

## Отчёт по домашнему заданию

- **Тема:** Разработка игры с использованием Pygame.
  - **Выполнил:** Егоров Виктор
  - **Группа:** ИУ5-32Б
- 

### Описание задания

Задача заключалась в создании игры на основе библиотеки Pygame. Игрок управляет самолётом, уничтожает врагов и собирает бонусы. Основные элементы игры включают:

- Главный персонаж (самолёт).
  - Враги, разделённые по уровням сложности.
  - Система здоровья, очков и улучшений.
  - Боссы, появляющиеся на каждом уровне.
  - Три уровня с постепенным увеличением сложности.
- 

### Описание программы

Программа состоит из следующих основных частей:

1. **Главный цикл игры**
    - Отвечает за обработку событий, обновление состояния объектов и рендеринг.
  2. **Классы объектов**
    - Самолёт игрока: управление, стрельба, улучшения оружия.
    - Враги и боссы: движение, взаимодействие с игроком и снарядами.
    - Бонусы: восстановление здоровья и улучшения.
  3. **Функции интерфейса**
    - Отображение текстовой информации (очки, здоровье).
    - Меню начала игры и переходов между уровнями.
  4. **Система столкновений**
    - Обработка попаданий пуль, взаимодействия с врагами и бонусами.
  5. **Использование базы данных (SQLite)**
    - Сохранение рекордов и другой информации об игроке.
- 

### Тексты программ

Программа представлена следующим кодом:

```
import random
import sqlite3
import sys

import pygame

width, height = 1400, 675
hero_y, hero_x = 75, 100
```

```

fps = 60
pwu_time = 10000

pygame.init()
pygame.mixer.init()
screen = pygame.display.set_mode((width, height))
pygame.display.set_caption("fancy space")
clock = pygame.time.Clock()
font_name = pygame.font.match_font('Arial Black')

def terminate():
    con = sqlite3.connect("zxc.db")
    cur = con.cursor()
    info = cur.execute(f'SELECT user FROM zxc WHERE user=""')
    if info.fetchone() is None:
        cur.execute(
            f""INSERT INTO zxc(user, score) VALUES('', '0')""
        )
        con.commit()
    txt = ['Вы достигли НЛЮ', 'Вы достигли Тёмного мага', 'Вы достигли  
Культиста']
    sc = [0, 10000, 20000]
    intro_text = [f"Итоговый счёт: {settings.score + sc[settings.level -  
1]}",
                  "",
                  f"Убито боссов: {settings.bosses}",
                  "",
                  f"{txt[settings.level - 1]}",
                  "",
                  "",
                  "Для выхода нажмите esc"]
    fon =
pygame.transform.scale(pygame.image.load('./images/background.jpg'), (1800,
675))
    screen.blit(fon, (0, 0))
    font = pygame.font.Font(None, 30)
    text_coord = 200
    for line in intro_text:
        string_rendered = font.render(line, True, pygame.Color('white'))
        intro_rect = string_rendered.get_rect()
        text_coord += 10
        intro_rect.top = text_coord
        intro_rect.x = 800
        text_coord += intro_rect.height
        screen.blit(string_rendered, intro_rect)

    while True:
        for event in pygame.event.get():
            key_pressed = pygame.key.get_pressed()
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if key_pressed[pygame.K_ESCAPE]:
                pygame.quit()
                sys.exit()
        pygame.display.flip()

def start_screen():
    intro_text = ["Первый уровень"]
    fon =
pygame.transform.scale(pygame.image.load('./images/background.jpg'), (1800,
675))
    screen.blit(fon, (0, 0))

```

```

font = pygame.font.Font(None, 30)
text_coord = 300
for line in intro_text:
    string_rendered = font.render(line, True, pygame.Color('white'))
    intro_rect = string_rendered.get_rect()
    text_coord += 10
    intro_rect.top = text_coord
    intro_rect.x = 800
    text_coord += intro_rect.height
    screen.blit(string_rendered, intro_rect)

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            terminate()
        elif event.type == pygame.KEYDOWN or event.type ==
pygame.MOUSEBUTTONDOWN:
            return
    pygame.display.flip()
    clock.tick(fps)

def between_screen(lvl):
    lvls = ['Второй', 'Третий']
    intro_text = [f"{lvls[lvl]} уровень"]
    fon =
pygame.transform.scale(pygame.image.load('./images/background.jpg'), (1800,
675))
    screen.blit(fon, (0, 0))
    font = pygame.font.Font(None, 30)
    text_coord = 300
    for line in intro_text:
        string_rendered = font.render(line, True, pygame.Color('white'))
        intro_rect = string_rendered.get_rect()
        text_coord += 10
        intro_rect.top = text_coord
        intro_rect.x = 800
        text_coord += intro_rect.height
        screen.blit(string_rendered, intro_rect)

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            terminate()
        elif event.type == pygame.KEYDOWN and event.key ==
pygame.K_SPACE:
            return
    pygame.display.flip()
    clock.tick(fps)

class Settings:
    def __init__(self):
        self.screen_size = (width, height)
        self.all_sprites = pygame.sprite.Group()
        self.score = 0
        self.damage = 1
        self.last_score = 4000
        self.bosses = 0
        self.level = 1
        self.bosses_killed = 0
        self.last_bosses = 0

