```java
package com.example.gridgamefinal;

import java.util.ArrayList;

public class Player {
    private String name;
    private ArrayList<BoardSquare> owned = new ArrayList<>();

    private ArrayList<BoardPiece> pieces = new ArrayList<>();

    public Player(String n, ArrayList<BoardPiece> p){
        name = n;
        for (BoardPiece piece: p) {
            pieces.add(piece);
        }
    }
    public void removeFromOwned(BoardSquare r){
        owned.remove(r);
    }
    public void setOwned(BoardSquare b) {
        owned.add(b);
    }

    public String getName() {
        return name;
    }

    public ArrayList<BoardPiece> getPieces() {
        return pieces;
    }

    public ArrayList<BoardSquare> getOwned() {
        return owned;
    }

    public void removePieces(BoardPiece piece) {
        pieces.remove(piece);
    }
}
```

```java
package com.example.gridgamefinal;

public class Location {
    private int row;
    private int column;

    public Location(int r, int c){
        row = r;
        column = c;
    }
    public int getRow() {
        return row;
    }

    public int getColumn() {
        return column;
    }

    public void setColumn(int column) {
        this.column = column;
    }

    public void setRow(int row) {
        this.row = row;
    }

    // done
}

```

```java
package com.example.gridgamefinal;

import javafx.scene.image.Image;

public class BoardPiece {
    private String name;
    private Image image;

    private int health;

    private int power;
    public BoardPiece(String n, Image img){
        name = n;
        image = img;
    }


    public Image getImage() {
        return image;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public int getHealth() {
        return health;
    }

    public int getPower() {
        return power;
    }
}

```

```java
package com.example.gridgamefinal;

import javafx.scene.image.Image;

import java.util.ArrayList;

public class BoardSquare {
    private String name;
    private Location loc;
    private Image image;

    private int health;
    private int owner;

    private BoardPiece piece;

    private ArrayList<BoardPiece> troops = new ArrayList<>();
    private Image image2;

    public BoardSquare(int row, int column, String n, Image img,int own,int h){
        name = n;
        loc = new Location(row,column);
        image = img;
        //0,1,2  0=unowned
        owner = own;
        health = h;
    }

    public BoardSquare(int row, int column, String n, Image img,int own,BoardPiece p){
        name = n;
        loc = new Location(row,column);
        image = img;
        //0,1,2  0=unowned
        owner = own;
        piece = p;
        health = p.getHealth();
        image2 = p.getImage();
    }
```

```
39
40      public int getHealth() {
41          return health;
42      }
43
44      public void changeHealth(int health) {
45          this.health += health;
46      }
47
48      public void changeLocation(int r, int c){
49          loc.setRow(r);
50          loc.setColumn(c);
51      }
52
53      public Image getImage() {
54          return image;
55      }
56
57      public void setName(String name) {
58          this.name = name;
59      }
60
61      public int getOwner() {
62          return owner;
63      }
64
65      public BoardPiece getPiece() {
66          return piece;
67      }
68
69      public Image getImage2() {
70          return image2;
71      }
72
73      public void setImage(Image image) {
74          this.image = image;
75      }
76
77      public String getName() {
78          return name;
79      }
```

```
 80
 81      public Location getLoc() {
 82          return loc;
 83      }
 84      public int getRowLoc(){
 85          return loc.getRow();
 86      }
 87      public  int getColLoc(){
 88          return loc.getColumn();
 89      }
 90
 91      public void setPiece(BoardPiece piece) {
 92          this.piece = piece;
 93      }
 94
 95      public ArrayList<BoardPiece> getTroops() {
 96          return troops;
 97      }
 98
 99      public void setTroops(BoardPiece troop) {
100          this.troops.add(troop);
101      }
102 }
103
```

```java
package com.example.gridgamefinal;

