

# Refactoring

Sr No.	File Name	Line Number	Target	Description
1.	Multiple Files	-	State Pattern	State Pattern Implemented
2.	Multiple Files	-	Command Pattern	Command Pattern Implemented
3.	Multiple Files	-	Order Processing	To allow different type of orders generation
4.	GameEngine	106	String l_s	Although it's a local variable but give it proper name so one can understand it easily
5.	GamePlay Class	-	private String d_gamePhase	No need getter setter for this variable
6.	GamePlay Class	-	private String d_fileName	No need of get method for this variable
7.	GamePlay Class	-	private int d_maxNumberOfTurns	No need getter setter for this variable
8.	Deployorder	48	int l_getArmy	for 2 conditions,assign multiple times
9.	GameConfigServiceImpl	50	String l_showMapOfCountris	for 2 conditions,assign multiple times
10.	MapHandlingImpl	246	List<String> l_countryNn	Need to change name of variable
11.	MapHandlingImpl	300	boolean l_result	need to remove this variable as it's a unused variable
12.	WarMap	47	boolean d_isValid	No need of this variable as it is unused variable
13.	CommandResponse	60 to 65	Command fails (else and if)	Merge both with OR condition
14.	Country Class	74 to 78	return false	merge both if conditions using OR
15.	MapHandlingImpl	585 and 633	int l_continentId	need to remove this variable as it is of no use
16.	MapHandlingImpl	603-625 and 670-687	remove neighbour logic	common function can be possible
17.	MapHandlingImpl	658	Country	Need to rename this variable i.e. l_country
18.	GameConfigServiceImpl	63 to 71	simplify for loop	if(l_j==0) and then check for each l_j i.e (if(l_j==0) and (l_j!=0) )
19.	GameConfigServiceImpl	214	int l_i = 0, l_j = 0;	No need of this as in next line it automatically set to zero because of for loop
20.	GameConfigServiceImpl	222	int rand_int1	Need to rename this variable to l_rand_int1

# Rational explanation

- State Pattern - We wanted to alter the behaviour of an object when it's internal state changes, and make it easy to add new varying behaviour that comes with new states.
- Command Pattern - We don't want to know what is the actual method being executed to process the request, or what object will eventually execute it. In these cases, we may want to create a request for a result, store it, then process it elsewhere or sometime after.
- Order Processing - Made order generation to generic so that all distinct types of order can be process in the same way and also give flexibility for adding or removing another types of order without changing order generation process.
- Set Appropriate Variable Name - For better code understanding
- Removed unused variable - For better code readability as well as to reduce memory utilization.
- No Need of Getter/Setter - We didn't remove this variable as it may be required in next build.
- Common Function Can Be Possible - This change was not as important as other changes. So, we leave it as of now.
- Other File Changes - We focused on necessary changes, which was crucial and required more attention as quickly as possible. So, we neglected this minor changes.

# Actual refactoring targets

Refactoring Target	Test	Purpose
State Pattern	1.MapHandligImplTest.java 2. GameEnginImplTest.java	To allows an object to alter its behaviour when it's internal state changes.
Command Pattern	1.MapHandligImplTest.java	To achieve complete decoupling between the sender and the receiver.
Order Process	1. GameEnginImplTest.java	To allow different type of orders generation
Set the appropriate variable name	1.MapHandligImplTest.java 2.GameEnginImplTest.java 3.GameConfigServiceTest.java	Change variable name into meaningful name to enhance readability of code
Remove unused variable	1.MapHandligImplTest.java 2.GameEnginImplTest.java 3.GameConfigServiceTest.java	Remove unused variable

# Before/After depiction

- State Pattern -

<pre>93     } 94 95     /** 96     * 97     * @ -99,8 +116,8 @@ void backToWelcome(ActionEvent p_event) { 98 99     */ 100     @FXML 101     void toStartGame(ActionEvent p_event) { 102 - 103 -         d_stageManager.switchScene(FxmlView.GAMEENGINE, d_gamePlay); 104     } 105 106     /**</pre>	<pre>110     } 111 112     /** 113 114     */ 115     @FXML 116     void toStartGame(ActionEvent p_event) { 117 +         StartupPhase st = (StartupPhase) d_gameEngine.getPhase(); 118 +         st.next(d_gameData); 119     } 120 121 122 123     /**</pre>
<pre>115     CommandResponse l_gmConfigres = new CommandResponse(); 116 117     if (l_command.toLowerCase().startsWith(SHOW_MAP)) { 118         //condition if user gives input to show the map 119 -         if (d_gamePlay.get_warMap() != null) { 120 -             l_gmConfigres = d_gameConfigService.showPlayerMap(d_gamePlay); 121         } else { 122             l_gmConfigres.setD_invalid(false); 123             l_gmConfigres.setD_responseString("Please load the map first"); 124 125     @@ -115,8 +132,8 @@ public void getData(ActionEvent p_event) { 126 127     CommandResponse l_gmConfigres = new CommandResponse(); 128 129     if (l_command.toLowerCase().startsWith(SHOW_MAP)) { 130         //condition if user gives input to show the map 131 +         if (d_gameData.get_warMap() != null) { 132 +             l_gmConfigres = d_gameConfigService.showPlayerMap(d_gameData); 133         } else { 134             l_gmConfigres.setD_invalid(false); 135             l_gmConfigres.setD_responseString("Please load the map first"); 136 137     @@ -144,15 +161,15 @@ public void getData(ActionEvent p_event) { 138 139     l_gmConfigres.setD_invalid(false); 140 141     l_gmConfigres.setD_invalid(false);</pre>	<pre>132     CommandResponse l_gmConfigres = new CommandResponse(); 133 134     if (l_command.toLowerCase().startsWith(SHOW_MAP)) { 135         //condition if user gives input to show the map 136 +         if (d_gameData.get_warMap() != null) { 137 +             l_gmConfigres = d_gameConfigService.showPlayerMap(d_gameData); 138         } else { 139             l_gmConfigres.setD_invalid(false); 140             l_gmConfigres.setD_responseString("Please load the map first"); 141 142     @@ -144,15 +161,15 @@ public void getData(ActionEvent p_event) { 143 144     l_gmConfigres.setD_invalid(false); 145 146     l_gmConfigres.setD_invalid(false);</pre>

- Command Pattern –

<pre>32     public Parent execute() { 33 34         return null; 35     } 36 37     @Override 38     public void next(Object p_nextObject) { 39 40 41     @Override 42     public void executeOrder(Object p_gameData) { 43         //Implement logic here 44 45         List&lt;CommandResponse&gt; l_orderStatus = new ArrayList&lt;&gt;(); 46         for (int l_i = 0; l_i &lt; d_gameData.get_maxNumberOfTurns(); l_i++) { 47             //main loop for giving the turn to player in round-robin 48             for (int l_j = 0; l_j &lt; d_gameData.get_playerList().size(); l_j++) { 49                 if (d_gameData.get_playerList().get(l_j).hasOrder()) { //checks if 50                     the player has an order or not 51                     Order l_order = d_gameData.get_playerList().get(l_j).next_order(); 52                     String l_countryName = ((DeployOrder) l_order).get_CountryName(); 53                     ((DeployOrder) l_order).setD_player(d_gameData.get_playerList().get(l_j)); 54                     //to add the player to use in execution 55                     boolean l_executeOrder = l_order.executeOrder(); 56                     //invokes the order 57                     if (l_executeOrder) {</pre>	<pre>28     public Parent execute() { 29 +         this.printInvalidCommandMessage(); 30         return null; 31     } 32 33     @Override 34     public void next(Object p_nextObject) { 35 +         IssueOrderPhase l_issueOrderPhase = new IssueOrderPhase(d_gameEngine); 36 +         l_issueOrderPhase.d_gameData = (GameData) d_gameData; 37 +         l_issueOrderPhase.d_commandResponses = d_commandResponses; 38 +         d_gameEngine.setPhase(l_issueOrderPhase); 39     } 40 41     @Override 42 +     public void executeOrder() { 43 44         List&lt;CommandResponse&gt; l_orderStatus = new ArrayList&lt;&gt;(); 45         for (int l_i = 0; l_i &lt; d_gameData.get_maxNumberOfTurns(); l_i++) { 46             //main loop for giving the turn to player in round-robin 47             for (int l_j = 0; l_j &lt; d_gameData.get_playerList().size(); l_j++) { 48                 if (d_gameData.get_playerList().get(l_j).hasOrder()) { //checks if 49                     the player has an order or not 50                     Order l_order = d_gameData.get_playerList().get(l_j).next_order(); 51                     String l_countryName = ((DeployOrder) l_order).get_CountryName(); 52                     ((DeployOrder) l_order).setD_player(d_gameData.get_playerList().get(l_j)); 53                     //to add the player to use in execution 54                     boolean l_executeOrder = l_order.executeOrder(); 55                     //invokes the order 56                     if (l_executeOrder) {</pre>
--	--

- Order Processing –

```

29 +
30 + @Autowired
31 + GeneralUtil d_generalUtil;
32 +
33 + @Override
34 + public CommandResponse processOrder(String p_orderCommand,GameData p_gameData) {
35 +     CommandResponse l_commandResponse = new CommandResponse();
36 +     List<String> l_commandData = Arrays.asList(p_orderCommand.split(" "));
37 +     String l_orderName = d_generalUtil.toTitleCase(l_commandData.get(0).toLowerCase());
38 +     List<String> l_args = l_commandData.subList(1, l_commandData.size());
39 +
40 +     l_orderName += "Order";
41 +     Class l_classObj = null;
42 +     Object l_orderObj = null;
43 +     try {
44 +         l_classObj = Class.forName("com.soen6441.warzone.model." + l_orderName);
45 +         l_orderObj = l_classObj.getDeclaredConstructor().newInstance();
46 +     } catch (ClassNotFoundException | NoSuchMethodException | SecurityException |
InstantiationException | IllegalAccessException | IllegalArgumentException |
InvocationTargetException ex) {
47 +         //Invalid Command
48 +         l_commandResponse.setIsValid(false);
49 +         l_commandResponse.setResponseString("Command Is not valid");
50 +     }
51 +
52 +     Field l_mandatoryField = null;
53 +     Field l_player = null;
54 +     Field l_gameData = null;
55 +
56 +     int l_noOfMandatoryFields = 0;
57 +     Field[] l_fields = l_classObj.getDeclaredFields();

