Avatar AI Assistant:

The Project Manual: (Till 1st Month progress)

1. Environment Setup Procedure:

In order to run the “Avatar AI Assistant” system, the following library and tools are required:

1. **Programming language : Python**
   a. **Python Installation:**
      i. If Python is not already installed on your system, download the latest version from the official Python website (https://www.python.org/downloads/) and follow the installation instructions for your operating system.This will also install pip for managing the packages.

2. **Python dependencies:** Some of the prerequisite libraries and tools for the AI Assistant Project
   a. **SpeechRecognition Python Library:**
      i. The SpeechRecognition library in Python provides a simple interface for performing speech recognition tasks. It supports multiple speech recognition engines, including Google Speech Recognition, IBM Speech to Text, and CMU Sphinx.
      ii. In our project we use this to convert the speech collected from VRChat app to text.
   b. **Pyaudio:**
      i. Pyaudio is a Python library providing bindings for PortAudio, facilitating cross-platform audio I/O operations. It enables working with audio streams in real-time, including recording from microphones, playback through speakers, and audio manipulation.
      ii. In our project we use this to stream audio to VRChat via the VirtualCable.
   c. **Pyttsx3:**
      i. Pyttsx3 is a Python library for text-to-speech (TTS) conversion, providing a simple interface to convert text into spoken words..
      ii. We leverage this library to convert the AI response text to Speech so that it can be sent and played as the voice of the AI avatar in VRchat app.
   d. **Python-osc:**
      i. Python-osc library for Open Sound Control (OSC) communication in Python. This allows for networking sound and osc -message making it useful for communication between audio applications and other multimedia systems.
      ii. Python-osc is used to send OSC messages from our python code VRChat via UDP connection to manipulate the Chat Box for showing AI avatars response as a speech box over the head as well as thinking dots visualization.

e. **Fastapi:**
   i. FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints.
   ii. It is designed to be easy to use and to provide high performance for both synchronous and asynchronous scenarios.
   iii. We leverage fastapi for making the text chat in VRchat possible in our project. VRChat sends Http requests using the “string-loading” method which is handled and responded to by the FastAPI backend.

f. **Uvicorn**:
   i. Lighting-fast server implementation which is used to run the FastAPI server in our project.

➢ The following should be installed for the system to work.

1. SpeechRecognition==3.10.3
2. openai==1.16.2
3. pygame==2.5.2
4. pyaudio==0.2.14
5. pyttsx3==2.90
6. python-osc==1.8.3
7. setuptools==69.5.1
8. fastapi==0.111.0
9. uvicorn==0.29.0

In order to install the above required packages (dependencies) go to the project folder in CMD. Then type in:

```
pip install SpeechRecognition==3.10.3 openai==1.16.2 pygame==2.5.2 pyaudio==0.2.14 pyttsx3==2.90 python-osc==1.8.3 setuptools==69.5.1 fastapi==0.111.0 uvicorn==0.29.0
```

Or, I have already listed this requirements in requirements.txt file inside the project folder from where these can be installed by typing:

```
pip install -r requirements.txt
```

3. APP and Tools :

a. VR-Chat App and account to play the VRChat.

   i. VRChat: The avatar in this project is meant for VRChat app. Thus, in order to download and install VRChat on PC, please follow the following instruction”
   ii. Requirements for VRChat:Steam Account (https://store.steampowered.com/)
   iii. PC meeting minimum system requirements:
      1. Requires a 64-bit processor and operating system
      2. OS *: Windows 8.1, Windows 10
      3. Processor: Intel® i5-4590 / AMD FX 8350 equivalent or greater

4. Memory: 4 GB RAM
5. Graphics: NVIDIA GeForce® GTX 970 / AMD Radeon™ R9 290 equivalent or greater
6. DirectX: Version 11
7. Network: Broadband Internet connection
8. Storage: 1 GB available space
9. VR Support: SteamVR

If need more detail refer this: (https://store.steampowered.com/app/438100/VRChat/)

iv. Downloading and Installing:
1. Install Steam on your PC if you haven't already.
2. Open Steam and create an account if you don't have one. (Once created, can use the same id to play the game as well)
3. Search for "VRChat" in the Steam Store.
4. Click on "VRChat" in the search results.
5. Click "Install" on the VRChat store page.
6. Follow the on-screen instructions to complete the installation.

v. Launching VRChat:
1. Once the installation is complete, you'll see "Play" turn into "VRChat" in your Steam library.
2. Click "VRChat" to launch the game.

vi. Account creation:
1. For this project we require two accounts for VRChat, thus make one additional account. One account will be used for AI assistant and another for the user testing the system

b. OpeanAI Api key:

i. The OpenAI API grants developers access to OpenAI's language models, such as the GPT series. It allows for sending text prompts to OpenAI's servers and receiving AI-generated responses, supporting various natural language processing tasks.

ii. Access to the OpenAI API is available through registration on the OpenAI website.

iii. OpenAI API Authentication:
1. Register for access to the OpenAI API on the OpenAI website (https://openai.com/).
2. Obtain your API key after registration.
3. Store your API key securely and do not share it publicly.

c. VB-Audio Cables (A+B):

Downloading and Installing VB-Audio Cable:

i. Visit the VB-Audio website (https://vb-audio.com/Cable/) and download the VB-Virtual Audio Cable software.

ii. Download the VB-Cable A+B as both channels are required for this project. A donation of 5 euros is required.

iii. Follow the installation instructions provided by the software installer if needed.

iv. Step to install:

1. After downloading, extract the zip file.
2. Inside, there will be separate folders for A and B channels installers named “VBCABLE_B_Driver_Pack43” and “VBCABLE_A_Driver_Pack43”.
3. Navigate into these folders individually and install the drivers by double-clicking “VBCABLE_Setup_x64” and following the installer instructions.
4. Upon completion, restart your computer to apply any changes made during the installation process.

