COMP2007 - Game Development

Week 8 - Code session

Particles in Unity

Unity has two particle systems available for different purposes:

- Unity Particle System
- Visual Effect Graph

The **Unity Particle System** draws FX to the **default renderer** and uses the **CPU** to process FX.

The **Visual Effect Graph** draws FX to the **High Definition Render Pipeline** system (HDRP) and uses the **GPU** (your graphics card) to process FX. The visual effect graph is capable of **many** more particles due to its using the GPU.

	Particle system (CPU)	**Visual Effect Graph (GPU)**
Render pipeline	Built-in Renderer	URP/HDRP Universal or High Definition pipelines
Number of particles	Thousands*	Millions*
Physics	Yes	Sort of
Script interaction	Yes	Limited to properties
Other	-	Frame buffer access

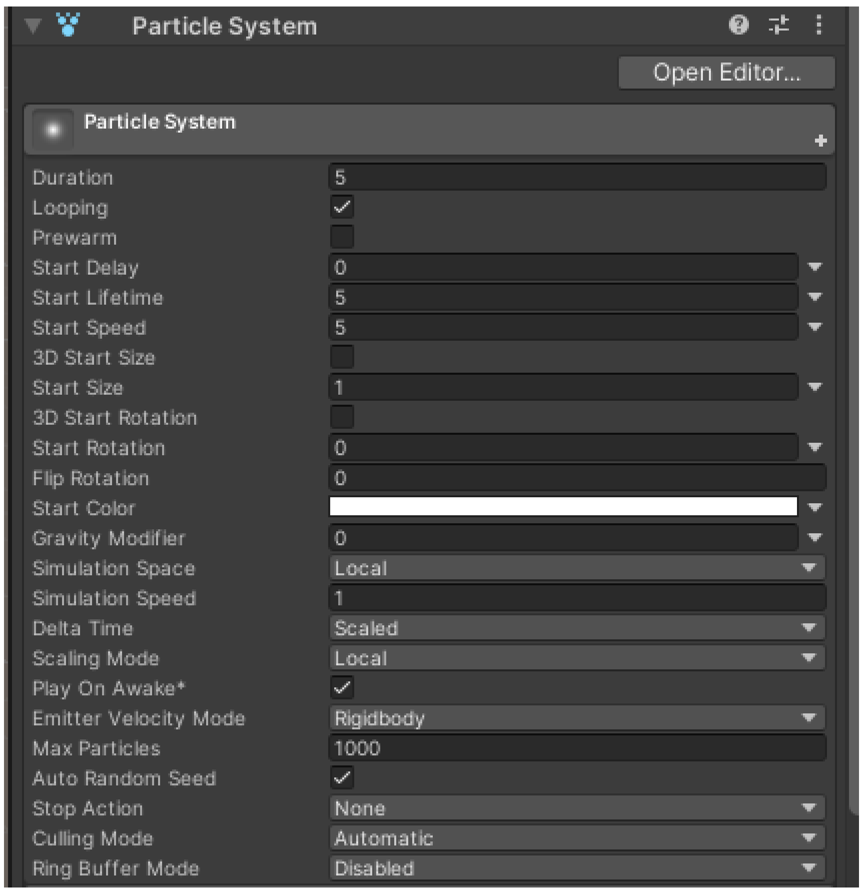
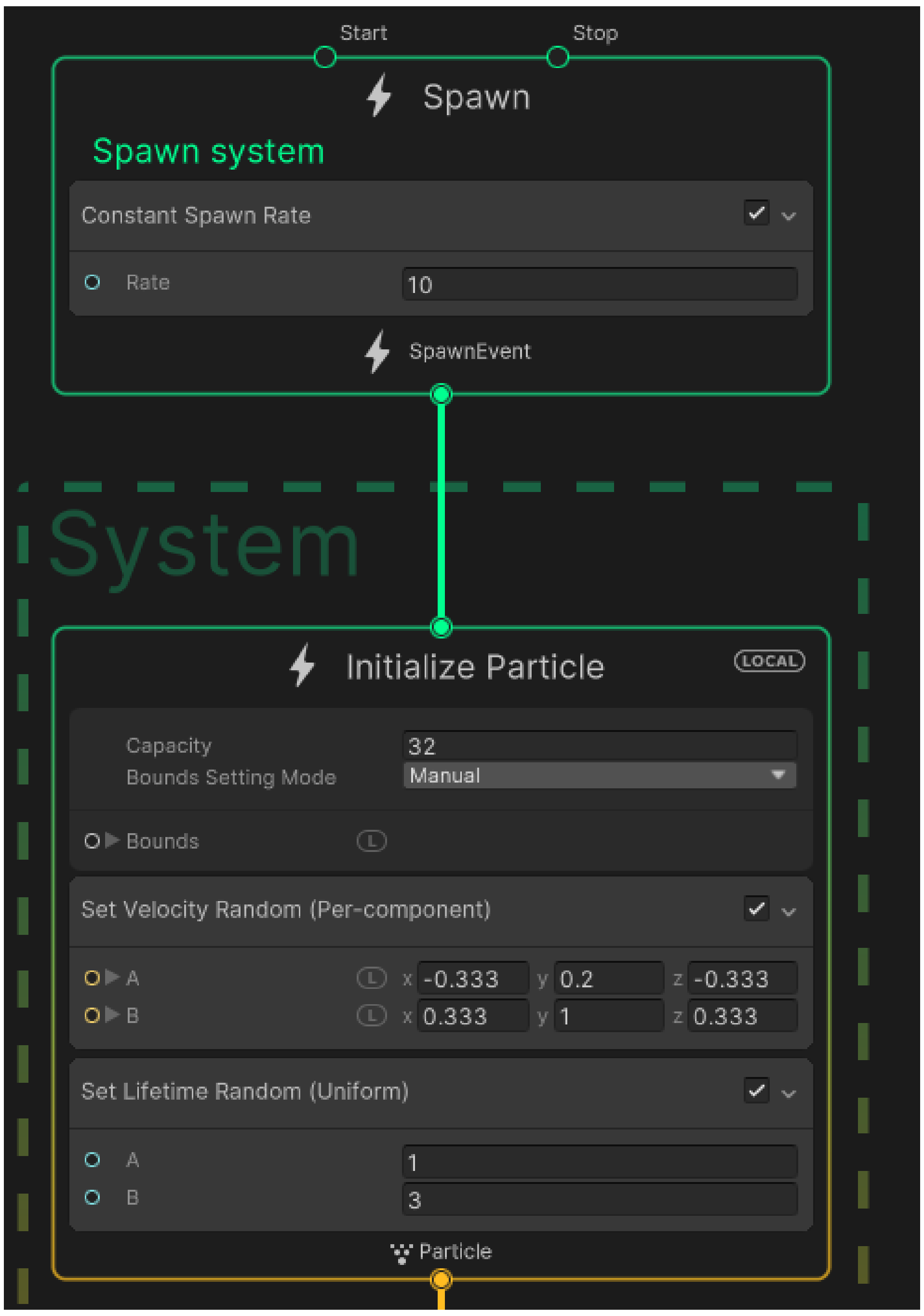
*The number of particles a system is able to handle is dependant on its processing power, so results may vary

