Jaden Booher
405-333-407
January 19, 2023

ECE 180D Lab 2 Writeup

Tasks:

1.
```
Received message: "0.4626054473174738" on topic "ece180d/team1" with QoS 1
Received message: "0.0467267308177316" on topic "ece180d/team1" with QoS 1
Received message: "0.5669882299813243" on topic "ece180d/team1" with QoS 1
Received message: "0.33142116624655793" on topic "ece180d/team1" with QoS 1
Received message: "0.7613954493869307" on topic "ece180d/team1" with QoS 1
Received message: "0.8123739998406743" on topic "ece180d/team1" with QoS 1
Received message: "0.6758966148060799" on topic "ece180d/team1" with QoS 1
Received message: "0.7964676246503285" on topic "ece180d/team1" with QoS 1
Received message: "0.6703180182403764" on topic "ece180d/team1" with QoS 1
Received message: "0.30843686620453326" on topic "ece180d/team1" with QoS 1
```

   a. Communication between a publisher and a client is possible using MQTT, which will be useful for our project. Though setting up the broker can be difficult, and latency can be a bit of an issue, the software is not too challenging to use. The typical latency would be approximately 100-300 milliseconds, depending on how we set up our code. Overall, we will most likely use this software for remote communication during our project.

2.
```
(base) C:\Users\Jaden\180DA-WarmUp\Lab 2>python Recognizer.py
Start speaking
result2:
{   'alternative': [   {'confidence': 0.952371, 'transcript': 'Orange'},
                       {'transcript': 'Lawrence'},
                       {'transcript': 'Florence'}],
    'final': True}
You said: Orange
Runtime: 4.0696046352386475

(base) C:\Users\Jaden\180DA-WarmUp\Lab 2>python Recognizer.py
Start speaking
result2:
{   'alternative': [   {'confidence': 0.87587702, 'transcript': 'a'},
                       {'transcript': 'hey'},
                       {'transcript': 'Bay'}],
    'final': True}
You said: a
Runtime: 8.25125765800476
```

- Saying letters is exponentially more complicated for the software to identify and thus takes a lot more computation and a performance hit to do it accurately.

- As part of the *listen* function in the speech recognition python software, we can specify phase time length to be able to accurately detect what we are looking for.
- Noise is a very prevalent problem, and an effective filtering method, quiet environment, or good microphone would be essential.

Individual thoughts:

a. We can use the speech recognition to act as inputs for the game we are creating.
b. We want the speech recognition to be simple: if we want to use speech recognition for the game, commands should be easy and fast to say, both to allow the player to effectively play the game and will enable the software to process the words effectively.
c. Accuracy is extremely important; if a word is incorrectly identified, it can drastically impact how an individual plays the game and can ruin the game altogether. Additionally, the speech recognition should have a very limited delay, hopefully under 1 second.
d. I believe that setting up the speech recognition correctly in the software to allow for fast and accurate translation is the most important. However, noise removal and/ or a decently tuned microphone would also be important.