

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет инфокоммуникационных технологий

Образовательная программа Мобильные и сетевые технологии

Направление подготовки 09.03.03 Прикладная информатика

О Т Ч Е Т

о производственной, технологической (проектно-технологической)
практике

Тема задания: Автоматизация озвучивания вопросов в игре "Что? Где? Когда?"

Обучающийся: Шугинин Ю. А., группа К34402

Руководитель практики от университета: Коцюба И. Ю., доцент ФИКТ
Университета ИТМО

Практика пройдена с оценкой ____

Дата ____

Санкт-Петербург, 2022

РЕФЕРАТ

В ходе практики разрабатывался алгоритм автоматизации озвучивания вопросов ведущим при проведении игры «Что? Где? Когда?». Таким образом решалась проблема дефицита людей, обладающих навыками ведущего игры «Что? Где? Когда?» и желающих исполнять эту роль.

Цель практики – программная автоматизация озвучивания вопросов в игре «Что? Где? Когда?».

Для достижения цели были решены следующие задачи:

- формализация специфики чтения вопросов в игре "Что? Где? Когда?",
- изучение средств программного преобразования текста в речь,
- сравнительный анализ найденных инструментов и выбор наиболее подходящего для достижения поставленной цели,
- программная реализация алгоритма озвучивания вопроса,
- разработка утилиты для игры с озвученным пакетом вопросов,
- интеграция разработанного алгоритма в созданный ранее Telegram-бот для игры в "Что? Где? Когда?".

В ходе разработки применялись следующие технологии: язык программирования Python 3.10, а также подключаемые библиотеки pdfplumber, gtts, pathlib, tempfile, playsound, re, time.

Результатом практики стала реализация алгоритма озвучивания пакета вопросов игры «Что? Где? Когда?», который впоследствии был интегрирован в разработанную на его основе утилиту для проведения игры, а также в разработанный в рамках предыдущей практики Telegram-бот.

Отчёт содержит 16 страниц, 3 рисунка и 7 интернет-источников.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Технологические основы.....	5
1.1 Синтаксический анализ пакета вопросов	5
1.2 Преобразование текста в речь.....	6
1.3 Воспроизведение аудиофайла.....	7
2 Реализация.....	9
2.1 Формализация требований к чтению	9
2.2 Определение функционала.....	9
2.3 Архитектура программ	10
3 Полученный результат.....	12
3.1 Пример работы утилиты.....	12
3.2 Пример работы бота.....	13
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16

ВВЕДЕНИЕ

Спортивное «Что? Где? Когда?» – соревновательная версия известной телевизионной игры, по которой проводится множество турниров различной сложности, доступных любому заинтересованному коллективу из 6 человек. В Университете ИТМО есть студенческий клуб интеллектуальных игр «ИТМО.INTELLECT», в котором еженедельно проводятся отыгрыши турниров по «Что? Где? Когда?», в которых зачастую принимают участие не только студенты ИТМО, но и команды из других вузов Санкт-Петербурга.

Руководители клуба ещё не преодолели возрастной порог участия в студенческих турнирах, поэтому они сами заинтересованы в том, чтобы быть участниками проводимых игр в составе своих команд. Часто приходится на роль ведущего приглашать людей, которые сами никогда не играли в «Что? Где? Когда?» и очень отдалённо знакомы с игрой.

Специфика проведения игры заключается в том, что большинство игроков привыкли записывать вопросы во время их озвучивания. Поэтому ведущему недостаточно просто красиво прочитать вопросы в микрофон. Во-первых, это нужно делать медленнее, чем в обычной жизни. Во-вторых, необходимо делать достаточно продолжительные паузы между предложениями и чуть менее продолжительные – в районе знаков препинания в середине предложений. На практике оказывается, что у сторонних обывателей не получается соблюдать все требования и неопытный человек рефлексивно возвращается к привычному темпу чтения, даже если ему предварительно всё подробно объяснили. К сожалению, это очень сильно портит впечатление от игры и влияет на результат даже опытных команд.

