

# THE WAR GAMES

Segmentation Fault



Github Repository: <https://github.com/sloththedev/cos214-project>

Link to Document:

[https://docs.google.com/document/d/1wIGAnmKHzTf5SyiAyfpMH3J3QU\\_R3UXf-i8mV9dXS0/edit#heading=h.4cdojdwsic](https://docs.google.com/document/d/1wIGAnmKHzTf5SyiAyfpMH3J3QU_R3UXf-i8mV9dXS0/edit#heading=h.4cdojdwsic)

# Task 2 - DESIGN

Functional Requirements:	<b>3</b>
The War Simulator will:	3
Provide a God of War:	3
Provide a Player:	3
• The CPU will:	3
• The User will:	3
• The GUIUser will:	3
Provide a Map	3
Provide a means to interact with the Map (extends GoW)	4
Provide Areas that:	4
• Can have factories which provide:	4
Provide an internal Battle Engine:	4
Provide platoons that:	4
Provide a means to construct platoons:	4
Design Patterns	<b>5</b>
UML	<b>6</b>
Activity Diagrams	<b>7</b>
Main War Loop	7
Alliance turn	8
Sequence Diagrams:	<b>9</b>
Battle Sequence Diagram	9
Change Strategy	10
Request Factory from adjacent areas	11
Builder Sequence Diagram:	12
Communication Diagrams	<b>13</b>
Basic Visitor Functionality (countVisitor used to check if war ended in this case)	13
Retreat from Battle (Attackers)	13
Object Diagrams	<b>14</b>
Area during battle object diagram	14
Platoon object diagram	14
State Diagrams	<b>15</b>
Area state diagram:	15
Platoon State diagram:	15
Appendix:	<b>16</b>
Pre-Initial UML	16
Working UML	17

## Functional Requirements:

### The War Simulator will:

#### Provide a God of War:

- Select simulation modes
- Initialisation of sides
- Let users make strategic decisions in a turn-based approach
- Allow users to view the map
- Allow users to inspect areas

#### Provide a Player:

- Either a CPU or a User:
- The CPU will:
  - Initialisation of the game (no user input)
  - Choose an action each round (no user input)
    - Move platoons,
    - Request factories
    - Attack transport routes
  - Make decisions in battles (no user input)
    - Change strategy
    - Retreat
    - Request reinforcements.
- The User will:
  - Initialise the game through Terminal
  - Make decisions on the actions to be performed each round
  - Make decisions about where to move platoons, add factories to areas/request from factories and when to attack transport routes
  - Make decisions about the platoon's strategy mid-battle, this includes when to change strategy, when to request reinforcements and when to retreat.
- The GUIUser will:
  - Make decisions on the initialization of the game through GUI
  - Make decisions on the actions to be performed each round
  - Make decisions about where to move platoons, add factories to areas/request from factories and when to attack transport routes
  - Make decisions about the platoon's strategy mid-battle, this includes when to change strategy, when to request reinforcements and when to retreat.

#### Provide a Map

- That contains and manages areas and transport routes
- That contains a reference to all countries (active and inactive)
- That manages the creation/destruction of transport routes between areas.
- That manages the requests for additional factories to specified areas.

- That manages the requests for reinforcements by platoons
- That manages the ability of platoons to move between areas
- That controls the resolution of battles

Provide a means to interact with the Map (extends GoW)

- By allowing players to view the map
- By allowing players to get detailed information about specific areas on the map
- By allowing players to march platoons in and out of areas
- By colouring the map to show the progress of the war
- By managing the creation of transport routes
- By allowing users to destroy selected transport routes

Provide Areas that:

- Consists of both Air and Land Theatres of War
- Can be occupied by an alliance/country by marching in
- Initiate battles between defenders and attackers
- Can have factories which provide:
  - Goods which increase the morale of the platoon
  - Medics which increase the health of the platoon
  - Ammo which replenishes the ammo of the platoon

Provide an internal Battle Engine:

- Which controls the flow of battles between alliances/platoons in an area
- Which works on a turn-based approach
- Which allows users to change their strategy
- Which enables users to attack
- Which enables users to request reinforcements from neighbouring areas
- Which allows users to retreat to adjacent areas

Provide platoons that:

- Have the ability to attack other platoons
- Takes damage from other platoons
- Keeps track of the units' morale, health and damage
- Consists of soldiers and vehicles
- Has an active weapon strategy (precision/explosive)

