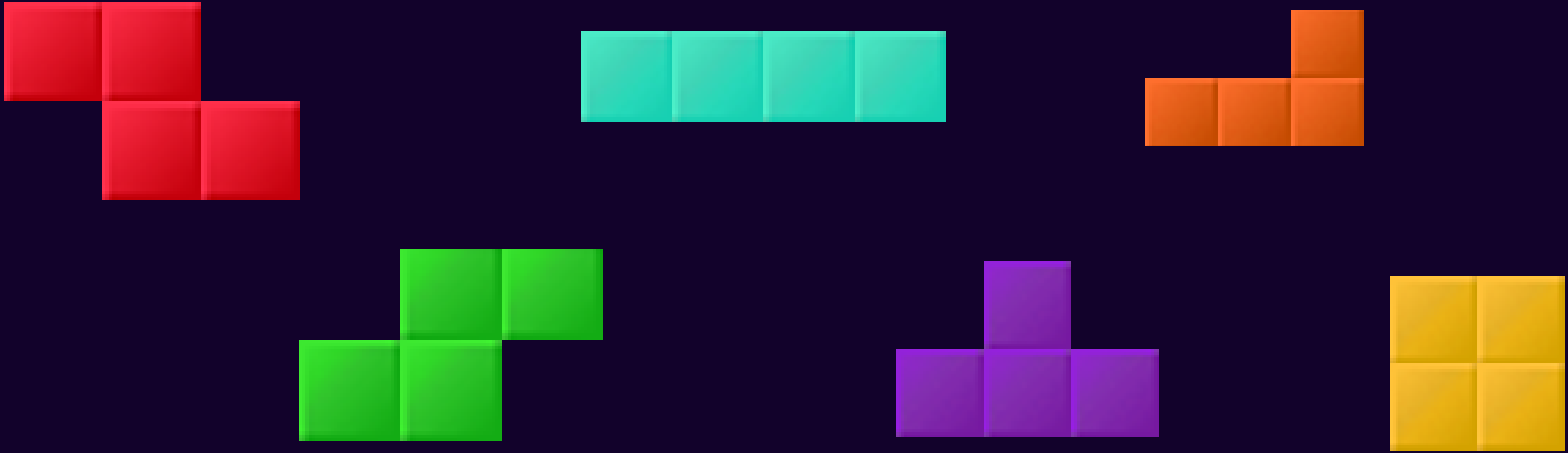
TETRIS TRAINER

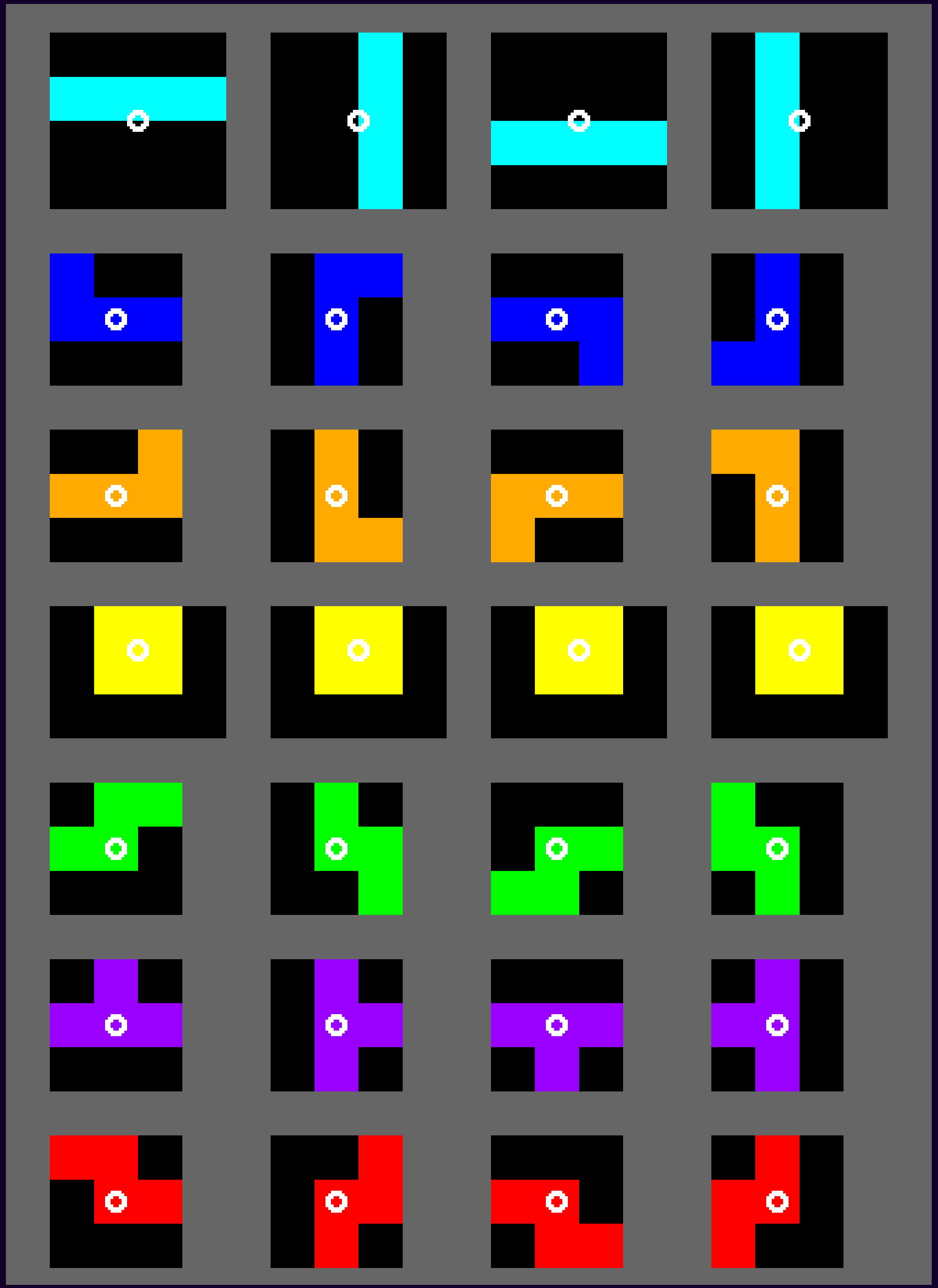
GRUPO QUALITY

PIEZAS

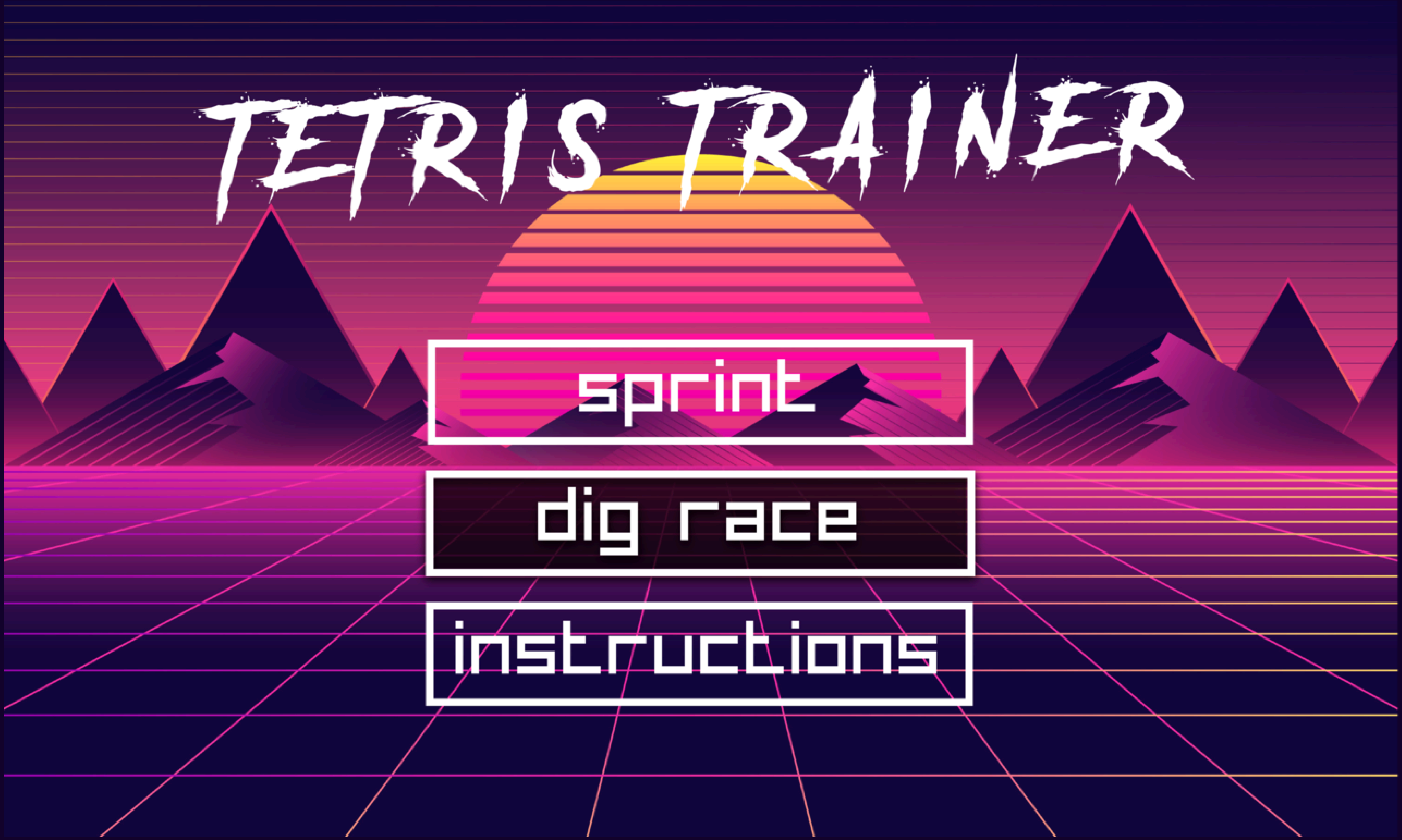


ROTACIONES

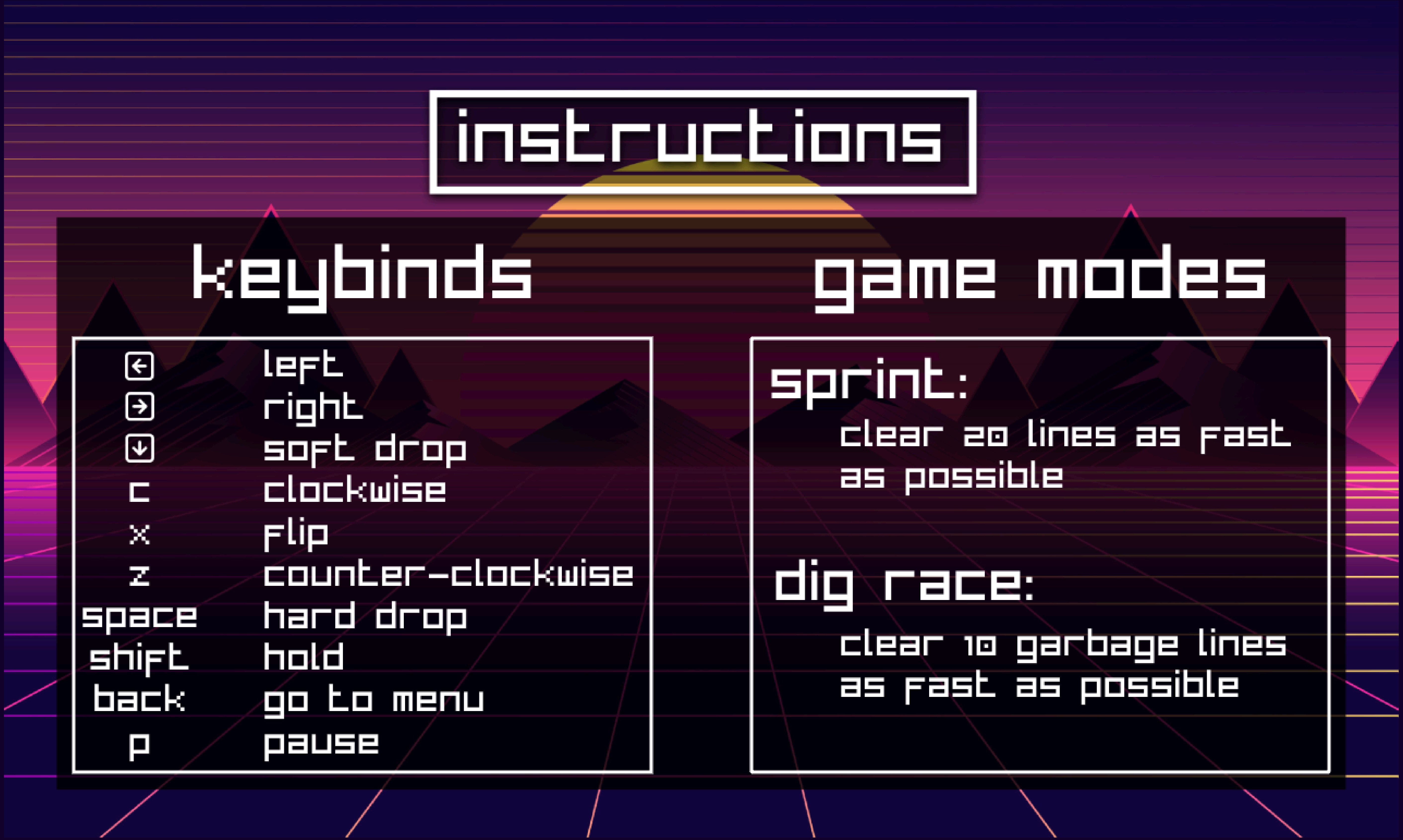
ROTACIONES



MENU



INSTRUCTIONS



instructions

keybinds

←	left
→	right
↓	soft drop
c	clockwise
x	flip
z	counter-clockwise
space	hard drop
shift	hold
back	go to menu
p	pause

