The University of Edinburgh

School of Informatics

System Design Project

TOPIC:

User Guide

Leo, an Assistant Gaming Robot for the Elderly

DATE: March 22nd , 2018

1 Introduction

Leonidas is an assistant robot constructed with Lego pieces. Since elderly people respond better to robots than tablets[1][2], we have programmed Leo to play interactive games that they would normally play on a tablet. This helps them maintain their cognitive fitness as it tests their memory and reaction times for the games. Leo has the ability to move his arms. As a user, you are expected to press the buttons on the arms indicated by the corresponding LEDs lighting up. Leo has the Amazon Alexa integrated into its system, which allows users to speak to it and choose games verbally. The reason for using both the Alexa and button pressing as an interaction mechanism between the user and Leo is that people benefit from congruent multi-sensory information[3]. Leo also has face detection and table edge detection abilities which make sure it faces the user and doesn't fall off the table. This document contains guidelines on how to set up the system and how to make Leonidas perform as expected.

2 Hardware overview

2.1 Components

Leo comes with a suite of hardware components. A full-spec Leo will have equipped the following: (Note that the numbering corresponds to the numbers in Figure 1.)

1. 2 X Lego EV3 Bricks
2. Raspberry Pi 3 Model B
3. 2 X Lego EV3 Ultrasonic Sensors
4. Lego EV3 Gyro Sensor
5. 3 X Lego EV3 Touch Sensors
6. 4 X Lego EV3 Large Servo Motors
7. 2 X Lego EV3 Medium Servo Motors
8. Set of LEDs of colour RED
9. Set of LEDs of colour GREEN
10. Set of LEDs of colour YELLOW
11. Amazon Echo Dot
12. Power Bank
13. Logitech HD WEBCAM C270

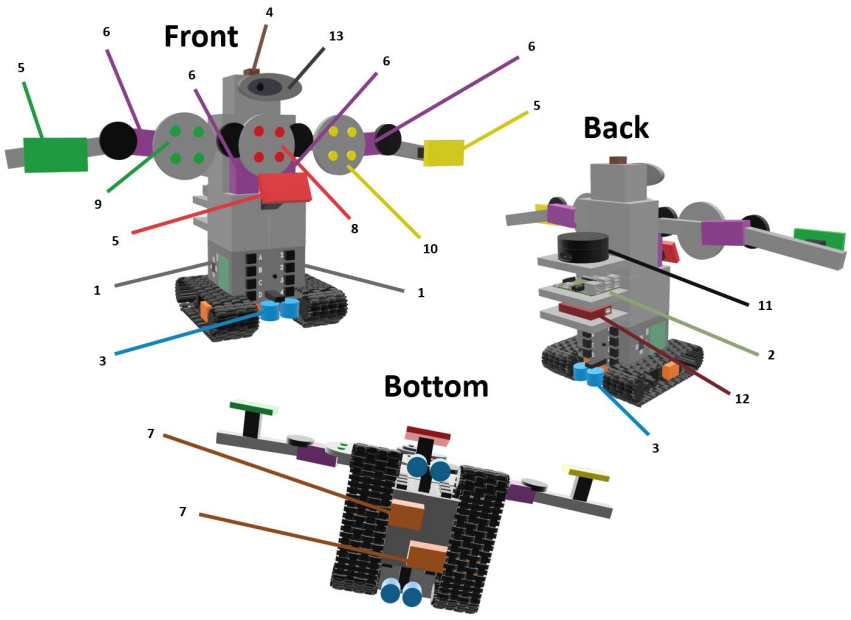


Figure 1: Leo Parts

Figure 1 offers three perspectives of how Leo looks from the front, back and the bottom. It also includes labels that show where Leo's hardware components can be found in relation to one another.

