SOFTWARE REQUIREMENTS SPECIFICATION

2D SWORD & SHIELD FIGHTING GAME

---

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to clearly define the **software requirements** for our project.

This document serves as a **structured guide** for the development process, helping to ensure that the project meets its intended **goals** and **functionality**.

It is intended for the **course instructor** who will review the assignment, as well as for the **developers** as a reference during implementation and testing.

1.2 Document Conventions

Bold text indicates important features and terms.

Functional requirements are numbered as **FR-x**, and non-functional requirements as **NFR-x**.*Terminology specific to the game:*

- **Dash:** A quick horizontal movement for repositioning and spacing.
- **Attack:** Player strikes in upward, horizontal, or downward directions.
- **Defend:** Blocking attacks from different directions.
- **Tutorial:** A short introductory guide that explains core mechanics.
- **HP (Health Points):** A numerical value representing player vitality.

Labeling Conventions Used in This Document:

Label	Meaning	Used In
FR-x	Functional Requirement	Section 3.3
NFR-x	Non-Functional Requirement	Section 4
UC-x	Use Case	Section 3.2

1.3 Intended Audience

- **Course Instructor:** To review the assignment and check whether the project meets the defined requirements.
- **Developers:** To use this document as a reference during the design, implementation, and testing, and to understand system features and implementation requirements.

1.4 Project Scope

In Scope:

- 1v1 local multiplayer gameplay with sword and shield combat mechanics.
- Player actions: movement (right or left), dash (right or left), attacking (upward, horizontal, downward), and defending (upward, horizontal, downward).
- Game menus: Main Menu, Options, Tutorial, Credits, and Pause Menu.
- Game results: match end screen with options to rematch or return to the main menu.
- Audio and resolution settings adjustable in the Options menu.

Out of Scope:

- AI-controlled opponents.
- Multiple playable characters.
- Online multiplayer functionality.

1.5 References

- **GitHub Repository Link:https://github.com/OguzhanCel/YZM2021-Project**
- **GitHub Project Board Link: github.com/users/OguzhanCel/projects/3**

---

2. OVERALL DESCRIPTION

2.1 Product Perspective

Our game is a **standalone application** developed using the **Godot Engine**.

It does not rely on any external systems, servers, or APIs.

The game is **self-contained**, focusing on providing **local 1v1 gameplay** between two players on a single computer.

2.2 Product Functions

- Display a **Main Menu** with options such as *Start Game*, *Tutorial*, *Options*, *Credits*, and *Exit*.
- Provide **attack**, **defend**, **dash**, and **movement** mechanics.
- Show a **Pause Menu** during gameplay with *Resume*, *Options*, and *Main Menu* choices.
- Display an **End Screen** after each match with *Rematch* and *Return to Main Menu* options.
- Allow players to adjust **volume**, **screen resolution**, and **fullscreen/windowed** mode in the *Options* menu.
- Present a short **Tutorial Section** explaining the basic controls and mechanics.

2.3 User Classes and Characteristics

- **Player:**
    - Expected to have basic computer knowledge.
    - Expected to be interested in 1v1 combat gameplay.

2.4 Operating Environment

The game will run as a desktop application on **Windows OS**.

Players will use **keyboard input** to control their characters.

2.5 Design and Implementation Constraints

- The game must be developed using the **Godot Engine**.
- Only **local multiplayer** is required; no network functionality should be implemented.
- Assets such as sprites, sounds, and backgrounds will be obtained from **online sources**.

2.6 Assumptions and Dependencies

- No external API or internet connection is required for gameplay.
- The **Godot Engine** provides all necessary tools for rendering, physics, and input management.
- Since some assets are obtained externally, certain original ideas may be modified or adapted to fit available resources.

---

3. FUNCTIONAL REQUIREMENTS

3.1 User Stories

Group 1: Main Menu

User Story 1 — Start Game

As a player

I want to start a new match from the main menu

So that I can begin playing the game immediately

Acceptance Criteria:

- **Given** the main menu is visible
- **When** the player clicks *Start Game*
- **Then** the game scene should load and both players should be able to control their characters

Priority: Must Have

User Story 2 — Access Tutorial

As a new player

I want to open the tutorial from the main menu

So that I can learn how to play before starting a match

Acceptance Criteria:

- **Given** the main menu is visible
- **When** the player clicks *Tutorial*
- **Then** the tutorial scene should load and display control explanations

