Islands - Project Design Document

The Project

A web application for the puzzle game <u>Nurikabe</u>, where given an n x n grid with a few numbers, the player must determine which squares are designated land and water. The user will do this using a few rules: the numbers represented the area of the land it is a part of, all water on the board must be connected, and no area of the water can be *2 x 2* or greater. In the application, the user will be able to use an interactive tutorial, play various sizes (n = 5, 7, 10, 12) of game, and keep track of some of their stats - games solved, number of puzzles solved.

Requirements

From the user's end, this web app should be easy to use, aesthetic, and fun to play. These all tie into design choices on the frontend portion.

As a learning tool and project on my resume/GitHub, this project should have front- and backend elements, described below.

Out of Scope

This project will not be collecting any identifiable user information. The user can log in by typing in a username and the user stats will be readily available, essentially open for anyone to access.

Architecture

I. User

- interacts with web application which has a user interface displaying React components

2. Web Application

- makes API requests to database to retrieve games, game metrics, user metrics

- web application will write and update game/user metrics upon completion of games or input of username

3. Database

- tables of puzzle games (start/solution), game metrics, user metrics

High Level Overview

Not really sure how to do this part :\

API Design

Puzzle - Retrieve a random game of selected size, game metrics
        Method and Endpoint: GET, from: /puzzles/{size}/{randint}
        Response: 'id' (int): Puzzle ID
                      'startboard' (array of array of int/str): Starting grid of puzzle
                      'solutionboard' (array of array of int): Solution of puzzle
        Method and Endpoint: GET, from: /game_metrics/{puzzle_id}
        Response: 'avg_time' (int): Average solve time in minutes
                      'solve_total' (int): Number of times solved

Puzzle Submission - Submits game and user metrics when puzzle is solved
        Method and Endpoint: PUT, to: /game_metrics/{puzzle_idx}
        Request Body:'solve_total' (int): Number of times solved + 1
                    'avg_time' (int): (prev avg time * solve_total+current_time)/(solve_total+1)
        If user:
        Method and Endpoint: GET then PUT, to: /user/{user_id}/{size}
        Response: 'solves_x' (int): Number of puzzles of size *x* solved, increases by 1

User Authentication - Find user and user_id, if not, create
        Method and Endpoint: GET, from: /user
        Response: 'user_id' (int): Unique user id number
            or None
        If not user:
        Method and Endpoint: POST, to: /user
        Request Body: 'user_id' (int): Unique user id number
                    'solves_x' (int): 0 for all puzzle sizes *x*

User Data - Retrieve user's game metrics
        Method and Endpoint: GET, from: /user/{user_id}
        Response: 'solves_x' (int): Number of puzzles of each size solved

Data Storage and Model

Relational Database (SQLite) - not too much data will be processed here…
Tables:
        Puzzles: 'id' (int), 'size' (int), 'startboard' (array), 'solutionboard' (array)
        Game_metrics: 'id' (int), 'solve_total' (int), 'avg_time' (int)
        User: 'username' (str), 'user_id' (int), 'solves_x' (int) (x = 5, 7, 10, 12)

Application/Component Level Design

Main Menu Page
        Buttons: Play, Rules, User, About

Game Page
>Select from available sizes: 5, 7, 10, 12
>Random game is retrieved from database and displayed, timer begins
>Game menu has puzzle in the middle, timer top left, and average solve time, solve numbers top right and buttons below: Home, Check Progress, Restart, New Game, Rules, Hint, Undo

Rules Page
>Interactive 'game-like' explanation of rules:
>>-number on grid = area of island it is on
>>-river must be all connected on board
>>-no area of river can by 2x2 or greater

User Page
>Display: username, total puzzles solved, number of puzzles solved for 5, 7, 10, 12 sizes

About Page
>Game was made in React, based on Nurikabe by Nikoli.

Future Improvements

-daily puzzles - users can access these puzzles of the day with leaderboard of top 10 user times
-social aspect - wordle things…