Space Shooter

Максим Харламов

Дарья Шевченко

Объем кода + комментарии + классы

```
    screen.blit(enter, enterR)

    pygame.display.flip()
    clock.tick(60)
    if is_level:
        end_game = True  # флаг того, как закончилась игра
        # выходом или заходом в уровень
        music_play = (
            True  # флаг того, чтоможно снова играть музыку победы/
            проигрыша
        )
        is_levels_window = False
        # r = False
        score = start_level(d[level_n])
        update(name, score)
        win = score > 0  # победил человек или нет
if is_settings_window:
    # окно с настройками
```

```
class Present(pygame.sprite.Sprite):
    """
    types:
    0) меняет тип игрока на 0
    1) меняет тип игрока на 1
    2) меняет тип игрока на 2
    3) увеличивает скорость пуль
    4) уменьшает задержку между пулями
    5) увеличивает скорость игрока
    """

    def __init__(self, x, y, type):
        if not (0 <= type < len(presents)):
            return
        super().__init__(presents_group)
```

```
278             animation_group.draw(screen)
279
280             if not enemy_group.sprites():
281                 running = False
282
283             pygame.display.flip()
284             clock.tick(FPS)
285         return score if len(events) == 1 else 0
286
287
288     if __name__ == "__main__":
289         print(start_level("q.txt"))
290
```

```
class Animation(pygame.sprite.Sprite):
    # в папке должны быть файлы 1.png, 2.png, 3.png ...
    def __init__(self, x, y, folder,
                 n=10):
        super().__init__(animation_group)
        self.images = [load_image(f"{folder}\{i}.png", None,
            in range(1, n + 1)]
        self.index = -1
```

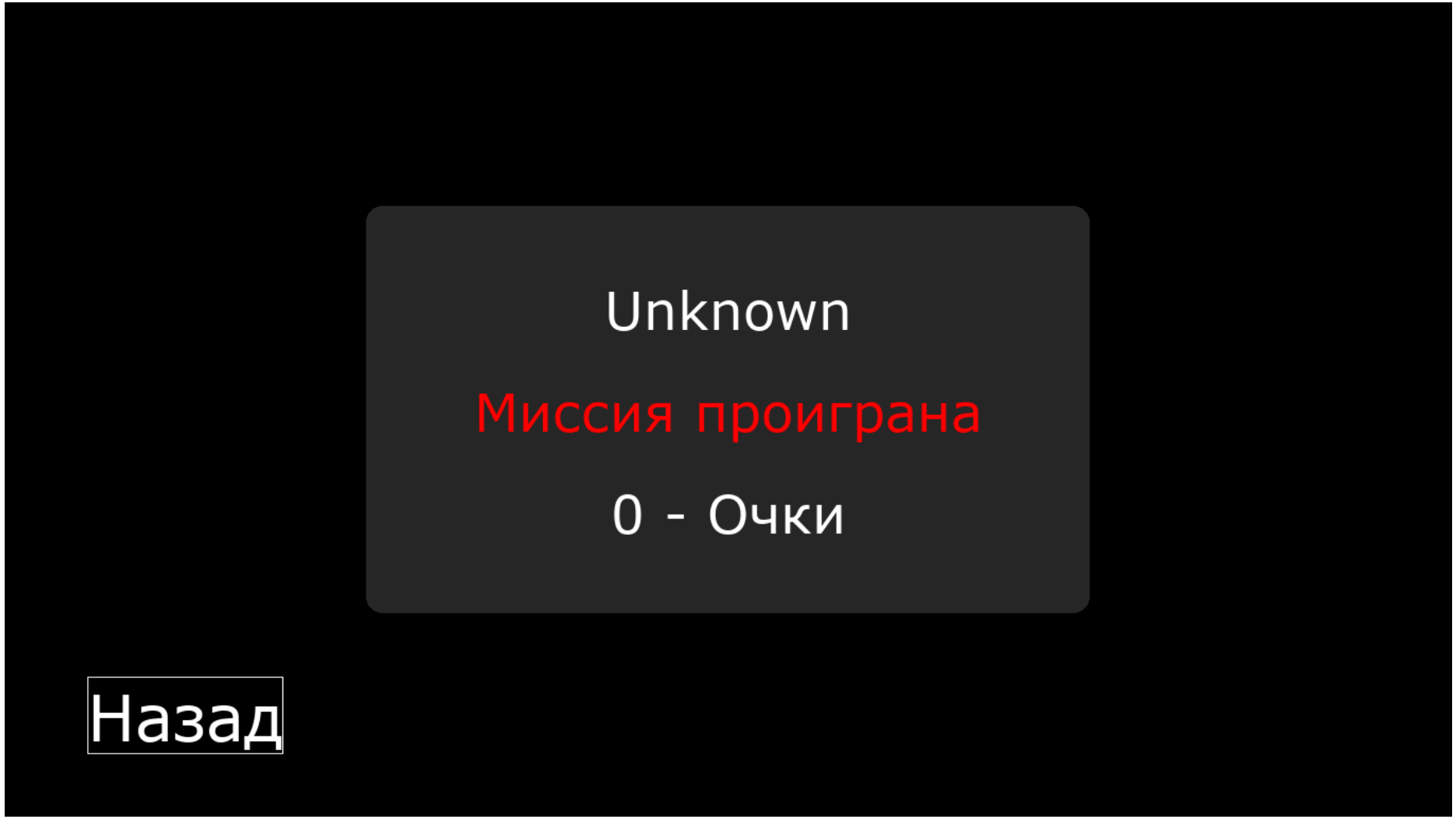
```
class Enemy(pygame.sprite.Sprite):

    def __init__(self, type, pos_x, pos_y):
        super().__init__(enemy_group)
        self.type, image, self.bullet_filename, duration, self.speeds, self.
            points = enemies[type]
        self.image = load_image(image)
        self.mask = pygame.mask.from_surface(self.image)
        self.rect = self.image.get_rect().move(pos_x, pos_y)
        self.rect.x -= self.rect.w // 2
        self.rect.y -= self.rect.h // 2

        EVENT = pygame.USEREVENT + len(events) + 1
        pygame.time.set_timer(EVENT, duration)
```

```
482                 pygame.display.flip()
483                 clock.tick(60)
484
485         pygame.quit()
486
487
488     start_window()
489
```

