

Project Description:

Our project is a new and improved Snake Game with a leaderboard, different game modes and settings. The game modes include various speeds, map sizes, obstacles, bosses, and features like border wrapping, peace mode, and poison apples.

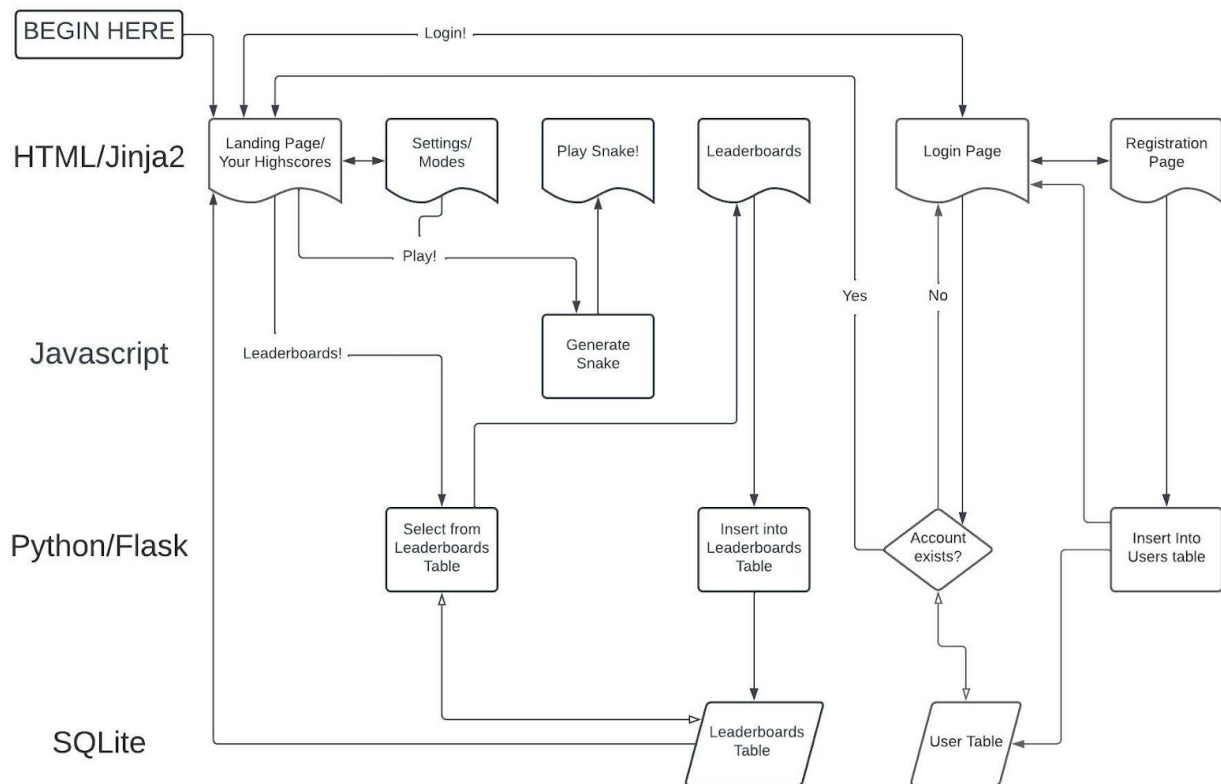
- Speeds: slow, medium, fast, insane
 - Different speeds have a different score multiplier, so fast score can be analogous to slow score in leaderboard
 - Multipliers: slow- 0.5; medium-1.0; fast-1.5; insane- 2.0
- Map sizes: small (12 x 6), medium (24 x 12), large (48 x 24)
 - Size of canvas does not change with map size, just how big each square is dictates how many squares are on the field
- Obstacles: for every 4 apples you eat, a block appear on the app
 - If score doesn't increment on a multiple of 4, obstacles appear on closest number rounded or floored to multiple of 4 (i.e. fast, on score 7.5, obstacle appears)
- Border wrapping: hitting the wall does not kill you, it loops you to other side
- Peace mode: can't die, you can hit wall (border wraps), and can hit yourself
- Flying apples: the apple moves around the map (can only move up or down, no diagonals)
- Poison apples: you can only eat the apple on one side; if you eat it from any other side, you lose points, but if you eat it correctly, you gain twice as many points than usual

Components:

- JavaScript
 - Game logic and animations
 - (mode from python, send to python score for db)
 - Leaderboard data matching with selected mode
 - (send mode to py, get data from py)
- Python
 - Send information using flask to create webpage
 - Differentiating between modes
 - (mode data sent to js for game)
 - Leaderboard
 - (get selected mode from js, send db data to js)
 - Log-in and register
- SQLite
 - Set up the database

- HTML/CSS, Jinja2
 - Making webpage look good
 - Functional frontend lists and dynamics
- Bootstrap
 - Front end framework
- Jinja2
 - Create the HTML template
- Flask
 - Create the site that the user will interact with
 - Create a session when the user logs in

