



Made by
Meshcherakou
Egor

Представляем beta version ГОЛОСОВОГО ассистента **TalkFlow**

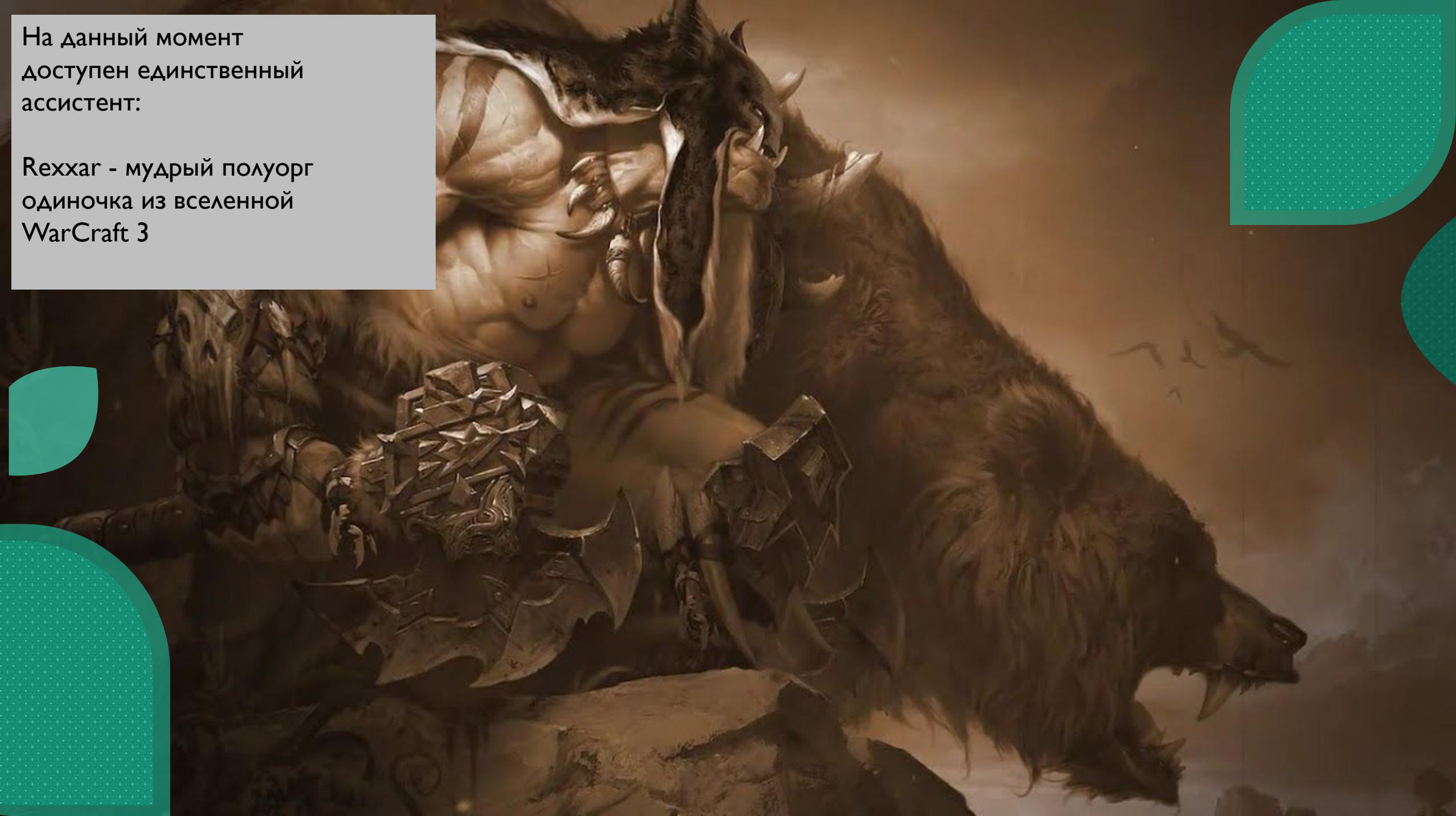
- Данное приложение создано по мотивам мечты личного ассистента компаньона в компьютерном мире с использованием ИИ

На данной стадии разработки ассистента реализованы следующие возможности:

- Общение с chat gpt
- Озвученные ответы
- Выполнение хоткеев по голосовым командам
- Простая система добавления новых команд
- Оффлайн работа по хоткеям

На данный момент
доступен единственный
ассистент:

Реххар - мудрый полуорг
одиночка из вселенной
WarCraft 3



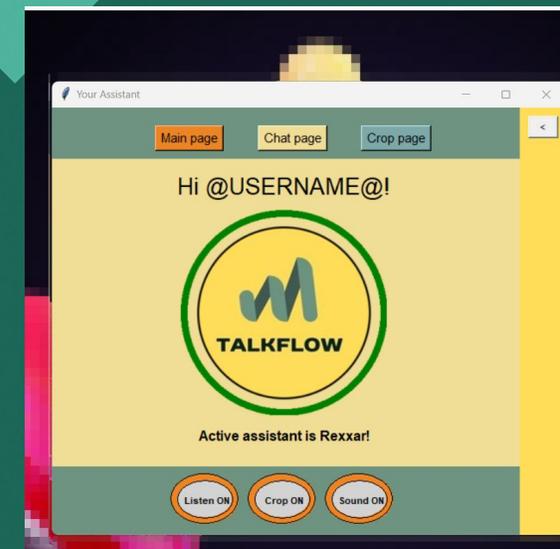
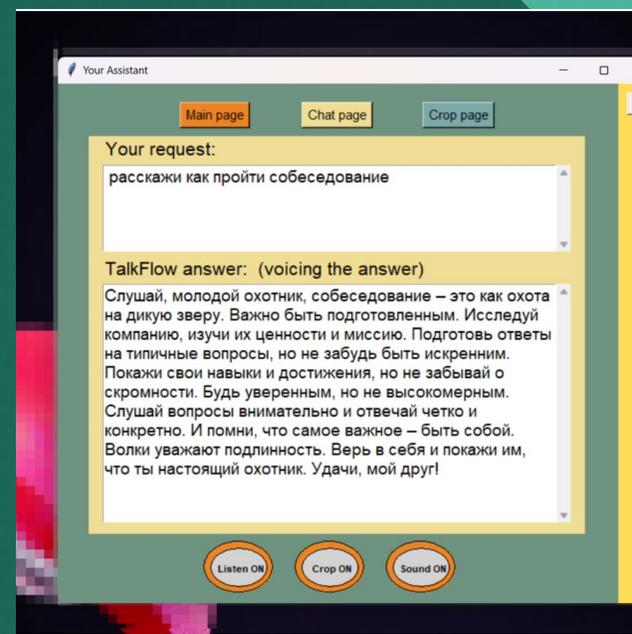
Реализация UX

