



Carlos Alberto Bello R.

14 de diciembre de 2025

Contenido

- 1 Introducción
- 2 Solución: Matrices
- 3 Ventajas
- 4 Matemáticas del Juego
- 5 Implementación

El problema de Tetris

¿Cómo representar piezas en código?

Cada pieza de Tetris tiene:

- Forma diferente
- Rotaciones diferentes (1-4)
- Necesita ser dibujada
- Debe detectar colisiones

Ejemplo: Pieza T

4 rotaciones diferentes

Representación con matrices

Idea principal

Usar matrices 2D (listas de listas en Python)

- 0 = espacio vacío
- 1, 2, 3... = tipo de pieza/color
- Cada rotación es una matriz diferente

Pieza T - Rotación 0

```
[[0,3,0],  
 [3,3,3],  
 [0,0,0]]
```

Código en Python

```
PIECE_SHAPES = [  
    [], # Índice 0: vacío  
  
    # Pieza I (2 rotaciones)  
    [  
        [[0,0,0,0], [1,1,1,1],  
         [0,0,0,0], [0,0,0,0]],  
        [[0,1,0,0], [0,1,0,0],  
         [0,1,0,0], [0,1,0,0]]  
    ],  
  
    # Pieza O (1 rotación)  
    [[[2,2], [2,2]]],  
  
    # Pieza T (4 rotaciones)  
    [  
        [[0,3,0], [3,3,3], [0,0,0]],  
        [[0,3,0], [0,3,3], [0,3,0]],  
        [[0,0,0], [3,3,3], [0,3,0]],  
        [[0,3,0], [3,3,0], [0,3,0]]  
    ],  
]
```

Visualización

Pieza I - Horizontal

0	0	0	0	→				
1	1	1	1		X	X	X	X
0	0	0	0					
0	0	0	0					

Pieza O - Cuadrado

2	2	→	X	X
2	2		X	X

Rotaciones de la pieza T

Rotación 0

0	3	0
3	3	3
0	0	0



	X	
X	X	X

Rotación 1

0	3	0
0	3	3
0	3	0



	X	
	X	X
	X	

Modelo Matemático del Tablero

Representación Matricial

El tablero se modela como una matriz de 20×10 :

$$T = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}_{20 \times 10}$$

donde $T_{i,j} = 0$ representa espacio vacío.

¿Por qué 10×20 ?

- **Ancho (10):** Permite rotación de piezas (la más larga mide 4).

Sistema de Puntuación

Tabla de Puntuación Clásica

Líneas	Puntos Base	Ejemplo (Nivel 2)
1 línea	100	200
2 líneas	300	600
3 líneas	500	1,000
4 líneas	800	1,600

Fórmulas de Cálculo

$$\text{Puntos} = \text{Puntos Base} \times \text{Nivel}$$

$$\text{Nivel} = \left\lfloor \frac{\text{Líneas Totales}}{10} \right\rfloor + 1$$

Estructuras de Datos Clave

Variables Globales

- `board_state`: Matriz 20×10 del tablero
- `current_piece`: Diccionario con pieza actual
- `score`: Puntuación actual

Representación de Pieza

`current_piece =`

`'shape' : 3, Tipodepieza(I, J, L, etc.)'rotation' : 0, Estadoderotacin(0`

Algoritmo de Colisiones

Lógica de Detección

Verificamos si la nueva posición está fuera de los límites o superpuesta con bloques existentes.

Pseudocódigo Python

```
def check_collision(shape, x, y) : for block in shape : bx =  
x + block.offset_x, by = y + block.offset_y  
if bx < 0 or bx >= ANCHO: return True if by >= ALTO:  
return True if by >= 0 and board[by][bx] != 0: return True  
return False
```