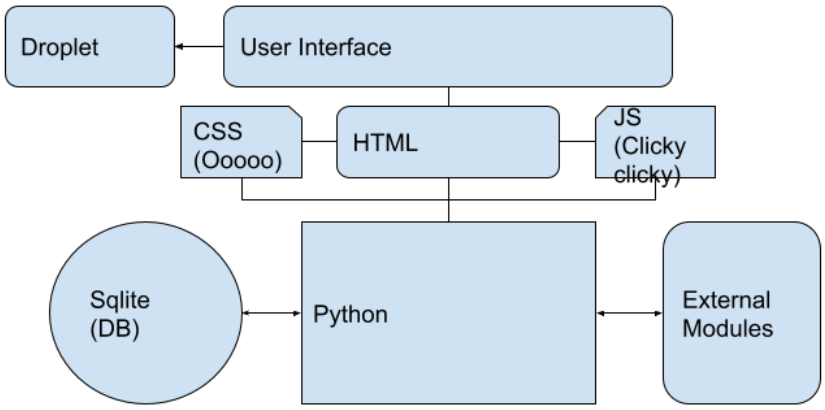
Doink! – Julia (Lia) Nelson (PM), Lucas (LTW) Tom-Wong, Liesel Wong, Tomas Acuna, Dahlia, Tobias, King Hagrid, Llamy the Llama
SoftDev P02
2022-03-04
Title Project: DOINK!

Description of Project:

Rhythm game where the user clicks the screen on pre-ordained intervals. These are randomly generated based on the tempo and time signature.
User Interaction
User clicks anywhere on the screen to register an action
→ If User fails to click on the screen when action is needed or clicks at the wrong time, a life is lost. The will be a delay before ability to lose another life
→ Score increases with each successful click

Components:



CSS: Makes everything pretty(ier).
JS: Interaction and animation.
HTML: What the user sees which will be influenced by everything.
Python: Generate random rhythms, interact with database and external modules, facilitates communication between other components.
Sqlite: Acts as the database. Stores data and information in a table which will be pulled for various nefarious purposes.
External Modules:

- *Hashlib* for password hashing

Front-End Framework: *Bootstrap*. Framework templates will allow easier formatting over creating custom stylesheets by using something that already exists. The documentation is also easier to understand than Foundation. We will largely be using its preexisting classes, especially with our buttons, in order to provide a consistent look.

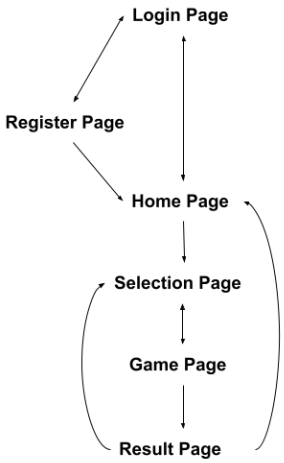
What is new compared to P01? **JavaScript!!!!!**
JavaScript helps facilitate user interaction by adding scripts to html allowing the page to react to user inputs.

What is needed for this project? **Audio!**
We need audio for a rhythm game…. Probably. We need to be able to generate a random rhythm as well as audio to compliment it. This will most likely be done through javascript or python.

Database:

User (Primary Key, Text)	Pass (Text, Ideally Hashed)	High Score (Int)
joeshmoe	f7e1df1d3eedc8bcbe928e03fedca73afbfe807e2ec147bbb277a8668b47443b	2187
bilbobaggins	2c1f58751dbb39f17f5d3575fd4baa0f6a76d8ad55df0c71e013afcc5ae0ce6a	24601

Site Map:



Understanding the Site Map:

- Register Page
  - Button to Login Page
  - Input
    - User
    - Pass
  - Button to enter input
    - Gives error or leads to home with user logged in
- Login Page
  - Hyperlink to Register Page
  - Input
    - User
    - Pass
  - Button to enter input
    - Gives error or leads to home with user logged in
- Home Page
  - Logout button
  - Welcome user
  - Description of Game
  - Instructions for game
  - Button to start game (leads to selection page)
  - Shows user's top score

