Spotlight – User Manual

Author: Owen Chilson

Date: November 2025

1. Overview

The **Spotlight** is a C++ desktop application that simulates a digital board game environment. It combines command-driven gameplay with SFML rendering, player/company management, and persistent configuration storage.

The purpose of this program is to demonstrate structured C++ design and gameplay state management using object-oriented programming.

2. Requirements

Operating System

Windows 10 or later (64-bit)

Dependencies

All dependencies are handled automatically by CMake:

- SFML 2.6.x (Simple and Fast Multimedia Library)

Required Tools

Tool	**Version**	**Purpose**
CMake	$\geq$ 3.15	Configures and builds the project
MSYS2 / MinGW (UCRT64)	Latest	Compiler and shell environment
Ninja	(with CMake)	Build system
Git	Optional	To clone the repository

3. Downloading the Program

Option 1 – Git (Recommended)

```
git clone https://github.com/Dazzlepuff/Spotlight.git
cd Spotlight
```

Option 2 – Manual Download

1. Visit the GitHub repository.
2. Click `Code → Download ZIP`.
3. Extract it to your preferred folder.

4. Building the Program

Step 1 – Open the Correct Terminal

Launch **MSYS2 UCRT64** from your Start Menu to ensure the correct compiler and environment.

Step 2 – Navigate to Project Folder

```
cd /c/Users/<YourUser>/Documents/CodingProjects/Spotlight
```

Step 3 – Use Existing Build Folder or Create New One

The repository includes a `build/` folder with precompiled files. You can use it directly, or create a separate build directory if you prefer:

Option A – Use existing build folder:

```
cd build
```

Option B – Create a new build folder (for clean compilation):

```
mkdir build-new
cd build-new
```

Step 4 – Configure with CMake (if using new build folder)

If you created a new build directory, configure it:

```
cmake -G "Ninja" ..
```

CMake will:

- Fetch SFML 2.6.x automatically from GitHub.
- Generate Ninja build files.

Step 5 – Compile

```
cmake --build .
```

The compiled program will appear as `boardgame.exe` in the `build/` directory.

5. Running the Program

From Command Line

```
./boardgame
```

From Visual Studio Code

1. Open the project in VS Code.
2. Build with `Ctrl + Shift + B` (runs "CMake: build").
3. Press `F5` to launch with the included debugger configuration.

6. Program Flow

After launching, the program begins in the **Startup Menu**, allowing users to:

- Start Game
- Open Settings
- Exit

Option	**Description**
Start Game	Launches a new game using the current configuration.
Settings	Opens the settings menu to customize players and companies.
Exit	Closes the application.

7. Settings Menu

The Settings Menu allows configuration of player and company details. You will be prompted to enter:

- Number of players (2–6)
- Player names
- Company names

- Company symbols

Example Interaction:

```
=== SETTINGS ===
Enter number of players (2-6): 6
Player 1 name: Lena
Company name: Orion Dynamics
Company symbol: $
Player 2 name: Kai
Company name: Solaris Group
Company symbol: #
Player 3 name: Mira
Company name: Eclipse Labs
Company symbol: %
Player 4 name: Dax
Company name: Zenith Freight
Company symbol: &
Player 5 name: Nova
Company name: Crimson Forge
Company symbol: *
Player 6 name: Riven
Company name: AstraCore Industries
Company symbol: @
Settings saved!
```

Settings are saved to:

```
/Documents/CodingProjects/Spotlight/config/settings.txt
```

8. Starting a Game

When **Start Game** is selected:

- The configuration is loaded.
- The board radius is chosen automatically.
- Each player and company is initialized.
- The main game loop begins.

```
Starting game with 6 players (Board radius: 4)
```

9. Gameplay Interface

Visual Display

A graphical SFML window opens showing a hexagonal board grid, where each tile may display:

- Color (type or status)
- Ownership (linked to a company)

Text Console

You can enter commands in the terminal to interact with the game world. Type `help` at any time to list available commands.

10. Available Console Commands

Command	**Description**
set_color <x> <y> <z> <color>	Sets the color of a tile.
set_owner <x> <y> <z> <company_index>	Assigns ownership of a tile.
build <x> <y> <z> <color> [player_index]	Constructs a building on a tile.
list_players	Lists all players and their companies.
show_resources [player_index]	Displays a player's current resources.
give_resource <resource> <amount> [player_index]	Adds resources to a player.
spend_resource <resource> <amount> [player_index]	Deducts resources from a player.
show_cards [player_index]	Lists cards held by a player.
draw_card <deck_name> <amount> [player_index]	Draws cards for a player.
play_card <card_name> [player_index]	Plays a card from the hand.
end_turn	Ends the current player's turn.
next	Scrolls to next page of output.
clear	Clears the console text.
help	Displays available commands.

Example:

```
> list_players
1: Lena (Orion Dynamics)
2: Kai (Solaris Group)
3: Mira (Eclipse Labs)
4: Dax (Zenith Freight)
5: Nova (Crimson Forge)
6: Riven (AstraCore Industries)

> set_color 0 1 -1 red
Tile (0, 1, -1) color set to red.

> build 0 0 0 blue 1
Player 1 built a structure on tile (0, 0, 0).
```

11. Saving and Loading

The game automatically saves configuration data when modified. Each session reuses `settings.txt`. Full in-game state saving may be implemented in future updates.

12. Troubleshooting

Problem	**Cause**	**Fix**
Error: Could not write to settings.txt	Missing config directory	Ensure write permissions; program creates directories automatically.
SFML errors on launch	Missing graphics context	Run from MSYS2 UCRT64 or ensure SFML DLLs exist.
Window doesn't open	Misconfigured environment	Rebuild with CMake to restore dependencies.

13. Example Session

```
=== MAIN MENU ===
1. Start Game
2. Settings
3. Exit
Choose an option: 1

Starting game with 6 players (Board radius: 3)
> help
Available commands:
  set_color <x> <y> <z> <color>  - Sets tile color
  build <x> <y> <z> <color> [player_index]  - Builds structure
```

```
end_turn  - Ends the current player's turn
```

14. Optional: Running the Prebuilt Version

If you cloned the repository and a precompiled executable exists:

- Double-click `boardgame.exe`.
- The main menu will appear immediately – no compilation required.

15. Summary

Step	**Action**
1	Install MSYS2, CMake, Ninja
2	Clone or extract repository
3	Configure with `cmake -G "Ninja" ..`
4	Build with `cmake --build .`
5	Run `./boardgame`
6	Use menus and commands to play