

SOEN 6441 - Advanced Programming Practices

Project - Wargame (Build #1)

Winter 2025

Submitted by **DABSV**

Arvind Nachiappan Lakshmanan - 40310757

Barath Sundararaj - 40324920

Devasenan Murugan - 40302170

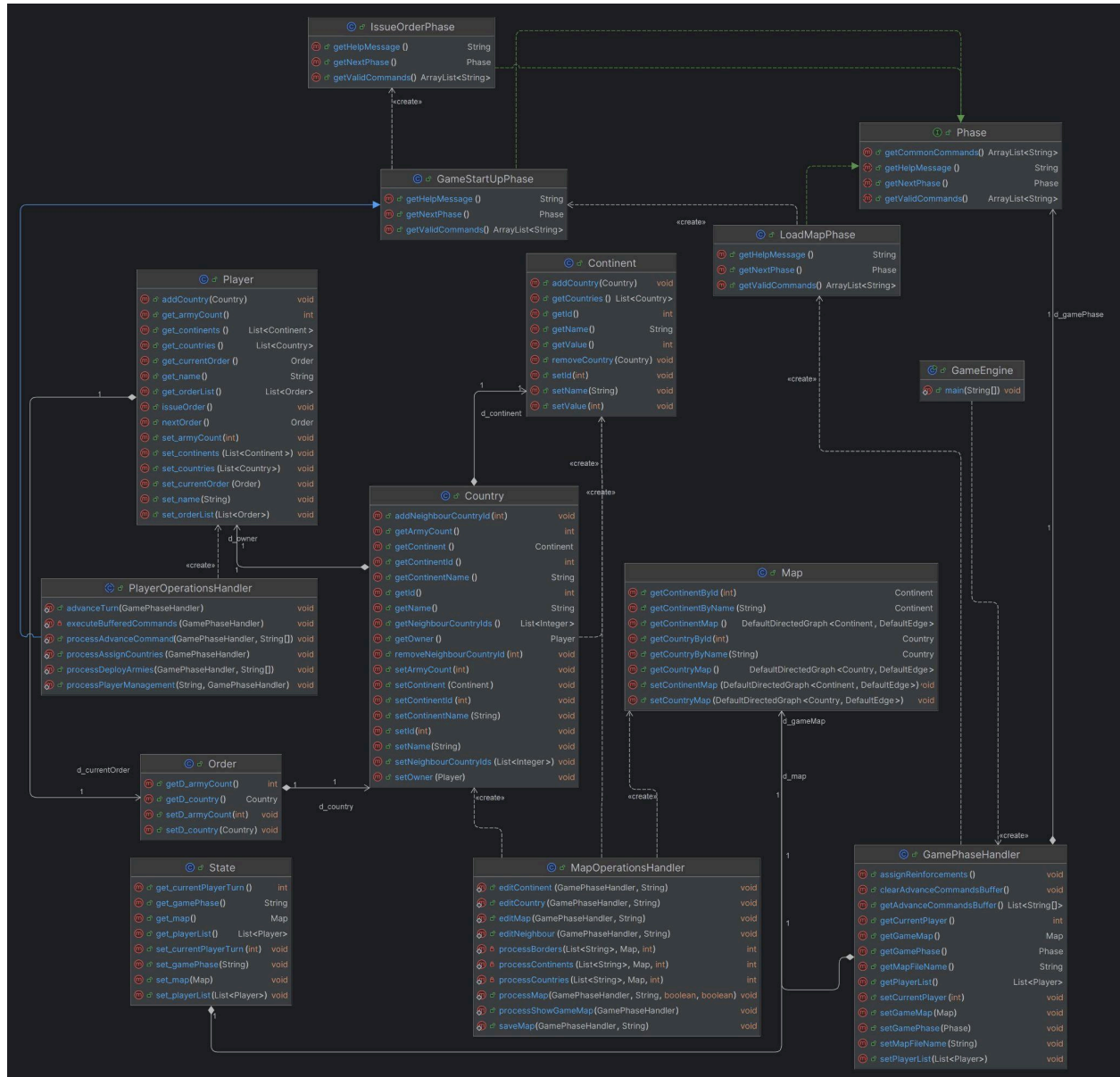
Swathi Priya Pasumarthi- 40322468

Submitted to

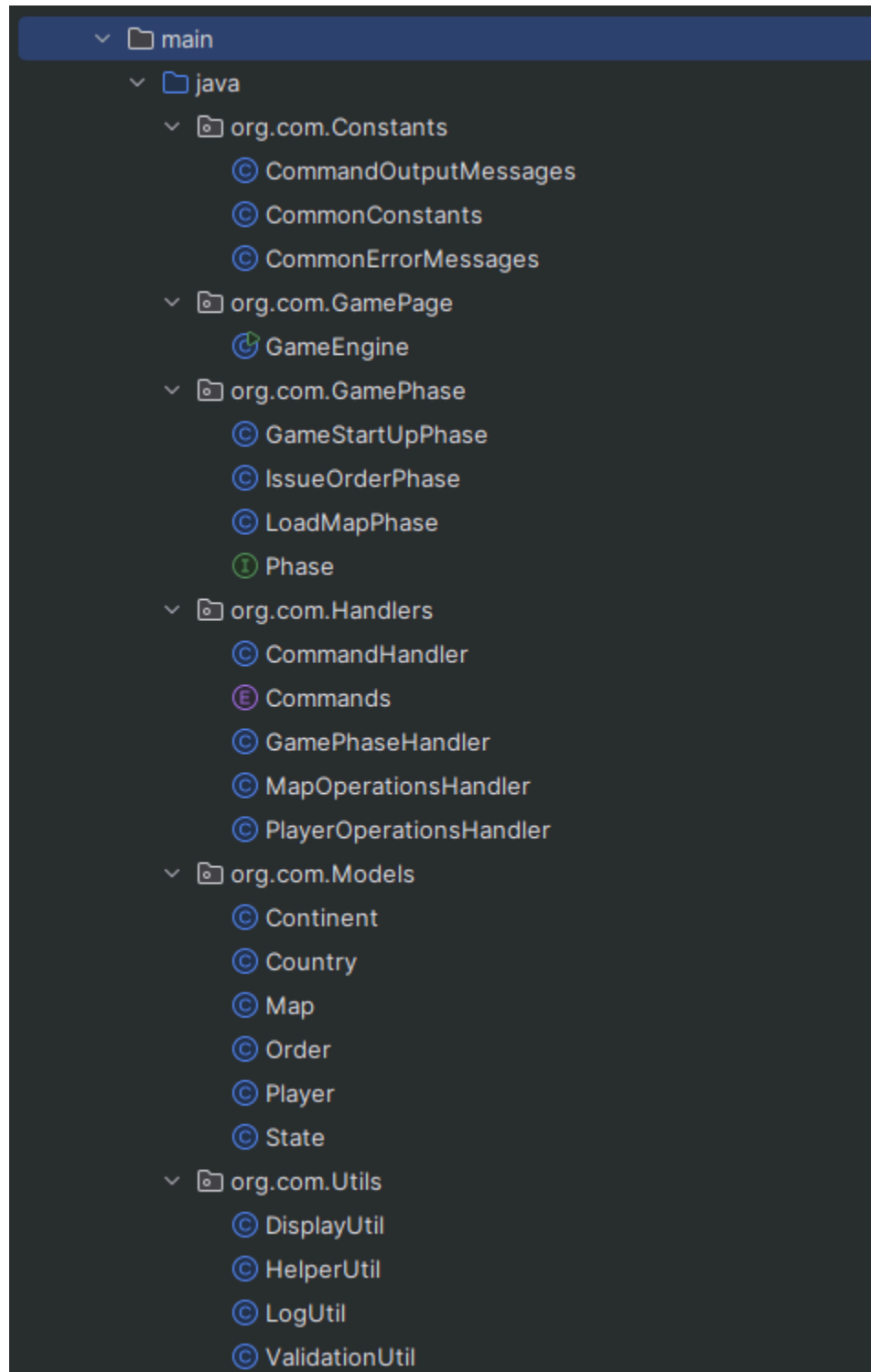
Prof. M. Taleb

Build #1 - Architectural Design

UML Diagram



Package Structure



Design Description

The application is structured into several modules to promote reusability, maintainability, and scalability. These modules are `Constants`, `GamePage`, `GamePhase`, `Handlers`, `Models`, and `Utils`.

1. Constants

This module provides a centralized location for constants and messages used throughout the application, promoting code consistency and ease of modification.

Java File	Description
CommandOutputMessages	This class contains output messages of each command.
CommonConstants	This class contains all global constants.
CommonErrorMessages	This class contains error messages for different failure cases.

2. GamePage

This module serves as the entry point for the game.

Java File	Description
GameEngine	The main starting point where the game starts with accepting the game commands.

3. GamePhase

Java File	Description
GameStartUpPhase	Game startup phase related functions and method calls are

	in this file.
IssueOrderPhase	Issue Order Phase implements the <i>Phase</i> interface and contains code related to the command given by the Player.
LoadMapPhase	Initial phase of the game where the player initialises a map and the code related to this are present here.
