



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №5
із дисципліни «Розробка мобільних застосунків під Android»
ДОСЛІДЖЕННЯ РОБОТИ З ВБУДОВАНИМИ ДАТЧИКАМИ

Виконала
студентка групи ІА–24
Сіденко Дар'я

Київ 2025

Мета роботи: ознайомитись з можливостями вбудованих датчиків мобільних пристроїв та дослідити способи їх використання для збору та обробки даних.

Завдання

БАЗОВЕ (10/20 балів). Написати програму під платформу Андроїд, яка має інтерфейс для виведення даних з обраного вбудованого датчика (тип обирається самостійно, можна відслідковувати зміни значень і з декількох датчиків).

ПОВНЕ (20/20). Функціональність базового додатку додатково розширюється обробкою отриманих даних та виведенням їх у відповідній формі.

Примітка: конкретного варіанту не передбачено, студент сам обирає завдання та вигляд програми. Приклади очікуваних робіт:

- «будівельний рівень» з виведенням лінії горизонту та кутом нахилу;
- компас з ілюстрацією стрілки (циферблату з позначеними сторонами світу);
- крокомір (підрахунок кількості кроків);
- додаток для вимірювання перевантажень в авто (G-force meter);
- автоматичне регулювання яскравості та екрану в залежності від рівня освітлення, але ще б додати автозаглушення екрану при піднесенні до перешкоди (до вуха під час розмови або «в кишені»), щоб уникнути ненавмисних дотиків;
- барометр з прогнозом погоди (мова про опади – зміна атмосферного тиску, а, можливо, і вологості з температурою).

Хід роботи

Під час виконання лабораторної роботи я вирішила створити компас, що показує напрям та чіткі координати. Для цього було створено розмітку з компасною стрілкою і додатковим підписом напрямку знизу.

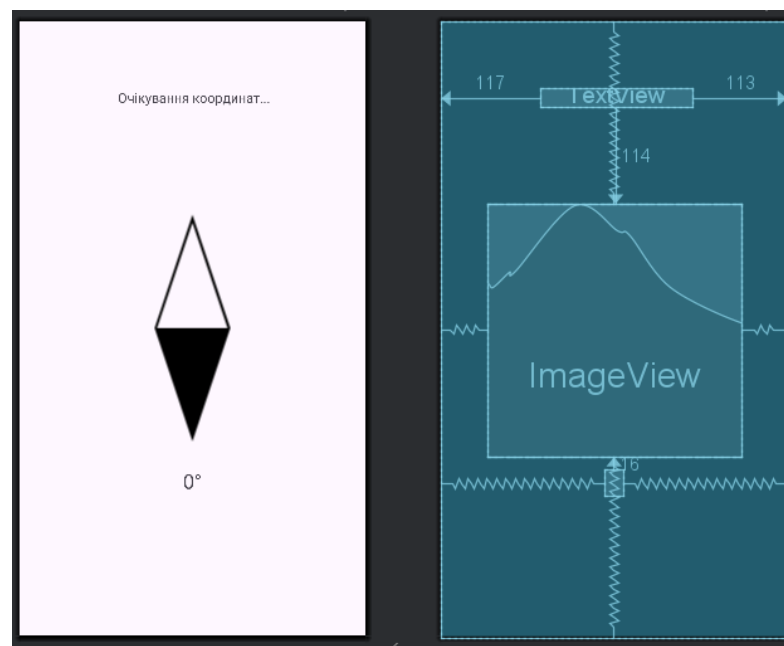


Рис. 1 – Розмітка активності

Лістинг коду activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/compass_image"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_centerInParent="true"
        android:src="@drawable/compass_needle" />

    <TextView
        android:id="@+id/azimuth_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0°"
        android:textSize="24sp"
        android:layout_below="@id/compass_image"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp" />

    <TextView
        android:id="@+id/location_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@id/compass_image"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_gravity="center_horizontal"
        android:layout_marginStart="117dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="113dp"
        android:layout_marginBottom="114dp"
        android:text="Очікування координат..."
        android:textSize="16sp" />

</RelativeLayout>
```

Для того щоб програма працювала коректно було надано певні дозволи:

```
<uses-permission android:name="android.permission.ACTIVITY_RECOGNITION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Перший дозвіл дозволяє відслідковувати рух користувача, щоб коректно відображати напрям компасу. Другий та третій відповідають за геолокацію користувача.

Головний клас після ініціалізації та налаштувань отримує данні з акселерометра та магнітометра, а потім обчислюється матриця повороту, з якої обчислюється азимут. Так і відбувається визначення кута на який має повернутись стрілка.

