Software Requirements Specification (SRS)
BPM guessing web-based game

Team: Group 3
Authors: Ali, Luiz, Masha, Blad, Kriston, Armando
Customer: SWE I
Instructor: Dr. Daly

Introduction

This SRS document will detail the processes that will be the guiding principals for our development of a BPM guessing game. The topics that will be covered here vary from what classes will perform what tasks and how they are related and how the users will be able to interact with the software(s) produced. We will also cover realistic expectations of the final product and go into the details of how the imagined final product will look like. The purpose of this document is expressely the aforementioned and it is only intended for review by Dr. Daly or any other graders along with Group 3 who will be using this document as a centralized and organized list of requirements that will be divvied up amongst the our team members

scope

- There will be only one (1) final product produced with a few (3 4) prototypes leading up to it
- The product will be the game which will be hosted on the website (see GitHub for the most up to date instructions on getting to the website)
  - The benefits of this is that the game (product) will be lightweight and virtually anyone with access to the internet on relatively modern machine will be able to run and play the game on the web
  - The big picture objectives are: Develop game logic, tie logic to the web-games UI through Javascript, Creating a database for users, creating a global data base for data tracking, and finally publishing the game on the website for all to enjoy and learn from
  - The domain description of the application (game) is that it is an web-embedded game for improving sense of tempo and musical knowledge.

- The final product will be able to give feedback to users based on their performance in different levels of rhythm tracking difficulty.
- The final product will also allow users to keep their feedback centralized by tying their username to their acquired data.

Definitions, acronyms and abbreviations

- **BPM** is a measure of the speed or tempo of a piece of music and indicates the number of beats (pulses or taps) that occur in one minute
- **Measure** is a unit of time within a piece of music that is defined by a specific number of beats
- **YAGNI** YAGNI stands for "You Aren't Gonna Need It." It is a principle in software development and agile methodologies that suggests not to implement features or functionality until they are actually needed to satisfy known requirements.

Organization

This document will go through a couple of sections that speak generally about the product to be produced but will then delve into examples of product operations and what will be used to perform certain tasks.

The first half of this document is general and big picture ideas while the second half is more focused on the development and the theory that will be followed through development.

Overall Description

This section will cover all the necessary objects needed to complete the product

Product Perspective

- The product will be a stand alone music improvement game hosted on the website aforementioned in the first section. It will have no external dependencies and all tracks and pieces of art used will be free of any copyright or created by a willing team member
- There is a multitude of constraints for the interface
  - in EASY mode the user shall have only 4 options (displayed as 4 buttons) to pick from 1 of which will be the correct BPM to the audio clip that they can click on to play. Once the user presses one of the options they will be moved to the next question or if they have answered 4 questions previously and this is their 5th now, then they will be sent to the results screen in which they will get a

percentage score based on how accurate they were. EASY mode will also allow the user to click on a ribbon that will display 1 useful hint per question. The hint system will not be present in any other mode
  - in MEDIUM mode the user will have all the features of easy mode but without the hint system.
  - in HARD mode the user will lose the functionality of hint and the functionality of picking from 4 options. Instead the user will have only the audio clip play/pause buttons and a text box in which they will input their guess into. If the guess is not numeric then it will be rejected Standard gameloop applies

- System interfaces will be as follows
  - **Audio Interface:**
    - **Input:** The game should be able to load and play audio tracks
    - **Output:** Display control options in the game's audio player
  - **User Input Interface:**
    - **Input:** Capture user BPM guesses and event listeners in JavaScript
    - **Output:** Provide feedback to the user based on their inputs, such as whether their guess was correct and the overall performance in percentage at the end screen
  - **Local Storage Interface:**
    - **Input:** Store and retrieve user data, including BPM guesses and performance results, using local storage in the browser. This storage will be a simple organized text or JSON file
    - **Output:** Retrieve historical user data to populate the local leaderboard
  - **Global Leaderboard Interface:**
    - **Input:** Implement communication with a server or a cloud database to store and retrieve global leaderboard data securely.
    - **Output:** Display the global leaderboard within the game, showcasing top performers across all users
  - **User Authentication Interface:**
    - **Input:** User shall be able to establish a secure authentication mechanism to validate user identity.
    - **Output:** Provide authenticated users access to their personalized data and secure local leaderboard information
- There will be no Hardware interface as the entire game will be hosted on the web.
- User interface will be graphical (GUI) and it will be simple to understand. (TODO: PUT LINK TO DIAGRAM HERE)

These are the essential interfaces for the successful completion of the final product