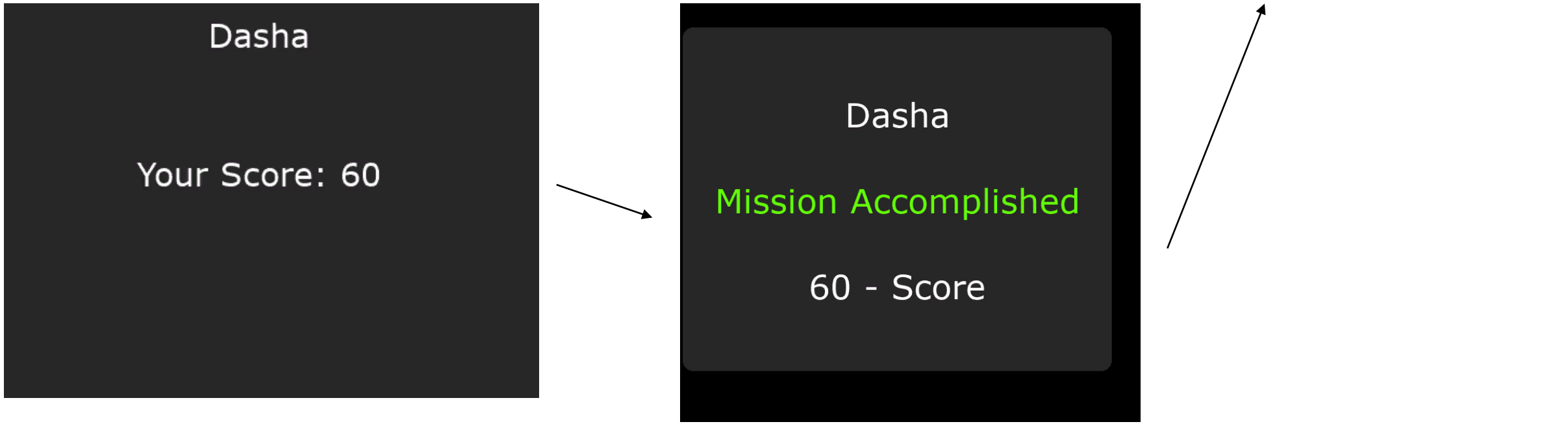
requirements.txt

Стартовое окно

Финальное окно



Подсчет результатов



Спрайты + collide

```
        self.rect.x += self.vx
        self.rect.y += self.vy
        if pygame.sprite.spritecollide(self, player_group, False, pygame.
            sprite.collide_mask):
            presents_group.remove(self)
            if self.type == 3:
```

```
    def update(self):
        global running
        if pygame.sprite.spritecollide(self, enemy_bullets, False, pygame.
            sprite.collide_mask):
            running = False
```

