THINGS TO COVER:

1. What you have - short intro to what the aim is
   1. What specific things you are given. This should also establish nomenclature we are using, e.g. Brick,, app EV3, Sensor Hub etc
   2. Describe how it is built and what is where
   3. Describe what software bits there are and what they are responsible for
2. What to do before the system can be used
   1. Turning on, connecting to a computer, setting up tethering, uploading files
   2. Sensor alignment, pointer attachment
   3. Downloading and installing the app (anything else - server?)
   4. What the environment is expected to be - white lines. Cross sections, markers, where paintings are, starting location, diagram of the map the robot uses
3. How to use it
   1. What to do each time??
   2. Run the program, make a tour, select paintings, fire up
   3. As above but for two users (i don’t know how it works)
   4. Assume a. Or b. Has been done, do some other stuff (I dunno)
   5. Describe what is possible as the tour is in progress (change speed, stop, toilet, etc). **Mention that you have to hit continue**
   6. Describe the meta-functions of the app - language selection, search, recommendations, speech processing etc
      1. A list of keywords?
   7. Press and hold the back button
4. What it should do - I think it is basically a description of what we consider nominal and correct behaviour. Perhaps made as a “story” of what the robot is doing
   1. Robot calculates route after the tour starts
   2. Robot follows line and detects branches, navigates accordingly
   3. On arrival, robot stops, points to painting, description plays to all users. Waits until continue is hit
   4. Robot detects the obstacle (how?), tries to avoid or stops (when?). Avoids by following the shape of the obstacle until it hits line again. If at junction, hits black. What happens then. What are the limitations of the obstacles (min size, has to touch the ground. Cannot have holes, where can be placed)
5. What to do if it doesn’t - we have it covered already, I think

EMILIA’S COMMENTS:

* While installing the app - get a way of verifying that the installation was successful
* The list of components is not obvious in its purpose
* The prerequisites for the systems are not clear are not obvious
* The role of the server is not clear
* Describe the environment - we never say anything on the lines etc
* Troubleshooting table - keep it on one page, don’t split
* We never say what the tour is - consider defining it

# 1 System Overview

RoboTour is a robotic tour guide that assists people in environments such as museums or art galleries. The system comprises of an autonomous robotic guide, a companion, purpose built Android application, and a web server mediating the communication between the two. RoboTour can be controlled by up to two Android devices, and the tour may be followed by many more. The app allows users to interact with RoboTour intuitively in multiple languages.

## 1.1 System Components and Prerequisites

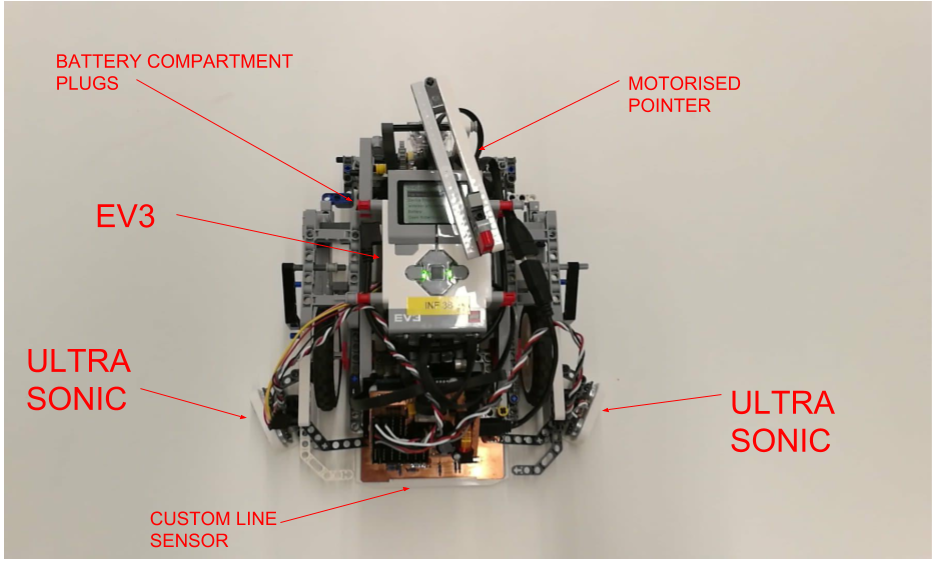
To follow this guide you’ll need:

* An Android device with an apk of at least 16, and an internet connection
* A computer with bluetooth and an internet connection.
* The built robot with all the necessary python files installed on the EV3 brick (this is provided for you)

## 1.2 Robot Structure

### 1.2.1 Robot

The robot provided to you should look like the robot in Fig 1. It consists of an EV3, a LEGO Ultrasonic sensor, LEGO colour sensors, 2 HC-SR04 Ultrasonic sensors, a Custom Line sensor, and an Arduino Sensor hub.