Лістинг коду MainActivity.java

```
public class MainActivity extends Activity implements SensorEventListener {

    private ImageView compassImage;
    private TextView azimuthText;
    private float currentDegree = 0f;
    private SensorManager sensorManager;
    private Sensor accelerometer;
    private Sensor magnetometer;

    private float[] gravity;
    private float[] geomagnetic;
    private FusedLocationProviderClient fusedLocationClient;
    private TextView locationText;
    private static final int LOCATION_PERMISSION_REQUEST = 1001;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        locationText = findViewById(R.id.location_text); // додай цей
        TextView y XML
        fusedLocationClient =
        LocationServices.getFusedLocationProviderClient(this);
        requestLocationUpdates();

        compassImage = findViewById(R.id.compass_image);
        azimuthText = findViewById(R.id.azimuth_text);

        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        accelerometer =
        sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        magnetometer =
        sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
    }

    @Override
    protected void onResume() {
        super.onResume();
        sensorManager.registerListener(this, accelerometer,
        SensorManager.SENSOR_DELAY_UI);
        sensorManager.registerListener(this, magnetometer,
        SensorManager.SENSOR_DELAY_UI);
    }

    @Override
    protected void onPause() {
        super.onPause();
        sensorManager.unregisterListener(this);
    }

    private void requestLocationUpdates() {
        if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
            String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            LOCATION_PERMISSION_REQUEST);
            return;
        }
    }
}
```

```

    }

    LocationRequest locationRequest = LocationRequest.create();
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    locationRequest.setInterval(5000); // 5 секунд

    fusedLocationClient.requestLocationUpdates(locationRequest, new
LocationCallback() {
        @Override
        public void onLocationResult(LocationResult locationResult) {
            if (locationResult == null) return;
            Location location = locationResult.getLastLocation();
            if (location != null) {
                double lat = location.getLatitude();
                double lon = location.getLongitude();
                double alt = location.hasAltitude() ?
location.getAltitude() : 0;
                locationText.setText("Lat: " + lat + "\nLon: " + lon +
"\nAlt: " + alt + " m");
            }
        }
    }, Looper.getMainLooper());
}

@Override
public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if (requestCode == LOCATION_PERMISSION_REQUEST) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            requestLocationUpdates();
        } else {
            Toast.makeText(this, "Потрібен дозвіл на доступ до
геолокації", Toast.LENGTH_LONG).show();
        }
    }
}

@Override
public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
gravity = event.values;
    if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD)
geomagnetic = event.values;

    if (gravity != null && geomagnetic != null) {
        float R[] = new float[9];
        float I[] = new float[9];
        boolean success = SensorManager.getRotationMatrix(R, I, gravity,
geomagnetic);
        if (success) {
            float orientation[] = new float[3];
            SensorManager.getOrientation(R, orientation);
            float azimuth = (float) Math.toDegrees(orientation[0]);
            azimuth = (azimuth + 360) % 360;

            RotateAnimation rotateAnimation = new RotateAnimation(
                currentDegree,
                -azimuth,
                Animation.RELATIVE_TO_SELF, 0.5f,
                Animation.RELATIVE_TO_SELF, 0.5f);

```

```

        rotateAnimation.setDuration(500);
        rotateAnimation.setFillAfter(true);

        compassImage.startAnimation(rotateAnimation);
        currentDegree = -azimuth;

        String direction = getDirectionText(azimuth);
        azimuthText.setText(Math.round(azimuth) + "° - " +
direction);
    }
}

private String getDirectionText(float azimuth) {
    if (azimuth >= 337.5 || azimuth < 22.5) return "North";
    else if (azimuth >= 22.5 && azimuth < 67.5) return "North-East";
    else if (azimuth >= 67.5 && azimuth < 112.5) return "East";
    else if (azimuth >= 112.5 && azimuth < 157.5) return "South-East";
    else if (azimuth >= 157.5 && azimuth < 202.5) return "South";
    else if (azimuth >= 202.5 && azimuth < 247.5) return "South-West";
    else if (azimuth >= 247.5 && azimuth < 292.5) return "West";
    else return "North-West";
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // Not used
}
}

```

В результаті отримала програму, що відображає чітку геолокацію та показує напрям.

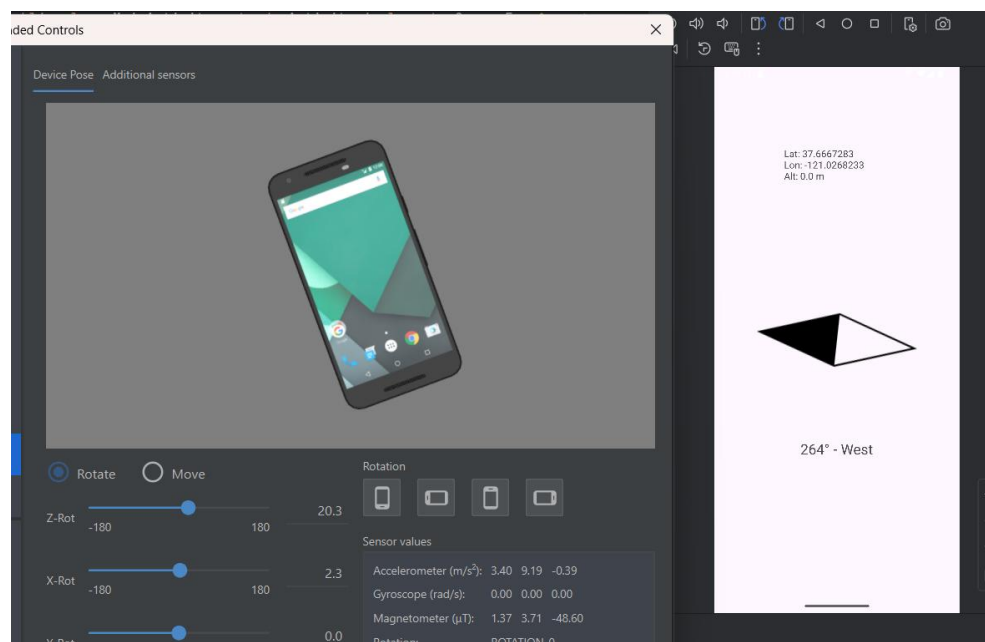


Рис. 2 – Результат роботи компаса

Висновок: під час виконання даної лабораторної роботи я навчилась працювати з датчиками мобільних пристроїв та використала знання на практиці.