

Maps - Control

Lab-08



location · marker · constraint ·
layout · landscape · googlemap
· mapview

PlacemarkActivity

(start lab08)

```
class PlacemarkActivity : AppCompatActivity(), AnkoLogger {  
  
    var placemark = PlacemarkModel()  
    lateinit var app: MainApp  
    var edit = false  
    val IMAGE_REQUEST = 1  
    val LOCATION_REQUEST = 2  
  
    override fun onCreate(savedInstanceState: Bundle?) {...}  
  
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {...}  
  
    fun save() {...}  
  
    override fun onOptionsItemSelected(item: MenuItem?): Boolean {...}  
  
    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {...}  
}
```

Placemark

SAVE

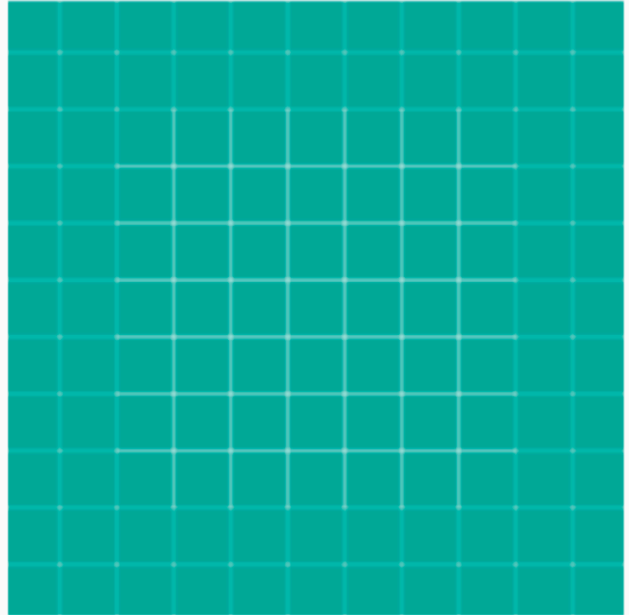
CANCEL

Placemark Title

Description

ADD IMAGE

SET LOCATION



PlacemarkActivity - onCreate

```
var placemark = PlacemarkModel()
lateinit var app: MainApp
var edit = false
val IMAGE_REQUEST = 1
val LOCATION_REQUEST = 2

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
    app = application as MainApp

    toolbarAdd.title = title
    setSupportActionBar(toolbarAdd)

    if (intent.hasExtra(name: "placemark_edit")) {...}

    chooseImage.setOnClickListener {...}

    placemarkLocation.setOnClickListener {...}
}
```

