**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Кафедра ЕОМ**



Звіт

до лабораторної роботи №6

з дисципліни: «Програмні технології мобільних обчислень»

на тему: «Управління геоконтекстними сервісами в ОС Android»

Варіант №14

Виконав студент групи КІ-44:

Ольховик О. С.

Перевірив:

Бачинський Р.В.

**Львів 2023**

**Мета:** Оволодіти навичками роботи з програмним інтерфейсом геоконтекстних сервісів в ОС Android.

**Завдання:**

* Тестова програма, яка показує розташування на карті того корпусу Львівської політехніки, в якому буде проводитись наступне заняття згідно розкладу занять.
* Тестова програма, яка визначає біжучу відстань між користувачем та заданим корпусом Львівської політехніки (номер корпусу користувач вибирає зі списку) та відображає її у вигляді відрізку прямої лінії на карті.

**Хід виконання**

1. **Порядок налаштування середовища розробки для роботи з сасобами позиціонуваття та сервісом Google Maps.**
   1. Додати фрагмент карти в аплікацію за допомогою:

<fragment  
 android:id="@+id/google\_map"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:name="com.google.android.gms.maps.SupportMapFragment"/>

* 1. Отримання API Key для Google Maps, для цього потрібно виконати наступні дії:

Щоб отримати API Key для Google Maps, виконайте наступні дії:

* Зареєструйтеся в Google Cloud Platform.
* Увійдіть в консоль Google Cloud Platform.
* Натисніть кнопку "Create Project".
* Введіть назву проекту і виберіть місце для його розташування.
* Натисніть кнопку "Create".
* У меню "API & Services" виберіть пункт "Credentials".
* Натисніть кнопку "Create Credentials".
* Виберіть пункт "API Key".
* Виберіть тип API Key "Web Server".
* Натисніть кнопку "Create".

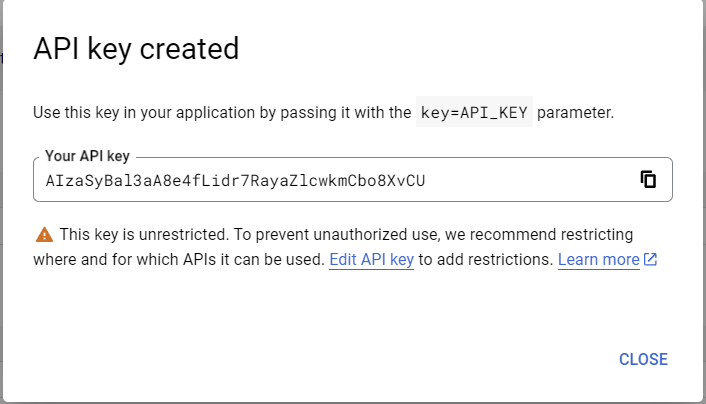


Рис.1. Отримання API ключа

* 1. Додавання необхідних залежностей в **build.gradle**

*implementation* ("com.karumi:dexter:6.2.3")  
*implementation* ("com.android.volley:volley:1.2.1")  
*implementation* ("com.google.android.gms:play-services-maps:18.2.0")  
*implementation* ("com.google.android.gms:play-services-location:21.0.1")

Ось коротке пояснення того, для чого використовується кожна залежність:

* **com.karumi:dexter:6.2.3**

Ця залежність використовується для спрощення процесу запиту дозволів під час виконання на Android. Вона надає простий API, який робить простим запит дозволів і обробку відповіді користувача.

* **com.android.volley:volley:1.2.1**

Ця залежність використовується для здійснення мережевих запитів. Це легка і проста у використанні бібліотека, яка може використовуватися для завантаження даних з веб-сайту, завантаження файлів тощо.

* **com.google.android.gms:play-services-maps:18.2.0**

Ця залежність використовується для додавання підтримки Google Maps до вашого додатка. Він дозволяє відображати карти, додавати маркери та включати маршрути.

* **com.google.android.gms:play-services-location:21.0.1**

Ця залежність використовується для додавання підтримки розташування до вашого додатка. Він дозволяє отримувати поточне розташування користувача, відстежувати рух користувача тощо.