Помимо этого, в рамках предыдущей практики был разработан Telegram-бот для игры в «Что? Где? Когда?». В нём вопросы представляются в формате текста. При этом ботом может пользоваться человек, привыкший воспринимать вопросы на слух. Либо может оказаться так, что бот используется для тренировки командой, в которой никто не хочет быть ведущим.

Таким образом, в ходе практики разрабатывался алгоритм автоматизации озвучивания вопросов. Актуальность индивидуального задания обеспечена, во-первых, решением проблемы студенческого клуба интеллектуальных игр «ITMO.INTELLECT», а во-вторых, нехваткой функционала озвучивания вопросов в разработанном ранее боте.

Цель практики – программная автоматизация озвучивания вопросов в игре «Что? Где? Когда?».

Для достижения цели были решены следующие задачи:

- формализация специфики чтения вопросов в игре "Что? Где? Когда?",
- изучение средств программного преобразования текста в речь,
- сравнительный анализ найденных инструментов и выбор наиболее подходящего для достижения поставленной цели,
- осуществление программной реализации озвучивания вопроса,
- разработка утилиты для озвучивания пакета вопросов игры,
- интеграция разработанного алгоритма в созданный ранее Telegram-бот для игры в "Что? Где? Когда?",
- тестирование разработанных программных средств.

1 Технологические основы

1.1 Синтаксический анализ пакета вопросов

Существует так называемый сайт сообщества знатоков (<https://rating.chgk.info/>). На нём анонсируются турниры и ведётся учёт участия игроков и команд в тех или иных соревнованиях. Также на сайте присутствует функционал регистрации площадки отыгрыша турнира. С его помощью в назначенный день пользователь, изъявивший желание провести игру в своём городе, получает пакет вопросов от организаторов.

Пакет вопросов на игру представляет собой PDF-файл, имеющий унифицированную структуру: преамбула (вводные слова, благодарности редакторам турнира и пожелания участникам), список редакторов турнира, вопросы (в формате: номер вопроса, текст вопроса, пустая строка, ответ, комментарий, источник, автор). Исходя из этого можно написать универсальный алгоритм получения исключительно текстов вопросов из данного файла.

В языке программирования Python есть встроенная функция `open()` для открытия файла внутри программы на чтение или на запись. Однако в случае с PDF-файлом этого недостаточно, потому что данный формат не является текстовым. Для получения текста из PDF-файла существует библиотека `pdfplumber`, в которой реализован метод `extract_text()` для постраничного извлечения текста из объекта открытого PDF-файла в строковом формате.

1.2 Преобразование текста в речь

В результате исследования предметной области были найдены две актуальные Python-библиотеки, которые осуществляют преобразование строковых данных в аудиоформат: `gtts` и `pyttsx3`, принципиально отличающиеся друг от друга.

Модуль `gtts` – библиотека Python и инструмент командной строки для взаимодействия с API Google Translate для преобразования текста в речь. Записывает озвученные mp3-данные файлоподобный объект (строку байтов) для дальнейшей обработки звука. [1] Т. е. преобразование текста происходит в результате обращения к удалённым серверам Google Translate, а значит, работа библиотеки требует подключения к интернету. Также стоит отметить, что модуль возвращает лишь файлоподобный объект, у которого есть метод `save()` для сохранения получившейся записи в mp3-файл, но нет функционала обращения к аудиосистеме устройства и воспроизведения содержимого получившегося объекта. Есть возможности выбора языка, в том числе русского, и регулирования темпа речи (2 режима: обычный и медленный).

Модуль `pyttsx3` – это также библиотека преобразования текста в речь на Python. В отличие от предыдущего модуля, этот работает в автономном режиме. [2] При работе с данным модулем создаётся так называемый «движок» – объект класса `pyttsx3`, имеющий широкий спектр методов: регулирование скорости и громкости воспроизведения; переключение между мужским и женским голосом. Немаловажно, что есть возможность воспроизведения озвученного текста без необходимости предварительного сохранения в файл.