Product Functions

Main functionalities will be

- User is able to create an account. This account creation will store user data in a JSON or a txt file on the backend of the website which will be used to authenticate users with their respective accounts.
- User will be able to change music and SFX volume in a settings option on the main menu. The main menu is the menu launched immediately after the web page of the game is fully loaded
- User will be able to start a game from the main menu which will take them to difficulty selection menu in which they will be able to choose from a/an EASY, MEDIUM or HARD mode. Once difficulty is selected the user will go into the game stage
- Regardless of game mode difficulty, the user will always be able to play, replay, scrub through and pause the audio clip accompanied by the prompt to guess the right BPM.
- User will be able to select 1 of 4 options during EASY and MEDIUM modes. EASY mode has hints which the user will be able to click on to get a useful hint. MEDIUM mode will not have the aforementioned feature concerning the hint system. The user will go through 5 questions and get a percentage score at the result screen
- User will not be given options for HARD mode but instead be met with a text box in which the only valid inputs are numerical ones.

User Characteristics

This game expects nothing of the user apart from minor musical knowledge when it comes to tempo and rhythm feeling. In effort of making this game accessible we will most likely be adding tts support so this game can reach as many people as possible. The previous is only hypothetical at the current standing.

Constraints

Below are a list of safety-critical properties of our game that will be hosted on the webpage

- **Data Security:**
  - Ensuring that user data, including personal information and game-related data, is securely stored and transmitted. This prevents unauthorized access, data breaches, and protects user privacy

- **Secure Authentication:**
  - Implementing robust user authentication mechanisms to verify the identity of players. This prevents unauthorized access to accounts and ensures that only legitimate users can participate in the game
- **Game Stability and Reliability:**
  - Ensuring the game's code is robust and free from critical bugs that could lead to unexpected behaviors or crashes. This contributes to a stable and reliable gaming experience, reducing the risk of disruptions or system failures

Below are a list of constraints properties of our game that will be hosted on the webpage

- **Browser Compatibility:**
  - The game may not be compatible with all web browsers. Ensuring cross-browser compatibility can be a constraint, especially when dealing with older browser versions. This may be avoided by advising to play on the latest versions of Firefox or Microsoft Edge for the best experience
- **Internet Connection Speed:**
  - The game's performance may be constrained by the user's internet connection speed. Large asset files, such as high-resolution graphics or extensive audio, may lead to slower loading times for users with slower internet connections
- **Device Compatibility:**
  - The game should be designed to work on a variety of devices, including desktops, laptops, tablets, and smartphones. Ensuring responsiveness across different screen sizes and resolutions can be a constraint

Assumptions and Dependencies

There are very little assumptions made about the users interactions and below will be the ones that are foreseable

- **Assumption: Stable Internet Connection:**
  - It is assumed that users have a stable and reliable internet connection to access and play the game without significant interruptions.
- **Dependency: Web Hosting Service:**

  - The game relies on a web hosting service to store and serve game files, ensuring availability and accessibility to users.

- **Assumption: Browser Compatibility:**
  - It is assumed that users will access the game using modern web browsers that support the necessary technologies and standards.

- **Assumption: User Familiarity with Web Navigation:**
  - It is assumed that users are familiar with basic web navigation principles to interact with and navigate through the game interface.

Apportioning of Requirements

The requirements that were outlined through customer negotiations are very low and realistic. This was done with the expectation that the final product will be done on time and then as updates come along if the customer desires we add more features to the game as need arises. (YAGNI development principle)

Specific Requirements

1. User Registration and Authentication
   - 1.1 Users can create accounts with unique usernames and passwords.
   - 1.2 Users can log in securely to access their profiles
2. Game Mechanics
   - 2.1 User can start a rhythm test game.
   - 2.2 The game assigns a sound clip with a specific BPM to the user.
   - 2.3 Users have to guess the BPM of the sound clip
   - 2.4 The game records user responses.
   - 2.5 Users can select the difficulty level (easy, medium, hard).
   - 2.6 Users can access hints and explanations of musical concepts during the game.
3. Scoring and Progress
   - 3.1 The application calculates and displays the user's accuracy in percentage after each test.
   - 3.2 Users can view their test history and individual results.
   - 3.3 After completing five tests, the application calculates and displays the average accuracy.
   - 3.4 Users can track their progress over time.

