Refactor Playgameoflife.live - Utilize JSXGraph the React Way

Leslie Wong, a web developer
Shenzhen, China
79917148leslie@gmail.com

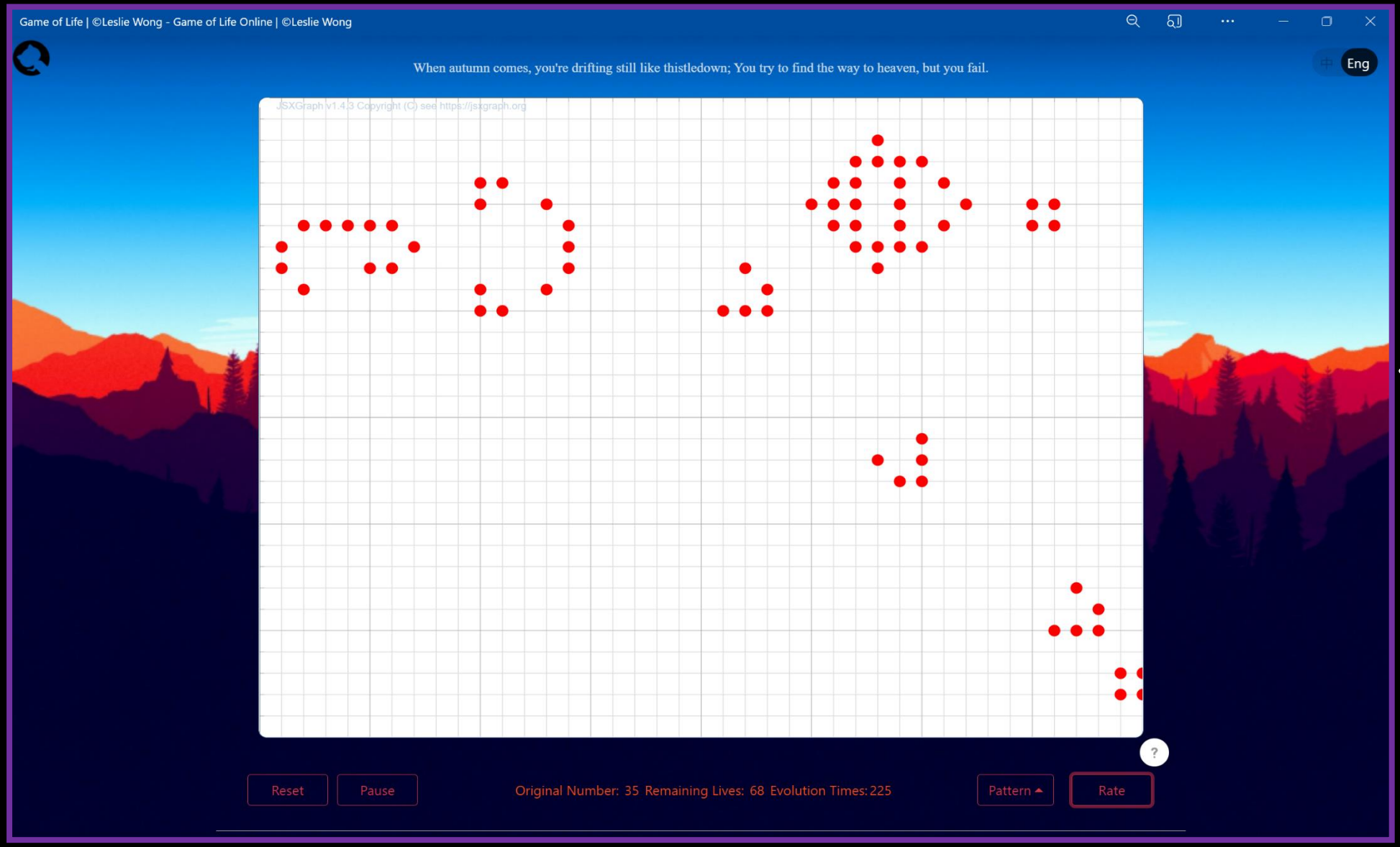
Agenda

- Overview of Playgameoflife.live
- Architecture migration
- Refactor the usage of JSXGraph
- Implementation to decode/encode RLE

