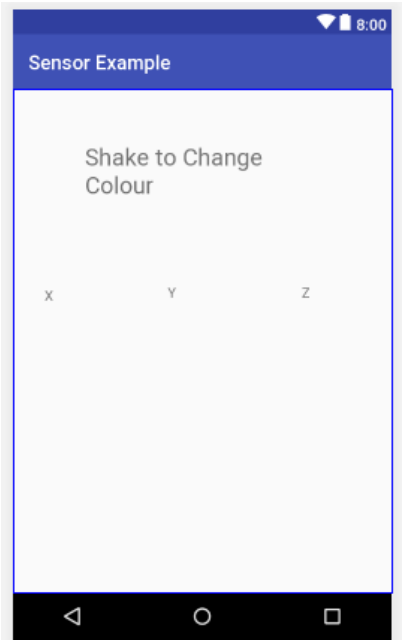
## Android Sensors — Semester 2, 2018

- Accelerometer

In this practical, we will build an application which will change its background color and read 'X', 'Y' and 'Z' cordinates if it is shuffled.

1. Design an activity with 04 Text Views




2. Define the following at the class level

```
private TextView xText, yText, zText;
private SensorManager sensorManager;
private Sensor mySensor;
private boolean color = false;
private View view;
private long lastUpdate;
```

3. Add the following code in the java file of the activity. This will read the X, Y and Z coordinates and display.

```
//Create the sensor manager
sensorManager =  (SensorManager) getSystemService(SENSOR_SERVICE);

//  Accelerometer Sensor
mySensor = sensorManager.getDefaultSensor (Sensor.TYPE_ACCELEROMETER);

//Register sensor listener
sensorManager.registerListener( listener: this,mySensor,SensorManager.SENSOR_DELAY_NORMAL);

//Assign Text Views
xText = (TextView) findViewById(R.id.textViewX);
yText = (TextView) findViewById(R.id.textViewY);
zText = (TextView) findViewById(R.id.textViewZ);
```

```
@Override
public void onSensorChanged(SensorEvent sensorEvent) {
    xText.setText("X: "+ sensorEvent.values[0]);
    yText.setText("Y: "+ sensorEvent.values[1]);
    zText.setText("Z: "+ sensorEvent.values[2]);
}

@Override
public void onAccuracyChanged(Sensor sensor, int i) {

}
```

**Exercise 01**

- Add another text view
- Assign a colour to the text view
- Modify the above code to change the background colour of the newly added text view when phone is shuffled

### Exercise 02 – Create a Compass

- Create the following Custom View Class as the Layout

```
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;

public class MyCompassView extends View {

    private Paint paint;
    private float position = 0;

    public MyCompassView(Context context) {
        super(context);
        init();
    }

    private void init() {
        paint = new Paint();
        paint.setAntiAlias(true);
        paint.setStrokeWidth(2);
        paint.setTextSize(25);
        paint.setStyle(Paint.Style.STROKE);
        paint.setColor(Color.RED);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        int xPoint = getMeasuredWidth() / 2;
        int yPoint = getMeasuredHeight() / 2;

        float radius = (float) (Math.max(xPoint, yPoint) * 0.6);
        canvas.drawCircle(xPoint, yPoint, radius, paint);
        canvas.drawRect(0, 0, getMeasuredWidth(), getMeasuredHeight(), paint);

        // 3.143 is a good approximation for the circle
        canvas.drawLine(
                xPoint,
                yPoint,
                (float) (xPoint + radius
                        * Math.sin((double) (-position) / 180 * 3.143)),
                (float) (yPoint - radius
                        * Math.cos((double) (-position) / 180 * 3.143)), paint);

        canvas.drawText(String.valueOf(position), xPoint, yPoint, paint);
    }

    public void updateData(float position) {
        this.position = position;
        invalidate();
    }

}
```

- Change the coding of your MainA*ctivity*.
  - Create an object from the created custom view and assign it as the layout
  - Register Sensor object
  - Modify public void onSensorChanged() to read the magnetic North value and update the compassView by calling updateData()