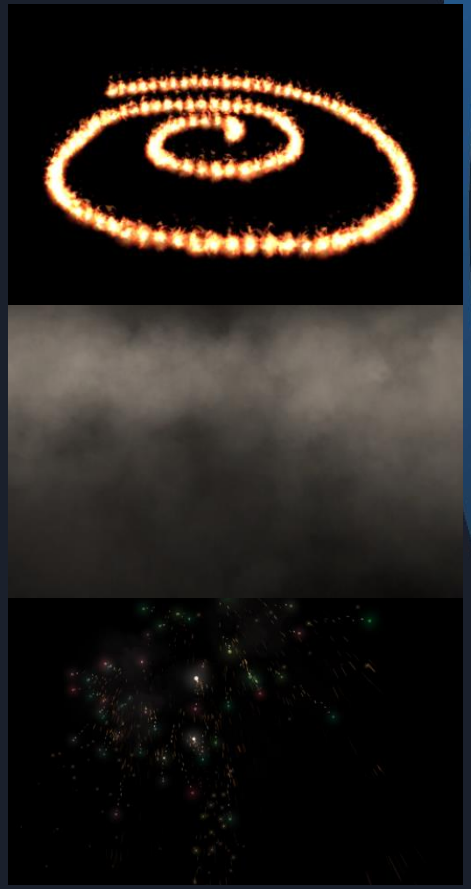
- Most things we render in our games are fairly solid
  - Cars, people, buildings, furniture, props
  - Represented well by 3D models
- Particle systems are useful for non-solid things
  - Smoke, flames, rain, water splashes, spell effects
  - Often difficult to represent as a 3D model
- Still images don’t do good particle systems justice!
  - Motion is an important part of their effect
  - Assets -> Import -> Particle Systems to see for yourself

Car model from Unity’s Standard Assets.

“Flare” particles from Unity’s Standard Assets.

What is a particle system?

- Made up of many individual particles
  - These are “emitted” by the system
- Each particle has a “lifetime” after which it disappears and is replaced
- The most significant part of a particle system is how its particles change over their lifetimes
  - Motion
  - Animated properties – colour, size, transparency



“Wildfire”, “DustStorm” and “FireWorks” particles from Unity’s Standard Assets.

What is a particle system?

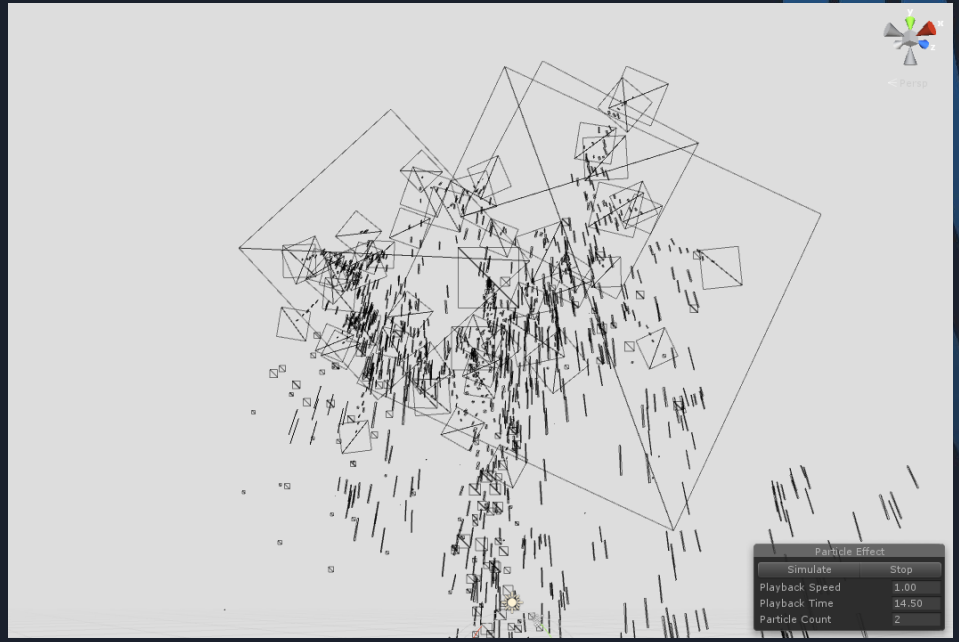
- Multiple “particle systems” can be combined to make a single particle effect.

