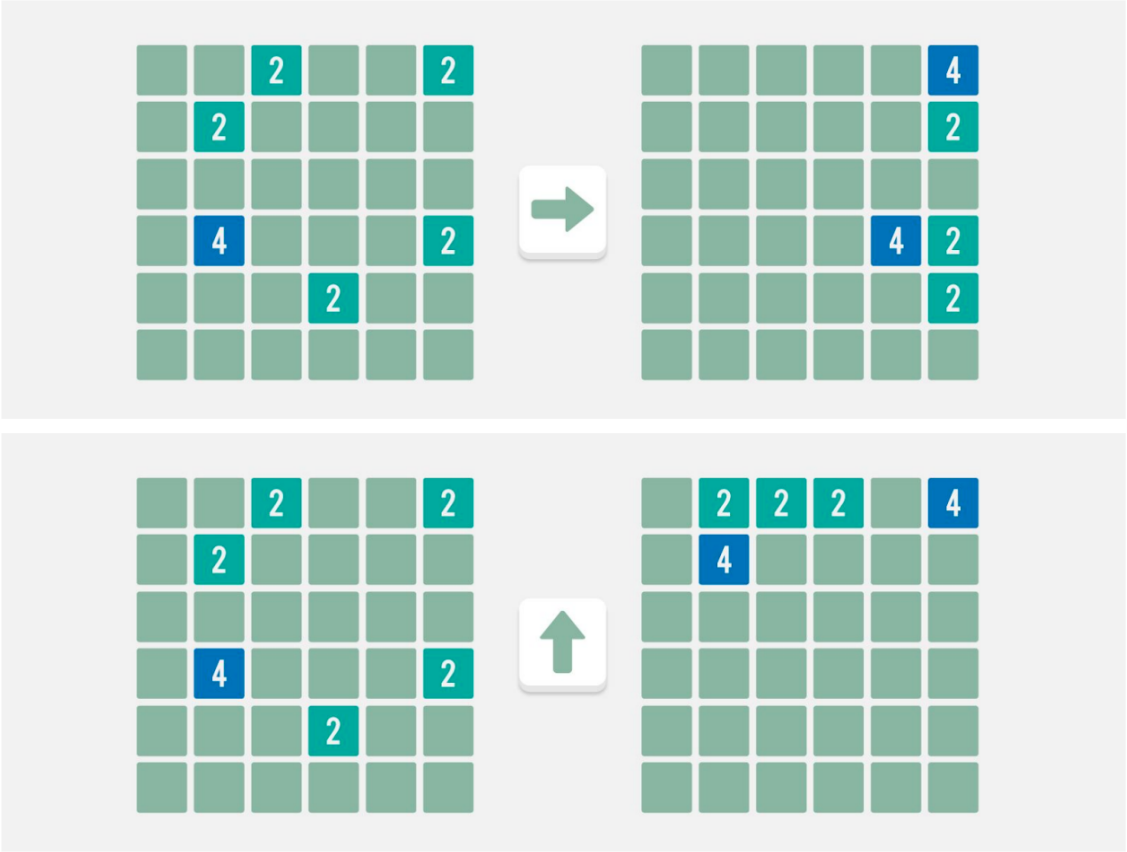
2048

Most people are familiar with the game 2048. You can find many implementations of it online, as well as many different variations.



Specifications

- The grid consists of 6x6 tiles
- At the beginning of a game the grid is empty, except for one tile of value 2 placed at
  random.
- The user can slide the tiles either up, down, left or right
- After each slide a new tile with value 1 will appear in a random free space.
  - If there is no free space to put the new tile the game is lost

- During the slide, tiles of equal values pushed into each other will merge into a new
  tile with the combined value. 2 + 2 = 4
  - If there are 3 values next to each other, e.g. 2 2 2, and the player slides right,
    the values closest to the wall should merge first resulting in 2 4.
- If any tile reaches the value 2048 the game is won.

Main task

Implement a frontend application allowing you to play this game. We use Vue as our frontend framework of choice, but you can use anything you prefer. Just to give the general idea of the scope of your choices, it could be

- Vue, React, Angular, or any other FE framework
- Vanilla JS with Canvas, or SVG
- WebGL

Anything you feel comfortable with is fine.

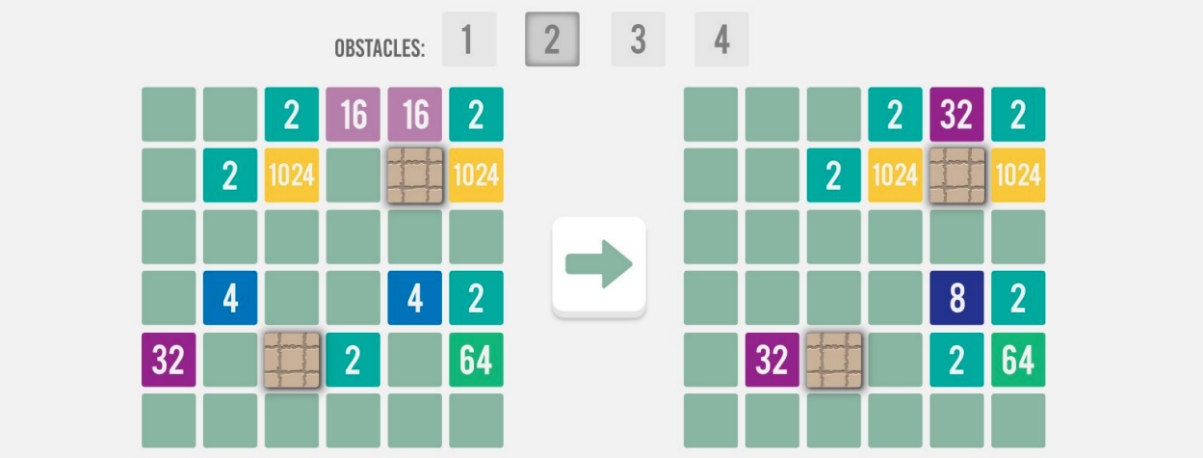
That being said, if you opt for Vue because that's what we use, we recommend using Vue 3 + Vite and setting up your project using instructions at https://vuejs.org/guide/quick-start.html#with-build-tools

This won't give you an advantage over other submissions. What we care most about is you being able to explain why you chose the tools you did.

Bonus tasks

None of these are needed, but doing one will make your submission look much more impressive

1. Add randomly placed obstacles in the grid (the user should be able to choose the
   number of obstacles at the start of the game). An obstacle acts as a wall and will not
   move



2. A backend with support for multiplayer via websockets (this one is potentially big, only do it if you’re comfortable with it)
3. Custom grid sizes

Judgement criteria

We care about well structured, maintainable code, specification covered as much as possible, with the ability to justify your choices and decisions.

Polish is great and we want to see it, but the ability to find a balance between time spent and output is much more highly valued.

You can submit your solution as a zip file with your first and last name in it, or as a link to a github repository.

Make sure the source code has instructions on how to run it. Ideally, it’s with a single command. If you have the ability to deploy it, that’s great, but we would like to be able to run it locally for potential debugging purposes.