

```

1  /**
2   * TourActivity.java
3   * Brock Butler
4   * Main activity for the tour portion of Brock Butler.
5   * Created by Taras Mychaskiw 2013-02-20
6   * Copyright (c) 2013 Sea Addicts. All rights reserved.
7   */
8  package edu.seaaddicts.brockbutler.tour;
9
10 import android.app.Activity;
11 import android.os.Bundle;
12 import android.view.Menu;
13 import android.view.MenuItem;
14 import android.widget.ImageButton;
15 import android.widget.RelativeLayout;
16 import android.widget.Toast;
17 import edu.seaaddicts.brockbutler.R;
18
19 public class TourActivity extends Activity {
20     private TourNode[] nodes;
21     private final int numImages = R.drawable.j328 - R.drawable._a301b + 1;
22     private TourInfo info;
23
24     @Override
25     protected void onCreate(Bundle savedInstanceState) {
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.activity_tour);
28
29         RelativeLayout rl = (RelativeLayout)findViewById(R.id.screen);
30         ImageButton[] buttons = new ImageButton[5];
31         buttons[0] = (ImageButton)findViewById(R.id.outerleft);
32         buttons[1] = (ImageButton)findViewById(R.id.innerleft);
33         buttons[2] = (ImageButton)findViewById(R.id.center);
34         buttons[3] = (ImageButton)findViewById(R.id.innerright);
35         buttons[4] = (ImageButton)findViewById(R.id.outerright);
36         info = new TourInfo(rl, buttons, getApplicationContext());
37         initNodes();
38         nodes[idx(R.drawable._a301f)].paint(info);
39     }
40
41     @Override
42     public boolean onCreateOptionsMenu(Menu menu) {
43         // Inflate the menu; this adds items to the action bar if it is present.
44         getMenuInflater().inflate(R.menu.activity_tour, menu);
45         return true;
46     }
47
48     /**
49     * Pops the previous TourNode from the stack and goes to that node.
50     */
51     public void goBack(MenuItem item){
52         if (!info.history.empty())
53             info.history.pop().paint(info);
54     }
55
56     /**
57     * Goes to the node in the tour which is logically turning around.
58     */
59     public void turnAround(MenuItem item){
60         if (info.current.canTurnAround()){
61             info.history.push(info.current);
62             info.current.turnAroundNode.paint(info);
63         }
64         else
65             Toast.makeText(info.context, "No data available", Toast.LENGTH_SHORT).show();
66     }
67
68     /**
69     * Teleports the user to the specified block.
70     */
71     public void teleport(MenuItem item){

```

```

72     info.history.push(info.current);
73     switch (item.getItemId()){
74     case R.id.teleport_a_block: nodes[idx(R.drawable._a301f)].paint(info); return;
75     case R.id.teleport_b_block: nodes[idx(R.drawable._b301f)].paint(info); return;
76     case R.id.teleport_c_block: nodes[idx(R.drawable._c301f)].paint(info); return;
77     case R.id.teleport_d_block: nodes[idx(R.drawable._d301b)].paint(info); return;
78     case R.id.teleport_e_block: nodes[idx(R.drawable._e301f)].paint(info); return;
79     case R.id.teleport_f_block: nodes[idx(R.drawable._f301f)].paint(info); return;
80     case R.id.teleport_g_block: nodes[idx(R.drawable._h338f)].paint(info); return;
81     case R.id.teleport_h_block: nodes[idx(R.drawable._f301f)].paint(info); return;
82     case R.id.teleport_j_block: nodes[idx(R.drawable._j301f)].paint(info); return;
83     }
84 }
85
86 /**
87  * Ends the tour and destroys the Activity
88  */
89 public void endTour(MenuItem item){
90     super.onBackPressed();
91 }
92
93 /**
94  * Overrides the back button to pop previous TourNode from the stack
95  * and goes to that node.
96  */
97 @Override
98 public void onBackPressed(){
99     goBack(null);
100 }
101
102 /**
103  * @param r - image resource value
104  * @return index in `nodes` of TourNode which shows the image `r`
105  */
106 private int idx(int r){
107     return(r - R.drawable._a301b);
108 }
109
110 /**
111  * Adds a new TourRoom to nodes.
112  * @param r - the image resource value
113  * @param roomNumber - the title of the room, such as A323
114  */
115 private void room(int r, String roomNumber){
116     nodes[idx(r)] = new TourRoom(r,roomNumber);
117 }
118
119 /**
120  * Adds a new TourHall to nodes.
121  * @param r - the image resource value
122  * @param ol - image resource value of outer left button
123  * @param ul - resource of inner left
124  * @param c - resource of center
125  * @param ur - resource of inner right
126  * @param or - resource of outer right
127  * @param ta - resource of turn around node
128  */
129 private void hall(int r, int ll, int ul, int c, int ur, int lr, int ta){
130     if (ta != -1)
131         nodes[idx(r)] = new TourHall(r,
132             (ll == -1)? null : nodes[idx(ll)],
133             (ul == -1)? null : nodes[idx(ul)],
134             (c == -1) ? null : nodes[idx(c)],
135             (ur == -1)? null : nodes[idx(ur)],
136             (lr == -1)? null : nodes[idx(lr)],
137             /*(ta == -1)?*/nodes[idx(ta)]);
138     else
139         nodes[idx(r)] = new TourHall(r,
140             (ll == -1)? null : nodes[idx(ll)],
141             (ul == -1)? null : nodes[idx(ul)],
142             (c == -1) ? null : nodes[idx(c)],

```

```

143         (ur == -1)? null : nodes[idx(ur)],
144         (lr == -1)? null : nodes[idx(lr)]);
145     }
146     private void hall(int r, int c){ hall(r,-1,-1,c,-1,-1,-1); }
147     private void hall(int r, int c, int ta){ hall(r,-1,-1,c,-1,-1,ta); }
148     private void hall (int r, int ll, int ul, int c, int ur, int lr){ hall(r,ll,ul,c,ur
,lr,-1); }
149
150     /**
151      * Giant method to link all of the images together. The fun stuff. Right here.
