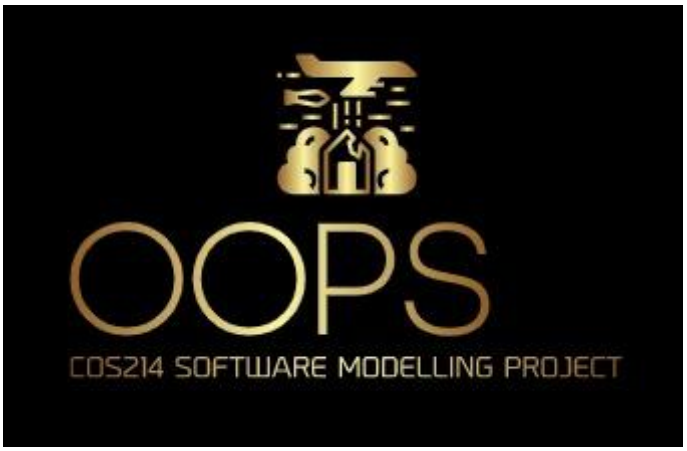
COS214 Project - Report

“War, war never changes…”

Please note that this document includes the research brief and report on design patterns as stipulated by Task 4. UML class, sequence, object, communication, state, and activity diagrams can also be found in the “Report” directory.

Link to the Google Doc version:
https://docs.google.com/document/d/17LYU0BKzvOeFWF8SqTmUtS7nt5N8vYokSYuyeVtK-Us/edit?usp=share_link



Research Brief

Effect of GDP on War:

Within our project, we will be using the GDP of a country, or the monetary value of total assets a country has as the main determining factor of many outcomes of war. Although it is possible for poorer countries to beat more wealthy countries in a war (e.g. Vietnam war) it is less often the case. This is due to the high cost of war, from supply costs, to soldiers salaries, to the cost of development of war technologies. This is explained in the provided article by Tejvan Pettiger: https://econ.economicshelp.org/2010/03/economic-cost-of-war.html. As well as this the financial state of nations affects how they approach a war with poorer nations perhaps being more defensive, and for this reason an EconomicState class is created enforcing a state design pattern based on the wealth of a country.

Battalions of an Army:

Within our project the units of an army are most often stored within a battalion. These battalions are designed similarly to Combined Arms Battalion, (As explained on the Battle Order website:https://www.battleorder.org/usa-cab-2016#:~:text=The%20Combined%20Arms%20Battalion%20(CAB,2%20Mechanized%20Infantry%20Rifle%20Companies.) as a single battalion can contain many different types of units, such as Land vehicles to Navy Soldiers with the units of a battalion fighting within their corresponding war theatre of a battle.

Theatre of War:

A war or battle occurs within its corresponding theatres. These theatres of war may include the land, sea or air theatres whereby the important military events and battles occur, as defined at https://en.wikipedia.org/wiki/Theater_(warfare)#Theater_of_war. Different units within a War may interact with different war theatres but often with far less impact. For example an air unit may interact with the sea theatre of war, but it will have less impact on said theatre than the sea units themselves.

Production of Units:

Obviously, for the creation of a battalion, the creation of units such as soldiers or vehicles are required. Therefore within our project we have created a UnitFactory that can produce soldiers or vehicles for land, sea or air purposes, and this is accurate to the actual development of soldiers in war as the different branches of the military, that being the Air force, navy and the army as explained on https://www.indexmundi.com/south_africa/military_branches.html, all train soldiers separately and for separate tasks.

Non-combat units:

Within a war there are many people caught within the theatre of war who are not fighting participants, such as medics and civilians. We have modeled this group of people separately as it is a war-crime to kill either of these types of people within a war and thus they are to be managed seperately. The spoken about war crimes and laws protecting against these crimes can be found at:https://dayofdifference.org.au/i-medical/is-it-a-war-crime-to-kill-a-medic.html)
(https://www.hrw.org/reports/2002/isrl-pa/ISRAELPA1002-04.htm#:~:text=Willful%20killing%2C%20that%20is%2C%20intentionally,wounding%20victims%2C%20are%20war%20crimes.&text=Persons%20who%20commit%2C%20order%2C%20or,humanitarian%20law%20for%20their%20crimes.