```

```

class Plane(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.hp = 500
        self.image = pygame.transform.scale(player_img, (hero_x, hero_y))
        self.image.set_colorkey((0, 0, 0))
        self.rect = self.image.get_rect()
        self.radius = int(self.rect.width * .6 / 2)

        self.rect.centerx = 300
        self.rect.bottom = 330
        self.plane_velocity = 10

        self.gun_power = 1
        self.gun_power_time = pygame.time.get_ticks()
        self.delay = 200
        self.shot = pygame.time.get_ticks()

    def shoot(self):
        time = pygame.time.get_ticks()
        if time - self.shot > self.delay:
            self.shot = time
            if self.gun_power == 1:
                25)         bullet = Bullet(self.rect.centerx + 50, self.rect.bottom -

                            all_sprites.add(bullet)
                            bullets.add(bullet)

            if self.gun_power == 2:
                10)         bullet_1 = Bullet(self.rect.centerx + 50, self.rect.bottom -

                            bullet_2 = Bullet(self.rect.centerx + 50, self.rect.bottom -
                40)

                            all_sprites.add(bullet_1)
                            all_sprites.add(bullet_2)
                            bullets.add(bullet_1)
                            bullets.add(bullet_2)

            if self.gun_power >= 3:
                25)         bullet_1 = Bullet(self.rect.centerx + 50, self.rect.bottom -

                            bullet_2 = Bullet(self.rect.centerx + 50, self.rect.bottom)
                            bullet_3 = Bullet(self.rect.centerx + 50, self.rect.bottom -
                50)

                            all_sprites.add(bullet_1)
                            all_sprites.add(bullet_2)
                            all_sprites.add(bullet_3)
                            bullets.add(bullet_1)
                            bullets.add(bullet_2)
                            bullets.add(bullet_3)

    def death(self):
        self.kill()

    def update(self):
        if self.gun_power >= 2 and pygame.time.get_ticks() -
self.gun_power_time > pwu_time:
            self.gun_power -= 1
            self.delay += 15
            if self.delay >= 290:
                self.delay = 290
            self.plane_velocity -= 2
            if self.plane_velocity < 10:
                self.plane_velocity = 10

```

```

        self.gun_power_time = pygame.time.get_ticks()

def gun_powerup(self):
    self.gun_power += 1
    self.delay -= 15
    if self.delay <= 200:
        self.delay = 200
    self.plane_velocity += 2
    if self.plane_velocity > 14:
        self.plane_velocity = 14
    self.gun_power_time = pygame.time.get_ticks()

class Bullet(pygame.sprite.Sprite):
    def __init__(self, x, y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.transform.scale(bullet_img, (60, 25))
        self.image.set_colorkey((0, 0, 0))
        self.rect = self.image.get_rect()

        self.rect.bottom = y
        self.rect.centerx = x
        self.speedx = -10

    def update(self):
        self.rect.x -= self.speedx
        if self.rect.right > 1800:
            self.kill()

class Enemy(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((30, 40))
        self.frame = 0
        self.enemy = random.random()

        if self.enemy > 0.65:
            self.mobs_11 = [pygame.image.load('./images/enemy_11/0.png'),
                             pygame.image.load('./images/enemy_11/1.png'),
                             pygame.image.load('./images/enemy_11/2.png'),
                             pygame.image.load('./images/enemy_11/3.png'),
                             pygame.image.load('./images/enemy_11/4.png')]

            self.image = pygame.transform.scale(self.mobs_11[self.frame],
            (100, 100))
        else:
            self.mobs_12 = [pygame.image.load('./images/enemy_12/0.png'),
                             pygame.image.load('./images/enemy_12/1.png'),
                             pygame.image.load('./images/enemy_12/2.png'),
                             pygame.image.load('./images/enemy_12/3.png'),
                             pygame.image.load('./images/enemy_12/4.png'),
                             pygame.image.load('./images/enemy_12/5.png')]

            self.image = pygame.transform.scale(self.mobs_12[self.frame],
            (100, 120))

        self.image.set_colorkey((0, 0, 0))
        self.rect = self.image.get_rect()
        self.radius = int(self.rect.width / 2)

        self.rect.x = random.randrange(width, width + 200)
        self.rect.y = random.randrange(0, 600)
        self.speedx = random.randrange(-5, -2)

