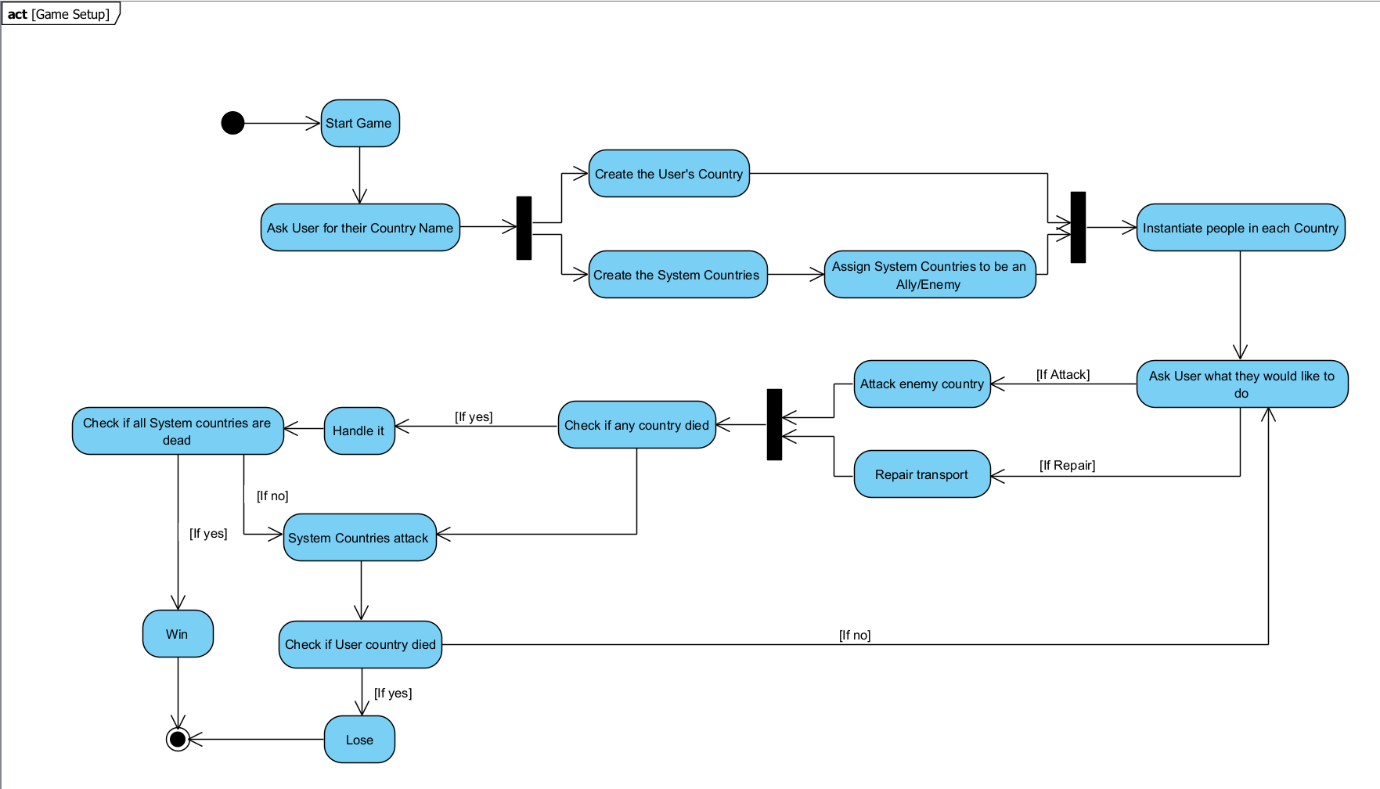
COS214 Project

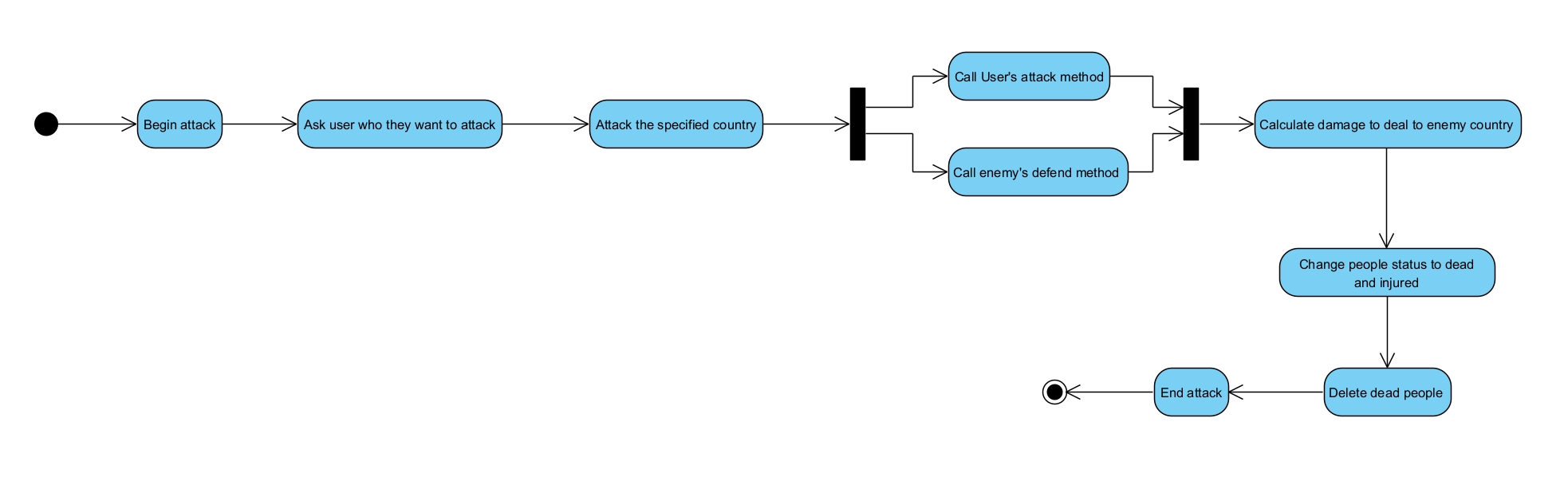
# Task 2:

## Functional Requirements

* Sets up a game for the user to play when the program is started
* Allows the user to choose who they would like to attack
* The system will change the algorithm used depending on the stage of the game
* If a specific country’s transport line gets damaged, that country will be severely hindered until they repair it
* Countries will hold different types of people and these types will do different damage numbers to enemy countries
