



## Introducción al desarrollo de videojuegos con SFML



### Actividades

Tomando como base el código del ejemplo realice las siguientes actividades

1. Agregar a la escena un nuevo sprite con una textura diferente. No hace falta que el sprite se mueva. Tenga en cuenta realizar los siguientes pasos:
  - Crear una nueva variable para la nueva textura
  - Cargar la textura desde un archivo (utilice el método `loadFromFile()`)
  - Crear una nueva variable para el sprite
  - Asignar la textura al sprite (utilice el método `setTexture()`)
  - Opcionalmente, puede darle una posición al sprite utilizando el método `setPosition()` (¿En qué posición aparecerá el sprite si no se invoca a éste método?)
  - Finalmente, no olvide dibujar el sprite en la escena llamando al método `draw()` de la **RenderWindow**.
2. Agregar una imagen de fondo.
3. Cambiar el ángulo del sprite agregado en el punto 2 utilizando el método `setRotation()`.
4. Agregar un sprite que se mueva solo a una razón de 3px en el eje x y a 2px en el eje y por cada iteración.
5. El sprite del ejemplo (la cara anaranjada) es bastante grande, modifique su tamaño (escala) utilizando el método `setScale()`
6. El sprite del ejemplo (la cara anaranjada) se mueve sólo hacia la derecha, hacer que se mueva en las otras direcciones utilizando las flechas correspondientes.
7. Agregar a la escena un nuevo sprite (elegir la textura) que siga al cursor del raton (ayuda: puede utilizar `setPosition()` para posicionar el sprite y `sf::Mouse::getPosition()` para obtener la posición del ratón).
8. Si realizó correctamente los puntos anteriores, ahora la cara anaranjada puede moverse en todas las direcciones. Teniendo en cuenta las dimensiones que eligió para la ventana y las dimensiones del sprite, realice las verificaciones necesarias para que éste no pueda salir de la pantalla.
9. Realice las modificaciones necesarias para que cada 3 segundos aparezca una nueva cara azul en la escena (no hace falta que se muevan). Pista: utilice un contador para saber cuándo pasaron 3 segundos, recuerde que (ya que en el ejemplo se impone un límite al framerate) cada iteración del bucle `while` toma 1/60 segundos.
10. Hacer que las caras azules emergentes que se programaron en el punto anterior se muevan con una velocidad de 10px en el eje x y 10px en el eje y, y que al llegar al borde de la ventana reboten, en lugar de salir.
11. **BONUS:** hacer que la velocidad de las caras emergentes programadas en el punto anterior sea aleatoria y varíe entre -10 y 10 para ambos ejes.
12. Hacer que la ventana se cierre y el juego termine si la cara anaranjada choca con alguna de las caras azules programadas en el punto anterior Ayuda: utilice el método `getGlobalBounds()` de los sprites, dicha función devuelve el rectángulo (**FloatRect**) que cada sprite ocupa en la pantalla, luego utilice la función `intersects()` de **FloatRect** para saber si un rectángulo se intersecta con el.
13. Guardar el puntaje del jugador en una variable, el puntaje es la cantidad de segundos que el jugador duró esquivando las caras azules sin que ninguna de ellas toque la cara anaranjada. Ayuda: utilice un contador y recuerde que cada iteración del bucle `while` toma 1/60 segundos.
14. **BONUS:** cada 5 segundos debe aparecer un item (elija la textura para el mismo). Si el jugador lo junta gana 10 puntos más. No puede haber más de un item simultáneamente en la escena. Cuando el jugador logra tocar el item se empiezan a contar nuevamente 5 segundos para la aparición del próximo.
15. **BONUS:** Agregue un texto en la escena que muestre el puntaje del jugador en un momento dado. Para ello, vea el tutorial sobre fuentes y texto.

### Material de referencia

- [Funciones de Sprite](#)
- [Funciones de Texture](#)
- [Funciones de Mouse](#)
- [Funciones de Keyboard](#)

- [Funciones de `RenderWindow`](#)

## Tutoriales

- [Crear ventana de juego](#)
- [Dibujar en la ventana de juego](#)
- [Sprites y texturas](#)
- [Teclado, mouse y joystick](#)
- [Fuentes y texto](#)

NOTA: No es necesario leer cada tutorial completo. Las partes que nos interesan generalmente están explicadas en las primeras secciones y el resto es para profundizar o hacer cosas más específicas que por el momento no nos interesan.