

Guillermo Pérez Trueba  
A01377162

## REPORTE ESCRITO

**Tema seleccionado:** Sistema de Partículas.

### Definición del problema

Se define como sistemas de partículas al grupo de objetos que representan algunos fenómenos naturales como las nubes, el humo, el agua, explosiones, truenos, electricidad o el fuego, representados por:

1. Un conjunto de puntos;
2. Un conjunto de reglas que definen su comportamiento.

Los sistemas de partículas representan un gran reto dentro del mundo del modelado de software ya que son objetos difusos, sin un tamaño y estructura definida.

Características de los sistemas de partículas:

1. Un objeto está representado por un conjunto de partículas primitivas que definen su volumen.
2. No es una entidad estática; nace nuevas partículas y mueren las antiguas.
3. El objeto es no determinístico por la forma y naturaleza de los objetos.

### Aplicaciones

En el modelado de sistemas gráficos se utiliza los sistemas de partículas para modelar distintos fenómenos, como pueden ser:

- The Genesis Demo de la película Star Trek II: The Warth of Khan.
- Fuegos artificiales
- Explosiones: utilizados en la película del Retorno del Jedi.
- Pasto: utilizado en la Figura *white.sand* de Alvy Ray Smith (Lucasfilm).

## Estado del Arte

**Autor:** William T. Reeves

**Fecha de publicación:** Julio 1983

**Descripción detallada de la técnica descrita en el artículo.**

Un sistema de partículas es un conjunto de muchas partículas representados por objetos difusos que contienen atributos individuales y cumplen el siguiente ciclo de vida:

1. Se generan nuevas partículas;
2. A cada partícula se le asignan atributos individuales (posición, velocidad, tamaño, color, transparencia, forma y tiempo de vida);
3. Cada partícula que ha existido en el sistema se extingue;
4. Las partículas que no se extinguen, se transforman;
5. Cualquier partícula viviente se renderiza en el frame buffer.

Para poder dibujar cada una de estas partículas, se debe dibujar cada partícula dentro de un determinado frame, los cuales se transformarán conforme avanza el tiempo como se planteó anteriormente. Dentro de estos sistemas de partículas, cada partícula es considerado un sistema de partícula, en donde si el padre cambia, todos los hijos también cambiarán (una nube puede ser representada por un conjunto de gotas de agua), representados por un árbol. Cada uno de estos sistemas cuentan con una *generación de estado* que define la región en que cada una de las partículas nuevas se va a ubicar, el cual definirá la forma, ubicación y dirección de cada una de las partículas del sistema. Cada partícula individual se transforma con el tiempo, en el cual su color, tamaño y transparencia individual se van transformando, al igual que podrían ser sus movimientos (lo cual agrega la capacidad de editar la gravedad). Finalmente, el problema de renderizar objetos que no son primitivos se puede resolver de dos maneras: (1) las partículas no intersecan con otros primitivos y (2) cada partícula funciona como una fuente de luz.

#### **Descripción del trabajo futuro que proponen los autores.**

Se plantea el problema de, como se dijo durante la explicación, poder especializarse y mejorar la técnica de modelado de nubes, ya que estas resultan ser complicadas al ser un conjunto de partículas que dependen de factores externos al sistema.

**Autores:** Tomas Akenine-Müller and Eric Haines

**Fecha de publicación:** 16 de septiembre de 2002

#### **Descripción detallada de la técnica descrita en el artículo.**

Los billboards nos ayudan a representar de una manera más precisa los sistemas de partículas (representaciones de objetos de la naturaleza). El uso de billboards nos ayudará para crear modelos 3D utilizando simples operaciones de dibujos en 2D. Para poder definir los vectores y las direcciones que usarán en los billboards, es necesario tomar en consideración su normal y "up vector".

- **Screen-Aligned Billboard:** utilizado en explosiones. La normal es la negación de la normal de la vista, donde la dirección es un vector visto por la cámara en el **world space**, y el up vector es la cámara.
- **World-Oriented Billboard:** utilizado en sprites circulares, donde el up vector es irrelevante debido a la simetría del objeto. Un problema al utilizarlos es que, al utilizar la misma matriz de rotación, la proyección deforma a los objetos circulares en objetos cilíndricos. Este método es utilizado para modelar flamas, humo y explosiones (eventos caóticos).

- **Axial Billboard:** las texturas no ven de frente al usuario, se gira a través del mundo para ver al usuario tanto como sea posible. Un problema es si el usuario vuela a través de los billboards y mira hacia abajo se pierde el efecto dado por el billboard. Son útiles para representar simetrías cilíndricas.

### Descripción del trabajo futuro que proponen los autores.

No aplica

### Propuesta

El proyecto consistirá en una escena que contenga un sistema de partículas utilizando billboards para crear el efecto 3D, es decir, ver algunos elementos más cercanos que otros dentro de una pantalla 2D. Los efectos que se crearán, es decir, los sistemas con los cuales contará la escena serán:

- Lluvia
- Nieve
- Fuego

Para cada elemento de los sistemas se nos permitirá configurar elementos individuales de cada elemento de los sistemas. Se buscará simular cada uno de estos elementos dentro de un espacio 2D, dando así, un efecto que estos elementos se encuentran en un espacio 3D.