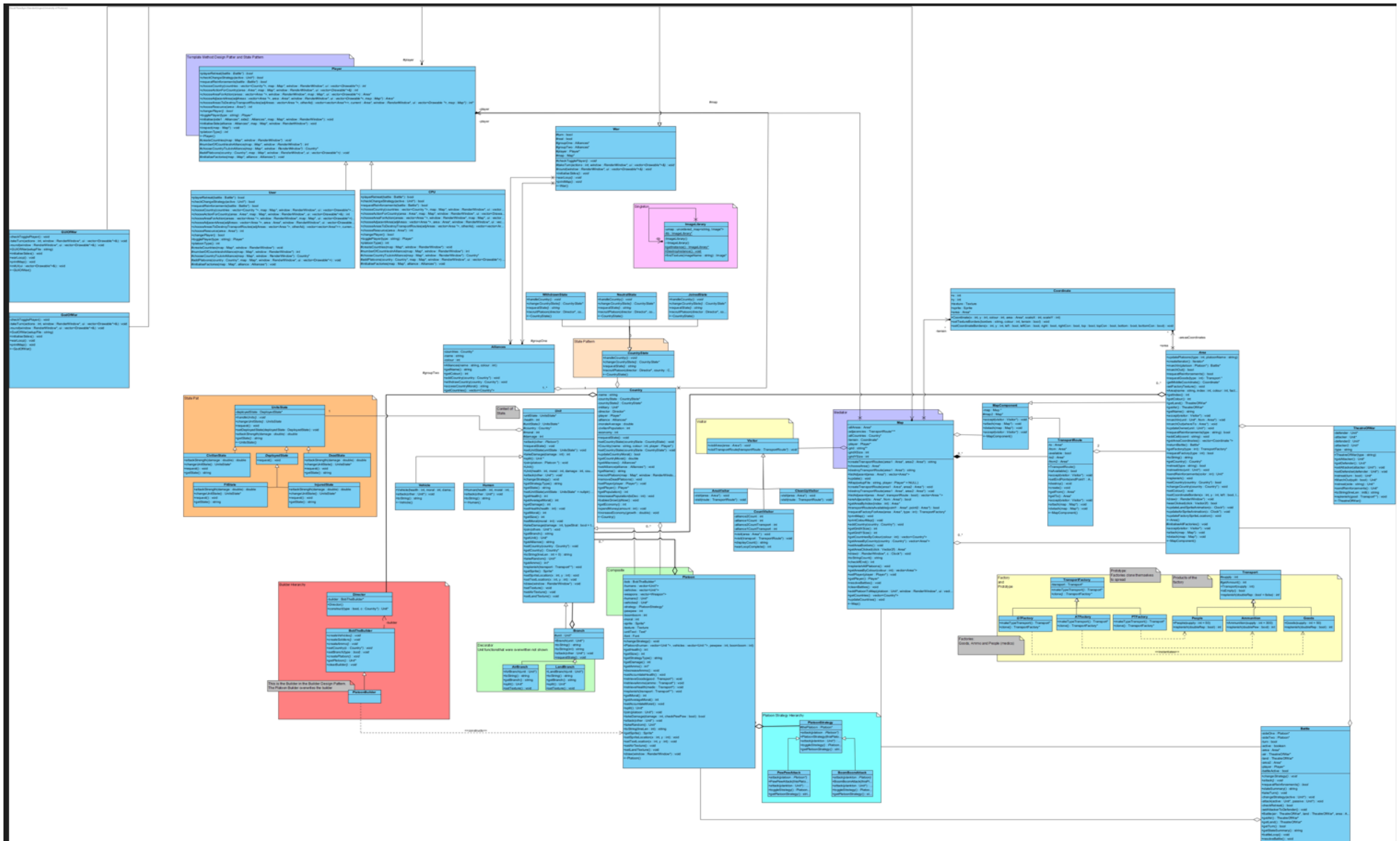
Provide a means to construct platoons:

- By adding a random amount of soldiers to the platoon (based on abstract criteria)
- By adding a random amount of vehicles to the platoon (based on abstract criteria)
- By initialising the platoon's ammo and strategy
- By adding the platoon to a country, and thus an alliance (based on Player Input)

## Design Patterns

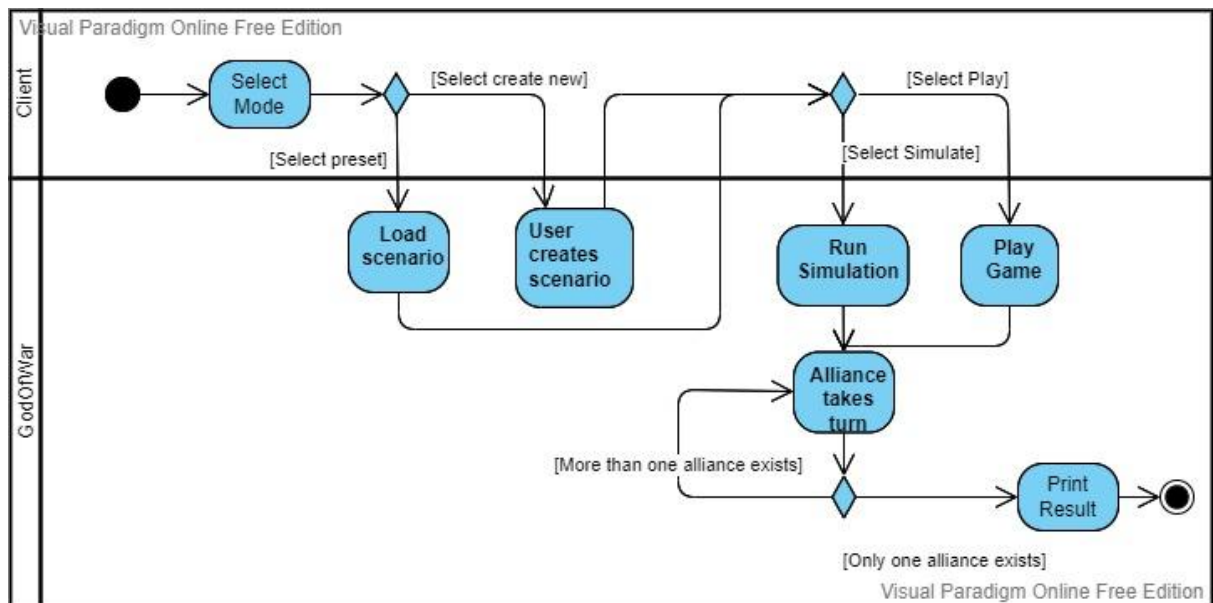
- State
  - Country (Withdrawn, Neutral, Joined)
  - Unit (Fit, Injured, Dead)
- Strategy
  - BoomBoom (explosive attacks)
  - PewPew (precision attacks)
- Composite
  - Unit hierarchy
    - Humans
    - Vehicles
    - Platoons
- Decorator
  - Branches (decorates Unit hierarchy)
  - AirBranch
  - LandBranch
- Prototype
  - Resources Factories clone to be distributed
- Mediator
  - Map manages area communication (Areas are colleagues)
- Builder
  - Create platoons
- Visitor
  - Perform operations over map to provide stable map interface
- Template method
  - Initialization of game (User vs CPU)
- Factory method
  - Factories Create resources to supply platoons

