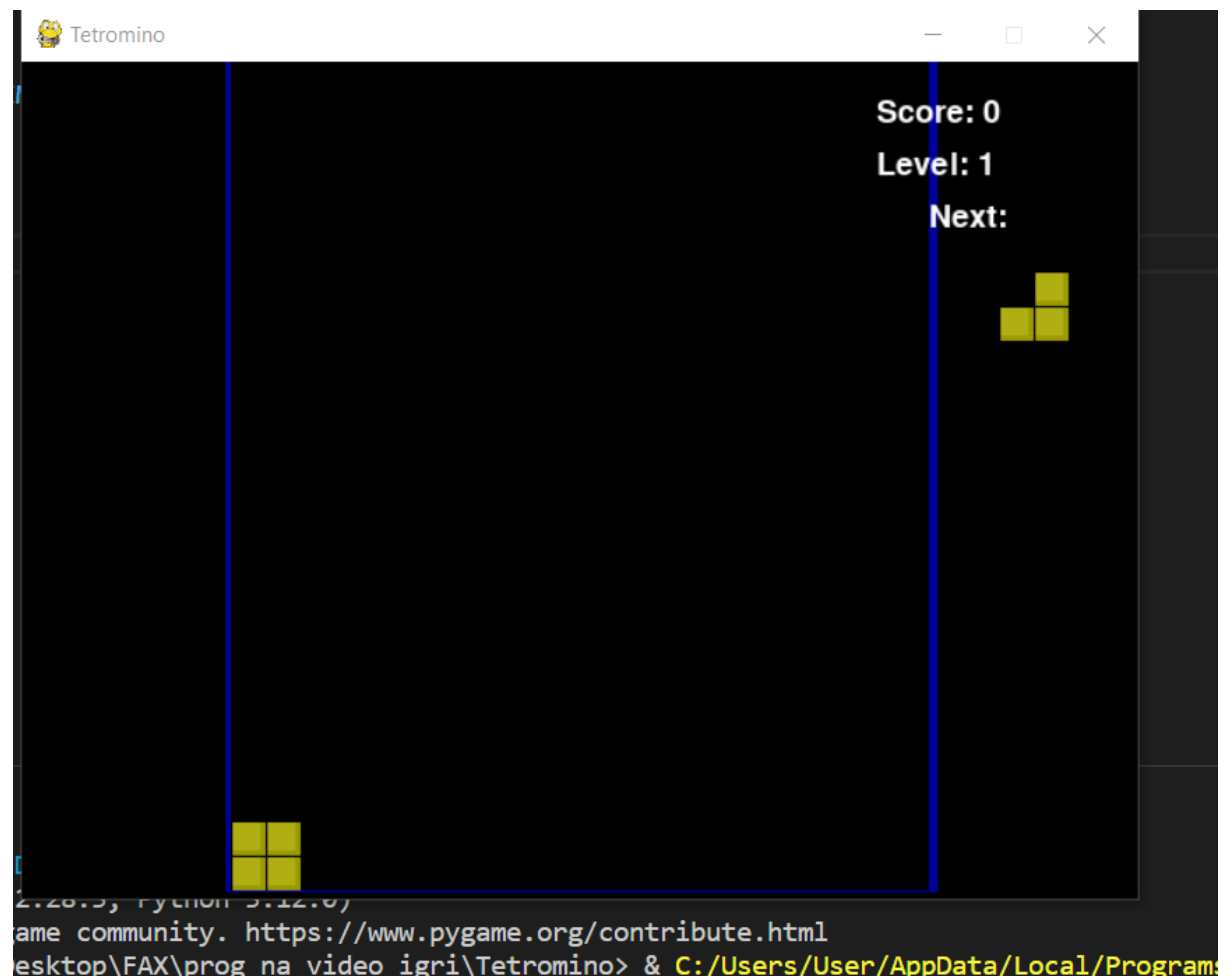


- Правиме промена на големината на board во 20x40 кутии.

```
BOARDWIDTH = 20
BOARDHEIGHT = 40
```