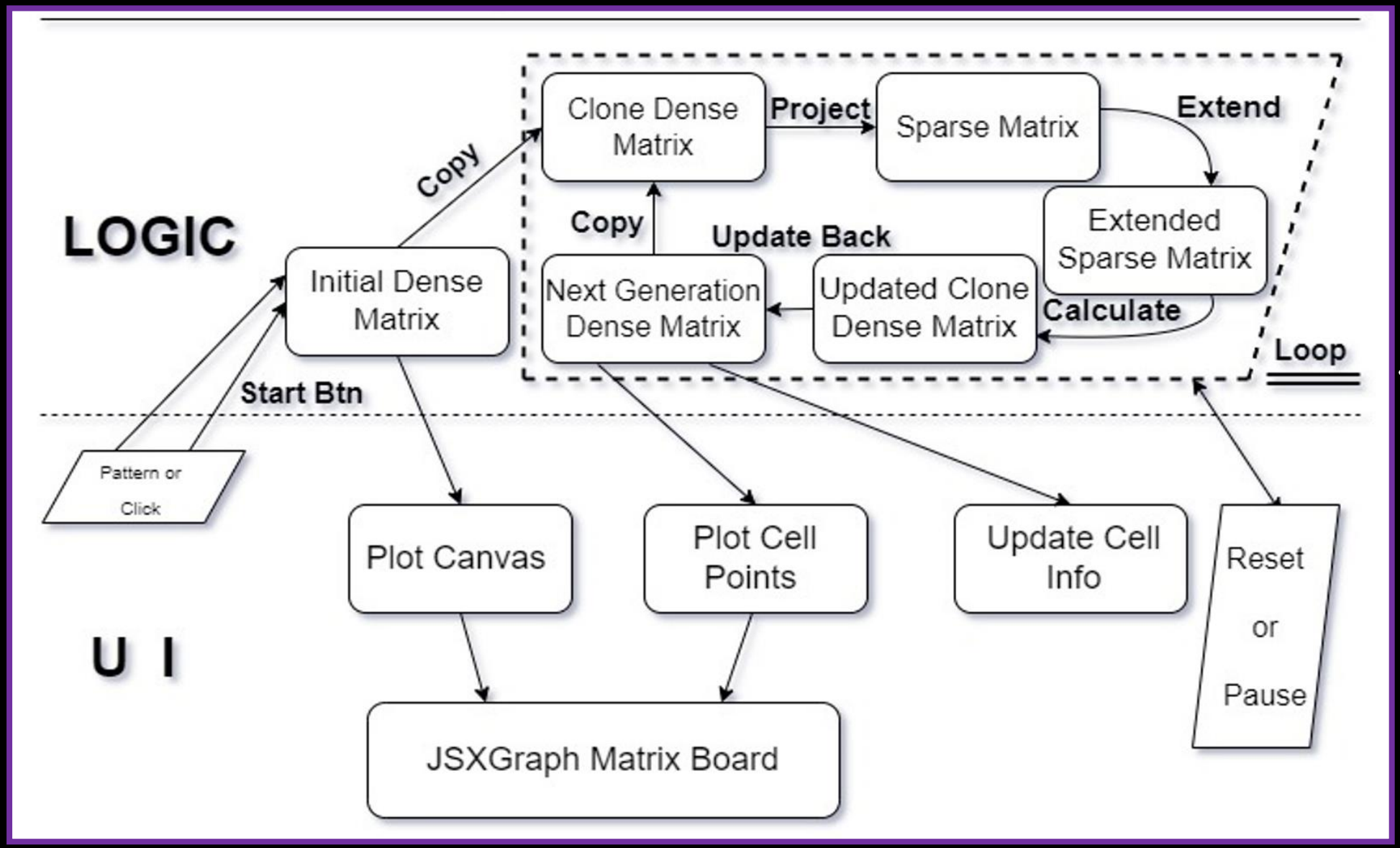
1

Overview of Playgameoflife.live

Screenshot


Game of Life | ©Leslie Wong - Game of Life Online | ©Leslie Wong
When autumn comes, you're drifting still like thistledown; You try to find the way to heaven, but you fail.
Eng
JSXGraph v1.4.3 Copyright (C) see https://jsxgraph.org
Reset
Pause
Original Number: 35 Remaining Lives: 68 Evolution Times: 225
Pattern
Rate