С точки зрения предоставляемого функционала для решения моей задачи лучше подходила библиотека `pyttsx3`. Однако, когда был реализован минимальный жизнеспособный образец будущего алгоритма, оказалось, что модуль совсем не подходит для использования с текстом на русском языке, несмотря на заявленную поддержку. Запись получается прерывистая, без какой-либо интонации. Создаётся впечатление, что с тобой общается робот очень плохого качества. В случае с модулем `gtts`, напротив, получается нормальная для восприятия запись: с правильной интонацией, паузами в нужных местах и отсутствием фонетических ошибок. На мой взгляд, ключевую роль здесь играет взаимодействие с API Google Translate, где наверняка применяется машинное обучение для приближения качества синтезированной речи к человеческой. В результате сравнения двух минимально жизнеспособных образцов программы был сделан однозначный вывод о том, что нужно продолжать работу исключительно с модулем `gtts`.

1.3 Воспроизведение аудиофайла

Для воспроизведения аудиофайла в процессе работы программы на Python существует несколько одностипных библиотек: `pydub`, `playsound`, `tksnack`, `simpleaudio`. [3] Принцип работы у них схожий: нужно передать методу путь аудиофайлу в файловой системе устройства. Отличаются, в основном, поддерживаемыми аудиоформатами. Был выбран модуль `playsound`.

На данном этапе единственной проблемой оставалась необходимость сохранять в системе устройства `mp3`-файл каждого вопроса, а после

воспроизведения, соответственно, удалять – либо руками, либо в коде. И то, и другое не очень рационально.

Оказалось, что существует библиотека `tempfile`, предоставляющая функционал создания так называемого временного файла. [4] То есть библиотека при обращении к ней автоматически создаёт временный файл в своей папке (или в папке, указанной пользователем), а после его закрытия в программе – удаляет. В моём случае лучше всего подходит метод `NamedTemporaryFile()` [5], потому что функция `playsound` требует передачи пути к файлу, в том числе его названия.

Таким образом, условный алгоритм озвучивания вопроса получается таким: с помощью модуля `gtts` преобразуем строковую переменную в файлоподобный объект с результатом озвучивания, затем сохраняем его в предварительно созданный с помощью библиотеки `tempfile` временный файл и, наконец, воспроизводим получившийся звук из временного файла с помощью библиотеки `playsound`.

2 Реализация

2.1 Формализация требований к чтению

Я уже упомянул, что при проведении игры «Что? Где? Когда?» нужно учитывать тот факт, что игроки записывают вопросы. Для этого необходимо делать паузы между предложениями и в месте знаков препинания. На основании собственного игрового опыта были установлены следующие значения для данных параметров: 6 секунд для паузы между предложениями и 3 секунды – в месте знаков препинания.

У изученных библиотек озвучивания текстов нет подобного функционала. Остаётся разбивать вопрос на части на этапе синтаксического анализа и воспроизводить каждую часть отдельно, останавливая выполнение программы на указанный временной промежуток в зависимости от того, какой знак препинания на конце.

Разбивать вопрос на части можно с помощью модуля `re`, используемый для работы с регулярными выражениями. У него есть метод `split()`, возвращающий список частей строки, разбитой согласно символам, указанным в переданном регулярном выражении. В моём случае это знаки препинания. За принудительные паузы в работе программы на языке Python отвечает встроенный модуль `time`.

2.2 Определение функционала

В результате анализа требований и доступных программных средств был установлен следующий набор опций для реализации:

- получение и обработка пути к файлу с пакетом вопросов,
- получение и обработка количества вопросов в пакете,

- проверка существования PDF-файла по указанному пути,
- воспроизведение вопросов. Вопросы произносятся по одному с учётом установленных требований к паузам в чтении. При этом чтение каждого следующего вопроса происходит по команде пользователя,
- возможность повторения уже воспроизведённого вопроса.