- The “Explosion” effect is made up of:
  - Starts with fireball system
  - Then has shockwave system
  - Then sparks, smoke and dust systems

“Explosion” from Unity’s Standard Assets.

What is a particle system?

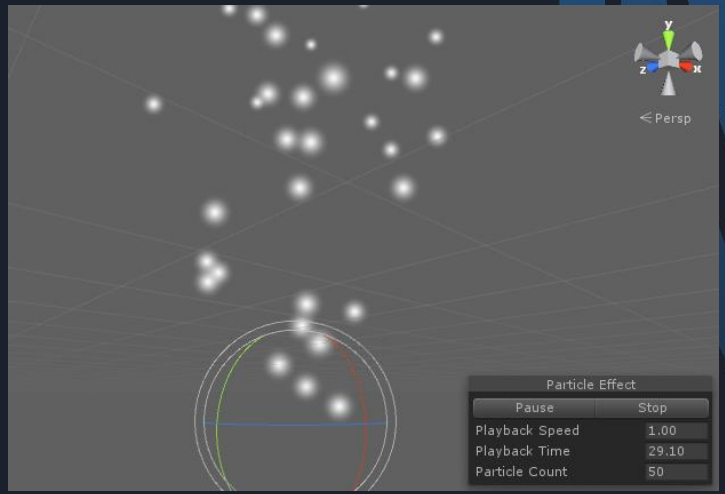
- Individual particles are usually rendered as quads
  - Simplest possible geometry
- Can apply effects such as scaling based on velocity
- Some engines let you use other geometry in particles
  - Be sparing!



"FireWorks" system displayed in wireframe mode.

Particle systems in Unity

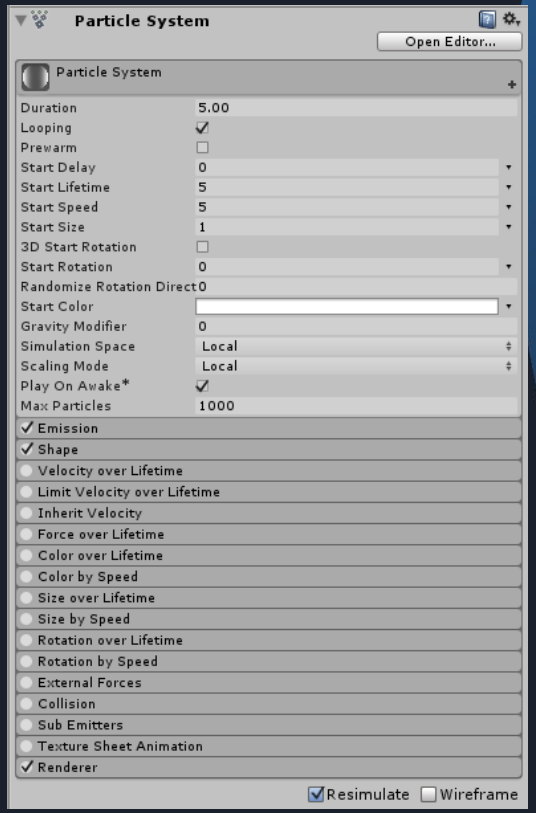
- Unity's particle system is called "Shuriken"
- Create a Particle System via GameObject -> Effects -> Particle System
- When selected, controls will appear in the Scene window



A default particle system in Unity's Scene view.

Particle systems in Unity

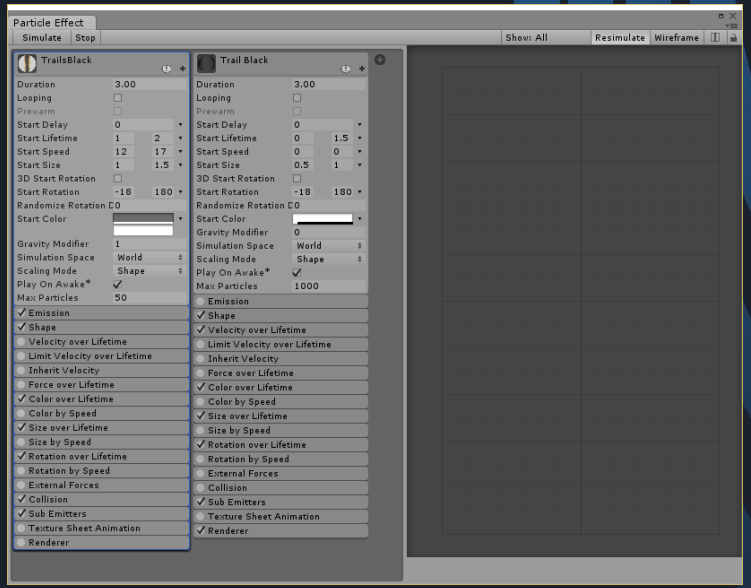
- The Particle System Inspector has many “Modules” which can be individually enabled or disabled.
- We will look at some modules shortly