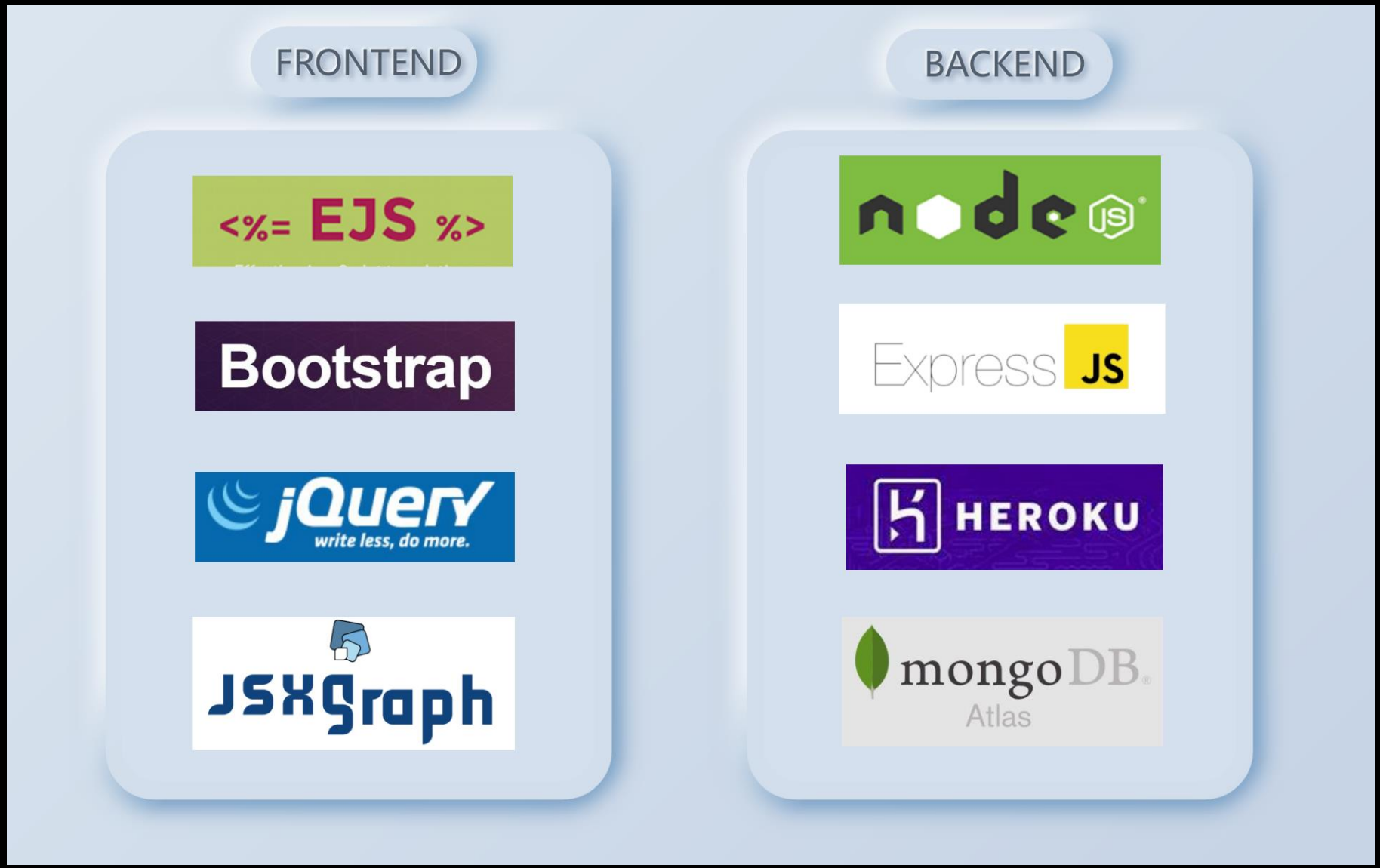
Backbone


LOGIC
Clone Dense Matrix
Project
Sparse Matrix
Extend
Copy
Copy
Update Back
Extended Sparse Matrix
Initial Dense Matrix
Next Generation Dense Matrix
Updated Clone Dense Matrix
Calculate
Loop
Start Btn
Pattern or Click
Plot Canvas
Plot Cell Points
Update Cell Info
Reset or Pause
U I
JSXGraph Matrix Board

