

YES

Utilizing JSXGraph to Implement John Conway's Game of Life

Leslie Wong 79917148leslie@gmail.com @LeslieWongH1



What is game of life? :

The Game of Life is a cellular automation model devised by the mathematician John Horton Conway from Cambridge University. It came to become well-known for the article published at Scientific American in 1970.

Rules of the Game of Life: The universe of the Game of Life is an infinite, two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive and dead, (or populated and unpopulated, respectively). Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

1. For a box with a live cell:

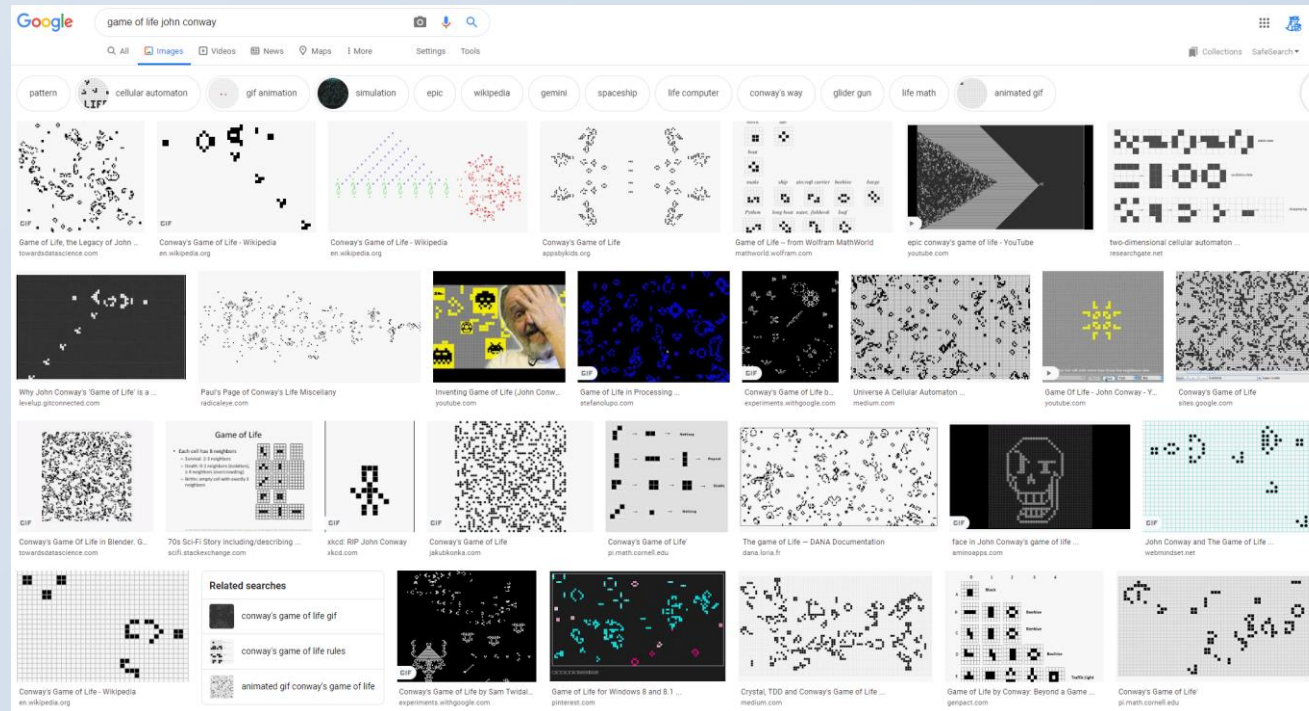
1. There existing fewer than two live neighbors, the cell dies, due to isolation.
2. There existing more than three live neighbors, the cell dies, due to over-crowding.
3. There existing two or three live neighbors, the cell lives on to the next generation.

2. For an empty box or a box with a dead cell:

1. There existing three live neighbors, the box generates a new live cell, as if by reproduction.

MOTIVATION:

Not able to spot an elegant and appealing implementation of game of life when searching in google



Just use vanilla box
to represent cell?

