

```
1 package com.example.gridgamefinal;
2
3 import java.util.ArrayList;
4
5 public class Player {
6     private String name;
7     private ArrayList<BoardSquare> owned = new
8         ArrayList<>();
9     private ArrayList<BoardPiece> pieces = new
10        ArrayList<>();
11
12    public Player(String n, ArrayList p){
13        name = n;
14        pieces = p;
15    }
16    public void removeFromOwned(BoardSquare r){
17        owned.remove(r);
18    }
19    public void setOwned(BoardSquare b) {
20        owned.add(b);
21    }
22
23    public String getName() {
24        return name;
25    }
26
27    public ArrayList<BoardPiece> getPieces() {
28        return pieces;
29    }
30
31    public void changeLocation(int droll, String[]
32        boardPieceNames, BoardSquare[][] board){
33        //find number of board piece
34        //      currentPieceNumber=0;
35        //      for (int i = 0; i < boardPieceNames.length
36        //          ; i++) {
37        //          if (currentBoardPiece.getName().equals(
38        //              boardPieceNames[i])) {
39        //              currentPieceNumber=i;
```

```
37 //           }
38 //           }
39 //           System.out.println("current piece # " +
40 //           currentPieceNumber);
41 //           if(currentPieceNumber+droll>boardPieceNames
42 //           .length-1){
43 //           currentPieceNumber = droll+
44 //           currentPieceNumber- boardPieceNames.length;
45 //           }else{
46 //           currentPieceNumber+=droll;
47 //           }
48 //           for (int i = 0; i < board.length; i++) {
49 //               for (int j = 0; j < board[0].length; j
50 //               ++){}
51 //               if (board[i][j].getName().equals(
52 //               boardPieceNames[currentPieceNumber])){
53 //                   playerLoc.setRow(board[i][j].
54 //                   getRowLoc());
55 //                   playerLoc.setColumn(board[i][j]
56 //                   .getColLoc());
57 //                   currentBoardPiece = board[i][j];
58 //               }
59 }
```

```
1 package com.example.gridgamefinal;
2
3 public class Location {
4     private int row;
5     private int column;
6
7     public Location(int r, int c){
8         row = r;
9         column = c;
10    }
11    public int getRow() {
12        return row;
13    }
14
15    public int getColumn() {
16        return column;
17    }
18
19    public void setColumn(int column) {
20        this.column = column;
21    }
22
23    public void setRow(int row) {
24        this.row = row;
25    }
26
27    // done
28 }
29
```

```
1 package com.example.gridgamefinal;
2
3 import javafx.scene.image.Image;
4
5 public class BoardPiece {
6     private String name;
7     private Image image;
8     private Image image2;
9     private int owner;
10    public BoardPiece(String n, Image img,int own){
11        name = n;
12        image = img;
13        //0,1,2,3,4  0=unowned
14        owner = own;
15    }
16
17    public BoardPiece(String n, Image img, Image img2
18 , int own){
19        name = n;
20        image = img;
21        image2 = img2;
22        //0,1,2  2=unowned
23        owner = own;
24    }
25
26    public Image getImage() {
27        return image;
28    }
29
30    public Image getImage2() {
31        return image2;
32    }
33
34    public void setName(String name) {
35        this.name = name;
36    }
37
38    public String getName() {
39        return name;
40    }
```

```
41     public int getOwner() {  
42         return owner;  
43     }  
44 }  
45
```

```
1 package com.example.gridgamefinal;
