

# Conway's Game of Life

## prova tecnica

### Premessa

Negli anni abbiamo notato che uno dei migliori modi per valutare le competenze tecniche dei nostri candidati è fargli risolvere un esercizio di programmazione. In questo modo siamo in grado di capire il livello, i punti di forza e i punti migliorabili di ogni candidato.

### Il compito

Consiste nella risoluzione di un esercizio che ha come tema il Game of Life di Conway.

### Elementi di valutazione

**Qualità del codice scritto.**

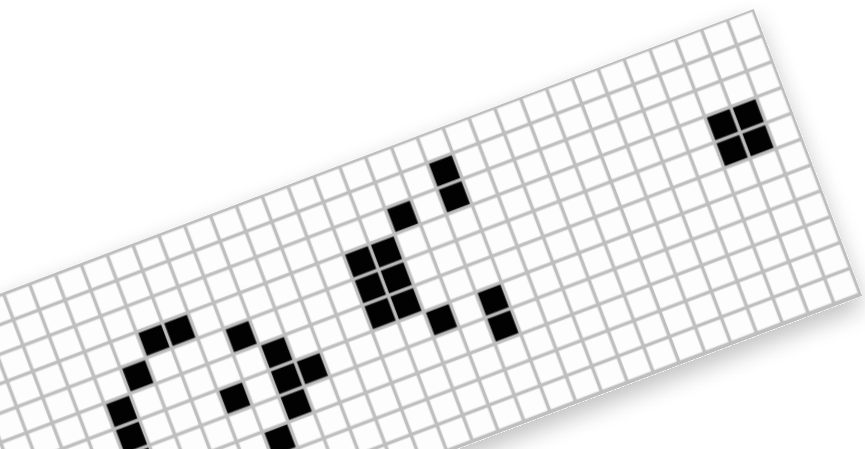
**Presenza di commenti nel codice** (valutiamo positivamente commenti nel codice. Lavoriamo in teams sui progetti ed è importante che il codice sia facilmente intelleggibile da parte di tutti i teammates).

**Velocità di esecuzione** in rapporto con la qualità della delivery (non è una gara, diamo più di una settimana di tempo perchè sappiamo che molti svolgono il compito al di fuori dell'orario lavorativo e universitario).

Front-End

 React + TypeScript

extendi



# Game Of Life instructions

## Problem description

Given a input generation the goal of this kata is to calculate the next generation. The world consists of a two dimensional grid of cells, where each cell is either dead or alive. For the purpose of this kata let's assume that the grid is finite and no life can exist off the edges.

Given a cell we define its eight neighbours as the cells that are horizontally, vertically, or diagonally adjacent.

When calculating the next generation you should follow these rules:

- Any live cell with fewer than two live neighbours dies.
- Any live cell with two or three live neighbours lives on to the next generation.
- Any live cell with more than three live neighbours dies.
- Any dead cell with exactly three live neighbours becomes a live cell.

## Implementation

The initial state (the current generation) will be provided via a text file that specifies:

- the current generation number
- the grid size
- the population state (\* represents a live cell, . represents a dead cell)

In the input file example below we can see an input file specifying the third generation on a 4 by 8 grid.

Generation 3:  
4 8

```
. . . . * . . .  
. . . . * . . .  
. . . * * . . .  
. . . . . . . .
```



Generation 4:  
4 8

```
. . . * * . . .  
. . . * * . . .  
. . . * * . . .  
. . . . . . . .
```

The output should display the correct number of the resulting generation, the grid size and the current population state.

## Notes

- Focus on writing the best code you can produce.
- When ready share with us a git repo (github/gitlab) with your solution.
- Provide a web application.

[See an example](#)