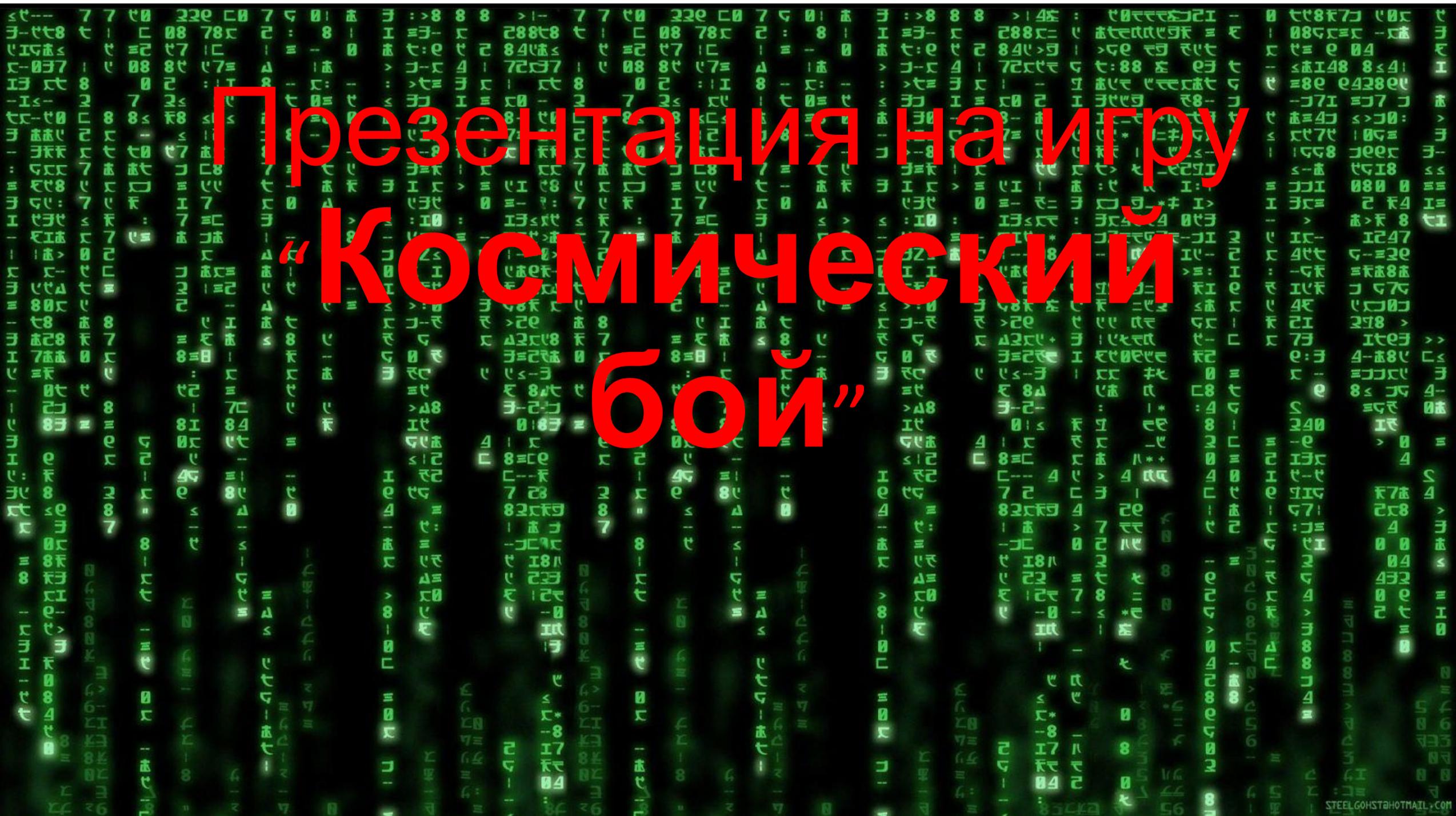


Презентация на игру "Космический бой"



Разделы

Создаем разделы:

- Враг
- Игрок
- И главный раздел, где

и описывается весь основной код.