2
3 import javafx.scene.image.Image;
4
5 import java.util.ArrayList;
6
7 public class BoardSquare {
8     private String name;
9     private Location loc;
10    private Image image;
11    private int power;
12
13    private int owner;
14    public BoardSquare(int row, int column, String n
15 , Image img,int own,int p){
16         name = n;
17         loc = new Location(row,column);
18         image = img;
19         //0,1,2  0=unowned
20         owner = own;
21         power = p;
22     }
23
24     public int getPower() {
25         return power;
26     }
27     public void changeLocation(int r,int c){
28         loc.setRow(r);
29         loc.setColumn(c);
30     }
31
32     public Image getImage() {
33         return image;
34     }
35
36     public void setName(String name) {
37         this.name = name;
38     }
39
40     public String getName() {
41         return name;
```

```
41      }
42
43      public Location getLoc() {
44          return loc;
45      }
46      public int getRowLoc(){
47          return loc.getRow();
48      }
49      public int getColLoc(){
50          return loc.getColumn();
51      }
52 }
53
```

```
1 package com.example.gridgamefinal;
2
3 import javafx.event.ActionEvent;
4 import javafx.event.EventHandler;
5 import javafx.fxml.FXML;
6 import javafx.scene.control.Button;
7 import javafx.scene.image.Image;
8 import javafx.scene.image.ImageView;
9 import javafx.scene.input.MouseEvent;
10 import javafx.scene.layout.AnchorPane;
11 import javafx.scene.layout.GridPane;
12
13 import java.io.FileInputStream;
14 import java.io.FileNotFoundException;
15 import java.lang.reflect.Array;
16 import java.util.ArrayList;
17 import java.util.Arrays;
18 import java.util.function.DoubleToIntFunction;
19
20 public class HelloController {
21
22     @FXML
23     GridPane gpane;
24     private ArrayList<Player> currentPlayers = new
25     ArrayList<>();
26     private BoardSquare[][][] board = new BoardSquare[15][15];
27     private ImageView[][][] boardImages = new ImageView[15][15];
28     private BoardPiece airbase, fighter, bomber, samSilo,
29     , nuke, radar, fleet, battleship, carrier, sub;
30     //startingPieces dont include nukes
31     private ArrayList<BoardPiece> startingPieces =
32     new ArrayList<>(Arrays.asList(airbase, airbase, airbase,
33     , fighter, fighter, fighter, bomber, bomber, samSilo,
34     , samSilo, samSilo, samSilo, samSilo, samSilo, samSilo,
35     , samSilo, samSilo, samSilo, radar, radar, radar, radar,
36     , radar, radar, radar, fleet, fleet, fleet, battleship,
37     , carrier, sub));
```

```
32
33     @FXML
34     private ImageView b0000, b0001, b0002, b0003,
35         b0004, b0005, b0006, b0007, b0008, b0009, b0010,
36         b0011, b0012, b0013, b0014, b0100, b0101, b0102,
37         b0103, b0104, b0105, b0106, b0107, b0108, b0109,
38         b0110, b0111, b0112, b0113, b0114, b0200, b0201,
39         b0202, b0203, b0204, b0205, b0206, b0207, b0208,
40         b0209, b0210, b0211, b0212, b0213, b0214, b0300,
41         b0301, b0302, b0303, b0304, b0305, b0306, b0307,
42         b0308, b0309, b0310, b0311, b0312, b0313, b0314,
43         b0400, b0401, b0402, b0403, b0404, b0405, b0406,
44         b0407, b0408, b0409, b0410, b0411, b0412, b0413,
45         b0414, b0500, b0501, b0502, b0503, b0504, b0505,
46         b0506, b0507, b0508, b0509, b0510, b0511, b0512,
47         b0513, b0514, b0600, b0601, b0602, b0603, b0604,
48         b0605, b0606, b0607, b0608, b0609, b0610, b0611,
49         b0612, b0613, b0614, b0700, b0701, b0702, b0703,
