

```
1 package Game;
2
3 import java.io.Serializable;
4
5 public class Die implements Serializable {
6     int facets;
7     public int result;
8
9     public Die(int facets) {
10         this.facets = facets;
11     }
12
13     public void roll() {
14         result = (int) (Math.random()*this.facets);
15     }
16 }
17
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.Group?>
4 <?import javafx.scene.control.Button?>
5 <?import javafx.scene.control.Label?>
6 <?import javafx.scene.layout.AnchorPane?>
7 <?import javafx.scene.layout.Pane?>
8 <?import javafx.scene.paint.LinearGradient?>
9 <?import javafx.scene.paint.Stop?>
10 <?import javafx.scene.shape.Circle?>
11 <?import javafx.scene.shape.Rectangle?>
12
13 <AnchorPane fx:id="board" maxHeight="-Infinity" maxWidth="-Infinity"
14   minHeight="-Infinity" minWidth="-Infinity" prefHeight="600.0" prefWidth="800.
15   0" xmlns="http://javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1
16   " fx:controller="Game.GUI">
17   <children>
18     <Group fx:id="plots">
19       <Rectangle fx:id="plot0" arcHeight="5.0" arcWidth="5.0" height="60.0"
20         layoutX="100.0" layoutY="120.0" stroke="BLACK" strokeType="INSIDE"
21         width="60.0">
22         <fill>
23           <LinearGradient endX="1.0" endY="1.0">
24             <stops>
25               <Stop color="PURPLE" />
26               <Stop color="BLACK" offset="1.0" />
27             </stops>
28           </LinearGradient>
29         </fill></Rectangle>
30       <Rectangle fx:id="plot1" arcHeight="5.0" arcWidth="5.0" height="60.0"
31         layoutX="10.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
32         <fill>
33           <LinearGradient endX="1.0" endY="1.0">
34             <stops>
35               <Stop color="RED" />
36               <Stop color="BLACK" offset="1.0" />
37             </stops>
38           </LinearGradient>
39         </fill></Rectangle>
40       <Rectangle fx:id="plot2" arcHeight="5.0" arcWidth="5.0" height="60.0"
41         layoutX="70.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
42         <fill>
43           <LinearGradient endX="1.0" endY="1.0">
44             <stops>
45               <Stop color="BLACK" />
46               <Stop color="WHITE" offset="1.0" />
47             </stops>
48           </LinearGradient>
49         </fill>
```

```
50          </LinearGradient>
51      </fill></Rectangle>
52      <Rectangle fx:id="plot4" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="190.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
53          <fill>
54              <LinearGradient endX="1.0" endY="1.0">
55                  <stops>
56                      <Stop color="BLACK" />
57                      <Stop color="WHITE" offset="1.0" />
58                  </stops>
59          </LinearGradient>
60      </fill></Rectangle>
61      <Rectangle fx:id="plot5" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="250.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
62          <fill>
63              <LinearGradient endX="1.0" endY="1.0">
64                  <stops>
65                      <Stop color="BLACK" />
66                      <Stop color="WHITE" offset="1.0" />
67                  </stops>
68          </LinearGradient>
69      </fill></Rectangle>
70      <Rectangle fx:id="plot6" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="310.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
71          <fill>
72              <LinearGradient endX="1.0" endY="1.0">
73                  <stops>
74                      <Stop color="BLACK" />
75                      <Stop color="WHITE" offset="1.0" />
76                  </stops>
77          </LinearGradient>
78      </fill></Rectangle>
79      <Rectangle fx:id="plot7" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="370.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
80          <fill>
81              <LinearGradient endX="1.0" endY="1.0">
82                  <stops>
83                      <Stop color="BLACK" />
84                      <Stop color="WHITE" offset="1.0" />
85                  </stops>
86          </LinearGradient>
87      </fill></Rectangle>
88      <Rectangle fx:id="plot8" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="430.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
89          <fill>
90              <LinearGradient endX="1.0" endY="1.0">
91                  <stops>
92                      <Stop color="BLACK" />
93                      <Stop color="WHITE" offset="1.0" />
94                  </stops>
95          </LinearGradient>
96      </fill></Rectangle>
97      <Rectangle fx:id="plot9" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="490.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
```

```
98          <fill>
99              <LinearGradient endX="1.0" endY="1.0">
100                 <stops>
101                     <Stop color="BLACK" />
102                     <Stop color="WHITE" offset="1.0" />
103                 </stops>
104             </LinearGradient>
105         </fill></Rectangle>
106         <Rectangle fx:id="plot10" arcHeight="5.0" arcWidth="5.0" height
107             ="60.0" layoutX="550.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE"
108             width="60.0">
109             <fill>
110                 <LinearGradient endX="1.0" endY="1.0">
111                     <stops>
112                         <Stop color="BLACK" />
113                         <Stop color="WHITE" offset="1.0" />
114                     </stops>
115                 </LinearGradient>
116             </fill></Rectangle>
117             <Rectangle fx:id="plot11" arcHeight="5.0" arcWidth="5.0" height
118             ="60.0" layoutX="610.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE"
119             width="60.0">
120             <fill>
121                 <LinearGradient endX="1.0" endY="1.0">
122                     <stops>
123                         <Stop color="BLACK" />
124                         <Stop color="WHITE" offset="1.0" />
125                     </stops>
126                 </LinearGradient>
127             </fill></Rectangle>
128             <Rectangle fx:id="plot12" arcHeight="5.0" arcWidth="5.0" height
129             ="60.0" layoutX="670.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE"
130             width="60.0">
131             <fill>
132                 <LinearGradient endX="1.0" endY="1.0">
133                     <stops>
134                         <Stop color="BLACK" />
135                         <Stop color="WHITE" offset="1.0" />
136                     </stops>
137                 </LinearGradient>
138             </fill></Rectangle>
139             <Rectangle fx:id="plot13" arcHeight="5.0" arcWidth="5.0" height
140             ="60.0" layoutX="730.0" layoutY="30.0" stroke="BLACK" strokeType="INSIDE"
141             width="60.0">
142             <fill>
143                 <LinearGradient endX="1.0" endY="1.0">
144                     <stops>
145                         <Stop color="BLACK" />
146                         <Stop color="WHITE" offset="1.0" />
```

```
148          </stops>
149      </LinearGradient>
150      </fill></Rectangle>
151      <Rectangle fx:id="plot15" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="730.0" layoutY="150.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
152          <fill>
153              <LinearGradient endX="1.0" endY="1.0">
154                  <stops>
155                      <Stop color="BLACK" />
156                      <Stop color="WHITE" offset="1.0" />
157                  </stops>
158              </LinearGradient>
159          </fill></Rectangle>
160          <Rectangle fx:id="plot16" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="730.0" layoutY="210.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
161              <fill>
162                  <LinearGradient endX="1.0" endY="1.0">
163                      <stops>
164                          <Stop color="BLACK" />
165                          <Stop color="WHITE" offset="1.0" />
166                      </stops>
167                  </LinearGradient>
168              </fill></Rectangle>
169              <Rectangle fx:id="plot17" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="730.0" layoutY="270.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
170                  <fill>
171                      <LinearGradient endX="1.0" endY="1.0">
172                          <stops>
173                              <Stop color="BLACK" />
174                              <Stop color="WHITE" offset="1.0" />
175                          </stops>
176                      </LinearGradient>
177                  </fill></Rectangle>
178                  <Rectangle fx:id="plot18" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="730.0" layoutY="330.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
179                      <fill>
180                          <LinearGradient endX="1.0" endY="1.0">
181                              <stops>
182                                  <Stop color="BLACK" />
183                                  <Stop color="WHITE" offset="1.0" />
184                              </stops>
185                          </LinearGradient>
186                      </fill></Rectangle>
187                      <Rectangle fx:id="plot19" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="730.0" layoutY="390.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
188                          <fill>
189                              <LinearGradient endX="1.0" endY="1.0">
190                                  <stops>
191                                      <Stop color="BLACK" />
192                                      <Stop color="WHITE" offset="1.0" />
193                                  </stops>
194                              </LinearGradient>
195                          </fill></Rectangle>
196                          <Rectangle fx:id="plot20" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="730.0" layoutY="450.0" stroke="BLACK" strokeType="INSIDE"
```

```
196 width="60.0">
197     <fill>
198         <LinearGradient endX="1.0" endY="1.0">
199             <stops>
200                 <Stop color="BLACK" />
201                 <Stop color="WHITE" offset="1.0" />
202             </stops>
203         </LinearGradient>
204     </fill></Rectangle>
205     <Rectangle fx:id="plot21" arcHeight="5.0" arcWidth="5.0" height
206         ="60.0" layoutX="730.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE"
207         width="60.0">
208         <fill>
209             <LinearGradient endX="1.0" endY="1.0">
210                 <stops>
211                     <Stop color="BLACK" />
212                     <Stop color="WHITE" offset="1.0" />
213                 </stops>
214             </LinearGradient>
215         </fill></Rectangle>
216         <Rectangle fx:id="plot22" arcHeight="5.0" arcWidth="5.0" height
217         ="60.0" layoutX="670.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE"
218         width="60.0">
219         <fill>
220             <LinearGradient endX="1.0" endY="1.0">
221                 <stops>
222                     <Stop color="BLACK" />
223                     <Stop color="WHITE" offset="1.0" />
224                 </stops>
225             </LinearGradient>
226         </fill></Rectangle>
227         <Rectangle fx:id="plot23" arcHeight="5.0" arcWidth="5.0" height
228         ="60.0" layoutX="610.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE"
229         width="60.0">
230         <fill>
231             <LinearGradient endX="1.0" endY="1.0">
232                 <stops>
233                     <Stop color="BLACK" />
234                     <Stop color="WHITE" offset="1.0" />
235                 </stops>
236             </LinearGradient>
237         </fill></Rectangle>
238         <Rectangle fx:id="plot24" arcHeight="5.0" arcWidth="5.0" height
239         ="60.0" layoutX="550.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE"
240         width="60.0">
241         <fill>
242             <LinearGradient endX="1.0" endY="1.0">
243                 <stops>
244                     <Stop color="BLACK" />
```

```
246             <Stop color="WHITE" offset="1.0" />
247         </stops>
248     </LinearGradient>
249   </fill></Rectangle>
250   <Rectangle fx:id="plot26" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="430.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
251       <fill>
252           <LinearGradient endX="1.0" endY="1.0">
253               <stops>
254                   <Stop color="BLACK" />
255                   <Stop color="WHITE" offset="1.0" />
256               </stops>
257           </LinearGradient>
258       </fill></Rectangle>
259       <Rectangle fx:id="plot27" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="370.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
260         <fill>
261             <LinearGradient endX="1.0" endY="1.0">
262                 <stops>
263                     <Stop color="BLACK" />
264                     <Stop color="WHITE" offset="1.0" />
265                 </stops>
266             </LinearGradient>
267         </fill></Rectangle>
268         <Rectangle fx:id="plot28" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="310.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
269           <fill>
270             <LinearGradient endX="1.0" endY="1.0">
271                 <stops>
272                     <Stop color="BLACK" />
273                     <Stop color="WHITE" offset="1.0" />
274                 </stops>
275             </LinearGradient>
276         </fill></Rectangle>
277         <Rectangle fx:id="plot29" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="250.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
278           <fill>
279             <LinearGradient endX="1.0" endY="1.0">
280                 <stops>
281                     <Stop color="BLACK" />
282                     <Stop color="WHITE" offset="1.0" />
283                 </stops>
284             </LinearGradient>
285         </fill></Rectangle>
286         <Rectangle fx:id="plot30" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="190.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
287           <fill>
288             <LinearGradient endX="1.0" endY="1.0">
289                 <stops>
290                     <Stop color="BLACK" />
291                     <Stop color="WHITE" offset="1.0" />
292                 </stops>
293             </LinearGradient>
294         </fill></Rectangle>
295         <Rectangle fx:id="plot31" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="130.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
```

```
295 ="60.0" layoutX="130.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE"
  width="60.0">
296     <fill>
297         <LinearGradient endX="1.0" endY="1.0">
298             <stops>
299                 <Stop color="BLACK" />
300                 <Stop color="WHITE" offset="1.0" />
301             </stops>
302         </LinearGradient>
303     </fill></Rectangle>
304     <Rectangle fx:id="plot32" arcHeight="5.0" arcWidth="5.0" height
="60.0" layoutX="70.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE"
  width="60.0">
305     <fill>
306         <LinearGradient endX="1.0" endY="1.0">
307             <stops>
308                 <Stop color="BLACK" />
309                 <Stop color="WHITE" offset="1.0" />
310             </stops>
311         </LinearGradient>
312     </fill></Rectangle>
313     <Rectangle fx:id="plot33" arcHeight="5.0" arcWidth="5.0" height
="60.0" layoutX="10.0" layoutY="510.0" stroke="BLACK" strokeType="INSIDE"
  width="60.0">
314     <fill>
315         <LinearGradient endX="1.0" endY="1.0">
316             <stops>
317                 <Stop color="BLACK" />
318                 <Stop color="WHITE" offset="1.0" />
319             </stops>
320         </LinearGradient>
321     </fill></Rectangle>
322     <Rectangle fx:id="plot34" arcHeight="5.0" arcWidth="5.0" height
="60.0" layoutX="10.0" layoutY="450.0" stroke="BLACK" strokeType="INSIDE"
  width="60.0">
323     <fill>
324         <LinearGradient endX="1.0" endY="1.0">
325             <stops>
326                 <Stop color="BLACK" />
327                 <Stop color="WHITE" offset="1.0" />
328             </stops>
329         </LinearGradient>
330     </fill></Rectangle>
331     <Rectangle fx:id="plot35" arcHeight="5.0" arcWidth="5.0" height
="60.0" layoutX="10.0" layoutY="390.0" stroke="BLACK" strokeType="INSIDE"
  width="60.0">
332     <fill>
333         <LinearGradient endX="1.0" endY="1.0">
334             <stops>
335                 <Stop color="BLACK" />
336                 <Stop color="WHITE" offset="1.0" />
337             </stops>
338         </LinearGradient>
339     </fill></Rectangle>
340     <Rectangle fx:id="plot36" arcHeight="5.0" arcWidth="5.0" height
="60.0" layoutX="10.0" layoutY="330.0" stroke="BLACK" strokeType="INSIDE"
  width="60.0">
341     <fill>
342         <LinearGradient endX="1.0" endY="1.0">
343             <stops>
```

```
344             <Stop color="BLACK" />
345             <Stop color="WHITE" offset="1.0" />
346         </stops>
347     </LinearGradient>
348 </fill></Rectangle>
349 <Rectangle fx:id="plot37" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="10.0" layoutY="270.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
350     <fill>
351         <LinearGradient endX="1.0" endY="1.0">
352             <stops>
353                 <Stop color="BLACK" />
354                 <Stop color="WHITE" offset="1.0" />
355             </stops>
356         </LinearGradient>
357     <fill></Rectangle>
358 <Rectangle fx:id="plot38" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="10.0" layoutY="210.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
359     <fill>
360         <LinearGradient endX="1.0" endY="1.0">
361             <stops>
362                 <Stop color="BLACK" />
363                 <Stop color="WHITE" offset="1.0" />
364             </stops>
365         </LinearGradient>
366     <fill></Rectangle>
367 <Rectangle fx:id="plot39" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="10.0" layoutY="150.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
368     <fill>
369         <LinearGradient endX="1.0" endY="1.0">
370             <stops>
371                 <Stop color="BLACK" />
372                 <Stop color="WHITE" offset="1.0" />
373             </stops>
374         </LinearGradient>
375     <fill></Rectangle>
376 <Rectangle fx:id="plot40" arcHeight="5.0" arcWidth="5.0" height="60.0" layoutX="10.0" layoutY="90.0" stroke="BLACK" strokeType="INSIDE" width="60.0">
377     <fill>
378         <LinearGradient endX="1.0" endY="1.0">
379             <stops>
380                 <Stop color="BLACK" />
381                 <Stop color="WHITE" offset="1.0" />
382             </stops>
383         </LinearGradient>
384     <fill></Rectangle>
385 <Group fx:id="playerTokens">
386     <Circle fx:id="playerToken1" fill="RED" layoutX="350.0" layoutY="280.0" radius="10.0" stroke="BLACK" strokeType="INSIDE" />
387     <Circle fx:id="playerToken2" fill="BLUE" layoutX="530.0" layoutY="280.0" radius="10.0" stroke="BLACK" strokeType="INSIDE" />
388     <Circle fx:id="playerToken3" fill="YELLOW" layoutX="710.0" layoutY="280.0" radius="10.0" stroke="BLACK" strokeType="INSIDE" />
389     <Circle fx:id="playerToken4" fill="PURPLE" layoutX="350.0" layoutY="460.0" radius="10.0" stroke="BLACK" strokeType="INSIDE" />
390     <Circle fx:id="playerToken5" fill="GREEN" layoutX="530.0" layoutY="460.0" radius="10.0" stroke="BLACK" strokeType="INSIDE" />
```

```
391             <Circle fx:id="playerToken6" fill="ORANGE" layoutX="710.0"
392                 layoutY="460.0" radius="10.0" stroke="BLACK" strokeType="INSIDE" />
393         </Group>
394     </Group>
395     <Rectangle fx:id="player1box" arcHeight="5.0" arcWidth="5.0" height="120.0"
396                 layoutX="220.0" layoutY="150.0" stroke="BLACK" strokeType="INSIDE"
397                 width="120.0">
398         <fill>
399             <LinearGradient endX="1.0" endY="1.0">
400                 <stops>
401                     <Stop color="BLACK" />
402                     <Stop color="RED" offset="1.0" />
403                 </stops>
404             </LinearGradient>
405         </fill></Rectangle>
406     <Rectangle fx:id="player2box" arcHeight="5.0" arcWidth="5.0" height="120.0"
407                 layoutX="400.0" layoutY="150.0" stroke="BLACK" strokeType="INSIDE"
408                 width="120.0">
409         <fill>
410             <LinearGradient endX="1.0" endY="1.0">
411                 <stops>
412                     <Stop color="BLACK" />
413                     <Stop color="BLUE" offset="1.0" />
414                 </stops>
415             </LinearGradient>
416         </fill></Rectangle>
417     <Rectangle fx:id="player3box" arcHeight="5.0" arcWidth="5.0" height="120.0"
418                 layoutX="580.0" layoutY="150.0" stroke="BLACK" strokeType="INSIDE"
419                 width="120.0">
420         <fill>
421             <LinearGradient endX="1.0" endY="1.0">
422                 <stops>
423                     <Stop color="BLACK" />
424                     <Stop color="YELLOW" offset="1.0" />
425                 </stops>
426             </LinearGradient>
427         </fill></Rectangle>
428     <Rectangle fx:id="player4box" arcHeight="5.0" arcWidth="5.0" height="120.0"
429                 layoutX="220.0" layoutY="330.0" stroke="BLACK" strokeType="INSIDE"
430                 width="120.0">
431         <fill>
432             <LinearGradient endX="1.0" endY="1.0">
433                 <stops>
434                     <Stop color="BLACK" />
435                     <Stop color="PURPLE" offset="1.0" />
436                 </stops>
437             </LinearGradient>
438         </fill></Rectangle>
439     <Rectangle fx:id="player5box" arcHeight="5.0" arcWidth="5.0" height="120.0"
440                 layoutX="400.0" layoutY="330.0" stroke="BLACK" strokeType="INSIDE"
441                 width="120.0">
442         <fill>
443             <LinearGradient endX="1.0" endY="1.0">
444                 <stops>
445                     <Stop color="BLACK" />
446                     <Stop color="GREEN" offset="1.0" />
447                 </stops>
448             </LinearGradient>
449         </fill></Rectangle>
450     <Rectangle fx:id="player6box" arcHeight="5.0" arcWidth="5.0" height="120.0"
451                 layoutX="580.0" layoutY="330.0" stroke="BLACK" strokeType="INSIDE"
452                 width="120.0">
453         <fill>
454             <LinearGradient endX="1.0" endY="1.0">
455                 <stops>
456                     <Stop color="BLACK" />
457                     <Stop color="PINK" offset="1.0" />
458                 </stops>
459             </LinearGradient>
460         </fill></Rectangle>
```

```
439 120.0" layoutX="580.0" layoutY="330.0" stroke="BLACK" strokeType="INSIDE"
440     width="120.0">
441         <fill>
442             <LinearGradient endX="1.0" endY="1.0">
443                 <stops>
444                     <Stop color="BLACK" />
445                     <Stop color="ORANGE" offset="1.0" />
446                 </stops>
447             </LinearGradient>
448         </fill></Rectangle>
449     <Button fx:id="tradeButton" layoutX="107.0" layoutY="227.0"
450         mnemonicParsing="false" onAction="#tradeMenu" text="Trade" />
451     <Button fx:id="propertiesPlots" layoutX="95.0" layoutY="287.0"
452         mnemonicParsing="false" onAction="#manageProperties" text="Properties" />
453     <Button fx:id="rollDice" layoutX="98.0" layoutY="347.0"
454         mnemonicParsing="false" onAction="#rollDiceAction" text="Roll Dice" />
455     <Button fx:id="savegameButton" layoutX="722.0" layoutY="1.0"
456         mnemonicParsing="false" onAction="#SaveGameAction" text="Save Game" />
457     <Pane fx:id="playerPane1" layoutX="220.0" layoutY="150.0" prefHeight="120.0"
458         prefWidth="120.0">
459         <children>
460             <Label fx:id="playerName1" layoutX="11.0" layoutY="14.0" text="Name"
461                 textFill="WHITE" />
462             <Label fx:id="playerWealth1" layoutX="41.0" layoutY="52.0" text="Wealth"
463                 textFill="WHITE" />
464             <Label fx:id="playerPosition1" layoutX="14.0" layoutY="89.0" text="position"
465                 textFill="WHITE" />
466         </children>
467     </Pane>
468     <Pane fx:id="playerPane2" layoutX="400.0" layoutY="150.0" prefHeight="120.0"
469         prefWidth="120.0">
470         <children>
471             <Label fx:id="playerName2" layoutX="11.0" layoutY="14.0" text="Name"
472                 textFill="WHITE" />
473             <Label fx:id="playerWealth2" layoutX="41.0" layoutY="52.0" text="Wealth"
474                 textFill="WHITE" />
475             <Label fx:id="playerPosition2" layoutX="11.0" layoutY="89.0" text="position"
476                 textFill="WHITE" />
477         </children>
478     </Pane>
479     <Pane fx:id="playerPane3" layoutX="580.0" layoutY="150.0" prefHeight="120.0"
480         prefWidth="120.0">
481         <children>
482             <Label fx:id="playerName3" layoutX="11.0" layoutY="14.0" text="Name"
483                 textFill="WHITE" />
484             <Label fx:id="playerWealth3" layoutX="41.0" layoutY="52.0" text="Wealth"
485                 textFill="WHITE" />
486             <Label fx:id="playerPosition3" layoutX="14.0" layoutY="89.0" text="position"
487                 textFill="WHITE" />
488         </children>
489     </Pane>
490     <Pane fx:id="playerPane4" layoutX="220.0" layoutY="330.0" prefHeight="120.0"
491         prefWidth="120.0">
492         <children>
493             <Label fx:id="playerName4" layoutX="11.0" layoutY="14.0" text="Name"
494                 textFill="WHITE" />
495             <Label fx:id="playerWealth4" layoutX="41.0" layoutY="52.0" text="Wealth"
496                 textFill="WHITE" />
497             <Label fx:id="playerPosition4" layoutX="14.0" layoutY="89.0" text="position"
498                 textFill="WHITE" />
```

```
478      </children>
479  </Pane>
480  <Pane fx:id="playerPane5" layoutX="400.0" layoutY="330.0" prefHeight="120.0" prefWidth="120.0">
481      <children>
482          <Label fx:id="playerName5" layoutX="11.0" layoutY="14.0" text="Name" textFill="WHITE" />
483          <Label fx:id="playerWealth5" layoutX="41.0" layoutY="52.0" text="Wealth" textFill="WHITE" />
484          <Label fx:id="playerPosition5" layoutX="14.0" layoutY="89.0" text="position" textFill="WHITE" />
485      </children>
486  </Pane>
487  <Pane fx:id="playerPane6" layoutX="580.0" layoutY="330.0" prefHeight="120.0" prefWidth="120.0">
488      <children>
489          <Label fx:id="playerName6" layoutX="11.0" layoutY="14.0" text="Name" textFill="WHITE" />
490          <Label fx:id="playerWealth6" layoutX="41.0" layoutY="52.0" text="Wealth" textFill="WHITE" />
491          <Label fx:id="playerPosition6" layoutX="14.0" layoutY="89.0" text="position" textFill="WHITE" />
492      </children>
493  </Pane>
494  <Pane fx:id="plotPane0" layoutX="101.0" layoutY="120.0" prefHeight="60.0" prefWidth="60.0">
495      <children>
496          <Label fx:id="plotValue0" layoutX="16.0" layoutY="30.0" text="Value" textFill="WHITE" />
497          <Label fx:id="plotTitle0" layoutX="17.0" layoutY="15.0" text="Title" textFill="WHITE" />
498      </children>
499  </Pane>
500  <Pane fx:id="plotPane1" layoutX="10.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
501      <children>
502          <Label fx:id="plotValue1" layoutX="16.0" layoutY="30.0" text="Value" textFill="WHITE" />
503          <Label fx:id="plotTitle1" layoutX="17.0" layoutY="15.0" text="Title" textFill="WHITE" />
504      </children>
505  </Pane>
506  <Pane fx:id="plotPane2" layoutX="70.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
507      <children>
508          <Label fx:id="plotValue2" layoutX="16.0" layoutY="30.0" text="Value" />
509          <Label fx:id="plotTitle2" layoutX="17.0" layoutY="15.0" text="Title" />
510      </children>
511  </Pane>
512  <Pane fx:id="plotPane3" layoutX="130.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
513      <children>
514          <Label fx:id="plotValue3" layoutX="16.0" layoutY="30.0" text="Value" />
515          <Label fx:id="plotTitle3" layoutX="17.0" layoutY="15.0" text="Title" />
516      </children>
517  </Pane>
```

```
518      <Pane fx:id="plotPane4" layoutX="190.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
519          <children>
520              <Label fx:id="plotValue4" layoutX="16.0" layoutY="30.0" text="Value" />
521              <Label fx:id="plotTitle4" layoutX="17.0" layoutY="15.0" text="Title" />
522          </children>
523      </Pane>
524      <Pane fx:id="plotPane5" layoutX="250.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
525          <children>
526              <Label fx:id="plotValue5" layoutX="16.0" layoutY="30.0" text="Value" />
527              <Label fx:id="plotTitle5" layoutX="17.0" layoutY="15.0" text="Title" />
528          </children>
529      </Pane>
530      <Pane fx:id="plotPane6" layoutX="310.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
531          <children>
532              <Label fx:id="plotValue6" layoutX="16.0" layoutY="30.0" text="Value" />
533              <Label fx:id="plotTitle6" layoutX="17.0" layoutY="15.0" text="Title" />
534          </children>
535      </Pane>
536      <Pane fx:id="plotPane7" layoutX="370.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
537          <children>
538              <Label fx:id="plotValue7" layoutX="16.0" layoutY="30.0" text="Value" />
539              <Label fx:id="plotTitle7" layoutX="17.0" layoutY="15.0" text="Title" />
540          </children>
541      </Pane>
542      <Pane fx:id="plotPane8" layoutX="430.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
543          <children>
544              <Label fx:id="plotValue8" layoutX="16.0" layoutY="30.0" text="Value" />
545              <Label fx:id="plotTitle8" layoutX="17.0" layoutY="15.0" text="Title" />
546          </children>
547      </Pane>
548      <Pane fx:id="plotPane9" layoutX="490.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
549          <children>
550              <Label fx:id="plotValue9" layoutX="16.0" layoutY="30.0" text="Value" />
551              <Label fx:id="plotTitle9" layoutX="17.0" layoutY="15.0" text="Title" />
552          </children>
553      </Pane>
554      <Pane fx:id="plotPane10" layoutX="550.0" layoutY="30.0" prefHeight="60.0" prefWidth="60.0">
555          <children>
556              <Label fx:id="plotValue10" layoutX="16.0" layoutY="30.0" text="Value" />
557              <Label fx:id="plotTitle10" layoutX="17.0" layoutY="15.0" text="Title" />
```

```
557 Title" />
558     </children>
559   </Pane>
560   <Pane fx:id="plotPane11" layoutX="610.0" layoutY="30.0" prefHeight="60
.0" prefWidth="60.0">
561     <children>
562       <Label fx:id="plotValue11" layoutX="16.0" layoutY="30.0" text="
Value" />
563       <Label fx:id="plotTitle11" layoutX="17.0" layoutY="15.0" text="
Title" />
564     </children>
565   </Pane>
566   <Pane fx:id="plotPane12" layoutX="670.0" layoutY="30.0" prefHeight="60
.0" prefWidth="60.0">
567     <children>
568       <Label fx:id="plotValue12" layoutX="16.0" layoutY="30.0" text="
Value" />
569       <Label fx:id="plotTitle12" layoutX="17.0" layoutY="15.0" text="
Title" />
570     </children>
571   </Pane>
572   <Pane fx:id="plotPane13" layoutX="730.0" layoutY="30.0" prefHeight="60
.0" prefWidth="60.0">
573     <children>
574       <Label fx:id="plotValue13" layoutX="16.0" layoutY="30.0" text="
Value" />
575       <Label fx:id="plotTitle13" layoutX="17.0" layoutY="15.0" text="
Title" />
576     </children>
577   </Pane>
578   <Pane fx:id="plotPane14" layoutX="730.0" layoutY="90.0" prefHeight="60
.0" prefWidth="60.0">
579     <children>
580       <Label fx:id="plotValue14" layoutX="16.0" layoutY="30.0" text="
Value" />
581       <Label fx:id="plotTitle14" layoutX="17.0" layoutY="15.0" text="
Title" />
582     </children>
583   </Pane>
584   <Pane fx:id="plotPane15" layoutX="730.0" layoutY="150.0" prefHeight="
60.0" prefWidth="60.0">
585     <children>
586       <Label fx:id="plotValue15" layoutX="16.0" layoutY="30.0" text="
Value" />
587       <Label fx:id="plotTitle15" layoutX="17.0" layoutY="15.0" text="
Title" />
588     </children>
589   </Pane>
590   <Pane fx:id="plotPane16" layoutX="730.0" layoutY="210.0" prefHeight="
60.0" prefWidth="60.0">
591     <children>
592       <Label fx:id="plotValue16" layoutX="16.0" layoutY="30.0" text="
Value" />
593       <Label fx:id="plotTitle16" layoutX="17.0" layoutY="15.0" text="
Title" />
594     </children>
595   </Pane>
596   <Pane fx:id="plotPane17" layoutX="730.0" layoutY="270.0" prefHeight="
60.0" prefWidth="60.0">
597     <children>
```

```
598          <Label fx:id="plotValue17" layoutX="16.0" layoutY="30.0" text="Value" />
599          <Label fx:id="plotTitle17" layoutX="17.0" layoutY="15.0" text="Title" />
600      </children>
601  </Pane>
602  <Pane fx:id="plotPane18" layoutX="730.0" layoutY="330.0" prefHeight="60.0" prefWidth="60.0">
603      <children>
604          <Label fx:id="plotValue18" layoutX="16.0" layoutY="30.0" text="Value" />
605          <Label fx:id="plotTitle18" layoutX="17.0" layoutY="15.0" text="Title" />
606      </children>
607  </Pane>
608  <Pane fx:id="plotPane19" layoutX="730.0" layoutY="390.0" prefHeight="60.0" prefWidth="60.0">
609      <children>
610          <Label fx:id="plotValue19" layoutX="16.0" layoutY="30.0" text="Value" />
611          <Label fx:id="plotTitle19" layoutX="17.0" layoutY="15.0" text="Title" />
612      </children>
613  </Pane>
614  <Pane fx:id="plotPane20" layoutX="730.0" layoutY="450.0" prefHeight="60.0" prefWidth="60.0">
615      <children>
616          <Label fx:id="plotValue20" layoutX="16.0" layoutY="30.0" text="Value" />
617          <Label fx:id="plotTitle20" layoutX="17.0" layoutY="15.0" text="Title" />
618      </children>
619  </Pane>
620  <Pane fx:id="plotPane21" layoutX="730.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
621      <children>
622          <Label fx:id="plotValue21" layoutX="16.0" layoutY="30.0" text="Value" />
623          <Label fx:id="plotTitle21" layoutX="17.0" layoutY="15.0" text="Title" />
624      </children>
625  </Pane>
626  <Pane fx:id="plotPane22" layoutX="670.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
627      <children>
628          <Label fx:id="plotValue22" layoutX="16.0" layoutY="30.0" text="Value" />
629          <Label fx:id="plotTitle22" layoutX="17.0" layoutY="15.0" text="Title" />
630      </children>
631  </Pane>
632  <Pane fx:id="plotPane23" layoutX="610.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
633      <children>
634          <Label fx:id="plotValue23" layoutX="16.0" layoutY="30.0" text="Value" />
635          <Label fx:id="plotTitle23" layoutX="17.0" layoutY="15.0" text="Title" />
636      </children>
637  </Pane>
```

```
638      <Pane fx:id="plotPane24" layoutX="550.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
639          <children>
640              <Label fx:id="plotValue24" layoutX="16.0" layoutY="30.0" text="Value" />
641              <Label fx:id="plotTitle24" layoutX="17.0" layoutY="15.0" text="Title" />
642          </children>
643      </Pane>
644      <Pane fx:id="plotPane25" layoutX="490.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
645          <children>
646              <Label fx:id="plotValue25" layoutX="16.0" layoutY="30.0" text="Value" />
647              <Label fx:id="plotTitle25" layoutX="17.0" layoutY="15.0" text="Title" />
648          </children>
649      </Pane>
650      <Pane fx:id="plotPane26" layoutX="430.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
651          <children>
652              <Label fx:id="plotValue26" layoutX="16.0" layoutY="30.0" text="Value" />
653              <Label fx:id="plotTitle26" layoutX="17.0" layoutY="15.0" text="Title" />
654          </children>
655      </Pane>
656      <Pane fx:id="plotPane27" layoutX="370.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
657          <children>
658              <Label fx:id="plotValue27" layoutX="16.0" layoutY="30.0" text="Value" />
659              <Label fx:id="plotTitle27" layoutX="17.0" layoutY="15.0" text="Title" />
660          </children>
661      </Pane>
662      <Pane fx:id="plotPane28" layoutX="310.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
663          <children>
664              <Label fx:id="plotValue28" layoutX="16.0" layoutY="30.0" text="Value" />
665              <Label fx:id="plotTitle28" layoutX="17.0" layoutY="15.0" text="Title" />
666          </children>
667      </Pane>
668      <Pane fx:id="plotPane29" layoutX="250.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
669          <children>
670              <Label fx:id="plotValue29" layoutX="16.0" layoutY="30.0" text="Value" />
671              <Label fx:id="plotTitle29" layoutX="17.0" layoutY="15.0" text="Title" />
672          </children>
673      </Pane>
674      <Pane fx:id="plotPane30" layoutX="190.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
675          <children>
676              <Label fx:id="plotValue30" layoutX="16.0" layoutY="30.0" text="Value" />
677              <Label fx:id="plotTitle30" layoutX="17.0" layoutY="15.0" text="Title" />
```

```
677 Title" />
678     </children>
679   </Pane>
680   <Pane fx:id="plotPane31" layoutX="131.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
681     <children>
682       <Label fx:id="plotValue31" layoutX="16.0" layoutY="30.0" text="Value" />
683       <Label fx:id="plotTitle31" layoutX="17.0" layoutY="15.0" text="Title" />
684     </children>
685   </Pane>
686   <Pane fx:id="plotPane32" layoutX="70.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
687     <children>
688       <Label fx:id="plotValue32" layoutX="16.0" layoutY="30.0" text="Value" />
689       <Label fx:id="plotTitle32" layoutX="17.0" layoutY="15.0" text="Title" />
690     </children>
691   </Pane>
692   <Pane fx:id="plotPane33" layoutX="10.0" layoutY="510.0" prefHeight="60.0" prefWidth="60.0">
693     <children>
694       <Label fx:id="plotValue33" layoutX="16.0" layoutY="30.0" text="Value" />
695       <Label fx:id="plotTitle33" layoutX="17.0" layoutY="15.0" text="Title" />
696     </children>
697   </Pane>
698   <Pane fx:id="plotPane34" layoutX="10.0" layoutY="450.0" prefHeight="60.0" prefWidth="60.0">
699     <children>
700       <Label fx:id="plotValue34" layoutX="16.0" layoutY="30.0" text="Value" />
701       <Label fx:id="plotTitle34" layoutX="17.0" layoutY="15.0" text="Title" />
702     </children>
703   </Pane>
704   <Pane fx:id="plotPane35" layoutX="10.0" layoutY="390.0" prefHeight="60.0" prefWidth="60.0">
705     <children>
706       <Label fx:id="plotValue35" layoutX="16.0" layoutY="30.0" text="Value" />
707       <Label fx:id="plotTitle35" layoutX="17.0" layoutY="15.0" text="Title" />
708     </children>
709   </Pane>
710   <Pane fx:id="plotPane36" layoutX="10.0" layoutY="330.0" prefHeight="60.0" prefWidth="60.0">
711     <children>
712       <Label fx:id="plotValue36" layoutX="16.0" layoutY="30.0" text="Value" />
713       <Label fx:id="plotTitle36" layoutX="17.0" layoutY="15.0" text="Title" />
714     </children>
715   </Pane>
716   <Pane fx:id="plotPane37" layoutX="10.0" layoutY="270.0" prefHeight="60.0" prefWidth="60.0">
717     <children>
```

```
718      <Label fx:id="plotValue37" layoutX="16.0" layoutY="30.0" text="Value" />
719      <Label fx:id="plotTitle37" layoutX="17.0" layoutY="15.0" text="Title" />
720      </children>
721    </Pane>
722    <Pane fx:id="plotPane38" layoutX="10.0" layoutY="210.0" prefHeight="60.0" prefWidth="60.0">
723      <children>
724        <Label fx:id="plotValue38" layoutX="16.0" layoutY="30.0" text="Value" />
725        <Label fx:id="plotTitle38" layoutX="17.0" layoutY="15.0" text="Title" />
726      </children>
727    </Pane>
728    <Pane fx:id="plotPane39" layoutX="10.0" layoutY="150.0" prefHeight="60.0" prefWidth="60.0">
729      <children>
730        <Label fx:id="plotValue39" layoutX="16.0" layoutY="30.0" text="Value" />
731        <Label fx:id="plotTitle39" layoutX="17.0" layoutY="15.0" text="Title" />
732      </children>
733    </Pane>
734    <Pane fx:id="plotPane40" layoutX="10.0" layoutY="90.0" prefHeight="60.0" prefWidth="60.0">
735      <children>
736        <Label fx:id="plotValue40" layoutX="16.0" layoutY="30.0" text="Value" />
737        <Label fx:id="plotTitle40" layoutX="17.0" layoutY="15.0" text="Title" />
738      </children>
739    </Pane>
740  </children>
741 </AnchorPane>
742
```

```

1 package Game;
2
3 import DataHandler.DataManager;
4 import Game.Management.ManagementController;
5 import Game.Plots.Plot;
6 import Game.PopUpMessage.ChanceInfo;
7 import Game.PopUpMessage.PayRentInfoNormal;
8 import Game.PopUpMessage.TaxPay;
9 import Game.PopUpMessage.QueryToBuyPlot;
10 import javafx.collections.FXCollections;
11 import javafx.event.Event;
12 import javafx.fxml.FXML;
13 import javafx.fxml.FXMLLoader;
14 import javafx.scene.Group;
15 import javafx.scene.Node;
16 import javafx.scene.Parent;
17 import javafx.scene.Scene;
18 import javafx.scene.control.Button;
19 import javafx.scene.control.Label;
20 import javafx.scene.layout.Pane;
21 import javafx.scene.paint.Color;
22 import javafx.scene.paint.CycleMethod;
23 import javafx.scene.paint.LinearGradient;
24 import javafx.scene.paint.Stop;
25 import javafx.scene.shape.Rectangle;
26 import javafx.stage.Stage;
27
28 import java.io.IOException;
29 import java.lang.reflect.Method;
30 import java.util.ArrayList;
31 import java.util.HashMap;
32 import java.util.List;
33
34 public class GUI {
35     public GameMaster theGame;
36
37     @FXML private Button rollDice;
38
39     @FXML private Group playerTokens;
40     @FXML private Label playerName1; @FXML private Label playerWealth1; @FXML
41     private Label playerPosition1;
42     @FXML private Label playerName2; @FXML private Label playerWealth2; @FXML
43     private Label playerPosition2;
44     @FXML private Label playerName3; @FXML private Label playerWealth3; @FXML
45     private Label playerPosition3;
46     @FXML private Label playerName4; @FXML private Label playerWealth4; @FXML
47     private Label playerPosition4;
48     @FXML private Label playerName5; @FXML private Label playerWealth5; @FXML
49     private Label playerPosition5;
50     @FXML private Label playerName6; @FXML private Label playerWealth6; @FXML
51     private Label playerPosition6;
52     ArrayList<HashMap<String,Label>> playerFXLabels = new ArrayList<>();
53     Color[] playerColors = {Color.RED, Color.BLUE, Color.YELLOW, Color.PURPLE
      , Color.GREEN, Color.ORANGE};
54
55     @FXML private Group plots;
56
57     @FXML
58     private void rollDiceAction(Event event) {
59         if (theGame.players.get(theGame.turnOrder).boardPosition == 0) {

