Report on Text-to-Speech

This report covers a Python script developed to simulate text-to-speech (TTS) functionalities with multiple character voices using the eSpeak library.

Script Overview

1.Character Configuration:

A list of dictionaries is used to store configurations for each character, including name, voice type, speech rate, and volume.

2.Function Definitions:

speak_with_settings (text, voice, rate, volume): This function constructs and executes an eSpeak command with the given text and character settings.

show_characters(): This function displays the list of available characters to the user.

3.Main Loop:

A while loop prompts the user to input text, select a character, and trigger the TTS function. The loop continues until the user types an exit command.

Code Breakdown

Character Configuration

The ‘characters’ list defines ten characters with specific attributes:

```
characters = [
    {"name": "Emily", "voice": "en+f1", "rate": 150, "volume": 180},
    {"name": "Matt", "voice": "en+m1", "rate": 120, "volume": 160},
    {"name": "Diane", "voice": "en+f2", "rate": 180, "volume": 140},
    {"name": "Helen", "voice": "de+m1", "rate": 100, "volume": 200},
    {"name": "Eva", "voice": "es+f1", "rate": 200, "volume": 120},
    {"name": "Emma", "voice": "it+m1", "rate": 140, "volume": 100},
    {"name": "Daniel", "voice": "pt+f1", "rate": 160, "volume": 80},
    {"name": "Marlene", "voice": "ru+m1", "rate": 130, "volume": 60},
    {"name": "Meryl", "voice": "sv+f1", "rate": 170, "volume": 40},
    {"name": "Lea", "voice": "tr+m1", "rate": 110, "volume": 20},
]
```

Each character dictionary includes:

-name: The character's name.

-voice: The eSpeak voice identifier.

-rate: Speech rate in words per minute.

-volume: Volume level.

Text-to-Speech Function

The speak_with_settings function constructs the eSpeak command and uses subprocess.run to execute it:

```
def speak_with_settings(text, voice, rate, volume):
    command = f'espeak "{text}" -v {voice} -s {rate} -a {volume}'
    subprocess.run(command, shell=True)
```

The command integrates user input text and character-specific settings.

Character Display

The show_characters function prints the list of available characters:

```
def show_characters():
    print("Available characters:")
    for i, char in enumerate(characters, 1):
        print(f"{i}. {char['name']}")
```

Main Program Loop

The main loop handles user input and character selection:

```
while True:
    answer = input("Enter what you want the robot to say (type 'exit', 'bye', or 'quit' to quit): \n")

    if answer.lower() in ['exit', 'bye', 'quit']:
        print("Exiting the program.")
        break

    show_characters()

    char_index = input("Enter character number (1-10): ")
    try:
        char_index = int(char_index) - 1
        if 0 <= char_index < 10:
            char_settings = characters[char_index]
            speak_with_settings(answer, char_settings["voice"], char_settings["rate"], char_settings["volume"])
        else:
            print("Invalid character number. Please enter a number between 1 and 10.")
    except ValueError:
        print("Invalid input. Please enter a number between 1 and 10.")
```

The loop continues until the user inputs an exit command, at which point the program terminates.