Criterion A: Planning

Scenario

My uncle lives in a small village in the south of Spain. He works as a social worker and he has many problems communicating with immigrants because he does not speak their languages. These people are usually very poor economically and have the urge to be employed. They often end up working in plantations that are in that zone. But to be able to work with aliments, they need to get a certificate of food manipulator and must pass a small course.

The trouble is that my uncle finds it impossible to teach these people because of discrepancies in the language. Since it's a very remote and rural area in Spain, there are no available people who speak their languages and are also certified to do the course. Firing a human interpreter would be a cost that the social services could not afford, that's why he asked me to develop a computer software to solve his problem. Because they can afford the cost of buying a stock of headsets so the attendants to the course could listen to a computer made translation.

I have done a small interview with my uncle that can be found on the Appendix.

Rationale and proposed solution

As stated above, my goal is to create an interpreter. It must be able to translate a conversation almost simultaneously (maybe just some seconds of delay) given a voice as input and output the translation in another language also in the form of voice.

As it's a software that is just needed by the person who wants to communicate with people who do not speak his language (in this case, my uncle), the best choice is to make a stand-alone application that can be downloaded on his computer. Of course, internet connection will be needed as the translation services are online.

I've decided to write the app in Python, as it's a very easy to use high level language and I've used it before. Moreover, it has the advantage that there are plenty of libraries and APIs written in Python that will facilitate the coding and simplify a lot the underlying complexity that translation requires. My app will rely on open source code for the translation, voice-text conversions and audio management.

The user interface of the application will also be developed in Python, using a GUI module. My goal is to write the whole app in this language.

Success criteria

I have adjusted the success criteria to a simpler version of the application that the client has requested, since his idea was way too complex for my skill level.

- ☐ Make a play/pause button to start/stop interpreting
- ☐ The translation must be simultaneous (reasonable delay and no omissions)
- ☐ User is able to select the source language and the target language

- [ ] Make a switch languages button
- [ ] Display both the text in its original language and its translation on the screen
- [ ] Make a translation history to store all translations
    - [ ] Reproduce button to hear again a translation
- [ ] Friendly and pretty user interface
- [ ] Connect more than one output device (each with a different language)
- [ ] Can be used on multiple operating systems (app is cross platform)

Wordcount: 396