152      */
153     private void initNodes(){
154         nodes = new TourNode[numImages];
155
156         /** A BLOCK ROOMS */
157         room(R.drawable.a323,"A323");
158
159         /** B BLOCK ROOMS */
160         room(R.drawable.b303a,"B303A");
161         room(R.drawable.b303b,"B303B");
162         hall(R.drawable.b303,-1,-1,R.drawable.b303b,R.drawable.b303a,-1);
163
164         /** C BLOCK ROOMS */
165         room(R.drawable.c300,"C300");
166         room(R.drawable.c303,"C303");
167         room(R.drawable.c304,"C304");
168         room(R.drawable.c306_1,"C306");
169         room(R.drawable.c306_2,"C306");
170         hall(R.drawable.c306_3,R.drawable.c306_2,-1,-1,-1,R.drawable.c306_1);
171         room(R.drawable.c308,"C308");
172         room(R.drawable.c310,"C310");
173
174         /** D BLOCK ROOMS */
175         room(R.drawable.d300,"D300");
176         room(R.drawable.d301,"D301");
177         room(R.drawable.d303,"D303");
178         room(R.drawable.d304,"D304");
179         room(R.drawable.d308,"D308");
180         room(R.drawable.d309_1,"D309");
181         room(R.drawable.d309_2,"D309");
182         room(R.drawable.d310,"D310");
183         room(R.drawable.d314,"D314");
184         room(R.drawable.d315,"D315");
185         room(R.drawable.d316,"D316");
186         room(R.drawable.d317,"D317");
187         room(R.drawable.d318,"D318");
188         room(R.drawable.d319,"D319");
189         room(R.drawable.d350l,"D350L");
190
191         /** E BLOCK ROOMS */
192         room(R.drawable.e302,"E302");
193         room(R.drawable.e303,"E303");
194         room(R.drawable.e304,"E304");
195
196         /** F BLOCK ROOMS */
197         /* N/A */
198         /** G BLOCK ROOMS */
199         room(R.drawable.g301_2,"G301");
200         room(R.drawable.g301_3,"G301");
201         hall(R.drawable.g301_1,R.drawable.g301_2,-1,-1,-1,R.drawable.g301_3);
202         room(R.drawable.g305,"G305");
203
204         /** H BLOCK ROOMS */
205         room(R.drawable.h300,"H300");
206         room(R.drawable.h302,"H302");
207         room(R.drawable.h303,"H303");
208         room(R.drawable.h304a,"H304A");
209         hall(R.drawable.h304,-1,-1,-1,-1,R.drawable.h304a);
210         hall(R.drawable.h306,/*R.drawable.h309a6*/-1,-1,-1,/*R.drawable.h306a6*/-1,-1);
211         hall(R.drawable.h309,-1,/*R.drawable.h306a9*/-1,-1,-1,/*R.drawable.h309a9*/-1);
212         hall(R.drawable.h306a6,-1,-1,-1,R.drawable.h309,-1);

```

```
213 hall(R.drawable.h306a9,-1,R.drawable.h306,-1,-1,-1,R.drawable.h306a6);
214 hall(R.drawable.h309a6,R.drawable.h309,-1,-1,-1,-1);
215 hall(R.drawable.h309a9,R.drawable.h306,-1,-1,-1,-1,R.drawable.h309a6);
216 nodes[idx(R.drawable.h306)].setOuterLeftNode(nodes[idx(R.drawable.h309a6)]);
217 nodes[idx(R.drawable.h306)].setInnerRightNode(nodes[idx(R.drawable.h306a6)]);
218 nodes[idx(R.drawable.h309)].setInnerLeftNode(nodes[idx(R.drawable.h309a9)]);
219 nodes[idx(R.drawable.h309)].setOuterRightNode(nodes[idx(R.drawable.h306a9)]);
220 room(R.drawable.h310,"H310");
221 room(R.drawable.h313,"H313");
222 room(R.drawable.h315,"H315");
223 room(R.drawable.h316,"H316");
224 room(R.drawable.h317,"H317");
225 room(R.drawable.h318a,"H318A");
226 hall(R.drawable.h318,-1,-1,-1,R.drawable.h318a,-1);
227 room(R.drawable.h320,"H320");
228 room(R.drawable.h321,"H321");
229 room(R.drawable.h322,"H322");
230 room(R.drawable.h323,"H323");
231 room(R.drawable.h324,"H324");
232
233 /** J BLOCK ROOMS */
234 room(R.drawable.j301,"J301");
235 room(R.drawable.j310,"J310");
236 room(R.drawable.j327,"J327");
237 room(R.drawable.j328,"J328");
238
239 /** J BLOCK FORWARD PASS */
240 hall(R.drawable._j315f,-1,-1,-1,-1,R.drawable.j301);
241 hall(R.drawable._j314f,R.drawable._j315f);
242 hall(R.drawable._j313f,R.drawable._j314f);
243 hall(R.drawable._j312f,-1,-1,-1,R.drawable._j313f,-1);
244 hall(R.drawable._j311f,R.drawable._j312f);
245 hall(R.drawable._j310f,R.drawable._j311f);
246 hall(R.drawable._j309f,R.drawable._j310f);
247 hall(R.drawable._j308f,R.drawable._j309f);
248 hall(R.drawable._j307f,-1);
249 hall(R.drawable._j306f,R.drawable._j307f,-1,-1,-1,R.drawable._j308f);
250 hall(R.drawable._j305f,R.drawable._j306f);
251 hall(R.drawable._j304f,-1,R.drawable.j327,R.drawable._j305f,R.drawable.j328,-1);
252 hall(R.drawable._j303f,R.drawable._j304f);
253 hall(R.drawable._j302f,-1,R.drawable.j310,-1,R.drawable.j301,-1);
254 hall(R.drawable._j301f,-1,-1,R.drawable._j303f,R.drawable._j302f,-1);
255
256 /** D BLOCK FORWARD PASS */
257 hall(R.drawable._d343f,/*R.drawable._d304b*/-1,-1,-1,/*R.drawable._j312b*/-1,-1);
258 hall(R.drawable._d342f,R.drawable._d343f);
259 hall(R.drawable._d341f,-1,R.drawable.g310,R.drawable._j301f,R.drawable._d342f,-1);
260 hall(R.drawable._d340f,-1,-1,R.drawable._d341f,-1,/*R.drawable._d307b*/-1);
261 hall(R.drawable._d339f,-1,-1,R.drawable._d340f,/*R.drawable._d307b*/-1,R.drawable
.d309_2);
262 hall(R.drawable._d338f,R.drawable._d339f);
263 hall(R.drawable._d337f,-1,-1,R.drawable._d338f,R.drawable.d309_1,-1);
264 hall(R.drawable._d336f,R.drawable._d337f);
265 hall(R.drawable._d335f,/*R.drawable._d306b*/-1,-1,-1,-1,R.drawable._d341f);
266 hall(R.drawable._d334f,-1,-1,R.drawable._d335f,-1,R.drawable.d310);
267 hall(R.drawable._d333f,-1,-1,R.drawable._d334f,-1,R.drawable.d314);
268 hall(R.drawable._d332f,R.drawable._d333f);
269 hall(R.drawable._d331f,-1,-1,R.drawable._d332f,-1,R.drawable.d315);
270 hall(R.drawable._d330f,-1,-1,R.drawable._d331f,-1,R.drawable.d316);
271 hall(R.drawable._d329f,-1,-1,R.drawable._d330f,-1,R.drawable.d317);
272 hall(R.drawable._d328f,-1,R.drawable._d329f,R.drawable.d318,-1,-1);
273 hall(R.drawable._d327f,R.drawable._d328f);
274 hall(R.drawable._d326f,R.drawable.d319,-1,R.drawable._d327f,-1,-1);
275 hall(R.drawable._d325f,R.drawable._d326f);
276 hall(R.drawable._d324f,R.drawable._d325f);
277 hall(R.drawable._d323f,R.drawable._d324f);
278 hall(R.drawable._d322f,R.drawable._d323f);
279 hall(R.drawable._d321f,R.drawable._d322f);
280 hall(R.drawable._d320f,R.drawable._d321f);
281 hall(R.drawable._d319f,R.drawable._d320f);
282 hall(R.drawable._d318f,-1,-1,R.drawable._d319f,/*R.drawable._d326b*/-1,-1);
```

```
283 hall(R.drawable._d317f,-1,R.drawable._d336f,R.drawable.d308,R.drawable._d318f,-1);
284 hall(R.drawable._d316f,R.drawable.d304,-1,R.drawable._d317f,-1,-1);
285 hall(R.drawable._d315f,R.drawable.d303,-1,R.drawable._d316f,-1,-1);
286 hall(R.drawable._d314f,R.drawable._d315f);
287 hall(R.drawable._d313f,R.drawable.d301,-1,R.drawable._d314f,-1,-1);
288 hall(R.drawable._d312f,/*R.drawable._d319b*/-1,-1,-1,-1,R.drawable._d319f);
289 hall(R.drawable._d311f,R.drawable._d312f);
290 hall(R.drawable._d310f,-1,R.drawable._d311f,-1,-1,-1);
291 hall(R.drawable._d309f,R.drawable._d310f);
292 hall(R.drawable._d308f,R.drawable._d309f);
293 hall(R.drawable._d307f,R.drawable._d308f);
294 hall(R.drawable._d306f,R.drawable._d307f);
295 hall(R.drawable._d305f,R.drawable._d306f);
296 hall(R.drawable._d304f,-1,R.drawable.d3501,-1,-1,-1);
297 hall(R.drawable._d303f,-1,R.drawable._d305f,-1,R.drawable._d304f,-1);