Интерфейс был сотворен на python с помощью Tkinter

Имеется 2 активных (на данный момент) окна:

- Главное меню
- Чат

На каждом сущетсвуют активные элементы управления голосовым помощником, а так же опознавательные знаки, по которым можно понять чем именно занимается ассистент

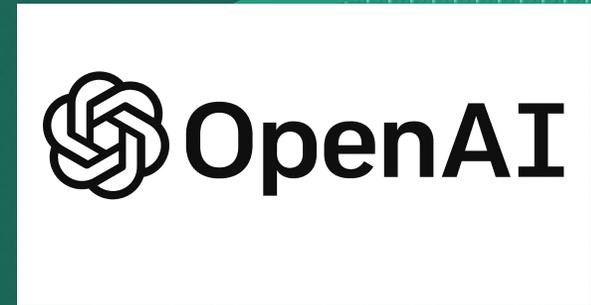


Picovoice/
porcupine



On-device wake word detection powered by deep learning

34 Contributors 22 Used by 3k Stars 443 Forks



Backend

- Реализует технологии

oobabooga/text-generation-webui

#956 **Advanced voice control for Silero TTS**



6 comments

KirillRepinArt opened on April 9, 2023



AutoHotkey

freshtorrent.ru

alphacep/**vosk**



VOSK Speech Recognition Toolkit

1 Contributor 3 Issues 332 Stars 40 Forks

Установка и запуск

Установка:

- Для установки достаточно скопировать репозиторий проекта, и установить библиотеки из requirements.txt
- На линукс может понадобиться установка portaudio19-dev и tkinter, а так же предоставить доступ к микрофону и динамикам
- По умолчанию в main.py используется модель для работы на линукс, при запуске с windows модель необходимо поменять на вторую, что лежит в папке
-

Запуск:

Из Assistens_manager

1. Предварительно запустить .\FindThePath.py для работы хоткеев (выполнять одинажды)
2. Запускать приложение через testing.py

Взаимодействие

- После экрана загрузки и приветственной фразы можно начинать работу с ассистентом
- - Для начала говорим кодовое слово *Ассистент*
 - Для этого используется библиотека porcupine которая очень быстро обрабатывает голос и реагирует только на кодовое слово
 - После чего в течении 5 секунд он будет нас слушать и мы можем дать ему команду или спросить о чем либо
 -
 -

(время вопроса можно изменить в main.py: self.delay - время в секундах)



Создание новых команд

В файле `commands.yaml` указаны все команды по следующей схеме, и для добавления новой команды достаточно прописать параметры и закинуть `.exe` файл

-
- `File` - `.exe` файл который будет выполняться по обнаружению слов похожих на триггерные
- `Command` - команда на питоне прописанная в `main.py` которая будет выполняться
- `Sound` - звуки, один из которых будет случайно воспроизведен при выполнении команды (находятся в папке `./sound/Rexxar`), для создания новых команд заготовлено 8 различных `ok.wav` звуков
- `Triggers` - слова, схожесть с которыми будет проверяться по алгоритму Левенштейна
-

```
open_browser:  
  file: "Run browser.exe"  
  command: dosmthing  
  sound:  
    - browser3.wav  
    - browser1.wav  
    - browser2.wav  
  triggers:  
    - открой браузер  
    - запусти браузер  
    - открой гугл хром  
    - гугл хром
```

Autohotkey и .exe

- К сожалению автохоткеи работают только на Windows, однако уже изучаются аналоги для линукс и к началу сентября базовые хоткеи будут работать и на линукс

Пока хоткеи умеют:

- Работать со спотифай
- Включать браузер
- А так же ютуб
-

Однако при желании не сложно сделать хоткей для любого процесса

исходники скриптов автохоткеев можно найти в папке `custom_commands`, по ним были собраны exe файлы

Планы по развитию

- Приложение предварительно разделено на front и back, таким образом предстоящая смена интерфейса на react не будет большой сложностью
-
- Использовать `countvectorizer` из `sklearn` для ускоренного и упрощенного опознания команд
-
- Добавить полноценную озвучку персонажами для уникальных ответов, для ускорения процесса и лучшего погружения
-
- Создать простой конструктор команд с графическим интерфейсом
-
- Создать сервер, который будет работать с внешними сервисами, что бы облегчить работу с ключами и Api, что так же значительно облегчит приложение



The background is a solid teal color. It features several large, overlapping abstract shapes in various shades of teal and light green. Some shapes have a fine, dotted pattern, while others are solid. The shapes include rounded rectangles, circles, and organic, leaf-like forms. The overall composition is modern and clean.

Буду рады
любому FeedBack-у