These two systems can be used side-by-side in your projects as they have different advantages, depending on your game.

- The Particle system can use physics interactions with its particles, the Visual effect graph cannot.
- We can easily use scripts to change and interact with the Particle System, but script integration is limited for the Visual Graph.

Comparison

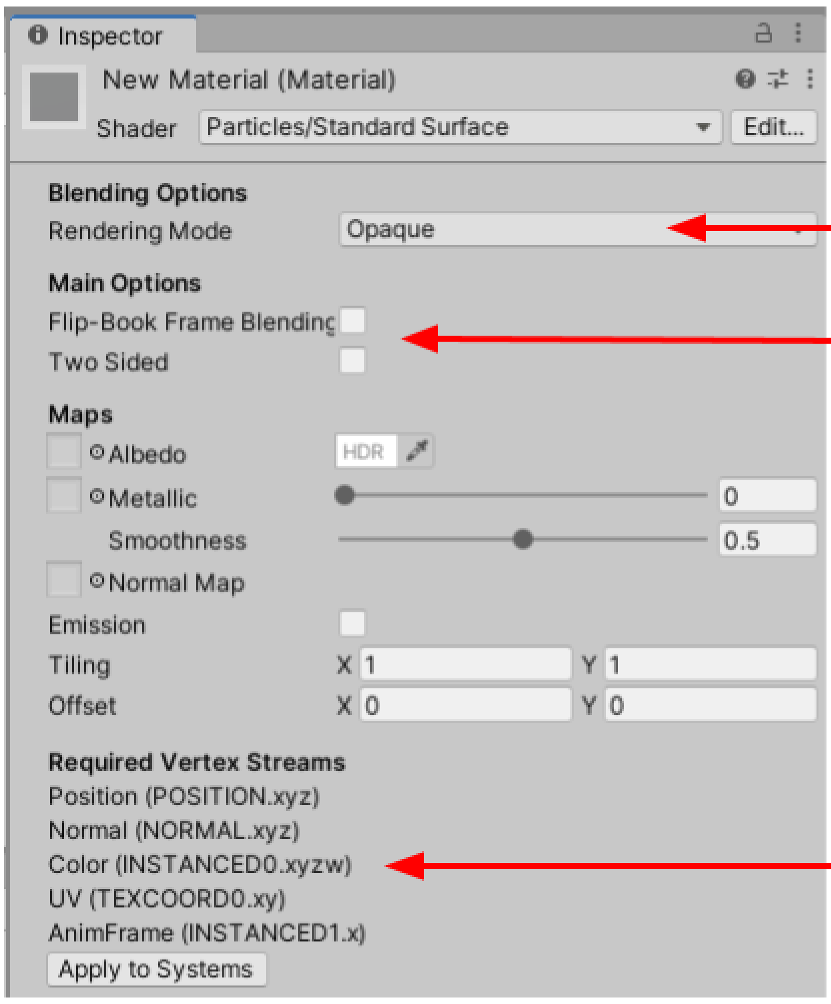
The particle system is contained within a single component, but the visual effect system is edited through the visual effect graph.

Particle System component	Visual effect graph
 Particle System Open Editor... Particle System Duration 5 Looping ☑ Prewarm ☐ Start Delay 0 Start Lifetime 5 Start Speed 5 3D Start Size ☐ Start Size 1 3D Start Rotation ☐ Start Rotation 0 Flip Rotation 0 Start Color Gravity Modifier 0 Simulation Space Local Simulation Speed 1 Delta Time Scaled Scaling Mode Local Play On Awake* ☑ Emitter Velocity Mode Rigidbody Max Particles 1000 Auto Random Seed ☑ Stop Action None Culling Mode Automatic Ring Buffer Mode Disabled	 Start Stop Spawn Spawn system Constant Spawn Rate ☑ Rate 10 SpawnEvent System Initialize Particle LOCAL Capacity 32 Bounds Setting Mode Manual Bounds L Set Velocity Random (Per-component) ☑ A L x -0.333 y 0.2 z -0.333 B L x 0.333 y 1 z 0.333 Set Lifetime Random (Uniform) ☑ A 1 B 3 Particle

Particle System

Unity’s **Particle System** is a type of Renderer that can spawn many smaller images and apply effects to them.
The Particle system can use standard shaders, but also has specific shaders to use with certain fx types like framed animation
There are two types of particle Shaders, **Standard Surface** and **Standard Unlit.**

Standard surface shader



Inspector

New Material (Material)

Shader: Particles/Standard Surface — Edit...

