The Field Automated Routing Module (FARM)

David Sheppard

Table of Contents

Background and Motivation

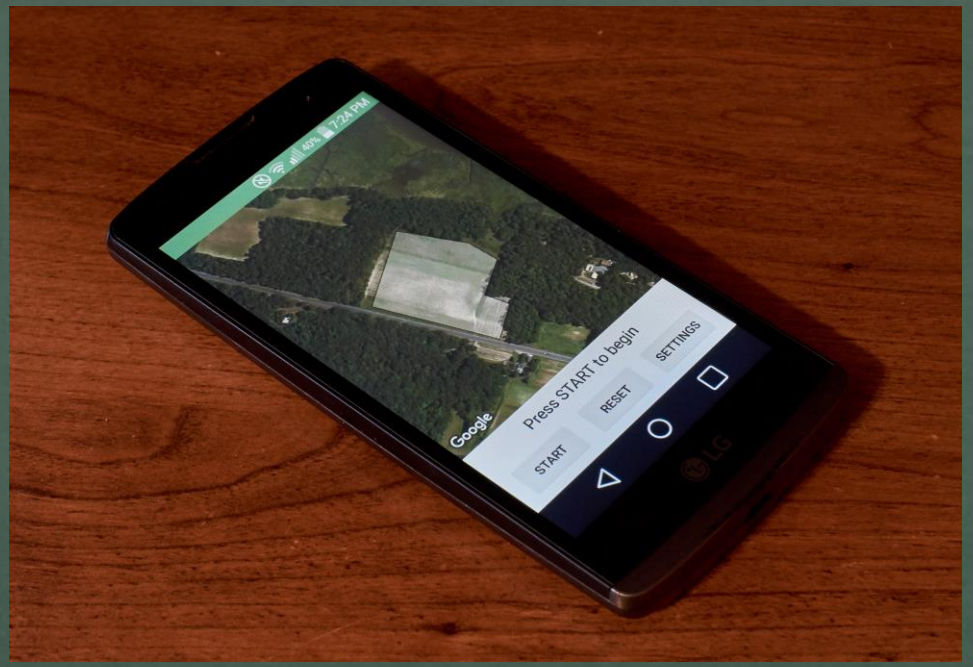
- It can often be challenging for farmers to keep track of their progress when fertilizing or planting a field. If they miss a spot, they are wasting part of their field. It they overlap rows, they are wasting valuable fertilizer or seeds.
- Existing (but problematic) solutions include:
  - GPS devices and software designed for agriculture
  - Mobile phone applications

The Goals

- The main goals of the project were to create an Android app that could:
  - Track the user as they travel across the field by plotting their path on a map
  - Allow their path to have an adjustable width so that the width of the path matches the width of the area covered by the tractor/sprayer/plow/etc.
  - Allow the user to define the boundaries of their field to make it easier to track progress

Work Done

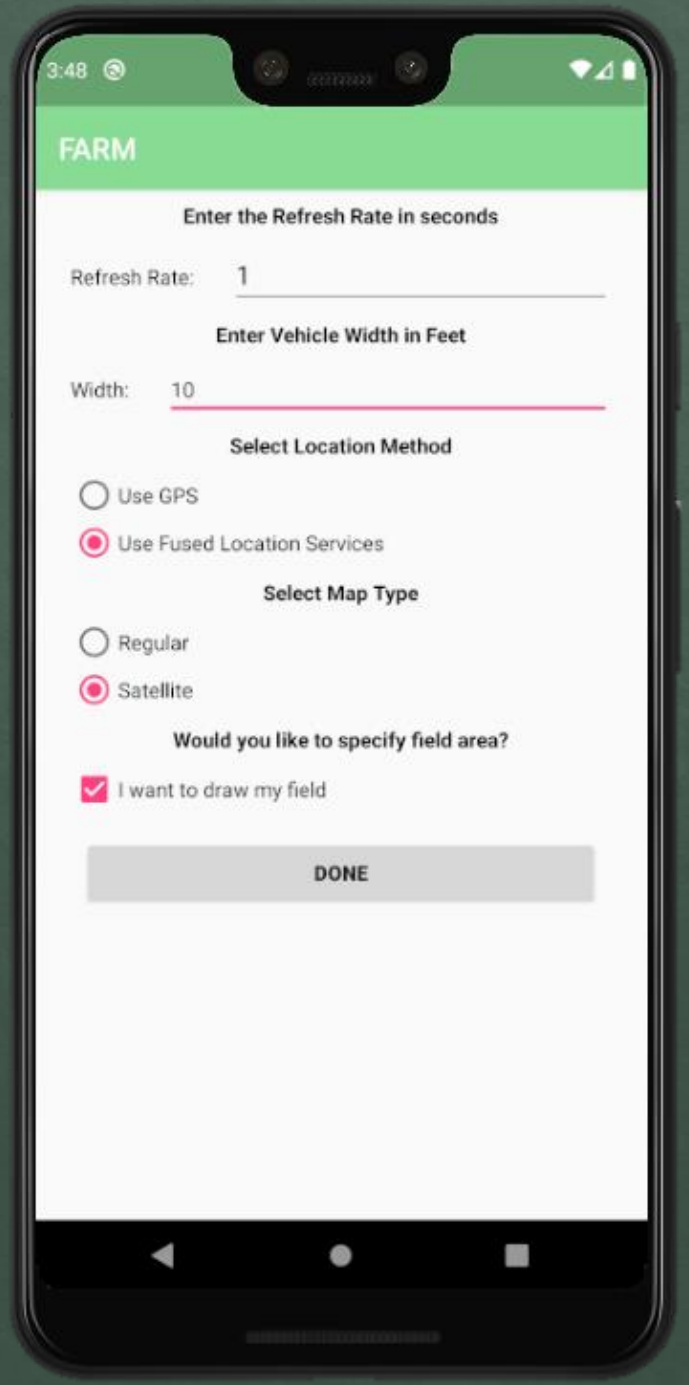
- The app has been developed and tested on an Android smartphone.
- Around 1200 lines of Java code were written with the Android Studio IDE
- Testing (outdoors and simulated)
- The app was made compatible with Android 5.0 (“Lollipop”) and above
  - Android Lollipop came out in 2014, and about 90% of Android users are running this operating system or a later one
    - source: https://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/




```
610
611     public void getPermissionsFINE(){   //gets permission to ACCESS_FINE_LOCATION and requests location updates
612         //if at least Marshmallow, need to ask user's permission to get GPS data
613         if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
614             //if permission is not yet granted, ask for it
615             if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
616                 requestPermissions(new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, requestCode: 0);
617                 if(ContextCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
618                     //if permission still not granted, tell user app will not work without it
619                     if(myToast != null) myToast.cancel();
620                     myToast = Toast.makeText( context: MapsActivity.this, text: "Need GPS permissions for app to function", Toast.LENGTH_SHORT);
621                     myToast.show();
622                 }
623                 else{
624                     locationManager.requestLocationUpdates(interval, minDistance: 0, locationCriteria, locationListener, looper: null);
625                     locationPermissionGranted = true;
626                 }
627             }
628             else {
629                 locationManager.requestLocationUpdates(interval, minDistance: 0, locationCriteria, locationListener, looper: null);
630                 locationPermissionGranted = true;
631             }
632
633         }  //else if below Marshmallow, we don't need to ask special permission
634         else if(ContextCompat.checkSelfPermission( context: this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED){
635             assert locationManager != null;
636             locationManager.requestLocationUpdates(interval, minDistance: 0, locationCriteria, locationListener, looper: null);
637             locationPermissionGranted = true;
638         }
639         else{
640             //if permission still not granted, tell user app will not work without it
641             if(myToast != null) myToast.cancel();
642             myToast = Toast.makeText( context: MapsActivity.this, text: "Need GPS permissions for app to function", Toast.LENGTH_SHORT);
643             myToast.show();
644         }
645     }
```

Functionality

- When the app first starts up, the user can choose their desired settings.
- The app allows the user to choose:
  - Location refresh rate
  - Path width (in feet)
  - Location update method
    - GPS (uses traditional GPS satellite signals)
    - Fused Location (uses a combination of GPS, cell towers, and Wi-Fi signals)
  - Map type (regular or satellite view)
  - The area of their field



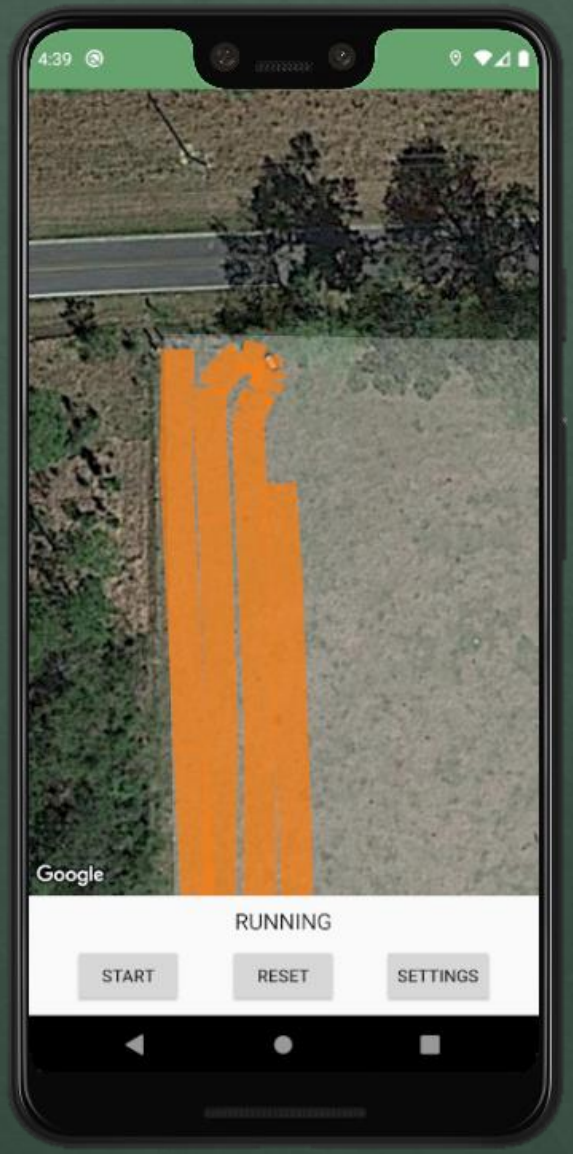
Functionality

- If the user wants to define the boundaries of their field, they can do so by simply tapping the edges of the field.
- A marker is added to each spot the user taps. A polygon is automatically drawn with each marker serving as a point on the polygon.
- If the user wants to undo the placement of a marker, they can tap the “undo” button.
- The polygon is shaded white and is 75% transparent so that the user can still see the map underneath the polygon.
- The area of the field (in acres) is shown in the text box at the bottom of the screen.

Functionality

- After selecting the area of their field, the user is taken back to the main screen where the field is now highlighted on the map and the red markers are removed.
- Once they press start, the user's location is tracked, and their path is plotted on the map.
- Their path is plotted as an orange polygon with 25% transparency

Images of Plotted Path



Possible Future Improvements

- While the app is currently fully functional, other features could be added to improve the user's experience:
  - Ability to track acreage covered in real time
  - Ability for user to set polygon colors and transparency values
  - Ability for user to save fields that they define for later use

Summary

- An Android application was developed that can track a user's path as they travel around a field or other area.
- The user can predefine the area of their field and the width of their tractor.
- The app has not shown to suffer from lag like other available apps
- An app like this can help farmers to ensure that they to do over-fertilizer or under-fertilize any parts of their field. This can help them to be more efficient and can save money.
- This application is open-source. The repository can be found at https://github.com/sheppardd1/FARM