import javafx.application.Platform;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.ListView;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.GridPane;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Timer;
import java.util.TimerTask;

public class HelloController {

    public ListView piecesView;
    @FXML
    GridPane gpane;
    private ArrayList<Player> currentPlayers = new ArrayList<>();
    private BoardSquare[][] board = new BoardSquare[15][15];
    private ImageView[][] boardImages = new ImageView[15][15];

    private BoardPiece airbase,fighter,bomber,silo,nuke,radar,battleship,carrier,sub;

    //startingPieces dont include nukes
    private ArrayList<BoardPiece> startingPieces = new ArrayList<BoardPiece>();

    private BoardPiece selectedPiece = null;

    @FXML
    private ImageView b0000, b0001, b0002, b0003, b0004, b0005, b0006, b0007, b0008, b0009, b0010,
```

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

```
43      TimerTask myTimerTask = new TimerTask(){
44          @Override
45          public void run() {
46              if (!finished) {
47                  Platform.runLater(new Runnable() {
48                      @Override public void run() {
49                          System.out.println("Time: "
    + time + " seconds");
50                      }
51                  });
52  //                System.out.println("Time: " + time
    + " seconds");
53                  time--;
54              }
55          }
56      };
57
58      int time = 300;
59
60      int timerTimes = 0;
61
62      public HelloController(){
63
64          FileInputStream redd,orangee,purplee,dbluee,
   lbluee,airbasee,fighterr,bomberr,siloo,nukee,radarr,
   battleshipp,carrierr,subb;
65          try {
66              redd = new FileInputStream("src/main/
   Pictures/red.png");
67              red = new Image(redd);
68              orangee= new FileInputStream("src/main/
   Pictures/orange.png");
69              orange = new Image(orangee);
70              purplee = new FileInputStream("src/main/
   Pictures/purple.jpeg");
71              purple = new Image(purplee);
72              dbluee = new FileInputStream("src/main/
   Pictures/dark-blue.jpeg");
73              dblue = new Image(dbluee);
74              lbluee = new FileInputStream("src/main/
   Pictures/ocean-blue.jpeg");
```

```
 75              lblue = new Image(lbluee);
 76
 77
 78              airbasee = new FileInputStream("src/main
    /Pictures/Airbase.png");
 79              airbasePic = new Image(airbasee);
 80              fighterr = new FileInputStream("src/main
    /Pictures/Fighter.png");
 81              fighterPic = new Image(fighterr);
 82              bomberr = new FileInputStream("src/main/
    Pictures/Bomber.png");
 83              bomberPic = new Image(bomberr);
 84
 85              siloo = new FileInputStream("src/main/
    Pictures/Sam.png");
 86              siloPic = new Image(siloo);
 87              nukee = new FileInputStream("src/main/
    Pictures/Nuke.png");
 88              nukePic = new Image(nukee);
 89              radarr = new FileInputStream("src/main/
    Pictures/Radar.png");
 90              radarPic = new Image(radarr);
 91
 92              battleshipp = new FileInputStream("src/
    main/Pictures/Battleship.png");
 93              battleshipPic = new Image(battleshipp);
 94              carrierr = new FileInputStream("src/main
    /Pictures/Carrier.png");
 95              carrierPic = new Image(carrierr);
 96              subb = new FileInputStream("src/main/
    Pictures/Sub.png");
 97              subPic = new Image(subb);
 98          } catch (FileNotFoundException e) {
 99              e.printStackTrace();
100          }
101
102          airbase = new BoardPiece("airbase",
    airbasePic);
103          fighter = new BoardPiece("fighter",
    fighterPic);
104          bomber = new BoardPiece("bomber", bomberPic
```

```
104 );
105         silo = new BoardPiece("silo", siloPic);
106         nuke = new BoardPiece("nuke", nukePic);
107         radar = new BoardPiece("radar", radarPic);
108         battleship = new BoardPiece("battleship",
    battleshipPic);
109         carrier = new BoardPiece("carrier",
    carrierPic);
