Software Requirements Specification for Whack A Prof Game

Contents

1 Introduction

1.1 Purpose

The purpose of this SRS document is to provide a description of the "Whack a Prof" game. This document is intended to be a guide for designers, developers, and testers (Quality Assurance), ensuring that the game is developed according to the specified requirements and that there is a clear understanding of its objectives, mechanics, and expected behavior.

The intended audience for this document includes:

- Backbone: Responsible for implementing the game's features and mechanics based on the requirements.

- Graphics Team: Responsible for developing the game's visuals, including characters, effects (e.g., blood splatter effects, hit effects), items (e.g., mallet), animations (e.g., mallet swing), sounds (e.g, mallet hit sound effects).

- QA Team: Responsible for verifying that the game functions as expected and meets the outlined specifications.

Each group will use this SRS to understand their roles and ensure that the game meets the overall project goals.

1.2 Scope

The software product to be produced followed by the SRS outline should be named "Whack A Prof". This software product would be a game that will follow a similar fashion to the original "Whack A Mole" game, where users would be timed and scored to hit characters in a whack a mole style fashion.

The application of the software being specified — "Whack A Prof" is an interactive time-based game designed to simulate a "Whack A Mole" experience but with a unique feel. In this game, users will attempt to hit characters (i.e., Professors or Board of Trustee Members) that pop up randomly on the screen within a limited time frame, i.e., in a "Whack A Mole" styled fashion. The player is rewarded with points for successful hits, and performance will be tracked through a score system. The player will be able to enter their name before playing the game, the player will be able to adjust settings in the game, the player will be able to pause/quit the game, and the player will be able to view their scores.

Benefits of Application

- User Engagement: The game will provide players with an engaging experience with a timed nature and with features such as score tracking during gameplay.

- Entertainment: The game will provide players with entertainment that challenges the players' reaction time.

Objectives of Application

- Timed Gameplay: Players have a limited time.

- Accurate Scoring: Players will be scored based on how many characters they have hit.

The overall goal of "Whack A Prof" is to replicate and simulate the original "Whack A Mole" game while giving players an enjoyable experience.

1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification.

- Whack A Mole: The game name, where players score points by hitting characters (usually moles) that randomly appear on the screen.

- UI: User interface, the elements through which players interact with the game.

- QA: Quality Assurance, the team responsible for testing the game to ensure its functionality and performance meet the specifications.

- Hit Detection: The system that detects when a player successfully hits a target and responds accordingly.

- Score System: A system that tracks the player's performance in the game, usually based on the number of targets hit or the accuracy of the hits.

- Target: Objects in the game, moles, that players need to hit to score points.

- FPS: Frames Per Second, referring to the refresh rate of the game's visuals, which ensures smooth gameplay.

- WIP: Work In Progress - optional/required requirements will be added soon.

1.4 References

- https://en.wikipedia.org/wiki/Whac-A-Mole

- Whack-A-Mole gameplay - Offline Version

1.5 Overview

This document is organized to guide the development, design, and testing of the game Whack A Prof ensuring that all teams involved clearly understand the objectives, scope, and requirements. It starts with the Introduction, which defines the purpose and intended audience of the document, including the Backbone, Graphics, and QA teams, each with specific roles to implement, design, and verify the game. The Scope outlines the game's core mechanics, modeled after the classic Whack A Mole game. It also lists the Objectives of the game, which include providing user engagement through timed gameplay and ensuring accurate scoring. The Definitions and Acronyms section clarifies key terms such as SRS, UI, and QA, which are essential for understanding the technical elements discussed throughout the document. The references section provides any material used or referenced in the creation of the Whack A Prof game. This document is structured to ensure that each team understands their role in fulfilling the game's design and functional goals, ensuring the game is developed according to the outlined requirements.

2 Product Perspectives

2.1 System interfaces

The game will consist of 3 things the player can access: Start/Title, Options, Game.

2.2 User interfaces

Start/Title Screen: The first thing the player sees when they load the app. It will have the buttons to start the game, exit to desktop, or options.

Options/Settings: Allows the user to configure the game to have the most comfortable experience. Options include brightness, quality, etc.

Game: The hammer will replace the mouse pointer, there will be 9 holes on screen in a 3x3 layout. Pressing Esc will bring up a menu for options or to return to the title screen.

2.3 Hardware interfaces

The game can be played in either fullscreen or windowed mode. It can be supported by most operating systems (i.e. Windows, Mac, Linux), or most browsers (i.e. Chrome, Firefox).