2

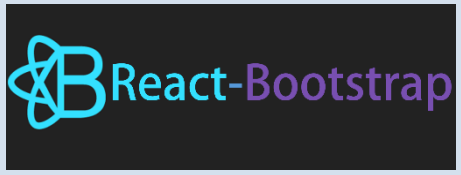
Architecture migration

Previous Architecture



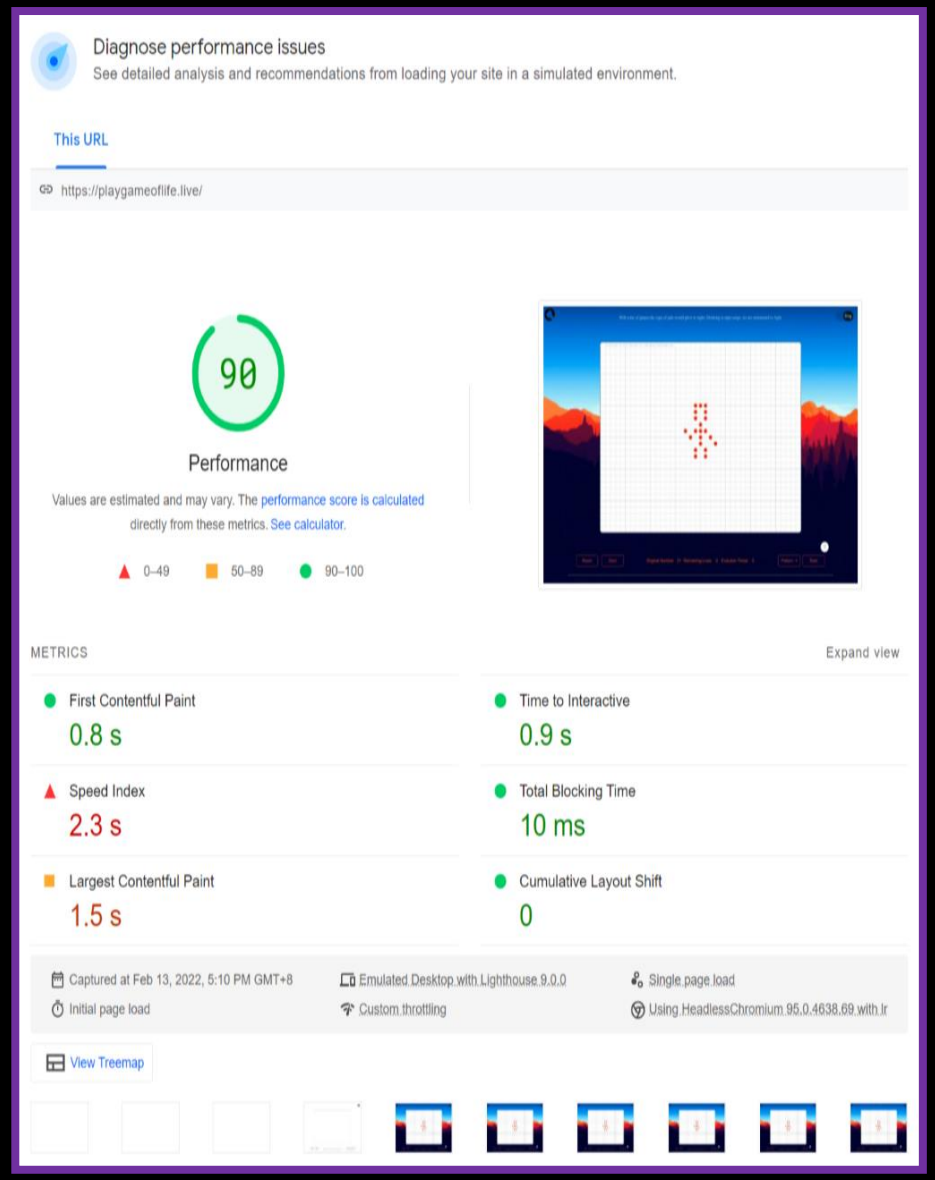
Current Architecture

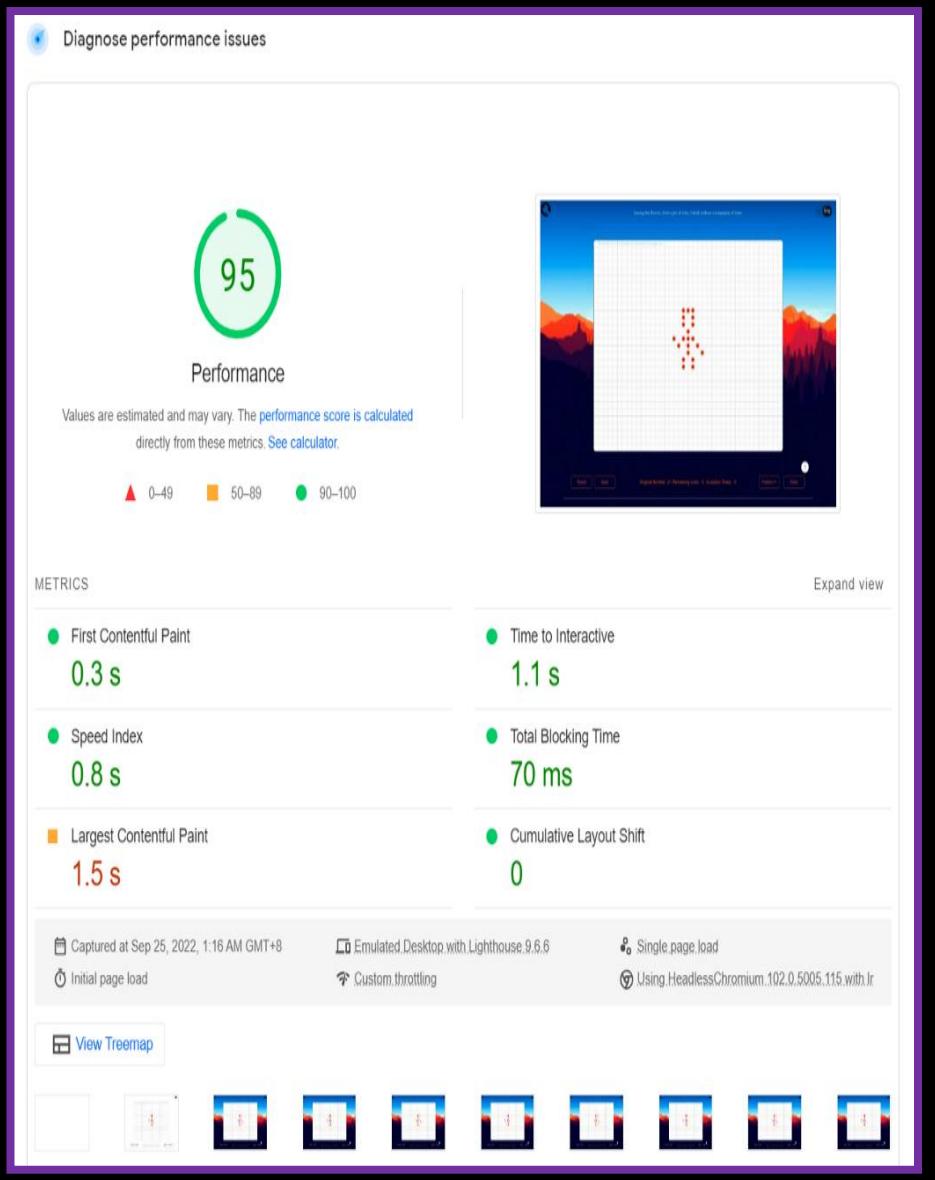
FRONTEND



