Software Requirements Specification

For Stephanie Li

Version 1.0

June 20, 2025

For Software Development

VCE Units 3 & 4

Prepared by Andrew Nguyen

NGU0193@balwynhs.vic.edu.au

Under the supervision of Mr. Toet

RTO@balwynhs.vic.edu.au

Table of Contents

Section 1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description, as well as lay out the functional and non-functional requirements of the rhythm game software that is to be created for Software Development Units 3/4. This document will explain the system's purpose, features, interfaces, functional and non-functional requirements, and constraints. This document is intended for the developers and proposed users of the system and will be presented to the proposed users for approval.

Section 2. Product Overview

2.1 Product Scope

The product will be a desktop-based rhythm game developed in the Unity game engine, based on the Japanese arcade rhythm game *Taiko no Tatsujin* and *osu!taiko*. The product aims to unify the two major file formats in the Taiko rhythm game community, TJA (used by *Taiko no Tatsujin* simulators) and .osz (used by *osu!taiko*). The product will allow users to select, manage and play songs (Charts) loaded from an external file (both TJA and osu! formats), view their scores and past gameplay statistics saved to their local machine, and customise their video, audio, gameplay and input preferences, allowing the product to cater to the needs of and remain accessible to individual users. The product will be designed to be responsive, customisable, and visually engaging. The product will emphasise input accuracy, and implement performance tracking, local leaderboards, and user interface customization. The product aims to address file format incompatibility, input lag, and gameplay stability issues found in existing rhythm games, and will emphasise crucial elements for competitive gaming, including low input latency and audio latency and fast loading times. The ultimate goal for the product is to deliver a modern, community-friendly, and stable alternative to existing platforms. Due to time constraints and the developer's inability to extensively test compatibility on other platforms, the product will be built exclusively for the Windows operating system.

2.2 Product Purpose

The software solution is being developed as an original product built to address long-standing usability and performance issues observed in existing Taiko rhythm game simulators. While the product will draw mechanical and aesthetic inspiration from *Taiko no Tatsujin* and *osu!taiko*, it is not affiliated with any existing simulator or platform and is built as a standalone product with an emphasis on stability and format compatibility. Existing platforms often suffer from audio and input latency, poor stability and outdated and unfriendly user interfaces. The software solution aims to be a unified, stable, performant and user-friendly alternative to existing systems.

The product seeks to bridge the gap between two existing Taiko "beatmap" ecosystems: the TJA format used in Taiko no Tatsujin simulators (closely emulating the

proprietary format by Bandai Namco) and the .osz format native to osu!taiko. The product's purpose is to allow users to easily and seamlessly import, play, and manage maps from both formats while experiencing stable gameplay that meets competitive rhythm gaming standards, that is, low input and audio latency, high FPS, smooth visuals, etc.

2.3 Product Constraints

The product is being developed within the Unity game engine due to its robust cross-platform capabilities, extensive documentation, and familiarity to the developer. However, this introduces certain limitations, including dependency on Unity's rendering pipeline, input system architecture, and scene management. The use of Unity also constrains the project to C# as the primary programming language, which may limit access to certain native libraries or low-level system optimizations that could otherwise reduce latency or improve performance. An additional legal constraint is Unity's licensing and distribution policies, including restrictions under the Personal license tier which the developer is using for non-commercial educational purposes.

Hardware and platform compatibility are additional constraints. The game must be fully operational on Windows systems without requiring administrative privileges for installation or execution. This means things such as low-level input hooks, drivers, or privileged access to hardware peripherals (such as low-level drum controller drivers) cannot be used.

Any file operations must be sandboxed within accessible directories to comply with any potential device restrictions that could be present, such as on a school-managed/provided device. This means restricted directories such as AppData on Windows may not be used. As users may run the game on low-spec machines with limited processing power and integrated graphics (e.g., school-provided machines/laptops), the software must be optimized to reduce CPU and memory consumption without compromising gameplay visuals or responsiveness.

Time and resource constraints are another constraint on the project. As this is a solo project that will be developed alongside additional academic commitments (Media Units 3 & 4 is of special concern), development time is limited to a fixed calendar with key progress, delivery and assessment checkpoints. Therefore, features like multiplayer support, online leaderboards, or extensive custom skinning must be excluded from the scope for the initial

release. These constraints will guide technical decisions throughout the development process to ensure a feasible and high quality delivery within the project's time frame.

2.4 User Characteristics and Audience

The product is being designed with existing members of the rhythm game community in mind with a diverse range of skill levels, however it will also cater to those who are currently inexperienced with rhythm games, but they will constitute a minority. Therefore, the majority of the users will have some prior rhythm game experience. These users seek software stability, audio and input precision, aesthetics, customisability and smooth performance, which have already been and will continue to be very strong considerations of the software solution. Most of these users are between the ages of 15 and 30, many are fluent in certain rhythm game-specific terminology, and familiar with many individual performance settings for optimal gameplay. It is important that the software solution therefore implements many settings found in other rhythm games.

