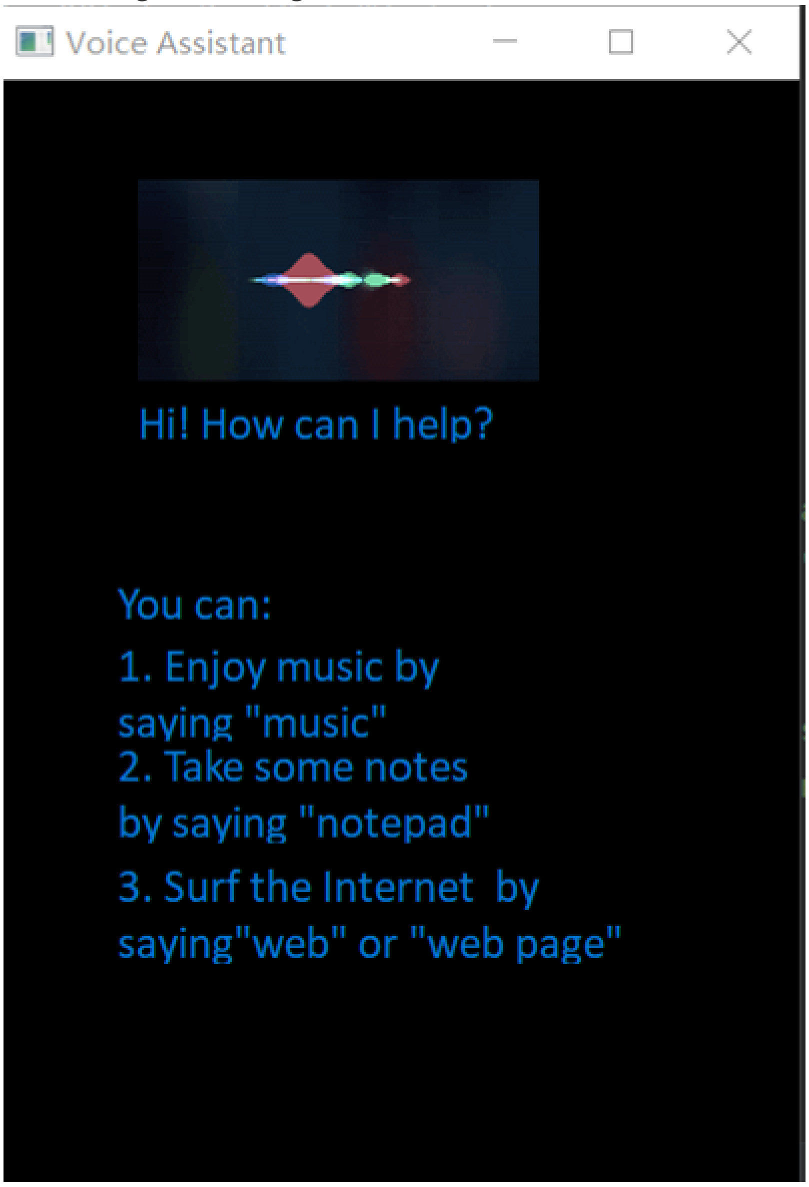
HCI Lab1 Report

1. The modifications to GUI

First I changed the background's icon to make it more beautiful like this:



Hi! How can I help?

You can:
1. Enjoy music by saying "music"
2. Take some notes by saying "notepad"
3. Surf the Internet by saying"web" or "web page"

Then,I modified and added some tips to help users easierly to use the program.

2. The modifications to Codes

2.1 Add functions to make the program stronger

- **Play music**

```
win32api.ShellExecute(0, 'open', 'music.mp3', '', '', 1)
```

- **Open a text file**

```
win32api.ShellExecute(0, 'open', 'notepad.exe', '', '', 1)
```

- **Open web browser**

```
webbrowser.open("https://www.baidu.com")
```

2.2 Use thread to control the recgonizition

```
class MyThread(threading.Thread):
    def __init__(self):
        super(MyThread, self).__init__()
        self._running=True

    def recognize_speech_from_mic(self,recognizer, microphone):
        """Transcribe speech from recorded from `microphone`.

        Returns a dictionary with three keys:
        "success": a boolean indicating whether or not the API request was
                   successful
        "error":   `None` if no error occured, otherwise a string containing
                   an error message if the API could not be reached or
                   speech was unrecognizable
        "transcription": `None` if speech could not be transcribed,
                   otherwise a string containing the transcribed text
        """
        # check that recognizer and microphone arguments are appropriate type
        if not isinstance(recognizer, sr.Recognizer):
            raise TypeError("`recognizer` must be `Recognizer` instance")

        if not isinstance(microphone, sr.Microphone):
            raise TypeError("`microphone` must be `Microphone` instance")

        # adjust the recognizer sensitivity to ambient noise and record audio
        # from the microphone
        with microphone as source:
            recognizer.adjust_for_ambient_noise(source)
            audio = recognizer.listen(source)

        # set up the response object
        response = {
            "success": True,
            "error": None,
            "transcription": None
        }

        # try recognizing the speech in the recording
        # if a RequestError or UnknownValueError exception is caught,
        #     update the response object accordingly
        try:
            response["transcription"] = recognizer.recognize_sphinx(audio)
        except sr.RequestError:
            # API was unreachable or unresponsive
            response["success"] = False
            response["error"] = "API unavailable"
        except sr.UnknownValueError:
            # speech was unintelligible
            response["error"] = "Unable to recognize speech"

        return response

    def stop(self):
        self._running=False
        print(self._running)

    def run(self):
        recognizer = sr.Recognizer()
        microphone = sr.Microphone()
        while self._running:
            res = self.recognize_speech_from_mic(recognizer, microphone)
            if res["error"]:
                print("ERROR: {}".format(res["error"]))
                continue
            words = res["transcription"]
            if self._running:
                print(words)
                if words.lower() in ["music","you think"]:
                    win32api.ShellExecute(0, 'open', 'f1lcapae.wav', '', '', 1)
                elif words.lower() in ["notepad","note pad","goat's head","goat
head"]:
                    win32api.ShellExecute(0, 'open', 'notepad.exe', '', '', 1)
                elif words.lower() in ["web page","webpage","web"]:
                    webbrowser.open("https://www.baidu.com")
            else:
                break
```

3.The accuracy of speech recognition

Due to the use of external speech recognition api, the recognition accuracy rate is relatively low, which is mainly reflected in:

1. **Recognize one word as another word**

   eg. recognize `play` as `today`.

2. **Recognize unclear words**

   The program sometimes recognizes ambiguous words from background noise.

4.How to improve the accuracy

1. **Speak in quiet environment**

   In a quiet environment, the program can avoid the interference of noise in the environment, thereby improving accuracy.

2. **Use words that are easier to recognize as commands**

   eg. At first I used `play music` as the command to play music, but soon I found that the program is not sensitive to the unvoiced consonant `/p/`,so this command can hardly be recognized accurately.Then I simply used `music` to take place of `play music`,and the recognition accuracy has been significantly improved.

3. **Associate the command to be recognized with words that sound similar to it.**

   There is still another way to improve the recognition accuracy.That is,associate the command with words that sound similar to it.
   eg. associate **`"notepad","note pad","goat's head","goat head"`** with command **`notepad`**.When the user speaks **`notepad`**,the program still execute correct command even it recogize the words as **`goat's head`** and so on.