

```

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

```