Component Map: [link](#)



Database Organization:

Users

Username(TEXT)	Password(TEXT)	PfP Number (INT)
<username1>	<password1>	<0 - n>
<username2>	<password2>	<0 - n>

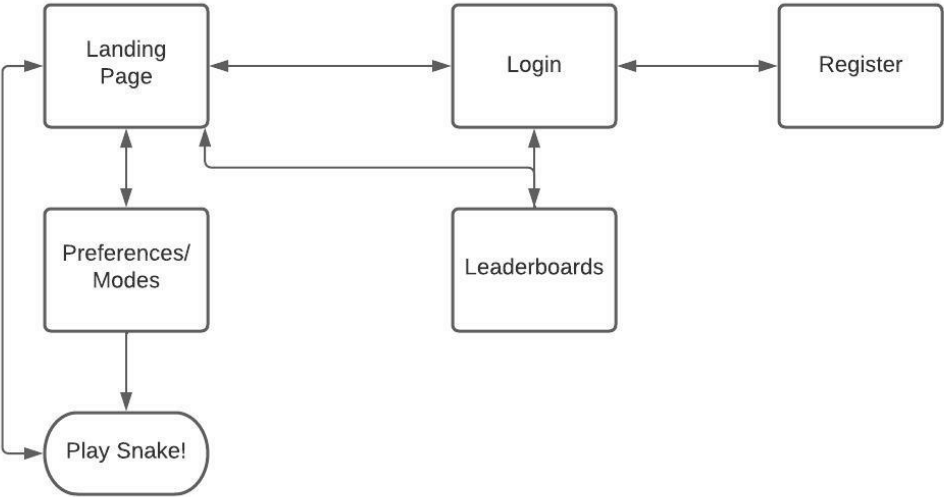
...
-----	-----	-----

Profile picture number corresponds to each available profile picture

Scores

Username(TEXT)	Highscore Mode1(INT)	Highscore Mode2(INT)	Highscore ModeEtc...(INT)
<username1>	<0-n>	<0-n>	<0-n>
<username2>	<0-n>	<0-n>	<0-n>
...

Site Map: [link](#)



Templates:

Home / Landing

Welcome, <User>! Login/Logout

Snake++

PLAY

Your HighScore: <N>

Mode: <Basic> Speed: <Medium> Map size: <Medium>

Change Settings

Game

Current Score: <18> HighScore: <18>

Snake ++ | <User> | Pause | Quit

Setting

Change Your Settings

Speed: Slow Medium Fast Insane

Map Size: Small Medium Large

Mode:

Basic Obstacles Border wrap

Peace Poison apple Boss

PLAY

Leaderboard

Leaderboard

Mode: Basic ▼

1. <User>: 1000,000

2. <User 2>: 750

3. <User 3>: 3

4. You: 2

5. <User 5>: 1

Back to Home

Login

Login

Username: User1

Password:

Login

Don't have an account? Register

Register

Register

Username: User1

Password:

Repeat Password:

Register

Frontend Framework:

We plan on using Bootstrap, utilizing the easy to import javascript and the clean and easy to use classes in order to make sure the information is properly laid out.

Good Bootstrap Features:

- Containers!
 - Put everything in containers: they're magical.
- Alerts are a nice type of container that (may not be used for intended purpose here but) look very nice and allow access to other colors for containers.
 - Some listed items on main pages will likely be in Alerts, as well as actual alerts, like login errors and such.
- Dropdowns are easy to create, look nice and are user-friendly.
 - Settings and/or modes perhaps?
- Buttons have a lot of support classes for aesthetics and links can be made to look like buttons
 - Navigation between pages
 - Form submission
- Display headings are special text classes that allow for much larger and better headings
 - Title, TNPG or other large headings
- Forms in bootstrap are supported nicely and come with good-looking default classes.

- Settings/modes and other submissions
- Tables and auto formatted (but controllable) columns
 - Positioning different page elements in rows
- Lots of text utility like **sizing**, s p a c i n g, margins , **font weight** and more.

Task Division:

Ivan Mijacika (PM): frontend devo

- Frontend: jinja2 and html
- Flask and website main functions
- Leaderboards

Noakai Aronesty: frontend devo

- Frontend: bootstrap, css
- Authentication functions
- Javascript and snake game

Justin Morrill: backend devo

- SQLite and db
- Leaderboards
- Flask and website utility functions

Qina Liu: backend devo

- SQLite and db
- Javascript and snake game
- Settings/game mode differentiation

TARGET SHIP DATE: Monday, 3/21/22

Notes:

Mode ideas:

- Ouroboros: your goal is to eat your own tail. The longer you are when you do so the more points it is worth
- <https://www.quora.com/What-are-the-most-interesting-variants-of-the-game-of-Snake>