

Programación 2D: Prácticas

Máster en Programación de Videojuegos



Javier San Juan Cervera

Presentación

- Hay un total de 11 prácticas a realizar.
- Se proporcionará el esqueleto de un motor gráfico escrito en C++ utilizando OpenGL.
- Cada práctica cubre la materia vista en un tema. Tienen dos partes:
 - Implementación en el motor gráfico de la funcionalidad vista en el tema.
 - Realización de un ejercicio de pruebas que utiliza dicha funcionalidad.

Clase Collision

- Clase abstracta que permite detectar colisiones con otros objetos Collision.
- Sus clases derivadas son: **CircleCollision** (círculos), **PixelCollision** (objetos CollisionPixelData), **RectCollision** (bounding boxes).
- La clase **CollisionManager** calcula las colisiones entre cualquier par de primitivas.

Clase CollisionPixelData

- Define un rectángulo, indicando si cada uno de los puntos contenidos es colisionable o no.
- Se genera a partir de un fichero de imagen, donde alpha = 255 significa colisionable.



Clase Emitter

- Emisor objetos Particle.
- Emite constantemente partículas, con una tasa dada, que pueden tener individualmente propiedades de velocidad, dirección, etc.



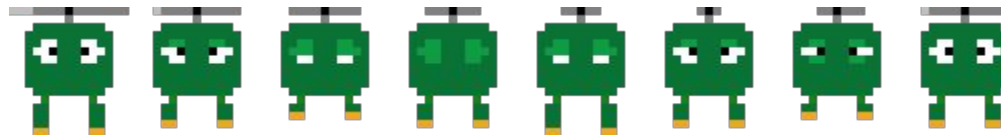
Clase Font

- Carga una imagen que contiene una matriz de caracteres ASCII.
- Permite dibujar textos utilizando los caracteres de la imagen.

HOLA, MUNDO

Clase Image

- Carga un archivo de imagen JPG, PNG, TGA, BMP, PSD, GIF, HDR o PIC.
- El ancho y alto de la imagen debe ser potencia de 2 (requisito de OpenGL).
- Soporta múltiples frames para animación.



Clase Map

- Carga un mapa creado con la herramienta Tiled Map Editor.
- Soporta mapas isométricos con la subclase **IsometricMap**.



Módulo Math

- Provee de funciones matemáticas que amplían las funciones estándar disponibles en C y C++.

Clase Renderer

- Dibujo de primitivas gráficas.
- Dibujo de imágenes.
- Dibujo de textos.
- Establece propiedades de pintado (blending, color...)
- Encapsula todas las llamadas a OpenGL utilizadas en el motor.

Clase ResourceManager

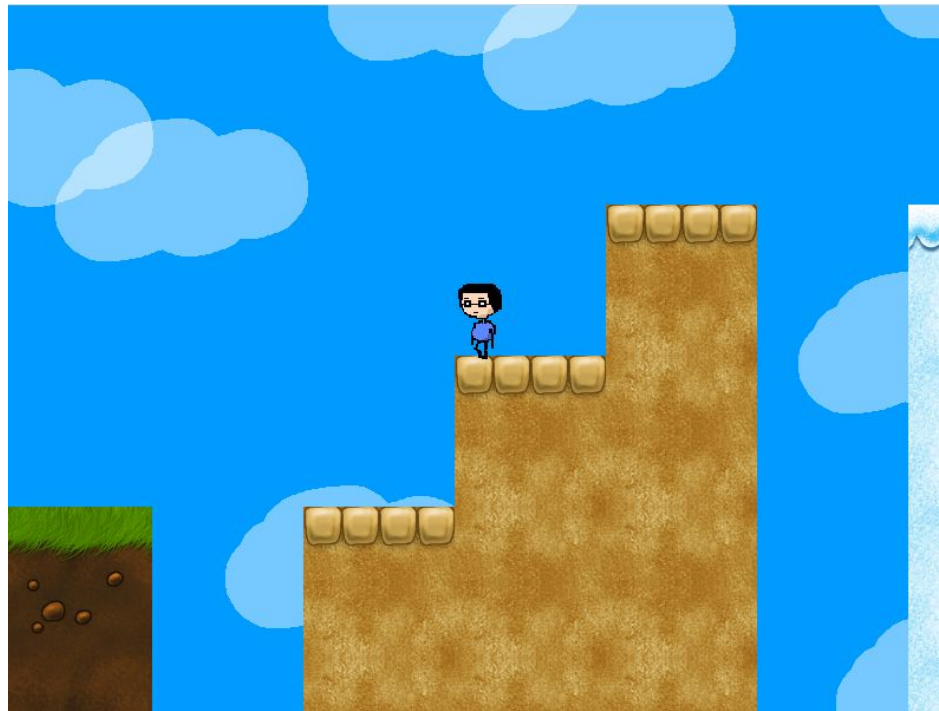
- Gestiona la carga de recursos del disco: imágenes, fuentes, mapas, datos de colisión.
- Controla que un recurso sea cargado una única vez para ahorrar memoria.
- Permite liberar de memoria los recursos cargados.

Clase Scene

- Gestiona escenas que contienen sprites y emisores de partículas.
- La escena se visualiza a través de una cámara.
- Soporta scroll parallax mediante la subclase **ParallaxScene**.
- Soporta escenas con mapas mediante la subclase **MapScene**.

Clase Scene

- Soporta escenas con mapas isométricos mediante la subclase **IsometricScene**.



Clase Screen

- Se encarga de la creación de una ventana o pantalla de OpenGL y de la gestión de los datos de entrada del ratón y el teclado vinculados a esa ventana.
- Encapsula todas las llamadas a GLFW.

Clase Sprite

- Crea una entidad que se representa mediante una imagen, y que tiene propiedades como posición en la escena, rotación, escala, animación, y detección de colisiones.
- Su clase derivada **Particle** tiene propiedades adicionales como velocidad y dirección, para representar partículas.