4. User Profile
    - 4.1 Users can view their profile information.
    - 4.2 Users can update their profile information.
    - 4.3 Users can view their sense of rhythm measurement.
5. User Interaction
    - 5.1 Users can receive feedback on their performance.
    - 5.2 Users can compete with others (leaderboards or challenges).
    - 5.3 Users can share their results on social media.
    - 5.4 Users can invite friends to play the game.
6. Technical Requirements
    - 6.1 The application must be a web-based platform accessible on modern browsers.
    - 6.2 The application should use secure encryption for user data and authentication.
    - 6.3 The application should be responsive and work well on different screen sizes.
    - 6.4 The game mechanics should provide a seamless and enjoyable user experience.
    - 6.5 The application should have a database to store user profiles and results.
7. Testing and Quality Assurance
    - 7.1 The application should undergo rigorous testing to ensure accuracy and reliability.
    - 7.2 User feedback should be collected and used for improvements.
8. Documentation
    - 8.1 Provide comprehensive user documentation on how to use the application.
    - 8.2 Provide technical documentation for developers and administrators.
9. Legal and Compliance
    - 9.1 Ensure compliance with relevant laws and regulations, including copyright and data protection laws.

Modeling Requirements

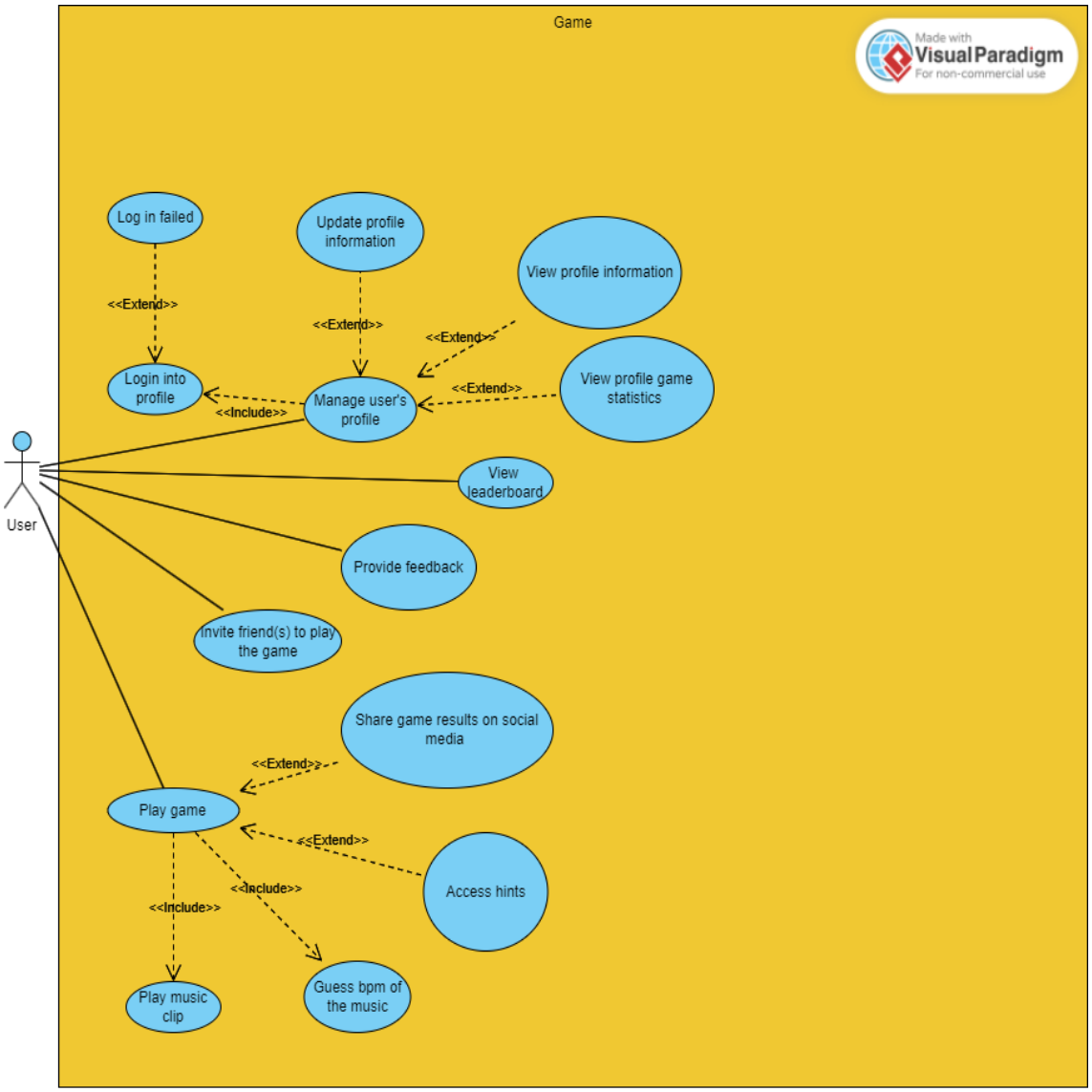


Figure 1: Use case diagram

Use Case Name:	Manage user's profile
Actors:	User
Description:	The user modifies and views profile information
Type:	Undefined
Includes:	Login Into Profile
Extends:	Update profile information, View profile information, View Profile game statistics
Cross-refs:	None
Uses Cases:	Manage user's profile, View leaderboard

