

```
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 void changeLocation(int droll, String[]
28        boardPieceNames, BoardSquare[][] board){
29        //find number of board piece
30
31        //        currentPieceNumber=0;
32        //        for (int i = 0; i < boardPieceNames.length
33        //            ; i++) {
34        //            if (currentBoardPiece.getName().equals(
35        //                boardPieceNames[i])) {
36        //                currentPieceNumber=i;
37        //            }
38        //        }
39        //        System.out.println("current piece # " +
40        //            currentPieceNumber);
```

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

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

```
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  2=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  2=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,
```



```
171 startingPieces));
172         currentPlayers.add(new Player("Player 2",
173             startingPieces));
174         for (int i = 0; i < 4; i++) {
175             currentPlayers.get(0).setOwned(board[0][
176                 i]);
177             }
178         }
179
180
181
182     }
183     @FXML
184     protected void handleStart(ActionEvent event){
185         boardImages[0][0]=b0000;
186         boardImages[0][1]=b0001;
187         boardImages[0][2]=b0002;
188         boardImages[0][3]=b0003;
189         boardImages[0][4]=b0004;
190         boardImages[0][5]=b0005;
191         boardImages[0][6]=b0006;
192         boardImages[0][7]=b0007;
193         boardImages[0][8]=b0008;
194         boardImages[0][9]=b0009;
195         boardImages[0][10]=b0010;
196         boardImages[0][11]=b0011;
197         boardImages[0][12]=b0012;
198         boardImages[0][13]=b0013;
199         boardImages[0][14]=b0014;
200
201         boardImages[1][0]=b0100;
202         boardImages[1][1]=b0101;
203         boardImages[1][2]=b0102;
204         boardImages[1][3]=b0103;
205         boardImages[1][4]=b0104;
206         boardImages[1][5]=b0105;
207         boardImages[1][6]=b0106;
208         boardImages[1][7]=b0107;
```

```
209         boardImages[1][8]=b0108;
210         boardImages[1][9]=b0109;
211         boardImages[1][10]=b0110;
212         boardImages[1][11]=b0111;
213         boardImages[1][12]=b0112;
214         boardImages[1][13]=b0113;
215         boardImages[1][14]=b0114;
216
217         boardImages[2][0]=b0200;
218         boardImages[2][1]=b0201;
219         boardImages[2][2]=b0202;
220         boardImages[2][3]=b0203;
221         boardImages[2][4]=b0204;
222         boardImages[2][5]=b0205;
223         boardImages[2][6]=b0206;
224         boardImages[2][7]=b0207;
225         boardImages[2][8]=b0208;
226         boardImages[2][9]=b0209;
227         boardImages[2][10]=b0210;
228         boardImages[2][11]=b0211;
229         boardImages[2][12]=b0212;
230         boardImages[2][13]=b0213;
231         boardImages[2][14]=b0214;
232
233         boardImages[3][0]=b0300;
234         boardImages[3][1]=b0301;
235         boardImages[3][2]=b0302;
236         boardImages[3][3]=b0303;
237         boardImages[3][4]=b0304;
238         boardImages[3][5]=b0305;
239         boardImages[3][6]=b0306;
240         boardImages[3][7]=b0307;
241         boardImages[3][8]=b0308;
242         boardImages[3][9]=b0309;
243         boardImages[3][10]=b0310;
244         boardImages[3][11]=b0311;
245         boardImages[3][12]=b0312;
246         boardImages[3][13]=b0313;
247         boardImages[3][14]=b0314;
248
249         boardImages[4][0]=b0400;
```

```
250         boardImages[4][1]=b0401;
251         boardImages[4][2]=b0402;
252         boardImages[4][3]=b0403;
253         boardImages[4][4]=b0404;
254         boardImages[4][5]=b0405;
255         boardImages[4][6]=b0406;
256         boardImages[4][7]=b0407;
257         boardImages[4][8]=b0408;
258         boardImages[4][9]=b0409;
259         boardImages[4][10]=b0410;
260         boardImages[4][11]=b0411;
261         boardImages[4][12]=b0412;
262         boardImages[4][13]=b0413;
263         boardImages[4][14]=b0414;
