

Particle Systems Crash Course

How To Initialize

There are two main ways to create a particle system. The first is to go into the Hierarchy and click the plus and then click Effects/Particle System. The second way is to go to an existing Game Object and click “Add Component” and search for the Particle System component. Now that we’re started, let’s look through the different modules available.

Particle System Module

The Particle System Module is arguably the most important module. It contains the initial behaviours of the particle which will be affected in later modules. These are some of the most important parameters:

- Start Color - Allows you to set a single color or a gradient for the particles
- Start Size - Sets initial particle size
- Start Lifetime - Sets how long until a particle will disappear
- Play On Awake - Will the Particle System run when the program starts
- Looping - Will the Particle System loop past it’s initial duration
- Simulation Speed - Speed of particle calculations
- Gravity - What scale of gravity affects the particles

Emission Module

The Emission Module is where you can choose the behaviour of how often particles are emitted. There are 3 main emission modifiers:

- Rate Over Time - How many particles are created over the course of one second
- Rate Over Distance - How many particles are created after a certain amount of distance has been crossed
- Burst - Multiple particles released at the same time. There are different parameters to measure how many particles, what interval bursts will be fired at, and the probability of the burst

Renderer Module

The Renderer Module covers how the particles will look in the game. There are a ton of important parameters so I’ll just go and list some of the most used.

- **Renderer Mode** - This is where you get to decide what kind of particles you're creating.
 - **Billboard** - Renders the particles as sprites that're always facing the camera.
 - **Stretched Billboard** - Similar to Billboard but particles are stretched in the direction of motion.
 - **Horizontal Billboard** - Particles always face up along the y-axis
 - **Vertical Billboard** - Particles always face up along the x-axis
 - **Mesh** - Particles are rendered as 3d meshes
 - **None** - Particles are not rendered
- **Material** - Determines the material for your particle. There are a few preset ones in Unity but by default Billboard particles look like Minecraft XP and Mesh's are grey with simple lighting
- **Sort Mode** - Determines if multiple particles are overlapping which one gets priority
- **Min and Max Particle Size** - Clamps the size of particles to make sure tiny particles don't cause visual artifacts and large particles don't cover the whole screen
- **Cast Shadows** - Do the particles cast shadows onto the world around them

Over Lifetime Modules

There are a number of modules that all cover how a particle will behave as it ages so I'm combining them all together into this one big section.

- **Velocity Over Lifetime** - General control over the speed of the particles throughout the simulation.
 - **Radial** - Speed the particle moves from the origin
 - **Simulation Modifier** - Multiplier of the particles overall speed
- **Limit Velocity Over Lifetime** - Slows down particles over time. Especially good for smoke and fire simulations.
 - **Speed** - Speed limit of the particle
 - **Dampen** - Percent of how much speed is reduced
 - **Drag** - Slows down particles over time as they travel
 - **Multiply by Size and Velocity** - Two different parameters that allow you to alter how drag affects particles based on their size and speed.
- **Forces Over Lifetime** - Calculates how much forces such as wind affect the particles. There are options to adjust each axis's level of influence

- Color Over Lifetime - Gives you a gradient to determine the color particles become as they age.
- Size Over Lifetime - Grow or shrink particles as they age
- Rotation Over Lifetime - Rotate Particles as they age

Collision Module

Particles have the capability of acting as colliders both in a global space and local. In a local space, a plane is defined which gives the particles a place to bounce off of. This is less Cpu intensive. In global, each particle interacts with every collider it hits. You can change how much each particle bounces with the “Bounce” section. Collision is one of my favorite modes because of one tab at the bottom that says “Send Collision Messages”. This tab allows you to use a function called “OnParticleCollision” which lets you determine the logic for every time a particle collides with an object.

Noise Module

Noise Module gives a way to add randomness to the motion of created particles. Unity generates a noise map which it displays on the side and the white and black values of the noise map to the movement of the particles. These are some of the main values to control.

- Strength - How much the noise map affects the velocity of the particles
- Frequency - The detail of the noise or how zoomed out the visual representation of the noise appears
- Scroll Speed - How fast the noise scrolls through it's infinite generation
- Octaves - Changes the detail and grain of the noise. Think of it almost like how harmonics change the quality of a sound
- Quality - Gives 3 different modes: 1D, 2D, 3D. Each dimension increase randomness as it gives more dimensions to translate through

There are many other modules and details within the modules that I didn't touch on but these are the main essential particle modifiers to get you started on making a particle system. For more information, you can see the Unity Documentation on Particle Systems here: <https://docs.unity3d.com/Manual/ParticleSystems.html>