```

- Set Appropriate Variable Name –

<pre> 1068         } 1069 1070         for (int i = 0; i &lt; l_currentContinent.getCountryList().size(); i++) 1071         { 1072             Country currentCountry = l_currentContinent.getCountryList().get(i); 1073             if (currentCountry.getCountryIndex() == Integer.parseInt(l_neighbourArray[0])) { 1074                 currentCountry.setNeighbourCountries(l_neighbourName); 1075                 l_currentContinent.getCountryList().set(i, currentCountry); 1076             } 1077             l_continentMap.put(l_currentContinent.getContinentIndex(), l_currentContinent); </pre>	<pre> 1065         } 1066 1067         for (int i = 0; i &lt; l_currentContinent.getCountryList().size(); i++) 1068         { 1069             Country l_currentCountry = l_currentContinent.getCountryList().get(i); 1070             if (l_currentCountry.getCountryIndex() == Integer.parseInt(l_neighbourArray[0])) { 1071                 l_currentCountry.setNeighbourCountries(l_neighbourName); 1072                 l_currentContinent.getCountryList().set(i, l_currentCountry); 1073             } 1074             l_continentMap.put(l_currentContinent.getContinentIndex(), l_currentContinent); </pre>
<pre> @@ -1100,11 +1097,11 @@ private Continent getContinentByCountryId(Map&lt;Integer, Continent&gt; p_continentMap 1100         //getting country list 1101         List&lt;Country&gt; l_countryList = continent.getCountryList(); 1102         if (l_countryList != null) { 1103             for (Country country : l_countryList) { 1104 1105                 if (country != null) { 1106                     //comparing index with country's which we want to find 1107                     if (country.getCountryIndex() == p_countryIndex) { 1108 1109                         return continent; 1110                     } </pre>	
<pre> @@ -1133,12 +1130,12 @@ private String getCountryNameByCountryId(Map&lt;Integer, Continent&gt; p_continentMap, 1133 </pre>	

- Removed unused variable –

297	}	297	}
298	String l_countryName = "";	298	String l_countryName = "";
299	String l_neighbourCountryName = "";	299	String l_neighbourCountryName = "";
300	<del>boolean l_result = false;</del>		
301	List<String> l_commandString = Arrays.asList(p_neighbour.split(" "));	300	List<String> l_commandString = Arrays.asList(p_neighbour.split(" "));
302	if (d_generalUtil.validateIOString(p_neighbour, "editneighbour((\\s-add\\s[a-zA-Z]+\\s[a-zA-Z]+)(\\s-remove\\s[a-zA-Z]+\\s[a-zA-Z]+)*") {	301	if (d_generalUtil.validateIOString(p_neighbour, "editneighbour((\\s-add\\s[a-zA-Z]+\\s[a-zA-Z]+)(\\s-remove\\s[a-zA-Z]+\\s[a-zA-Z]+)*") {
303	List<String> l_cName = getAvailableCountryName(d_warMap);	302	List<String> l_cName = getAvailableCountryName(d_warMap);
@@ -582,7 +581,6 @@ public boolean validateMap(WarMap p_warMap) {			
582	*/	581	*/
583	public boolean deleteContinent(String p_continentName) {	582	public boolean deleteContinent(String p_continentName) {
584	boolean l_result = false;	583	boolean l_result = false;
585	<del>int l_continentId;</del>		
586	List<String> l_countryOfContinent = new ArrayList();	584	List<String> l_countryOfContinent = new ArrayList();
587		585	
588	//to store country of deleting continent	586	//to store country of deleting continent
@@ -630,7 +628,6 @@ public boolean deleteContinent(String p_continentName) {			
630	if (d_warMap.getContinents() != null) {	628	if (d_warMap.getContinents() != null) {
631	for (Map.Entry<Integer, Continent> l_entry : d_warMap.getContinents().entrySet())	629	for (Map.Entry<Integer, Continent> l_entry : d_warMap.getContinents().entrySet())
632	{	630	{
	if (l_entry.getValue() != null &&		if (l_entry.getValue() != null &&
	p_continentName.equalsIgnoreCase(l_entry.getValue().getContinentName())) {		p_continentName.equalsIgnoreCase(l_entry.getValue().getContinentName())) {
633	<del>l_continentId = l_entry.getKey();</del>	631	d_warMap.getContinents().remove(l_entry.getKey());
634	d_warMap.getContinents().remove(l_entry.getKey());	632	l_result = true;
635	l_result = true;	633	break;
636	break;		
@@ -1068,10 +1065,10 @@ public WarMap readMap(String p_filename) throws IOException {			