Сразу создаем самого игрока как цикл

И даем ему “права” на движение.

Тоже самое с метеоритом, только движение будет осуществляться автоматически.

```
def update(self):
    keys = pygame.key.get_pressed()
    if keys[pygame.K_UP]:
        self.rect.top -= 5
    if keys[pygame.K_DOWN]:
        self.rect.top += 5
    if keys[pygame.K_LEFT]:
        self.rect.left -= 5
    if keys[pygame.K_RIGHT]:
        self.rect.left += 5
```

Спрайты и классы

Но, так как нам нечем управлять, некого уничтожать, добавляем:

1. Фон

```
background_image = pygame.image.load('cosmos-NASA.png')
background_image = background_image.convert()
colorkey = background_image.get_at((0, 0))
background_image.set_colorkey(colorkey)
background_image = pygame.transform.scale(background_image, (width, height))
```

2. Космолёт (игрок)

```
class PlayerObject(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = pygame.image.load('косм_корабль.png')
        self.image = self.image.convert_alpha()
        self.image = pygame.transform.scale(self.image, (70, 70))
        self.rect = self.image.get_rect()
        self.rect.top = 0
        self.rect.left = 0
```

3. Враг (астероиды)

```
class Enemy(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = pygame.image.load('астероид.png')
        self.image = self.image.convert()
        colorkey = self.image.get_at((0, 0))
        self.image.set_colorkey(colorkey)
        self.image = pygame.transform.scale(self.image, (100, 100))
        self.rect = self.image.get_rect()
```

Многочисленные инопришельцы

Для того, чтобы астероид был не один, создаем функцию “randit”, рандомный спавн астероидов.

```
enemy_sprites.draw(win)
```

Также выводим на экран, И делаем “живыми”

```
all_sprites.update()
```

```
player.rect.left = width - 290  
player.rect.top = height - 100
```

```
enemy = Enemy.Enemy()  
enemy.rect.left = 200
```

```
enemy1 = Enemy.Enemy()  
enemy1.rect.left = 100
```

```
enemy2 = Enemy.Enemy()  
enemy2.rect.left = 300
```

```
all_sprites.add(player)
```

```
enemy_sprites = pygame.sprite.Group()
```

```
enemy_sprites.add(enemy)  
enemy_sprites.add(enemy1)  
enemy_sprites.add(enemy2)
```

Пуля и её соприкосновение

Для того, чтобы уничтожить противников, требуется пуля, следовательно, требуется код, который будет считывать координаты пули и врага, и при наложении удалять оба спрайта.

- Создаем пулю, т.к. еще один класс делать нет смысла, пулю будет простым красным прямоугольником. А для того, чтобы она появлялась на конце корабля, считываем координаты x , y игрока и по y находим середину, там появится наша пуля.

```
x = player.rect.centerx  
y = player.rect.top - 3
```

- Соприкосновение пули с врагом

Пишем код, который будет набрасывать спрайт астероида на спрайт пули и смотреть, накладываются ли они. Тем самым пропадают и пули, теперь игровой счет, если уничтожили астероид, то в строку score добавляется +100, а если метеорит попал в нас, то из строки health: n -1.

Здоровье:

```
f1 = pygame.font.Font(None, 36)
text1 = f1.render('Health: '+str(3), True, (health))
```

Счет:

```
f2 = pygame.font.Font(None, 36)
text2 = f2.render('Score: '+str(0), True, (0))
```

Конец игры

- Для того чтобы игра закончилась, мы должны потратить все жизни. Следовательно, если жизни заканчиваются заливаем экран черным цветом, и пишем “Game Over”.

```
if health <= 0:  
    win.fill((255, 255, 255))  
    f3 = pygame.font.Font(None, 50)  
    text3 = f3.render('Game over ', True, ())
```

**Спасибо за
внимание!**