

# Multi-touch Tester Android Application

Milen Georgiev

mg00929@surrey.ac.uk

## 1. Purpose

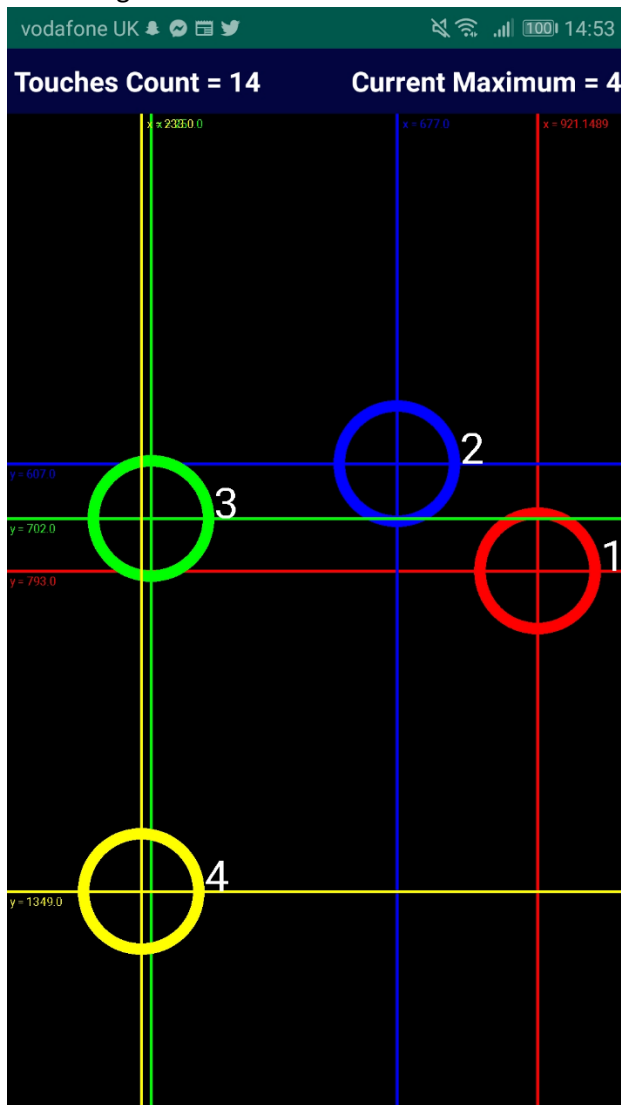
The purpose of this Android application is to enable the user to test the largest amount of screen touches his android device can register simultaneously. Additionally, it shows the largest amount, which has ever been achieved since installing the app and the amount, which has been achieved in the current use of the application. Moreover, the application is able to display all the touches made (only the first touches - without the movements on the screen) while testing the multitouch of the device.

The application primarily aim is for screen diagnostics and can be used by people, who need to check the maximum multitouch amount of their display. Such tests in reality may be needed when the screen of a device is changed (after breaking it or another reason).

## 2. Functionalities and how to use

### Multitouch Test Feature:

The first and primary feature of my application is to test the maximum touches the device's display can register simultaneously. It is done by clicking the button for starting the multitouch testing. After that the user must try to touch the display on as many places as possible simultaneously and see the result for his screen. The activity for testing looks like:



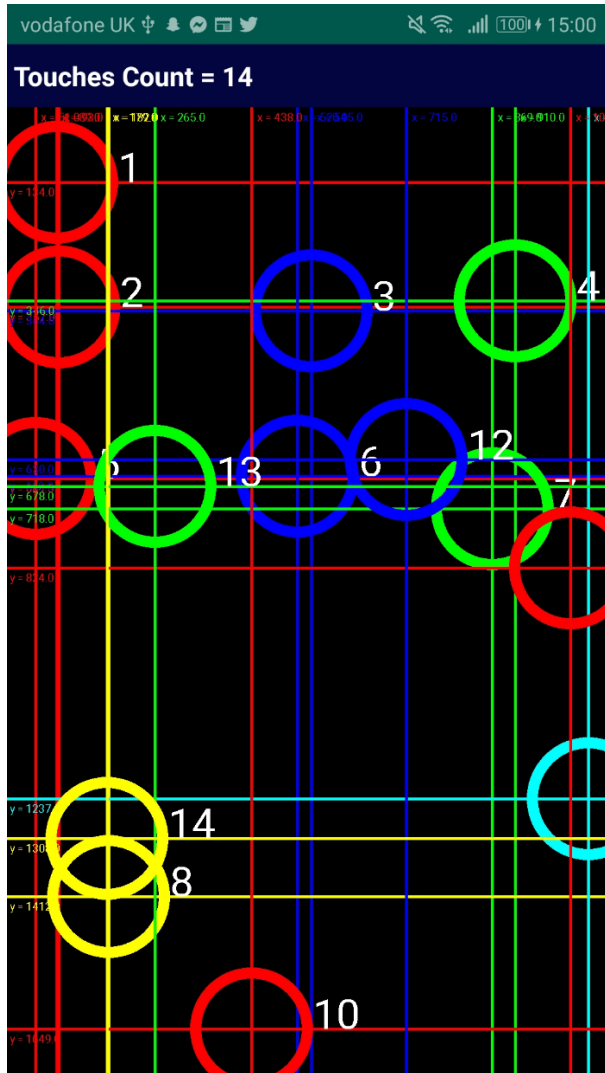
Screenshot 1: Multitouch Test Activity

Information about the current maximum detected in this testing session and the total touches made since installing the application is displayed on the top of the application.

### Display All Touches Since Installing the Application Feature:

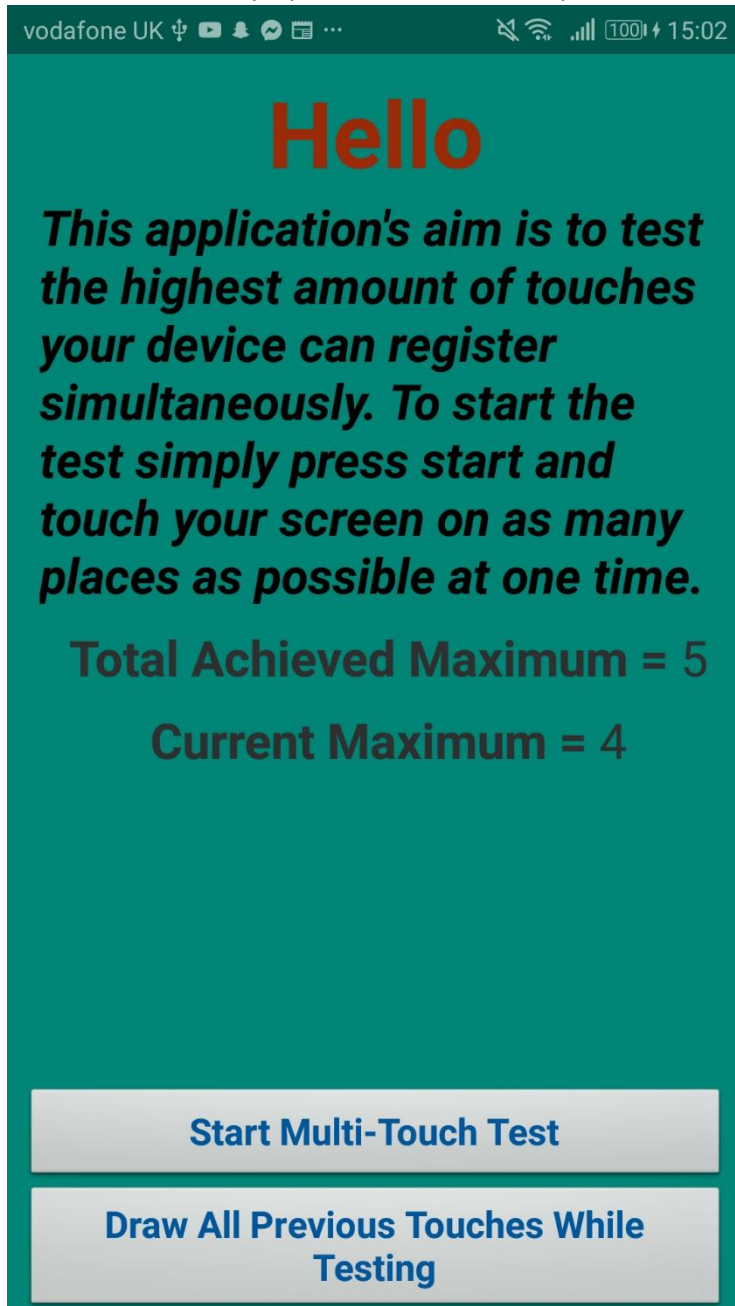
To try that feature the user must select it in the initial menu clicking the button for drawing all previous touches. After that the application will simply draw all the touches in the same way as they were initially made with the same colour and on the same places on the screen. Additionally, information about the total count of the touches made since the installing of the application while testing will be displayed on top of the application.

