**Universitatea Tehnica a Moldovei**

**Facultatea Calculatoare Informatica si Microelectronica**

**Raport**

Lucrarea de laborator Nr. 4

La desciplina: Medii interactive de dezvoltare a produselor soft

A elaborat st. gr. TI-154

N.Dilan

A verificat lect. univ.

I.Cojanu

**Chisinau 2017**

## Tema: Dezvoltarea unei aplicatii mobile

### Obiective:

* Cunostinte de baza privina arhitectura unei aplicatii mobile
* Cunostinte de baza ale platformei SDK

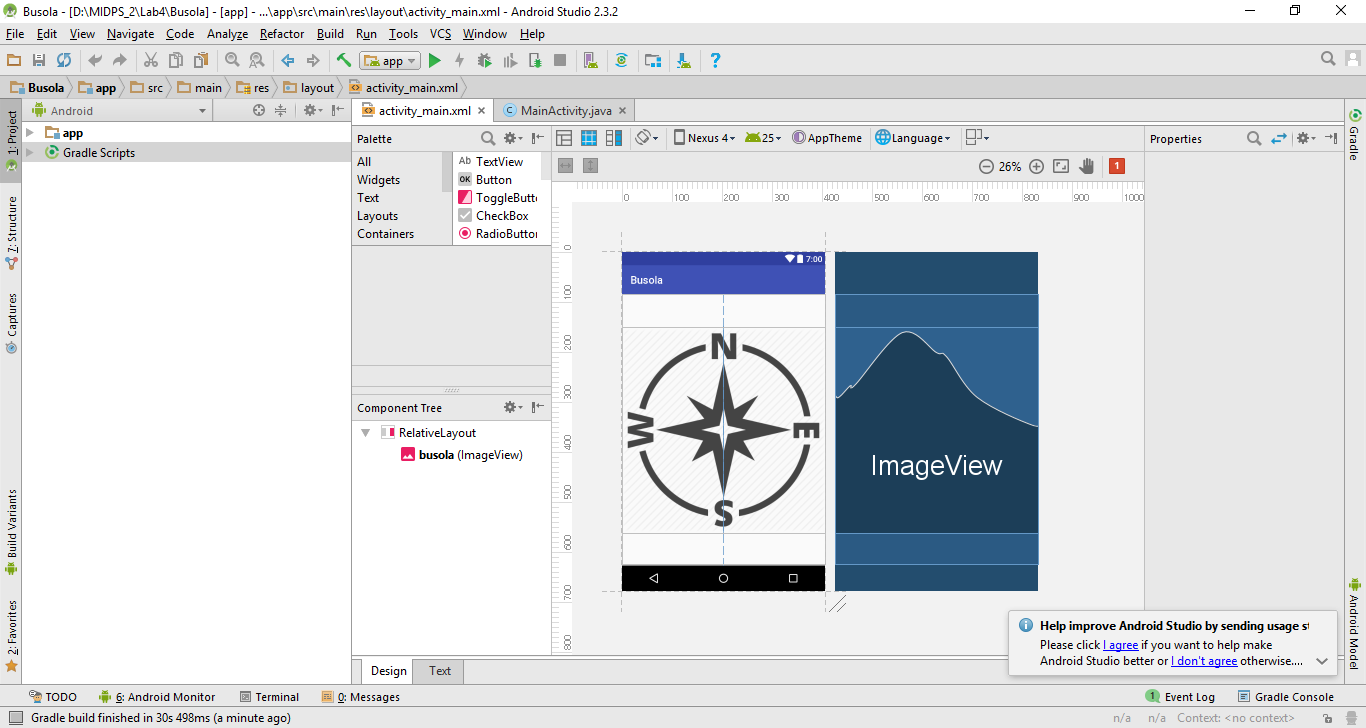
### Laboratory Requirements:

*Advanced Level* (nota 9 || 10):

* Realizeaza o aplicatie care va implimenta tehnica *Pomodoro* **SAU**
* O alta aplicatie sofisticata la alegere
  + Game

**Mersul lucrării:**

Pentru efectuarea lucrărei de laborator am instalat Android Sudio și Android SDK.