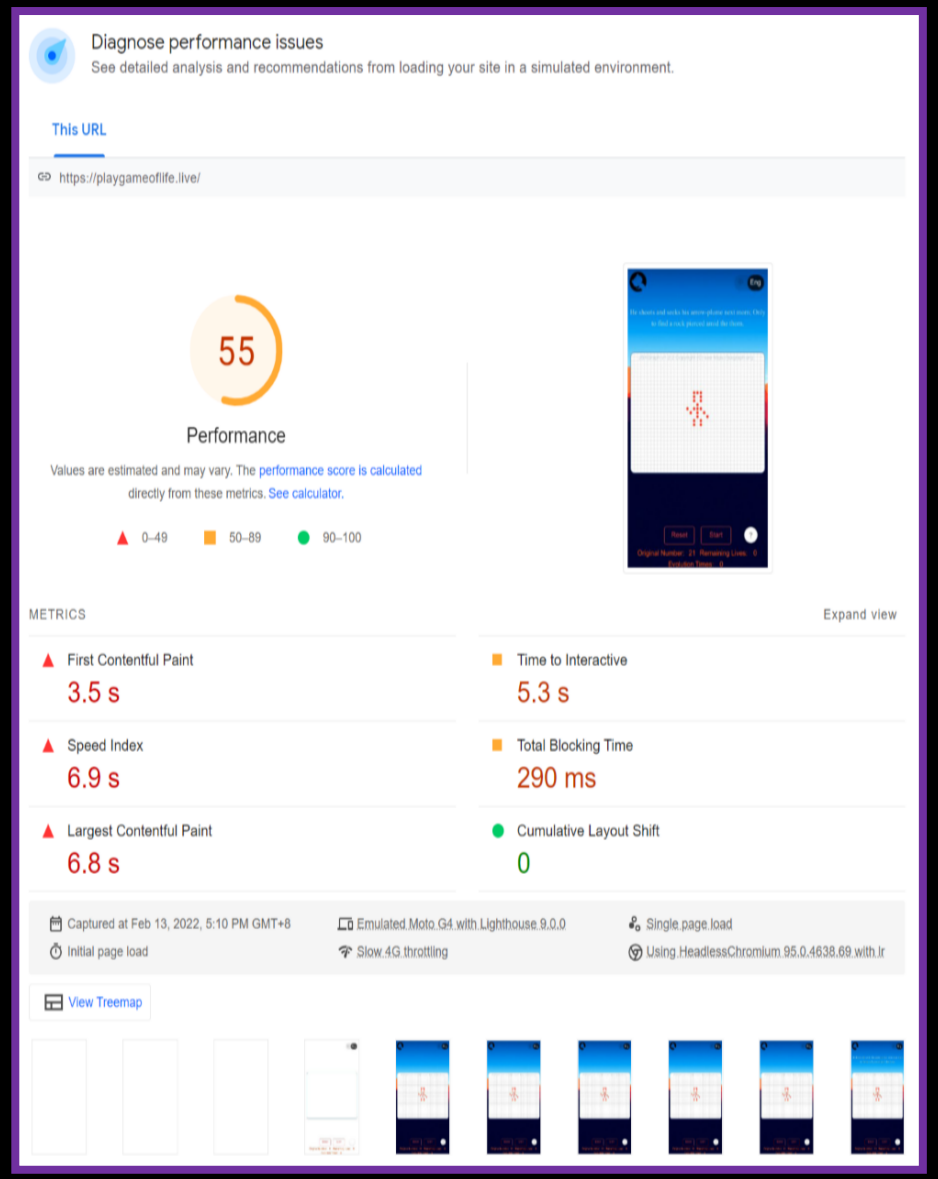
BACKEND

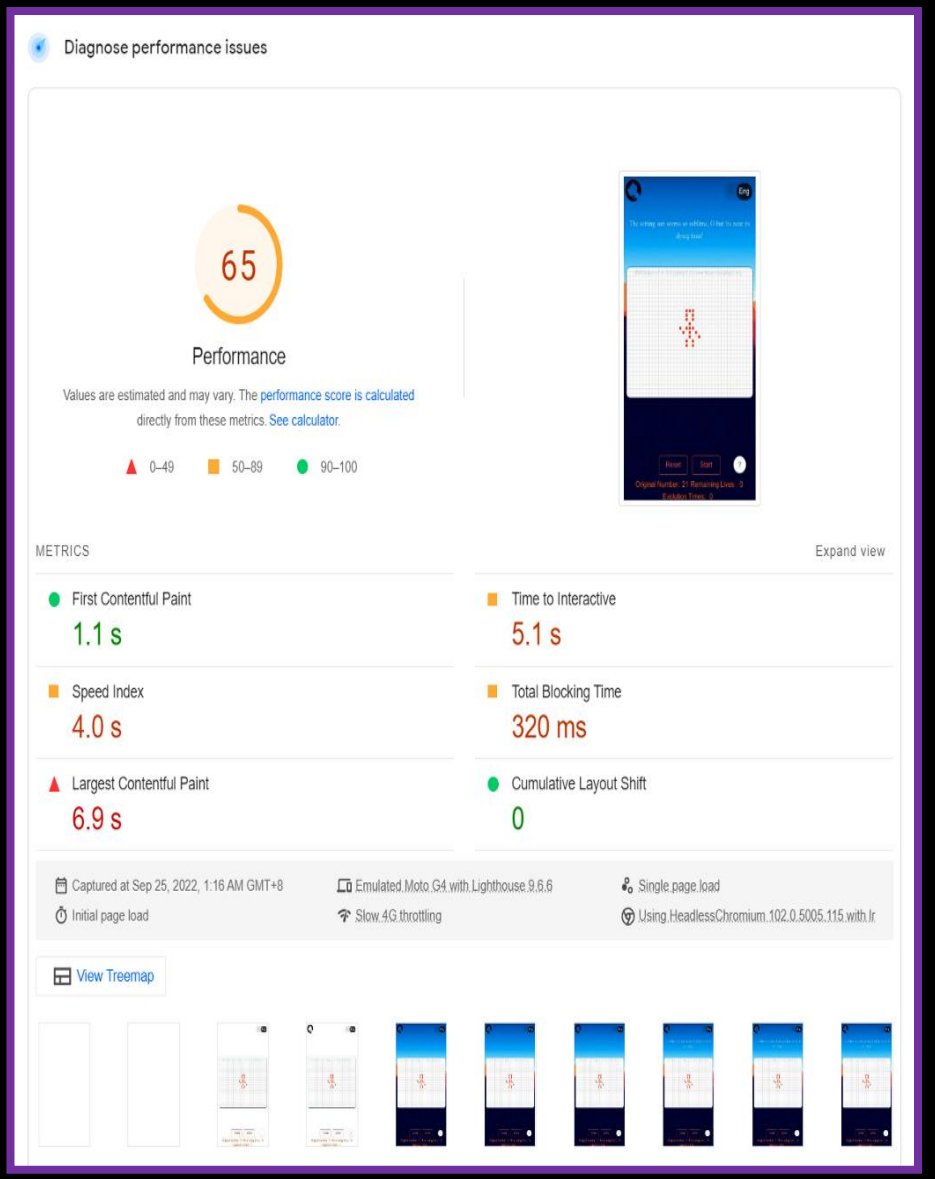
Outcome (Desktop)





