

```
1 package com.example.demo;
2
3 public class Weapon {
4     private String name;
5     private int attack;
6     private int agility;
7     private int defense;
8     private int luck;
9     private String image;
10    private int level;
11    private int cost;
12
13    public Weapon(String n, String i, int l, int at
14 , int ag, int d, int lk, int c){
15         name = n;
16         image = i;
17         level = l;
18         attack = at;
19         agility = ag;
20         defense = d;
21         luck = lk;
22         cost = c;
23     }
24
25     public String getName() {
26         return name;
27     }
28
29     public int getAttack() {
30         return attack;
31     }
32
33     public int getAgility() {
34         return agility;
35     }
36
37     public int getDefense() {
38         return defense;
39     }
40     public int getLuck() {
```

```
41         return luck;
42     }
43
44     public String getImage() {
45         return image;
46     }
47
48     public int getCost() {
49         return cost;
50     }
51
52     public int getLevel() {
53         return level;
54     }
55
56     public void setName(String name) {
57         this.name = name;
58     }
59
60     public void setAttack(int attack) {
61         this.attack = attack;
62     }
63
64     public void setImage(String image) {
65         this.image = image;
66     }
67
68     public void setLuck(int luck) {
69         this.luck = luck;
70     }
71
72     public void setAgility(int agility) {
73         this.agility = agility;
74     }
75
76     public void setCost(int cost) {
77         this.cost = cost;
78     }
79
80     public void setDefense(int defense) {
81         this.defense = defense;
```

```
82      }
83 }
84
```

```
1 package com.example.demo;
2
3 import java.util.ArrayList;
4
5 public class Character {
6     private String name;
7     private int totalHP;
8     private int hp;
9     private int level;
10    private int money;
11    private int defense;
12    private int attack;
13    private int agility;
14    private int luck;
15    private String card;
16    private ArrayList<Weapon> weapons = new
17        ArrayList<>();
18    private Weapon chosenWeapon;
19
20    public Character(String n, int h, int r, int at
21 , int ag, int l, String c, int lev){
22         name = n;
23         hp = h;
24         totalHP = h;
25         defense = r;
26         attack = at;
27         agility = ag;
28         luck = l;
29         card = c;
30         level = lev;
31         money = 50;
32     }
33
34 //             if(opponent.equals("Spades")){
35 //                 enemyCharacter.setAttack(randomNumber(
36 //                     where+1) * 50 - 20, (where+1) * 50 + 20));
37 //                 enemyCharacter.setDefense(randomNumber(
38 //                     where * 50 - 20, where * 50 + 20));
39 //                 enemyCharacter.setAgility(randomNumber(
40 //                     where * 50 - 20, where * 50 + 20));
41 //                 enemyCharacter.setLuck(randomNumber(10
```

```
36 + (where - 1)* 5 - 2, 10 + (where - 1) * 5 + 2));  
37 // } else if(opponent.equals("Hearts")){  
38 //     enemyCharacter.setDefense(randomNumber((  
39 //         where+1) * 50 - 20, (where+1) * 50 + 20));  
40 //     enemyCharacter.setAttack(randomNumber(  
41 //         where * 50 - 20, where * 50 + 20));  
42 //     enemyCharacter.setAgility(randomNumber(  
43 //         where * 50 - 20, where * 50 + 20));  
44 //     enemyCharacter.setLuck(randomNumber(10  
45 //         + (where - 1)* 5 - 2, 10 + (where - 1) * 5 + 2));  
46 // } else if(opponent.equals("Clubs")){  
47 //     enemyCharacter.setAgility(randomNumber((  
48 //         where+1) * 50 - 20, (where+1) * 50 + 20));  
49 //     enemyCharacter.setDefense(randomNumber(  
50 //         where * 50 - 20, where * 50 + 20));  
51 //     enemyCharacter.setAttack(randomNumber(  
52 //         where * 50 - 20, where * 50 + 20));  
53 //     enemyCharacter.setLuck(randomNumber(10 +  
54 //         where* 5 - 2, 10 + where * 5 + 2));  
55 // }  
56 //  
57 //  
58 public int getHp() {  
59     return hp;  
60 }  
61 //  
62 public int getTotalHP() {  
63     return totalHP;  
64 }
```

```
65
66     public int getLevel() {
67         return level;
68     }
69
70     public int getMoney() {
71         return money;
72     }
73
74     public int getDefense() {
75         return defense;
76     }
77
78     public int getAttack() {
79         return attack;
80     }
81
82     public int getAgility() {
83         return agility;
84     }
85
86     public int getLuck() {
87         return luck;
88     }
89
90     public String getCard() {
91         return card;
92     }
93
94     public ArrayList<Weapon> getWeapons(){
95         return weapons;
96     }
97
98     public Weapon getWeapon(int i){
99         return weapons.get(i);
100    }
101
102    public Weapon getChosenWeapon() {
103        return chosenWeapon;
104    }
105
```

```
106     //      public String getWeaponName(int i){  
107 //          return weapons.get(i).getName();  
108 //      }  
109  
110 //      public int getWeaponAttack(int i){  
111 //          return weapons.get(i).getAttack();  
112 //      }  
113 //  
114 //      public int getWeaponAgility(int i){  
115 //          return weapons.get(i).getAgility();  
116 //      }  
117 //  
118 //      public int getWeaponDefense(int i) {  
119 //          return weapons.get(i).getDefense();  
120 //      }  
121 //  
122 //      public int getWeaponLuck(int i){  
123 //          return weapons.get(i).getLuck();  
124 //      }  
125 //  
126 //      public String getWeaponImage(int i){  
127 //          return weapons.get(i).getImage();  
128 //      }  
129  
130 //      public int getWeaponLevel(int i){  
131 //          return weapons.get(i).getLevel();  
132 //      }  
133  
134     public void addWeapon(Weapon w) {  
135         this.weapons.add(w);  
136     }  
137  
138     public void setWeapons(ArrayList<Weapon> w) {  
139         this.weapons = w;  
140     }  
141  
142     public void setWeapon(Weapon w, int i){  
143         this.weapons.set(i, w);  
144     }  
145  
146     public void setChosenWeapon(Weapon
```

```
146 chosenWeapon) {  
147     this.chosenWeapon = chosenWeapon;  
148 }  
149  