&

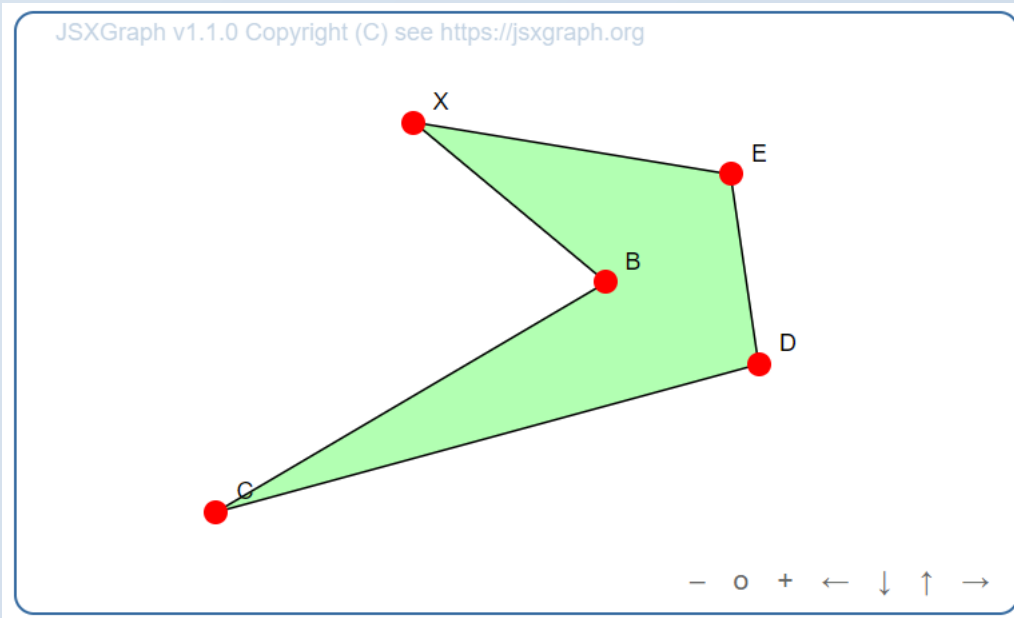
Always Black and White?

Bad User Experience!



MOTIVATION:

Came across JSXGraph by chance and
be obsessed with its fabulous UI



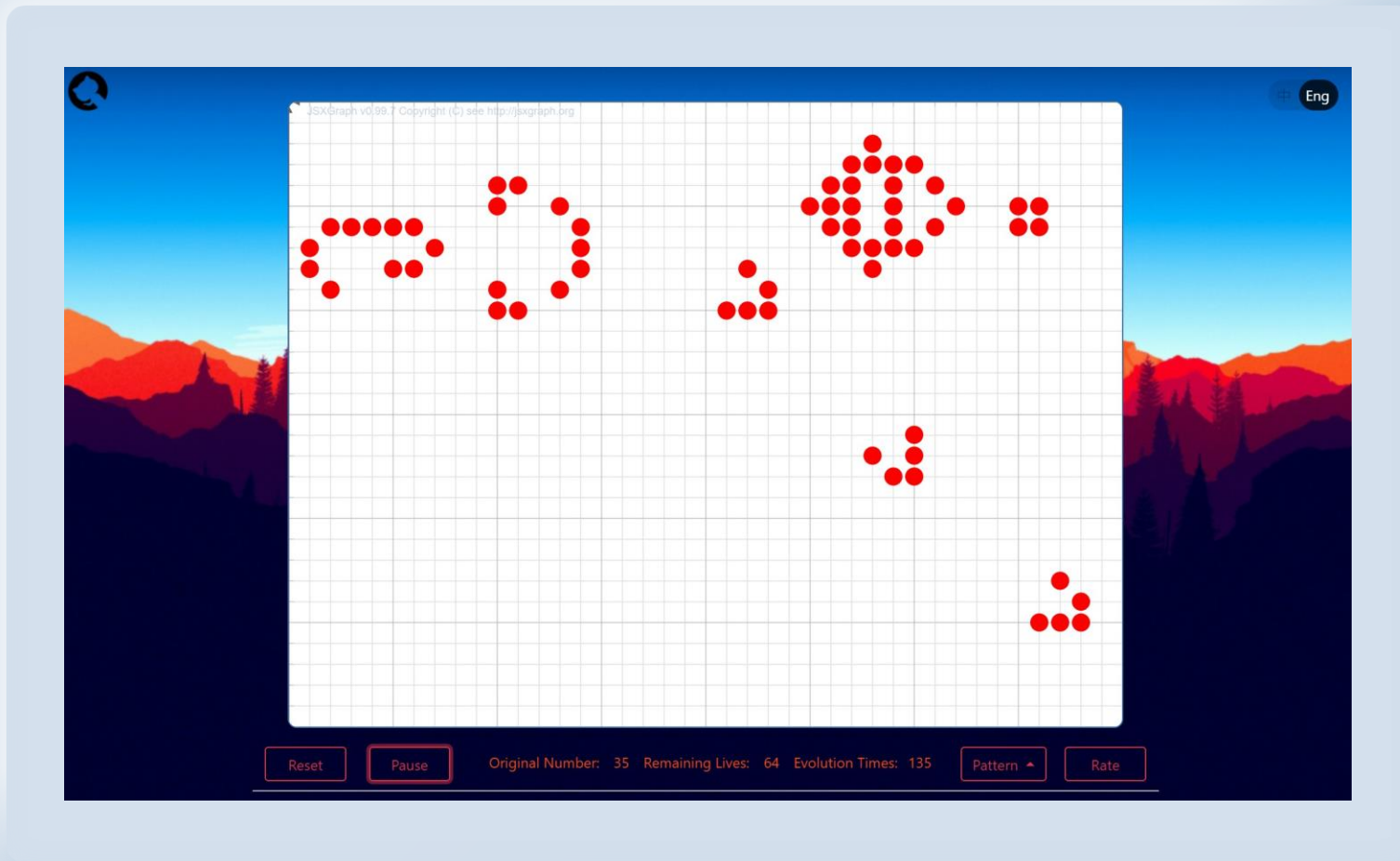
Impressed by the red point element
Nice to represent cell

