

Разработка и реализация интерактивного аудиоплеера с визуализацией на основе библиотеки Pygame

Языков Кирилл Александрович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье представлена разработка интерактивного аудиоплеера с визуализацией в виде имитации эквалайзера, реализованного с использованием библиотеки Pygame. Плеер обладает интуитивно понятным графическим интерфейсом, позволяющим пользователю выбирать аудиофайлы, управлять воспроизведением (воспроизведение, пауза, остановка), а также наблюдать визуальное отображение активности аудиосигнала в режиме реального времени. Предложенный подход позволяет создавать простые, но эффективные приложения для воспроизведения аудио с элементами визуализации, используя возможности свободно распространяемой библиотеки Pygame. Несмотря на имитационный характер визуализации, данная разработка может служить основой для дальнейших исследований в области интерактивных аудиоприложений.

Ключевые слова: Pygame, аудиоплеер, визуализация, эквалайзер, графический интерфейс, интерактивное приложение, Python, разработка программного обеспечения, имитация, визуализация звука.

Improving file upload security using Drag-and-Drop: A comprehensive approach to validation and protection

Yazykov Kirill Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article discusses the implementation of a secure file upload system in web applications based on multi-level validation of the client and server sides, as well as dynamic content analysis methods. A comprehensive protection method against downloading malicious files is presented, based on a combination of MIME type verification, file extensions, signature analysis, and integration with antivirus software. An experimental analysis of the effectiveness of the proposed system in detecting and preventing the download of various types of malicious code has been carried out. The possibilities of using this approach to solve the simplest tasks of ensuring the security of web applications are discussed and the limitations of the proposed methods are considered.

Keywords: File upload security, file validation, MIME type verification, content analysis, protection against malicious code, web applications, JavaScript, PHP, multi-level protection, Drag-and-Drop, protection against attacks.

1 Введение

1.1 Актуальность

Разработка интерактивных мультимедийных приложений с использованием открытого программного обеспечения является актуальной задачей в области информационных технологий. Библиотека Pygame предоставляет удобный инструментарий для создания графических интерфейсов и работы с аудио, что делает ее привлекательной для разработки подобных приложений. Разработка простого и эффективного аудиоплеера с визуализацией может служить учебным пособием для студентов, изучающих программирование на Python, а также основой для создания более сложных мультимедийных приложений.

1.2 Обзор исследований

А. С. Матвеева в своей работе демонстрирует применение паттерна проектирования «Команда» для создания модульных и гибких приложений на Tkinter, повышая их расширяемость и удобство модификации. [1] В области медицины, А. М. Вохидов, Д.А.Вохидов, Р.Ф. Фармонова и Д. Ш. Хафизова исследовали использование Tkinter для создания интуитивно понятных интерфейсов медицинских приложений, подчеркивая удобство данной технологии для медицинских работников и пациентов. [2] Д. А Алигулович, З. И. Махмудов и Р. Ф. Сайфуллаев анализируют преимущества и недостатки Tkinter в контексте разработки GUI-приложений, особо отмечая его применимость в образовании для создания кроссплатформенных решений. [3] М. Моиза представил практический подход к архивированию данных из социальных сетей, разработав Python-скрипт на основе библиотеки Instaloader, автоматизирующий процесс сохранения постов и генерирующий PDF-отчеты. [4] Наконец, П. Анжанейулу, Д. Приянка и Я. Манасвини разработали систему на основе ReportLab для автоматической транскрипции аудио в PDF, используя методы распознавания речи и обработки естественного языка, что особенно актуально для сфер здравоохранения и образования. [5]

1.3 Цель исследования

Разработать и реализовать функциональный аудиоплеер с интуитивно понятным графическим интерфейсом и визуализацией аудиосигнала с использованием библиотеки Pygame, демонстрируя возможности библиотеки для создания интерактивных мультимедийных приложений.

2 Материалы и методы

Для разработки аудиоплеера использовался язык программирования Python и библиотека Pygame. Графический интерфейс создавался с помощью встроенных функций Pygame для отрисовки графических элементов

(прямоугольники, текст). Для выбора файлов использовалась библиотека Tkinter. Визуализация эквалайзера реализована в виде имитации, использующей генерацию случайных значений, без анализа реальных частотных характеристик аудиосигнала.

3 Результаты и обсуждения

1. Выбор файла.

Функциональность выбора аудиофайла реализована с помощью библиотеки tkinter и функции `filedialog.askopenfilename()` (рис.1). Эта функция открывает стандартное диалоговое окно выбора файлов операционной системы, позволяя пользователю выбрать аудиофайл (поддерживаются форматы `.mp3`, `.wav`, `.ogg`). Выбранный путь к файлу сохраняется в глобальной переменной `file_path_global`.

```
def choose_file():
    root = tk.Tk()
    root.withdraw()
    file_path = filedialog.askopenfilename(filetypes=[("Audio Files", "*.mp3;*.wav;*.ogg")])
    if file_path:
        global file_path_global
        file_path_global = file_path
        print(f"Выбран файл: {file_path_global}")
```

Рисунок 1. Выбор файла

2. Управление воспроизведением.

