**Python Workshop Project Report**

**Husnu Melih Erdogan**

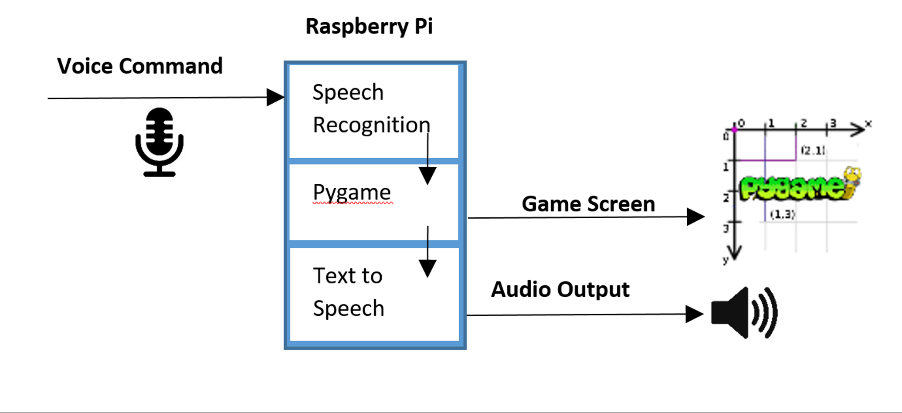
**Voice Controlled Video Game**

The goal of this project is to develop a voice controlled 2D video game in Python. I picked Raspberry Pi as computer that runs the game. The reason I use Raspberry Pi in this project is that it is powerful enough to do all the task that I need in this project such as speech recognition, speech synthesis, and running the 2D game system. It also has so many libraries which makes it very easy to prototype any software in a very short time. There is also a very large Raspberry Pi community on the Internet.

**Game:**

The game is basic a 2D video game. In the game there is a character called Vicky who collects food that appears on the map. In order to move Vicky to any direction, player uses voice commands such as ‘UP’, ‘DOWN’, ‘RIGHT’, ‘LEFT’. Player can also interact with Vicky by asking him some questions. Vicky responses to the question and commands such as ‘Vicky!’, ‘How are you?’, and What is your name? Vicky also speaks when user tells Vicky go beyond the borders of the map.

**Top Design of the Project:**



**System:**

* Raspberry Pi
* Microphone
* USB Stereo Sound Adaptor:I needed to buy this adaptor because Raspberry Pi doesn’t have any build-in sound card on it. <https://www.amazon.com/gp/product/B001MSS6CS/ref=oh_aui_detailpage_o05_s00?ie=UTF8&psc=1>
* Stereo Speaker
* Usb WiFi Module <https://www.amazon.com/Edimax-EW-7811Un-150Mbps-Raspberry-Supports/dp/B003MTTJOY/ref=sr_1_1?s=pc&ie=UTF8&qid=1465520067&sr=1-1&keywords=raspberry+pi+wifi>

**Libraries:**

**Speech Recognition Library:** I used CMU Pocket Sphinx for speech recognition. Speech recognition part consist of two different sub tasks. First task is voice recording. I used Pyaudio library for voice recording. Instructions and details of this library can be find at this link.[https://people.csail.mit.edu/hubert/pyaudio/](https://people.csail.mit.edu/hubert/pyaudio/%20)

The second part is using the audio file for speech recognition. In order to install the library, I followed the instruction at this link <http://cmusphinx.sourceforge.net/wiki/raspberrypi.>

This part is the most challenging part of this project. Since this library used by only a small group of people, in order to understand how the library works and how to implement it into my project, I had to spend so much time and do lots of research on the Internet. In order to get this library work, basically it needs two things. First is the audio file or microphone input, and the second thing is the language model. Language modal consist of the words that you want to use in your speech recognition library. I used only a few words to control the character in the game. Since I use this library on Raspberry Pi, the larger library means bigger error rate and longer time to get a result. Once the library gets the results ready in string format, game class uses them to move the character.

**Pygame:** Pygame library that includes a large set of Python modules designed for building a video game. The library has pretty much everything what a game developer could ask for. For example, developers can easily add graphical user interface, sound, or images to their projects by using the modules in this library. This library is also supported by a huge community. I used Pygame to build my game engine and create a display on where the characters move and other items are places. I also used the sound play module comes with the library to give sound feedback when player collects food and walks around the map.

<http://www.pygame.org/hifi.html>

**Text to Speech:**

Although there are several different speech synthesis options for python, I picked espeak. Unlike Google TTS or ATT TTS, espeak works offline and the result is good enough to make a robot or machine speak. I followed instruction at this link. [https://pythonspot.com/en/speech-engines-with-python-tutorial/](https://pythonspot.com/en/speech-engines-with-python-tutorial/%20). Basically, you give a text in string format to the library and it plays the sound. In order to get a better and a more realistic result, I plan to use a better TTS service such as Google TSS in the future.

**Conclusion:**

All in all, I learned a lot from this project. I learned lots of new things about speech recognition, Raspberry Pi, speech synthesis, game development and of course Python. It was a very helpful and instructive project to me.