&

Overwhelmed by JSXGraph's detailed
documentation and examples

Just do it!

RESULT: <https://playgameoflife.live>



PEACEFUL LAYOUT

CONSIDERATE
INTERACTIVITY

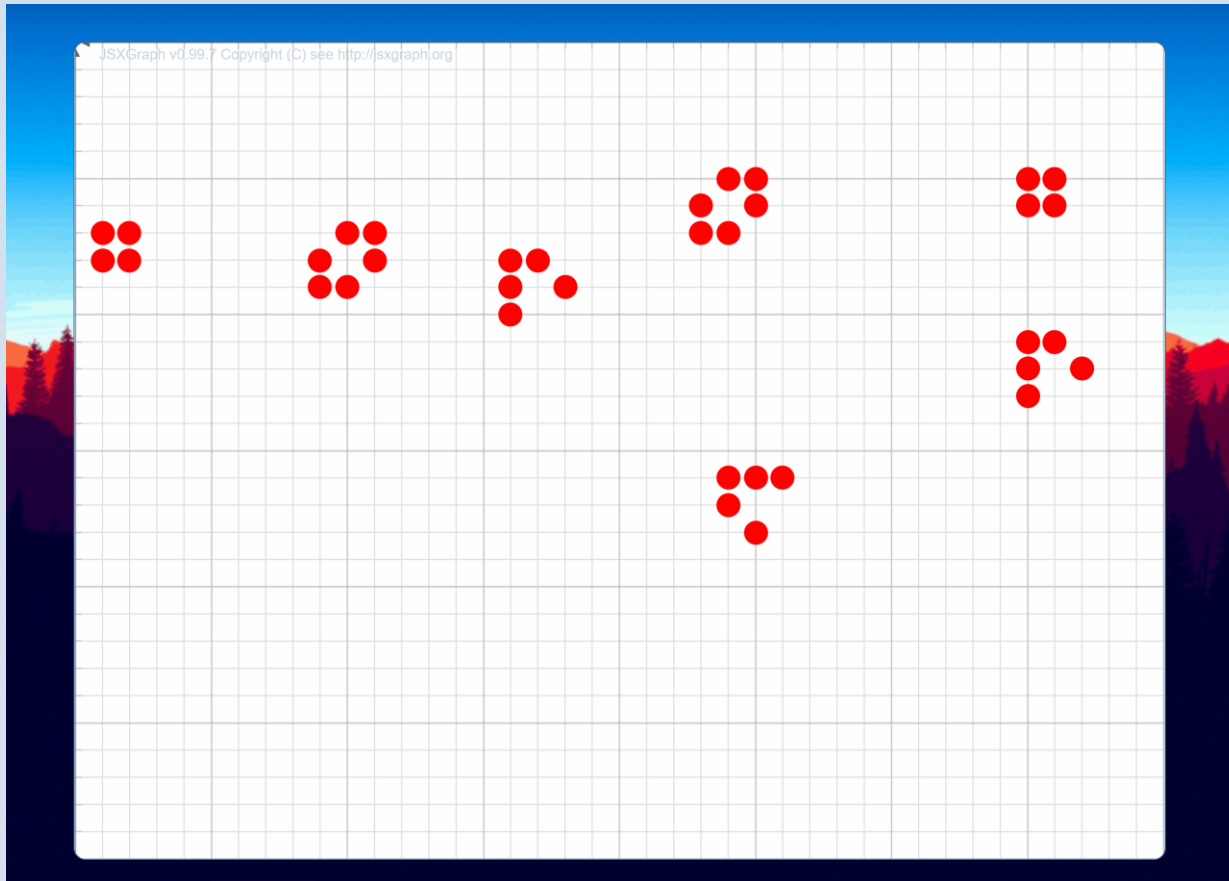
