

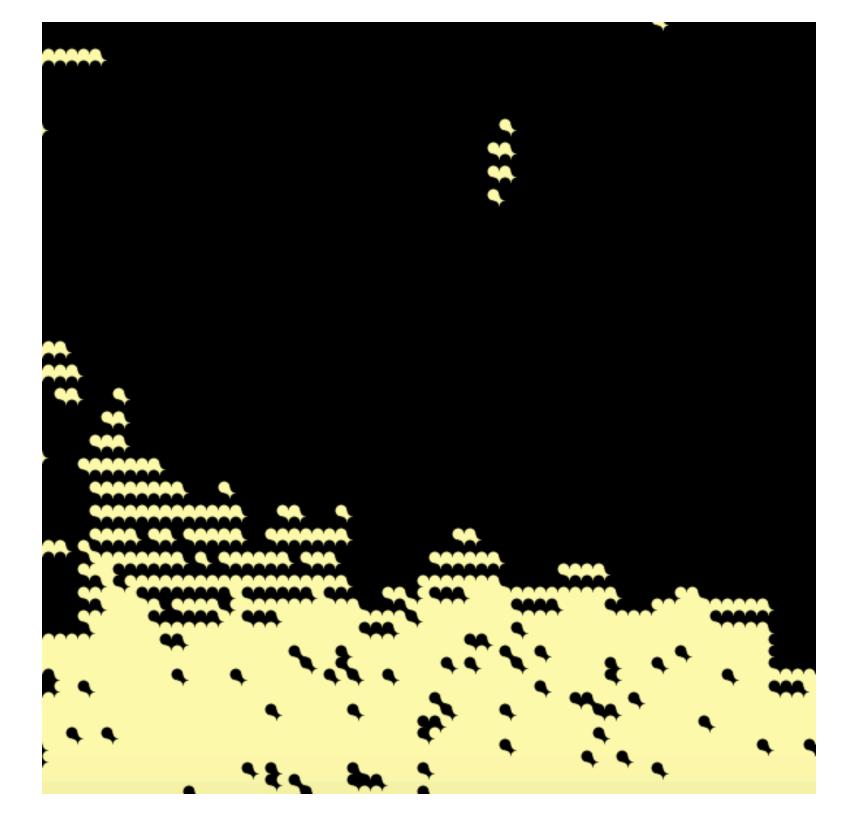
```
R. .
                                              function make2DArray (cols, rows){
                                                           let array = []
                                                            for (var i = 0; i < cols; i++){
                                                                             let row = []
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         TOTAL TOTAL CONTROL OF THE PARTY OF THE PART
                                                                             for (var j = 0; j < rows; j++){}
11
                                                                                             let obj = {
12
13
                                                                                                          x:i,
                                                                                                            y:j,
                                                                                                           state: random(0.05)
                                                                                              }
                                                                                             row.push(obj)
                                                                            }
 21
                                                                                           array.push(row)
                                                         return array
                                           }
                                            let grid;
                                           let cols;
                                           let rows;
                                           let resolution = 7;
```

```
E.
     let resolution = 7;
     function setup(){
       createCanvas(600,600);
       cols = width / resolution;
       rows = height / resolution;
       grid = make2DArray(cols, rows);
     function draw(){
       background(0)
       grid = eat()
       display()
     }
70
     function display(){
71
       for (var x = 0; x < grid.length; x++){
         let row = grid[x]
76
         for (var y = 0; y < row.length; y++){
           let cell = grid[x][y]
           fill(255)
           stroke(0)
           if (cell.state){
              fill(0)
           }
         ellipse(cell.x*resolution,cell.y*resolution,8,9)
```

```
II..
