DevoBeats: Danny Mok, Colyi Chen, Kevin Lin
SoftDev
P05:
2025-05-07
Time Spent: 2
TARGET SHIP DATE: 2025-06-09

DESIGN DOCUMENT (Version 1)

I. Description

This project is a rhythm game inspired by Piano Tiles. It will have a settings page, log-in functionality to keep track of scores on a leaderboard, a song selection page and the actual game page itself. The game page will consist of 4 lanes, each with tiles that scroll down at a set speed. Upon reaching a threshold point, the user can click or use a corresponding keybind to tap the tile and gain points. The accuracy of the timing will be taken into consideration as well, and failing to tap a tile in time will lead to penalties. When the song is completed or the user fails to complete the song, their score is compared to their high score and the leaderboard displaying the scores of all users is updated.

A. Program Components

a. User Accounts:
    i. Creation of accounts and login/logout functionality
    ii. Sessions
b. Node.js backend
    i. Needed for real time responsive inputs to visual-audio cues
c. Node.js middleware
    i. Needed for real time responsive inputs to visual-audio cues
d. SQLite3 Database:
    i. Stores user login info
    ii. Stores user scores
e. Jinja Templates
    i. Landing Page
        1. Flashy visuals and easy navigation
        2. Play button
        3. Login/Signup options
    ii. Leaderboard Page
        1. Displays leadership by points
    iii. Song Selection Page
        1. Select pre-existing song

2. Select pre-mapped song (optional)

iv. Game Page

1. Tile scrolling and keyboard/click based gameplay

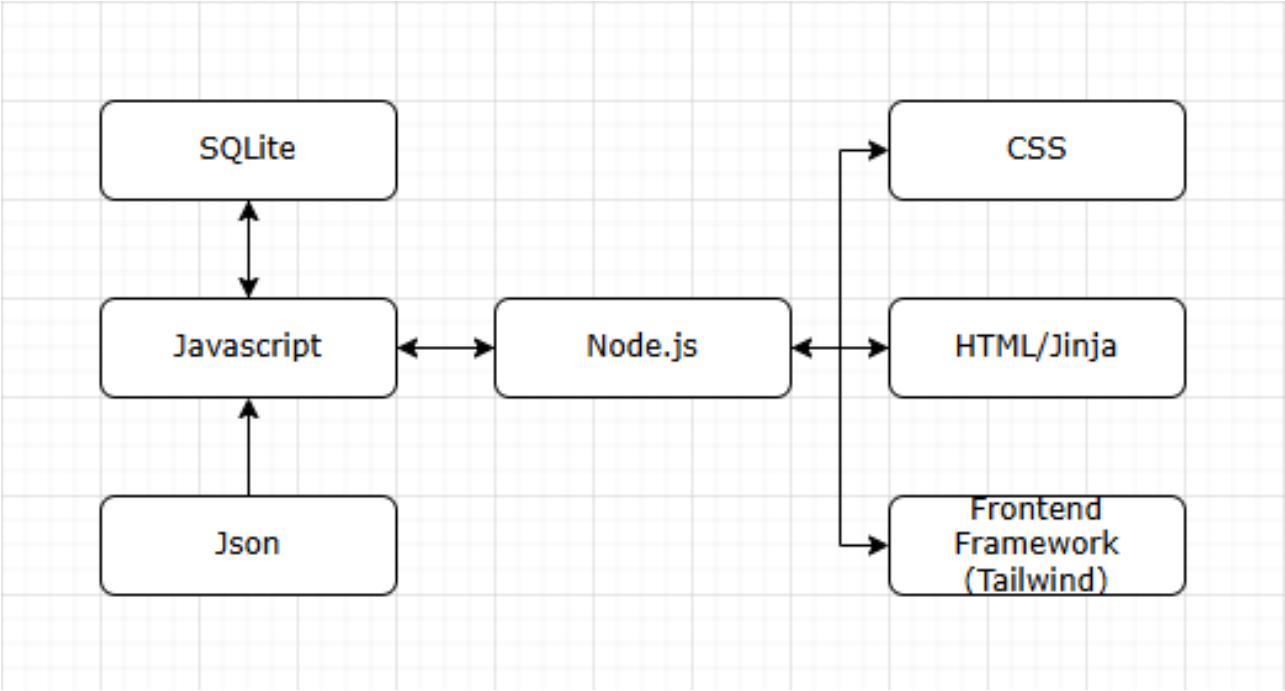
v. Settings

1. Change gameplay settings
    a. Keybinds
    b. Song volume (optional)
    c. Website style changes (optional)
    d. Song speed (optional)
    e. Note style (optional)

f. Javascript

g. Json files (Optional depending on implementation)

i. Storing beat maps



B. Site Map + Descriptions

a. Landing page/home (/):

i. Redirects to user login/logout/registration, leaderboard and the song selection page

b. Login (/login):

i. User login functionality

c. Registration Page (/register):

