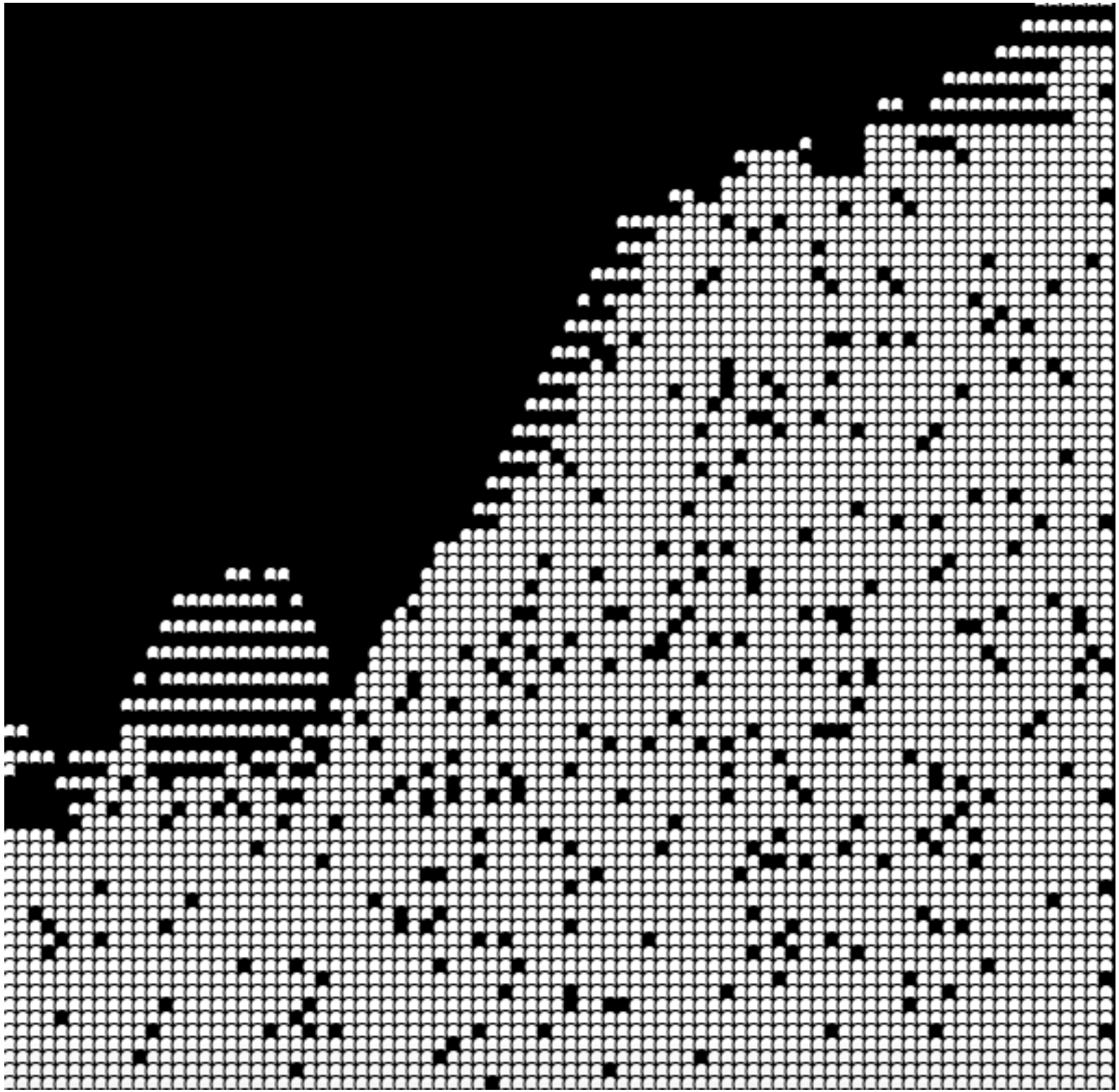
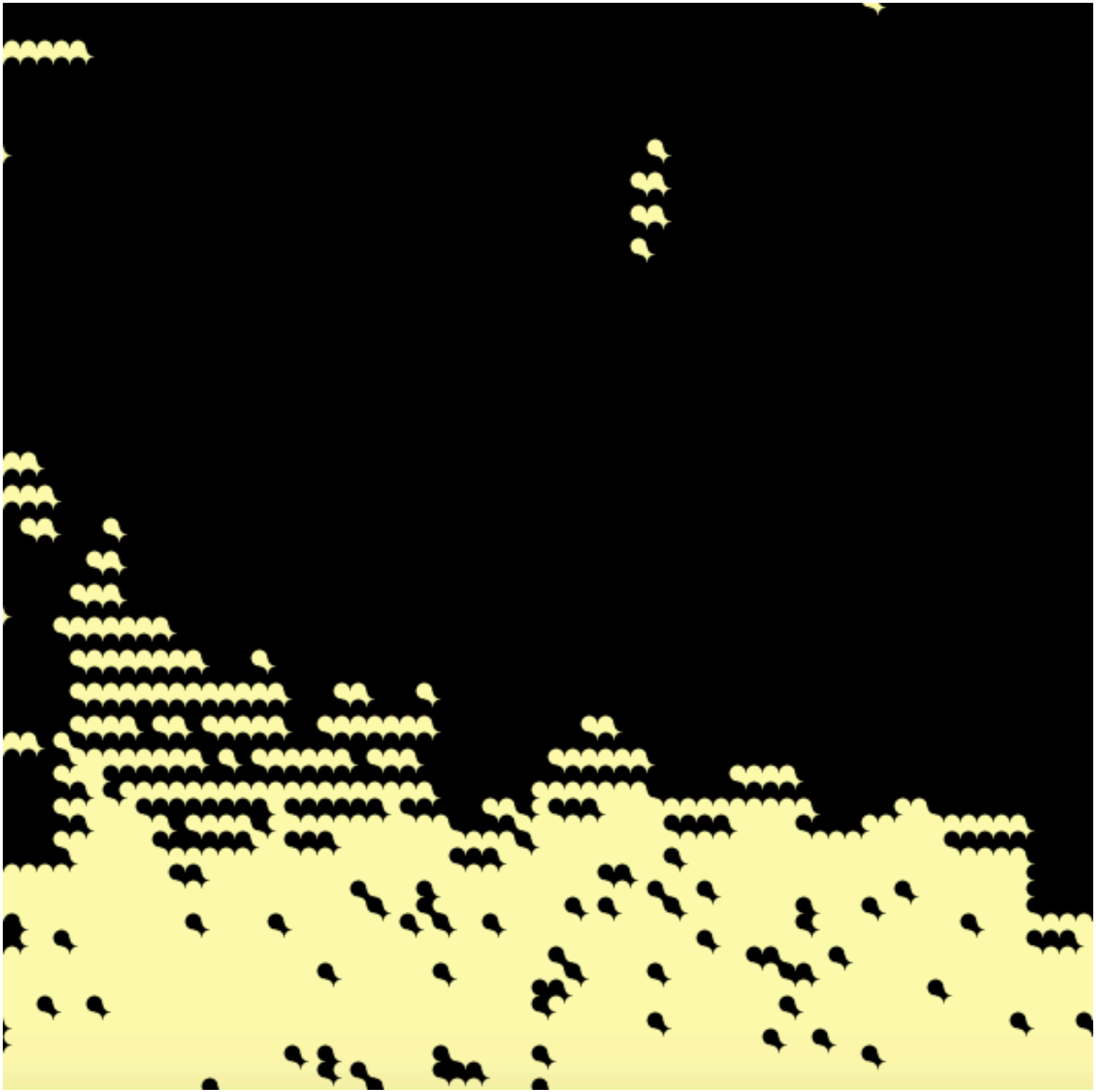


```
function make2DArray (cols, rows){

  let array = []

  for (var i = 0; i < cols; i++){

    let row = []

    for (var j = 0; j < rows; j++){

      let obj = {

        x:i,
        y:j,
        state: random(0.05)

      }

      row.push(obj)

    }

      array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 7;
```

```
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()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    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)




    }
```

```
140
141         }
142         // console.log(score,a,b,c,d)
143
144         let obj = {
145
146           x:i,
147           y:j,
148
149         }
150         if (score > 1){
151           obj.state = true
152         } else if (random()<0.05) {
153           obj.state = true
154
155         } else {
156
157           obj.state = false
158
159         }
160
161
162         newRow.push(obj)
163
164
165       }
166       newGrid.push(newRow)
167
168     }
169
170     return newGrid
171
172   }
173   // function draw(){
```

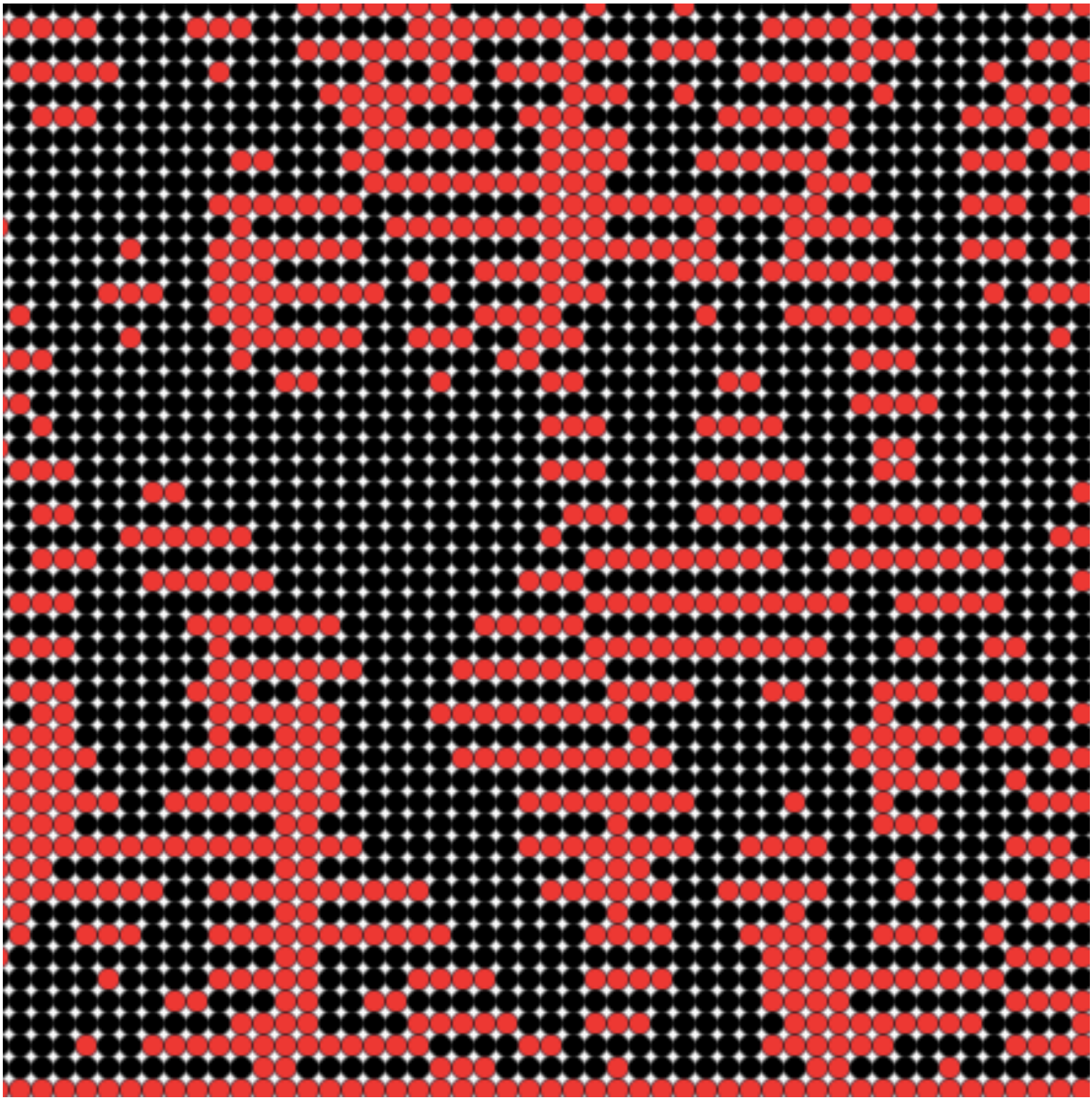
```
    }

  }

}

function eat(){