FEBL
140
142
            let obj = {
146
              x:i,
              y:j,
            if (score > 1){
150
              obj.state = true
            } else if (random()<0.05) {
               obj.state = true
154
155
            } else {
156
              obj.state = false
158
            }
159
            newRow.push(obj)
          newGrid.push(newRow)
        }
170
        return newGrid
171
172
```

```
94
          }
        }
      }
100
      function eat(){
        let newGrid = []
104
        for (var i = 0; i < grid.length; i++){
106
          let newRow = []
          let row = grid[i]
110
          for (var j = 0; j < row.length; j++){
111
112
            let score = 0
            let cell = grid[i][j]
113
114
            let a = constrain(i-1, 0, grid.length);
115
116
            let b = constrain(i+1, 0, grid.length);
            let c = constrain(j-1, 0, row.length);
117
            let d = constrain(j+1, 0, row.length);
118
119
120
121
           // let n1 = grid[a][c]
122
123
            let n3 = grid[a][d]
124
126
127
            let n4 = grid[i][c]
            let n6 = grid[i][d]
128
129
130
            let n7 = grid[a][j]
            let n8 = grid[a][c]
131
132
133
            let neighbours = [n7,n8,n3,n4,n6]
134
135
136
            for (var r = 2; r < neighbours.length; r++){
              if (neighbours[r]){
137
                score += neighbours[r].state
138
139
140
```

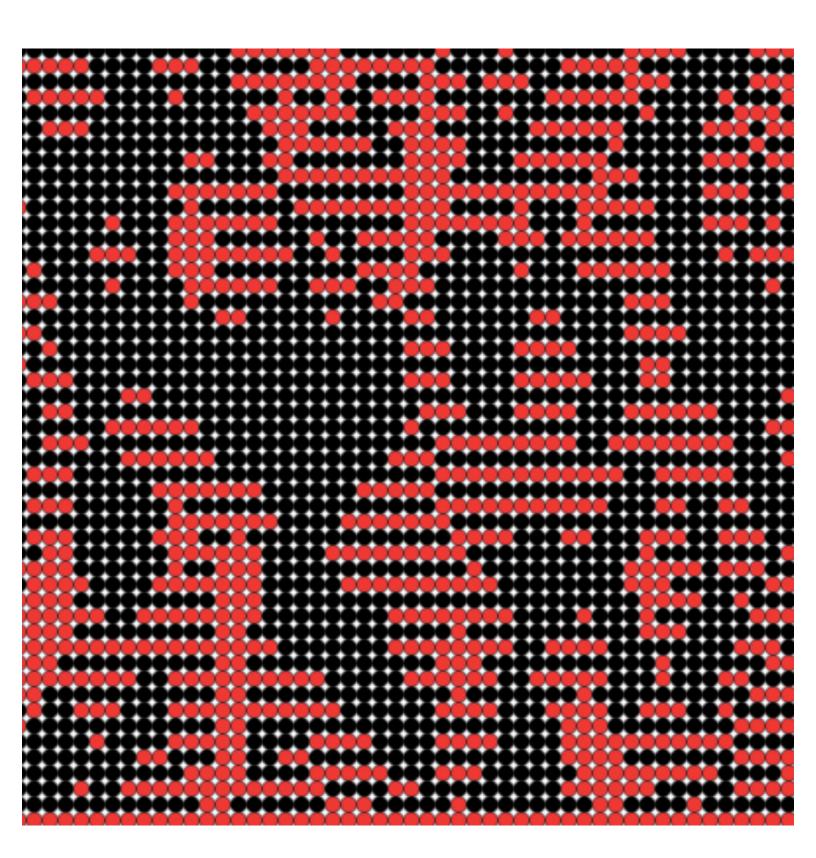
```
II...
                                                                                       FEMA.
The second secon
```



```
142
143
             let obj = {
144
145
               x:i,
146
               y:j,
147
148
             if (score < 2){
149
150
               obj.state = true
151
             } else if (random()<0.5) {</pre>
152
               obj.state = true
153
             } else {
154
155
               obj.state = false
156
157
158
159
160
161
             newRow.push(obj)
162
163
164
165
           newGrid.push(newRow)
166
         }
167
168
169
         return newGrid
170
171
172
      // function draw(){
173
            background(0);
174
```

```
function make2DArray (cols, rows){
       let array = []
       for (var i = 0; i < cols; i++){
         let row = []
         for (var j = 0; j < rows; j++){}
11
            let obj = {
12
13
             x:i,
             y:j,
15
             state: (random()>0.5)
           }
17
            row.push(obj)
21
         }
           array.push(row)
23
25
       return array
     }
     let grid;
     let cols;
     let rows;
     let resolution = 9;
```

```
51
     function setup(){
52
       createCanvas(600,600);
       cols = width / resolution;
       rows = height / resolution;
       grid = make2DArray(cols, rows);
     }
60
     function draw(){
       background(255)
62
       grid = eat()
       display()
     }
70
71
     function display(){
72
       for (var x = 0; x < grid.length; x++){
74
         let row = grid[x]
76
         for (var y = 0; y < row.length; y++){