50         b0704, b0705, b0706, b0707, b0708, b0709, b0710,
51         b0711, b0712, b0713, b0714, b0800, b0801, b0802,
52         b0803, b0804, b0805, b0806, b0807, b0808, b0809,
53         b0810, b0811, b0812, b0813, b0814, b0900, b0901,
54         b0902, b0903, b0904, b0905, b0906, b0907, b0908,
55         b0909, b0910, b0911, b0912, b0913, b0914, b1000,
56         b1001, b1002, b1003, b1004, b1005, b1006, b1007,
57         b1008, b1009, b1010, b1011, b1012, b1013, b1014,
58         b1100, b1101, b1102, b1103, b1104, b1105, b1106,
59         b1107, b1108, b1109, b1110, b1111, b1112, b1113,
60         b1114, b1200, b1201, b1202, b1203, b1204, b1205,
61         b1206, b1207, b1208, b1209, b1210, b1211, b1212,
62         b1213, b1214, b1300, b1301, b1302, b1303, b1304,
63         b1305, b1306, b1307, b1308, b1309, b1310, b1311,
64         b1312, b1313, b1314, b1400, b1401, b1402, b1403,
65         b1404, b1405, b1406, b1407, b1408, b1409, b1410,
66         b1411, b1412, b1413, b1414;
67     private String[] boardSquareNames = {"b1", "b2",
68         "b3", "b4", "r1", "r2", "r3", "r4", "unowned"};
69     private int playerTurn=0;
70     Image red,orange,purple,dblue,lblue,airbasePic,
71         fighterPic,bomberPic,samPic,siloPic,nukePic,radarPic,
72         fleetPic,battleshipPic,carrierPic,subPic;
```

```
38     public HelloController(){
39         FileInputStream redd,orangee,purplee,dbluee,
40             lbluee,airbasee,fighterr,bomber,samm,siloo,nukee,
41             radarr,fleett,battleshipp,carrierr,subb;
42         try {
43             redd = new FileInputStream("src/main/
44                 Pictures/red.png");
45             red = new Image(redd);
46             orangee= new FileInputStream("src/main/
47                 Pictures/orange.png");
48             orange = new Image(orangee);
49             purplee = new FileInputStream("src/main/
50                 Pictures/purple.jpeg");
51             purple = new Image(purplee);
52             dbluee = new FileInputStream("src/main/
53                 Pictures/dark-blue.jpeg");
54             dblue = new Image(dbluee);
55             lbluee = new FileInputStream("src/main/
56                 Pictures/ocean-blue.jpeg");
57             lblue = new Image(lbluee);
58
59             airbasee = new FileInputStream("src/main/
60                 Pictures/Airbase.png");
61             airbasePic = new Image(airbasee);
62             fighterr = new FileInputStream("src/main/
63                 Pictures/Fighter.png");
64             fighterPic = new Image(fighterr);
65             bomber = new FileInputStream("src/main/
66                 Pictures/Bomber.png");
67             bomberPic = new Image(bomber);
68
69             samm = new FileInputStream("src/main/
70                 Pictures/Sam.png");
71             samPic = new Image(samm);
72             siloo = new FileInputStream("src/main/
73                 Pictures/Silo.png");
74             siloPic = new Image(siloo);
75             nukee = new FileInputStream("src/main/
76                 Pictures/Nuke.png");
77             nukePic = new Image(nukee);
```

```
66         radarr = new FileInputStream("src/main/
67             Pictures/Radar.png");
68         radarPic = new Image(radarr);
69
70         fleett = new FileInputStream("src/main/
71             Pictures/Fleet.png");
72         fleetPic = new Image(fleett);
73         battleshipp = new FileInputStream("src/
74             main/Pictures/Battleship.png");
75         battleshipPic = new Image(battleshipp);
76         carrierr = new FileInputStream("src/main/