Phase	This interface is the basic structure of the entire phase management.

4. Handlers

This module contains classes responsible for handling different types of operations and commands.

Java File	Description
CommandHandler	This class handles all commands and invocation of all command methods.
Commands	This enum contains all commands and their valid attributes.
GamePhaseHandler	Handles the game phase and changes after each player's turn.
MapOperationsHandler	All important methods which are used for processing a map related command are written here.
PlayerOperationsHandler	All operations related to a Player like Deploycommand is written here

5. Models

This module defines the core game entities and its properties.

Java File	Description
Continent	This Class represents the Continents in the Map.
Country	This class represents the Country in the Map.
Map	The Map class represents the game map, which includes continents and countries. Also the border information
Order	This file is used to represent the "Order" given by a player (Each player will have an order) in the game.
Player	This java file is used to define a player in the game Contains all the attributes of a player.
State	The 'State' class is used to represent the current state of the game.

6. Utils

This module provides utility classes for common operations and tasks.

Java File	Description
DisplayUtil	The DisplayUtil class is mainly used for describing the outline of the map.
LogUtil	This class contains the base

	method for logging all game activities.
ValidationUtil	ValidationUtil Class is mainly used for validation in various phases of the Game, such as map validation, command validation etc.

This structure ensures a separation of concerns, making the codebase more organized and easier to maintain. Each module is responsible for a specific aspect of the game, and the classes within each module encapsulate the data and behavior related to that aspect.

Game Flow

"help" command will display all these features and its commands

Map loading Phase: This phase handles all the map related features

1. Loading a map
2. Editing an already existing map or Creating a map from scratch
 - a. Editing the Continents
 - b. Editing the Countries
 - c. Editing the Neighbouring countries
3. Validating the correctness of the map
 - a. Both after loading or editing a map, the validation takes place automatically
4. Show Map (at any phase of the game)

Gameplay Phase:

1. After loading a map, the actual game play starts
2. Enter the name of the players
3. When the player issues the "assigncountries" command, all of the players in the game are assigned with an equal number of countries present in the loaded map.
4. Along with assigning countries they are also given a fixed number of army to be deployed in their own country.
5. Only after a particular has deployed all of his/her armies will the next player's turn begin.

Tools

- **IntelliJ** - Code Editor
- **JUnit5** - is used for unit testing
- **Github** - is used for managing and collaborating on the project.
- **Javadoc** - implemented for all classes and methods.