

Walkthrough

I) .env (from example.env)

- 2: OpenAI_KEY: nhập openAI key ở đây để transcribe audio (sử dụng whisper)
- 5-8: nhập elevenlab key, tên của voice mà bạn đặt tên ở eleven lab (nếu bạn sử dụng eleven lab), voice mình đề xuất là young female, australian.
- 11-12: url của voicevox không cần sửa, với voicevox_log, set là True nếu các bạn muốn thấy phần log của voicevox
- 16-18: characterAI_key có thể để trống, ở dòng 18 các bạn vào characterAI chọn 1 nhân vật mình muốn, ví dụ https://beta.character.ai/chat?char=oL2IzOD15_wBIP_o6NAWDwiVyAnzz_3aGLu9aU7i254, các bạn lấy phần string sau char= dán vào, ở đây là oL2IzOD15_wBIP_o6NAWDwiVyAnzz_3aGLu9aU7i254
- 20: Nếu sử dụng elevenlab thì có thể set translate to JP là false
- 23: Voice: set là voicevox hoặc elevenlabs
- 25-26: Vì voicevox chỉ nói được tiếng nhật nên tts_jp là true, tts_en là false, nếu bạn xài elevenlabs thì set ngược lại
- 29-30: vtube studio: không cần sửa
- 33: torch device: với các bạn không có nvidia thì set là cpu

II) ./waifu/pyvenv.cfg:

- 1: set python path, vào cmd nhập where python
- 2: có xài site package của system không, các bạn không cần sửa
- 3: version python hiện tại của các bạn, vào cmd và nhập python --version

III) Start.bat:

- Đây là file các bạn chạy để khởi động chương trình
- 1: @echo off để turn off echo của command, tức là command không display lên console khi script được thực thi
- 3: This is a comment line
- 4-9: set codepage to 65001, tức là utf-8
- 11: set biến môi trường virtual_env = ./waifu
- 38: Vào thư mục character_ai và đọc file package.json để cài dependencies
- pyaudio không cài bằng pip thông thường được nên phải cài pipwin
- cls: clear console screen
- execute file runner.py để bắt đầu

IV) utils/dependencies.py:

- 4: chỉ ra directory của voicevox
- 12: link của ffmpeg
- 13: link của voicevox
- 16: gán VOICE = với giá trị của voice trong environment variables
- Hàm download khi có URL
- 40: hàm check , nếu voicevox chưa tồn tại thì down về, đồng thời tải ffmpeg và ffmpegprobe cho elevenlabs
- 65-72: check xem voice, torch_device, oai key, characterai_id được set chưa

V) utils/hotkeys.py:

- 6-7: Hàm này trả về True nếu right ctrl và right shift được nhấn cùng lúc
- 9-11: Hàm này chờ đến khi audio_input() trả về True, khi đó nó sẽ trả về True

VI) utils/audio.py:

- Audiosegment: allows working with audio files
- Wave: allows working with WAV audio files
- Pyaudio: Provides functionalities for audio input/output
- chunk ở đây là chunk size và rate là sample rate
- 20-22: Hàm này load audio data từ path dưới dạng AudioSegment và truyền vào hàm play_mp3_memory
- 25-49: Hàm này play MP3 audio file từ memory
- Tương tự ta có hàm play wav file
- 84: Hàm record(): Khi giữ 2 hotkey (trong file hotkeys) thì audio data được read in chunk và append vào frame, dừng khi không còn giữ nữa. Sau đó mình lưu audio vào file recording.wav và return

VII) utils/vtube_studio.py:

- **Control mouth movements**
- Hàm start:
 - Đợi đến khi kết nối với Vtube studio thành công
 - 128-139: check xem Voice_Level có thay đổi có thay đổi không, nếu có thì gửi request tới Vtube studio để update voice parameter
- 147: Hàm start real: là cầu nối giữa asynchronous code và main program flow, nó chạy hàm start trong event loop
- 152: Hàm run_async: Tạo 1 thread mới với start_real là hàm mục tiêu . Daemon true nghĩa là thread sẽ chạy ở background và không block main program from exiting.