Placemark

SAVE

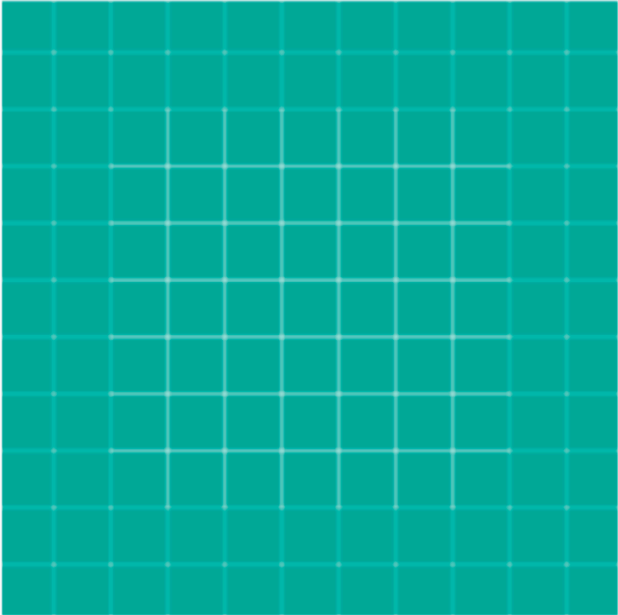
CANCEL

Placemark Title

Description

ADD IMAGE

SET LOCATION



PlacemarkActivity - Edit

```
var placemark = PlacemarkModel()
lateinit var app: MainApp
var edit = false
val IMAGE_REQUEST = 1
val LOCATION_REQUEST = 2

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
    app = application as MainApp

    toolbarAdd.title = title
    setSupportActionBar(toolbarAdd)

    if (intent.hasExtra(name: "placemark_edit")) {
        edit = true
        placemark = intent.extras.getParcelable<PlacemarkModel>(key: "placemark_edit")
        placemarkTitle.setText(placemark.title)
        description.setText(placemark.description)
        placemarkImage.setImageBitmap(readImageFromPath(context: this, placemark.image))
        if (placemark.image != null) {
            chooseImage.setText("Change image")
        }
    }

    chooseImage.setOnClickListener {...}

    placemarkLocation.setOnClickListener {...}
}
```

Placemark

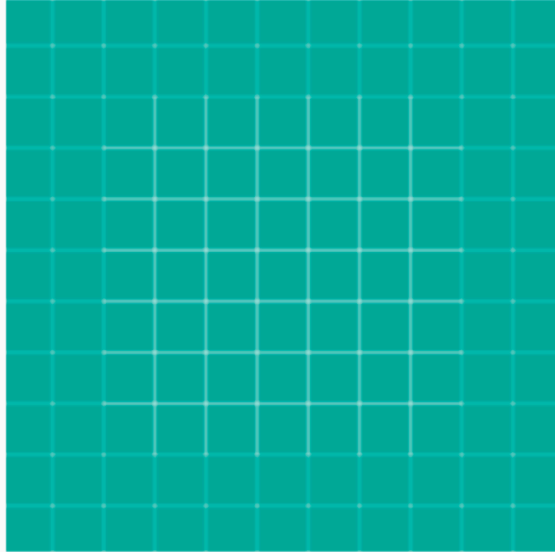
SAVECANCEL

Placemark Title

Description

ADD IMAGE

SET LOCATION



PlacemarkActivity - Edit

```
var placemark = PlacemarkModel()
lateinit var app: MainApp
var edit = false
val IMAGE_REQUEST = 1
val LOCATION_REQUEST = 2

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
    app = application as MainApp

    toolbarAdd.title = title
    setSupportActionBar(toolbarAdd)

    if (intent.hasExtra(name: "placemark_edit")) {...}

    chooseImage.setOnClickListener { it: View!
        showImagePicker(parent: this, IMAGE_REQUEST)
    }

    placemarkLocation.setOnClickListener { it: View!
        val location = Location(lat: 52.245696, lng: -7.139102, zoom: 15f)
        if (placemark.zoom != 0f) {
            location.lat = placemark.lat
            location.lng = placemark.lng
            location.zoom = placemark.zoom
        }
        startActivityForResult(intentFor<MapsActivity>().putExtra(name: "location", location), LOCATION_REQUEST)
    }
}
```

