ROCK – PAPER - SCISSORS

Emily Rekewitz

Alina Römmelt

Kätlin Tamm

Table of Contents

1. Introduction

1.1 Purpose of the requirements document

The purpose of this document is to outline the functional and non-functional requirements for the Rock-Paper-Scissors Game. It provides a detailed description of the game's features, intended user interactions, and the technical constraints that will guide its development. The document serves as a reference for developers, testers, and stakeholders involved in the project.

1.2 Scope of the product

The product is a simple rock-paper-scissors game where a player competes against a computer. The game will provide immediate feedback on each round, track the player's score, and offer a seamless user experience across various platforms. This game is intended for casual play, with a future possibility for enhanced features like multiplayer mode and game statistics tracking.

1.3 Definitions, acronyms, and abbreviations

- RPS: Rock-Paper-Scissors
- UI: User Interface
- DFD: Data Flow Diagram

1.4 References

Random number generation algorithm for computer selection
User interface design guidelines for game applications

1.5 Overview of the remainder of the document

The remainder of this document provides an in-depth explanation of the game's general and specific requirements, including technical constraints, user characteristics, and assumptions. The final sections cover the appendices and the index for easy navigation.

2. General Description

2.1 Product perspective

The Rock-Paper-Scissors Game is a standalone application designed for single-player use. The player competes against the computer, which selects an option at random. The game will be simple, fast, and accessible to a wide range of users

2.2 Product Functions

The core functions of the game include:

- Allowing the player to select rock, paper, or scissors.
- Randomly generating the computer's choice.
- Determining the winner based on standard rules.
- Displaying immediate feedback to the player.
- Tracking the score and maintaining it across multiple rounds.

2.3 User Characteristics

The target users are casual players of all ages, from children to adults, who are familiar with simple games. No special skills are required to play the game. The UI will be designed to ensure ease of use for players of all experience levels.

2.4 General Constraints

The game must load quickly and run smoothly on all major browsers and devices.
Randomness must be consistent and fair for each round.
The game should provide a clear and responsive user interface.

2.5 Assumptions and Dependencies

The product assumes that users have access to a device capable of running a web-based application.
The system relies on external libraries or algorithms for generating random computer choices.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **UI Design:** The game will have a simple and intuitive user interface that displays the options (rock, paper, scissors), the current score, and the results of each round.
- **Feedback Mechanism:** Immediate feedback will be provided after each selection, showing the player's choice, the computer's choice, and the outcome.

3.1.2 Hardware Interfaces

- **Input Devices:** The game should accept input from standard devices such as a mouse or touchscreen.
- **Output Devices:** The game will display on standard computer screens, tablets, and mobile devices.

3.1.3 Software Interfaces

- B**rowser Compatibility:** The game should be compatible with all major web browsers (e.g., Chrome, Firefox, Safari, Edge).
- **Third-Party Libraries:** If applicable, the game may use third-party libraries for random number generation or UI components.

3.1.4 Communications Interfaces

The game can function offline; however, if future multiplayer features are added, it will require internet connectivity for player matching.

3.2 Functional Requirements

3.2.1 Mode 1: Single Player

3.2.1.1 Functional Requirement 1.1: User Selection
The player must be able to select either rock, paper, or scissors.

3.2.1.2 Functional Requirement 1.2: Computer Selection
The computer must randomly select one of the three options.

3.2.1.3 Functional Requirement 1.3: Determine Winner

The game must evaluate the winner based on established rules and display the result.

3.2.2 Mode 2: Multiplayer (Future Enhancement)

3.2.2.1 Functional Requirement 2.1: Player Matching
The system should allow players to join a game with a friend online.

3.2.2.2 Functional Requirement 2.2: Real-Time Feedback
Players should receive real-time updates on choices and results during multiplayer matches.

3.3 Performance Requirements

The game should load within 2 seconds on standard internet connections.
User interactions (button clicks, selections) should be registered with less than 200ms delay.

3.4 Design Constraints

The user interface must adhere to accessibility standards to accommodate users with disabilities.
The game must be designed to function across a variety of screen sizes without loss of usability.

3.5 Software System Attributes

- **Reliability:** The game must have a failure rate of less than 1% during normal use.
- **Usability:** The interface should be easy to navigate, with a minimum success rate of 90% in user testing.

3.6 Other Requirements

- **Documentation:** Comprehensive user documentation should be provided, including installation instructions and gameplay guidelines.
- **Support and Maintenance:** A plan should be in place for addressing bugs and releasing updates.

4. Appendices

4.1 Test Cases

- **User input testing:** Verify that the player's choice is registered correctly.
- **Result testing:** Ensure that the game accurately displays a win, loss, or draw according to the rules.
- **Scoring testing:** Check that the game score updates correctly after each round.
- **Smooth transition:** Test that the game responds without noticeable lag to user selections (less than 200ms).

4.2 Data Flow Diagram (DFD)