Blending Options
Rendering Mode: Opaque

Main Options
Flip-Book Frame Blending: □
Two Sided: □

Maps
Albedo: HDR
Metallic: 0
Smoothness: 0.5
Normal Map
Emission: □
Tiling: X 1 Y 1
Offset: X 0 Y 0

Required Vertex Streams
Position (POSITION.xyz)
Normal (NORMAL.xyz)
Color (INSTANCED0.xyzw)
UV (TEXCOORD0.xy)
AnimFrame (INSTANCED1.x)

Apply to Systems

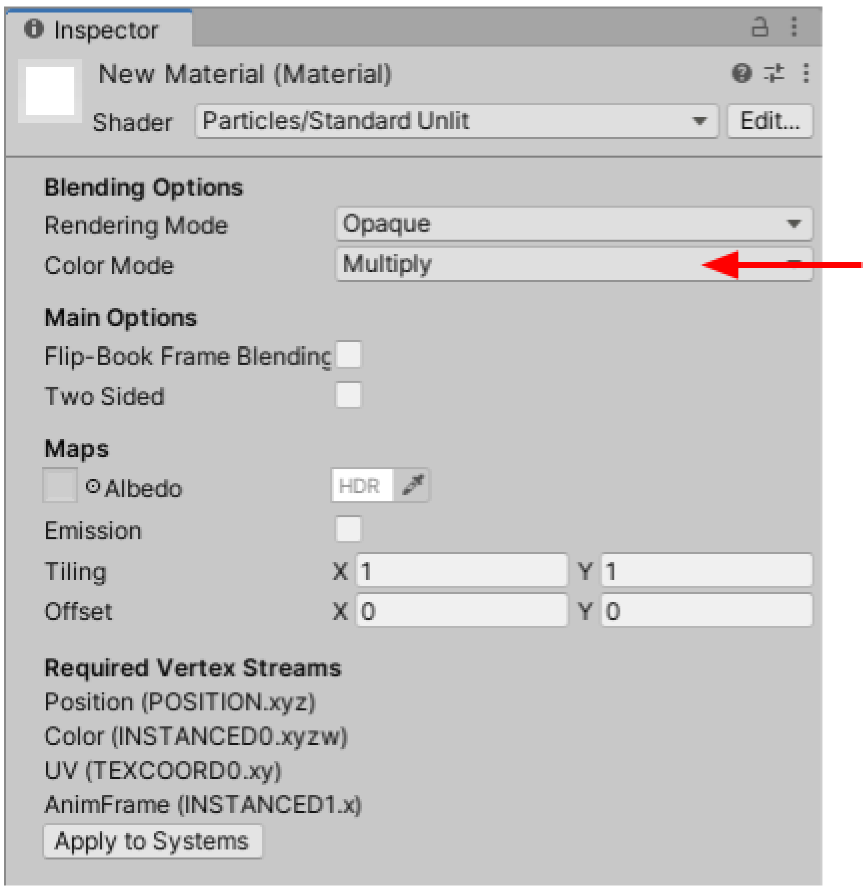
Render settings same as standard materials with some additions

Flip book animation settings

Same options as a standard material

Vertex streams give the shader extra data from the renderer

Standard Unlit



Inspector

New Material (Material)

Shader: Particles/Standard Unlit — Edit...