110         sub = new BoardPiece("sub", subPic);
111 
112         startingPieces.add(airbase);
113         startingPieces.add(airbase);
114         startingPieces.add(airbase);
115         startingPieces.add(fighter);
116         startingPieces.add(fighter);
117         startingPieces.add(fighter);
118         startingPieces.add(fighter);
119         startingPieces.add(fighter);
120         startingPieces.add(fighter);
121         startingPieces.add(bomber);
122         startingPieces.add(bomber);
123         startingPieces.add(bomber);
124         startingPieces.add(silo);
125         startingPieces.add(silo);
126         startingPieces.add(silo);
127         startingPieces.add(silo);
128         startingPieces.add(silo);
129         startingPieces.add(silo);
130         startingPieces.add(silo);
131         startingPieces.add(radar);
132         startingPieces.add(radar);
133         startingPieces.add(radar);
134         startingPieces.add(radar);
135         startingPieces.add(radar);
136         startingPieces.add(radar);
137         startingPieces.add(radar);
138         startingPieces.add(radar);
139         startingPieces.add(battleship);
140         startingPieces.add(battleship);
141         startingPieces.add(battleship);
142         startingPieces.add(battleship);
```

```
143         startingPieces.add(battleship);
144         startingPieces.add(battleship);
145         startingPieces.add(carrier);
146         startingPieces.add(carrier);
147         startingPieces.add(carrier);
148         startingPieces.add(carrier);
149         startingPieces.add(carrier);
150         startingPieces.add(sub);
151         startingPieces.add(sub);
152         startingPieces.add(sub);
153         startingPieces.add(sub);
154         startingPieces.add(sub);
155
156
157         for (int i = 0; i < 5; i++) {
158             for (int j = 0; j < 5; j++) {
159                 board[i][j] = new BoardSquare(i,j,
    boardSquareNames[0],red,1,-1);
160             }
161         }
162         board[0][5] = new BoardSquare(0,5,
    boardSquareNames[0],red,1,-1);
163         board[0][6] = new BoardSquare(0,6,
    boardSquareNames[0],red,1,-1);
164         board[1][5] = new BoardSquare(1,5,
    boardSquareNames[0],red,1,-1);
165         board[1][6] = new BoardSquare(1,6,
    boardSquareNames[0],red,1,-1);
166         board[2][5] = new BoardSquare(2,5,
    boardSquareNames[0],red,1,-1);
167
168         board[5][0] = new BoardSquare(5,0,
    boardSquareNames[0],red,1,-1);
169         board[5][1] = new BoardSquare(5,1,
    boardSquareNames[0],red,1,-1);
170         board[5][2] = new BoardSquare(5,2,
    boardSquareNames[0],red,1,-1);
171         board[6][0] = new BoardSquare(6,0,
    boardSquareNames[0],red,1,-1);
172         board[6][1] = new BoardSquare(6,1,
    boardSquareNames[0],red,1,-1);
```

```
173
174
175         for (int i = 0; i < 5; i++) {
176             for (int j = 0; j < 5; j++) {
177                 board[i][14-j] = new BoardSquare(i,
    14-j,boardSquareNames[1],orange,2,-1);
178             }
179         }
180         board[2][9] = new BoardSquare(2,9,
    boardSquareNames[1],orange,2,-1);
181         board[3][8] = new BoardSquare(3,8,
    boardSquareNames[1],orange,2,-1);
182         board[3][9] = new BoardSquare(3,9,
    boardSquareNames[1],orange,2,-1);
183         board[4][8] = new BoardSquare(4,8,
    boardSquareNames[1],orange,2,-1);
184         board[4][9] = new BoardSquare(4,9,
    boardSquareNames[1],orange,2,-1);
185
186         board[5][11] = new BoardSquare(5,11,
    boardSquareNames[1],orange,2,-1);
187         board[5][12] = new BoardSquare(5,12,
    boardSquareNames[1],orange,2,-1);
188         board[5][13] = new BoardSquare(5,13,
    boardSquareNames[1],orange,2,-1);
189         board[5][14] = new BoardSquare(5,14,
    boardSquareNames[1],orange,2,-1);
190         board[6][14] = new BoardSquare(6,14,
    boardSquareNames[1],orange,2,-1);
191
192         for (int i = 0; i < 5; i++) {
193             for (int j = 0; j < 5; j++) {
194                 board[14-i][j] = new BoardSquare(14-
    i,j,boardSquareNames[2],dblue,3,-1);
195             }
196         }
197         board[8][4] = new BoardSquare(8,4,
    boardSquareNames[2],dblue,3,-1);