- Ја намалуваме брзината на пагање на елементите оставаме повеќе временски простор пред да се појави наредната фигура



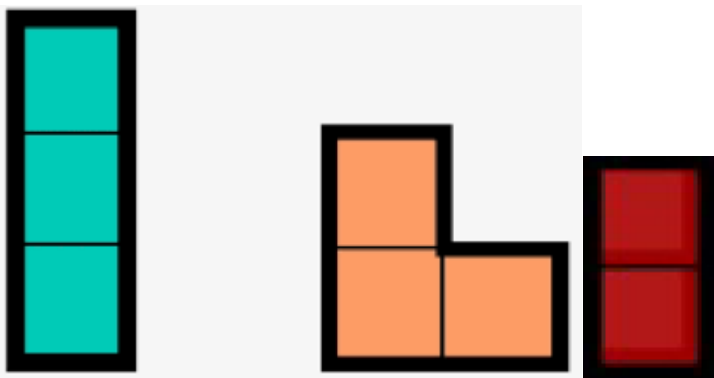
- Дефинираме 3 нови фигури x1,x2,x3 во dictionary. X1 фигура која се состои од 2 полиња, X2 фигура од 3 полиња и X3 од 3 полиња.

```
X1_SHAPE_TEMPLATE = [['.....', #2box shape
    '..0..',
    '..0..',
    '.....',
    '.....'],
    ['.....',
    '.....',
    '..00.',
    '.....',
    '.....'],
    ['.....',
    '.....',
    '..0..',
    '..0..',
    '.....'],
    ['.....',
    '.....',
    '.....',
    '.....',
    '.....']]

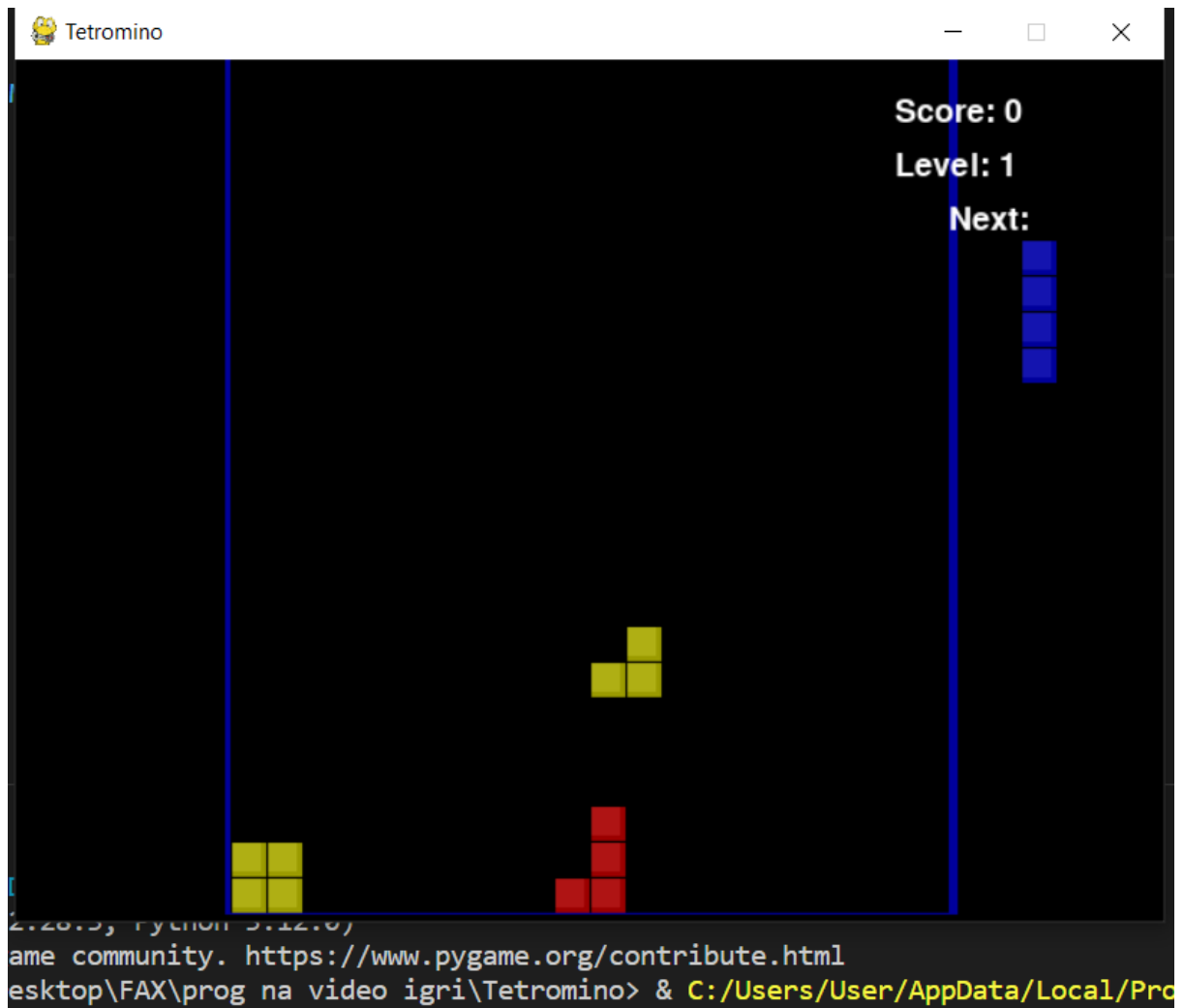
X2_SHAPE_TEMPLATE = [['.....', #3box shape
    '..0..',
    '..00.',
    '.....',
    '.....'],
    ['.....',
    '..0..',
    '..00.',
    '.....',
    '.....'],
    ['.....',
    '..00.',
    '..0..',
    '.....',
    '.....'],
    ['.....',
    '.....',
    '.....',
    '.....',
    '.....']]

X3_SHAPE_TEMPLATE = [['.....', #3box shape
    '..0..',
    '..0..',
    '..0..',
    '.....'],
    ['.....',
    '.....',
    '..000',
    '.....',
    '.....'],
    ['.....',
    '.....',
    '.....',
    '.....',
    '.....'],
    ['.....',
    '.....',
    '..0..',
    '..0..',
    '..0..'],
    ['.....',
    '.....',
    '000..',
    '.....',
    '.....']]
```

- Ова е формата на новите елементи



- Една од фигурите во игра:



```

# let the piece fall if it is time to fall
if time.time() - lastFallTime > fallFreq:
    # see if the piece has landed
    if not isValidPosition(board, fallingPiece, adjY=1):
        # falling piece has landed, set it on the board
        addToBoard(board, fallingPiece)

        score1 = score
        score += removeCompleteLines(board)
        if score1 != score:
            flag = 1
        else:
            flag = 0
        level, fallFreq = calculateLevelAndFallFreq(score)

    if flag == 1:
        if (score1 - score) % 2 == 0:
            #print(score)
            #print(score1)
            fallFreq = fallFreq + (level * 0.02)
        if (score1 - score) % 2 != 0:
            fallFreq = fallFreq - (level * 0.02)
            #print(score)
            #print(score1)

```

- Декларираме нова променлива score1 во која ќе се чува стариот score, flag кој се користи како резултат од проверка кога score ќе се промени, односно ред ќе се пополни или избрише.
- Кога ќе се случи ова се врши нова проверка односно разликата помеѓу стариот score1 и новиот score и дали се избришале парен или непарен број на редови и во однос на тоа динамички се променува брзината на паѓање на фигурите со фактор (level * 0.02)