Verificarea aplicației:



**Codul programului:**

**activity\_main.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout 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"  
 tools:context="com.example.nixan.busola.MainActivity"**>  
  
  
 <**ImageView  
 android:id="@+id/busola"  
 android:layout\_width="390dp"  
 android:layout\_height="390dp"  
 android:layout\_centerInParent="true"  
 android:src="@drawable/busola"** />  
  
</**RelativeLayout**>

**MainActivity.java:**

**package** com.example.nixan.busola;  
  
**import** android.hardware.Sensor;  
**import** android.hardware.SensorEvent;  
**import** android.hardware.SensorEventListener;  
**import** android.hardware.SensorManager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.animation.Animation;  
**import** android.view.animation.RotateAnimation;  
**import** android.widget.ImageView;  
  
**public class** MainActivity **extends** AppCompatActivity **implements** SensorEventListener{  
  
 **private** ImageView **imageView**;  
 **private float**[] **mGravity** = **new float**[3];  
 **private float**[] **mGeomagnetic** = **new float**[3];  
 **private float azimuth** = 0f;  
 **private float currectAzimuth** = 0f;  
 **private** SensorManager **mSensorManager**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **imageView** = (ImageView) findViewById(R.id.***busola***);  
 **mSensorManager** = (SensorManager) getSystemService(***SENSOR\_SERVICE***);  
 }  
  
 @Override  
 **protected void** onResume() {  
 **super**.onResume();  
 **mSensorManager**.registerListener(**this**,**mSensorManager**.getDefaultSensor(Sensor.***TYPE\_MAGNETIC\_FIELD***),  
 SensorManager.***SENSOR\_DELAY\_GAME***);  
 **mSensorManager**.registerListener(**this**,**mSensorManager**.getDefaultSensor(Sensor.***TYPE\_ACCELEROMETER***),  
 SensorManager.***SENSOR\_DELAY\_GAME***);  
 }  
  
 @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 **mSensorManager**.unregisterListener(**this**);  
 }  
  
 @Override  
 **public void** onSensorChanged(SensorEvent sensorEvent) {  
 **final float** alpha = 0.97f;  
 **synchronized** (**this**){  
 **if**( sensorEvent.**sensor**.getType() == Sensor.***TYPE\_ACCELEROMETER***)  
 {  
 **mGravity**[0]=alpha\***mGravity**[0]+(1-alpha)\*sensorEvent.**values**[0];  
 **mGravity**[1]=alpha\***mGravity**[1]+(1-alpha)\*sensorEvent.**values**[1];  
 **mGravity**[2]=alpha\***mGravity**[2]+(1-alpha)\*sensorEvent.**values**[2];  
 }  
  
 **if**( sensorEvent.**sensor**.getType() == Sensor.***TYPE\_MAGNETIC\_FIELD***)  
 {  
 **mGeomagnetic**[0]=alpha\***mGeomagnetic**[0]+(1-alpha)\*sensorEvent.**values**[0];  
 **mGeomagnetic**[1]=alpha\***mGeomagnetic**[1]+(1-alpha)\*sensorEvent.**values**[1];  
 **mGeomagnetic**[2]=alpha\***mGeomagnetic**[2]+(1-alpha)\*sensorEvent.**values**[2];  
 }  
  
 **float** R[] = **new float**[9];  
 **float** I[] = **new float**[9];  
 **boolean** succes = SensorManager.*getRotationMatrix*(R,I,**mGravity**,**mGeomagnetic**);  
 **if** (succes)  
 {  
 **float** orientation []=**new float**[3];  
 SensorManager.*getOrientation*(R,orientation);  
 **azimuth** = (**float**)Math.*toDegrees*(orientation[0]);  
 **azimuth** = (**azimuth**+360)%360;  
  
 Animation anim = **new** RotateAnimation(-**currectAzimuth**,**azimuth**,Animation.***RELATIVE\_TO\_SELF***,0.5f,Animation.***RELATIVE\_TO\_SELF***,0.5f);  
 **currectAzimuth** = **azimuth**;  
  
 anim.setDuration(500);  
 anim.setRepeatCount(0);  
 anim.setFillAfter(**true**);  
  
 **imageView**.startAnimation(anim);  
 }  
  
 }  
  
 }  
  
 @Override  
 **public void** onAccuracyChanged(Sensor sensor, **int** accuracy) {  
  
 }  
}

**Concluzie**

Efectuînd lucrarea de laborator am studiat Android studio și posibilitățile ei. Am obținut cunostințe noi in programarea pe platforma Android și am aflat cum se testeaza aplicațiea creată de noi pe dispozitiv și pe emulator.