



# Programación de Tetris para C++

Ana Cristina Molina  
Andrés Arteaga

Física Computacional II  
20202



# Contenido

1. Introducción
2. Librerías
3. Movimientos
4. Código en C++\*
5. Demostración



\*Repositorio de Github: <https://github.com/anacmolina/CompII-20202/tree/main/Proyecto>

# Introducción

- Es un videojuego diseñado originalmente por los científicos rusos Alekséi Pázhitnov y Dmitry Pavlovsky.
- Fue lanzado el 6 de junio de 1984 y en 1989 se lanza la versión portátil para Gameboy con la cual el juego se popularizó con gran rapidez.
- Se ha convertido en uno de los videojuegos más icónicos de todos los tiempos.





## Librería SFML

Simple and Fast Multimedia Library (SFML), provee una interfaz simple para diferentes componentes de un computador y facilitar el desarrollo de videojuegos y aplicaciones multimedia.





## **sf::RenderWindow**

Es la clase principal del módulo *Graphics*, esta define una ventana en el sistema operativo la cual puede ser utilizada como lienzo para dibujar objetos en 2D obtenidos de las demás clases también del módulo *Graphics*.

## **sf::RectangleShape squares**

Clase especializada para la construcción de un rectángulo, las funciones de la clase son: *position, rotation, color, scale,...etc.*



## **sf::VideoMode**

Esta clase se utiliza para configurar una ventana al momento de la creación.

## **sf::Event**

Define un evento del sistema y sus parámetros, guarda toda la información de un evento que acaba de ocurrir, entre los tipos de eventos están: mover ratón, presionar una tecla, cerrar la venta... etc. (*mouse moved, key pressed, window closed, ...*)



## **sf::Vector2<T>**

Clase que permite crear y manipular vectores bidimensionales,  $\langle T \rangle$  representa el tipo *Type* de las entradas del vector (*float*, *int* ...).

## **sf::Color**

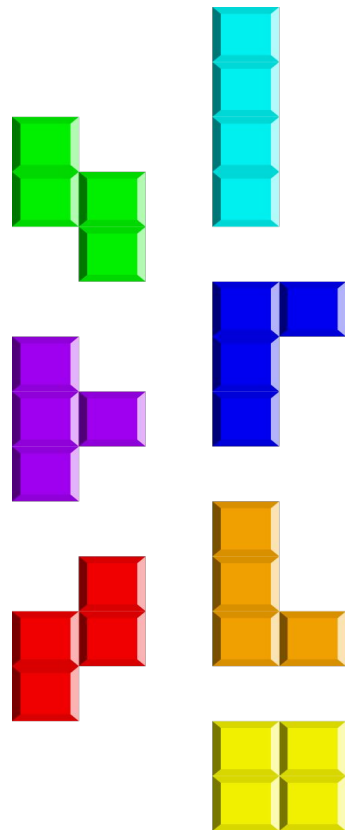
Se utiliza para manipular colores en RGB.

## **sf::Clock**

Esta clase se usa para medir el tiempo transcurrido.

# Reglas del juego

- Para el juego se utilizan 7 piezas cad una conformada por 4 cuadrados conectados por sus lados.
- El jugador no puede evitar la caída de las piezas pero si puede rotarlas y también hacer traslaciones a la izquierda o derecha.
- Cuando la última fila se encuentra llena, esta desaparece y todas las piezas de encima descienden una posición.
- El juego termina cuando las piezas se amontonan hasta llegar al tope.