INTEGRATE
GAME OF LIFE
LEXICON

OPEN SOURCE

ANYWAY

COOL~

RESULT: <https://playgameoflife.live>

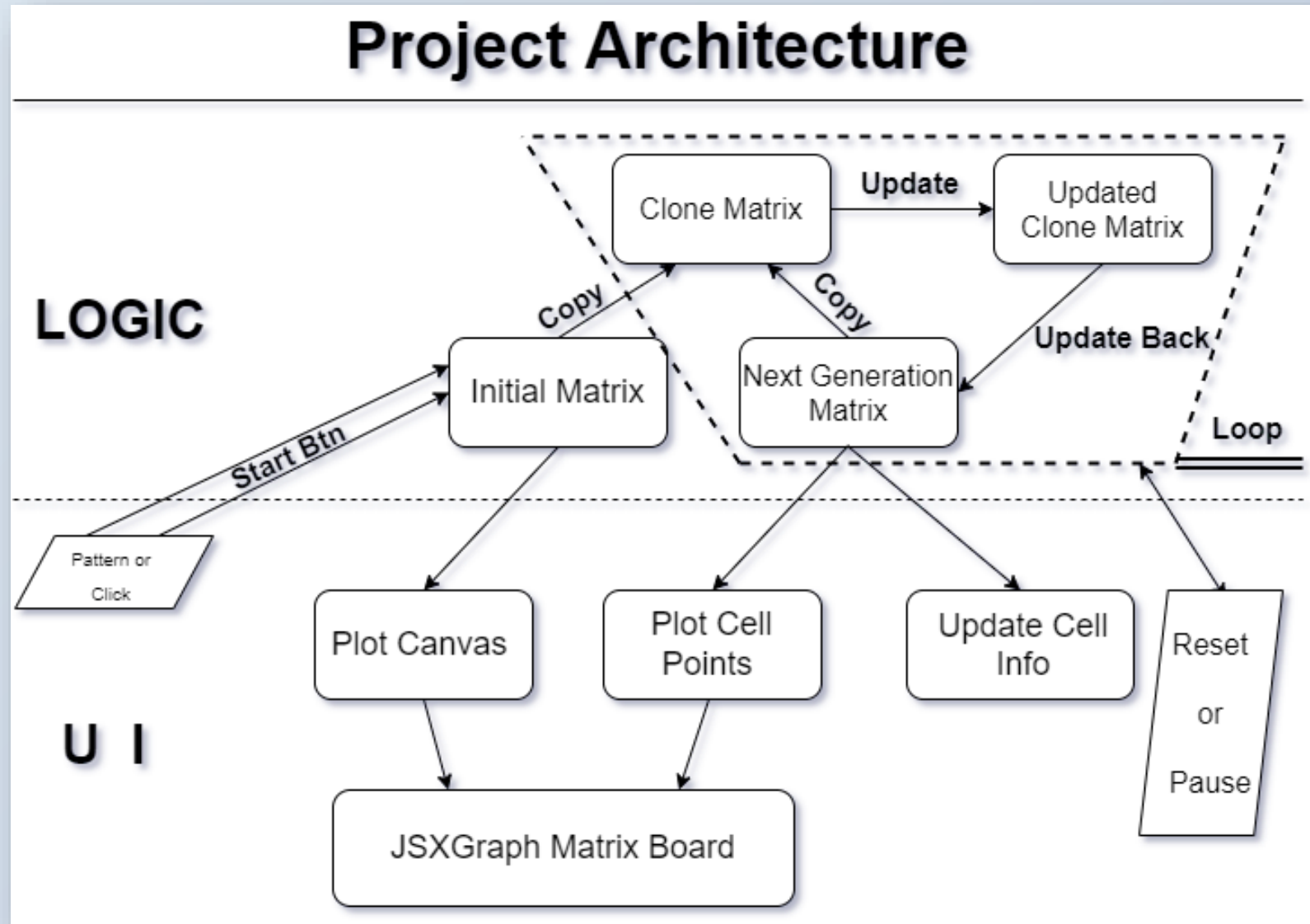


← ANIMATION
ILLUSTRATION

[GITHUB
README](#)

