

모바일 GIS 실습

A. 실습할 프로젝트 소개

- 첫째, 스마트 폰 또는 태블릿 PC의 화면에 지도를 표시하고
- 둘째, 지도를 레이어(Layer) 단위로 구성하며
- 셋째, 구성된 레이어의 색상 등의 심벌을 지정하고
- 넷째, 구성된 레이어의 라벨을 표시하며
- 다섯째, 표시된 건물을 터치하면 터치된 건물의 속성 정보를 제공하고
- 여섯째, 현재 내 위치로 지도를 이동함

B. 사용할 지도 데이터 살펴보기

레이어 이름	설명	형상
SJ_EMD	세종시의 '읍면동' 단위 행정구역	Polygon
SJ_LI	세종시의 '리' 단위 행정구역	Polygon
SJ_MANAGE	세종시의 전체 도로 중심선	Polyline
SJ_RW	세종시의 전체 실폭 도로	Polygon
SJ_BULD	세종시의 전체 건물	Polygon

*SJ_EMD*의 속성 구조

필드명	설명	타입
EMD_CD	읍면동코드	VARCHAR2(10)
EMD_KOR_NM	읍면동명_한글	VARCHAR2(40)
EMD_ENG_NM	읍면동명_영문	VARCHAR2(40)

*SJ_LI*의 속성 구조

필드명	설명	타입
LI_CD	리코드	VARCHAR2(10)
LI_KOR_NM	리명_한글	VARCHAR2(40)
LI_ENG_NM	리명_영문	VARCHAR2(40)

*SJ_BULD*의 속성 구조

필드명	설명	타입
SIG_CD	시군구코드	VARCHAR2(5)
BUL_MAN_NO	건물일련번호	NUMBER(7)
RN_CD	도로명코드	VARCHAR2(7)

RDS_MAN_NO	도로구간일련번호	NUMBER(12)
BSI_INT_SN	기초구간일련번호	NUMBER(10)
EQB_MAN_SN	건물군일련번호	NUMBER(10)
BULD_SE_CD	건물구분코드	VARCHAR2(1)
BULD_MNNM	건물본번	NUMBER(5)
BULD_SLNO	건물부번	NUMBER(5)
BULD_NM	건물명	VARCHAR2(40)
BUL_ENG_NM	건물영문명	VARCHAR2(40)
BULD_NM_DC	상세건물명	VARCHAR2(100)
BULD_STTUS	건물상태코드	VARCHAR2(40)
BDTYP_CD	건물용도코드	VARCHAR2(5)
BUL_DPN_SE	건물종속구분	VARCHAR2(1)
GRO_FLO_CO	지상층수	NUMBER(3)
UND_FLO_CO	지하층수	NUMBER(3)
ZIP	우편번호	VARCHAR2(7)
POS_BUL_NM	다량배달처건물명	VARCHAR2(40)
POS_BUL_YN	다량배달처여부	VARCHAR2(1)
REG_PUB_NM	등록공공문서명	VARCHAR2(20)
EMD_CD	읍면동코드	VARCHAR2(3)
LI_CD	리코드	VARCHAR2(2)
MNTN_YN	산여부	VARCHAR2(1)
LNBR_MNNM	지번본번	NUMBER(4)
LNBR_SLNO	지번부번	NUMBER(4)
COMPET_DE	완료일자	VARCHAR2(8)
NTFC_DE	고시일자	VARCHAR2(8)
MVM_RES_CD	이동사유코드	VARCHAR2(10)
MVMN_RESN	이동사유	VARCHAR2(254)
MVMN_DE	이동일자	VARCHAR2(8)
OPE_MAN_ID	작업자아이디	VARCHAR2(20)
OPERT_DE	작업일시	VARCHAR2(14)
IMA_FIL_SN	이미지파일일련번호	NUMBER(11)
BSI_ZON_NO	기초구역번호	NUMBER(5)
NTI_TRG_YN	고지대상여부	VARCHAR2(1)
INPUT_MTHD	입력방법	VARCHAR2(1)
BD_MGT_SN	이전건물관리번호	VARCHAR2(25)
ISSU_YN	발급여부	VARCHAR2(1)
XGEOMETRY	도형정보	LONG RAW
ZIP_BUL_NM	다량배달처건물명	VARCHAR2(1)

ETC_BUL_NM	기타상세건물명	VARCHAR2(40)
ZIP_NO	우편번호일련번호	VARCHAR2(3)
RDS_SIG_CD	도로구간시군구코드	VARCHAR2(5)

*SJ_MANAGE*의 속성 구조

필드명	설명	타입
SIG_CD	시군구코드	VARCHAR2(5)
RDS_MAN_NO	도로구간일련번호	NUMBER(12)
RN	도로명	VARCHAR2(80)
RN_CD	도로명코드	VARCHAR2(7)
ENG_RN	영문도로명	VARCHAR2(80)
NTFC_DE	고시일자	VARCHAR2(8)
RN_DLB_DE	도로명심의일자	VARCHAR2(8)
ROA_MAN_ES	도로제정권자	VARCHAR2(20)
WDR_RD_CD	광역도로구분코드	VARCHAR2(10)
ROA_CLS_SE	도로위계기능구분	VARCHAR2(2)
RDS_DPN_SE	도로구간종속구분	VARCHAR2(1)
RBP_CN	기점	VARCHAR2(80)
REP_CN	종점	VARCHAR2(80)
ROAD_BT	도로폭	NUMBER(10,3)
ROAD_LT	도로길이	NUMBER(10,3)
BSI_INT	기초간격	VARCHAR2(2)
NLR_LCL_NO	국도/지방도번호	VARCHAR2(10)
ALWNC_RESN	부여사유	VARCHAR2(254)
ALWNC_DE	부여일자	VARCHAR2(8)
MVM_RES_CD	이동사유코드	VARCHAR2(10)
MVMN_RESN	이동사유	VARCHAR2(254)
MVMN_DE	이동일자	VARCHAR2(8)
OPE_MAN_ID	작업자아이디	VARCHAR2(20)
OPERT_DE	작업일시	VARCHAR2(14)
PAR_SIG_CD	분기시군구코드	VARCHAR2(5)
PAR_RDS_NO	분기도로구간일련번호	NUMBER(12)
INPUT_MTHD	입력방법	VARCHAR2(1)
CRSRD_CNT	교차로수	NUMBER(3)
ISSU_YN	발급여부	VARCHAR2(1)

*SJ_RW*의 속성 구조

필드명	설명	타입
-----	----	----

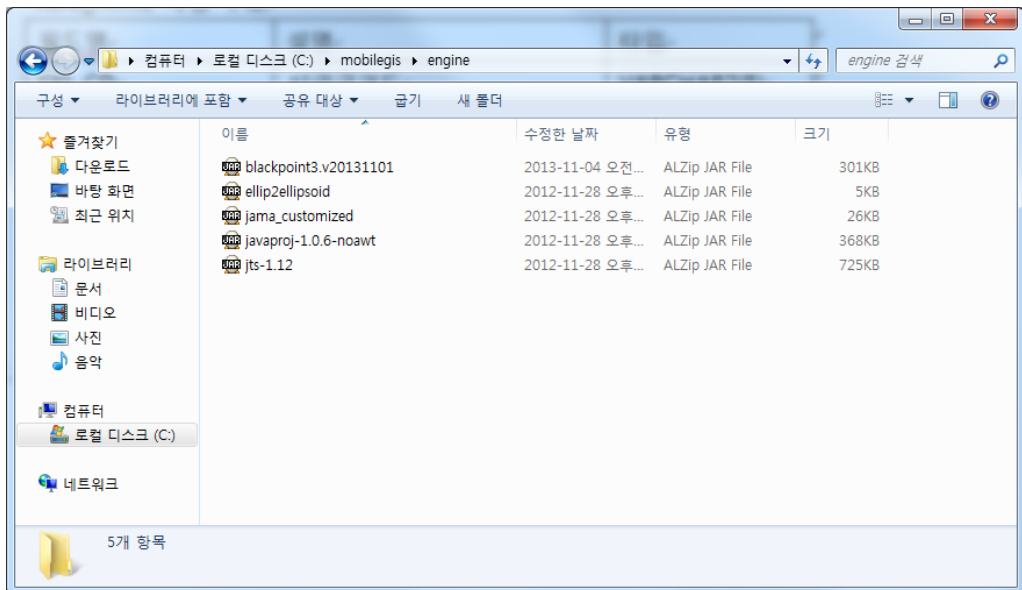
SIG_CD	시군구코드	VARCHAR2(5)
RW_SN	실폭도로일련번호	NUMBER(12)
ROA_CLS_SE	도로위계기능구분	VARCHAR2(2)
RDS_MAN_NO	도로구간일련번호	NUMBER(12)
OPE_MAN_ID	작업자아이디	VARCHAR2(20)
OPERT_DE	작업일시	VARCHAR2(14)