```

```

def update(self):
    self.frame += 1
    if self.enemy > 0.65:
        if self.frame == len(self.mobs_11) * 10:
            self.frame = 0
            self.image = pygame.transform.scale(self.mobs_11[self.frame //
10], (70, 70))
        else:
            if self.frame == len(self.mobs_12) * 10:
                self.frame = 0
                self.image = pygame.transform.scale(self.mobs_12[self.frame //
10], (70, 90))
            self.rect.x += self.speedx
            if self.rect.right <= 0:
                self.kill()

class Enemy2(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((30, 40))
        self.frame = 0
        self.enemy = random.random()

        if self.enemy > 0.65:
            self.mobs_21 = [pygame.image.load('./images/enemy_21/0.png'),
                             pygame.image.load('./images/enemy_21/1.png'),
                             pygame.image.load('./images/enemy_21/2.png'),
                             pygame.image.load('./images/enemy_21/3.png')]

            self.image = pygame.transform.scale(self.mobs_21[self.frame],
(80, 80))
        else:
            self.mobs_22 = [pygame.image.load('./images/enemy_22/0.png'),
                             pygame.image.load('./images/enemy_22/1.png'),
                             pygame.image.load('./images/enemy_22/2.png'),
                             pygame.image.load('./images/enemy_22/3.png'),
                             pygame.image.load('./images/enemy_22/4.png'),
                             pygame.image.load('./images/enemy_22/5.png')]

            self.image = pygame.transform.scale(self.mobs_22[self.frame],
(80, 60))

        self.image.set_colorkey((0, 0, 0))
        self.rect = self.image.get_rect()
        self.radius = int(self.rect.width / 2)

        self.rect.x = random.randrange(width, width + 200)
        self.rect.y = random.randrange(0, 600)
        self.speedx = random.randrange(-7, -3)

    def update(self):
        self.frame += 1
        if self.enemy > 0.65:
            if self.frame == len(self.mobs_21) * 10:
                self.frame = 0
                self.image = pygame.transform.scale(self.mobs_21[self.frame //
10], (80, 80))
            else:
                if self.frame == len(self.mobs_22) * 10:
                    self.frame = 0
                    self.image = pygame.transform.scale(self.mobs_22[self.frame //
10], (80, 60))

```

```

        self.rect.x += self.speedx
        if self.rect.right <= 0:
            self.kill()

class Enemy3(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((30, 40))
        self.frame = 0
        self.enemy = random.random()

        if self.enemy > 0.9:
            self.mobs_31 = [pygame.image.load('./images/enemy_31/0.png'),
                             pygame.image.load('./images/enemy_31/1.png'),
                             pygame.image.load('./images/enemy_31/2.png'),
                             pygame.image.load('./images/enemy_31/3.png'),
                             pygame.image.load('./images/enemy_31/4.png'),
                             pygame.image.load('./images/enemy_31/5.png')]

            self.image = pygame.transform.scale(self.mobs_31[self.frame],
            (250, 240))
        elif 0.9 > self.enemy > 0.5:
            self.mobs_32 = [pygame.image.load('./images/enemy_32/0.png'),
                             pygame.image.load('./images/enemy_32/1.png'),
                             pygame.image.load('./images/enemy_32/2.png'),
                             pygame.image.load('./images/enemy_32/3.png')]

            self.image = pygame.transform.scale(self.mobs_32[self.frame],
            (50, 60))
        else:
            self.mobs_33 = [pygame.image.load('./images/enemy_33/0.png'),
                             pygame.image.load('./images/enemy_33/1.png'),
                             pygame.image.load('./images/enemy_33/2.png'),
                             pygame.image.load('./images/enemy_33/3.png'),
                             pygame.image.load('./images/enemy_33/4.png'),
                             pygame.image.load('./images/enemy_33/5.png')]

            self.image = pygame.transform.scale(self.mobs_33[self.frame],
            (70, 75))

        self.image.set_colorkey((0, 0, 0))
        self.rect = self.image.get_rect()
        self.radius = int(self.rect.width / 2)

        self.rect.x = random.randrange(width, width + 200)
        self.rect.y = random.randrange(0, 550)
        self.speedx = random.randrange(-7, -3)

    def update(self):
        self.frame += 1
        if self.enemy > 0.9:
            if self.frame == len(self.mobs_31) * 40:
                self.frame = 0
            self.image = pygame.transform.scale(self.mobs_31[self.frame //
            40], (250, 240))
        elif 0.9 > self.enemy > 0.5:
            if self.frame == len(self.mobs_32) * 40:
                self.frame = 0
            self.image = pygame.transform.scale(self.mobs_32[self.frame //
            40], (50, 60))
        else:
            if self.frame == len(self.mobs_33) * 10:
                self.frame = 0