Priority: Should Have

User Story 3 — Access Options Menu

As a player

I want to open the options menu

So that I can adjust settings such as volume and resolution

Acceptance Criteria:

- **Given** the main menu is visible
- **When** the player clicks *Options*
- **Then** an options screen should appear with adjustable settings

Priority: Should Have

User Story 4 — Exit Game

As a player

I want to quit the game safely from the main menu

So that I can close the game when finished

Acceptance Criteria:

- **Given** the main menu is visible
- **When** the player clicks *Exit*
- **Then** the game should close safely

Priority: Must Have

User Story 5 — Adjust Volume

As a player

I want to adjust the game's music and sound effect volume

So that I can customize the audio experience to my preference

Acceptance Criteria:

- **Given** the Options menu is open
- **When** the player changes the volume sliders
- **Then** the new audio levels are applied immediately

Priority: Should Have

User Story 6 — View Credits Screen

As a player

I want to view the credits from the main menu

So that I can see the developers and contributors of the game

Acceptance Criteria:

- **Given** the main menu is visible
- **When** the player clicks *Credits*
- **Then** a credits screen appears showing developer names and roles

Priority: Could Have

Group 2: Gameplay Mechanics

User Story 7 — Player Movement

As a player

I want to move my character left and right

So that I can position myself strategically during combat

Acceptance Criteria:

- **Given** both players are active in the game scene
- **When** the player presses the movement keys
- **Then** the character should move smoothly in the corresponding direction

Priority: Must Have

User Story 8 — Attack Mechanic

As a player

I want to perform attacks

So that I can damage my opponent and win the match

Acceptance Criteria:

- **Given** both players are active in the game scene
- **When** the attack button is pressed
- **Then** the character should perform the correct attack animation and deal damage

Priority: Must Have

User Story 9 — Defense Mechanic

As a player

I want to block incoming attacks

So that I can avoid taking damage

Acceptance Criteria:

- **Given** an opponent is attacking
- **When** the player holds the defense button
- **Then** the character should block the attack successfully if timed correctly

Priority: Must Have

User Story 10 — Dash Mechanic

As a player

I want to quickly dash in a direction

So that I can evade attacks or close distance between myself and my opponent

Acceptance Criteria:

- **Given** the player is in an idle state
- **When** the dash key is pressed
- **Then** the character should perform a short dash movement in the intended direction

Priority: Should Have

User Story 11 — Health Bar Display

As a player

I want to see my health bar and my opponent's health bar

So that I can track my progress and know who is closer to winning

Acceptance Criteria:

- **Given** the match is in progress
- **When** attacks land successfully
- **Then** both health bars should update accordingly
- **And** the match should end when one player's health reaches zero

Priority: Must Have

Group 3: UI and Menus

User Story 12 — Pause Menu

As a player

I want to pause the game during a match

So that I can take a break or access additional options

Acceptance Criteria:

- **Given** the match is ongoing
- **When** the player presses the pause button
- **Then** a pause menu should appear with *Resume*, *Options*, and *Main Menu* buttons
- **And** selecting *Resume* should continue the match from where it was paused

Priority: Must Have

User Story 13 — End Screen

As a player

I want to see the winner and post-match options after a game ends

So that I can choose to replay or return to the main menu

Acceptance Criteria:

- **Given** a match ends
- **When** one player's health reaches zero or the timer runs out
- **Then** an end screen should appear showing the winner
- **And** it should display *Rematch* and *Main Menu* buttons

Priority: Must Have

User Story 14 — Save Settings

As a player

I want to have my audio and display settings saved

So that I don’t need to readjust them every time I start the game

Acceptance Criteria:

- **Given** the player modifies settings
- **When** the game restarts
- **Then** the settings remain applied

Priority: Should Have

User Story 15 — Background Music Toggle

As a player

I want to mute or unmute the background music

So that I can focus or play silently if I prefer

Acceptance Criteria:

- **Given** the Options or Pause menu is open
- **When** the player toggles the *Music On/Off* button
- **Then** background music should start or stop accordingly

Priority: Could Have

3.2 Use Cases

UC-01 — Start Game from Main Menu

Actors: Player 1, Player 2

Preconditions: Game is running and Main Menu is on screen.

Main Flow:

1. Player selects **Start Game**.
2. Game scene loads.
3. Both players can control their characters and the match starts.
   Alternative Flows:

- If loading fails, show an error and return to the Main Menu.
  Postconditions: Match scene is active and both players are ready.

