|  |  |
| --- | --- |
| **Project Requirements** | **Student: Xin Dong** |

Project Purpose:

Voice command-controlled mouse and keyboard make computer assessable for people in special needs. Beyond that, it also free up hands to operate a computer in normal people’s life. This software development project will develop such a tool from scratch in Python environment. This tool enable human to use voice control computer mouse and keyboard and operate corresponding commands.

Project Structure:

This tool will basically connect 3 major library/module of python, which are SpeechRecognition, PyAudio, and PyAutoGUI.

* **SpeechRecognition** is an open source library which can perform Speech to Text function. It supports several engines and API such as CMU Sphinx, Google Speech Recognition, Google Cloud Speech API, Microsoft Bing Voice Recognition, Wit.ai etc. This project will only use two type of input, microphone to capture natural speech as input and audio file with .WAV format as input.
* **PyAudio** provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms. PyAudio is required in this library if using microphone as input.
* **PyAutoGUI** lets Python scripts control the mouse and keyboard to automate interactions with other applications.

Besides above three required python module/libraries, the python **“mouse”** library may benefit the project so will be installed as well.

The project starter code to capture voice command will be look like:

import speech\_recognition as sr

# obtain audio from the microphone

r = sr.Recognizer()

with sr.Microphone() as source:

    print("Say something!")

    audio = r.listen(source)

print(r.recognize\_google(audio))

# obtain path to "recording.wav" in the project folder as this script

from os import path

AUDIO\_FILE = path.join(path.dirname(path.realpath(\_\_file\_\_)), "C:\LWTECH\CSD480-CapStone\Recording.wav")

# use the audio file as the audio source

r = sr.Recognizer()

with sr.AudioFile(AUDIO\_FILE) as source:

    audio = r.record(source)  # read the entire audio file

print(r.recognize\_google(audio))

The voice commands are converted to text and stored in variables. PyAutoGUI will use these variables to perform corresponding commands.

Barrier and Constrains:

Speech-to Text accuracy is a challenge for this project. Many times the computer may interpreter voice to incorrect text, due to speech environment, English speaker’s dialect etc., which will cause project failure.