Use Case Name:	Play Game
Actors:	User
Description:	The user plays the game.
Includes:	Play music clip, Guess BPM of the music
Extends:	Access hints, Share game results on social media
Cross-refs:	None
Uses Cases:	Play Game

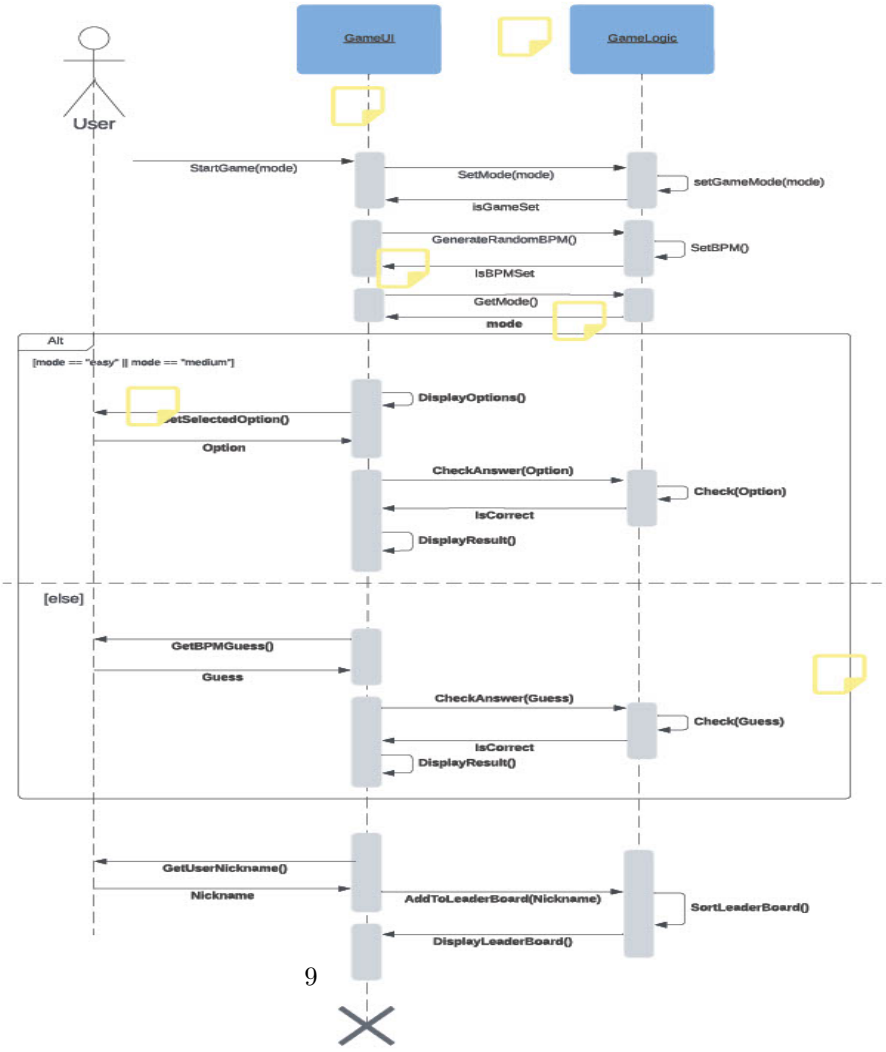


Figure 2: Sequence 1

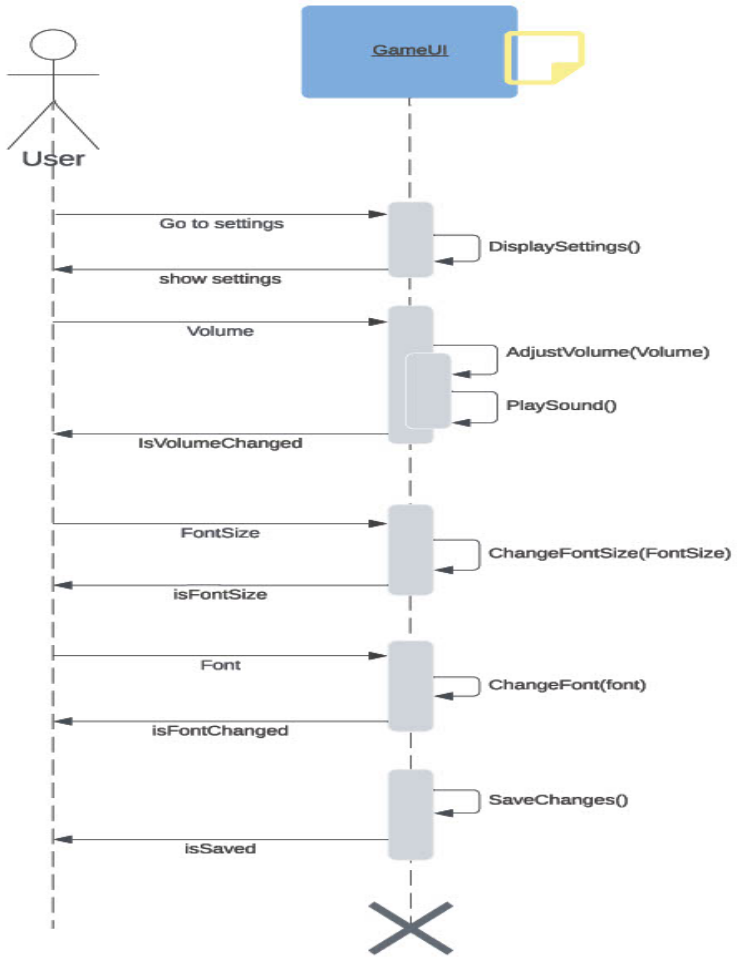


Figure 3: Sequence 2

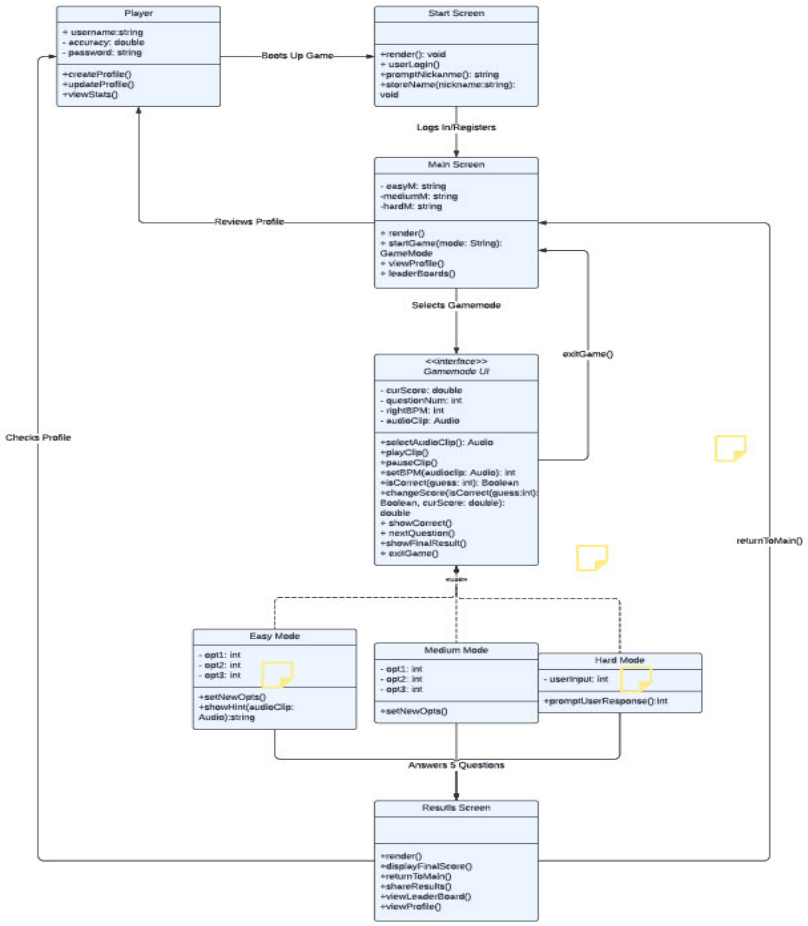


Figure 4: Class

Prototype

How to Run Prototype

To run the prototype you need to have a device that is able to run a web browser that is configured to run JavaScript. To access the prototype first extract the SWE-Project folder. First Open the folder named "SWE-Project" then open the folder named "Pages", then open the folder named "1.Start Page", then open the file "StartScreen".