Phases of war:

We have modelled a total of 7 phases of a war for each war that occurs, with the early, middle, and late phases being the main phases, as defined at https://www.researchgate.net/figure/Chronological-Phases-of-Conflict-Intensity-Levels-Swanstrom-and-Weissmann-2005-The_fig1_318960397. These phases do have stages within each one with the early and late phases having Unstable Peace, Open Conflict and Crisis phases while the mid phase is the main war phase. These phases of war ultimately determine the tension between states and frequency of battles leading to the war, and occuring at the end of the war. These characteristics are defined within each phase as Unstable peace having building of tension, Open Conflict having visible conflicts, Crisis phase having extremely high tensions and the corresponding states being at the brink of warfare with rare battles occurring and the War phase being an official War.

Alliances:

Within a war, multiple countries can form an alliance with wars often consisting of two main alliances fighting against each other, such as the allies vs the axis in World war 2. These alliances could change the tide of battle with allied countries providing each other with supplies, money, or participation within battles to win wars as described in: (https://www.encyclopedia.com/social-sciences-and-law/sociology-and-social-reform/sociology-general-terms-and-concepts/alliances#:~:text=The%20primary%20function%20of%20an,advance%20collective%20and%20individual%20purposes.). As well as this there are neutral countries within wars that do not participate within the war itself, but said neutral countries can often join alliances if they do decide to fight in the war.

Transport:

The operation of transport lines, whether it be rail, car, or plane within war is of utmost importance. Without supply lines, food, water, and ammunition wouldn't be able to be delivered to the battlefield eventually rendering the units useless as they cannot continue to fight (https://www.nato.int/cps/en/natolive/topics_61741.htm). As well as this medical transport is needed to remove wounded soldiers from the battlefield and heal those who are injured/sick.

Military General:

Within a country, there is usually a military general who commands and controls the operations of the military branches. The military general can of course order the armies of a country to do actions as necessary, whether that be an attack on a transport line, the entering of a battle, surrender, or any other desired move. The definition of a general can be found at: https://www.britannica.com/topic/general.

Strategies of an Army:

Often a military general chooses the specific strategy for an army to follow, whether they be offensive, defensive or neutral (Not focused on either offense or defense) as described at: (https://en.wikipedia.org/wiki/List_of_military_strategies_and_concepts). Such strategies often determine the outcome of a War, for example a country currently on the losing end of a war wouldn't choose to be offensive while the enemy attacks them and if a defensive strategy is chosen it can change the tide of battle. By selecting different strategies, it obviously provides benefits for that mode of battle. However if for example an offensive strategy is chosen, the defense of that army would be weakened and vice versa.

Upgrading of Supply or Unit Production:

Should a country have enough resources (determined by gdp) to improve its supply or military unit production, it can be done. This is done by producing more, higher quality supplies, and the production of more, better trained military units. This results in a stronger army that can help win battles, and subsequently the war.

Report on Design Patterns Employed

The following is a report highlighting the core mechanisms of the system and the design patterns used to implement them.

The complete class diagram can be viewed here:
ClassDiagram-V22.pdf

And is also included as both a pdf and as a VPP project in the “Report” directory.

Effect of GDP on a Country’s decision making (State):

