# **Refactoring Introduction**

# Refactoring in the map package

# (1 Selected Refactoring option has been marked)

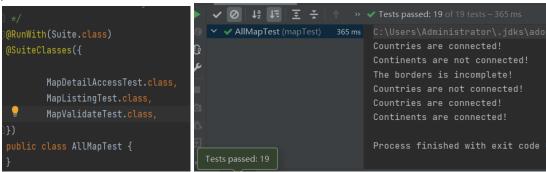
In the map package, there are 5 classes related to the detecting map function. So, those classes have to merge or modify. We listed 5 potential targets to refactor:

- 1. Combining all classes in the map package into one class.
- 2. Creating one class that inherits all map classes in the map package.
- 3. Combining all classes in the map package except the class that can detect maps.
- **4.** Combining classes that are related to get the file information(the country's storage location), and get the map information(functions that return a list of countries or continents).

Option #1 and #3 may cause a bigger mess than before because the new class may be too long to maintain. Option #2 is easy to call, but it doesn't make the project structure clear. Therefore, #4 is more useful than other options. The methods classified by the function are more convenient to call.

The following two graphs show the results of the test runs before the refactoring. Nineteen tests are all passed.

After we refactor, there are only three classes which relate to the map validate function. The MapCheck.java combined into the MapValidate.java, The MapLineAccess.java combined into the MapDetialAccess.java. The following two graphs show the results of the test runs after the refactoring. Nineteen tests are all passed.



# Refactoring GameEngine/GameData to MVC structure

(we are not fully implement MVC, View part is not complete)

# (2 Selected Refactoring options have been marked)

5. In order to introduce the observer mode in the current version, we move all relevant game elements and game data from the previous version to a package named model as the program data in the model that responds to requests for data about its state. Also, Some classes like (GameData.java, GamePhase.java, LogEntryBuffer.java) in the package model extend Observable.java class that used to allow observers to watch the data change of them.

```
▼ ■ Team_05 [SOEN_6441] ~/Desktop/Team_05
                                                                   package model;
  ▶ idea
                                                          16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
  ▼ 🖿 classesuml
        project.png
                                                                    * This class is a wrapper class containing the elements and the current states of the game
     domination
   ▼ 🗎 src
                                                                         * Map file
        ▼ lava
          ► 🖿 command
                                                                       File d_MapFile;
           ▶ □ controller
                                                                         * Map Object, read only
             ▶ 🖿 gameelements
              ► 🖿 map
                                                                       public MapDetailAccess d_MapDetailAccess;
              © GameData
                                                                         * used to get/set current model.map

    GamePhase

    LogEntryBuffer

                                                                       public MapListing d_MapListing;
                Observable
          ▶ 🖿 view
             © GameDriver
                                                                         * Game current phase
                                                          38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
                                                                       private Phase d_CurrentPhase;
     👢 .gitignore
     m pom.xml
                                                                       private int d_TotalPlayer;
     # SOEN_6441.iml
  III External Libraries
                                                                        * how many countries now in game
  Scratches and Consoles
                                                                       private int d_TotalCountry;
                                                                         * Country object array list
                                                                       private ArrayList<Country> d_CountryList;
                                                                         * Continent object array list
                                                                       private ArrayList<Continent> d_ContinentList;
```

```
Team_05 [SOEN_6441] ~/Desktop/Team_05
                                                      package model;
github 🔳 .
                                                      import ...
▶ Iidea
classesum!
                                                8
                                                      public class LogEntryBuffer extends Observable {
    project.png
    m project.uml
                                               10
                                                           * Byte Array OutputStream
domination
JavaDoc
                                                          private ByteArrayOutputStream d_Baos;
▼ ■ src
  ▼ main
                                               14
                                                           * previous output stream
    ▼ 📄 java
                                               16
                                                          private PrintStream d Previous;
      ▶ □ command
       ▶ □ controller
                                                           * is capturing flag
                                               18
       ▼ 🖿 model
                                               19
         ▶ □ gameelements
                                               20
                                                          private boolean d_Capturing;
         map
         ▶ ■ state
                                                           * loa buffer strina list
            GameData
                                                          private List<String> d_LogBuffer;
           GamePhase
                                               24
            CogentryBuffer
                                               26
            Observable
                                               27
                                                           * file path
            state.zip
                                               28
                                                          private String d_FilePath = "log/Log.txt";
                                               29
         © GameDriver
                                               30
```