Sample Scenarios

Imagine stepping into a scenario where a student wants to play our game. The student then connects to their device, one that supports a browser with JavaScript enabled, and then looks up the URL to our website and types it right into the address bar and presses enter. The browser then takes the user to our website, where they will start their gaming session.

Once the user lands on our website, they will first encounter a registration page that looks exactly like the one shown in Figure 5, where the user will be asked to register themselves by typing a name and pressing the button "register" located to the right of the textbox. Once that is done, the user will be taken to the main screen of the game, where they will be able to choose whether they want to start playing the game right away, check the leaderboards, or change some of the settings of the game. The main screen is shown in Figure 6 so we can have a better understanding of how the layout is displayed.

Once the user has decided where to go, they can press one of the three buttons that will perform the desired action. If we suppose that the student wants to change some of the settings before they start playing, then once they press the button "Settings" they will be taken taken to the page shown in Figure 7 where they will be able to see one slider that allows them to change the font size of the game, and one dropdown menu that allows them to change the font of the text. Once they are done customizing the game, they can press the black back arrow located on the top left corner of the page which will make them return to the main screen. Then, if we assume the player wants to start a new game, then once they press the "Play Game" button they will be taken to the screen shown if Figure 8 where the user is asked to decide whether it wants to play on easy, medium, or hard mode. Depending on the choice of the student, the game dynamics will change making the game either more challenging or more educational. This can be seen in Figure 9, Figure 10, and Figure 11). The differences are quite simple. When playing on hard mode, the user does not get any options to choose from. Instead, they must type how many beats per minute they think they are listening to. If they are correct, they score, if they aren't, they don't. When playing on Easy, the user gets 4 options to choose from plus a tooltip

located to the right of the stage number that will give them hints of what the answer should be. In this mode, the players are allowed to make two mistakes before finally getting the question wrong (as seen in Figure 12, and Figure 13). the more mistakes they make the less they score. The medium mode works pretty much the same way as the easy mode, except that the players do not have access to hints and are only allowed to make one mistake per question.

The game is over once the user answers all five questions. after that they will be taken to a screen similar to the one in Figure 14, where they are able to see and register their score, play again, change difficulty, and check the leaderboard. The leaderBoard (Figure 15) is quite simple and can be accessed from both the main screen and from the end screen (the one that displays your results). Once the student gets to that screen, they will see the highest scores from both a global perspective (top scores in the world) and from a personal perspective (their own top scores). This way the user is able to keep track of their performance and set challenges for themselves in order to improve in the game.

