Game Of Life

Introduction

In this exercise you have to implement **Conway's Game of Life** in TypeScript. Before starting with this exercise, make yourself familiar with *Game of Life* by reading the **Wikipedia article**.

Setup

- Download the attached skeleton project `game-of-life.zip` and extract it.
- This contains a `app.ts` file which you can use to solve this exercise.
- Make sure to restore the required Node packages using `npm install`.
- Run the simulation using `npm start`.

Requirements

- Your *Game of Life* grid must have a size of 200 by 200 cells.
- Initially, random 3% of all grid cells must be alive. All other cells must initially be dead.
- Each cell must be represented on the screen by a black square with side length of 4 pixel.
- Write your code so that it is very simple to experiment with other grid sizes, pixel sizes and initial ratios of alive cells.
- Implement the *Game of Life* algorithm with the classic rules:
  - Any live cell with fewer than two live neighbors dies, as if caused by underpopulation.
  - Any live cell with two or three live neighbors lives on to the next generation.
  - Any live cell with more than three live neighbors dies, as if by overpopulation.
  - Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.
- Your *Game of Life* algorithm must start immediately after the HTML page has been loaded. - It must run continuously in an endless loop until the user closes the browser or reloads the page.

Here is a screenshot showing how the game board must finally look like when the game is running:

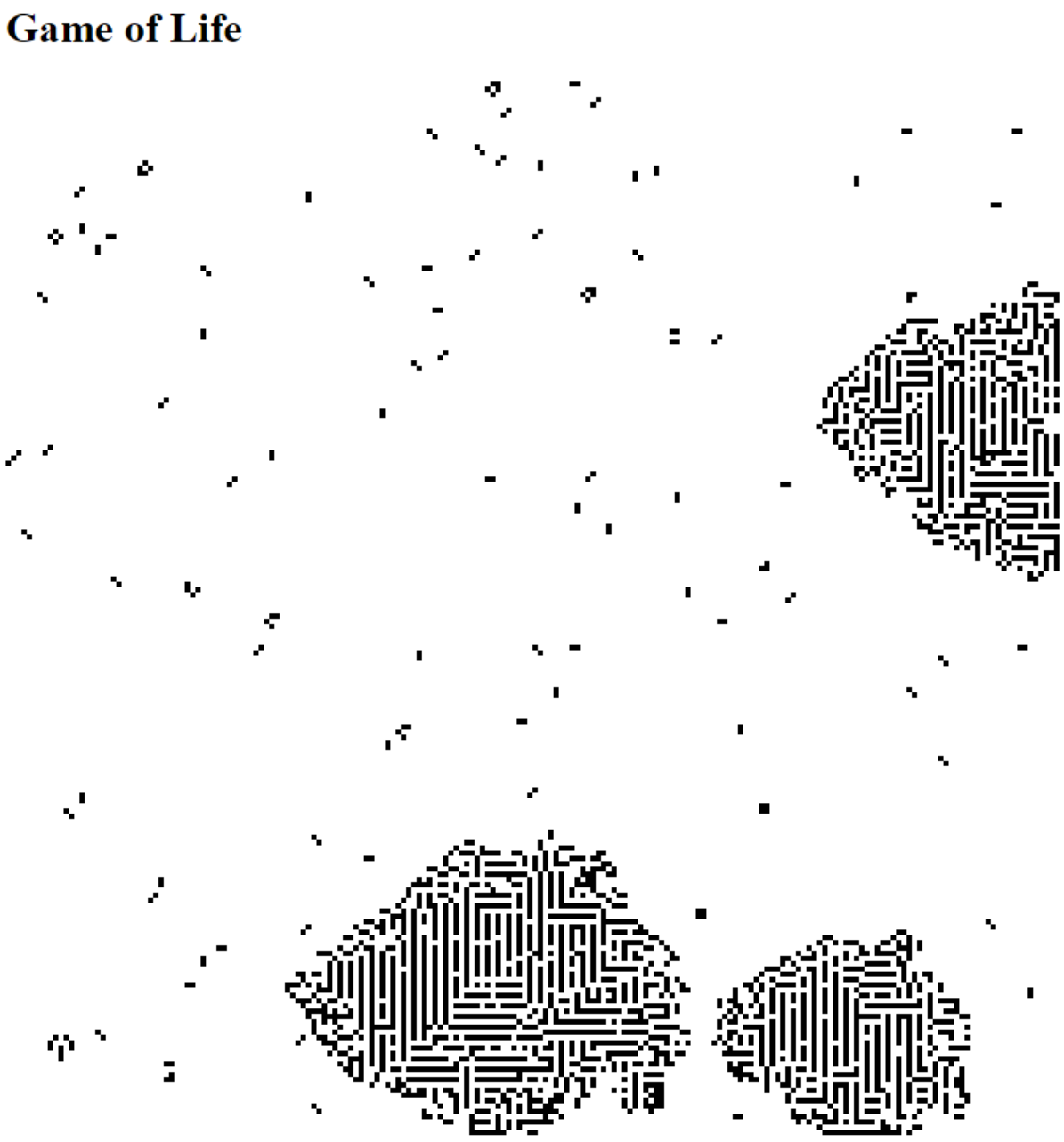

Figure 1: Game of Life board