UML

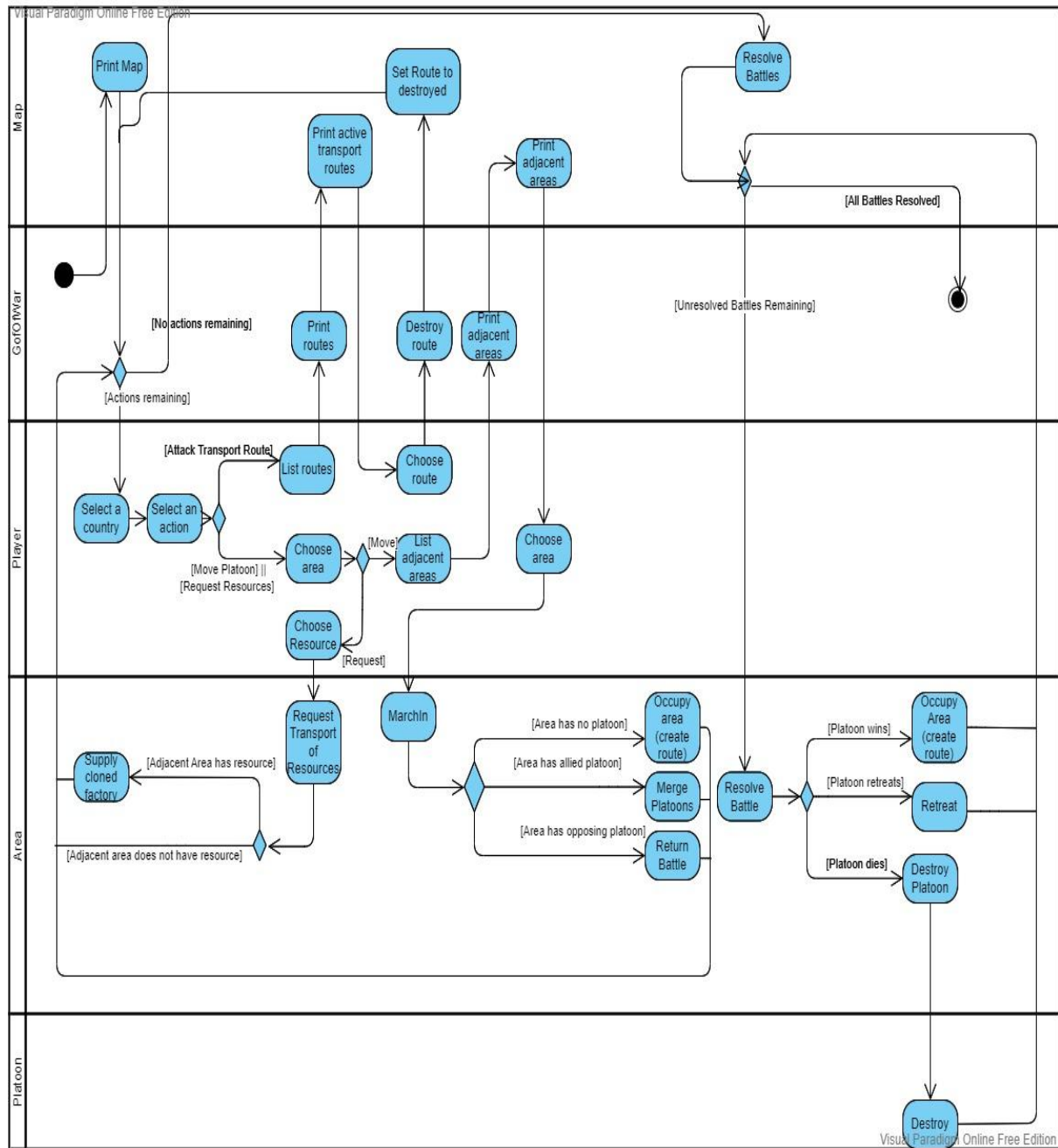


## Activity Diagrams

### Main War Loop



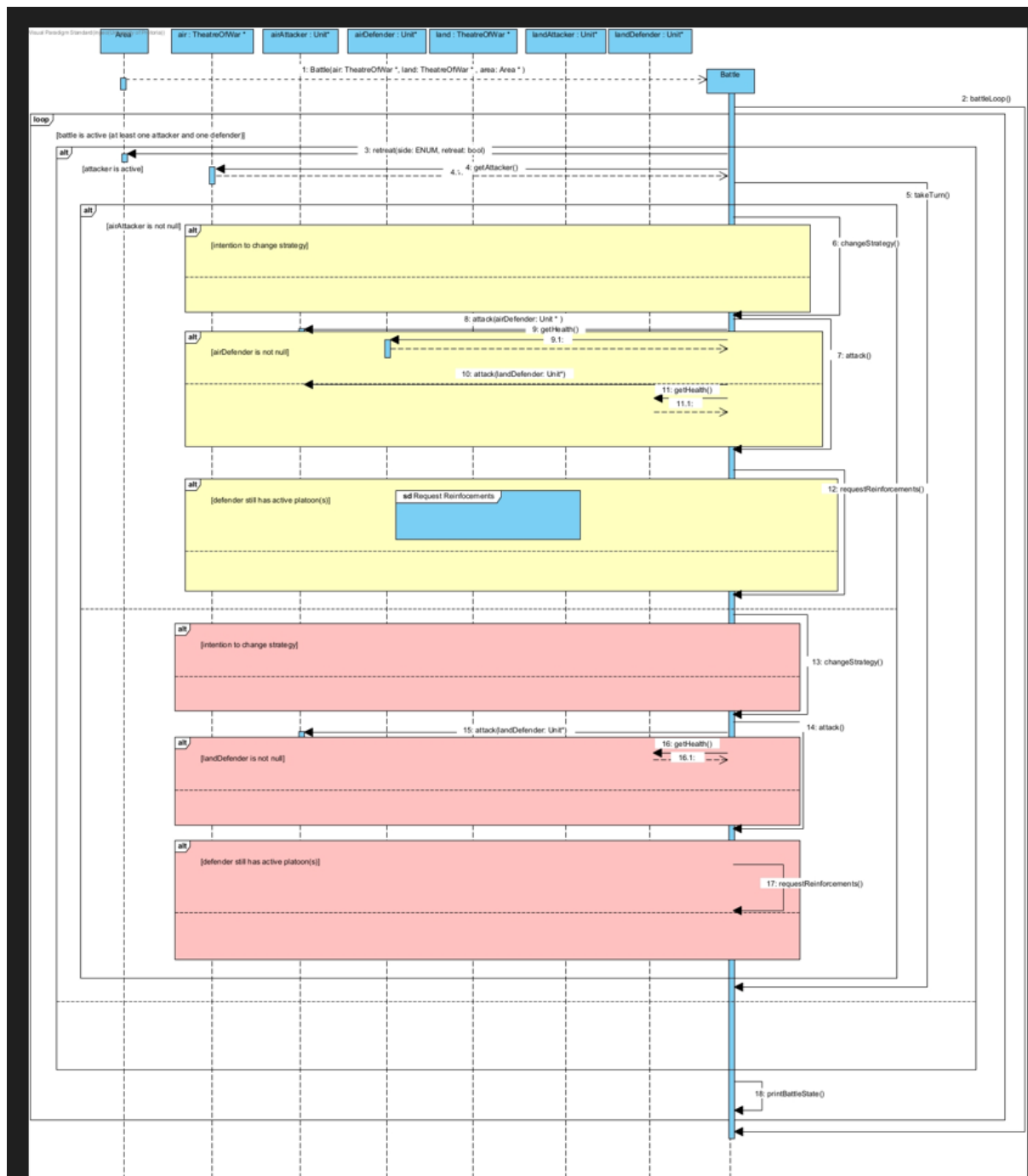
## Alliance turn





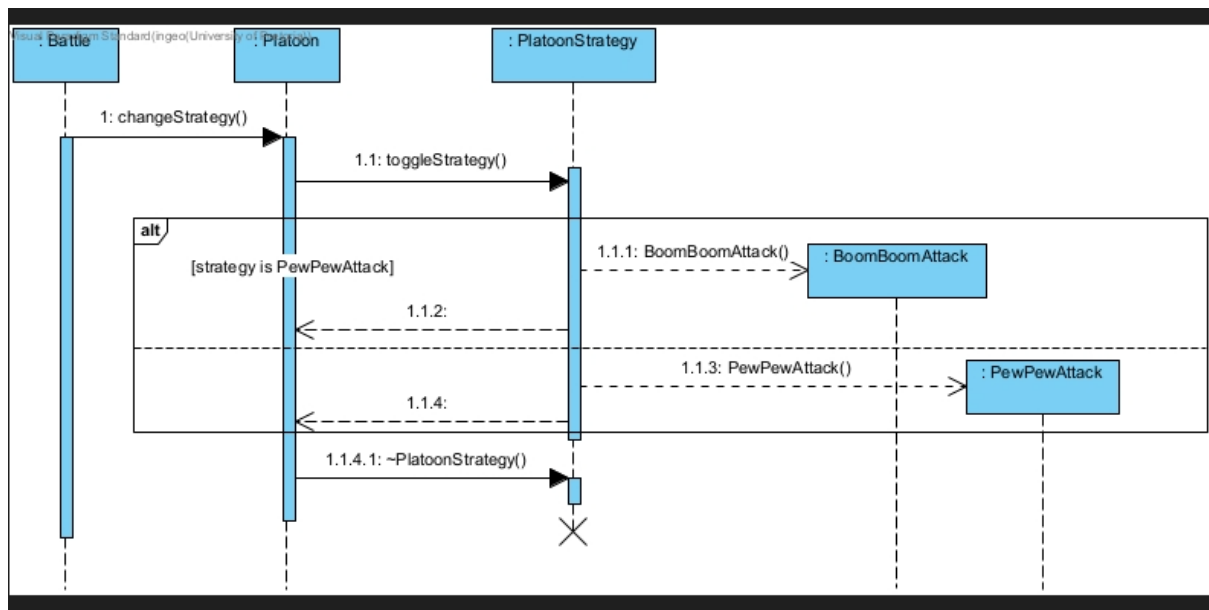