* 1. В manifest.xml додати наступне:

<uses-permission android:name="android.permission.INTERNET"/>  
<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"/>  
<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>  
<uses-permission android:name="android.permission.ACCESS\_BACKGROUND\_LOCATION" />

* **android.permission.INTERNET** - дозволяє додатку отримувати доступ до Інтернету. Це потрібно для більшості додатків, оскільки вони повинні спілкуватися з віддаленими серверами для завантаження даних і надання функціональності.
* **android.permission.ACCESS\_COARSE\_LOCATION** - дозволяє додатку отримувати доступ до приблизного розташування користувача. Це можна використовувати для таких функцій, як відображення прогнозу погоди на основі розташування користувача або знаходження найближчих бізнесів.
* **android.permission.ACCESS\_FINE\_LOCATION** - дозволяє додатку отримувати доступ до точного розташування користувача. Це потрібно для таких функцій, як навігація покроково або відстеження прогресу фітнесу користувача.
* **android.permission.ACCESS\_BACKGROUND\_LOCATION** - дозволяє додатку отримувати доступ до розташування користувача навіть тоді, коли він не в передньому плані. Це потрібно для таких функцій, як відстеження руху користувача протягом дня або надання сповіщень на основі розташування.
  1. Додати API GMAPS ключ в manifest.xml

<meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="AIzaSyBq3feCjhxQ-qtVGO-xO7ZgGt9KbSb2Zns">  
</meta-data>

1. **Лістинг тестової програми**

**MainActivity.java**

package com.example.myapplication;  
  
import static java.lang.String.*format*;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
  
import android.Manifest;  
import android.annotation.SuppressLint;  
import android.content.Context;  
import android.content.pm.PackageManager;  
import android.graphics.Color;  
import android.graphics.Point;  
import android.location.Location;  
import android.location.LocationManager;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.view.WindowManager;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Spinner;  
import android.widget.Toast;  
  
import com.android.volley.toolbox.JsonObjectRequest;  
import com.android.volley.toolbox.Volley;  
import com.google.android.gms.location.FusedLocationProviderClient;  
import com.google.android.gms.location.LocationServices;  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.MarkerOptions;  
import com.google.android.gms.maps.model.Polyline;  
import com.google.android.gms.maps.model.PolylineOptions;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.karumi.dexter.Dexter;  
import com.karumi.dexter.PermissionToken;  
import com.karumi.dexter.listener.PermissionDeniedResponse;  
import com.karumi.dexter.listener.PermissionGrantedResponse;  
import com.karumi.dexter.listener.PermissionRequest;  
import com.karumi.dexter.listener.single.PermissionListener;  
  
import java.util.Arrays;  
import java.util.HashMap;  
import java.util.Map;  
  
public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {  
  
 SupportMapFragment supportMapFragment;  
 FusedLocationProviderClient fusedLocationProviderClient;  
  
 private static Map<String, mPoint> *locations* = new HashMap<>();  
 private GoogleMap mMap;  
 protected LocationManager locationManager;  
 PolylineOptions polylineOptions = new PolylineOptions();  
  
 LatLng myPos;  
  
 class mPoint  
 {  
 double first;  
 double second;  
  
 mPoint(double first, double second)  
 {  
 this.first = first;  
 this.second = second;  
 }  
 }  
  
 {  
 *locations*.put("Головна будівля", new mPoint(49.835664417899494, 24.01443679983333));  
 *locations*.put("Перший корпус", new mPoint(49.83534091276235, 24.010644727103653));  
 *locations*.put("Другий корпус", new mPoint(49.83611705969751, 24.012392109423697));  
 *locations*.put("Третій корпус", new mPoint(49.83653599231934, 24.01360437589593));  
 *locations*.put("Четвертий корпус", new mPoint(49.836423507581806, 24.011372301879856));  
 *locations*.put("П'ятий корпус", new mPoint(49.83499578639943, 24.00810262479517));  
 *locations*.put("Шостий корпус", new mPoint(49.83522227115248, 24.006496308753526));  
 *locations*.put("Сьомий корпус", new mPoint(49.83466293290917, 24.009693295262277));  
 *locations*.put("Восьмий корпус", new mPoint(49.837843234646435, 24.01253141246118));  
 *locations*.put("Дев'ятий корпус", new mPoint(49.836571707842964, 24.01433568177108));  
 *locations*.put("Десятий корпус", new mPoint(49.8365301891598, 24.01510815798094));  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*, WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  
  
 supportMapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.*google\_map*);  
  
 fusedLocationProviderClient = LocationServices.*getFusedLocationProviderClient*(this);  
  
 Dexter.*withContext*(getApplicationContext()).withPermission(Manifest.permission.*ACCESS\_FINE\_LOCATION*)  
 .withListener(new PermissionListener() {  
 @Override  
 public void onPermissionGranted(PermissionGrantedResponse permissionGrantedResponse) {  
 myPos = getCurrentLocation();  
 }  
  
 @Override  
 public void onPermissionDenied(PermissionDeniedResponse permissionDeniedResponse) {  
  