298 hall(R.drawable._d302f,R.drawable.d300,-1,R.drawable._d313f,-1,R.drawable._d303f);
299 hall(R.drawable._d301f,R.drawable._d302f);
300
301 /** C BLOCK FORWARD PASS */
302 hall(R.drawable._c312f,-1,-1,-1,R.drawable._d301f,-1);
303 hall(R.drawable._c311f,-1,R.drawable.c310,R.drawable._c312f,-1,-1);
304 hall(R.drawable._c310f,R.drawable._c311f);
305 hall(R.drawable._c309f,R.drawable.c306_3,-1,R.drawable._c310f,-1,-1);
306 hall(R.drawable._c308f,R.drawable._c309f);
307 hall(R.drawable._c307f,-1,R.drawable._c308f,-1,-1,-1);
308 hall(R.drawable._c306f,-1,-1,/*R.drawable._c306b*/-1,R.drawable._c307f,-1);
309 hall(R.drawable._c305f,R.drawable.c304,-1,R.drawable._c306f,-1,-1);
310 hall(R.drawable._c304f,R.drawable.c303,-1,R.drawable._c305f,-1,-1);
311 hall(R.drawable._c303f,R.drawable._c304f);
312 hall(R.drawable._c302f,R.drawable._c303f);
313 hall(R.drawable._c301f,-1,-1,-1,R.drawable._c302f,-1);
314 hall(R.drawable._c316f,-1,R.drawable._c306f,/*R.drawable._c310b*/-1,-1,-1);
315 hall(R.drawable._c315f,R.drawable._c316f);
316 hall(R.drawable._c314f,R.drawable.c308,-1,R.drawable._c315f,-1,-1);
317 hall(R.drawable._c313f,-1,-1,-1,R.drawable._c314f,-1);
318
319 /** F BLOCK FORWARD PASS */
320 hall(R.drawable._f308f,R.drawable._c313f);
321 hall(R.drawable._f307f,R.drawable._f308f);
322 hall(R.drawable._f306f,R.drawable._f307f);
323 hall(R.drawable._f305f,R.drawable._f306f);
324 hall(R.drawable._f304f,R.drawable._f305f);
325 hall(R.drawable._f303f,R.drawable._f304f);
326 hall(R.drawable._f302f,R.drawable._f303f);
327 hall(R.drawable._f301f,-1,R.drawable._f302f,-1,-1,/*R.drawable._h301b*/-1);
328
329 /** G BLOCK FORWARD PASS */
330 hall(R.drawable._g313f,-1);
331 hall(R.drawable._g312f,R.drawable._g313f);
332 hall(R.drawable._g311f,R.drawable.g305,-1,R.drawable._g312f,-1,-1);
333 hall(R.drawable._g310f,R.drawable._g311f);
334 hall(R.drawable._g309f,R.drawable._g310f);
335 hall(R.drawable._g308f,R.drawable._g309f);
336 hall(R.drawable._g307f,R.drawable._g308f);
337 hall(R.drawable._g306f,R.drawable._g307f);
338 hall(R.drawable._g305f,R.drawable.g301_1,-1,R.drawable._g306f,-1,-1);
339 hall(R.drawable._g304f,R.drawable._g305f);
340 hall(R.drawable._g303f,R.drawable._g304f);
341 hall(R.drawable._g302f,-1,-1,R.drawable._g303f,-1,-1);
342 hall(R.drawable._g301f,-1,R.drawable._g302f,-1,-1,-1,R.drawable._f301f);
343
344 /** H BLOCK FORWARD PASS */
345 hall(R.drawable._h338f,-1,R.drawable._g301f,-1,R.drawable._f302f,-1);
346 hall(R.drawable._h337f,R.drawable._h338f);
347 hall(R.drawable._h336f,R.drawable._h337f);
348 hall(R.drawable._h335f,R.drawable._h336f);
349 hall(R.drawable._h334f,R.drawable._h335f);
350 hall(R.drawable._h333f,-1,/*R.drawable._h303b*/-1,R.drawable._h334f,-1,-1);
351 hall(R.drawable._h332f,R.drawable._h333f);
352 hall(R.drawable._h331f,R.drawable._h332f);
353 hall(R.drawable._h330f,R.drawable._h331f);
```

```
354 hall(R.drawable._h329f,-1,R.drawable._h334f,/*R.drawable._h303b*/-1,-1,
/*R.drawable._h322b*/-1);
355 hall(R.drawable._h328f,-1,R.drawable.h324,R.drawable._h329f,-1,-1);
356 hall(R.drawable._h327f,R.drawable.h323,-1,R.drawable._h328f,-1,-1);
357 hall(R.drawable._h326f,R.drawable._h327f);
358 hall(R.drawable._h325f,R.drawable._h326f);
359 hall(R.drawable._h324f,R.drawable.h320,-1,R.drawable._h325f,-1,-1);
360 hall(R.drawable._h323f,-1,-1,R.drawable._h324f,R.drawable.h322,R.drawable.h321);
361 hall(R.drawable._h322f,R.drawable.h318,-1,R.drawable._h323f,-1,-1);
362 hall(R.drawable._h321f,-1,R.drawable.h316,R.drawable._h322f,-1,R.drawable.h317);
363 hall(R.drawable._h320f,R.drawable._h321f);
364 hall(R.drawable._h319f,-1,-1,-1,-1,R.drawable._h320f);
365 hall(R.drawable._h318f,-1,R.drawable.h315,R.drawable._h319f,-1,-1);
366 hall(R.drawable._h317f,R.drawable.h313,-1,R.drawable._h318f,-1,-1);
367 hall(R.drawable._h316f,R.drawable._h317f);
368 hall(R.drawable._h315f,R.drawable._h316f);
369 hall(R.drawable._h314f,-1);
370 hall(R.drawable._h313f,R.drawable._h314f);
371 hall(R.drawable._h312f,-1,R.drawable.h310,R.drawable._h313f,R.drawable._h315f,-1);
372 hall(R.drawable._h311f,-1,-1,-1,R.drawable.h310,R.drawable._h312f);
373 hall(R.drawable._h310f,-1,-1,R.drawable._h311f,R.drawable.h309,-1);
374 hall(R.drawable._h309f,-1,-1,R.drawable._h310f,-1,R.drawable.h306);
375 hall(R.drawable._h308f,-1,-1,R.drawable._h309f,-1,R.drawable.h304);
376 hall(R.drawable._h307f,R.drawable.h303,-1,R.drawable._h308f,-1,-1);
377 hall(R.drawable._h306f,R.drawable._h307f);
378 hall(R.drawable._h305f,-1,-1,R.drawable._h306f,-1,R.drawable.h302);
379 hall(R.drawable._h304f,-1,R.drawable.h300,R.drawable._h305f,-1,-1);
380 hall(R.drawable._h303f,-1,R.drawable._h304f,-1,R.drawable._h330f,-1);
381 hall(R.drawable._h302f,R.drawable._h303f);
382 hall(R.drawable._h301f,R.drawable._h302f);
383
384 /** E BLOCK FORWARD PASS */
385 hall(R.drawable._e311f,-1);
386 hall(R.drawable._e310f,-1,R.drawable._h301f,-1,R.drawable._e311f,-1);
387 hall(R.drawable._e309f,R.drawable._e310f);
388 hall(R.drawable._e308f,R.drawable._e309f);
389 hall(R.drawable._e307f,-1,-1,R.drawable._e308f,-1,-1);
390 hall(R.drawable._e306f,R.drawable.e304,-1,R.drawable._e307f,-1,-1);
391 hall(R.drawable._e305f,R.drawable.e303,-1,R.drawable._e306f,-1,-1);
392 hall(R.drawable._e304f,R.drawable._e305f);
393 hall(R.drawable._e303f,-1,R.drawable.e302,R.drawable._e304f,-1,-1);
394 hall(R.drawable._e302f,R.drawable._e303f);
395 hall(R.drawable._e301f,-1,-1,-1,R.drawable._e302f,-1);
396
397 /** B BLOCK FORWARD PASS */
398 hall(R.drawable._b314f,R.drawable._c301f);
399 hall(R.drawable._b313f,R.drawable._b314f);
400 hall(R.drawable._b312f,R.drawable._b314f);
401 hall(R.drawable._b311f,-1,-1,-1,R.drawable._b312f,-1);
402 hall(R.drawable._b310f,-1,R.drawable._b311f,R.drawable._b313f,-1,-1);
403 hall(R.drawable._b309f,R.drawable._b310f);
404 hall(R.drawable._b308f,-1,R.drawable._b309f,-1,-1,-1);
405 hall(R.drawable._b307f,R.drawable._e301f);
406 hall(R.drawable._b306f,R.drawable._b307f);
407 hall(R.drawable._b305f,R.drawable._b306f);
408 hall(R.drawable._b304f,-1,-1,R.drawable._b305f,R.drawable.b303,-1);
409 hall(R.drawable._b303f,R.drawable._b304f);
410 hall(R.drawable._b302f,-1,R.drawable._b303f,-1,R.drawable._b308f,-1);
411 hall(R.drawable._b301f,-1,-1,-1,R.drawable._b302f,-1);
412
413 /** A BLOCK FORWARD PASS */
414 hall(R.drawable._a309f,-1,R.drawable._b301f,-1,-1,-1);
415 hall(R.drawable._a308f,R.drawable._a309f);
416 hall(R.drawable._a307f,R.drawable._a308f);
417 hall(R.drawable._a306f,R.drawable._a307f);
418 hall(R.drawable._a305f,-1,-1,R.drawable._a306f,-1,R.drawable.a323);
419 hall(R.drawable._a304f,R.drawable._a305f);
420 hall(R.drawable._a303f,R.drawable._a304f);
421 hall(R.drawable._a302f,R.drawable._a303f);
422 hall(R.drawable._a301f,R.drawable._a302f);
423
```

```
424     /** A BLOCK BACK PASS */
425     hall(R.drawable._a307b,-1,R.drawable._a302f);
426     hall(R.drawable._a306b,R.drawable._a307b,R.drawable._a303f);
427     hall(R.drawable._a305b,R.drawable._a306b,R.drawable._a304f);
428     hall(R.drawable._a304b,-1,R.drawable.a323,R.drawable._a305b,-1,-1,R.drawable.