198         board[9][3] = new BoardSquare(9,3,
    boardSquareNames[2],dblue,3,-1);
199         board[9][4] = new BoardSquare(9,4,
```

```
199 boardSquareNames[2],dblue,3,-1);
200         board[9][5] = new BoardSquare(9,5,
    boardSquareNames[2],dblue,3,-1);
201         board[10][5] = new BoardSquare(10,5,
    boardSquareNames[2],dblue,3,-1);
202 
203         board[12][6] = new BoardSquare(12,6,
    boardSquareNames[2],dblue,3,-1);
204         board[13][5] = new BoardSquare(13,5,
    boardSquareNames[2],dblue,3,-1);
205         board[13][6] = new BoardSquare(13,6,
    boardSquareNames[2],dblue,3,-1);
206         board[14][5] = new BoardSquare(14,5,
    boardSquareNames[2],dblue,3,-1);
207         board[14][6] = new BoardSquare(14,6,
    boardSquareNames[2],dblue,3,-1);
208 
209         for (int i = 0; i < 5; i++) {
210             for (int j = 0; j < 5; j++) {
211                 board[14-i][14-j] = new BoardSquare(
    14-i,14-j,boardSquareNames[3],purple,4,-1);
212             }
213         }
214         board[8][12] = new BoardSquare(8,12,
    boardSquareNames[3],purple,4,-1);
215         board[9][9] = new BoardSquare(9,9,
    boardSquareNames[3],purple,4,-1);
216         board[9][12] = new BoardSquare(9,12,
    boardSquareNames[3],purple,4,-1);
217         board[9][13] = new BoardSquare(9,13,
    boardSquareNames[3],purple,4,-1);
218         board[9][14] = new BoardSquare(9,14,
    boardSquareNames[3],purple,4,-1);
219 
220         board[10][8] = new BoardSquare(10,8,
    boardSquareNames[3],purple,4,-1);
221         board[10][9] = new BoardSquare(10,9,
    boardSquareNames[3],purple,4,-1);
222         board[11][8] = new BoardSquare(11,8,
    boardSquareNames[3],purple,4,-1);
223         board[11][9] = new BoardSquare(11,9,
```

```
223 boardSquareNames[3],purple,4,-1);
224         board[12][9] = new BoardSquare(12,9,
    boardSquareNames[3],purple,4,-1);
225         //board pieces
226 
227         for (int i = 0; i < 15; i++) {
228             for (int j = 0; j < 15; j++) {
229                 if (board[i][j] == null) {
230                     board[i][j] = new BoardSquare(i,
    j,boardSquareNames[4],lblue,0,-1);
231                 }
232 
233             }
234         }
235         currentPlayers.add(new Player("Player 1",
    startingPieces));
236         currentPlayers.add(new Player("Player 2",
    startingPieces));
237         currentPlayers.add(new Player("Player 3",
    startingPieces));
238         currentPlayers.add(new Player("Player 4",
    startingPieces));
239         for (int i = 0; i < 15; i++) {
240             for (int j = 0; j < 15; j++) {
241                 if (board[i][j].getImage() == red
    )  {
242                     currentPlayers.get(0).setOwned(
    board[i][j]);
243                 }
244                 if (board[i][j].getImage() == dblue
    )  {
245                     currentPlayers.get(1).setOwned(
    board[i][j]);
246                 }
247                 if (board[i][j].getImage() == purple
    )  {
248                     currentPlayers.get(2).setOwned(
    board[i][j]);
249                 }
250                 if (board[i][j].getImage() == orange
    )  {
```

```
251                     currentPlayers.get(3).setOwned(
    board[i][j]);
252                 }
253             }
254         }
255     }
256
257
258     private void startTimer() {
259         timerTimes++;
260         myTimer.scheduleAtFixedRate(myTimerTask, 0,
    1000 * timerTimes);
261     }
262
263
264     @FXML
265     protected void handleStart(ActionEvent event){
266
267         if (time == -1){
268             startTimer();
269         } else{
270             time = 120;
271         }
272
273         finished = false;
274
275         boolean idk = true;
276
277         if (idk){
278             boardImages[0][0]=b0000;
279             boardImages[0][1]=b0001;
280             boardImages[0][2]=b0002;
281             boardImages[0][3]=b0003;
282             boardImages[0][4]=b0004;
283             boardImages[0][5]=b0005;
284             boardImages[0][6]=b0006;
285             boardImages[0][7]=b0007;
286             boardImages[0][8]=b0008;
287             boardImages[0][9]=b0009;
288             boardImages[0][10]=b0010;