UC-02 — Perform Attack & Deal Damage

Actors: Player 1, Player 2

Preconditions: Both players are active in the match.

Main Flow:

1. Player presses the **attack** key.
2. The attack animation plays.
3. If it hits the opponent, damage is applied.
4. The opponent's HP decreases.
5. If HP reaches zero, the match ends.
   Alternative Flows:

- If the opponent blocks, damage is reduced.
- If the opponent is dashing, the attack may miss.
  Postconditions: HP values are updated; the round continues or ends.

UC-03 — Block or Dash to Avoid Damage

Actors: Player 1, Player 2

Preconditions: Match is active and both players can move.

Main Flow:

1. Player uses **Block** in the right direction.
2. Damage is blocked or reduced depending on timing.
3. Player can also use **Dash** to escape or reposition.
   Alternative Flows:

- Wrong timing → full damage is applied.
  Postconditions: Player avoids or takes reduced damage.

UC-04 — Pause and Resume Game

Actors: Player 1

Preconditions: A match is in progress.

Main Flow:

1. Player presses the **Pause** key.
2. The game stops and a Pause Menu appears.
3. Player chooses **Resume** to continue.
   Alternative Flows:

- Player opens **Options** to adjust volume.
- Player selects **Main Menu** to leave the match.
  Postconditions: Game continues or returns to Main Menu.

UC-05 — End Match and Rematch

Actors: Player 1, Player 2

Preconditions: One player's HP is 0 or timer runs out.

Main Flow:

1. The game checks who has more HP.
2. The winner is shown on the **End Screen**.
3. Players can choose **Rematch** or **Main Menu**.
   Alternative Flows:

- If both players have the same HP, it's a draw.
  Postconditions: A new match starts or players return to the Main Menu.

UC-06 — View Tutorial

Actors: Player 1

Preconditions: Main Menu is visible.

Main Flow:

1. Player selects **Tutorial**.
2. Tutorial scene opens and explains controls.
3. Player exits back to Main Menu.
   Alternative Flows:

- Player can directly start the game if they don't need to watch the tutorial.
  Postconditions: Player returns to Main Menu or match scene.

3.3 Functional Requirements List

Main Menu

- **FR-1:** The system shall display a **Main Menu** with options: Start Game, Tutorial, Options, Credits, and Exit.
- **FR-2:** The system shall start a new match when the player selects **Start Game**.
- **FR-3:** The system shall open the **Tutorial** scene when the player selects Tutorial.
- **FR-4:** The system shall open the **Options Menu** where players can adjust settings.
- **FR-5:** The system shall close the game when the player selects **Exit** from the Main Menu.
- **FR-6:** The system shall display a **Credits Screen** showing developer names and roles.

Gameplay Mechanics

- **FR-7:** The system shall allow each player to **move left and right** using their assigned keys.
- **FR-8:** The system shall allow each player to **perform attacks** (upward, horizontal, downward).
- **FR-9:** The system shall allow each player to **block attacks** from different directions.
- **FR-10:** The system shall allow each player to **dash** for repositioning or evading attacks.
- **FR-11:** The system shall display **health bars** for both players and update them after each hit.
- **FR-12:** The system shall determine a **winner** when one player's HP reaches zero or the timer ends.

UI and Menus

- **FR-13:** The system shall display a **Pause Menu** during gameplay when the player presses the pause key.
- **FR-14:** The Pause Menu shall include options for **Resume**, **Options**, and **Main Menu**.
- **FR-15:** The system shall display an **End Screen** showing the match result and options for Rematch or Main Menu.
- **FR-16:** The system shall save and apply **audio and display settings** changed by the player.
- **FR-17:** The system shall allow players to **toggle background music** on or off.
- **FR-18:** The system shall display **confirmation dialogs** before critical actions such as Exit Game, Restart, or Rematch.

---

4. NON-FUNCTIONAL REQUIREMENTS

4.1 Performance Requirements

- **NFR-1:** The game shall keep a **minimum frame rate of 60 FPS** on a standard mid-range computer.
- **NFR-2:** The time between a player’s input and the on-screen action shall not be **more than 10 ms**.
- **NFR-3: Loading screens** shall not take longer than **3 seconds** under normal conditions.

4.2 Usability Requirements

- **NFR-4:** The **tutorial** shall clearly explain all main mechanics (movement, attack, block, dash).
- **NFR-5:** All **menus** (Main Menu, Pause, End Screen) shall have a consistent layout and style.
- **NFR-6: UI elements** must have enough contrast so text and HP bars are easy to see.
- **NFR-7:** A **new player** shall be able to understand the controls and start playing within **2 minutes** after launching the game.