C. 개발 프로젝트 구성

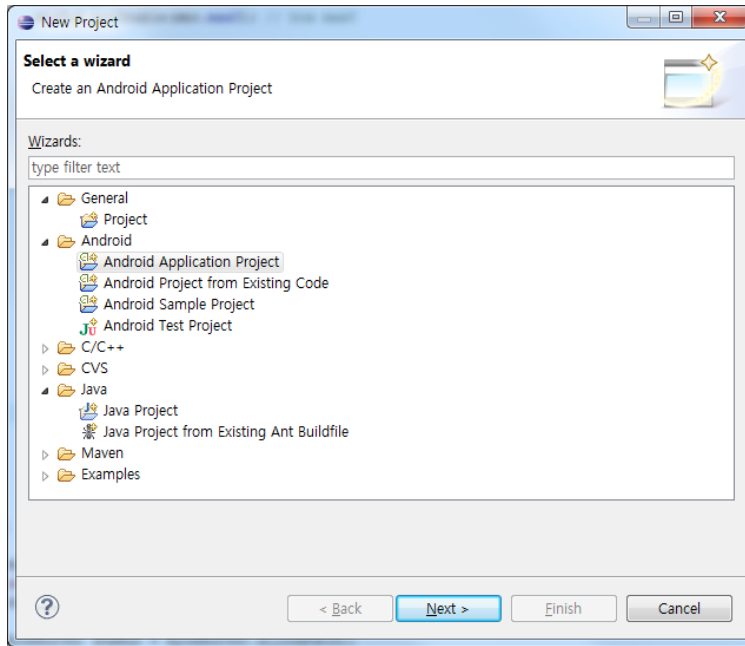
가장 먼저 모바일 GIS 엔진인 블랙포인트를 아래의 URL을 통해 다운로드 받습니다.

http://www.geoservice.co.kr/upload/20131101/blackpoint_v3.1.20131101.zip

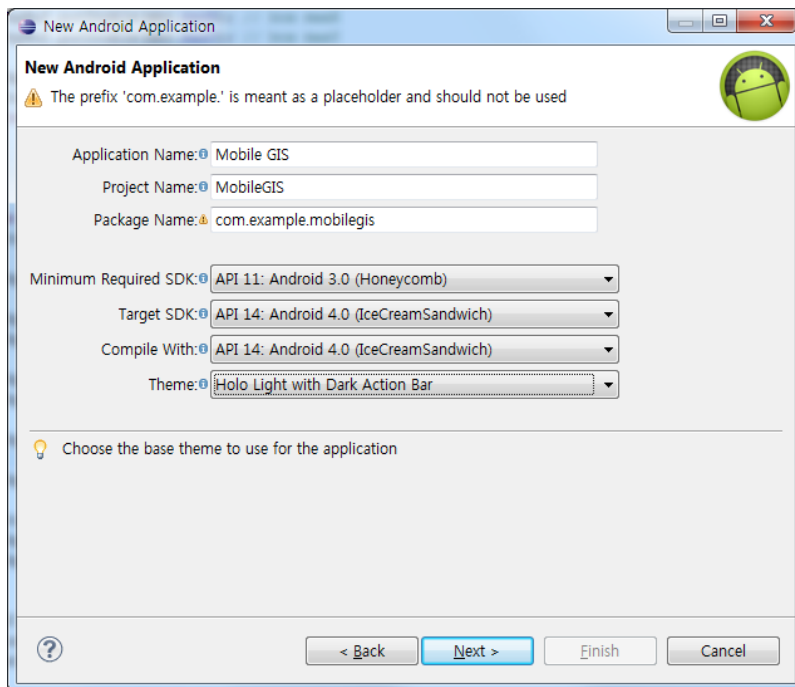
다운로드 받은 압축 파일을 C 드라이브의 mobilegis 폴더의 서브 폴더인 engine에 압축을 풉니다. (폴더가 없으면 생성합니다)



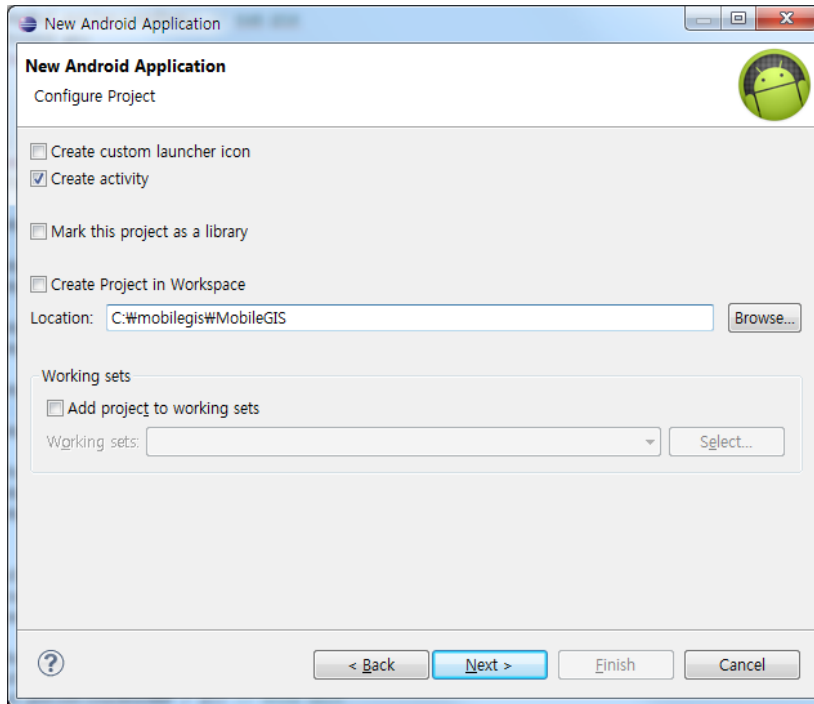
개발툴인 이클립스(Eclipse)를 실행하고 New Project에서 Android Application Project를 선택하고 Next 버튼을 클릭합니다.



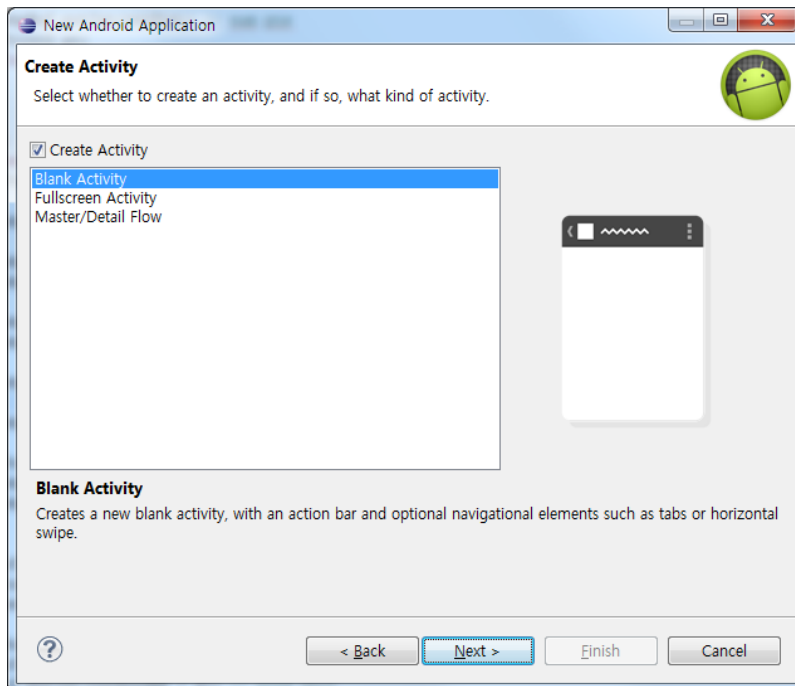
새롭게 나타나는 대화창에서 아래와 같이 입력하고 Next 버튼을 클릭합니다.