Blending Options
Rendering Mode: Opaque
Color Mode: Multiply

Main Options
Flip-Book Frame Blending: □
Two Sided: □

Maps
Albedo: HDR
Emission: □
Tiling: X 1 Y 1
Offset: X 0 Y 0

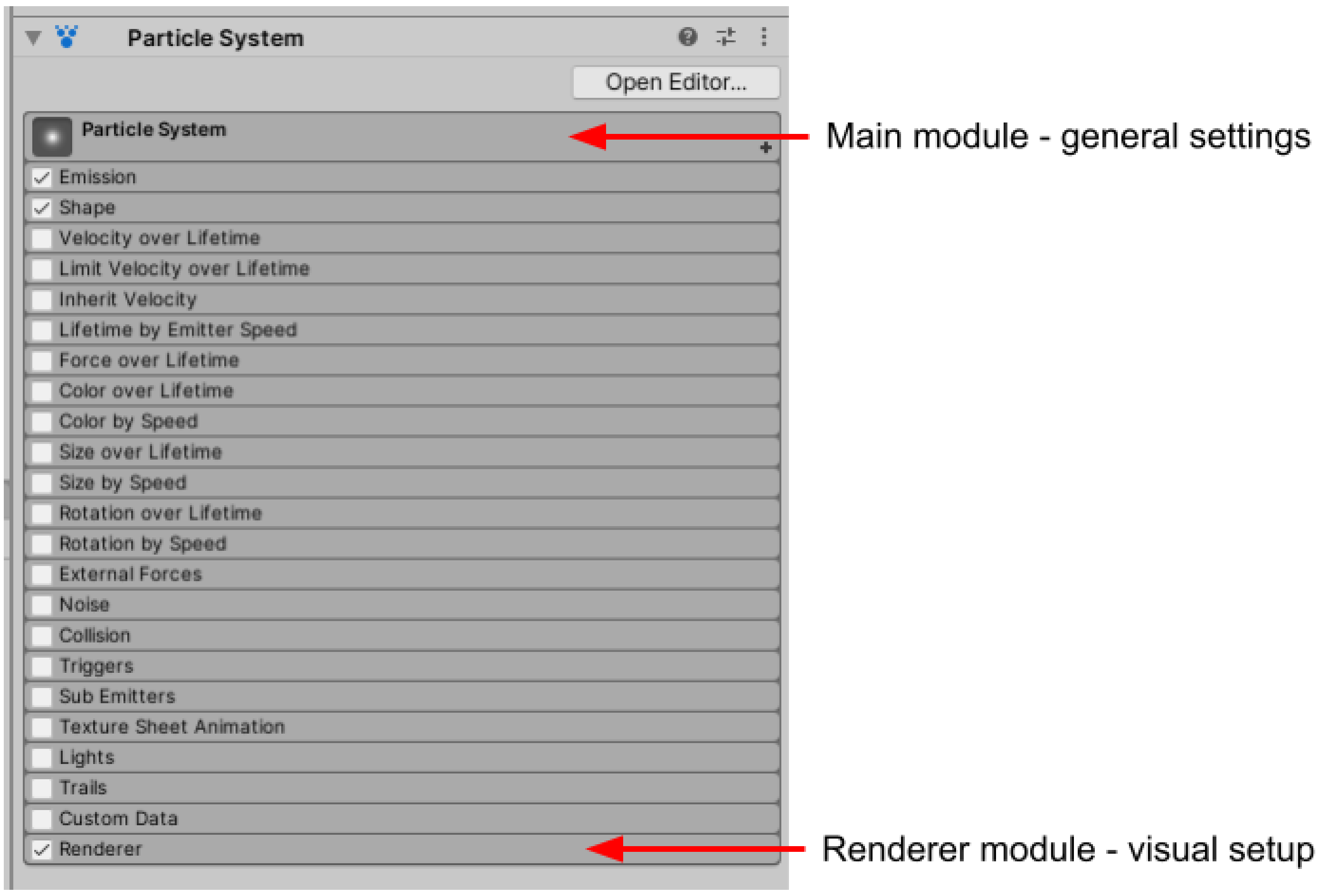
Required Vertex Streams
Position (POSITION.xyz)
Color (INSTANCED0.xyzw)
UV (TEXCOORD0.xy)
AnimFrame (INSTANCED1.x)

Apply to Systems

Color mode can mask the colour from the texture
Similar to photoshop color mode

The Particle System Component

The component has several **Modules** for the different kind of particle FX available
The Main and Renderer modules are for general setup, with other modules for more specific FX
List of all Particle System module settings



Particle System

Open Editor...

Particle System

Main module - general settings

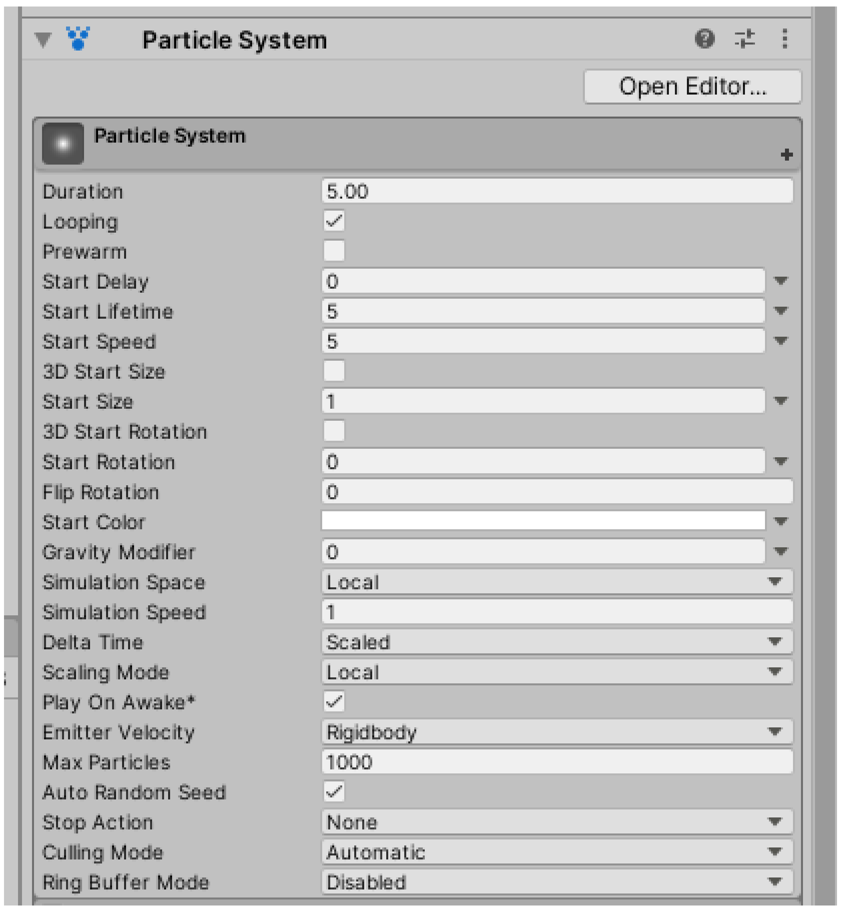
- ☑ Emission
- ☑ Shape
- □ Velocity over Lifetime
- □ Limit Velocity over Lifetime
- □ Inherit Velocity
- □ Lifetime by Emitter Speed
- □ Force over Lifetime
- □ Color over Lifetime
- □ Color by Speed
- □ Size over Lifetime
- □ Size by Speed
- □ Rotation over Lifetime
- □ Rotation by Speed
- □ External Forces
- □ Noise
- □ Collision
- □ Triggers
- □ Sub Emitters
- □ Texture Sheet Animation
- □ Lights
- □ Trails
- □ Custom Data
- ☑ Renderer

Renderer module - visual setup

Main Module

Used for setup of a particle system

- Duration
- Size
- Colour
- Speed



Particle System

Open Editor...

