Avatar AI Assistant:

The Project Overview (Till 1st Month progress)

1. Errors encountered:

1. Solved Errors:

a. Error 1: gtts saved ‘wav’ file format not working

Error:
Failed to open file file.wav as a WAV due to: file does not start with RIFF id

Solution:
Upon investigation, I found that GTTS (Google Text-to-Speech) saved the file in MP3 format despite maintaining the '.wav' extension. Consequently, the file lacked the appropriate 'wav' format. Resolution: Given the necessity for WAV format to ensure high-quality audio output for avatar streaming, I opted to substitute the GTTS module with another text-to-speech solution known as pyttsx3, a Python library proficient in translating text to speech."

b. Error 2: OSC-message gave no effect in facial expression:

i. Error:
While implementing OSC messaging for facial expression on the Avatar, I encountered an issue. The code I utilized worked seamlessly for one avatar but failed for another.

```
def facial_express(emo_val):
    osc_address = '/avatar/parameters/FacialExpression'
    osc_value = emo_val
    client.send_message(osc_address, osc_value)
```

ii. Solution:
It became apparent that not all avatars share the same OSC address. Additionally, creators have the freedom to define default OSC addresses for their avatars. Therefore, it's crucial to either set the OSC address

during the avatar's creation or have a comprehensive understanding of the parameters supported by the avatar.

To ascertain the parameters supported by a specific avatar, one can navigate to VRChat's settings > debug > Avatar debug. Alternatively, locally, this information can be found within the config file located at: *`~\AppData\LocalLow\VRChat\VRChat\OSC\{userId}\Avatars\{avatarId}.json`*

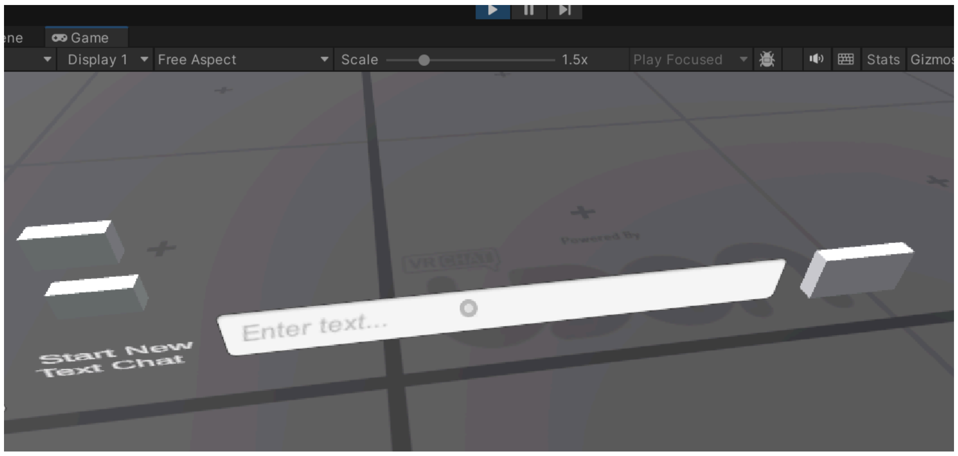
Upon inspecting the config file, I discovered that the OSC address for facial expression on the Avatar I was working with was actually *"/avatar/parameters/Face/Facial"*, rather than *"/avatar/parameters/FacialExpression"*. Making this adjustment resolved the issue.

```
def facial_express(emo_val):

    osc_address = '/avatar/parameters/Face/Facial'
    osc_value = emo_val
    client.send_message(osc_address, osc_value)
```

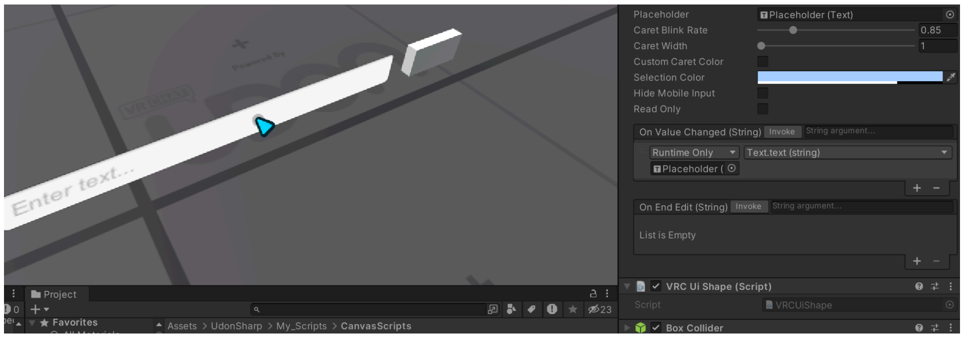
c. Error 3:: Text Input field was not write able in Unity Game:

iii. Error:
 I was not able to write on both the legacy as well as TMX Pro input field when in the "game" scene in Unity.



iv. Solution:
I was able to write on the input field in the "game" scene after adding a component called "VRC Ui Shape (Script) on the inspector window of that

Input Text field.