```

```

        self.image = pygame.transform.scale(self.mobs_33[self.frame //
10], (70, 75))

        self.rect.x += self.speedx
        if self.rect.right <= 0:
            self.kill()

class Cultist(pygame.sprite.Sprite):
    def __init__(self, y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((30, 40))
        self.frame = 0
        self.cult = [pygame.image.load('./images/boss_31/0.png'),
            pygame.image.load('./images/boss_31/1.png'),
            pygame.image.load('./images/boss_31/2.png'),
            pygame.image.load('./images/boss_31/3.png'),
            pygame.image.load('./images/boss_31/4.png'),
            pygame.image.load('./images/boss_31/5.png')]

        self.image = pygame.transform.scale(self.cult[self.frame], (80, 105))

        self.image.set_colorkey((0, 0, 0))
        self.rect = self.image.get_rect()
        self.radius = int(self.rect.width / 2)

        self.rect.x = width + 100
        self.rect.y = y
        self.speedx = -1

    def update(self):
        self.frame += 1
        if self.frame == len(self.cult) * 40:
            self.frame = 0
        self.image = pygame.transform.scale(self.cult[self.frame // 40], (80,
105))

        self.rect.x += self.speedx
        if self.rect.right <= 0:
            self.kill()

class Crows(pygame.sprite.Sprite):
    def __init__(self, y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((30, 40))
        self.frame = 0
        self.crow = [pygame.image.load('./images/boss_21/0.png'),
            pygame.image.load('./images/boss_21/1.png'),
            pygame.image.load('./images/boss_21/2.png'),
            pygame.image.load('./images/boss_21/3.png')]

        self.image = pygame.transform.scale(self.crow[self.frame], (60, 60))

        self.image.set_colorkey((0, 0, 0))
        self.rect = self.image.get_rect()
        self.radius = int(self.rect.width / 2)

        self.rect.x = width + 100
        self.rect.y = y
        self.speedx = -1

    def update(self):
        self.frame += 1

```



```

        if self.frame == len(self.crow) * 40:
            self.frame = 0
        self.image = pygame.transform.scale(self.crow[self.frame // 40], (60,
60))

        self.rect.x += self.speedx
        if self.rect.right <= 0:
            self.kill()

class Ufos(pygame.sprite.Sprite):
    def __init__(self, y):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.Surface((30, 40))
        self.frame = 0
        self.cult = [pygame.image.load('./images/boss_11/0.png'),
                      pygame.image.load('./images/boss_11/1.png'),
                      pygame.image.load('./images/boss_11/2.png'),
                      pygame.image.load('./images/boss_11/3.png')]

        self.image = pygame.transform.scale(self.cult[self.frame], (72, 60))

        self.image.set_colorkey((0, 0, 0))
        self.rect = self.image.get_rect()
        self.radius = int(self.rect.width / 2)

        self.rect.x = width + 100
        self.rect.y = y
        self.speedx = -1

    def update(self):
        self.frame += 1
        if self.frame == len(self.cult) * 40:
            self.frame = 0
        self.image = pygame.transform.scale(self.cult[self.frame // 40], (72,
60))

        self.rect.x += self.speedx
        if self.rect.right <= 0:
            self.kill()

class Boss(pygame.sprite.Sprite):
    def __init__(self, *group):
        super().__init__(*group)
        self.hp = 10 * settings.level
        self.frame = 0

        self.images_1 = [pygame.image.load('./images/boss_1/0.png'),
                          pygame.image.load('./images/boss_1/1.png'),
                          pygame.image.load('./images/boss_1/2.png'),
                          pygame.image.load('./images/boss_1/3.png'),
                          pygame.image.load('./images/boss_1/4.png'),
                          pygame.image.load('./images/boss_1/5.png'),
                          pygame.image.load('./images/boss_1/6.png'),
                          pygame.image.load('./images/boss_1/7.png')]
        self.images_2 = [pygame.image.load('./images/boss_2/0.png'),
                          pygame.image.load('./images/boss_2/1.png'),
                          pygame.image.load('./images/boss_2/2.png'),
                          pygame.image.load('./images/boss_2/3.png'),
                          pygame.image.load('./images/boss_2/4.png')]
        self.images_3 = [pygame.image.load('./images/boss_3/0.png'),
                          pygame.image.load('./images/boss_3/1.png'),
                          pygame.image.load('./images/boss_3/2.png')]