Figure 5: Start screen

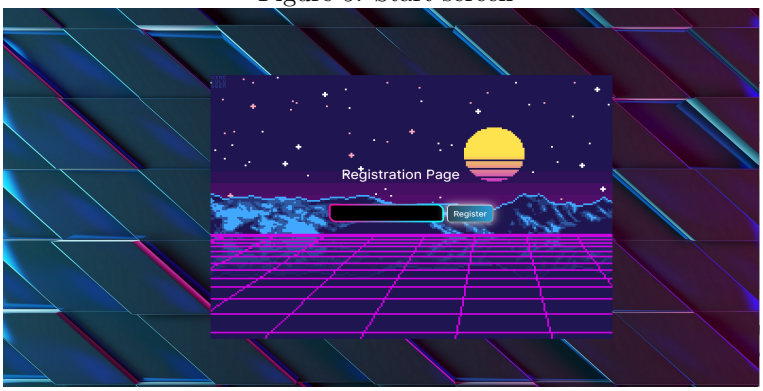


Figure 6: Main screen

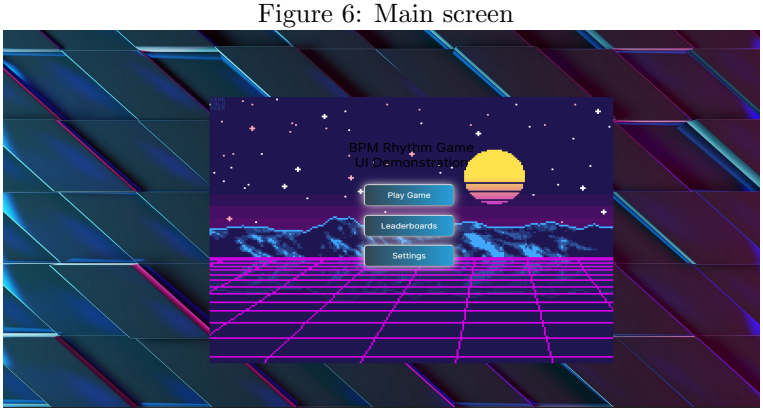


Figure 7: Setting screen

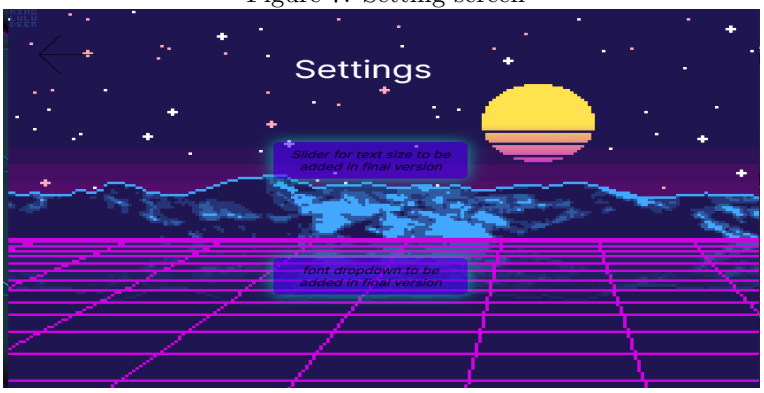


Figure 8: Select Mode screen

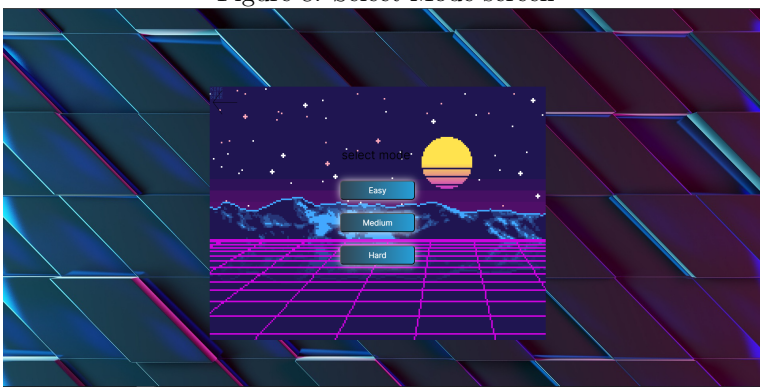


Figure 9: Hard screen

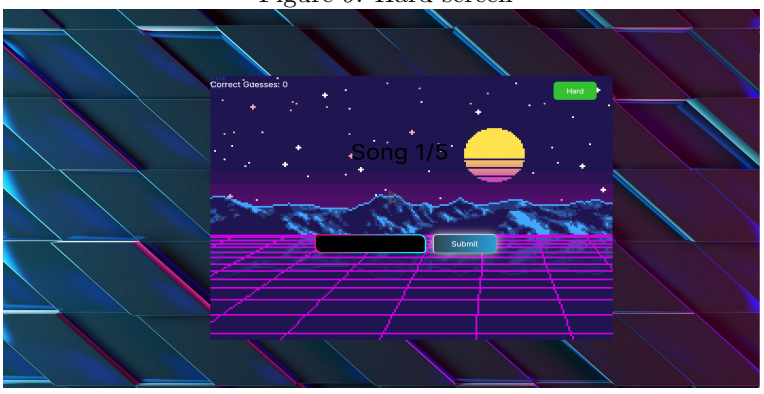


Figure 10: Medium screen

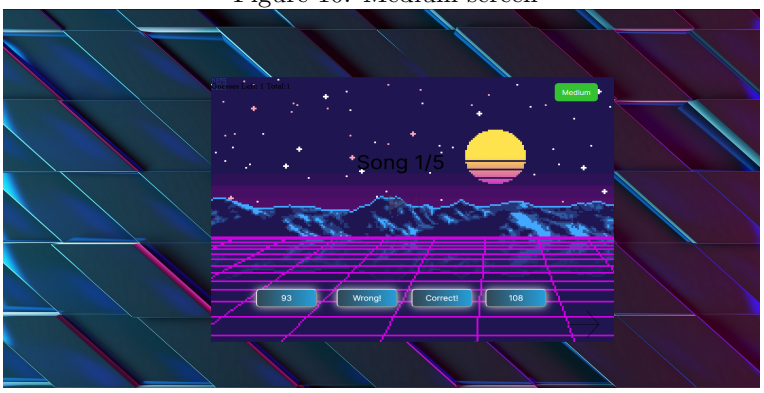


Figure 11: Easy screen

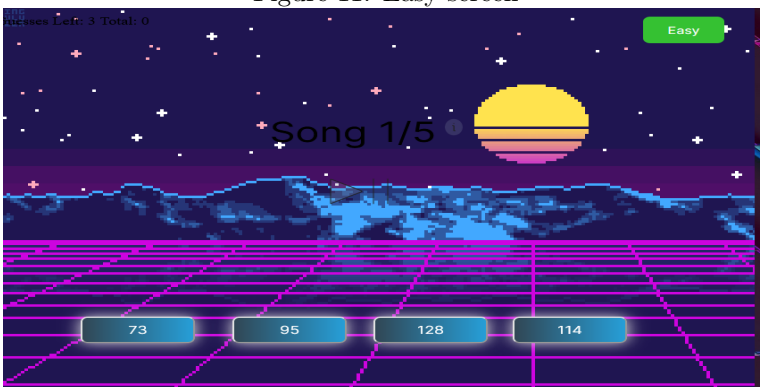


Figure 12: Easy screen one wrong

Figure 13: Easy screen couple wrong

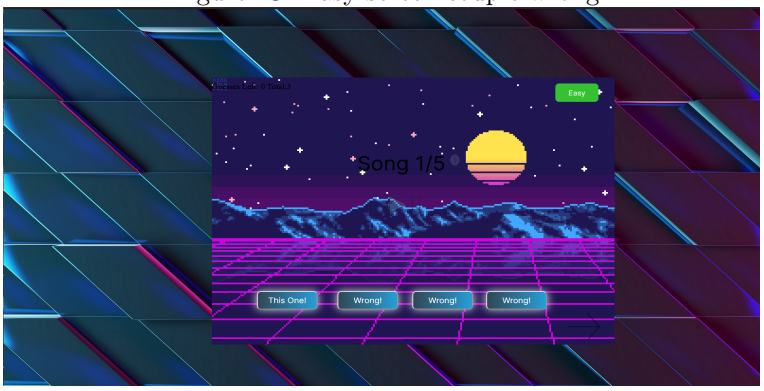


Figure 14: End screen

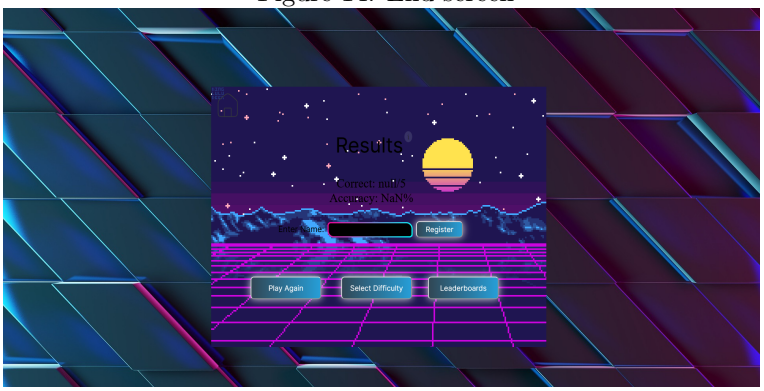
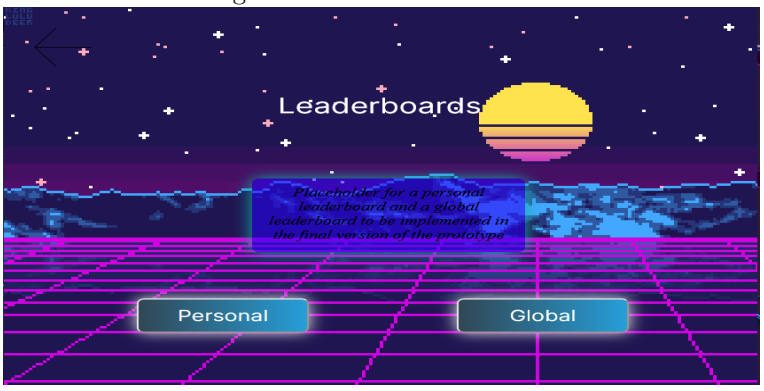


Figure 15: Leaderboard screen



References

Pixabay Image: Glass Sci-Fi (CC0)
https://pixabay.com/photos/glass-scifi-violet-ultraviolet-3389935/
License: https://pixabay.com/service/license-summary/

DeviantArt GIF: Custom Box Background AESTHETIC IS DEAD by King-Lulu-Deer (Published on Jun 12, 2018)
https://www.deviantart.com/king-lulu-deer/art/Custom-Box-Background-AESTHETIC-IS-DEAD-74939

Font Used: November Font by Brandon S. or Tepid Monkey Fonts
https://www.1001fonts.com/november-font.html
Last Modified: February 24th, 2013

Point of Contact

For further information regarding this document and project, please contact Ali Qattan at University of Massachusetts Lowell (Ali_Qattan at student.uml.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.