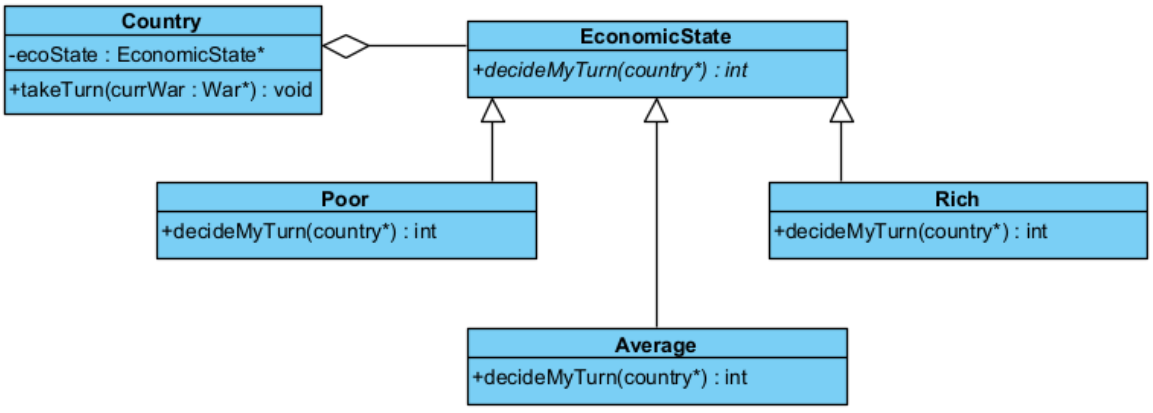
In real life, countries need to take into account their current economic position when deciding to execute tasks pertaining to war (such as raising an army or improving their capacity to produce supplies). To address this required functionality, a State pattern has been implemented to manage and track a Country’s current economic state. The decision making process is simulated by a function that returns a numerical value corresponding to a specific decision, and each Country (acting as the context participant) has an EconomicState object which is used to determine the possible numerical values to be returned, based on how positive or negative an economic position a Country is in. The use of this pattern makes it easy to dynamically track a Country’s economic position over time and have its behaviour vary depending on its economic position.

Participants:

Context: Country

State: EconomicState

ConcreteState: Rich, Poor, Average



ArmyComponent, Battalion, Soldiers and Vehicles (Composite):

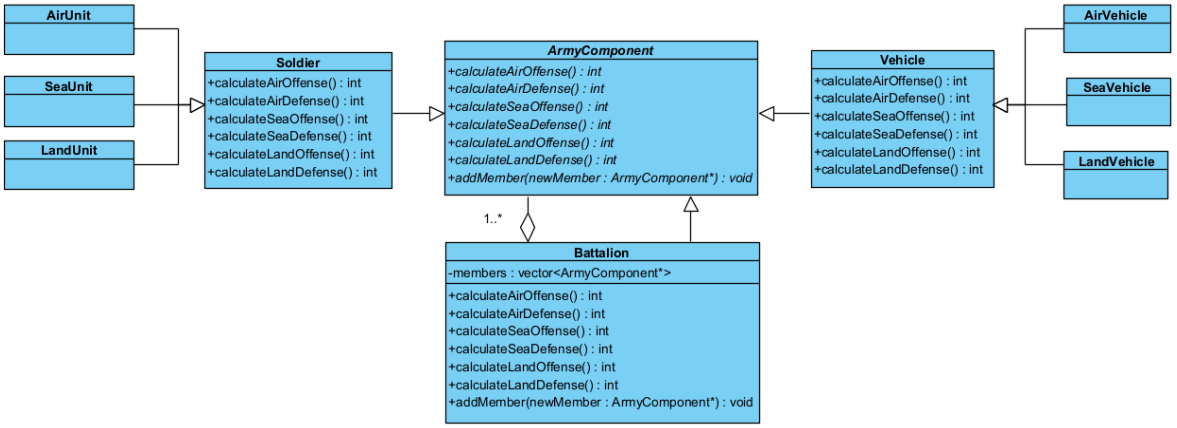
Our system requires being able to construct structured battalions and organisations of individual soldiers and vehicles as well as treat these composite entities as entities in their own right. This problem lends itself to the use of the Composite pattern, as this will allow us to construct composite army component objects, consisting of other army component objects. For example, one can easily construct a battalion army component which consists of a number of soldier and vehicle army components, and treat that battalion as if it were a single entity. The typical structure of an army as used in our system is to consist of two battalions (each consisting of a number of soldiers and vehicles) as well as a number of individual soldiers and vehicles, not belonging to a specific battalion.