```

```

        self.image = pygame.Surface((30, 40))

        if settings.level == 1:
            self.image = pygame.transform.scale(self.images_1[self.frame],
(400, 150))

        if settings.level == 2:
            self.image = pygame.transform.scale(self.images_2[self.frame],
(180, 140))

        if settings.level == 3:
            self.image = pygame.transform.scale(self.images_3[self.frame],
(82, 120))

        self.image.set_colorkey((0, 0, 0))
        self.rect = self.image.get_rect()
        self.radius = int(self.rect.width / 2)
        # pygame.draw.circle(self.image, (0, 0, 255), self.rect.center,
self.radius)

        self.rect.x = width + 50
        self.rect.y = height // 4 + 125
        self.speedx = -1

    def boss_death(self):
        settings.bosses_killed += 1
        settings.bosses += 1
        if settings.level == 1:
            for ufo in ufos:
                ufo.kill()
        if settings.level == 2:
            for crow in crows:
                crow.kill()
        if settings.level == 3:
            for cultist in cultists:
                cultist.kill()
        self.kill()

    def update(self):
        self.frame += 1

        if settings.level == 1:
            if self.frame == len(self.images_2) * 15:
                self.frame = 0
                self.image = pygame.transform.scale(self.images_1[self.frame //
15], (400, 150))

        if settings.level == 2:
            if self.frame == len(self.images_2) * 15:
                self.frame = 0
                self.image = pygame.transform.scale(self.images_2[self.frame //
15], (180, 140))

        if settings.level == 3:
            if self.frame == len(self.images_3) * 50:
                self.frame = 0
                self.image = pygame.transform.scale(self.images_3[self.frame //
50], (82, 120))

        self.rect.x += self.speedx
        if self.rect.right <= 0:
            main_plane.hp -= 300
            self.hp = 0

```

```

        self.boss_death()

class PowerUps(pygame.sprite.Sprite):
    def __init__(self, center):
        pygame.sprite.Sprite.__init__(self)
        self.type = random.choice(['med', 'gun'])
        self.image = powerups_images[self.type]
        self.rect = self.image.get_rect()
        self.rect.center = center
        self.speedx = 1

    def update(self):
        self.rect.x -= self.speedx
        if self.rect.right <= 0:
            self.kill()

def draw_text(surf, text, size, x, y):
    font = pygame.font.Font(font_name, size)
    text_surface = font.render(text, True, (70, 150, 200))
    text_rect = text_surface.get_rect()
    text_rect.midtop = (x, y)
    surf.blit(text_surface, text_rect)

def draw_hp_bar(surf, x, y, value):
    if value < 0:
        value = 0
    bar_length = 120
    bar_height = 20
    fill = (value / 500) * bar_length
    outline_rect = pygame.Rect(x, y, bar_length, bar_height)
    fill_rect = pygame.Rect(x, y, fill, bar_height)
    if main_plane.hp >= 250:
        pygame.draw.rect(surf, (100, 240, 70), fill_rect)
        pygame.draw.rect(surf, (255, 255, 255), outline_rect, 2)
    elif main_plane.hp < 250:
        pygame.draw.rect(surf, (255, 0, 0), fill_rect)
        pygame.draw.rect(surf, (255, 255, 255), outline_rect, 2)

def draw_boss_delay(surf, x, y, value):
    bar_length = 130
    bar_height = 20
    fill = (value / 3000) * bar_length
    if fill > 130:
        fill = 130
    outline_rect = pygame.Rect(x, y, bar_length, bar_height)
    fill_rect = pygame.Rect(x, y, fill, bar_height)
    pygame.draw.rect(surf, (70, 140, 200), fill_rect)
    pygame.draw.rect(surf, (255, 255, 255), outline_rect, 2)

def main():
    start_screen()
    running = True
    invis = False
    background_x = width
    size = (width, height)
    while running:
        screen.blit(background, (-background_x + 1200, 0))
        screen.blit(background, (-background_x, 0))
        screen.blit(background, (-background_x - 1200, 0))