2.2 How components are used

Figure 1 shows the various components installed on our robot, Leonidas. There are two motors (7) at the bottom which are responsible for rotating Leo's caterpillar tracks, allowing Leo to move in any direction. The two ultrasonic sensors (3) located at the front and the back core, together with the gyro sensor (4) on Leo's head, prevent Leo from falling off his playing environment (table). The two EV3 bricks (1) receive data from sensors and send commands to the motors. The two motors (6) located behind the Red colored LEDs rotate Leo's left and right shoulders up or down during gaming. The motors (6) located behind the colored Green LEDs and colored Yellow LEDs on Leo's left and right arms, rotate Leo's left and right elbows. There are three touch sensors (5) attached to Leo; the green on the left, the red in the center, and the yellow on the right, which the user pushes while playing a game. On the top is the camera (13), which is used to help Leo face the user from an appropriate distance. On the back is the Raspberry Pi (2) which is used for connecting all other components together. The Echo dot (11) is used for Alexa support and the power bank (12) provides a power supply to the raspberry pi and the Echo dot.

3 Software overview

3.1 Software Architecture

Figure 2 explains the software architecture of Leo. Because we have 6 functional motors on our robot, we have used 2 EV3s, one for moving Leo's hands and playing user games, and the other for optimal positioning when the games are being played. Since both the EV3s can receive instructions from Alexa, the android app, and data from the web camera, we needed a central control unit for data flow and connecting the various components in our architecture. We decided that a Raspberry Pi would be optimal for our needs, optimal for our budget, and also physically small enough to easily fit on Leo.

We wanted to enable an easier and more pleasant gaming experience, and therefore we decided to add the Echo dot as the external speaker to play music during the games and the component that recognises speech commands. If a voice command is detected, the Echo dot will send the voice record to the Alexa Skills Service provided by Amazon. After the voice records are processed and interpreted to certain specific commands that can control Leo, Alexa Skills Service will send these commands to the Raspberry Pi to perform a task or execute a game accordingly.

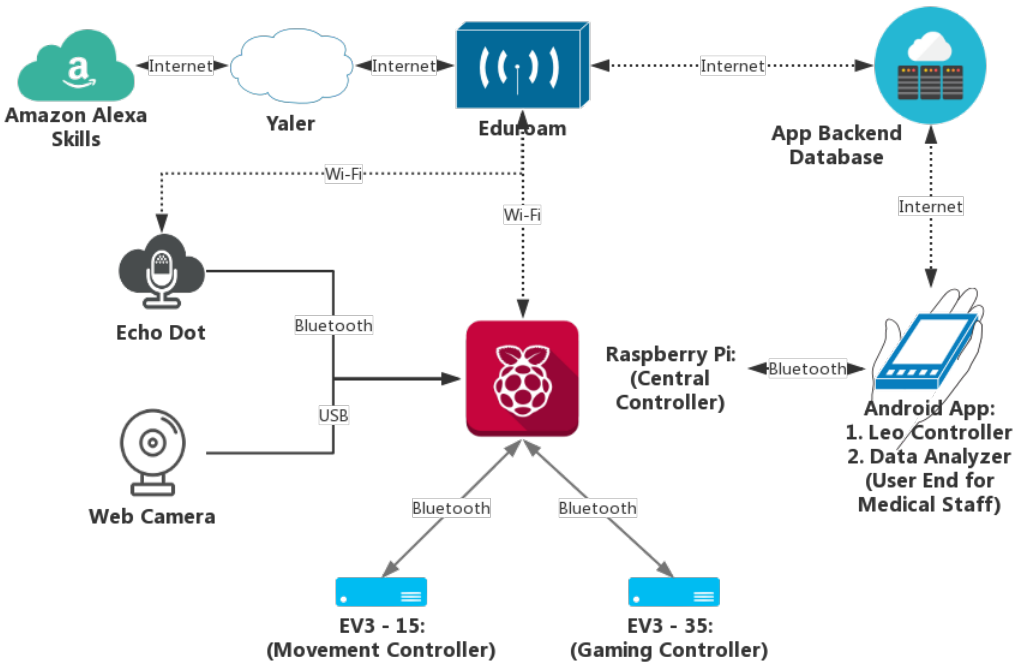


Figure 2: Leo's Architecture

4 Setup

Notice: Please follow the given steps in *order* to properly set Leo.

