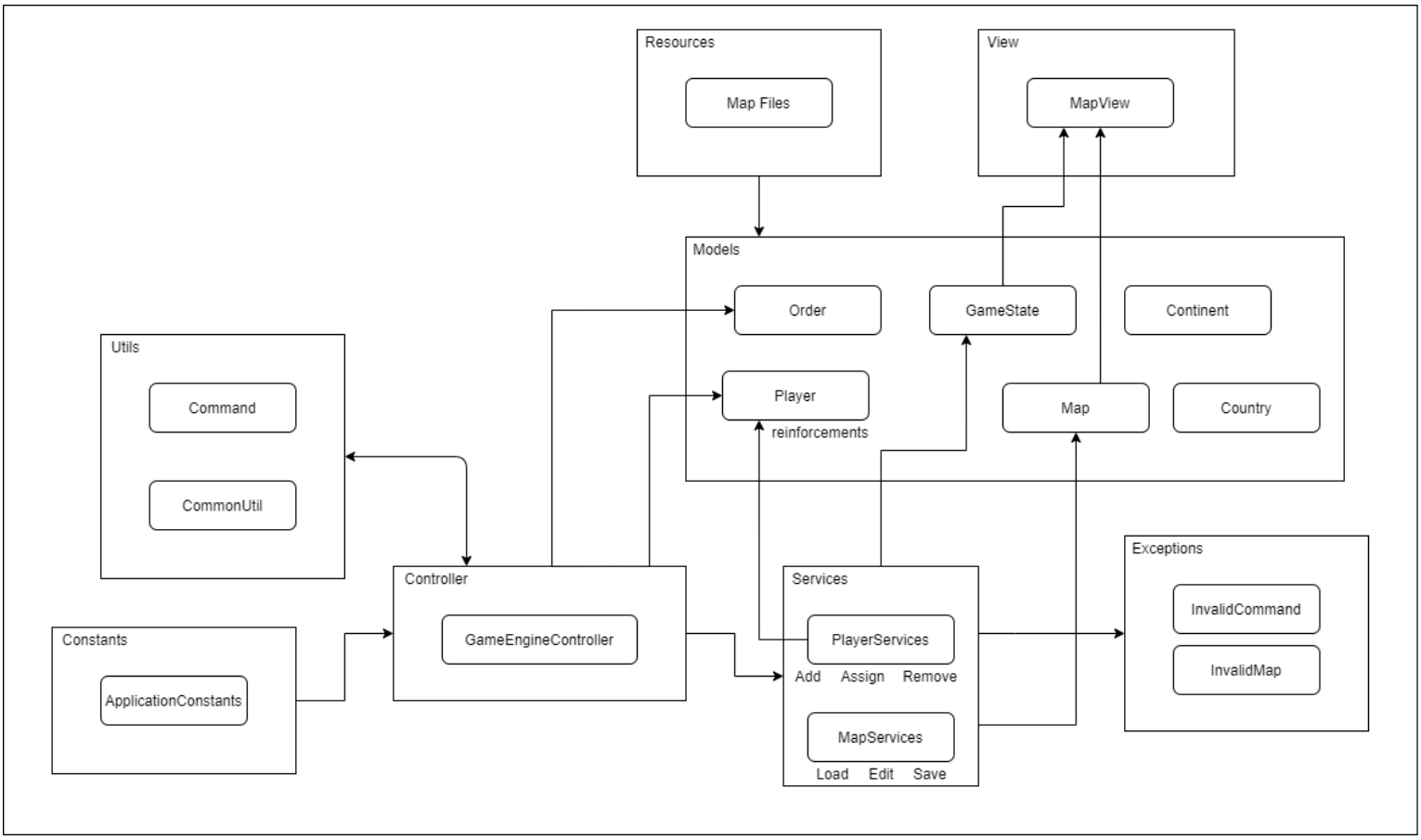
Build #1 – Architecture Diagram



A. Models

- Player: This class manages all the continents, countries, the orders given, and armies owned by each Player object.
- Order: This class handles the order action, the source and destination country and the number of armies to move.
- GameState: Manages the Player and Order objects and maintains the list of orders which are in the list (yet to be executed).
- Map: Manages the Country and Continent objects and Validates the Map by checking the connectivity between the existing countries and continents.
- Continent: Manages all the Continents along with the bonus value of each continent. Also, Country object is linked with the Continent object to map all the countries that exist in the Continent.
- Country: Manages all the Countries along with the number of armies present on each of them and the Countries adjacent to it.

B. View

- MapView: Displays the current state of the Map when user selects *showmap* command in the GamePlay State and displays the overall existing Map while in EditMap phase.

C. Controller

- GameEngineController: The main command handling logic is controlled from this class.

D. Services

- MapServices: The main logic to handle the state of the Map is handled here, along with map load, edit and save functionalities.
- PlayerServices: It enables the services related to Player – add, remove, assign continent, countries and armies.

E. Utils

- Command: Takes and formats the commands given by the players.
- CommonUtil: Handles the input files.

F. Constants

- ApplicationConstants: All the static members are defined here and can be used anywhere in the application at any point.

G. Exceptions

- InvalidCommand: Displays meaningful message to the player when invalid command is entered.
- InvalidMap: Displays meaningful message to the player when invalid map is created.

H. Resources

- Map Files: All the .map files, existing and newly created by users are stored and managed here.