Patterns used to address the functionality

- Memento - used to restore the state of entities(units/vehicles) after they have been helped by a medic/mechanic. If a unit is damaged in battle and they are attended to by a medic or mechanic, then the memento pattern is used to restore their original state. Also used to save the state of the entire system on the interrupts in the design mode in case the user wishes to pause the game or there is a system crash. This state can then be restored at a later stage if the user wishes to leave the program or if there is a program crash.
- Template - used to define the rough layout of an entity (units/vehicles). The general layout is used to define a person or vehicle unit and is used to create its shared methods and attributes that all types of units will have.
- AbstractFactory - used to produce the prototypes that are used in the creation of units. The abstract factory will produce the specific types of units from the template. I.e., a person is used to create soldiers, medics and mechanics. Vehicle is used to create tanks, ships and planes.
- Prototype - used by the AbstractFactory to create multiple instances of concrete products. Multiple instances of each unit will be created for the war simulation so that the number of units in an army are represented by actual objects and not just a number.
- Strategy - used to define the algorithms that units use depending on the overall state of the war engine program. Depending on the state of the program units may behave differently. The strategy pattern is used to select what methods are available to units based on the current state of the program. People can engage in peace talks or fighting, and vehicles can be used for transportation or fighting.
- State - used to control the phases of war that the war engine program is currently in. The states (as described above) will change how the program runs and how all the entities behave depending on the state currently being used.
- Decorator - used to add new methods/attributes to a unit as the war progresses. As turns take place in the war theatre in phase 3, countries will make use of the decorator to “upgrade” their units. E.g., health, armour, abilities etc. The decorator will simulate a research and development aspect of a country.

- Iterator - can be used to progress through a turn in Phase 3 (eg: attack->manage resources->evacuate personnel, etc.). The iterator will step through the methods in the war theatre to produce an outcome.
- Singleton - used to create a single instance of the war engine as there will only be one instance of this class used as it is the overall program.
- Command - might be able to use this in diplomacy/negotiations?
- Observer - the observer design pattern will act as the UN in the simulation. The observer will be notified when battles take place and then the outcomes of peace talks and negotiations.
- Composite - The lists of attacking/defending countries in a war theatre will be a composite, so that we can call each of their functions at the same time.
- Builder - the WarTheatre will have an attackingSquad, and a defendingSquad, these squads will either be SeaSquads(that can consist of navy and air units) or LandSquads(that can consist of troops and air units). These squads will be built using a SquadBuilder