Для управления воспроизведением (воспроизведение, пауза, остановка) используются кнопки, размещенные на графическом интерфейсе. Обработка событий нажатия кнопок осуществляется в основном цикле программы. Функции `play_music()`, `pause_music()`, `unpause_music()`, и `stop_music()` используют методы библиотеки `pygame.mixer` для управления воспроизведением аудиофайла (рис.2).

```
def play_music():
    global playing, file_path_global
    try:
        if file_path_global:
            pygame.mixer.music.load(file_path_global)
            pygame.mixer.music.play()
            playing = True
        else:
            print("Пожалуйста, выберите файл.")
    except pygame.error as e:
        print(f"Ошибка при воспроизведении файла: {e}")
    except Exception as e:
        print(f"Произошла ошибка: {e}")

def pause_music():
    global paused, playing
    if playing:
        pygame.mixer.music.pause()
        paused = True
        playing = False
    else:
        print("Музыка не запущена")

def unpause_music():
    global paused, playing
    if paused:
        pygame.mixer.music.unpause()
        paused = False
        playing = True
    else:
        print("Музыка не приостановлена")

def stop_music():
    global playing, paused
    pygame.mixer.music.stop()
    playing = False
    paused = False
```

Рисунок 2. Функции управления воспроизведением

3. Визуализация (Имитация эквалайзера)

Визуализация аудиосигнала реализована в виде имитации эквалайзера. Для упрощения, вместо анализа частотного спектра аудио, используется генерация псевдослучайных значений, которые отображаются в виде движущихся полос. Функция `update_eq_bands()` обновляет значения высоты полос эквалайзера, создавая эффект динамической визуализации (рис.3).

```
def update_eq_bands():
    global eq_bands
    if playing:
        for i in range(num_bands):
            eq_bands[i] = int(random.triangular(0, band_height, band_height/2))
```

Рисунок 3. Эквалайзер

4. Основной цикл программы (рис.4).

Основной цикл программы постоянно работает, пока пользователь не закроет окно. Внутри цикла происходит обработка событий: закрытие окна и нажатия кнопок мыши. При нажатии на кнопку “Choose File” открывается диалоговое окно для выбора аудиофайла. Нажатия на кнопки Play, Pause и Stop управляют воспроизведением музыки с помощью функций `play_music`,

pause_music, unpause_music и stop_music, которые используют возможности библиотеки Pygame для загрузки и воспроизведения аудио.

В каждой итерации цикла происходит обновление имитации эквалайзера с помощью функции update_eq_bands(), которая генерирует случайные значения для высоты полос эквалайзера, создавая эффект движения. Затем экран очищается и отрисовываются полосы эквалайзера и кнопки. Цвет кнопок динамически меняется при наведении курсора и нажатии. Функция pygame.display.flip() обновляет экран, а clock.tick(30) ограничивает частоту кадров до 30 в секунду для плавной работы. В конце работы программы выполняется функция pygame.quit() для корректного завершения Pygame.

```
running = True
clock = pygame.time.Clock()

while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.MOUSEBUTTONDOWN:
            for name, rect in buttons.items():
                if rect.collidepoint(event.pos):
                    if name == "choose":
                        choose_file()
                    elif name == "play":
                        if paused:
                            unpause_music()
                        elif not playing:
                            play_music()
                    elif name == "pause" and playing:
                        pause_music()
                    elif name == "stop" and playing:
                        stop_music()

    update_eq_bands()
    screen.fill(WHITE)
    # Отрисовка эквалайзера
    for i, height in enumerate(eq_bands):
        x = i * band_width + band_margin
        y = eq_y - height
        pygame.draw.rect(screen, BLUE, (x, y, band_width - 2 * band_margin, height))
    # Отрисовка кнопок
    for name, rect in buttons.items():
        color = GRAY
        if rect.collidepoint(pygame.mouse.get_pos()):
            color = LIGHT_GRAY
        if event.type == pygame.MOUSEBUTTONDOWN and rect.collidepoint(event.pos):
            color = BLUE
        pygame.draw.rect(screen, color, rect)
        text = font.render(name.capitalize(), True, BLACK)
        screen.blit(text, (rect.centerx - text.get_width() // 2, rect.centery - text.get_height() // 2))
    pygame.display.flip()
    clock.tick(30)
pygame.quit()
```

Рисунок 4. Основной цикл программы

5. Внешний вид системы (рис.5)