**Fig 1**: Labelled Top View of RoboTour

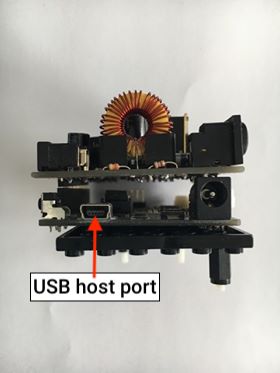
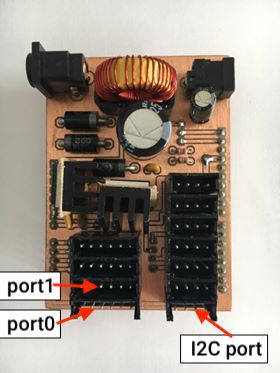
### 

### 1.2.2 Ports

Table 1 shows which EV3 port connects with which sensor or motor so make sure that these match up upon receiving the robot.

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Port** | **Element** | **Port** |
| Left drive motor | EV3 PORT B | Front ultrasonic sensor | EV2 PORT2 |
| Right drive motor | EV3 PORT D | Sensor hub | EV3 USB host port (Fig 10) |
| Pointer motor | EV3 PORT C | Left ultrasonic sensor | Sensor hub sensor port0 (Fig 9) |
| Left colour sensor | EV3 PORT 4 | Right ultrasonic sensor | Sensor hub sensor port1 (Fig 9) |
| Right colour sensor | EV3 PORT 1 | Front line sensor | Sensor hub I2C port (Fig 9) |

**Table 1**: Showing which EV3 port connects with each sensor / motor



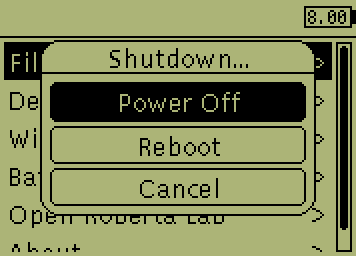
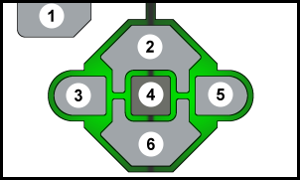
**Fig 9**: Top of the board **Fig 10**: Front of the board

### 1.2.3 Buttons

Figure 7 labels each button on the EV3 brick so from now on when referring to the EV3 buttons please consult Fig 7 (Ev3dev.org, 2018a).

The brick can be turned on by long pressing button 4 (Fig 7). It takes approximately 40 seconds from the point of pressing the button for the EV3 to start up.

To turn off, keep pressing button 1 (Fig 7) until Fig 8 appears on the screen,then press button 4 (Fig 7) and the EV3 will shutdown (Ev3dev.org, 2018b), this takes approximately 30 seconds to complete.



**Fig 7**: EV3 buttons (ev3dev 2018)  **Fig 8**: Exit menu triggered (ev3dev,2018)

### 1.2.4 Replacing and Recharging Batteries

To replace the battery first detach the EV3 Brick from the body of the robot. There are eight red safety pins holding the Brick in place. Pull them all one notch away from the brick. You should now be able to lift the brick from the chassis. To release the battery, press on the two latches on the battery pack underneath the Brick.

Insert a new battery pack making sure that the two restrictors feed into their housings. Apply moderate pressure until you hear the latches snap in place.