## Sequence Diagrams:

### Battle Sequence Diagram

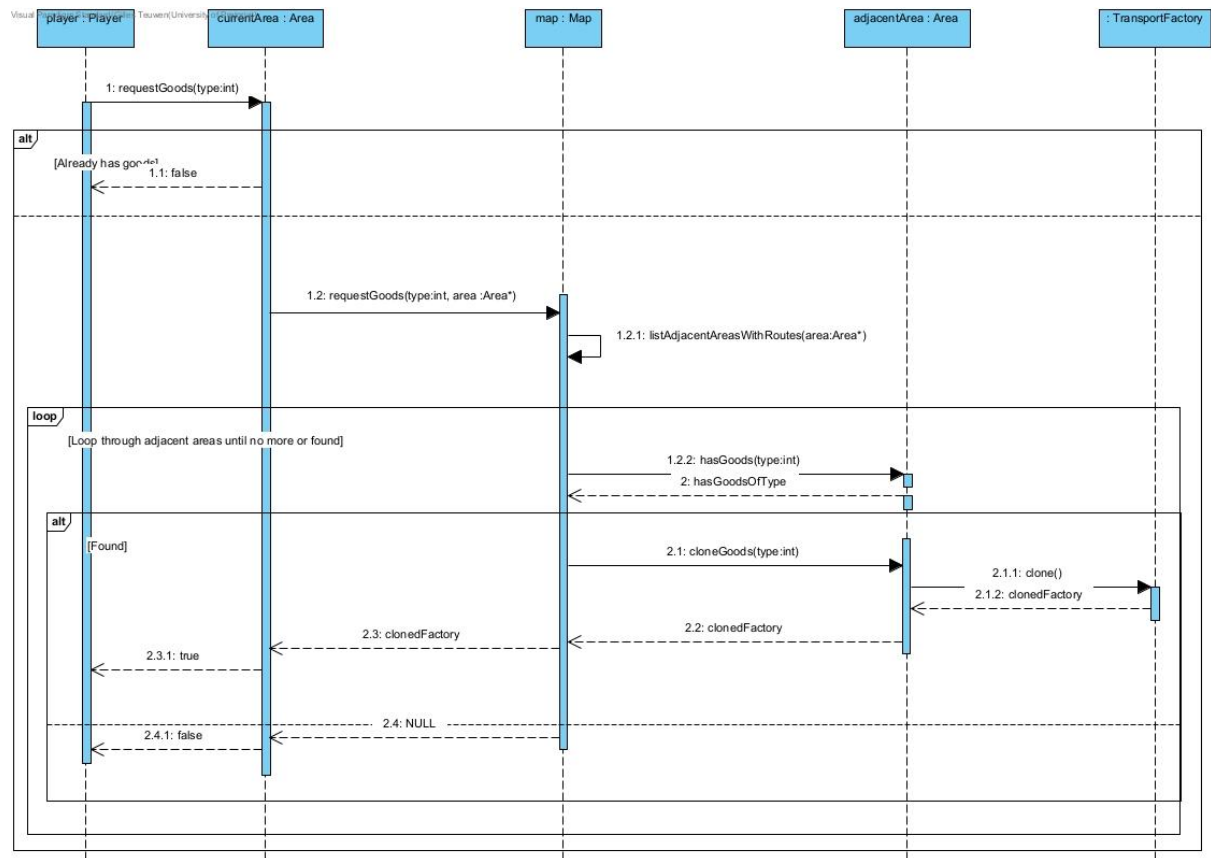


Note that the same holds true for the alternative (defender is active) with roles of attacker and defender just reversing

