docs.godotengine.org

Particle shaders

3–4 minutes

Particle shaders are a special type of vertex shader that runs before the object is drawn. They are used for calculating material properties such as color, position, and rotation. They are drawn with any regular material for CanvasItem or Spatial, depending on whether they are 2D or 3D.

Particle shaders are unique because they are not used to draw the object itself; they are used to calculate particle properties, which are then used by the CanvasItem of Spatial shader. They contain only a vertex processor function that outputs multiple properties (see built-ins below).

Particle shaders use a transform feedback shader, which is a special type of vertex shader that runs on its own. It takes in data in a buffer like a regular vertex shader does, but it also outputs to data buffers instead of outputting to the fragment shader for pixel-processing. Because of this, transform feedback shaders can build on themselves each run, unlike other shaders that discard the data they have calculated once they draw to the frame buffer.

Note

Particle shaders are only available in the GLES3 backend. If you need particles in GLES2, use CPUParticles.

Render modes¶

Render mode	Description
keep_data	Do not clear previous data on restart.
disable_force	Disable attractor force. (Not currently implemented in 3.1)
disable_velocity	Ignore **VELOCITY** value.

Built-ins¶

Values marked as "in" are read-only. Values marked as "out" are for optional writing and

will not necessarily contain sensible values. Values marked as "inout" provide a sensible default value, and can optionally be written to. Samplers are not subjects of writing and they are not marked.

Global built-ins¶

Global built-ins are available everywhere, including custom functions.

Built-in	Description
in float **TIME**	Global time, in seconds.

Vertex built-ins¶

In order to use the COLOR variable in a SpatialMaterial, set `use_vertex_as_albedo` to `true`. In a ShaderMaterial, access it with the COLOR variable.

Built-in	Description
inout vec4 **COLOR**	Particle color, can be written to and accessed in mesh's vertex function.
inout vec3 **VELOCITY**	Particle velocity, can be modified.
out float **MASS**	Particle mass, use for attractors (not implemented in 3.1).
inout bool **ACTIVE**	`true` when Particle is active, can be set to `false`.
in bool **RESTART**	`true` when particle must restart (lifetime cycled).
inout vec4 **CUSTOM**	Custom particle data. Accessible from shader of mesh as **INSTANCE_CUSTOM**.
inout mat4 **TRANSFORM**	Particle transform.
in float **LIFETIME**	Particle lifetime.
in float **DELTA**	Delta process time.
in uint **NUMBER**	Unique number since emission start.

Built-in	Description
in int **INDEX**	Particle index (from total particles).
in mat4 **EMISSION_TRANSFORM**	Emitter transform (used for non-local systems).
in uint **RANDOM_SEED**	Random seed used as base for random.