79
           let cell = grid[x][y]
           fill(255,255,174)
82
           stroke(0)
           strokeWeight(0)
           if (cell.state){
87
              fill(0)
         rect(cell.x*resolution,cell.y*resolution,resolution,resolution)
          ellipse(cell.x*resolution, cell.y*resolution, resolution)
94
         }
       }
```

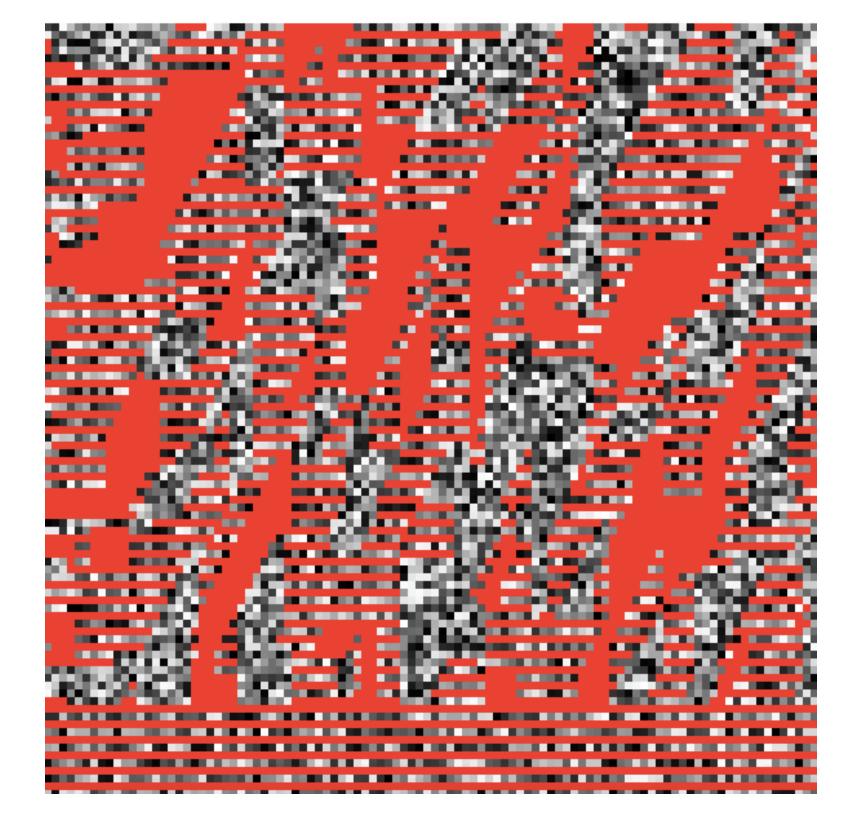


```
function make2DArray (cols, rows){
       let array = []
       for (var i = 0; i < cols; i++){
         let row = []
         for (var j = 0; j < rows; j++){}
11
            let obj = {
12
13
             x:i,
14
             y:j,
15
             state: (random()>0.5)
17
           }
            row.push(obj)
21
        }
23
           array.push(row)
24
25
29
34
       return array
     }
     let grid;
     let cols;
47
     let rows;
     let resolution = 9;
```

```
48
     let resolution = 9;
     function setup(){
       createCanvas(800,800);
       cols = width / resolution;
       rows = height / resolution;
       grid = make2DArray(cols, rows);
     }
     function draw(){
       background(255)
       grid = eat()
       display()
70
71
     }
     function display(){
       for (var x = 0; x < grid.length; <math>x++){
76
         let row = grid[x]
79
         for (var y = 0; y < row.length; y++){
           let cell = grid[x][y]
           fill(0)
           stroke(0)
           strokeWeight(0.5)
           if (cell.state){
              fill(677,34,53)
           circle(cell.x*resolution,cell.y*resolution,resolution,resolution)
```

```
97
        }
100
      function eat(){
104
        let newGrid = []
        for (var i = 1; i < grid.length; i++){</pre>
           let newRow = []
109
          let row = grid[i]
110
111
           for (var j = 1; j < row.length; j++){</pre>
112
113
114
             let score = 0
115
             let cell = grid[i][j]
116
             let a = constrain(i-1, 0, grid.length);
             let b = constrain(i+1, 0, grid.length);
118
119
             let c = constrain(j-1, 0, row.length);
120
             let d = constrain(j+1, 0, row.length);
121
122
123
124
125
126
             let n3 = grid[a][d]
127
128
             let n4 = grid[i][c]
129
130
             let n6 = grid[i][d]
131
132
             let n7 = grid[a][j]
             let n8 = grid[a][c]
133
134
135
136
             let neighbours = [n7,n8,n3,n4,n6]
137
             for (var r = 1; r < neighbours.length; r++){</pre>
138
139
               if (neighbours[r]){
                 score += neighbours[r].state/2
140
141
142
143
```

```
142
143
144
145
             let obj = {
146
147
148
               x:i,
149
               y:j,
150
151
152
             if (score > 1){
153
               obj.state = false
             } else if (neighbours.n7*n3 < 5) {
154
155
               obj.state = false
156
             } else {
157
158
159
               obj.state = true
160
161
             }
162
163
164
             newRow.push(obj)
165
166
167
168
           newGrid.push(newRow)
169
