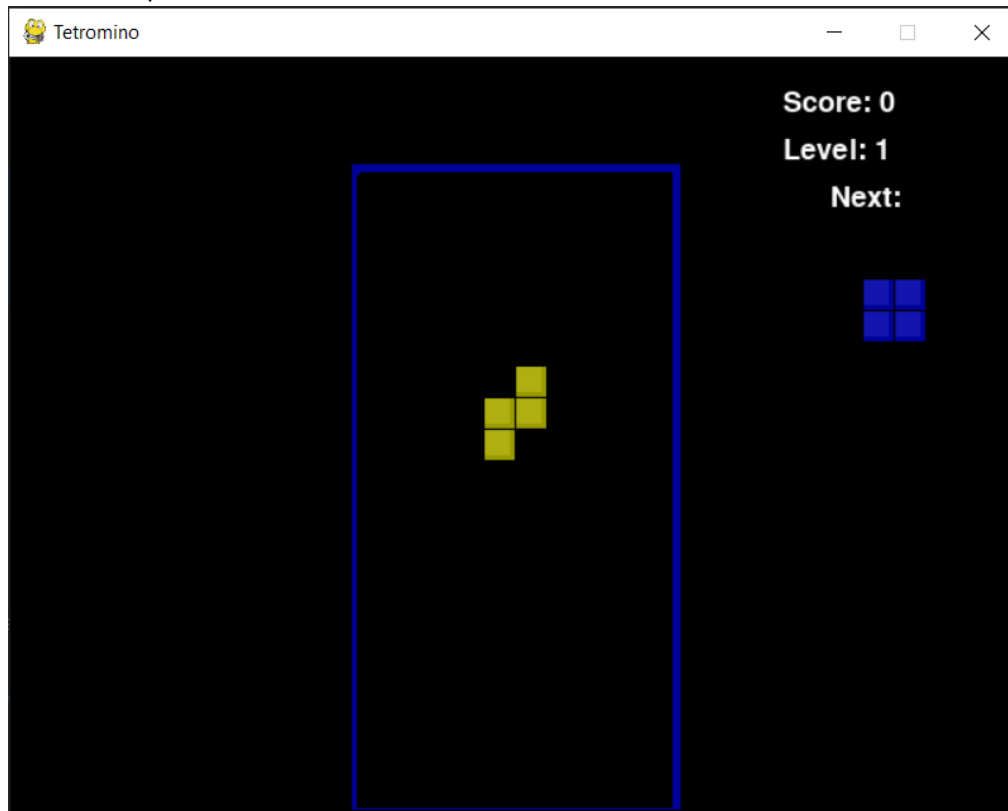


Работа на час Tetromino

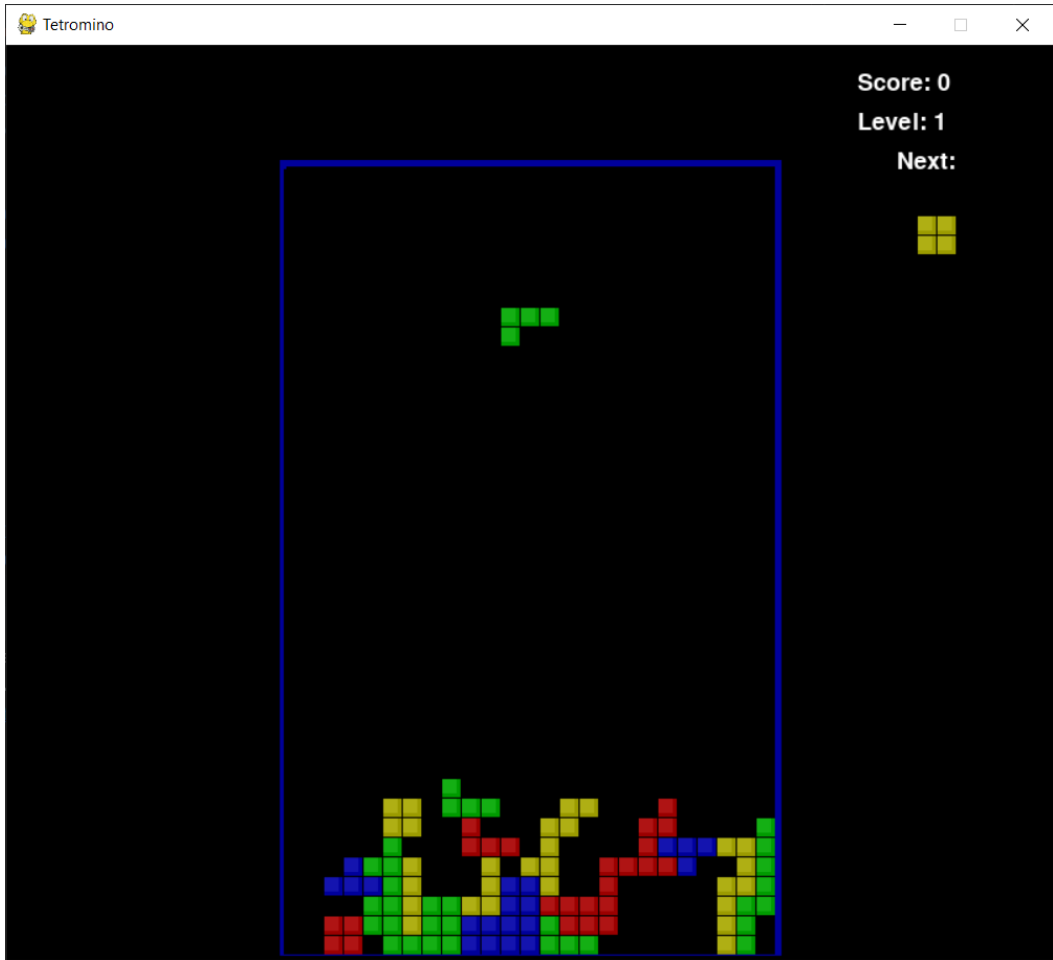
1. Почетен екран



2. Change the variables so that the board is 25 boxes wide and 40 boxes high.

```
WINDOWWIDTH = 800
WINDOWHEIGHT = 700
BOXSIZE = 15
BOARDWIDTH = 25
BOARDHEIGHT = 40
```

BOARDWIDTH го поставуваме на 25 и BOARDHEIGHT на 40 соодветно ги зголемуваме димензиите на екранот и ја намалуваме големината на едно поле со цел да играта да има соодветен приказ.



