Senior Design
ENG EC 463

Memo

To: Professor Pisano

From: Arley Trujillo, Shivani Bhatia, Laura Salinas, Priya Kapadia, Steven Graham

Team: 14

Date: 11/14/17

Subject: Vobot First Deliverable Test Plan

1.0 Android Application

Description & Goal:

The goal of the phone application is to create an interactive way for the user to control the robot. “Control” in our case is defined as choosing behaviors such as dancing, playing a playful sound, or directing robots movements. These behaviors exhibit positive reinforcement, thus motivating the child to continue practicing the word. The app, which connects to the robot via bluetooth, listens to the user’s vocalizations, displays what has been heard on screen, and makes the robot exhibit positive feedback.

Procedure:

The application was developed using Java in the Android Studio environment. The application uses Gradle, which is an open source build automation system. The application uses Java for the functionality of the application, and XML for the view or display of the application. There are two pages including the home page, and the drive page. The app can be built using Gradle, and uploaded to the Android Phone. The home page uses the MiP library to automatically detect any MiP robots available in the bluetooth connectivity range. It uses the MiP library authentication to connect to the phone without requiring the user to select the robot and input a PIN number. To prompt the robot to carry out functionality, the library sends a command to the sensors inside the MiP.

On the home page, there is a button called “Start Listening” that begins the entire listening process. It first outputs an audio file to the speaker that prompts the child to say a particular word. For instance, this could be “Say the word table.” Following the prompt, the application starts listening for the child’s attempt. If the child correctly says the word table, it sends a message to the robot to celebrate. If the child does not say the correct word, the robot does not provide any feedback. The application uses the speech to text API to listen to the user, and change the text on the screen to display what the user has said. If the API does not detect a recognizable word, it simply prompts the user to try again. The application’s speech recognition employs the Google Speech Recognition component built into the Android phone. The source code to turn the speech to text was modeled after the code found on: https://www.nextgenearn.com/android-speech-text_android-voice-recognition_no-deprecation/.

The second page, which is the drive page, allows the user to move the robot by clicking on the screen, and dragging their finger in the direction of movement. It uses the same library to send these commands to the robot.

Verifiable Result:
Once the app is compiled with the Gradle build on Android Studio, the user may unplug their phone from the computer and is able to control the robot manually, as long as it is still connected through bluetooth. The user can also record their voice and have the app display the results onto the screen.

2.0 Speech Recognition Algorithm

Description & Goal:
Our motivation in speech recognition is to understand what our users are saying. We will be able to monitor a user's speech progression by using an API that receives vocalizations as input and translates them to text. Handling user input in this way allows us to shape a child's vocalizations dynamically in the short term and ultimately getting them to speak a particular set of words with less difficulty. We are currently capable of detecting intelligible speech - vocalizations that are partial pronunciations either flag an error or return a next best approximation to a known word. In deliverables further down the line, we would like to score these vocalizations which would allow us to implement a leveling system for user progress and reward behavior.

Procedure:
In order to run the algorithm we first need to download and modify the source code that uses Google speech recognition and port it to supporting Android architecture. When the code is compiled on its own it can be used to detect spoken words by users. By combining the algorithm with the application framework we created, we are able to select a “start listening” option on the Android app to bring up the onboard microphone which begins to listen for input. We can test that the application is working properly by running multiple tests with connectivity - making sure we can connect successfully to the MiP robot and that it responds in a timely manner when we select the record option.

Verifiable Result:
Once the user finishes speaking and the algorithm compiles, we will print to a screen on the app the words that have been understood. By capturing users voice data we will be able to compare their vocalizations to what the model word from a database *should* sound like. The comparison will allow us to score user and give feedback on speech progress. Voice recognition is one of the foundations of our project and is a necessary component in order to move forward with more advanced performance.

3.0 Wireless Communication

Description & Goal:
The goal of wireless communication is to transmit vocalizations via bluetooth to the application. The wireless communication system will take in vocalizations made into a microphone and process them on an Arduino platform. The Arduino will then transmit the speech-to-text conversion via bluetooth to an application, which will display the text on the screen. This method of wireless communication creates an efficient system which will take in, process, and transmit information with low latency.

Procedure:
In order to set up the wireless communication system, we need an Arduino Uno, Adafruit Bluefruit LE SPI friend, Adafruit Electret Microphone Amplifier, and the Bluefruit LE Connect application either an Android or iOS device. Begin by setting up your circuit using the following pinouts:

Device Pinouts (BLE and Microphone)	**Arduino Pinouts**
SCK (BLE)	13
MISO (BLE)	12
MOSI (BLE)	11
CS (BLE)	8
IRQ (BLE)	7
OUT (Microphone)	A0
VIN (BLE), VCC (Microphone)	3.3V
GND (BLE), GND (Microphone)	GND

After the circuit is constructed, download the uSpeech and BLE libraries for the Arduino. Then, compile and upload the Arduino code to the Arduino Uno. Once the program is running, we can connect the Adafruit BLE device to the downr0loaded Bluefruit LE Connect app on a phone. Open the application on the phone, click the ‘Connect’ button next to the Adafruit BLE device, and then click on the UART option on the next screen. This will open a sending and receiving screen on the application. We can test that the wireless communication is running by typing a message into the serial monitor and pressing ‘Send’, and verifying that it appears on the applications UART screen. If the devices are not connected, ‘Failed to Send’ will be printed on the serial monitor.

Verifiable Result:
Once the circuit is set-up and the code is uploaded to the Arduino, we can speak into the microphone and view the detected phonemes being sent via bluetooth on the serial monitor. We can compare the sent phonemes to the received phonemes that appear on the UART screen on the application, as long as the bluetooth device is still connected to the application.