         }
170
171
         return newGrid
172
173
174
       }
```

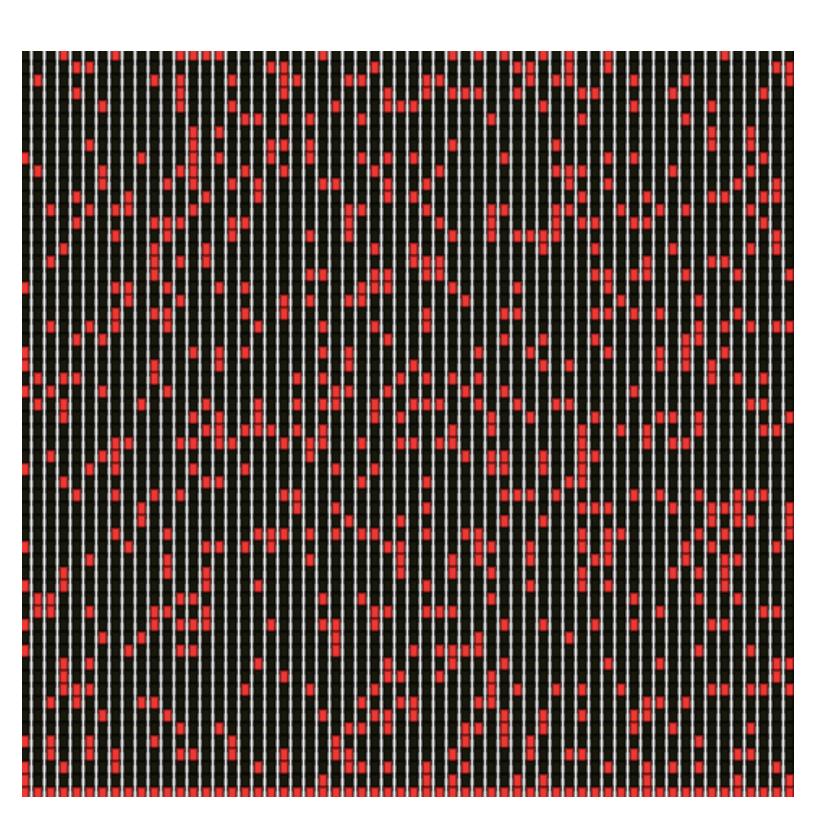


```
1
     function make2DArray (cols, rows){
        let array = []
        for (var i = 0; i < cols; i++){
          let row = []
          for (var j = 0; j < rows; j++){}
10
11
            let obj = {
12
13
              x:i,
14
              y:j,
15
              state: (random()>0.5)
17
           }
            row.push(obj)
19
20
21
          }
22
23
            array.push(row)
        }
24
25
        // console.log(array[4][3])
27
29
30
31
32
33
        // n2 = array[i+1][j]
34
35
36
37
38
41
       return array
42
43
     }
44
      let grid;
      let cols;
47
      let rows;
      let resolution = 6:
```

```
51
      function setup(){
52
        createCanvas(600,600);
53
       cols = width / resolution;
        rows = height / resolution;
        grid = make2DArray(cols*9, rows);
     }
      function draw(){
61
        background(255)
62
63
       grid = eat()
        display()
67
     }
70
71
      function display(){
72
73
       for (var x = 0; x < grid.length; x++){
74
          let row = grid[x]
76
          for (var y = 0; y < row.length; y++){
78
            let cell = grid[x][y]
79
81
            fill(235,65,13)
82
            stroke(0)
83
            strokeWeight(0)
85
            if (cell.state){
87
               fill(random(255))
            }
90
        rect(cell.x*resolution,cell.y*resolution,resolution*9,resolution*9)
91
92
         }
        }
```

```
100
101
       function eat(){
103
         let newGrid = []
104
105
         for (var i = 0; i < grid.length; i++){</pre>
           let newRow = []
108
           let row = grid[i]
110
           for (var j = 0; j < row.length; j++){
111
112
             let score = 0
113
             let cell = grid[i][j]
114
115
             let a = constrain(i-1, 0, grid.length);
116
             let b = constrain(i+1, 1, grid.length);
             let c = constrain(j-1, 0, row.length);
117
118
             let d = constrain(j+1, 1, row.length);
119
120
121
122
            // let n1 = grid[a][c]
123
             //let n2 = grid[a][j]
124
             let n3 = grid[a][d]
125
126
127
             let n4 = grid[i][c]
128
             let n6 = grid[i][d]
129
130
             let n7 = grid[a][j]
             let n8 = grid[a][c]
131
132
133
134
             let neighbours = [n7,n8,n3,n4,n6]
135
136
             for (var r = 2; r < neighbours.length; r++){</pre>
137
               if (neighbours[r]){
138
                 score += neighbours[r].state
               }
139
141
142
143
             let obj = {
144
145
146
```

```
146
               x:i,
147
               y:j,
148
149
             }
150
151
           if (score < 2){
152
               obj.state = true
153
             } else if (score +neighbours) {
               obj.state = true
154
155
156
               if (score = 3){
157
                 obj.state = false
158
159
               }else if (score>3+neighbours)
160
               obj.state= false
