INTERACTIVE ITC MAP

TECHNICAL MANUAL

Developer: Adam Strahan
Supervisor: Oisin Cawley

Contents

Introduction

My goal for this project is to create a map application of the IT Carlow campus. The map should allow students enter a location and plot a route to it, either from their location or an entered starting point. It should make getting around the campus easier for students who may not be familiar with it.

In this document I will go through the various technical aspects of the project, including:

- A step-by-step guide to installing the application
- Screenshots of the code

Installation Guide

The application can be found on OneDrive at the following link:

https://instituteoftechnol663-my.sharepoint.com/:u:/g/personal/c00216499_itcarlow_ie/EcmHKC2OlJFJrhqMCxaWv7MBouwIvPqctbhTkDCJpU78vA?e=xxY868.

Once you have the zip folder downloaded:

1. Extract the files.
2. Open the project in your IDE (*I recommend Android Studio as the application was created in Java*).
3. Link your Android device to your computer using a USB cable

   OR

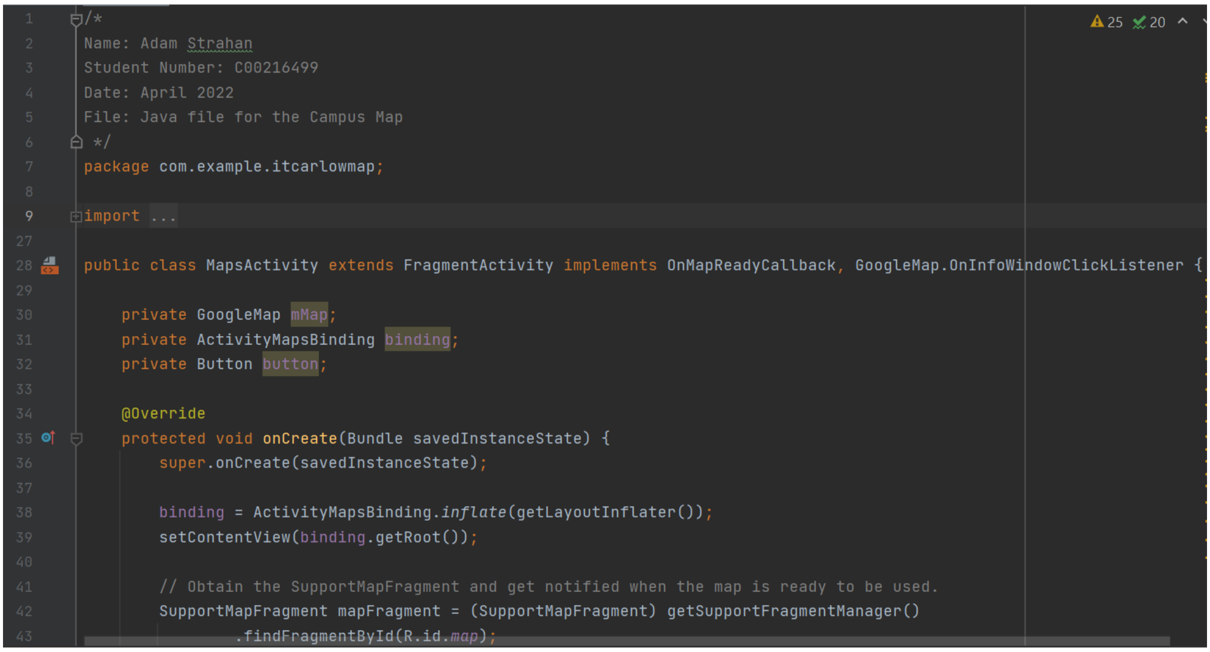
   Set up an Android emulator in Android Studio.

4. Build and run the project.

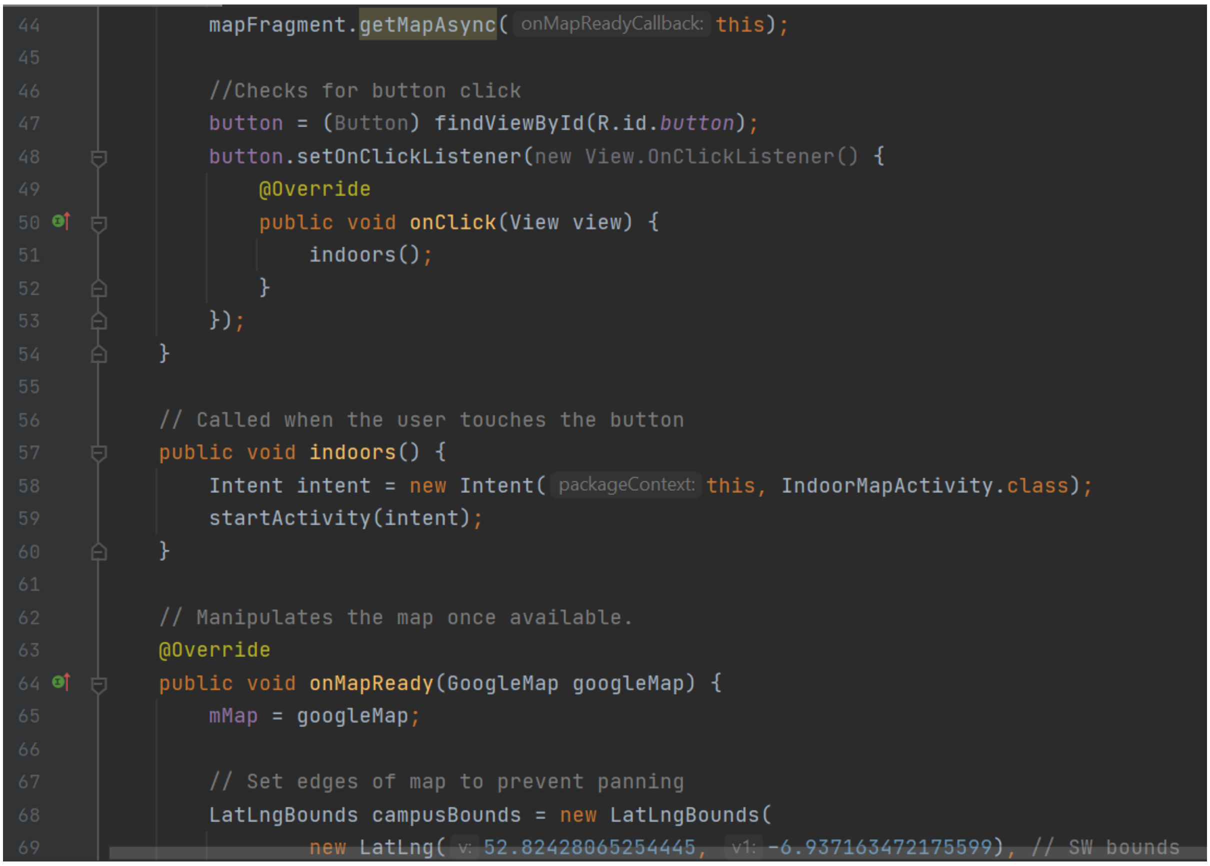
At this point, the application should be installed on your Android device, or running inside your Android emulator. Enjoy!