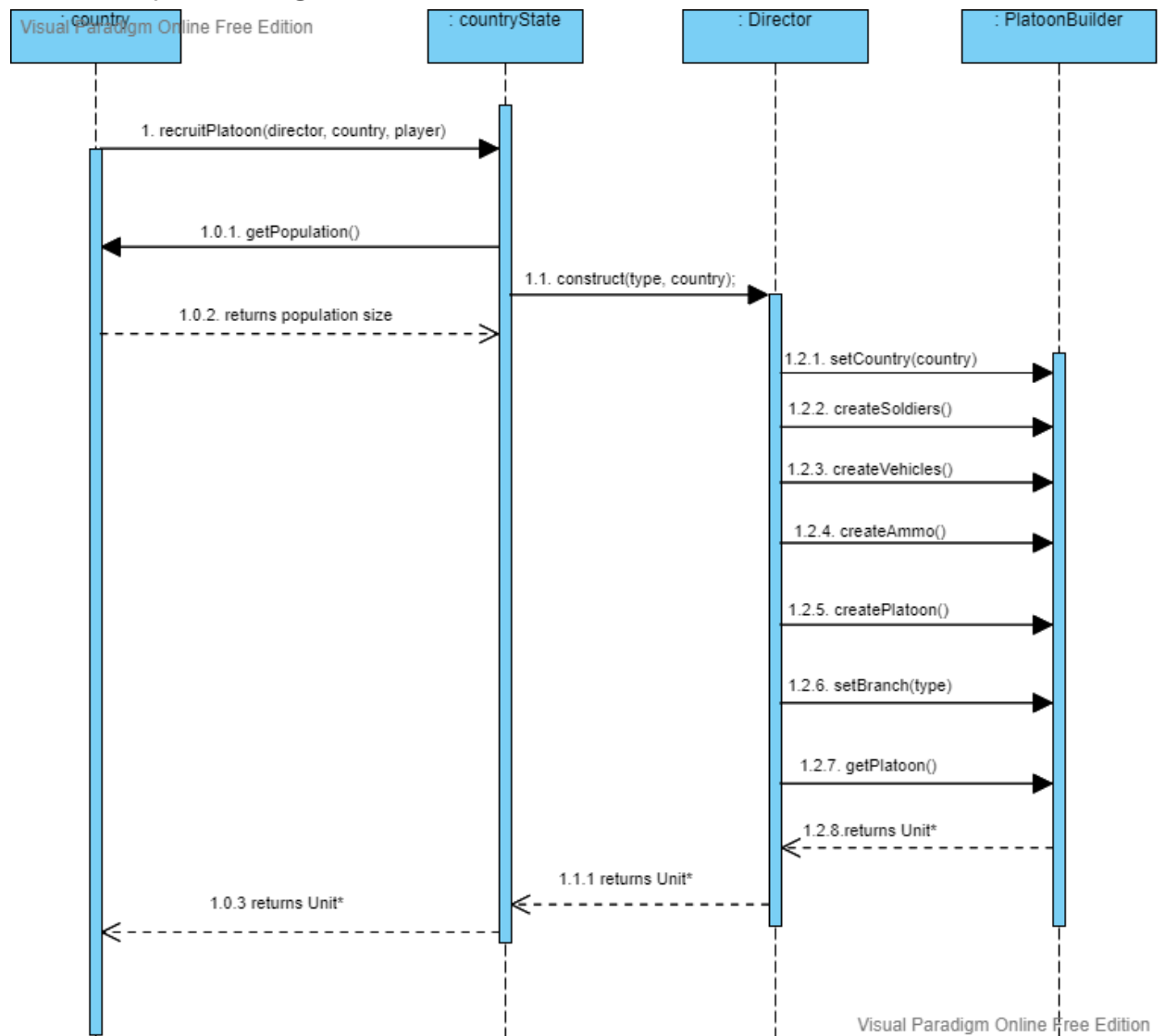
## Change Strategy



## Request Factory from adjacent areas

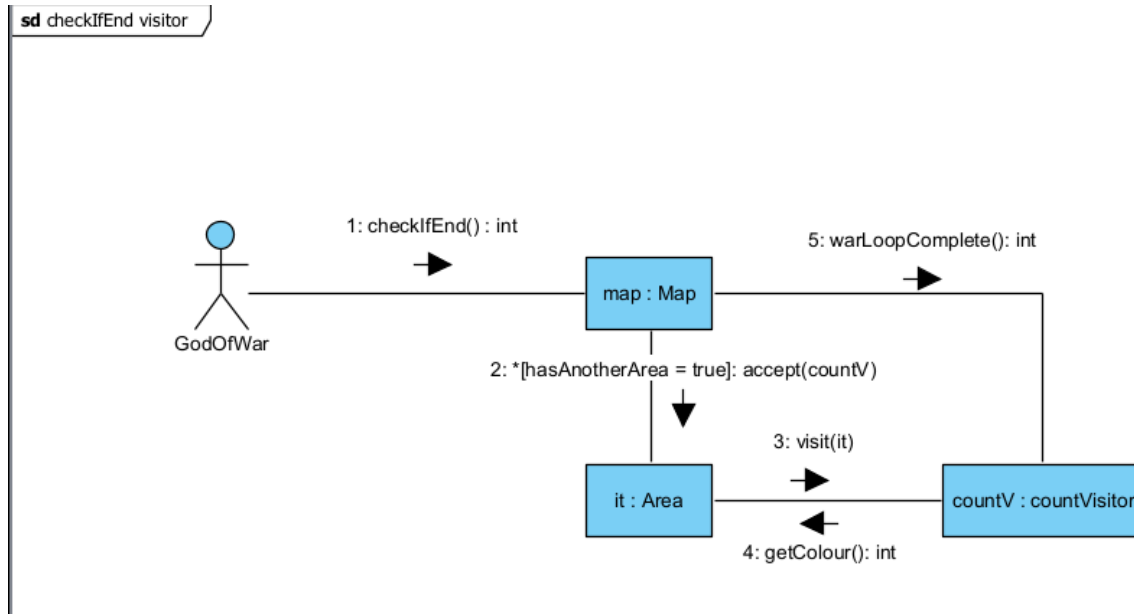


