

puzzle 15 document

SESAMI

request

1. On page load tiles order should be randomized.
2. The board size (number of columns and rows) should be configurable, and support a
3. rectangular configuration (M x N board size)
4. The tile size should be configurable.
5. Must be performant for a large grid (e.g above 30×30).
6. Should support adding multiple boards.
7. Validity checks. Every board should be solvable.
8. A success message should be displayed when all the elements are reordered properly.
9. Tiles should be animated and slide to the designated area (Nice to have).

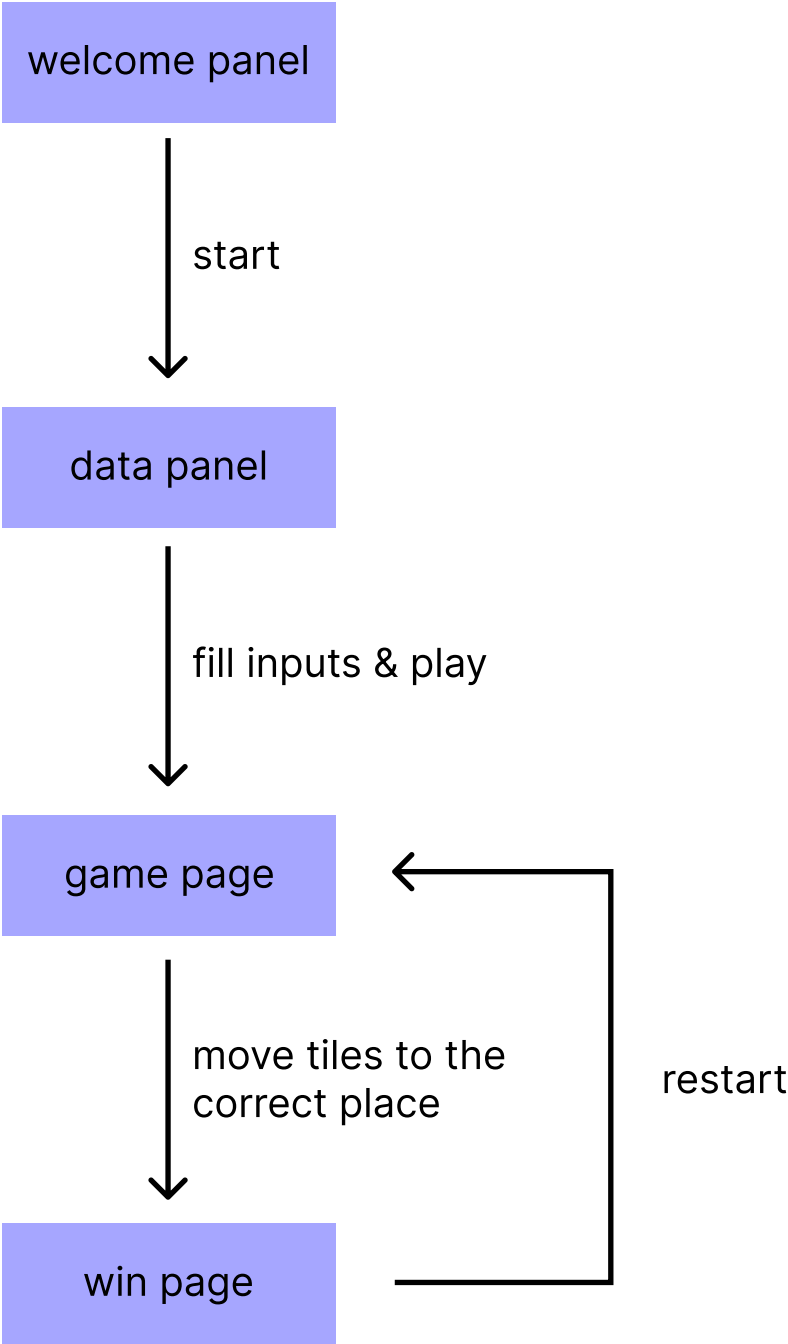
UI Requirements

Please implement the simple UI wireframes outlined below. Minimal styling is acceptable.