The Code

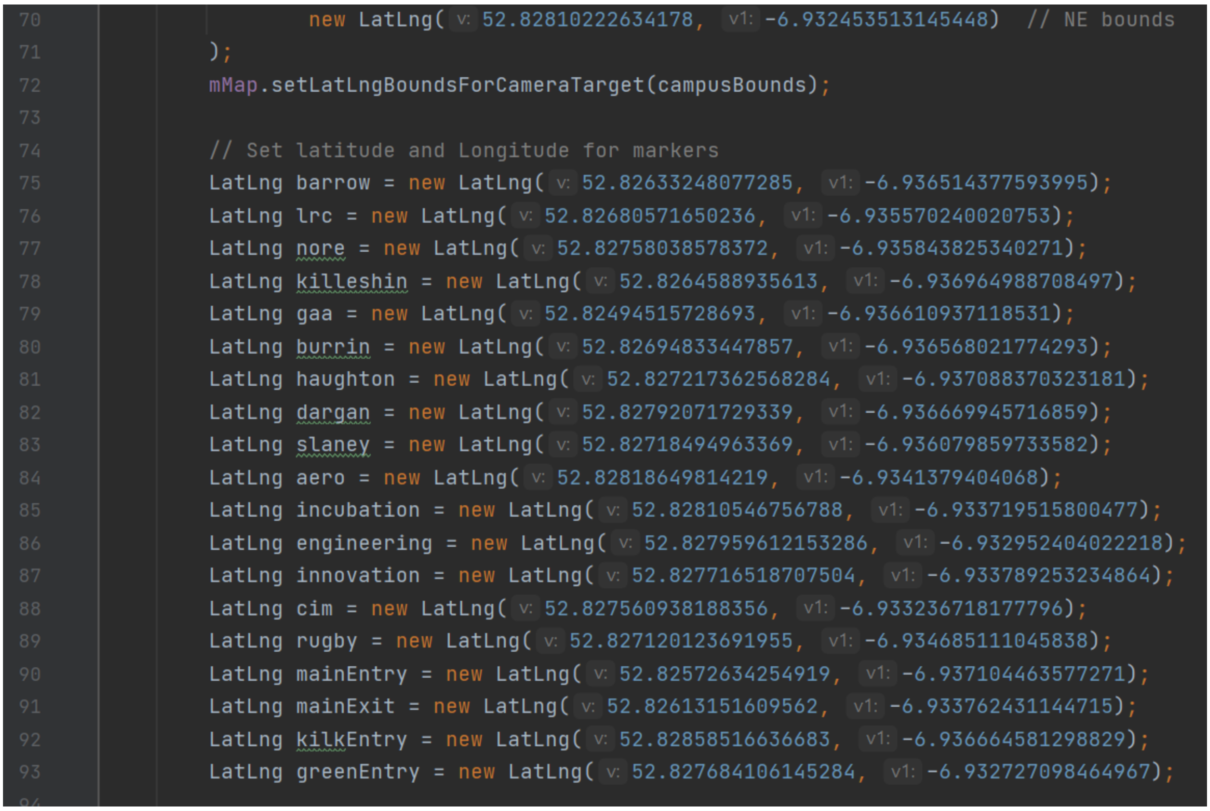
MapsActivity.java



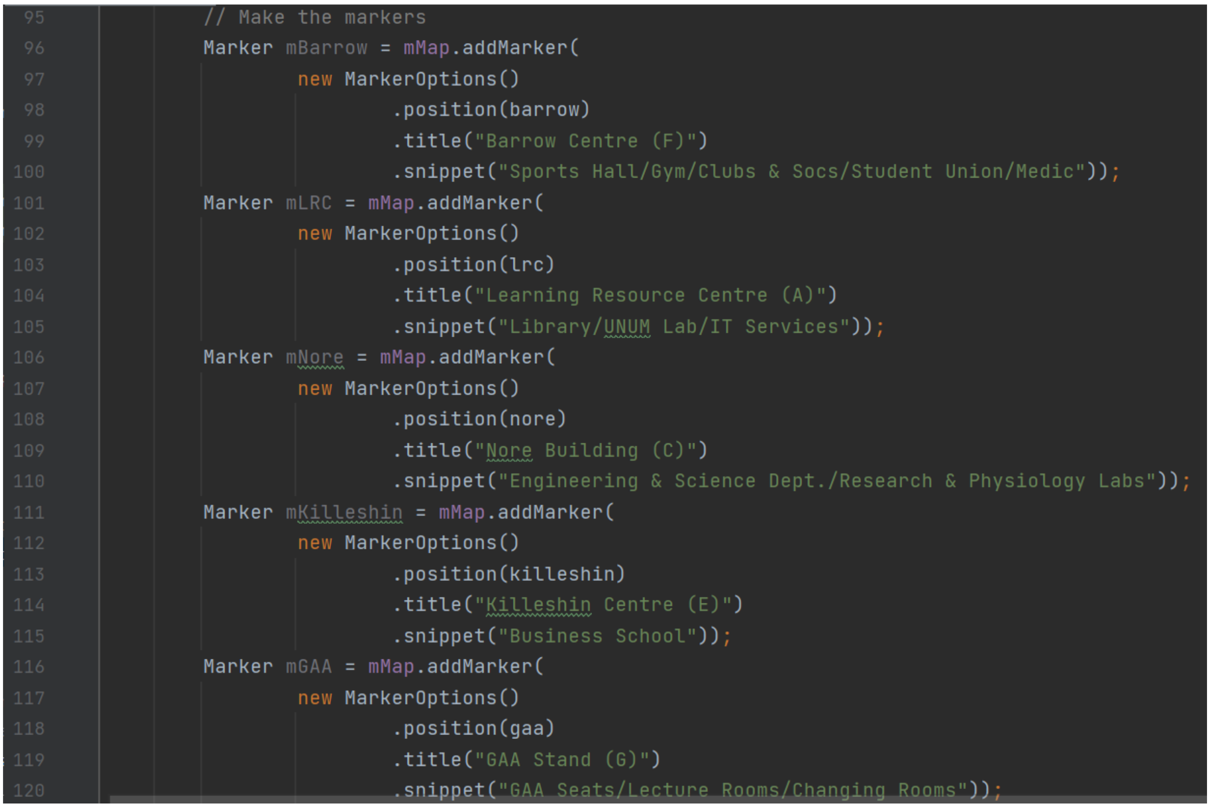
```java
/*
Name: Adam Strahan
Student Number: C00216499
Date: April 2022
File: Java file for the Campus Map
 */
package com.example.itcarlowmap;

import ...

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback, GoogleMap.OnInfoWindowClickListener {

    private GoogleMap mMap;
    private ActivityMapsBinding binding;
    private Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMapsBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
```



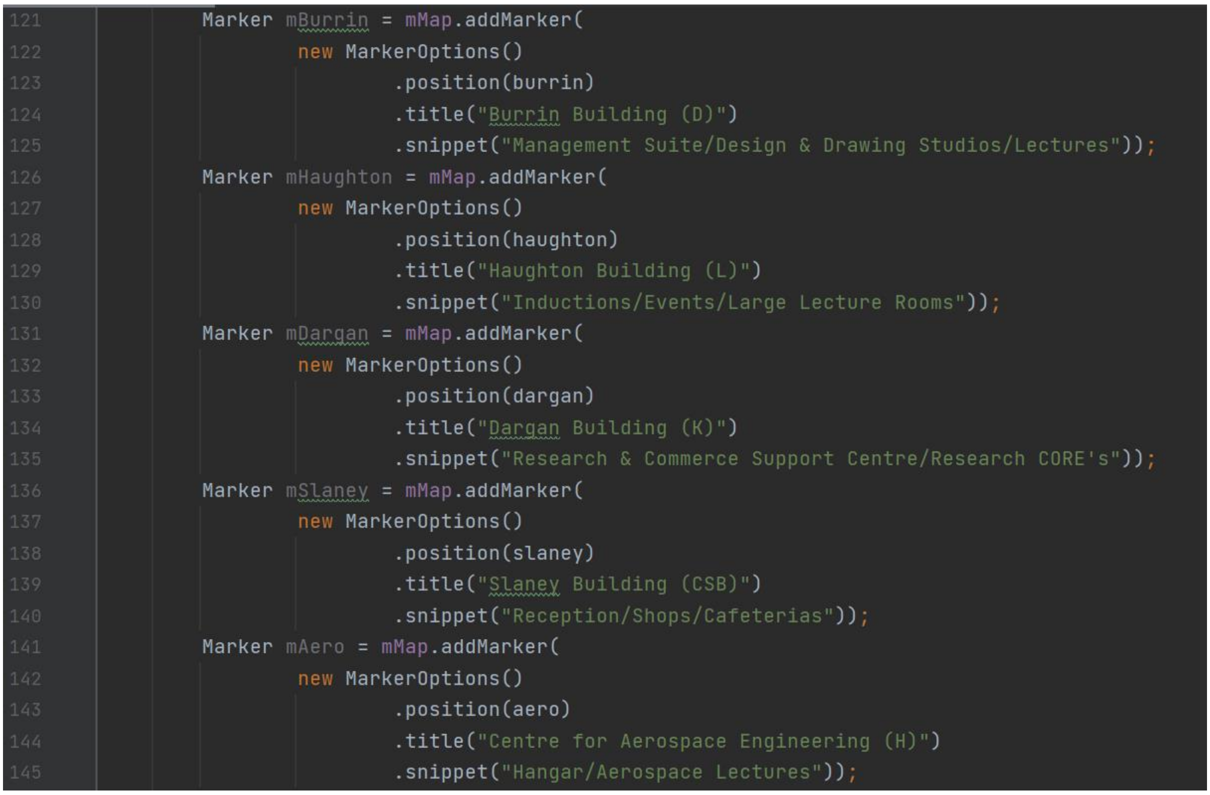
```java
        mapFragment.getMapAsync( onMapReadyCallback: this);

        //Checks for button click
        button = (Button) findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                indoors();
            }
        });
    }

    // Called when the user touches the button
    public void indoors() {
        Intent intent = new Intent( packageContext: this, IndoorMapActivity.class);
        startActivity(intent);
    }

    // Manipulates the map once available.
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        // Set edges of map to prevent panning
        LatLngBounds campusBounds = new LatLngBounds(
                new LatLng( v: 52.82428065254445,  v1: -6.937163472175599), // SW bounds
```