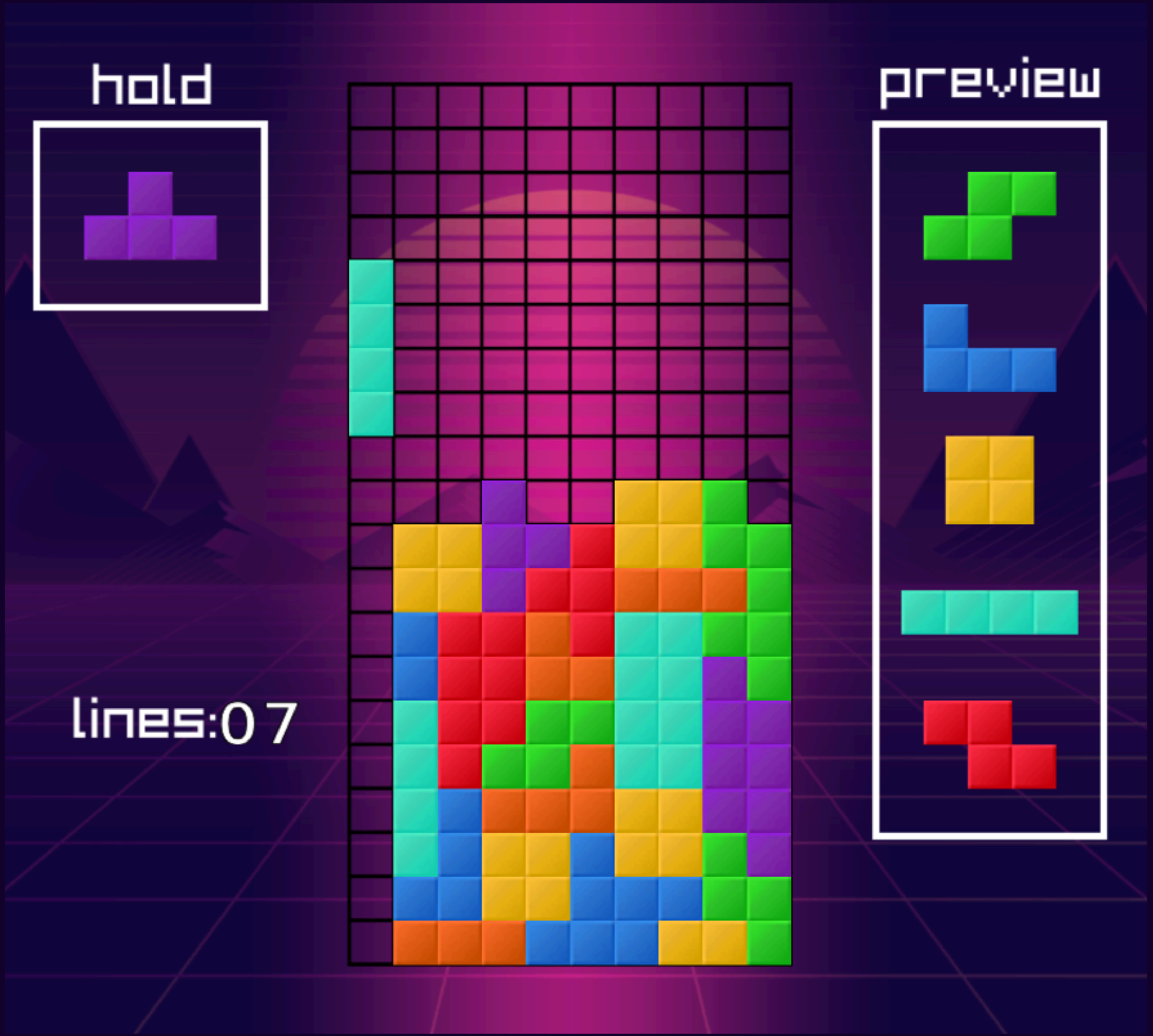
game modes

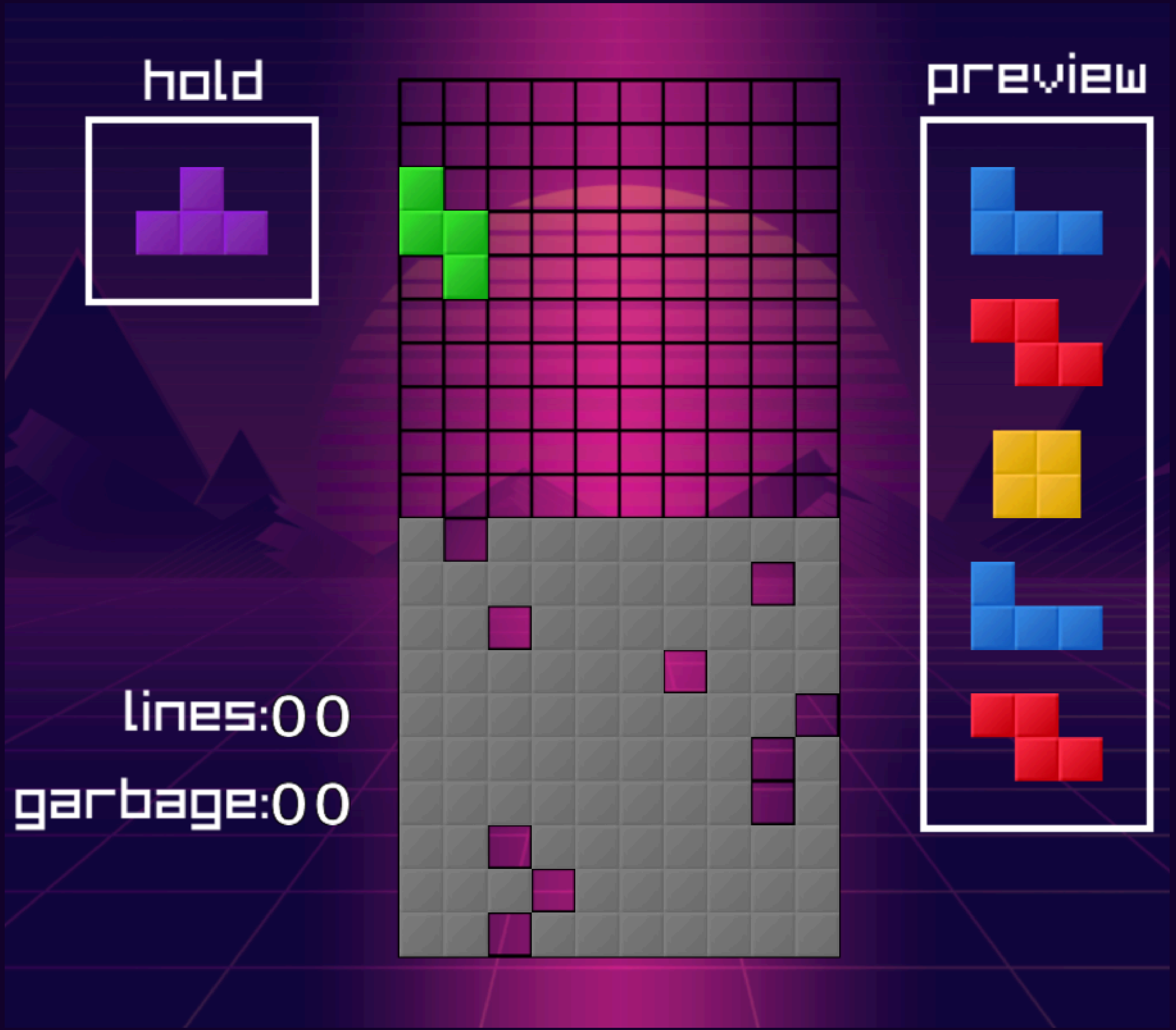
sprint:
clear 20 lines as fast as possible