The Particle System Inspector.

Particle systems in Unity

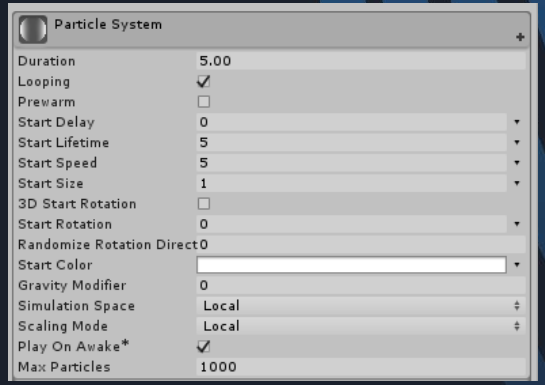
- The Particle Editor is useful for editing nested particle effects together.
- Press “Open Editor” at the top of the Particle System Inspector to open this.



The Particle System Editor.

Base Particle System settings

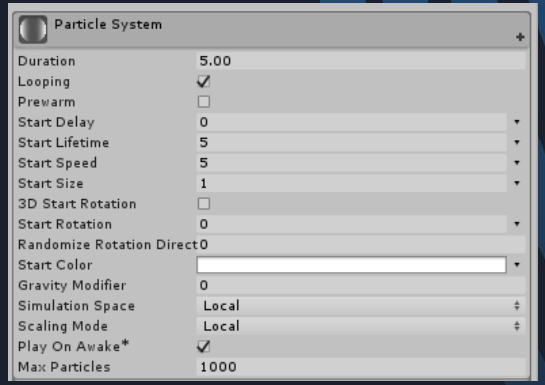
- Common settings for all particle systems
  - Duration: How long the system lasts.
  - Looping: Does it restart when the duration expires?
  - Start Lifetime: The default lifetime of particles emitted by the system.
  - Start Size: The default size of particles emitted by the system.
  - Start Color: The default colour of particles emitted by the system.



The top section of the Particle System Inspector.

Base Particle System settings

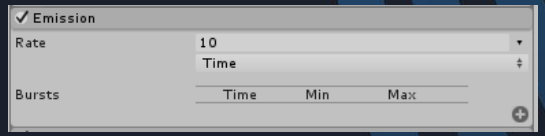
- Common settings for all particle systems (continued)
  - Simulation Space: Do particles move with the system or retain their world positions?
  - Play On Awake: Should the particle system start playing on its own? If disabled you will need a script to start the system.
  - Max Particles: The system will not emit once this limit is reached until some particles expire.



The top section of the Particle System Inspector.

Emission module settings

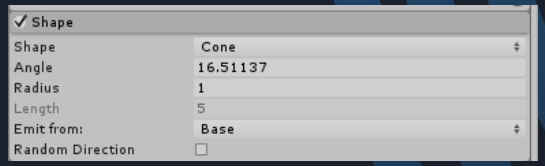
- Controls when particles are emitted by the system
  - Rate can be based on Time or Distance
- Time:
  - Rate specifies particles emitted per second
  - Can also specify “bursts” of particles released at a specific time
- Distance:
  - Rate specifies particles emitted per unit of movement
- Disable this if you want to control emission by script



The Emission module in the Particle System Inspector.

Shape module settings

- Specifies the initial location and direction of emitted particles.
- A number of shapes are available, each with different properties and behaviour.
- Note that initial velocity is one of the base settings of the Particle System

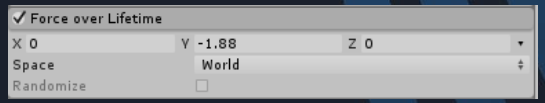


The Shape module in the Particle System Inspector.

