

## Лабораториска вежба 5: Tetromino

Студент: Викторија Мијалчева 181070

### 1. Зголемете ја димензијата на рамката на 20x40 полиња.

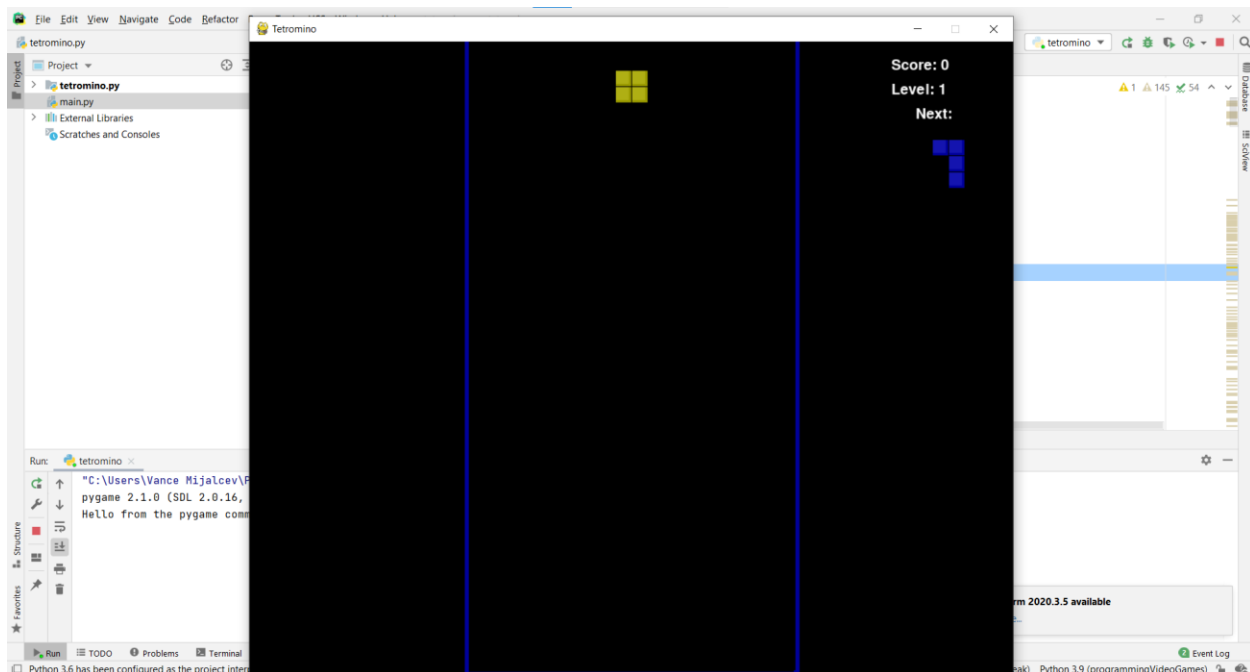
За да го задоволиме ова потребно е да ги промениме следниве глобални променливи:

```
BOARDWIDTH = 20 #CHANGE  
BOARDHEIGHT = 40 #CHANGE
```

За да овозможиме поубав приказ на корисникот со менување на овие променливи потребно е менување и на следниве две променливи:

```
WINDOWHEIGHT = 480 + 300 #CHANGE  
WINDOWWIDTH = 640 + 300 #CHANGE
```

Излезниот екран по направените промени е следниот:



