# 实验6 Android在物联网中的应用

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 202231060901 | 姓名 | 刘洋 |
| 专业 | 物联网工程 | 年级 | 2022级 |

特别说明：实验报告必须命名为【学号紧跟姓名\_实验项目名】的格式，不包含括号【】。

例如【202231060901刘洋\_Androi\_实验6.docx】，不按此命名不得分。

实验报告提交地址：<https://pintia.cn/>

## 选择项目： GPS轨迹记录与存储 (90分)

(1)XML代码（30分）

|  |
| --- |
| <?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">  <TextView  android:id="@+id/distance"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hello World!"  android:textAppearance="@style/TextAppearance.AppCompat.Large"  android:textSize="60sp"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />  <Button  android:id="@+id/button"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="获取当前坐标"  android:textSize="36sp"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.498"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/distance"  app:layout\_constraintVertical\_bias="0.247" />  </androidx.constraintlayout.widget.ConstraintLayout> |

(2)Java代码(30分)

每个Java文件放一个表格。如果有多个。

MainActivity.java

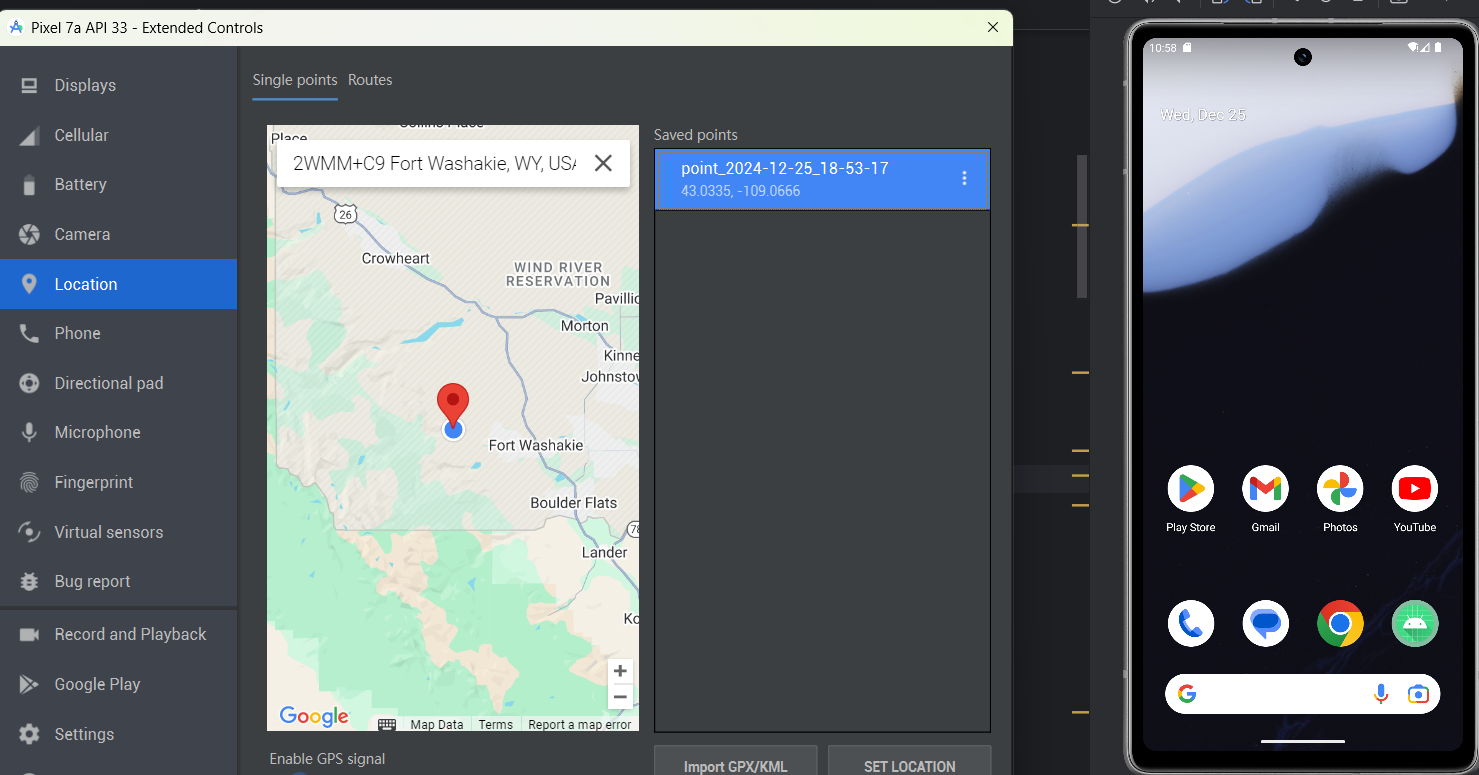
|  |
| --- |
| package edu.swpu.iot2022.liuyang;  import android.Manifest;  import android.content.ComponentName;  import android.content.Context;  import android.content.Intent;  import android.content.ServiceConnection;  import android.content.pm.PackageManager;  import android.os.Bundle;  import android.os.Handler;  import android.os.IBinder;  import android.widget.Button;  import android.widget.TextView;  import android.widget.Toast;  import androidx.activity.EdgeToEdge;  import androidx.annotation.NonNull;  import androidx.appcompat.app.AppCompatActivity;  import androidx.core.app.ActivityCompat;  import androidx.core.content.ContextCompat;  import androidx.core.graphics.Insets;  import androidx.core.view.ViewCompat;  import androidx.core.view.WindowInsetsCompat;  import java.util.Locale;  public class MainActivity extends AppCompatActivity {  private static Button mButton;  private ServiceConnection mConnection;  public static Button getButton() {  return mButton;  }  private final int PERMISSION\_REQUEST\_CODE=1;  private OdometerService