

**SOEN 6441 (Advance Programming Practices)**

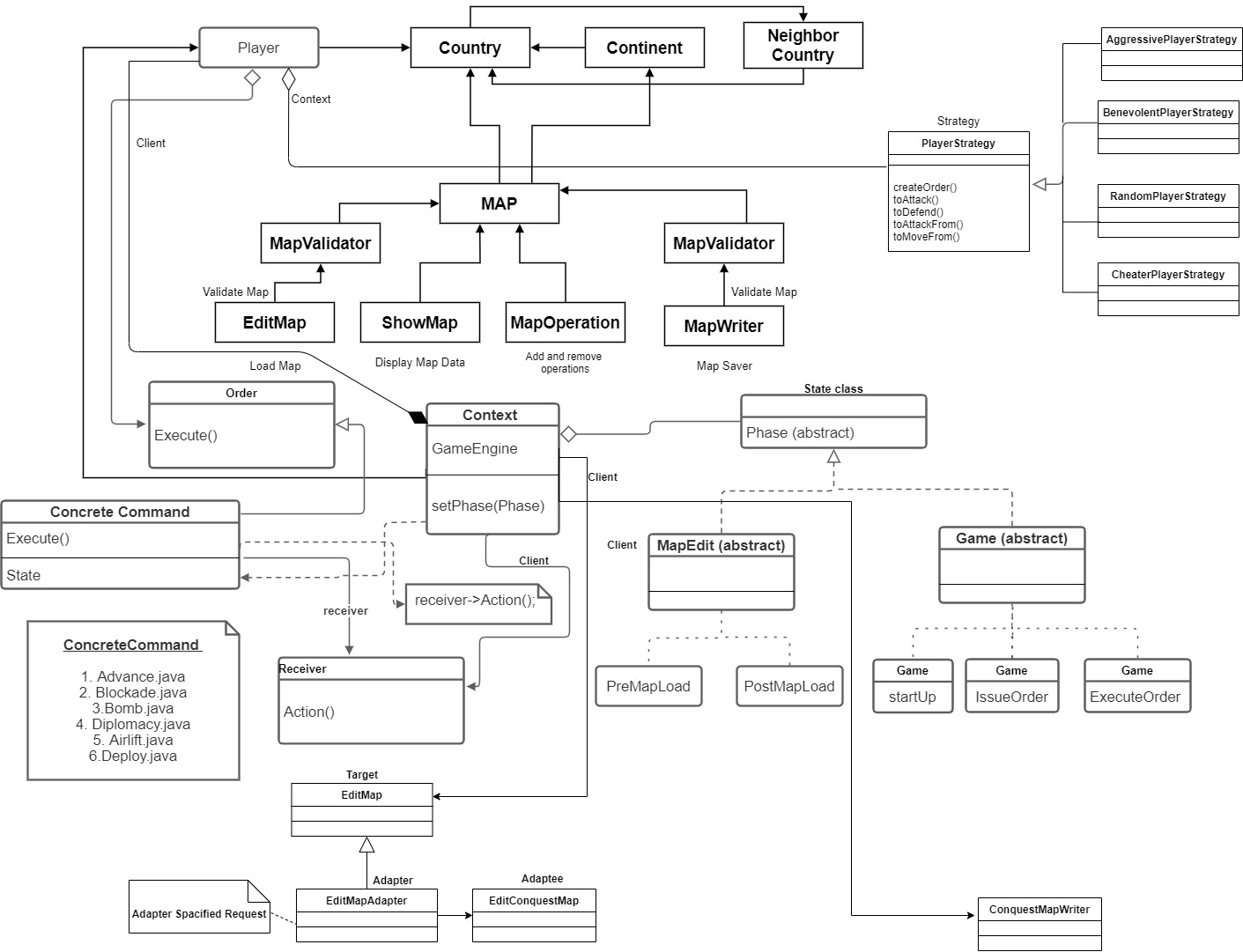
**Build - 3**

**Professor:** Dr. Joey Paquet

**Team 19**

|  |  |
| --- | --- |
| Chirag Patel | **40160656** |
| Harsh Patel | **40165709** |
| Harshil Patel | **40163431** |
| Parth Navsariwala | **40178800** |
| Rishabh Patel | **40170811** |

* **Architectural Diagram:**

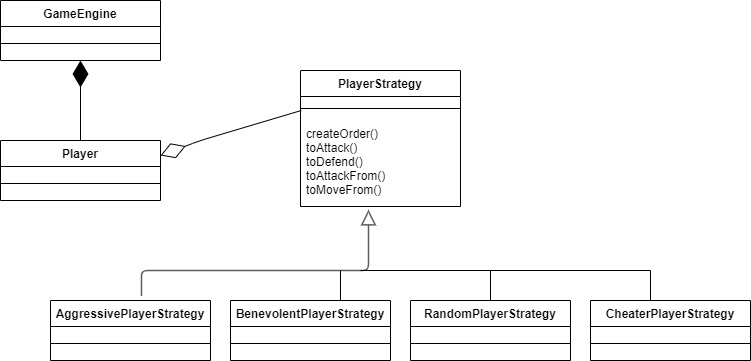
****

**Architectural Diagram**

* In build 3 we have main **5 major changes** in our existing code which are as below: -

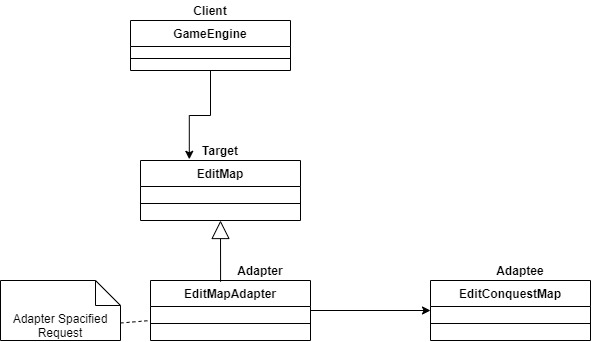
1. **Implementation of Strategy Pattern**
2. **Implementation of Adapter Pattern**
3. **Implementation of Single mode & Tournament Mode**
4. **Implementation of Game save/load**
5. **Refactoring existing code**

* **Strategy Pattern:**

****

**Strategy Pattern**

* Here GameEngine works as a client.
* Player class works as a context.
* PlayerStrategy works as a strategy class and it contain mainly 5 methods
  + CreateOrder()
  + toAttack()
  + toDefend()
  + toAttackFrom()
  + toMoveFrom()
* There are mainly four concrete strategy which are as below:
* AggressivePlayerStrategy,java
* BenevolentPlayerStrategy,java
* RandomPlayerStrategy.java
* CheaterPlayerStrategy.java
* **Adapter Pattern**

****

**Adapter Pattern**

* Here GameEngine class work as a client.
* Here we use mainly two targets: MapWriter & EditMap.
* Here we use mainly two adapters: MapWriterAdapter & EditMapAdapter
* Here we use mainly two adaptees: ConquestMapWriter & EditConquestMap

**Possible Refactoring Targets :**

Listed below are 15 potential refactoring targets:

1. Implement Strategy Pattern

2. Implement Adapter Pattern

3. Modify gameplayer command to pass strategy as an argument

4. Modify savemap command to pass Map type (Conquest or Domination) to save map

5.

6. Remove ContinentID from Country Class since BelongsToContinent is already present. [Country.java]

7. Remove getPlayerFromPlayerID since it is not used anywhere. [Player.java]

8. Rename getCountryFromCountryName() to getCountry(). [Country.java]

9. Rename getContinentFromContinentName() to getContinent(). [Continent.java]

10. Merge displayEditorMap() and displayGameMap() into one. [ShowMap.java]

11. Rename d\_PlayerList to Players to store all the players of the game. [Main.java]

12. Rename removeContinentFromContinentList() to removeContinent(). [Map.java]

13. Rename addContinentToContinentList() to addContinent(). [Map.java]

14. Rename removeCountryFromCountryList() to removeCountry(). [Continent.java]

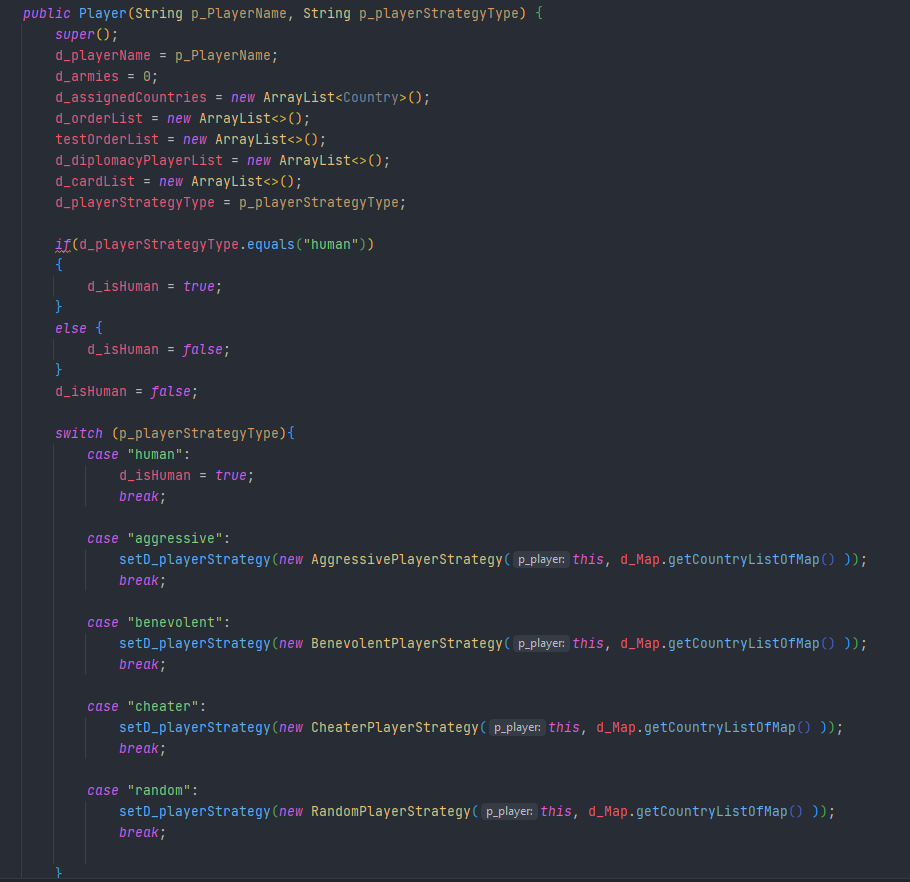
15. Rename addCountryToCountryList() to addCountry(). [Continent.java]