Setting Up VB-Virtual Audio Cable:

- Configure the virtual audio cables as needed
    - Set one cable to capture user audio from VRChat (input).:
    - Set another cable to transmit AI-generated responses (output).

1. Use CABLE B to capture user audio from VRChat (input) : To make it happen configuration should be done both in Windows Sound settings as well as in python file as follows:
    - Setting in Windows Audio: Go to volume mixture in windows and then select the VRChat option (VR chat should be open to set it for the first time). On the output device audio of VRchat option choose `CABLE-B Input (VB-Audio Cable B)`
    - Setting in Python: In the project folder’s “default_values.py” file set the values for “VB_Cable_B_channel” to the index of `CABLE-B Output (VB-Audio Cable B)`

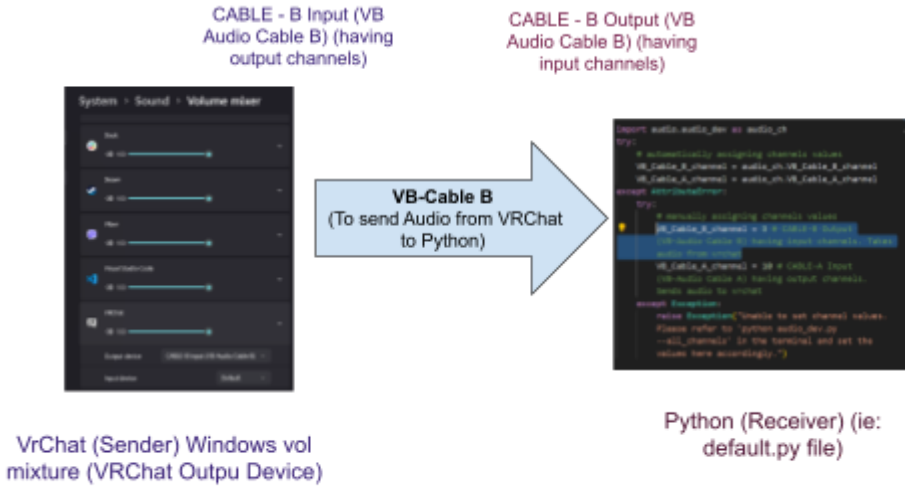
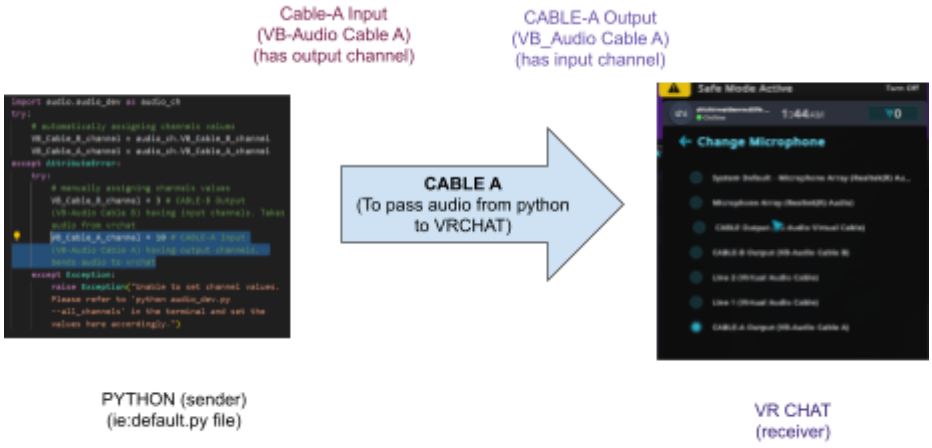


Fig:

2. Use CABLE A to transmit AI-generated audio responses (output) to VRChat. To make it happen configuration should be done both in python file as well as VRChat app as follows:
    - Setting in python file: In the project folder's "default_values.py" file set the values for "VB_Cable_A_channel" to the index of `CABLE-A Input (VB-Audio Cable A)`
    - Setting in VRChat app: Open VRchat app and go to settings for audio and in there change the Microphone to `CABLE-A Output (VB-Audio Cable A)`



Note: For details about the index of available audio channels for audio setups, navigate to the project folder in the terminal and run:

"`python audio_dev.py --all_channels`"

This will provide information about the available virtual channels for A and B input and outputs, which can be set in the "default.py" file. However, the project has already automated the setting for audio channels in the "default.py" file. Nevertheless, in the audio settings for the VR-Chat app and in Windows, the values should be set manually as instructed above.