Для интеграции алгоритма озвучивания вопросов в разработанный в рамках предыдущей практики Telegram-бот для игры в «Что? Где? Когда?» актуальны только требования осуществления звукового воспроизведения вопросов, а также возможности повторения уже озвученного вопроса. Алгоритмы получения вопросов из Базы в нужном формате там уже были реализованы. А паузы между предложениями не так актуальны, потому что бот не предназначен для проведения полноценных публичных игр (достаточно пауз, встроенных по умолчанию в ответах Google Translate API).

2.3 Архитектура программ

Получившаяся утилита имеет следующую структуру:

- подключение сторонних модулей. В первой главе подробно описано использование библиотек pdfplumber, gtts, tempfile, playsound. Также был использован модуль pathlib для проверки существования PDF-файла по переданному пути; модуль time, для остановки выполнения программы во время требуемых пауз, и модуль re для работы с регулярными выражениями при разбиении вопроса на части по знакам препинания [6],
- функции, описанные далее.

main() -> None: точка входа в программу. В ней происходит считывание пути к файлу с вопросами, проверка существования такого PDF-файла, считывание количества вопросов, а также последовательный вызов остальных функций.

`parse_questions(file_path: str, quest_num: int) -> list[list[str]]`: принимает путь к файлу с пакетом вопросов и их количество; возвращает список вопросов, при этом каждый вопрос представлен списком строк, исходя из разбиения по знакам препинания.

`read(question: str) -> None`: принимает строку для озвучивания, запускает воспроизведение, а также останавливает работу программы на 3 или на 6 секунд в зависимости от того, какой знак препинания на конце (точка, восклицательный знак, вопросительный знак – конец предложения, 6 секунд; остальные – середина предложения, 3 секунды).

`game(questions: list) -> None`: принимает список вопросов и для каждого вызывает функцию `read()`. Также запрашивает команду для перехода к следующему вопросу или повторения уже воспроизведённого.

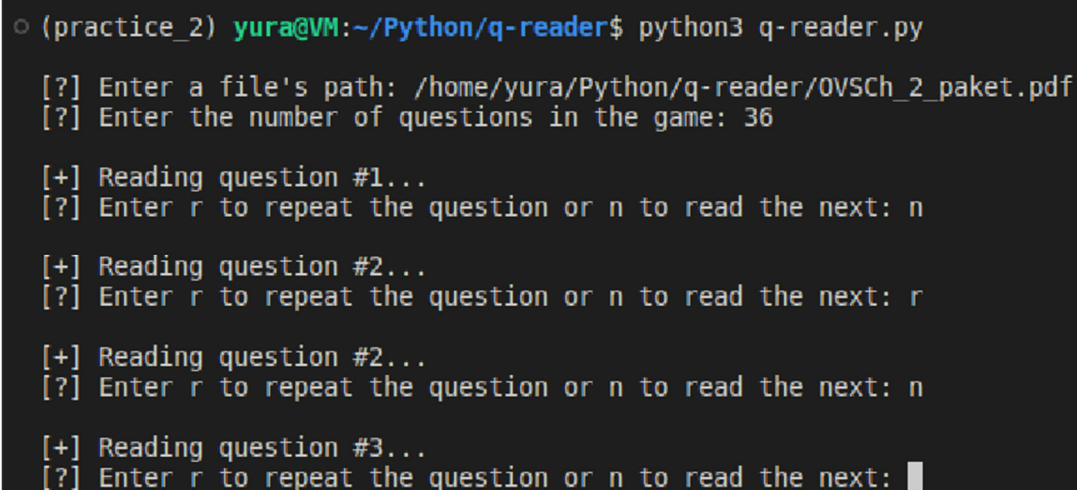
Для интеграции алгоритма озвучивания вопросов в разработанный ранее Telegram-бот для игры в «Что? Где? Когда?» актуальна функция чтения вопроса. Это реализуется через вызов метода `send_audio()` у объекта класса бота из библиотеки `pyTelegramBotApi` [7]. Также необходимо было добавить обработку дополнительного вопроса перед началом игровой сессии: в каком формате пользователь хочет получать вопросы (текст или аудио). Варианта ответа у пользователя всего два, поэтому рационально предоставить возможность сделать свой выбор с помощью нажатия на одну из двух появившихся кнопок, а не через ввод ответа на клавиатуре.