2.4 Software interfaces

The game's name is "Whack A Prof". While the game is in development, there should be beta or alpha versions to make sure the game is at least usable before moving to the next step of development. An example would be "Beta 0.01". When the game is finished, it should be released as "Version 1.0".

2.5 Communications interfaces

The game itself should not require WiFi to play unless the game is on a website. If achievements and the ability to share achievements on social media (i.e. Steam) is developed, it will have to require WiFi.

2.6 Memory

The base game should aim to need 1-2GB worth of storage space due to the simplicity of the game. However, this is not a strict requirement and exceeding that amount is acceptable.

2.7 Operations

When the player starts the game, they are first met with the Title screen. If they need to change the settings to improve their experience, they can go to the Options/Settings button to change it. To actually start the game, the player must press the Start button.

To play the game, the player must press the hit button (default: Left-Click) when the hammer/mouse pointer is over a Prof to score points.

To leave the game, the player must press Esc and return to the Title screen.

2.8 Site adaptation requirements

The game will likely not need any specific site requirements other than the game being embedded.

3 Product Functions

3.1 User Interface Functions

- **Main Menu:** In the main menu, I will display options such as "Start Game," "Settings," "High Scores," and "Exit." This interface will feature clear visual cues, allowing users to effortlessly select their desired action.

- **Game Interface:** The game interface will showcase the playing field where characters, such as professors or board members, appear for players to hit. It will include a timer countdown to add urgency and a score display to keep players informed of their progress.

- **Settings Menu:** Players will have the option to customize their experience through the settings menu. Here, they can adjust sound levels, graphics settings, and gameplay preferences to suit their needs.

3.2 Gameplay Functions

- **Character Generation:** The game will randomly generate characters that pop up on the screen, ensuring a diverse and engaging gameplay experience. The variety in character appearance and frequency will keep players on their toes.

- **Hit Detection:** To enhance player interaction, the game will capture hits on characters through mouse or touch input. Immediate feedback will be provided via sound effects and visual cues, such as blood splatter effects, reinforcing the game's theme.

- **Scoring System:** The scoring system will award points for successful hits, with bonuses for speed and accuracy. This system will track the total score throughout the game and display it at the end of each session, encouraging competitive play.

3.3 User Interaction Functions

- **Player Input:** Players will be able to start, pause, or restart the game using keyboard or mouse inputs. Additionally, they will have the opportunity to input their name for the high score leaderboard before starting the game, fostering a sense of achievement.

- **Feedback Mechanisms:** To enhance the gaming experience, auditory and visual feedback will be provided for both successful hits and misses. Messages will be displayed after each round, offering encouragement or constructive feedback, such as "Great Job!" or "Try Again!"

3.4 Data Management Functions

- **Score Tracking:** The game will save scores to a local leaderboard or database, allowing for the tracking of player performance. After completing a game session, users will have the option to view high scores, adding a competitive element.

- **User Profile Management:** To personalize gameplay, the game will store player names and high scores associated with their profiles. This feature will facilitate continuity for players during future sessions.

3.5 Quality Assurance Functions

- **Error Handling:** To ensure a smooth gaming experience, the game will detect and report any gameplay errors or glitches, such as characters failing to appear. When issues arise, the game will provide instructions or suggestions for resolving them.

- **Testing Tools:** I will include tools for QA testers to simulate various gameplay scenarios. These tools will facilitate thorough testing, ensuring that all functions operate as intended before release.

3.6 Performance Tracking Functions

- **Statistics Display:** After each game session, players will be presented with statistics like hit accuracy, total hits, and average reaction time. This data will help players understand their performance and areas for improvement.

- **Gameplay Analytics:** To continuously enhance the game, I plan to collect data on player interactions. This gameplay analytics will inform future updates and adjustments, creating a more engaging experience.

3.7 Help and Support Functions

- **Tutorial Mode:** To support new players, I will implement a tutorial mode that introduces gameplay mechanics and controls through guided steps. This mode will ensure that all players feel confident as they start their gaming journey.

- **Help Section:** Lastly, a help section will be included, providing FAQs and troubleshooting tips for common issues. This resource will enhance user satisfaction and retention by addressing player concerns effectively.

4 User Characteristics

4.1 User Characteristics

The general characteristics of the users of 'Whack A Prof' are:

- **Educational Level:** Players can range from high school students to college graduates and beyond. The game's simplicity makes it accessible to a wide audience, regardless of their formal education.

- **Experience:**

  - **Casual Gamers:** Gamers who enjoy playing simple, quick games for relaxation or entertainment.
  - **Experienced Gamers:** Gamers who have extensive experience with various types of games, including both casual and complex games.

