

Trabajo Práctico nº 2

Game Loop con Clases y Objetos en Visual Studio C++

Trabajo Preliminar

Crear una solución llamada **TrabajoPractico-3** que contendrá los proyectos: **Ejercicio-0, Ejercicio-1, Ejercicio-2.**

Ejercicio 0

Deberá realizar un programa que dibuje un marco de 1,1,24,80 e imprima * a medida que se mueva según se presionen las teclas flecha, el programa terminará cuando se toque cualquiera de los bordes o se presione la tecla escape. Deberá utilizar la librería.

El programa comienza con el * en el centro de la pantalla y muestra "game over" en 30,24 al terminar.

Usar la siguiente función para capturar las teclas

```
void input(){  
    if(_tecla=getKey(true)){  
        switch(_tecla){  
            case KEY_UP: _y--; // Ir arriba  
                        break;  
            case KEY_LEFT: _x--; // Ir izquierda  
                        break;  
            case KEY_RIGHT: _x++; // Ir derecha  
                        break;  
            case KEY_DOWN: _y++; // Ir abajo  
                        break;  
            case KEY_ESC: {  
                _gameOver=true;  
                break;  
            }  
        }  
    }  
}
```

Compile, linkee y pruebe el ejercicio tal que muestre lo siguiente en pantalla:



Ejercicio 1

Este ejercicio cuyo proyecto se llama **Ejercicio-1** se codificará utilizando el ejercicio-0 como base y deberá escribir la clase **Juego** según el game loop

El **constructor** es por defecto.

El método **init** inicializa todos los atributos.

El método **input** captura el movimiento de las teclas.

El método **draw** dibuja el * en la pantalla en las coordenadas **_x, _y**.

El método **gameover** devuelve el estado de **_gameover**.

El método **play** es el game loop.

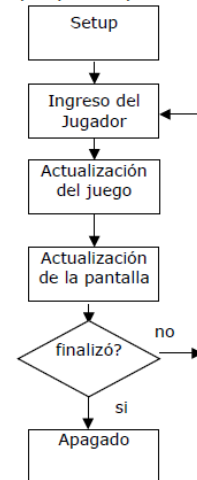
El método **update** que es la lógica del juego, verifica si se toca cualquiera de los bordes entonces pone a **_gameover** en true.

El método **result** imprime el resultado del juego en 30,24.

Juego
- int _x
- int _y
- int _tecla
- bool _gameOver
+ Juego()
+ void init()
+ bool gameOver()
+ void input()
+ void update()
+ void draw()
+ void play()
+ void result()
Representa a un Juego

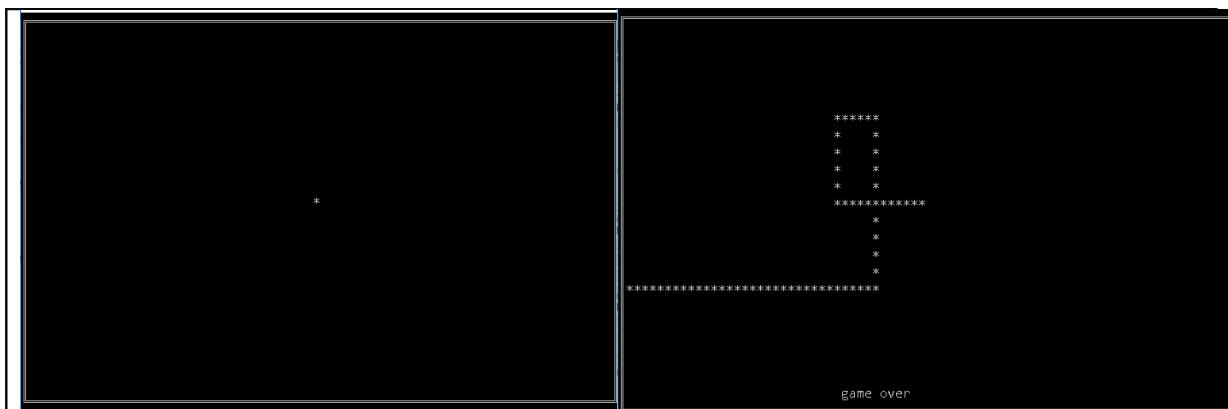
Veamos las partes del ciclo de un juego:

- **Setup.** Implica los ajustes iniciales del juego, como el sonido, música y gráficos. También puede presentarse el trasfondo del juego y sus objetivos.
- **Ingreso del jugador.** Se trata de la captura del ingreso del jugador que puede provenir desde el teclado, mouse, joystick, trackball, o algún otro dispositivo de entrada.
- **Actualización del juego.** Implica la lógica del juego y las reglas que se aplican al mundo del juego, teniendo en cuenta los aportes del jugador. Puede tomar la forma de un sistema interacción física de objetos o implicar cálculos de Inteligencia Artificial del enemigo.
- **Actualización de la pantalla.** En la mayoría de los juegos, este proceso es el más exigente del hardware del equipo, ya que a menudo implica gráficos complejos. Sin embargo, este proceso puede ser tan simple como mostrar una línea de texto.
- **Comprobación de que el juego ha finalizado.** Si el juego no ha terminado (si jugador todavía está vivo y que no ha salido), el control se bifurca de nuevo a la etapa de ingreso del jugador. Si el juego ha terminado, el control se desvía a la fase de cierre.
- **Apagado.** En este punto, el juego termina. El jugador ve a menudo la información final como ser el puntaje, tiempo, etc. El programa libera todos los recursos, si es necesario, y se cierra.



Cree un objeto juego y llame al método **play**.

Compile, linkee y pruebe el ejercicio tal que muestre lo siguiente en pantalla:



Compile, linkee y pruebe el ejercicio tal que muestre lo siguiente en pantalla:

