



# ANDROID

# DEV

Geo-Paint

ANDROID

Being able to harness and react to a device's location is what makes Android apps different than "regular" computers, and is arguably the entire point of developing them ! For this project, I am building

An app that will

**uses the device's  
*physical location* as  
input for a simple  
drawing program.**



# What are the User Stories?

The user stories of the YAMA app are:

As a user, I want to...

- Be able to draw a picture mirroring my movements
- See my drawing displayed on a map
- Draw in different colors
- Save my drawing and share it with others

# Location and Google Maps API

- The app uses the device's Location Service for receiving and parsing information about the user's current location (exact latitude and longitude).
- Google Map API is used for displaying an interactive map, as well as the "painting canvas" for the app.



Google Maps

# Third-Party Libraries

- To leverage the work of implementing a color picker for the paint brush, I used the third-party package ColorPicker created by someone else!
- The package is loaded in the app's build.gradle and initiated in the app's main activity, displaying as a pop-up dialog when called.

Library Documentation: <https://github.com/xdtianyu/ColorPicker>



# Saving and Sharing the Drawing

→ An essential feature of Geo-Paint is to save and share the drawing.

Here's how it works:

→ The drawing is saved as a set of coordinates in a **JSON file** and stored **privately on External Storage**, so that the file world-readable and sharable.

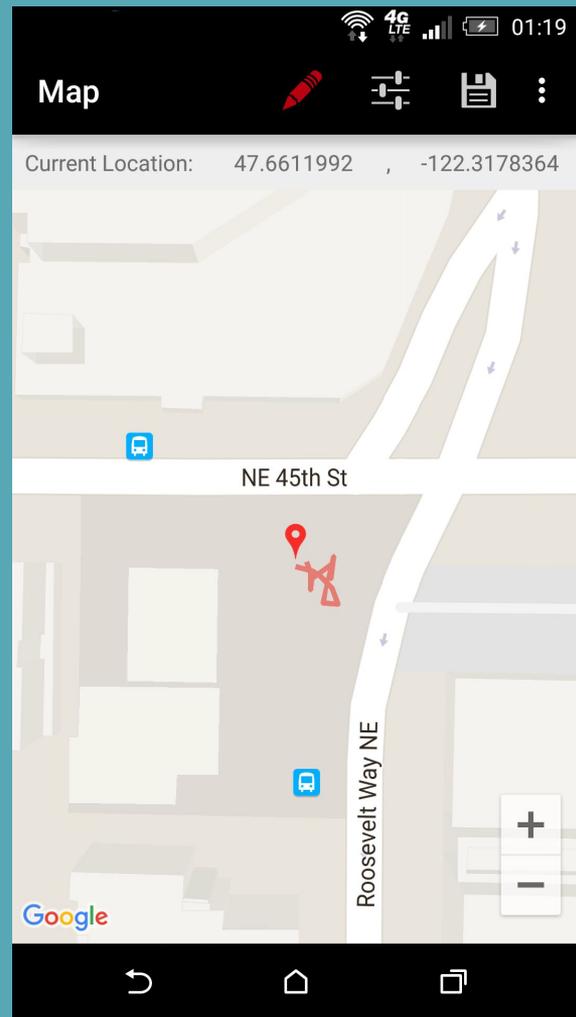
◆ 

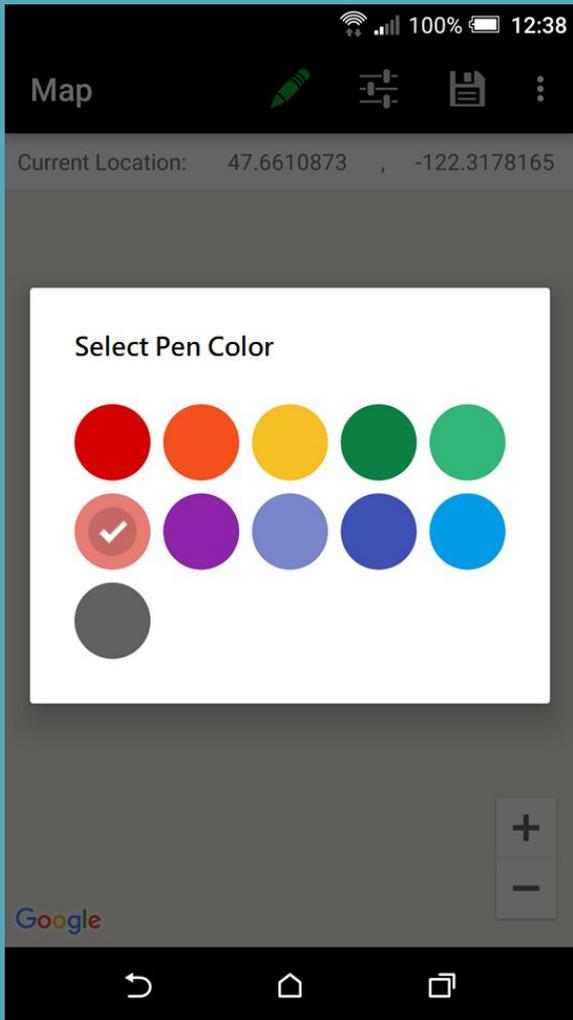
```
File file = new File(this.getExternalFilesDir(null), "drawing.geojson");  
FileOutputStream outputStream = new FileOutputStream(file);  
outputStream.write(string.getBytes()); //write the string to the file  
outputStream.close(); //close the stream
```

→ The drawing file is shared through an **Action Provider**, a handy widget which allows the user to select a channel to share the JSON file.

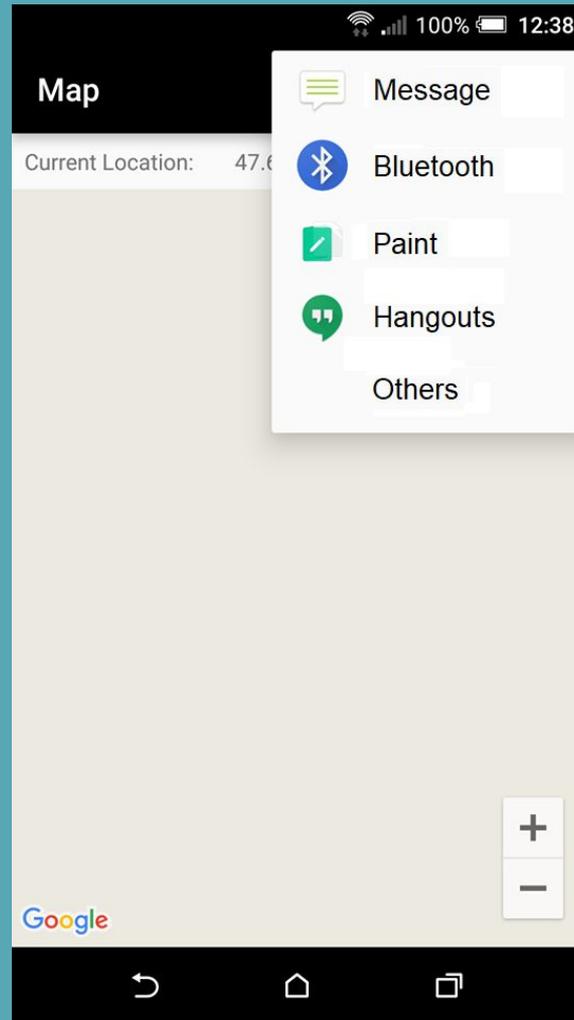
# Drawing on Map

- The device's precise current location is displayed at the top.
- The pink path is an example of the drawing path, which follows the red location icon.
- The path is drawn only when the pen is down (pen icon in red), otherwise the icon will only be moving with the device on the map when the pen is in non-drawing state (pen icon in green)





The third-party color picker for adjusting the color of the drawn path.



Geo-Paint uses the Action Provider for calling different sharing channels.

# What Skills have I demonstrated?

Through completing this project, I've demonstrated the following skills:

- **Accessing location sensors** (e.g., GPS) found on mobile devices
- **Integrating Google Play Services** and displaying interactive maps
- **Handling run-time permission requests**
- Creating and using menus
- Loading and using packages and **third-party libraries in Android**
- **Saving files and sharing** them with other apps

# How might the app move forward in the future?

Next-step features to improve usability for the Geo-Paint may include:

- Loading a previously saved or downloaded JSON file
- Analyzing drawing patterns (i.e distance vs. time) to send push notification/suggestion on an approximate time to leave
- Using the tracked pattern to suggest least time consuming or shortest route



**THANKS!**

ANDROID

**ANDROID DEV**  
Geo-Paint