 }  
  
 @Override  
 public void onPermissionRationaleShouldBeShown(PermissionRequest permissionRequest, PermissionToken permissionToken) {  
 permissionToken.continuePermissionRequest();  
 }  
 }).check();  
  
  
 String[] items = {"Головна будівля", "Перший корпус", "Другий корпус", "Третій корпус", "Четвертий корпус", "П'ятий корпус", "Шостий корпус", "Сьомий корпус", "Восьмий корпус", "Дев'ятий корпус", "Десятий корпус"};  
 ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_spinner\_item*, items);  
 adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
  
 Spinner spinner = findViewById(R.id.*spinner*);  
 spinner.setAdapter(adapter);  
  
 adapter.notifyDataSetChanged();  
 spinner.setSelection(5);  
  
  
 spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
 String selectedItem = (String) parent.getItemAtPosition(position);  
 mPoint mPoint = *locations*.get(selectedItem);  
 assert mPoint != null;  
 setloc(mPoint.first, mPoint.second);  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
 // Do nothing.  
 }  
 });  
  
 }  
  
 public void setloc(double lat, double lng) {  
 if(myPos != null) {  
 LatLng corpPos = new LatLng(lat, lng);  
 float[] results = new float[1];  
 Location.*distanceBetween*(myPos.latitude, myPos.longitude,  
 corpPos.latitude, corpPos.longitude, results);  
 mMap.clear();  
 mMap.addMarker(new MarkerOptions().position(corpPos).title("Корпус, відстань " + results[0] + " m"));  
 mMap.addMarker(new MarkerOptions().position(myPos).title("Я"));  
 Polyline line = mMap.addPolyline(new PolylineOptions().width(5).color(Color.*RED*));  
 mMap.addPolyline(polylineOptions);  
 line.setPoints(Arrays.*asList*(corpPos, myPos));  
 mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(corpPos, 15.7f));  
 }  
 }  
  
 public LatLng getCurrentLocation() {  
 final LatLng[] latLng = new LatLng[1];  
  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED* && ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_COARSE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 // *TODO: Consider calling* // *ActivityCompat#requestPermissions* // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.  
 return null;  
 }  
 Task<Location> task = fusedLocationProviderClient.getLastLocation();  
 task.addOnSuccessListener(new OnSuccessListener<Location>() {  
 @Override  
 public void onSuccess(Location location) {  
 supportMapFragment.getMapAsync(new OnMapReadyCallback() {  
 @Override  
 public void onMapReady(@NonNull GoogleMap googleMap) {  
 if(location != null) {  
 latLng[0] = new LatLng(location.getLatitude(),location.getLongitude());  
 myPos = latLng[0];  
 MarkerOptions markerOptions = new MarkerOptions().position(latLng[0]).title("Current Location !");  
 googleMap.addMarker(markerOptions);  
 googleMap.animateCamera(CameraUpdateFactory.*newLatLngZoom*(latLng[0],15));  
 mMap = googleMap;  
 mPoint mPoint1 = *locations*.get("Головна будівля");  
 assert mPoint1 != null;  
 setloc(mPoint1.first, mPoint1.second);  
  
  
 }  
 else {  
 Toast.*makeText*(MainActivity.this, "Please on your Location App Permissions", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
 });  
 }  
 });  
  
 return latLng[0];  
 }  
  
 @Override  
 public void onMapReady(GoogleMap googleMap) {  
 mMap = googleMap;  
 mMap.addMarker(new MarkerOptions().position(myPos).title("Я"));  
 mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(myPos, 18f));  
 }  
}

**activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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=".MainActivity">  
  
 <fragment  
 android:id="@+id/google\_map"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:name="com.google.android.gms.maps.SupportMapFragment"/>  
  
 <Spinner  
 android:id="@+id/spinner"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="55dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.949"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.023"  
 android:radius="10dp"  
 android:color="#FF0000"  
 style="@style/CustomSpinnerTheme"  
 />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. **Результати виконання тестової програми**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Рис.2. Запит на отримання доступу до геолокації | | | Рис.3. Відстань між моєю поточною локацією та 5-им корпусом НУЛП | |
| Рис.4. Список вибору корпусів | Рис.5. Відстань між моєю поточною локацією та головним корпусом НУЛП | |

**Висновок:**  Під час виконання даної лабораторної роботи, я оволодів навичками роботи з програмним інтерфейсом геоконтекстних сервісів в ОС Android та створив програму яка відображає відстань від поточного місцязнаходження до вибраного зі списку корпусу НУЛП.