Not My Forte Piano
By: Team Not My Forte Piano

Jake Hamblin, Nick Latham, Vaibhav Patel,
Alexander Shampton, and Alex Yeoh

Elevator Pitch

We want to build a browser-based piano game because we feel that it is a void that might be able to be filled. While there are a number of app-equivalent games, not many are browser-based. More importantly, most of them do not offer a Guitar Hero-style of gameplay. The game will include a sandbox mode, an interactive timing-based skill mode, and a follow-along mode. Users will be able to select between many formats of gameplay, from casual to more in-depth, competitive styles, allowing nearly everyone to have an interest in the game.

Overview

The architecture of our piano game consists of four modules:

- User Interface (Alexander Shampton)
- Front End (Jake Hamblin)
- Back End (Alex Yeoh)
- High Scores tables and songs directory (Nick Latham)

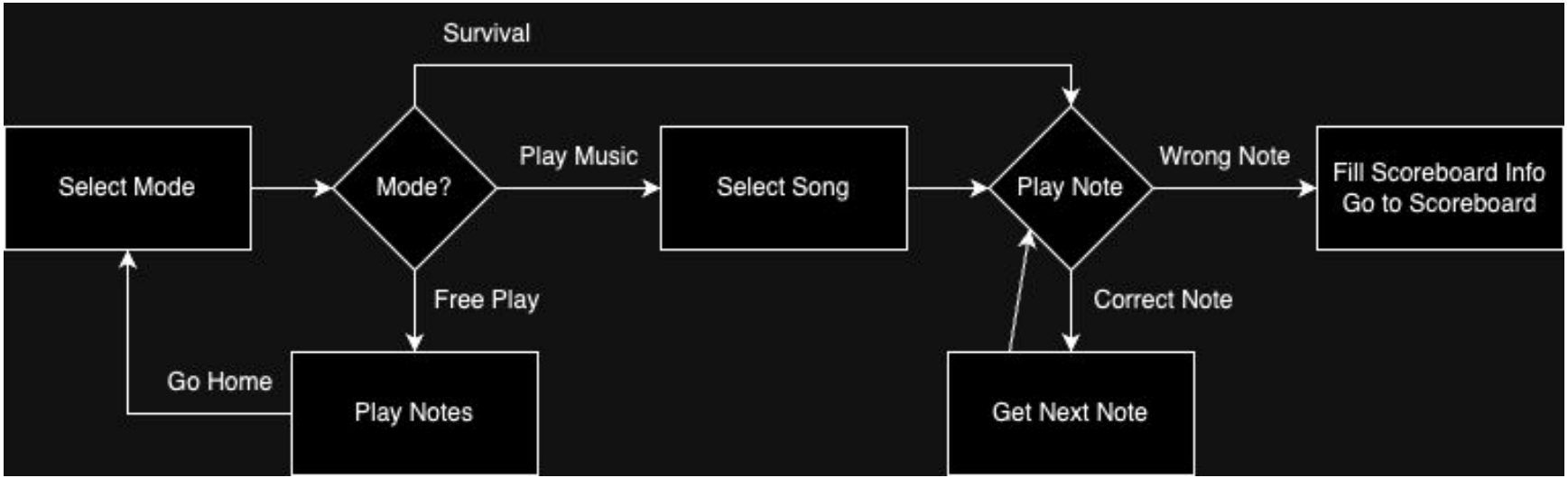
The testing plan was created by Vaibhav Patel