Placemark

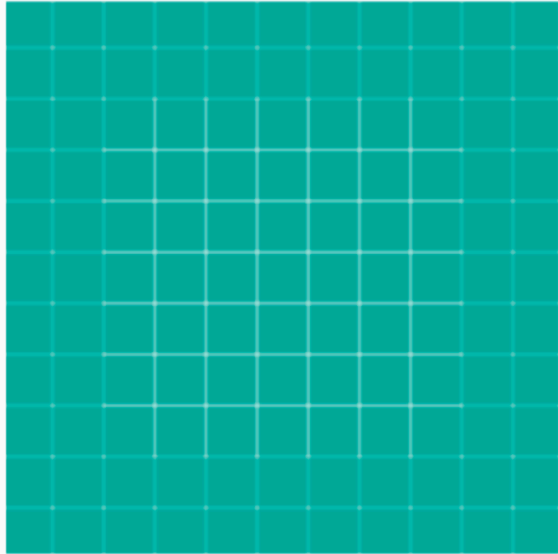
SAVECANCEL

Placemark Title

Description

ADD IMAGE

SET LOCATION



PlacemarkActivity - Menu

```
override fun onCreateOptionsMenu(menu: Menu?): Boolean {  
    menuInflater.inflate(R.menu.menu_placemark, menu)  
    return super.onCreateOptionsMenu(menu)  
}  
  
override fun onOptionsItemSelected(item: MenuItem?): Boolean {  
    when (item?.itemId) {  
        R.id.item_save -> {  
            save()  
        }  
        R.id.item_cancel -> {  
            setResult(RESULT_CANCELED)  
            finish()  
        }  
    }  
    return super.onOptionsItemSelected(item)  
}
```

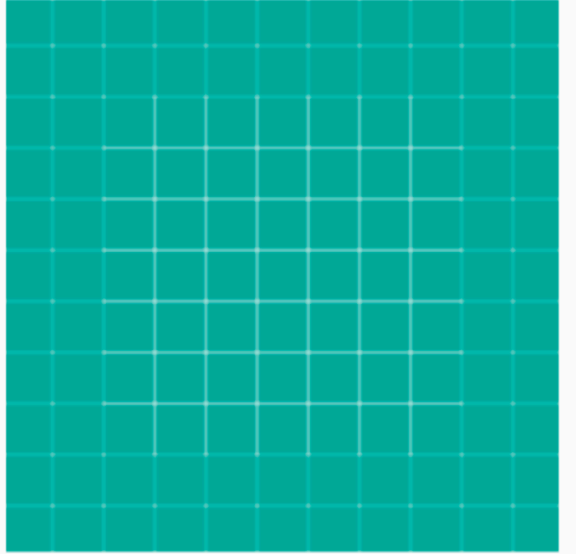
Placemark

SAVECANCEL

Placemark Title

Description

ADD IMAGESET LOCATION



PlacemarkActivity - Save

```
fun save() {  
    placemark.title = placemarkTitle.text.toString()  
    placemark.description = description.text.toString()  
  
    if (edit) {  
        app.placemarks.update(placemark.copy())  
        setResult(201)  
        finish()  
    } else {  
        if (placemark.title.isEmpty()) {  
            app.placemarks.create(placemark.copy())  
            setResult(200)  
            finish()  
        } else {  
            toast("Please enter a Placemark Title")  
        }  
    }  
}
```

