Lab 1 Report

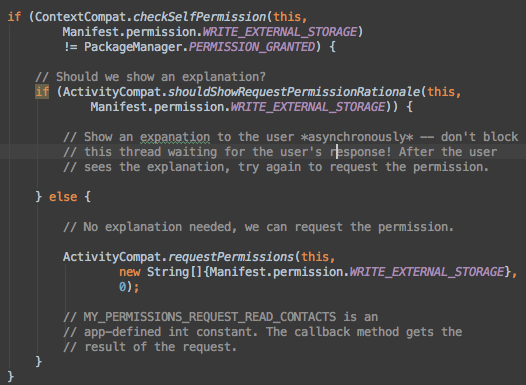
Group 7

1. Code structure and explanation

Main code is included in MainActivity.java. And methods are listed below. Main content of each method is shown meanwhile.

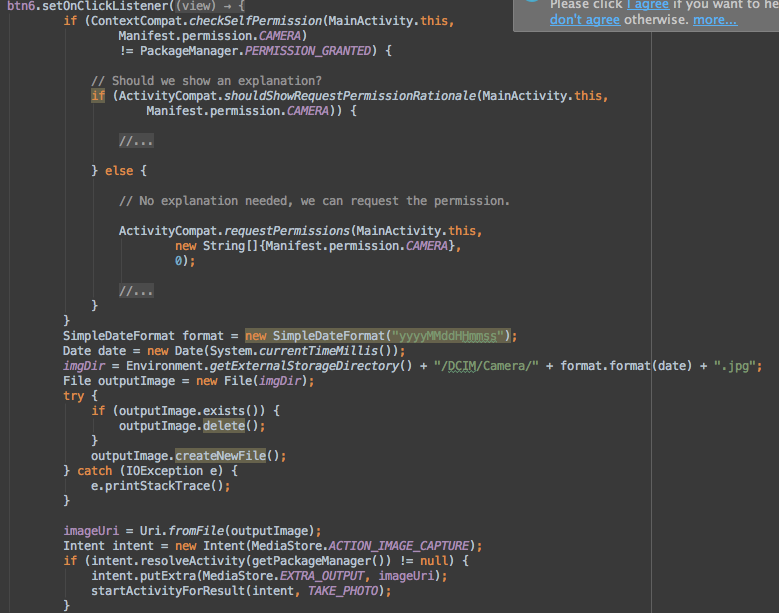
# Since all records will be written in the same file, all these arguments which store the data are set global.

1. onCreate()

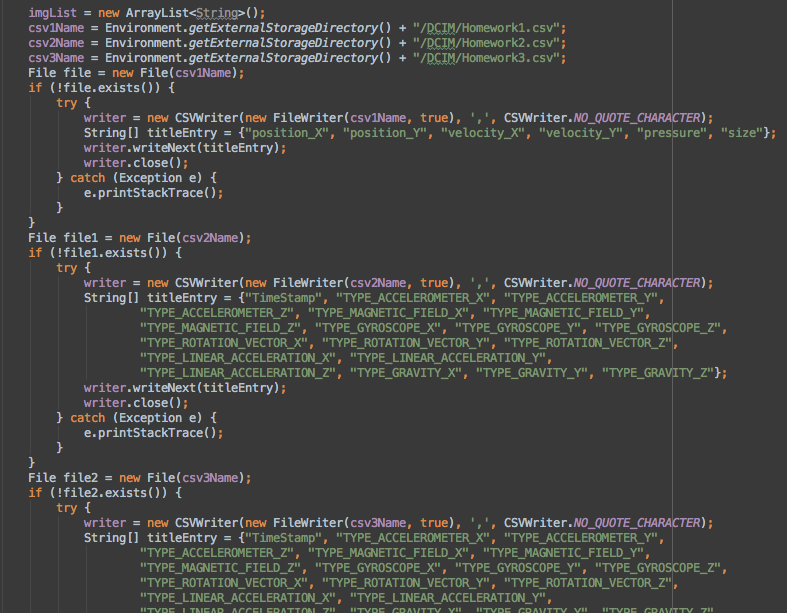
Check permission and get permission of WRITE\_EXTERNAL\_STORAGE & READ\_EXTERNAL\_STORAGE.

Declare buttons and imageView, connecting them with widgets in main UI. Set click listener to each of them.

Initialize camera button and set onclickListener. Set paths of new photos and call the onActivityResult() by static value TAKE\_PHOTO.



Initialize imgList which is used for storing image to check whether the image has been rated. Create new .csv file (homework1,2,3) and add titles to them.



Create sensor manager and all 6 different sensors. Register listeners (mySensorListener) to them. The function mySensorListener is put outside onCreate().

Initialize 6 3-size String arrays to store the sensors’ data.