289             boardImages[0][11]=b0011;
```

```
290             boardImages[0][12]=b0012;
291             boardImages[0][13]=b0013;
292             boardImages[0][14]=b0014;
293
294             boardImages[1][0]=b0100;
295             boardImages[1][1]=b0101;
296             boardImages[1][2]=b0102;
297             boardImages[1][3]=b0103;
298             boardImages[1][4]=b0104;
299             boardImages[1][5]=b0105;
300             boardImages[1][6]=b0106;
301             boardImages[1][7]=b0107;
302             boardImages[1][8]=b0108;
303             boardImages[1][9]=b0109;
304             boardImages[1][10]=b0110;
305             boardImages[1][11]=b0111;
306             boardImages[1][12]=b0112;
307             boardImages[1][13]=b0113;
308             boardImages[1][14]=b0114;
309
310             boardImages[2][0]=b0200;
311             boardImages[2][1]=b0201;
312             boardImages[2][2]=b0202;
313             boardImages[2][3]=b0203;
314             boardImages[2][4]=b0204;
315             boardImages[2][5]=b0205;
316             boardImages[2][6]=b0206;
317             boardImages[2][7]=b0207;
318             boardImages[2][8]=b0208;
319             boardImages[2][9]=b0209;
320             boardImages[2][10]=b0210;
321             boardImages[2][11]=b0211;
322             boardImages[2][12]=b0212;
323             boardImages[2][13]=b0213;
324             boardImages[2][14]=b0214;
325
326             boardImages[3][0]=b0300;
327             boardImages[3][1]=b0301;
328             boardImages[3][2]=b0302;
329             boardImages[3][3]=b0303;
330             boardImages[3][4]=b0304;
```

```
331             boardImages[3][5]=b0305;
332             boardImages[3][6]=b0306;
333             boardImages[3][7]=b0307;
334             boardImages[3][8]=b0308;
335             boardImages[3][9]=b0309;
336             boardImages[3][10]=b0310;
337             boardImages[3][11]=b0311;
338             boardImages[3][12]=b0312;
339             boardImages[3][13]=b0313;
340             boardImages[3][14]=b0314;
341
342             boardImages[4][0]=b0400;
343             boardImages[4][1]=b0401;
344             boardImages[4][2]=b0402;
345             boardImages[4][3]=b0403;
346             boardImages[4][4]=b0404;
347             boardImages[4][5]=b0405;
348             boardImages[4][6]=b0406;
349             boardImages[4][7]=b0407;
350             boardImages[4][8]=b0408;
351             boardImages[4][9]=b0409;
352             boardImages[4][10]=b0410;
353             boardImages[4][11]=b0411;
354             boardImages[4][12]=b0412;
355             boardImages[4][13]=b0413;
356             boardImages[4][14]=b0414;
357
358             boardImages[5][0]=b0500;
359             boardImages[5][1]=b0501;
360             boardImages[5][2]=b0502;
361             boardImages[5][3]=b0503;
362             boardImages[5][4]=b0504;
363             boardImages[5][5]=b0505;
364             boardImages[5][6]=b0506;
365             boardImages[5][7]=b0507;
366             boardImages[5][8]=b0508;
367             boardImages[5][9]=b0509;
368             boardImages[5][10]=b0510;
369             boardImages[5][11]=b0511;
370             boardImages[5][12]=b0512;
371             boardImages[5][13]=b0513;
```

```
372             boardImages[5][14]=b0514;
373
374             boardImages[6][0]=b0600;
375             boardImages[6][1]=b0601;
376             boardImages[6][2]=b0602;
377             boardImages[6][3]=b0603;
378             boardImages[6][4]=b0604;
379             boardImages[6][5]=b0605;
380             boardImages[6][6]=b0606;
381             boardImages[6][7]=b0607;
382             boardImages[6][8]=b0608;
383             boardImages[6][9]=b0609;
384             boardImages[6][10]=b0610;
385             boardImages[6][11]=b0611;
386             boardImages[6][12]=b0612;
387             boardImages[6][13]=b0613;
388             boardImages[6][14]=b0614;
389
390             boardImages[7][0]=b0700;
391             boardImages[7][1]=b0701;
392             boardImages[7][2]=b0702;
393             boardImages[7][3]=b0703;
394             boardImages[7][4]=b0704;
395             boardImages[7][5]=b0705;