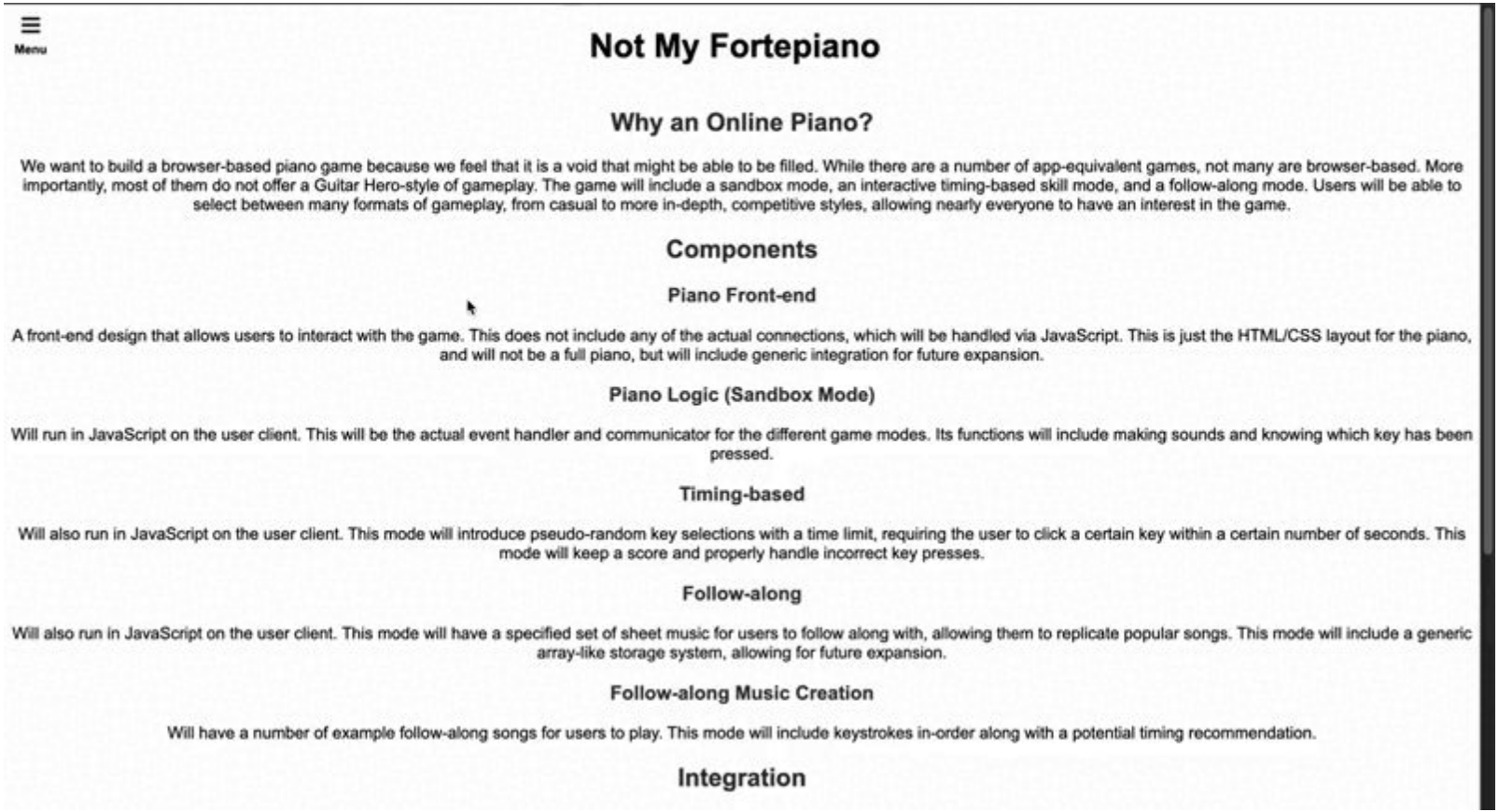
Piano Logic (Alex Yeoh)

User Story
As a musician, when I play a song in the song mode and don’t get a high score, I want to receive feedback so I can understand how to improve and try again.

Requirement
The system shall have functionality to play a song again in Play Music.

Piano Logic Diagram




Menu
Not My Fortepiano
Why an Online Piano?
We want to build a browser-based piano game because we feel that it is a void that might be able to be filled. While there are a number of app-equivalent games, not many are browser-based. More importantly, most of them do not offer a Guitar Hero-style of gameplay. The game will include a sandbox mode, an interactive timing-based skill mode, and a follow-along mode. Users will be able to select between many formats of gameplay, from casual to more in-depth, competitive styles, allowing nearly everyone to have an interest in the game.
Components
Piano Front-end
A front-end design that allows users to interact with the game. This does not include any of the actual connections, which will be handled via JavaScript. This is just the HTML/CSS layout for the piano, and will not be a full piano, but will include generic integration for future expansion.
Piano Logic (Sandbox Mode)
Will run in JavaScript on the user client. This will be the actual event handler and communicator for the different game modes. Its functions will include making sounds and knowing which key has been pressed.
Timing-based
Will also run in JavaScript on the user client. This mode will introduce pseudo-random key selections with a time limit, requiring the user to click a certain key within a certain number of seconds. This mode will keep a score and properly handle incorrect key presses.
Follow-along
Will also run in JavaScript on the user client. This mode will have a specified set of sheet music for users to follow along with, allowing them to replicate popular songs. This mode will include a generic array-like storage system, allowing for future expansion.
Follow-along Music Creation
Will have a number of example follow-along songs for users to play. This mode will include keystrokes in-order along with a potential timing recommendation.
Integration

High Score Table (Alex Shampton)

- User Story:
  - As a competitive gamer, I want to be challenged by other survival mode players' high scores, even if I cannot beat their scores.
- Requirements:
  - The system shall have a leaderboard that contains scores
  - The system shall have a leaderboard that shows all highscores

High Score Table (Alex Shampton)

- Design Specification:
  - The frontend shall have a menu that allows the user to select between "free-play", "survival", "play song", and "see high scores" modes. This shall be contained within a menu, so the user may select which mode to play in.