dig race:
clear 10 garbage lines as fast as possible

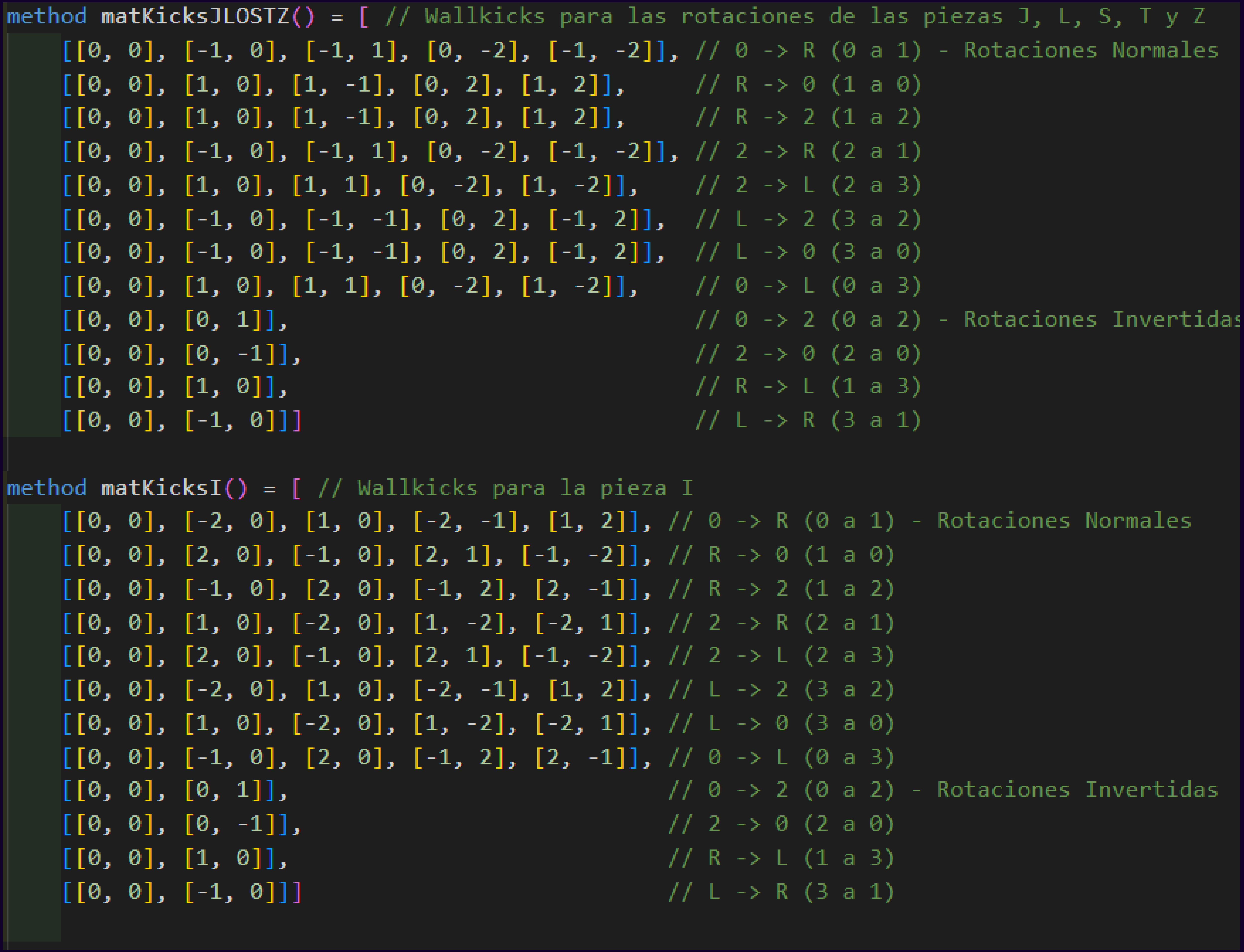
SPRINT

DIG RACE



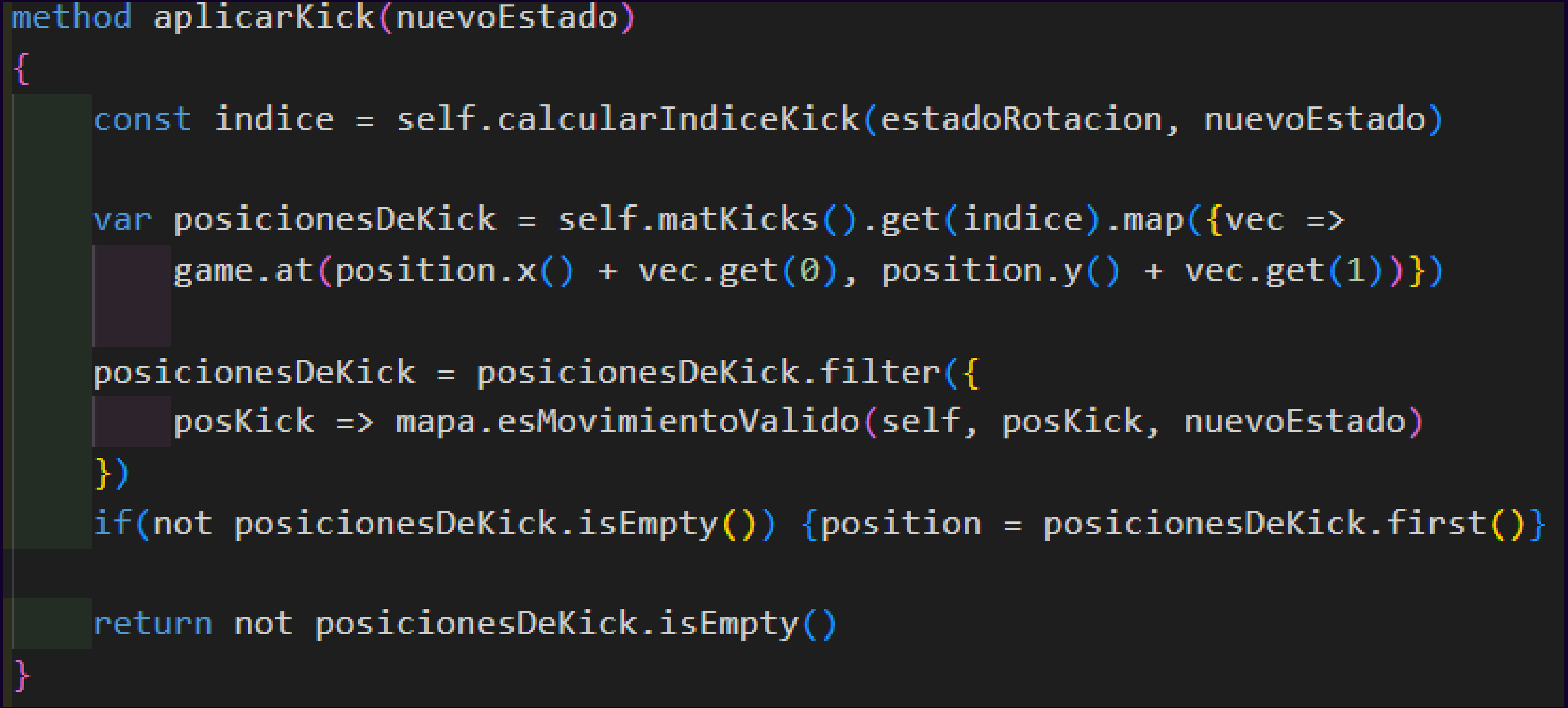


WALL KICK



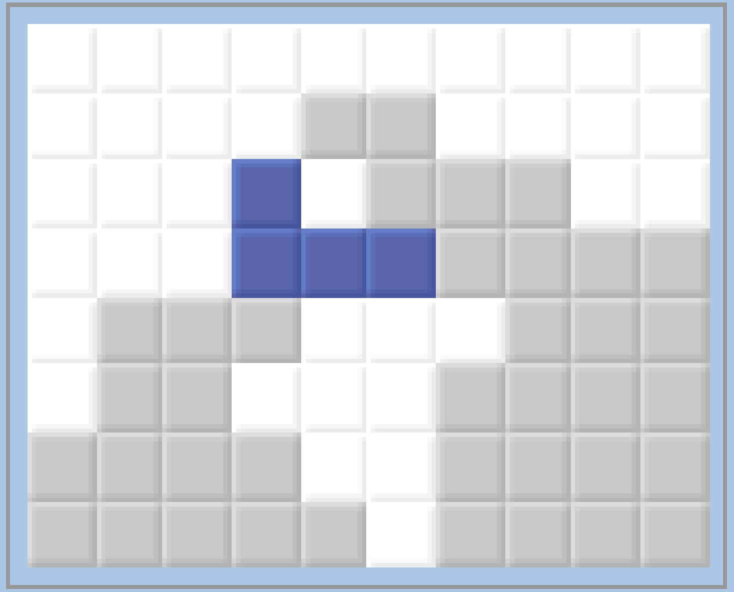
```
method matKicksJLOSTZ() = [ // Wallkicks para las rotaciones de las piezas J, L, S, T y Z
        [[0, 0], [-1, 0], [-1, 1], [0, -2], [-1, -2]], // 0 -> R (0 a 1) - Rotaciones Normales
        [[0, 0], [1, 0], [1, -1], [0, 2], [1, 2]],     // R -> 0 (1 a 0)
        [[0, 0], [1, 0], [1, -1], [0, 2], [1, 2]],     // R -> 2 (1 a 2)
        [[0, 0], [-1, 0], [-1, 1], [0, -2], [-1, -2]], // 2 -> R (2 a 1)
        [[0, 0], [1, 0], [1, 1], [0, -2], [1, -2]],    // 2 -> L (2 a 3)
        [[0, 0], [-1, 0], [-1, -1], [0, 2], [-1, 2]],  // L -> 2 (3 a 2)
        [[0, 0], [-1, 0], [-1, -1], [0, 2], [-1, 2]],  // L -> 0 (3 a 0)
        [[0, 0], [1, 0], [1, 1], [0, -2], [1, -2]],    // 0 -> L (0 a 3)
        [[0, 0], [0, 1]],                              // 0 -> 2 (0 a 2) - Rotaciones Invertidas
        [[0, 0], [0, -1]],                             // 2 -> 0 (2 a 0)