150 //     public void setWeaponImage(int i, String  
151 //     s){  
151 //         this.weapons.get(i).setImage(s);  
152 //     }  
153  
154  
155     public void setName(String name) {  
156         this.name = name;  
157     }  
158  
159     public void setHp(int hp) {  
160         this.hp = hp;  
161     }  
162  
163     public void setTotalHP(int totalHP) {  
164         this.totalHP = totalHP;  
165         this.hp = totalHP;  
166     }  
167  
168     public void setLevel(int level) {  
169         this.level = level;  
170     }  
171  
172     public void setMoney(int money) {  
173         this.money = money;  
174     }  
175  
176     public void setDefense(int defense) {  
177         this.defense = defense;  
178     }  
179  
180     public void setAttack(int attack) {  
181         this.attack = attack;  
182     }  
183  
184     public void setAgility(int agility) {  
185         this.agility = agility;
```

```
186      }
187
188      public void setLuck(int luck) {
189          this.luck = luck;
190      }
191
192      public void setCard(String card) {
193          this.card = card;
194      }
195
196  }
197
```

```
1 package com.example.demo;
2
3 import javafx.event.ActionEvent;
4 import javafx.fxml.FXML;
5 import javafx.scene.control.*;
6 import javafx.scene.image.Image;
7 import javafx.scene.image.ImageView;
8 import javafx.scene.input.MouseEvent;
9 import javafx.scene.layout.AnchorPane;
10
11 import java.io.*;
12 import java.lang.reflect.Array;
13 import java.nio.file.Files;
14 import java.nio.file.Paths;
15 import java.util.ArrayList;
16 import java.util.Locale;
17 import java.util.Scanner;
18
19 //I deserve an A because I gave the incorporated
20 //concepts that we have used in past projects such as
21 //a timer and automatically updating based on user
22 //textfield. i also
23 //I also simplified my code by calling the same
24 //function sometimes but chnaging the output thorugh
25 //parameters. I also called
26 //the same function in scenebuilder through buttons
27 //and then used action events in order to
28 //differentiate.
29
30
31 public class HelloController {
32     @FXML
33     private Label introText, cardInfo, infoText,
34     displayImprove, costText, errorText, rewardText,
35     infoText2, oppError, playerStats, enemyStats,
36     nameMC, nameEnemy, turnText, rewardsList,
37     weaponStats, weaponLabel, weaponError, weaponStats2
38 ;
39     @FXML
40     private TextField nameText, pointsText;
41     @FXML
```

```
30     private ImageView city, second, mainAvatar,
      battleIcon, mainAvatar2, battlePlayer, battleEnemy
      , mainRewards, backHome, weaponIcon, selectedWeapon
      ;
31     @FXML
32     private AnchorPane startingPane, storyPanel1,
      letterPane, chooseCardPane, chooseAvatarPane,
      characterPane, sidePane, chooseOpponentPane,
      battlePane, rewardPane, weaponPane, weaponInfo,
      sideWeapon, gameOver;
33     @FXML
34     private Button chooseButton, collectRewards,
      buyButton;
35     @FXML
36     private ListView statsList, weaponsList;
37     @FXML
38     private ProgressBar mainHP, enemyHP, xp;
39
40
41     int sceneCount = 0;
42     String[] types = {"Axe", "Crossbow", "Hammer",
      "Mace", "Shield", "Spear", "Sword"};
43     ArrayList<Weapon> allWeapons = new ArrayList
      <>();
44     ArrayList<String> weaponNames = new ArrayList
      <>();
45     ArrayList<Character> allEnemies = new ArrayList
      <>();
46     //Axe: +15 Luck & -5 Agility
47     //Crossbow: +15 Attack & +20 Agility
48     //Hammer: +10 Attack & +5 Luck & -5 Agility
49     //Mace: +10 Attack & -5 Defense & +10 Luck
50     //Shield: +25 Defense & +5 Luck
51     //Spear: +5 Attack & +10 Agility
52     //Sword: +20 Attack & +15 Agility
53     String mainImage = "src/main/resources/Avatars/
      pinkGirl.png";
54     Character mainCharacter;
55     Character enemyCharacter;
56     String enemyImage;
57     Weapon showWeapon;
```

```
58     int classChosen = 1;
59     int visa = 3;
60     int where = 0;
61     boolean playerTurn = true;
62     boolean playerWon = false;
63     String opponent;
64     long visaTime;
65
66     String[] images = {"aquaGirl", "blackBoy", "blondeBoy", "blondeBoy2", "blueBoy", "brownBoy", "cyanBoy", "cyanGirl", "grayBoy", "greenBoy", "greenGirl1", "greenGirl2", "hairGirl", "magentaGirl", "purpleBoy", "pinkGirl", "purpleGirl", "redBoy", "redGirl", "silverBoy", "violetGirl", "whiteBoy", "whiteGirl"};
67
68     public HelloController(){
69         //runs for every element in the type of weapons array
70         for (int x = 0; x < 7; x++){
71             //runs for ever possible weapon level
72             for(int i = 1; i < 5; i++){
73                 String image = "src/main/resources/weapons/" + types[x].toLowerCase(Locale.ROOT) + "Level" + i + ".png";
74                 switch (x){
75                     case 0: allWeapons.add(new Weapon(types[x], image, i, 0, -5*i, 0, 15*i, 50*i));
76                     break;
77                     case 1: allWeapons.add(new Weapon(types[x], image, i, 15*i, 20*i, 0, 0, 70*i));
78                     break;
79                     case 2: allWeapons.add(new Weapon(types[x], image, i, 10*i, -5*i, 0, 5*i, 20*i));
80                     break;
81                     case 3: allWeapons.add(new Weapon(types[x], image, i, 10*i, 0, -5*i, 10*i, 40*i));
82                     break;
83                     case 4: allWeapons.add(new Weapon(types[x], image, i, 10*i, 0, 0, 10*i, 30*i));
84                     break;
85                     case 5: allWeapons.add(new Weapon(types[x], image, i, 10*i, 0, 0, 10*i, 30*i));
86                     break;
87                     case 6: allWeapons.add(new Weapon(types[x], image, i, 10*i, 0, 0, 10*i, 30*i));
88                     break;
89                 }
90             }
91         }
92     }
93 }
```

```

83 i));
84                                     break;
85             case 4: allWeapons.add(new
86             Weapon(types[x], image, i, 0, 0, 25*i, 5*i, 60*i
87             ));
88             break;
89             case 5: allWeapons.add(new
90             Weapon(types[x], image, i, 5*i, 10*i, 0, 0, 30*i
91             ));
92             break;
93             case 6: allWeapons.add(new
94             Weapon(types[x], image, i, 20*i, 15*i, 0, 0, 70*i