## 2. Дефинирајте ги и додадете ги во структурата dictionary сите форми што можат да се креираат со 3 и 2 полиња

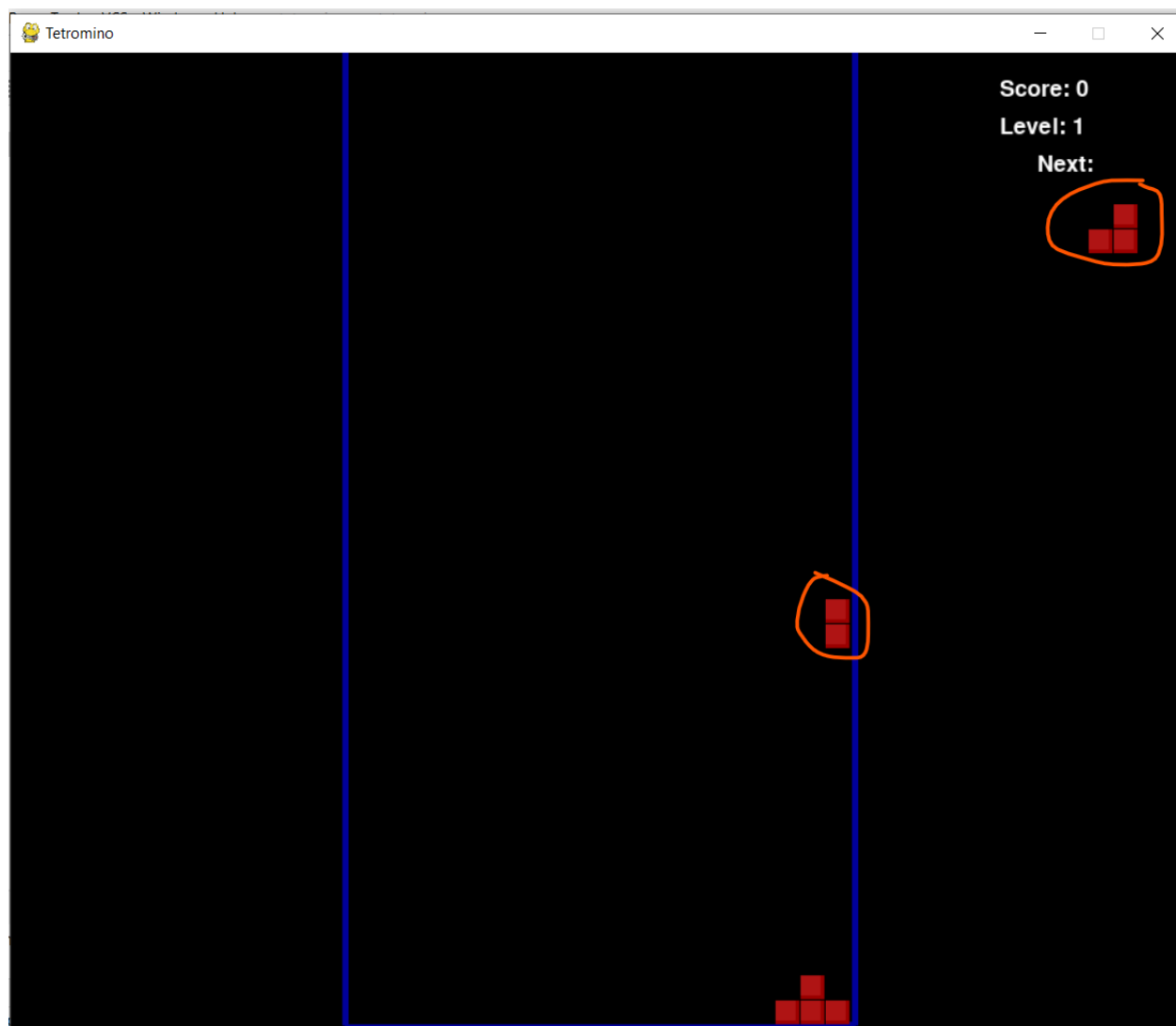
Такви структури во овој случај постојат 3. Истите се креираат и се додаваат во речник со име *three\_or\_two\_fields\_shapes*. За секоја форма се дефинираат исто и позиции во кои истата може да се најде. Речникот е следен:

```
three_or_two_fields_shapes = {
    'three_fields_1': [
        ['..O..',
         '..O..',
         '..O..',
         '....',
         '....'],
        ['....',
         '....',
         '....',
         'OOO..',
         '....',
         '....']],
    'three_fields_2': [
        ['.....',
         '....00',
         '....0.',
         '.....',
         '.....'],
        ['.....',
         '....00',
         '....0.',
         '.....',
         '.....'],
        ['.....',
         '....0.',
         '....00',
         '.....',
         '.....'],
        ['.....',
         '....0.',
         '....00',
         '.....',
         '.....']],
    'two_fields_1': [
        ['.....',
         '....0.',
         '....0.',
         '.....',
         '.....'],
        ['.....',
         '....00',
         '.....',
         '.....']}]
```

Следно овие 3 форми се додаваат во речникот на *PIECES* на следниот начин:

```
PIECES = {'S': S_SHAPE_TEMPLATE,
          'Z': Z_SHAPE_TEMPLATE,
          'J': J_SHAPE_TEMPLATE,
          'L': L_SHAPE_TEMPLATE,
          'I': I_SHAPE_TEMPLATE,
          'O': O_SHAPE_TEMPLATE,
          'T': T_SHAPE_TEMPLATE,
          'TH_1': three_or_two_fields_shapes['three_fields_1'],
          'TH_2': three_or_two_fields_shapes['three_fields_2'],
          'TW_1': three_or_two_fields_shapes['two_fields_1']}
```

Излезният екран по направените промени е следният:



3. Обезбедете динамичко подесување на брзината на паѓање на фигурите. Ако играчот исполни парен број редови, зголемете ја брзината за фактор  $\text{NumberOfLevel} * 0.02$ . Во спротивно, ако играчот исполни непарен број редови, тогаш намалете ја брзината за фактор  $\text{NumberOfLevel} * 0.02$ .

Единствените промени се потребни во функцијата за пресметка на брзината - *calculateLevelAndFallFreq*. Додаваме услов кој доколку score-от, т.е. бројот на редови кои се исполнети е парен – тогаш ја зголемуваме брзината на паѓање. Во обратниот случај истата ја намалуваме.

```
def calculateLevelAndFallFreq(score):  
    # Based on the score, return the level the player is on and  
    # how many seconds pass until a falling piece falls one space.  
    level = int(score / 10) + 1  
    if score % 2 == 0:  
        fallFreq = 0.27 - (level * 0.02) #CHANGE: IF THE SCORE IS AN EVEN NUMBER WE ARE INCREASING THE FALLING FREQUENCY  
    else:  
        fallFreq = 0.27 + (level * 0.02) #CHANGE: IF THE SCORE IS AN ODD NUMBER WE ARE DECREASING THE FALLING FREQUENCY  
    return level, fallFreq
```

Излезот од програмата по додавање на овој услов, не може да се долови на слика.