

SP-22 – Dynamic Rhythm Game
Software Design Document
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 Purpose.....	3
1.3 Definitions and Acronyms	3
2.0 General Overview	4
2.1 General Information.....	4
2.2 Assumptions.....	4
2.3 Constraints	4
2.3.1 Environment.....	4
2.3.2 User Characteristics	4
3.0 Design Considerations	5
3.1 Goals and Guidelines	5
5.0 System Design	6
5.1 UX Process Flow Diagram	6
5.2 User Interface Design Diagrams	6
5.2.1 Main Menu Screen.....	6
5.2.2 Song Play Screen	7
5.2.3 Pause Menu.....	8
Appendix A: Glossary.....	9

1.0 Introduction

1.1 Overview

The Dynamic Rhythm Game will utilize Unity to help create an environment that is easy for the user to navigate along with having all the functionality requirements necessary to make the game run smoothly. We will create a “Home” (or start/beginning) screen that will have options for playing the game, opening the settings, and an exit button. After selecting the “Play” option, the user will be navigated to the “Song Selection” screen, which will allow the user to either select a pre-downloaded/uploaded song or upload new music to generate a new beatmap. After selecting a song, the user wishes to play, they will be navigated to the “Game” screen where they will play the beatmap, and a score will be established based on how well the user hits the notes generated on cue. After the gameplay, they will see a summary screen that will showcase their song stats and have a button that will redirect them back to the song selection menu. If the user selects “Settings” at any point - from either the “Home” or “Song Selection” Screen - they will be redirected to the Settings screen/overlay where they can customize their experience with a variety of things from sound to stage color. We plan to add more design elements as we progress, but with what we have outlined above we believe we will have a fully functioning rhythm game.

1.2 Purpose

The purpose of this design document is to assist us with planning and decision-making as we create our rhythm game throughout the semester. It outlines what our game will include and how we will build it as well as visually assist as to what the UI will look like and how it will function. It will also ensure that we stay on track while creating the game.

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

2.0 General Overview

2.1 General Information

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 100 million songs and near-endless possibilities.

2.2 Assumptions

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

2.3 Constraints

2.3.1 Environment

Unity allows for a wide range of applications to be created and many possibilities, but there are some tradeoffs. Platform compatibility is something to keep in mind. If we decide to have cross-platform compatibility (Windows, macOS, Android, etc.), we need to ensure it works correctly and efficiently on each of them. Performance is another constraint. We need to ensure efficiency and playability on a wide range of systems. That includes a framerate that is both smooth and consistent and has efficient memory usage.

2.3.2 User Characteristics

User accessibility considerations for rhythm games are essential to ensure that players with different abilities can enjoy the game. This includes providing comprehensive tutorials and practice modes to help players learn the game mechanics and improve their skills at their own pace, along with different difficulty levels for a wide range of players. Other considerations are:

- Customizable controls - Provide options for players to customize controls, including keybindings, button layouts, and game speed.
- Adjustable Timing Windows - Allow players to adjust the timing windows for hitting notes. This feature can accommodate players with different reaction times, motor skills, and habits.
- Colorblind Modes and Inclusive Design - Implement colorblind-friendly modes that use color combinations suitable for players with vision deficiencies, along with an interface easy for everyone to read.
- Warnings - Photosensitivity warnings for those with epileptic disabilities

3.0 Design Considerations

3.1 Goals and Guidelines

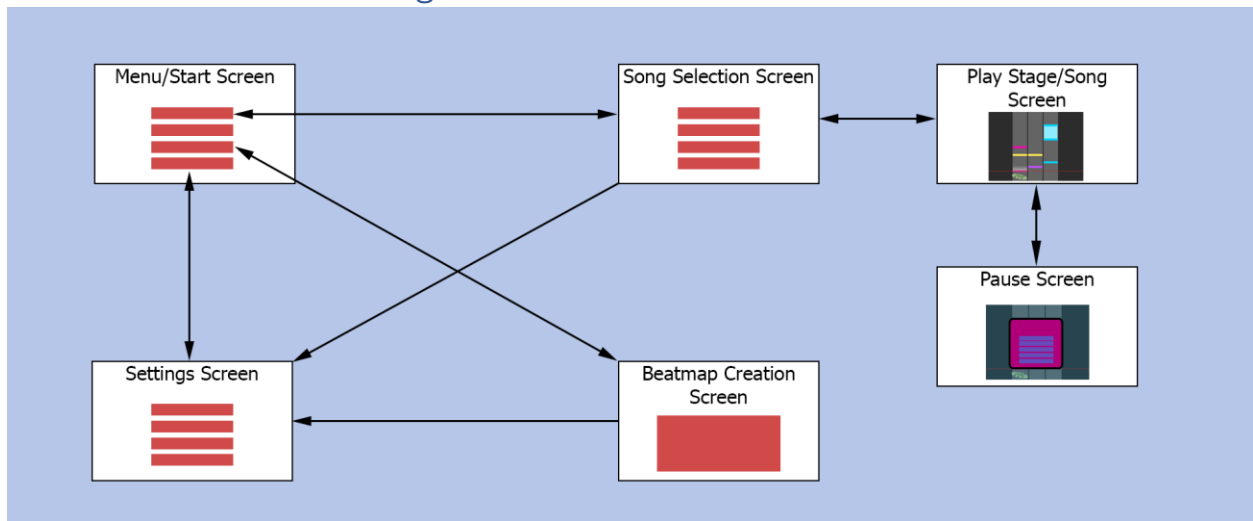
In our project, we have added an additional phase beyond the two. We have defined each phase below.

