

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет
телекоммуникаций и информатики»

Лабораторная работа №4
«Компас»

Выполнили: студенты 3 курса
ИВТ, гр. ИП-713
Трусов К.В.
Михеев Н.А.

Новосибирск, 2020 г.

Задание

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

Решение поставленной задачи

Вся программа была реализованная в MainActivity. В нем были реализованы добавление картинки компаса, отслеживание датчика пространства: ORIENTATION_SENSOR и изменение угла наклона картинки компаса по направлению на сервер.

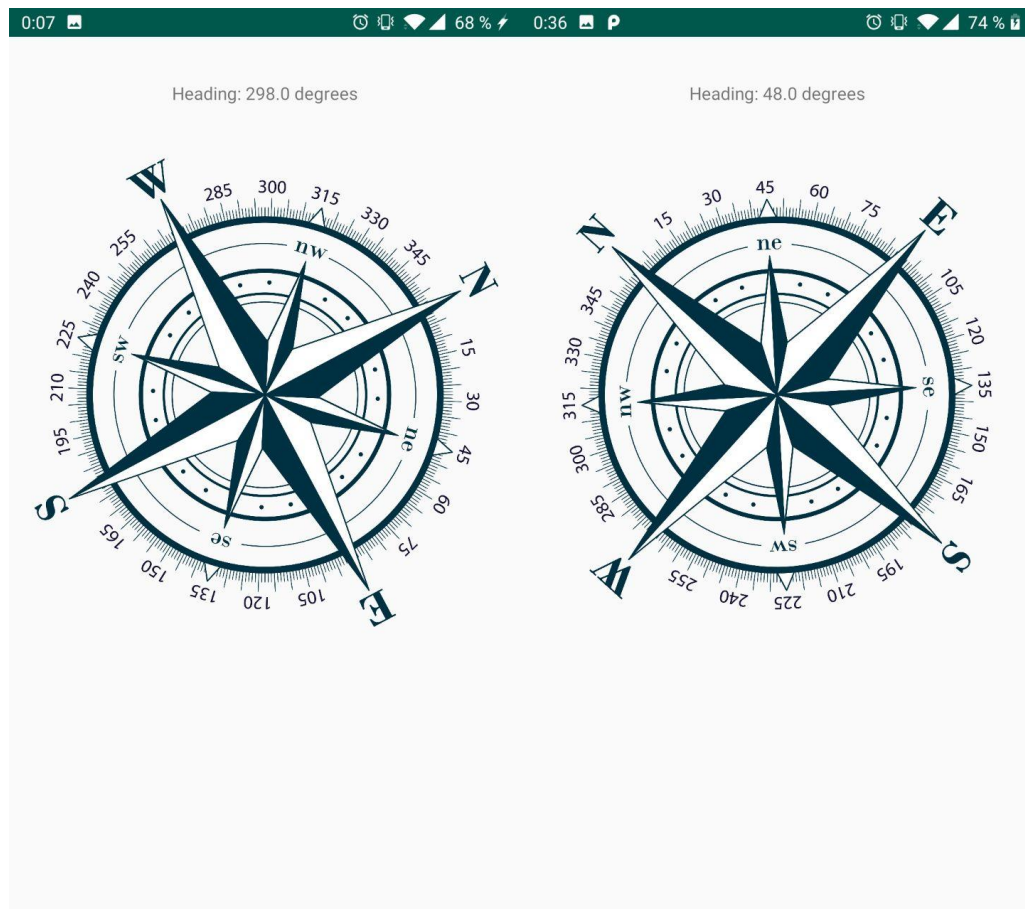


Рис.1-2 – состояния программы с различным положением в пространстве

Листинг программы

Класс MainActivity:

```
public class MainActivity extends Activity implements SensorEventListener {  
    private ImageView image;  
    private float currentDegree = 0f;  
    private SensorManager mSensorManager;  
    TextView tvHeading;
```

```

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    image = (ImageView) findViewById(R.id.imageViewCompass);

    tvHeading = (TextView) findViewById(R.id.tvHeading);

    mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

}

@Override

protected void onResume() {

    super.onResume();

    mSensorManager.registerListener(this, mSensorManager.getDefaultSensor(Sensor.TYPE_ORIENTATION),

        SensorManager.SENSOR_DELAY_GAME);

}

@Override

protected void onPause() {

    super.onPause();

    mSensorManager.unregisterListener(this);

}

@Override

public void onSensorChanged(SensorEvent event) {

    float degree = Math.round(event.values[0]);

    tvHeading.setText("Heading: " + Float.toString(degree) + " degrees");

    RotateAnimation ra = new RotateAnimation(

        currentDegree,

        -degree,

        Animation.RELATIVE_TO_SELF, 0.5f,

        Animation.RELATIVE_TO_SELF,

        0.5f);

    ra.setDuration(210);

    ra.setFillAfter(true);

    image.startAnimation(ra);

    currentDegree = -degree;

}

@Override

public void onAccuracyChanged(Sensor sensor, int accuracy) {

}

}

```