# Builder Sequence Diagram:

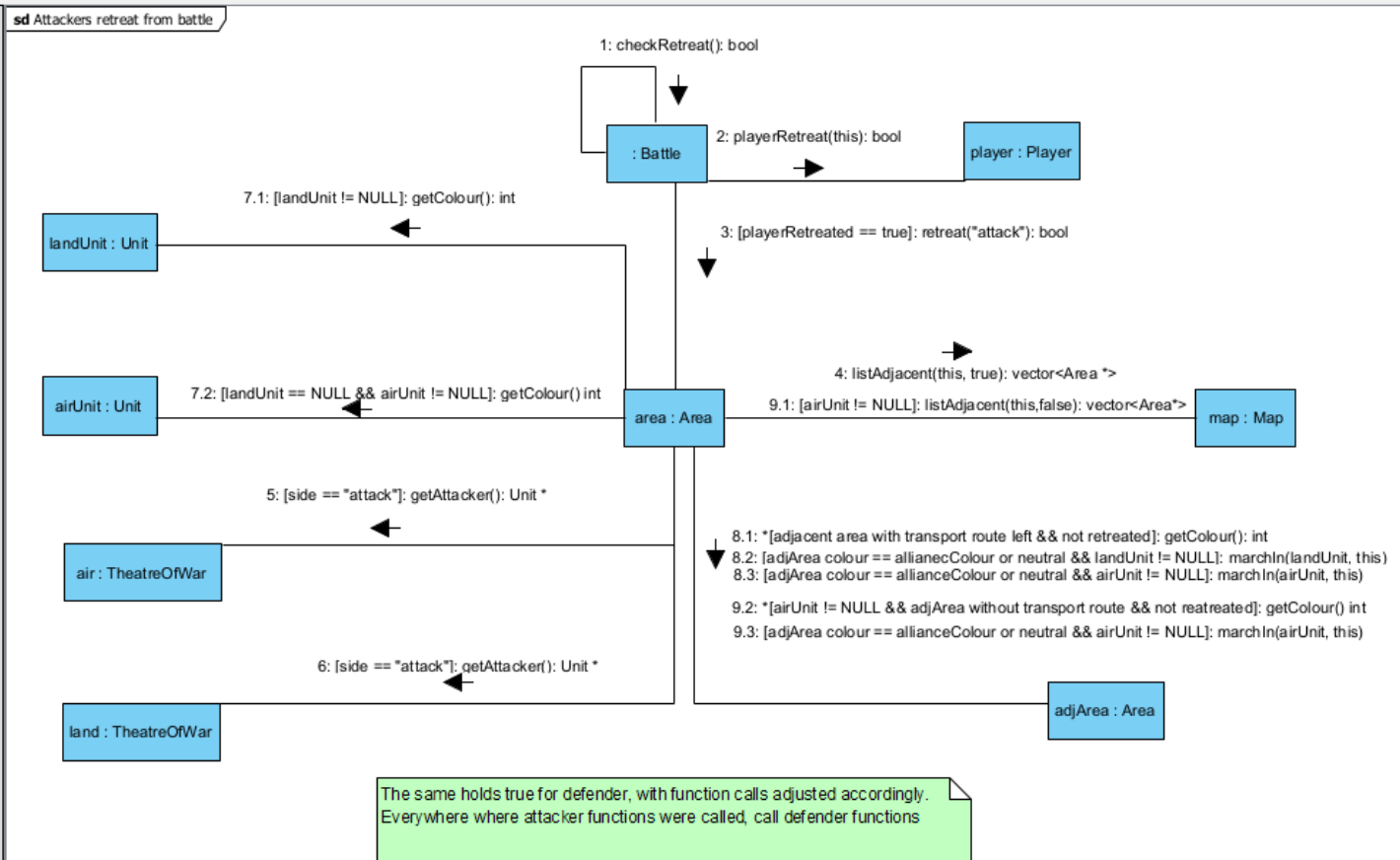


# Communication Diagrams

Basic Visitor Functionality (countVisitor used to check if war ended in this case)