Placeholder: Placeholder (Text)
Caret Blink Rate: 0.85
Caret Width: 1
Custom Caret Color
Selection Color
Hide Mobile Input
Read Only

On Value Changed (String) Invoke String argument...
Runtime Only | Text.text (string)
Placeholder (

On End Edit (String) Invoke String argument...
List is Empty

VRC Ui Shape (Script)
Script: VRCUiShape
Box Collider

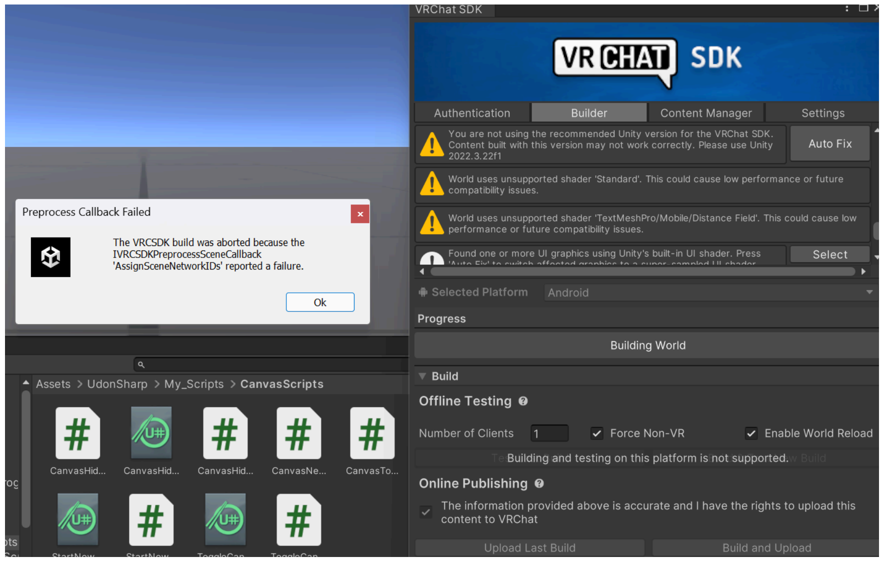
d. Error 4: Error in audio channel settings:

Error might code due to mismatch of audio channels and sampling frequency given by the user. So, in that case please run the “audio_dev.py” file using “python audio_dev.py” to get the information about the available audio channels for both VB-audio A and B. Then set them accordingly as specified in the above sections.

e. Error 5: Error due to python versions: Please set the python version to 3.12.0 if any new updates may depreciate dependencies. Also if you use 3.12.0 please install setup tools manually if not automatically installed using “pip install setuptools” (ref: https://stackoverflow.com/questions/69919970/no-module-named-distutils-but-distutils-installed)

2. Unsolved Errors:

a. When I converted some C# Script to Udon# consequently this didn’t let me build and upload the world threw this error:



Preprocess Callback Failed

The VRCSDK build was aborted because the IVRCSDKPreprocessSceneCallback 'AssignSceneNetworkIDs' reported a failure.

You are not using the recommended Unity version for the VRChat SDK. Content built with this version may not work correctly. Please use Unity 2022.3.22f1

World uses unsupported shader 'Standard'. This could cause low performance or future compatibility issues.

World uses unsupported shader 'TextMeshPro/Mobile/Distance Field'. This could cause low performance or future compatibility issues.

Found one or more UI graphics using Unity's built-in UI shader. Press

Selected Platform: Android

Progress

Building World

Build

Offline Testing

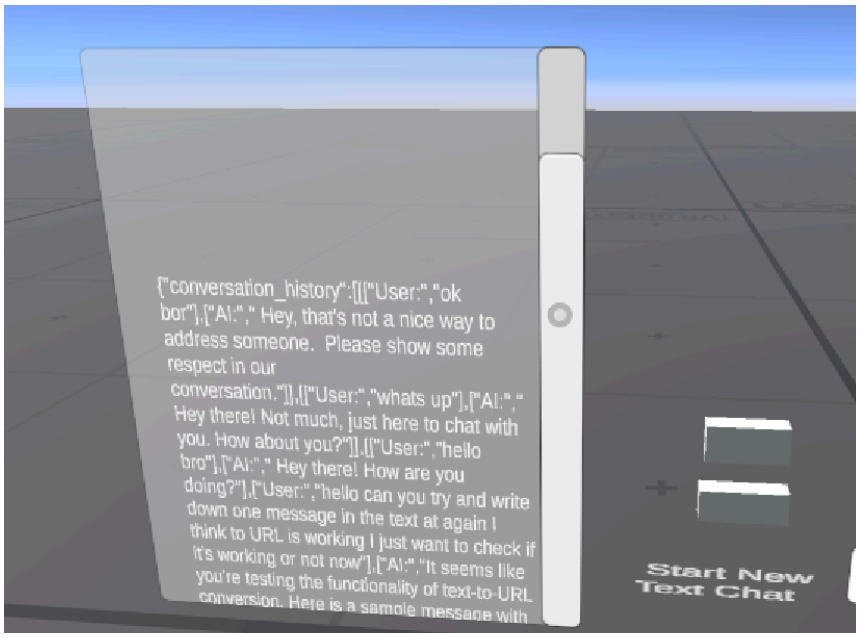
Number of Clients: 1 | ✓ Force Non-VR | ✓ Enable World Reload

