

Object-oriented Graphics Rendering Engine

Práctica 2

2.2 - Sistema de partículas

Alberto Núñez

Departamento de Sistemas Informáticos y Computación

Universidad Complutense de Madrid

- ❑ En esta práctica vamos a crear tres efectos con sistema de partículas
 - ❑ Una estela de humo en la intro para ogrehead
 - ❑ Efecto de fuego en la intro
 - ❑ Humo en el laberinto
- ❑ Recordad que los sistemas de partículas deben definirse en un fichero
 - ❑ Con extensión `.material`
- ❑ En el directorio `media/IG2App` deben estar:
 - ❑ Texturas
 - ❑ Fichero de materiales `.material`
 - ❑ Fichero con los sistemas de partículas `.particle`
- ❑ El primer paso es definir un material para el humo
 - ❑ Tenéis esta definición en el Tema 7

- ❑ Necesitamos

- ❑ Sistema de partículas

- ```
Ogre::ParticleSystem* nombreDelSistemaDePartículas;
```

- ❑ Nodo que adjunta el Sistema de partículas

- ```
Ogre::SceneNode* NodoDondeSeAdjuntaSP;
```

- ❑ Creamos tanto el nodo como el Sistema de partículas en la clase correspondiente

- ❑ Con `setEmitting(bool)` activamos/desactivamos la emisión de partículas

- ❑ Definimos el Sistema de partículas en el fichero `.particle`

- ❑ Utilizará el material del humo definido en el fichero `.material`

- ❑ Características de los billboards

- ❑ Transparencias

- ❑ Sin estar afectadas por la iluminación

- ❑ Sin escritura en el buffer de profundidad

- ❑ Sistema de partículas para ogrehead
 - ❑ Utilizaremos un emisor de tipo Point
 - ❑ Utilizaremos un modificador (*affector*) de tipo ColourFader
 - ❑ Modificaremos la transparencia de las partículas
 - ❑ El color (atributo *colour* del emisor)
 - ❑ recuerda que blanco es 1, negro es 0
 - ❑ translúcido es 0.5, transparente es 0, opaco es 1
 - ❑ Combina valores de los siguientes atributos para conseguir el efecto
 - ❑ Quota
 - ❑ Tamaño de las partículas
 - ❑ Ángulo
 - ❑ Velocidades (minima y máxima)
 - ❑ Tiempo de vida de las partículas

- ❑ Sistema de partículas para el fuego de la intro
 - ❑ Utilizaremos un emisor de tipo Point
 - ❑ Utilizaremos varios modificadores (*affector*) de tipo
 - ❑ ColourFader, ColourImage, Rotator, Scaler y DirectionRandomiser
 - ❑ Para el modificador ColourImage
 - ❑ Utilizaremos la imagen smokeColors.png
 - ❑ Consideraciones
 - ❑ Hay varios sistemas de partículas, utiliza un vector o array para almacenarlos.
 - ❑ El tamaño del plano utilizado en el video es de 300x150
 - ❑ Tamaño de las partículas
 - ❑ Ángulo (es pequeño)
 - ❑ Velocidades (mínima y máxima)
 - ❑ Tiempo de vida de las partículas
 - ❑ Duración (menos de lo que dura la animación)
 - ❑ Se repite en bucle
 - ❑ Utiliza el parámetro repeat_delay

- ❑ Sistema de partículas para el humo del laberinto
 - ❑ Similar al que vimos en clase
 - ❑ Podéis ajustarlo como consideréis a vuestro laberinto
 - ❑ Oscuridad, densidad, dispersión, transparencia, etc.
 - ❑ La idea es que se simule el efecto humo/niebla y se dificulte la visibilidad
 - ❑ Cuidado al asignar/adjuntar el sistema de partículas al nodo del bloque
 - ❑ Dará problemas con la *bounding box* de la perla
 - ❑ Y hará que sea más complicado detectar la colisión
 - ❑ Una posible solución
 - ❑ Utilizar un vector en la clase que gestiona los laberintos
 - ❑ Para cada laberinto, se colocan los sistemas de partículas
 - ❑ Independientes del bloque
 - ❑ Cada uno se indica con una 's' en el fichero del laberinto

- ❑ Sistema de partículas para el humo del laberinto
 - ❑ El utilizado para las capturas utiliza
 - ❑ Emisor de tipo Point
 - ❑ Dos modificadores: ColourFader y Scaler
 - ❑ Duración de 5 sg y se repite en bucle
 - ❑ Quota de 200 (con valores altos, y muchos sistemas, baja mucho los fps)
 - ❑ Ángulo con valor alto (simula mayor dispersión)