264
265         boardImages[5][0]=b0500;
266         boardImages[5][1]=b0501;
267         boardImages[5][2]=b0502;
268         boardImages[5][3]=b0503;
269         boardImages[5][4]=b0504;
270         boardImages[5][5]=b0505;
271         boardImages[5][6]=b0506;
272         boardImages[5][7]=b0507;
273         boardImages[5][8]=b0508;
274         boardImages[5][9]=b0509;
275         boardImages[5][10]=b0510;
276         boardImages[5][11]=b0511;
277         boardImages[5][12]=b0512;
278         boardImages[5][13]=b0513;
279         boardImages[5][14]=b0514;
280
281         boardImages[6][0]=b0600;
282         boardImages[6][1]=b0601;
283         boardImages[6][2]=b0602;
284         boardImages[6][3]=b0603;
285         boardImages[6][4]=b0604;
286         boardImages[6][5]=b0605;
287         boardImages[6][6]=b0606;
288         boardImages[6][7]=b0607;
289         boardImages[6][8]=b0608;
290         boardImages[6][9]=b0609;
```

```
291         boardImages[6][10]=b0610;
292         boardImages[6][11]=b0611;
293         boardImages[6][12]=b0612;
294         boardImages[6][13]=b0613;
295         boardImages[6][14]=b0614;
296
297         boardImages[7][0]=b0700;
298         boardImages[7][1]=b0701;
299         boardImages[7][2]=b0702;
300         boardImages[7][3]=b0703;
301         boardImages[7][4]=b0704;
302         boardImages[7][5]=b0705;
303         boardImages[7][6]=b0706;
304         boardImages[7][7]=b0707;
305         boardImages[7][8]=b0708;
306         boardImages[7][9]=b0709;
307         boardImages[7][10]=b0710;
308         boardImages[7][11]=b0711;
309         boardImages[7][12]=b0712;
310         boardImages[7][13]=b0713;
311         boardImages[7][14]=b0714;
312
313         boardImages[8][0]=b0800;
314         boardImages[8][1]=b0801;
315         boardImages[8][2]=b0802;
316         boardImages[8][3]=b0803;
317         boardImages[8][4]=b0804;
318         boardImages[8][5]=b0805;
319         boardImages[8][6]=b0806;
320         boardImages[8][7]=b0807;
321         boardImages[8][8]=b0808;
322         boardImages[8][9]=b0809;
323         boardImages[8][10]=b0810;
324         boardImages[8][11]=b0811;
325         boardImages[8][12]=b0812;
326         boardImages[8][13]=b0813;
327         boardImages[8][14]=b0814;
328
329         boardImages[9][0]=b0900;
330         boardImages[9][1]=b0901;
331         boardImages[9][2]=b0902;
```

```
332         boardImages[9][3]=b0903;
333         boardImages[9][4]=b0904;
334         boardImages[9][5]=b0905;
335         boardImages[9][6]=b0906;
336         boardImages[9][7]=b0907;
337         boardImages[9][8]=b0908;
338         boardImages[9][9]=b0909;
339         boardImages[9][10]=b0910;
340         boardImages[9][11]=b0911;
341         boardImages[9][12]=b0912;
342         boardImages[9][13]=b0913;
343         boardImages[9][14]=b0914;
344
345         boardImages[10][0]=b1000;
346         boardImages[10][1]=b1001;
347         boardImages[10][2]=b1002;
348         boardImages[10][3]=b1003;
349         boardImages[10][4]=b1004;
350         boardImages[10][5]=b1005;
351         boardImages[10][6]=b1006;
352         boardImages[10][7]=b1007;
353         boardImages[10][8]=b1008;
354         boardImages[10][9]=b1009;
355         boardImages[10][10]=b1010;
356         boardImages[10][11]=b1011;
357         boardImages[10][12]=b1012;
358         boardImages[10][13]=b1013;
359         boardImages[10][14]=b1014;
360
361         boardImages[11][0]=b1100;
362         boardImages[11][1]=b1101;
363         boardImages[11][2]=b1102;
364         boardImages[11][3]=b1103;
365         boardImages[11][4]=b1104;
366         boardImages[11][5]=b1105;
367         boardImages[11][6]=b1106;
368         boardImages[11][7]=b1107;
369         boardImages[11][8]=b1108;
370         boardImages[11][9]=b1109;
371         boardImages[11][10]=b1110;
372         boardImages[11][11]=b1111;
```

```
373         boardImages[11][12]=b1112;
374         boardImages[11][13]=b1113;
375         boardImages[11][14]=b1114;
376
377         boardImages[12][0]=b1200;
378         boardImages[12][1]=b1201;