**6.** We refactored the *GameEngine.java* class in the previous version and move it from *gameplay* package to the package named *controller*, now the behavior of this class become the role of controller in Observer pattern and named *GameEngineController.java* 

```
Team_05 [SOEN_6441] ~/Desktop/Team_05
                                                         package controller;
aithub ...
                                                         import ...
▶ ■ .idea
classesuml
                                                         /** this is game main controller */
      project.png
                                                         public class GameEngineController {
    m project.uml
  domination
                                                 19
20
23
24
JavaDoc
                                                             /** the GameData you design from the MainLoop */
▼ N src
                                                             public GameData d GameData;
   ▼ 🗎 main
     ▼ 📄 java
                                                 25
30
37
                                                             /** GameEngineController Constructor ...*/
                                                             public GameEngineController(GameData p_GameData) {...}
       ▼ lacommand
            CommandType
             CommandValidator
                                                             /** add new player to the game ...*/
                                                 44
67
                                                             public void addNewPlayer(String p_colour) {...}
           GameEngineController
            MainPlayController
                                                             /** remove player from the game ...*/
       model
                                                             public void removePlayer(Player p_Player) {...}
                                                 82
83
          ▶ □ gameelements
          ► 🖿 map
                                                             /** this method is current player phase process, player's order will be processed here. */
                                                 84
          ▶ ■ state
            © GameData
            GamePhase
                                                100
            © LogEntryBuffer
                                                             /** this method used to player show model.map in game ...*/
            © Observable
                                                106
128
                                                             public void showMap(Player p_Player) {...}
       ▶ 🖿 view
         © GameDriver
  ► test
                                                             /** Game Engine get correct model.map file path function, return model.map path ...*/
► larget
                                                136
                                                             public String getMapFilePath() throws IOException {...}
   륂 .gitignore
                                                             /** Allows the user to create/remove the players for the game by typing command. */
                                                156
  mx.moq m
                                                             public void gamePlayerCommand() {...}
  README.md
                                                192
                                                             /** Randomly assign all the countries to the players. */
III External Libraries
                                                             public void assignCountries() {...}
                                                237
238
Scratches and Consoles
                                                             /** Calculate the reinforcement armies to be award. ...*/
                                                244
                                                             public int getReinforcementBonus(Player p_Player) {...}
```

### **Related Tests**

LogEntryBufferTest: writeTest ViewTest: obsListAttachTest GameDataTest: loadMapTest

