

УДК 004.423

РАЗРАБОТКА ИГР С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ PYGAME НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Кулинча П.В.

*студент направления подготовки информатика и вычислительная техника,
Хакасский государственный университет имени Н.Ф. Катанова,
г. Абакан, Россия ¹*

Аннотация: Разработка игр является одним из наиболее популярных направлений в сфере программирования. В последние годы игры стали не только развлечением, но и предметом серьезного изучения и коммерческого успеха. Существует множество инструментов и библиотек, которые облегчают процесс создания игр. В данной статье мы рассмотрим разработку игр с использованием библиотеки PyGame на языке программирования Python. PyGame предоставляет простой и удобный интерфейс для создания 2D игр и приложений, что делает его популярным выбором для начинающих разработчиков.

Ключевые слова: Языки программирования, Python, PyGame.

GAME DEVELOPMENT USING THE PYGAME LIBRARY IN THE PYTHON PROGRAMMING LANGUAGE

Kulincha P.V.

*student of computer science and computer engineering department,
N.F. Katanov Khakass State University,
Abakan, Russia*

Abstract: Game development is one of the most popular areas in the field of programming. In recent years, games have become not only entertainment, but also the subject of serious study and commercial success. There are many tools and libraries

¹ Научный руководитель: Спирин Д.В. доцент кафедры ЦТиД, Хакасский государственный университет имени Н.Ф. Катанова, г. Абакан, Россия

that facilitate the process of creating games. In this article, we will look at the development of games using the PyGame library in the Python programming language. PyGame provides a simple and user-friendly interface for creating 2D games and applications, which makes it a popular choice for novice developers.

Keywords: Programming languages, Python, PyGame.

Перед началом работы с PyGame необходимо установить его на компьютер [1]. Для этого можно воспользоваться менеджером пакетов `pip`, выполнив команду в командной строке (Рис. 1).

```
1 pip install pygame
```

Рисунок 1 – Установка пакета PyGame [разработано автором]

После установки PyGame можно приступить к созданию игр. В начале программы необходимо импортировать библиотеку PyGame (Рис. 2).

```
3 import pygame
```

Рисунок 2 – Импорт библиотеки PyGame [разработано автором]

PyGame предоставляет широкий набор функций и классов для работы с графикой, звуком, анимацией и управлением. Рассмотрим базовую структуру программы на PyGame (Рис. 3).

```
1 import pygame
2 pygame.init()
3
4 # Создание окна игры
5 screen = pygame.display.set_mode((800, 600))
6 pygame.display.set_caption("Моя игра")
7
8 # Основной цикл игры
9 running = True
10 while running:
11     for event in pygame.event.get():
12         if event.type == pygame.QUIT:
13             running = False
14
15     screen.fill((255, 255, 255))
16     pygame.display.flip()
17
18 pygame.quit()
```

Рисунок 3 – Базовая структура программы PyGame [разработано автором]

В данном примере мы создаем окно игры размером 800x600 пикселей и задаем ему заголовок "Моя игра". Затем мы запускаем основной цикл игры, который будет выполняться до тех пор, пока переменная `running` равна `True`. Внутри цикла мы проверяем события, такие как нажатия клавиш или закрытие окна, и обновляем графическое состояние игры.

PyGame предоставляет мощные инструменты для работы с изображениями и анимацией. Для загрузки изображения в игру можно использовать функцию `pygame.image.load()` (Рис. 4) [2].

```
player_image = pygame.image.load("player.png")
```

Рисунок 4 – Загрузка изображения в PyGame [разработано автором]

Здесь "player.png" - это путь к файлу изображения. Загруженное изображение можно отобразить на экране с помощью команды `screen.blit(player_image, (x, y))`. Где `x` и `y` - координаты позиции изображения на экране. Для создания анимации можно использовать последовательность изображений и переключать их с определенной частотой.

