

# Supplementary Specification

## Introduction

This document captures non-functional requirements, constraints, and supplementary features for the Rokue-Like Game project. These requirements define the overall quality attributes and design constraints for the system.

## Usability

### User Interface Design:

- Intuitive UI with minimal complexity for players of all experience levels.
- Real-time visual indicators for:
  - Remaining time (timer display).
  - Player lives.
  - Collected enchantments and their effects.

### Player Feedback:

- Sound and visual cues for important actions, such as:
  - Finding a rune.
  - Successfully using an enchantment.
  - Being hit by a monster.

### Accessibility:

- Ensure text and visuals are clear and easy to read, even on smaller screens.
- Avoid colour combinations that may be problematic for colourblind users.

## Reliability

### Recoverability:

- In the event of a crash or unexpected shutdown, allow the game to restore progress from the last hall completed.

### Fault Tolerance:

- Handle errors such as failed resource loading or invalid user input without crashing the game.

## Performance

### Loading Times:

- Ensure hall transitions and Build Mode load times are under 3 seconds on supported hardware.

### Real-Time Responsiveness:

- Maintain frame rates of at least 30 FPS during gameplay to ensure smooth player interactions.

### Scalability:

- Support generation of halls with increasing complexity without significant performance degradation.

## Supportability

### Extensibility:

- Modular design to allow future expansions, such as:
  - New hall themes.
  - Additional enchantments or monsters.
  - Multiplayer functionality.

### Configuration:

- Allow customisation of Build Mode parameters (e.g., maximum and minimum objects per hall).

### Compatibility:

- Ensure the game runs smoothly on all desktop platforms, with minimal differences in user experience.

## **Implementation Constraints**

### **Technology Stack:**

- Use Java as the primary programming language.
- Use Java Swing as the GUI library.

### **Hardware Requirements:**

- Minimal hardware is enough.

## **Interfaces**

### **Player Interface:**

- Provide a main menu with the following options:
  - Start Game.
  - View Help Screen.
  - Exit Game.
- In-game interface must display:
  - Timer and life counters.
  - Collected enchantments.
  - Pause and return main menu buttons

## **Application-Specific Domain Rules**

### **Rune Collection Rules:**

- Each hall must contain exactly one rune, hidden within an interactive object.
- The player cannot exit a hall without first collecting the rune.
- Rune can be teleported away to another object by a specific monster.

### **Monster Rules:**

- Monsters must behave consistently based on their types:
  - Archer Monsters shoot at the player if within range.
  - Fighter Monsters pursue the player when nearby.
  - Wizard Monsters teleport runes periodically.

### **Enchantment Rules:**

- Storable enchantments must be usable at any point after collection.
- Immediate enchantments (e.g., Extra Time, Extra Life) take effect as soon as collected.