

FrontEnd and Dashboard Developer

Exercise

The candidate has to build a working replica of the 2048 game. It can be console-based or with a UI. We will provide a link to the original game so the candidate can download and see the game.

The candidate needs to replicate 2048 in any platform of their choosing.

Evaluation Criteria

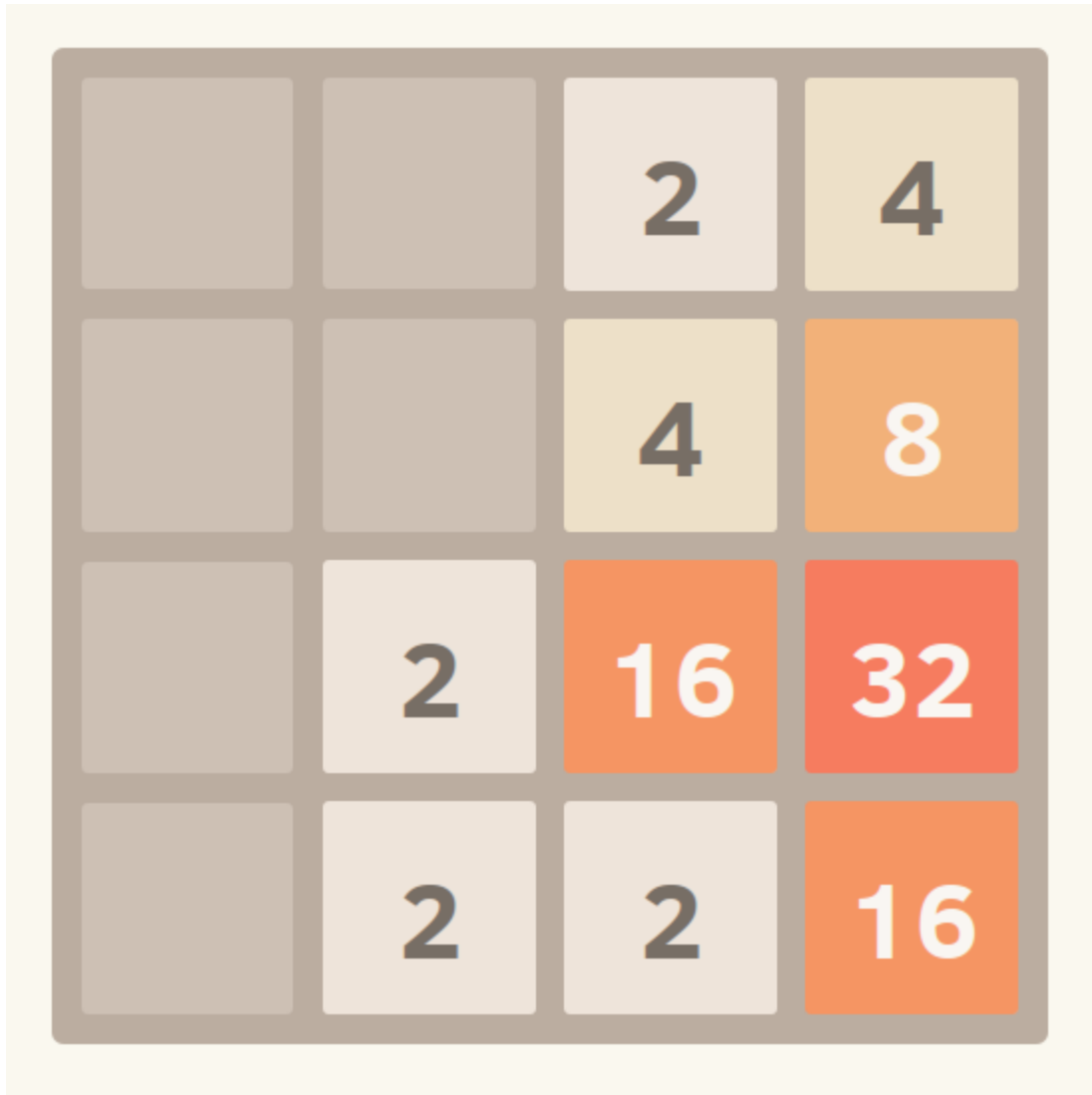
The results will be evaluated on the basis of -

- Understanding the problem.
- Plan to tackle.
- Coding Standards
- Readable Code

Expected output

A Working 2048 game.

2048



2048 is played on a gray 4×4 grid, with numbered tiles that slide when a player moves them using the four arrow keys. Every turn, a new tile will randomly appear in an empty spot on the board with a value of either 2 or 4. Tiles slide as far as possible in the chosen direction until they are stopped by either another tile or the edge of the grid. If two tiles of the same number collide while moving, they will merge into a tile with the total value of the two tiles that collided. The resulting tile cannot merge with another tile again in the same move. If a move causes three consecutive tiles of the same value to slide together, only the two tiles farthest along the

direction of motion will combine. If all four spaces in a row or column are filled with tiles of the same value, a move parallel to that row/column will combine the first two and last two.

Link to game sample : <https://play2048.co/>

Implementation Details

1. Print the 4 x 4 board on each turn in the console and wait for user input. This will initially have two cells populated at random with a 2 or 4.
2. User will input 1, 2, 3, 4 for left, right, up and down movements
3. Program will then merge all the tiles in given direction and show the latest sums according to rules mentioned above
4. Next it should select a random empty location in tiles and place a 2 or a 4
5. Repeat steps 1 - 4 till one of the cell reaches 2048

Language to be used

Use either Python, Javascript or Java for implementation

Coding Criteria :

- The history of the coding challenge must be shown on github along with relevant git commits, branches etc.
- NO UI Needed! Console game works

Expected output

Working code with outputs of each stage printed on the console. Code should be committed to Github and a link to be shared. Do not share zipped files.

Discussion Objectives

- Design principles used
- Thoughts / Probable solutions / Problems faced while designing the solution
- Code walkthrough
- Scope of making this an 8x8 from 4x4
- Change from 2048 to 4096 as end number

(Do not implement these above two points, but have a way to do this)