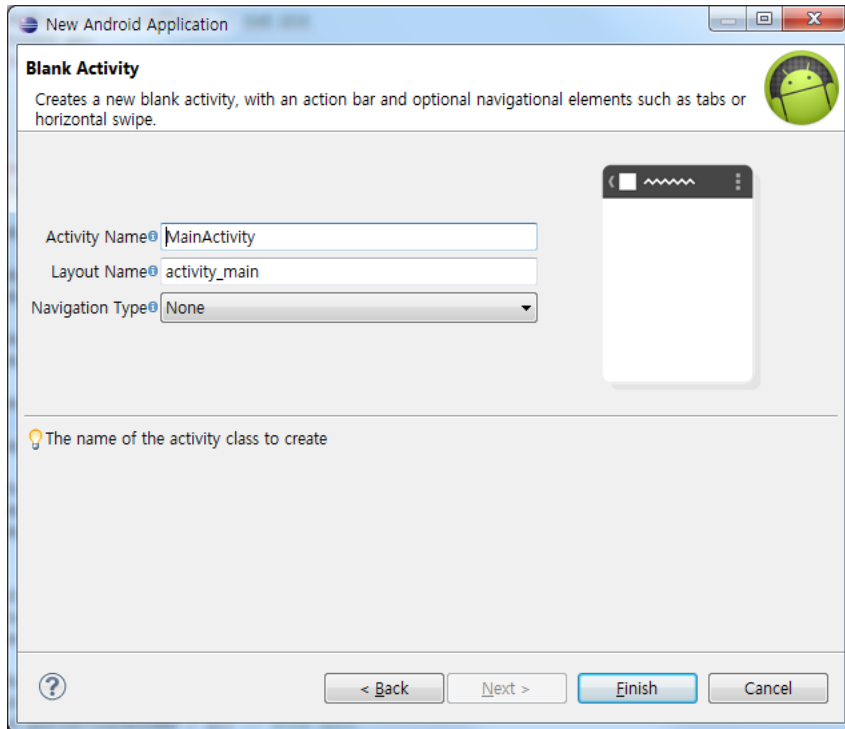
새롭게 표시되는 대화창에서 아래와 같이 입력하고 Next 버튼을 클릭합니다.



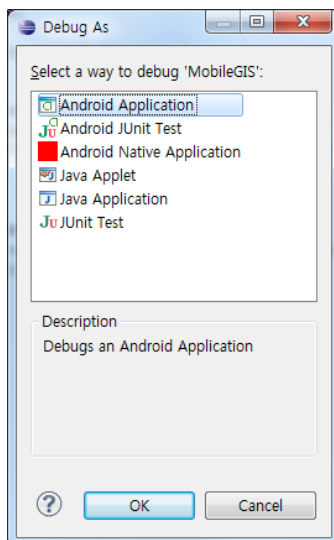
새롭게 표시되는 대화창에서 아래와 같이 입력(기본값)하고 Next 버튼을 클릭합니다.



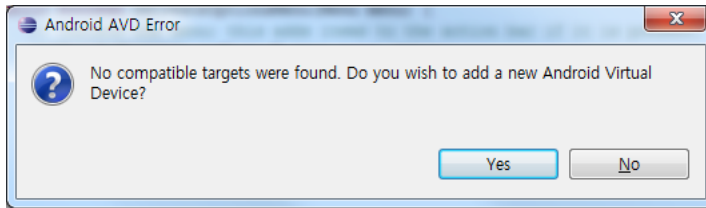
새롭게 표시되는 대화창에서 아래와 같이 입력(기본값)하고 Finish 버튼을 클릭합니다.



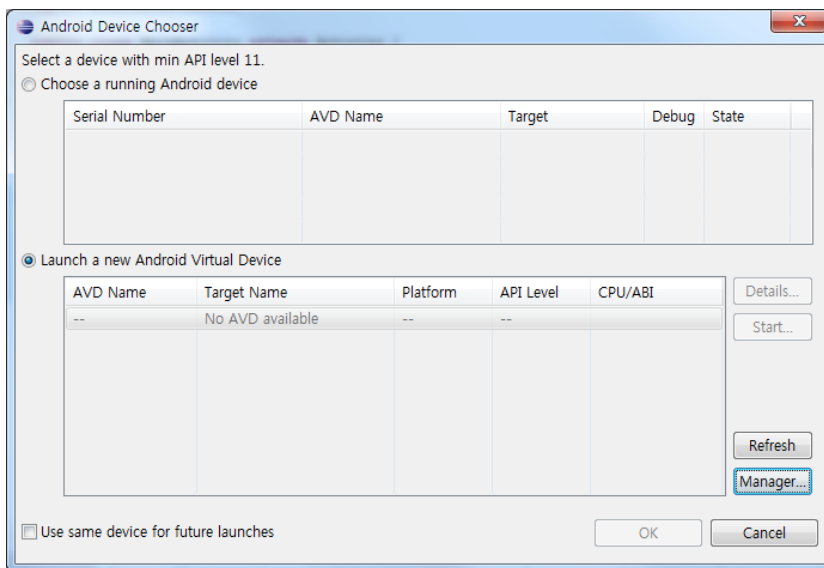
실행을 위해 F11 키를 누르면 아래와 같은 대화창이 표시되는데 여기서 Android Application을 선택하고 OK 버튼을 클릭합니다.



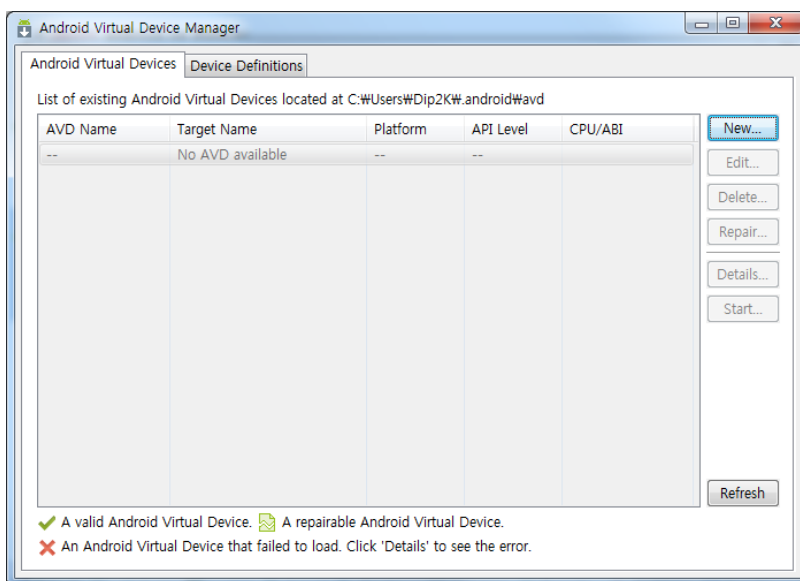
아래와 같은 대화창이 표시되는데 안드로이드 가상 단말기(Android Virtual Device, 이하 AVD라 함)를 생성할 것을 묻는 것으로 Yes를 클릭합니다.



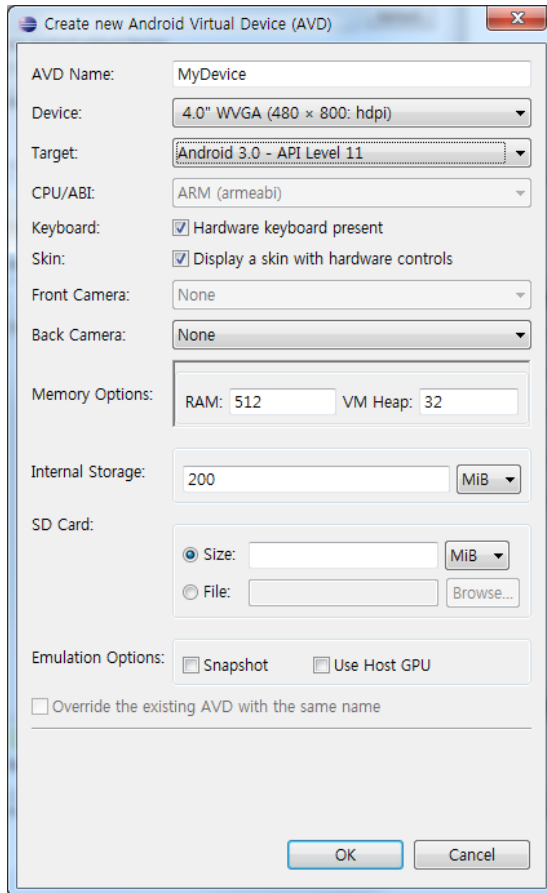
아래와 같은 대화창이 표시되며, Launch a new Android Virtual Device를 선택하고 Manager 버튼을 클릭합니다.