mOdometerService;  private boolean bound =false;//用于记录绑定的状态  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  mButton=findViewById(R.id.button);  mConnection = new ServiceConnection() {  @Override  public void onServiceConnected(ComponentName name, IBinder service) {  //用于绑定服务  OdometerService.OdometerBinder binder = (OdometerService.OdometerBinder)service;  mOdometerService =binder.getOdometer();  bound = true;  }  @Override  public void onServiceDisconnected(ComponentName name) {  //用于解绑服务  bound=false;  }  };  displayDistance();  }  @Override  public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  super.onRequestPermissionsResult(requestCode, permissions, grantResults);  switch (requestCode) {  case PERMISSION\_REQUEST\_CODE: {  if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {  Intent intent = new Intent(this, OdometerService.class);  bindService(intent, mConnection, Context.BIND\_AUTO\_CREATE);  } else {  Toast.makeText(this, "Location permission required", Toast.LENGTH\_SHORT).show();  }  }  break;  default:  throw new IllegalStateException("Unexpected value: " + requestCode);  }  }  @Override  protected void onStart() {  super.onStart();  if(ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION)  != PackageManager.PERMISSION\_GRANTED){  ActivityCompat.requestPermissions(this,new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION},PERMISSION\_REQUEST\_CODE);  }else{  Intent intent = new Intent(this,OdometerService.class);  bindService(intent,mConnection, Context.BIND\_AUTO\_CREATE);  }  }  @Override  protected void onStop() {  super.onStop();  if(bound){  unbindService(mConnection);  bound =false;  }  }  private void displayDistance(){  final TextView distanceView = findViewById(R.id.distance);  final Handler handler = new Handler();  handler.post(new Runnable() {  @Override  public void run() {  double distance = 12.4;  if(bound&&mOdometerService !=null){  distance= mOdometerService.getDistanceMeters();  }  String distanceStr = String.format(Locale.getDefault(),"%1$,.2fKM",distance);  distanceView.setText(distanceStr);  handler.postDelayed(this,1000);  }  });  }  } |

