

SP-22 – Dynamic Rhythm Game
Software Requirements Specification
CS 4850 – Section 01 – Fall 2023
Professor Sharon Perry
Sept 18, 2023



Katlin Scott



Willie Doyle



Bethany Cadman



Angel Magallanes

Table of Contents

1.0 Introduction.....	3
1.1 Overview.....	3
1.2 Project Goals.....	3
1.3 Definitions and Acronyms	4
1.4 Assumptions.....	4
2.0 Functional Requirements	5
3.0 Non-Functional Requirements	6
3.1 Accessibility.....	6
3.2 Capacity & Scalability	7
3.3 Usability	7
4.0 External Interface Requirements.....	7
4.1 User Interface Requirements.....	7
4.2 Hardware Interface Requirements	7
4.3 Software Interface Requirements.....	7
4.4 Communication Interface Requirements	7
Appendix A: Glossary.....	8

1.0 Introduction

1.1 Overview

Have you ever listened to a song and wished you could play that particular song in a rhythm game? Or maybe even wanted more of your own songs in a rhythm game without having to wait for a new DLC pack to be released?

Well, our dynamic rhythm game will create procedurally generated ‘rhythm game levels’ (also known as ‘maps’) as songs are selected. We plan on achieving this by utilizing AI to detect various beats and inflections within the music. The music will be sourced from the largest music library in the world, Spotify, giving us access to over one hundred million songs and near-endless possibilities.

1.2 Project Goals

In our project, we have added an additional phase beyond the two. We have defined each phase below.

- Phase 1 is absolutely necessary for our game.
- Phase 2 is quality of life aspects that would improve the game’s performance massively.
- Phase 3 is absolutely optional and not required for the game to function.

	Goal	Description
Phase 1	Base game platform	The base interface
	API Spotify implementation	Having Spotify connectivity with songs
	User interaction design	To be able to have the user be able to interact with the stage and beats with game input (“Perfect!” “Super!” “Miss,” etc.) (Combo tracking)
	AI development to detect beats	To make the AI detect beats
	Playable game prototype	
Phase 2	Optimization	Making the game run smoother
	Design review and rework	Relooking at the game
	Tutorial	Will have an in-game interactable

		tutorial or a video in the game
Phase 3	Storage of (pre-)generated songs	Stores the AI-generated songs (or other) in a database or file
	Beatmap editor	
	Auto-play feature	Have maps get played through automatically by the computer
	Possible in-game story	
	Mobile Application Development Build	

1.3 Definitions and Acronyms

Beatmap* / Notechart = The full progression of notes to be pressed on a single song

DRG = Dynamic Rhythm Game

Lane = a column notes are separated into, usually connected to a specific button

Note = a basic single step in a notechart/beatmap

Player - a person who plays any type of game, whether it be video games, board games, role-playing games, etc.

Spotify = digital music/audio streaming and media service provider

Stage = the area where the notes move, and the player interacts with said notes

* = Term is preferred in this document

1.4 Assumptions

We are assuming that the system that will be used to run the game has the sufficient specifications that will be needed for the game to properly run on the proper operating system.

2.0 Functional Requirements

1.0 Display Main Menu

- 1.1 Play Button redirects to beatmap selector screen
- 1.2 Beatmap Creator redirects to beatmap creation screen
- 1.3 Settings Button redirects to settings screen
- 1.4 Quit Button exits the game

2.0 Beatmap Selector Screen

- 2.1 List of beatmaps displayed
- 2.2 Search bar and category filter/sort
- 2.3 Back Button redirects to main menu

3.0 Stage

- 3.1 User interaction with notes
- 3.2 Feedback based on the success of hitting notes
- 3.3 Results screen

4.0 Beatmap Creation

- 4.1 Adding a song from the Spotify library
- 4.2 Selecting a difficulty

5.0 Settings Screen

- 5.1 Customize Settings
 - 5.1.1 Visuals
 - 5.1.2 Audio/Volume Controls
 - 5.1.3 Controls Customization
 - 5.1.4 Accessibility
 - 5.1.4.1 Color Blindness Options
 - 5.1.4.2 Game Subtitles

- 5.2 Timing Window Adjustment

6.0 Pause Screen

- 6.1 Resume
- 6.2 Restart the level

6.3 Options/Settings

6.4 Return to Song Selection

6.5 Return to Main Menu

According to Unity, the user's PC/Mac graphic and operating system requirements require the user to have:

MINIMUM:

OS: Windows 7

Processor: Dual Core 2.00GHz

Memory: 2 GB RAM

Graphics: Intel HD 520

Storage: 100 MB available space

RECOMMENDED:

OS: Windows 10

Processor: i5

Memory: 8 GB RAM

Graphics: GeForce GTX 660

Storage: 100 MB available space

3.0 Non-Functional Requirements

3.1 Accessibility

The DRG will place emphasis on having options in the game to improve the quality of gameplay a player has while playing the game.

Some accessibility features that will be included are:

- Stage color preference setting (lighter colors, darker colors, neutral colors, pastel colors, etc.)
- Alternative coloring for colorblindness (for deuteranomaly, protanomaly, protanopia, deuteranopia, tritanomaly, tritanopia)
- Subtitles
- Dynamic (according to settings) button labels for the lanes
- Stage/Lane color variations/skins

3.2 Capacity & Scalability

The game should have the availability for storage expansion in case Phase 2 is completed. Each beatmap should take a minimal amount of space but should also allow for songs of different lengths and sizes. The game should also allow for a virtually infinite number of songs in its' storage, only limited by the user's system.

3.3 Usability

The game will be easy to learn, engaging, efficient, effective, and error tolerant. The player does not need to have prior experience with playing rhythm games to play our rhythm game. The game will be simple enough to be able to learn through our game, having sound cues when the player hits a note. Along with that, the plan is to make it pleasant and satisfying to play through the responsiveness and accuracy of the game and the capability to configure settings.

4.0 External Interface Requirements

4.1 User Interface Requirements

The fonts and icons should be similar throughout the final product. The controls that are most commonly used should stay consistent throughout the final product. The color schemes should avoid visually jarring combinations. There should be accommodations for the visually impaired (coloring for colorblindness). Certain shortcut keys for specific functions (Ex: Esc for pausing)

4.2 Hardware Interface Requirements

Supported device types are PC/Mac

4.3 Software Interface Requirements

This product will make use of the Spotify API.

4.4 Communication Interface Requirements

This product will require an internet connection.

Appendix A: Glossary

AI = Artificial Intelligence

Artificial intelligence uses computers and machines to mimic the problem-solving and decision-making capabilities of the human mind. To read more:

<https://www.ibm.com/topics/artificial-intelligence#:~:text=Artificial%20intelligence%20leverages%20computers%20and,capabilities%20of%20the%20human%20mind>

Algorithm

A procedure used for solving a problem. Algorithms act as a list of instructions that help to conduct specified actions step by step in either hardware or software-based routines. To read more: <https://www.techtarget.com/whatis/definition/algorithm>

PC = Personal Computer

A PC is a computing device that utilizes a microprocessor and is typically designed for use by one person. To read more: <https://computer.howstuffworks.com/pc.htm>