Newer users may not have deeper technical knowledge and often play with the default pre-configured settings. Therefore, it is important to ensure that the settings, by default, should represent the best possible user experience, which the user can then change as they see fit. For newer users, clean navigation, ease-of-use and visual polish must be kept as high priorities. The user's technical ability should have no affect on their ability to use the software solution and play the game effectively.
Across all user classes, there will be individuals with accessibility needs, including those with motor impairments or visual sensitivities. These users may require customizable color themes, the ability to toggle or reduce animations, adjustable difficulty scaling, or remappable controls to accommodate alternative input methods.

2.5 Technical Environment

The solution will be designed to run efficiently and stably on a wide range of hardware with varying specifications, including lower-end laptops and high-performance PCs. The solution will be developed on the Unity game engine (version 2021.3.39f1), as specified in the design brief.

For Windows machines, the minimum hardware requirements include a system running Windows 10 or newer. The CPU must support x86 or x64 architectures with SSE2 instruction set support, as well as Arm or Arm64 systems, and the graphics card must be capable of running DirectX 10/11/12, or alternatively, Vulkan. All GPU drivers must be officially licensed/supported by the hardware vendor to ensure system compatibility, as well as system stability.

For MacOS machines, the minimum requirements include a system running Big Sur 11 or newer, running on Apple Silicon or x64 architecture with SSE2 instruction set compatibility. Graphics rendering depends on a Metal-capable GPU, which includes Intel and AMD GPUs officially supported by Apple.

The software solution does not have any network requirements, as it does not store leaderboards on an external server and instead on the local machine. There are also no integrations with external APIs, but the solution does support the file formats of other rhythm game software.

Section 3. Requirements

3.1 Functional

FR-1: The system shall support a rhythm-based input system that includes hit detection for two types of drum notes (don and ka), with timing-based scoring.
FR-2: The system shall display a real-time scrolling note chart synchronized to background music.
FR-3: The system shall track and display the player's score, accuracy, combo, and progress during gameplay.
FR-4: The system shall support a version of Go-Go Time (high-energy visual and scoring boost segment), with toggleable visual effects.
FR-5: The system shall support gameplay using both keyboard and external controllers.
FR-6: The system shall support customizable keybindings for all input actions.
FR-7: The system shall provide user interface customization, including:

- Custom color palettes,
- Toggleable visual effects (e.g., background dim, animations, sparkles).

FR-8: The system shall allow the user to adjust audio settings, including the audio offset and volume levels.
FR-9: The system shall include a searchable chart (song) selection screen.
FR-10: The system shall allow sorting by attributes such as song title, artist name, difficulty, length, and date added.
FR-11: The system shall display detailed metadata for each chart, including the song title, artist, song length, total note count, difficulty, and preview audio.
FR-12: The system shall maintain local leaderboards.
FR-13: The system shall support .osz and TJA files.
FR-14: The system shall allow users to disable flashing lights.

3.2 Non-Functional

3.2.1 Performance

NFR-01: The system shall maintain a stable framerate of at least 60 FPS during gameplay on standard desktop configurations
NFR-02: The system shall import songs in less than seven (7) seconds.
NFR-03: The average time to load into gameplay from the song selection screen shall not exceed 5 seconds on a system meeting minimum specs.

3.2.2 Usability

NFR-04: The user interface shall be designed such that core functions (play, chart select, settings) can be accessed within 3 or fewer clicks from each other.
NFR-05: The system shall support keyboard, mouse, touchscreen, and controller inputs with full UI operability.
NFR-06: The text in the game shall have a minimum contrast ratio of 4.5:1.
NFR-07: The software shall have adequate accommodations for users with accessibility needs.

3.2.3 Reliability

NFR-08: Upon encountering a corrupted chart or file, the system shall fail gracefully by skipping the content and notifying the user.

NFR-09: The system shall have a mean time between failures (MTBF) of at least one (1) hour of continuous use under normal conditions.

3.2.4 Maintainability

NFR-10: The codebase shall have extensive internal documentation for future extensibility.
NFR-11: The codebase shall follow clean architecture principles, allowing the addition or replacement of modules (e.g., input, skins, score system) with minimal coupling.

3.2.5 Portability

NFR-12: The system shall be compatible with Windows 10/11 and ideally support Android/iOS for future mobile ports.
NFR-13: The application shall be resolution-independent, functioning correctly on displays ranging from 720p (mobile/tablet) to 4K monitors.
NFR-14: All user data (scores, settings, skins) shall be stored in a portable format and support easy backup and migration between devices.
NFR-15: Date/time format shall be automatically adapted based on local machine settings.