77             Pictures/Carrier.png");
78         carrierPic = new Image(carrierr);
79         subb = new FileInputStream("src/main/
80             Pictures/Sub.png");
81         subPic = new Image(subb);
82     } catch (FileNotFoundException e) {
83         e.printStackTrace();
84     }
85
86     airbase = new BoardPiece("airbase",
87         airbasePic, 2);
88     fighter = new BoardPiece("fighter",
89         fighterPic, 2);
90     bomber = new BoardPiece("bomber", bomberPic
91         , 2);
92     samSilo = new BoardPiece("sam", samPic,
93         siloPic, 2);
94     nuke = new BoardPiece("nuke", nukePic, 2);
95     radar = new BoardPiece("radar", radarPic, 2
96 );
97     fleet = new BoardPiece("fleet", fleetPic, 2
98 );
99     battleship = new BoardPiece("battleship",
100        battleshipPic, 2);
101    carrier = new BoardPiece("carrier",
102        carrierPic, 2);
103    sub = new BoardPiece("sub", subPic, 2);
104
105
106    for (int i = 0; i < 5; i++) {
```

```
94         for (int j = 0; j < 5; j++) {
95             board[i][j] = new BoardSquare(i,j,
96                 boardSquareNames[0],red,0,2);
97         }
98     board[0][5] = new BoardSquare(0,5,
99         boardSquareNames[0],red,0,2);
100    board[0][6] = new BoardSquare(0,6,
101        boardSquareNames[0],red,0,2);
102    board[1][5] = new BoardSquare(1,5,
103        boardSquareNames[0],red,0,2);
104    board[1][6] = new BoardSquare(1,6,
105        boardSquareNames[0],red,0,2);
106    board[2][5] = new BoardSquare(2,5,
107        boardSquareNames[0],red,0,2);
108
109
110
111    for (int i = 0; i < 5; i++) {
112        for (int j = 0; j < 5; j++) {
113            board[i][14-j] = new BoardSquare(i,
114                14-j,boardSquareNames[1],orange,0,2);
115        }
116    board[2][9] = new BoardSquare(2,9,
117        boardSquareNames[1],orange,0,2);
118    board[3][8] = new BoardSquare(3,8,
119        boardSquareNames[1],orange,0,2);
120    board[3][9] = new BoardSquare(3,9,
121        boardSquareNames[1],orange,0,2);
122    board[4][8] = new BoardSquare(4,8,
```

```
119 boardSquareNames[1],orange,0,2);  
120         board[4][9] = new BoardSquare(4,9,  
121             boardSquareNames[1],orange,0,2);  
122         board[5][11] = new BoardSquare(5,11,  
123             boardSquareNames[1],orange,0,2);  
124         board[5][12] = new BoardSquare(5,12,  
125             boardSquareNames[1],orange,0,2);  
126         board[5][13] = new BoardSquare(5,13,  
127             boardSquareNames[1],orange,0,2);  
128         board[5][14] = new BoardSquare(5,14,  
129             boardSquareNames[1],orange,0,2);  
130         board[6][14] = new BoardSquare(6,14,  
131             boardSquareNames[1],orange,0,2);  
132     }  
133     for (int i = 0; i < 5; i++) {  
134         for (int j = 0; j < 5; j++) {  
135             board[14-i][j] = new BoardSquare(14-  
136                 i,j,boardSquareNames[2],dblue,0,2);  
137         }  
138     }  
139     board[8][4] = new BoardSquare(8,4,  
140         boardSquareNames[2],dblue,0,2);  
141     board[9][3] = new BoardSquare(9,3,  
142         boardSquareNames[2],dblue,0,2);  
143     board[9][4] = new BoardSquare(9,4,  
144         boardSquareNames[2],dblue,0,2);  
145     board[9][5] = new BoardSquare(9,5,  
146         boardSquareNames[2],dblue,0,2);  
147     board[10][5] = new BoardSquare(10,5,  
148         boardSquareNames[2],dblue,0,2);  
149     board[12][6] = new BoardSquare(12,6,  
150         boardSquareNames[2],dblue,0,2);  
151     board[13][5] = new BoardSquare(13,5,  
152         boardSquareNames[2],dblue,0,2);  