429         _a307f);
430     hall(R.drawable._a303b,R.drawable._a304b,R.drawable._a308f);
431     hall(R.drawable._a302b,-1,-1,-1,R.drawable._a303b,-1,R.drawable._a309f);
432     hall(R.drawable._a301b,R.drawable._a302b,R.drawable._b301f);
433
434     /** B BLOCK BACK PASS */
435     hall(R.drawable._b317b,R.drawable._a301b,R.drawable._b301f);
436     hall(R.drawable._b316b,-1,R.drawable._b317b,-1,R.drawable._b303f,-1,R.drawable.
437         _b302f);
438     hall(R.drawable._b313b,-1,-1,-1,R.drawable._b316b,-1,R.drawable._b309f);
439     hall(R.drawable._b312b,R.drawable._b313b,R.drawable._b313f);
440     hall(R.drawable._b311b,R.drawable._b313b,R.drawable._b311f);
441     hall(R.drawable._b310b,R.drawable._b311b,R.drawable._b312f);
442     hall(R.drawable._b309b,-1,-1,R.drawable._b312b,R.drawable._b310b,-1,R.drawable.
443         _b314f);
444     hall(R.drawable._b308b,-1,R.drawable._b309b,-1,-1,-1,R.drawable._b314f);
445     hall(R.drawable._b306b,-1,-1,-1,R.drawable._b308b,-1,R.drawable._c301f);
446     hall(R.drawable._b305b,-1,R.drawable._b308f,-1,R.drawable._b317b,-1,R.drawable.
447         _b303f);
448     hall(R.drawable._b304b,R.drawable._b305b,R.drawable._b304f);
449     hall(R.drawable._b303b,R.drawable.b303,-1,R.drawable._b304b,-1,-1,R.drawable.
450         _b305f);
451     hall(R.drawable._b302b,R.drawable._b303b,R.drawable._b306f);
452     hall(R.drawable._b301b,R.drawable._b302b,R.drawable._e301f);
453
454     /** E BLOCK BACK PASS */
455     hall(R.drawable._e309b,-1,-1,R.drawable._b301b,-1,R.drawable.e302,R.drawable.
456         _e303f);
457     hall(R.drawable._e308b,R.drawable._e309b,R.drawable._e304f);
458     hall(R.drawable._e307b,-1,-1,R.drawable._e308b,-1,R.drawable.e303,R.drawable.
459         _e305f);
460     hall(R.drawable._e306b,-1,-1,R.drawable._e307b,-1,R.drawable.e304,R.drawable.
461         _e306f);
462     hall(R.drawable._e305b,R.drawable._e306b,R.drawable._e307f);
463     hall(R.drawable._e304b,R.drawable._e305b,R.drawable._e308f);
464     hall(R.drawable._e303b,R.drawable._e304b,R.drawable._e309f);
465     hall(R.drawable._e302b,R.drawable._e303b,R.drawable._e311f);
466     hall(R.drawable._e301b,R.drawable._e311f,-1,R.drawable._e303b,-1,-1,R.drawable.
467         _h301f);
468
469     /** H BLOCK BACK PASS */
470     hall(R.drawable._h326b,R.drawable._e301b,R.drawable._h302f);
471     hall(R.drawable._h325b,-1,-1,R.drawable._h326b,-1,R.drawable._h304f,R.drawable.
472         _h330f);
473     hall(R.drawable._h324b,R.drawable._h325b,R.drawable._h331f);
474     hall(R.drawable._h323b,R.drawable._h324b,R.drawable._h332f);
475     hall(R.drawable._h322b,R.drawable._h323b,R.drawable._h333f);
476     hall(R.drawable._h321b,R.drawable._h330f,-1,-1,-1,R.drawable._h326b,R.drawable.
477         _h304f);
478     hall(R.drawable._h320b,R.drawable.h302,-1,R.drawable._h321b,R.drawable.h300,-1,R.
479         drawable._h305f);
480     hall(R.drawable._h319b,-1,-1,R.drawable._h320b,-1,R.drawable.h303,R.drawable.
481         _h307f);
482     hall(R.drawable._h318b,-1,R.drawable.h304,R.drawable._h319b,-1,-1,R.drawable.
483         _h308f);
484     hall(R.drawable._h317b,-1,R.drawable.h306,R.drawable._h318b,-1,-1,R.drawable.
485         _h309f);
486     hall(R.drawable._h316b,-1,R.drawable.h309,R.drawable._h317b,-1,-1,R.drawable.
487         _h311f);
488     hall(R.drawable._h315b,-1,R.drawable._h316b,-1,R.drawable.h310,R.drawable._h313f,
489         R.drawable._h315f);
490     hall(R.drawable._h314b,R.drawable._h315b,R.drawable._h316f);
491     hall(R.drawable._h313b,-1,-1,R.drawable._h314b,R.drawable.h313,-1,R.drawable.
492         _h317f);
493     hall(R.drawable._h312b,-1,-1,R.drawable._h313b,R.drawable.h315,-1,R.drawable.
494         _h319f);
```

```
476 hall(R.drawable._h311b,-1,R.drawable.h317,R.drawable._h312b,R.drawable.h316,R.  
drawable.h318,R.drawable._h322f);  
477 hall(R.drawable._h310b,R.drawable._h311b,R.drawable._h323f);  
478 hall(R.drawable._h309b,-1,-1,R.drawable._h310b,-1,R.drawable.h320,R.drawable.  