**Refactoring: -**

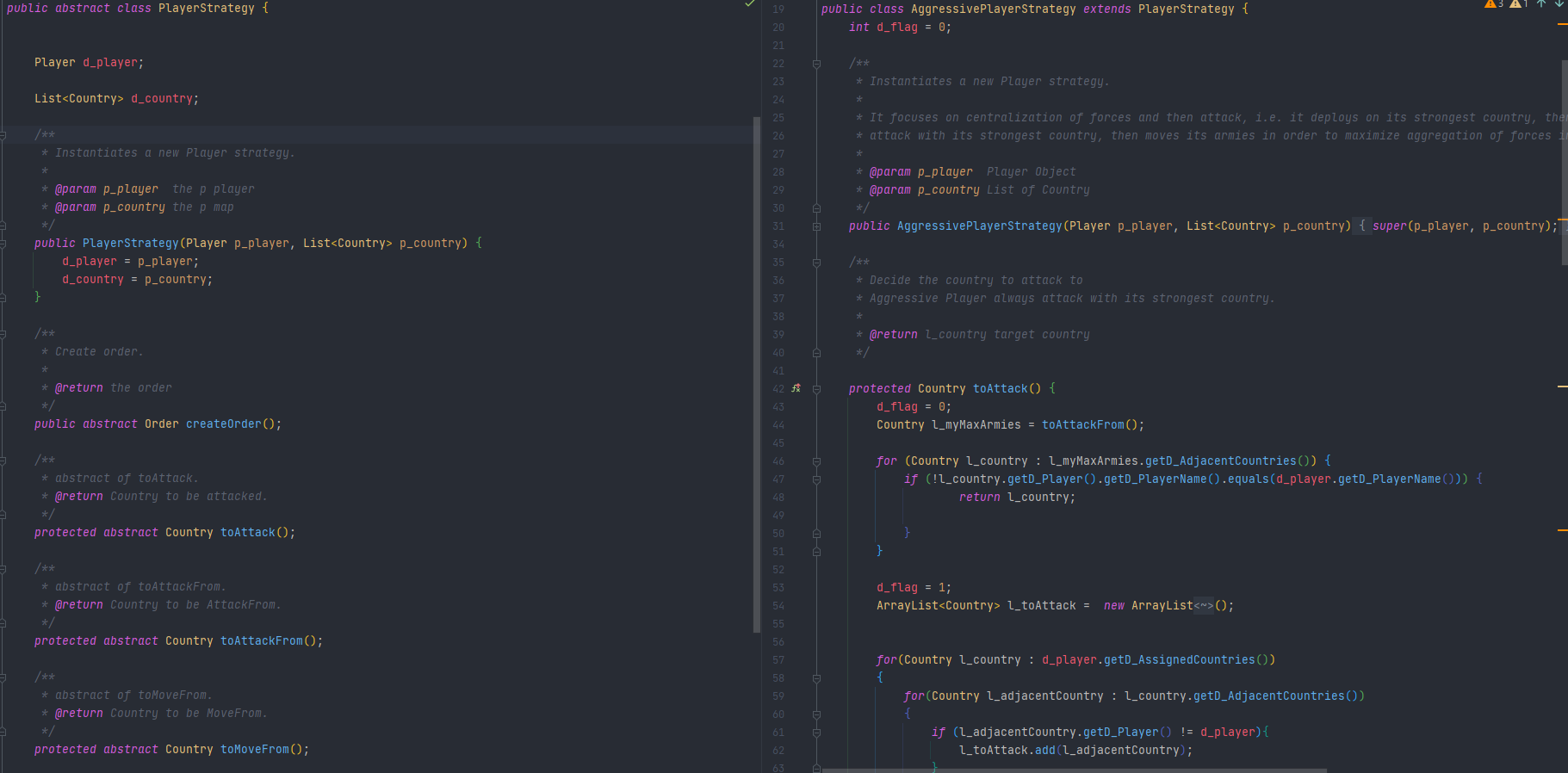
1. Implemented Strategy Pattern

Refactored our current user-driven player code so that the implementation of the Player’s issueOrder() method’s behavior is using the Strategy pattern. Then, during the main development phase, implemented different computer player behaviors using the Strategy pattern, where the strategies provide varying behavior that support the Player class to expose varying behavior when executing the issueOrders() method





* 1. Constructor added in Player Class to store strategy type (Before and After)