Participants:

Component: ArmyComponent

Composite: Battalion

Leaf: Soldier (AirUnit, SeaUnit, LandUnit), Vehicle (AirVehicle, SeaVehicle, LandVehicle)



UnitFactory (AbstractFactory):

The units of the battalions also need to be created in such a way that allows the creation of both vehicles and soldiers, as well as specialisations of those vehicles or soldiers, for either air, land or sea. This creation process can thus become complicated when a large number of different types of entities are being created. The Abstract Factory design pattern lends itself to this scenario and alleviates some complexity as it allows one to create a uniform factory hierarchy, capable of producing either a soldier or a vehicle, with separate ConcreteFactories for either air, land or sea. This structure simplifies the construction of multiple Soldiers and Vehicles as well as allows factories to track how much of their budget for creating units has been spent, such that each unit has an associated cost and if the costs exceed the factory's budget (as stipulated by a Country) it will refuse to produce more units.

Participants:

AbstractFactory: UnitFactory

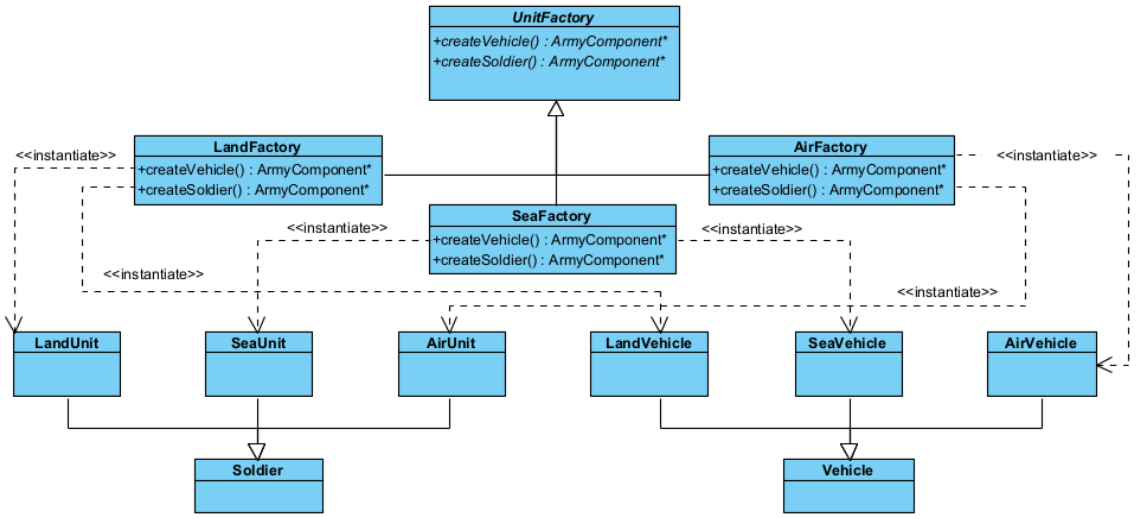
ConcreteFactory: SeaFactory, AirFactory, LandFactory

AbstractProductA: Soldier (which is a child class of ArmyComponent)

AbstractProductB: Vehicle(which is a child class of ArmyComponent)

ConcreteProductA: SeaUnit, AirUnit, LandUnit

ConcreteProductB: SeaVehicle, AirVehicle, LandVehicle



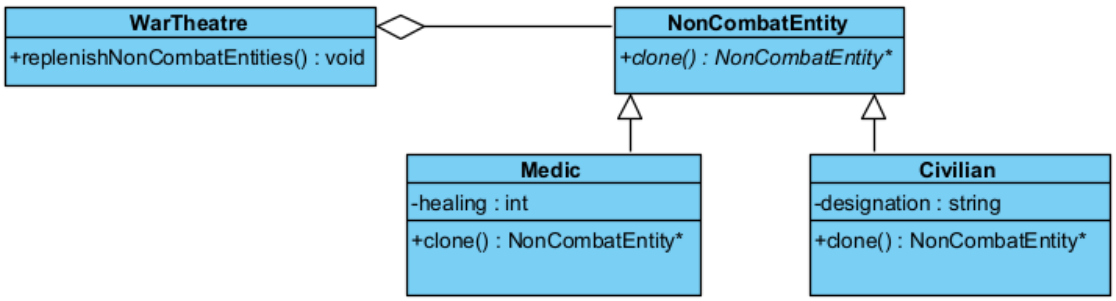
Non-combat units (Prototype):