- 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.)
	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	

5.0 System Design

5.1 UX Process Flow Diagram

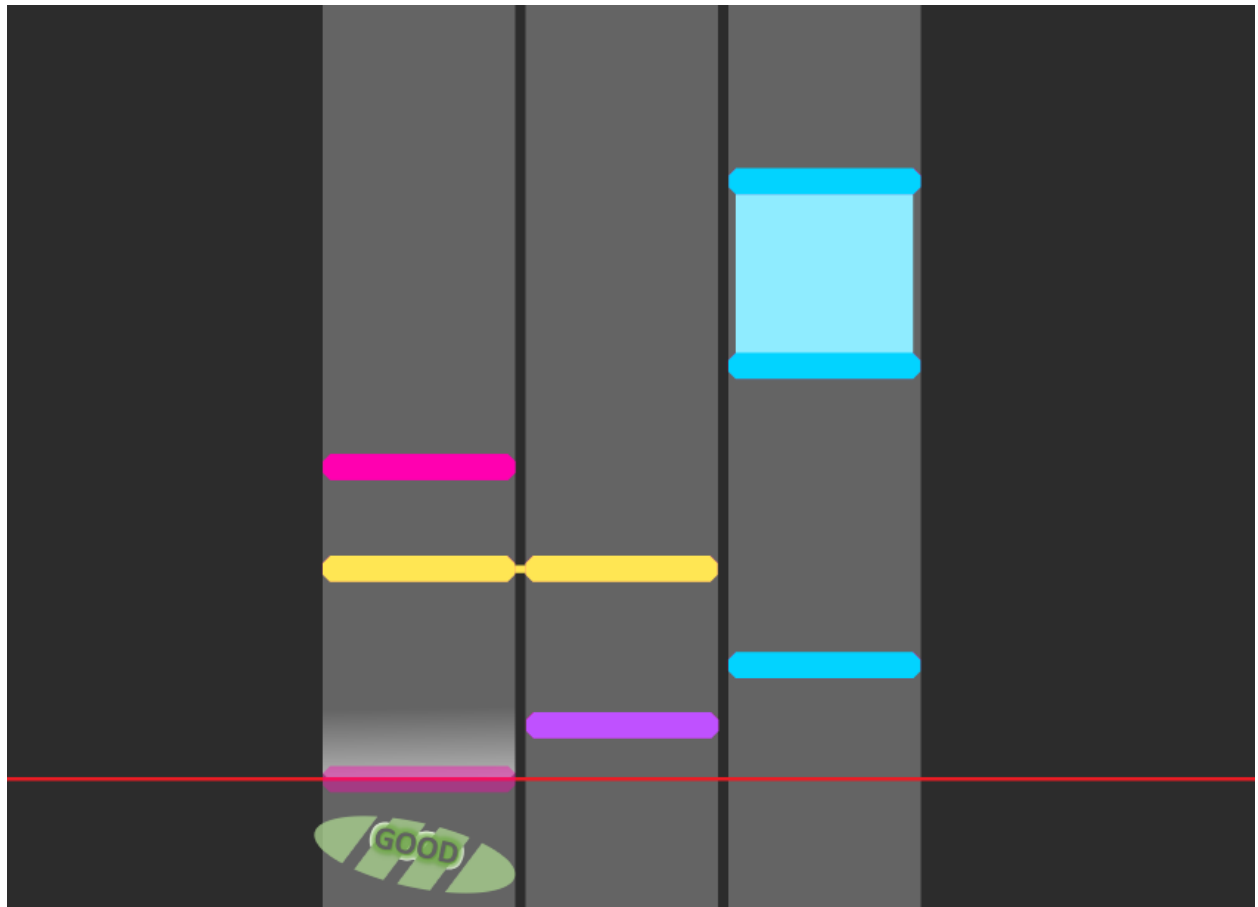


5.2 User Interface Design Diagrams

5.2.1 Main Menu Screen



5.2.2 Song Play Screen



5.2.3 Pause Menu



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>