Particle System

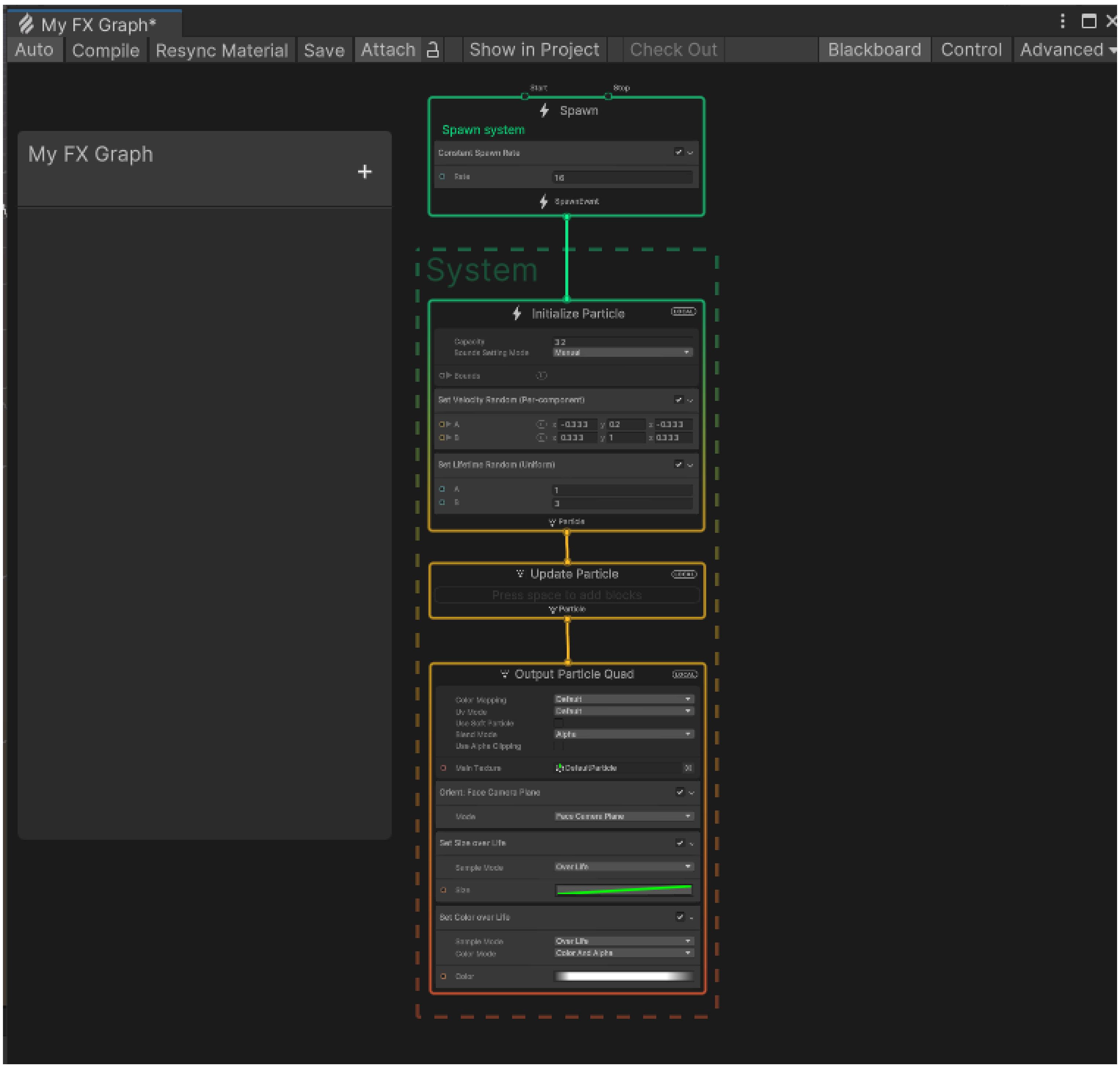
Duration	5.00
Looping	☑
Prewarm	□
Start Delay	0
Start Lifetime	5
Start Speed	5
3D Start Size	□
Start Size	1
3D Start Rotation	□
Start Rotation	0
Flip Rotation	0
Start Color	
Gravity Modifier	0
Simulation Space	Local
Simulation Speed	1
Delta Time	Scaled
Scaling Mode	Local
Play On Awake*	☑
Emitter Velocity	Rigidbody
Max Particles	1000
Auto Random Seed	☑
Stop Action	None
Culling Mode	Automatic
Ring Buffer Mode	Disabled

Visual Effect Graph

NOTE: you must have the HDRP (High Definition Render Pipeline) package installed in your project to use the Visual Effect Graph.
The Visual Effect Graph is an official Unity package we can access from the package manager.