153     board[13][6] = new BoardSquare(13,6,  
154         boardSquareNames[2],dblue,0,2);  
155     board[14][5] = new BoardSquare(14,5,  
156         boardSquareNames[2],dblue,0,2);  
157     board[14][6] = new BoardSquare(14,6,
```

```
143 boardSquareNames[2], dblue, 0, 2);
144
145     for (int i = 0; i < 5; i++) {
146         for (int j = 0; j < 5; j++) {
147             board[14-i][14-j] = new BoardSquare(
148                 14-i, 14-j, boardSquareNames[3], purple, 0, 2);
149         }
150     }
151     board[8][12] = new BoardSquare(8, 12,
152         boardSquareNames[3], purple, 0, 2);
153     board[9][9] = new BoardSquare(9, 9,
154         boardSquareNames[3], purple, 0, 2);
155     board[9][12] = new BoardSquare(9, 12,
156         boardSquareNames[3], purple, 0, 2);
157     board[9][13] = new BoardSquare(9, 13,
158         boardSquareNames[3], purple, 0, 2);
159     board[9][14] = new BoardSquare(9, 14,
160         boardSquareNames[3], purple, 0, 2);
161
162     board[10][8] = new BoardSquare(10, 8,
163         boardSquareNames[3], purple, 0, 2);
164     board[10][9] = new BoardSquare(10, 9,
165         boardSquareNames[3], purple, 0, 2);
166     board[11][8] = new BoardSquare(11, 8,
167         boardSquareNames[3], purple, 0, 2);
168     board[11][9] = new BoardSquare(11, 9,
169         boardSquareNames[3], purple, 0, 2);
170     board[12][9] = new BoardSquare(12, 9,
171         boardSquareNames[3], purple, 0, 2);
172     //board pieces
173
174     for (int i = 0; i < 15; i++) {
175         for (int j = 0; j < 15; j++) {
176             if (board[i][j] == null) {
177                 board[i][j] = new BoardSquare(i,
178                     j, boardSquareNames[8], lblue, 2, 0);
179             }
180         }
181     }
182     currentPlayers.add(new Player("Player 1",
183         "Player 2"));
184 }
```

```
171 startingPieces));
172         currentPlayers.add(new Player("Player 2",
173             startingPieces));
174         currentPlayers.add(new Player("Player 3",
175             startingPieces));
176         currentPlayers.add(new Player("Player 4",
177             startingPieces));
178     for (int i = 0; i < 15; i++) {
179         for (int j = 0; j < 15; j++) {
180             if (board[i][j].getImage() == red)
181                 currentPlayers.get(0).setOwned(
182                     board[i][j]);
183             if (board[i][j].getImage() == dblue)
184                 currentPlayers.get(1).setOwned(
185                     board[i][j]);
186             if (board[i][j].getImage() == purple)
187                 currentPlayers.get(2).setOwned(
188                     board[i][j]);
189             if (board[i][j].getImage() == orange)
190                 currentPlayers.get(3).setOwned(
191                     board[i][j]);
192     }
193 }
194 }
195 @FXML
196 protected void handleStart(ActionEvent event){
197     boolean idk = true;
198
199     if (idk){
200         boardImages[0][0]=b0000;
```

```
201         boardImages[0][1]=b0001;
202         boardImages[0][2]=b0002;
203         boardImages[0][3]=b0003;
204         boardImages[0][4]=b0004;
205         boardImages[0][5]=b0005;
206         boardImages[0][6]=b0006;
207         boardImages[0][7]=b0007;
208         boardImages[0][8]=b0008;
209         boardImages[0][9]=b0009;
210         boardImages[0][10]=b0010;
211         boardImages[0][11]=b0011;
212         boardImages[0][12]=b0012;
213         boardImages[0][13]=b0013;
214         boardImages[0][14]=b0014;
215
216         boardImages[1][0]=b0100;
217         boardImages[1][1]=b0101;