Building and testing on this platform is not supported.

Online Publishing

✓ The information provided above is accurate and I have the rights to upload this content to VRChat

b. The UI ScrollView wont work (content is not moving the when slides) when and the text is not placed in the right place in Unity “game” window. However it works when I keep the camera from World Space to Screen Space.



```
{"conversation_history":[[["User:","ok
bor"],["AI:"," Hey, that's not a nice way to
address someone.  Please show some
respect in our
conversation."]],[["User:","whats up"],["AI:","
Hey there! Not much, just here to chat with
you. How about you?"]],[["User:","hello
bro"],["AI:"," Hey there! How are you
doing?"],["User:","hello can you try and write
down one message in the text at again I
think to URL is working I just want to check if
it's working or not now"],["AI:","It seems like
you're testing the functionality of text-to-URL
conversion. Here is a sample message with
```

c. I have utilized the VRCUrl input for taking the user text message from the VRChat users. This is forcing the input to include the URL and user has to type the URL before any message they want to send. I am not able to tweak this or found any effective alternative.

2. IMPORTANT REFERENCES

1. OSC Use:
    1. https://qiita.com/will-yama/items/2945ab43c28352c2817e This link uses OSC to manipulate chat avatar. I got the hint to leverage OSC for showing facial expressions to my avatar via this.
    2. https://docs.vrchat.com/docs/osc-as-input-controller This is the official link of documentation for OSC use in VRChat.
        a. This explains to how can use the OSC as Input controller.
        b. This link provided me with the hint to use the “chatbox/input” to show text over avatar’s head and “chatbox/typing” to make the avatar display thinking dots when the backend is processing for AI response generation.

3. https://docs.vrchat.com/docs/osc-debugging This link describes how we can visualize the osc messeges being passed on our VRChat. This lets us to test if our OSC-messege is getting any changes in the parameters of avatar or not.
4. https://docs.vrchat.com/docs/osc-avatar-parameters This link describes how we can get information about the parameters of the Avatar that can be manipulated using OSC. It gives idea about the config file of the avatar that is saved inside the VRChat' OSC folder inside our C drive directory. The config file can be essential if we want to know about the OSC parameters, OSC address and types that the Avatar we are using supports.

2. VRCUrl, VRCUrlInputField and String Loading:
    1. https://udonsharp.docs.vrchat.com/vrchat-api/#vrcurlinputfield This link provided me with information on VRCUrl and VRCURLInputField
        a. VRCUrl: This allows us to set a URL during editor time not in runtime. Thus we need to set this up in unity itself before building and uploading in VRchat world. I leveraged this to use it to set the consistent fastapi endpoint for showing chat history and creating new text chat sessions.
        b. VRCURLInputField: This field allows users to enter any URL during the game runtime. A Http request can be made to this URL. I have leveraged the VRCUrlInputField as an input field for user messages. When a user types in a message preceded by the URL of my backend fastapi (responsible for responding with AI answers) and makes a HTTP request, the fast api gets the request and provides the AI answers back to VRChat (Udon Script).
    2. https://creators.vrchat.com/worlds/udon/string-loading/: This link (String loading) provides essential concepts on how to get the data downloaded from server to udon sharp. Userful for chat applications between external servers and VrChat..

    3. https://qiita.com/namanonamako/items/96338ac346bafa912f08 : gives a good idea on how to make a request from U# to any external server and get the response back. This combines the idea of VRCUrl and String Loading. Moreover, the work involves receiving the string from the server but not sending string data to the server.
    4. https://docbase.io/posts/3074819/sharing/02ddccc2-f53a-4138-975b-0560cd48d6af This document in the link provided me more idea on how to ulitlize the VRCUrInfputFiled to make two way string communication possible from VRchat to external server and viceversa. I slightly improvised this method to make a fastapi server based chat system using similar approach of using VRCUrlInputfield instead of VRCUrl field that enables runtime http request to make along with user message passing to external server.