- Selection Page
  - Select tempo
  - Select time signature
  - View current score
  - Button back to home page
  - Start Button to Game Page
  - Current lives counter
- Game Page
  - Display rhythms
  - Cursor to follow rhythms and get user input
  - Interact with user input
  - Display tempo
  - Display time signature
  - Display lives (3 lives, if you make a mistake you lose a life)
  - Display score on the side
  - When the game ends (Loss due to lives or if ended game), display score
  - Button to Result Page (end game)
- Result Page
  - States if high score
    - Asks to save score in database
  - Button to Home Page (home)
  - Button to Selection Page (start new game)

Goals:

1. Get a metronome playing
   a. Make metronome optional
   b. Add optional flash with metronome
2. Show tempo and time signature
3. Generate rhythms
   a. Have maximum length of time
   b. Determine time left
   c. Evaluate feasible rhythms
   d. Select rhythm from list
   e. Stop at end of time
4. Display rhythms generated (bars and notes should be shown)
5. Add cursor
6. Determine user responses and evaluate if correct
7. Display result of said evaluation

8. Add other tempos
9. Add other time signatures
10. Establish point scoring system
    a. Three lives
    b. Given a grace period before they can lose a life again
    c. Each correct rhythm gets a points
11. Display lives and points on screen
12. Make accounts
13. Hash passwords
14. Generate background sound to match rhythms

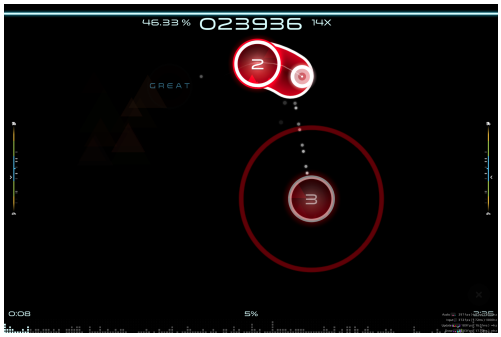
Roles:

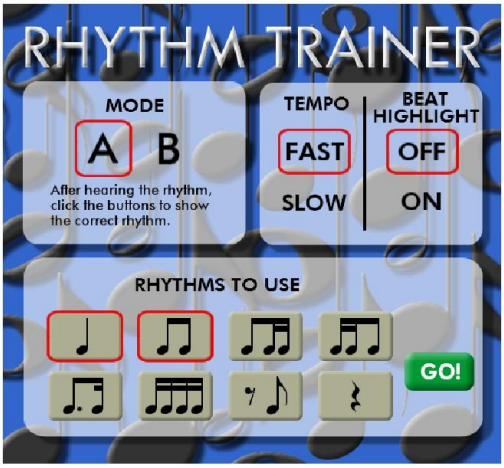
Lia (PM): Audio/Rhythm Generation (Create Functions for JavaScript to Call)
LTW: HTML + CSS (Design, Front-End)
Liesel: Sqlite and Python Interaction (Account-Based Interaction) - (Going between pages, interacting with database, randomizing sound file)
Tomas: JavaScript (User Interaction and Animation)

Potentially Important and Useful Links:

https://developer.mozilla.org/en-US/docs/Web/API/HTMLAudioElement/Audio
https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API
https://developer.mozilla.org/en-US/docs/Web/API/HTMLAudioElement
https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio
https://en.wikipedia.org/wiki/WAV
https://docs.python.org/3/library/wave.html
https://piazza.com/class/kv0wqn7faux3ye?cid=33
https://developer.chrome.com/blog/canvas2d/
https://github.com/tabs4acoustic/GetSong-Metronome
https://getsongbpm.com/api?fbclid=IwAR1PKDUzz218Cm0QLaC53IIoOTAhjgQ2Wp284T5TL16IB0CoE9Af7FdSUQ4
https://api.jquery.com/

Inspiration:





Target Ship Date:
2022-03-28