Third-party medics operate within theatres of war to heal the wounded, while civilians exists with the sole purpose to avoid being caught in the crossfire (a saddening reality). Over time, these entities leave and join the war theatre over time, and in terms of programming, it would be simplest to clone these entities as more join the theatre (since each entity of each type is almost identical). The Prototype design pattern provides for this functionality. Hence, we can easily use a reference NonCombatEntity object of type Medic or Civilian and simply clone that object when it is necessary to replenish the non combat entities in a specific war theatre.

Participants:

Prototype: NonCombatEntity

ConcretePrototype: Medic, Civilian



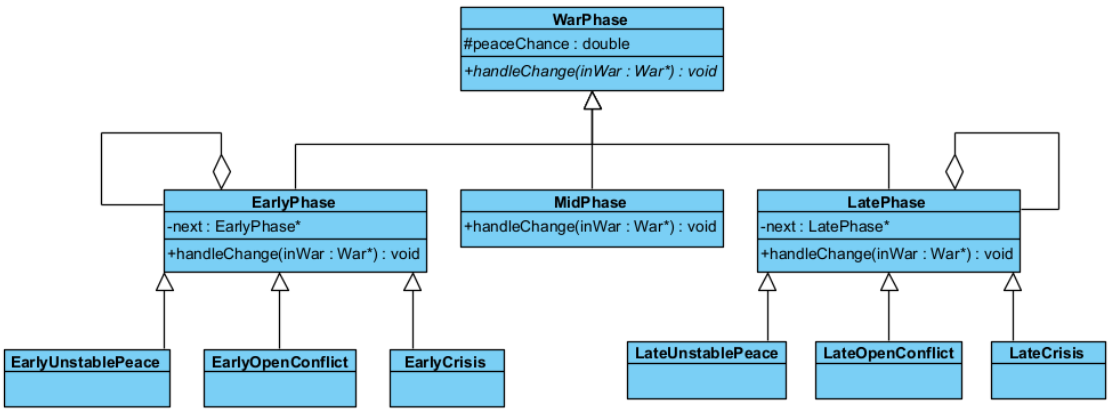
Phases of war (Chain of Responsibility):

In our system, we have elected to model a total of 7 phases of a war. Namely, the early, middle, and late phases being the main phases. Some of these phases have stages within themselves. The early and late phases have Unstable Peace, Open Conflict and Crisis phases while the mid phase is the main war phase (and consists of no sub-phases). These phases of war affect the chance of a war being resolved peacefully with the chance for peace decreasing throughout the early phases, it not being possible in the mid phase (as war would currently be happening), and the chance for peace decreasing through the late phases. Since the war phase is effectively a handler for certain functionality that takes place in the war (namely deciding whether peace has occurred) and that all possible phases need to be available to handle a request from the War at any time, this structure lends itself to a Chain of Responsibility pattern, whereby each specific phase of the war is a ConcreteHandler. Naturally, the appropriate phase (depending on how long the war has been going on for) associated with deciding whether peace has been reached will handle the request each time.

Participants:

Handler: WarPhase

ConcreteHandler: EarlyPhase (with specialisations EarlyUnstablePeace, EarlyOpenConflict, EarlyCrisis), MidPhase, LatePhase (with specialisations LateUnstablePeace, LateOpenConflict, LateCrisis)



Transport (Mediator):

In order to model real life, our system requires the modelling and use of transport lines for transporting medical and ammunition supplies from an army's home country, to the army itself. Since the Country class and Army class are tightly coupled, in order to avoid further complexity, a Mediator pattern is employed here to coordinate the transport of supplies between Corresponders (which is the parent class of both Country and Army). The Transporter mediator is then responsible for managing the interaction that takes place between the classes when supplies are transported. Even though in our system this process has been implemented such that countries send supplies to armies and the reverse is not possible, if at a later date one wanted to implement this reverse functionality, the use of the Mediator pattern would allow for such an expansion (whereby armies can send captured or surplus supplies back to their country of origin). Furthermore, these "transport routes" can be destroyed by an enemy army, (ie: the mediator attribute has been set to null) meaning that a country can no longer communicate with/send supplies to its army.

