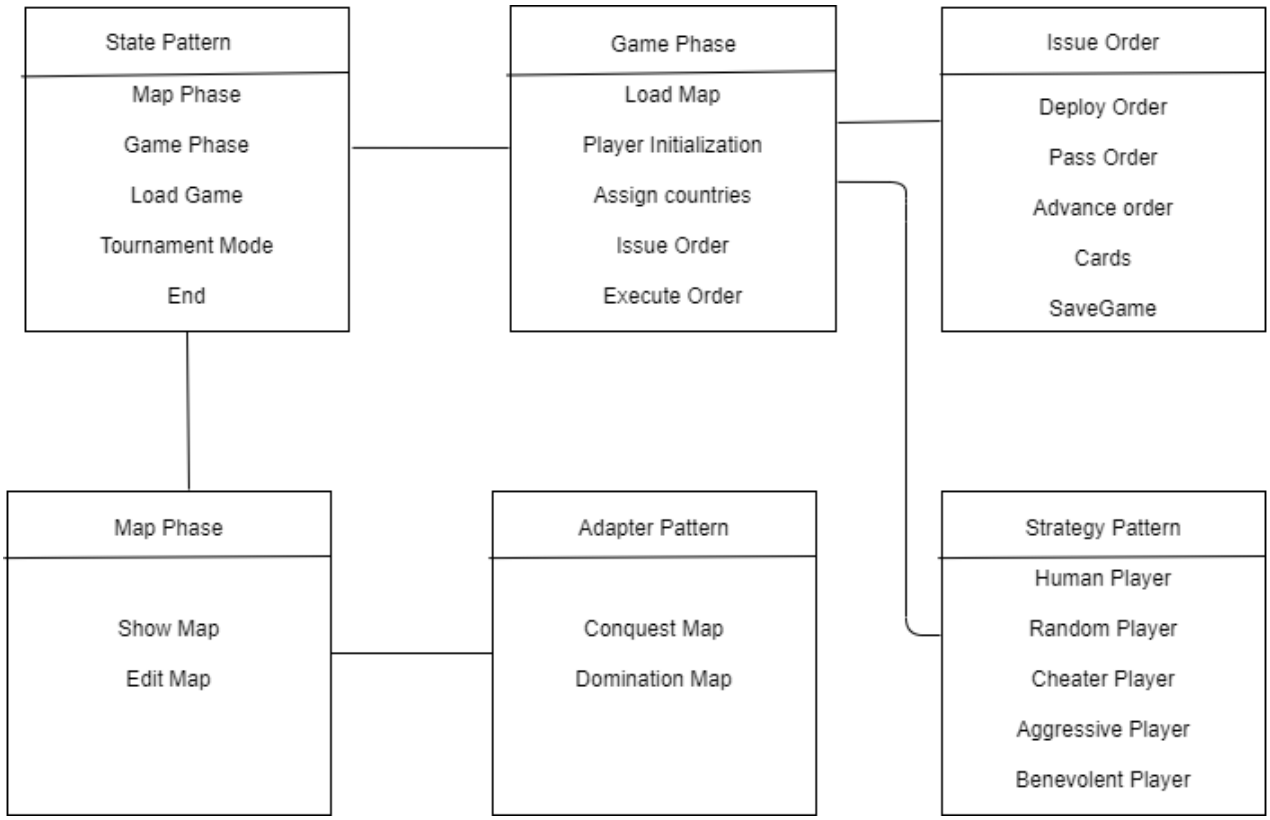
Architectural Diagram



Strategy Pattern: All the different strategies of the different players are stored at the src/main/java/org/StrategyPattern folder. So the different strategies are Random Player Strategy which is issuing the orders randomly, the Aggressive Player Strategy which is moving its army on the strongest country or is attacking the neighbouring country from the country which has his highest number of the armies, Benevolent Player Strategy is issuing the order to deploy the armies on its weakest country, Cheater Player Strategy's orders are directly executed, Human Player Strategy includes taking order from the human player. So this pattern includes the abstract class named as PlayerStrategy which contains the

abstract method like toAttack, toAttackFrom, toMoveFrom, toDefend, createOrder which are being used by the above mentioned strategies.