Placemark

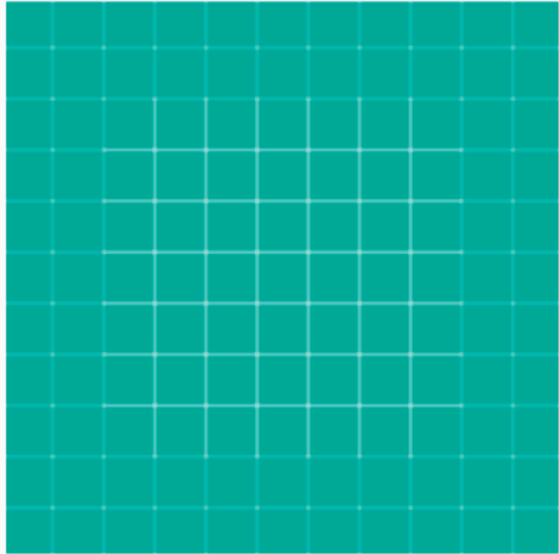
SAVECANCEL

Placemark Title

Description

ADD IMAGE

SET LOCATION



PlacemarkActivity - onActivityResult

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  
    super.onActivityResult(requestCode, resultCode, data)  
    when (requestCode) {  
        IMAGE_REQUEST -> {  
            if (data != null) {  
                placemark.image = data.getData().toString()  
                placemarkImage.setImageBitmap(readImage(activity: this, resultCode, data))  
                chooseImage.setText("Change image")  
            }  
        }  
        LOCATION_REQUEST -> {  
            if (data != null) {  
                val location = data.extras.getParcelable<Location>(key: "location")  
                placemark.lat = location.lat  
                placemark.lng = location.lng  
                placemark.zoom = location.zoom  
            }  
        }  
    }  
}
```

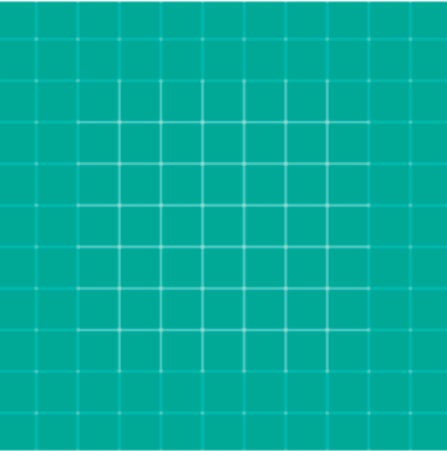
Placemark

SAVECANCEL

Placemark Title

Description

ADD IMAGESET LOCATION



Placemark

SAVE

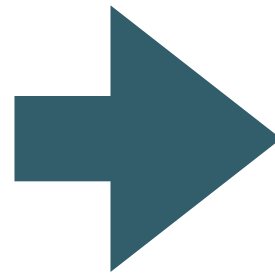
CANCEL

Placemark Title

Description

ADD IMAGE

SET LOCATION

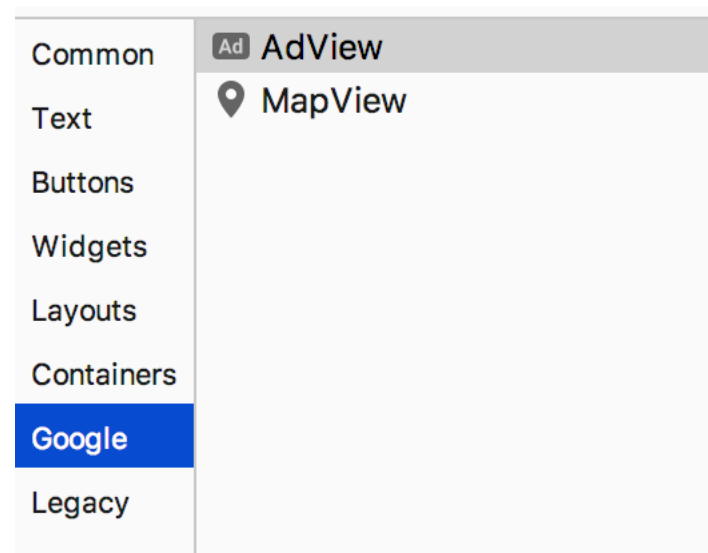


Placemark Title

Description

ADD IMAGE

SET LOCATION



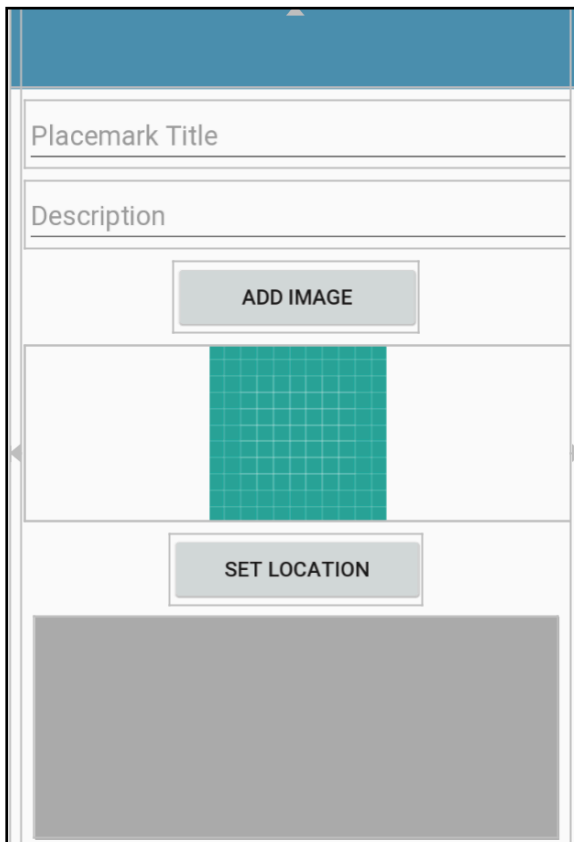
MapView
added to
layout

Revised Placemark