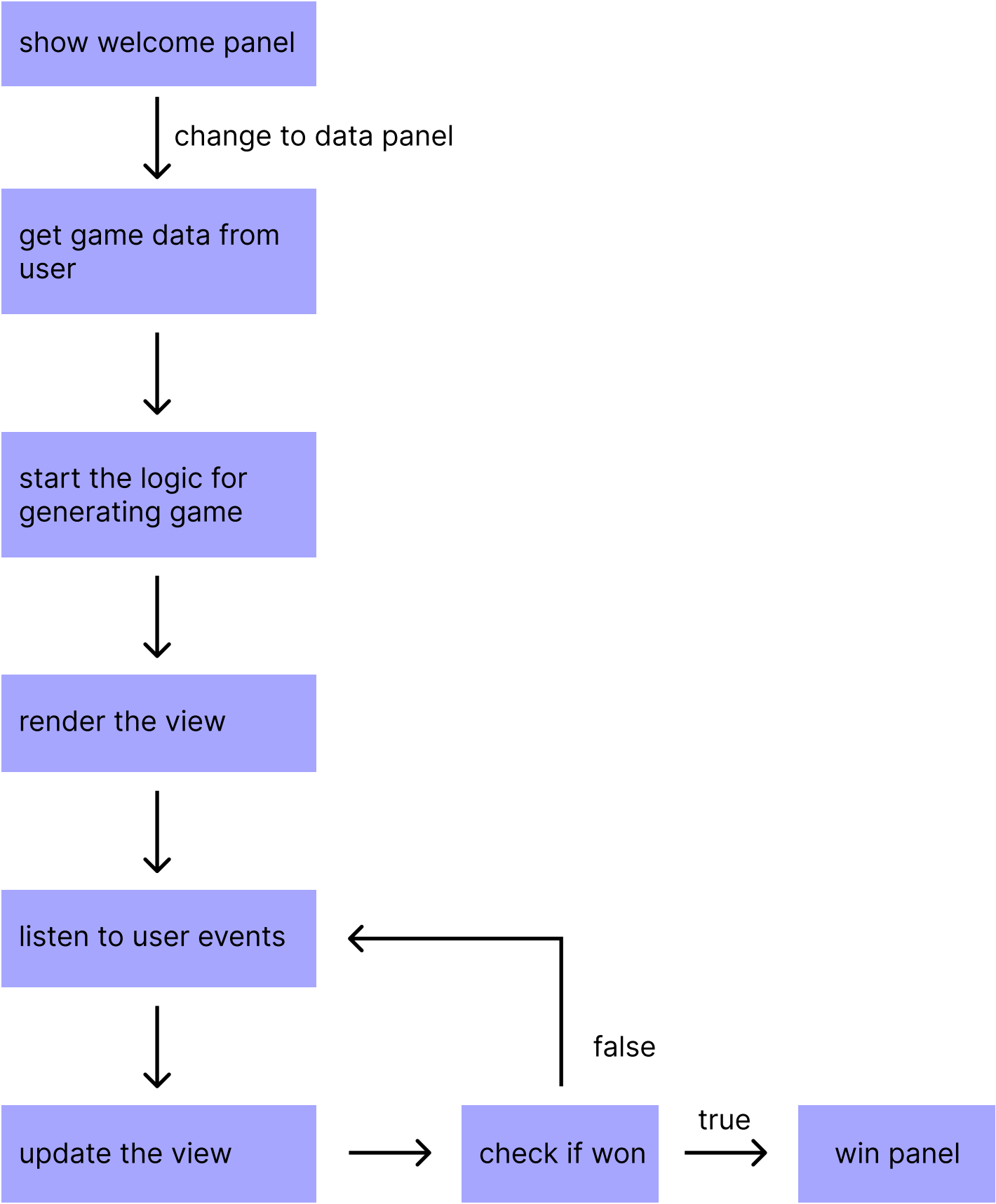
my takes

1. goal is to show how i analyze the request and how i find the solution (problem solving skill)
2. UI is not the goal but it should look good
3. code should be flexible for further development and maintenance
4. code should handle large calculation without no drop in performance
5. "Should support adding multiple boards." , have no idea what that means !
6. game must be 100% solvable , better to check before generating for user
7. game must have a stable main loop , start → gameplay → win , repeat
8. animation is nice
9. no animation library only css animation
10. "Assume that your code will be served from a simple web server" , logic should be separated so it can be run from server without need of browser
11. vanilla javascript !
12. "Follows a module pattern" , no need to mention
13. "UI remains responsive" , base on "UI Requirements" i get this meaning that its not suppose to be fancy but it must stay functional over any device screen