        [[0, 0], [1, 0]],                              // R -> L (1 a 3)
        [[0, 0], [-1, 0]]]                             // L -> R (3 a 1)

method matKicksI() = [ // Wallkicks para la pieza I
        [[0, 0], [-2, 0], [1, 0], [-2, -1], [1, 2]], // 0 -> R (0 a 1) - Rotaciones Normales
        [[0, 0], [2, 0], [-1, 0], [2, 1], [-1, -2]], // R -> 0 (1 a 0)
        [[0, 0], [-1, 0], [2, 0], [-1, 2], [2, -1]], // R -> 2 (1 a 2)
        [[0, 0], [1, 0], [-2, 0], [1, -2], [-2, 1]], // 2 -> R (2 a 1)
        [[0, 0], [2, 0], [-1, 0], [2, 1], [-1, -2]], // 2 -> L (2 a 3)
        [[0, 0], [-2, 0], [1, 0], [-2, -1], [1, 2]], // L -> 2 (3 a 2)
        [[0, 0], [1, 0], [-2, 0], [1, -2], [-2, 1]], // L -> 0 (3 a 0)
        [[0, 0], [-1, 0], [2, 0], [-1, 2], [2, -1]], // 0 -> L (0 a 3)
        [[0, 0], [0, 1]],                            // 0 -> 2 (0 a 2) - Rotaciones Invertidas
        [[0, 0], [0, -1]],                           // 2 -> 0 (2 a 0)
        [[0, 0], [1, 0]],                            // R -> L (1 a 3)
        [[0, 0], [-1, 0]]]                           // L -> R (3 a 1)
```