```
class PlacemarkActivity : AppCompatActivity(), AnkoLogger {  
  
    var placemark = PlacemarkModel()  
    lateinit var app: MainApp  
    var edit = false  
    val IMAGE_REQUEST = 1  
    val LOCATION_REQUEST = 2  
  
    override fun onCreate(savedInstanceState: Bundle?) {...}  
  
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {...}  
  
    fun save() {...}  
  
    override fun onOptionsItemSelected(item:  
  
    override fun onActivityResult(requestCode:  
}
```

New Objects

- GoogleMap
- defaultLocation



```
class PlacemarkActivity : AppCompatActivity(), AnkoLogger {  
  
    var placemark = PlacemarkModel()  
    lateinit var app: MainApp  
    lateinit var map: GoogleMap  
    var edit = false  
    val IMAGE_REQUEST = 1  
    val LOCATION_REQUEST = 2  
    val defaultLocation = Location(lat: 52.245696, lng: -7.139102, zoom: 15f)  
  
    override fun onCreate(savedInstanceState: Bundle?) {...}  
  
    fun configureMap() {...}  
  
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {...}  
  
    fun save() {...}  
  
    override fun onOptionsItemSelected(item: MenuItem?): Boolean {...}  
  
    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {...}  
  
    override fun onDestroy() {...}  
  
    override fun onLowMemory() {...}  
  
    override fun onPause() {...}  
  
    override fun onResume() {...}  
  
    override fun onSaveInstanceState(outState: Bundle?) {...}  
}
```

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
    app = application as MainApp

    toolbarAdd.title = title
    setSupportActionBar(toolbarAdd)

    if (intent.hasExtra( name: "placemark_edit")) {...}

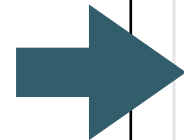
    chooseImage.setOnClickListener {...}

    placemarkLocation.setOnClickListener {...}
}

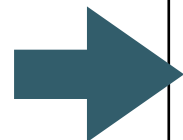
```

Revised onCreate

initialise MapView



Request
GoogleMap
object reference



```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_placemark)
    mapView.onCreate(savedInstanceState);
    app = application as MainApp

    toolbarAdd.title = title
    setSupportActionBar(toolbarAdd)

    mapView.getMapAsync { it: GoogleMap!
        map = it
        configureMap()
    }
}