Ps: onCreate() should be refactored cause it has 200+ lines of code and some code can be moved outside such as creating of .csv files.

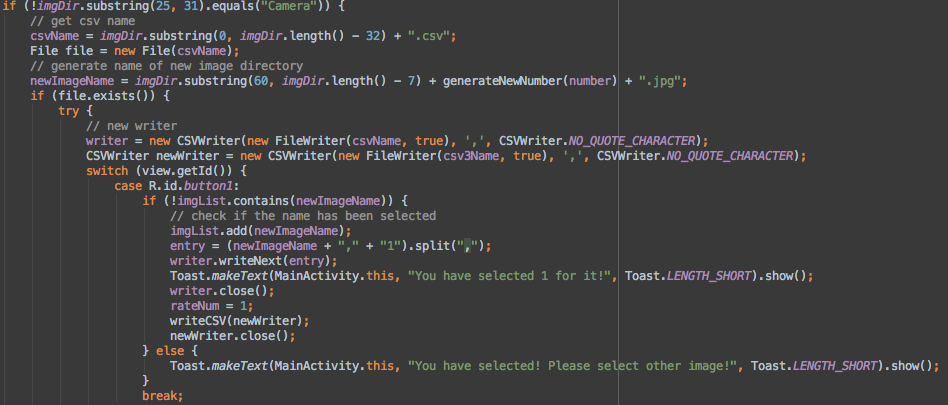
1. Class MyClickListener

This class is called when each of buttons is clicked and as it implements OnClickListener, onClick(View view) must be overridden.The class is divided into 2 parts. By checking whether the String ‘Camera’ is included in the image name, the function creates different .csv names and write the record into these .csv files.

1st part:

This part is used for pictures in /DCIM/group7.

1. Substring the imgDir which represents the image name to get new csvName. Because the String imgDir is the absolute path of an image, I substring the image name from it. Ex: get 400-spxxxx-smile5.jpg from /storage/emulated/0/DCIM/group7/4000-spxxxx-smile5/400-spxx-smile5.jpg (just an example not a real path)
2. Get id of the clicked button and use switch/case to write the record of each image and its rate number into .csv name. The argument ‘writer’ is used for writing in Lab1A’s .csv file and newWriter is for the last .csv file in Lab1B. Make toast when clicking the button but it will cause delay.



2nd part:

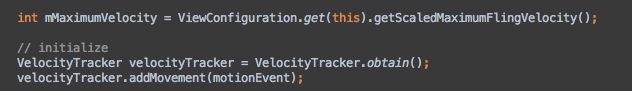
This part is almost same as 1st part. The difference is the .csv name.

1. onTouchEvent(MotionEvent motionEvent)

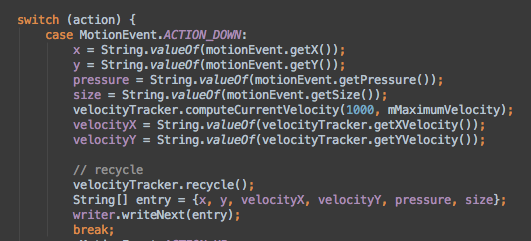
This method is to monitor all touch events and store data into different arguments.

Main parts are shown below.

Get velocity when moving.



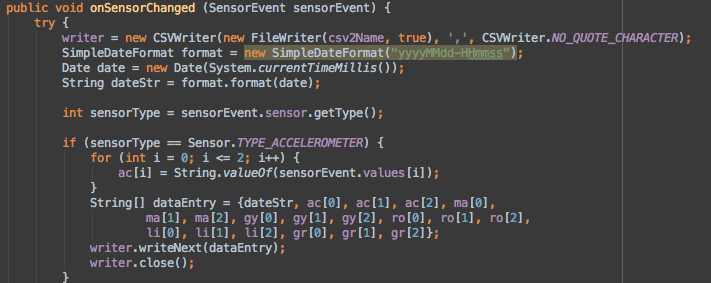
If the action is ACTION\_DOWN, store the data. So does ACTION\_MOVE.



1. SensorEventListener mySensorListener.

In sensorEventListener, onSensorChanged and onAccuracyChanged are overridden.

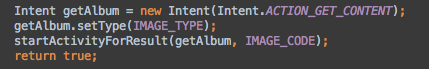
In onSensorChanged, sensorType is used to store sensor type and by checking which sensor it is, different values are passed into specific string arrays. Each time data is transferred, a record is written into .csv file.



../Desktop/Screen%20Shot%202016-09-24%20at%2016.50.28.png

1. onOptionsItemSelected(MenuItem item)

Get the id of the action bar in menu. In this application the item is ‘browse folder’. When the id is matched, it’ll ask system to get permission and then open the gallery.



1. backwordImage and forwardImage.