_h324f);  
479 hall(R.drawable._h308b,R.drawable._h309b,R.drawable._h325f);  
480 hall(R.drawable._h307b,R.drawable.h322,R.drawable.h321,R.drawable._h308b,-1,-1,R.  
drawable._h326f);  
481 hall(R.drawable._h306b,-1,-1,R.drawable._h307b,R.drawable.h323,R.drawable.h324,R.  
drawable._h327f);  
482 hall(R.drawable._h305b,R.drawable._h306b,R.drawable._h328f);  
483 hall(R.drawable._h304b,R.drawable._h305b,R.drawable._h329f);  
484 hall(R.drawable._h303b,-1,R.drawable._h322b,-1,R.drawable._h304b,-1,R.drawable.  
_h334f);  
485 hall(R.drawable._h302b,R.drawable._h303b,R.drawable._h335f);  
486 hall(R.drawable._h301b,R.drawable._h302b,R.drawable._h337f);  
487  
488 /** G BLOCK BACK PASS */  
489 hall(R.drawable._g309b,-1,-1,-1,R.drawable._f301f,-1,R.drawable._g302f);  
490 hall(R.drawable._g308b,-1,R.drawable._g309b,-1,-1,-1,R.drawable._g303f);  
491 hall(R.drawable._g307b,R.drawable._g308b,R.drawable._g304f);  
492 hall(R.drawable._g306b,-1,-1,R.drawable._g307b,-1,R.drawable.g301_1,R.drawable.  
_g305f);  
493 hall(R.drawable._g305b,R.drawable._g306b,R.drawable._g306f);  
494 hall(R.drawable._g304b,R.drawable._g305b,R.drawable._g309f);  
495 hall(R.drawable._g303b,R.drawable._g304b,R.drawable._g310f);  
496 hall(R.drawable._g302b,-1,-1,R.drawable._g303b,-1,R.drawable.g305,R.drawable.  
_g311f);  
497 hall(R.drawable._g301b,-1,-1,R.drawable._g302b,R.drawable._g313f,-1,R.drawable.  
_g312f);  
498  
499 /** F BLOCK BACK PASS */  
500 hall(R.drawable._f305b,-1,-1,R.drawable._f301f,R.drawable._g301f,-1,R.drawable.  
_f303f);  
501 hall(R.drawable._f304b,R.drawable._f305b,R.drawable._f304f);  
502 hall(R.drawable._f303b,R.drawable._f304b,R.drawable._f305f);  
503 hall(R.drawable._f302b,R.drawable._f303b,R.drawable._f306f);  
504 hall(R.drawable._f301b,-1,-1,-1,R.drawable._f302b,-1,R.drawable._f308f);  
505  
506 /** C BLOCK BACK PASS */  
507 hall(R.drawable._c315b,R.drawable._b306b,R.drawable._c302f);  
508 hall(R.drawable._c314b,-1,-1,R.drawable._c315b,-1,R.drawable.c300,R.drawable.  
_c303f);  
509 hall(R.drawable._c313b,R.drawable._c314b,R.drawable._c304f);  
510 hall(R.drawable._c312b,-1,-1,R.drawable._c313b,-1,R.drawable.c303,R.drawable.  
_c305f);  
511 hall(R.drawable._c311b,-1,-1,R.drawable._c312b,-1,R.drawable.c304,R.drawable.  
_c306f);  
512 hall(R.drawable._c310b,R.drawable._c311b,R.drawable._c306f);  
513 hall(R.drawable._c309b,-1,R.drawable._f301b,-1,-1,-1,R.drawable._c313f);  
514 hall(R.drawable._c308b,-1,-1,R.drawable._c309b,-1,R.drawable.c308,R.drawable.  
_c314f);  
515 hall(R.drawable._c307b,R.drawable._c308b,R.drawable._c315f);  
516 hall(R.drawable._c306b,R.drawable._c307b,R.drawable._c316f);  
517 hall(R.drawable._c305b,-1,-1,-1,R.drawable._c310b,R.drawable._c306b,R.drawable.  
_c308f);  
518 hall(R.drawable._c304b,-1,-1,R.drawable._c305b,-1,R.drawable.c306_3,R.drawable.  
_c309f);  
519 hall(R.drawable._c303b,R.drawable._c304b,R.drawable._c310f);  
520 hall(R.drawable._c302b,-1,-1,-1,R.drawable._c303b,-1,R.drawable._c311f);  
521 hall(R.drawable._c301b,-1,R.drawable._c302b,R.drawable.c310,-1,-1,R.drawable.  
_c312f);  
522  
523 /** D BLOCK BACK PASS */  
524 hall(R.drawable._d334b,R.drawable._c301b,R.drawable._d301f);  
525 hall(R.drawable._d333b,R.drawable._d334b,-1,R.drawable.d300,-1,R.drawable._d313f,  
R.drawable._d303f);  
526 hall(R.drawable._d332b,R.drawable._d333b,R.drawable._d303f);  
527 hall(R.drawable._d331b,-1,-1,R.drawable._d304f,R.drawable._d332b,-1,R.drawable.  
_d305f);  
528 hall(R.drawable._d330b,R.drawable._d331b,R.drawable._d306f);
```



```
529 hall(R.drawable._d329b,R.drawable._d330b,R.drawable._d307f);
530 hall(R.drawable._d328b,R.drawable._d329b,R.drawable._d308f);
531 hall(R.drawable._d327b,R.drawable._d328b,R.drawable._d309f);
532 hall(R.drawable._d326b,-1,-1,-1,R.drawable._d327b,-1,R.drawable._d310f);
533 hall(R.drawable._d325b,-1,R.drawable._d303f,R.drawable._d334b,R.drawable.d300,R.
drawable.d301,R.drawable._d314f);
534 hall(R.drawable._d324b,-1,-1,R.drawable._d325b,-1,R.drawable.d303,R.drawable.
_d315f);
535 hall(R.drawable._d323b,-1,-1,R.drawable._d324b,-1,R.drawable.d304,R.drawable.
_d317f);
536 hall(R.drawable._d322b,R.drawable.d308,R.drawable._d318f,R.drawable._d323b,-1,-1,
R.drawable._d337f);
537 hall(R.drawable._d321b,R.drawable.d309_1,-1,R.drawable._d322b,-1,-1,R.drawable.
_d338f);
538 hall(R.drawable._d320b,R.drawable.d309_2,-1,R.drawable._d321b,-1,-1,R.drawable.
_d340f);
539 hall(R.drawable._d319b,-1,R.drawable._d326b,R.drawable._d323b,R.drawable._d336f,-
1,R.drawable._d319f);
540 hall(R.drawable._d318b,R.drawable._d319b,R.drawable._d320f);
541 hall(R.drawable._d317b,R.drawable._d318b,R.drawable._d321f);
542 hall(R.drawable._d316b,R.drawable._d317b,R.drawable._d322f);
543 hall(R.drawable._d315b,R.drawable._d316b,R.drawable._d323f);
544 hall(R.drawable._d314b,R.drawable._d315b,R.drawable._d323f);
545 hall(R.drawable._d313b,-1,-1,R.drawable._d314b,-1,R.drawable.d319,R.drawable.
_d325f);
546 hall(R.drawable._d312b,R.drawable._d313b,R.drawable._d327f);
547 hall(R.drawable._d311b,R.drawable.d317,R.drawable.d318,-1,R.drawable._d312b,-1,R.
drawable._d329f);
548 hall(R.drawable._d310b,R.drawable.d316,-1,R.drawable._d311b,-1,-1,R.drawable.
_d330f);
549 hall(R.drawable._d309b,R.drawable.d315,-1,R.drawable._d310b,-1,-1,R.drawable.
_d331f);
550 hall(R.drawable._d308b,R.drawable.d314,-1,R.drawable._d309b,-1,-1,R.drawable.
_d333f);
551 hall(R.drawable._d307b,R.drawable.d310,-1,R.drawable._d308b,-1,-1,R.drawable.
_d334f);
552 hall(R.drawable._d306b,R.drawable._d307b,-1,R.drawable._d320b,-1,-1,R.drawable.
_d340f);
553 hall(R.drawable._d305b,-1,R.drawable._d307b,R.drawable._d306b,-1,-1,R.drawable.