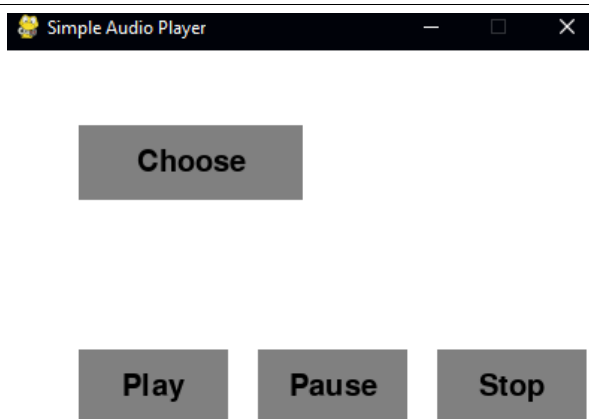


Рисунок 5. Внешний вид

Сначала нужно выбрать файл для воспроизведения нажав кнопку «Choose» (рис.6).

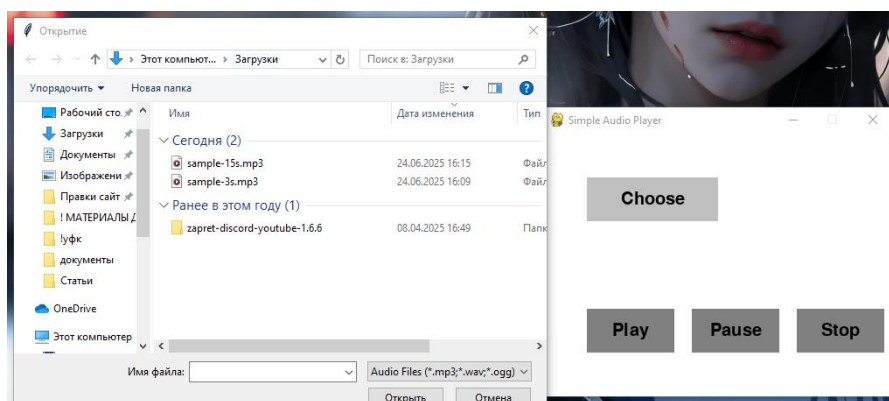


Рисунок 6. Выбор файла

После чего нажимаем кнопку «Play» и аудио начинает воспроизводиться.

4 Выводы

В ходе проведенного исследования была разработана и реализована программа — интерактивный аудиоплеер с элементами визуализации, основанный на библиотеке Pygame. Разработка подтвердила удобство и эффективность использования Pygame для создания интерактивных мультимедийных приложений с относительно простым графическим интерфейсом. Реализованный плеер позволяет пользователю выбирать аудиофайлы, управлять воспроизведением (воспроизведение, пауза, остановка) и наблюдать визуальное отображение активности аудиосигнала в виде имитации эквалайзера.

Несмотря на то, что визуализация эквалайзера в данном проекте носит имитационный характер и не отражает реальные частотные характеристики аудиосигнала, она успешно выполняет функцию визуальной обратной связи, повышая привлекательность и интерактивность приложения. Полученный

опыт может быть использован для дальнейшей разработки более сложных аудиоплееров с расширенным функционалом и более точной визуализацией, в том числе с применением специализированных библиотек для анализа аудиоданных. Проект демонстрирует потенциал Pygame как доступного и мощного инструмента для создания интерактивных мультимедийных приложений. Дальнейшие исследования могут быть направлены на интеграцию алгоритмов анализа частот и улучшение качества визуализации, а также добавление таких функций, как регулировка громкости, создание плейлистов и другие.

Библиографический список

1. Матвеева А. С. Реализация паттерна поведения «Команда» в графическом интерфейсе пользователя на Python // Постулат. 2025. №1.
2. Вохидов А.М., Вохидов Д.А., Фармонова Р.Ф., Хафизова Д.Ш. Разработка графическим пользовательским интерфейсом программ в пакете tkinter с использованием современных педагогических технологий в области медицины // Роль информационных технологий в управлении ресурсами персонала здравоохранения 2022. Т. 34. С. 181-184
3. Aligulovich D.A., Mahmudova Z.I., Sayfullayev R.F. Tibbiyot yo'nalishida zamonaviy pedagogik texnologiyalarini qo'llab tkinter paketida gui dasturlarini tuzish // Евразийский журнал математической теории и компьютерных наук. 2022. №2(12). С. 31-35.
4. Anjaneyulu P., Priyanka D., Manaswini Y. Reportlab Based Transcription Transformer // Computing & Intelligent Systems, SCRS, India. 2025. С. 85-96.
5. Moiz M. Workflow: Using A Python Script to Capture and Summarize Instagram Feeds at Rauner Special Collections // Rauner Special Collections Digital Archives Practice. 2024. С. 1-8.