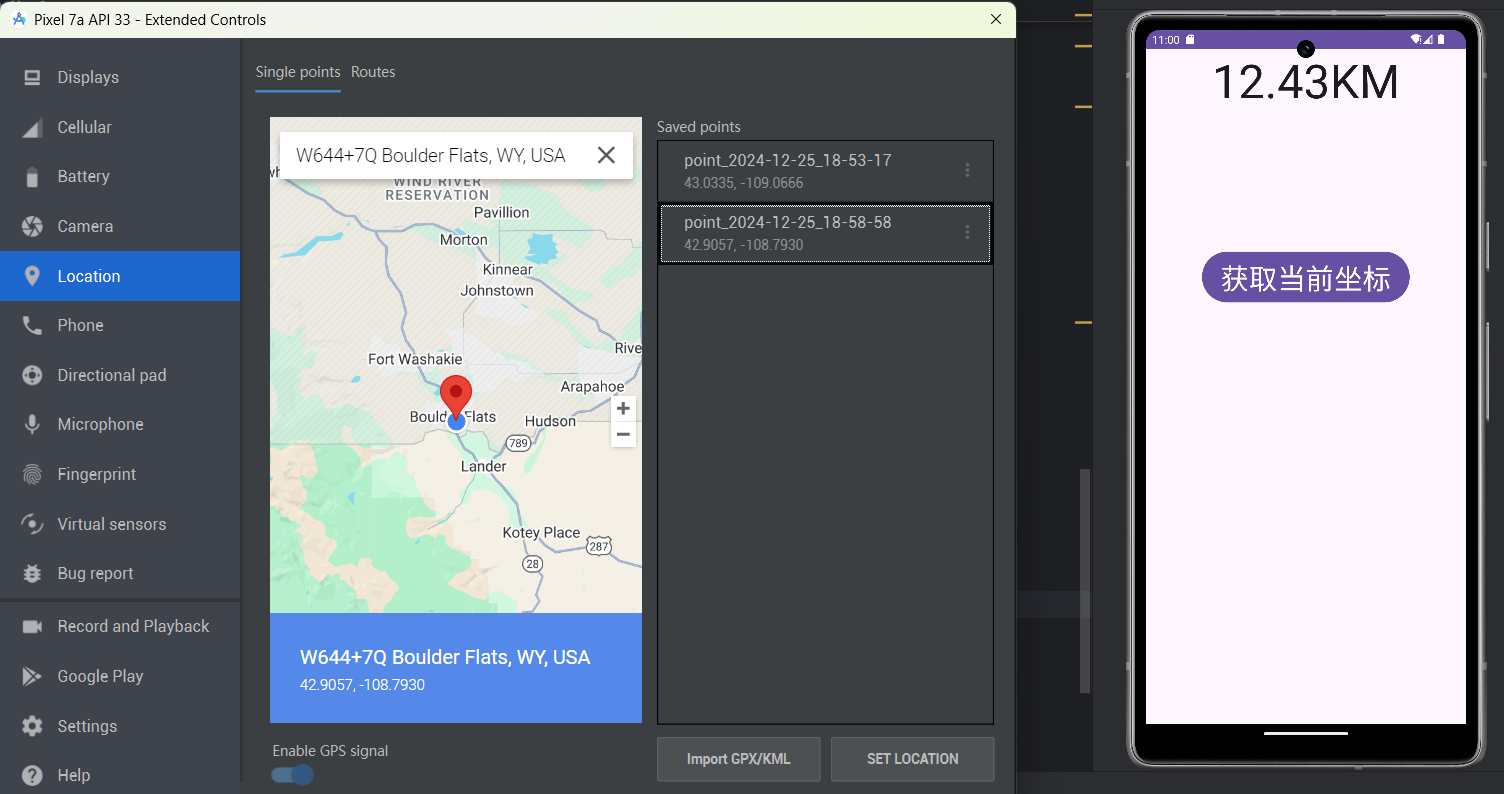
OdometerService.java

|  |
| --- |
| package edu.swpu.iot2022.liuyang;  import android.Manifest;  import android.app.Service;  import android.content.Context;  import android.content.Intent;  import android.content.pm.PackageManager;  import android.location.Criteria;  import android.location.Location;  import android.location.LocationListener;  import android.location.LocationManager;  import android.os.Binder;  import android.os.Bundle;  import android.os.IBinder;  import android.util.Log;  import android.view.View;  import android.widget.Button;  import androidx.annotation.NonNull;  import androidx.core.app.ActivityCompat;  import androidx.core.content.ContextCompat;  import java.util.List;  public class OdometerService extends Service {  private Location mFirstLocation = null;  private Location mLocation;  public class OdometerBinder extends Binder {  OdometerService getOdometer() {  return OdometerService.this;  }  }  private double distanceMeters;  private LocationManager mLocationManager;  private LocationListener mLocationListener;  @Override  public void onCreate() {  super.onCreate();  mLocationManager = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);    // 创建位置请求的监听器  mLocationListener = new LocationListener() {  private Location lastLocation;  @Override  public void onLocationChanged(@NonNull Location location) {  if (lastLocation == null) {  lastLocation = location;  } else {  distanceMeters = location.distanceTo(lastLocation);  if(lastLocation==null)  Log.d("Log:location","上次纬度:"+location.getLatitude()+", 经度是: "+location.getLongitude());  else  Log.d("Log:location","上次纬度:"+lastLocation.getLatitude()+", 经度是: "+lastLocation.getLongitude());  Log.d("Log:location","当前纬度:"+location.getLatitude()+", 经度是: "+location.getLongitude());  lastLocation = location;  Log.d("Log:location", "当前距离: " + distanceMeters + "米");  }  }  @Override  public void onStatusChanged(String provider, int status, Bundle extras) {  }  @Override  public void onProviderEnabled(String provider) {  }  @Override  public void onProviderDisabled(String provider) {  }  };  // 请求位置更新  if (ActivityCompat.checkSelfPermission(this,  Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED) {  mLocationManager.requestLocationUpdates(  LocationManager.GPS\_PROVIDER, // 使用GPS提供者  1000, // 最小时间间隔（毫秒）  1, // 最小距离间隔（米）  mLocationListener  );  }  }  private Location getLastKnownLocation() {  mLocationManager = (LocationManager) getApplicationContext().getSystemService(LOCATION\_SERVICE);  List<String> providers = mLocationManager.getProviders(true);  Location bestLocation = null;  for (String provider : providers) {  if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  Log.d("Log:location", "权限未获取");  return null;  }  Location l = mLocationManager.getLastKnownLocation(provider);  if (l == null) {  continue;  }  if (bestLocation == null || l.getAccuracy() < bestLocation.getAccuracy()) {  // Found best last known location: %s", l);  bestLocation = l;  }  }  return bestLocation;  }  public double getDistanceMeters() {  return distanceMeters/1000;  }  @Override  public void onDestroy() {  super.onDestroy();  if(mLocationManager !=null && mLocationListener!=null){  mLocationManager.removeUpdates(mLocationListener);  }  mLocationManager=null;  mLocationListener=null;  }  private IBinder mIBinder = new OdometerBinder();  @Override  public IBinder onBind(Intent intent) {  return mIBinder;  }  } |

(3)运行截图(30分)

组图，多个图





## 实验心得体会(10分)

要求不少于100汉字，必须是自己原创，不得雷同。

|  |
| --- |
| 空指针异常问题  问题描述：在调用mLocation.getLatitude()时出现NullPointerException  解决方案：在使用mLocation对象前添加了null检查，避免了空指针异常  具体实现：添加了if (mLocation != null)的判断，并在为null时输出相应的日志信息  点击按钮无法获取位置信息问题  改为使用位置监听器,继承LocationListener,再调用mLocationManager.requestLocationUpdates通知系统这里有监听需求 |