



*HellHeaven Fx is a realtime 3D solution for high-quality rendering in the movie and video-game industry.*

*Mail : [contact@persistant.fr](mailto:contact@persistant.fr)*

*Site : [www.hellheaven.fr](http://www.hellheaven.fr)*

## **Editor**

Customizable layout configurations, keyboard and color customization, etc.

### **Asset workflow:**

Plug-ins for Autodesk 3ds Max allows to import models with mesh topology, mapping coordinates, skeleton structure, animation data, smoothing groups and materials names.

Support for importing content from industry-standard Autodesk FBX file formats.

Built-in support for most common image formats: dds, jpeg, png, exr. Other formats can be added through custom codecs.

Resource watcher allows hot-reloading when a resource is modified through an external editor (photoshop, 3dsmax, maya, etc...)

### **Content Browser**

- Organizing game assets of all types. (create, search, filter, etc ...)

### **Resource Editors**

- Nodal
- Curve
- Script

### **Particle Editor**

- Realtime viewport
- Time control
- Custom mesh backdrops
- Recording system
- Emitter layers

### **Debugging**

- Inspect individual particles by selecting them in the realtime viewport
- Inspect previous update states through the frame history
- Debug particle highlighting
- Particle fields vector visualization (velocity, rotation axis, ...)
- Overdraw visualization
- Realtime visual profiler (graph & histogram visualizations for performance and memory)
- Performance impact estimation in script editor for each source line
- Advanced statistics (particle count, bytes per particle, CPU cycles per particle, ...)

## FEATURES OVERVIEW

### Runtime

Modular high performance C++ runtime and API, optimized for PC. Coming soon on Xbox360 and Playstation3.

#### FULLY SCRIPTABLE

##### Simple

- abstracts controls for a single particle
- user particle fields accessible in-script
- instant feedback during edition
- particle states: one evolve script per state
- dynamic per-particle state switches

##### Efficient

- automatically streamed/vectorized behind the scenes for maximum performance
- can be JIT-compiled to native SIMD code with an appropriate backend
- ability to add a GPU backend that will build shaders or CUDA kernels from the particle scripts

#### SAMPLING

##### Shape sampling (Surface + Volume)

- Spheres, Ellipsoids, Cylinders, Cones, Capsules, Shape collections

##### Curve sampling

- Linear, Hermite
- fully generic curves with arbitrary control points and tangents
- optimized equally-spaced control points
- can be bound to any particle property

##### Mesh sampling (Surface + Volume)

- Uniform or triangle-constant distribution
- Probability density maps
- Sample textures from mesh UVs, and use sampled values inside script

#### SPAWNERS

##### Configurable continuous spawners

- spawn flux curves
- particles per second or total particle count
- stream or burst

##### Trail spawners (particles that spawn particles)

- position and time spawn metrics

##### Performance and precision

- thread pool-friendly update tasks and high performance SIMD implementation
- sub-time step accuracy
- sub-time step probabilistic multi-shape spawns

#### LIGHTING

##### Light-receiving particles

- receive and cast shadows
- opaque particles lit by N lights
- alpha particles currently lit by 1 light

##### Light-emitting particles

- per-particle light radius and color
- custom soft falloff curve
- thousands of lights (light prepass ready)

#### RENDERERS

##### Generic renderer nodes

- Billboards, Lights, Meshes, custom...

##### Various Billboard types

- screen-aligned quads
- screen-aligned triangles (maximizes vertex throughput)
- viewpoint-facing quads
- axis-aligned stretched quads
- 4-point stretched spheroid
- 6-point screen-aligned stretched capsules

##### Texture atlases

- per-particle sub-texture indexing
- animated textures
- random texture assignments
- auto-atlas creation from texture lists

#### NEWTONIAN DYNAMICS

- per-particle mass
- scene collisions
- particle to particle collisions
- bounce restitution
- impulse distribution in world
- solid friction
- fluid drag
- uniform acceleration fields
- force fields
- velocity(wind) fields
- procedural turbulence fields
- closed-form force fields (point-mass attractors)

#### API ENTRY POINTS

##### Extensibility

- fully customizable nodes (evolvers, spawners, renderers, descriptors, etc..)
- scripting entry points allowing custom language extensions
- Hook custom allocators and thread pool managers
- Straightforward access to all particle statistics

##### Raycast queries

- particle/scene collisions through scene raycast queries

##### Per-particle events

- builtin collision and death events
- script-triggered custom events
- hook user callbacks to events
- spawn other particles

##### Particle interaction

- provide velocity & vorticity fields
- specify custom particle-system attributes
- change attributes at runtime