_d341f);
554 hall(R.drawable._d304b,-1,R.drawable._d305b,-1,R.drawable._d341f,-1,R.drawable.
_d342f);
555 hall(R.drawable._d303b,R.drawable._d343f,R.drawable._d343f,-1,-1,-1);
556 hall(R.drawable._d302b,R.drawable._d342f,R.drawable._d306b,-1,-1,R.drawable.g310,
R.drawable._j301f);
557 hall(R.drawable._d301b,R.drawable._d302b,R.drawable._j301f);
558
559 /** J BLOCK BACK PASS */
560 hall(R.drawable._j312b,-1,R.drawable._d303b,R.drawable._d301b,R.drawable._j303f,-
1,R.drawable._j302f);
561 hall(R.drawable._j311b,-1,R.drawable._j302f,-1,R.drawable._d301b,-1,R.drawable.
_j304f);
562 hall(R.drawable._j310b,-1,R.drawable.j328,R.drawable._j311b,-1,R.drawable.j327,R.
drawable._j305f);
563 hall(R.drawable._j309b,R.drawable._j310b,R.drawable._j306f);
564 hall(R.drawable._j308b,-1,-1,R.drawable._j308f,R.drawable._j309b,-1,R.drawable.
_j307f);
565 hall(R.drawable._j307b,-1,R.drawable._j309b,R.drawable._j307f,-1,-1,R.drawable.
_j308f);
566 hall(R.drawable._j306b,R.drawable._j307b,R.drawable._j309f);
567 hall(R.drawable._j305b,-1,R.drawable._j306b,-1,-1,-1,R.drawable._j312f);
568 hall(R.drawable._j304b,R.drawable._j305b,R.drawable._j313f);
569 hall(R.drawable._j303b,R.drawable.j310,-1,R.drawable._j304b,-1,-1,R.drawable.
_j313f);
570 hall(R.drawable._j302b,R.drawable._j303b,R.drawable._j314f);
571 hall(R.drawable._j301b,R.drawable._j302b,R.drawable._j315f);
572
573 /** BUTTON FIXES */
574
//hall(R.drawable._d343f,/*R.drawable._d304b*/-1,-1,-1,/*R.drawable._j312b*/-1,-1)
;
```

```
575 nodes[idx(R.drawable._d343f)].setOuterLeftNode(nodes[idx(R.drawable._d304b)]);
576 nodes[idx(R.drawable._d343f)].setInnerRightNode(nodes[idx(R.drawable._j312b)]);
577 //hall(R.drawable._d340f,-1,-1,R.drawable._d341f,-1,/*R.drawable._d307b*/-1);
578 nodes[idx(R.drawable._d340f)].setOuterRightNode(nodes[idx(R.drawable._d307b)]);
579
//hall(R.drawable._d339f,-1,-1,R.drawable._d340f,/*R.drawable._d307b*/-1,R.drawable
e.d309_2);
580 nodes[idx(R.drawable._d339f)].setInnerRightNode(nodes[idx(R.drawable._d307b)]);
581 //hall(R.drawable._d335f,/*R.drawable._d306b*/-1,-1,-1,-1,R.drawable._d341f);
582 nodes[idx(R.drawable._d335f)].setOuterLeftNode(nodes[idx(R.drawable._d306b)]);
583 //hall(R.drawable._d318f,-1,-1,R.drawable._d319f,/*R.drawable._d326b*/-1,-1);
584 nodes[idx(R.drawable._d318f)].setInnerRightNode(nodes[idx(R.drawable._d326b)]);
585 //hall(R.drawable._d312f,/*R.drawable._d319b*/-1,-1,-1,-1,R.drawable._d319f);
586 nodes[idx(R.drawable._d312f)].setOuterLeftNode(nodes[idx(R.drawable._d319b)]);
587 //hall(R.drawable._c306f,-1,-1,/*R.drawable._c306b*/-1,R.drawable._c307f,-1);
588 nodes[idx(R.drawable._c306f)].setCenterNode(nodes[idx(R.drawable._c306b)]);
589 //hall(R.drawable._c316f,-1,R.drawable._c306f,/*R.drawable._c310b*/-1,-1,-1);
590 nodes[idx(R.drawable._c316f)].setCenterNode(nodes[idx(R.drawable._c310b)]);
591 //hall(R.drawable._f301f,-1,R.drawable._f302f,-1,-1,/*R.drawable._h301b*/-1);
592 nodes[idx(R.drawable._f301f)].setInnerRightNode(nodes[idx(R.drawable._h301b)]);
593 //hall(R.drawable._h333f,-1,/*R.drawable._h303b*/-1,R.drawable._h334f,-1,-1);
594 nodes[idx(R.drawable._h333f)].setInnerLeftNode(nodes[idx(R.drawable._h303b)]);
595
//hall(R.drawable._h329f,-1,R.drawable._h334f,/*R.drawable._h303b*/-1,-1,/*R.drawa
ble._h322b*/-1);
596 nodes[idx(R.drawable._h329f)].setCenterNode(nodes[idx(R.drawable._h303b)]);
597 nodes[idx(R.drawable._h329f)].setOuterRightNode(nodes[idx(R.drawable._h322b)]);
598
599 /** TURN AROUND FIXES **/
600 nodes[idx(R.drawable._a301f)].setTurnAroundNode(nodes[idx(R.drawable._a307b)]);
601 nodes[idx(R.drawable._a305f)].setTurnAroundNode(nodes[idx(R.drawable._a305b)]);
602 nodes[idx(R.drawable._a306f)].setTurnAroundNode(nodes[idx(R.drawable._a305b)]);
603
604 nodes[idx(R.drawable._b307f)].setTurnAroundNode(nodes[idx(R.drawable._b302b)]);
605 nodes[idx(R.drawable._b308f)].setTurnAroundNode(nodes[idx(R.drawable._b317b)]);
606 nodes[idx(R.drawable._b310f)].setTurnAroundNode(nodes[idx(R.drawable._b313b)]);
607
608 nodes[idx(R.drawable._c307f)].setTurnAroundNode(nodes[idx(R.drawable._c310b)]);
609
610 nodes[idx(R.drawable._d302f)].setTurnAroundNode(nodes[idx(R.drawable._d334b)]);
611 nodes[idx(R.drawable._d311f)].setTurnAroundNode(nodes[idx(R.drawable._d326b)]);
612 nodes[idx(R.drawable._d312f)].setTurnAroundNode(nodes[idx(R.drawable._d326b)]);
613 nodes[idx(R.drawable._d313f)].setTurnAroundNode(nodes[idx(R.drawable._d325b)]);
614 nodes[idx(R.drawable._d316f)].setTurnAroundNode(nodes[idx(R.drawable._d324b)]);
615 nodes[idx(R.drawable._d318f)].setTurnAroundNode(nodes[idx(R.drawable._d319b)]);
616 nodes[idx(R.drawable._d324f)].setTurnAroundNode(nodes[idx(R.drawable._d313b)]);
617 nodes[idx(R.drawable._d326f)].setTurnAroundNode(nodes[idx(R.drawable._d313b)]);
618 nodes[idx(R.drawable._d328f)].setTurnAroundNode(nodes[idx(R.drawable._d312b)]);
619 nodes[idx(R.drawable._d332f)].setTurnAroundNode(nodes[idx(R.drawable._d309b)]);
620 nodes[idx(R.drawable._d335f)].setTurnAroundNode(nodes[idx(R.drawable._d307b)]);
621 nodes[idx(R.drawable._d336f)].setTurnAroundNode(nodes[idx(R.drawable._d322b)]);
622 nodes[idx(R.drawable._d339f)].setTurnAroundNode(nodes[idx(R.drawable._d306b)]);
623
624 nodes[idx(R.drawable._e302f)].setTurnAroundNode(nodes[idx(R.drawable._e309b)]);
625 nodes[idx(R.drawable._e310f)].setTurnAroundNode(nodes[idx(R.drawable._e303b)]);
626
627 nodes[idx(R.drawable._f302f)].setTurnAroundNode(nodes[idx(R.drawable._f305b)]);
628 nodes[idx(R.drawable._f307f)].setTurnAroundNode(nodes[idx(R.drawable._f302b)]);
629
630 nodes[idx(R.drawable._g307f)].setTurnAroundNode(nodes[idx(R.drawable._g305b)]);
631 nodes[idx(R.drawable._g308f)].setTurnAroundNode(nodes[idx(R.drawable._g304b)]);
632 nodes[idx(R.drawable._g313f)].setTurnAroundNode(nodes[idx(R.drawable._g301b)]);
633
634 nodes[idx(R.drawable._h303f)].setTurnAroundNode(nodes[idx(R.drawable._h326b)]);
635 nodes[idx(R.drawable._h306f)].setTurnAroundNode(nodes[idx(R.drawable._h320b)]);
636 nodes[idx(R.drawable._h310f)].setTurnAroundNode(nodes[idx(R.drawable._h317b)]);
637 nodes[idx(R.drawable._h312f)].setTurnAroundNode(nodes[idx(R.drawable._h316b)]);
638 nodes[idx(R.drawable._h318f)].setTurnAroundNode(nodes[idx(R.drawable._h313b)]);
639 nodes[idx(R.drawable._h320f)].setTurnAroundNode(nodes[idx(R.drawable._h311b)]);
640 nodes[idx(R.drawable._h321f)].setTurnAroundNode(nodes[idx(R.drawable._h311b)]);
641 nodes[idx(R.drawable._h336f)].setTurnAroundNode(nodes[idx(R.drawable._h302b)]);
```

```
642         nodes[idx(R.drawable._h338f)].setTurnAroundNode(nodes[idx(R.drawable._h301b)]);
643
644         nodes[idx(R.drawable._j303f)].setTurnAroundNode(nodes[idx(R.drawable._j311b)]);
645         nodes[idx(R.drawable._j310f)].setTurnAroundNode(nodes[idx(R.drawable._j306b)]);
646         nodes[idx(R.drawable._j311f)].setTurnAroundNode(nodes[idx(R.drawable._j305b)]);
647     }
648 }
649
```

```

1  /**
2   * TourHall.java
3   * Brock Butler
4   * A hallway node in the tour. Hallways have multiple branching points,
5   * and can turn around. Each hallway knows about all of it's branching points.
