

# ОСНОВЫ РАБОТЫ С PYGAME

01001

00101

01100

010

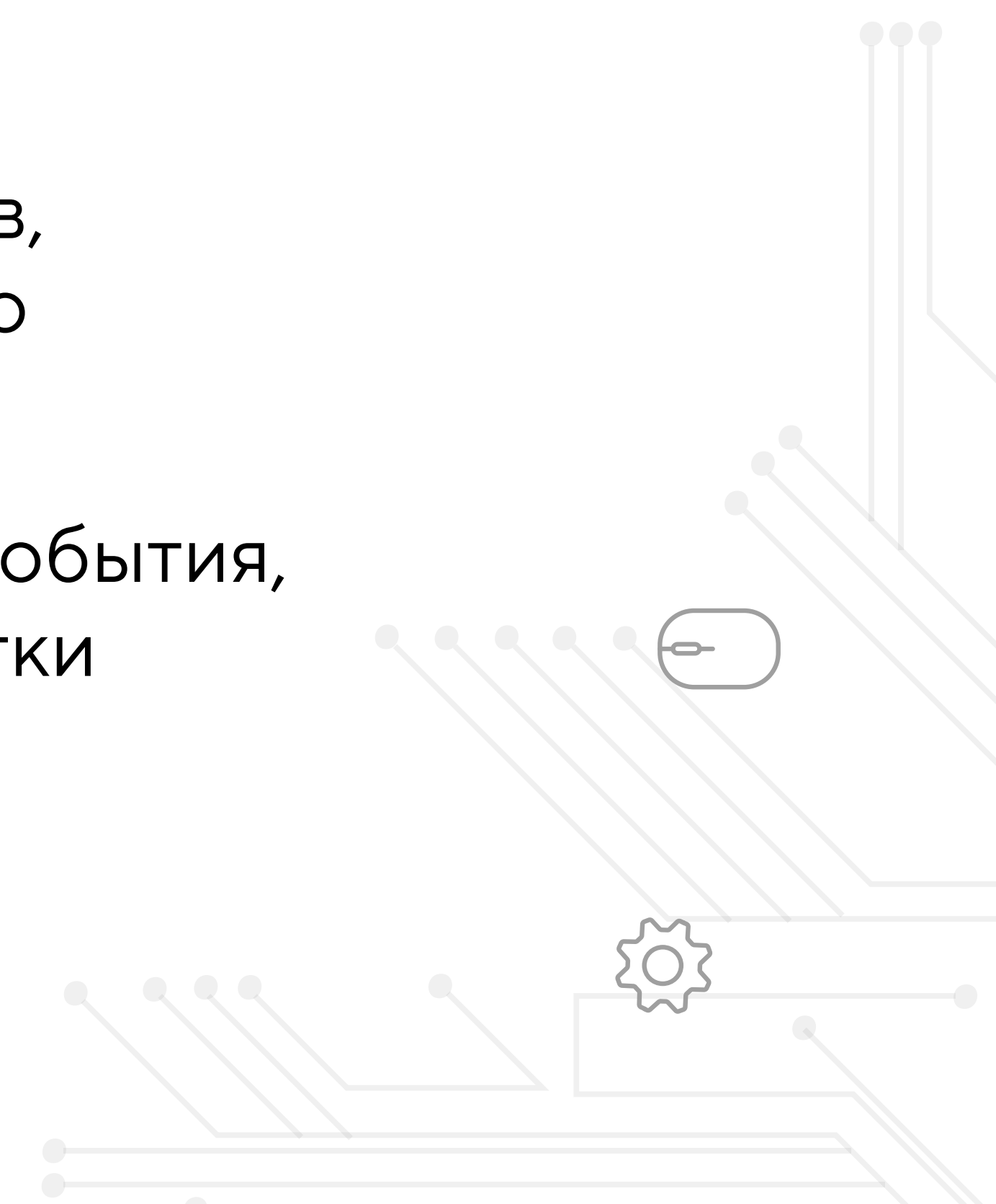
001

0110

# Работа с pygame

Pygame предоставляет множество инструментов для создания игр и мультимедийных приложений. Вот некоторые из основных элементов работы с Pygame:

- Создание окна в Pygame: Pygame позволяет создавать окна различных размеров и форматов. Для создания окна в Pygame необходимо использовать функцию `pygame.display.set_mode()`.
- Работа с графикой в Pygame: Pygame предоставляет множество инструментов для работы с графикой, включая создание спрайтов, текстур и анимации. Для работы с графикой в Pygame необходимо использовать модуль `pygame.draw`.
- Обработка событий в Pygame: Pygame позволяет обрабатывать события, такие как нажатие клавиш, движение мыши и другие. Для обработки событий в Pygame необходимо использовать функцию `pygame.event.get()`.