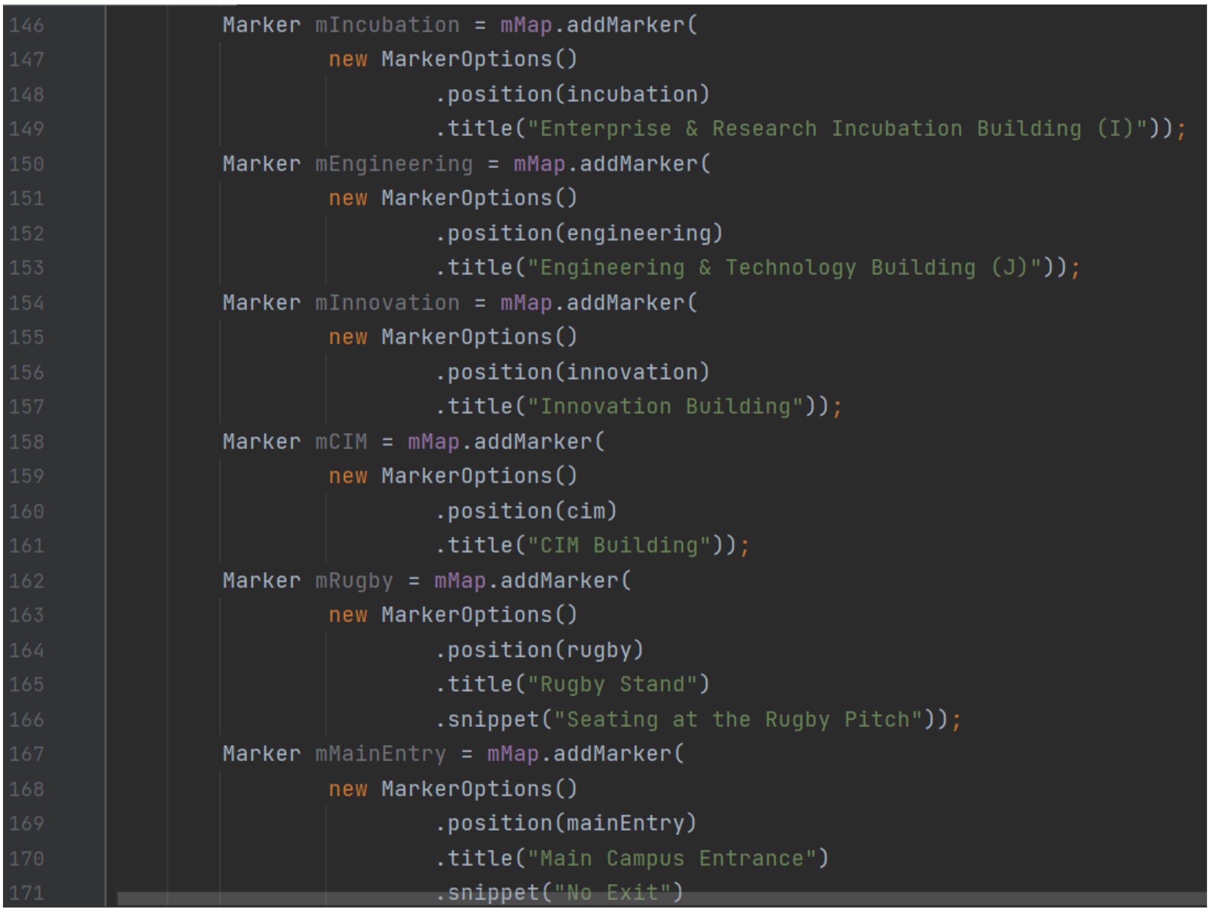
```
                new LatLng( v: 52.82810222634178,  v1: -6.932453513145448)  // NE bounds
        );
        mMap.setLatLngBoundsForCameraTarget(campusBounds);

        // Set latitude and Longitude for markers
        LatLng barrow = new LatLng( v: 52.82633248077285,  v1: -6.936514377593995);
        LatLng lrc = new LatLng( v: 52.82680571650236,  v1: -6.935570240020753);
        LatLng nore = new LatLng( v: 52.82758038578372,  v1: -6.935843825340271);
        LatLng killeshin = new LatLng( v: 52.8264588935613,  v1: -6.936964988708497);
        LatLng gaa = new LatLng( v: 52.82494515728693,  v1: -6.936610937118531);
        LatLng burrin = new LatLng( v: 52.82694833447857,  v1: -6.936568021774293);
        LatLng haughton = new LatLng( v: 52.827217362568284,  v1: -6.937088370323181);
        LatLng dargan = new LatLng( v: 52.82792071729339,  v1: -6.936669945716859);
        LatLng slaney = new LatLng( v: 52.82718494963369,  v1: -6.936079859733582);
        LatLng aero = new LatLng( v: 52.82818649814219,  v1: -6.9341379404068);
        LatLng incubation = new LatLng( v: 52.82810546756788,  v1: -6.933719515800477);
        LatLng engineering = new LatLng( v: 52.827959612153286,  v1: -6.932952404022218);
        LatLng innovation = new LatLng( v: 52.827716518707504,  v1: -6.933789253234864);
        LatLng cim = new LatLng( v: 52.827560938188356,  v1: -6.933236718177796);
        LatLng rugby = new LatLng( v: 52.827120123691955,  v1: -6.934685111045838);
        LatLng mainEntry = new LatLng( v: 52.82572634254919,  v1: -6.937104463577271);
        LatLng mainExit = new LatLng( v: 52.82613151609562,  v1: -6.933762431144715);
        LatLng kilkEntry = new LatLng( v: 52.82858516636683,  v1: -6.936664581298829);
        LatLng greenEntry = new LatLng( v: 52.827684106145284,  v1: -6.932727098464967);
```