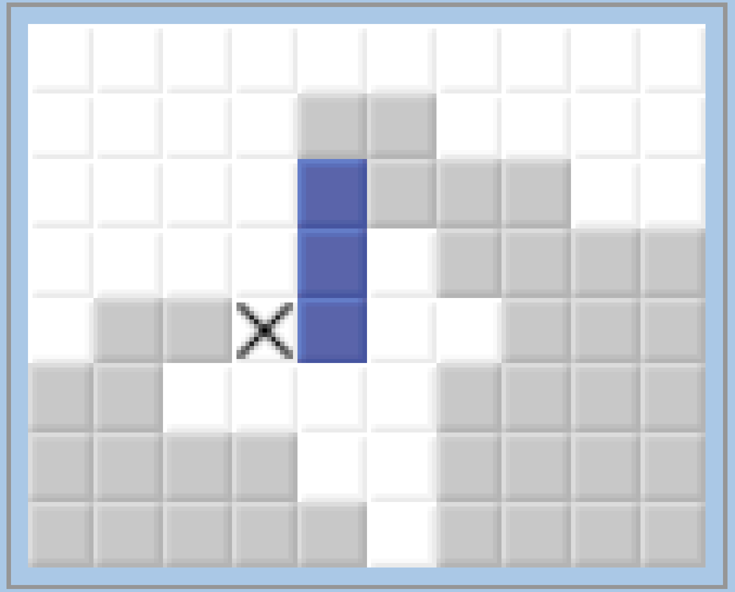
WALL KICK

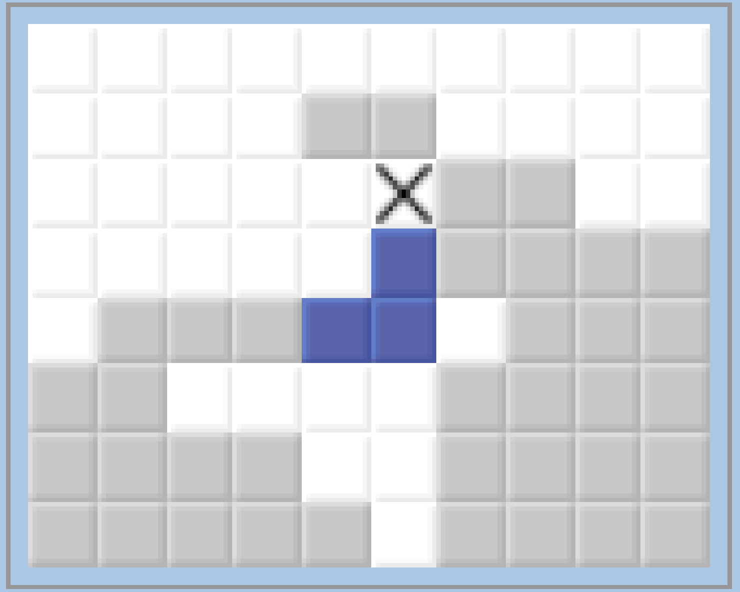


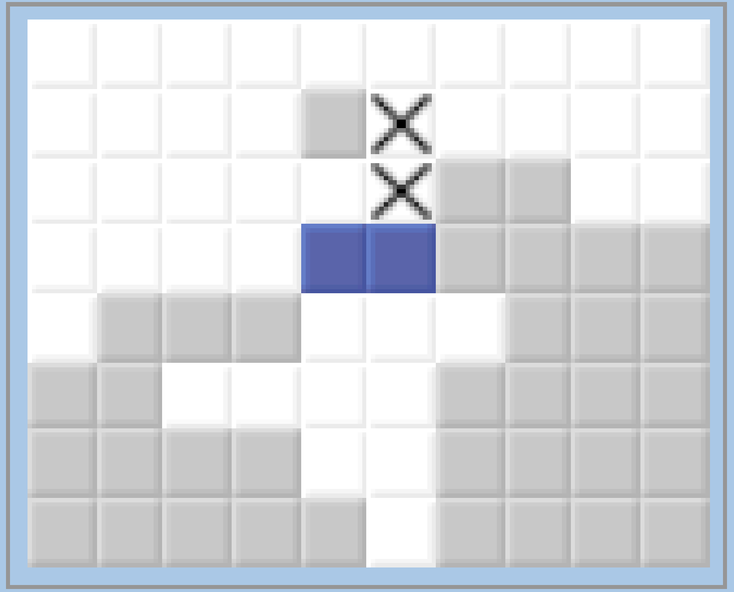
```
method aplicarKick(nuevoEstado)
{
    const indice = self.calcularIndiceKick(estadoRotacion, nuevoEstado)

    var posicionesDeKick = self.matKicks().get(indice).map({vec =>
        game.at(position.x() + vec.get(0), position.y() + vec.get(1))})

    posicionesDeKick = posicionesDeKick.filter({
        posKick => mapa.esMovimientoValido(self, posKick, nuevoEstado)
    })
    if(not posicionesDeKick.isEmpty()) {position = posicionesDeKick.first()}

    return not posicionesDeKick.isEmpty()
}
```