4.1 Turning on Leo

Step 1: Turn on the EV3s by pressing the middle button on them.
The EV3s' buttons should display a red light for a short amount of time while the operating system on them boots, and then these lights turn green indicating that the EV3s are on.

Step 2: Connect the Raspberry Pi to the power bank. Check that the Raspberry Pi lights up.

4.2 Bluetooth connection between EV3s and Raspberry Pi

The next step is connecting the EV3s to the Raspberry Pi via Bluetooth

Step 1:
(a) On an EV3, go to Wireless and Networks / Bluetooth.
(b) Make sure Powered and Visible are ticked.

Step 2:
(a) On the EV3, go to Wireless and Networks / All Network Connections.
(b) Search for "raspberrypi" and click the middle button on the EV3.
(c) Press Connect.

If the message "State: Connected" pops up and the EV3 displays an IP address, the EV3 has successfully connected to the Raspberry Pi.

4.3 Set up Echo Dot and Alexa Skills

Step 1: To set up the Echo dot on your mobile device, please download the Alexa app from the Apple Store or the Google Play store. If you are using a PC to set up the Echo dot, please visit https://alexa.amazon.co.uk.

Step 2: Plug in your Echo Dot and wait for it to initialize. If you want to reset the Echo, or the setup process didn't start properly, please press and hold the **Action** button until the light ring turns orange.

Step 3: Open the Alexa app/website and connect your mobile/PC to the Wi-Fi hotpot created by Echo Dot (The network name will be: Amazon-xxxx). Then, go to **Settings - Set up a new device** and follow the individual step instructions given by the Alexa app/website.

Step 4: After finishing the setup, please configure Echo dot as the external Bluetooth speaker on the **Settings** menu. To finish that, you need to connect the Echo Dot to Raspberry Pi via Bluetooth. You can find more information on Bluetooth connection in Section 5.1, or visit https://www.amazon.com/gp/help/customer/display.html?nodeId=202011820 alternatively.

Step 5: Open the Alexa app and go to the **Skills** menu. Alternatively, go to the Alexa Skills Store on the Amazon website https://amazon.co.uk/skills. Use the **Search** option to find a skill named **"Leo Control"** and select the **Enable Skill** option.

4.4 Making Leo ready to take commands

This section assumes that the two EV3s are connected via Bluetooth to the raspberry pi. (Please refer to Section 4.2). There are two ways of proceeding from here with the setup. If you you successfully completed steps in Section 4.3, we strongly recommend using following method to set Leo up (If needed, a more technical solution for setup is available in the Appendix A).

Step 1: Speaking **"Alexa, ask Leo to set up"** will automatically set up Leo for taking voice commands. After Leo responds with "Setup is done", you can start using voice commands like "Ask Leo to say hello".

Step 2: If you want to control Leo with your mobile phone:
(1) Make sure that your phone is paired to the raspberry pi via Bluetooth, and internet sharing is enabled under the options for pairing
(2) Saying **"Alexa, tell Leo to connect my phone"** will ensure that Leo is ready to be controlled by the mobile app introduced in Section 4.5. After Leo responds with "Connection is done", you can start controlling Leo from the app.

4.5 Setting up phone app

Step 1: Download the app from Github from the following link https://github.com/akshayCha/LeoApp by clicking on the **Clone or download** button and selecting **Download ZIP**.

Step 2: Unzip the file onto your desktop as a folder.

Step 3: On your android device, go to settings, then select **about device** and click on the **build number** 7 times to enable developer mode.

Step 4: Go back to settings, and under **developer options**, enable **USB Debugging**.

Step 5: Go to **security** under settings and enable **Unknown resources**.

Step 6: Download Android File Transfer from https://www.android.com/filetransfer/.