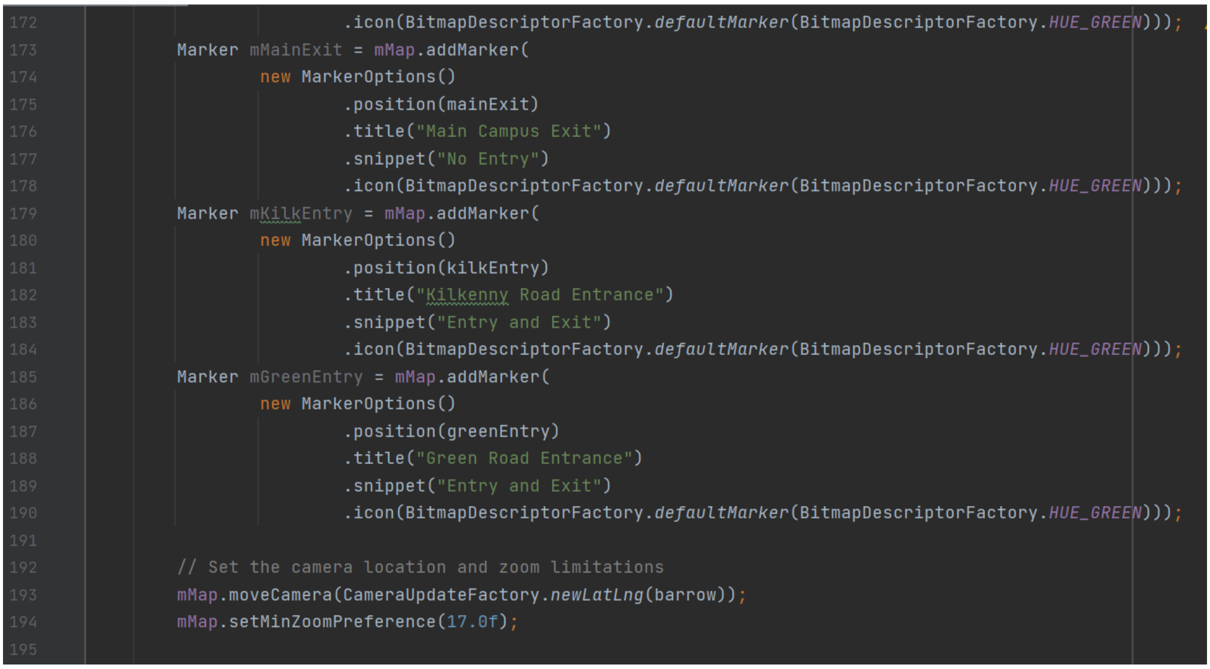
```
        // Make the markers
        Marker mBarrow = mMap.addMarker(
                new MarkerOptions()
                        .position(barrow)
                        .title("Barrow Centre (F)")
                        .snippet("Sports Hall/Gym/Clubs & Socs/Student Union/Medic"));
        Marker mLRC = mMap.addMarker(
                new MarkerOptions()
                        .position(lrc)
                        .title("Learning Resource Centre (A)")
                        .snippet("Library/UNUM Lab/IT Services"));
        Marker mNore = mMap.addMarker(
                new MarkerOptions()
                        .position(nore)
                        .title("Nore Building (C)")
                        .snippet("Engineering & Science Dept./Research & Physiology Labs"));
        Marker mKilleshin = mMap.addMarker(
                new MarkerOptions()
                        .position(killeshin)
                        .title("Killeshin Centre (E)")
                        .snippet("Business School"));
        Marker mGAA = mMap.addMarker(
                new MarkerOptions()
                        .position(gaa)
                        .title("GAA Stand (G)")
                        .snippet("GAA Seats/Lecture Rooms/Changing Rooms"));
```



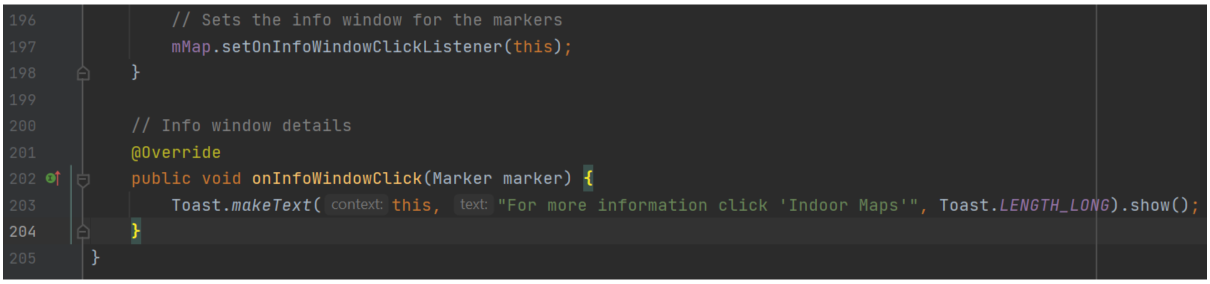
```java
        Marker mBurrin = mMap.addMarker(
                new MarkerOptions()
                        .position(burrin)
                        .title("Burrin Building (D)")
                        .snippet("Management Suite/Design & Drawing Studios/Lectures"));
        Marker mHaughton = mMap.addMarker(
                new MarkerOptions()
                        .position(haughton)
                        .title("Haughton Building (L)")
                        .snippet("Inductions/Events/Large Lecture Rooms"));
        Marker mDargan = mMap.addMarker(
                new MarkerOptions()
                        .position(dargan)
                        .title("Dargan Building (K)")
                        .snippet("Research & Commerce Support Centre/Research CORE's"));
        Marker mSlaney = mMap.addMarker(
                new MarkerOptions()
                        .position(slaney)
                        .title("Slaney Building (CSB)")
                        .snippet("Reception/Shops/Cafeterias"));
        Marker mAero = mMap.addMarker(
                new MarkerOptions()
                        .position(aero)
                        .title("Centre for Aerospace Engineering (H)")
                        .snippet("Hangar/Aerospace Lectures"));
```



```java
        Marker mIncubation = mMap.addMarker(
                new MarkerOptions()
                        .position(incubation)
                        .title("Enterprise & Research Incubation Building (I)"));
        Marker mEngineering = mMap.addMarker(
                new MarkerOptions()
                        .position(engineering)
                        .title("Engineering & Technology Building (J)"));
        Marker mInnovation = mMap.addMarker(
                new MarkerOptions()
                        .position(innovation)
                        .title("Innovation Building"));
        Marker mCIM = mMap.addMarker(
                new MarkerOptions()
                        .position(cim)
                        .title("CIM Building"));
        Marker mRugby = mMap.addMarker(
                new MarkerOptions()
                        .position(rugby)
                        .title("Rugby Stand")
                        .snippet("Seating at the Rugby Pitch"));
        Marker mMainEntry = mMap.addMarker(
                new MarkerOptions()
                        .position(mainEntry)
                        .title("Main Campus Entrance")
                        .snippet("No Exit")
```

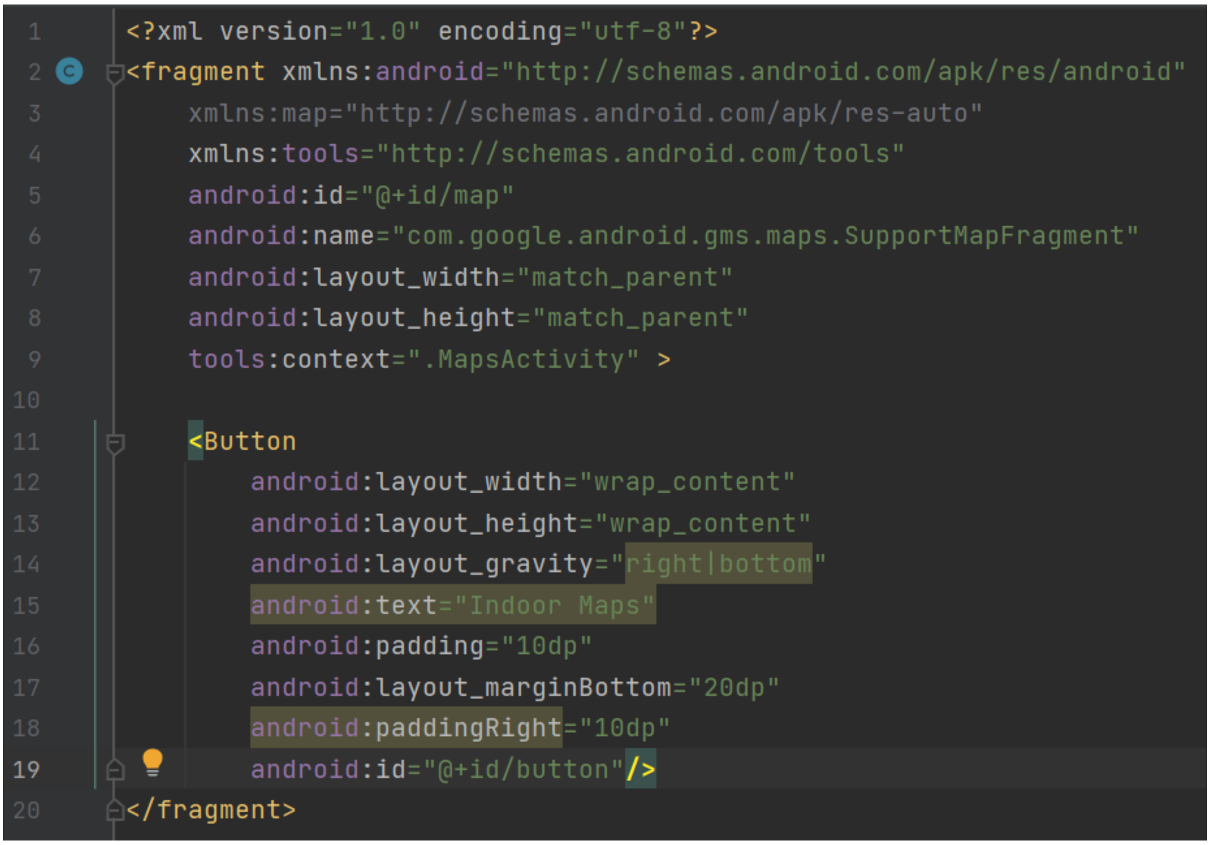


```java
                        .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN)));
        Marker mMainExit = mMap.addMarker(
                new MarkerOptions()
                        .position(mainExit)
                        .title("Main Campus Exit")
                        .snippet("No Entry")
                        .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN)));
        Marker mKilkEntry = mMap.addMarker(
                new MarkerOptions()
                        .position(kilkEntry)
                        .title("Kilkenny Road Entrance")
                        .snippet("Entry and Exit")
                        .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN)));
        Marker mGreenEntry = mMap.addMarker(
                new MarkerOptions()
                        .position(greenEntry)
                        .title("Green Road Entrance")
                        .snippet("Entry and Exit")
                        .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN)));

        // Set the camera location and zoom limitations
        mMap.moveCamera(CameraUpdateFactory.newLatLng(barrow));
        mMap.setMinZoomPreference(17.0f);

