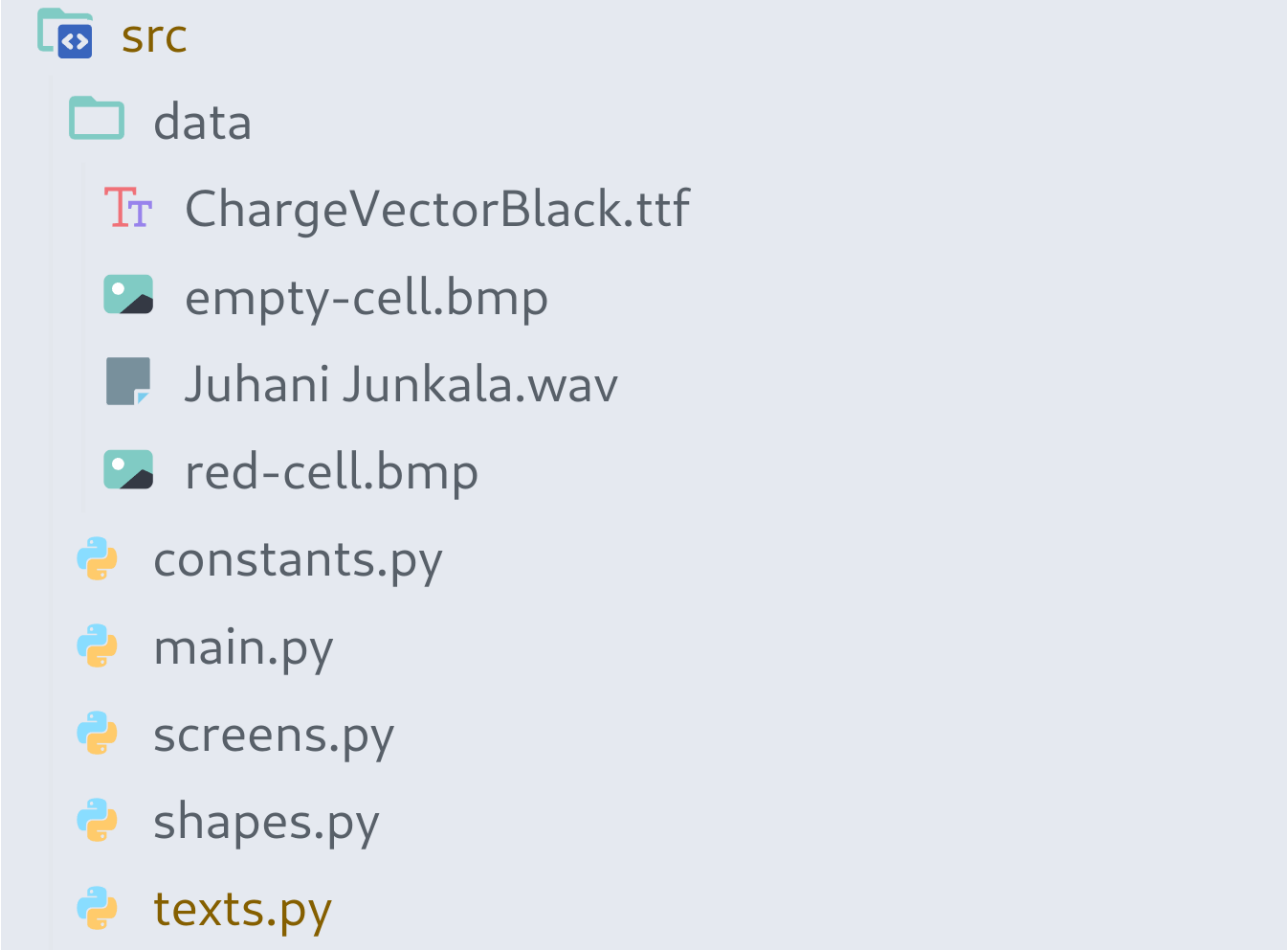
Суриков Никита **@syricov** и Рыбалков Михаил **@erytw**

Проект Pygame

Вступление

Целью проекта было воссоздание оригинальной и никогда не устаревающей игры Tetris с помощью языка Python и модуля PyGame, добавление в проект и дополнительных фишек, например, паузы или дополнительных цветов блоков. Для выполнения данной задачи в команде Использовалась система контроля версий Git и сервис GitHub.

