**DSP Lab Final Project Report**

Members:

Eugene Lan (yl8241),

Chan-Yu Cheng (cc7283)

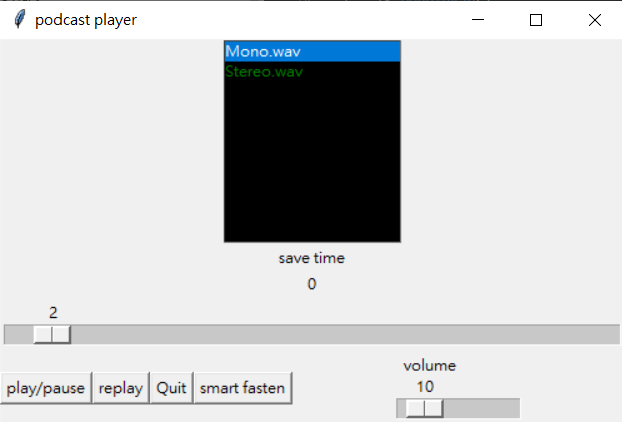
**Motivation**

Inspired by Smart Speed by Overcast a podcast player application. When we were using a normal music player to play some speech or podcasts. We found that there were sometimes having some unpleasant blank (non-speaking) parts and noises due to various reasons (recording quality, lecturer’s’ pauses, etc.). We would like to save some time when listening to the podcasts without fast-forward the informative parts. Hence, we think using some DSP techniques to skip the unpleasant blank would be a great approach to accomplish the task. Therefore, we use material mentioned in class to design a podcast player with similar function like “Smart Speed”.

**Library Used**

All the library we used are from the course: math, pyaudio, struct, tkinter, wave, time, numpy, os

**Implementation**



Using tkinter design a user interface to interact with user

Function description:

Listbox : tkinter’s listbox module. Let user to select wav file to play under podcast file.

Slider : tkinter’s slider module. Similar function like bar of music player. Let user to jump to certain time point in the wav file, and it displays the time we have played in wav file.

play/pause button : tkinter’s button module. When the file is playing, user press this button can pause the wav file. Vice versa, when the file is pause, this button will resume the wav file.

Replay button : tkinter’s button module. Return timeline to the beginning of wav file.

Quit button : tkinter’s button module. Finish the program and leave the windows.

Volume slider : tkinter’s slider module. Use to control the output volume. The basic volume at 10.

Smart fasten button : tkinter’s button module. Switch smart fasten function on or off. Smart fasten function is analysis by Fourier transform block by block. According to the “EQing vocals : What is happening in each frequency range in human voice” on FLYPAPER, the human voice is generally located on 80 to 300. Therefore, we transform data to frequency domain and if the frequency between 80 to 300, this band of frequency exceed the value we set, then we keep this 0.1 second block of track and play it out. Vice versa, if this band is not pass the criteria we set, we skip this block of track and record the total skip time to the label.

**Reference :**

EQing vocals : What is happening in each frequency range in human voice by Kenneth Estrada y Santiago

https://flypaper.soundfly.com/produce/eqing-vocals-whats-happening-in-each-frequency-range-in-the-human-voice/