```

```

screen.blit(background, (-background_x - 2400, 0))

background_x += 3
if background_x >= 0:
    background_x = background_x - 2400
clock.tick(fps)

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        terminate()

key_pressed = pygame.key.get_pressed()
if key_pressed[pygame.K_s]:
    coord_y = main_plane.rect.bottom + main_plane.plane_velocity
    if coord_y >= size[1]:
        print('down')
    else:
        main_plane.rect.bottom += main_plane.plane_velocity
if key_pressed[pygame.K_w]:
    coord_y = main_plane.rect.bottom - main_plane.plane_velocity
    if coord_y <= size[1] and coord_y < 70:
        print('up')
    else:
        main_plane.rect.bottom -= main_plane.plane_velocity
if key_pressed[pygame.K_a]:
    coord_x = main_plane.rect.centerx - main_plane.plane_velocity
    if coord_x <= size[0] and coord_x < 50:
        print('left')
    else:
        main_plane.rect.centerx -= main_plane.plane_velocity
if key_pressed[pygame.K_d]:
    coord_x = main_plane.rect.centerx + main_plane.plane_velocity
    if coord_x >= size[0] - 25:
        print('right')
    else:
        main_plane.rect.centerx += main_plane.plane_velocity
if key_pressed[pygame.K_SPACE]:
    main_plane.shoot()
all_sprites.update()
if len(mobs) <= 5:
    spawn()

hits = pygame.sprite.groupcollide(mobs, bullets, True, True)
for hit in hits:
    settings.score += 100
    m = 1
    if settings.level == 1:
        m = Enemy()
        all_sprites.add(m)
        if random.random() > 0.95:
            pow = PowerUps(hit.rect.center)
            all_sprites.add(pow)
            powerups.add(pow)

    if settings.level == 2:
        m = Enemy2()
        all_sprites.add(m)
        if random.random() > 0.93:
            pow = PowerUps(hit.rect.center)
            all_sprites.add(pow)
            powerups.add(pow)

    if settings.level == 3:
        m = Enemy3()

```

```

        all_sprites.add(m)
        if random.random() > 0.9:
            pow = PowerUps(hit.rect.center)
            all_sprites.add(pow)
            powerups.add(pow)

    if settings.score == 1000:
        spawn_boss()

    if settings.last_score - settings.score <= 0:
        if settings.bosses_killed - settings.last_bosses == 1:
            settings.last_score += 4000
            settings.last_bosses = settings.bosses_killed
            spawn_boss()
    m.kill()

hits = pygame.sprite.groupcollide(bos, bullets, invis, True)
for hit in hits:
    boss.hp -= settings.damage
    if boss.hp == 1:
        invis = True

    if boss.hp <= 0:
        boss.boss_death()
        settings.score += 200
        invis = False

    for i in range(settings.level + 1):
        pow = PowerUps(hit.rect.center)
        all_sprites.add(pow)
        powerups.add(pow)

    boss.hp = 10 * settings.level
hits = pygame.sprite.spritecollide(main_plane, powerups, True)
for hit in hits:
    if hit.type == 'med':
        main_plane.hp += random.choice([50, 60, 70, 80, 90, 100])
        if main_plane.hp >= 500:
            main_plane.hp = 500
    if hit.type == 'gun':
        main_plane.gun_powerup()

hits = pygame.sprite.spritecollide(main_plane, mobs, True,
pygame.sprite.collide_circle)
for hit in hits:
    main_plane.hp -= hit.radius * 2
    if main_plane.hp <= 0:
        terminate()

summons = [ufos, crows, cultists]
sms = ['ufos', 'crows', 'cultists']

hits = pygame.sprite.groupcollide(summons[settings.level - 1],
bullets, False, True)
hits = pygame.sprite.spritecollide(main_plane, summons[settings.level
- 1], False, pygame.sprite.collide_circle)
for hit in hits:
    print(sms[settings.level - 1])
    main_plane.hp -= (hit.radius // 20) * settings.level
    if main_plane.hp <= 0:
        terminate()

# КОЛЛИЗИЯ БОССОВ
hits = pygame.sprite.spritecollide(main_plane, bos, False,

