Software Requirements Specification (SRS)

by TophUwO

June 2024

Project **Noriko**

Component **Noriko**

Noriko is a cross-platform 2-D RPG game engine, focused on simplicity, ease of use, and scalability.

Table of Contents

Version History

The following table logs all changes made to this document during the analysis phase of the project. Author names may hold hyperlinks to the author's preferred way of contact. Dates are given in ISO-8601 **`MM-DD-YYY`** format. Version numbering is as follows: **`MAJOR.MINOR.PATCH-ITERATION`**. Version **`1.0.0`** marks the release version of this document according to which the software will be developed.

DATE	VERSION	CONTRIBUTORS	CHANGES
06-16-24	0.0.1-1	TophUwO	**INITIAL COMMIT** **ADD** table of contents
06-16-24	0.0.1-2	TophUwO	**ADD** version history **DO** accent color setup
06-16-24	0.0.1-3	TophUwO	**ADD** remaining major sections **FIX** formatting of tables
06-17-24	0.0.2-1	TophUwO	**ADD** initial content for all sub-sections within section *1. Introduction*
06-17-24	0.0.2-2	TophUwO	**ADD** text contents for section *2.1. Product Perspective and System Context*

1. Introduction

The following section provides basic information on not only the product described by this document, but also the document itself, such as document conventions, structure, and various pointers for readers.

1.1. Purpose

This document is a Software Requirements Specification (SRS) for the component called **`Noriko`** of project **`Noriko`**. Project Noriko is a cross-platform 2-D role-playing game (RPG) game engine. Other documents of this type exist for components **`NorikoEd`** and **`NorikoRt`**. Throughout the remainder of this document, the term **`Noriko`** will refer to the game engine component only.

1.2. Document Conventions

The document at its core closely adheres to IEEE 830: Software Requirements Specification. Additional sections providing miscellaneous information may exist, especially in the latter half of this document. The document's language is English (United States). Terms which are **`highlighted like this`** may be either technical terms, variables, or other information of special significance. In the Version History table, terms in **`CAPITAL BOLDFACE`** signify commit message actions, whereas terms in *`underlined highlighted italics`* are links to other sections within this document. When external links are inserted, a superscript `(»)` is appended to the link texts, whereas e-mail or other contact links are annotated via a superscript `(@)`.

1.3. Intended Audience and Reading Suggestions

The document is to be understood by all stakeholders. That includes project managers, designers, software engineers, and programmers. What sections are most important depends on the individual's function in the project. While project managers should be familiar with the entire document, software engineers and designers should focus on the latter half of this document which delves into low-level details of the system.

1.4. Product Scope

Games have seen a rapid increase in popularity in recent decades. With a milliard of genres to choose from, effortlessly crossing country-, demographics-, as well as

cultural boundaries, games have become a vital part of today's entertainment industry. Whereas today's focus of AAA studios is mostly on realistic 3-D worlds, there was a time not long ago when computers and especially consoles lacked the hardware power to simulate scenery this complex. In the 1990s and early 2000s, handheld consoles became increasingly prevalent, with the Nintendo® Game Boy® having been introduced as far back as 1989. Due to the stringent hardware limitations of early-generation handheld devices, games had to not only heavily optimize game- and drawing routines, but also had to place strict requirements on complexity as well as code size.

Having grown up in the early 2000s, I came in contact with handheld consoles, with the Nintendo® Game Boy Color® being my first one. I quickly developed a deep interest in RPG games which were virtually always set in tile-based worlds due to the aforementioned limitations. This project, Noriko, aims to recapture the iconic charm that managed and still manages to captivate millions of kids across the globe by focusing on what made these games special, with the objective of making development of such games as easy, accessible, efficient, and scalable as possible.

1.5. References

The following table contains resources (web pages, books, articles, ...) to key topics and technologies mentioned throughout this document. They may serve as an introductory pointer to the reader; however, sticking to them is not required. Additional research is advised.

TERM	TYPE	DATE	INFO

1.6. Applications and Tools Used

This section lists all tools and external applications used for creating this document, grouped by scope.

SCOPE	TOOL(S) USED
LAYOUT & TYPESETTING	Microsoft® Word® 365 (Version: 2405)

2. General Description

This section focuses on key aspects of the system, such as high-level features, user-, and system context analysis.

2.1. Product Perspective and System Context

All three main components of Noriko live in the same ecosystem and are designed to depend on one another in order to function properly. The ecosystem is considered to be largely self-sufficient; dependencies from outside the ecosystem are non-existent or minimal.

The following system context diagram visualizes the relationships between Noriko's three main components.

2.2. Product Functions

2.3. User Classes and Characteristics

2.4. Operating Environment

2.5. Design and Implementation Constraints

2.6. User Documentation

2.7. Assumptions and Dependencies

3. External Interface Requirements

4. System Features

5. Non-functional Requirements

6. Other Requirements

7. Optional Requirements

8. Future Requirements and Plans

Appendix A: Glossary

Appendix B: Supplementary Resources

Appendix C: Original User Story

Appendix D: Analysis Models

Appendix E: TBD