user flow



code flow



code structure

index.html

style.css

global.js

panels.js

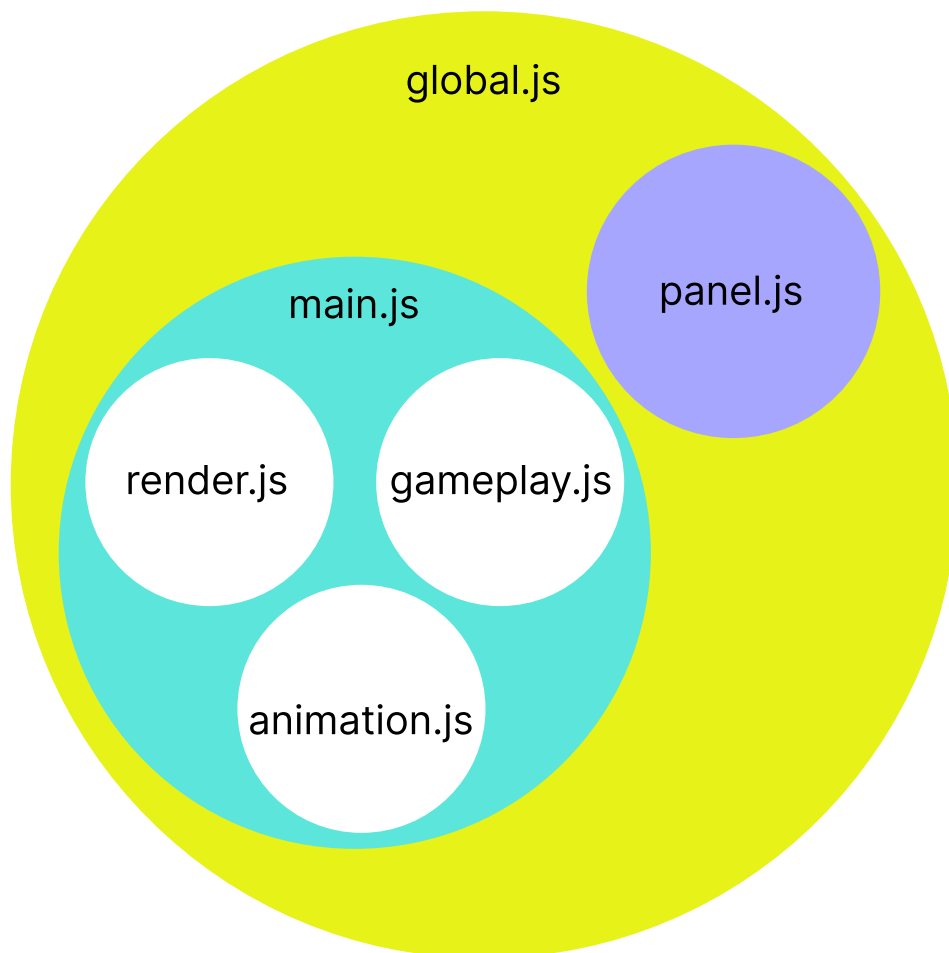
main.js

render.js

gamePlay.js

animation.js

- global.js : this file have the role of a state management , keep every variable that is used over all the app
- panels.js : handle the user interaction with UI , this is the start of the code and this code will give the order to generate the game
- main.js : start the rendering of the game (render.js) and then start listening to user events (gamePlay.js)
- render.js : includes functions to generate , validate and render the puzzle
- gamePlay.js : includes functions to listen , move tiles , update game status and check if user won
- animation.js : handle any animation and changes on UI by logic



data flow

