Neam Time - Patrick Ging, Aaron Contreras, David Chong, Deven Maheshwari, Ryan Wang
SoftDev
P02 -- Design Document
2022-03-04
time spent: 1 hour

Proposal

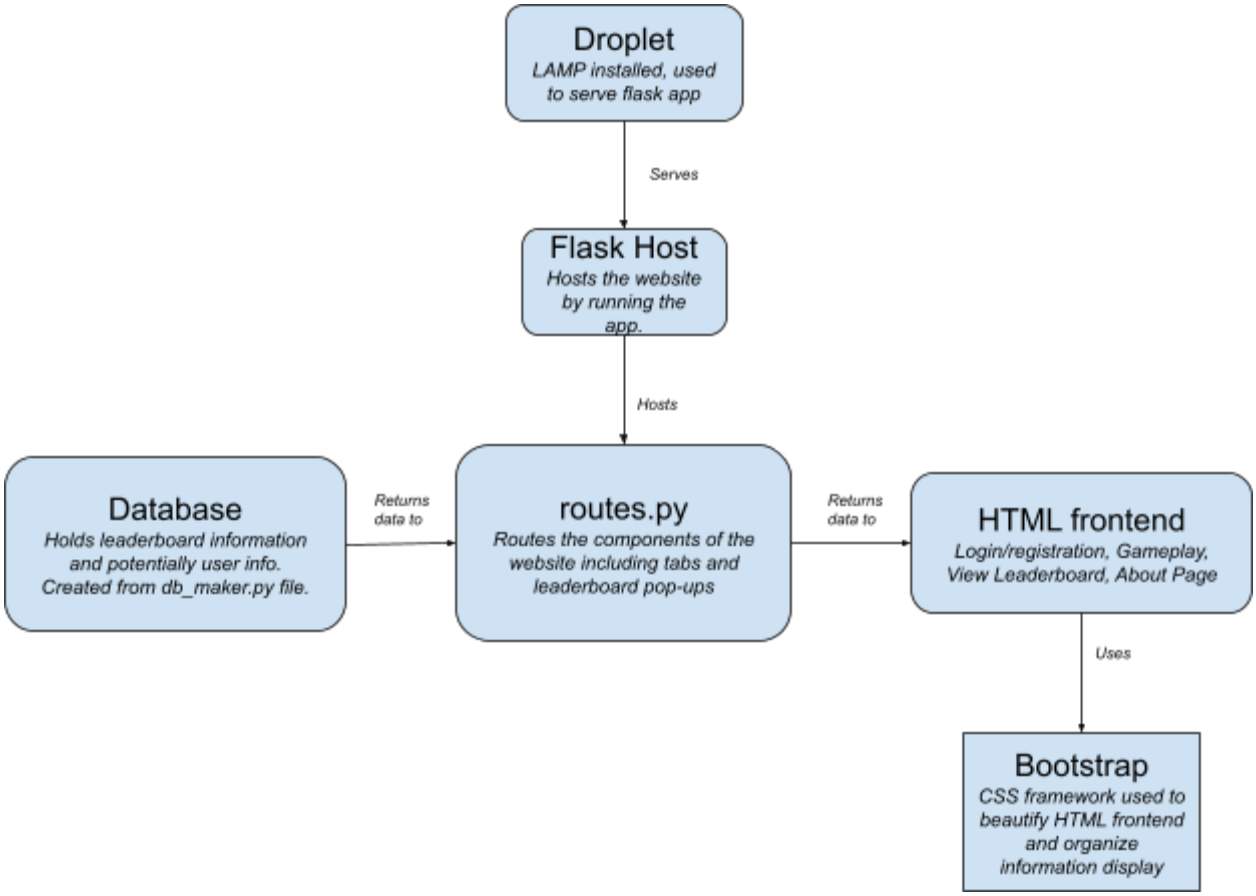
Enhanced *Wack-A-Mole*. Essentially, this app will allow the user to rediscover their love for the classic game *Wack-A-Mole* with various minigame additions. One plan is diving into the mole's perspective a bit more, adding an extra scene where the user dodges flying hammers from the sky as a mole. We will also experiment with mix-ups during the actual game, like inverted controls and random bursts of speed.