3 Полученный результат

3.1 Пример работы утилиты

Я разработал утилиту, в которую интегрировал созданный алгоритм воспроизведения вопросов игры «Что? Где? Когда?». Утилита покрывает полный цикл проведения игры.

Для демонстрационного примера (рисунок 1) использовался пакет вопросов второго тура Открытого Всероссийского Синхронного Чемпионата (ОВСЧ) 2022/2023.



```
o (practice_2) yura@VM:~/Python/q-reader$ python3 q-reader.py
[?] Enter a file's path: /home/yura/Python/q-reader/OVSh_2_paket.pdf
[?] Enter the number of questions in the game: 36

[+] Reading question #1...
[?] Enter r to repeat the question or n to read the next: n

[+] Reading question #2...
[?] Enter r to repeat the question or n to read the next: r

[+] Reading question #2...
[?] Enter r to repeat the question or n to read the next: n

[+] Reading question #3...
[?] Enter r to repeat the question or n to read the next: █
```

Рисунок 1 – Демонстрация работы утилиты

После запуска утилита предлагает ввести путь к файлу с пакетом вопросов; затем спрашивает, сколько вопросов в пакете. Затем начинается поочерёдное воспроизведение вопросов. Переход к следующему вопросу происходит при нажатии клавиши n (next). Повторение воспроизведённого вопроса происходит при нажатии клавиши r (repeat).

Код разработанной утилиты доступен в моём репозитории на GitHub (<https://github.com/YuriiShuginin/q-reader>).

3.2 Пример работы бота

Интеграция алгоритма озвучивания вопросов в разработанный в рамках предыдущей практики бот для игры в «Что? Где? Когда?» заключалась в добавлении возможности выбрать в начале игровой сессии, в каком формате получать вопрос. Для ответа на данный вопрос пользователь имеет возможность нажать на одну из двух кнопок (рисунок 2).

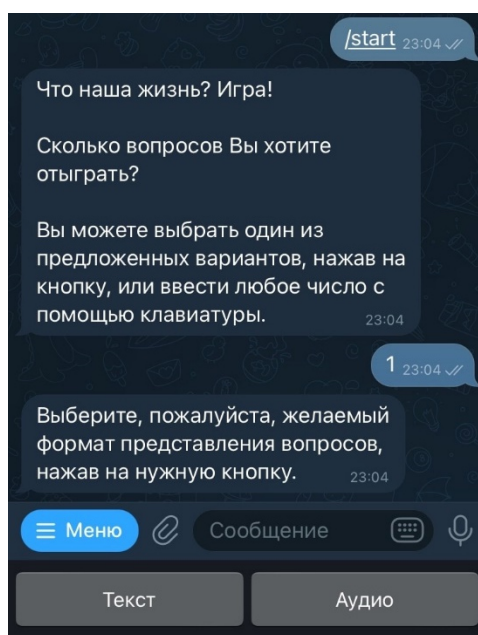


Рисунок 2 – Демонстрация выбора формата вопросов

Бот сохраняет этот выбор на протяжении всей игровой сессии. В случае выбора аудиоформата, во время получения очередного вопроса пользователю приходит не текст вопроса, а аудиофайл, доступный для воспроизведения (рисунок 3). При необходимости пользователь может запустить звуковой файл повторно и прослушать вопрос ещё раз.