```

```

54             //Make jail actions. Break and Return functions to break method.
55         } else {
56             theGame.throwDiceToMove(theGame.players.get(theGame.turnOrder));
57         }
58         updatePlayerTokens();
59         resolvePlayerPlacement();
60         updatePlayerStatistics();
61         if (theGame.die1.result == theGame.die2.result) {
62             theGame.throwDiceToMove(theGame.players.get(theGame.turnOrder));
63             updatePlayerTokens();
64             resolvePlayerPlacement();
65             updatePlayerStatistics();
66             //Take an extra turn.
67             if (theGame.die1.result == theGame.die2.result) {
68                 theGame.throwDiceToMove(theGame.players.get(theGame.
69             turnOrder));
69                 updatePlayerTokens();
70                 resolvePlayerPlacement();
71                 updatePlayerStatistics();
72                 if (theGame.die1.result == theGame.die2.result) {
73                     //Go to jail, you criminal, think of the dice!
74                     theGame.players.get(theGame.turnOrder).boardPosition = 0
75                 ;
76                 updatePlayerTokens();
77             } else {
78                 //Take another extra turn.
79                 updatePlayerTokens();
80                 resolvePlayerPlacement();
81                 updatePlayerStatistics();
82             }
83         }
84     }
85     //Check for bankruptcy and game objectives.
86     theGame.endTurn();
87 }
88
89     private void resolvePlayerPlacement() {
90         if (theGame.board.plotsOnBoard.get((theGame.players.get(theGame.
91             turnOrder).boardPosition)).event != null) {
91             int placement = theGame.players.get(theGame.turnOrder).
92             boardPosition;
92             if (theGame.board.plotsOnBoard.get((theGame.players.get(theGame.
93                 turnOrder).boardPosition)).
93                 event.contentEquals("CHANCE")) {
94                 theGame.chanceDeck.shuffle();
95                 new ChanceInfo(theGame.chanceDeck.drawCard(), theGame);
96                 updatePlayerTokens();
97             } else if (theGame.board.plotsOnBoard.get((theGame.players.get(
97                 theGame.turnOrder).boardPosition)).
98                 event.contentEquals("TAX4000/10%")) {
99                 calculatePlayerValue();
100                 new TaxPay(theGame.players.get(theGame.turnOrder), "
100             TAX4000/10%");
101             } else if ((theGame.board.plotsOnBoard.get((theGame.players.get(
101                 theGame.turnOrder).boardPosition)).
102                 event.contentEquals("GOTOJAIL")))) {
103                 theGame.players.get(theGame.turnOrder).boardPosition = 0
103             ;
104             } else if (theGame.board.plotsOnBoard.get((theGame.players.get(
104