4.3 Reliability and Availability

- **NFR-8:** If a loading or input error happens, the game shall show an **error message** and return to the Main Menu.
- **NFR-9: HP, timer, and pause state** shall stay the same after pausing and resuming.
- **NFR-10:** If **saving settings** fails, the game shall automatically use the last valid data.

4.4 Maintainability Requirements

- **NFR-11:** The game codebase shall be **modular**, separating gameplay logic, UI, and physics components to make debugging and updates easier.
- **NFR-12:** New features or bug fixes shall be added without changing more than **20% of the existing code**.
- **NFR-13:** The system shall follow consistent **naming conventions** and **code formatting** based on the team’s style guide.

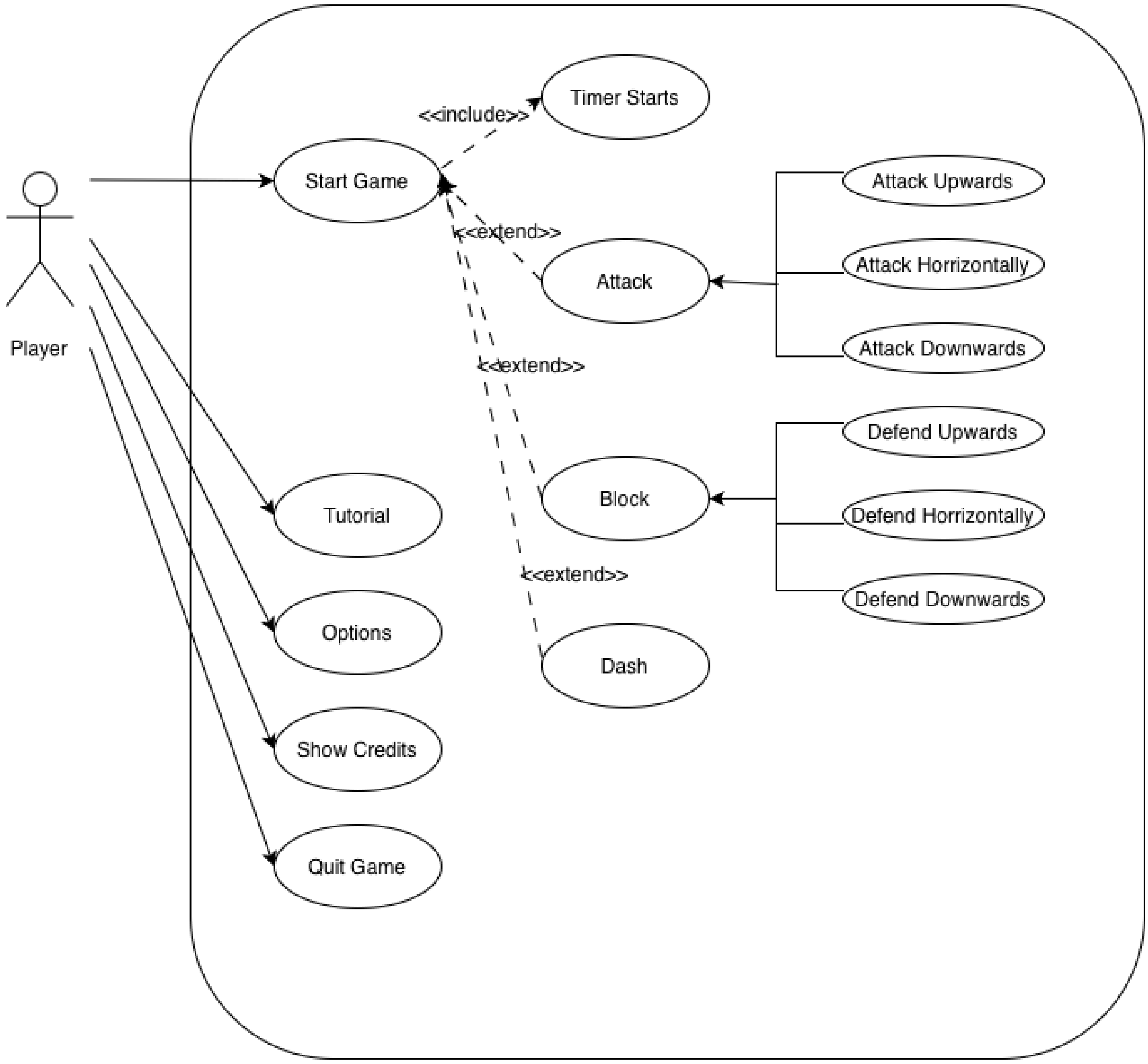
4.5 Compatibility Requirements

- **NFR-14:** The game shall be compatible with the **Windows operating system**.
- **NFR-15:** The game shall work properly on both **laptops and desktop computers** with standard setups.
- **NFR-16:** All **menus and UI components** shall display correctly on different aspect ratios (**16:9**, **16:10**).