### Metas

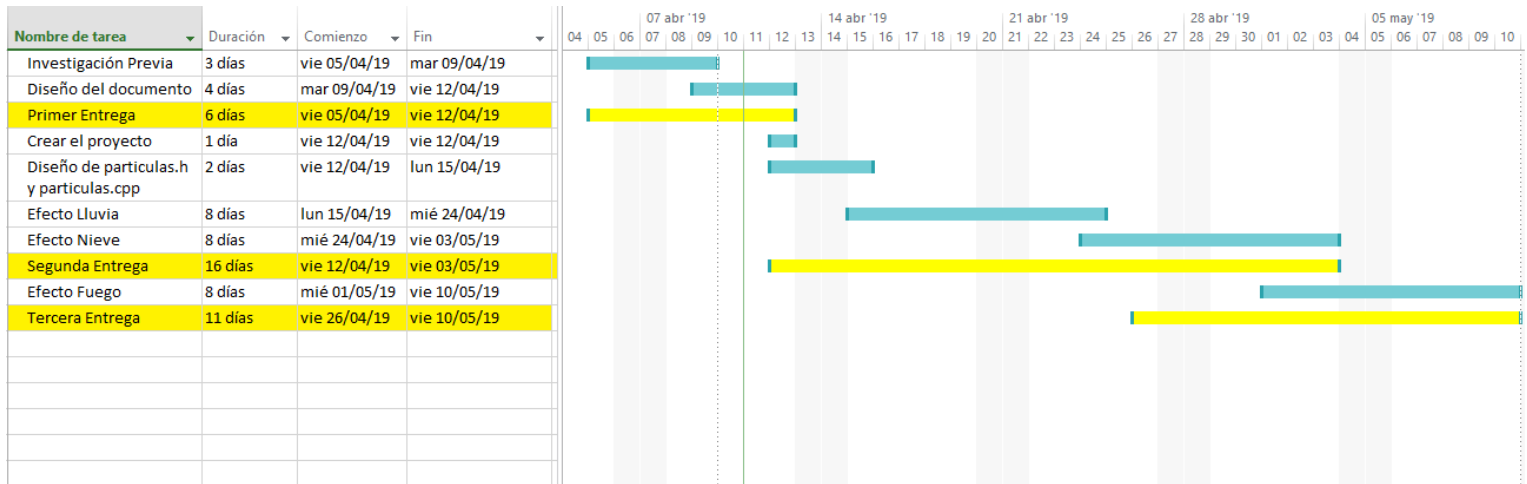
El sistema tendrá un máximo de 1000 partículas. Inicialmente, se tendrá la escena de lluvia con los siguientes valores iniciales:

- float life; //tiempo de vida
- float fade; //duración de la vida
- float[] rgb;
- GLfloat x, y, z; //posición inicial 0,20,0
- GLfloat vx, vy, vz; // velocidad en x, y, z
- GLfloat ax, ay, az; // aceleración en x, y, z

La posición de las partículas cambiará conforme al tiempo con respecto a su velocidad en cada eje. Las partículas deberán tener una aceleración constante negativa, para dar el efecto que van cayendo. El efecto del viento será dado por una aceleración en x. Todas las partículas nacerán en el punto (0, 20, 0). El tiempo de vida de las partículas será variable a cada una, y será determinado por una constante de entre 0 y 1f y almacenado en fade. Este valor se le irá restando a life hasta llegar a 0. Una vez alcanzado este valor, la partícula volverá a nacer con los valores iniciales. Los demás atributos serán constantes durante el tiempo de vida de las partículas.

Para poder cambiar entre escenas, se podrá hacer click sobre los botones '+' y '-' para ir al efecto siguiente o al anterior, siendo el orden: lluvia, nieve, fuego (en caso de estar en el primer efecto se regresará al último). Estos tres efectos no podrán estar al mismo tiempo dibujados en la pantalla.

## Diagrama de Gantt



## Fuentes de consulta

William T. Reeves. (1983). Particle Systems A Technique for Modeling a Class of Fuzzy Objects . <https://www.lri.fr/~mbl/ENS/IG2/devoir2/files/docs/fuzzyParticles.pdf>:

Opengl-tutorial. Billboards. <http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/billboards/>

Opengl-tutorial. Particles / Instancing. <http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/particles-instancing/>

Information Coding / Computer Graphics, ISY, LiTH. Billboards, A texture mapped polygon, which always faces the viewer. <http://computer-graphics.se/TSBK07/files/pdf17/10c%20Billboards.pdf>

Billboarding (2002) Tomas Akenine-Möller and Eric Haines [http://www.flipcode.com/archives/Billboarding-Excerpt\\_From\\_iReal-Time\\_Rendering\\_2E.shtml](http://www.flipcode.com/archives/Billboarding-Excerpt_From_iReal-Time_Rendering_2E.shtml)

[ABB\*07] Andujar C., Boo J., Brunet P., Fairen M., Navazo I., Vazquez P., Vinacua A.: *Omni-directional relief impostors*. *Computer Graphics Forum (Eurographics'07)* 26, 3 (2007), 553–560.