95             ));
96             break;
97         }
98     }
99
100    //runs for every possible level
101    for (int lvl = 1; lvl < 13; lvl++){
102        //makes 10 enemies of each card level
103        for(int c = 0; c < 10; c++){
104            //Spades
105            allEnemies.add(new Character("EnemY",
106                randomNumber(170 + (lvl - 1)*50, 230 + (
107                    lvl - 1)*50), randomNumber(lvl * 50 - 20, lvl * 50
108                    + 20), randomNumber((lvl+1) * 50 - 20, (lvl+1) *
109                    50 + 20), randomNumber(lvl * 50 - 20, lvl * 50 +

```

```

105 20), randomNumber(10 + (lvl - 1)* 5 - 2, 10 + (lvl
- 1) * 5 + 2), "Spades", lvl));
106                                //Hearts
107                                allEnemies.add(new Character("Enemy",
108                                randomNumber(170 + (lvl - 1)*50, 230 + (
109                                lvl - 1)*50), randomNumber((lvl+1) * 50 - 20, (lvl
+1) * 50 + 20), randomNumber(lvl * 50 - 20, lvl *
50 + 20), randomNumber(lvl * 50 - 20, lvl * 50 +
20), randomNumber(10 + (lvl - 1)* 5 - 2, 10 + (lvl
- 1) * 5 + 2), "Hearts", lvl));
110                                //Clubs
111                                allEnemies.add(new Character("Enemy",
112                                randomNumber(170 + (lvl - 1)*50, 230 + (
113                                lvl - 1)*50), randomNumber(lvl * 50 - 20, lvl * 50
+ 20), randomNumber(lvl * 50 - 20, lvl * 50 + 20
), randomNumber((lvl+1) * 50 - 20, (lvl+1) * 50 +
20), randomNumber(10 + (lvl - 1)* 5 - 2, 10 + (lvl
- 1) * 5 + 2), "Clubs", lvl));
114                                //Diamonds
115                                allEnemies.add(new Character("Enemy",
116                                randomNumber(170 + (lvl - 1)*50, 230 + (
117                                lvl - 1)*50), randomNumber(lvl * 50 - 20, lvl * 50
+ 20), randomNumber(lvl * 50 - 20, lvl * 50 + 20
), randomNumber(lvl * 50 - 20, lvl * 50 + 20),
randomNumber(10 + lvl* 5 - 2, 10 + lvl * 5 + 2), "
Diamonds", lvl));
118                            }
119                        }
120
121                        //Assigns all the weapons in certain level
122                        //to enemies in certain levels.
123                        for (Character enemy: allEnemies){
124                            for (Weapon w: allWeapons){
125                                if(enemy.getLevel() >=0 && enemy.
126                                getLevel() <= 3 && w.getLevel() == 1){
127                                    enemy.addWeapon(w);
128                                } else if (enemy.getLevel() >=4
129                                && enemy.getLevel() <= 6 && w.getLevel() == 2){
130                                    enemy.addWeapon(w);
131                                } else if (enemy.getLevel() >=7
132                                && enemy.getLevel() <= 9 && w.getLevel() == 3){

```

```
123                     enemy.addWeapon(w);
124             } else if (enemy.getLevel() >=10
125             && enemy.getLevel() <= 12 && w.getLevel() == 4){
126                 enemy.addWeapon(w);
127             }
128
129
130 //           System.out.print("Level: " + enemy.
131 //                           getLevel() + "Weapon Level: ");
132 //           for (int i = 0; i < enemy.getWeapons
133 //                           ().size(); i++){
134 //               System.out.print(enemy.getWeapon
135 //                               (i).getLevel() + " ");
136 //           }
137
138         mainCharacter = new Character("Siya", 200
139             , 50, 50, 50, 10, "Diamond", 1);
140         enemyCharacter = new Character("Enemy",
141             200, 50, 50, 50, 10, "Diamond", 1);
142
143     @FXML
144     public void startGame() {
145         startingPane.setVisible(false);
146         characterPane.setVisible(false);
147         storyPane1.setVisible(true);
148         visa = 3;
149         sceneCount = 0;
150         city.setVisible(true);
151         second.setVisible(false);
152         gameOver.setVisible(false);
153         mainCharacter = new Character("Siya", 200
154             , 50, 50, 50, 10, "Diamond", 1);
155
156     @FXML
```

```
157     public void loadGame() {
158         startingPane.setVisible(false);
159         handleLoad();
160         setCharacterPane();
161     }
162
163     @FXML
164     public void sceneOne() {
165         if (sceneCount == 0){
166             introText.setText("After a whole day
of pointlessly wandering around it seems that the
whole city has shut down. No electricity. No
people. No life. Suddenly in the pitch dark there
seems to be glowing yellow light coming from one
of the buildings. Thrilled to see some form of
light you make your way over to the building.");
167             try {
168                 FileInputStream input = new
FileInputStream("src/main/resources/buildingLit.
png");
169                 city.setImage(new Image(input));
170             } catch (Exception e) {
171                 e.printStackTrace();
172             }
173             sceneCount++;
174         } else if (sceneCount == 1){
175             city.setVisible(false);
176             second.setVisible(true);
177             introText.setText("As you enter the
building and approach the main desk you greeted
with a small envelope. Filled with curiosity you
open it and find a key card for Room 1.");
178             sceneCount++;
179         } else if (sceneCount == 2){
180             introText.setText("Opening the door to
Room 1, it seems to be an small empty room. In
the middle, there is a wooden table with all of
the 4 King of Cards lying next to them lays a
letter.");
181             try {
182                 FileInputStream input3 = new
```

```
182 FileInputStream("src/main/resources/desk.png");
183             second.setImage(new Image(input3
184         ));
185         } catch (Exception e) {
186             e.printStackTrace();
187         }
188         sceneCount++;
189     } else if (sceneCount == 3){
190         storyPanel1.setVisible(false);
191         letterPane.setVisible(true);
192         sceneCount++;
193     }
194
195     @FXML
196     public void createCharacter(){
197 //         mainCharacter = new Character(nameText.