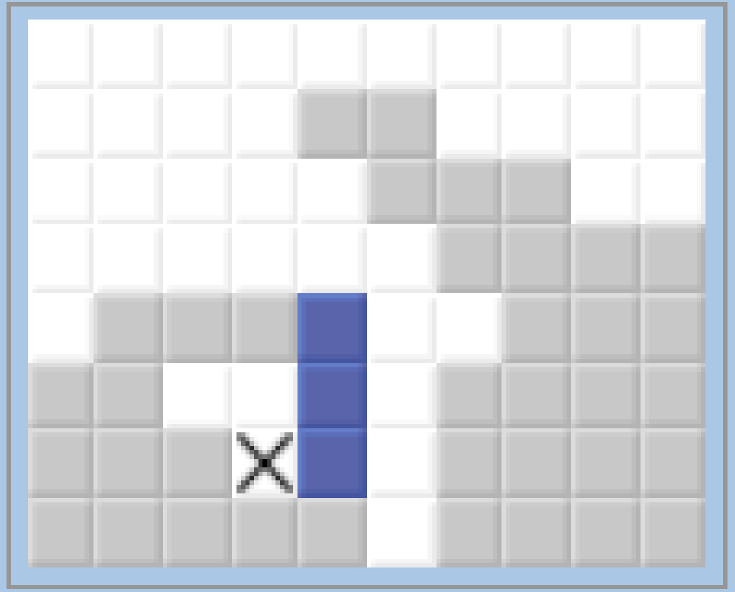
WALL KICK

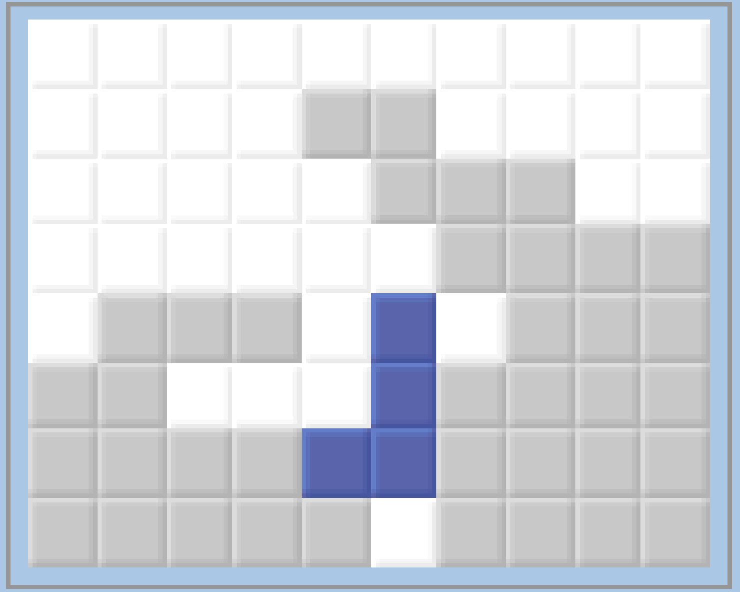

1. Initial position.
Attempt to rotate 0->L.


2. Test 1, (0, 0) fails.
(Basic rotation fails.)

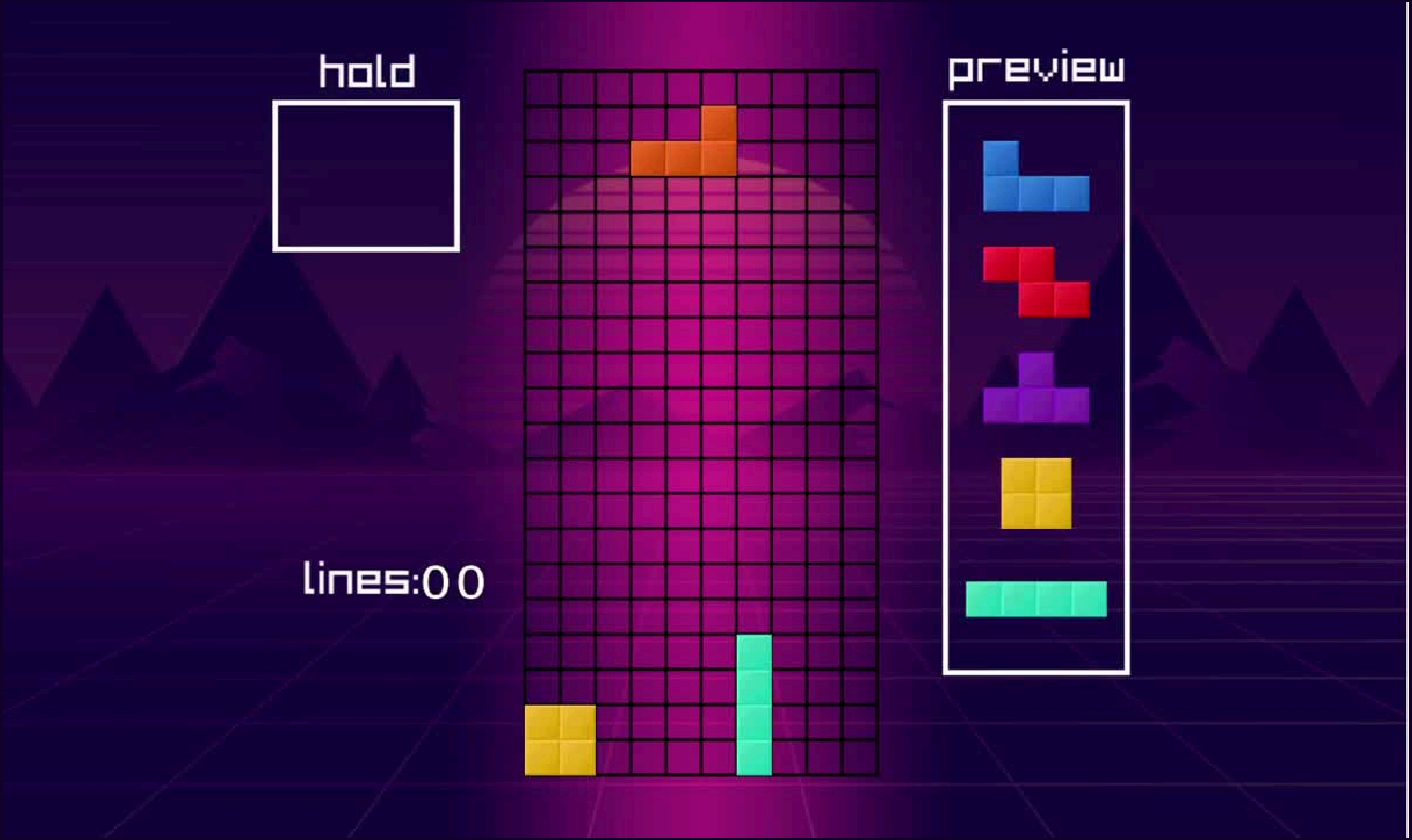

3. Test 2, (+1, 0) fails.