```



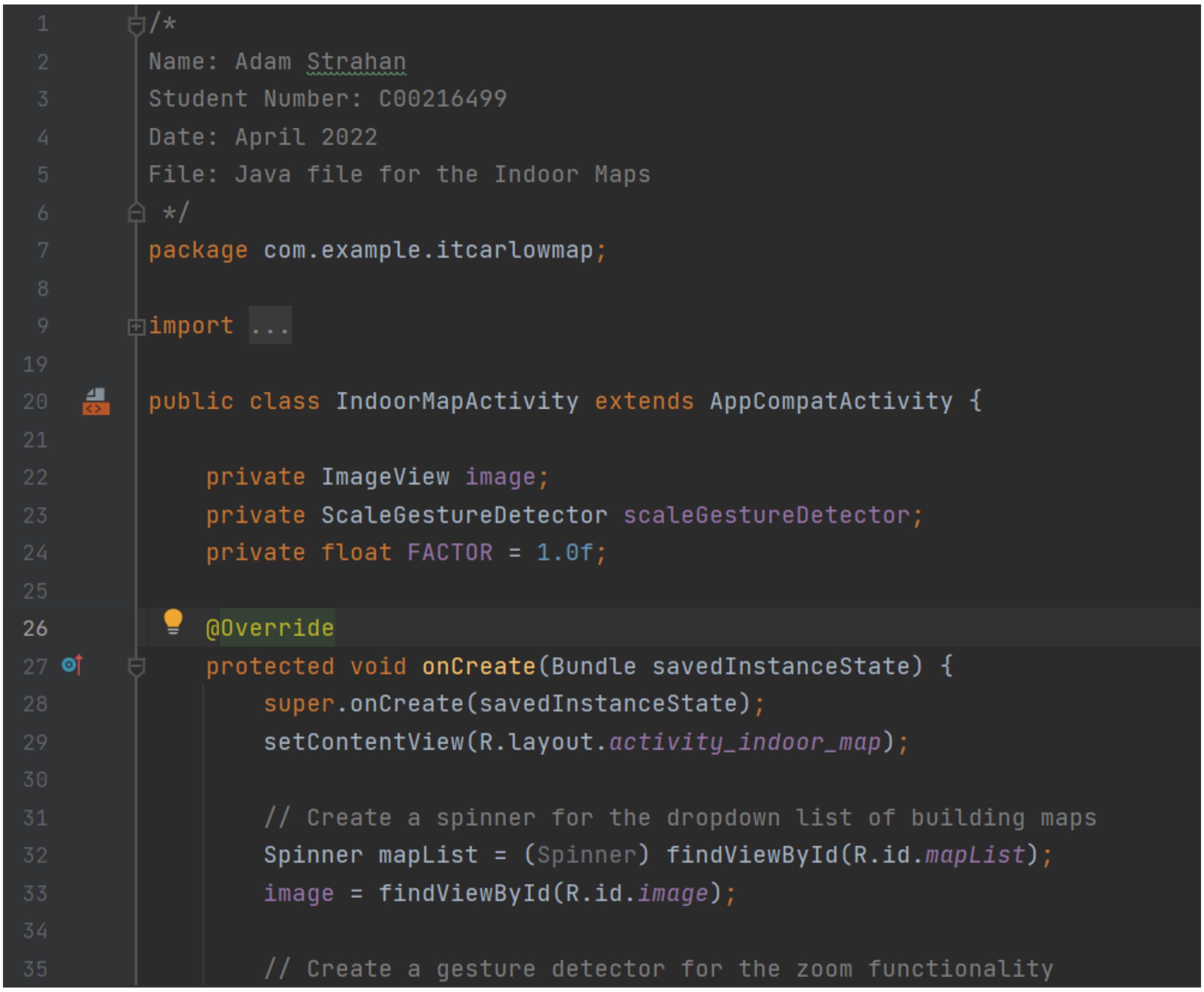
```java
        // Sets the info window for the markers
        mMap.setOnInfoWindowClickListener(this);
    }

    // Info window details
    @Override
    public void onInfoWindowClick(Marker marker) {
        Toast.makeText( context: this,  text: "For more information click 'Indoor Maps'", Toast.LENGTH_LONG).show();
    }
}
```

activity_maps.xml

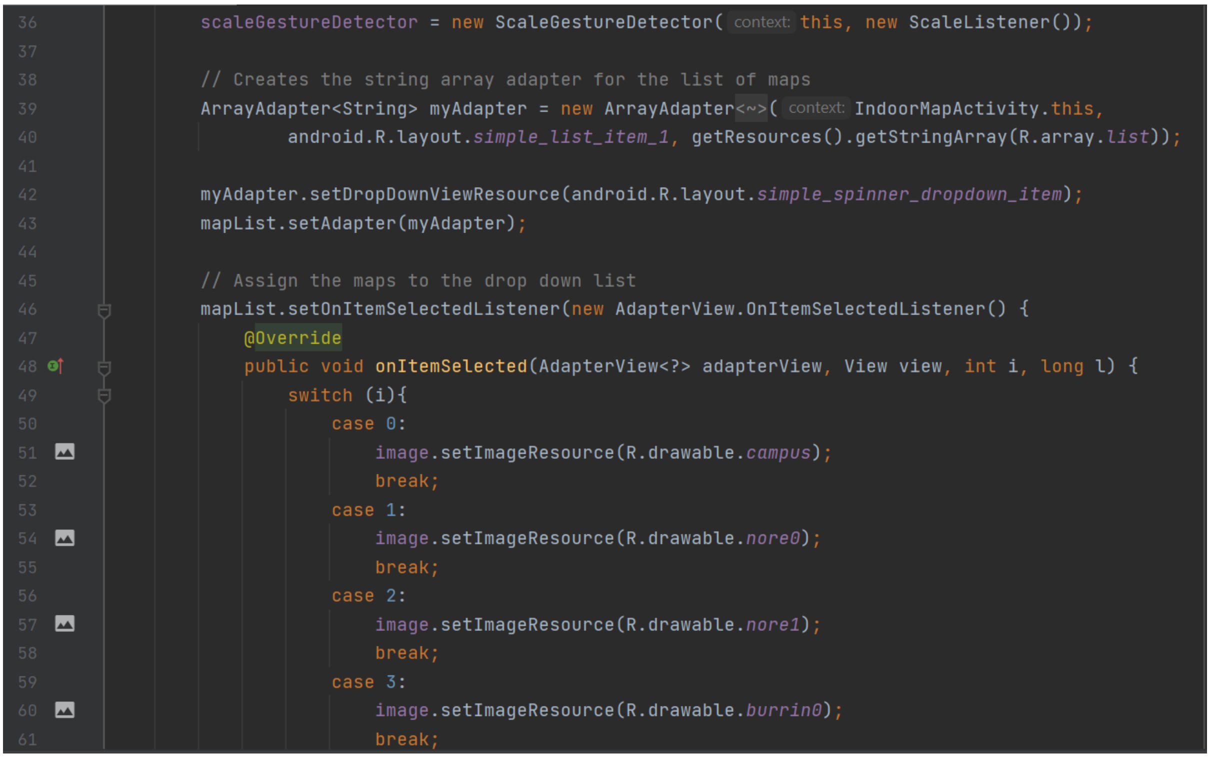


```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity" >

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="right|bottom"
        android:text="Indoor Maps"
        android:padding="10dp"
        android:layout_marginBottom="20dp"
        android:paddingRight="10dp"
        android:id="@+id/button"/>
</fragment>
```