Step 7: Connect the android device using a USB to your computer, and make sure that the phone is connected as a media device (MTP). You can configure this by dragging down the notifications bar.

Step 8: Drag the apk file from the folder into the **Android** folder on the File Transfer screen showing you device storage.

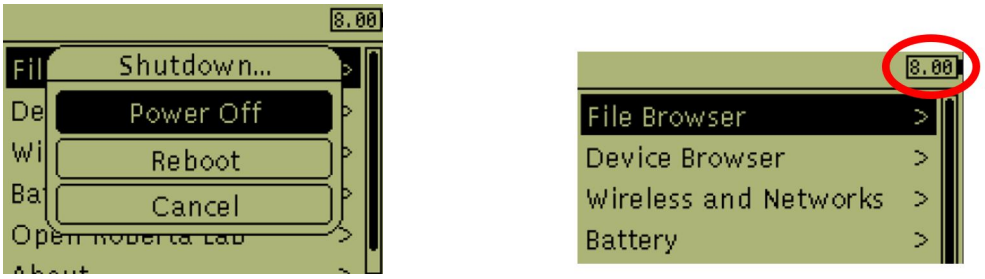
Step 9: Go to **My Files** on your phone and click on the app under the **Android** folder to install it onto your phone.

4.6 Turning Off Leo

In order to turn off Leo, you should shut down the EV3s and power off the raspberry pi. Follow the steps below to do that.

Step 1: Turn off the EV3s by pressing the back button from the main menu screen or by pressing and holding the back button from any screen. This will open a dialog as shown in Figure 3 (a). Select "Power Off" from the list and press Enter.

Step 2: Disconnect the raspberry pi from the power bank



(a) Turning Off the EV3s

(b) Battery Status

Figure 3: Operate EV3s

4.7 Battery Status and Charging

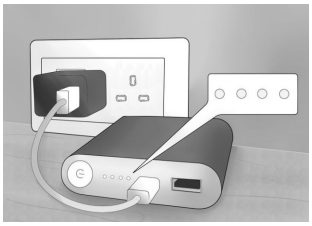
Leo involves 3 devices that need to be charged in order to make it operate normally: 2 EV3 bricks and the power bank that powers the raspberry pi and echo dot.

1. EV3

   - **Monitoring Battery Status on EV3 Bricks**: You can see the voltage in the upper right corner of the EV3's screen when it is turned on, as shown in Figure 3(b). The voltage ranges between 5V to 8V. If the voltage drops below 5V, then the EV3 brick will turn off and all unsaved progress will get lost. You should also consider that it will take a much longer time to discharge from 8V to 6.5V than from 6.5V down to 5V.

   - **Charging the EV3 Bricks**: Using one of the two adapter cords provided for each EV3, connect the EV3 to a wall outlet as shown in Figure 4(a). When you plug the EV3 into a wall socket, a red indicator light will flash. When charging is complete, the red light will go out and a green light will flash. The recharging process will generally take three to four hours. If you use Leo while the EV3s battery is charging, it will take longer.

(a) Charging EV3



(b) Charging Power Bank

Figure 4: Charge EV3 and Power Bank

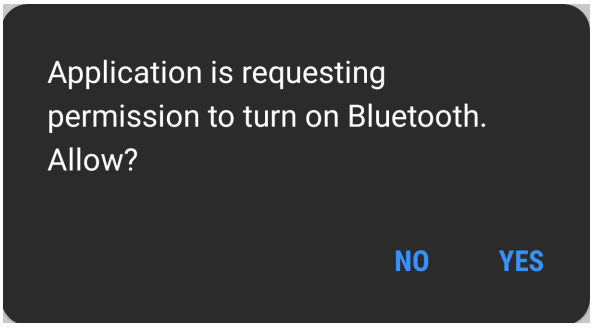
2. Power Bank

   - **Monitoring Battery Status on the Power Bank**: The battery levels on the power bank is indicated by the flashing LEDs on it. Four flashing LEDs indicate that the power bank is fully charged. If only one LED is flashing, the power bank needs to be charged.
   - **Charging the Power Bank**: Plug the provided A-Male to Mini-B cable into the input port of the Power Bank. Plug the other end into a standard USB port, which can be either a computer port or wall charger as shown in Figure 4(b). The input current for the power bank is adjustable according to the power source. The bigger the source, the faster it will recharge.