Файловая структура



С целью соблюдения принципов ООП проект разделен на файлы: в папке data хранятся ресурсы, необходимые программе, а сам код разделен на 5 python-файлов.

Структура кода

Рассмотрим входящие в проект python-файлы поподробнее:

- **constants.py**
  В файле задаются основные глобальные переменные, которые используются в проекте: Цвета и формы клеток, размер и форма окна, частота обновления экрана. Так же, именно в этом файле организована функция завершения работы программы, таким образом, файл используется при запуске и завершении работы игры.

```
4   FPS = 60
5   SIZE = WIDTH, HEIGHT = 800, 1000
6   FIELD_SIZE = FIELD_HEIGHT, FIELD_WIDTH = 20, 10
7   BLOCK = pygame.Rect(0, 0, 40, 40)
8   BORDER_W = 5
9
10  # Событие падения блока
11  DOWNEVENT = pygame.USEREVENT + 1
12  # Места хранения спрайтов для клеток разных цветов
13  CELL_COLORS = ("black", "blue", "green", "yellow", ···
15  # Формы падающих фигур для генерации
16  BLOCK_SHAPES = (···
55
56
57  def terminate():···
60
```

- **texts.py**
  В файле заданы класс для красивого отображения текста с использованием нестандартного шрифта(Text), функция мерцающего текста(blink_text), а так же класс для отрисовки и обновления скорборда — Score.

```
6   def blink_text(surface,···
27
28
    Syricov, 4 недели назад | 1 author (Syricov)
29  class Text:
30      def __init__(self, text, size, rect, title=False, color='white'):···
40
41      def draw(self, surface, dim=False):···
45
46      def dim_screen(self, surface):···
50
51      def get_rect(self):···
53
54      def set_color(self, color):···
57
58
    Syricov, 4 недели назад | 1 author (Syricov)
59  class Score:      Syricov, 4 недели назад • Реорганизация структуры прое
60      def __init__(self, pos):···
70
71      def draw(self, surface):···
77
78      def update(self, lines):···
96
97      def get_score(self):···
99
100     def get_level(self):···
102
```

- **screens.py**
  В файле заданы функции для начального экрана(start_screen), экрана паузы(pause_screen), экрана загрузки, экрана завершения попытки — геймовера(game_over), и экрана выхода из игры(exit_screen)

```
7   def start_screen(surface):···
12
13
14  def pause_screen(surface):···
19
20
21  def game_over(surface, score):···
52
53
54  def exit_screen(surface):···
86
```

- **shapes.py**
  Самый массивный и сложный файл программы, состоит из более, чем 300 строк. В нем реализованы классы для хранения, отрисовки и просчитывания взаимодействия клеток(Cell), блоков(Block) и игрового поля(Field)

```
Syricov, 4 недели назад | 1 author (Syricov)
 9  class Cell: ···
46
47
Syricov, 4 недели назад | 1 author (Syricov)
48  class Block: ···
137
138
Syricov, 4 недели назад | 1 author (Syricov)
139  class Field: ···
307
```

- **main.py**
  Главный исполняемый файл программы, создает объекты и использует функции из других модулей, работает с музыкой и основным игровым циклом.

```
11  if __name__ == '__main__':
12      pygame.init()
13      pygame.mixer.music.load("src/data/Juhani Junkala.wav")
14      pygame.display.set_caption('NashTetris')
15      screen = pygame.display.set_mode(SIZE)
16      clock = pygame.time.Clock()
17      pygame.key.set_repeat(200, 100)
18      pygame.time.set_timer(DOWNEVENT, 1000)
19      # Создание объектов
20      field = Field(*FIELD_SIZE)
21      title = Text('NashTetris', 80,···
24      score = Score((WIDTH * 0.77, HEIGHT * 0.55))
25      current_speed = 0
26      # Стартовый экран
27      running = start_screen(screen)
28      # Запускаем музыку
29      pygame.mixer.music.play(loops=-1)
30      while running:···
71      terminate()
72
```