198 //             getText(), 200, 50, 50, 50, 10, "Diamond", 1);
199         letterPane.setVisible(false);
200         chooseCardPane.setVisible(true);
201     }
202
203     @FXML
204     public void classSpades(){
205         chooseButton.setVisible(true);
206         cardInfo.setText("Class: Spade \nStats: +
207         50 Attack \nWeapon: Sword");
208         classChosen = 1;
209     }
210
211     @FXML
212     public void classHearts(){
213         chooseButton.setVisible(true);
214         cardInfo.setText("Class: Heart \nStats: +
215         50 Defense \nWeapon: Shield");
216         classChosen = 2;
217     }
218
219     @FXML
220     public void classClubs(){
221         chooseButton.setVisible(true);
```

```
219     cardInfo.setText("Class: Club \nStats: +50  
220         Agility \nWeapon: Spear");  
221         classChosen = 3;  
222     }  
223  
224     @FXML  
225     public void classDiamonds(){  
226         chooseButton.setVisible(true);  
227         cardInfo.setText("Class: Diamond \nStats  
228 : +10 Luck \nWeapon: Crossbow");  
229         classChosen = 4;  
230     }  
231  
232     @FXML  
233     public void setClassChosen(){  
234         if(classChosen == 1){  
235             mainCharacter.setCard("Spades");  
236             mainCharacter.setAttack(mainCharacter.  
getAttack() + 50);  
237             mainCharacter.addWeapon(allWeapons.get  
(24));  
238         } else if (classChosen == 2){  
239             mainCharacter.setCard("Hearts");  
240             mainCharacter.setDefense(mainCharacter  
.getDefense() + 50);  
241             mainCharacter.addWeapon(allWeapons.get  
(16));  
242         } else if (classChosen == 3){  
243             mainCharacter.setCard("Clubs");  
244             mainCharacter.setAgility(mainCharacter  
.getAgility() + 50);  
245             mainCharacter.addWeapon(allWeapons.get  
(20));  
246         } else if (classChosen == 4){  
247             mainCharacter.setCard("Diamonds");  
248             mainCharacter.setLuck(mainCharacter.  
getLuck() + 10);  
249             mainCharacter.addWeapon(allWeapons.get  
(4));  
250         }  
251         chooseCardPane.setVisible(false);
```

```
250         chooseAvatarPane.setVisible(true);
251     }
252
253     @FXML
254     public void avatarClicked(MouseEvent t){
255         mainImage = "src/main/resources/Avatars/"
256         + t.getPickResult().getIntersectedNode().getId()
257         () + ".png";
258         System.out.println(mainImage);
259 //        try {
260 //            String cssDefault = "-fx-border-
261 //color: blue; \n" + "-fx-border-insets: 5; \n" + "-fx
262 //border-width: 3; \n" + "-fx-border-style: dashed; \n";
263 //            brownBoy.setStyle(cssDefault);
264 //        } catch(Exception e) {
265 //            System.out.println(errorText);
266 //        }
267 //        t.getPickResult().getIntersectedNode().
268 //        getId()
269
270     }
271
272     @FXML
273     public void setCharacterPane(){
274         battlePane.setVisible(false);
275         chooseAvatarPane.setVisible(false);
276         characterPane.setVisible(true);
277         visaTimer();
278
279         try {
280             FileInputStream input = new
281             FileInputStream(mainImage);
282             mainAvatar.setImage(new Image(input));
283             System.out.println(mainAvatar);
284         } catch (Exception e) {
285             e.printStackTrace();
286         }
287
288         updateList(statsList);
289
290     }
```

```

284         infoText.setText("Level: " + mainCharacter
285             .getLevel() + "\nMoney: $" + mainCharacter.
286             getMoney()
287             + "\nVisa: " + Integer.
288             toString(visa) + " Days");
289
290     }
291
292     @FXML
293     public void checkLevel(){
294         int total = mainCharacter.getTotalHP() +
295             mainCharacter.getAgility() + mainCharacter.
296             getAttack() + mainCharacter.getDefense();
297         if (total >= 200 * (mainCharacter.getLevel
298             ()+1) + 100){
299             mainCharacter.setLevel(mainCharacter.
300             getLevel() + 1);
301
302             //finding the number of xp needed to reach
303             //the next level using the difference of the goal
304             //for next level
305             //and current xp over 200 because 200 is
306             //the interval between each cutting mark
307             xp.setProgress(1 - ((200 * (mainCharacter.
308             getLevel()+1) + 100) - total)/200.0);
309         }
310
311     }
312
313     @FXML
314     public void updateList(ListView list){
315         list.getItems().clear();
316         list.getItems().add(mainCharacter.getName
317             ().toUpperCase(Locale.ROOT));
318         list.getItems().add("Class: " +
319             mainCharacter.getCard());
320         list.getItems().add("HP: " + mainCharacter
321             .getTotalHP());
322         list.getItems().add("Attack: " +
323             mainCharacter.getAttack());
324         list.getItems().add("Defense: " +
325             mainCharacter.getDefense());

```

```
309         list.getItems().add("Agility: " +
310             mainCharacter.getAgility());
310         list.getItems().add("Luck: " +
311             mainCharacter.getLuck());
311         list.getItems().add("WEAPON");
312
313         for(int i = 0; i <mainCharacter.getWeapons
314             ().size(); i++){
314             list.getItems().add("Level " +
315                 mainCharacter.getWeapon(i).getLevel() + " " +
316                 mainCharacter.getWeapon(i).getName());
315         }
316     }
317
318     @FXML
319     public void changeStats(MouseEvent x) throws
320         FileNotFoundException {
320         where = statsList.getSelectionModel().