  let newGrid = []

  for (var i = 0; i < grid.length; i++){

    let newRow = []
    let row = grid[i]

    for (var j = 0; j < row.length; j++){

      let score = 0
      let cell = grid[i][j]

      let a =  constrain(i-1, 0, grid.length);
      let b =  constrain(i+1, 0, grid.length);
      let c =  constrain(j-1, 0, row.length);
      let d =  constrain(j+1, 0, row.length);



     // let n1 = grid[a][c]
      //let n2 = grid[a][j]
      let n3 = grid[a][d]
      // console.log(n1)

      let n4 = grid[i][c]
      let n6 = grid[i][d]

      let n7 = grid[a][j]
      let n8 = grid[a][c]
      // let n9 = grid[b][d]

      let neighbours = [n7,n8,n3,n4,n6]

      for (var r = 2; r < neighbours.length; r++){
        if (neighbours[r]){
          score += neighbours[r].state
        }

      }
```



```
142
143         let obj = {
144
145           x:i,
146           y:j,
147
148         }
149         if (score < 2){
150           obj.state = true
151         } else if (random()<0.5) {
152           obj.state = true
153
154         } else {
155
156           obj.state = false
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++){

      let obj = {

        x:i,
        y:j,
        state: (random()>0.5)

      }

      row.push(obj)

    }

    array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 9;
```

```
function setup(){
  createCanvas(600,600);
  cols = width / resolution;
  rows = height / resolution;
  grid = make2DArray(cols, rows);



}

function draw(){
  background(255)

  grid = eat()
  display()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    for (var y = 0; y < row.length; y++){

      let cell = grid[x][y]

      fill(255,255,174)
      stroke(0)
      strokeWeight(0)


      if (cell.state){
        fill(0)
      }

    rect(cell.x*resolution,cell.y*resolution,resolution,resolution)
     ellipse(cell.x*resolution,cell.y*resolution,resolution,resolution)



    }

  }
```

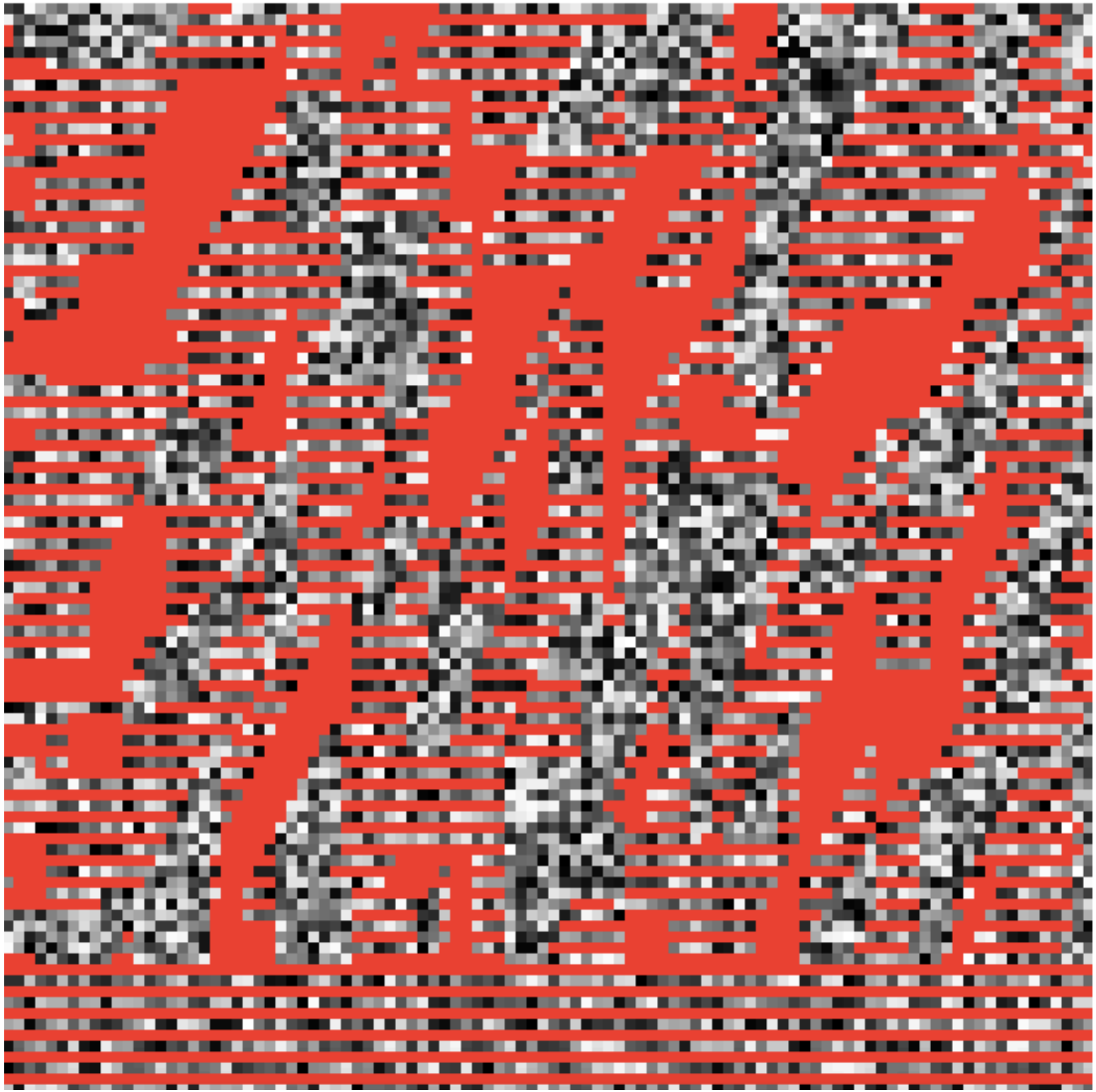


```
function make2DArray (cols, rows){

  let array = []

  for (var i = 0; i < cols; i++){

    let row = []

    for (var j = 0; j < rows; j++){

      let obj = {

        x:i,
        y:j,
        state: (random()>0.5)

      }

      row.push(obj)

    }

      array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 9;
```

```