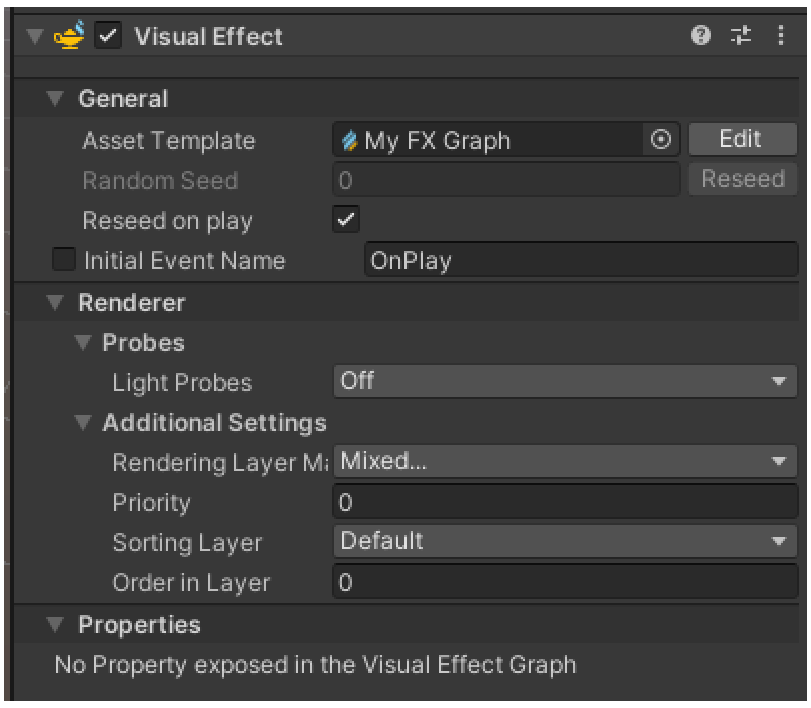
The Visual Effect Graph window is an editor for Visual Effect Graph Assets (FX Assets), which we can use on our GameObjects

The Visual Effect Graph window



An **FX asset** in the Project view, this is the file we can use to add effects to Gameobjects.

We use the **Visual Effect component** to display effects on our GameObjects.
NOTE: the **Asset Template** property is where we add the **FX asset** file.



☑ Visual Effect

General

Asset Template: My FX Graph — Edit

Random Seed: 0 — Reseed

Reseed on play: ☑

□ Initial Event Name: OnPlay

Renderer

Probes

Light Probes: Off

Additional Settings

Rendering Layer Ma: Mixed...

Priority: 0

Sorting Layer: Default

Order in Layer: 0

Properties

No Property exposed in the Visual Effect Graph

Visual Effect Graphs

The graph represents two workflows:
Processing (Vertical)

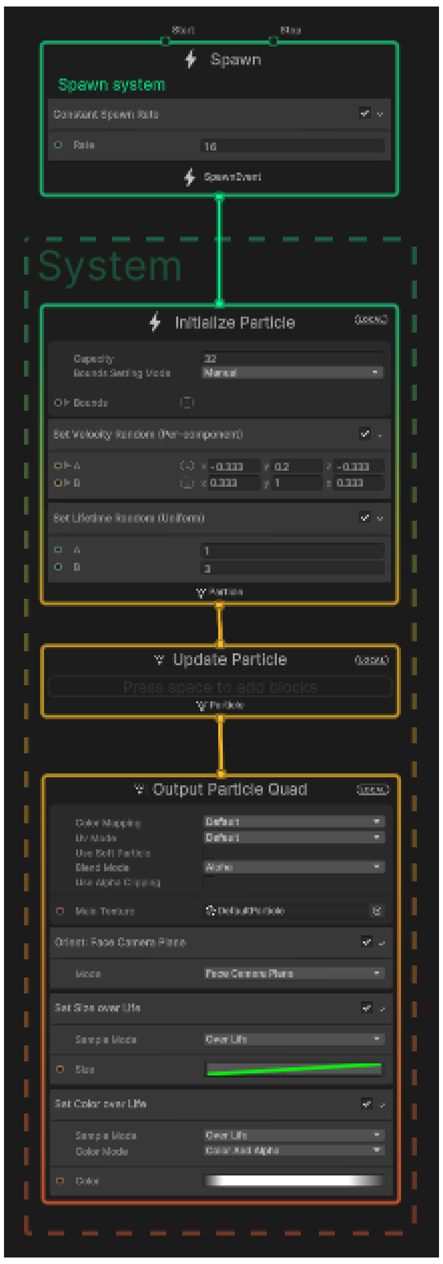
- Defines how a system works and low long an effect lasts

Property (Horizontal)

- Connects different contexts to define the look and behavior of an effect

A graph contans the structure of a effect system
The graph contains a chain of connected contexts, each of which manages a part of the system

A basic graph with a chain of connected contexts (The coloured boxes)