Force over Lifetime module settings

- Changes the velocity of particles over time

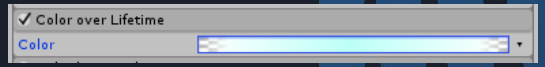
- Very useful for emulating:
  - Gravity
  - Wind



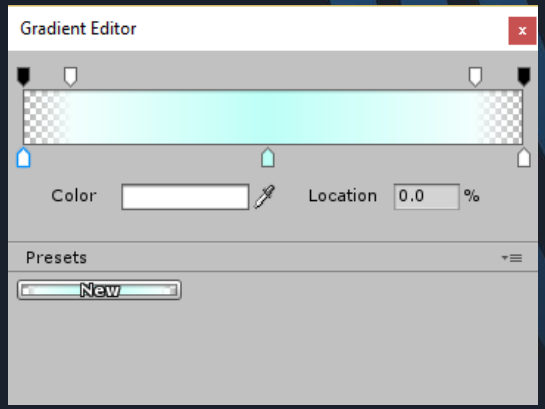
The Force over Lifetime module in the Particle System Inspector.

Color over Lifetime module settings

- Changes the color of particles over time.
- Use the Gradient Editor to configure a colour gradient.
  - Alpha (transparency) values set at the top of the bar.
  - Colour values set at the bottom of the bar.
- Very useful for having particles fade in/out instead of “popping”!



The Color over Lifetime module in the Particle System Inspector.



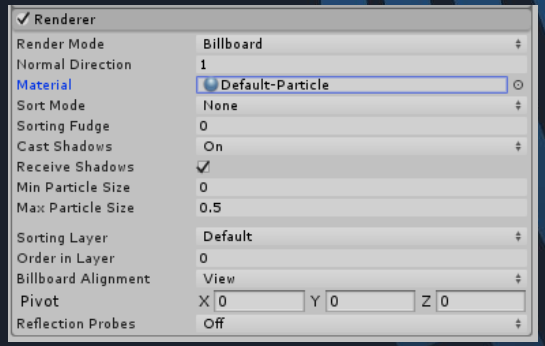
The Gradient Editor.

aie
SPECIALIST EDUCATORS IN GAMES, ANIMATION & FILM VFX

Renderer module settings

- Determines how the system is rendered in the scene
- Provides a number of render modes each with different settings:
  - Billboard
  - Stretched Billboard
  - Vertical / Horizontal Billboard
  - Mesh (use this sparingly!)

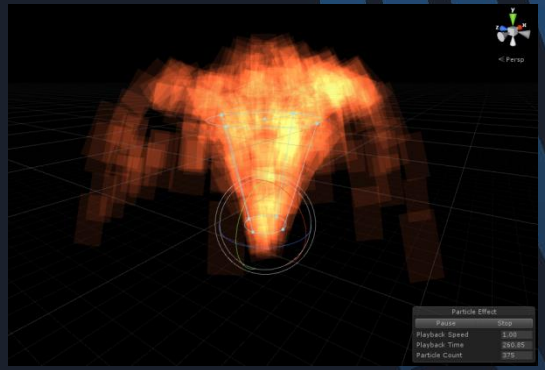
  Try these out and experiment with their settings.
- Allows you to set the Material for the particle system.
  - Make a new Material to change your particle's texture.



The Renderer module in the Particle System Inspector.

Performance tips

- Fewer particles = better performance
  - Saves memory, CPU time and GPU fill rate
- Increase opacity and particle size, decrease particle count
  - This allows you to get a more dense effect with fewer particles
- Fill your particle textures as much as possible
  - Transparent pixels aren't free, so don't just use the middle of your texture
  - Do leave enough room for mip-mapping to work.
- Use the Scene view's Wireframe and Overdraw modes to check that you're getting the most out of your effects.



The Unity Scene view displaying a particle effect in Overdraw mode.

Summary

- Particle systems are useful for effects like smoke, fire, rain
- A particle system has many particles. Individual particles are usually quads.
- Particle systems achieve their effects by moving and changing the display properties of their particles
- Unity's particle systems have their properties and functionality broken into various modules
- Fewer particles = better performance

References

- *Unity Manual*, Particle Systems chapter, Unity Technologies, accessed 21/01/2015
  - http://docs.unity3d.com/Manual/ParticleSystems.html