Fundamentos de programación gráfica
Práctica 5: Sprites

En esta práctica aprenderemos a implementar la funcionalidad de la clase Sprite en el motor. Un sprite contiene:

- Información visual: una imagen, un modo de pintado y valores RGBA.
- Unas coordenadas en la escena (tiene también coordenada Z, aunque ésta se ignora y será empleada posteriormente para crear sprites con perspectiva isométrica). Como no hemos implementado el soporte para escenas aún, las coordenadas se refieren a la posición de pantalla.
- Un ángulo, que representa la rotación de la imagen en grados sexagesimales y sentido antihorario sobre el eje Z.
- Un coeficiente de escala sobre los ejes X e Y del sprite.
- Otros datos que iremos viendo según corresponda.

Incluiremos por tanto dos nuevos ficheros a nuestro motor, **sprite.cpp** y **sprite.h** (este último debe ser incluido por **u-gine.h** para que la clase esté disponible cuando utilicemos el motor).

De nuevo esta práctica se divide en dos partes que se calificarán de forma independiente.

Primera parte (0,5 puntos):

Partiendo de una clase Sprite que contiene la declaración de las variables miembro y los esqueletos de sus métodos, implementaremos lo siguiente:

- Inicializaremos todas las variables miembro en el constructor.
- Implementaremos los siguientes métodos: SetImage, GetImage, SetBlendMode, GetBlendMode, SetColor, GetRed, GetGreen, GetBlue, GetAlpha, SetPosition, SetX, SetY, SetZ, GetX, GetY, GetZ, GetScreenX, GetScreenY, SetAngle, GetAngle, SetScale, SetScaleX, SetScaleY, GetScaleX, GetScaleY y Render.

Una vez implementada la funcionalidad básica del sprite, modificaremos la práctica 3 para que dibuje la imagen de la pelota de baloncesto (y también la de fútbol si habíamos completado la segunda parte de dicha práctica) utilizando sprites.

Segunda parte (0,5 puntos):

Aunque la clase Sprite permite mover y rotar un objeto cambiando su posición y ángulo directamente, vamos a implementar un sistema que permita realizar estas tareas de forma animada.

En primer lugar, mediante el método RotateTo, debemos poder especificar al sprite que rote desde su ángulo actual hasta un nuevo ángulo especificado en grados sexagesimales. Se especificará además una velocidad de rotación en ángulos por segundo. El sprite deberá encontrar el sentido de rotación por el que llegará antes a la nueva rotación (horario o antihorario).

Si anguloDestino - anguloActual < anguloActual - anguloDestino, entonces el sentido de rotación será antihorario. En caso contrario, será horario. Guardamos este valor, que será el número total de grados que debemos rotar.

En el método Update de la clase Sprite, debemos ir actualizando la rotación del sprite a la velocidad indicada hasta que lo hayamos girado el número total de grados.

En segundo lugar, mediante el método MoveTo, podremos realizar una tarea similar a la anterior, que dado un punto de destino y una velocidad (con uno o dos valores, según queramos especificar una velocidad uniforme o bien una velocidad para el desplazamiento horizontal y otra para el vertical), mueva el sprite de sus coordenadas actuales al punto dado.

Si únicamente se ha especificado una velocidad (es decir, el segundo parámetro de velocidad es 0), debemos calcular la velocidad horizontal y vertical para que llegue al punto de destino de manera lineal (que alcance las coordenadas X e Y de destino al mismo tiempo). Lo podemos hacer calculando en primer lugar el tiempo que nos va a llevar desplazarnos la distancia entre el punto de origen y el de destino a la velocidad dada. Para obtener la velocidad horizontal y vertical necesarias, dividimos la distancia horizontal y vertical respectivamente entre el tiempo que va a llevar el movimiento, con lo que obtenemos los dos valores de velocidad deseados.

Al igual que para el caso de la rotación, debemos añadir al método Update de la clase Sprite el código necesario para que realice el movimiento a la velocidad indicada.

Finalmente, probaremos que se ha implementado correctamente esta funcionalidad realizando una aplicación de prueba en la que cargaremos el sprite **“data/alien.png”,** y podremos mover dicho sprite por la pantalla utilizando las teclas de cursor. Cuando el sprite se está moviendo hacia la izquierda, debe girar hasta un ángulo de 15 grados; si se mueve hacia la derecha, debe girar hasta 345 (o −15) grados. Si no se mueve en ninguna de estas dos direcciones, se debe llevar de vuelta a un ángulo de 0 grados.

Todos los movimientos y rotaciones de esta parte de la práctica se deben hacer con los métodos MoveTo y RotateTo.

Una posible solución daría el siguiente resultado: