USING PARTICLE SYSTEMS TO MODULATE CELEBRATIONS WITH FIREWORKS
Rafael H. C. de MELO (rmelo@ic.uff.br)
Evelyn de A. VIEIRA (evieira@ic.uff.br)
Aura CONCI (aconci@ic.uff.br)
Departamento de Ciência da Computação, IC,
UFF – Federal Fluminense University
Schedule

> Goal

> Basic concepts

> Proposed model

> Results

> Conclusions
Goal

> Simulate all kind of celebrations using fireworks (with different shapes, sizes and colors).

> Construct a model that we can add:
  
  - wind conditions (ideal or real)
  - lightness conditions (ideal and real)
  - smoke
  - sound
  - distance between fireworks
  - lime-light
  - possible obstacles and any other influences that the fireworks may suffer
Basic concepts

> The use of particle systems on computers began on 1983 with William T. Reeves on a project of special effects of the film “Star Trek II: The Wrath of Khan”.

> Using laws of the mechanics of Newton he shows the possibility of creating elements by a collection of particles.

> Particle systems are a way to model objects like fire, clouds, smoke and water.
Particle systems

> An object isn’t represented by an set of elements of primitive surface like polygons, but like clouds of primitive particles that defines its volume.

> A particle system is not a static entity, its particles can change their forms and move themselves.

> New particles are created and old ones are destroyed.

> An object represented by a particle system is not deterministic, its form is not completely specified.
Particle systems

> A particle is an object that has:
  Mass
  Position
  Velocity
  Color

> And could be affected by other forces, like:
  gravity
  Air resistance
  Wind.
Proposed model

> We used, as inspiration, past year celebrations taped on some Brazilian beaches.

> Different types of fireworks were formulated to simulate situations like:
  New Year’s Day
  opening events of:
  - olympics games
  - and world championships.

> The initial position components \((x, y, z)\), like the color \((r, g, b)\) of each firework are taking randomly.
Particle characteristics

> Each new particle has as attributes:
  Initial position (x, y and z);
  Initial speed (x, y and z);
  Size;
  Color;
  Initial transparency;
  Shape;
  Life time;
  Speed that the live goes away (by frame);
  Acceleration (x, y and z).
Particle characteristics

> The position of each particle on each frame could be found by knowing its velocity.

> This can be modified by the acceleration.

> The color, the opacity and size of the particle can be modified by parameters.

> This parameters can be the same to all the particles or stochastic for each particle.
Particle characteristics

> When each particle is created a life time is associated to it.

> After each frame, the life time is reduced by the speed that the live goes away (by frame), when the life time is zero the particle is destroyed.

> When a particle is outside an interest region (the screen of the computer, for example) it could be destroyed.
Fireworks modeled (circle)

\[ v_x = v \times \cos(\theta) \times \sin(\phi) \]
\[ v_y = v \times \cos(\phi) \]
\[ v_z = v \times \sin(\theta) \times \sin(\phi) \]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>(v)</td>
<td>([0.2, 1])</td>
</tr>
<tr>
<td>(\theta)</td>
<td>([0, 2\pi])</td>
</tr>
<tr>
<td>(\phi)</td>
<td>([0, \pi])</td>
</tr>
</tbody>
</table>
Fireworks modeled (square)

\[ v_x = v \times \cos(\theta) \]
\[ v_y = v \times \cos(\phi) \]
\[ v_z = v \times \sin(\theta) \times \sin(\phi) \]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v )</td>
<td>[0.2, 1]</td>
</tr>
<tr>
<td>( \theta )</td>
<td>[0, 2\pi]</td>
</tr>
<tr>
<td>( \phi )</td>
<td>[0, \pi]</td>
</tr>
</tbody>
</table>
Fireworks modeled (iglu)

\[ v_x = v \times \cos(\theta) \]
\[ v_y = v \times \sin(\theta) \]
\[ v_z = v \times \sin(\theta) \times \sin(\phi) \]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>(v)</td>
<td>([0, 2, 1])</td>
</tr>
<tr>
<td>(\theta)</td>
<td>([0, \pi])</td>
</tr>
<tr>
<td>(\phi)</td>
<td>([0, \pi])</td>
</tr>
</tbody>
</table>
Fireworks modeled (disk)

\[ v_x = v \times \cos(\theta) \]
\[ v_y = v \times \sin(\theta) \]
\[ v_z = v \times \sin(\theta) \times \sin(\phi) \]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v )</td>
<td>([0, 2, 1])</td>
</tr>
<tr>
<td>( \theta )</td>
<td>([0, 2\pi])</td>
</tr>
<tr>
<td>( \phi )</td>
<td>([0, \pi])</td>
</tr>
</tbody>
</table>
Real example (inspiration)
Result (instant 1)
Result (instant 2)
Acceleration of examples

> We simulate a little wind through the x (ax) axis.

> The gravity (ay) to have a good estimation of a reality.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ax</td>
<td>0.01</td>
</tr>
<tr>
<td>ay</td>
<td>-0.08</td>
</tr>
<tr>
<td>az</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Conclusions

> Particle systems are a very good approach to simulate fireworks.

> With this technique we can easily construct a model to simulate any kind of celebration that use fireworks.

> And this model could be very useful to a great number of situations considering the design of this type of celebration.
Conclusions

> The project was implemented in C++ using OpenGL and paradigms of object orientation.

> In the future, we can also add to this model some things that were not taking into consideration here like the smoke and sound of the fireworks.

> More illustrations can be found at the web address: http://www.ic.uff.br/~rmelo/fogosArtificio.htm.