```



Store reference
+ initialise the
map

configureMap

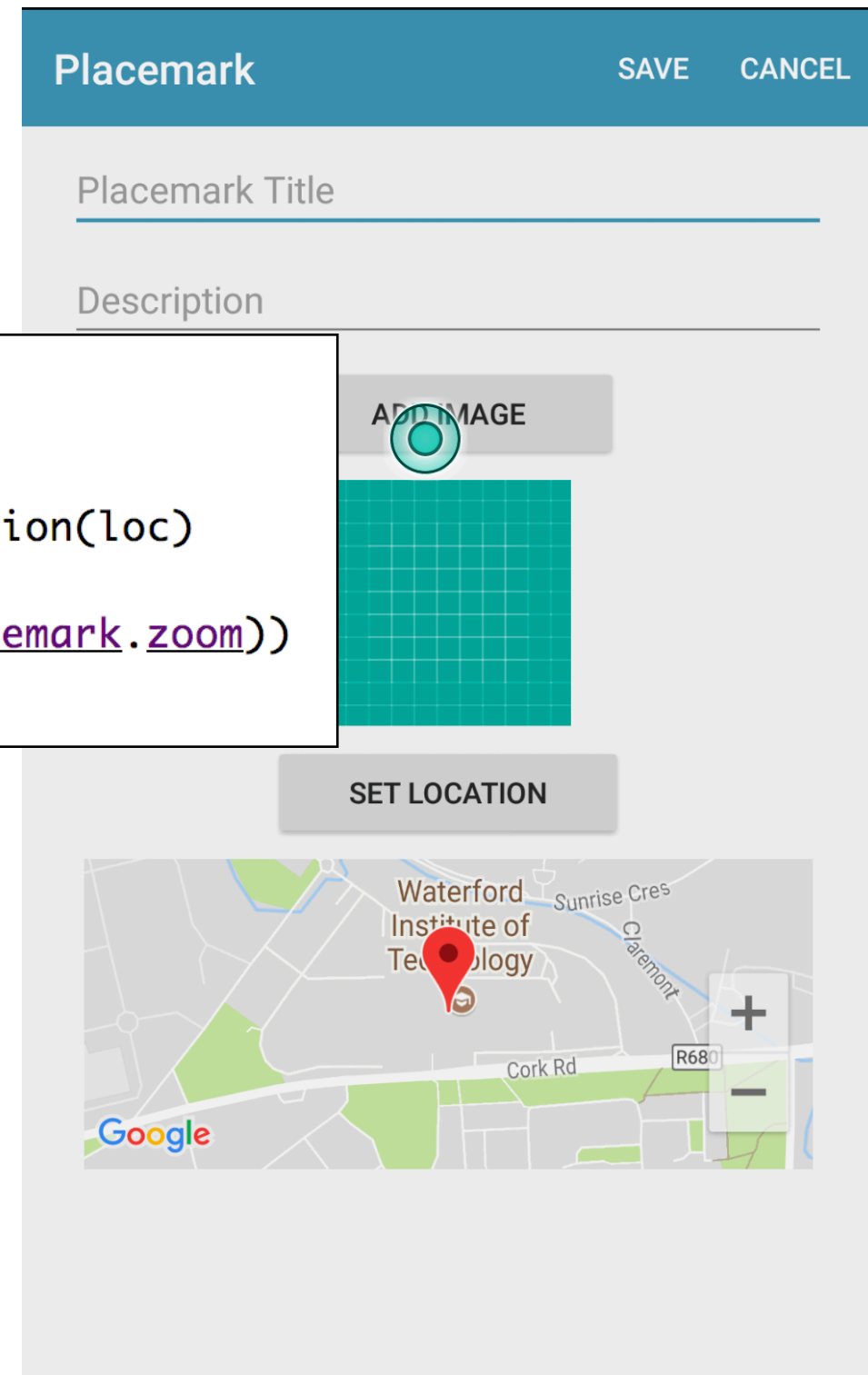
```
fun configureMap() {  
    map.uiSettings.setZoomControlsEnabled(true)  
    val loc = LatLng(placemark.lat, placemark.lng)  
    val options = MarkerOptions().title(placemark.title).position(loc)  
    map.addMarker(options)  
    map.moveCamera(CameraUpdateFactory.newLatLngZoom(loc, placemark.zoom))  
}
```

Enable Zoom controls (+-)

Create a marker

Add Marker to map

Zoom to placemark zoom setting



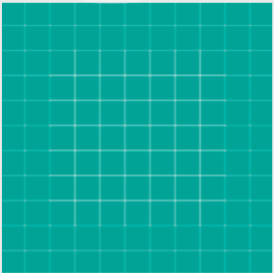
GoogleMap Lifecycle Management

Placemark SAVE CANCEL


Placemark Title

Description

ADD IMAGE



SET LOCATION



```
override fun onLowMemory() {  
    super.onLowMemory()  
    mapView.onLowMemory()  
}  
  
override fun onPause() {  
    super.onPause()  
    mapView.onPause()  
}  
  
override fun onResume() {  
    super.onResume()  
    mapView.onResume()  
}  
  
override fun onSaveInstanceState(outState: Bundle?) {  
    super.onSaveInstanceState(outState)  
    mapView.onSaveInstanceState(outState)  
}
```