Design / Usage Document

Project Overview: This project is an adventure game where the player navigates through different terrains, collects items, and interacts with the environment. The game is built using the SplashKit SDK in C#.

Design: The program uses several object-oriented programming principles and design patterns to achieve a modular and maintainable structure. The main components include:

- **GameMap Class:** Manages the game map, loading it from a text file and setting up various game objects like land, bombs, gold, swords, and potions.
- **GameProcessor Class:** Handles the game loop, initializes the game map and player, and manages game states (start, play, game over).
- **Player Class:** Represents the player character, managing its state, movement, and interactions with game objects. It implements strategies for movement, jumping, and collecting items, and uses the observer pattern to notify changes.
- **Observer Pattern:** Observers like SwordUnlocked, FastMovementUnlocked, HighJumpUnlocked, and ScoreObserver handle game events and update the game state or provide feedback to the player.
- **Strategy Pattern:** Different strategies for movement (NormalMovement, FastMovement) and jumping (NormalJump, HighJump) are implemented. The collection strategy is updated based on the player's progress.
- **Singleton Pattern:** ClockSingleton manages timers for various events.
- **Command Pattern:** Commands like LookCommand are used for player interactions with inventory.

Usage Instructions:

1. **Starting the Game:**
   - Run the Main method in Program.cs.
   - The game window will open, and you can start the game by pressing the space key.
2. **Playing the Game:**
   - Use the arrow keys to move the player.
   - Collect items like gold, swords, and potions to unlock abilities and increase your score.
   - The game map will update based on your movements and interactions.

3. **Game Over:**
    - The game will display a "Game Over" screen if the player hits a bomb.
    - Press 'R' to restart or 'Q' to quit the game.

Key Features:

- **GameMap Setup:** Reads the map layout from a text file and initializes the game objects based on the map data.
- **Player Movement:** Implements different movement strategies and handles collisions with the environment.
- **Item Collection:** Collect gold, swords, speed potions, and jump potions to enhance abilities and score.
- **Observers:** Notifies changes in the player's state, such as collecting items or unlocking abilities.
- **Game State Management:** Manages the game states and transitions between start, play, and game over screens.