

Real-Time Special Effects for a Computer Game Using Particle Systems

Particle System described is part of Parsec

Parsec: 3-D Network Space Fight Game

originally a project for the CG 2 & 3 lab in
the summer term of 1996

Rendering pipeline entirely based on depth sorting of objects represented by BSP trees

Problem #1: All objects have to be b-reps. Too inflexible for weapons: more complex things than simple laserbeam objects, missile objects and the like are not feasible.

Problem #2: No impressive special effects possible. Explosions are simply scaled and animated bitmaps.

Solution: Particle System

yields:

- Particle Weapons
- Particle Special Effects

essentially two components:

- Rendering of Particles
- Behavior of Particles

Particle System Renderer

Every particle is drawn as bitmap scaled proportionally to z-distance to view point.

Bitmap assigned to particle has constant z-value and is efficiently drawn using a z-buffer to solve the visibility problem.

Particles can be grouped into clusters to speed up viewing frustum culling.

Particles \leftrightarrow Polygon Rendering

Polygons of single object are drawn using object-local BSP tree. Whole objects are drawn in depth sorted order.

Polygon drawing never checks the z-buffer, only fills it: No conditional branching in inner loop of texture mapper.

Particles are drawn after all visible polygons have filled the z-buffer appropriately.

Attributes of Particles

- Current position in 3-space
- Current velocity (both speed and direction)
- Corresponding bitmap and size
- Lifetime
- Particle type (important for animation)

Particle type also determines:

- Autonomous particles
- Closely/loosely bound particles

Application: Lightning beams consisting entirely of particles

Main objective is visual appearance, not physically correct simulation.

General direction of beam determined by seed segment: overall linear shape.

Shape is reevaluated every n frames to achieve a sizzling effect.