# Definición de las piezas

Comenzamos definiendo las piezas de la siguiente manera

I =	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7	Z =	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7	S =	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7	T =	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7	L =	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7	J =	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7	O =	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td></tr></table>	0	1	2	3	4	5	6	7
0	1																																																																				
2	3																																																																				
4	5																																																																				
6	7																																																																				
0	1																																																																				
2	3																																																																				
4	5																																																																				
6	7																																																																				
0	1																																																																				
2	3																																																																				
4	5																																																																				
6	7																																																																				
0	1																																																																				
2	3																																																																				
4	5																																																																				
6	7																																																																				
0	1																																																																				
2	3																																																																				
4	5																																																																				
6	7																																																																				
0	1																																																																				
2	3																																																																				
4	5																																																																				
6	7																																																																				
0	1																																																																				
2	3																																																																				
4	5																																																																				
6	7																																																																				
	(1,3,5,7)		(2,4,5,7)		(3,5,4,6)		(3,5,4,7)		(2,3,5,7)		(3,5,7,6)		(2,3,4,5)																																																								




## Posición en x, y

Tenemos que cada pieza está conformada por cuatro bloques cada uno identificado con un número de 0 a 7, para determinar la posición de la pieza accedemos a las posiciones en (x, y) de cada bloque.

Para obtener la posición en x calculamos el residuo del número del bloque entre 2 y para y dividimos el número del bloque entre 2.

$$x = \text{bloque} \% 2 ; \quad y = \text{bloque} \div 2$$



Por ejemplo si tomamos la pieza uno, las coordenadas para las posiciones de cada uno de sus bloques son

0	1
2	3
4	5
6	7

Bloque 1:  $x = 1 \% 2 = 1$  ;  $y = 1 / 2 = 0$  posición (1,0)

Bloque 3:  $x = 3 \% 2 = 1$  ;  $y = 3 / 2 = 1$  posición (1,1)

Bloque 5:  $x = 5 \% 2 = 1$  ;  $y = 5 / 2 = 2$  posición (1,2)

Bloque 7:  $x = 7 \% 2 = 1$  ;  $y = 7 / 2 = 3$  posición (1,3)




## Movimientos: Traslación

Una vez obtenidas las posiciones de los bloques que forman cada pieza para efectuar una traslación a la derecha o la izquierda simplemente sumamos o restamos uno a cada posición de  $x$  para cada bloque y dejamos la posición en  $y$  igual.



## Movimientos: Rotación

Para realizar una rotación a cualquier pieza primero escogemos un bloque de los cuatro el cual usaremos como centro de rotación, tomando como centro el segundo bloque entonces podemos definir un vector relativo entre las posiciones del centro de rotación y la posición de otro bloque adyacente.




Por ejemplo volviendo a tomar la pieza 1 y tomando el segundo bloque como centro de rotación, bloque 3, podemos definir el vector relativo,  $V_r$ , entre las posiciones del bloque 3 y el bloque 1.

0	1
2	3
4	5
6	7

$$v_r = v_1 - v_{cr}$$


$$v_r = (x_1 - x_3, y_1 - y_3)$$



Ahora tomamos este vector  $V_r$  y lo rotamos usando la matriz de rotación en sentido antihorario evaluada para el ángulo de 90 grados obteniendo así un vector rotado  $V_t$ , esto es:

$$v_t = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} v_r \qquad \begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} -y_r \\ x_r \end{pmatrix}$$

Ahora tomamos este vector  $V_r$  y lo rotamos usando la matriz de rotación en sentido antihorario evaluada para el ángulo de 90 grados obteniendo así un vector rotado  $V_t$ , esto es:



El vector  $V_t$ , es el vector  $V_r$  rotado por lo que si queremos el vector con las coordenadas del bloque  $i$  después de la rotación, debemos sumar  $V_{cr}$ .