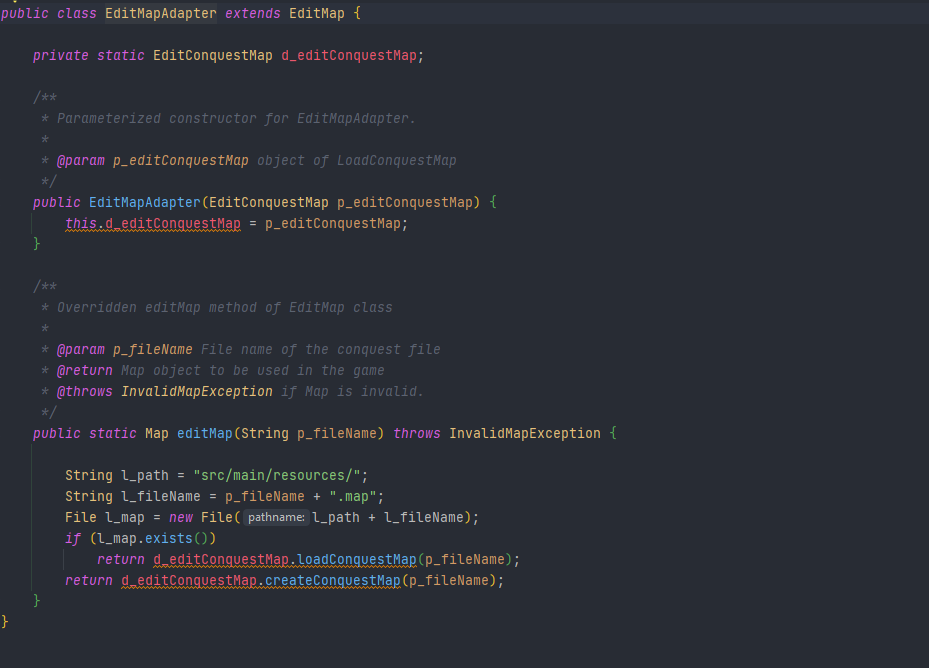


* 1. Abstract Strategy Class and Concrete Aggressive Strategy Class

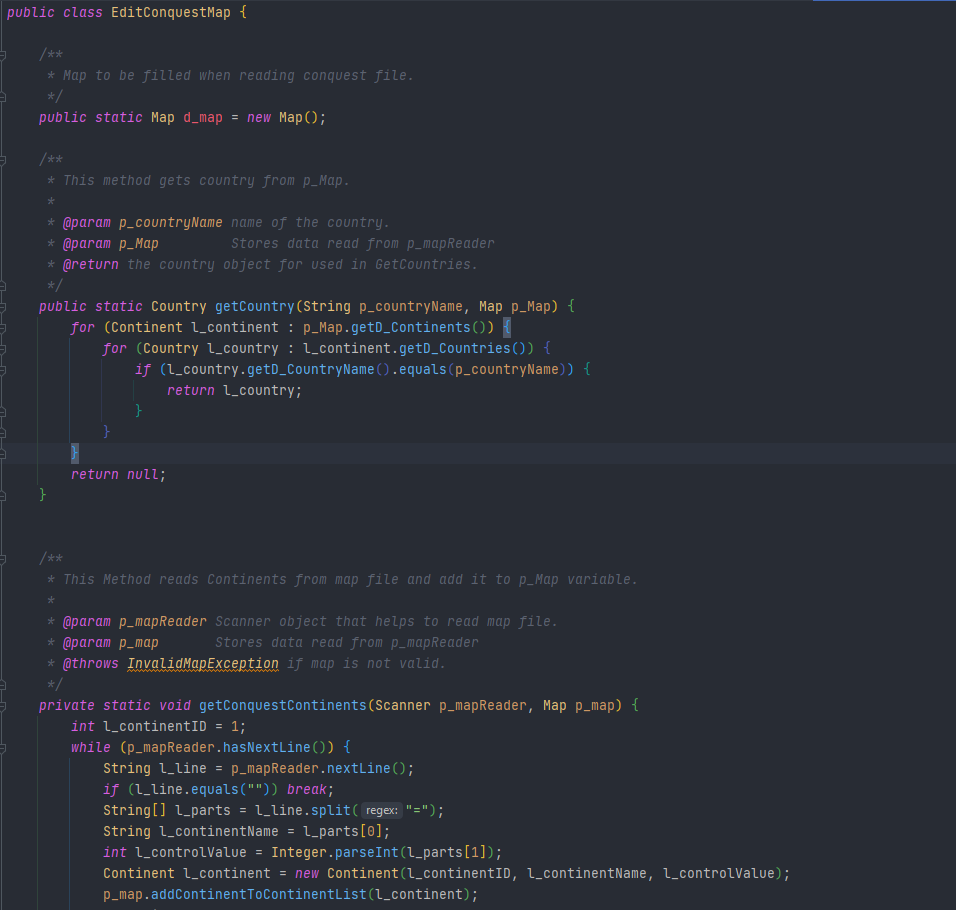
Tests : Created tests to check the validity of all kinds of strategies and their order creation.

1. Implemented Adapter Pattern

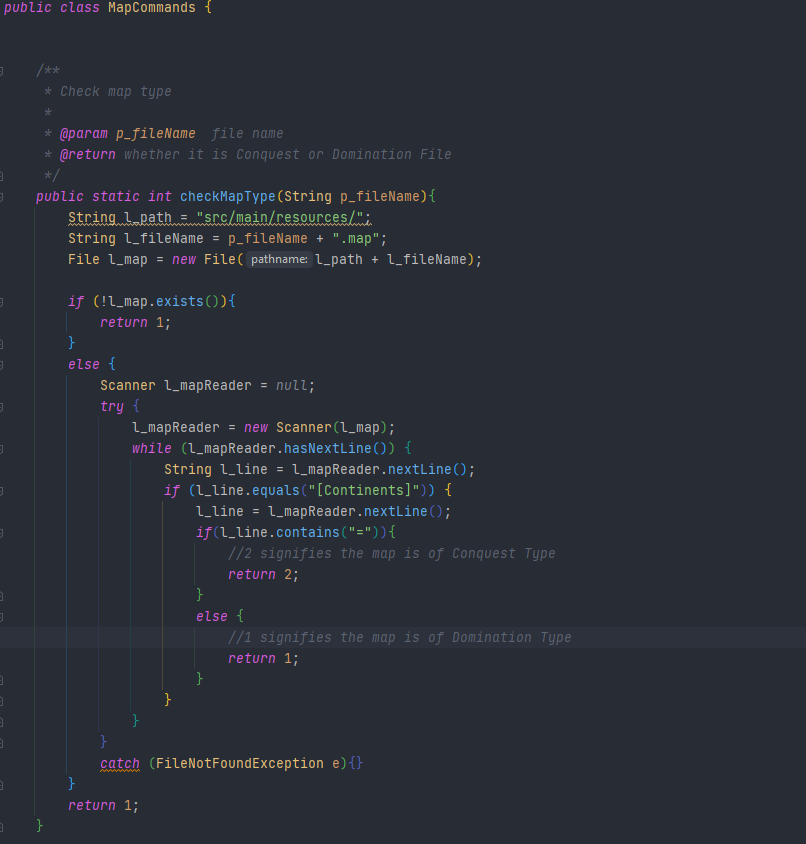
Refactored our code to use the Adapter pattern to enable the application to read/write from/to a file using the “conquest” game map format. The Game is be able to decide to use either the original “domination” file reader or the “conquest” file reader adapter when a file is opened, depending on the file type. When a map file is saved, the user is given the option as to which file format to use as output.