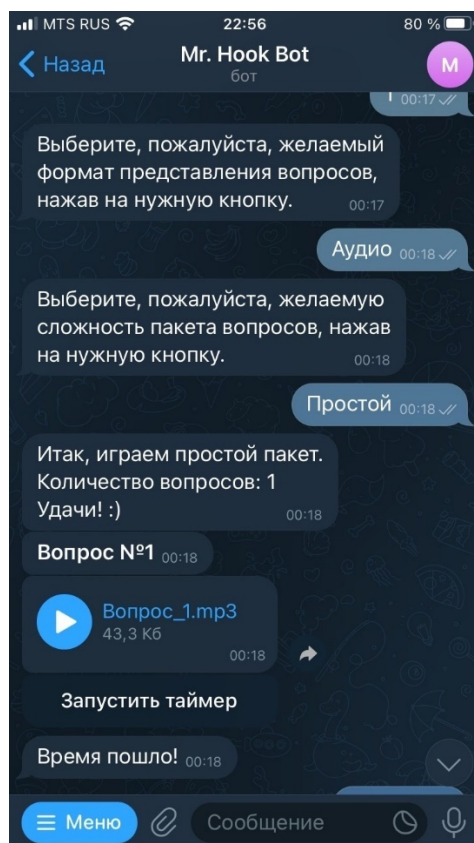


Рисунок 3 – Получение вопроса в аудиоформате

Код разработанного бота с добавленным функционалом озвучивания вопросов доступен в моём репозитории на GitHub (<https://github.com/YuriiShuginin/MrHookBot>).

ЗАКЛЮЧЕНИЕ

В ходе практики были получены практические навыки реализации алгоритма преобразования текста в речь с использованием языка программирования Python. Освоена работа с библиотеками `gtts`, `pyttsx3`, `tempfile`, `playsound` и `pdfplumber`, которые являются важными инструментами для решения задач в своих областях. Полученные навыки программной реализации преобразования текста в речь могут быть актуальны, например, в разработке голосовых ассистентов.

Помимо этого, был разработан скрипт в формате консольной утилиты, решающей прикладную задачу проведения полного цикла игры «Что? Где? Когда?» с автоматизированным озвучиванием вопросов с учётом специфики их чтения на серьёзных турнирах.

Также были получены навыки добавления нового функционала озвучивания вопросов в относительно большой проект Telegram-бота без ущерба уже реализованным в нём функциям.

В результате прохождения практики решены все поставленные задачи и достигнута цель программной автоматизации озвучивания вопросов в игре «Что? Где? Когда?».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация библиотеки `gtts` на английском языке [Электронный ресурс]. URL: <https://gtts.readthedocs.io/en/latest/index.html> (дата обращения: 12.11.2022).
2. Документация библиотеки `pyttsx3` на английском языке [Электронный ресурс]. URL: <https://pyttsx3.readthedocs.io/en/latest/> (дата обращения: 14.11.2022).
3. Англоязычная статья «Play sound in Python» [Электронный ресурс]. URL: <https://www.geeksforgeeks.org/play-sound-in-python/> (дата обращения: 17.11.2022).
4. Статья «Работа с временным файлом в Python» [Электронный ресурс]. URL: <https://andreyex.ru/programmirovanie/python/rabota-s-vremennym-fajlom-v-python/> (дата обращения: 24.11.2022).
5. Документация библиотеки `tempfile` на английском языке [Электронный ресурс]. URL: <https://docs.python.org/3/library/tempfile.html> (дата обращения: 24.11.2022).
6. Англоязычная статья «Split a string on punctuation marks in Python» [Электронный ресурс]. URL: <https://bobbyhadz.com/blog/python-split-string-on-punctuation> (дата обращения: 30.11.2022).
7. Официальная документация по работе с Telegram Bot API. [Электронный ресурс]. URL: <https://core.telegram.org/bots/api> (дата обращения: 08.12.2022).