

Hoja de ejercicios del Tema 3

1. Se trata de mostrar una escena animada muy sencilla con un fondo y un par de elementos animados que se mueven por la pantalla. Sigue los siguientes pasos:

- Escribe un programa que cree una ventana SDL de 800x600 de título “My first test with SDL”, cargue la imagen `background.png` (incluida en la plantilla de proyecto SDL para Visual Studio), la muestre ocupando toda la ventana SDL, espere 5 segundos (ver función `SDL_Delay`) y se cierre.
- Extiende el programa del apartado anterior de manera que el programa además cargue la imagen `dog.png`, y muestre el primer frame de la izquierda escalado a 100x100 encima de la imagen de fondo en la parte izquierda sobre el césped.
- Escribe un bucle que se repita hasta que se cierre la ventana (evento `SDL_Quit`, ver pág. 15 de las diapositivas del tema 3) y que vaya moviendo y repintando el perro de izquierda a derecha. Cuando el perro desaparezca por la derecha debe volver a aparecer por la izquierda de manera cíclica.
- Usa `SDL_GetTicks` y/o `SDL_Delay` (ver documentación de SDL) para que el movimiento se produzca a menor velocidad. Haz que el perro se desplace aproximadamente 10 píxeles cada 50 ms.
- Mejora la animación de manera que cada 100 ms (define la constante correspondiente) se cambie al siguiente frame de la textura del perro cíclicamente (de izquierda a derecha) al mismo tiempo que el perro se desplaza de izquierda a derecha. Parecerá que el perro se desplaza corriendo.
- Añade una animación análoga de un helicóptero (`helicopter.png`) moviéndose por la parte superior de la ventana de derecha a izquierda (y volviendo a aparecer por la derecha de manera cíclica).

2. Implementa el soporte necesario en el programa del ejercicio anterior de manera que tanto el perro como el helicóptero pasen de estar parados a ponerse en movimiento y viceversa mediante la pulsación de una tecla (tecla `h` para el helicóptero y tecla `d` para el perro). Implementa también el soporte para que la velocidad de movimiento del perro y el helicóptero se pueda aumentar y disminuir (teclas `j` y `g` para aumentar/disminuir la velocidad del helicóptero y teclas `f` y `s` para el perro).

3. Re-estructura el código del ejercicio anterior usando clases y métodos (ver diapositivas del tema 2). Primeramente hazlo para la versión del apartado f) del ejercicio 1 y después ya para las nuevas funcionalidades del ejercicio 2. Debes definir al menos las siguientes clases:

- Clase `Texture`: contiene información relativa a una textura (puntero a `SDL_Texture`, anchura, altura, etc.) y métodos para cargarla y pintarla.
- Clases `Helicopter` y `Dog`: contienen al menos un puntero a su textura asociada (`Texture*`) y métodos para pintar y actualizar sus posiciones.
- Clase `Game`: contiene información de los objetos del juego y métodos para pintar y actualizar.

4. Se quiere implementar un juego que contiene un laberinto de un tamaño fijo (NROWS x NCOLS). El laberinto se representará como una matriz de booleanos y se pintará colocando una textura de un ladrillo o muro en las posiciones en las que haya muro sobre un fondo fijo. Implementa la clase Game con al menos los siguientes métodos:

- **bool** load(): carga el contenido de la matriz de un fichero que contiene ceros y unos (con espacios y saltos de línea como separadores de las columnas y filas respectivamente).
- **void** render(): pinta en la ventana SDL el laberinto como se ha indicado.

5. Extiende el juego del ejercicio anterior de manera que tras cargar el laberinto aparezcan sobre ciertas posiciones libres (elegidas aleatoriamente) una serie de bolas (opcionalmente de distintos colores). La matriz del laberinto ahora ya no será de booleanos sino de un tipo enumerado con tres posibles valores {Empty, wall, Ball}. Conviene además añadir un atributo con el vector de bolas (clase Ball). Reimplementa los métodos load y render de la clase Game. Implementa además el soporte necesario para que al pulsar el cursor de arriba/abajo todas las bolas se desplacen hacia arriba/abajo hasta que no puedan moverse más (como si el laberinto se inclinase hacia arriba/abajo).

6. Reimplementa el ejercicio 7 del tema 1 usando SDL para la visualización del tablero y orientación a objetos.