Design Components

- Flask - Will serve as the python backend, routing the various python files and html front end code. Handle score and leaderboard inputs, as well as lives (3) for the user.
- HTML and CSS/JS (Bootstrap) - Use to display the front end and make the website aesthetically pleasing. Handle form inputs. JS will be used for a canvas and backdrop to show the mole animation. CSS for styling.
- Sqlite3 - Stores leaderboard information. After every score by any user, a name input field is asked and the user will be able to input their preferred name and respective score. Only the top 10 scores will be presented on the front end. Think of a PacMan arcade game where there is no login but still a leaderboard.
- Digital Ocean Droplet

Component Map

- Main/Home page
- Leaderboard page
- Game page
- About page



Database Organization (Includes stretch goals)

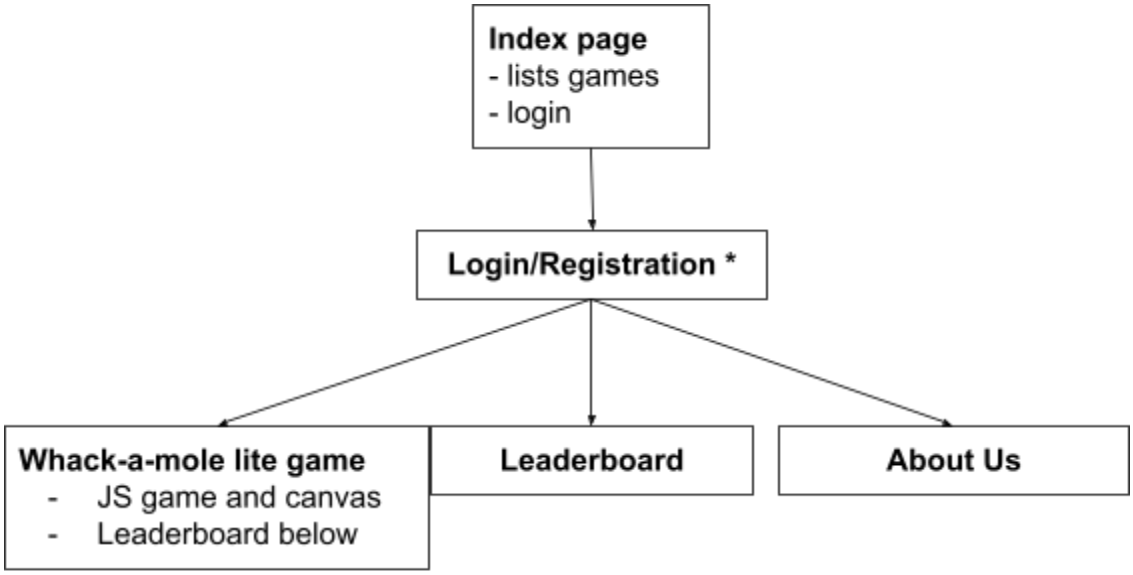
Leaderboard

USER	**SCORE**
PRIMARY _KEY TEXT	INTEGER

User Info

USERNAME	**PASSWORD**	**TOPSCORE**	**TIMESTAMP**
PRIMARY _KEY TEXT	TEXT	INTEGER	TEXT

Sitemap (Frontend)



* Stretch goal

- / : home page
    - Link to login/register
    - Can play game without being a user
    - Can view about page
- /login : a login page
    - Redirect to main page once logged in
- /register : a sign up page
    - Allows user to sign up for an account
    - Redirects to login page
- /wack : wack-a-mole gamepage
    - Choose between main *Wack-A-Mole* game or toggle between minigames
    - Buttons for start and pause
    - If logged in, top score from each run is recorded in the database along with timestamp
    - Leaderboard input field pop up after each run
- /leaderboard : displayed leaderboard
    - Displays top 10 scores
    - Search functionality by user and by score

- /about : about us page & meet the team
    - Basic tutorial and project vision

Game Mechanics:

- Player starts with 3 lives, grids will randomly flash a mole
    - Backdrop of the canvas will be an arcade setup. We will use mole sprites and mallet sprites in order to visually simulate the game
    - Moles switch between base mole and squashed mole and mallets appear based on mouseX and mouseY
- While mole is flashed player will be able to click it, mole will squash and disappear
- Players score and streak will be recorded
    - A streak of 10 (subject to change) will result in the user being prompted (optional) to play one of the minigames to quickly improve their score. Game will be decided randomly.
- Leaderboard prompt for name will be added at the end of each run. Can be a button for adding to the leaderboard and will record every score.

Task Breakdown

- **Pat(PM):** Bootstrap/HTML frontend. JS animations
- **Deven(Devo):** Sqlite & JS animations
- **Aaron(Devo):** Routes.py, minigame 1: *Mole Perspective*
- **David(Devo):** Routes.py, minigame 2: *Game modes*
- **Ryan Wang(Devo):** CSS

Stretch Goals:

- ☐ User accounts added to Sqlite - stores mole specifications and sprite choice, personal high score, the ability for more personal mini games

Ship Date

March 21, 2022