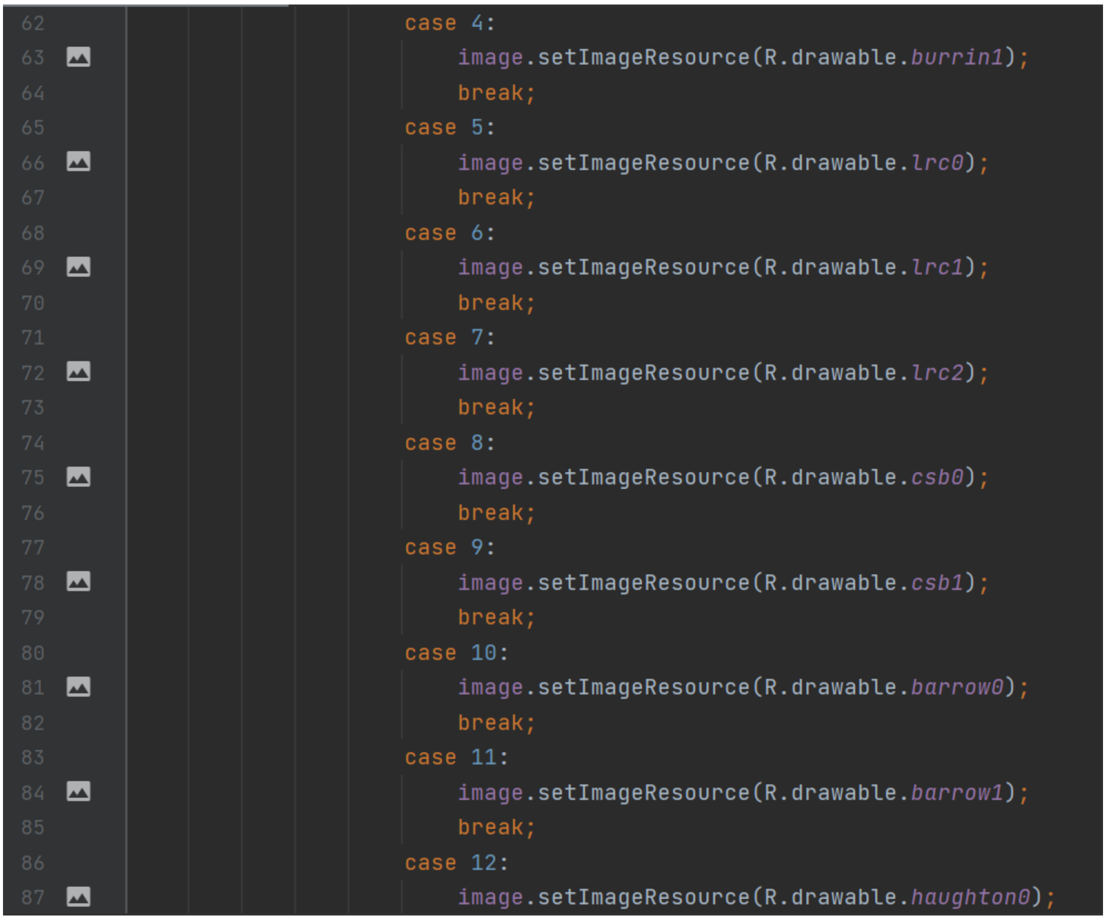
IndoorMapActivity.java



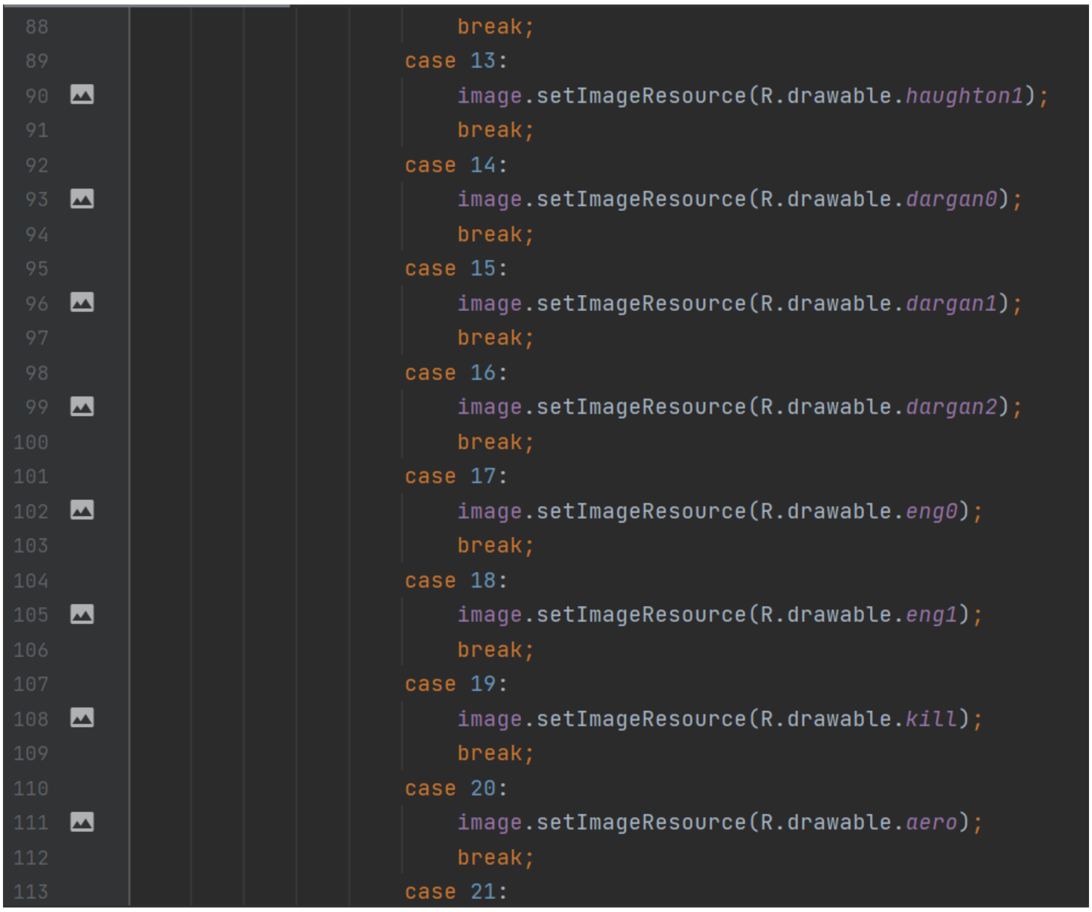
```java
/*
Name: Adam Strahan
Student Number: C00216499
Date: April 2022
File: Java file for the Indoor Maps
 */
package com.example.itcarlowmap;

import ...

public class IndoorMapActivity extends AppCompatActivity {

    private ImageView image;
    private ScaleGestureDetector scaleGestureDetector;
    private float FACTOR = 1.0f;

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

        // Create a spinner for the dropdown list of building maps
        Spinner mapList = (Spinner) findViewById(R.id.mapList);
        image = findViewById(R.id.image);

        // Create a gesture detector for the zoom functionality
```