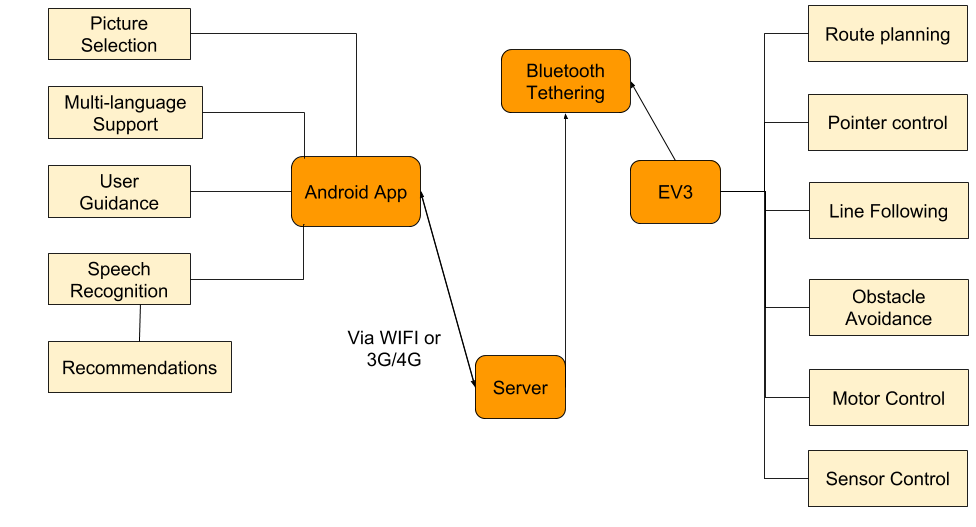
The batteries last approximately 40 minutes from a full charge, and they take approximately 2 hours to charge from 0 to 100%.

The battery can be recharged with the supplied 10V/700ma barrel plug charger. Connect the charger to the power supply, locate the charging port in the battery underneath the brick (accessible from the back of the robot). Green LED will indicate good connection. Red LED will shine until the battery is fully charged.

## 1.3 Software Structure

The robot and the Android app communicate with each other via a server which hosts a php file. Both the app and robot are able to read the server and send a post request to the server to update it. The php file used for the server can be accessed at this link. <http://proparoxytone-icing.000webhostapp.com/receiver.php>

The purpose of this is to allow for multiple users to communicate with the robot. Fig 6 shows the communication between the Android app and EV3 and the process each part does.



**Fig 6**: Communication structure between Android app and EV3

# 2 Setup and Preparation

## 2.1 Working Environment

The first step to setting up the robot is to recreate the map provided here, being careful to recreate the same lengths and use 25mm thick white and black tape as indicated by the map:

\*\*\*Provide Jaou’s Map / Preferably make our own\*\*\*

## 2.2 Robot Hardware

Once the map has been made set the robot facing north with the 2 EV3 colour sensors between the white line at the start position of the map as indicated by fig X (Insert image?)

## 2.3 Robot Software

Turn on the robot by following the steps of 1.2.3

Connect the EV3 to the internet by executing these instructions:

1. On the EV3, first verify that Bluetooth is powered on. In brickman, open the Wireless and Networks menu and select Bluetooth. Make sure the Powered checkbox is checked. Furthermore, make sure the Visible box is checked to make the EV3 discoverable.
2. Turn on Bluetooth on your Android device, then go to Tethering & Mobile Hotspot and turn on Bluetooth tethering, then go back to Settings and open Bluetooth and select your EV3 to pair it.
3. Confirm the passkey on both devices when requested.
4. On the EV3, find your Android in the list of Bluetooth devices and and select it.
5. Select Network Connection then select Connect.
6. State should change to Online. The Bluetooth icon will also indicate that you are connected and the IP address will be displayed at the top of the screen.

For further details how to connect with other devices, for bluetooth tethering, please visit see [Ev3dev.org (2018c)](http://www.ev3dev.org/docs/tutorials/connecting-to-the-internet-via-bluetooth/).

## 2.4 Android App

To download the app simply go to [homepages.inf.ed.ac.uk/s1553593/download.php](http://homepages.inf.ed.ac.uk/s1553593/download.php) and download the app apk onto your phone. Once the app is downloaded go to your Downloads app on your phone, find and tap on the file called “RoboTour-App.apk” and follow the installation instructions. Once installed the app will be in your App drawer under “RoboTour”.

Tap the app to open it.

# 3 Tour Guide

Begin by following the setup and preparation steps. At this point the robot should be on, in the correct position of the map, and connected to the internet and the app should be open on your Android device.

