CONCORDIA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

SOEN 6441, Winter 2019

Instructor: Amin Ranj Bar

RISK Game (Build-1)

<u>Coding Standards</u>

BY:

Team - 13

Jemish Kishor Paghadar	(40080723)
Deep Alpesh Patel	(40087798)
Raj Patel	(40085012)

Hardik Vora (40087606)

Dolly Modha (40084358)

Coding Conventions

Code Layout

- Blank lines have been left between code blocks to increase readability
- Curly braces are written on the same line on which code block starts, rather than on separate line.
- The body of a function indented with respect to its header.
- The body of a for, while, or switch statement is indented with respect to its first line; and similarly for if statements and other nested structures.

```
PlayGameController.java
                                            🔃 PlayerModel.java 🔀 🔃 PlayerModel.java
                                                                                                                  Main.java
               public void reinforcementPhase(Territory territory, ObservableList<Territory> terrList, TextArea txtAreaMsg) {
    ArrayList<Territory> terrArList = new ArrayList<Territory>(terrList);
  344 €
  346
  347
                    if(playerList.size()<=1)
    return;</pre>
  350
                    // Run the task in a background thread
                    if(currentPlayer.getStrategy() instanceof Human || (!Config.isThreadingForTournament)) {
    System.out.println("Inside this");
  352
                         currentPlayer.getStrategy().reinforcementPhase(terrList, territory, currentPlayer,terrArList, null);
if (currentPlayer.getArmies() <= 0 && playerList.size() > 1) {
  353
                               GameUtils.addTextToLog("===Reinforcement phase Ended! ===\n");
  355
  356
                                    setChanged();
  357
                                    notifyObservers("Attack");
  358
                    }else {
    Thread backgroundThread = new Thread(new Runnable() {
  359
  360 ⊖
                               @Override
public void run() {
  361 €
△362
                                    try {
    Thread.sleep(Config.waitBeweenTurn);
} catch (InterruptedException e) {
    e.printStackTrace();
  363
  364
  366
  367
                                     .
currentPlayer.getStrategy().reinforcementPhase(terrList, territory, currentPlayer,terrArList,mull);
                                    if (currentPlayer.getArmies() <= 0 && playerList.size() > 1) {
   GameUtils.addTextToLog("===Reinforcement phase Ended! ===\n");
  369
  370
                                          Platform.runLater(()
  372
373
                                               setChanged();
notifyObservers("updateReinforceArmy");
                                               setChanged();
  375
376
                                               notifyObservers("Attack");
                                          });
                                    }
  378
379
                              }
                         });
                              Terminate the running thread if the application exits
                         backgroundThread.setDaemon(true);
// Start the thread
  381
  382
                         backgroundThread.start();
  384
               }
```

Indentation

Code is indented according to its nesting level to improve code readability.