4. Test 3, (+1,+1) fails.

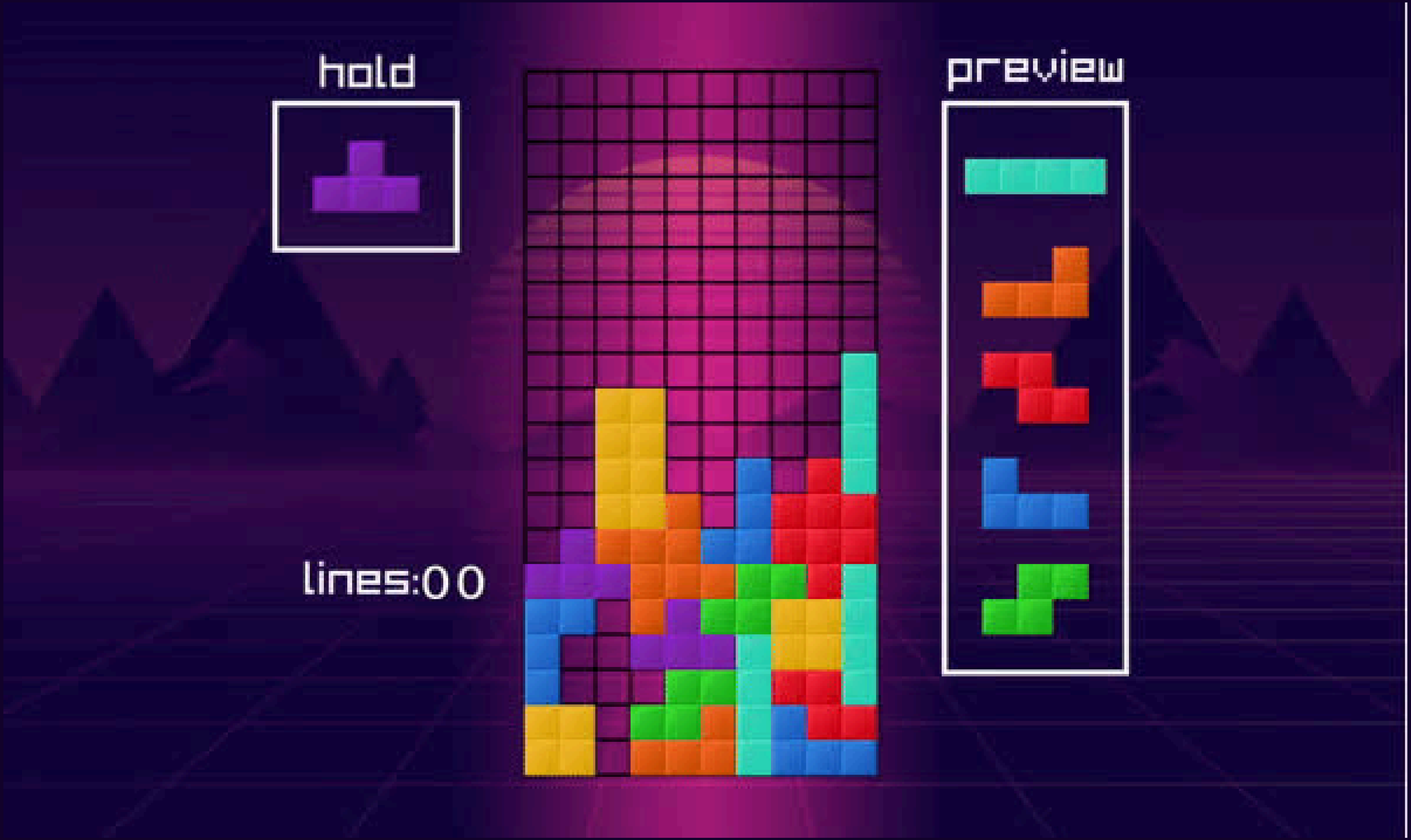

5. Test 4, (0,-2) fails.


6. Final position.

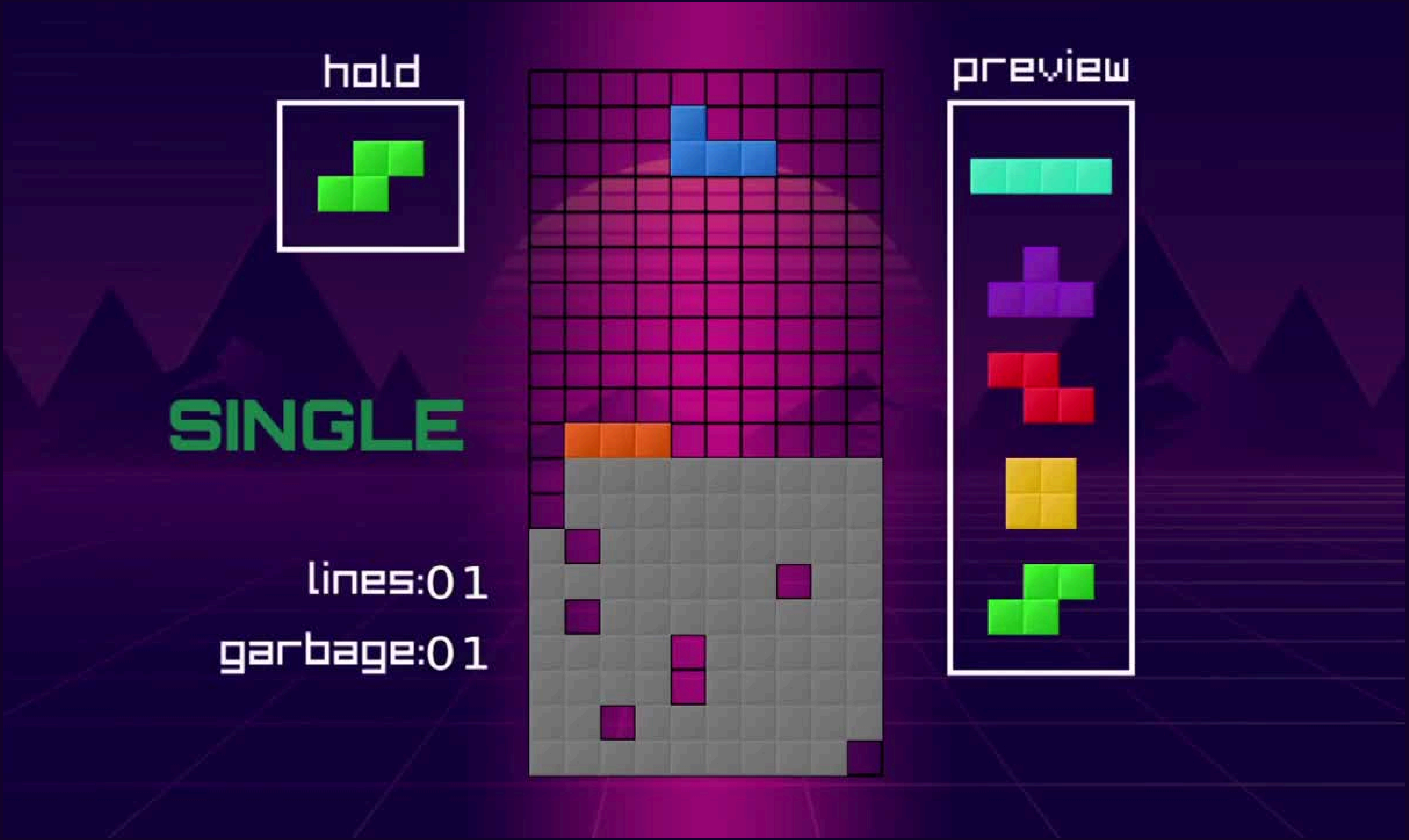
DT-CANNON



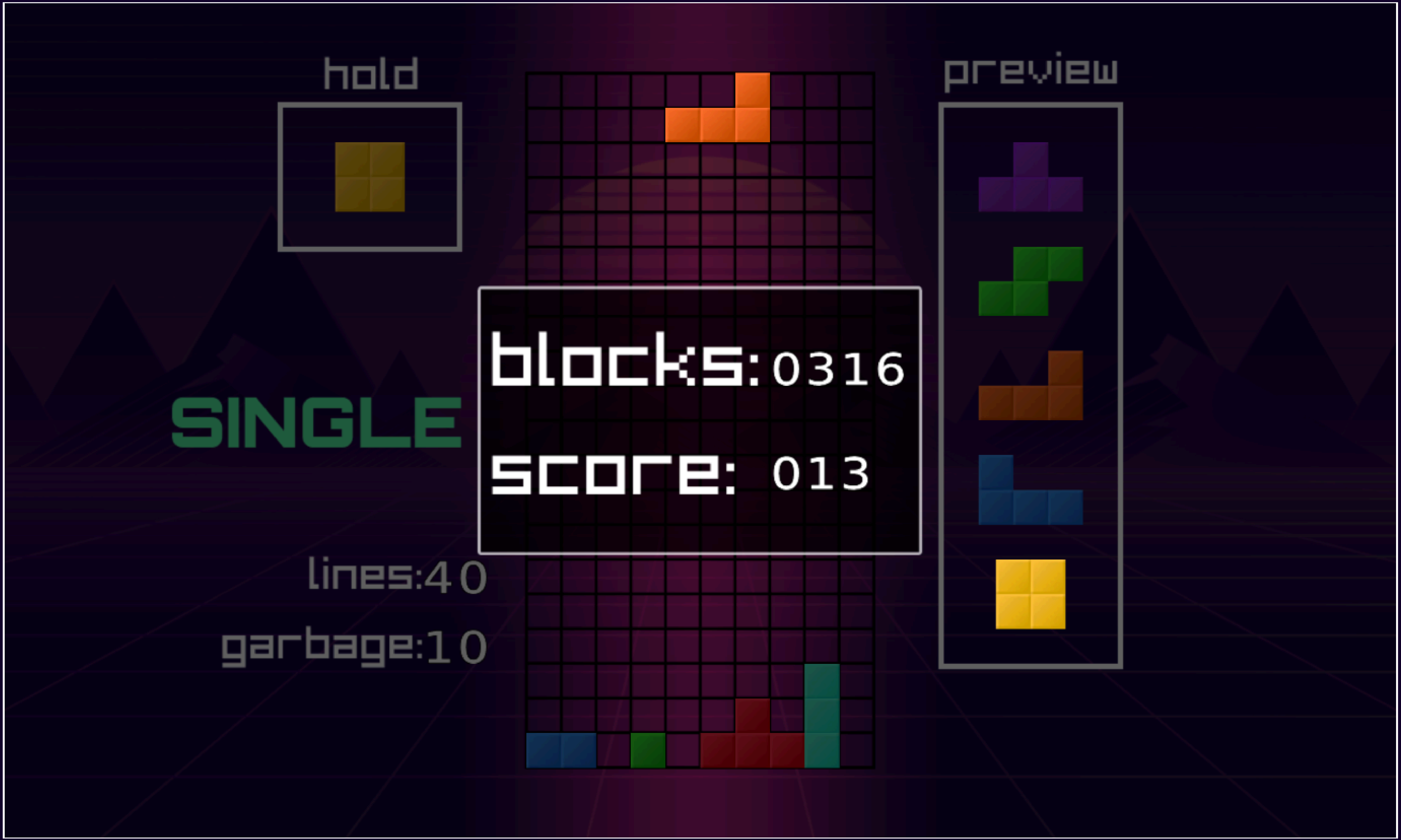
T-SPINS



DIG RACE



END SCREEN



blocks:0316

score: 013