These two method are click listeners of the buttonBackward and buttonForward separately which is used for moving images backward or forward. Set a number and do the operation number —or number ++ each time.

1. onActivityResult

This method is mentioned before that it’ll be called when the camera or the gallery is opened.

If the requestCode is TAKE\_PHOTO, then it’ll create new intent and image uri to store new image and turn to CROP\_PHOTO. If the requestCode is neither TAKE\_PHOTO nor CROP\_PHOTO, it means an image is choosed and it should be displayed on screen so the selecImage() will be called.

1. selectImage(data)

The argument data is the image uri. And in this method, it will resolve the uri and get the path of this image. Call displayImage and create .csv file.

1. displayImage()

Picasso is used in this method to display Image on screen.

1. writeCSV()

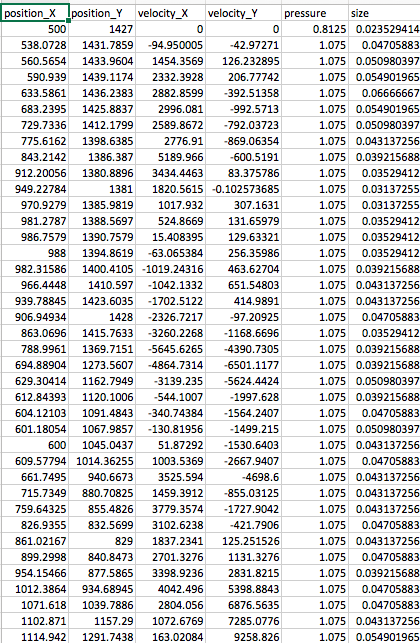
Write record of all the data into Homework3.csv.

And in onPause() the listener is unregistered so when the app shuts down, sensors won’t work.

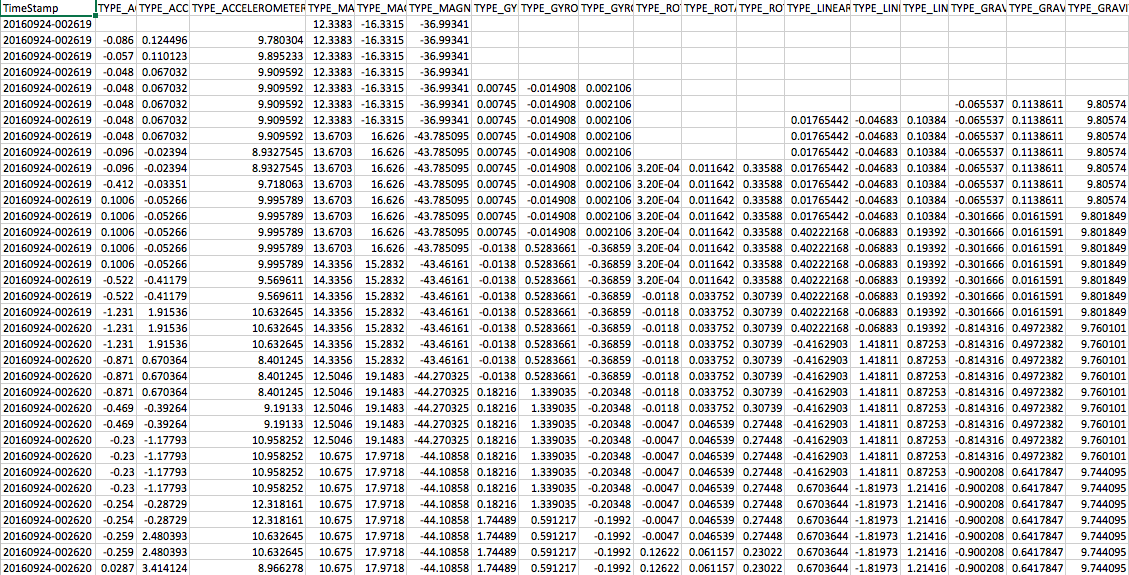
1. Result

CSV file:

Homework1:

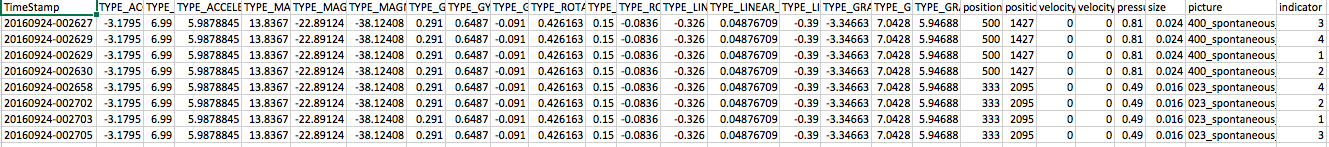


Homework2:



The blank areas mean that there’s no data at that moment.

Homework3:



Screenshot:

