Testing Manual

Table of Contents:

(1) Introduction

This testing manual outlines the testing plan, strategy, and scenarios for the Snake Game application. The testing approach aims to ensure that the game functions as intended under various conditions, both typical and edge cases. This document also includes a comprehensive set of test cases that cover different aspects of the game's functionality, user interface, and performance.

---

(2) Testing Plan

Objectives

The primary objectives of the testing process are:

- To verify that all game features function correctly according to the specified requirements.
- To identify and fix any bugs or inconsistencies in the game logic and user interface.
- To ensure that the game performs well under different scenarios, including edge cases and stress conditions.
- To validate that the game settings are saved and loaded correctly across sessions.

Scope

The testing process will cover:

- Game initialization and settings.
- Single-player and two-player gameplay.
- User interface elements and interactions.
- Performance under different grid sizes and difficulty levels.
- Error handling and recovery.

Resources

- **Testing Environment:** The game will be tested on multiple platforms, including Windows and macOS.
- **Testers:** All members in the team.
- **Tools:** Manual testing tools and Java debugging tools.

Deliverables

- A summary report of the test results, including pass/fail status for each test case.

---

(3) Testing Strategy

Manual Testing Approach

The testing process will primarily involve manual testing due to the interactive nature of the game. Testers will manually interact with the game, following predefined scenarios, and observing the behavior of the application.

Test Categories

1. **Functional Testing:**
    - Ensure that all features work as expected.
    - Verify game mechanics, including snake movement, food spawning, and collision detection.
2. **User Interface Testing:**
    - Validate the layout, color schemes, and user interaction flows.
    - Ensure that the settings menu functions correctly.
3. **Performance Testing:**
    - Test the game's performance under different conditions, such as large grid sizes and high difficulty levels.

4. **Boundary Testing:**
    - Test the game's behavior at the edge of allowable input values (e.g., minimum and maximum grid sizes).
5. **Error Handling Testing:**
    - Verify that the game handles errors gracefully, such as invalid input or missing settings files.

---

(4) Testing Scenarios

Scenario 1: Game Initialization

- **Objective:** Verify that the game initializes correctly with default settings.
- **Steps:**
    1. Launch the game.
    2. Observe the initial settings menu.
    3. Start the game with default settings.
- **Expected Outcome:** The game should start without errors, and the snake should appear on the game board.

Scenario 2: Customizing Game Settings

- **Objective:** Ensure that the user can customize game settings and that these settings are applied correctly.
- **Steps:**
    1. Open the settings menu.
    2. Change various settings (e.g., grid size, snake color, difficulty).
    3. Start the game and verify that the settings are reflected in the gameplay.
- **Expected Outcome:** The game should reflect the customized settings, and gameplay should proceed as expected.

Scenario 3: Single-Player Gameplay

- **Objective:** Test the functionality of single-player mode.
- **Steps:**
    1. Start the game in single-player mode.
    2. Play the game, attempting to eat food and grow the snake.
    3. Observe the snake's movement, collision detection, and score increment.
- **Expected Outcome:** The snake should move correctly, grow when eating food, and the game should end when the snake collides with the wall or itself.

Scenario 4: Two-Player Gameplay

- **Objective:** Test the functionality of two-player mode.
- **Steps:**
    1. Start the game in two-player mode.
    2. Control both snakes simultaneously.
    3. Observe the movement of both snakes and ensure no collision occurs between them.
- **Expected Outcome:** Both snakes should move independently. The game should end for each player only when their respective snake collides with a wall, collides with itself, or collides with the other snake.

Scenario 5: Borderless Mode

- **Objective:** Verify that the borderless mode functions as expected.
- **Steps:**
    1. Enable borderless mode in the settings menu.
    2. Start the game and move the snake across the edges of the board.
- **Expected Outcome:** The snake should reappear on the opposite side of the board when it moves off the edge, without any issues.

Scenario 6: Game Over Screen

- **Objective:** Ensure that the game over screen displays correctly and shows the correct scores.
- **Steps:**
    1. Play the game until the snake collides and the game ends.
    2. Observe the game over screen.
    3. Check that the scores are displayed and that the "Back to Menu" button functions.
- **Expected Outcome:** The game over screen should show the correct scores and allow the user to return to the menu.

