Федеральное агентство связи

Сибирский государственный университет телекоммуникаций и   
 информатики

Кафедра прикладной математики и кибернетики (ПМ и К)

Лабораторная работа «Компас»

по дисциплине «Программирование для мобильных устройств»

Выполнил:

студент ИВТ,  
группы ИП-813

Бурдуковский И. А.

Проверила:  
 Ассистент кафедры ПМиК

Павлова У.В.

Новосибирск 2021

Оглавление  
Текст задания ….……………………………………………………………………………………………3

Теория …………………………………………………………….…………………………………………….3

Скриншоты ……………………………………………………….…………………………………………..4

Листинг кода …………………………………………………………………………………………………4

**Текст задания**Создайте приложение "Компас". На экране отображается циферблат компаса, вращение циферблата осуществляется в зависимости от работы датчика местоположения.

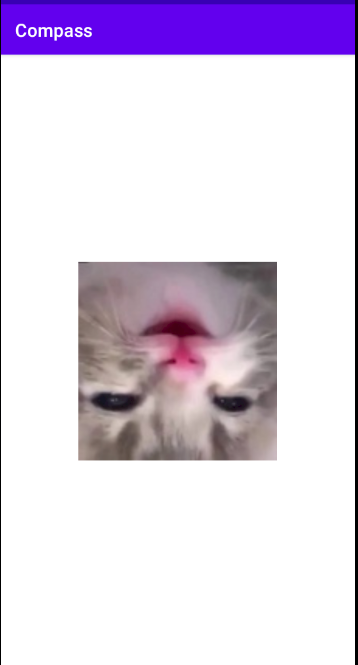
**Теория**

Как заставить компас реагировать и находить стороны света?   
Для этого необходимо воспользоваться датчиками, расположенными в смартфоне. Данные датчики так же используются во многих приложениях, для которых необходимо точное положение телефона в пространстве (например, 2ГИС, где помимо системы GPS используются датчики-сенсоры телефона).  
Для этого было реализовано 2 метода, которые помогают программе вычислить положение смартфона в пространстве и относительно его изголовья (будем считать, что изголовье смартфона находится там, где расположена фронтальная камера).

Методы:  
*TYPE\_MAGNETIC\_FIELD-*Датчик магнитного поля, определяющий текущие показатели магнитного поля в микротеслах по трём осям.

*TYPE\_ACCELEROMETER-*Измеряет ускорение в пространстве по осям X, Y, Z

**Скриншот**



**Листинг кода**

**MainActivity.java**

*package com.example.Compass;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.animation.Animation;  
import android.view.animation.RotateAnimation;  
import android.widget.ImageView;  
import android.hardware.Sensor;  
import android.hardware.SensorEvent;  
import android.hardware.SensorEventListener;  
import android.hardware.SensorManager;  
  
public class MainActivity extends AppCompatActivity implements SensorEventListener {  
  
 //картинка киси  
 private ImageView imageView;  
 private final float[] mGravity = new float[3];  
 private final float[] mGeomagnetic = new float[3];  
  
 //азимут поворота  
 private float azimuth = 0f;  
 private float correctAzimuth = 0f;  
  
 //объявляем работу с сенсером устройства  
 private SensorManager mSensorManager;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 //Связываем объект ImageView с нашим изображением:  
 imageView = (ImageView) findViewById(R.id.imageView);  
  
 //Инициализируем возможность работать с сенсором устройства:  
 mSensorManager = (SensorManager) getSystemService(SENSOR\_SERVICE);  
 }  
  
 @Override  
 protected void onResume(){  
 super.onResume();  
 //Устанавливаем слушатели ориентации сенсера  
 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();  
 //Останавливаем при надобности слушателя ориентации  
 //сенсора с целью сбережения заряда батареи:  
 mSensorManager.unregisterListener(this);  
 }  
  
 @Override  
 //гравитация и преобразование магнитного поля относительно x y z  
 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];  
 }  
  
 //Создаем анимацию вращения:  
 float[] R =new float[9];  
 float[] I =new float[9];  
 boolean success = SensorManager.getRotationMatrix(R,I,mGravity,mGeomagnetic);  
 if(success){  
 float[] orientation = new float[3];  
 SensorManager.getOrientation(R,orientation);  
 azimuth = (float)Math.toDegrees(orientation[0]);  
 azimuth = (azimuth+360)%360;  
 Animation anim = new RotateAnimation(-correctAzimuth,-azimuth,Animation.RELATIVE\_TO\_SELF,0.5f,Animation.RELATIVE\_TO\_SELF,0.5f);  
 correctAzimuth = azimuth;  
 //Продолжительность анимации в миллисекундах:  
 anim.setDuration(500);  
 anim.setRepeatCount(0);  
 //Настраиваем анимацию после завершения подсчетных действий датчика:  
 anim.setFillAfter(true);  
 //Запускаем анимацию:  
 imageView.startAnimation(anim);  
 }  
 }  
 }  
  
 @Override  
 public void onAccuracyChanged(Sensor sensor,int i){  
  
 }  
}*

**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:id="@+id/activity\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="16dp"  
 android:paddingLeft="16dp"  
 android:paddingRight="16dp"  
 android:paddingTop="16dp"  
 tools:context=".MainActivity">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="300dp"  
 android:layout\_height="300dp"  
 android:layout\_centerInParent="true"  
 android:layout\_marginStart="70dp"  
 android:layout\_marginTop="116dp"  
 android:layout\_marginEnd="70dp"  
 android:layout\_marginBottom="38dp"  
 android:contentDescription="@string/todo"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/kisya" />  
  
</RelativeLayout>*