VIII) utils/transcriber.py:

- 6: set voicevox_url

- 9: hàm transcribe: transcribe sử dụng whisper và return text
- 21: hàm speak_jp: Sử dụng voicevox api để generate speech in Japanese và play
 - request.content chứa audio data response
 - chuyển đổi qua dạng bytesIO (in-memory file-like object) và truyền vào hàm play_wav để play audio

IX) utils/characterAI.py:

- 21: Hàm run_websocket: Tạo một kết nối websocket đến local characterAI server, khi websocket đóng thì gọi hàm on_close, khi nhận được message thì gọi hàm on_message
- 28: send message thông qua websocket
- 33: send message thông qua standard input
- 48: Hàm handle output: Handle output (stderr hoặc stdout) of CharacterAI process asynchronously. Nó liên tục đọc từ stream và print nó ra
- 55: Hàm start: Tạo 1 subprocess để chạy characterAI process
 - Nếu characterai_log là True thì nó create task để xử lí stdout và stderr output
- 82: Hàm run_async: Tạo 1 event loop mới và set nó là event loop hiện tại, cuối cùng là bắt đầu 1 thread để chạy hàm start, các bạn có thể search event loop in asynchronous programming để tìm hiểu thêm

X) utils/translator.py:

- Perform translation using Google Translate service
- 3: Tạo 1 instance của Translator class
- 5: Hàm translate text to japan

XI) utils/speech.py:

- File này được tạo ra để trường hợp voice là voicevox và option text to speech là english (vì như mình đã nói voicevox chỉ có thể phát âm tiếng nhật
- 3: device sử dụng là gì, mặc định là cpu
- 7: Hàm prepare, check xem cwd có [model.pt](#) không, nếu không thì tải về
- 14: Hàm silero_tts: model.save_wav được call để generate audio waveform từ input message, hàm này sau đó trả về đường dẫn đến audio path

XII) utils/punctuation_fixer.py:

- Chuyển dấu .. thành ',' và chuyển vài dấu khác như ')', '!', '*', '_' .

XIII) utils/elevenlabs.py:

- Import elevenlabslib
- Dựa vào key để khởi tạo 1 elevenlabsusers
- Lấy tên voice đã được khai báo trong .env để getvoice, nếu không có thì tạo 1 voice female, american, young

- 19: Hàm speak sẽ generate voice dưới dạng mp3 từ message sau đó gọi hàm play_mp3 để speak

XIV) utils/multi_thread.py:

- Hàm run_in_new_thread: tạo một thread mới và thực thi hàm target

XIV) utils/voicevox.py:

- Hàm start: chạy subprocess (./run.exe của voicevox) , xử lý các stdout và stderr của subprocess
- Hàm run_async: chạy hàm start asynchronously ở trong thread mới

XV) Runner.py:

- 6-7: check tại wd nếu tồn tại example.env mà không tồn tại .env thì đổi tên example.env thành .env
- 10-16: mở file .env , với mỗi file thì gọi hàm strip (loại bỏ leading hoặc trailing whitespace character),
 - tiếp theo check xem line này có empty hay bắt đầu với kí tự # hay không, nếu phải thì continue vòng lặp
 - 15: split line thành 2 phần dựa vào kí tự '=' , số 1 để giới hạn việc split ở dấu = đầu tiên
 - 16: gán giá trị value cho biến môi trường key
- 20-23: set openai.api_key = key mình đã cung cấp ở .env
- 26-35: check xem có trong virtual environment không
- 43: Khi program nhận SIGINT signal, thì hàm signal handler được gọi
- 45-53: import các file đã được giải thích ở trên
- 55: gán giá trị của biến môi trường VOICE cho voice
- 57: Gọi hàm check của dependencies
- 59-63: Nếu là elevenlabs thì import elevenlabss, voicevox thì import voicevox rồi chạy hàm run_async
- 72: Nếu enable thì kết nối vtube studio
- 78: Tạo 1 semaphore với giá trị 0
- 85: Hàm character replied: Đưa message để nhận được speech
 - 97-108: xử lý speak audio với voicevox hoặc elevenlabs
 - 111: set mouth to rest point (audio level=0)
 - 113: release semaphore (semaphore này acquire trong vòng lặp, việc có semaphore giúp cho reply được xử lí hết trước khi synthesis speech cho 1 reply mới)
- 117: Vòng lặp chờ user nhấn 2 phím hotkeys sau đó lắng nghe audio và gửi message đến characterAI sau đó nhận response và thực hiện text to speech

THIS IS THE END, TO START PROGRAM, EXECUTE FILE START.BAT