* 1. EditMapAdapter Class



* 1. Snippet of newer ConquestMapReader class

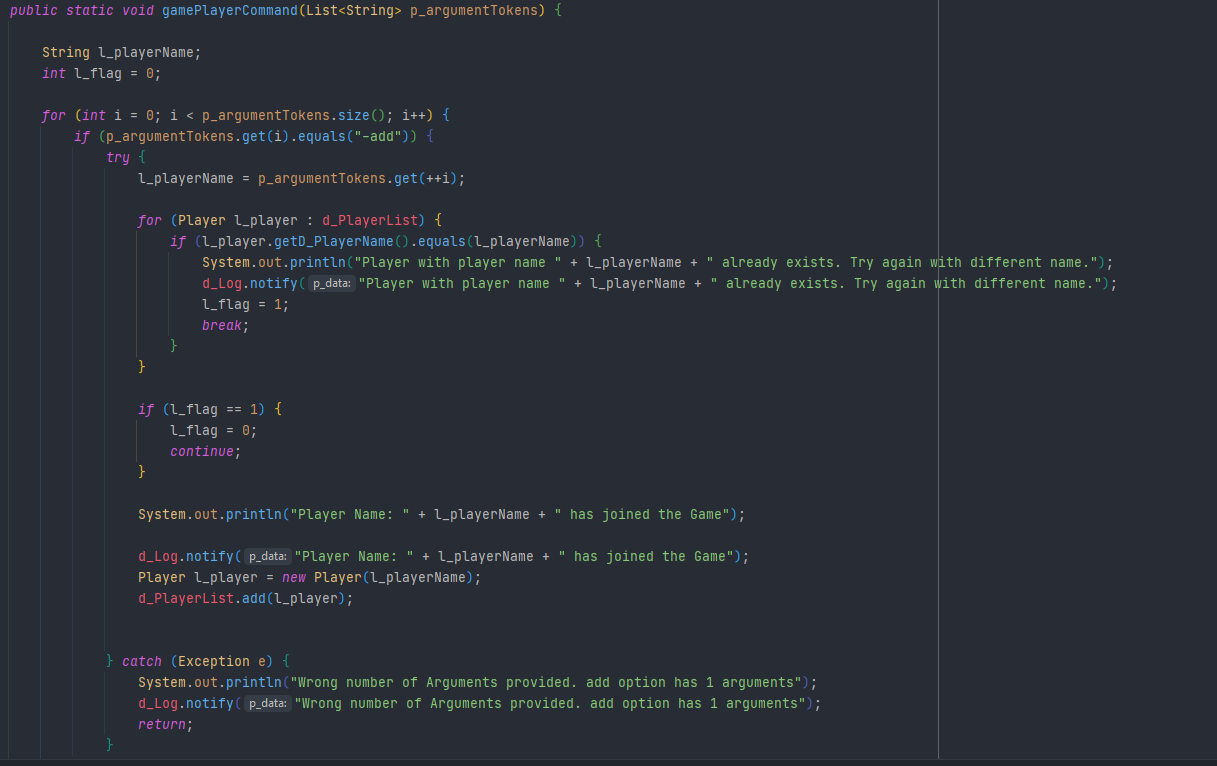


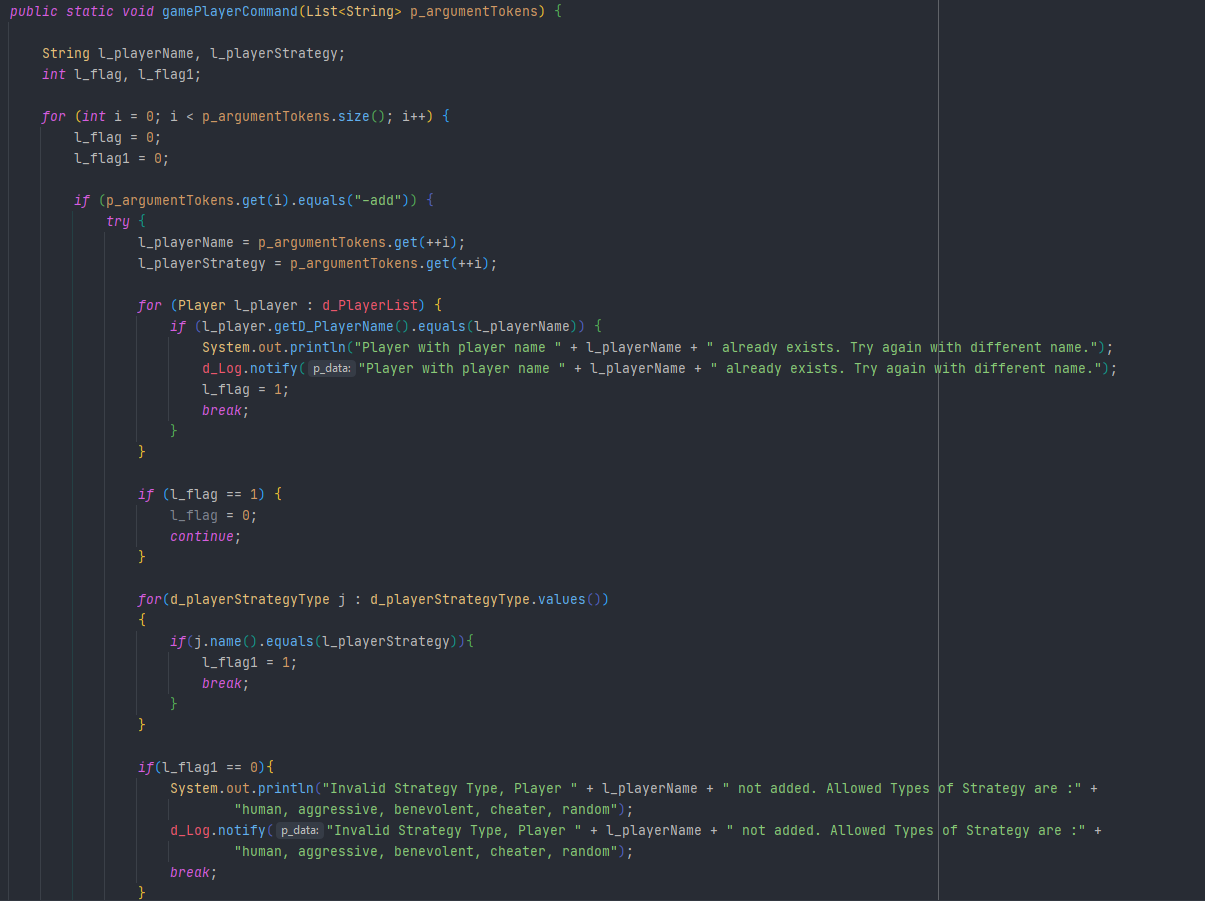
2.3 Checking format of Map and reading the map accordingly

Tests: Created Tests to load both Conquest and Domination Map files

1. Modify gameplayer command to pass strategy as an argument.

Previously gameplayer command only had one argument for both add and remove options. Since Strategy pattern is introduced, gameplayer command has been modified to take two arguments for add option. The second argument is used to set the strategy type of the player added.