File Organization

- The package name should be in lowercase. For this project, every package name starts with com.risk6441 and then append with the functionality that the classes in package are going to implement. ex. **com.risk6441.models.**
- Any external file that is being used such as map file, should be in lowercase and should be in resource folder.

```
_ _
                                          6
                                                     public boolean isFortificationPhasePossible(Map map, Player
    for (Continent continent : map.getContinents()) {
        for (Territory territory : continent.getTerritories())
                                             569⊜
▼ 🏭 > Team13 [team13 master]
                                             570
                                             571

▼ 

main/java

                                                                   if (territory.getPlayer().equals(currentPlayer)
                                             572
     # # com.risk6441.configuration
                                                                        for (Territory adjterritory: territory.getA
   if (adjterritory.getPlayer().equals(curr
                                             573
       CardType.java
                                             574
       Configuration.java
                                             575
                                                                                 Platform.runLater(()->{
       PlayerStrategy.java
                                             576
                                                                                      setChanged();
                                                                                      notifyObservers("Fortification")

▼ # > com.risk6441.controller

                                             577
       DiceController.java
                                             578
       MainController.java
                                             580
       ▶ ☐ MapRedactorController.java
                                             581
       ▶ ☐ MapRedactorSplashController.java
                                             582
                                                                        }
       PlayerStrategyChooserController.ja
       PlayGameController.java
                                                              }
     # com.risk6441.entity
                                             585
       Card.java
                                                          Platform.runLater(()->{
                                             586
       ▶ ☐ Continent.java
                                             587
                                                              setChanged();
                                             588
                                                              notifyObservers("NoFortification");
       ▶ ☐ Country.java
                                             589
       Map.java
                                             590
       Player.java
                                             591
                                                          return false;
    ► # com.risk6441.exception
                                                     }
     593
     ▼ # com.risk6441.main
                                             594⊖
     Main.java
                                             595
                                                     * This method is used to exchange cards for army.
     596
                                             597
                                                        @param selectedCardsByThePlayer List of cards selected by
       CommonMapUtilities.java
                                             598
                                                        @param txtAreaMsg
                                                                                               The area where the messag
       ▶ № MapOperations.java
                                             599
                                                                                               displayed.
       ▶ MapReader.java
                                            600
       MapVerifier.java
       MapWriter.java
                                           🥋 Problems 🏿 @ Javadoc 🚇 Declaration 📮 Console 🕱
     ▼ 🞝 > com.risk6441.models
                                           <terminated> Main (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_191.jdk/Content
       GameUIStateModel.java
                                           Total No of Countries : 6
Total No of Countries : 6
Total No of Countries : 6
       ▶ ♣ PlaverModel.iava
       ▶ □ WorldDominationModel.java
                                           Intializer Called
     ▼ 表 com.risk6441.strategy
```

▶ Naming Conventions

• Constant have been named with uppercase letters and separated by underscore.

```
public static final Integer ARMIES_THREE_PLAYER = 35;

25
26⊕ /**

* The ARMIES_TWO_PLAYER Constant assigns 30 armies to 4 player in the game

*/

public static final Integer ARMIES_FOUR_PLAYER = 30;
```

- All class names start with uppercase letter and words are separated using case change.
- Folder and Package names are written in lowercase.
- File names are written in UpperCamelCase.

```
▼ 등 > src/main/java
▼ 등 com.risk6441.configuration
▼ GardType.java
▼ CardType
```

- Method and Parameter names are written in lowerCamelCase.
- Local variable names are written in lowerCamelCase.

```
26
        private Player player;
27
        private List<Country> adjacentCountries;
28
        private boolean isProcessed;
29
30 ⊖
        /**
31
         * Setter method for the player.
32
         * @param player set player
33
         */
        public void setPlayer(Player player) {
34⊖
35
            this.player = player;
36
```

Comments

- Commenting is done as per conventions for Java Doc.
- Each class declaration precedes by a comment explaining what the class is for.

```
package com.risk6441.controller;
  2⊕ import java.net.URL;
 26
 27⊜ /**
     * This class controller allows to choose a player Strategies for selected number of player
     * which are listed below
     * HUMAN
     * COMPUTER
 33
     * 
     * @author Deep
     * @author Jemish
 35
 36
     * @author Dolly
 37
 38 public class PlayerStrategyChooserController extends Observable implements Initializable{
 40⊝
 41
        private VBox vBox;
 42
 43⊖
        @FXML
 44
        private Button btnSubmit;
 45
```

• Each method or function have comments explaining what it does, as well as what is the purpose of parameters and return type description if the method's return is non-void.

```
Main.java
              🚺 PlayerStrategyChooserController.java 🔀
  эт
  92⊖
  93
           * This method is used for getting the strategy for the player.
  94
           * @param str string defining type of strategy for the player
  95
           * @return strategy object
  96
           */
          private IStrategy strategyObjectForPlayer(String str) {
  97⊝
  98
              IStrategy strategy = null;
  99
              if (str.equals("HUMAN")) {
 100
                  strategy = new Human();
              } else if (str.equals("COMPUTER")) {
 101
 102
                  strategy = new Computer();
              }
 103
 104
              return strategy;
          }
 105
```