218         boardImages[1][2]=b0102;
219         boardImages[1][3]=b0103;
220         boardImages[1][4]=b0104;
221         boardImages[1][5]=b0105;
222         boardImages[1][6]=b0106;
223         boardImages[1][7]=b0107;
224         boardImages[1][8]=b0108;
225         boardImages[1][9]=b0109;
226         boardImages[1][10]=b0110;
227         boardImages[1][11]=b0111;
228         boardImages[1][12]=b0112;
229         boardImages[1][13]=b0113;
230         boardImages[1][14]=b0114;
231
232         boardImages[2][0]=b0200;
233         boardImages[2][1]=b0201;
234         boardImages[2][2]=b0202;
235         boardImages[2][3]=b0203;
236         boardImages[2][4]=b0204;
237         boardImages[2][5]=b0205;
238         boardImages[2][6]=b0206;
239         boardImages[2][7]=b0207;
240         boardImages[2][8]=b0208;
241         boardImages[2][9]=b0209;
```

```
242         boardImages[2][10]=b0210;
243         boardImages[2][11]=b0211;
244         boardImages[2][12]=b0212;
245         boardImages[2][13]=b0213;
246         boardImages[2][14]=b0214;
247
248         boardImages[3][0]=b0300;
249         boardImages[3][1]=b0301;
250         boardImages[3][2]=b0302;
251         boardImages[3][3]=b0303;
252         boardImages[3][4]=b0304;
253         boardImages[3][5]=b0305;
254         boardImages[3][6]=b0306;
255         boardImages[3][7]=b0307;
256         boardImages[3][8]=b0308;
257         boardImages[3][9]=b0309;
258         boardImages[3][10]=b0310;
259         boardImages[3][11]=b0311;
260         boardImages[3][12]=b0312;
261         boardImages[3][13]=b0313;
262         boardImages[3][14]=b0314;
263
264         boardImages[4][0]=b0400;
265         boardImages[4][1]=b0401;
266         boardImages[4][2]=b0402;
267         boardImages[4][3]=b0403;
268         boardImages[4][4]=b0404;
269         boardImages[4][5]=b0405;
270         boardImages[4][6]=b0406;
271         boardImages[4][7]=b0407;
272         boardImages[4][8]=b0408;
273         boardImages[4][9]=b0409;
274         boardImages[4][10]=b0410;
275         boardImages[4][11]=b0411;
276         boardImages[4][12]=b0412;
277         boardImages[4][13]=b0413;
278         boardImages[4][14]=b0414;
279
280         boardImages[5][0]=b0500;
281         boardImages[5][1]=b0501;
282         boardImages[5][2]=b0502;
```

```
283         boardImages[5][3]=b0503;
284         boardImages[5][4]=b0504;
285         boardImages[5][5]=b0505;
286         boardImages[5][6]=b0506;
287         boardImages[5][7]=b0507;
288         boardImages[5][8]=b0508;
289         boardImages[5][9]=b0509;
290         boardImages[5][10]=b0510;
291         boardImages[5][11]=b0511;
292         boardImages[5][12]=b0512;
293         boardImages[5][13]=b0513;
294         boardImages[5][14]=b0514;
295
296         boardImages[6][0]=b0600;
297         boardImages[6][1]=b0601;
298         boardImages[6][2]=b0602;
299         boardImages[6][3]=b0603;
300         boardImages[6][4]=b0604;
301         boardImages[6][5]=b0605;
302         boardImages[6][6]=b0606;
303         boardImages[6][7]=b0607;
304         boardImages[6][8]=b0608;
305         boardImages[6][9]=b0609;
306         boardImages[6][10]=b0610;
307         boardImages[6][11]=b0611;
308         boardImages[6][12]=b0612;
309         boardImages[6][13]=b0613;
310         boardImages[6][14]=b0614;
311
312         boardImages[7][0]=b0700;
313         boardImages[7][1]=b0701;
314         boardImages[7][2]=b0702;
315         boardImages[7][3]=b0703;
316         boardImages[7][4]=b0704;
317         boardImages[7][5]=b0705;
318         boardImages[7][6]=b0706;