```

```

104 theGame.turnOrder).boardPosition));
105             event.contentEquals("TAX2000")) {
106                 new TaxPay(theGame.players.get(theGame.turnOrder), "
107 TAX2000");
108             }
109         if (theGame.players.get(theGame.turnOrder).boardPosition !=
110 placement) {
111             resolvePlayerPlacement();
112         }
113         if (theGame.board.ownerships.get(theGame.players.get(theGame.
turnOrder).boardPosition) == null &&
114             theGame.board.plotsOnBoard.get((theGame.players.get(theGame.
turnOrder).boardPosition)).value > 0) {
115                 new QueryToBuyPlot (theGame);
116                 if (theGame.players.get(theGame.turnOrder).ID ==
117                     theGame.board.ownerships.get(theGame.players.get(theGame
.turnOrder).boardPosition)) {
118                     changeOwnershipGraphics(
119                         theGame.board.plotsOnBoard.get(theGame.players.get(
theGame.turnOrder).boardPosition));
120                 }
121             } else if (theGame.board.plotsOnBoard.get((theGame.players.get(
theGame.turnOrder).boardPosition)).event == null) {
122                 new PayRentInfoNormal(theGame);
123             }
124         }
125     private void calculatePlayerValue() {
126         int value = 0;
127         value = value + theGame.players.get(theGame.turnOrder).wealth;
128         for (int i = 0; i < theGame.board.ownerships.size(); i++) {
129             if (theGame.board.ownerships.get(i) != null &&
130                 theGame.board.ownerships.get(i) == theGame.players.get(
theGame.turnOrder).ID) {
131                 value = value + theGame.board.plotsOnBoard.get(i).value;
132             }
133         }
134         theGame.players.get(theGame.turnOrder).value = value;
135     }
136
137 @FXML
138     private void SaveGameAction (Event event) {
139         DataManager dataManager = new DataManager("saveFiles/test");
140         dataManager.saveGameObjectToFile(this.theGame);
141     }
142
143 @FXML
144     private void tradeMenu (Event event) {
145
146     }
147
148 @FXML
149     private void manageProperties (Event event) {
150         try {
151             Stage managementStage = new Stage();
152             FXMLLoader loader = new FXMLLoader();
153             Pane root = (Pane)loader.load(getClass().getResource("/Game/
Management/Management.fxml"));
154             ManagementController managementController = (