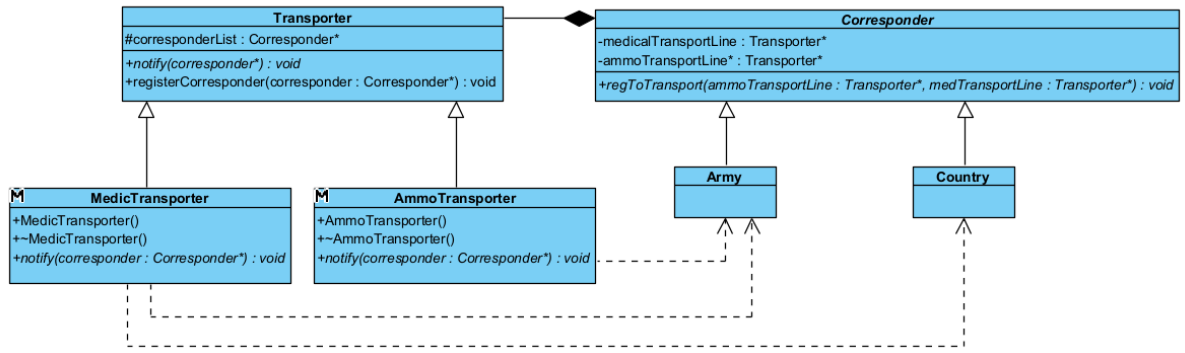
Participants:

Colleague: Corresponder

ConcreteColleague: Army, Country

Mediator: Transporter

ConcreteMediator: MedicTransporter, AmmoTransporter



Military General (Command):

In an effort to model real life as well as provide a mechanism for an army to be instructed to perform certain tasks, the system requires a way to be able to send requests from a Country to its Army. Once again, since the Army and Country classes are already tightly coupled, it is desirable to make use of a MilitaryCommander class that is able to issue commands/requests to the army on behalf of the Country. The Command pattern lends itself to this type of interaction and allows for these requests to be encapsulated as objects, meaning that for example a command for an army to enter a war theatre can hold a pointer to the target theatre to enter as well as the army to enter into that theatre.

Participants:

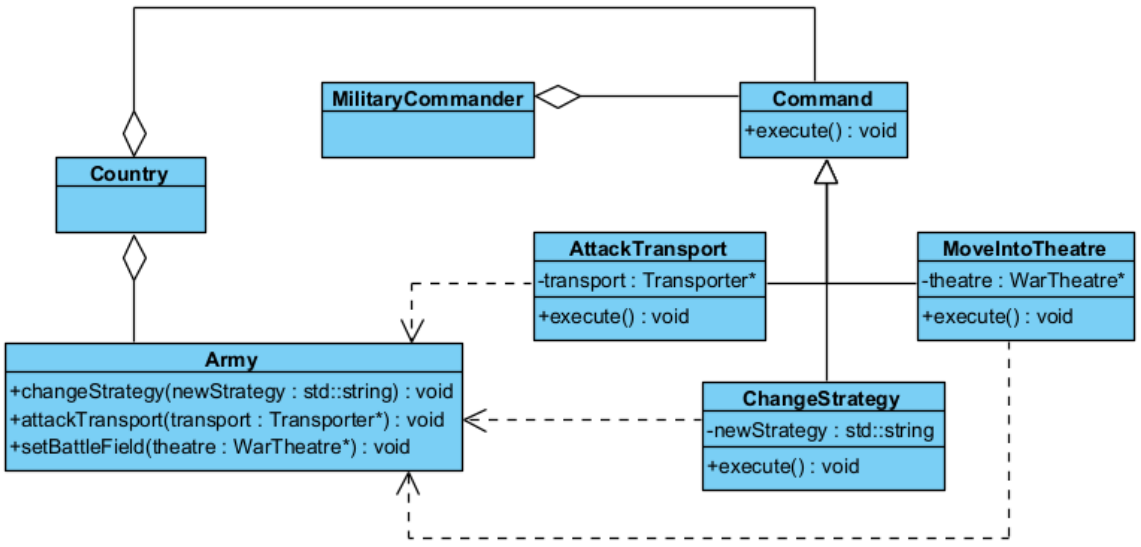
Invoker: MilitaryCommander

Command: Command

ConcreteCommand: MoveIntoTheatre, AttackTransport, ChangeStrategy

Receiver: Army

Client: Country



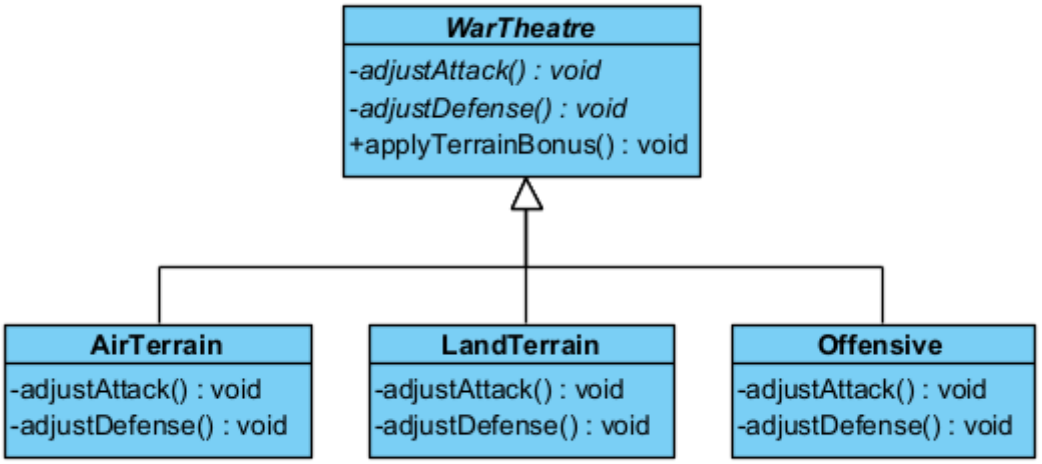
War Theatres and Terrain (Template Method):

Battles take place in multiple areas and terrain during a war. For example, naval battles at sea or trench warfare on land. In our simulation, we have modelled War to consist of land, air and sea theatres where rival armies fight head to head. Furthermore, different terrains lend advantages to troops and vehicles that are suited for that specific terrain, and naturally increases and decreases are to be applied to different armies' stats depending on how its constituent components suit the terrain. To manage this, attack stats and defence stats need to be adjusted during conflict in the terrain, and how these stats are adjusted is dependent on the terrain. Hence, a Template Method is used that calls the appropriate primitiveOperations (to adjust defence and attack) is implemented, while how these specific adjustments are individually made is implemented in the LandTerrain, SeaTerrain and AirTerrain classes that inherit from WarTheatre.

Participants:

AbstractClass: WarTheatre

ConcreteClass: AirTerrain, SeaTerrain, LandTerrain



Army Strategies (Strategy):

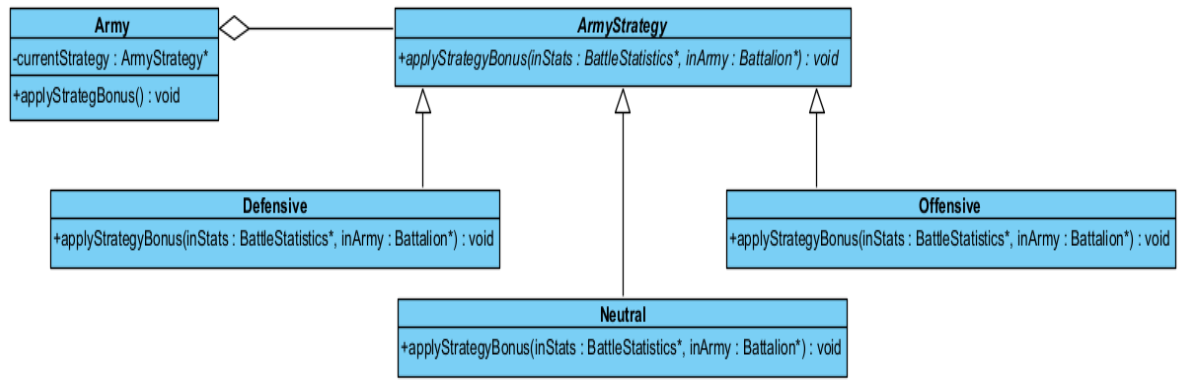
Armies adopt different strategies during battle in an effort to gain an advantage or achieve a specific goal, increasing their likelihood of victory. To model this, we have (appropriately) employed the use of the Strategy pattern such to extract the logic behind the adjustments that need to be made to an army based on its current strategy (Offensive, Defensive or Neutral). This also allows an army to dynamically change its strategy over time or between turns on the instruction of a MilitaryCommander, while the BattleStatistics adjustments (determined by the Strategy) are performed dynamically as and when needed.