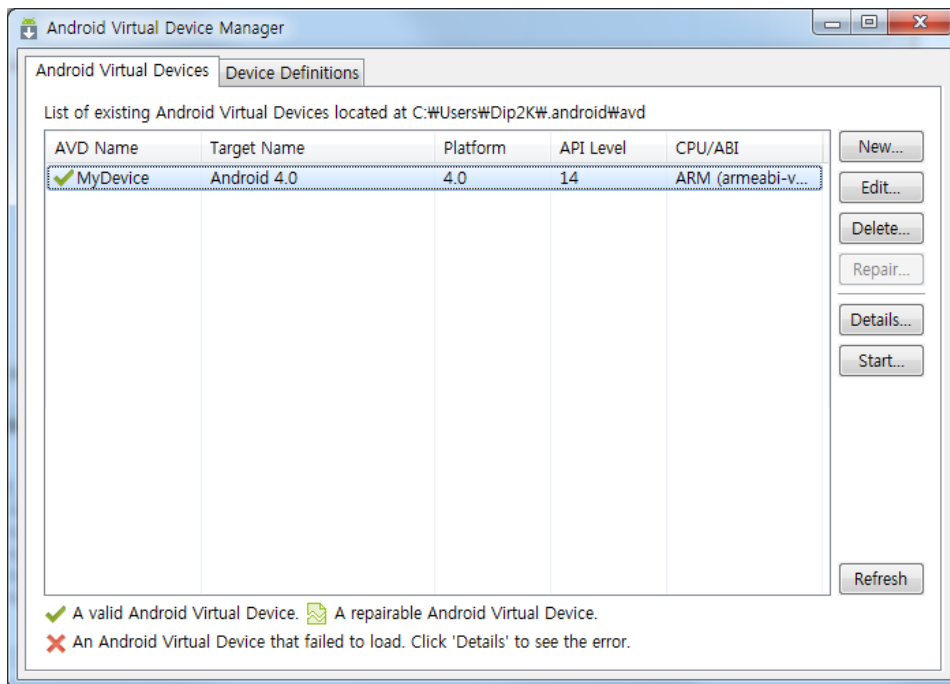
아래와 같은 대화창이 표시되며 New 버튼을 클릭합니다.



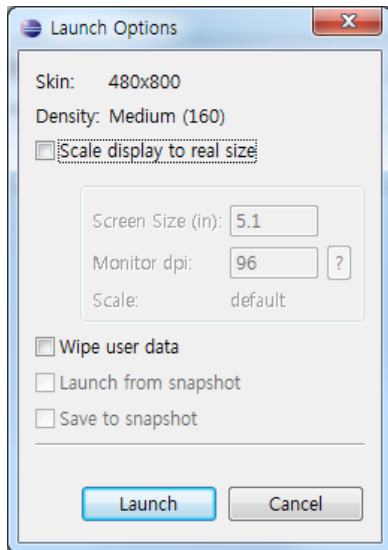
새롭게 표시되는 대화창에서 다음처럼 입력합니다.



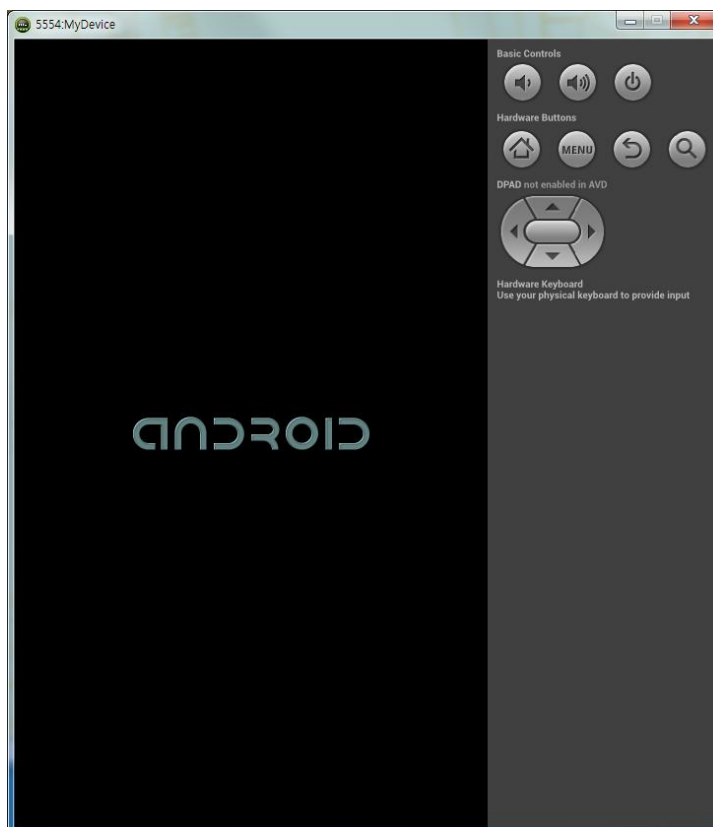
다음 처럼 MyDevice라는 이름의 AVD가 추가된 것을 확인할 수 있습니다. MyDevice를 선택하고 Start 버튼을 클릭합니다.



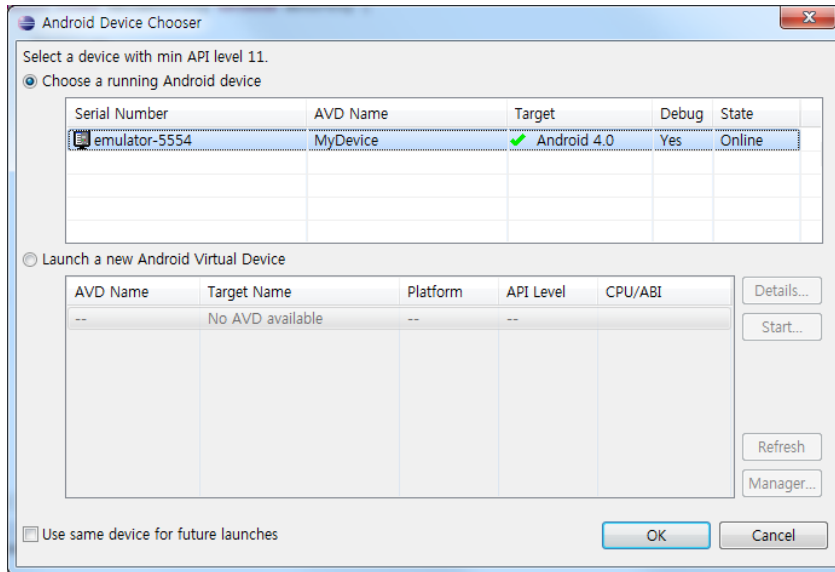
다음처럼 새로운 대화창이 표시되며 Launch 버튼을 클릭합니다.



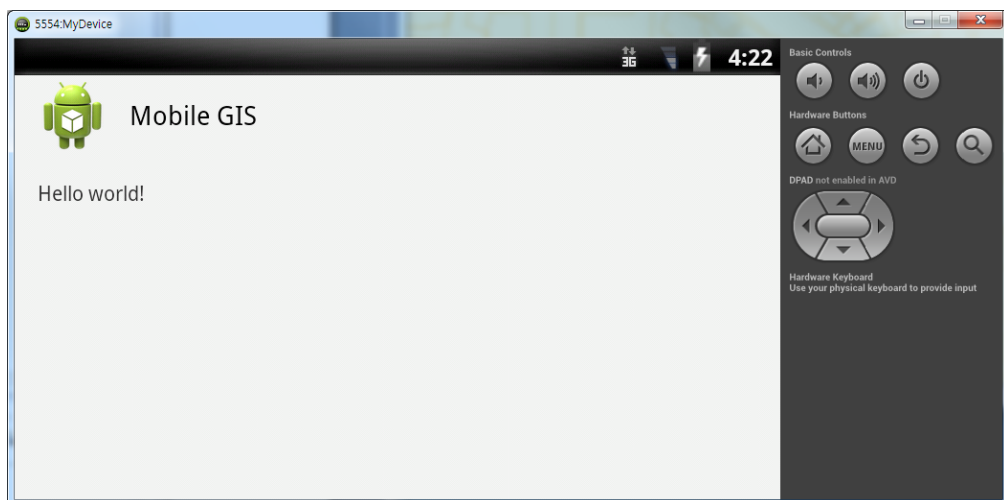
다음처럼 AVD가 실행되는 것을 볼 수 있습니다.



앞서 F11 키를 눌러 표시된 Android Device Chooser 대화창으로 돌아왔으며 아래처럼 MyDevice를 선택하고 OK 버튼을 클릭합니다.



