

```

1  package edu.seaaddicts.brockbutler.maps;
2
3  import android.app.Activity;
4  import android.app.AlertDialog;
5  import android.content.DialogInterface;
6  import android.content.Intent;
7  import android.os.Bundle;
8  import android.os.Handler;
9  import android.os.Message;
10 import android.util.Log;
11 import android.view.Menu;
12 import android.view.MenuItem;
13 import android.view.View;
14 import android.view.Window;
15 import android.widget.EditText;
16 import android.widget.LinearLayout;
17 import android.widget.RelativeLayout;
18 import android.widget.Toast;
19 import edu.seaaddicts.brockbutler.R;
20 import edu.seaaddicts.brockbutler.help.HelpActivity;
21
22 public class MapsActivity extends Activity {
23     private static final String TAG = "MapsActivity";
24
25     private EditText mSearchEditText;
26
27     private Handler mHandler;
28     private MapsTouchImageView mMapImage;
29     private MapsHandler mMapHandler;
30
31     private Position mStartPosition;
32     private Position mGoalPosition;
33     private Astar school;
34
35     @Override
36     protected void onCreate(Bundle savedInstanceState) {
37         super.onCreate(savedInstanceState);
38         getWindow().requestFeature(Window.FEATURE_ACTION_BAR_OVERLAY);
39         setContentView(R.layout.activity_map);
40
41         RelativeLayout layout = (RelativeLayout) findViewById(R.id.maps_layout);
42         layout.setSystemUiVisibility(View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN);
43         getActionBar().setBackgroundDrawable(
44             getResources().getDrawable(R.drawable.actionbar_bg));
45         init();
46
47         mHandler = new Handler() {
48
49             // Handles Messages sent from Thomas.
50             @Override
51             public void handleMessage(Message msg) {
52                 switch (msg.what) {
53                     case MapsHandler.MAPS_REQUEST_UPDATE:
54                         Log.d("MAIN HANDLER", "YAYAAA!!!");
55                         break;
56                     case MapsHandler.THREAD_UPDATE_POSITION:
57                         Log.d("TEST", msg.getData().getString("pos"));
58                         break;
59                     default:
60                         Log.d(TAG, "-----+++++ Got THREAD_UPDATE_POSITION message. +++++-----");
61                         break;
62                 }
63             }
64         };
65         mMapHandler = new MapsHandler(mHandler, this);
66         school = new Astar();
67     }
68
69     @Override
70     public boolean onCreateOptionsMenu(Menu menu) {
71         // Inflate the menu; this adds items to the action bar if it is present.

```

```

72     getMenuInflater().inflate(R.menu.activity_map, menu);
73     return true;
74 }
75
76 private void init() {
77     mMapImage = (MapsTouchImageView) findViewById(R.id.imgv_map);
78     // mMapImage.setOnClickListener(new OnClickListener() {
79     //
80     //     public void onClick(View v) {
81     //         mMapsHandler.sendMessage(MapsHandler.MAPS_REQUEST_UPDATE);
82     //     }
83     // });
84     // start = (Button) findViewById(R.id.btnstart);
85     // start.setOnClickListener(new OnClickListener() {
86     //
87     //     public void onClick(View v) {
88     //         mMapsHandler.sendMessage(MapsHandler.THREAD_REQUEST_START);
89     //     }
90     // });
91     // stop = (Button) findViewById(R.id.btnstop);
92     // stop.setOnClickListener(new OnClickListener() {
93     //
94     //     public void onClick(View v) {
95     //         mMapsHandler.sendMessage(MapsHandler.THREAD_REQUEST_PAUSE);
96     //     }
97     // });
98     //
99     // resume = (Button) findViewById(R.id.btnresume);
100    // resume.setOnClickListener(new OnClickListener() {
101    //
102    //     public void onClick(View v) {
103    //         mMapsHandler
104    //             .sendMessage(MapsHandler.THREAD_REQUEST_RESUME);
105    //     }
106    // });
107 }
108
109 @Override
110 public void onBackPressed() {
111     mMapsHandler.sendMessage(MapsHandler.THREAD_REQUEST_PAUSE);
112     mMapsHandler = null;
113     super.onBackPressed();
114 }
115
116 /**
117  *
118  * @param item
119  */
120 public void exitMaps(MenuItem item) {
121     onBackPressed();
122 }
123
124 /**
125  * Prompts user to search for existence of a location.
126  *
127  * @param item
128  */
129 public void displaySearchDialog(MenuItem item) {
130     AlertDialog.Builder editalert = new AlertDialog.Builder(this);
131
132     editalert.setTitle("MChown Location Search");
133     editalert.setMessage("Enter block or room name (i.e. B314)");
134
135     mSearchEditText = new EditText(this);
136     mSearchEditText.setSingleLine(true);
137     LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(
138         LinearLayout.LayoutParams.MATCH_PARENT,
139         LinearLayout.LayoutParams.MATCH_PARENT);
140     mSearchEditText.setLayoutParams(lp);
141     editalert.setView(mSearchEditText);
142