Participants:

Context: Army

Strategy: ArmyStrategy

ConcreteStrategy: Defensive, Offensive, Neutral



Supplies and Supply Factories (Factory Method):

To support armies in their battle endeavours, they require medical and ammunition supplies to heal the wounded at the end of each offensive or attack the rival army with ballistics respectively. Before a country can send these supplies to their army, they need some mechanism for creating these supplies and to track how much budget is allowed for making these supplies. This problem has been addressed by using a Factory Method to create a uniform interface for creating medical supplies (MedicalFactory) or ammo supplies (AmmoFactory). Not only does this simplify the creation of Supply objects, it allows for the factories to be able to track the amount being spent to create these supplies, as well as the level (which can be improved by spending GDP to upgrade the factory) of the factory, which can influence how the quantity of supplies can be produced at a time.

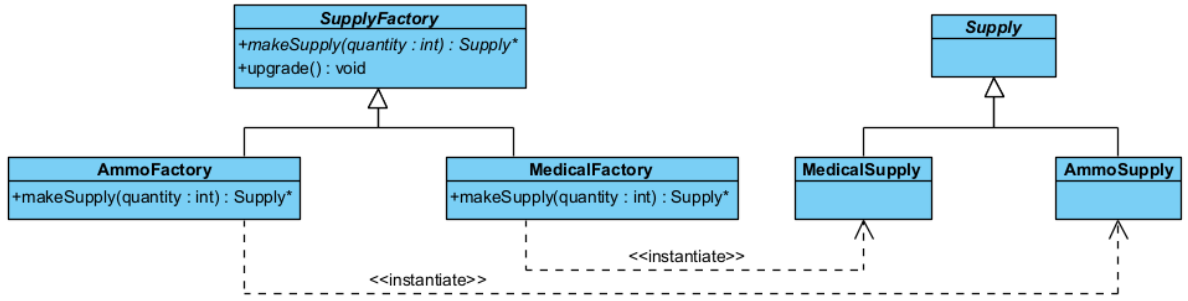
Participant:

Creator: SupplyFactory

ConcreteCreator: AmmoFactory, MedicalFactory

Product: Supply

ConcreteProduct: AmmoSupply, MedicalSupply



Raising Armies (Builder):

Armies are complicated objects that consist of multiple components (namely battalion, individual soldiers and vehicles, medical and ammo supplies) and the construction of armies is best abstracted to some external class. The Builder pattern is employed to accomplish this. We have an ArmyBuilder (which is the traditional GoF Builder hierarchy compressed into a single class) which consists of various methods each responsible for constructing the various constituent components of an army, and then an ArmyDirector, which calls these methods in the appropriate order to construct the entire army. By making use of this pattern, we have separated the creation of an army from the army itself, meaning that in future could modify the construction process to create armies if needed. Note that the reason for the compression of the builder hierarchy is that in our implementation, there exists only one type of Product (Army) class in our system, and thus expanding the Builder into a hierarchy would be redundant since we would only have one ConcreteBuilder.

Participants:

Director: ArmyDirector

Builder/ConcreteBuilder: ArmyBuilder

Product: Army