396             boardImages[7][6]=b0706;
397             boardImages[7][7]=b0707;
398             boardImages[7][8]=b0708;
399             boardImages[7][9]=b0709;
400             boardImages[7][10]=b0710;
401             boardImages[7][11]=b0711;
402             boardImages[7][12]=b0712;
403             boardImages[7][13]=b0713;
404             boardImages[7][14]=b0714;
405
406             boardImages[8][0]=b0800;
407             boardImages[8][1]=b0801;
408             boardImages[8][2]=b0802;
409             boardImages[8][3]=b0803;
410             boardImages[8][4]=b0804;
411             boardImages[8][5]=b0805;
412             boardImages[8][6]=b0806;
```

```
413                 boardImages[8][7]=b0807;
414                 boardImages[8][8]=b0808;
415                 boardImages[8][9]=b0809;
416                 boardImages[8][10]=b0810;
417                 boardImages[8][11]=b0811;
418                 boardImages[8][12]=b0812;
419                 boardImages[8][13]=b0813;
420                 boardImages[8][14]=b0814;
421
422                 boardImages[9][0]=b0900;
423                 boardImages[9][1]=b0901;
424                 boardImages[9][2]=b0902;
425                 boardImages[9][3]=b0903;
426                 boardImages[9][4]=b0904;
427                 boardImages[9][5]=b0905;
428                 boardImages[9][6]=b0906;
429                 boardImages[9][7]=b0907;
430                 boardImages[9][8]=b0908;
431                 boardImages[9][9]=b0909;
432                 boardImages[9][10]=b0910;
433                 boardImages[9][11]=b0911;
434                 boardImages[9][12]=b0912;
435                 boardImages[9][13]=b0913;
436                 boardImages[9][14]=b0914;
437
438                 boardImages[10][0]=b1000;
439                 boardImages[10][1]=b1001;
440                 boardImages[10][2]=b1002;
441                 boardImages[10][3]=b1003;
442                 boardImages[10][4]=b1004;
443                 boardImages[10][5]=b1005;
444                 boardImages[10][6]=b1006;
445                 boardImages[10][7]=b1007;
446                 boardImages[10][8]=b1008;
447                 boardImages[10][9]=b1009;
448                 boardImages[10][10]=b1010;
449                 boardImages[10][11]=b1011;
450                 boardImages[10][12]=b1012;
451                 boardImages[10][13]=b1013;
452                 boardImages[10][14]=b1014;
453
```

```
454                 boardImages[11][0]=b1100;
455                 boardImages[11][1]=b1101;
456                 boardImages[11][2]=b1102;
457                 boardImages[11][3]=b1103;
458                 boardImages[11][4]=b1104;
459                 boardImages[11][5]=b1105;
460                 boardImages[11][6]=b1106;
461                 boardImages[11][7]=b1107;
462                 boardImages[11][8]=b1108;
463                 boardImages[11][9]=b1109;
464                 boardImages[11][10]=b1110;
465                 boardImages[11][11]=b1111;
466                 boardImages[11][12]=b1112;
467                 boardImages[11][13]=b1113;
468                 boardImages[11][14]=b1114;
469
470                 boardImages[12][0]=b1200;
471                 boardImages[12][1]=b1201;
472                 boardImages[12][2]=b1202;
473                 boardImages[12][3]=b1203;
474                 boardImages[12][4]=b1204;
475                 boardImages[12][5]=b1205;
476                 boardImages[12][6]=b1206;
477                 boardImages[12][7]=b1207;
478                 boardImages[12][8]=b1208;
479                 boardImages[12][9]=b1209;
480                 boardImages[12][10]=b1210;
481                 boardImages[12][11]=b1211;
482                 boardImages[12][12]=b1212;
483                 boardImages[12][13]=b1213;
484                 boardImages[12][14]=b1214;
485
486                 boardImages[13][0]=b1300;
487                 boardImages[13][1]=b1301;
488                 boardImages[13][2]=b1302;
489                 boardImages[13][3]=b1303;
490                 boardImages[13][4]=b1304;
491                 boardImages[13][5]=b1305;
492                 boardImages[13][6]=b1306;
493                 boardImages[13][7]=b1307;
