Architecture Description  
Team: Overlords

The architecture of our build 1 of Warzone game is divided into several cohesive modules, each with a clear responsibility:

1. **map Module**
   * **MapReader:** Loads map files into memory by reading text files and parses the map’s structure, including continents, countries, and connections.
   * **MapWriter:** Saves maps from memory back into text files, ensuring correct formatting and structure.
   * **MapEditor:** Manages map editing commands based on user input, validating the consistency and connectivity of the map.
2. **iohandler Module**
   * **InputHandler:** Handles all input and output operations during map editing and gameplay, including processing map editing commands, game setup, and validating user inputs.
3. **model Module**
   * **Country and Continent:** Define the map’s structure with ownership relationships between countries and continents, supporting strategic gameplay elements.
4. **gameplay Module**
   * **GameEngine:** Controls the overall game flow, including phases like startup, reinforcement, attack, and fortification, as well as managing player turns and state transitions.
   * **Player:** Manages player-specific information such as owned countries, available armies, and issued orders.
   * **Order:** Represents commands issued by players, such as deploying or moving armies, ensuring they are executed sequentially and follow game rules.
   * **Mainmenu:** Manages the main menu with options to edit maps, play the game, or exit, serving as the entry point to different game functionalities.
5. **Command Handling**
   * **Command:** Responsible for parsing and organizing user inputs into labeled arguments for execution, validating command syntax, and forwarding them to appropriate handlers.

This modular design ensures clear separation of concerns, cohesion within modules, and scalability, making the system maintainable and extendable.