

## Refactoring Document

### Optimizing Design Patterns:

1. *Command Pattern*: Implementing structured command encapsulation for user actions.
2. *Observer Pattern*: Introducing an observer pattern for improved communication between game components.
3. *State Pattern*: Enhancing modularity by implementing the State Pattern for distinct game phases.

**Code Efficiency Enhancements:** 4) *Removal of Redundant Commands*: Eliminating unnecessary command line calls to enhance code efficiency.

5. *Reducing Object Overhead*: Refactoring temp order creation to minimize extra object instantiation.
6. *Improved Termination Conditions*: Updating termination conditions for the game with clarity and conciseness.

**Enhancing Readability:** 7) *Smarter Conditional Checks*: Breaking down concatenated checks for better code readability.

8. *Method Refactoring*: Breaking down large methods into smaller functions for improved readability.
9. *Centralized Validation*: Discarding validation in individual classes, considering centralization.

**Cleaner User Interaction:** 10) *Streamlined Map Editing*: Removing redundant code related to `editmap` and `loadmap` from the EDITMAP phase.

11. *Reduced Redundancy in ShowMap*: Refactoring `showmap` functionality to eliminate redundancy.
12. *Special Order Conditions Clarification*: Clarifying and refactoring conditions for issuing special orders.

**Complexity Reduction:** 13) *Addressing Nested Code*: Considering and addressing deeply nested code for improved clarity.

14. *Minor Refactoring for Object Creation*: Minimizing the creation of unnecessary temporary objects.
15. *Naming Consistency*: Renaming methods for a more consistent and understandable codebase.

## CHOSEN REFACTORING TARGETS

1. *Command Pattern*: Implementing structured command encapsulation for user actions.
2. *Observer Pattern*: Introducing an observer pattern for improved communication between game components.
3. *State Pattern*: Enhancing modularity by implementing the State Pattern for distinct game phases.
4. *Improved Termination Conditions*: Updating termination conditions for the game with clarity and conciseness.
5. *Reducing Object Overhead*: Refactoring temp order creation to minimize extra object instantiation.
6. *Centralized Validation*: Discarding validation in individual classes, considering centralization.