

Assignment 3: Particle Systems

Computer Graphics – CS 4300/5310

Due: *March 2nd, 11:59pm*

Educational Objectives

- Gain comfort with realtime 3D rendering and 3D transformations
- Learn about how to make simple special effects for games and movies

Assignment Description

For this assignment, you will be implementing an interactive particle system in 3D and aiming to simulate a natural phenomenon. You may choose which phenomenon you are aiming for; examples include fireworks, flames, clouds, water, grass, hair, genesis waves, and flocking creatures. Or you might just be trying to make something that looks pretty!

You are required to program this particle system yourself, using either Processing, OpenGL, DirectX, or some other similar graphics library. You may not use a particle system built into existing rendering software or game engines.

Required and Optional Features

This is an intentionally open-ended and exploratory assignment. The following features are absolutely required:

- Particles that move in varying directions from a point-emitter
- Particles with color that varies over time
- Particles that die after either a certain amount of time has elapsed, a certain maximum number of particles on screen has been reached, or some combination of these
- Some kind of physics system – at the very least, they must have a speed that updates their position over time. Depending on the effect you are aiming for, you may also wish to implement acceleration (e.g. due to gravity).

You should then choose different ways to extend it, depending on the kind of effect you want to produce and what grabs your interest. At least one of these features (or a similar effort to go above the base requirements) should be implemented:

- Textured particles
- An emitter that has one or more faces (e.g. a triangle, a cube, a dodecahedron) and particles are emitted from all over the surface rather than from a single point
- Emitters that can turn on and off in a scripted fashion (this is how you would simulate a Las Vegas style water fountain show)
- A particle that spawns a new emitter when it collides with a surface. This is typically how an effect like rain is produced: raindrop particles fall from a surface emitter and then spawn a small emitter for a fountain-like effect when they hit the ground
- Particle that are rendered all at once, rather than distributed over time, commonly used to produce effects like hair, fur, and grass
- Additional physics effects, such as variable wind
- Particles that collide with a volume (either inside or outside the volume)
- Flocking Boids [Based on Craig Reynolds' work, described informally here: <http://www.red3d.com/cwr/boids/>]

Note that some effects are harder to recreate than others, and many involve artists working in concert with engineers. I do not expect students in this class to be artists! Your work will be judged primarily on its technical merit.

Finding Other Resources

There are several resources available online for learning more about particle systems than what was covered in class. I encourage you to seek these out and share them on Piazza, especially if there is a particular effect you are aiming to replicate. Google Scholar is a great resource for finding academic papers, and popular graphics conferences where papers about particle systems might be published include SIGGRAPH and Eurographics. Gamasutra and GameDev.net are good resources for articles and tutorials from the games industry, which commonly employs particle systems for special effects in games.

Submission Instructions

A zip file containing the following must be uploaded to Blackboard:

- Your well-documented code
- A Windows or Linux executable
- **A short video showing your work** – this will be used during our class discussion of assignment 3
- A README listing the number of late days you wish to use, any instructions required for running your code, and a brief description of the effect you were aiming to reproduce and the features you implemented in order to do so

Emailed assignments will **not** be accepted.