Outcome (Mobile)





3

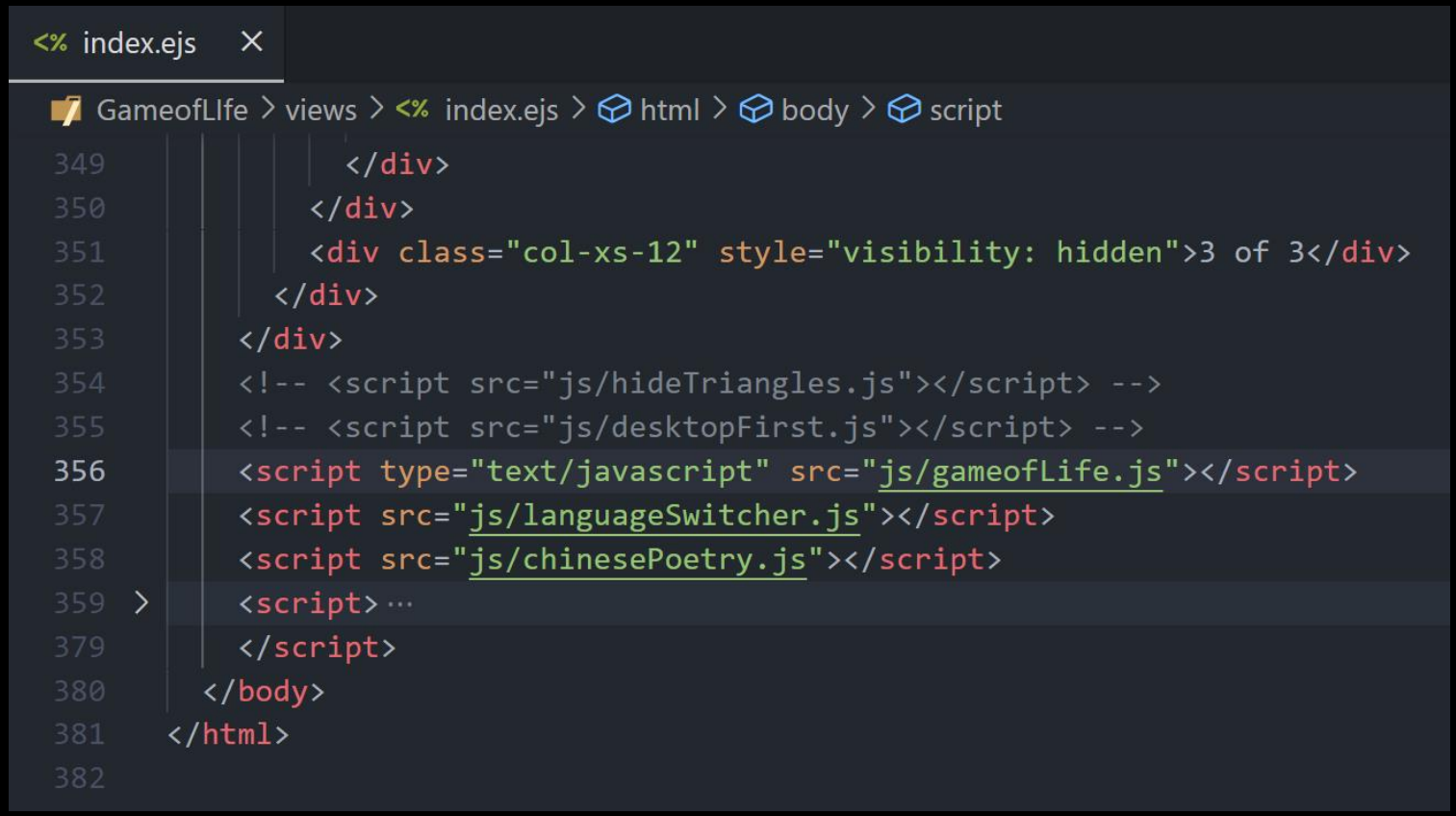
Refactor the usage of JSXGraph

<%- index.ejs JS gameofLife.js ×

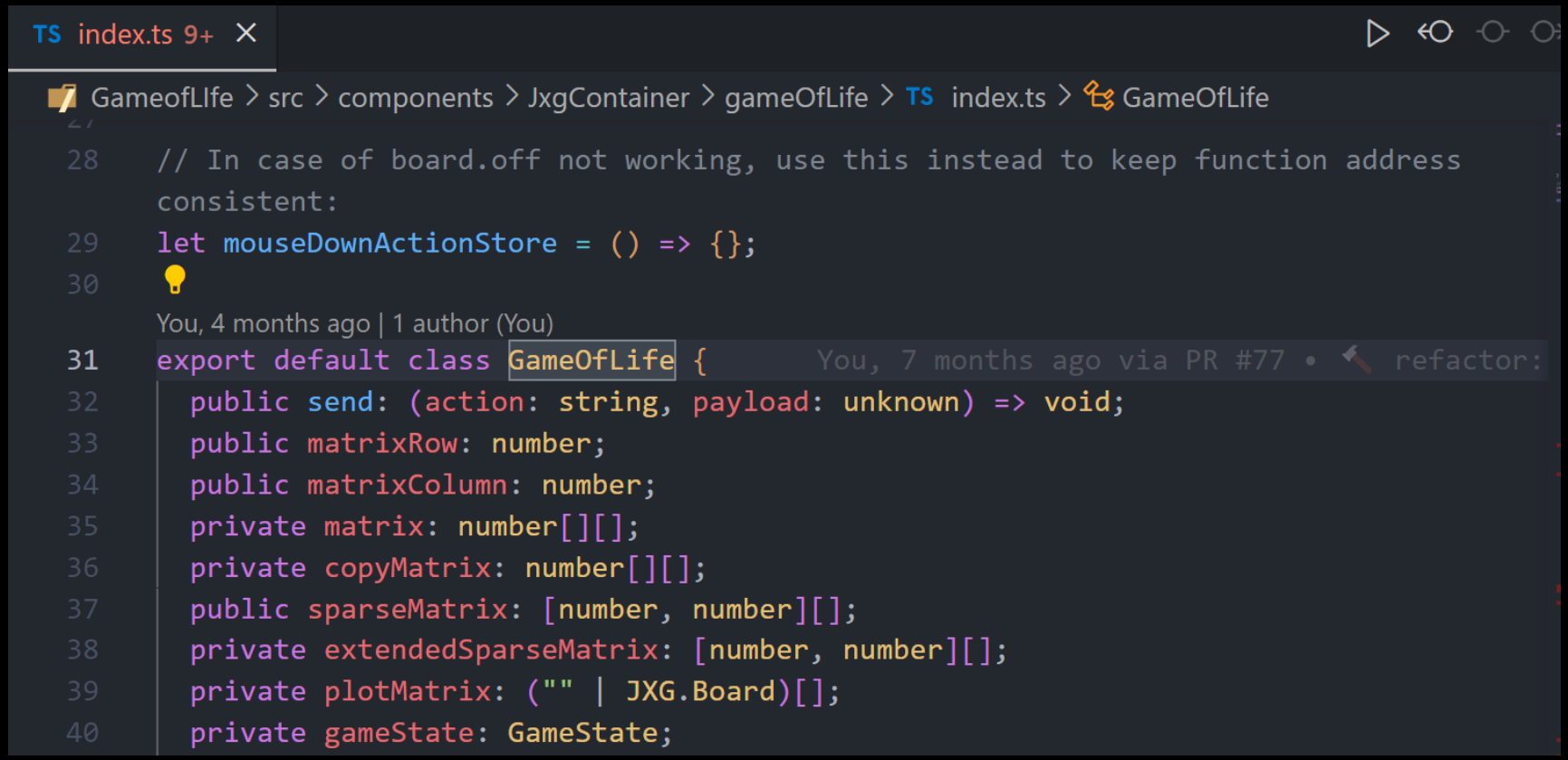
GameofLIfe > public > js > JS gameofLife.js > ...

```
10
11  var matrix = [];
12  // var initialPlotMatrix = [];
13  var copyMatrix = [];
14  var sparseMatrix = [];
15  var extendedSparseMatrix = [];
16  var plotMatrix = [];
17  var matrixRow, matrixColumn;
18  var start = document.getElementsByClassName("start")[0];
19
20  // var stop = document.getElementsByClassName("stop")[0];
21  var random = document.getElementById("random");
22  var glider = document.getElementById("glider");
23  var smallexploder = document.getElementById("smallexploder");
24  var exploder = document.getElementById("exploder");
25
26  var tencellcolumn = document.getElementById("tencellcolumn");
27
28  var lightweightspaceship = document.getElementById("lightweightspaceship");
29  var tumbler = document.getElementById("tumbler");
```

https://github.com/Leslie-Wong-H/game_of_life/blob/sparse_matrix_version/public/js/gameofLife.js



https://github.com/Leslie-Wong-H/game_of_life/blob/sparse_matrix_version/views/index.ejs



https://github.com/Leslie-Wong-H/game_of_life/blob/master/src/components/JxgContainer/gameOfLife/index.ts