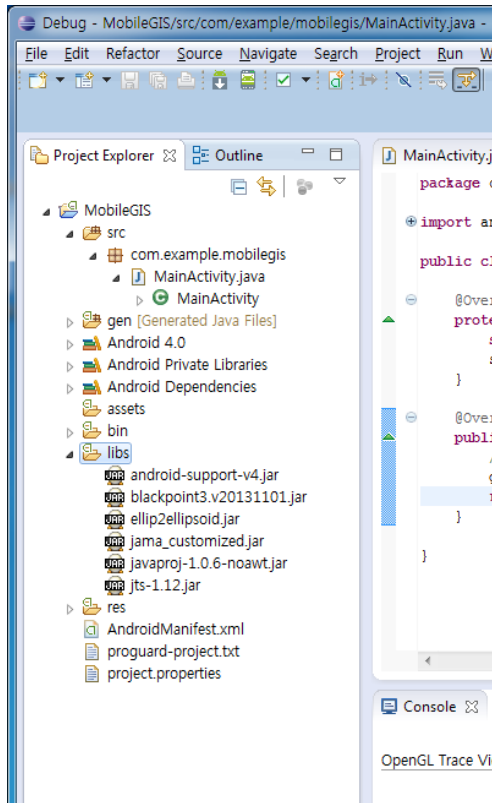
실행 결과는 다음과 같습니다.



(위의 화면은 AVD에서 Ctrl+F11를 눌러 화면을 회전시켰습니다)

D. 모바일 GIS 엔진 블랙포인트 라이브러리 참조

이제 모바일 GIS 엔진을 사용하기 위해 프로젝트의 libs 폴더에 블랙포인트의 lib 파일을 모두 복사합니다.



그리고 AndroidManifest.xml 파일에 다음과 같이 6개의 퍼미션(Permission)을 추가합니다.

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

E. 화면에 지도뷰 배치

activity_main.xml 파일을 열어 다음처럼 변경합니다.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <geoservice.blackpoint.XrMap
        android:id="@+id/map"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />

</RelativeLayout>
```

MainActivity.java 파일을 다음처럼 변경합니다.

```

package com.example.mobilegis;

import geoservice.blackpoint.XrMap;
import android.app.Activity;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.Menu;
import android.view.Window;

public class MainActivity extends Activity {

    private XrMap map = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        requestWindowFeature(Window.FEATURE_NO_TITLE);

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

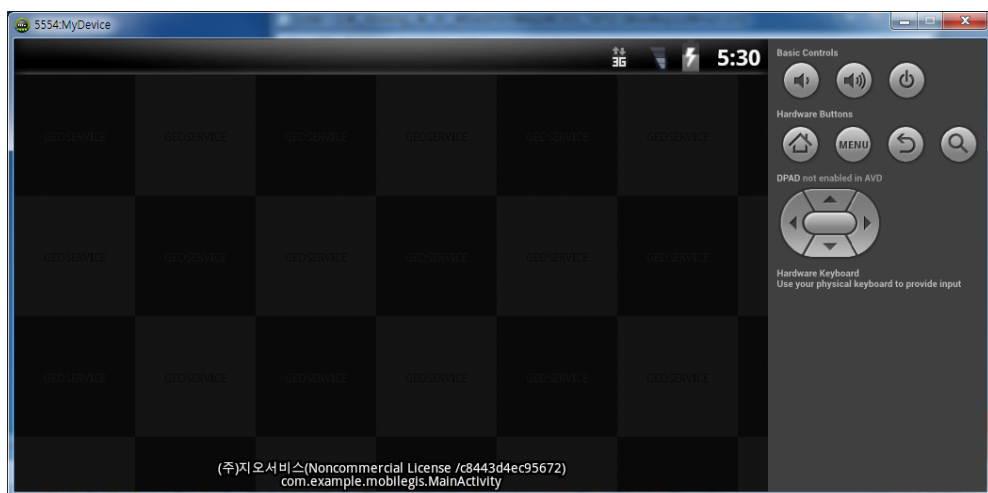
        map = (XrMap) findViewById(R.id.map);

        DisplayMetrics outMetrics = new DisplayMetrics();
        getWindowManager().getDefaultDisplay().getMetrics(outMetrics);
        int DPI = outMetrics.densityDpi;
        map.setDPI(DPI);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}

```

이제 F11를 눌러 실행하면 다음과 같은 화면이 표시됩니다.

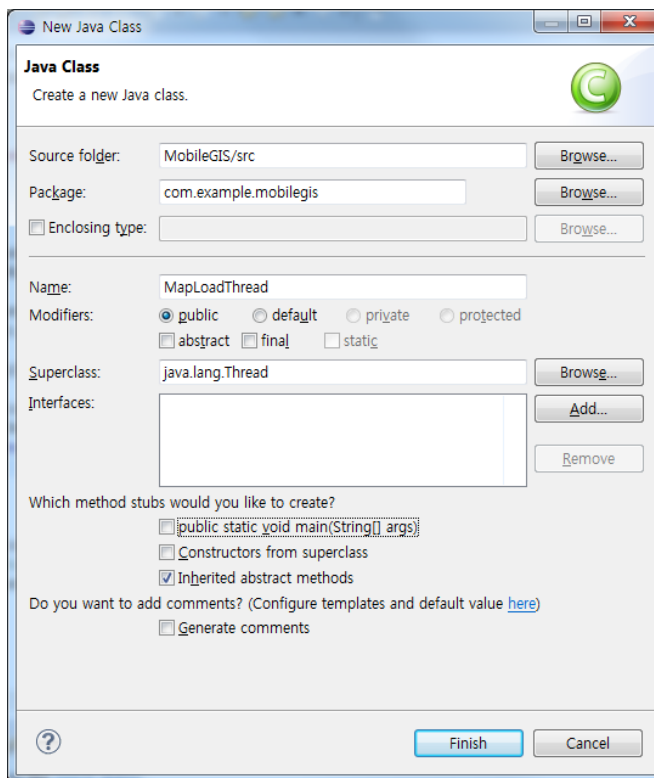


F. 레이어 구성

모바일 GIS 엔진인 블랙포인트는 지도 데이터에 대해서 ESRI 사의 SHP 파일과 모바일 환경에 최적화된 XrS 파일을 지원함은 물론 공간 서버로부터 공간 데이터를 받아

사용할 수 있습니다. 이 교육에서는 공간 서버로부터 공간 데이터를 받아 사용하는 것을 위주로 사용하겠습니다.

지도 데이터를 추가하는 작업은 연산 시간이 많이 소요될 수 있으므로 별도의 스레드(Thread)를 사용합니다. 아래처럼 Thread로부터 상속을 받는 MapLoadThread.java 파일을 추가합니다.



그리고 MapLoadThread.java 파일을 다음처럼 입력합니다.

```
package com.example.mobilegis;

import geoservice.blackpoint.MouseMode;
import geoservice.blackpoint.XrMap;
import geoservice.blackpoint.base.CoordMapper;
import geoservice.blackpoint.managers.LayerManager;
import geoservice.blackpoint.view.layers.ShapeLayer;
import android.os.Handler;
import android.os.Message;

public class MapLoadThread extends Thread {

    private XrMap map = null;

    public MapLoadThread(XrMap map) {
        this.map = map;
    }

    public void run() {
        map.getRenderManager().waitForDrawing(true);
    }
}
```

```

map.setMouseMode(MouseMode.NoneMode);
LayerManager layerMan = map.getLayerManager();

ShapeLayer lyrEMD = new ShapeLayer("읍면동",
    "http://www.geoservice.co.kr:8080/Xr?layerName=SJ_EMD");
layerMan.addLayer(lyrEMD);

map.setMouseMode(MouseMode.MapViewMode);

handler.sendMessage(0);
}

private Handler handler = new Handler() {
    public void handleMessage(Message msg) {
        if(msg.what == 0) {
            CoordMapper coordMapper = map.getCoordMapper();
            LayerManager layerMan = map.getLayerManager();
            coordMapper.zoomByMBR(layerMan.getLayerByName("읍면동").getMBR());
        }
        map.update();
    }
};
}

```

이제 MapLoadThread에서 제공하는 스레드를 실행하기 위해 MainActivity.java 파일에서 제공하는 MainActivity 클래스가 다음처럼 OnMapReadyEventListener 인터페이스를 구현하도록 지정합니다.

```
public class MainActivity extends Activity implements OnMapReadyEventListener {
```

그리고 onCreate의 마지막 줄에 다음 코드를 추가합니다.