PyGame позволяет легко обрабатывать пользовательский ввод, такой как нажатия клавиш или движение мыши. Например, чтобы обрабатывать нажатия клавиш, можно использовать код, представленный на рисунке 5 [3].

```
1 > for event in pygame.event.get():  
2 >     if event.type == pygame.KEYDOWN:  
3 >         if event.key == pygame.K_SPACE:  
4 >             # выполнить действие при нажатии пробела
```

Рисунок 5 – Обработка ввода в PyGame [разработано автором]

Анимация добавляет живость и реалистичность в игру. PyGame обладает мощными средствами для создания и управления анимацией [4]. На рисунке 6 приведен пример кода, демонстрирующий анимацию движения спрайта.

```
import pygame

# Инициализация PyGame
pygame.init()

# Создание окна игры
screen = pygame.display.set_mode((800, 600))

# Загрузка спрайта
sprite_sheet = pygame.image.load('spritesheet.png')

# Разделение спрайта на кадры
frames = []
frame_width = 64
frame_height = 64
for i in range(4):
    frame = sprite_sheet.subsurface((i * frame_width, 0, frame_width, frame_height))
    frames.append(frame)

# Управление анимацией
frame_index = 0
animation_speed = 0.2 # Скорость смены кадров (кадры в секунду)
last_frame_time = pygame.time.get_ticks()

# Основной цикл игры
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    current_time = pygame.time.get_ticks()
    if current_time - last_frame_time >= animation_speed * 1000:
        frame_index = (frame_index + 1) % len(frames)
        last_frame_time = current_time

    current_frame = frames[frame_index]
    screen.blit(current_frame, (0, 0))

    pygame.display.update()

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

Рисунок 6 – Анимация движения в PyGame [разработано автором]

PyGame также поддерживает воспроизведение звуковых эффектов. Для этого необходимо загрузить звуковой файл с помощью функции `pygame.mixer.Sound()` и воспроизводить его при необходимости (Рис. 6) [5].

```
sound = pygame.mixer.Sound("sound.wav")
sound.play
```

Рисунок 6 – Загрузка и воспроизведение звука в PyGame [разработано автором]

На рисунке 7 приведен пример кода, демонстрирующий воспроизведение звука при нажатии клавиши.

```
import pygame

# Инициализация PyGame
pygame.init()

# Создание окна игры
screen = pygame.display.set_mode((800, 600))

# Загрузка звукового файла
sound = pygame.mixer.Sound('sound.wav')

# Основной цикл игры
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                # Воспроизведение звука
                sound.play()

    pygame.display.update()

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

Рисунок 7 – Воспроизведение звука во время движения в PyGame [разработано автором]

В данной статье мы рассмотрели основы разработки игр с использованием библиотеки PyGame на языке программирования Python. PyGame предоставляет удобные инструменты для работы с графикой, звуком и управлением, что делает процесс создания игр доступным даже для начинающих разработчиков. Мы рассмотрели примеры кода и объяснили их работу. Приобретенные знания могут быть использованы для разработки собственных игр и приложений на основе PyGame.

Библиографический список:

1. Кондратьева, В.А. Основы программирования на языке Python [Текст] / В.А. Кондратьева. – М.: МГП, 2022. – 68 с.

2. Макаренко Л.Ф. Программирование на языке Python [Текст] / Л.Ф. Макаренко, И.С. Шувалова. – М.: Московский автомобильно-дорожный государственный технический университет (МАДИ), 2022. – 88 с.
3. Бэрри, П. Изучаем программирование на Python [Текст] / П. Бэрри. – М.: Вильямс, 2014. – 243 с.
4. Гуриков, С.Р. Основы алгоритмизации и программирования на Python [Текст] / С.Р. Гуриков. – М.: ИНФРА-М, 2023. – 343 с.
5. Гэддис, Т. Начинаем программировать на Python [Текст] / Т. Гэддис. – СПб.: БХВ-Петербург, 2021. – 768 с.

Оригинальность 77%