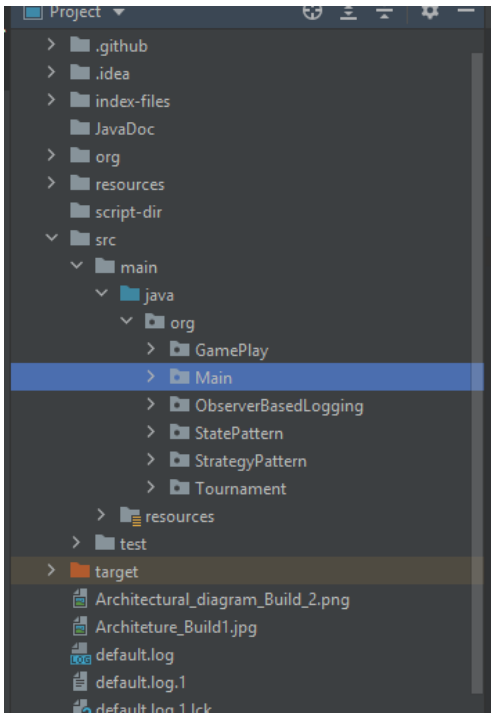
State Pattern: All the files for the state pattern are available at the src/main/java/org/StatePattern. So there is basically a main file named as the Phase.java file which consists of all the phases of the game. So basically there are main 2 main phases Map Phase and the GamePlay Phase. This is the abstract class. All the phases are extending this particular class to implement the phases. It contains many abstract functions such as editMap, showMap etc.

Command Pattern: This pattern represents that first of all the orders of the player has to be stored into the list and then the order should get executed. So this pattern is helping us to first store the orders of all the players into the queue and the orders are executed in the serial order the orders were issued.

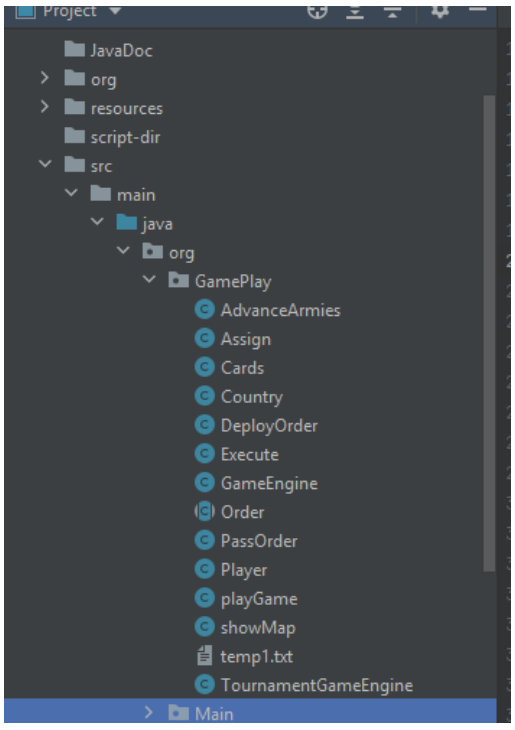
Observer Pattern: The observer pattern which is used for keeping the log file is implemented in the ObserverBasedLogging package. All the files related to the Observer pattern are available at the src/main/java/org/ObserverBasedLogging. So this folder contains the 3 files LogEntryBuffer, LogFile and LogWriter. LogEntryBuffer is extending the observable which is used for notifying the observers for the changes that has occurred. LogFile is implementing the Observer whose main purpose is to send the call the LogWriter class when ever some thing is changed. And the last one is the LogWriter which is writing the Log into the file.

Adapter Pattern: Adapter Pattern is used as the bridge between the two incompatible interface. So all the files for the Adapter Pattern are stored in the src/main/java/org/Main folder. So we've created the class to read and write the conquest map which is being used by the adapter pattern depending on the file that is read by the reader class of the Adapter Pattern.

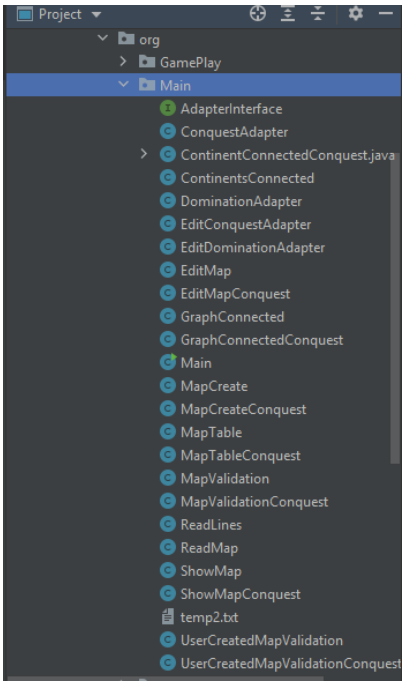
Project Structure: Inside java.org Package, there are 6 main folders which are GamePlay, Main, ObserverdBasedLogging, StrategyPattern, Tournament.