```
map.setOnMapReadyListener(this);
```

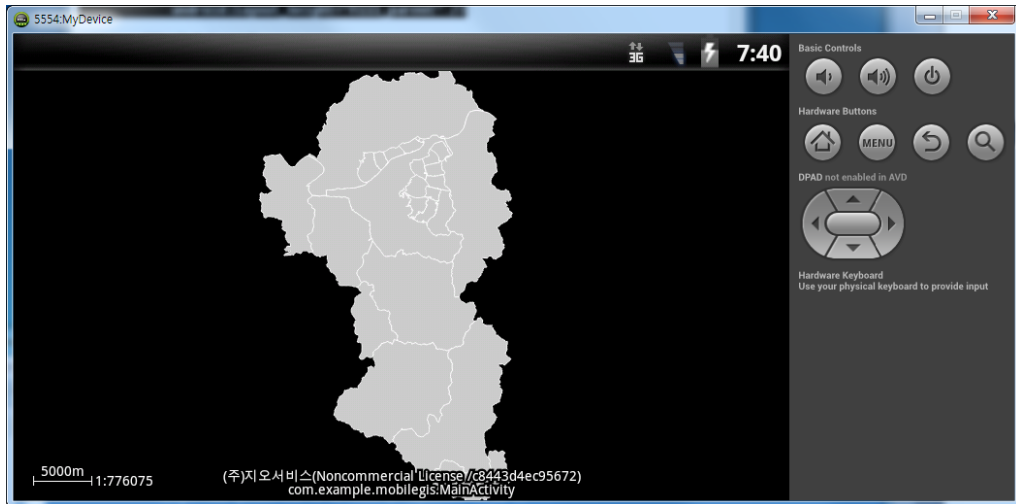
OnMapReadyEventListener 인터페이스에 대해서 onMapReady라는 함수를 구현해야 하며 다음처럼 입력합니다.

```

@Override
public void onMapReady(MapEvent arg0) {
    MapLoadThread dataLoadThread = new MapLoadThread( map);
    dataLoadThread.start();
}

```

실행하면 다음과 같은 결과를 볼 수 있습니다.



지도에 대해서 확대, 축소를 위한 메뉴를 추가합니다. 이를 위해서 main.xml 파일을 열어 다음처럼 입력합니다.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
  <item
    android:id="@+id/zoomIn"
    android:showAsAction="never"
    android:title="확대"
  />

  <item
    android:id="@+id/zoomOut"
    android:showAsAction="never"
    android:title="축소"
  />

  <item
    android:id="@+id/identifyAttribute"
    android:showAsAction="never"
    android:title="건물속성"
  />

  <item
    android:id="@+id/moveGPS"
    android:showAsAction="never"
    android:title="내 위치"
  />
</menu>
```

이제 이 메뉴에 대한 코드를 위해 MainActivity.java에서 onOptionsItemSelected 함수를 다음처럼 추가합니다.

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
  switch(item.getItemId()) {
    case R.id.zoomIn: {
      double scale = map.getCoordMapper().getScale();
      map.getCoordMapper().setScale(scale * 2);
      map.update();
    }
  }
}
```



```

        break;
    }

    case R.id.zoomOut: {
        double scale = map.getCoordMapper().getScale();
        map.getCoordMapper().setScale(scale * 0.5);
        map.update();

        break;
    }
}

return true;
}

```

실행하여 MENU를 통해 지도를 확대하고 축소할 수 있습니다.

이제 제주도의 "도"에 대한 행정구역 뿐만 아니라 시, 읍면동, 리, 도로, 건물에 대한 레이어를 구성하기 위해 MapLoadThread의 run 함수를 다음처럼 변경합니다. 이와 함께 축척에 따라 각 레이어를 보이고 감추는 기능을 추가합니다.

레이어 이름	축척범위
읍면동	1:2000000 — 1:200000
리	1:200000 — 1:1
도로_중심선	1:100000 — 1:1
도로_실폭	1:50000 — 1:1
건물	1:25000 — 1:1

```

public void run() {
    map.getRendererManager().waitForDrawing(true);

    map.setMouseMode(MouseMode.NoneMode);
    LayerManager layerMan = map.getLayerManager();

    ShapeLayer lyrEMD = new ShapeLayer("읍면동",
        "http://www.geoservice.co.kr:8080/Xr?layerName=SJ_EMD");
    layerMan.addLayer(lyrEMD);

    ShapeLayer lyrLI = new ShapeLayer("리",
        "http://www.geoservice.co.kr:8080/Xr?layerName=SJ_LI");
    layerMan.addLayer(lyrLI);

    ShapeLayer lyrMANAGE = new ShapeLayer("도로_중심선",
        "http://www.geoservice.co.kr:8080/Xr?layerName=SJ_MANAGE");
    layerMan.addLayer(lyrMANAGE);

    ShapeLayer lyrRW = new ShapeLayer("도로_실폭",
        "http://www.geoservice.co.kr:8080/Xr?layerName=SJ_RW");
    layerMan.addLayer(lyrRW);

    ShapeLayer lyrBULD = new ShapeLayer("건물",
        "http://www.geoservice.co.kr:8080/Xr?layerName=SJ_BULD");
    layerMan.addLayer(lyrBULD);
}

```

```

Visibility visbEMD = lyrEMD.getVisibility();
visbEMD.setVisibleByScale(true);
visbEMD.setFromScale(1);
visbEMD.setToScale(2000000);

Visibility visbLI = lyrLI.getVisibility();
visbLI.setVisibleByScale(true);
visbLI.setFromScale(1);
visbLI.setToScale(200000);

Visibility visbMANAGE = lyrMANAGE.getVisibility();
visbMANAGE.setVisibleByScale(true);
visbMANAGE.setFromScale(1);
visbMANAGE.setToScale(100000);

Visibility visbRW = lyrRW.getVisibility();
visbRW.setVisibleByScale(true);
visbRW.setFromScale(1);
visbRW.setToScale(50000);

Visibility visbBULD = lyrBULD.getVisibility();
visbBULD.setVisibleByScale(true);
visbBULD.setFromScale(1);
visbBULD.setToScale(25000);

map.setMouseMode(MouseMode.MapViewMode);

handler.sendEmptyMessage(0);
}

```

G. 레이어의 그리기 심벌 지정

이제 레이어의 그리기 심벌을 지정해 합니다. MapLoadThread의 run 함수에 다음 코드를 추가합니다.

```

SimpleDrawShapeTheme themeEMD = (SimpleDrawShapeTheme)lyrEMD.getTheme();
themeEMD.getStrokeSymbol().setWidth(3);

SimpleDrawShapeTheme themeLI= (SimpleDrawShapeTheme)lyrLI.getTheme();
themeLI.getStrokeSymbol().setWidth(1);
themeLI.getFillSymbol().setHollow(true);

SimpleDrawShapeTheme themeMANAGE = (SimpleDrawShapeTheme)lyrMANAGE.getTheme();
themeMANAGE.getStrokeSymbol().setColor(Color.DKGRAY);

SimpleDrawShapeTheme themeRW = (SimpleDrawShapeTheme)lyrRW.getTheme();
themeRW.getStrokeSymbol().setColor(Color.DKGRAY);
themeRW.getFillSymbol().setColor(Color.DKGRAY);

SimpleDrawShapeTheme themeBULD = (SimpleDrawShapeTheme)lyrBULD.getTheme();
themeBULD.getStrokeSymbol().setColor(Color.RED);
themeBULD.getFillSymbol().setColor(Color.YELLOW);

```

실행하여 지도를 확대해 보면 다음과 같습니다.

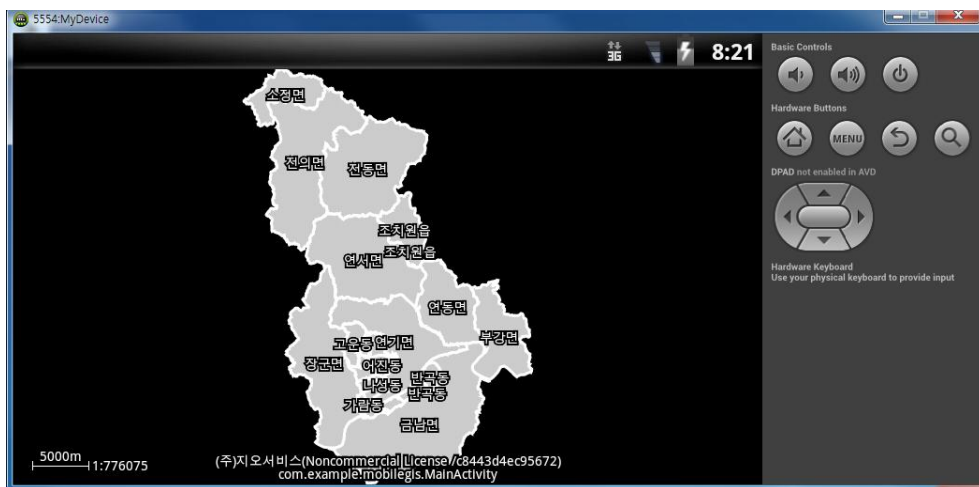


H. 레이어의 라벨 지정

레이어의 라벨을 표시하도록 합니다. 먼저 읍면동에 대한 행정구역 명칭을 표시하기 위해 MapLoadThread의 run 함수에 다음 코드를 추가합니다.

