In Class Lab: Making a google Maps Application
Create a google maps activity from the Android Wizard, add a toolbar with a spinner for map type. Add some menu items for places where you should go.

Create a new project, choose 'Google Maps Activity"

Starts with manifest visible with the following text

```
<!--
    TODO: Before you run your application, you need a Google Maps API key.

    To get one, follow the directions here:

       https://developers.google.com/maps/documentation/android-sdk/get-api-key

    Once you have your API key (it starts with "AIza"), define a new property in your
    project's local.properties file (e.g. MAPS_API_KEY=Aiza...), and replace the
    "YOUR_API_KEY" string in this file with "${MAPS_API_KEY}".
-->
<meta-data
   android:name="com.google.android.geo.API_KEY"
   android:value="YOUR_API_KEY" />
```

Go to the link shown above, create a key and copy it to the YOUR_API_KEY above. This will allow you to retrieve map tiles from Google. *Forget to do this, you get no tiles.*

In activity_maps.xml – change the lone fragment to the following, this gives you a toolbar with a spinner as well as a map fragment

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
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=".MapsActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary">
            <Spinner
                android:id="@+id/spinner"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:paddingRight="10dp" />
        </androidx.appcompat.widget.Toolbar>

    </com.google.android.material.appbar.AppBarLayout>

    <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" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

Add strings and menu items:

Add the following to strings.xml under res/values
```
<string-array name="map_types">
    <item>Normal</item>
    <item>Hybrid</item>
    <item>Satellite</item>
    <item>Terrain</item>
    <item>None</item>
</string-array>
```

Add a menu folder and a file called menu.xml under res. Put the following in menu.xml. These will be the location selection choices available in the overflow menu
```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/action_KP"
        app:showAsAction="never"
        android:title="KP"></item>

    <item
        android:id="@+id/action_NZ"
        app:showAsAction="never"
        android:title="Christchurch"></item>
    <item
        android:id="@+id/action_MT"
        app:showAsAction="never"
        android:title="Milford"></item>
    <item
        android:id="@+id/action_RT"
        app:showAsAction="never"
        android:title="Routeburn"></item>

</menu>
```

in res/values/styles change the theme, otherwise your app fails because you are creating an actionbar when one already exists
change
```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
```
  to
```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
```

In MainActivity

Change the interface from FragmentActivity to AppCompatActivity. This lets you set the actionbar (AppCompatActivity is a child of FragmentActivity).
```
public class MapsActivity extends AppCompatActivity implements OnMapReadyCallback {
```

Modify onCreate to set your actionBar
```
@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_maps);
   Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
   setSupportActionBar(toolbar);
   // Obtain the SupportMapFragment and get notified when the map is ready to be used.
   SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
         .findFragmentById(R.id.map);
   mapFragment.getMapAsync(this);
}
```

And in onMapReady(...)
Set up the spinner after the map is loaded and ready otherwise the user will try to change the map type before its loaded

```
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    //start at my house
    mMap.addMarker(new MarkerOptions().position(KP_HOUSE).title("Marker KP"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(KP_HOUSE));

    setupSimpleSpinner();
}
```

Override and add so that the overflow button appears in the Appbar

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu, menu);
    return true;
}

//handle item selection from the overflow menu
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {

        case R.id.action_KP:
            goToKP();
            break;

        case R.id.action_NZ:
            goToNZ();
            break;
        case R.id.action_MT:
            goToMT();
            break;
        case R.id.action_RT:
            goToRT();
            break;

        default:
            break;
    }
    return true;
}
```

Add some locations as class member variables

```
private static final LatLng KP_HOUSE = new LatLng(37.047291, -76.493837);
private static final LatLng  CC_NZ = new LatLng(-43.5321, 172.6362);

private static final LatLng  NZ_MT = new LatLng(-44.9083700, 167.9100500);
private static final LatLng  NZ_RT = new LatLng( -44.7283600, 168.1800600);


//some locations to travel to
private void goToKP() {
  CameraUpdate camera = CameraUpdateFactory.newLatLngZoom(KP_HOUSE, 15);
  mMap.addMarker(new MarkerOptions().position(CC_NZ).title("Keith and Lynns house"));
  mMap.animateCamera(camera);
}


private void goToNZ() {
    CameraUpdate camera = CameraUpdateFactory.newLatLngZoom(CC_NZ, 15);
    mMap.addMarker(new MarkerOptions().position(CC_NZ).title("Christchurch NZ"));
    mMap.animateCamera(camera);
}
private void goToMT() {
    CameraUpdate camera = CameraUpdateFactory.newLatLngZoom(NZ_MT, 15);
    mMap.addMarker(new MarkerOptions().position(NZ_MT).title("Milford Track NZ\nworlds best hike"));
    mMap.animateCamera(camera);
}

private void goToRT() {
    CameraUpdate camera = CameraUpdateFactory.newLatLngZoom(NZ_RT, 15);
    mMap.addMarker(new MarkerOptions().position(NZ_RT).title("Routeburn Track NZ\nworlds best hike"));
    mMap.animateCamera(camera);
}
```

And finally the spinner

```
AdapterView.OnItemSelectedListener mySpinnerListener;
private void setupSimpleSpinner() {

    Spinner spinner = (Spinner) findViewById(R.id.spinner);
    // Create an ArrayAdapter using the string array and a default spinner layout
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
            R.array.map_types, android.R.layout.simple_spinner_item);
    // Specify the layout to use when the list of choices appears
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    // Apply the adapter to the spinner
    spinner.setAdapter(adapter);

    //set listener
    mySpinnerListener = new AdapterView.OnItemSelectedListener() {

        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
            switch (position) {
                // Sets the map type
                case 0:
                    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
                    break;
                case 1:
                    mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
                    break;
                case 2:
                    mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
                    break;
                case 3:
                    mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
                    break;
                case 4:
                    mMap.setMapType(GoogleMap.MAP_TYPE_NONE);
                    break;
                default:
                    break;
            }

        }

        /**
         * Callback method to be invoked when the selection disappears from this
         * view. The selection can disappear for instance when touch is activated
         * or when the adapter becomes empty.
         *
         * @param parent The AdapterView that now contains no selected item.
         */
        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
    };

    //respond when spinner clicked
    spinner.setOnItemSelectedListener(mySpinnerListener);
}
```