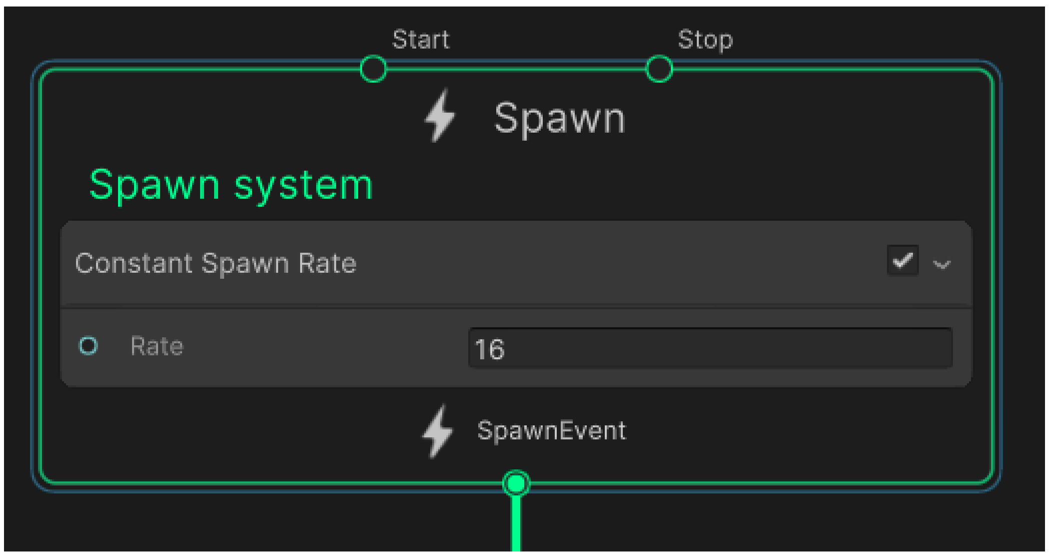
Spawn system
Constant Spawn Rate
Rate 16
SpawnEvent

Initialize Particle
Capacity 32
Bounds Setting Mode Manual
Bounds
Set Velocity Random (Per-component)
A x -0.333 y 0.2 z -0.333
B x 0.333 y 1 z 0.333
Set Lifetime Random (Uniform)
A 1
B 3
Particle

Update Particle
Press space to add blocks
Particle

Output Particle Quad
Color Mapping Default
Uv Mode Default
Use Soft Particle
Blend Mode Alpha
Use Alpha Clipping
Main Texture DefaultParticle
Orient: Face Camera Plane
Mode Face Camera Plane
Set Size over Life
Sample Mode Over Life
Size
Set Color over Life
Sample Mode Over Life
Color Mode Color And Alpha
Color

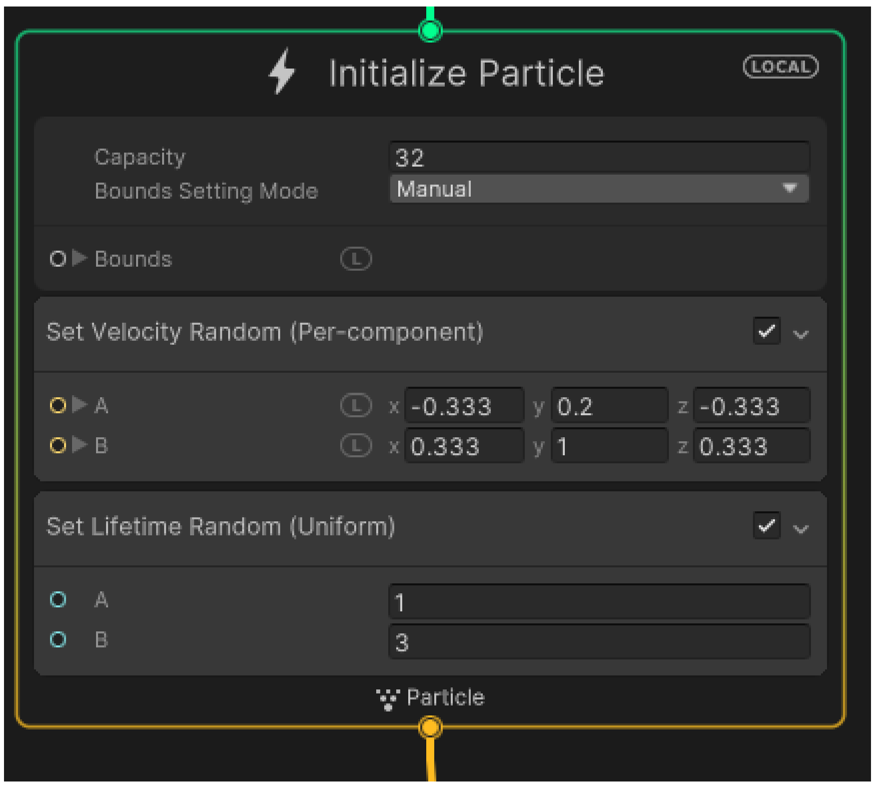
At the top of the graph is the spawn Context, this defines when we spawn particles and how many we can spawn.



Spawn system
Constant Spawn Rate
Rate 16
SpawnEvent

The initialize Context sets up the starting values of our particles.

- Where to spawn
- How long does each particle last
- How many can we spawn

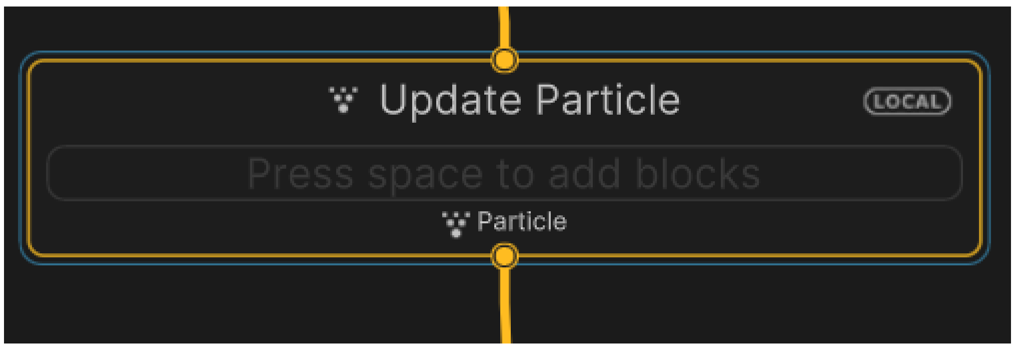


Capacity 32
Bounds Setting Mode Manual
Bounds

Set Velocity Random (Per-component) ✓
A x -0.333 y 0.2 z -0.333
B x 0.333 y 1 z 0.333

