

ILR 02 – Progress Review 1

Dorothy Kirlew

Team Daedalus Members: Pranav Maheshwari, Richa Varma, Mohak Bhardwaj, and Shivam Gautam

October 23, 2015

1. Individual Progress

a. Android App

For the progress review on October 22, I worked on completing our App Development goal of creating a basic user interface through an Android app. To do so, I downloaded Android Studio and began using Udacity to familiarize myself with it. Although I still have much more to do, I was able to learn enough to add buttons and text to the screen, as well as modify the colors and font as seen in Figure 1: Basic GUI.

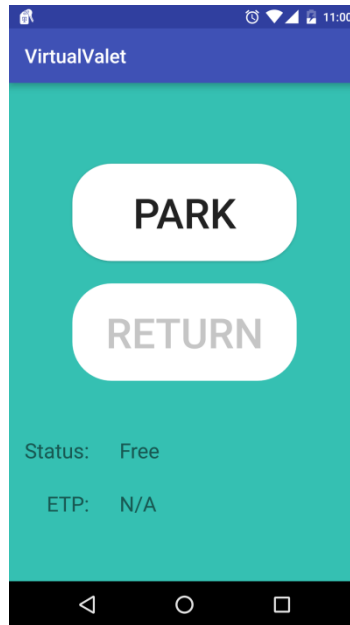


Figure 1: Basic GUI

I also created the following list of capabilities that the app should be able to do:

- Bluetooth
 - Laptop
 - From laptop to phone
 - From phone to laptop
 - App
 - From app to laptop
 - From laptop to app
 - Mobile platform
 - From app to mobile platform
 - From mobile platform to app
- Buttons
 - Buttons can be pressed
 - Buttons send commands to output
 - Buttons send commands via Bluetooth to phone
- App
 - Shows buttons

- Shows status
- Shows ETP and ETA (estimated time to park/arrival)
- Receives status via Bluetooth from mobile platform
- Updates display based on status
 - Buttons enabled/disabled
 - Status updated
 - ETP/ETA update as appropriate
 - Shows ETP or ETA
 - Shows time or N/A
- Time displayed appropriately
 - “1 minute and 30 seconds”, “1 second” (change units based on values)
 - App counts down or receives updates or both?
- Notifications
 - Notify user when car is parked
 - Notify user when car has arrived
 - Notify user of error
- Other
 - App is dummy-proof
 - Cannot press “return” when car is “free”, etc.
 - Create strict command format to and from app
 - Create unique vehicle/.app communication
 - App only talks to one, correct vehicle
 - Vehicle only talks to one, correct app
 - Perhaps create way to introduce phone to car?
- Possible bells and whistles:
 - Phone can communicate to multiple cars from drop down
 - Car can receive from multiple phones (after appropriate handshake). Good for multiple owners
 - Maybe some kind of security?
 - Stats: average time to park, etc.

b. Conceptual Design Review (CoDR)

I worked with Pranav to modify the CoDR as per the suggestions given in the feedback. We also worked together to extensively update the website with information gained from the CoDR updates and from the research studies performed by other team members.

2. Challenges

My only challenge for this progress review was familiarizing myself with Arduino Studio. I have never worked with an app. There are many aspects that are new to me, such as creating separate formats for buttons, instead of formatting them explicitly in their definition.

Additionally, Mohak and I are again facing resource challenges. Android Studio needs an update in order to work on his laptop, so any changes to the app will be made on my laptop for the time being. Additionally, I do not have an Android phone, which means that any app testing will be done on his phone.

3. Teamwork

We divided the work amongst our five team members as evenly as possible, with respect to our experience and skills. Shivam researched communication options and discovered VANET – Vehicle Ad Hoc Network, which can be used for Point-to-Point communication between vehicles. Pranav studied vision based obstacle detection with Mohak and they decided to use Kinect for the vision system. Richa worked on integrating the DFRobot 4WD Platform with the Arduino Mega 2560. Mohak and I collaborated on creating the app. Mohak also researched Bluetooth communication and how to implement Bluetooth communication between an Android app and a laptop, as well as between an Android app and a mobile platform.

4. Plans

The team has five basic goals to complete by Progress Review 2:

1. Select and place order for the first set of communication equipment.
2. Request Kinect from inventory and start implementation of obstacle detection.
3. Interface the Arduino Mega as a node with ROS
4. Show progress regarding backend functionality for the android app.
5. Follow up with sponsor regarding mobile platforms and present progress regarding it.

5. GUI Code

a. activity_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity"
    android:background="#35c0b2">

    <Button
        android:layout_width="230dp"
        android:layout_height="100dp"
        android:textSize="50sp"
        android:background = "@drawable/unpressed_button"
        android:text="Park"
        android:id="@+id/button"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="67dp"
        android:clickable="true" />
```

```

<Button
    android:layout_width="230dp"
    android:layout_height="100dp"
    android:textSize="50sp"
    android:background = "@drawable/pressed_button"
    android:text="Return"
    android:id="@+id/button2"
    android:enabled="true"
    android:layout_centerVertical="true"
    android:layout_alignStart="@+id/button"
    android:clickable="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="25sp"
    android:text="Status:"
    android:id="@+id/label1Text"
    android:layout_marginTop="57dp"
    android:layout_below="@+id/button2"
    android:layout_alignParentStart="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="25sp"
    android:text="Free"
    android:id="@+id/statusText"
    android:layout_marginStart="31dp"
    android:layout_alignTop="@+id/label1Text"
    android:layout_toEndOf="@+id/label1Text" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="25sp"
    android:text="ETP:"
    android:id="@+id/label2Text"
    android:layout_below="@+id/label1Text"
    android:layout_toStartOf="@+id/statusText"
    android:layout_marginTop="26dp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="25sp"
    android:text="N/A"
    android:id="@+id/ETText"
    android:layout_alignTop="@+id/label2Text"
    android:layout_alignStart="@+id/statusText" />

</RelativeLayout>

```

b. button_states.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

```

```
<item android:state_pressed="false"
      android:drawable="@drawable/unpressed_button"/>

<item android:state_pressed="true"
      android:drawable="@drawable/pressed_button"/>

</selector>
```

c. pressed_button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
      android:shape="rectangle">
  <corners android:radius="40dp"/>
  <gradient android:startColor="#FFFFFF" android:endColor="#FFFFFF" />
</shape>
```

d. unpressed_button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
      android:shape="rectangle">
  <corners android:radius="40dp"/>
  <gradient android:startColor="#FFFFFF" android:endColor="#FFFFFF" />
</shape>
```