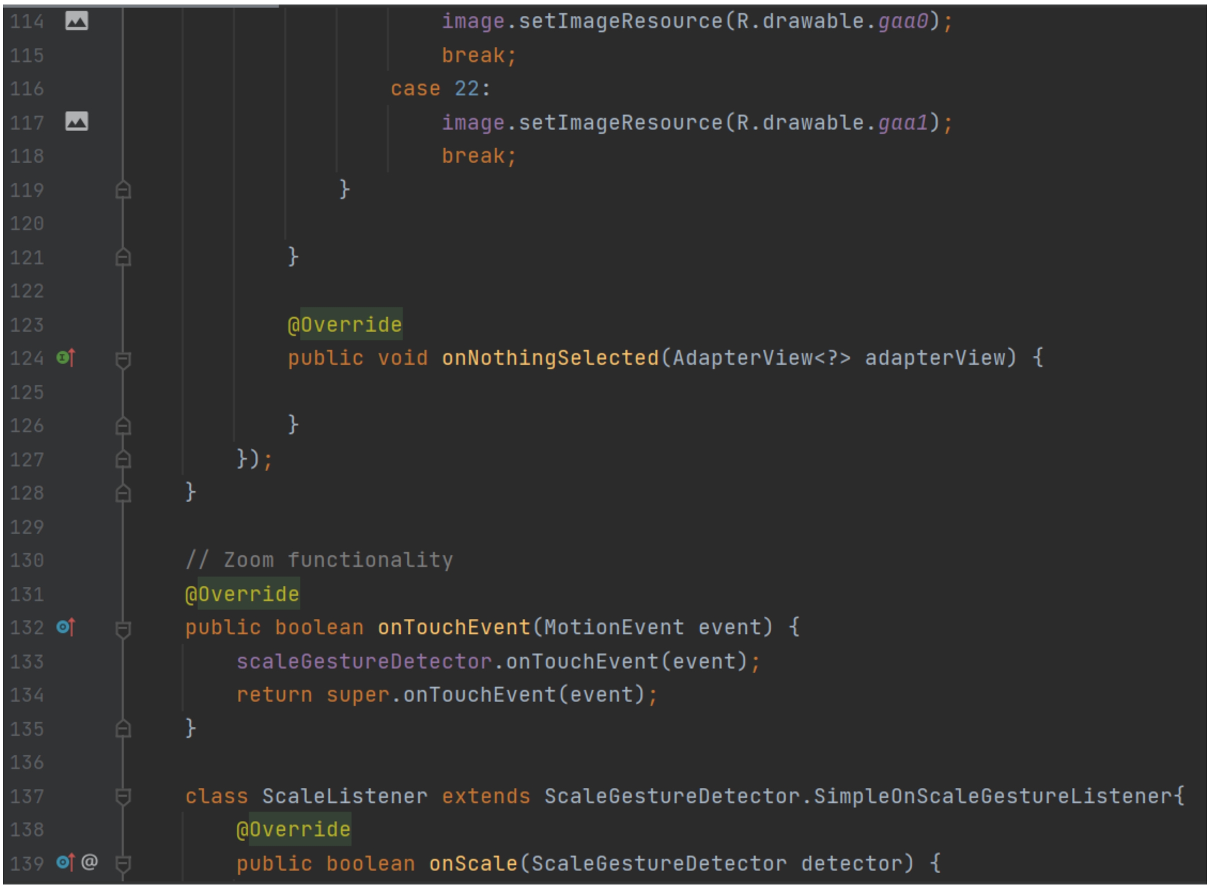
```java
        scaleGestureDetector = new ScaleGestureDetector( context: this, new ScaleListener());

        // Creates the string array adapter for the list of maps
        ArrayAdapter<String> myAdapter = new ArrayAdapter<~>( context: IndoorMapActivity.this,
                android.R.layout.simple_list_item_1, getResources().getStringArray(R.array.list));

        myAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        mapList.setAdapter(myAdapter);

        // Assign the maps to the drop down list
        mapList.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
                switch (i){
                    case 0:
                        image.setImageResource(R.drawable.campus);
                        break;
                    case 1:
                        image.setImageResource(R.drawable.nore0);
                        break;
                    case 2:
                        image.setImageResource(R.drawable.nore1);
                        break;
                    case 3:
                        image.setImageResource(R.drawable.burrin0);
                        break;
```



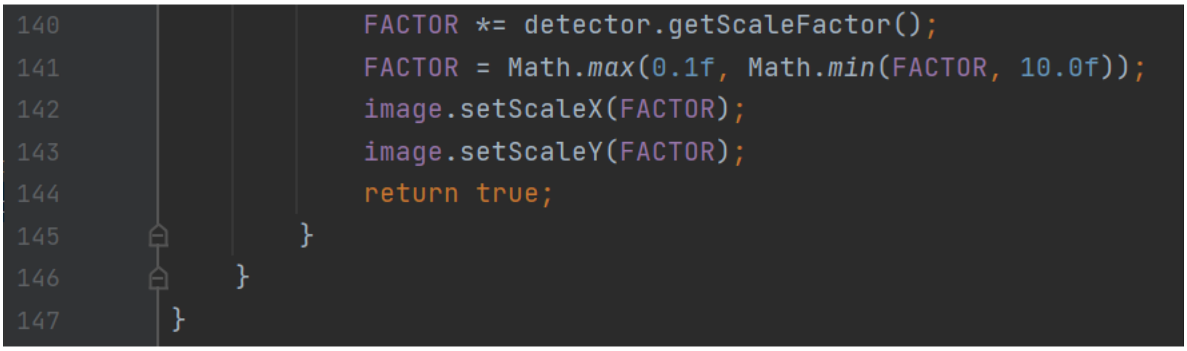
```
                    case 4:
                        image.setImageResource(R.drawable.burrin1);
                        break;
                    case 5:
                        image.setImageResource(R.drawable.lrc0);
                        break;
                    case 6:
                        image.setImageResource(R.drawable.lrc1);
                        break;
                    case 7:
                        image.setImageResource(R.drawable.lrc2);
                        break;
                    case 8:
                        image.setImageResource(R.drawable.csb0);
                        break;
                    case 9:
                        image.setImageResource(R.drawable.csb1);
                        break;
                    case 10:
                        image.setImageResource(R.drawable.barrow0);
                        break;
                    case 11:
                        image.setImageResource(R.drawable.barrow1);
                        break;
                    case 12:
                        image.setImageResource(R.drawable.haughton0);
```



```
                        break;
                    case 13:
                        image.setImageResource(R.drawable.haughton1);
                        break;
                    case 14:
                        image.setImageResource(R.drawable.dargan0);
                        break;
                    case 15:
                        image.setImageResource(R.drawable.dargan1);
                        break;
                    case 16:
                        image.setImageResource(R.drawable.dargan2);
                        break;
                    case 17:
                        image.setImageResource(R.drawable.eng0);
                        break;
                    case 18:
                        image.setImageResource(R.drawable.eng1);
                        break;
                    case 19:
                        image.setImageResource(R.drawable.kill);
                        break;
                    case 20:
                        image.setImageResource(R.drawable.aero);
                        break;
                    case 21:
```