```

```
154 ManagementController) loader.getController();
155         Scene scene = new Scene(root);
156         managementController.game = this.theGame;
157         managementController.fillTable();
158         managementStage.setScene(scene);
159         managementStage.setTitle("Property Management");
160         managementStage.setResizable(true);
161         managementStage.show();
162     } catch (IOException e) {
163         e.printStackTrace();
164     }
165 }
166
167 public void changeOwnershipGraphics(Plot plot) {
168     Node nodeRectangle = plots.getChildren().get(plot.ID);
169     if (nodeRectangle instanceof Rectangle) {
170         Stop[] stops = new Stop[]{new Stop(0, Color.valueOf(
171             (plot.district)),
172             new Stop(1, playerColors[theGame.turnOrder]))};
173         LinearGradient linearGradient = new LinearGradient(0, 0, 1, 1,
174 true, CycleMethod.NO_CYCLE, stops);
175         ((Rectangle) nodeRectangle).setFill(linearGradient);
176     }
177 }
178
179 public void updatePlayerTokens() {
180     playerTokens.getChildren().get(theGame.turnOrder).setLayoutY(plots.
181     getChildren().get(theGame.players.
182         get(theGame.turnOrder).boardPosition).getLayoutY()+(10+(8*
183 theGame.turnOrder)));
184     playerTokens.getChildren().get(theGame.turnOrder).setLayoutX(plots.
185     getChildren().get(theGame.players.
186         get(theGame.turnOrder).boardPosition).getLayoutX()+(10));
187 }
188
189 public void updateAllPlayerTokens() {
190     for (int i = 0; i < theGame.players.size(); i++) {
191         playerTokens.getChildren().get(i).setLayoutY(plots.getChildren().
192         get(theGame.players.get(i).boardPosition).
193             getLayoutY()+(10+(8*i)));
194         playerTokens.getChildren().get(i).setLayoutX(plots.getChildren().
195         get(theGame.players.get(i).boardPosition).
196             getLayoutX()+(10));
197     }
198 }
199
200 public void updatePlayerStatistics() {
201     for (int i = 0; i < theGame.players.size(); i++) {
202         playerFXLabels.get(i).get("Name").setText(theGame.players.get(i).
203             name);
204         playerFXLabels.get(i).get("Wealth").setText(theGame.players.get(i).
205             wealth + ", -");
206         playerFXLabels.get(i).get("Placement").setText("Placement: " +
207             theGame.players.get(i).boardPosition);
208     }
209 }
210
211 public void setupBoard() {
212     for (int i = 0; i < plots.getChildren().size(); i++) {
213         Node nodeRectangle = plots.getChildren().get(i);
```

```

205         if (nodeRectangle instanceof Rectangle && theGame.board.
206             plotsOnBoard.get(i).district != null) {
207             Stop[] stops = new Stop[] {new Stop(0, Color.valueOf(theGame
208                 .board.plotsOnBoard.get(i).district)),
209                 new Stop(1, Color.BLACK)};
210             LinearGradient linearGradient = new LinearGradient(0,0,1,1,
211                 true, CycleMethod.NO_CYCLE, stops);
212             ((Rectangle) nodeRectangle).setFill(linearGradient);
213         }
214     }
215
216     public void setupPlayers() {
217         for (int i = 0; i < theGame.players.size(); i++) {
218             playerFXLabels.add(new HashMap<>());
219         }
220         playerFXLabels.get(0).put("Name", playerName1);
221         playerFXLabels.get(0).put("Wealth",playerWealth1);
222         playerFXLabels.get(0).put("Placement",playerPosition1);
223         playerFXLabels.get(1).put("Name",playerName2);
224         playerFXLabels.get(1).put("Wealth",playerWealth2);
225         playerFXLabels.get(1).put("Placement",playerPosition2);
226         playerFXLabels.get(2).put("Name",playerName3);
227         playerFXLabels.get(2).put("Wealth",playerWealth3);
228         playerFXLabels.get(2).put("Placement",playerPosition3);
229         if (theGame.players.size() > 3) {
230             playerFXLabels.get(3).put("Name",playerName4);
231             playerFXLabels.get(3).put("Wealth",playerWealth4);
232             playerFXLabels.get(3).put("Placement",playerPosition4);
233             if (theGame.players.size() > 4) {
234                 playerFXLabels.get(4).put("Name",playerName5);
235                 playerFXLabels.get(4).put("Wealth",playerWealth5);
236                 playerFXLabels.get(4).put("Placement",playerPosition5);
237                 if (theGame.players.size() > 5) {
238                     playerFXLabels.get(5).put("Name",playerName6);
239                     playerFXLabels.get(5).put("Wealth",playerWealth6);
240                     playerFXLabels.get(5).put("Placement",playerPosition6);
241                     } else {
242                         playerName6.setVisible(false);
243                         playerWealth6.setVisible(false);
244                         playerPosition6.setVisible(false);
245                         playerTokens.getChildren().get(5).setVisible(false);
246                     }
247                 } else {
248                     playerName5.setVisible(false);
249                     playerWealth5.setVisible(false);
250                     playerPosition5.setVisible(false);
251                     playerTokens.getChildren().get(4).setVisible(false);
252                     playerName6.setVisible(false);
253                     playerWealth6.setVisible(false);
254                     playerPosition6.setVisible(false);
255                 }
256             } else {
257                 playerName4.setVisible(false);
258                 playerWealth4.setVisible(false);
259                 playerPosition4.setVisible(false);
260                 playerTokens.getChildren().get(3).setVisible(false);
261                 playerName5.setVisible(false);
262                 playerWealth5.setVisible(false);