The activity will look like this:



Screenshot 2: Display all touches activity

Additionally, information about the total maximum ever achieve in the application and the current achieved maximum will be displayed in the initial activity like:



Screenshot 3: Main Activity

### 3. App navigation

The navigation in the application is made by clicking the 2 buttons in the main activity. They are shown in the picture above (Screenshot 3). This activity has a lot of text, explaining the purpose of the application to the user. Consequently, if the display of the device on which it is used is too small, the text, including the other information (except the button), will be scrollable, so that no information is unreachable.

If the button for starting Multi-touch Testing is clicked, this will start the activity shown on Screenshot 1. There the user will be able to test the multitouch of his device. Information about everything needed there is on top of that view. Unfortunately, I removed the action bar, because it is more useful in this app to be able to touch on as much of an area as possible and therefore there is no parent navigation using the action bar (I implemented it, but it cannot be established, due to the lack of action bar, however if I return the action bar parent navigation

will be possible). Therefore, only pressing the back button can return the user to the initial activity. If the display is rotated everything will be back in place not only in this activity, but on all activities.

If the button for Drawing all previous touches is clicked, this will start the activity shown on Screenshot 2. On that activity, all touches on the screen while testing the application will be drawn on the screen with the right colour and number and on the exact position of the screen. Again, as the previous one this activity does not offer parent navigation for the same reason. Going to the main activity is established only by pressing the back button.

This application does not offer much navigation, because I wanted to implement a real application doing a real job and not just mocking anything. Therefore, most of the effort I put was in the functionality of the application.

## 4. Using of threads, SQLite and touchscreen sensor

### Threads:

All the threads (I used AsyncTask for them) I have implemented in the application are for using the database. One of them puts a new entry of a touch in the database with its coordinates and colour when new touch is made in the activity in Screenshot 1. Another one retrieves all the entries for them to be drawn in the activity in Screenshot 2. Another one changes the maximum achieved simultaneous touches since installing the app and another retrieves it for displaying it on the UI. There is also one AsyncTask, which retrieves the total number of touches (entries in the touches table). All AsyncTasks which retrieve information either return it by storing it on List and Map or by printing the information on a specific view in the method executed after the task is finished (onPostExecute).

All threads are safely using the database, because I implemented locking in the database helper class with a semaphore with 1 permit. Each time a database connection is established the semaphore is acquired, and when the helper is closed the semaphore is released.

### SQLite:

For using the SQLite platform for the database of the application I have implemented a database contract with all the names of tables and columns in the database and a thread-safe database helper for instantiating a connection with the database.

The database is very simple, all it stores are the coordinates of touches and their colour in one table and the maximum achieved touches ever detected by the device in another table with only 1 row. The 2 tables have no relationship. The relational database schema of the database is as follows:

Table	Columns
last_touches_coordinates	int primary key: Id; real x_coordinate; real y_coordinate; integer colour
max_touches_count	int primary key: Id; int count;

### Touchscreen sensor:

The use of touch screen sensor is established by using the onTouchEvent of the custom view MultiTouchTestView. In that method all the coordinates of the different touch pointers are extracted using the getX and getY methods with a parameter the specific touch pointer of the MotionEvent. The only check of the action of the event is to check whether the last pointer is out of the screen and if so to remove the its coordinates, because the onTouchDisplay method is triggered even if a touch is removed from the screen. Everything else in the method I have implemented is to check whether the count of touches is greater than the

largest recorded and if so to change it with the current. Also, if a new touch pointer is added then its coordinates must be added to the database as it represents a new touch.