* The system must be capable of undoing a previous move made by the user
* The system will create different people depending on the kind of area that the war is taking place in



Full Game loop



Attack/Defend process

Diagram

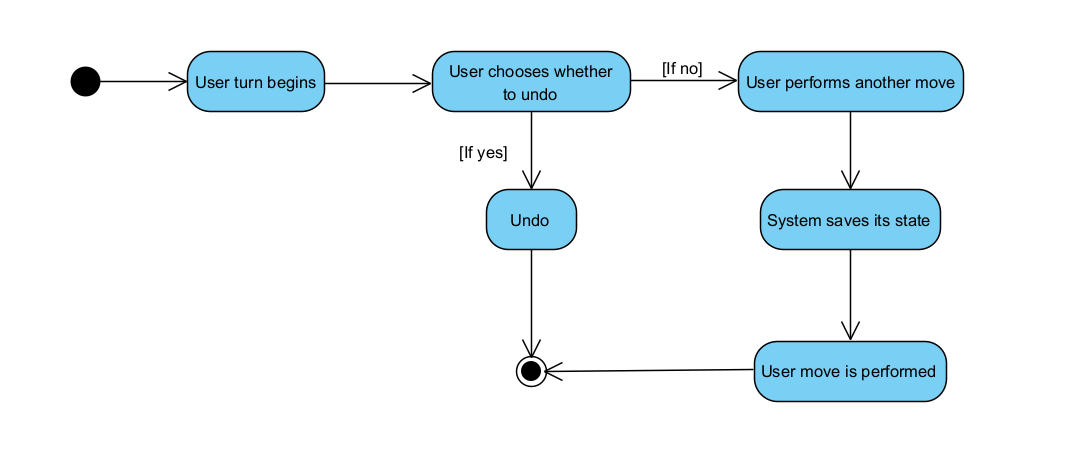
Description automatically generated

Stages of the game – Making use of the Strategy pattern here would allow us to easily swap out the game loop algorithm in the middle of the game itself.

