

Test/QA Plan

2 Player Online Game

1. Game Overview:

Game Concept: 2 Player classic online game.

Objective: Players can successfully join the lobby and play different games.

Target Audience: Any players that want to have some fun playing classic games.

Platforms: Web browser (Chrome, Safari, Firefox, Atom), Phone, Tablet.

2. Testing Objectives:

- Functional Testing: Ensure all game features work as intended.
- Performance Testing: Evaluate the game's performance under various conditions.
- Multiplayer Stability: Test the game's stability during online multiplayer sessions.
- User Experience: Assess the overall user experience, including UI/UX design and controls.
- Compatibility: Verify compatibility across different devices, operating systems, and browsers (if applicable).

3. Test Cases:

Gameplay Testing:

- Test basic functions from both the front end and back end for each player.
- Verify game rules and win/lose conditions.

Multiplayer Testing:

- Test online matchmaking and player connectivity.
- Evaluate server stability under different loads.
- Test synchronization between players' responses in the lobby.

UI/UX Testing:

- Test menus, HUD elements, and in-game displays.
- Evaluate the readability, intuitiveness, and responsiveness of UI elements.

Performance Testing:

- Test loading times for different games.
- Monitor the connection between the frontend and with backend(web socket)

Compatibility Testing:

- Test the game on various devices and screen sizes.
- Ensure compatibility with different operating systems and versions.
- Test different web browsers.

4. Bug Tracking and Reporting:

- Categorize bugs based on severity (critical, major, minor) and priority (high, medium, low) and store them in Jira or other tools.
- Provide detailed steps to reproduce each issue, including screenshots or videos if necessary.

5. Automation Testing:

- Conduct regression automation testing after each code change or bug fix to ensure existing features are unaffected.
- Create and maintain a regression test suite covering all core game features.

6. User Acceptance Testing:

- Conduct UAT with select players to gather feedback on gameplay, user experience, and overall satisfaction.
- Use player feedback to make necessary improvements before the official release.

7. Code Quality and Builds

Before pushing a change to the repo or finishing editing each game, a build of the game should be created in a new branch to ensure that the new code will not cause any issues with future builds. Ideally, code should also be tested across the network to ensure replication in accordance with the project requirements above.

Optional Testing If Time Allowed:

Security Testing:

- Evaluate the game for potential security vulnerabilities such as cheating, data manipulation, or unauthorized access.
- Implement encryption and secure communication protocols to protect player data.

Support Plan:

- Outline the plan for post-launch support, including regular updates, bug fixes, and feature enhancements based on player feedback.
- Set up customer support channels to address player issues and inquiries.

Documentation:

- Include information about game mechanics, controls, known issues, and troubleshooting tips.