321             getSelectedIndex();
321         sideWeapon.setVisible(false);
322         sidePane.setVisible(false);
323
324         if (where == 2) {
325             sidePane.setVisible(true);
326             displayImprove.setText("Improve HP
327 Stat");
327     } else if (where == 3) {
328         sidePane.setVisible(true);
329         displayImprove.setText("Improve
329 Attack Stat");
330     } else if (where == 4) {
331         sidePane.setVisible(true);
332         displayImprove.setText("Improve
332 Defense Stat");
333     } else if (where == 5) {
334         sidePane.setVisible(true);
335         displayImprove.setText("Improve
335 Agility Stat");
336     } else if (where == 6) {
337         sidePane.setVisible(true);
338         displayImprove.setText("Improve Luck
```

```

338 Stat");
339         } else if (where > 7) {
340             sideWeapon.setVisible(true);
341             showWeapon = mainCharacter.getWeapons
342                 ().get(where - 8);
343             FileInputStream input = new
344                 FileInputStream(showWeapon.getImage());
345             selectedWeapon.setImage(new Image(
346                 input));
347             weaponStats2.setText("Attack: " +
348                 showWeapon.getAttack() + "\nAgility: " +
349                 showWeapon.getAgility() + "\nDefense: " +
350                 showWeapon.getDefense() + "\nLuck: " + showWeapon.
351                 getLuck());
352             }
353             displayImprove.setText(displayImprove.
354                 getText() + "\nEach Stat Point Costs 5 Dollar");
355         }
356     }
357     @FXML
358     public void updateCost(){
359         try{
360             int points = 5 * Integer.parseInt(
361                 pointsText.getText());
362             costText.setText("Cost: $" + String.
363                 valueOf(points));
364         } catch(NumberFormatException e){
365             costText.setText("Cost: Enter an
366                 Integer");
367         }
368     }
369     @FXML
370     public void upgradeStats(){
371         int points = 5 * Integer.parseInt(
372             pointsText.getText());
373         if (mainCharacter.getMoney() >= points){
374             mainCharacter.setMoney(mainCharacter.
375                 getMoney() - points);
376             switch (where) {

```

```

366             case 2: mainCharacter.setTotalHP(
367                 mainCharacter.getTotalHP() + points/5);
368             break;
369             case 3: mainCharacter.setAttack(
370                 mainCharacter.getAttack() + points/5);
371             break;
372             case 4: mainCharacter.setDefense(
373                 mainCharacter.getDefense() + points/5);
374             break;
375             case 5: mainCharacter.setAgility(
376                 mainCharacter.getAgility() + points/5);
377             break;
378             case 6: mainCharacter.setLuck(
379                 mainCharacter.getLuck() + points/5);
380             break;
381         }
382     }
383     @FXML
384     public void equipWeapon(){
385         mainCharacter.setChosenWeapon(showWeapon);
386     }
387
388     @FXML
389     public void pickOpponent(){
390         battlePane.setVisible(false);
391         characterPane.setVisible(false);
392         chooseOpponentPane.setVisible(true);
393         try {
394             FileInputStream input = new
395             FileInputStream(mainImage);
396             mainAvatar2.setImage(new Image(input
397             ));
398         } catch (Exception e) {
399             e.printStackTrace();
400         }
401     }

```

```
399         infoText2.setText("Level: " +
    mainCharacter.getLevel() + "\nClass: " +
    mainCharacter.getCard()
400                     + "\nVisa: " + Integer.toString(
    visa) + " Days");
401     }
402
403     @FXML
404     public void spadesOpp(){
405         rewardText.setText("Attack: +" +
    mainCharacter.getLevel()*10 + "\nVisa: +" +
    mainCharacter.getLevel() + " Days" + "\nMoney: +$" +
    + mainCharacter.getLevel()*10);
406         opponent = "Spades";
407     }
408
409     @FXML
410     public void diamondOpp(){
411         rewardText.setText("Luck: +" +
    mainCharacter.getLevel()*10 + "\nVisa: +" +
    mainCharacter.getLevel() + " Days" + "\nMoney: +$" +
    + mainCharacter.getLevel()*10);
412         opponent = "Diamonds";
413     }
414
415     @FXML
416     public void heartOpp(){
417         rewardText.setText("Defense: +" +
    mainCharacter.getLevel()*10 + "\nVisa: +" +
    mainCharacter.getLevel() + " Days" + "\nMoney: +$" +
    + mainCharacter.getLevel()*10);
418         opponent = "Hearts";
419     }
420
421     @FXML
422     public void clubOpp(){
423         rewardText.setText("Agility: +" +
    mainCharacter.getLevel()*10 + "\nVisa: +" +
    mainCharacter.getLevel() + " Days" + "\nMoney: +$" +
    + mainCharacter.getLevel()*10);
424         opponent = "Clubs";
```

```
425      }
426
427      @FXML
428      public void oppChosen(){
429          boolean lookingOpp = true;
430          //This is generating random enemies from
431          //the enemies array until the enemy and player have
432          //the same level and the enemy has the selected
433          //class
434          while(lookingOpp){
435              enemyCharacter = allEnemies.get(
436                  randomNumber(0, allEnemies.size()-1));
437              enemyCharacter.setChosenWeapon(
438                  enemyCharacter.getWeapon(randomNumber(0,
439                  enemyCharacter.getWeapons().size()-1)));
440              if (enemyCharacter.getCard().equals(
441                  opponent) && enemyCharacter.getLevel() ==
442                  mainCharacter.getLevel())){
443                  lookingOpp = false;
444              }
445          }
446          enemyImage = "src/main/resources/Avatars/"
447          + images[randomNumber(0, images.length-1)] + ".
448          png";
449 //          System.out.println(enemyCharacter.
450 //              getTotalHP());
451          chooseOpponentPane.setVisible(false);
452          battlePane.setVisible(true);
453
454          //This is selecting a random weapon for
455          //the player if they selected battle without picking
456          //a weapon
457          if(mainCharacter.getChosenWeapon() == null
458          ){
459              mainCharacter.setChosenWeapon(
460                  mainCharacter.getWeapon(randomNumber(0,
461                  mainCharacter.getWeapons().size()-1)));
462          }
463          try {
464              FileInputStream input = new
465              FileInputStream(mainImage);
```

```

449         battlePlayer.setImage(new Image(input
    ));
450         FileInputStream enemy = new
    FileInputStream(enemyImage);
451         battleEnemy.setImage(new Image(enemy
    )));
452     } catch (Exception e) {
453         e.printStackTrace();
454     }
455         nameMC.setText(mainCharacter.getName() +
    "Level: " + mainCharacter.getLevel());
456         nameEnemy.setText(enemyCharacter.getName
    () + "Level: " + enemyCharacter.getLevel());
457         turnText.setText("Turn: " + mainCharacter.
    getName());
458         enemyCharacter.setHp(enemyCharacter.
