

BLACK CUBE

Автор проекта: Игнатов Михаил

ВВЕДЕНИЕ

Моя игра представляет собой пародию на популярный платформер «Red Ball». Задача игрока добраться до портала, находящегося в конце и уровня, собрав как можно больше монет. Для того, чтобы усложнить прохождение игры, на каждом уровне предусмотрены, как статичные, так и подвижные противники. При контакте с противниками игроку наносится урон, если количество очков здоровья игрока опуститься до нуля, то игрок погибает, и игра завершается

Описание реализации

Основные классы:

- Player
- Camera
- Button
- Классы противников
- Классы игровых объектов

```
# Ящик
class Box(pygame.sprite.Sprite):
    def __init__(self, tile_type, pos_x, pos_y):
        super().__init__(box_group, all_sprites)
        self.image = tile_images[tile_type]
        self.rect = self.image.get_rect().move(
            tile_width * pos_x, tile_height * pos_y)

    def erase(self):
        self.kill()

# Каменная стена
class StoneWall(pygame.sprite.Sprite):
    def __init__(self, pos_x, pos_y, width, height):
        super().__init__(stone_wall_group, all_sprites)
        self.image = pygame.image.load('data/images/stone.png')
        self.rect = pygame.Rect(pos_x, pos_y, width, height)

    def erase(self):
        self.kill()

# Кольцо
class Spike(pygame.sprite.Sprite):
    def __init__(self, pos_x, pos_y, width, height):
        super().__init__(spike_group, all_sprites)
        self.image = pygame.image.load('data/images/spike.png')
        self.type = 'spike'
        self.rect = pygame.Rect(pos_x, pos_y, width, height)
        self.damage = 1

    def update(self, screen):
        if pygame.sprite.collide_mask(self, player):
            player.health -= self.damage
            self.kill()

class Button():
    def __init__(self, image, pos, level=''):
        self.image = image
        self.x_pos = pos[0]
        self.y_pos = pos[1]
        self.level = level
        self.rect = self.image.get_rect(center=(self.x_pos, self.y_pos))
        self.sound_button = pygame.mixer.Sound("data/sounds/pressing the button.ogg")

    def update(self, screen):
        if self.image is not None:
            screen.blit(self.image, self.rect)

    def checkForInput(self, position):
        if position[0] in range(self.rect.left, self.rect.right) \
            and position[1] in range(self.rect.top, self.rect.bottom):
            self.sound_button.play()
            if self.level != '':
                return self.level
            else:
                return True
        return False
```

Описание реализации

Основные функции:

- generate_level()
- place_text()
- load_image()
- start_screen()
- main_menu()
- end_screen()
- game_rule_screen()
- game()

```
import pygame
from functions import terminate, place_text
from button import Button
from load_image import load_image
from game_rule_screen import game_rules_screen
```

```
def main_menu(screen, WIDTH, HEIGHT, clock, FPS): # Главное меню
    file = open("data/result.txt", "r", encoding='utf-8')
    file_text = file.read().strip().split()
    points = 'Всего монет собрано: ' + str(int(file_text[0]))
    file.close()

    fon = pygame.transform.scale(load_image('Backgro.jpg'), (WIDTH, HEIGHT))
    screen.blit(fon, (0, 0))

    text = ''
```

```
    image1 = load_image('level1.png')
    image2 = load_image('level2.png')
    image3 = load_image('level3.png')
    image4 = load_image('Rules.png')

    level1_button = Button(image1, (340, 200), 'level1')
    level2_button = Button(image2, (355, 300), 'level2')
    level3_button = Button(image3, (355, 400), 'level3')
    rules_button = Button(image4, (350, 500))
```

```
    menu_running = True
    while menu_running:
```

```
        menu_running = bool = True
```

```
        place_text(screen, points, 210, 60)
        place_text(screen, text, 80, 90)
        level1_button.update(screen)
```

```
import pygame
from functions import terminate, place_text
from variables import START_SOUND
from load_image import load_image
```

```
def start_screen(screen, WIDTH, HEIGHT, clock, FPS): # Стартовый экран
    intro_text = 'Для продолжения нажмите ПКМ или ЛКМ'
    START_SOUND.set_volume(0.5)
```

```
    fon = pygame.transform.scale(load_image('startfon.jpg'), (WIDTH, HEIGHT))
    screen.blit(fon, (0, 0))
```

```
    place_text(screen, intro_text, 100, 500)
    START_SOUND.play()
```

```
    while True:
```

```
        for event in pygame.event.get():
```

```
            if event.type == pygame.QUIT:
```

```
                terminate()
```

```
            elif event.type == pygame.MOUSEBUTTONDOWN:
```

```
                return
```

```
        pygame.display.flip()
```

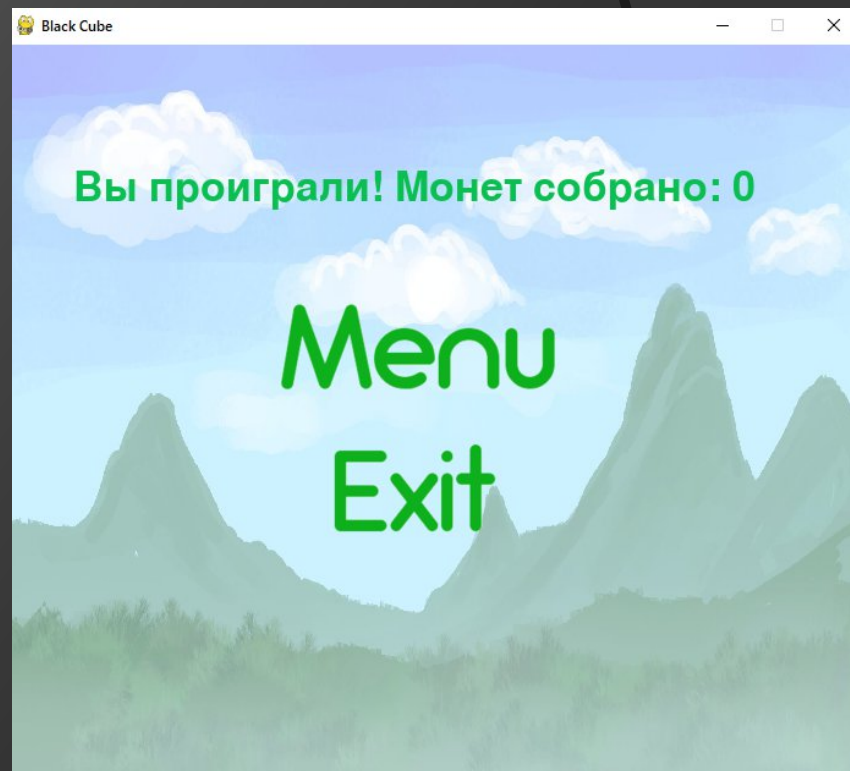
```
        clock.tick(FPS)
```

Интерфейс

Главное меню



Финальное окно



Интерфейс

