

Particle System Overview

This is the module responsible for the simulation of particle effects in CryEngine. It provides facilities to create, manipulate and render particles of different kinds.

Chapters:

- [Particle System Overview](#)
- [Concepts](#)

Concepts

Particle System

This is the engine's manager of objects. Games usually start interacting with particles through this object.

Particle Effect

An individual asset usually created by an artist that represents a placeable object. Examples could include fireworks, lightning, magic spells and many more.

Component

Effects are composed of multiple components. Each component represents a particular aspect of the effect. For example, a firework effect might be composed out of rockets, glitter, flash bangs, colorful willows, etc.

Feature

Features implement a particular behavior in a component. Each component contains a stack of features. Is this stack that makes each component function differently from each other.

Modifiers and Effectors

Some features behaviors can be further manipulated over each particle life time or other aspect. Combining diferent features, modifiers and effectors together can create stunningly complex behaviors.

Child (Second Generation)

One of the keypoints of CryEngine's particle system is the Child features. This allows to connect multiple components together as parent-children relationships. What this means is that existing particles can act as parents and spawn children particles.

Attributes

Attributes allows to manipulate a particle effect using external factors usually from FlowGraph or TrackView but also from game code. Attributes serve as a layer of communication between the effect and the game and allows artist full visual control over how an effect should behave in different conditions.

Particle Emitter

When Particle Effects get spawned into a level either as an entity or by game code or other means, They are spawned as Particle Emitters. this object contains the full working logic of a Particle Effect.

Runtime

Furthermore, a Runtime contains the full running logic of a Component and is the one that actually does the simulation and rendering of particles.