319         boardImages[7][7]=b0707;
320         boardImages[7][8]=b0708;
321         boardImages[7][9]=b0709;
322         boardImages[7][10]=b0710;
323         boardImages[7][11]=b0711;
```

```
324         boardImages[7][12]=b0712;
325         boardImages[7][13]=b0713;
326         boardImages[7][14]=b0714;
327
328         boardImages[8][0]=b0800;
329         boardImages[8][1]=b0801;
330         boardImages[8][2]=b0802;
331         boardImages[8][3]=b0803;
332         boardImages[8][4]=b0804;
333         boardImages[8][5]=b0805;
334         boardImages[8][6]=b0806;
335         boardImages[8][7]=b0807;
336         boardImages[8][8]=b0808;
337         boardImages[8][9]=b0809;
338         boardImages[8][10]=b0810;
339         boardImages[8][11]=b0811;
340         boardImages[8][12]=b0812;
341         boardImages[8][13]=b0813;
342         boardImages[8][14]=b0814;
343
344         boardImages[9][0]=b0900;
345         boardImages[9][1]=b0901;
346         boardImages[9][2]=b0902;
347         boardImages[9][3]=b0903;
348         boardImages[9][4]=b0904;
349         boardImages[9][5]=b0905;
350         boardImages[9][6]=b0906;
351         boardImages[9][7]=b0907;
352         boardImages[9][8]=b0908;
353         boardImages[9][9]=b0909;
354         boardImages[9][10]=b0910;
355         boardImages[9][11]=b0911;
356         boardImages[9][12]=b0912;
357         boardImages[9][13]=b0913;
358         boardImages[9][14]=b0914;
359
360         boardImages[10][0]=b1000;
361         boardImages[10][1]=b1001;
362         boardImages[10][2]=b1002;
363         boardImages[10][3]=b1003;
364         boardImages[10][4]=b1004;
```

```
365         boardImages[10][5]=b1005;
366         boardImages[10][6]=b1006;
367         boardImages[10][7]=b1007;
368         boardImages[10][8]=b1008;
369         boardImages[10][9]=b1009;
370         boardImages[10][10]=b1010;
371         boardImages[10][11]=b1011;
372         boardImages[10][12]=b1012;
373         boardImages[10][13]=b1013;
374         boardImages[10][14]=b1014;
375
376         boardImages[11][0]=b1100;
377         boardImages[11][1]=b1101;
378         boardImages[11][2]=b1102;
379         boardImages[11][3]=b1103;
380         boardImages[11][4]=b1104;
381         boardImages[11][5]=b1105;
382         boardImages[11][6]=b1106;
383         boardImages[11][7]=b1107;
384         boardImages[11][8]=b1108;
385         boardImages[11][9]=b1109;
386         boardImages[11][10]=b1110;
387         boardImages[11][11]=b1111;
388         boardImages[11][12]=b1112;
389         boardImages[11][13]=b1113;
390         boardImages[11][14]=b1114;
391
392         boardImages[12][0]=b1200;
393         boardImages[12][1]=b1201;
394         boardImages[12][2]=b1202;
395         boardImages[12][3]=b1203;
396         boardImages[12][4]=b1204;
397         boardImages[12][5]=b1205;
398         boardImages[12][6]=b1206;
399         boardImages[12][7]=b1207;
400         boardImages[12][8]=b1208;
401         boardImages[12][9]=b1209;
402         boardImages[12][10]=b1210;
403         boardImages[12][11]=b1211;
404         boardImages[12][12]=b1212;
405         boardImages[12][13]=b1213;
```

```
406         boardImages[12][14]=b1214;
407
408         boardImages[13][0]=b1300;
409         boardImages[13][1]=b1301;
410         boardImages[13][2]=b1302;
411         boardImages[13][3]=b1303;
412         boardImages[13][4]=b1304;
413         boardImages[13][5]=b1305;
414         boardImages[13][6]=b1306;
415         boardImages[13][7]=b1307;
416         boardImages[13][8]=b1308;
417         boardImages[13][9]=b1309;