$$v'_1 = v_t + v_{cr}$$





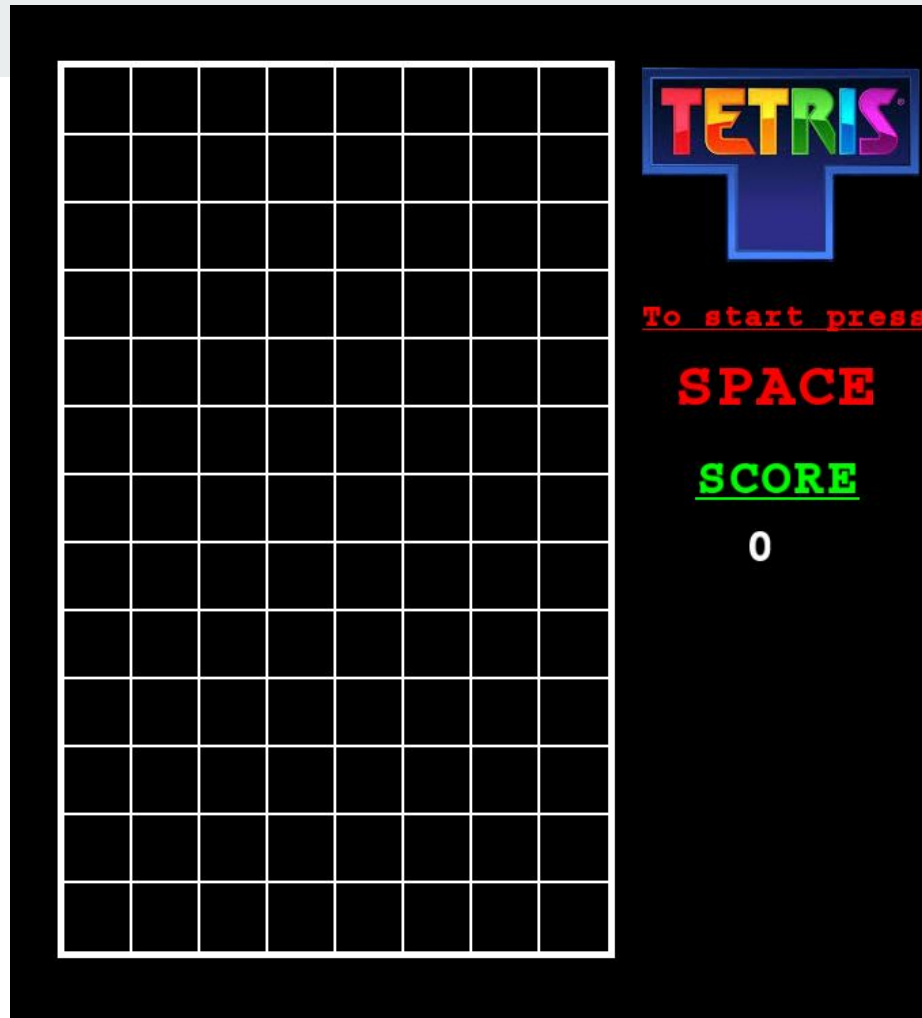
## Código: Funciones usadas

- background()
- random\_color()
- squares()
- top()
- check()
- no\_play()
- remove\_line()
- rotation()
- traslation()
- gen\_figure()
- draw\_figure()
- fixed\_figure()