6   * Created by Taras Mychaskiw 2013-02-20
7   * Copyright (c) 2013 Sea Addicts. All rights reserved.
8   */
9  package edu.seaaddicts.brockbutler.tour;
10
11  import android.view.View;
12  import android.view.View.OnClickListener;
13  import android.widget.Toast;
14  import edu.seaaddicts.brockbutler.R;
15
16  public class TourHall extends TourNode {
17      protected TourNode[] nodes; //each of the nodes this node branches off to
18
19      /**
20       * Forward pass constructor. Defines each button, and inits the turn around
21       * location to null.
22       * @param img - the image resource value
23       * @param ll - outer left node
24       * @param ul - upper left node
25       * @param c - center node
26       * @param ur - upper right node
27       * @param lr - outer right node
28       */
29      public TourHall(int img, TourNode ll, TourNode ul, TourNode c, TourNode ur,
30          TourNode lr){
31          image = img;
32          nodes = new TourNode[5];
33          nodes[0] = ll;
34          nodes[1] = ul;
35          nodes[2] = c;
36          nodes[3] = ur;
37          nodes[4] = lr;
38          turnAroundNode = null;
39          title = makeTitle(img);
40      }
41
42      /**
43       * Second pass constructor. Defines each button, and also defines the turn
44       * around node. Links `turnAroundNode` back to this node via turning around.
45       * @param ta - turn around node
46       */
47      public TourHall(int img, TourNode ll, TourNode ul, TourNode c, TourNode ur,
48          TourNode lr, TourNode ta){
49          this(img,ll,ul,c,ur,lr);
50          turnAroundNode = ta;
51          turnAroundNode.setTurnAroundNode(this);
52      }
53
54      public void setOuterLeftNode(TourNode node){ nodes[0] = node; }
55      public void setInnerLeftNode(TourNode node){ nodes[1] = node; }
56      public void setCenterNode(TourNode node){ nodes[2] = node; }
57      public void setInnerRightNode(TourNode node){ nodes[3] = node; }
58      public void setOuterRightNode(TourNode node){ nodes[4] = node; }
59
60      /**
61       * @param img - the resource value of the image of this node
62       * @return - the value for `title`
63       */
64      private String makeTitle(int img){
65          switch (img){
66              case R.drawable.b303: return("B303");
67              case R.drawable.c306_3: return("C306");
68              case R.drawable.g301_1: return("G301");
69              case R.drawable.h304: return("H304");
70              case R.drawable.h306: return("H306");
71              case R.drawable.h306a6: case R.drawable.h306a9: return("H306A");

```

```

70     case R.drawable.h309: return("H309");
71     case R.drawable.h309a6: case R.drawable.h309a9: return("H309A");
72     case R.drawable.h318: return("H318");
73     case R.drawable._a301f: case R.drawable._a301b:
74         return("A Block");
75     case R.drawable._b301f: case R.drawable._b301b: case R.drawable._b306b:
76         return("B Block");
77     case R.drawable._c301f: case R.drawable._c301b: case R.drawable._c313f:
78         return("C Block");
79     case R.drawable._d301f: case R.drawable._d301b: case R.drawable._d304b:
80         return("D Block");
81     case R.drawable._e301f: case R.drawable._e301b:
82         return("E Block");
83     case R.drawable._f301b: case R.drawable._f302f:
84         return("F Block");
85     case R.drawable._g301f:
86         return("G Block");
87     case R.drawable._h301f: case R.drawable._h301b:
88         return("H Block");
89     case R.drawable._j301f: case R.drawable._j312b:
90         return("J Block");
91     }
92     return(null);
93 }
94
95 /**
96  * Changes the image displayed on the screen and redefines where the buttons lead
97  * us to.
98  * Also pushes this node onto the TourInfo's history.
99  */
100 @Override
101 public void paint(final TourInfo info){
102     info.rl.setBackgroundResource(image);
103     info.current = this;
104     if (title != null)
105         Toast.makeText(info.context,title,Toast.LENGTH_SHORT).show();
106     for (int i = 0; i < nodes.length; i++){
107         final int idx = i;
108         info.buttons[idx].setOnClickListener(new OnClickListener(){
109             public void onClick(View v){
110                 if (nodes[idx] != null){
111                     info.history.push(TourHall.this);
112                     nodes[idx].paint(info);
113                 }
114             }
115         });
116         int resID = (nodes[idx] == null)? R.drawable.qb_empty : info.arrows[idx];
117         info.buttons[idx].setImageResource(resID);
118     }
119 }
120

```

```
1  /**
2   * TourInfo.java
3   * Brock Butler
4   * Wrapper class for all passing required information around throughout the tour.
5   * Created by Taras Mychaskiw 2013-02-20
6   * Copyright (c) 2013 Sea Addicts. All rights reserved.
7   */
8  package edu.seaaddicts.brockbutler.tour;
9
10 import java.util.Stack;
11
12 import android.content.Context;
13 import android.widget.ImageButton;
14 import android.widget.RelativeLayout;
15 import edu.seaaddicts.brockbutler.R;
16
17 public class TourInfo {
18     public RelativeLayout rl;
19     public ImageButton[] buttons;
20     public int[] arrows;
21     public Context context;
22     public TourNode current;
23     public Stack<TourNode> history;
24
25     /**
26      * @param r - the layout of which the background is changed from node to node
27      * @param b - array of ImageButtons where one would tap to change nodes
28      * @param c - getApplicationContext(), for toasts
29      */
30     public TourInfo(RelativeLayout r, ImageButton[] b, Context c){
31         rl = r;
32         buttons = b;
33         context = c;
34         current = null;
35         history = new Stack<TourNode>();
36         arrows = new int[5];
37         arrows[0] = R.drawable.qb_outerleft;
38         arrows[1] = R.drawable.qb_innerleft;
39         arrows[2] = R.drawable.qb_center;
40         arrows[3] = R.drawable.qb_innerright;
41         arrows[4] = R.drawable.qb_outerright;
42     }
43 }
44
```

```
1  /**
2   * TourNode.java
3   * Brock Butler
4   * Simple abstract wrapper for TourRoom and TourHall.