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()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    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)


```

```
    }

  }

}

function eat(){

  let newGrid = []

  for (var i = 1; i < grid.length; i++){

    let newRow = []
    let row = grid[i]

    for (var j = 1; j < row.length; j++){

      let score = 0
      let cell = grid[i][j]

      let a =  constrain(i-1, 0, grid.length);
      let b =  constrain(i+1, 0, grid.length);
      let c =  constrain(j-1, 0, row.length);
      let d =  constrain(j+1, 0, row.length);



     // let n1 = grid[a][c]
      //let n2 = grid[a][j]
      let n3 = grid[a][d]
      // console.log(n1)

      let n4 = grid[i][c]
      let n6 = grid[i][d]

      let n7 = grid[a][j]
      let n8 = grid[a][c]
      // let n9 = grid[b][d]

      let neighbours = [n7,n8,n3,n4,n6]

      for (var r = 1; r < neighbours.length; r++){
        if (neighbours[r]){
          score += neighbours[r].state/2
        }

      }
      // console.log(score,a,b,c,d)
```

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

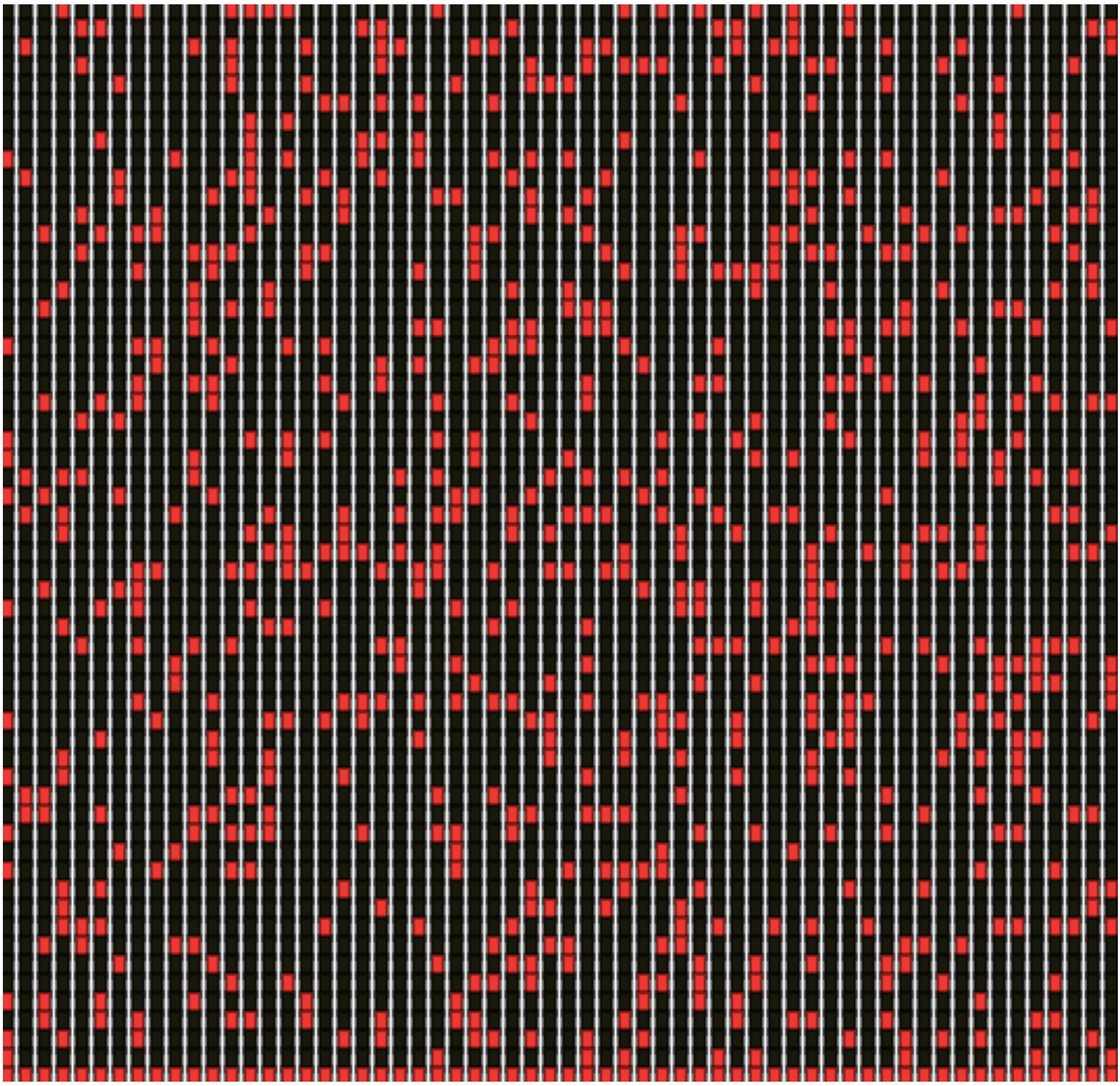


```
function make2DArray (cols, rows){

  let array = []

  for (var i = 0; i < cols; i++){

    let row = []