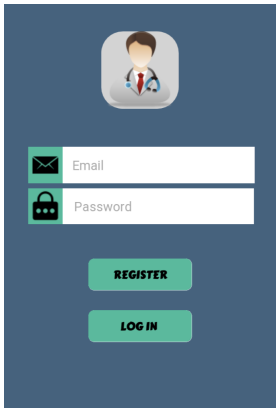
5 Using the app and playing with Leo

5.1 LeoApp

5.1.1 Connecting and registration



(a) Connect to Bluetooth



(b) User Login

Figure 5: UI for Connection and Login

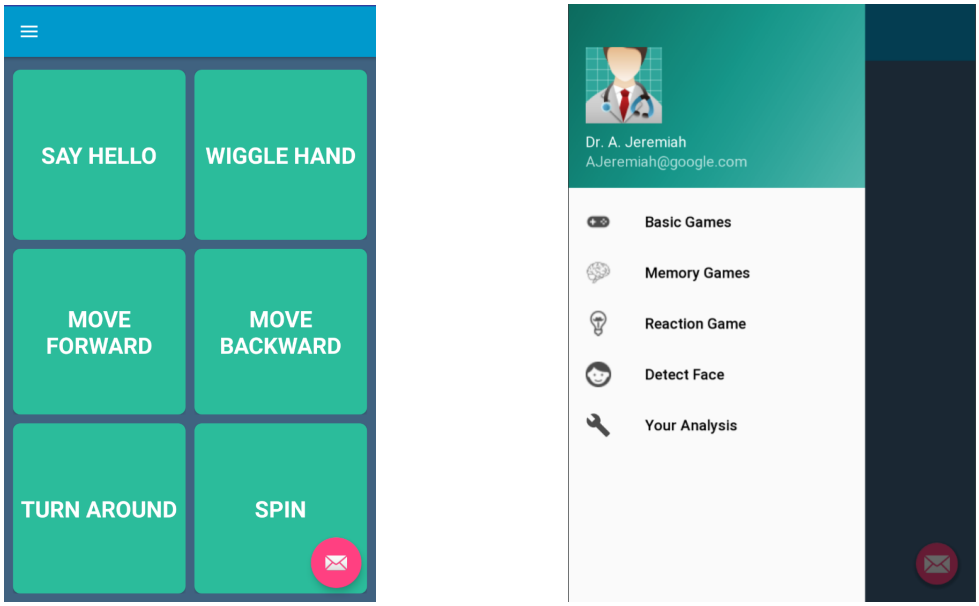
Step 1: After installing the app, you will see the app icon on the device screen.

Step 2: Make sure that your the Internet connection is switched off, you are connected to the raspberry pi using Bluetooth, and the Internet sharing is enabled under the options for the Bluetooth connection with the raspberry pi (Figure 5(a)).

Step 3: You will then be redirected to the login page, where you can login into the app using your previous credentials, or register yourself as a new user (Figure 5(b)). If you have any problem with this, please refer to section 7.

5.1.2 User Interface overview

- Logging in successfully will lead you to the page below, and you can click on the buttons to interact with the robot. (Figure 6(a))
- To switch to other menus, tap on the show slide bar option. (Figure 6(b))
- Tapping on a specific menu option will redirect you to another screen with different games for the user. You can also directly switch to different menus by swiping the screen left or right.



(a) Basic Layout (b) Slidebar

Figure 6: UI Explanation

5.1.3 Patient's result and getting information

- While the user is playing different games, various reaction times and scores that show up on the screen get saved to the backend (Figure 7).
- To access the users information, click on the send email option (the small red button at the bottom of the screen (Figure 6(a))and the current users results will get sent to your email.