### **Refactoring for State Pattern:**

# (1 Selected Refactoring option has been marked)

7. We removed the enum class *GamePhase.java*. Now, each game phase will base on the user input turning to the specified game phase in the state pattern, and remind the user what the current game phase they are in. Also, We changed the if-else statement in the previous version of Demo class mainloop.java to a switch case statement that is more consistent with the state pattern

```
▶ ■ .github
                                                                         mystart = l Scanner.nextLine();
▶ idea
                                                    80
                                                                         switch (mystart.toLowerCase()) {
▼ I classesuml
                                                    81
                                                                             case "edit":
      project.png
                                                                                 // Set the state to Preload
     m project.uml
                                                    83
                                                                                 setPhase(new Preload( p_ml: this)):
▶ ■ domination
                                                                                 //gamePhase.loadMap();
▶ JavaDoc
 ▼ 🖿 src
                                                    86
                                                                                     System.out.println(" ===
   ▼ 🖿 main
                                                                                     System.out.println("| # PHASE
                                                                                                                                         : command
                                                                                                                                                             |");
      java
                                                    88
                                                                                      System.out.println(" =
                                                                                                                                                              ");
                                                                                     System.out.println("| 1. Edit:PreLoad
                                                    89
                                                                                                                                         : load map
                                                                                                                                                             |");
        ▼ 🖿 command
                                                                                     System.out.println("| 2. Edit:PreLoad
             CommandType
                                                                                     System.out.println("| 3. Edit:PostLoad
System.out.println("| 4. Edit:PostLoad
                                                                                                                                         : edit map
                                                    91
                                                                                                                                                              I"):
             © CommandValidator
                                                    92
                                                                                                                                         : save map
        ▼ la controller
                                                                                     System.out.println("| 10. Any
             © GameEngineController
                                                                                     System.out.println("| 11. Any
System.out.println("| 12. Any
                                                                                                                                         : next phase
                                                                                                                                                              I");
             MainPlayController
                                                    95
                                                                                                                                                             j");
                                                                                                                                         : previous phase
        ▼ 🖿 model
                                                                                     System.out.println("enter a " + gamePhase.getClass().getSimpleName() + " phase command: mycommand = \_Scanner.nextInt();
          gameelements
                                                    97
                                                    98
          ▶ 🛅 man
                                                                                     System.out.println(" =
           ▼ 🖿 state
                                                    100
             ▼ 🛅 play
                                                                                     // Calls the method corresponding to the action that the user has selected.
                  C LoadMap
                                                                                      // Depending on what it the current state object, these method calls will
                  © MainPlay
                                                                                     // have a different implementation.
                                                    103
                  Play
                  © Startup
                                                                                     switch (mycommand) {
                📵 Edit
                                                                                         case 1:
                © End
                Phase
                                                    108
                                                                                             break;
                © PostLoad
                                                                                         case 2:
                © Preload
                                                                                             break;
                State
                                                                                         case 3:
              GameData
                                                                                              gamePhase.editMap();
             GamePhase
                                                                                             break;
             CogEntryBuffer
                                                                                         case 4:
             Observable
                                                                                              gamePhase.saveMap();
        ▶ □ view
                                                                                             break;
                                                                                          case 10:
          © GameDriver
                                                                                              gamePhase.endGame();
  ▶ test
                                                    119
                                                                                             break;
▶ larget
                                                   120
   aitianore.
                                                                                              gamePhase.next():
  m pom.xml
                                                                                             break;

    README.md

   SOEN_6441.iml
                                                                                              gamePhase.previous();
                                                                                              System.out.println("this command does not exist");
Scratches and Consoles
```

- 8. User needs to enter the corresponding number for the represent game phase to get in before entering the command for the phase
- 9. We do not need to ask the user for which model (Map edit/Game play / Exit). they are going to get in. Users can based on the entered number to get into different states.
- 10. Divide the gameplay phase into 3 different states, reinforcement phase, issue order phase, execution phase. The user needs to enter the command for the current phase before getting into the next phase.

### **Related Tests**

PlayStateTest: testEditState, testPlayState StateTest: testEditState, testPlayState

GameEngine Test: testAddRemovePlayer, testShowMap

# **General refactoring:**

## (0 Selected Refactoring option has been marked)

- 11. Combine the showmap function of GameEngin and Editmap of previous version, so that the program can view the map by calling showmap function.
- 12. Improve the Mainloop constructor, so that we do not need to input a map file when we define the mainloop as the parameter in the main function.
- 13. combining the new function card in order creation phase (like Bomb, Blockade, etc.) into the issue order phase, so that the user can use the function card durning issue order state.

# **Refactoring Command Pattern:**

# (1 Selected Refactoring option has been marked)

- 14. Command Pattern allows us to hide the logic of order execution from the GameEngine. It also makes it easier for us to add more Order types to the game later if needed.
- 15. Combine function printInvalidCommandIncurrentPhase() and printInvalidArgument() into one method, so that the program will print the same error message when the user enters Invalid command arguments or Invalid command in the current game phase.

### **Related Tests**

DeployOrderTest: testOrderInvalidGivenNullDestination, testOrderInvalidGivenNegativeArmyNumber, testOrderInvalidGivenDestinationNotInControl, testOrderValid

OrderFactoryTest: testDeployOrderCreation