    for (var j = 0; j < rows; j++){

      let obj = {

        x:i,
        y:j,
        state: (random()>0.5)

      }

      row.push(obj)

    }

      array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 6;
```

```
function setup(){
  createCanvas(600,600);
  cols = width / resolution;
  rows = height / resolution;
  grid = make2DArray(cols*9, rows);



}

function draw(){
  background(255)

  grid = eat()
  display()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    for (var y = 0; y < row.length; y++){

      let cell = grid[x][y]

      fill(235,65,13)
      stroke(0)
      strokeWeight(0)


      if (cell.state){
         fill(random(255))
      }

  rect(cell.x*resolution,cell.y*resolution,resolution*9,resolution*9)




    }

  }

```

```
 99  }
100
101  function eat(){
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, 0, grid.length);
116        let b =  constrain(i+1, 1, grid.length);
117        let c =  constrain(j-1, 0, row.length);
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        // console.log(n1)
126
127        let n4 = grid[i][c]
128        let n6 = grid[i][d]
129
130        let n7 = grid[a][j]
131        let n8 = grid[a][c]
132        // let n9 = grid[b][d]
133
134        let neighbours = [n7,n8,n3,n4,n6]
135
136        for (var r = 2; r < neighbours.length; r++){
137          if (neighbours[r]){
138            score += neighbours[r].state
139          }
140
141        }
142        // console.log(score,a,b,c,d)
143
144        let obj = {
145
146          x:i,
```

```
146             x:i,
147             y:j,
148
149           }
150
151         if (score < 2){
152             obj.state = true
153           } else if (score +neighbours) {
154             obj.state = true
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++){

      let obj = {

        x:i,
        y:j,
        state: random(0.05)

      }

      row.push(obj)

    }

      array.push(row)
  }

  // console.log(array[4][3])

  // let i = 3
  // let j = 3


  // let cell = array[i][j]
  // n2 = array[i+1][j]
  //  n1 = array[i-][j]
  // let arr = new Array(cols);
  // for (let i =0; i < arr.length; i++) {
  //   arr [i] = new Array(rows);

  // }

  return array

}

let grid;
let cols;
let rows;
let resolution = 9;
```

```
function setup(){
  createCanvas(600,600);
  cols = width / resolution;
  rows = height / resolution;
  grid = make2DArray(cols, rows);



}

function draw(){
  background(255)

  grid = eat()
  display()



}

function display(){

  for (var x = 0; x < grid.length; x++){

    let row = grid[x]

    for (var y = 0; y < row.length; y++){

      let cell = grid[x][y]

      fill(25,24,12)
      stroke(3)



      if (cell.state){
        fill(22,34,123)
      }

    rect(cell.x*resolution,cell.y*resolution,6,11)




    }

  }
```

```
 99  }
100
101  function eat(){
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);
116        let b =  constrain(i+1, 100, grid.length);
117        let c =  constrain(j-1, 75, row.length);
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        // console.log(n1)
126
127        let n4 = grid[i][c]
128        let n6 = grid[i][d]
129
130        //let n7 = grid[a][j]
131        //let n8 = grid[a][c]
132        // let n9 = grid[b][d]
133
134        let neighbours = [n1,n2,n3,n4,n6]
135
136        for (var r = 3; r < neighbours.length; r++){
137          if (neighbours[r]){
138            score += neighbours[r].state
139          }
140
141        }
142        // console.log(score,a,b,c,d)
143
144        let obj = {
145
146          x:i,
```

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