## 3.1 Starting the Tour (Robot)

Run the program from the brick. This is done by clicking on *File Browser -> main.py*.

The robot will then complete self-diagnosis and calibrate by rotating left and right. Once this is done the robot will make a short sound message.

Then press button 3 or 5 (Fig 7) to select Single or Multi User respectively.

Once you hear “Please select the paintings you want to go”, then you can start running the app and proceed to making selections for the paintings you wish to visit.

## 3.2 Starting the Tour (App)

Once 3.1 is completed click the “Start” button on your Android device.

The next screen (Fig 12) will display languages for you to select, if your language is not shown, select the “?”. The description of the art pieces, audio description and all the settings will be in your selected language (provided that your language is supported).

In Fig 13, you can select which art pieces you’d like to see by tapping on the images. You may also choose to search paintings or ask for recommendations via speech commands by tapping on the microphone or search bar.

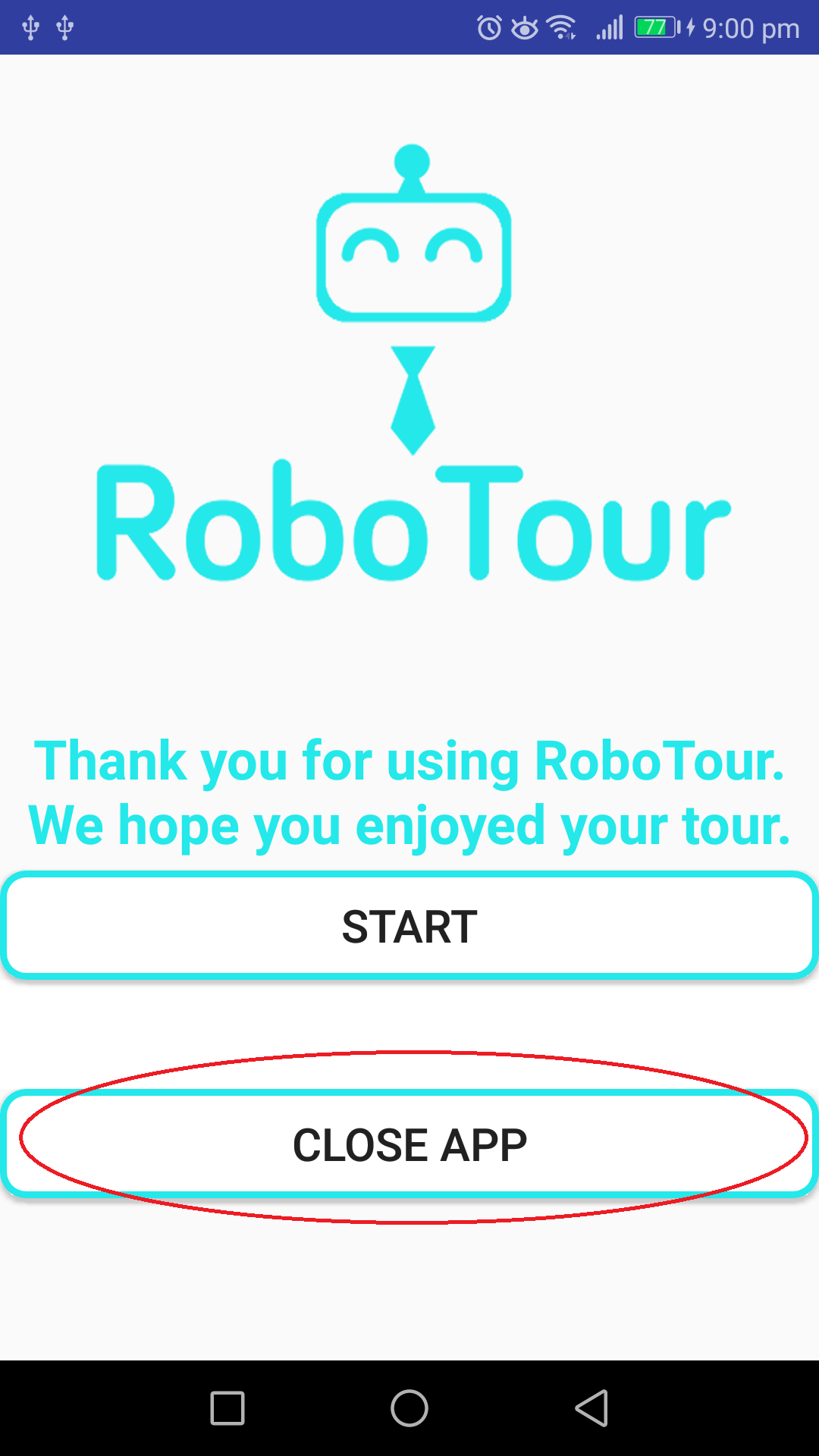
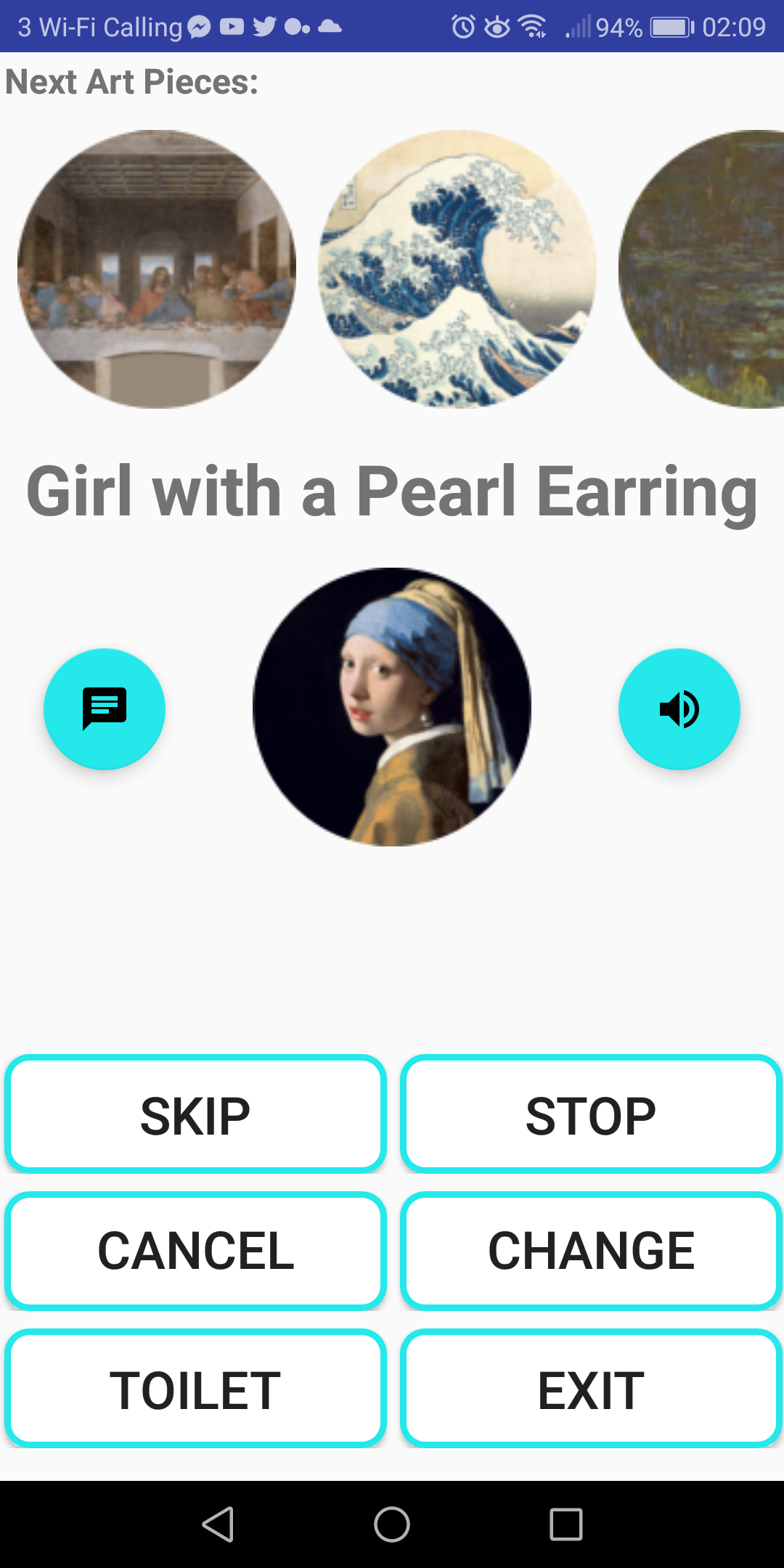
The searches that work with the app are any sentence containing a own substring of:

* Art Piece name
* Artist name
* “Best”
* “Popular”
* “Recommend”
* “New”

(In your selected language).

