

SEMANA 7 – PRACTICA 1

Instrucciones:

1. Modifica el código del activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:id="@+id/btnLienzo"
        android:text="Lienzo"
    ></Button>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="X"
        android:id="@+id/txtx"
        android:layout_weight="1"
        android:textSize="40dp"
        android:gravity="center_vertical"
        android:layout_gravity="center"
    />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Y"
        android:id="@+id/txtty"
        android:layout_weight="1"
        android:textSize="40dp"
        android:gravity="center_vertical"
        android:layout_gravity="center"
    />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Z"
        android:id="@+id/txtz"
        android:layout_weight="1"
        android:textSize="40dp"
        android:gravity="center_vertical"
        android:layout_gravity="center"
    />
</LinearLayout>
```

El diseño debe quedar como se muestra:



2. Modifica el MainActivity.java con el código que se muestra a continuación:

```
public class MainActivity extends AppCompatActivity {
    TextView txtx,txty,txtz;
    SensorManager sensorManager;
    Sensor sensor;

    // elLienzo miLienzo;

    //sensores generan datos, para recibir los datos generados por los sensores
    SensorEventListener sensorEventListener;
    boolean ejecutar=false;
    int movs=0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // miLienzo = new elLienzo(this);
        //setContentView(miLienzo);

        txtx=findViewById(R.id.txtx);
        txty=findViewById(R.id.txty);
        txtz=findViewById(R.id.txtz);
        // Toolbar myToolbar = (Menu) findViewById(R.id.men_principal);
        // setSupportActionBar(myToolbar);
        sensorManager=(SensorManager) getSystemService(SENSOR_SERVICE);
        sensor=sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        //implicitamente se le dice que traiga los datos desde el puerto de memoria
        del acelerómetro
        if (sensor==null) {
            Toast.makeText(this, "Acelerómetro no disponible",
            Toast.LENGTH_SHORT).show();
            finish();
        }
        sensorEventListener = new SensorEventListener() {
            @Override
            public void onSensorChanged(SensorEvent valorenensor) {
                float x=valorenensor.values[0];
                float y=valorenensor.values[1];
                float z=valorenensor.values[2];

                txtx.setText(String.valueOf(x));
                txty.setText(String.valueOf(y));
                txtz.setText(String.valueOf(z));
            }
        }
    }
}
```



```
        if (x < 0) {
            getWindow().getDecorView().setBackgroundColor(Color.YELLOW);
        }
        if (x>0) { getWindow().getDecorView().setBackgroundColor(Color.RED);
            movs++;
            ejecutar=true;
        }
        if (movs==3 && ejecutar){
            movs=0;
            ejecutar=false;
            playsound();
        }
    }

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

    }
};
iniciar();
}

private void playsound() {
    MediaPlayer mediaPlayer= MediaPlayer.create(this, R.raw.grit);
    mediaPlayer.start();
}

private void iniciar() {
    sensorManager.registerListener(sensorEventListener, sensor,
    SensorManager.SENSOR_STATUS_ACCURACY_LOW);
}
private void stop(){
    sensorManager.unregisterListener(sensorEventListener);
}

@Override
protected void onPause() {
    stop();
    super.onPause();
}

@Override
protected void onResume() {
    iniciar();
    super.onResume();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()){
        case R.id.men_principal:{
            Intent i = new Intent(this, MainActivity.class);
            startActivity(i);
            break;
        }
        case R.id.men_listasensores:{
            Intent i = new Intent(this, ListaSensores.class);
            startActivity(i);
            break;
        }
    }
    // case R.id.men_lienzo:{
    //     Intent i = new Intent(this, elLienzo.class);
    // }
```



```
//          startActivity(i);  
//          break;  
//      }  
  
    case R.id.men_nivel:{  
        Intent i = new Intent(this, sensorgrafico.class);  
        startActivity(i);  
        break;  
    }  
}  
return true;  
}
```

3. Para probar la aplicación en el emulador activa la