    getTotalHP());
459         mainCharacter.setHp(mainCharacter.
    getTotalHP());
460         updateStats();
461
462     }
463
464     @FXML
465     public void updateStats(){
466         try {
467             playerStats.setText("Class: " +
    mainCharacter.getCard() + "\nAttack: " +
    mainCharacter.getAttack() + "\nDefense: " +
    mainCharacter.getDefense() + "\nAgility: " +
    mainCharacter.getAgility() + "\nLuck: " +
    mainCharacter.getLuck() + "\nLevel " +
    mainCharacter.getChosenWeapon().getLevel() + " "
    + mainCharacter.getChosenWeapon().getName());
468             enemyStats.setText("Class: " +
    enemyCharacter.getCard() + "\nAttack: " +
    enemyCharacter.getAttack() + "\nDefense: " +
    enemyCharacter.getDefense() + "\nAgility: " +
    enemyCharacter.getAgility() + "\nLuck: " +
    enemyCharacter.getLuck() + "\nLevel " +
    enemyCharacter.getChosenWeapon().getLevel() + " "
    )
}

```

```
468 + enemyCharacter.getChosenWeapon().getName());
469 } catch (NullPointerException e){
470     System.out.println(e.getCause());
471 }
472     mainHP.setProgress(mainCharacter.getHp()/
Double.valueOf(mainCharacter.getTotalHP()));
473     enemyHP.setProgress(enemyCharacter.getHp()/
Double.valueOf(enemyCharacter.getTotalHP()));
474 }
475
476 @FXML
477 public void enemyMove(ActionEvent e){
478     int which = randomNumber(0, 4);
479     System.out.println(which);
480     if (!playerTurn){
481         if(which <= 2){
482             attack(e);
483         } else if (which == 3){
484             recover(enemyCharacter);
485         } else {
486             specialMove(enemyCharacter,
mainCharacter);
487         }
488     }
489     playerTurn = true;
490 }
491
492 @FXML
493 public void recoverClicked(){
494     if(playerTurn){
495         recover(mainCharacter);
496     }
497     playerTurn = false;
498 }
499
500 @FXML
501 public void recover(Character a){
502     if(a.equals(enemyCharacter) &&
enemyCharacter.getHp() + 100 > enemyCharacter.
getTotalHP()){
503         specialMove(enemyCharacter,
```

```

503 mainCharacter);
504     }
505     double num = randomNumber(1, 30);
506     a.setHp((int) (a.getHp() + num/100 * a.
507     getHp()));
508     turnText.setText(a.getName() + " Improved
509     HP By " + num + "%");
510     updateStats();
511 }
512 @FXML
513 public void attack(ActionEvent e){
514     if(playerTurn && e.getSource().toString().
515     equals("Button[id=playerAttack, styleClass=button
516 ]' Attack'")){
517         move(mainCharacter, enemyCharacter);
518         playerTurn = false;
519     } else if (!playerTurn && e.getSource().
520     toString().equals("Button[id=enemyAttack,
521     styleClass=button]'Enemy Move'")){
522         move(enemyCharacter, mainCharacter);
523         playerTurn = true;
524     }
525     updateStats();
526     if (enemyCharacter.getHp() <= 0){
527         turnText.setText("You Won :)");
528         collectRewards.setVisible(true);
529         playerWon = true;
530     } else if(mainCharacter.getHp() <= 0){
531         turnText.setText("You Lost :( ");
532         collectRewards.setVisible(true);
533         playerWon = false;
534     }
535     if(randomNumber(1, 50) <= (b.getAgility

```

```

535 () + b.getChosenWeapon().getAgility()) - (a.
      getAgility() + a.getChosenWeapon().getAgility())){
536         turnText.setText(b.getName() + "
Dodged The Hit.\nNext Turn: " + b.getName());
537     } else if ((double)((a.getLuck() + a.
      getChosenWeapon().getLuck())/500) >= Math.random
      ()) {
538         if(randomNumber(1, 100) == 1) {
539             turnText.setText(b.getName() + "
Was Frightened and Lost All HP");
540             b.setHp(0);
541         }
542     } else {
543         int damage = (int) (Math.pow((a.
      getAttack() + a.getChosenWeapon().getAttack()), 2
      ) / ((a.getAttack() + a.getChosenWeapon().
      getAttack()) + (b.getDefense() + b.getChosenWeapon()
      ().getDefense())));
544         b.setHp(b.getHp()-damage);
545         turnText.setText("Damage: -" + damage
      + " HP\nNext Turn: " + b.getName());
546     }
547     updateStats();
548 }
549
550 @FXML
551 public void specialMoveClicked(){
552     if (playerTurn){
553         specialMove(mainCharacter,
      enemyCharacter);
554     }
555     playerTurn = false;
556 }
557
558 @FXML
559 public void specialMove(Character a, Character
      b){
560     if(randomNumber(1, 50) == 1){
561         turnText.setText(a.getName() + " Found
      A Strong Recovery Potion");
562         a.setHp(a.getHp() + 100);

```

```

563         } else if(randomNumber(1, 10) == 1){
564             if(a.equals(mainCharacter)){
565                 turnText.setText(a.getName() + " "
566                 Found " + a.getLevel() * 10 + " Lying Around");
567                 a.setMoney(a.getMoney() + a.
568                 getLevel() * 10);
569             } else {
570                 turnText.setText(a.getName() + " "
571                 Robbed You");
572                 if (a.getMoney() > 0){
573                     a.setMoney(a.getMoney() - 10);
574                 } else {
575                     turnText.setText(a.getName()
576                     () + " Tried Robbing You But Found Out You Are
577                     Broke");
578                 }
579             }
580             updateStats();
581         }
582
583     @FXML
584     public void setRewards(){
585         rewardPane.setVisible(true);
586         try {
587             FileInputStream input = new
588             FileInputStream(mainImage);
589             mainRewards.setImage(new Image(input
590             ));
591         } catch (Exception e) {
592             e.printStackTrace();
593         }
594         if(playerWon){
595             visa += mainCharacter.getLevel();
596         }
597     }
598 }
```

```
594         mainCharacter.setMoney(mainCharacter.  
      getMoney() + mainCharacter.getLevel() * 10);  
595         rewardsList.setText("Money: $" +  
      mainCharacter.getMoney() + " +" + mainCharacter.  