```

```
262         playerPosition5.setVisible(false);
263         playerTokens.getChildren().get(4).setVisible(false);
264         playerName6.setVisible(false);
265         playerWealth6.setVisible(false);
266         playerPosition6.setVisible(false);
267         playerTokens.getChildren().get(5).setVisible(false);
268     }
269 }
270 }
271 }
```

```
1 package Game;
2
3 import Game.Plots.Plot;
4
5 import java.io.Serializable;
6 import java.util.ArrayList;
7 import java.util.HashMap;
8
9 public class Board implements Serializable {
10     public ArrayList<Plot> plotsOnBoard = new ArrayList<>();
11     public HashMap<Integer, Integer> ownerships = new HashMap<>(); //First
integer is the plotID, second integer is playerID.
12
13 }
14
```

```
1 package Game;
2
3 import java.io.Serializable;
4
5 public class Player implements Serializable {
6     public String name;
7     public int ID;
8     public int wealth;
9     public int value;
10    public int boardPosition;
11    public int jailBreakTokens;
12
13    public Player(String name, int ID) {
14        this.name = name;
15        this.boardPosition = 1;
16        this.wealth = 30000;
17        this.ID = ID;
18        this.jailBreakTokens = 0;
19    }
20}
21
```

```
1 package Game;
2
3 import java.io.Serializable;
4
5 public class ChanceCard implements Serializable {
6     public String text;
7     public int WealthChangePlayer;
8     public int StealFromOpponents;
9     public int Move;
10    public int MoveTo;
11    public boolean trainMove;
12    public boolean doublePay;
13    public int houseTax;
14    public int hotelTax;
15    public boolean jailBreakToken;
16    public int playerValueBelow;
17    public int conditionMetWealthChange;
18
19    public ChanceCard(String text, int wealthChangePlayer, int
stealFromOpponents, int move, int moveTo,
20                      boolean trainMove, boolean doublePay, int houseTax, int
hotelTax, boolean jailBreakToken,
21                      int playerValueBelow, int conditionMetWealthChange) {
22        this.text = text;
23        this.WealthChangePlayer = wealthChangePlayer;
24        this.StealFromOpponents = stealFromOpponents;
25        this.Move = move;
26        this.MoveTo = moveTo;
27        this.trainMove = trainMove;
28        this.doublePay = doublePay;
29        this.houseTax = houseTax;
30        this.hotelTax = hotelTax;
31        this.jailBreakToken = jailBreakToken;
32        this.playerValueBelow = playerValueBelow;
33        this.conditionMetWealthChange = conditionMetWealthChange;
34    }
35 }
36
```

```
1 package Game;
2
3 import java.io.Serializable;
4 import java.util.ArrayList;
5 import java.util.Collections;
6
7 public class ChanceDeck implements Serializable {
8     public ArrayList<ChanceCard> theDeck = new ArrayList<>();
9     public ArrayList<ChanceCard> discardPile = new ArrayList<>();
10
11    public ChanceDeck() {
12    }
13
14    //Should be placed under players
15    public ChanceCard drawCard() {
16        ChanceCard cardDrawn = this.theDeck.get(0);
17        this.discardPile.add(this.theDeck.get(0));
18        this.theDeck.remove(0);
19        if (this.theDeck.size() == 0) {
20            fill();
21            shuffle();
22        }
23        return cardDrawn;
24    }
25
26    public void shuffle() {
27        Collections.shuffle(theDeck);
28    }
29
30    private void fill() {
31        for (int i = 0; i < discardPile.size(); i++) {
32            this.theDeck.add(this.discardPile.get(i));
33        }
34        this.discardPile.clear();
35    }
36 }
37
38
```

```
1 package Game;
2
3 import java.io.Serializable;
4 import java.util.ArrayList;
5 import java.util.Collections;
6
7 public class GameMaster implements Serializable {
8     public ArrayList<Player> players;
9     public Board board;
10    int roundsLeft;
11    String boardType;
12    boolean auctionIfNotBought;
13    boolean fastStart;
14    boolean endOnBankruptcy;
15    boolean endOnOverwhelmingWealth;
16    boolean differingInheritance;
17    public boolean randomizeTurnOrder;
18    public int turnOrder;
19    public ChanceDeck chanceDeck;
20    public Die die1;
21    public Die die2;
22
23    public GameMaster(int roundsLeft, String boardType, boolean
auctionIfNotBought,
24                      boolean fastStart, boolean endOnBankruptcy, boolean
endOnOverwhelmingWealth,
25                      boolean differingInheritance, boolean
randomizeTurnOrder) {
26        this.players = new ArrayList<>();
27        this.roundsLeft = roundsLeft;
28        this.boardType = boardType;
29        this.auctionIfNotBought = auctionIfNotBought;
30        this.fastStart = fastStart;
31        this.endOnBankruptcy = endOnBankruptcy;
32        this.endOnOverwhelmingWealth = endOnOverwhelmingWealth;
33        this.differingInheritance = differingInheritance;
34        this.randomizeTurnOrder = randomizeTurnOrder;
35        this.turnOrder = 0;
36        this.die1 = new Die(6);
37        this.die2 = new Die(6);
38    }
39
40    public void randomizeTurnOrder() {
41        Collections.shuffle(this.players);
42    }
43
44    public void endTurn() {
45        this.turnOrder++;
46        if (this.turnOrder == this.players.size()) {
47            this.turnOrder = 0;
48        }
49    }
50
51    public void throwDiceToMove(Player playerToMove) {
52        this.die1.roll(); this.die2.roll();
53        playerToMove.boardPosition = playerToMove.boardPosition + this.die1.
result + this.die2.result;
54        if (playerToMove.boardPosition > this.board.plotsOnBoard.size()-1) {
55            playerToMove.wealth = playerToMove.wealth + 4000;
56            playerToMove.boardPosition = playerToMove.boardPosition -(this.
```

```
56 board.plotsOnBoard.size()-1);
57     }
58 }
59 }
60
```

```
1 package Game.Plots;
2
3 import java.io.Serializable;
4
5 public class Plot implements Serializable {
6     public int ID;
7     public int value;
8     public String event; public String title;
9     public int[] rent;
10    public int upgrades = 0;
11    public String district;
12
13    //Creates dummy plot.
14    public Plot(int ID) {
15        this.ID = ID;
16    }
17
18    //Creates event plot.
19    public Plot(int ID, String event) {
20        this.ID = ID;
21        this.event = event;
22    }
23
24    //Creates a plot belonging to a coloured monopoly
25    public Plot(int ID, String title, int value, int rent, int U1, int U2,
26    int U3, int U4, int U5, String district) {
27        this.ID = ID;
28        this.title = title;
29        this.value = value;
30        this.rent = new int[]{rent,U1,U2,U3,U4,U5};
31        this.district = district;
32    }
33}
34
```

```
1 package Game.Plots;
2
3 public class Train extends Plot {
4
5     public Train(int ID, String title) {
6         super(ID, title);
7         this.ID = ID;
8         this.title = title;
9         this.value = 4000;
10        this.rent = new int[]{500, 1000, 2000, 4000};
11        this.district = "GOLD";
12    }
13}
14
```

```
1 package Game.Plots;  
2  
3 public class Utility extends Plot {  
4  
5     public Utility(int ID, String title) {  
6         super(ID, title);  
7         this.ID = ID;  
8         this.title = title;  
9         this.value = 3000;  
10        this.district = "SILVER";  
11    }  
12}  
13
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.control.Button?>
4 <?import javafx.scene.control.Label?>
5 <?import javafx.scene.control.TableView?>
6 <?import javafx.scene.layout.AnchorPane?>
7
8
9 <AnchorPane prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/
  javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Game.
  Management.ManagementController">
10   <children>
11     <TableView fx:id="tableOfProperties" layoutX="14.0" layoutY="14.0"
  prefHeight="328.0" prefWidth="567.0" />
12     <Button layoutX="541.0" layoutY="360.0" mnemonicParsing="false" text=""
  Done" />
13     <Label fx:id="playerWealth" layoutX="14.0" layoutY="343.0" text="Player
  Wealth" />
14   </children>
15 </AnchorPane>
16
```

```
1 package Game.Management;
2
3 import Game.GameMaster;
4 import Game.Player;
5 import Game.Plots.Plot;
6 import javafx.collections.FXCollections;
7 import javafx.collections.ObservableList;
8 import javafx.fxml.FXML;
9 import javafx.scene.control.Button;
10 import javafx.scene.control.TableColumn;
11 import javafx.scene.control.TableView;
12 import javafx.scene.control.cell.PropertyValueFactory;
13
14 import java.util.Map;
15
16 public class ManagementController {
17     @FXML TableView<Plot> tableOfProperties;
18     TableColumn<Plot, String> titleColumn = new TableColumn<>("Title");
19     TableColumn<Plot, Integer> valueColumn = new TableColumn<>("Value");
20     TableColumn<Plot, Boolean> monopolyColumn = new TableColumn<>("Monopoly");
21     TableColumn<Plot, Integer> housesColumn = new TableColumn<>("Houses");
22     TableColumn<Plot, Boolean> hotelColumn = new TableColumn<>("Hotel");
23     TableColumn<Plot, Integer> rentColumn = new TableColumn<>("Rent");
24     TableColumn<Plot, Button> upgradeColumn = new TableColumn<>("Upgrade");
25     TableColumn<Plot, Button> sellColumn = new TableColumn<>("Sell Assets");
26     TableColumn<Plot, Button> pawnColumn = new TableColumn<>("Pawn");
27
28     public GameMaster game;
29     ObservableList<Plot> data = FXCollections.observableArrayList();
30
31     public void fillTable() {
32         Player activePlayer = game.players.get(game.turnOrder);
33         for (Map.Entry<Integer, Integer> entry : game.board.ownerships.
34             entrySet()) {
35             if (entry.getValue() == activePlayer.ID) {
36                 data.add(game.board.plotsOnBoard.get(entry.getKey()));
37             }
38             titleColumn.setCellValueFactory(new PropertyValueFactory<>("title"));
39             valueColumn.setCellValueFactory(new PropertyValueFactory<>("value"));
40             tableOfProperties.setItems(data);
41             tableOfProperties.getColumns().addAll(titleColumn, valueColumn,
42             monopolyColumn, housesColumn, hotelColumn,
43             rentColumn, upgradeColumn, sellColumn, pawnColumn);
44         }
45     }

```

```
1 package Game.PopUpMessage;
2
3 import Game.ChanceCard;
4 import Game.Player;
5 import javafx.scene.control.Alert;
6 import javafx.scene.control.ButtonType;
7
8 import java.util.Optional;
9
10 public class TaxPay {
11
12     public TaxPay(Player player, String taxType) {
13         if (taxType.contentEquals("TAX4000/10%")) {
14             Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
15             alert.setTitle("Tax.");
16             alert.setHeaderText(player.name + " you must either pay 10% of "
+ player.value +
17                     " which is the value of your assets or 4000,-?");
18             alert.setContentText("Will you pay to 10%?");
19             Optional<ButtonType> result = alert.showAndWait();
20             if (result.get() == ButtonType.OK) {
21                 player.wealth = player.wealth - (player.value / 10);
22             } else {
23                 player.wealth = player.wealth - 4000;
24             }
25         } else if (taxType.contentEquals("TAX2000")) {
26             Alert alert = new Alert(Alert.AlertType.INFORMATION);
27             alert.setTitle("Tax.");
28             alert.setHeaderText(player.name + " you must either pay 2000,- in "
+ taxes.);
29             alert.setContentText("Press to pay, you have no choice.");
30             Optional<ButtonType> result = alert.showAndWait();
31             if (result.get() == ButtonType.OK) {
32                 player.wealth = player.wealth - 2000;
33             } else {
34             }
35         } else {
36             System.out.println("Unknown tax type.");
37         }
38     }
39 }
40
```



```

50                     //INSERT CODE FOR PAYMENT, BOOLEAN TYPE?
51                 }
52             trainFound = true;
53         break;
54     }
55 }
56 if (!trainFound) {
57     for (int i = 0; i < game.board.plotsOnBoard.size(); i
58    ++) {
59         if (game.board.plotsOnBoard.get(i) instanceof Train
60     ) {
61             game.players.get(game.turnOrder).boardPosition
62         = game.board.plotsOnBoard.get(i).ID;
63         //Code for double pay
64         if (chanceCard.doublePay) {
65             //INSERT CODE FOR PAYMENT, BOOLEAN TYPE?
66         }
67     }
68 }
69 //Determine if start have been passed.
70 if (game.players.get(game.turnOrder).boardPosition <
71 lastPlacement
72         && game.players.get(game.turnOrder).boardPosition
73 != 0) {
74     game.players.get(game.turnOrder).wealth = game.players.
75     get(game.turnOrder).wealth;
76 }
77 //HouseTax and Hotel Tax
78 //Jail Break Token
79 if (chanceCard.jailBreakToken) {
80     game.players.get(game.turnOrder).jailBreakTokens++;
81 }
82 //Determine Legation.
83 if (game.players.get(game.turnOrder).wealth < chanceCard.
84 playerValueBelow) {
85     game.players.get(game.turnOrder).wealth =
86         game.players.get(game.turnOrder).wealth + chanceCard
87 .conditionMetWealthChange;
88 }
89

```

```
1 package Game.PopUpMessage;
2
3 import Game.GameMaster;
4 import Game.Player;
5 import Game.Plots.Plot;
6 import javafx.scene.control.Alert;
7 import javafx.scene.control.ButtonType;
8
9 import java.util.HashMap;
10 import java.util.Optional;
11
12
13 public class QueryToBuyPlot {
14
15
16     public QueryToBuyPlot(GameMaster gameMaster) {
17         //Plot plot, Player player, HashMap<Integer, Integer> ownerships
18         Player player = gameMaster.players.get(gameMaster.turnOrder);
19         Plot plot = gameMaster.board.plotsOnBoard.get(player.boardPosition);
20         HashMap<Integer, Integer> ownerships = gameMaster.board.ownerships;
21
22         Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
23         alert.setTitle(plot.title);
24         alert.setHeaderText(player.name + " will you buy " + plot.title + "?"
25 );
26         alert.setContentText("The price for this property is: " + plot.value
27             + ", -, you currently have: " +
28                 player.wealth + ", -");
29         Optional<ButtonType> result = alert.showAndWait();
30         if(result.get() == ButtonType.OK) {
31             player.wealth = player.wealth - plot.value;
32             ownerships.put(plot.ID, player.ID);
33         } else {
34     }
35 }
```

```
1 package Game.PopUpMessage;
2
3 import Game.GameMaster;
4 import Game.Player;
5 import Game.Plots.Plot;
6 import javafx.scene.control.Alert;
7 import javafx.scene.control.ButtonType;
8
9 import java.util.Optional;
10
11 public class PayRentInfoNormal {
12
13     public PayRentInfoNormal(GameMaster game) {
14         Player playerToPay = game.players.get(game.turnOrder);
15         Plot plot = game.board.plotsOnBoard.get(playerToPay.boardPosition);
16         Player playerToReceive = game.players.get(game.board.ownerships.get(
17             plot.ID));
18         int rent = plot.rent[plot.upgrades];
19
20         Alert alert = new Alert(Alert.AlertType.INFORMATION);
21         alert.setTitle("Pay Rent.");
22         alert.setHeaderText(playerToPay.name + " must pay " + playerToReceive
23             .name + " a rent of " +
24             rent + " for visiting " + plot.title);
25         alert.setContentText("Press to pay, you have no choice.");
26         Optional<ButtonType> result = alert.showAndWait();
27         if (result.get() == ButtonType.OK) {
28             playerToPay.wealth = playerToPay.wealth - rent;
29             playerToReceive.wealth = playerToReceive.wealth + rent;
30         } else {
31     }
32 }
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import java.lang.*?>
4 <?import java.util.*?>
5 <?import javafx.scene.*?>
6 <?import javafx.scene.control.*?>
7 <?import javafx.scene.layout.*?>
8
9 <AnchorPane xmlns="http://javafx.com/javafx"
10      xmlns:fx="http://javafx.com/fxml"
11      fx:controller="Game.TradeInterface.TradeInterfaceController"
12      prefHeight="400.0" prefWidth="600.0">
13
14 </AnchorPane>
15
```

```
1 package Game.TradeInterface;  
2  
3 public class TradeInterfaceController {  
4 }  
5
```

```
1 package Options;  
2  
3 public class Configuration {  
4 }  
5
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import java.lang.*?>
4 <?import java.util.*?>
5 <?import javafx.scene.*?>
6 <?import javafx.scene.control.*?>
7 <?import javafx.scene.layout.*?>
8
9 <AnchorPane xmlns="http://javafx.com/javafx"
10      xmlns:fx="http://javafx.com/fxml"
11      fx:controller="Options.OptionsInterface"
12      prefHeight="400.0" prefWidth="600.0">
13
14 </AnchorPane>
15
```

```
1 package Options;  
2  
3 public class OptionsInterface {  
4 }  
5
```

```
1 Manifest-Version: 1.0
2 Main-Class: StartMenu.Main
3
4
```

```
1 package StartMenu;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.stage.Stage;
8
9 public class Main extends Application {
10
11     public void start(Stage stage) throws Exception {
12         Parent root = (Parent) FXMLLoader.load(getClass().getResource("start.
13             fxml"));
14         Scene scene = new Scene(root);
15         stage.setScene(scene);
16         stage.setTitle("The Landlord's Game");
17         stage.show();
18     }
19
20     public static void main(String[] args){
21         launch(args);
22     }
23 }
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.control.Button?>
4 <?import javafx.scene.control.Label?>
5 <?import javafx.scene.layout.AnchorPane?>
6
7 <AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="400.0" xmlns="http://
javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="
StartMenu.StartMenuController">
8     <children>
9         <Button fx:id="newGame" layoutX="25.0" layoutY="25.0" maxHeight="-
Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
mnemonicParsing="false" onAction="#runNewGame" prefHeight="50.0" prefWidth="
350.0" text="New Game" />
10        <Button fx:id="loadGame" layoutX="25.0" layoutY="100.0" maxHeight="-
Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
mnemonicParsing="false" onAction="#runLoadGame" prefHeight="50.0" prefWidth="
350.0" text="Load Game" />
11        <Button fx:id="options" layoutX="25.0" layoutY="175.0" maxHeight="-
Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
mnemonicParsing="false" onAction="#runOptions" prefHeight="50.0" prefWidth="
350.0" text="Options" />
12        <Button fx:id="exit" layoutX="25.0" layoutY="250.0" maxHeight="-
Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
mnemonicParsing="false" onAction="#exitGame" prefHeight="50.0" prefWidth="350
.0" text="Exit" />
13        <Label layoutX="25.0" layoutY="325.0" maxHeight="-Infinity" maxWidth="-
Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="50.0"
prefWidth="350.0" text="Created by Daniel Jacobsen, Josephine Romby, Enise
Yilmaz & Asger Ågidius" wrapText="true" />
14    </children>
15 </AnchorPane>
16
```

```

1 package StartMenu;
2
3 import GameCreator.GameCreator;
4 import LoadGameInterface.LoadGameScreen;
5 import Options.OptionsInterface;
6 import javafx.embed.swing.JFXPanel;
7 import javafx.event.ActionEvent;
8 import javafx.fxml.FXML;
9 import javafx.fxml.FXMLLoader;
10 import javafx.fxml.Initializable;
11 import javafx.scene.Scene;
12 import javafx.scene.control.Button;
13 import javafx.scene.layout.Pane;
14 import javafx.scene.layout.VBox;
15 import javafx.stage.FileChooser;
16 import javafx.stage.Stage;
17
18 import java.io.File;
19 import java.io.IOException;
20 import java.net.URL;
21 import java.util.ResourceBundle;
22
23 public class StartMenuController implements Initializable {
24     @FXML
25     private Button newGame;
26     @FXML
27     private Button loadGame;
28     @FXML
29     private Button options;
30     @FXML
31     private Button exit;
32     private JFXPanel stage;
33
34     @Override
35     public void initialize(URL location, ResourceBundle resources) {
36     }
37
38     @FXML
39     private void runNewGame(ActionEvent event) {
40         try {
41             Stage gameCreatorStage = new Stage();
42             FXMLLoader loader = new FXMLLoader();
43             Pane root = (Pane)loader.load(getClass().getResource("/
GameCreator/GameCreator.fxml").openStream());
44             GameCreator gameCreator = (GameCreator)loader.getController();
45             Scene scene = new Scene(root);
46             gameCreator.updateItems();
47             gameCreatorStage.setScene(scene);
48             gameCreatorStage.setTitle("Start new game");
49             gameCreatorStage.setResizable(false);
50             gameCreatorStage.show();
51         } catch (IOException e) {
52             e.printStackTrace();
53         }
54     }
55     @FXML
56     private void runLoadGame(ActionEvent event) {
57         try {
58             Stage loadGameStage = new Stage();
59             FXMLLoader loader = new FXMLLoader();

```

```
60         FileChooser fileChooser = new FileChooser();
61         Button btn = new Button("Select File");
62         btn.setOnAction(e -> {
63             File selectedFile = fileChooser.showOpenDialog(loadGameStage
64         );
65         VBox vBox = new VBox(btn);
66         Scene scene = new Scene(vBox, 400, 400);
67         loadGameStage.setScene(scene);
68         loadGameStage.show();
69         Pane root = (Pane)loader.load(getClass().getResource("/
LoadGameInterface/LoadGameScreen.fxml").openStream());
70         LoadGameScreen loadGameScreen = (LoadGameScreen) loader.
getController();
71         loadGameStage.setScene(scene);
72         loadGameStage.setTitle("Start new game");
73         loadGameStage.setResizable(false);
74         loadGameStage.show();
75     } catch (IOException e) {
76         e.printStackTrace();
77     }
78 }
79 @FXML
80 private void runOptions(ActionEvent event) {
81     try {
82         Stage optionsStage = new Stage();
83         FXMLLoader loader = new FXMLLoader();
84         Pane root = (Pane)loader.load(getClass().getResource("/Options/
OptionsInterface.fxml").openStream());
85         OptionsInterface optionsInterface = (OptionsInterface) loader.
getController();
86         Scene scene = new Scene(root);
87         optionsStage.setScene(scene);
88         optionsStage.setTitle("Options");
89         optionsStage.setResizable(false);
90         optionsStage.show();
91     } catch (IOException e) {
92         e.printStackTrace();
93     }
94 }
95 @FXML
96 private void exitGame(ActionEvent event) {
97     System.exit(1);
98 }
99 }
100 }
```

```
1 package DataHandler;
2
3 import Game.GameMaster;
4 import java.io.*;
5
6 public class DataManager implements Serializable{
7     String filepath;
8
9     public DataManager(String filepath) {
10         this.filepath = filepath;
11     }
12
13     //Saves a Game object to a file given in the DataManagers filepath.
14     public void saveGameObjectToFile (GameMaster currentGame) {
15         try {
16             FileOutputStream fileOut = new FileOutputStream(filepath);
17             ObjectOutputStream objectOut = new ObjectOutputStream(fileOut);
18             objectOut.writeObject(currentGame);
19             objectOut.close();
20             System.out.println("Game successfully saved.");
21         } catch (Exception ex) {
22             System.out.println("Something went wrong with the save.");
23             ex.printStackTrace();
24         }
25     }
26     //Loads a Game object from a file given in the DataManagers filepath.
27     public GameMaster loadGameObjectFromFile() {
28         try {
29             FileInputStream fileIn = new FileInputStream(filepath);
30             ObjectInputStream objectIn = new ObjectInputStream(fileIn);
31             Object currentGame = objectIn.readObject();
32             objectIn.close();
33             System.out.println("The Game has been read.");
34             return (GameMaster) currentGame;
35         } catch (Exception ex) {
36             ex.printStackTrace();
37             System.out.println("Something went wrong with the load.");
38             return null;
39         }
40     }
41 }
```

```

1 package DataHandler;
2
3 import Game.Board;
4 import Game.ChanceCard;
5 import Game.ChanceDeck;
6 import Game.Plots.Plot;
7 import Game.Plots.Train;
8 import Game.Plots.Utility;
9 import org.json.simple.JSONArray;
10 import org.json.simple.JSONObject;
11 import org.json.simple.parser.JSONParser;
12 import org.json.simple.parser.ParseException;
13
14 import java.io.FileReader;
15 import java.io.IOException;
16 import java.util.ArrayList;
17
18 //Følger denne guide https://howtodoinjava.com/library/json-simple-read-write-json-examples/
19
20 public class JSONImporter {
21     JSONParser parser = new JSONParser();
22     FileReader reader;
23
24     public ChanceDeck createChanceDeckJson(String boardName) {
25         ArrayList<ChanceCard> cardsForDeck = new ArrayList<>();
26         try {
27             reader = new FileReader("package.json");
28             Object object = parser.parse(reader);
29             JSONObject initialFile = (JSONObject) object;
30             JSONObject boardObject = (JSONObject) initialFile.get("chanceCards");
31             JSONArray arrayOfCards = (JSONArray) boardObject.get(boardName);
32             for (int i = 0; i < arrayOfCards.size(); i++) {
33                 JSONObject cardJSONObject = (JSONObject) arrayOfCards.get(i);
34                 String cardText = cardJSONObject.get("Text").toString();
35                 int wealthPlayerChange;
36                 if (cardJSONObject.get("WealthChangePlayer") == null) {
37                     wealthPlayerChange = 0;
38                 } else {
39                     wealthPlayerChange = (int) (long) cardJSONObject.get("WealthChangePlayer");
40                 }
41                 int move;
42                 if (cardJSONObject.get("Move") == null) {
43                     move = 0;
44                 } else {
45                     move = (int) (long) cardJSONObject.get("Move");
46                 }
47                 int moveTo;
48                 if (cardJSONObject.get("MoveTo") == null) {
49                     moveTo = -1;
50                 } else {
51                     moveTo = (int) (long) cardJSONObject.get("MoveTo");
52                 }
53                 boolean trainMove;
54                 if (cardJSONObject.get("TrainMove") == null) {
55                     trainMove = false;
56                 } else {
57                     trainMove = (boolean) cardJSONObject.get("TrainMove");
58                 }
59             }
60         } catch (IOException e) {
61             e.printStackTrace();
62         }
63     }
64 }
```

```

58         }
59         boolean doublePay;
60         if (cardJSONObject.get("DoublePay") == null) {
61             doublePay = false;
62         } else {
63             doublePay = (boolean) cardJSONObject.get("DoublePay");
64         }
65         int stealFromOpponents;
66         if (cardJSONObject.get("StealFromOpponents") == null) {
67             stealFromOpponents = 0;
68         } else {
69             stealFromOpponents = (int) (long) cardJSONObject.get(
70                 "StealFromOpponents");
71         }
72         int houseTax;
73         if (cardJSONObject.get("HouseTax") == null) {
74             houseTax = 0;
75         } else {
76             houseTax = (int) (long) cardJSONObject.get("HouseTax");
77         }
78         int hotelTax;
79         if (cardJSONObject.get("HotelTax") == null) {
80             hotelTax = 0;
81         } else {
82             hotelTax = (int) (long) cardJSONObject.get("HotelTax");
83         }
84         boolean jailBreakToken;
85         if (cardJSONObject.get("JailBreakToken") == null) {
86             jailBreakToken = false;
87         } else {
88             jailBreakToken = (boolean) cardJSONObject.get(
89                 "JailBreakToken");
90         }
91         int playerValueBelow;
92         if (cardJSONObject.get("PlayerValueBelow") == null) {
93             playerValueBelow = 0;
94         } else {
95             playerValueBelow = (int) (long) cardJSONObject.get(
96                 "PlayerValueBelow");
97         }
98         int conditionMetWealthChange;
99         if (cardJSONObject.get("ConditionMetWealthChange") == null
100            ) {
101             conditionMetWealthChange = 0;
102         } else {
103             conditionMetWealthChange = (int) (long) cardJSONObject.
104             get("ConditionMetWealthChange");
105         }
106         ChanceCard chanceCard = new ChanceCard(cardText,
107             wealthPlayerChange,stealFromOpponents,move,moveTo,
108             trainMove,doublePay,houseTax,hotelTax,jailBreakToken
109             ,playerValueBelow,conditionMetWealthChange);
110         cardsForDeck.add(chanceCard);
111     }
112     } catch (ParseException | IOException e) {
113         e.printStackTrace();
114     }
115     ChanceDeck chanceDeck = new ChanceDeck();
116     chanceDeck.theDeck.addAll(cardsForDeck);
117     return chanceDeck;

```

```

111     }
112
113     public Board createBoardJSON(String boardName) {
114         ArrayList <Plot> plotsForBoard = new ArrayList<>();
115         try {
116             reader = new FileReader("package.json");
117             Object object = parser.parse(reader);
118             Plot plot;
119             JSONObject initialFile = (JSONObject) object;
120             JSONObject boardObject = (JSONObject) initialFile.get("plots");
121             JSONArray arrayOfPlots = (JSONArray) boardObject.get(boardName);
122             for (int i = 0; i < arrayOfPlots.size(); i++) {
123                 JSONObject plotJSONObject = (JSONObject) arrayOfPlots.get(i);
124                 int plotID = Integer.parseInt(plotJSONObject.get("ID").toString());
125                 if (plotJSONObject.size() == 2) {
126                     if (plotJSONObject.get("EVENT") == null) {
127                         if (plotJSONObject.get("TRAIN") != null) {
128                             plot = new Train(plotID, plotJSONObject.get("TRAIN").toString());
129                         } else if (plotJSONObject.get("UTILITY") != null) {
130                             plot = new Utility(plotID, plotJSONObject.get("UTILITY").toString());
131                         } else {
132                             plot = new Plot(plotID);
133                             System.out.println("Error in file parsing.");
134                         }
135                     } else {
136                         plot = new Plot(plotID, plotJSONObject.get("EVENT").toString());
137                     }
138                 } else if (plotJSONObject.size() == 10) {
139                     plot = new Plot(plotID, plotJSONObject.get("TITLE").toString(),
140                         Integer.parseInt(plotJSONObject.get("PRICE").toString()),
141                         Integer.parseInt(plotJSONObject.get("RENT").toString()),
142                         Integer.parseInt(plotJSONObject.get("U1").toString()),
143                         Integer.parseInt(plotJSONObject.get("U2").toString()),
144                         Integer.parseInt(plotJSONObject.get("U3").toString()),
145                         Integer.parseInt(plotJSONObject.get("U4").toString()),
146                         Integer.parseInt(plotJSONObject.get("U5").toString(),
147                         plotJSONObject.get("DISTRICT").toString()));
148                 } else {
149                     plot = new Plot(plotID);
150                     System.out.println("Error in file parsing.");
151                 }
152                 plotsForBoard.add(plot);
153             }
154         } catch (ParseException | IOException e) {
155             e.printStackTrace();
156         }
157         Board board = new Board();