High Score Table (Alex Shampton)

score	name
10000000	Alex Shampton
9999999	AS
9999998	Not as good

Play Along (Jake Hamblin)

- User Story:
  - As a competitive musician, I want to be able to play a provided song and get a high score so I can compete for the best score.
- Requirement:
  - The system shall have a song selection menu
  - The system shall allow the user to input their score into the leaderboard for Play Music mode

Play Along (Jake Hamblin)

- Design Specification:
  - The system shall have a database for songs. These songs will be broken down into notes that the front end will display to the user. The backend shall keep track of the number of notes the user plays that correctly correspond to the notes of the song, and determine the user's score based on the this.

Play Along (Jake Hamblin)

- Select song
- Read song database
- Show notes to play
- Determine if correct note played
- End collects name and sets score
- Score appears in leaderboard

Free Play (Nick Latham)

- User Story:
  - As an aspiring musician, I want to be able to select and use the Free Play mode to experiment with writing my own music.
- Requirements:
  - The system shall contain a Free Play mode
  - The system shall contain a menu selection for Free Play mode

Free Play (Nick Latham)

- Design Specification:
  - The logical components of the system shall have logic to allow the user to play individual piano notes. The notes will be acquired from a MIDI file and each piano key will correspond to its proper note.

Free Play (Nick Latham)

- Select Free Play mode from Menu
- User presses a key
- Front end sends note pressed to back end
- Back end returns note sound to front end
- Note is played
- Continue sending and returning the notes the user presses until user decides to quit and return to Menu

Free Play (Nick Latham)

```
@app.route("/freeplay")
def render_freeplay():
    session["mode"] = "freeplay"
    session["previous_note"] = 60  # default start note is middle c
    session["score"] = 0
    return render_template("freeplay.html", mode="freeplay")
```

Survival mode (Vaibhav Patel)

- User story:
  - As a gamer, I want to be able to play survival mode and save my scores into a leaderboard, so I can see my progress as I get better.
- Requirements:
  - The system shall contain a survival Play mode
  - The backend shall handle the logic for checking high scores and updating the leaderboard.

Survival mode (Vaibhav Patel)

- Design Specification:
  - The design specification for the user story will involve implementing Survival mode where the user can play and track their score. If the score qualifies as a high score, the system will prompt the user to enter their name and save the score to a leaderboard stored in database. The game will allow players to see their progress after each session.

Survival mode (Vaibhav Patel)

- Select Survival mode from Menu
- The front end initiates Survival mode gameplay
- The backend calculates the score and checks if it qualifies as a high score based on performance
- If the score is a high score, the front end prompts the user to input their name, and the backend saves it to the leaderboard
- After saving the score or returning to the menu, the user is taken back to the main menu.

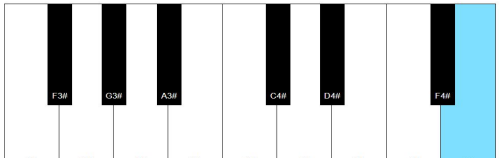
Scoreboard

Survival Highscores

Updated Score

Old Score

score	name
60	Vaibhav
26	Alex Shampton
16	Vaibhav



ceg4110-group-project-not-my-fortepiano > code > templates > survival.html

```
1   {% extends 'base.html' %}
2
3   {% block title %}Survival{% endblock %}
4
5   {% block header %}
6       <h1>Survival</h1>
7   {% endblock %}
8
9   {% block content %}
10      {% include 'piano.html' %}
11  {% endblock %}
```

Unit of Complexity (Vaibhav Patel)

- Low Complexity:
  - Survival Mode (No High Score)
  - Song Mode (No High Score)
  - Free Play Mode (User Exits)
- Medium Complexity:
  - Survival Mode (High Score Achievement)
  - Song Mode (High Score Achievement)
  - Invalid Name Input (Leaderboard Submission)
  - Score Entry Edge Cases
- High Complexity:
  - Unexpected Game Exit (During Gameplay)