      getLevel()*10);  
596         rewardsList.setText(rewardsList.  
      getText() + "\nVisa: " + visa + " Days +" +  
      mainCharacter.getLevel());  
597  
598         mainCharacter.setTotalHP(mainCharacter.  
      getTotalHP() + mainCharacter.getLevel() * 10);  
599         rewardsList.setText(rewardsList.  
      getText() + "\nHP: " + mainCharacter.getTotalHP()  
      () + " +" + mainCharacter.getLevel()*10);  
600  
601     if(opponent.equals("Spades")){  
602         mainCharacter.setAttack(  
      mainCharacter.getAttack() + mainCharacter.getLevel()  
      () * 10);  
603         rewardsList.setText(rewardsList.  
      getText() + "\nAttack: " + mainCharacter.getAttack()  
      () + " +" + mainCharacter.getLevel()*10);  
604  
605 } else if(opponent.equals("Hearts")){  
606         mainCharacter.setDefense(  
      mainCharacter.getDefense() + mainCharacter.  
      getLevel() * 10);  
607         rewardsList.setText(rewardsList.  
      getText() + "\nDefense: " + mainCharacter.  
      getDefense() + " +" + mainCharacter.getLevel()*10  
      );  
608  
609 } else if (opponent.equals("Clubs")){  
610         mainCharacter.setAgility(  
      mainCharacter.getAgility() + mainCharacter.  
      getLevel() * 10);  
611         rewardsList.setText(rewardsList.  
      getText() + "\nAgility: " + mainCharacter.  
      getAgility() + " +" + mainCharacter.getLevel()*10  
      );  
612
```

```

613             } else if (opponent.equals("Diamonds")
614             ){
615                 mainCharacter.setLuck(
616                     mainCharacter.getLuck() + mainCharacter.getLevel()
617                     () * 10);
618                 rewardsList.setText(rewardsList.
619                     getText() + "\nLuck: " + mainCharacter.getLuck()
620                     () + " +" + mainCharacter.getLevel()*10);
621             }
622             @FXML
623             public void exitRewards(){
624                 rewardPane.setVisible(false);
625                 collectRewards.setVisible(false);
626             }
627
628             @FXML
629             public void setWeaponPane(){
630                 weaponPane.setVisible(true);
631                 weaponsList.getItems().clear();
632                 for(String name: types){
633                     weaponsList.getItems().add(name);
634                 }
635             }
636
637             @FXML
638             public void showWeapon(){
639                 String which = weaponsList.
640                     getSelectionModel().getSelectedItem().toString();
641                 weaponInfo.setVisible(true);
642                 int where = weaponNames.indexOf(which);
643                 showWeapon = allWeapons.get(where);
644
645                 while(mainCharacter.getWeapons().indexOf(
646                     showWeapon) >= 0 && showWeapon.getName().equals(

```

```
645 which)) {
646         where += 1;
647         showWeapon = allWeapons.get(where);
648     }
649     System.out.println(showWeapon.getName());
650     System.out.println(mainCharacter.
651         getWeapons().indexOf(showWeapon));
651 //        System.out.println(mainCharacter.
652 //            getWeapons().indexOf(showWeapon) > 0);
652     System.out.println(showWeapon.equals(
653         mainCharacter.getWeapon(0)));
653     System.out.println(mainCharacter.getWeapon
654         (0).getName() + "\nLevel: " + mainCharacter.
655         getWeapon(0).getLevel() + "\nCost: $" +
656         mainCharacter.getWeapon(0).getCost());
657     System.out.println("Attack: " +
658         mainCharacter.getWeapon(0).getAttack() + "\n
659         Agility: " + mainCharacter.getWeapon(0).getAgility
660         () + "\nDefense: " + mainCharacter.getWeapon(0).
661         getDefense() + "\nLuck: " + mainCharacter.
662         getWeapon(0).getLuck());
663     System.out.println(mainCharacter.getWeapon
664         (0).getImage());
665     System.out.println(showWeapon.getName() +
666         "\nLevel: " + showWeapon.getLevel() + "\nCost: $" +
667         showWeapon.getCost());
668     System.out.println(("Attack: " +
669         showWeapon.getAttack() + "\nAgility: " +
670         showWeapon.getAgility() + "\nDefense: " +
671         showWeapon.getDefense() + "\nLuck: " + showWeapon.
672         getLuck()));
673     System.out.println(showWeapon.getImage());
674     try {
675         FileInputStream input = new
676         FileInputStream(showWeapon.getImage());
677         weaponIcon.setImage(new Image(input));
678     } catch (Exception e) {
679         e.printStackTrace();
680     }
681
682
683
684
685
686
```

```

667     if(showWeapon.getLevel() * 3 - 2 >=
       mainCharacter.getLevel() && (showWeapon.getLevel()
       () <= mainCharacter.getLevel()/3 || showWeapon.
       getLevel() <= mainCharacter.getLevel()%3)){
668         buyButton.setDisable(false);
669         weaponError.setVisible(false);
670     } else {
671         buyButton.setDisable(true);
672         weaponError.setText("Player Level Not
High Enough To Purchase");
673         weaponError.setVisible(true);
674     }
675 //      System.out.println(showWeapon.getLevel
676 //      () * 3 - 2 >= mainCharacter.getLevel());
676 //      System.out.println(showWeapon.getLevel
677 //      () <= mainCharacter.getLevel()/3);
677 //      System.out.println(showWeapon.getLevel
678 //      ());
678 //      System.out.println((double)(

mainCharacter.getLevel())/3);
679
680         weaponLabel.setText(showWeapon.getName
680         () + "\nLevel: " + showWeapon.getLevel() + "\nCost
: $" + showWeapon.getCost() );
681         weaponStats.setText("Attack: " +
showWeapon.getAttack() + "\nAgility: " +
showWeapon.getAgility() + "\nDefense: " +
showWeapon.getDefense() + "\nLuck: " + showWeapon.
getLuck());
682     }
683
684     @FXML
685     public void buyWeapon(){
686         boolean pushed = false;
687         if(mainCharacter.getMoney() < showWeapon.
getCost()){
688             weaponError.setText("Not Enough Money"
);
689             weaponError.setVisible(true);
690         } else {
691             mainCharacter.setMoney(mainCharacter.

