Particle Systems

Mike Bailey

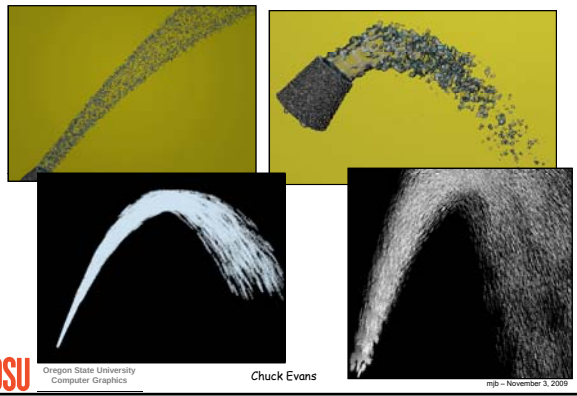
Oregon State University

Particle Systems

- Are used to simulate the appearance of particulate, hairy, or fuzzy phenomena.
- Involve the animation of large collections of tiny particles with various graphics characteristics.
- Were originally developed by Bill Reeves for the film *Star Trek II: The Wrath of Khan*
- Have been used to create effects of fire, smoke, rain, snow, fireworks, disintegration, dust, sand, explosions, flow, waterfalls, stars, comets, plants, hair, fuzz. Surely many more.

Particle Systems Examples

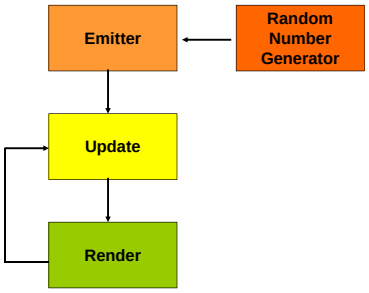
Particle Systems Examples



Chuck Evans

The Process

The basic process is:



The Emitter

The Emitter gives each particle a:

- Birth time
- Death time
- Start location
- Start velocity
- Start color
- Start size
- Start alpha (blending factor)

$$C = (1-\alpha)C_0 + \alpha C_1$$

Creating Random Values for the Emitter

```
#include <stdlib.h>


float
Ranf( float low, float high )
{
        float r = (float) rand();               // 0 - RAND_MAX
        float t = r  /  (float) RAND_MAX;       // 0. – 1.

        return   low  +  t * ( high - low );
}


int
Ranf( int ilow, int ihigh )
{
        float low = (float)ilow;
        float high = (float)ihigh + 0.9999f;

        return (int)(  Ranf(low,high) );
}
```


The Update

And the simulation updates the:

- Position
- Color
- Size
- Alpha
- Interaction with other particles and other objects

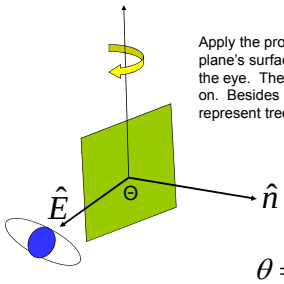
Note that these can change as a function of time, position, or anything else

The Renderer

And the renderer draws the scene using different graphics techniques such as:

- Dots
- Polygons with billboarding
- Quads with textures and billboarding
- Sprites
- Lighting
- Blending
- Smearing

Billboarding



Apply the proper Y-axis rotation such that the plane's surface normal is always pointed towards the eye. The eye always sees the surface head-on. Besides particle systems, this is often used to represent trees simply.

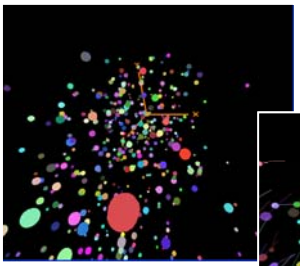
$$\theta = \cos^{-1}\left(\hat{n} \bullet \hat{E}\right)$$

Sprites

A "sprite" is a 3D object pre-rendered to a flat 2D texture and "slipped" into a certain depth in the scene.

http://sdb.drshnaps.com

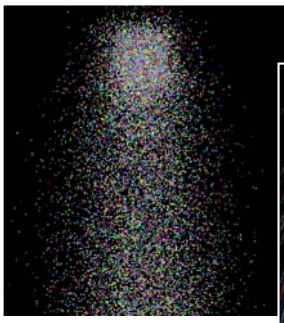
Particle Systems



Circles only

Circles with traces

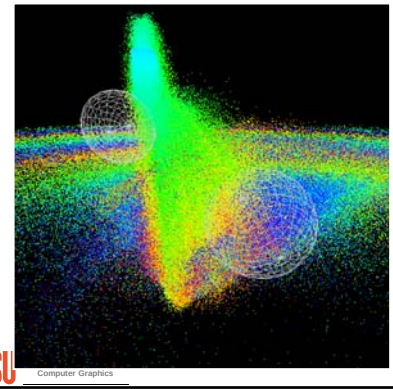
Particle Systems



Points only

Points with traces

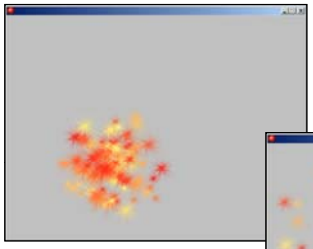
Particle Systems in CS 519



1,000,000 points

Ben Weiss

Particle Systems in Game Maker



http://www.yoyogames.come