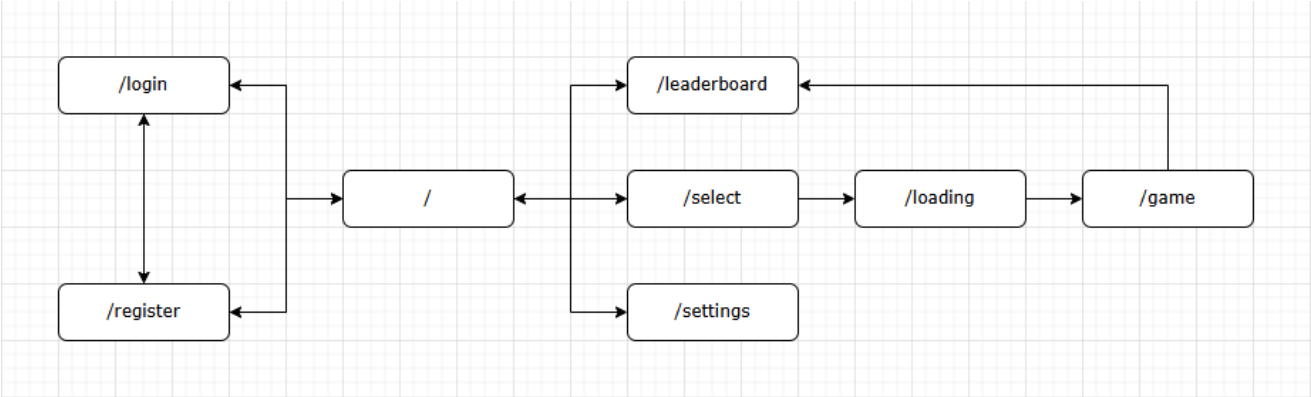
i. User registration functionality

d. Game Page (/game):

i. Tiles scroll down along one of the four paths

ii. Threshold line to gauge the timing of the keypresses to the oncoming tiles
iii. Keybind indicators
iv. Score indicators
v. Song progression bar
vi. Song audio from Spotify API (or other alternatives)
vii. Song distortion upon mistiming a note (optional)

e. Song Selection Page (/select):
   i. Browse through Spotify API playlists (or alternatives)
      1. These songs will have procedurally generated notes. We will analyze the beat patterns in the song with libraries capable of processing audio through FFT algorithms. This will provide the rough note timings, which will then be put into an array. If possible, we will then group notes with particular timing into common rhythm game note patterns for better gameplay. If not feasible, then the beats will just be randomly selected to go down one of the four lanes/
   ii. Browse through pre-mapped songs
      1. Pre-mapped songs are songs with manually placed notes by the developers
f. Settings Page (/settings):
   i. Change keybinds of the game
   ii. Change song volume (optional)
   iii. Change website styling (optional)
   iv. Song speed settings (optional)
   v. Change color of tiles (optional)
g. Leaderboard Page (/leaderboard):
   i. Displays usernames and their high scores from highest to lowest
h. Loading Screen (/loading):
   i. Implemented for variable wait times (we're aiming for under 30 seconds)



C. Database Organization

a. User Table

id	username	password	highscore
INTEGER PRIMARY KEY AUTOINCREMENT	TEXT UNIQUE NOT NULL	TEXT NOT NULL	INTEGER NOT NULL

D. Frontend Framework

Tailwind

We chose tailwind because of its ease of use and since our CSS lead has experience with it. Our primary focus is on functionality before style, and tailwind provides a fast but powerful way to stylize our website.

E. APIs

Spotify API (Subject to change)

F. Task Breakdown

(Very likely to change)

Danny Mok:

- Project Manager
- Javascript lead
- HTML and CSS + Tailwind

Colyi Chen:

- Undetermined

Kevin Lin:

- Database engineer
- Backend
- Middleware