5   * Created by Taras Mychaskiw 2013-02-20
6   * Copyright (c) 2013 Sea Addicts. All rights reserved.
7   */
8  package edu.seaaddicts.brockbutler.tour;
9
10 public abstract class TourNode {
11     protected int image;    //resource for the image on this node
12     protected TourNode turnAroundNode;
13     protected String title; //text for Toast to display, if any
14
15     public boolean canTurnAround(){ return(turnAroundNode != null); }
16     public TourNode getTurnAroundNode(){ return(turnAroundNode); }
17
18     public void setTurnAroundNode(TourNode node){ turnAroundNode = node; }
19
20     public void setOuterLeftNode(TourNode node){}
21     public void setInnerLeftNode(TourNode node){}
22     public void setCenterNode(TourNode node){}
23     public void setInnerRightNode(TourNode node){}
24     public void setOuterRightNode(TourNode node){}
25
26     /**
27      * Changes the background image and redefines what the buttons do,
28      * ie where the buttons will take you.
29      * @param info - current tour state
30      */
31     public abstract void paint(TourInfo info);
32 }
33
```

```
1  /**
2   * TourRoom.java
3   * Brock Butler
4   * A room node in the tour. Rooms are simple, they don't go anywhere.
5   * Created by Taras Mychaskiw 2013-02-20
6   * Copyright (c) 2013 Sea Addicts. All rights reserved.
7   */
8  package edu.seaaddicts.brockbutler.tour;
9
10 import android.view.View;
11 import android.view.View.OnClickListener;
12 import android.widget.Toast;
13 import edu.seaaddicts.brockbutler.R;
14
15 public class TourRoom extends TourNode {
16
17     /**
18      * Initializes this room, rooms only need the image resource value
19      * @param img - the resource value of the image of this room
20      */
21     public TourRoom(int img, String s){
22         image = img;
23         title = s;
24         turnAroundNode = null;
25     }
26
27     @Override
28     public void paint(TourInfo info) {
29         info.rl.setBackgroundResource(image);
30         info.current = this;
31         if (title != null) //should never be null, but just in case
32             Toast.makeText(info.context,title,Toast.LENGTH_SHORT).show();
33         for (int i = 0; i < info.buttons.length; i++){
34             info.buttons[i].setOnClickListener(new OnClickListener(){
35                 public void onClick(View v){}
36             });
37             info.buttons[i].setImageResource(R.drawable.qb_empty);
38         }
39     }
40 }
41
```



```
1 <menu xmlns:android="http://schemas.android.com/apk/res/android" >
2
3     <item
4         android:id="@+id/menu_go_back"
5         android:onClick="goBack"
6         android:title="@string/menu_go_back"
7         android:titleCondensed="@string/menu_go_back">
8     </item>
9     <item
10        android:id="@+id/menu_turn_around"
11        android:onClick="turnAround"
12        android:title="@string/menu_turn_around"
13        android:titleCondensed="@string/menu_turn_around">
14    </item>
15    <item
16        android:id="@+id/menu_teleport"
17        android:title="@string/menu_teleport"
18        android:titleCondensed="@string/menu_teleport">
19    <menu>
20        <item
21            android:id="@+id/teleport_a_block"
22            android:onClick="teleport"
23            android:title="@string/a_block"
24            android:titleCondensed="@string/a_block"/>
25        <item
26            android:id="@+id/teleport_b_block"
27            android:onClick="teleport"
28            android:title="@string/b_block"
29            android:titleCondensed="@string/b_block"/>
30        <item
31            android:id="@+id/teleport_c_block"
32            android:onClick="teleport"
33            android:title="@string/c_block"
34            android:titleCondensed="@string/c_block"/>
35        <item
36            android:id="@+id/teleport_d_block"
37            android:onClick="teleport"
38            android:title="@string/d_block"
39            android:titleCondensed="@string/d_block"/>
40        <item
41            android:id="@+id/teleport_e_block"
42            android:onClick="teleport"
43            android:title="@string/e_block"
44            android:titleCondensed="@string/e_block"/>
45        <item
46            android:id="@+id/teleport_f_block"
47            android:onClick="teleport"
48            android:title="@string/f_block"
49            android:titleCondensed="@string/f_block"/>
50        <item
51            android:id="@+id/teleport_g_block"
52            android:onClick="teleport"
53            android:title="@string/g_block"
54            android:titleCondensed="@string/g_block"/>
55        <item
56            android:id="@+id/teleport_h_block"
57            android:onClick="teleport"
58            android:title="@string/h_block"
59            android:titleCondensed="@string/h_block"/>
60        <item
61            android:id="@+id/teleport_j_block"
62            android:onClick="teleport"
63            android:title="@string/j_block"
64            android:titleCondensed="@string/j_block"/>
65    </menu>
66 </item>
67 <item
68     android:id="@+id/menu_end_tour"
69     android:onClick="endTour"
70     android:title="@string/menu_end_tour"
71     android:titleCondensed="@string/menu_end_tour">
```

```
72         </item>
73
74     </menu>
```

```
1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:id="@+id/screen"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:gravity="center"
7   tools:context=".TourHandler" >
8
9   <ImageButton
10      android:id="@+id/outerleft"
11      android:contentDescription="@string/go_left"
12      android:background="@android:color/transparent"
13      android:src="@drawable/qb_outerleft"
14      android:padding="5dp"
15      android:scaleType="fitEnd"
16      android:layout_width="75dp"
17      android:layout_height="400dp"
18      android:layout_alignParentLeft="true"
19      android:layout_marginLeft="10dp"
20      android:layout_centerVertical="true" />
21
22   <ImageButton
23      android:id="@+id/innerleft"
24      android:contentDescription="@string/go_left"
25      android:background="@android:color/transparent"
26      android:src="@drawable/qb_innerleft"
27      android:padding="5dp"
28      android:scaleType="fitEnd"
29      android:layout_width="75dp"
30      android:layout_height="400dp"
31      android:layout_marginLeft="20dp"
32      android:layout_toRightOf="@+id/outerleft"
33      android:layout_centerVertical="true" />
34
35   <ImageButton
36      android:id="@+id/center"
37      android:contentDescription="@string/go_forward"
38      android:background="@android:color/transparent"
39      android:src="@drawable/qb_center"
40      android:padding="5dp"
41      android:scaleType="fitEnd"
42      android:layout_width="75dp"
43      android:layout_height="400dp"
44      android:layout_centerHorizontal="true"
45      android:layout_centerVertical="true" />
46
47   <ImageButton
48      android:id="@+id/innerright"
49      android:contentDescription="@string/go_right"
50      android:background="@android:color/transparent"
51      android:src="@drawable/qb_innerright"
52      android:padding="5dp"
53      android:scaleType="fitEnd"
54      android:layout_width="75dp"
55      android:layout_height="400dp"
56      android:layout_marginRight="20dp"
57      android:layout_toLeftOf="@+id/outright"
58      android:layout_centerVertical="true" />
59
60   <ImageButton
61      android:id="@+id/outright"
62      android:contentDescription="@string/go_right"
63      android:background="@android:color/transparent"
64      android:src="@drawable/qb_outright"
65      android:padding="5dp"
66      android:scaleType="fitEnd"
67      android:layout_width="75dp"
68      android:layout_height="400dp"
69      android:layout_alignParentRight="true"
70      android:layout_marginRight="10dp"
71      android:layout_centerVertical="true" />
```

```
72
73     </RelativeLayout>
74
```