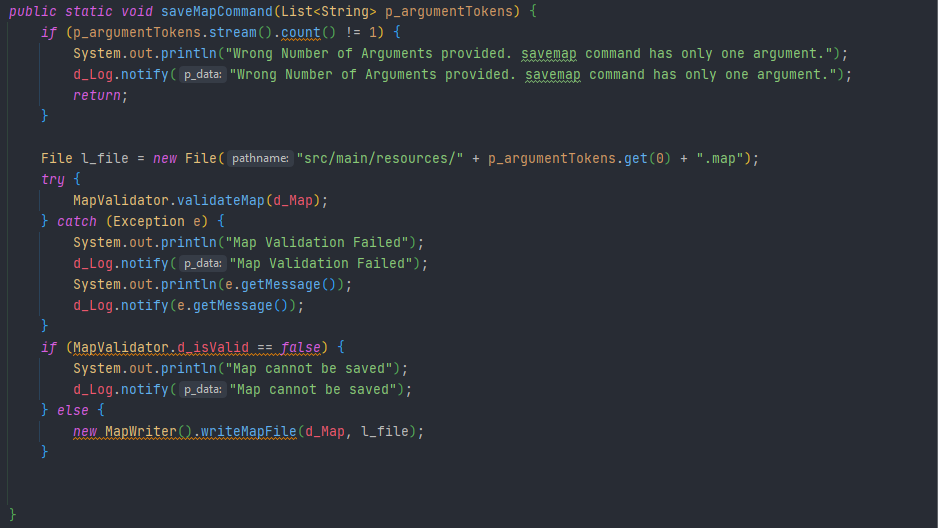


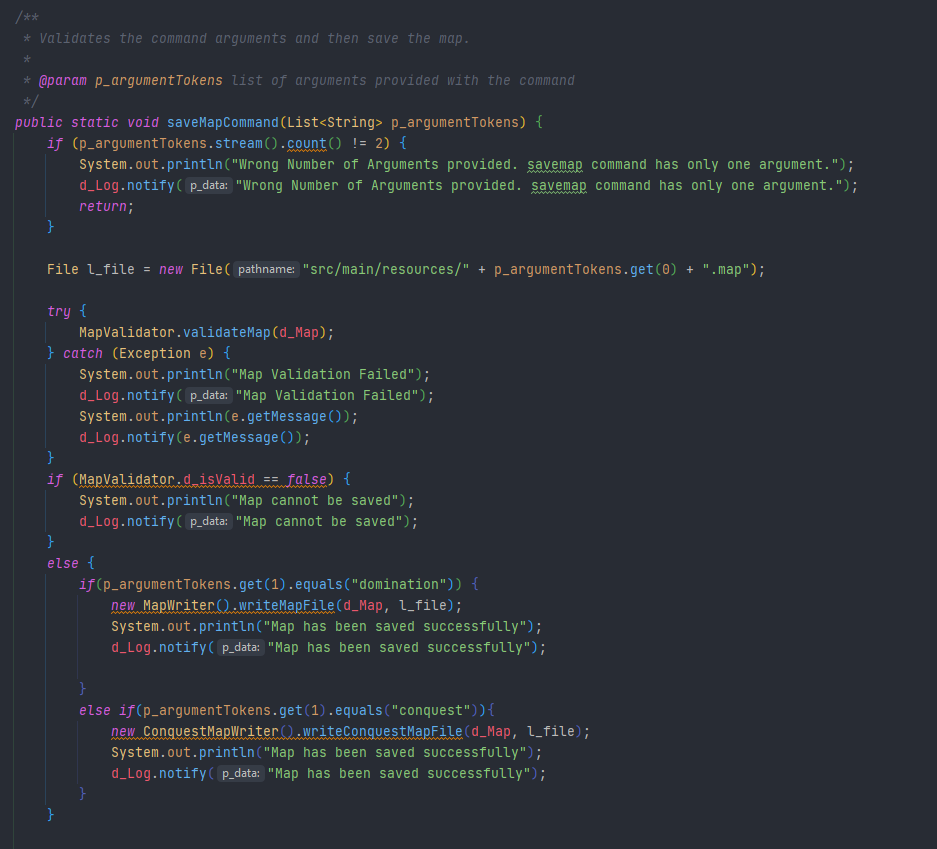
* 1. Gameplayer method (Before and After)

Tests : Create players method in testcases now have an additional argument of strategy type

1. Modify savemap command to pass Map type (Conquest or Domination) to save the map

Previously savemap command only had one argument that is file name to save the map as. Since Adapter pattern is introduced, savemap command has been modified to take two arguments. The second argument is used to set the format in which map is to be saved. The two formats are Conquest and Domination.





* 1. Savemap method Before and After

Test: Tested saving map in Conquest map format after loading a Domination Map

1. Migrated (Left to Do)

Functions including Reinforcement armies assignment, Random Number Generator for assigning countries to players randomly, assign cards to players on winning a battle all were refactored into separate class and moved under GameUtils package.

**Before**

public static Map editMap(String p\_fileName) throws InvalidMapException {  
  
 return *d\_editConquestMap*.*editConquestMap*(p\_fileName);  
}

**After**

public static Map editMap(String p\_fileName) throws InvalidMapException {  
  
 String l\_path = "src/main/resources/";  
 String l\_fileName = p\_fileName + ".map";  
 File l\_map = new File(l\_path + l\_fileName);  
 if (l\_map.exists())  
 return *d\_editConquestMap*.*loadConquestMap*(p\_fileName);  
 return *d\_editConquestMap*.*createConquestMap*(p\_fileName);  
}