- **Technical Expertise:**

  - **Basic computer skills:** Players will have computer skills, such as using a mouse or keyboard, navigating the internet, and installing the software.
  - **Intermediate skills:** Some players might have intermediate technical skills, including familiarity with different gaming platforms and troubleshooting minor technical issues.
  - **Advanced skills:** A subset of players might have technical expertise, such as knowledge of game development, programming, or modding games.

5 Constraints

5.1 Constraints

- **Regulatory policies:**

  - **Intellectual property rights:** 'Whack A Prof' avoids using copyrighted material without permission. This includes music, images, and characters.
  - **Violence and ethical content:** 'Whack A Prof' ensures the depiction of violence is appropriate for the game's context and avoids promoting real-world violence or stereotypes.
  - **Fair play and cheating measures:** 'Whack A Prof' implements measures to prevent cheating and ensure fair play.

- **Hardware Limitations:**

  - (TBD) Game has not been designed, but examples of hardware limitations can include memory, operating system (compatibility with Windows 10/11), development tools.

- **Interfaces to other applications:**

  - (TBD) We can allow players to share their achievements, scores, and game progress on social media.
  - Ensure your game can run on multiple platforms (PC, consoles, mobile devices) by using cross-platform development tools like Unity or Unreal Engine.

- **Parallel operation:**

  - (TBD) Does the game implement task parallelism (dividing a task into smaller sub-tasks that can be processed in parallel).

- **Audit functions:**

  - (TBD) Automated testing: does 'Whack A Prof' implement automated testing tools to continuously monitor and audit the game's functionality.

- **Control functions:**

  - (TBD) Input devices: 'Whack A Prof' supports various devices such as keyboards, mice, and touchscreens.
  - Control mapping: arrow up button means go up.

- **Higher-order language requirements:**

  - Abstraction and modularity: 'Whack A Prof' is written in JavaScript, which offers high levels of abstraction, allowing developers to write more modular and maintainable code.

- **Signal handshake protocols:**

  - (TBD) Does 'Whack A Prof' implement security for multiplayer interactions and data exchanges? This involves a handshake process where the client and server exchange keys and authenticate each other before data is transmitted (e.g., TLS, SSS, etc.).

- **Reliability requirements:**

  - Testing and quality assurance: 'Whack A Prof' is put through testing, including unit tests, integration tests, and user acceptance tests, to ensure the game functions as expected.
  - User feedback and bug reporting: provides easy-to-use mechanisms for players to report bugs and provide feedback.

- **Criticality of the application:**

  - User experience: the game provides a seamless and enjoyable experience.

- **Safety and security considerations:**

  - Access controls: 'Whack A Prof' uses strict access controls to limit who can access different parts of the game and its data. To ensure only authorized persons can perform certain actions
  - Secure coding practices: 'Whack A Prof' follow secure coding practices to minimize vulnerabilities in the game's code. This include input validation,and proper error handling.

6   Assumptions and Dependencies

6.1   Assumptions and Dependencies

- **Assumption:** The game will be on a web server that supports modern browsers for example (Chrome, Firefox, Safari) with JavaScript enabled.

- **Dependency:** If a user tries to play the game in an outdated browser or one that does not support JavaScript the game might not work as expected. In such a case, changes to the requirements would be needed, such as checking browser compatibility.

- **Assumption:** The game will be deployed on a website that has a stable internet connection and as well can handle multiple users accessing the website at the same time.

- **Dependency:** If the website hosting the game experiences downtime or poor performance due to limited bandwidth, the requirements might need to change to include performance optimization or the game might be able to be played offline.

- **Assumption:** The game will require users to interact via mouse clicks.

- **Dependency:** If the game needs to support additional and or more input methods, such as keyboard control, game controllers or touch screen, the SRS outline would need to change to accommodate the new input methods and there would need to be necessary changes in game mechanics.

7   Apportioning of Requirements or Optional Requirements

7.1   Apportioning of Requirements or Optional Requirements

- **Multiplayer Mode:** The initial release of the game will only support single-player mode. A future version of the game may include a multiplayer mode, allowing multiple users to compete simultaneously.

- **Delay:** Multiplayer support will be postponed to the final version.

- **Additional Characters and Themes:** The initial game will feature a single theme with a limited number of characters (ie, professors, board of trustees). Future versions may introduce additional professors or in game background themes, such as new character designs, animations.

- **Delay:** Additional characters and themes will/may be part of Version 3.0.

- **Mobile App Version:** The initial release of the game will be designed for web browsers, while a future version may include a mobile app version for Android and/or iOS devices.

- **Delay:** The mobile app version will be postponed for a potential future release.