



1. **Implement the Adapter Pattern** – The user can input filenames for both the domination and conquest maps. Using an adapter class, the conquest map data can be converted to domination map data, and vice versa.
2. **Add regex to mapLoader.java** – The maploader class will utilize regex instead of fixed array indices to retrieve data from the map file.
3. **Implement the strategy pattern** – The player strategies will undergo a complete reconstruction to adhere to the specifications outlined in the strategy pattern.
4. **Remove explicitly writing names of phases to enter a new phase** - The player strategies will be entirely changed to comply with the specifications in the strategy pattern.
5. **Switch from 'System.exit()' to return** –
The application will gracefully finish and close, moving away from the previous dependency on System.exit(), which directly closes the JVM and the application abruptly which is not preferred.
6. **New Parameter (WinnerPlayer)** - A new parameter named WinnerPlayer will be added in the connectivity class to run the process of deciding the winner more efficiently.
7. **Refactor setPlayers (gameplayer)** – setPlayers is being refactored to use strategies as player names for the tournament phase.
8. **Refactor MapLoader to load maps with the various characters** – MapLoader has to successfully load maps with characters such as ',', '.', '&', '-' etc.
9. Create function recordResult to save the result of game played – recordResult must save the details of the outcome of the game played
10. Reformatted the print messages of commands on the terminal to make it more user oriented.
11. Update the help method to add more commands for the user.
12. Execute order in the attack phase will be modified to handle the null order returned by the method next_order in the player.java class.
13. Make the indentation according to the set project standard in attack.java

14. Remove all the unnecessary print statements from the code which were being used by the team for the testing purpose.
15. Remove all the unnecessary comments inside the codes to make the code look cleaner and are added to the Javadoc of the respective function.

The 5 chosen refactoring targets –

1. Implement the Adapter Pattern

- a) User is now able to input map filename of both domination and the conquest file name; the adapter class will convert the conquest map data to domination map data and vice-versa.
- b) Tests implemented – tournamentValidation used for testing the execution of Adapter pattern to convert conquest map data to domination map data

2. Add regex to mapLoader.java

- a) Instead of hardcoding the index of array to retrieve the data from the map file, regex is used in the map loader class to do the same.
- b) Tests implemented – tournamentValidation, phaseValidation, MapLoaderTest

3. Implement the strategy pattern

- a) The whole player strategies is built and refactored according to the specifications of the strategy pattern.
- b) Tests implemented – tournamentValidation, phaseValidation

4. Remove explicitly writing names of phases to enter a new phase –

- a) The need to write name of phases such as “deploy”, “attack”, “fortify”, etc. is removed in this gameplay and instead the phases seamlessly transition from one to the other.
- b) Tests implemented – phaseValidation

5. Switch from ‘System.exit()’ to return

- a) The application gracefully completes and closes instead of the previous use of System.exit() which is not preferred as it closes the JVM directly and then the application.
- b) Tests implemented - tournamentValidation, phaseValidation, loadSaveTest

