Kuang Si Thu, Maisha Supritee Chowdhury

Code Review

We decided to tackle each of the classes the other person wrote, including the test classes, and fix any code smells we find. Some of the fixes are identified code smells, while some are just for better code quality (eg. exception handling)

Leaderboard :

- Renamed constant “playerListSize” to “leaderboardCapacity” (confusing variable name)

```
10  10
11      - import static utilities.Constants.playerListSize;
    11  + import static utilities.Constants.leaderboardCapacity;
12  12
```

```
58      -           if (playerScores.size() < playerListSize) {
    58  +           if (playerScores.size() < leaderboardCapacity) {
```

(in utilities.Constants)

```
21  21      // Leaderboard related
22      -   public static final int playerListSize = 5;
    22  +   public static final int leaderboardCapacity = 5;
23  23      public static final String leaderboardFile = "src/main/java/leaderboard/leaderBoard.txt";
24  24      // Entities related
```

- Refactored writeToFile() method for better exception handling and make sure fileW is closed

```
128 128     public void writeToFile() {
    129 +       FileWriter fileW = null;
129 130         // creating a file
130 131         try {
131 132             File file = new File(Constants.leaderboardFile);
132 133             file.createNewFile();
133     -           FileWriter fileW = new FileWriter(file);
    134 +           fileW = new FileWriter(file);
134 135
135 136             // outputting to the file
136 137             fileW.write(this.toString());
137 138
138 139             // closing the file
139 140             fileW.close();
140 141         } catch (IOException e) {
141     -           System.out.println("There was an error.");
    142 +           System.out.println("I/O error");
142 143             e.printStackTrace();
    144 +       } finally {
    145 +           //make sure that fileW is closed
    146 +           try {
    147 +               if (fileW != null) fileW.close();
    148 +           } catch (IOException e) {
    149 +               System.out.println("Cannot close the file");
    150 +           }
143 151         }
144 152     }
```

- Refactored readFromFile() method for better exception handling and make sure fileReader is closed

```
151 159       private void readFromFile() {
    160 +         Scanner fileReader = null;
152 161           try {
153 162               File file = new File(Constants.leaderboardFile);
154 163               // if a file doesn't exist
```

↧ Show 20 lines ↕ Show all unchanged lines ↥ Show 20 lines

```
156 165                   file.createNewFile();
157 166
158 167               // reading from the file
159     -             Scanner fileReader = new Scanner(file);
160     -             ArrayList<PlayerScore> leaderboardData = new ArrayList<PlayerScore>();
    168 +             fileReader = new Scanner(file);
    169 +             ArrayList<PlayerScore> leaderboardData = new ArrayList<>();
161 170
162 171               while (fileReader.hasNextLine()) {
163 172                   String data = fileReader.nextLine();
```

```
174 183                   this.playerScores = leaderboardData;
175 184               } catch (IOException e) {
176     -                 System.out.println("There was an error.");
    185 +                 System.out.println("I/O error");
177 186                   e.printStackTrace();
    187 +             } finally {
    188 +                 //make sure fileReader is closed
    189 +                 if (fileReader != null) fileReader.close();
178 190               }
179 191           }
```

- Included description for javadocs (lack of documentation)

```
98      -         * @param index
    98  +         * @param index index of playerScore in leaderboard
```

- Changed the typing to generic typing

```
23 23       private Leaderboard() {
24     -        this.playerScores = new ArrayList<PlayerScore>();
   24  +        this.playerScores = new ArrayList<>();
25 25           readFromFile();
26 26       }
```

```
210 222
211 223     public static Leaderboard Clone() {
212     -       ArrayList<PlayerScore> playerScores = new ArrayList<PlayerScore>();
213     -       Leaderboard l = new Leaderboard(playerScores);
214     -       return l;
    224 +       ArrayList<PlayerScore> playerScores = new ArrayList<>();
    225 +       return new Leaderboard(playerScores);
215 226       }
216 227   }
```

Buttons :

- added title param description in javadoc (lack of documentation)

- anonymous action listener - replaced with lambda expression (for all buttons)

```
public static void addPlayButton(JPanel panel, String buttonName) {
    JButton playButton = new JButton(buttonName);
    playButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
```

```
public static void addPlayButton(JPanel panel, String buttonName) {
    JButton playButton = new JButton(buttonName);
    playButton.addActionListener(e -> {
```

- removed unused import - static ui.GameUI.revalidateMainScreen, java.awt.event.ActionEvent, and java.awt.event.ActionListener

Elements :

- addRewardPanel(JPanel panel, int numOfRewards) - numOfRewards is not used in this method/any of its derived methods, so change addRewardPanel(this, int) to addRewardPanel(this) everywhere its been used (unused variable)

PanelWithBackgoundImage :

- changed private image to private final image

EntitiesTest :

- changed variables entities, tesPos and testPos2 to final
- removed unused import character.Player and import org.junit.jupiter.api.BeforeEach

EntitiesGeneratorTest :

- simplified assertTrue/False to assertEquals/NotEquals

PlayerTest :

- changed variable testScore to final

PlayerScoreTest :

- removed unused import static org.junit.Assert

Trap :

- changed switch statement to enhanced switch

```
switch (trapType) {
    case BOOBYTRAP:
        points = Constants.boobyTrapDmg;
        break;
    case TRAPFALL:
        points = Constants.trapFallDmg;
        break;
    default:
        throw new IllegalArgumentException("Invalid TrapType");
}
```

```
int points = switch (trapType) {
    case BOOBYTRAP -> Constants.boobyTrapDmg;
    case TRAPFALL -> Constants.trapFallDmg;
    default -> throw new IllegalArgumentException("Invalid TrapType");
};
```

CharacterModel :

- replace switch with enhanced switch