5.1.4 Data Analysis in App

- The *Your Analysis* option compares the current users performance data to the general history of performance data for all users.
- It accordingly suggests an increase or decrease in the difficulty of the games being played

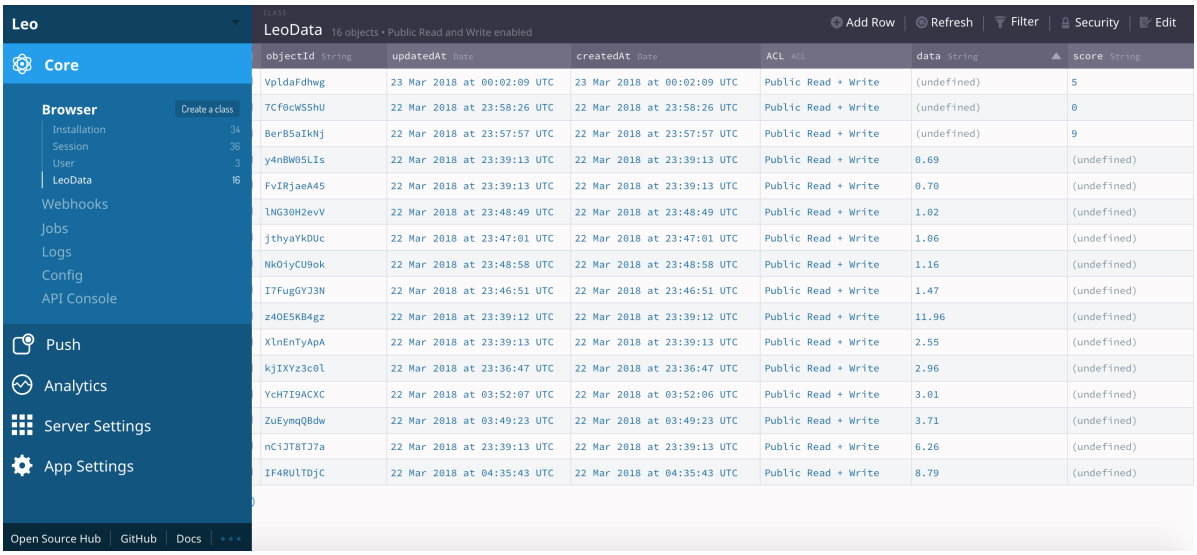


Figure 7: Performance data stored in the backend

5.2 Alexa commands

In order to interact with Leo using Alexa, you have to start your sentence with "Alexa, tell Leo to". Leo will process the following instructions:

1. "say hello" : Leo will wave his arms and say "Hello!"
2. "play reaction game X", Leo will start playing the reaction game numbered with X. If X is not a valid reaction game number (or not a number), Leo will say "Sorry, I don't know this game"
3. play memory game X", Leo will start playing the memory game numbered with X. If X is not a valid memory game number (or not a number), Leo will say "Sorry, I don't know this game"

5.3 Games

Recent studies indicate that cognitive activities delay onset of memory decline in persons who develop dementia[4]. With this fact in mind, we developed the following types of games:

5.3.1 Reaction Game

A reaction game consists of a sequence of movements done by Leo. At some point, Leo will stop, and the LEDs will indicate to the player which buttons should be pressed. The player has then to press the buttons corresponding to the LEDs which are lit. After the game ends, the reaction times are sent to the backend.

5.3.2 Memory Game

A memory game has two phases. In the first phase, Leo uses LEDs to indicate a pattern (for example red red green). The second phase consists of the player trying to reproduce the pattern shown by Leo by pressing the appropriate buttons (in this case pressing the red button once and the green button twice). The game ends when the player gets it wrong, or reaches the final stage and wins the game. The final user score is sent to the backend.