494                 boardImages[13][8]=b1308;
```

```
495             boardImages[13][9]=b1309;
496             boardImages[13][10]=b1310;
497             boardImages[13][11]=b1311;
498             boardImages[13][12]=b1312;
499             boardImages[13][13]=b1313;
500             boardImages[13][14]=b1314;
501
502             boardImages[14][0]=b1400;
503             boardImages[14][1]=b1401;
504             boardImages[14][2]=b1402;
505             boardImages[14][3]=b1403;
506             boardImages[14][4]=b1404;
507             boardImages[14][5]=b1405;
508             boardImages[14][6]=b1406;
509             boardImages[14][7]=b1407;
510             boardImages[14][8]=b1408;
511             boardImages[14][9]=b1409;
512             boardImages[14][10]=b1410;
513             boardImages[14][11]=b1411;
514             boardImages[14][12]=b1412;
515             boardImages[14][13]=b1413;
516             boardImages[14][14]=b1414;
517         }
518
519         refreshGridPane();
520     }
521     ImageView firstClick;
522     ImageView secondClick;
523     int click1X,click2X,click1Y,click2Y;
524     @FXML
525     private void handleClickImage(MouseEvent event
    ) {
526         System.out.println(event);
527         if (selectedPiece != null){
528             firstClick = (ImageView) (event.
    getSource());
529             click1Y = GridPane.getRowIndex(
    firstClick);
530             click1X = GridPane.getColumnIndex(
    firstClick);
531             checkResult();
```

```
532             selectedPiece = null;
533             firstClick = null;
534         } else {
535             if (firstClick == null) {
536                 firstClick = (ImageView) (event.
    getSource());
537                 click1Y = GridPane.getRowIndex(
    firstClick);
538                 click1X = GridPane.getColumnIndex(
    firstClick);
539 
540             } else {
541                 secondClick = (ImageView) (event.
    getSource());
542                 click2Y = GridPane.getRowIndex(
    secondClick);
543                 click2X = GridPane.getColumnIndex(
    secondClick);
544                 checkResult();
545                 firstClick = null;
546                 secondClick = null;
547             }
548         }
549     }
550     public void checkResult(){
551         if (selectedPiece != null){
552             if (selectedPiece.getName().equals("
    fighter") || selectedPiece.getName().equals("fighter
    ")){
553                 if (board[click1X][click1Y].getPiece
    ().getName().equals("airbase")){
554                     board[click1X][click1Y].
    setTroops(selectedPiece);
555                     currentPlayers.get(0).
    removePieces(selectedPiece);
556                 } else {
557                     System.out.println("Invalid
    placement");
558                 }
559             }
560             else if (selectedPiece.getName().equals(
```

```
560 "battleship") || selectedPiece.getName().equals("
    carrier") || selectedPiece.getName().equals("sub")){
561                 if (board[click1X][click1Y].getImage
    () == lblue){
562                     board[click1X][click1Y] = new
    BoardSquare(click1X, click1Y, boardSquareNames[0],
    board[click1X][click1Y].getImage(), 1, selectedPiece
    );
563                     currentPlayers.get(0).
    removePieces(selectedPiece);
564                 } else {
565                     System.out.println("Invalid
    placement");
566                 }
567             } else if (board[click1X][click1Y].
    getImage() != lblue) {
568                     board[click1X][click1Y] = new
    BoardSquare(click1X, click1Y, boardSquareNames[0],
    board[click1X][click1Y].getImage(), 1, selectedPiece
    );
569                     currentPlayers.get(0).
    removePieces(selectedPiece);
570             } else {
571                 System.out.println("Invalid
    placement");
572             }
573         } else {
574
575             BoardSquare first = board[click1X][
    click1Y];
576             BoardSquare second = board[click2X][
    click2Y];
577             if (second.getHealth() == -1) {
578                 if (first.getHealth() != -1){
579                     if (first.getPiece().getName().