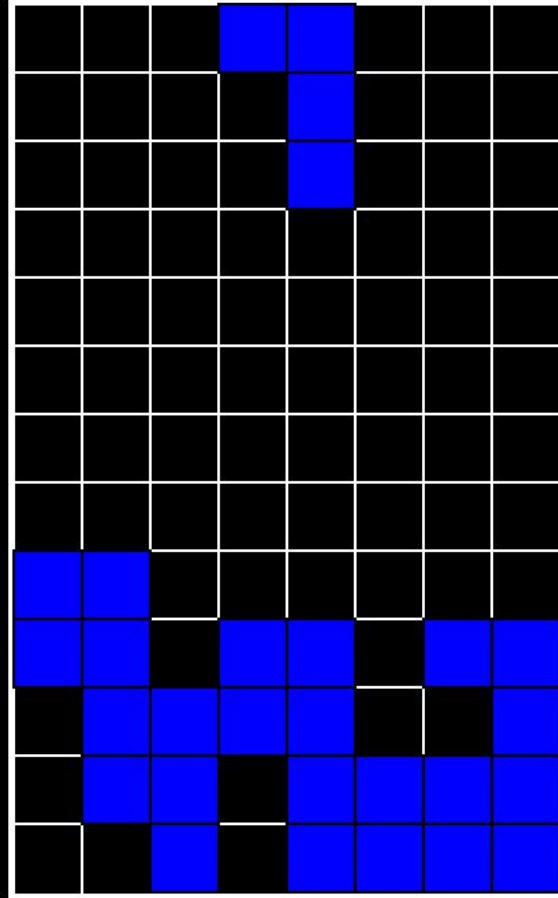
## Código: Compilación

g++ tetris.cpp -lsfml-graphics -lsfml-window -lsfml-system

# Fondo



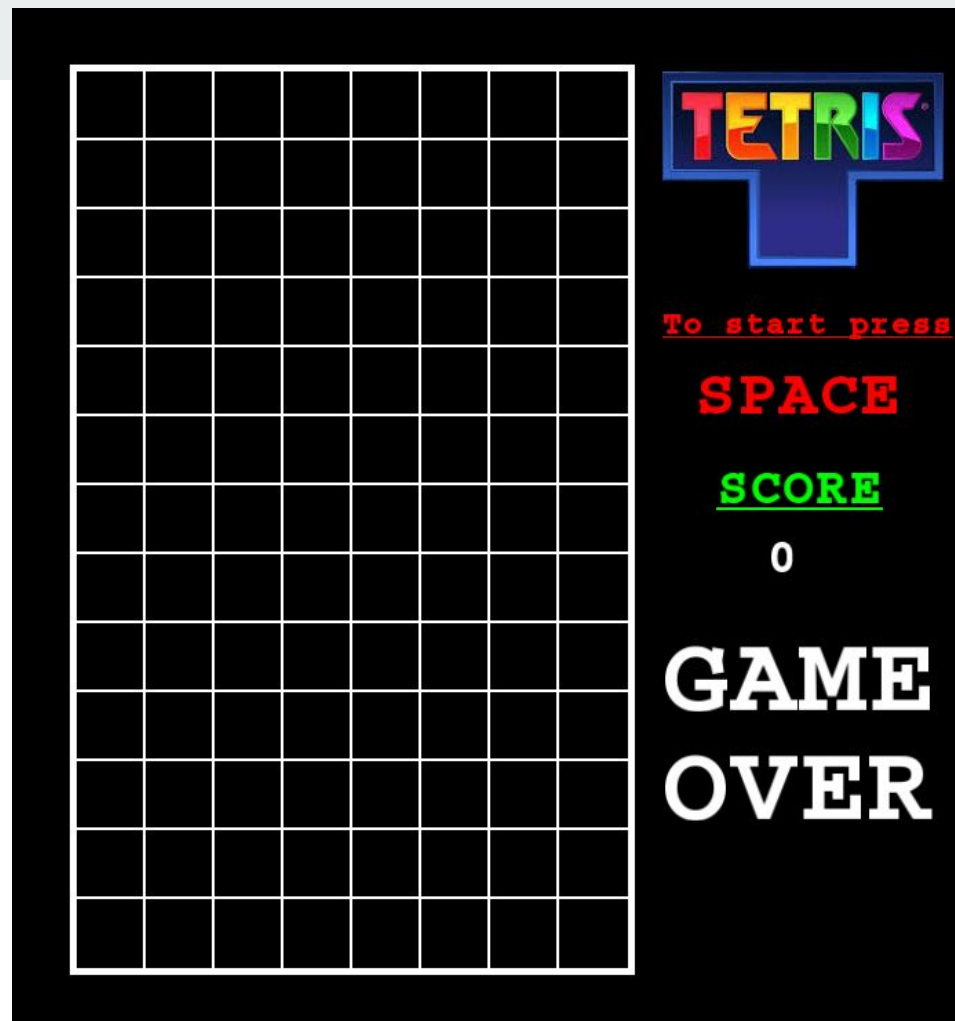
# Juego



SCORE

0

# Final





## Conclusiones

- A partir de la creada se empezar a construir distintos niveles de una forma más sencilla.
- La librería SFML es útil pero requiere tiempo tener cierto conocimiento de POO debido a los objetos que crea, teniendo eso en cuenta puede ser más sencilla que otras como GLFW, GLEW.



## Referencias

- <https://en.wikipedia.org/wiki/Tetris>
- [https://www.youtube.com/watch?v=zH\\_omFPqMO4](https://www.youtube.com/watch?v=zH_omFPqMO4)