Scenario 7: Saving and Loading Settings

- **Objective:** Verify that game settings are saved between sessions.
- **Steps:**
    1. Customize the game settings and start the game.
    2. Close and reopen the game.
    3. Check if the settings are preserved.
- **Expected Outcome:** The settings should be saved and loaded correctly when the game is restarted.

---

(5) Test Cases

Test Case 1: Default Game Initialization

- **Description:** Verify the game starts with default settings.
- **Steps:**
    1. Launch the game.
    2. Start the game without changing any settings.
- **Expected Result:** The game should start with a 20x20 grid, default colors, and medium difficulty.

Test Case 2: Invalid Grid Size Handling

- **Description:** Ensure that the game handles grid sizes smaller than the minimum allowable size.
- **Steps:**
    1. Enter a grid size of 5x5 in the settings menu.
    2. Attempt to start the game.
- **Expected Result:** The game should display an error message and prevent the game from starting.

Test Case 3: Changing Snake Colors

- **Description:** Verify that changing the snake color works correctly.
- **Steps:**
    1. Open the settings menu.
    2. Change the color of Snake 1 to red and Snake 2 to yellow.
    3. Start the game in two-player mode.
- **Expected Result:** Snake 1 should appear in red, and Snake 2 should appear in yellow on the game board.

Test Case 4: Borderless Mode Functionality

- **Description:** Test the functionality of borderless mode.
- **Steps:**
    1. Enable borderless mode.
    2. Start the game and move the snake to the edge of the board.
- **Expected Result:** The snake should appear on the opposite edge of the board when it moves off the screen.

Test Case 5: Game Over Condition

- **Description:** Ensure that the game ends correctly when the snake collides with itself or a wall.
- **Steps:**
    1. Play the game and intentionally collide with a wall or the snake's body.
    2. Observe the game over screen.
- **Expected Result: The game should end, and the game over screen should display the correct score.**

Test Case 6: Two-Player Movement

- **Description:** Verify that both snakes can move independently without interfering with each other.
- **Steps:**
    1. Start the game in two-player mode.
    2. Move Snake 1 with the arrow keys and Snake 2 with the WASD keys.
- **Expected Result:** Both snakes should move as directed without colliding with each other.

Test Case 7: Countdown Timer Visibility

- **Description:** Ensure that the countdown timer appears before the game starts.
- Steps:
    1. Start a new game.
    2. Observe the countdown timer.
- **Expected Result:** The countdown timer should display for 3 seconds before the game starts.

Test Case 8: Saving Settings

- **Description:** Verify that the game settings are saved correctly.
- **Steps:**
    1. Change the game settings (e.g., grid size, colors).
    2. Start the game and then close it.
    3. Reopen the game and check the settings.
- **Expected Result:** The settings should be preserved from the previous session.

Test Case 9: High Difficulty Performance

- **Description:** Test the game's performance and responsiveness at the highest difficulty level.
- **Steps:**
    1. Set the difficulty to "Hard" in the settings menu.
    2. Start the game and play for at least 5 minutes.
    3. Observe the game's performance and responsiveness.
- **Expected Result:** The game should remain responsive, with smooth movement of the snake, even at high speed.

Test Case 10: Minimal Grid Size Gameplay

- **Description:** Verify gameplay functionality on the smallest allowed grid size (10x10).
- **Steps:**
    1. Set the grid size to 10x10 in the settings menu.
    2. Start the game and attempt to play for several minutes.
- **Expected Result:** The game should function correctly, with the snake moving smoothly and food spawning properly on the 10x10 grid.

Test Case 11: Large Grid Size Performance

- **Description:** Test the game's performance on a large grid size (e.g., 50x50).
- **Steps:**
    1. Set the grid size to 50x50 in the settings menu.
    2. Start the game and play for at least 5 minutes.
- **Expected Result:** The game should handle the large grid size without performance issues, and the snake should move fluidly.