```

```

pygame.sprite.collide_circle)
    for hit in hits:
        main_plane.hp -= hit.radius // 40
        if main_plane.hp <= 0:
            terminate()

    # смена уровней
    if settings.level == 1 and settings.score >= 10000 and
settings.bosses_killed >= 3:
        between_screen(0)
        settings.last_score = 4000
        settings.level = 2
        for item in mobs:
            item.kill()
            mobs.clear(screen, background)
            mobs.draw(screen)
        for item in powerups:
            item.kill()
            powerups.clear(screen, background)
        for item in bos:
            item.kill()
            bos.clear(screen, background)
        for item in ufos:
            item.kill()
            ufos.clear(screen, background)
        boss.hp = 10 * settings.level
        settings.score = 0
        settings.bosses_killed = 0
        settings.last_bosses = 0

    if settings.level == 2 and settings.score >= 20000 and
settings.bosses_killed >= 5:
        between_screen(1)
        settings.last_score = 4000
        settings.level = 3
        for item in mobs:
            item.kill()
            mobs.clear(screen, background)
            mobs.draw(screen)
        for item in powerups:
            item.kill()
            powerups.clear(screen, background)
        for item in bos:
            item.kill()
            bos.clear(screen, background)
        for item in crows:
            item.kill()
            crows.clear(screen, background)
        boss.hp = 10 * settings.level
        settings.score = 0
        settings.bosses_killed = 0
        settings.last_bosses = 0

    all_sprites.update()
    all_sprites.draw(screen)
    draw_text(screen, 'Счёт: ' + str(settings.score), 18, width / 2, 10)
    draw_text(screen, 'Здоровье', 18, 110, 10)
    draw_hp_bar(screen, 50, 50, main_plane.hp)
    if settings.score >= 1000:
        draw_text(screen, 'Наступление', 18, 265, 10)
        draw_boss_delay(screen, 200, 50, settings.last_score -
settings.score)

    pygame.display.flip()

```

```

def spawn():
    if settings.level == 1:
        for i in range(10):
            e = Enemy()
            all_sprites.add(e)
            mobs.add(e)
    if settings.level == 2:
        for i in range(15):
            e = Enemy2()
            all_sprites.add(e)
            mobs.add(e)
    if settings.level == 3:
        for i in range(20):
            e = Enemy3()
            all_sprites.add(e)
            mobs.add(e)

def spawn_boss():
    if settings.level == 1:
        y = [150, 550]
        for i in range(2):
            ufo = Ufos(y[i])
            ufos.add(ufo)
            all_sprites.add(ufo)
    if settings.level == 2:
        y = [50, 150, 450, 550]
        for i in range(4):
            crow = Crows(y[i])
            crows.add(crow)
            all_sprites.add(crow)
    if settings.level == 3:
        y = [150, 450]
        for i in range(2):
            cu = Cultist(y[i])
            cultists.add(cu)
            all_sprites.add(cu)
    print('boss was spawned')
    q = Boss(all_sprites)
    all_sprites.add(q)
    bos.add(q)

powerups_images = {}
pressed = {}
powerups_images['med'] = pygame.image.load('./images/med.png')
powerups_images['gun'] = pygame.image.load('./images/power.png')
background = pygame.image.load('./images/background.jpg')
player_img = pygame.image.load('./images/hero_sprite.png')
bullet_img = pygame.image.load('./images/bullet.png')

all_sprites = pygame.sprite.Group()
mobs = pygame.sprite.Group()
bos = pygame.sprite.Group()
ufos = pygame.sprite.Group()
crows = pygame.sprite.Group()
cultists = pygame.sprite.Group()
bullets = pygame.sprite.Group()
powerups = pygame.sprite.Group()

settings = Settings()
main_plane = Plane()