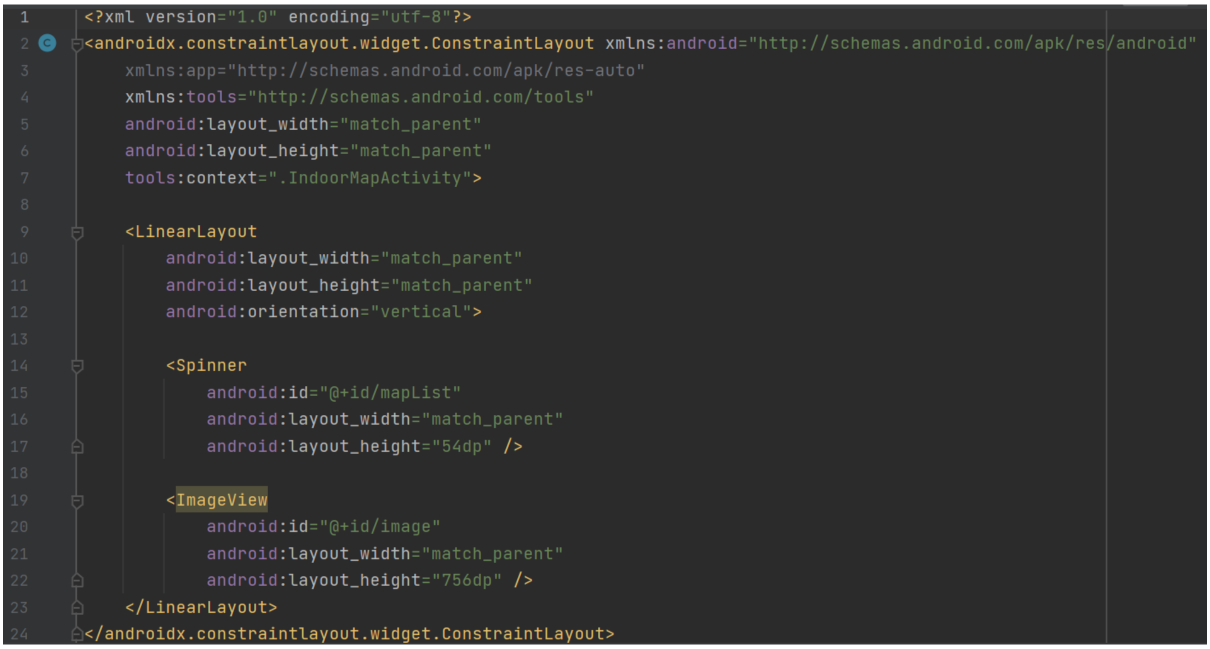


```java
                        image.setImageResource(R.drawable.gaa0);
                        break;
                    case 22:
                        image.setImageResource(R.drawable.gaa1);
                        break;
                }

            }

            @Override
            public void onNothingSelected(AdapterView<?> adapterView) {

            }
        });
    }

    // Zoom functionality
    @Override
    public boolean onTouchEvent(MotionEvent event) {
        scaleGestureDetector.onTouchEvent(event);
        return super.onTouchEvent(event);
    }

    class ScaleListener extends ScaleGestureDetector.SimpleOnScaleGestureListener{
        @Override
        public boolean onScale(ScaleGestureDetector detector) {
```



```java
            FACTOR *= detector.getScaleFactor();
            FACTOR = Math.max(0.1f, Math.min(FACTOR, 10.0f));
            image.setScaleX(FACTOR);
            image.setScaleY(FACTOR);
            return true;
        }
    }
}
```

activity_indoor_map.xml

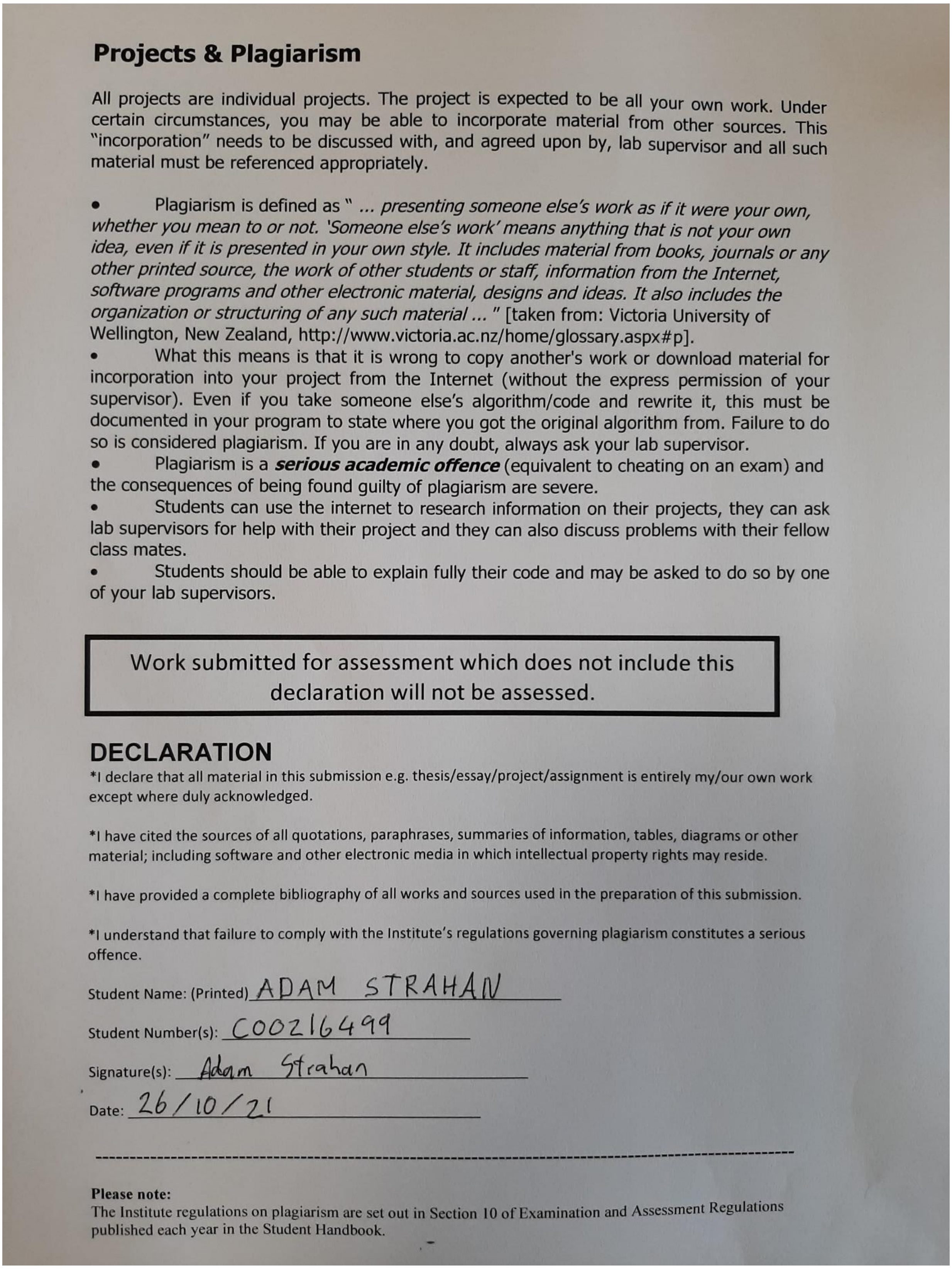


```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".IndoorMapActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <Spinner
            android:id="@+id/mapList"
            android:layout_width="match_parent"
            android:layout_height="54dp" />

        <ImageView
            android:id="@+id/image"
            android:layout_width="match_parent"
            android:layout_height="756dp" />
    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

strings.xml

Appendix

Plagiarism Declaration



Projects & Plagiarism

All projects are individual projects. The project is expected to be all your own work. Under certain circumstances, you may be able to incorporate material from other sources. This "incorporation" needs to be discussed with, and agreed upon by, lab supervisor and all such material must be referenced appropriately.

- Plagiarism is defined as " ... *presenting someone else's work as if it were your own, whether you mean to or not. 'Someone else's work' means anything that is not your own idea, even if it is presented in your own style. It includes material from books, journals or any other printed source, the work of other students or staff, information from the Internet, software programs and other electronic material, designs and ideas. It also includes the organization or structuring of any such material ...* " [taken from: Victoria University of Wellington, New Zealand, http://www.victoria.ac.nz/home/glossary.aspx#p].
- What this means is that it is wrong to copy another's work or download material for incorporation into your project from the Internet (without the express permission of your supervisor). Even if you take someone else's algorithm/code and rewrite it, this must be documented in your program to state where you got the original algorithm from. Failure to do so is considered plagiarism. If you are in any doubt, always ask your lab supervisor.
- Plagiarism is a ***serious academic offence*** (equivalent to cheating on an exam) and the consequences of being found guilty of plagiarism are severe.
- Students can use the internet to research information on their projects, they can ask lab supervisors for help with their project and they can also discuss problems with their fellow class mates.
- Students should be able to explain fully their code and may be asked to do so by one of your lab supervisors.

Work submitted for assessment which does not include this declaration will not be assessed.

DECLARATION

*I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.

*I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.

*I have provided a complete bibliography of all works and sources used in the preparation of this submission.

*I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offence.

Student Name: (Printed) ADAM STRAHAN

Student Number(s): C00216499

Signature(s): Adam Strahan

Date: 26/10/21

Please note:
The Institute regulations on plagiarism are set out in Section 10 of Examination and Assessment Regulations published each year in the Student Handbook.