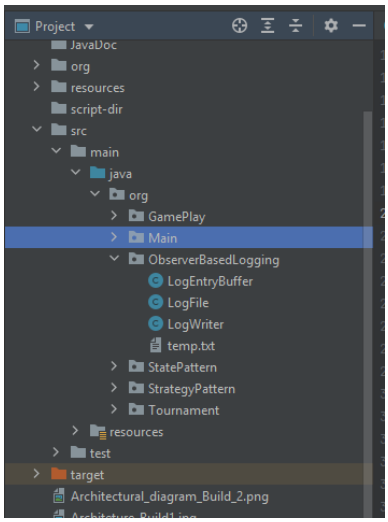
GamePlay: This is the package which contains all the files required for playing the game. This is connected to the Map Phase as we require to read the map to play the game.



Main (Map Editor Phase): This is the Map editor phase where all the operations related to maps are performed. This folder is connected with the GamePhase as gamePhase require Map Reader file to read the Map and to play the Game.

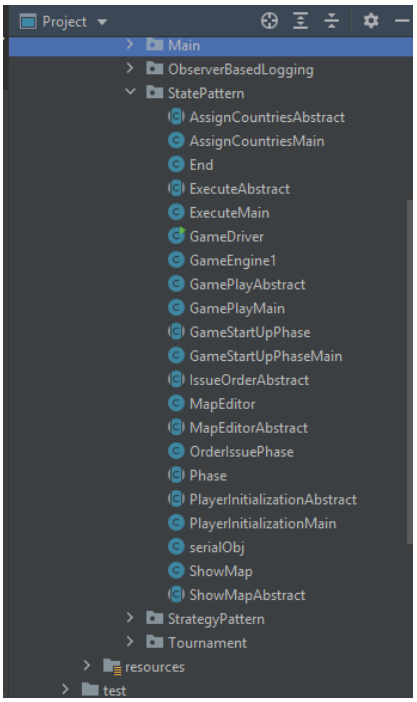


Observer Based Logging: This folder is used for creating the Log files. This folder is connected with all the folders as whenever some operations are performed by the human player or the automatic players, log is created for them.

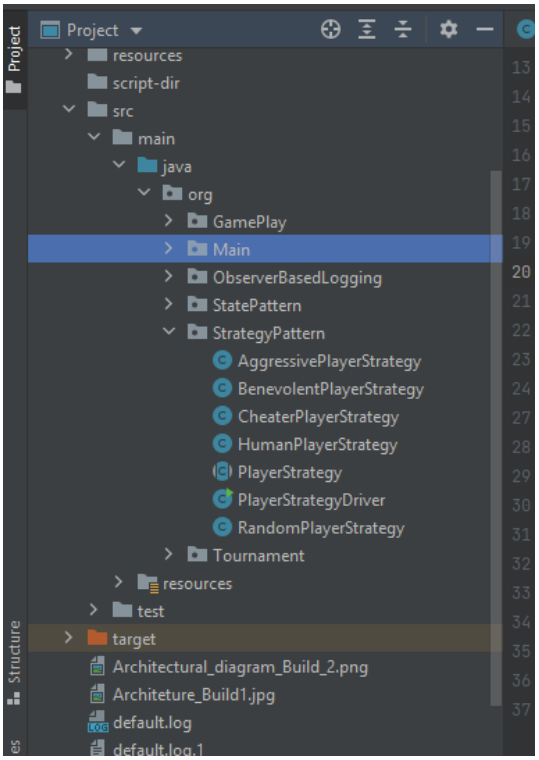


State Pattern Folder: State Pattern to change the Phases. This is connected with

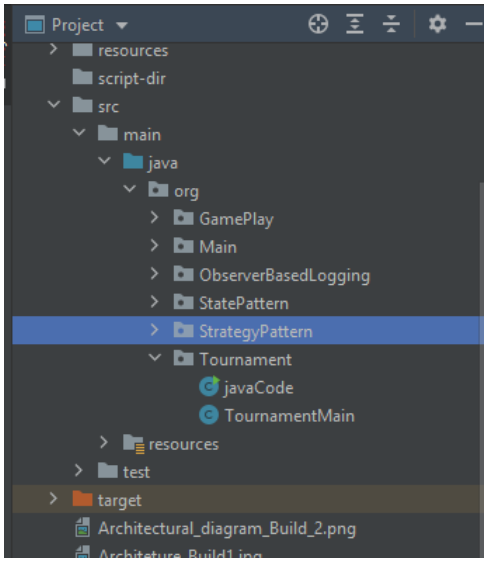
the Map editor package and the game play package.



Strategy Pattern: Package to manage the different strategies. Connected with GamePlay and Map Editor Package.



Tournament Mode: Package to start the tournament mode. Connected with Main Folder and the GamePlay



Test Cases: Folder to store all the test cases. Connected with all the other folders.