```

```
boss = Boss()
all_sprites.add(main_plane)
enemy = Enemy()

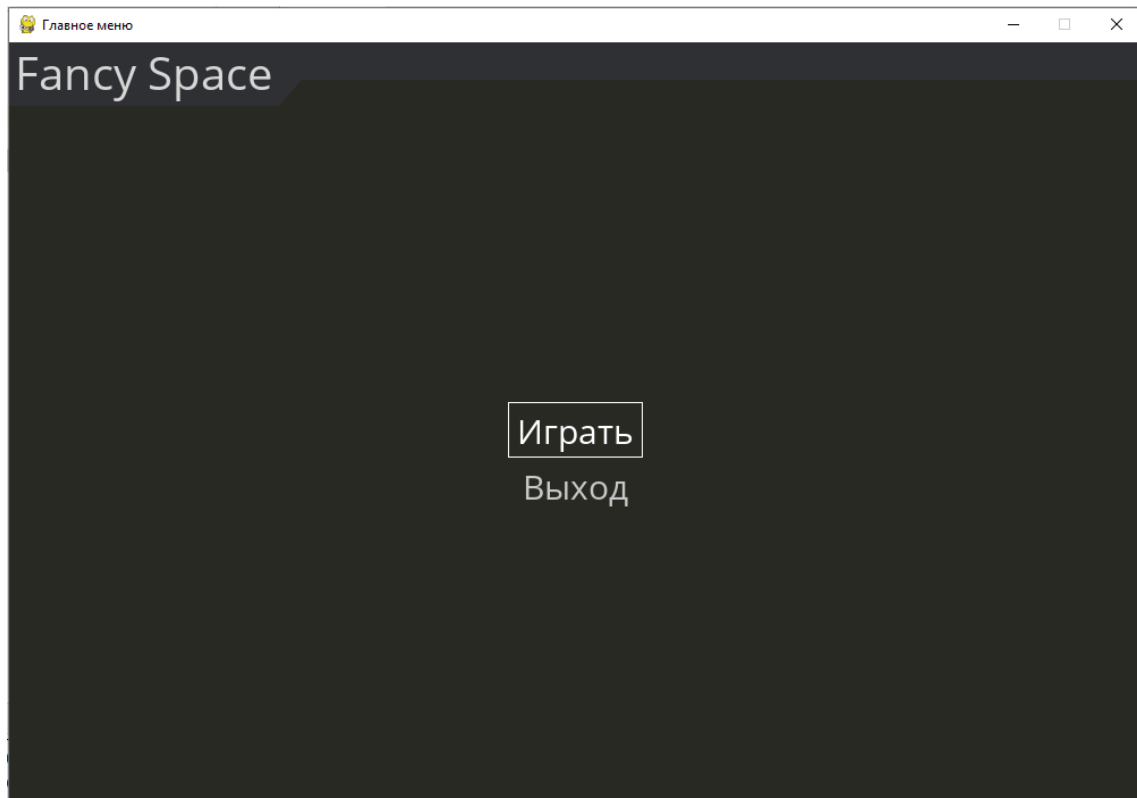
main()
pygame.quit()
```

---

## Экранные формы

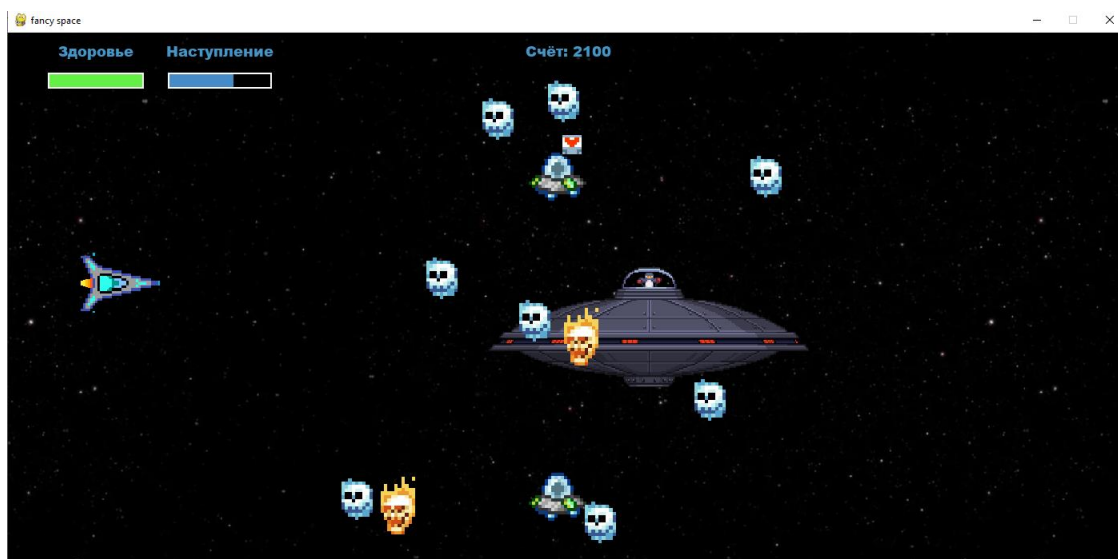
### 1. Главное меню

- Отображает текст начала игры и вводит игрока в первый уровень.



### 2. Игровой процесс

- Фон с бесшовным скроллингом, спрайты самолёта, врагов и бонусов.
- Панель очков и здоровья в верхней части экрана.





### 3. Переход между уровнями

- Отображается сообщение о следующем уровне и пауза перед началом.



### 4. Конец игры

- Итоговая таблица с результатами игрока (очки, количество убитых боссов).