Set Lifetime Random (Uniform) ✓
A 1
B 3

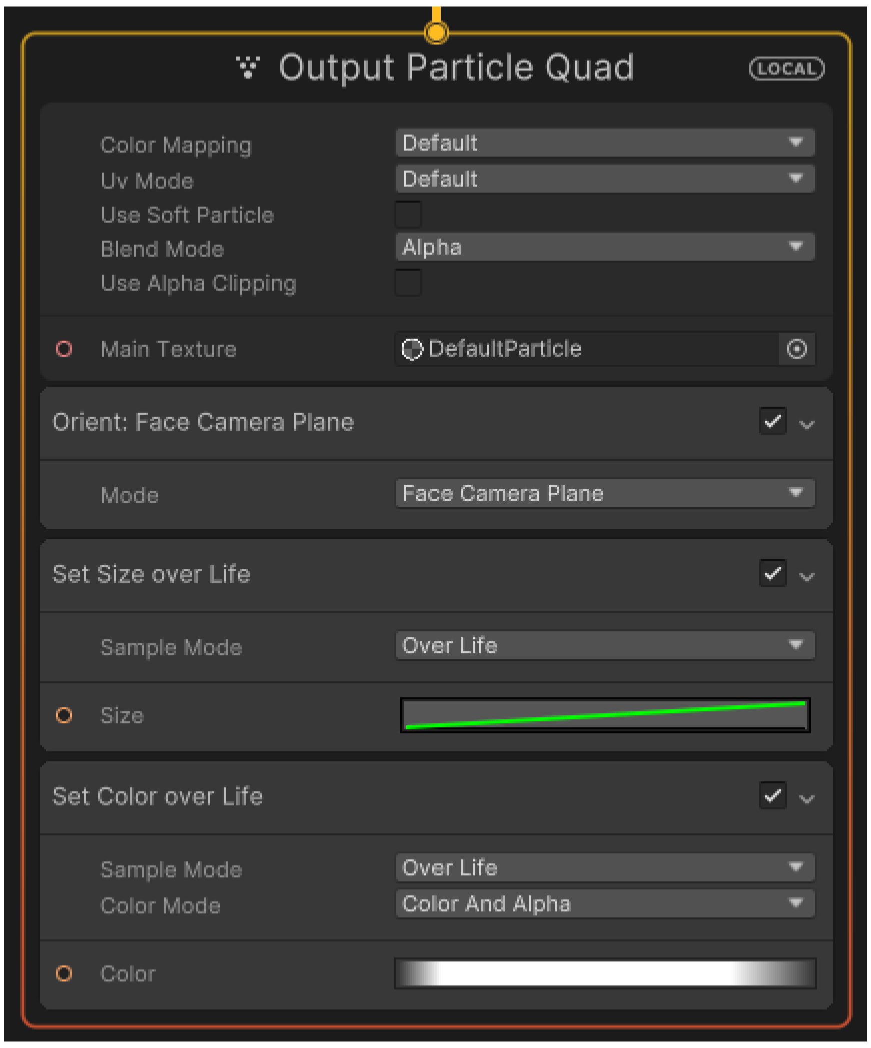
The Update Context allows us to control what happens to the particle over time, similar to our Update method in our C# scripts. We can add Blocks to the context to define specific behaviors.



The Output Context is the final node, where we decide what the final particle will look like and be rendered to the screen.
By default, we can control

- The size
- Colour
- Facing the camera

Note that the output context does not connect to any nodes below, as it is the final render of the effect.



Color Mapping Default
Uv Mode Default
Use Soft Particle □
Blend Mode Alpha
Use Alpha Clipping □
Main Texture DefaultParticle

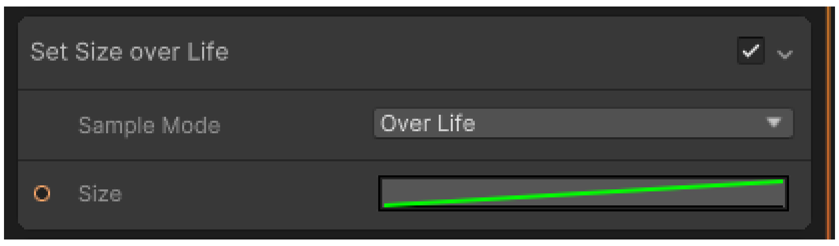
Orient: Face Camera Plane ✓
Mode Face Camera Plane

Set Size over Life ✓
Sample Mode Over Life
Size

Set Color over Life ✓
Sample Mode Over Life
Color Mode Color And Alpha
Color

Blocks

Blocks are small interchangeable features we can add to customize parts of our graph
The default graph comes with some blocks like the one below
Note the block has only a few settings for a specific task.
We can mix and match blocks to create different effects easily.



Set Size over Life ☑

Sample Mode: Over Life

Size

Links

Choosing your particle system

Particle System
List of all Particle System module settings

Visual Effect Graph package documentation

Unity particle pack on the Asset Store

GAME DEV. TECH.

UNIVERSITY OF PLYMOUTH