```
ShapeLayerLabel lblEMD = (ShapeLayerLabel)lyrEMD.getLabel();
lblEMD.setFieldName("EMD_KOR_NM");
lblEMD.setEnabled(true);
lblEMD.getFontSymbol().setTextSize(14);
lblEMD.getFontSymbol().setTextColor(Color.WHITE);
Visibility visLblEMD = lblEMD.getVisibility();
visLblEMD.setVisibleByScale(true);
visLblEMD.setFromScale(1);
visLblEMD.setToScale(2000000);
```

실행해 보면 다음과 같은 화면을 볼 수 있습니다.



이제 나머지 레이어인 '리' 레이어와 도로 중심선에 대한 라벨도 지정합니다. 아래처럼 코드를 추가합니다.

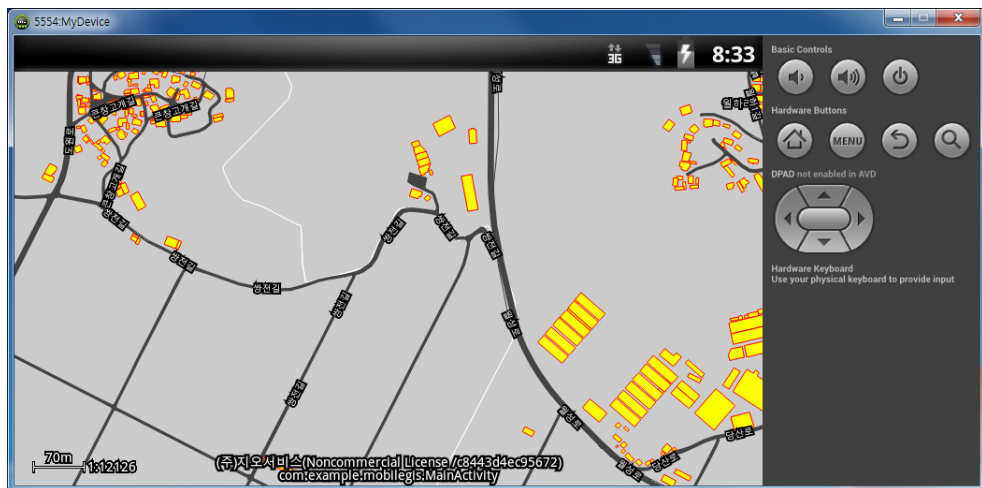
```

ShapeLayerLabel lblLI= (ShapeLayerLabel)lyrLI.getLabel();
lblLI.setFieldName("LI_KOR_NM");
lblLI.setEnabled(true);
lblLI.getFontSymbol().setTextSize(11);
lblLI.getFontSymbol().setTextColor(Color.LTGRAY);
Visibility visbLblLI= lblLI.getVisibility();
visbLblLI.setVisibleByScale(true);
visbLblLI.setFromScale(1);
visbLblLI.setToScale(200000);

ShapeLayerLabel lblMANAGE = (ShapeLayerLabel)lyrMANAGE.getLabel();
lblMANAGE.setFieldName("RN");
lblMANAGE.setEnabled(true);
lblMANAGE.getFontSymbol().setTextSize(10);
lblMANAGE.getFontSymbol().setTextColor(Color.WHITE);
Visibility visbLblMANAGE = lblMANAGE.getVisibility();
visbLblMANAGE.setVisibleByScale(true);
visbLblMANAGE.setFromScale(1);
visbLblMANAGE.setToScale(20000);

```

실행하고 지도를 확대해 보면 다음과 같은 화면을 볼 수 있습니다.



I. 건물을 터치하여 속성 확인

이제 화면에 표시된 건물을 선택하면 선택된 건물의 속성 정보를 제공하는 기능을 추가하겠습니다. 먼저 건물의 속성 정보가 필요하므로 MapLoadThread의 run 함수에서 건물 레이어를 추가하는 코드인 layerMan.addLayer(lyrBULD); 바로 밑에 다음 코드를 추가합니다.

```
((ShapeServiceAccess)lyrBULD.getAccess()).bAlwaysRequestAttribute = true;
```

그리고 선택된 건물에 대해서 시각적으로 다르게 표시하기 위해 그래픽 레이어를 하나 추가합니다. MapLoadThread의 run 함수에서 map.setMouseMode(MouseMode.MapViewMode); 코드 바로 위에 다음 코드를 추가합니다.

```

GraphicLayer grpLyr = new GraphicLayer("gl");
layerMan.addLayer(grpLyr);
grpLyr.setTopLayer(true);

```

그리고 지도에서 화면 터치에 대해 이벤트를 추가하기 위해 MainActivity 클래스에 대해서 OnTapUpEventListener 인터페이스를 구현하도록 다음처럼 지정합니다.

```

public class MainActivity extends Activity
    implements OnMapReadyEventListener, OnTapUpEventListener {

```

그리고 onCreate에 다음 코드를 추가합니다.

```

map.setOnTapUpListener(this);

```

OnTapUpEventListener 인터페이스에 대해 구현해야 할 함수를 추가합니다.

```

@Override
public void onTapUp(MotionEvent arg0) {
    // TODO Auto-generated method stub
}

```

건물에 대해서 속성 정보를 확인하겠다는 명령은 메뉴 중에 건물속성을 선택하는 것으로 시작합니다. onOptionsItemSelected 함수에서 switch 문에 다음 case 문을 추가합니다.

```

case R.id.identifyAttribute: {
    if(MouseMode.PickingMode != map.getMouseMode()) {
        map.setMouseMode(MouseMode.PickingMode);
        Toast.makeText(this, "정보를 확인할 건물을 선택하세요.", Toast.LENGTH_LONG).show();
    } else {
        map.setMouseMode(MouseMode.MapViewMode);
    }
}
}

```

onTapUp 이벤트는 map의 MouseMode 값이 MouseMode.PickingMode 일때만 발생합니다. onTapUp 이벤트 함수를 다음처럼 입력합니다.

```

@Override
public void onTapUp(MotionEvent arg0) {
    boolean bDrawing = map.getRenderManager().isWorking();
    map.getRenderManager().waitForDrawing(true);

    ShapeLayer layer = (ShapeLayer)map.layers().getLayerByName("건물");
    ShapeAccess sa = (ShapeAccess)layer.getAccess();

    PointD coord = map.getCoordMapper().V2W(new PointD(arg0.getX(), arg0.getY()));

    try {
        Vector<Integer> fids = layer.getFIDsByMapCoord(coord);
        if(fids != null) {
            int cntFIDs = fids.size();

```

```

    if(cntFIDs > 0) {
        sa.beginLoading();

        int FID = fids.get(0);
        AttributeRow ar = sa.loadAttributeById(FID);
        AttributeRowSet ars = (AttributeRowSet)sa.getAttributeRowSet();

        ShapeRow sr = (ShapeRow)sa.loadById(FID);
        highlightSelected(sr);

        if(bDrawing) map.update();
        else map.refresh();

        map.getRendererManager().waitForDrawing(false);
        map.finishDrawing();

        String strInfo = getAttributeInfo(ars, ar);

        new AlertDialog.Builder(MainActivity.this)
            .setTitle("건물 정보")
            .setMessage(strInfo)
            .setPositiveButton("확인", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {}
            }).show();

        map.setMouseMode(MouseMode.MapViewMode);

        sa.unload(sr);
        sa.unload(ar);

        sa.endLoading();

        return;
    }
} catch (IOException e) {
    e.printStackTrace();
}
}

```

위의 코드는 2개의 사용자 정의 함수를 사용하고 있습니다. highlightSelected 함수는 사용자가 선택한 건물을 다른 색상으로 표시하여 사용자에게 피드백(Feedback)을 제공하고 getAttributeInfo 함수는 선택된 건물의 속성 정보를 문자열로 구성하여 전달하는 함수입니다.

highlightSelected 함수는 다음과 같이 입력합니다.

```

private void highlightSelected(ShapeRow sr) {
    GraphicLayer gl = (GraphicLayer)map.layers().getLayerByName("gl");
    gl.getAccess().getSpatialRowSet().reset();

    PolygonShape polygon = (PolygonShape)sr.getShape();
    ArrayList<ArrayList<PointD>> polygons = polygon.getPolygons();

    PolygonGraphic grp = new PolygonGraphic(polygons);
    grp.getStrokeSymbol().setColor(Color.CYAN);
    grp.getStrokeSymbol().setWidth(4);
    grp.getStrokeSymbol().setAntiAlias(true);
    grp.setFillSymbol().setColor(Color.CYAN);
}

```

```

grp.setFillSymbol().setAlpha(60);
GraphicRow gr = new GraphicRow(0, grp);

gl.addGraphic(gr);
}

```

다음과 같은 `getAttributeInfo` 함수를 입력합니다.