418         boardImages[13][10]=b1310;
419         boardImages[13][11]=b1311;
420         boardImages[13][12]=b1312;
421         boardImages[13][13]=b1313;
422         boardImages[13][14]=b1314;
423
424         boardImages[14][0]=b1400;
425         boardImages[14][1]=b1401;
426         boardImages[14][2]=b1402;
427         boardImages[14][3]=b1403;
428         boardImages[14][4]=b1404;
429         boardImages[14][5]=b1405;
430         boardImages[14][6]=b1406;
431         boardImages[14][7]=b1407;
432         boardImages[14][8]=b1408;
433         boardImages[14][9]=b1409;
434         boardImages[14][10]=b1410;
435         boardImages[14][11]=b1411;
436         boardImages[14][12]=b1412;
437         boardImages[14][13]=b1413;
438         boardImages[14][14]=b1414;
439     }
440
441     for (int i = 0; i <boardImages.length; i
442         ++
443     {
444         for (int j = 0; j < boardImages.length;
445             j++)
446             {
447                 boardImages[i][j].setImage(board[i][
448                     j].getImage());
```

```
444         }
445     }
446
447     for (BoardPiece piece: currentPlayer.get(0)
448         .getPieces()) {
449
450     }
451     ImageView firstClick;
452     ImageView secondClick;
453     int click1X,click2X,click1Y,click2Y;
454     @FXML
455     private void handleClickImage(MouseEvent event
456 ) {
457         System.out.println(event);
458         if(firstClick == null){
459             firstClick = (ImageView) (event.
460             getSource());
461             System.out.println(firstClick);
462             click1Y = GridPane.getRowIndex(
463             firstClick);
464             click1X = GridPane.getColumnIndex(
465             firstClick);
466             int imgClicked = GridPane.
467             getColumnIndex((ImageView) event.getSource());
468             System.out.println(imgClicked);
469
470             }else{
471                 secondClick = (ImageView) (event.
472             getSource());
473                 click2Y = GridPane.getRowIndex(
474             secondClick);
475                 click2X = GridPane.getColumnIndex(
476             secondClick);
477                 checkResult();
478                 firstClick =null;
479                 secondClick = null;
480             }
481
482     }
483
484     public void checkResult(){
```

```
476     BoardSquare first = board[click1X][click1Y];
477     BoardSquare second = board[click2X][click2Y];
478     if (first.getPower() * 2 > second.getPower)
479     {
480         first.changeLocation(click2X, click2Y);
481         board[click1X][click1Y] = new BoardSquare(
482             click1X, click1Y, boardSquareNames[8], nukePic, 2, 0);
483         board[click2X][click2Y] = first;
484         boardImages[click1X][click1Y].setImage(
485             nukePic);
486         boardImages[click2X][click2Y].setImage(
487             first.getImage());
488     }
489 }
490
491 }
492
```

```
1 package com.example.gridgamefinal;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Scene;
6 import javafx.stage.Stage;
7
8 import java.io.IOException;
9
10 public class HelloApplication extends Application {
11     @Override
12     public void start(Stage stage) throws IOException
13     {
14         FXMLLoader fxmlLoader = new FXMLLoader(
15             HelloApplication.class.getResource("hello-view.fxml"
16         ));
17         Scene scene = new Scene(fxmlLoader.load(),
18             1000, 1000);
19         stage.setTitle("Hello!");
20         stage.setScene(scene);
21         stage.show();
22     }
23 }
```