Diagram

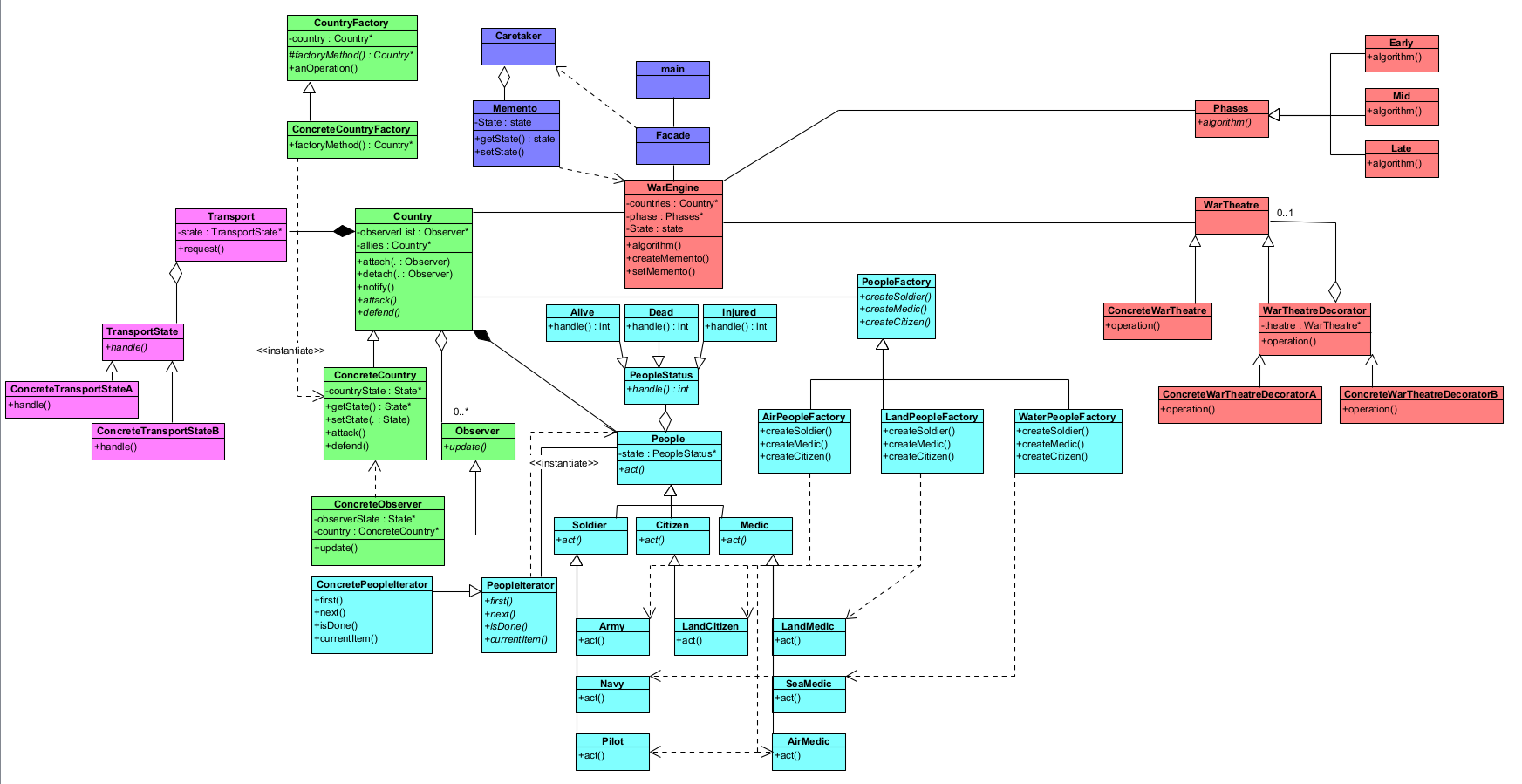
Description automatically generated

Transport Lines – Using the State pattern here allows us to change the state of the transport lines and affect the amount of damage a country will do depending on the state of it’s transport.



Undo – Using the Memento pattern allows us to save the state of the system and will give the user the option to undo a previous move that they have made.

1. Abstract Factory – Using an abstract factory will allow us to create all the different people that will exist within the war. For example, we will have Soldiers, Citizens, and Medics however there are different types of each of these. Abstract Factory takes care of this.
2. Iterator – The Iterator pattern will be used to iterate through all the people within a country.
3. State – The state pattern is used for multiple things in this program. It is used to control the state of the transport lines, as well as to track the state of each person in each country. Each state will act like a damage modifier for the country. The PeopleStatus will control the amount of damage that each person will do, and the TransportState will control the amount of damage that a Country can do.
4. Factory Method – The Factory Method pattern will be used to instantiate all the countries.
5. Template Method – The Template Method pattern will be used for the “act” and “attack/defend” methods in the people and country classes respectively.
6. Strategy – The Strategy pattern will be used to control the algorithm for the main game loop and will be changed at different stages of the game. This allows us to use a single gameloop method and just switch out the strategy when we need to change the stage of the game.
7. Decorator – Decorator will be used to control the WarTheatre, this will control the types of people that will be created by the Abstract People Factory. Eg. Land, Water, Air. Depending on the types of combat that can take place in the current location.
8. Memento – This will give the user the ability to undo a previous move in the case that they made an error.
9. Observer – The Observer pattern will be used to monitor a country’s transport state.
10. Façade – The Façade pattern will be used to create a clean interface for the user to interact with.



UML Class Diagram