Test Case 12: Handling Rapid Direction Changes

- **Description:** Ensure that the game handles rapid direction changes correctly.
- **Steps:**
    1. Start the game and continuously change the snake's direction rapidly using the arrow keys.
    2. Observe the snake's movement and responsiveness.
- **Expected Result:** The snake should follow the direction changes accurately, without skipping or reversing unintentionally.

Test Case 13: Snake Length and Game Over

- **Description:** Test how the game handles a long snake and potential self-collision.
- **Steps:**
    1. Play the game until the snake grows to a considerable length (e.g., 10 segments).
    2. Maneuver the snake to collide with itself.
- **Expected Result:** The game should end correctly when the snake collides with itself, and the game over screen should display.

Test Case 14: Color Contrast Verification

- **Description:** Verify that the selected colors for snakes, food, and background provide sufficient contrast.
- **Steps:**
    1. Choose a combination of colors in the settings menu (e.g., dark green snake on a black background).
    2. Start the game and observe the visibility of the snakes and food.
- **Expected Result:** The snakes and food should be clearly visible against the background, and the gameplay should not be hindered by poor color contrast.

Test Case 15: Food Spawning and Consumption

- **Description:** Ensure that food spawns in valid locations and is consumed correctly by the snake.
- **Steps:**
    1. Start a game and observe the food spawning behavior.
    2. Move the snake to consume food and check if the score increases and the snake grows.
- **Expected Result:** Food should spawn in locations that are not occupied by the snake, and consuming food should increase the snake's length and score correctly.

Test Case 16: Restarting the Game

- **Description:** Test the behavior when restarting the game after a game over.
- Steps:
    1. Play until the game ends (game over).
    2. Click "Back to Menu" and start a new game with the same or different settings.
- **Expected Result:** The game should restart correctly, with no residual state from the previous game affecting the new session.

Test Case 17: Invalid Input Handling

- **Description:** Ensure that invalid inputs in the settings menu are handled gracefully.
- **Steps:**
    1. Enter non-numeric characters in the width and height fields.
    2. Attempt to start the game.
- **Expected Result:** The game should display an error message and prevent starting the game until valid inputs are provided.

Test Case 18: Screen Resizing During Gameplay

- **Description:** Verify the game's behavior when the window is resized during gameplay.
- **Steps:**
    1. Start the game and resize the game window.
    2. Observe the game's layout and functionality after resizing.
- **Expected Result:** The game should handle window resizing gracefully, maintaining proper display and functionality.

Test Case 19: Snake Collides with Itself

- **Description**: Verify that Snake dies when its head collides with its own body.
- **Steps**:
    1. Start the game in one-player mode or two-player mode.
    2. Maneuver Snake to collide its head with its own body.
    3. Observe the game's response.
- **Expected Result**: Snake should die and the game will end(for both player in two-player mode)

Test Case 20: Snake 1 Collides with Snake 2's Body

- **Description**: Verify that Snake 1 dies when its head collides with any part of Snake 2's body.
- **Steps**:
    1. Start the game in two-player mode.
    2. Maneuver Snake 1 to collide its head with Snake 2's body (not the head).
    3. Observe the game's response.
- **Expected Result**: The game should end, and both snakes should die when Snake 1's head collides with Snake 2's body.

Test Case 21: Snake 2 Collides with Snake 1's Body

- **Description**: Verify that Snake 2 dies when its head collides with any part of Snake 1's body.
- **Steps**:
    1. Start the game in two-player mode.
    2. Maneuver Snake 2 to collide its head with Snake 1's body (not the head).
    3. Observe the game's response.
- **Expected Result**: The game should end, and both snakes should die when Snake 2's head collides with Snake 1's body.

Test Case 22: Head-to-Head Collision Between Snakes

- **Description**: Verify that both snakes die when their heads collide directly with each other.
- **Steps**:
    1. Start the game in two-player mode.
    2. Maneuver Snake 1 and Snake 2 so that their heads collide with each other.
    3. Observe the game's response.
- **Expected Result**: The game should end, and both snakes should die when their heads collide directly.