```

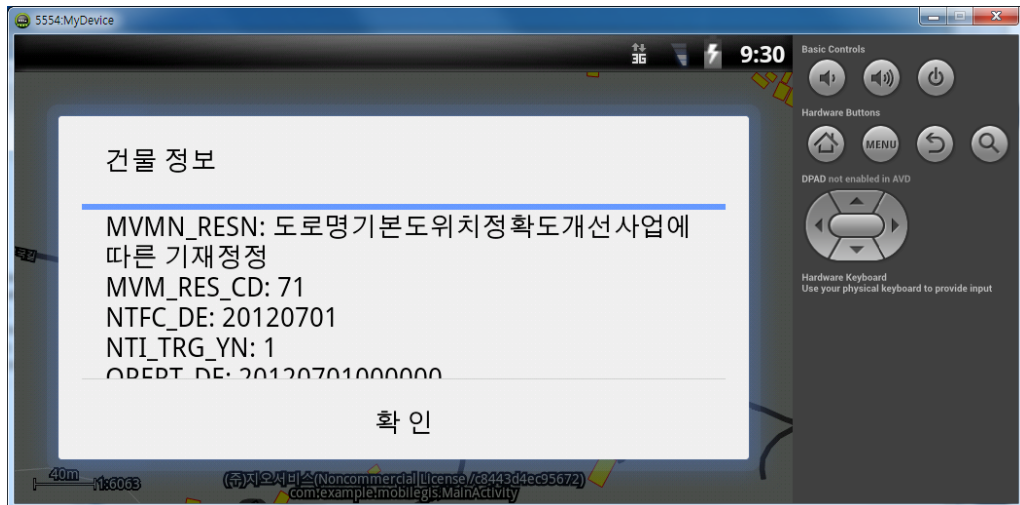
private String getAttributeInfo(AttributeRowSet ars, AttributeRow ar) {
    StringBuilder result = new StringBuilder();

    FieldSet fs = ars.getFieldSet();
    int cntFields = fs.getFieldsCount();
    for(int iField=0; iField<cntFields; iField++) {
        Field field = fs.getField(iField);
        result.append(field.getFieldName());
        result.append(": ");
        String value = ar.getValueAsString(iField);
        result.append(value);
        result.append('\n');
    }

    return result.toString();
}

```

이제 실행하고 건물의 속성 정보를 확인해 보면 다음과 같은 결과를 볼 수 있습니다.

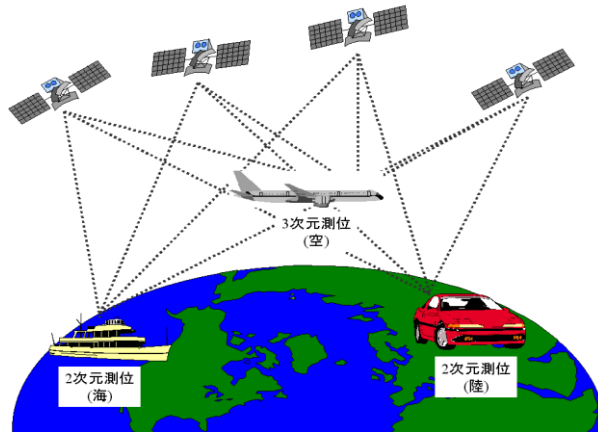


J. GPS로 현재 내 위치 확인

이제 GPS를 이용하여 현재 자신의 위치로 지도를 이동하는 기능을 추가하겠습니다. 이 GPS 연동 기능을 위해서 기본적인 GPS의 원리와 좌표계에 대해 알고 있어야 합니다.

i. GPS의 원리

안드로이드 기반의 디바이스는 GPS 수신기가 기본적으로 장착되어 있습니다.



GPS 수신기는 우주에 배치된 여러 대의 GPS 인공위성으로부터 신호를 받아 해석하고 계산하여 현재의 위치를 취득합니다.

GPS 수신기 1대를 가지고 위치를 얻는 것을 단독 측위라고 하며 오차는 수미터 — 수십미터 정도됩니다. 이러한 단독 측위로 수mm 또는 수cm의 정밀한 위치를 측정할 수 없으나 선박, 비행기, 자동차 등과 같은 네비게이션 분야에 매우 효과적으로 활용되고 있습니다.

ii. 좌표계의 원리

GPS의 좌표계는 WGS84 타원체의 경위도로서 3차원 좌표계입니다. 이를 종이 지도나 컴퓨터와 같은 화면에 나타내기 위해서 2차원으로 투영(Projection)하여야 하는데 본 교육에서 사용하는 세종시의 행정구역도 및 도로, 건물은 Bessel1841 타원체의 TM 좌표계이며 2차원 좌표계입니다. 즉, GPS에서로부터 측정된 3차원 좌표를 2차원 상에 표시하기 위해서는 좌표의 변환이 필요합니다.

블랙포인트는 기본적으로 GPS 좌표를 2차원의 화면에 표시하기 위한 좌표변환 기능을 제공하여 개발자가 쉽게 GPS 좌표와 지도를 연동할 수 있도록 돕고 있습니다.

이제 기본적인 GPS의 원리와 좌표계에 대해 파악했으므로 코드를 작성해 보도록 하겠습니다.

먼저 모바일 GIS 엔진에서 위치 기능을 위해 MainActivity 클래스의 onCreate에 다음 코드를 추가합니다.

```
IGPSProjection proj = new BesselUTMKProjection();
LocationManager lm = map.getLocationManager();
lm.setProjection(proj);
```



```
lm.start();
```

그리고 내 위치 MENU에 대한 실행 코드를 `onOptionsItemSelected` 함수에서 switch 문에 다음 case 문을 추가합니다.

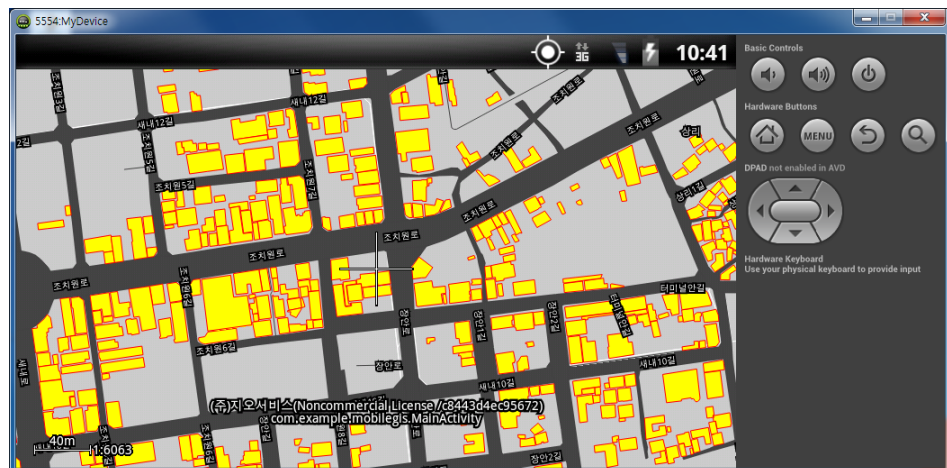
```
case R.id.moveGPS: {
    LocationManager lm = map.getLocationManager();
    Values3 pt = map.getLocationManager().getProjectedPoint();

    if(pt == null) {
        Toast.makeText(this, "GPS 위치 신호가 없습니다.", Toast.LENGTH_LONG).show();
    } else {
        lm.setTrackingEnable(!lm.isTrackingMode());

        if(lm.isTrackingMode()) {
            Toast.makeText(this, "내 위치를 추적합니다.", Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(this, "내 위치를 추적하지 않습니다.", Toast.LENGTH_LONG).show();
        }
    }
}

break;
}
```

이제 실행하고 내 위치 메뉴를 실행합니다. 그리고 AVD에서 GPS 좌표를 입력해 주기 위해서 DDMS의 Emulator Control에서 Longitude와 Latitude를 각각 127.298096과 36.601293을 입력해 줍니다. 결과는 다음과 같이 현재의 위치(에뮬레이터와 같은 AVD에서는 앞서 입력한 좌표 위치)로 이동하여 지도를 표시합니다.



블랙포인트의 보다 자세한 내용은 도서출판사 지앤서의 “모바일 GIS 프로그래밍 — 김형준, 이근상, 박진호 공저”을 참고하시기 바랍니다.