```

```

143     editalert.setPositiveButton("Search",
144         new DialogInterface.OnClickListener() {
145             public void onClick(DialogInterface dialog, int whichButton) {
146                 try {
147                     Position p1, p2=new Position(), p3=new Position(), p4=new Position(),
148                         p5=new Position();
149
150                     p1 = school.findPosition(mSearchEditText.getText().toString());
151
152                     p2.xPosition = p1.xPosition - 1;
153                     p2.yPosition = p1.yPosition - 1;
154
155                     p3.xPosition = p1.xPosition + 1;
156                     p3.yPosition = p1.yPosition + 1;
157
158                     p4.xPosition = p1.xPosition + 1;
159                     p4.yPosition = p1.yPosition - 1;
160
161                     p5.xPosition = p1.xPosition - 1;
162                     p5.yPosition = p1.yPosition + 1;
163
164                     Position[] pTest = {p5,p3,p4,p2,p5,p4,p3,p2};
165
166                     if(p1 != null && p2 != null)
167                         mMapImage.drawPosition(pTest,30);
168                 }
169                 catch (NullPointerException e) {
170                     Toast.makeText(getApplicationContext(), "Location not found",Toast.
171                         LENGTH_LONG).show();
172                 }
173             }
174         });
175     editalert.setNegativeButton("Cancel",
176         new DialogInterface.OnClickListener() {
177             public void onClick(DialogInterface dialog, int whichButton) {
178                 // do nothing
179             }
180         });
181     editalert.show();
182 }
183 /**
184  * Displays AlertDialog for user to enter destination. First the location is
185  * determined to exist, then if true path is drawn on map.
186  *
187  * @param item
188  */
189 public void displayGetDirectionsDialog(MenuItem item) {
190     school = new Astar();
191     AlertDialog.Builder alert = new AlertDialog.Builder(this);
192
193     alert.setTitle("Get Directions");
194     alert.setMessage("Enter Start and End Points");
195
196     LinearLayout lilal= new LinearLayout(this);
197     lilal.setOrientation(1); //1 is for vertical orientation
198     final EditText input = new EditText(this);
199     final EditText input1 = new EditText(this);
200     lilal.addView(input);
201     lilal.addView(input1);
202     alert.setView(lilal);
203
204     alert.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
205         public void onClick(DialogInterface dialog, int whichButton) {
206             try {
207                 mStartPosition = school.findPosition(input.getText().toString());
208                 mGoalPosition = school.findPosition(input1.getText().toString());
209
210                 Position[] p = school.pathGeneration(mStartPosition, mGoalPosition);
211                 if(mStartPosition != null && mGoalPosition != null)

```

```
212         mMapImage.drawPosition(p,8);
213     }
214     catch (NullPointerException e){
215         Toast.makeText(getApplicationContext(), "Location not found",Toast.
216             LENGTH_LONG).show();
217     }
218 }));
219
220 alert.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
221     public void onClick(DialogInterface dialog, int whichButton) {
222         // Canceled.
223     }
224 });
225
226 alert.show();
227 }
228
229 public void showHelp(MenuItem item)
230 {
231     Intent intent = new Intent(MapsActivity.this,HelpActivity.class);
232     Bundle bundle = new Bundle();
233     bundle.putString("activity", "maps");
234     intent.putExtras(bundle);
235     startActivity(intent);
236 }
237
238 /**
239  * Menu item onClick that calls mapping function to manual update user
240  * location in case first reported location isnot accurate.
241  * @param item
242  */
243 public void updatePosition(MenuItem item) {
244     Toast.makeText(getApplicationContext(), "Thomas' method goes here.",
245         Toast.LENGTH_LONG).show();
246     // call Thomas' method to update current position.
247 }
248 }
249
```