After all selections are made, press “Start Tour” and RoboTour will create an optimal route plan.

(In Multi User mode you’ll need to wait until the other user has made their selections and pressed “Start Tour” before you can continue.)



**Fig 12**: Language selection **Fig 13**: Painting Selection **Fig 14**: Navigation **Fig 15**: End

## 3.3 Following a Tour

After completing 3.1 & 3.2, the robot will now follow the white lines guiding you to all the paintings that were selected on the Android device(s). Once it reaches each painting it will stop, the motorised pointer will point at the painting and the app will use text-to-speech to talk about the painting. After the app has finished talking the robot will continue to the next closest painting using Dijkstra’s algorithm.

## 3.3 Robot Interactions During the Tour

When the robot is following a white line and detects an obstacle while it is in the inside perimeter (Fig X), the robot will just stop and not enter obstacle avoidance mode.

If it’s in the outside perimeter and detects an obstacle, it will enter obstacle avoidance mode:

1. The robot will first rotate 90 degrees in the direction towards the outward direction of the perimeter
2. Use the corresponding ultrasonic sensor to keep a certain distance between the obstacle and around the obstacle
3. The obstacle avoidance is completed once it returns to the white line.

## 3.4 App Interactions During the Tour

During the tour the app will be on Fig 14. Here you can select any of the buttons to execute the following commands:

* **SKIP**: Skip the current painting

(In Multi User mode an alert will pop up for the other user so that they can accept or cancel the decision).

* **STOP/CONTINUE**: Pause the robot, or continue moving towards the painting
* **CANCEL**: Cancel the tour
* **CHANGE SPEED**: Adjust the speed of the robot
* **TOILET**: Navigate to the toilet
* **EXIT**: Navigate to the exit

Selecting a picture from the carousel will bring up its description, its ETA, and the option to cancel going to that painting. Selecting the text floating action button will bring up the ETA and description of the current painting. Pressing the speaker button will use text to speech to read out the description in the voice of your chosen language.

## 3.5 Finishing the Tour

Once all the selected paintings have been visited the robot will return to the exit (the starting position). Once it reaches the exit it will turn around and is ready to be used again for a new tour. The app will then change to Fig 15 you can either start a new tour or close the app.

# 

# 5 Troubleshooting Guide

|  |  |  |
| --- | --- | --- |
| **ID** | **Problem** | **Solution** |
| 1 | Robot beeps once and does not start moving after starting main.py | Check if bluetooth tethering between EV3 and Android device is setup properly, and make sure the device is connected to the internet. |
| 2 | Robot beeps three times and does not start moving after starting main.py | Check if all sensors and motors are connected to appropriate ports. All disconnected devices will be shown on the terminal. Restart the program. If problem persists recharge the battery and try again. |
| 3 | Robot makes a series of short beeps or the front line sensor LED flashes. | Repeatedly press the reset button on the sensor hub and the reset button on the line sensor until the LED starts slowly fading on and off. |
| 4 | Robot fails to avoid an obstacle | Exit the program by pressing and holding the “Back” button on EV3. Hold it until the EV3 exits to the main menu. Move the robot to the starting position and restart the program. Make sure that the ultrasonic sensors are connected properly. Keep in mind that the robot can only avoid obstacles if it’s in the outside perimeter. |
| 5 | Robot does not follow the line properly | Make sure that the front line sensor is at approx. 4mm from the ground and parallel to it. Restart the program to allow recalibration of the sensor. During the calibration make sure that the entire sensor passes over both the line and the floor. |
| 6 | Robot stops at wrong markers | Make sure that the lego colour sensors are at approx 4mm from the ground. Restart the program. |
| 7 | Robot is turned on but not responsive to any commands | Force the EV3 to reset by removing the battery and placing it back in again. Turn on the EV3 brick. |

# 

# 