```

```
691 getMoney() - showWeapon.getCost());
692
693         //Looking through the current array to
694         //see if the weapon is already there and if it is
695         //then replacing it
696         for(int i = 0; i < mainCharacter.
697             getWeapons().size(); i++){
698             if(mainCharacter.getWeapon(i).
699                 getName().equals(showWeapon.getName())){
700                 mainCharacter.setWeapon(
701                     showWeapon, i);
702                 pushed = true;
703             }
704             if (!pushed){
705                 mainCharacter.addWeapon(showWeapon
706             );
707         }
708     @FXML
709     public void exitWeapons(){
710         System.out.println("running");
711         weaponPane.setVisible(false);
712     }
713
714     @FXML
715     public void saveButton(){
716         save();
717     }
718
719     @FXML
720     public void save(){
721         String outFile = "src/main/resources/
722         mainStats.txt";
723         try{
724             PrintWriter out = new PrintWriter(
725                 outFile);
```

```
724             out.println(mainCharacter.getName());
725             out.println(mainCharacter.getTotalHP
    ());
726             out.println(mainCharacter.getLevel());
727             out.println(mainCharacter.getMoney());
728             out.println(mainCharacter.getDefense
    ());
729             out.println(mainCharacter.getAttack
    ());
730             out.println(mainCharacter.getAgility
    ());
731             out.println(mainCharacter.getLuck());
732             out.println(mainCharacter.getCard());
733             out.println(visa);
734             out.println(mainImage);
735             out.close();
736         } catch (FileNotFoundException e){
737             System.out.println("no file");
738         }
739
740         String outFile2 = "src/main/resources/
mainWeapon.txt";
741         try{
742             PrintWriter out2 = new PrintWriter(
    outFile2);
743             for (Weapon a: mainCharacter.
    getWeapons()){
744                 out2.println(a.getName());
745                 out2.println(a.getImage());
746                 out2.println(a.getLevel());
747                 out2.println(a.getAttack());
748                 out2.println(a.getAgility());
749                 out2.println(a.getDefense());
750                 out2.println(a.getLuck());
751                 out2.println(a.getCost());
752             }
753             out2.close();
754         } catch (FileNotFoundException e2){
755             System.out.println("no file 2");
756         }
757     }
```

```
758
759     @FXML
760     public void handleLoad(){
761         try {
762             FileReader reader = new FileReader("src/main/resources/mainStats.txt");
763             Scanner in = new Scanner(reader);
764
765             ArrayList <String> stats = new
766             ArrayList<>();
767
768             while(in.hasNextLine()) {
769                 stats.add(in.nextLine());
770             }
771
772             setStats(stats);
773         } catch (IOException ex){
774             System.out.println("Something is very
775             wrong");
776         }
777
778         try{
779             FileReader reader2 = new FileReader("src/main/resources/mainWeapon.txt");
780             Scanner in2 = new Scanner(reader2);
781             ArrayList <String> weaponLoad = new
782             ArrayList<>();
783
784             while (in2.hasNextLine()){
785                 weaponLoad.add(in2.nextLine());
786             }
787
788             loadWeapons(weaponLoad);
789         } catch (FileNotFoundException ex2){
790             System.out.println("Something is very
791             wrong 2");
792         }
793
794     @FXML
795     public void setStats(ArrayList<String> x){
```

```
793         mainCharacter.setName(x.get(0));
794         mainCharacter.setTotalHP(Integer.parseInt(
795             x.get(1)));
795         mainCharacter.setLevel(Integer.parseInt(x.
796             get(2)));
796         mainCharacter.setMoney(Integer.parseInt(x.
797             get(3)));
797         mainCharacter.setDefense(Integer.parseInt(
798             x.get(4)));
798         mainCharacter.setAttack(Integer.parseInt(x.
799             .get(5)));
799         mainCharacter.setAgility(Integer.parseInt(
800             x.get(6)));
800         mainCharacter.setLuck(Integer.parseInt(x.
801             get(7)));
801         mainCharacter.setCard(x.get(8));
802         visa = Integer.parseInt(x.get(9));
803         mainImage = x.get(10);
804     }
805
806     @FXML
807     public void loadWeapons(ArrayList<String> x){
808         mainCharacter.getWeapons().clear();
809         for(int i = 0; i < x.size(); i += 8){
810             mainCharacter.getWeapons().add(new
811                 Weapon(x.get(i), x.get(i+1), Integer.parseInt(x.
812                     get(i + 2)), Integer.parseInt(x.get(i+3)), Integer
813                     .parseInt(x.get(i+4)), Integer.parseInt(x.get(i+5
814                     )), Integer.parseInt(x.get(i+6)), Integer.parseInt
815                     (x.get(i+7))));
816         }
817         System.out.println(mainCharacter.getWeapon
818             (0).getName());
819     }
820
821     @FXML
822     public void visaTimer(){
823         long current = System.nanoTime();
824
825         double timeElapsed = (current - visaTime
826             ) / 1_000_000_000;
```

```
820         timeElapsed = roundToPlace(timeElapsed, 1
821     );
822     if(timeElapsed > 30.0){
823         visa -= 1;
824         visaTime = current;
825         if(visa==0){
826             gameOver.setVisible(true);
827         }
828         if(visa==1){
829             infoText.setText("Level: " +
mainCharacter.getLevel() + "\nMoney: $" +
mainCharacter.getMoney()
+ "\nVisa: WARNING ONLY "
+ Integer.toString(visa) + " DAY");
830         }
831     }
832 }
833 }
834
835     public int randomNumber(int a, int b) {
836         double x = Math.floor(Math.random() * (b
- a + 1) + a);
837         return (int) x;
838     }
839
840     public double roundToPlace(double num, int
place) {
841         num*=Math.pow(10, place);
842         num = Math.round(num);
843         num/=Math.pow(10, place);
844         return num;
845     }
846 }
```

```
1 package com.example.demo;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Scene;
6 import javafx.stage.Stage;
7
8 import java.io.IOException;
9
10 public class HelloApplication extends Application {
11     @Override
12     public void start(Stage stage) throws
13         IOException {
14         FXMLLoader fxmlLoader = new FXMLLoader(
15             HelloApplication.class.getResource("hello-view.fxml"
16         ));
17         Scene scene = new Scene(fxmlLoader.load(),
18             600, 600);
19         stage.setTitle("Hello!");
20         stage.setScene(scene);
21         stage.show();
22     }
23 }
```