3. Include a new shape in the game by your choice, define the shape template list and update shape dictionary.

[illegible]

```

'.OO..',
'.....']]

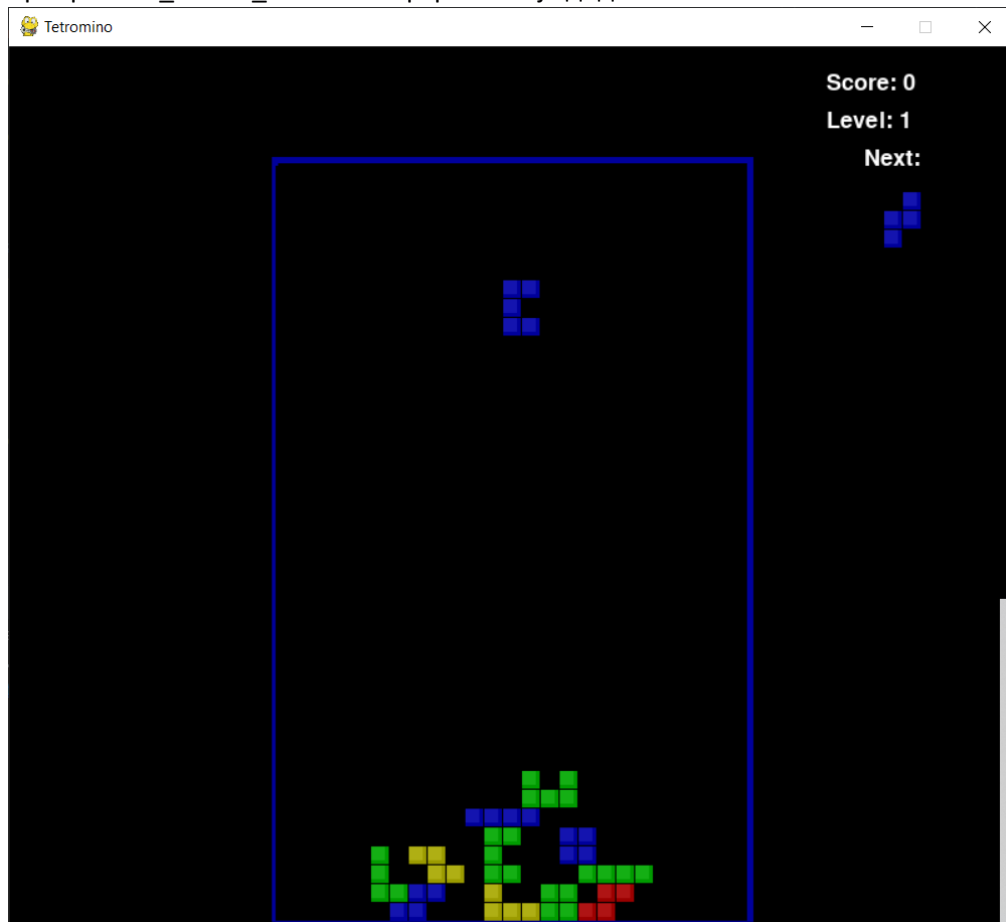
```

```

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,
          # Change for requirement 3:
          'U': U_SHAPE_TEMPLATE,
          }

```

Креираме U_SHAPE_TEMPLATE форма U и ја додаваме во PIECES



4. Write the number of each command that should be commented to provide playing the game without audio-device.

```

# pygame.mixer.music.play(-1, 0.0)

```

Доволни е само командата за пуштање на музика да се искоментира за играта да се игра без звук односно тоа се две линии во `main()` функцијата и во `runGame()` во делот за пауза.

Забелешка: Аудио фајловите сепак ќе се изчитаат во меморија но тие нема да се пуштат односно секогаш ќе бидат во состојба на пауза. Доколку би сакале да не се случува тоа би требало дополнително да закоментираме уште неколку команди и тоа за вчитување на аудио фајлови и командите за стопирање на музиката.

5. First, decrease the falling speed in the game. Then, decrease the rate used to update falling speed in each level. In the text field write the commands that were changed (together with their number). Then answer, at which level the piece cannot fall faster with the given changes?

```
def calculateLevelAndFallFreq(score):  
    # Change for requirement 5:  
    # fallFreq = 0.27 - (level * 0.02)  
    fallFreq = 0.4 - (level * 0.05)
```

Во функцијата која го пресметува нивото и фреквенцијата на паѓање `calculateLevelAndFallFreq(score)`

Првин го зголемуваме почетниот временски интервал од 0.27 секунди во 0.4 секунди со тоа ја намалуваме брзината на играта потоа го зголемуваме факторот со кој секое ново ниво го намалува временскиот интервал од 0.02 во 0.05 и со тоа при секое зголемување на нивото играта ќе се забрзува повеќе.

Кога играта ќе стигне на ниво 8 `fallFreq` ќе стане 0 и со тоа се постигнува максимална брзина.

При секој наредно ниво брзината на играта нема да се зголемува.

Променете е само една наредба

6. Provide possibility to dynamically change the number of erased rows required to change the game level. Start with 10 rows to change the first level to the second and for each next level increase this number for 2. Comment all the added or modified commands with # added or modified to provide full functionality.

```
def calculateLevelAndFallFreq(score):
```

```
...
```

```
# Change for requirement 6:  
  
# level = int(score / 10) + 1  
  
level = 1  
  
while score >= 10:  
    score -= 10+2*level  
    level += 1  
  
...
```

Во функцијата `calculateLevelAndFallFreq(score)` каде што се пресметува нивото на играта го менуваме кодот со цел да се исполни барањето.