[YOUTUBE
VIDEO](#)

TECHNICAL PART – HOW I DEVELOP IT



TECHNICAL PART – HOW I DEVELOP IT

Game of Life > public > js > JS gameofLife.js > nextGeneration

```
103
104 var board = JXG.JSXGraph.initBoard("box", {
105     boundingbox: [0, 0, -40, -30],
106     keepaspectratio: true,
107     axis: true,
108     grid: true,
109     showCopyright: true,
110     shownavigation: false,
111     pan: {
112         //panning interaction(i.e.moving the origin)
113         enabled: false, // disallow panning
114         needTwoFingers: false, // panning could not be done with two fingers on touch devices
115         needShift: false, // mouse panning needs pressing of the shift key
116     },
117     zoom: {
118         factorX: 1, // horizontal zoom factor (multiplied to JXG.Board#zoomX)
119         factorY: 1, // vertical zoom factor (multiplied to JXG.Board#zoomY)
120         wheel: false, // allow zooming by mouse wheel or
121         // by pinch-to-zoom gesture on touch devices
122         needShift: true, // mouse wheel zooming needs pressing of the shift key
123         min: 1, // minimal values of JXG.Board#zoomX and JXG.Board#zoomY, limits zoomOut
124         max: 1, // maximal values of JXG.Board#zoomX and JXG.Board#zoomY, limits zoomIn
125         pinchHorizontal: false, // Allow pinch-to-zoom to zoom only horizontal axis
126         pinchVertical: false, // Allow pinch-to-zoom to zoom only vertical axis
127         pinchSensitivity: 0, // Sensitivity (in degrees) for recognizing horizontal or vertical pinch-to-zoom gestures.
128     },
129 });
130
```


TECHNICAL PART – HOW I DEVELOP IT

Game of Life > public > js > JS gameofLife.js > nextGeneration

```
375
376 //Update back the state of next generation cell from clone matrix to original matrix, and plot
377 board.suspendUpdate();
378 for (i = 0; i < matrixRow; i++) {
379     for (j = 0; j < matrixColumn; j++) {
380         matrix[i][j] = copyMatrix[i][j];
381         if (matrix[i][j] == 1) {
382             if (plotMatrix[i][j] != "") {
383                 board.removeObject(plotMatrix[i][j]);
384             }
385             plotMatrix[i][j] = board.create("point", [-j, -i], {
386                 size: cellSize,
387                 name: "",
388                 fixed: true,
389             });
390         } else {
391             board.removeObject(plotMatrix[i][j]);
392             plotMatrix[i][j] = "";
393         }
394     }
395 }
396 board.unsuspendUpdate();
397
```

TECHNICAL PART – HOW I DEVELOP IT

FRONTEND

<%= **EJS** %>

Bootstrap

jQuery
write less, do more.


JSXGraph

BACKEND

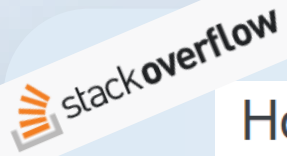
node JS

Express **JS**

 **HEROKU**

 **mongoDB**
Atlas

GREAT THANKS FOR HELP FROM MR. ALFRED WASSERMANN



How to delete the created point when using JSXGraph?

Asked 1 year, 2 months ago · Active 1 year, 2 months ago · Viewed 225 times

▲ JSXGraph elements are removed by calling `board.removeObject(object);` or `board.removeObject([array of objects]);`. Your example would look like:

0

▼
✓
↻

```
var p = board.create('point',[-3,1]);
var q = board.create('point',[-3,1], {face:'x', size:16});
board.removeObject(p);
board.removeObject(q);
// or
board.removeObject([p, q]);
```

share edit follow

answered Jun 16 '19 at 13:59

 Alfred Wassermann
1,358 ● 1 ● 7 ● 8

add a comment

FINALLY, IN MEMORY OF JOHN CONWAY

 **The Mathematical Association** @Mathematical_A · Apr 12
We're devastated to hear of the death of one of our Honorary Members, John Horton Conway. Here's a nice profile from The Guardian from 2015. theguardian.com/science/2015/j...



15 286 317

 **Leslie Wong** @LeslieWongH1 · Apr 13
Just implemented an interesting website project for Game of Life not long ago. It is also devastated of me to hear the death of John Conway. 😞😞😞
lesliewong.cn/gameoflife/

Thanks for devising
such a wonderful cell
automation world
(● - ●)