ReactJXGBoard.tsx 9+

GameofLIfe > src > components > JxgContainer > ReactJXGBoard.tsx > ...

```
197
198     // called only after initial render
199     // now that div exists, create new JSXGraph board with it
200     useEffect(() => {
201       // set "send" of XState to GameOfLife so that it can dispatch the
          number-update events
202       const GOL = new GameOfLife(send);
203       GOL.initMatrix();
204       GOL.initBoard();
205       GOL.easterEgg();
206       GOL.resizeThrottlerWrapper();
207       GOL.scrollHandler();
208       setGOLInstance(GOL);
209     }, [""]);
210
211     return <div id="box" className="jxgbox" />;
212   };
213
214   export default ReactJXGBoard;
215
```

https://github.com/Leslie-Wong-H/game_of_life/blob/master/src/components/JxgContainer/ReactJXGBoard.tsx

4

Implementation to decode/encode RLE

https://conwaylife.com/wiki/Run_Length_Encoded

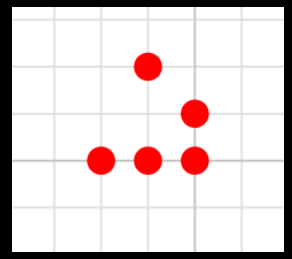
RLEpattern.rle

```
1  #C This is a glider.
2  x = 3, y = 3, rule = B3/S23
3  bo$2bo$3o!
4
```

Encode RLE

```
{result: Array(5)}
  result: Array(5)
    0: (2) [13, 19]
    1: (2) [14, 20]
    2: (2) [15, 18]
    3: (2) [15, 19]
    4: (2) [15, 20]
    length: 5
    [[Prototype]]: Array(0)
  [[Prototype]]: Object
```

Decode RLE



https://github.com/Leslie-Wong-H/game_of_life/blob/master/src/components/JxgContainer/gameOfLife/RLE.ts

Decode RLE

RLEpattern.rle

```
#C This is a glider.
x = 3, y = 3, rule = B3/S23
bo$2bo$3o!
```

```
/**********************/
/* rle-decoder starts */
/**********************/

//on number repeat next letter number of times.
//End by splitting on $ creating multiple lines
let decoded: mediumDecodedPattern = rleString
  .slice(0, -1)
  .replace(/(\d+)(\D)/g, function (match, num) {
    return match.split(num)[1].repeat(num);
  })
  .split("$");

//replace letter 'o' with 1's & b with 0's ie - alive: 1 , dead: 0
decoded = decoded.map((row) => row.replace(/o/g, "1"));
decoded = decoded.map((row) => row.replace(/b/g, "0"));

//for each row split into its own arrow containing single #'s
decoded = decoded.map((row) => [...row.split("")]);

//row length less than specifications add filler 0's
decoded = decoded.map((row) => {
  if (row.length < x) {
    const filler = new Array(x - row.length).fill(0);
    const value = row.concat(filler);
    return value;
  } else {
    return row;
  }
});

//convert all string numbers to type of Number
decoded = decoded.map((row) => row.map((string) => Number(string)));

/**********************/
/*  rle-decoder ends  */
/**********************/
```

['bo', 'bbo', 'ooo']

['01', '001', '111']

[['0', '1'], ['0', '0', '1'], ['1', '1', '1']]

[['0', '1', 0], ['0', '0', '1'], ['1', '1', '1']]

[[0, 1, 0], [0, 0, 1], [1, 1, 1]]

```
/**********************/
/*  rle-decoder ends  */
/**********************/

if (decoded.length === y) {
  let widthOffset = Math.floor((40 - x) / 2);
  let heightOffset = Math.floor((30 - y) / 2);
  widthOffset = widthOffset > 0 ? widthOffset : 0;
  heightOffset = heightOffset > 0 ? heightOffset : 0;
  for (let i = 0; i < y; i++) {
    for (let j = 0; j < x; j++) {
      if (decoded[i][j] === 1) {
        const tempcoord: [number, number] = [
          heightOffset + i,
          widthOffset + j,
        ];
        result.push(tempcoord);
      }
    }
  }
}
```

[[13, 19], [14, 20], [15, 18], [15, 19], [15, 20]]

https://github.com/Leslie-Wong-H/game_of_life/blob/master/src/components/JxgContainer/gameOfLife/RLE.ts

Encode RLE (Mainly reverse the logic of decoding RLE)

```
53    /**
54     * RLE Decipher, rule description: https://conwaylife.com/wiki/Run_Length_Encoded
55     * @param rawRLEtext the raw RLE text
56     * @returns Object { width:number, height:number, result:CoordinatedPattern}
57     */
58    function RLEDecipher(rawRLEtext: string): RLEDecipherResult {···
158   }
159
160   /**      You, 4 months ago via PR #105 • 🎉 feat: add RLEEncipher and its test set
161    * RLE Encipher, rule description: https://conwaylife.com/wiki/Run_Length_Encoded
162    * @param Object { width:number, height:number, result:CoordinatedPattern}
163    * @return rawRLEtext the raw RLE text
164    */
165   function RLEEncipher(patternInfo: RLEDecipherResult): string {···
251   }
252
```

https://github.com/Leslie-Wong-H/game_of_life/blob/master/src/components/JxgContainer/gameOfLife/RLE.ts

Thanks for your listening