# Создание окна

```
import pygame  
  
# Инициализация Pygame  
pygame.init()
```

Первым шагом мы импортируем модуль `pygame` и инициализируем его с помощью функции `pygame.init()`. Это необходимо для корректной работы всех функций и инструментов Pygame.

```
# Задание размеров окна  
  
width = 800  
height = 600
```

Затем мы задаем размеры окна, указывая ширину и высоту в пикселях. В данном примере мы установили ширину 800 пикселей и высоту 600 пикселей.

```
# Создание окна  
  
screen = pygame.display.set_mode((width, height))
```



# Основной цикл игры

```
# Основной игровой цикл
```

```
running = True  
while running:
```

```
# Обработка событий
```

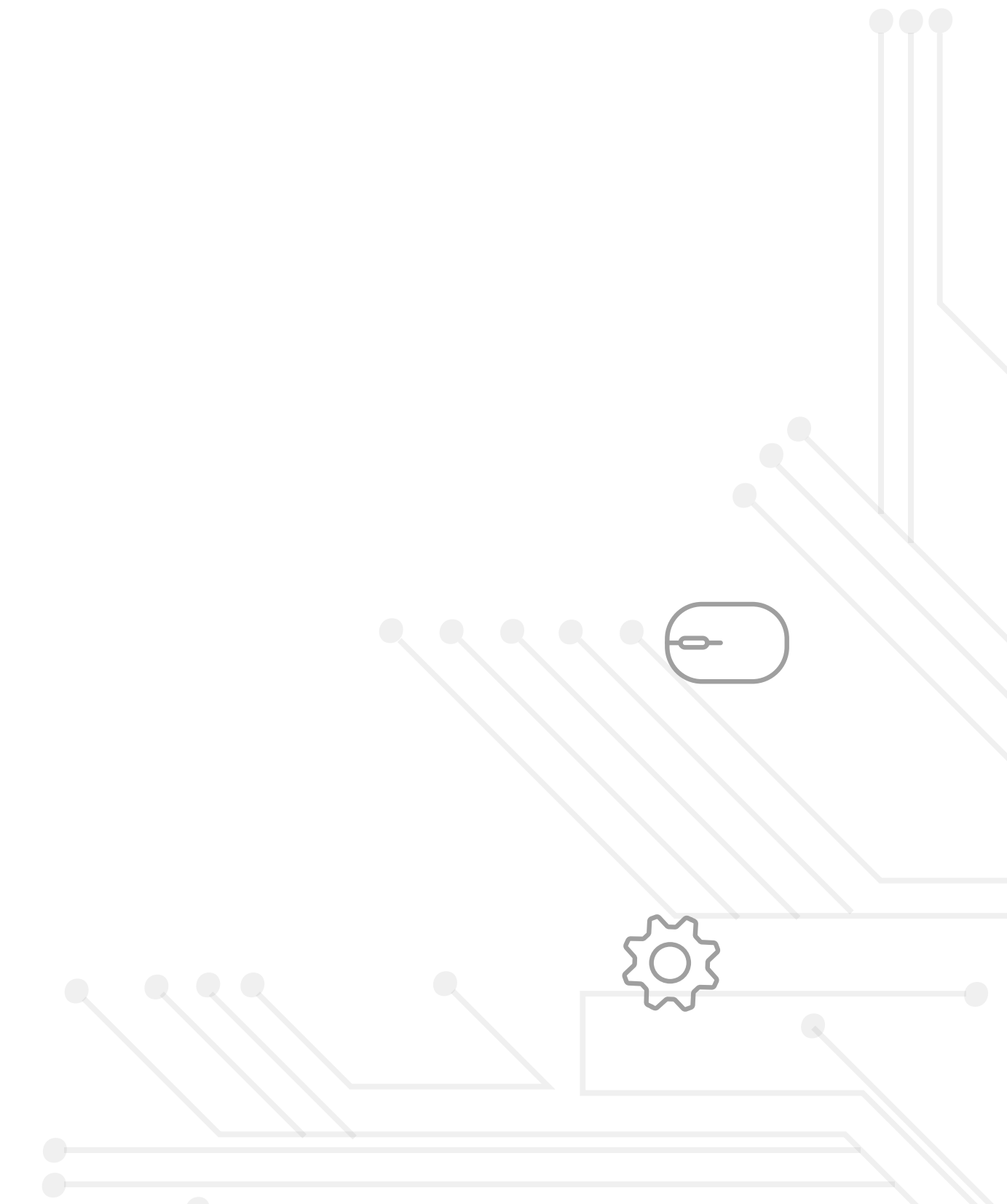
```
for event in pygame.event.get():  
    if event.type == pygame.QUIT:  
        running = False
```

```
# Отрисовка на экране
```

```
screen.fill((0, 0, 0)) # Заливка экрана черным цветом
```

```
# Обновление экрана
```