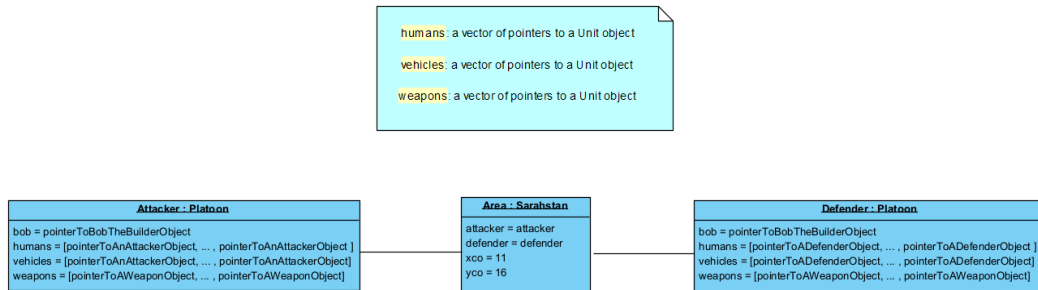


## Retreat from Battle (Attackers)

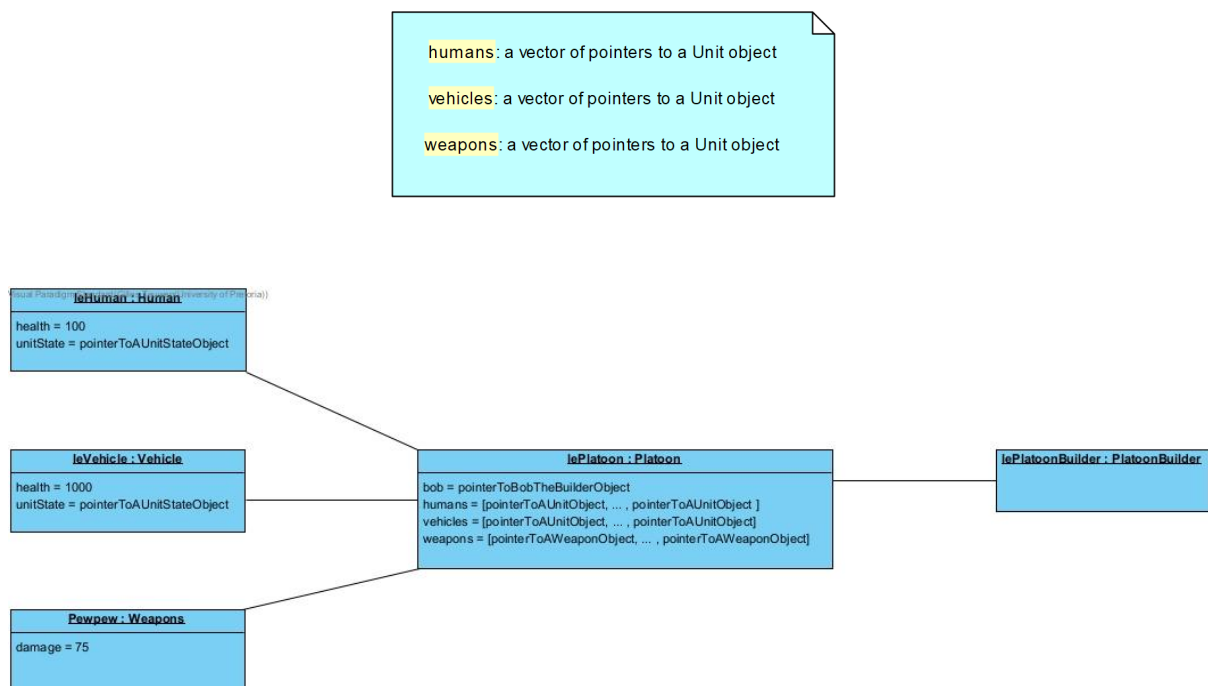


# Object Diagrams

## Area during battle object diagram



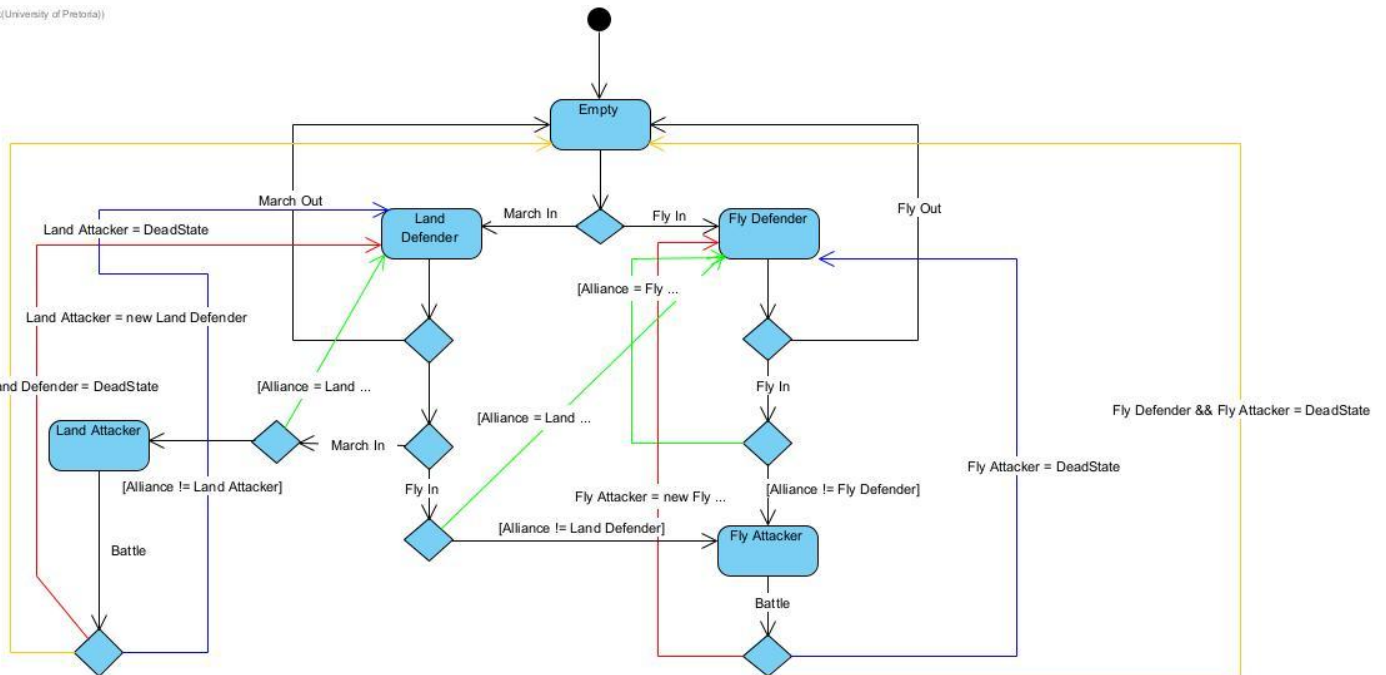
## Platoon object diagram



## State Diagrams

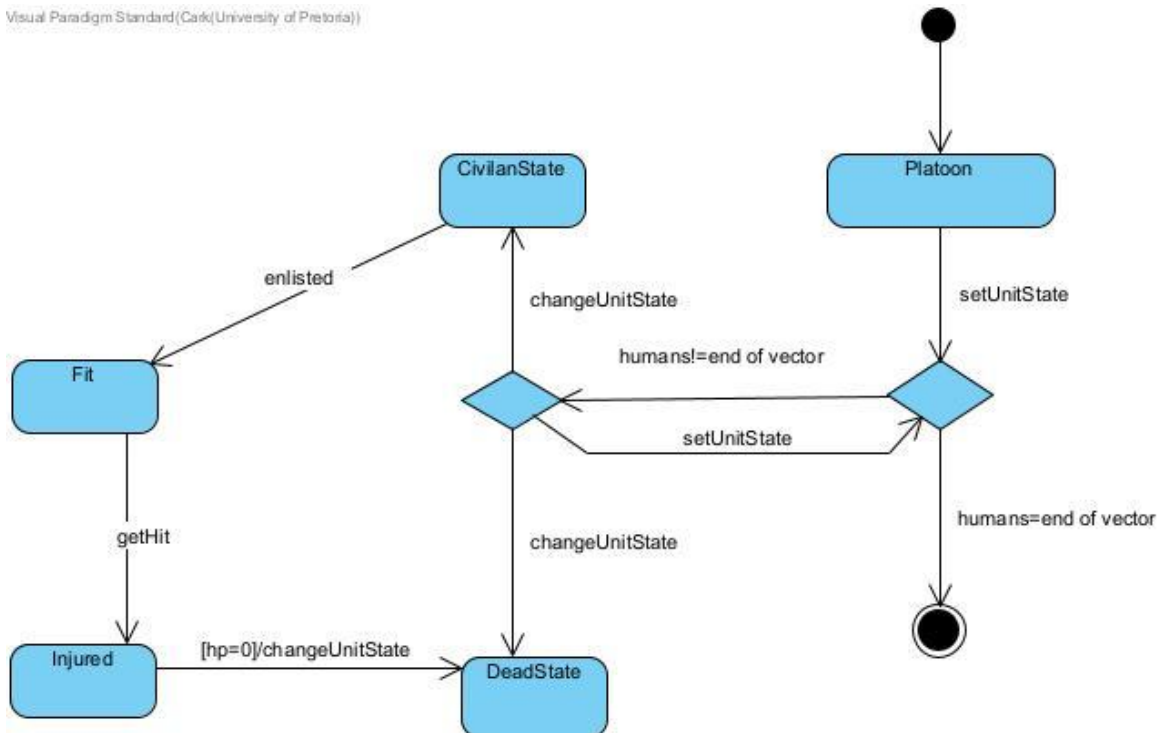