379         boardImages[12][2]=b1202;
380         boardImages[12][3]=b1203;
381         boardImages[12][4]=b1204;
382         boardImages[12][5]=b1205;
383         boardImages[12][6]=b1206;
384         boardImages[12][7]=b1207;
385         boardImages[12][8]=b1208;
386         boardImages[12][9]=b1209;
387         boardImages[12][10]=b1210;
388         boardImages[12][11]=b1211;
389         boardImages[12][12]=b1212;
390         boardImages[12][13]=b1213;
391         boardImages[12][14]=b1214;
392
393         boardImages[13][0]=b1300;
394         boardImages[13][1]=b1301;
395         boardImages[13][2]=b1302;
396         boardImages[13][3]=b1303;
397         boardImages[13][4]=b1304;
398         boardImages[13][5]=b1305;
399         boardImages[13][6]=b1306;
400         boardImages[13][7]=b1307;
401         boardImages[13][8]=b1308;
402         boardImages[13][9]=b1309;
403         boardImages[13][10]=b1310;
404         boardImages[13][11]=b1311;
405         boardImages[13][12]=b1312;
406         boardImages[13][13]=b1313;
407         boardImages[13][14]=b1314;
408
409         boardImages[14][0]=b1400;
410         boardImages[14][1]=b1401;
411         boardImages[14][2]=b1402;
412         boardImages[14][3]=b1403;
413         boardImages[14][4]=b1404;
```

```
414         boardImages[14][5]=b1405;
415         boardImages[14][6]=b1406;
416         boardImages[14][7]=b1407;
417         boardImages[14][8]=b1408;
418         boardImages[14][9]=b1409;
419         boardImages[14][10]=b1410;
420         boardImages[14][11]=b1411;
421         boardImages[14][12]=b1412;
422         boardImages[14][13]=b1413;
423         boardImages[14][14]=b1414;
424
425     for (int i = 0; i <boardImages.length; i
426 ++
427     for (int j = 0; j < boardImages.length;
428 j++) {
429         System.out.println(i + ", " + j);
430         boardImages[i][j].setImage(board[i][
431             j].getImage());
432     }
433     ImageView firstClick;
434     ImageView secondClick;
435     int click1X,click2X,click1Y,click2Y;
436     @FXML
437     private void handleClickImage(MouseEvent event
438 ) {
439         System.out.println(event);
440         if(firstClick == null){
441 //           click1X = GridPane.getRowIndex(((
442 //               ImageView) event.getSource()));
443         firstClick = (ImageView) (event.
444             getSource());
445 //           System.out.println(firstClick);
446         click1X = GridPane.getRowIndex(
447             firstClick);
448         click1Y = GridPane.getColumnIndex(
449             firstClick);
450 //           int imgClicked = GridPane.
451             getColumnIndex((ImageView) event.getSource());
```

```
446 //           System.out.println(imgClicked);
447
448     }else{
449         secondClick = (ImageView) (event.
450             getSource());
450         click2X = GridPane.getRowIndex(
451             secondClick);
451         click2Y = GridPane.getColumnIndex(
452             secondClick);
452         checkResult();
453         firstClick =null;
454         secondClick = null;
455     }
456
457 }
458 public void checkResult(){
459     BoardSquare first = board[click1X][click1Y];
460     BoardSquare second =board[click2X][click2Y];
461     if (first.getPower()>second.getPower()){
462         first.changeLocation(click2X,click2Y);
463         board[click1X][click1Y]=new BoardSquare(
463             click1X,click1Y,boardSquareNames[8],red,2,0);
464         board[click2X][click2Y] = first;
465         boardImages[click1X][click1Y].setImage(
465             red);
466         boardImages[click2X][click2Y].setImage(
466             first.getImage());
467     }else{
468         currentPlayers.get(0).removeFromOwned(
468             first);
469         boardImages[click1X][click1Y].setImage(
469             red);
470         board[click1X][click1Y]=new BoardSquare(
470             click1X,click1Y,boardSquareNames[8],red,2,0);
471     }
472 }
473
474 }
475
```

```
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         320, 240);
19         stage.setTitle("Hello!");
20         stage.setScene(scene);
21         stage.show();
22     }
23 }
```