6 Troubleshooting

The bricks on the robot won't turn on!	• Have you made sure the bricks have enough charge? Try connecting the bricks to a power source and switching them on after some time.
The EV3 brick is not responding.	• Try resetting the brick by pressing the center button and back button for five seconds.
The raspberry pi won't connect and run.	• Is the raspberry pi connected to a power source? • Make sure that the Ethernet cable you're using to connect to the raspberry pi to your computer is not faulty.
I can't login into the app.	• Have you registered yourself as a user? • Are you using the right login credentials? • Does your mobile phone have the Internet connection? • If you still can't login, please contact the app administrator for details.
The app won't connect to the robot.	This involves a lot of stages of debugging in ascending order of things to check: • Has the raspberry pi been paired through Bluetooth to the EV3 bricks on the robot? • Has the mobile been paired to the raspberry pi through Bluetooth? • Have you run the command on your laptop to make Leo ready to take commands? • Have you turned off the Internet connection on your mobile phone after logging in and switched on the Internet sharing under options for pairing with the raspberry pi?
The face detection doesn't seem to work.	• Make sure that your background doesn't have a lot of objects in it. • Try going a bit closer to the robot and use a white background if possible.
I can't see my gaming data or user logins and sessions in the backend.	• Try refreshing the dashboard or logging in and logging out of your account.
Alexa does not respond to my commands/I can't hear any music while playing games.	• Is the Alexa connected to the raspberry pi? Try looking for details on the Alexa app on your mobile. There might be an error message concerning what went wrong. • Please check if you have set up the connection properly. Reboot the Alexa if necessary to establish a new connection.

References

[1] Jordan A.Mann, Bruce A.MacDonald, I.-Han Kuo, Xingyan Li, Elizabeth Broadbent *People respond better to robots than computer tablets delivering healthcare instructions.* https://www.sciencedirect.com/science/article/pii/S0747563214005524

[2] Marla Kosowicz, Sarah E.MacPherson *Improving multitasking assessment in healthy older adults using a prop-based version of the Breakfast task.* https://www.tandfonline.com/doi/full/10.1080/23279095.2015.1136310

[3] Hunter EM, Phillips LH, MacPherson SE. *Effects of age on cross-modal emotion perception.* https://www.ncbi.nlm.nih.gov/pubmed/21186914

[4] C B. Hall, R B. Lipton, M Sliwinski, M J. Katz, A. Derbyand, J Verghese *Cognitive activities delay onset of memory decline in persons who develop dementia.* https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2725932/

Appendices

Appendix A: Using an Ethernet cable to establish a connection from the app to the raspberry pi

Step 1: Connect your computer to the raspberry pi with the Ethernet cable provided in the package. If you don't have an Ethernet port on your computer, you can connect to the Raspberry Pi via Bluetooth by searching for "Bluetooth SSH" on https://www.raspberrypi.org/forums, or you an buy a USB to Ethernet Adapter.

Step 2: If you have a Secure Shell (SSH) Client on your computer, go to Step 4.

Step 3: **SSH Client on Windows:** Download and Install PuTTY from https://www.chiark.greenend.org.uk/ sgtatham/putty/latest.html.
SSH Client on Linux: If you are using a Debian based distribution, open a terminal and run "sudo apt-get install openssh-client". If you are using a Red Hat based distribution, open a terminal and run "yum install openssh-clients".

Step 4: Make an SSH connection with the Raspberry Pi by using hostname "pi@raspberrypi.local". The password is "sparta".

Step 5: Run the following commands:
cd SDP-2018/RaspberryPi/Communication
python initialiseScript.py.
The script should print out "EV3 15 ready" and "EV3 35 ready"

THE NEXT 2 STEPS SHOULD BE DONE EACH TIME A NEW PHONE NEEDS TO INTERACT WITH LEO

Step 6: Pair your phone to the Raspberry Pi with Bluetooth and enable internet access under options for this connection

Step 7: On the Raspberry Pi SSH, run the following commands:
SDP-2018/RaspberryPi/Communication
python3 mqttrpi.py.