    equals("fighter") || first.getPiece().getName().
    equals("bomber") || first.getPiece().getName().
    equals("silo") || first.getPiece().getName().equals(
    "battleship") || first.getPiece().getName().equals("
    carrier") || first.getPiece().getName().equals("sub"
    )) {
```

```
580                         if (first.getPiece().getName
    ().equals("battleship") || first.getPiece().getName
    ().equals("carrier") || first.getPiece().getName().
    equals("sub")) {
581                             if (board[click2X][
    click2Y].getImage() == lblue) {
582                                 first.changeLocation
    (click2X, click2Y);
583                                 board[click2X][
    click2Y] = new BoardSquare(click2X, click2Y, board[
    click2X][click2Y].getName(), board[click2X][click2Y
    ].getImage(), board[click1X][click1Y].getOwner(),
    board[click1X][click1Y].getPiece());
584                                 board[click1X][
    click1Y] = new BoardSquare(click1X, click1Y, board[
    click1X][click1Y].getName(), board[click1X][click1Y
    ].getImage(), board[click1X][click1Y].getOwner(), -1
    );
585                             } else {
586                                 System.out.println("
    Invalid placement");
587                             }
588                         } else {
589                             first.changeLocation(
    click2X, click2Y);
590                             board[click2X][click2Y
    ] = new BoardSquare(click2X, click2Y, board[click2X
    ][click2Y].getName(), board[click2X][click2Y].
    getImage(), board[click1X][click1Y].getOwner(),
    board[click1X][click1Y].getPiece());
591                             board[click1X][click1Y
    ] = new BoardSquare(click1X, click1Y, board[click1X
    ][click1Y].getName(), board[click1X][click1Y].
    getImage(), board[click1X][click1Y].getOwner(), -1);
592                         }
593                         if (first.getPiece().getName
    ().equals("silo")){
594
595                         }
596                     } else {
597                         System.out.println("You can
```

```
597 only move planes or ships and shoot missiles from
    silos");
598                     }
599                 } else {
600                     System.out.println("You can only
     move planes or ships and shoot missiles from silos"
    );
601                 }
602             } else {
603                 if (Math.abs(click1X-click2X) == 1
     || Math.abs(click1Y-click2Y) == 1){
604                     attack(first,second);
605                 }
606                 currentPlayers.get(0).
    removeFromOwned(first);
607                 board[click1X][click1Y] = new
    BoardSquare(click1X, click1Y, boardSquareNames[4],
    second.getImage(), 3, -1);
608             }
609         }
610         refreshGridPane();
611     }
612
613     private void attack(BoardSquare first,
    BoardSquare second) {
614
615         second.changeHealth((int) (Math.random()*20
    ) * first.getPiece().getPower());
616         if (second.getHealth()<= 0){
617             second.setImage(second.getImage());
618         }
619     }
620
621     public void handlePlaceImage() {
622         int index = piecesView.getSelectionModel().
    getSelectedIndex();
623
624         selectedPiece = currentPlayers.get(0).
    getPieces().get(index);
625         System.out.println(selectedPiece.getName());
626     }
```

```
627
628     public void refreshGridPane() {
629         for (int i = 0; i < 15; i++) {
630             for (int j = 0; j < 15; j++) {
631                 if (board[i][j].getPiece() != null){
632                     boardImages[i][j].setImage(board
    [i][j].getImage2());
633                 } else {
634                     boardImages[i][j].setImage(board
    [i][j].getImage());
635                 }
636             }
637         }
638
639         piecesView.getItems().clear();
640         for (BoardPiece piece: currentPlayers.get(0
    ).getPieces()) {
641             piecesView.getItems().add(piece.getName
    ());
642         }
643     }
644 }
645
```

```java
package com.example.gridgamefinal;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.io.IOException;

public class HelloApplication extends Application {
    @Override
    public void start(Stage stage) throws IOException {
        FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));
        Scene scene = new Scene(fxmlLoader.load(), 1000, 1000);
        stage.setTitle("Hello!");
        stage.setScene(scene);
        stage.show();
    }

    public static void main(String[] args) {
        launch();
    }
}
```