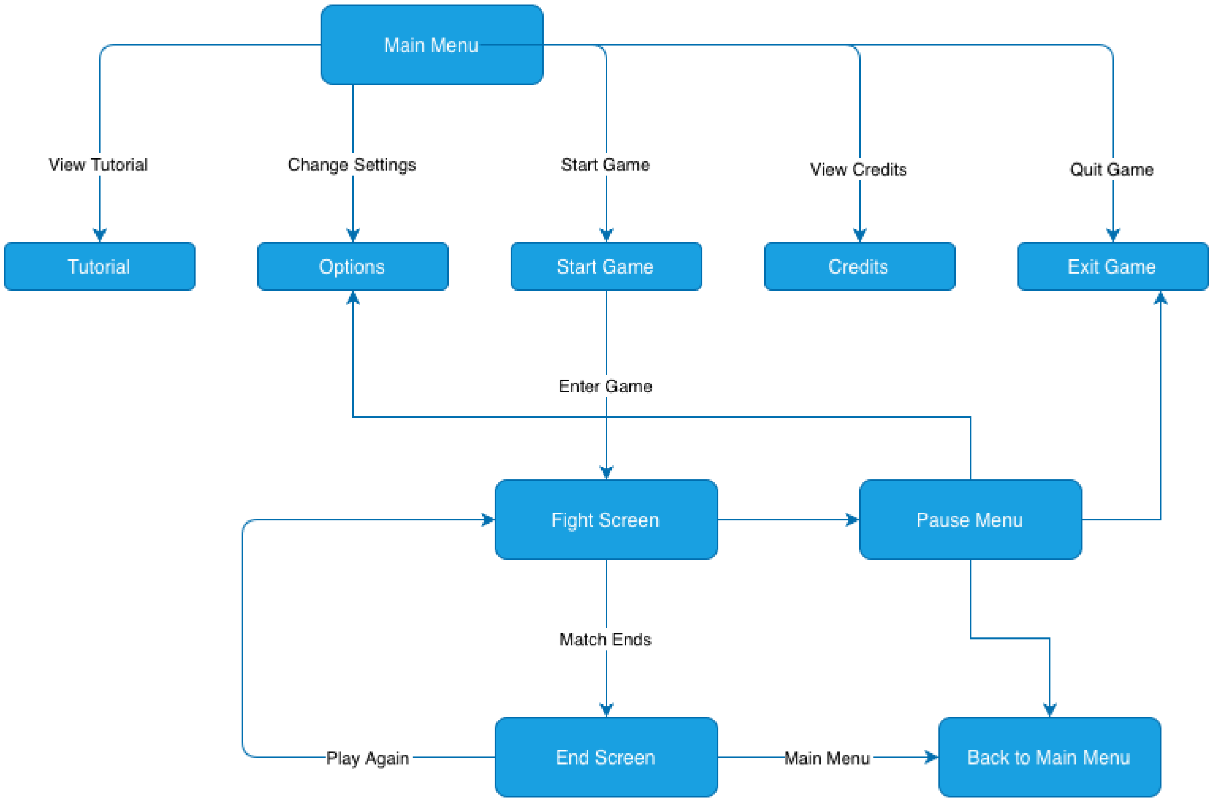
---

5. SYSTEM MODELS (DIAGRAMS)

5.1 Use Case Diagram



5.2 User Flow Diagram



6. DEVELOPMENT DETAILS

6.1 Development Setup

Project Manager and Core Systems Engineer:

Oğuzhan Çelik

Gameplay Engineer:

Yunus Emre Yılmaz

Menu and UI Engineer:

Tuğana Öykü Yıldız

6.2 GitHub Links

GitHub Repository Link:

https://github.com/OguzhanCel/YZM2021-Project

GitHub Project Board Link:

github.com/users/OguzhanCel/projects/3