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



## Описание кода

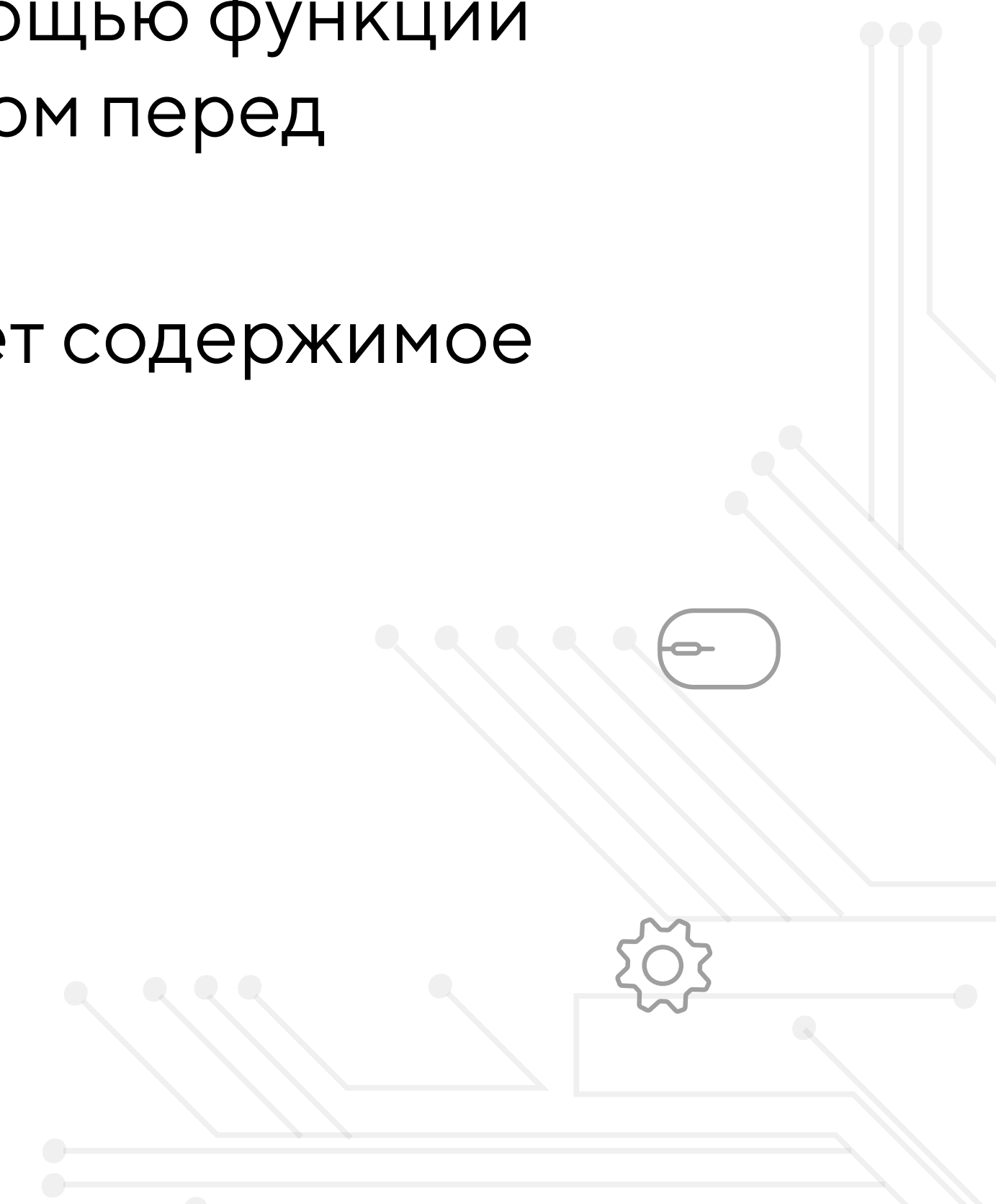
Затем мы создаем основной игровой цикл, который будет выполняться до тех пор, пока переменная `running` равна `True`. Внутри цикла мы обрабатываем события, такие как нажатие клавиши закрытия окна. Если пользователь нажимает на кнопку закрытия окна, мы изменяем значение переменной `running` на `False`, чтобы выйти из цикла и завершить программу.

Внутри цикла мы также отрисовываем на экране черный цвет с помощью функции `screen.fill((0, 0, 0))`. Это очищает экран и заполняет его черным цветом перед отрисовкой нового кадра.

Затем мы вызываем функцию `pygame.display.flip()`, которая обновляет содержимое окна и отображает новый кадр на экране.

```
# Завершение работы Pygame  
pygame.quit()
```

В конце программы мы вызываем функцию `pygame.quit()`, которая завершает работу Pygame и освобождает ресурсы, занятые им.



# Заключение

Pygame – это мощная библиотека для создания игр и мультимедийных приложений на Python. Она предоставляет множество инструментов для работы с графикой, звуком, управлением и другими аспектами игр. Для изучения Pygame можно использовать различные ресурсы, такие как документацию, учебники и сообщества разработчиков. Pygame позволяет создавать игры различных жанров, включая платформеры, шутеры, головоломки и многое другое. Если вы заинтересованы в создании игр на Python, Pygame – отличный выбор