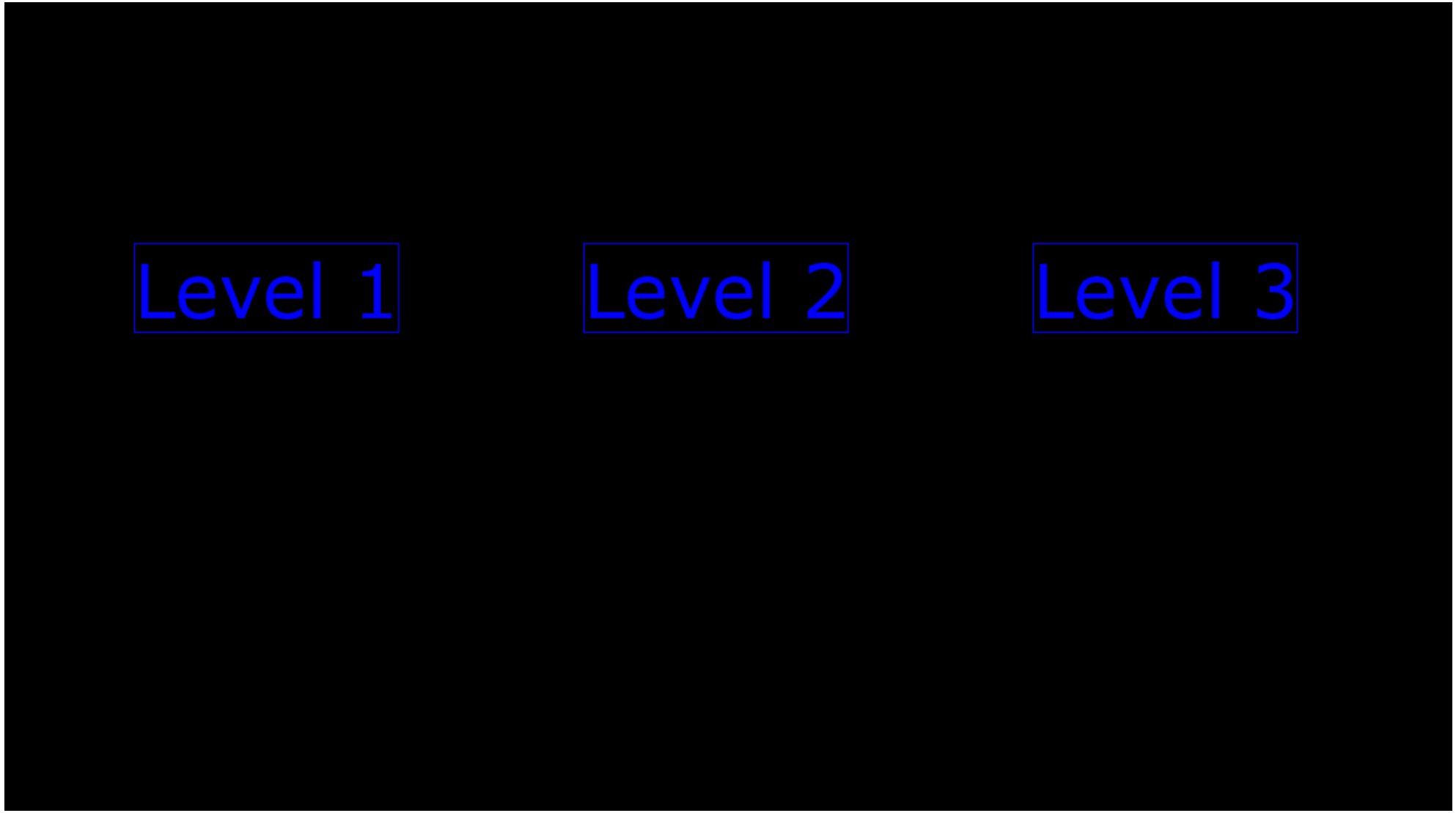
```
player_group = pygame.sprite.Group()
enemy_group = pygame.sprite.Group()
bullet_group = pygame.sprite.Group()
player_bullets = pygame.sprite.Group()
enemy_bullets = pygame.sprite.Group()
presents_group = pygame.sprite.Group()
animation_group = pygame.sprite.Group()
```

```
    player_group.update()
    player_group.draw(screen)

    bullet_group.update()
    bullet_group.draw(screen)

    enemy_group.update()
    enemy_group.draw(screen)

    presents_group.update()
    presents_group.draw(screen)

    animation_group.update()
    animation_group.draw(screen)
```

Анимация

```
class Animation(pygame.sprite.Sprite):
    # в папке должны быть файлы 1.png, 2.png, 3.png ...
    def __init__(self, x, y, folder,
                 n=10):
        super().__init__(animation_group)
        self.images = [load_image(f"{folder}\{i}.png", None, 100, 100) for i
            in range(1, n + 1)]
        self.index = -1
        self.x = x
        self.y = y

    def update(self):
        self.index += 1
        if self.index >= len(self.images):
            animation_group.remove(self)
            return
        self.image = self.images[self.index]
        self.rect = self.image.get_rect().move(self.x, self.y)
        self.rect.x -= self.rect.w // 2
        self.rect.y -= self.rect.h // 2
```

Несколько уровней



Хранение данных

```
connection = sqlite3.connect("data/shooter_bd.db")
cursor = connection.cursor()
cursor.execute(
    """
    CREATE TABLE IF NOT EXISTS Players (
    id INTEGER PRIMARY KEY,
    name TEXT NOT NULL,
    score INTEGER
    );
    """
)
connection.commit()
```