Test Plan: Robot Duel Game

1. Test Setup:

- Install and launch the Robot Duel game.
- Ensure that both "Your robot" and "Computer's robot" are present and functional.
- Verify that the health, attack damage, and other relevant attributes of the robots are displayed correctly.
- Confirm that the calculation for determining the winner is based on the specified formula.

2. **Functional Testing:**

a. Robot Attributes:

- Test that the health and attack damage values are correctly assigned to "Your robot" and "Computer's robot."
- Validate that the attributes can be modified within acceptable ranges.
- Verify that invalid attribute inputs are handled properly (e.g., non-numeric values, negative values).

b. Duel Outcome Calculation:

- Execute duels between different combinations of robots with varying attributes.
- Confirm that the outcome calculation accurately determines the winner based on the given formula.
- Test scenarios where the result could be a tie or a negative value.

c. User Interface:

- Interact with the game interface and verify that all buttons, menus, and controls are functional.
- Test different screen resolutions and aspect ratios to ensure proper display and usability.
- Check for any visual glitches, overlapping elements, or misaligned UI components.

Performance Testing:

- Conduct stress testing by simulating a large number of duels in a short time frame.
- Monitor system resources (CPU, memory, etc.) during the stress test.
- Evaluate the game's performance and responsiveness under heavy load.
- Identify any performance bottlenecks or issues that may affect the gameplay experience.

Bug Reporting:

- Document any bugs encountered during testing with detailed steps to reproduce.

- Include information about the observed behavior and expected behavior.
- Assign appropriate severity and priority levels to each reported bug.
- Provide screenshots or video recordings, if necessary, to support bug reports.

Compatibility Testing:

- Test the game on different platforms (Windows, macOS, Linux, iOS, Android) if applicable.
- Verify compatibility with various web browsers or gaming consoles, if applicable.
- Ensure that the game functions correctly on different hardware configurations.

Usability Testing:

- Gather a group of users to play the game and provide feedback on its usability.
- Observe how users interact with the game and identify any areas that cause confusion or frustration.
- Collect feedback on the game's controls, visuals, and overall user experience.
- Use the feedback to make improvements and enhance the game's usability.

Regression Testing:

- Repeat the functional tests mentioned above after each bug fix or game update.
- Verify that the reported bugs have been resolved and no new issues have emerged.
- Ensure that the overall functionality and performance of the game have not regressed.

Final Documentation:

- Compile a final report summarizing the testing process, including test results and bug fixes.
- Provide recommendations for further improvements based on the test findings.
- Include any additional observations, suggestions, or user feedback gathered during testing.