Testing Plan (Vaibhav Patel)

- Testing Plan for Game Modes:
  - Testing requirements for all game modes: Survival Mode, Play Music Mode, Free Play Mode
  - Focus on functionality, score handling, user interface, and database interactions
- Mode Selection Testing:
  - Test if user can select between modes (Survival, Play Music, Free Play, High Scores)
  - Test if each mode loads properly upon selection

Testing Plan

- Gameplay Functionality Testing:
  - Survival Mode: Verify correct start, score tracking, leaderboard integration, name input for high scores
  - Play Music Mode: Test song selection, score calculation, leaderboard display
  - Free Play Mode: Ensure free play starts and ends correctly with no score tracking

Testing Plan

- Score and Leaderboard Testing:
  - Verify leaderboard displays high scores for Survival and Play Music modes
  - Ensure scores are stored and retrieved correctly from the database
  - Test functionality to overwrite scores with higher scores

Testing Plan

- Database and Data Persistence Testing:
  - Verify database is storing scores, song data, and user names correctly
  - Ensure data persistence across gameplay sessions
  - Test data retrieval for songs and leaderboard

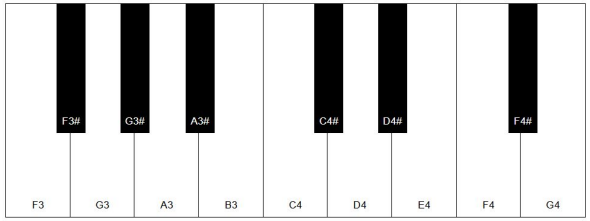
Testing Plan

- User Interface (UI) Testing:
  - Test the piano-style keyboard GUI in each mode
  - Validate menu navigation between modes (Survival, Play Music, Free Play)
  - Ensure score input popup for high scores functions correctly

Testing Plan

- Error Handling & Performance Testing:
  - Test error handling for invalid inputs and database failures
  - Ensure the game runs smoothly across platforms (PC, mobile, browsers)
- Endgame & Session Flow Testing:
  - Ensure correct redirection to the main menu after gameplay
  - Test smooth transition between game modes and exit functionality

Freeplay



Scoreboard

Survival Highscores

score	name
60	Vaibhav
26	Alex Shampton
16	Vaibhav

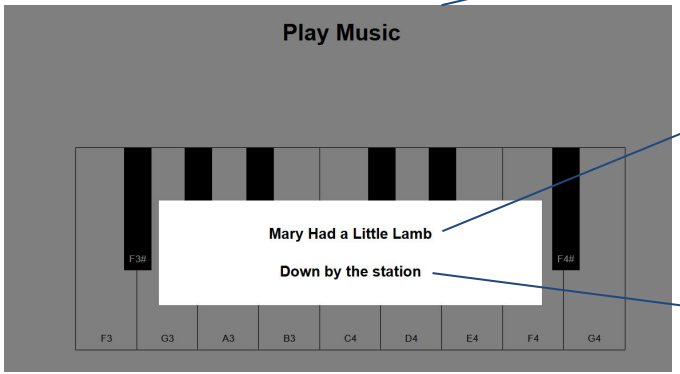
Mary Had a Little Lamb

score	name
14	asd
0	
0	asd

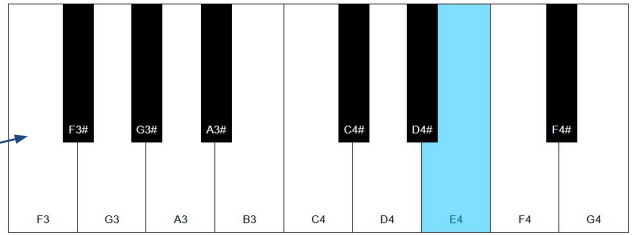
Down by the station

score	name
0	
0	
0	

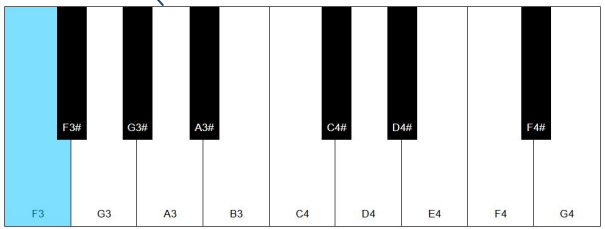
Play Music



Play Music



Survival



Conclusion

We have created a browser-based piano game that offers three different modes of Guitar Hero-style play and keeps scores for the user. The game we have created can be enjoyed by anyone from experienced musicians to competitive gamers.

Questions?