

Task 4

4.2. How patterns were used to address functional requirements

State-Defensive/Attack/Surrender country state

Composite-Forming country alliances by joining/disjoining the composite hierarchy

Factory Method-Used to create all war participants i.e., soldiers, weapons, medical supply, etc.

Prototype-Used to clone instantiated objects of the war participant hierarchy

Observer-Used to observe country state during the course of a war so that relevant objects can be notified i.e., Allies, medical services.

Iterator-Used to iterate through composite structure of allies and the relevant data structure that holds war Participants

Template - this design pattern is used for the war theatre. The WarTheatre class acts as an abstract class and the primitive functions are implemented/described in the three child classes that inherit from it.

Strategy- used to switch between types of strategies the user can use when attacking

Memento - used to store the previous state(s) of a country at a certain point during simulation to enable the user to retrieve the previous states anytime during the course of the game

Chain of Responsibility - This design pattern solves the problem of the user having to choose between a state of defend, attack, surrender and etc... It checks the amount of war participants/ weapons a user has is enough for the user to either attack, defend, ask for assistance from their alliance or surrender. It then switches between the states to find the one that is most suitable for the user given the number of weapons at their disposal.