```

```
158         board.plotsOnBoard.addAll(plotsForBoard);
159     return board;
160 }
161
162 }
163
```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.scene.control.Button?>
4 <?import javafx.scene.control.CheckBox?>
5 <?import javafx.scene.control.ComboBox?>
6 <?import javafx.scene.control.TextField?>
7 <?import javafx.scene.layout.AnchorPane?>
8
9 <AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
   minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://
   javafx.com/javafx/11.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="
   GameCreator.GameCreator">
10    <children>
11       <ComboBox fx:id="maxNumberOfRounds" layoutX="60.0" layoutY="275.0"
12          maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-
13          Infinity" onAction="#updateItems" onMouseClicked="#updateItems" prefHeight="
14          25.0" prefWidth="150.0" promptText="Max number of rounds" />
15       <ComboBox fx:id="boardOfChoice" layoutX="60.0" layoutY="310.0"
16          maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-
17          Infinity" onAction="#updateItems" onMouseClicked="#updateItems" prefHeight="
18          25.0" prefWidth="150.0" promptText="Board" />
19       <CheckBox fx:id="auctionIfNotBought" layoutX="350.0" layoutY="60.0"
20          mnemonicParsing="false" text="Auction if not bought" />
21       <CheckBox fx:id="fastStart" layoutX="350.0" layoutY="80.0"
22          mnemonicParsing="false" text="Fast start" />
23       <ComboBox fx:id="numberOfPlayers" layoutX="60.0" layoutY="60.0"
24          maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-
25          Infinity" onMouseClicked="#updateItems" prefHeight="25.0" prefWidth="150.0"
26          promptText="Number of players" />
27       <CheckBox fx:id="endOnBankruptcy" layoutX="350.0" layoutY="100.0"
28          mnemonicParsing="false" text="End on bankruptcy" />
29       <CheckBox fx:id="endOnOverwhelmingWealth" layoutX="350.0" layoutY="120.
30          0" mnemonicParsing="false" text="End on overwhelming wealth" />
31       <CheckBox fx:id="differingInheritance" layoutX="350.0" layoutY="140.0"
32          mnemonicParsing="false" text="Differing inheritance" />
33       <Button fx:id="startNewGame" layoutX="350.0" layoutY="310.0" maxHeight
34          ="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
35          mnemonicParsing="false" onAction="#startNewGameAction" prefHeight="25.0"
36          prefWidth="150.0" text="Start New Game" />
37       <TextField fx:id="namePlayer1" layoutX="60.0" layoutY="90.0" maxHeight
38          ="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
39          prefHeight="25.0" prefWidth="150.0" text="Name of Player 1" />
40       <TextField fx:id="namePlayer6" layoutX="60.0" layoutY="240.0" maxHeight
41          ="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
42          prefHeight="25.0" prefWidth="150.0" text="Name of Player 6" />
43       <TextField fx:id="namePlayer5" layoutX="60.0" layoutY="210.0" maxHeight
44          ="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
45          prefHeight="25.0" prefWidth="150.0" text="Name of Player 5" />
46       <TextField fx:id="namePlayer4" layoutX="60.0" layoutY="180.0" maxHeight
47          ="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
48          prefHeight="25.0" prefWidth="150.0" text="Name of Player 4" />
49       <TextField fx:id="namePlayer3" layoutX="60.0" layoutY="150.0" maxHeight
50          ="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
51          prefHeight="25.0" prefWidth="150.0" text="Name of Player 3" />
52       <TextField fx:id="namePlayer2" layoutX="60.0" layoutY="120.0" maxHeight
53          ="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
54          prefHeight="25.0" prefWidth="150.0" text="Name of Player 2" />
55       <CheckBox fx:id="randomizeTurnOrder" layoutX="350.0" layoutY="160.0"
56          mnemonicParsing="false" text="Randomize turn order" />
57    </children>

```

```
28 </AnchorPane>
```

```
29
```

```

1 package GameCreator;
2
3 import DataHandler.JSONImporter;
4 import Game.GUI;
5 import Game.GameMaster;
6 import Game.Player;
7 import javafx.collections.FXCollections;
8 import javafx.collections.ObservableList;
9 import javafx.event.ActionEvent;
10 import javafx.fxml.FXML;
11 import javafx.fxml.FXMLLoader;
12 import javafx.scene.Scene;
13 import javafx.scene.control.*;
14 import javafx.scene.layout.Pane;
15 import javafx.stage.Stage;
16 import java.io.IOException;
17
18 public class GameCreator {
19     @FXML
20     private ComboBox number0fPlayers;
21     @FXML
22     private TextField namePlayer1;
23     @FXML
24     private TextField namePlayer2;
25     @FXML
26     private TextField namePlayer3;
27     @FXML
28     private TextField namePlayer4;
29     @FXML
30     private TextField namePlayer5;
31     @FXML
32     private TextField namePlayer6;
33     @FXML
34     private ComboBox maxNumber0fRounds;
35     @FXML
36     private ComboBox board0fChoice;
37     @FXML
38     private CheckBox auctionIfNotBought;
39     @FXML
40     private CheckBox fastStart;
41     @FXML
42     private CheckBox endOnBankruptcy;
43     @FXML
44     private CheckBox endOnOverwhelmingWealth;
45     @FXML
46     private CheckBox differingInheritance;
47     @FXML
48     private CheckBox randomizeTurnOrder;
49     @FXML
50     private Button startNewGame;
51
52     public GameCreator() {
53     }
54
55     public void updateItems() {
56         ObservableList playerNumber = FXCollections.observableArrayList("3",
57             "4", "5", "6");
58         this.number0fPlayers.setItems(playerNumber);
59         ObservableList number0fRounds = FXCollections.observableArrayList("0",
60             "10", "15", "20", "25");

```

```

59         this.maxNumberOfRounds.setItems(numberOfRounds);
60         ObservableList listOfBoards = FXCollections.observableArrayList("Matador");
61         this.boardOfChoice.setItems(listOfBoards);
62     }
63
64     @FXML
65     private void startNewGameAction(ActionEvent event) {
66         if (maxNumberOfRounds.getValue() != null && boardOfChoice.getValue()
67             () != null && numberOfPlayers != null) {
68             GameMaster theGame = new GameMaster(Integer.parseInt((String)
69             maxNumberOfRounds.getValue()), (String) boardOfChoice.getValue(),
70             auctionIfNotBought.isSelected(), fastStart.isSelected()
71             (), endOnBankruptcy.isSelected(),
72             endOnOverwhelmingWealth.isSelected(),
73             differingInheritance.isSelected(),
74             randomizeTurnOrder.isSelected());
75             JSONImporter jsonImporter = new JSONImporter();
76             theGame.board = jsonImporter.createBoardJSON((String)
77             boardOfChoice.getValue());
78             theGame.chanceDeck = jsonImporter.createChanceDeckJson((String)
79             boardOfChoice.getValue());
80             Player player1 = new Player(namePlayer1.getCharacters().toString
81             (), 0);
82             theGame.players.add(player1);
83             Player player2 = new Player(namePlayer2.getCharacters().toString
84             (), 1);
85             theGame.players.add(player2);
86             Player player3 = new Player(namePlayer3.getCharacters().toString
87             (), 2);
88             theGame.players.add(player3);
89             if (Integer.parseInt((String) numberOfPlayers.getValue()) > 3) {
90                 Player player4 = new Player(namePlayer4.getCharacters().
91                 toString(), 3);
92                 theGame.players.add(player4);
93                 if (Integer.parseInt((String) numberOfPlayers.getValue())
94                     > 4) {
95                     Player player5 = new Player(namePlayer5.getCharacters().
96                     toString(), 4);
97                     theGame.players.add(player5);
98                     if (Integer.parseInt((String) numberOfPlayers.getValue()
99                     ()) > 5) {
100                         Player player6 = new Player(namePlayer6.
101                         getCharacters().toString(), 5);
102                         theGame.players.add(player6);
103                     }
104                 }
105             }
106             if (theGame.randomizeTurnOrder) {
107                 theGame.randomizeTurnOrder();
108             }
109             try {
110                 Stage gameCreatorStage = new Stage();
111                 FXMLLoader loader = new FXMLLoader();
112                 Pane root = (Pane) loader.load(getClass().getResource("/Game
113                 /GUI.fxml").openStream());
114                 GUI GUI = (GUI) loader.getController();
115                 GUI.theGame = theGame;
116                 GUI.setupPlayers();
117                 GUI.setupBoard();
118             }
119         }
120     }

```

```
103         GUI.updatePlayerStatistics();
104         GUI.updateAllPlayerTokens();
105         Scene scene = new Scene(root);
106         gameCreatorStage.setScene(scene);
107         gameCreatorStage.setTitle("Landlord's Game");
108         gameCreatorStage.setResizable(false);
109         gameCreatorStage.show();
110     } catch (IOException e) {
111         e.printStackTrace();
112     }
113 }
114 }
115 }
116 }
```

```

1 package TestFeature;
2
3 import Game.GameMaster;
4 import Game.Player;
5 import Game.Plots.Plot;
6 import javafx.scene.control.Alert;
7 import javafx.scene.control.ButtonType;
8 import javafx.scene.control.Label;
9 import javafx.stage.Stage;
10
11 import java.util.HashMap;
12 import java.util.Optional;
13
14 public class AsgerTemp {
15
16     public GameMaster theGame;
17
18     public void jailFunctions(Stage primaryStage) throws Exception {
19         Label label;
20         Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
21         alert.setTitle("Get out of jail options");
22         alert.setHeaderText("Choose options for how to get out of jail");
23         alert.setContentText("I hope we don't see you in here again!");
24
25         Player player = theGame.players.get(theGame.turnOrder);
26         Plot plot = theGame.board.plotsOnBoard.get(player.boardPosition);
27         HashMap<Integer, Integer> ownerships = theGame.board.ownerships;
28
29         ButtonType jailRollDice = new ButtonType("Roll dice");
30         ButtonType getOutToken = new ButtonType("Use token");
31         ButtonType payOption = new ButtonType("Pay 1000");
32
33         alert.getButtonTypes().clear();
34
35         alert.getButtonTypes().addAll(jailRollDice, getOutToken, payOption);
36
37         Optional<ButtonType> result = alert.showAndWait();
38         if (payOption.getText() == "pay 1000") {
39             player.wealth = player.wealth - 1000;
40             theGame.throwDiceToMove(theGame.players.get(theGame.turnOrder));
41             //make the user pay 1000 from player.wealth
42         } else if (jailRollDice.getText() == "Roll dice") {
43             for (int i = 0; i < 3; i++) {
44                 theGame.die1.roll();
45                 theGame.die2.roll();
46                 if (theGame.die1.result == theGame.die2.result) {
47                     theGame.players.get(theGame.turnOrder).boardPosition =
48 + theGame.die1.result + theGame.die2.result;
49                     break;
50                 }
51             } else {
52                 theGame.endTurn();
53             }
54         }
55     }
56 }
57
58
59

```

```

60 /*
61
62
63
64
65
66
67 // @Override
68 public void confirmationDialog(Stage primaryStage) throws Exception {
69     Alert alert = new Alert(Alert.AlertType.CONFIRMATION);
70     alert.setTitle("Confirmation Alert");
71     alert.setHeaderText("Do you wish to buy this property?");
72     alert.setContentText("Are you sure?");
73
74     Optional<ButtonType> result = alert.showAndWait();
75     if (result.get() == ButtonType.OK) {
76         // user picks okay
77     } else {
78         // user picks cancel
79     }
80     Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));
81     primaryStage.setTitle("new button test");
82     primaryStage.setScene(new Scene(root, 300, 275));
83     primaryStage.show();
84 }
85
86
87 // @Override
88 public void payDialog(Stage primaryStage) throws Exception {
89     Alert alert1 = new Alert(Alert.AlertType.INFORMATION);
90     alert1.setTitle("Payment Alert");
91     alert1.setHeaderText("You have landed on a property, pay up!"); // evt indsæt beløb der skal betales
92
93     Optional<ButtonType> result = alert1.showAndWait();
94     if (result.get() == ButtonType.OK) {
95
96     } else {
97
98     }
99 }
100
101
102 // @Override
103 public void LostDialog(Stage primaryStage) throws Exception {
104     Alert alert2 = new Alert(Alert.AlertType.INFORMATION);
105     alert2.setTitle("Information Alert");
106     alert2.setHeaderText("You have lost");
107     alert2.setContentText("Better Luck next time =)");
108
109     Optional<ButtonType> result = alert2.showAndWait();
110     if(result.get() == ButtonType.OK) {
111
112     } else {
113     }
114 }
115 // @Override
116 public void chanceDialog(Stage primaryStage) throws Exception {
117     Alert alert3 = new Alert(Alert.AlertType.INFORMATION);

```

```
118     alert3.setTitle("You have landed on a chance card!");
119     alert3.setHeaderText(""); //mulighed for at insertere chance card text
120     her
121     Optional<ButtonType> result = alert3.showAndWait();
122     if (result.get() == ButtonType.OK) {
123         } else {
124         }
125     */
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import java.lang.*?>
4 <?import java.util.*?>
5 <?import javafx.scene.*?>
6 <?import javafx.scene.control.*?>
7 <?import javafx.scene.layout.*?>
8
9 <AnchorPane xmlns="http://javafx.com/javafx"
10      xmlns:fx="http://javafx.com/fxml"
11      fx:controller="LoadGameInterface.LoadGameScreen"
12      prefHeight="400.0" prefWidth="600.0">
13
14 </AnchorPane>
15
```

```
1 package LoadGameInterface;  
2  
3 public class LoadGameScreen {  
4 }  
5
```