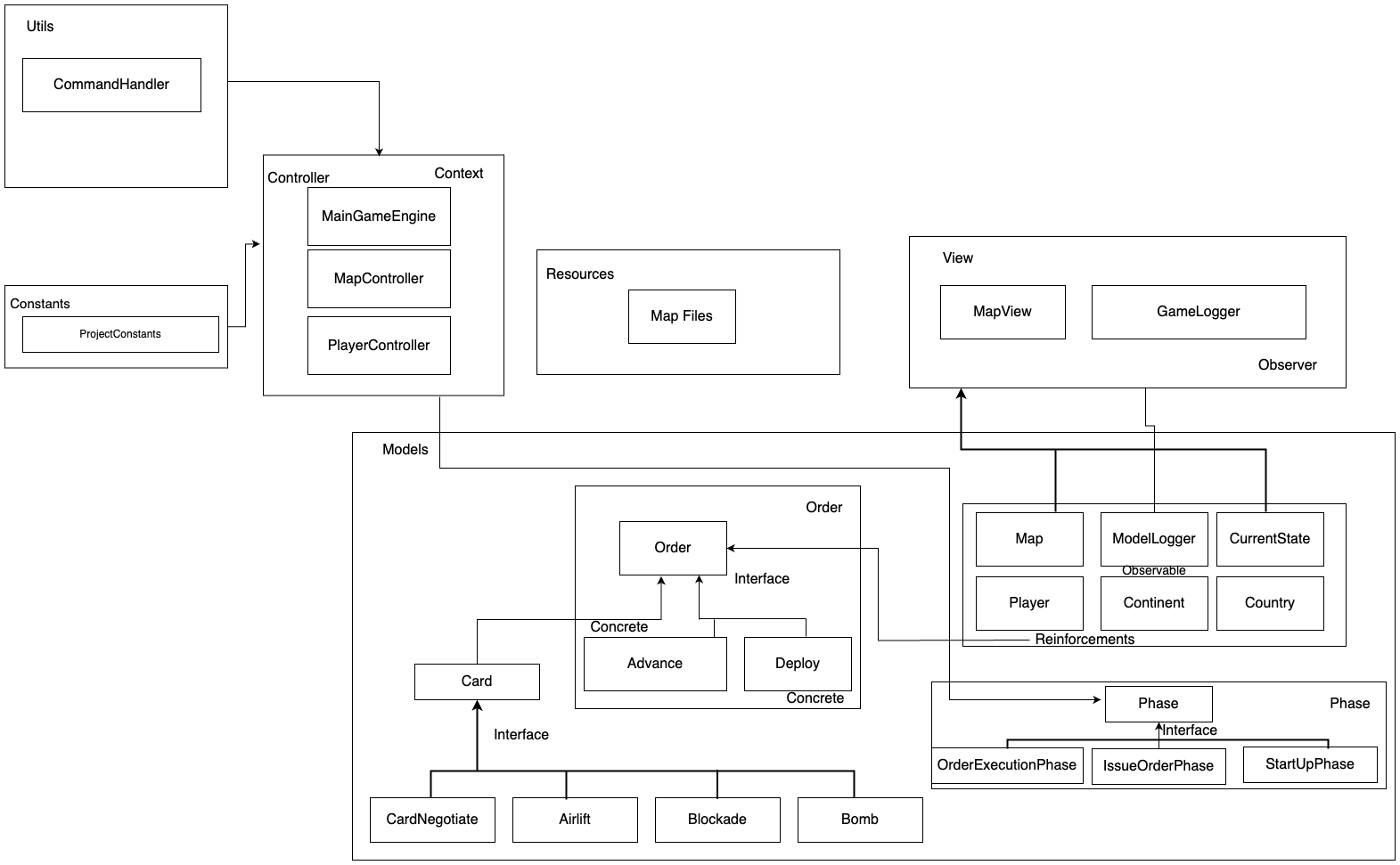
Group 15: Architecture Diagram

COURSE: SOEN 6441

BUILD #2



**Models**

**Advance: -** Stores the advance order details and contains the functions required to perform advance order.

**Card: -** Card interface is implemented by 4 special cards and it inherits the Order class.

**CardAirlift: -** It stores the airlift order details and contains the functions required to perform airlift operation.

**CardBlockade: -** It stores the Blockade order details and contains methods required to perform blockade operations.

**CardBomb:** - It stores the Bomb order details and contains methods required to perform bomb operation.

**CardNegotiate: -** It stores the Negotiate order details and contains methods required to perform Negotiate operation.

**Continent: -** This class contains the data and functions related to the Continents such as list of countries, continent value, etc

**Country: -** It includes operations to manage countries, armies and its neighbours.

**CurrentState: -** CurrentState is used throughout the game to get current state of map, player information and operations to manipulate the players in a game.

Deploy: - Stores the deploy Order details and contains the functions required to perform deploy Order.

**IssueOrderPhase: -** It is second phase of the game containing methods to take Orders from the user in round robin way.

**Map: -** It manages map related data and operations such as validate map, etc.

**ModelLogger: -** This class acts as a Subject for logger Observer pattern implementation.

**OrderExecutionPhase: -** This is the last phase in our game contains execute method that executes all the orders taken in the issueOrder Phase.

**Orders: -** It manages the player orders and is inherited by deploy, advance and card models.

**Phase: -** Phase is inherited by 3 other phases and it contains common functions needed in all the phases such as handling commands, etc.

**Player: -** This class manages the data and functions related to the player such as addPlayer, issueOrder, nextOrder, etc.

**StartupPhase: -** The game starts from this phase and it contains operations for commands allowed in this phase.

**Controller**

**• MainGameEngine: -**Game starts from this phase and it calls the init method of Startup Phase.

**• MapController: -**It handles all the map related operations of the game.

• **PlayerController:** - It manages all the player related operations of the game.

Views

**• MapView: -** This view is used to display the map during the execution of the game.

• **GameLogger: -** This class stores all the game operations in a logger text file and acts as a observer to the ModelLogger class.

Utils

• CommandHandler: -This class handles the input command given by the user by separating different components of the command that are further used to perform operations accordingly