Area state diagram:

Visual Paradigm Standard (Cape University of Pretoria)



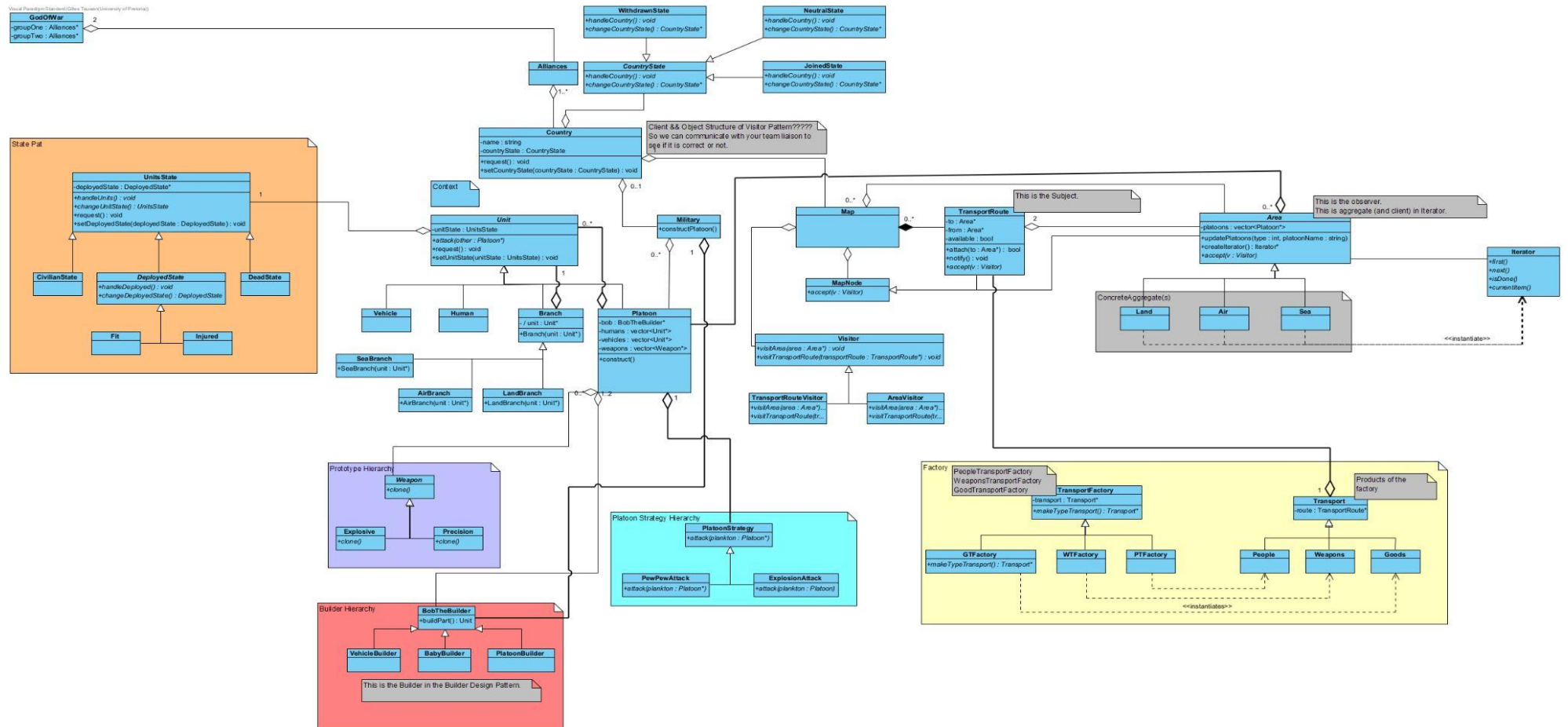
Platoon State diagram:

Visual Paradigm Standard (Cape University of Pretoria)



## Appendix:

### Pre-Initial UML





## Institute of Management Studies (University of Pretoria)