161
162
163
164
165
             newRow.push(obj)
166
167
168
169
170
           newGrid.push(newRow)
171
         }
172
173
174
         return newGrid
175
176
```



```
function make2DArray (cols, rows){
       let array = []
       for (var i = 7; i < cols; i++){
         let row = []
         for (var j = 9; j < rows; j++){}
11
           let obj = {
12
13
             x:i,
14
             y:j,
             state: random(0.05)
17
           }
           row.push(obj)
         }
21
23
           array.push(row)
       }
24
     return array
42
     }
     let grid;
     let cols;
     let rows;
     let resolution = 9;
```

```
51
     function setup(){
52
       createCanvas(600,600);
53
       cols = width / resolution;
        rows = height / resolution;
       grid = make2DArray(cols, rows);
     }
61
     function draw(){
       background(255)
62
63
       grid = eat()
       display()
     }
70
71
     function display(){
72
       for (var x = 0; x < grid.length; x++){
73
74
         let row = grid[x]
75
76
         for (var y = 0; y < row.length; y++){
78
            let cell = grid[x][y]
79
            fill(25,24,12)
81
82
            stroke(3)
83
85
            if (cell.state){
               fill(22,34,123)
            }
         rect(cell.x*resolution,cell.y*resolution,6,11)
91
92
93
         }
95
        }
```

```
}
 99
100
       function eat(){
101
102
103
         let newGrid = []
104
105
         for (var i = 0; i < grid.length; i++){
106
107
           let newRow = []
108
           let row = grid[i]
109
110
           for (var j = 0; j < row.length; j++){
111
112
              let score = 0
113
              let cell = grid[i][j]
114
115
             let a = constrain(i-1, 5, grid.length);
                       constrain(i+1, 100, grid.length);
constrain(j-1, 75, row.length);
116
             let b =
117
              let c =
              let d = constrain(j+1, 1, row.length);
118
119
120
121
122
             let n1 = grid[a][c]
123
             let n2 = grid[a][j]
124
              let n3 = grid[a][d]
125
126
127
             let n4 = grid[i][c]
128
             let n6 = grid[i][d]
129
130
131
132
133
134
             let neighbours = [n1,n2,n3,n4,n6]
135
136
              for (var r = 3; r < neighbours.length; r++){</pre>
                if (neighbours[r]){
137
138
                  score += neighbours[r].state
139
                }
140
141
142
143
144
              let obj = {
145
146
```

```
138
                  score += neighbours[r].state
139
140
141
             // console.log(score,a,b,c,d)
142
143
              let obj = {
144
145
146
               x:i,
147
               y:j,
148
149
             if (score <1){
150
                obj.state = true
151
             } else if (random()<0.2) {</pre>
152
                obj.state = true
153
154
             }
155
156
157
158
159
             newRow.push(obj)
160
161
           }
162
           newGrid.push(newRow)
163
164
         }
165
166
167
         return newGrid
168
169
```