

# 利用 HTML5 和 JavaScript 实现生命游戏

---

## 1 课题设计

---

### 1.1 课题背景

近日，Stephen Wolfram 发表了一篇热情洋溢的博客《Finally We May Have a Path to the Fundamental Theory of Physics... and It's Beautiful》，文章的主要思想是一些简单的规则在经过多次迭代后，会产生意想不到的复杂性。受该文的启发，我放弃了原来进展并不顺利的天体运动模型，转而实现生命游戏。

### 1.2 设计思路

1. 在html文件中，需要
  - 定义canvas
  - 定义“开始”，“停止”，“步进”3个接口
1. 在JavaScript文件中，需要
  - 画出网格
  - 计算每个网格的生死
  - 实时更新
  - 提供3个接口的函数实现

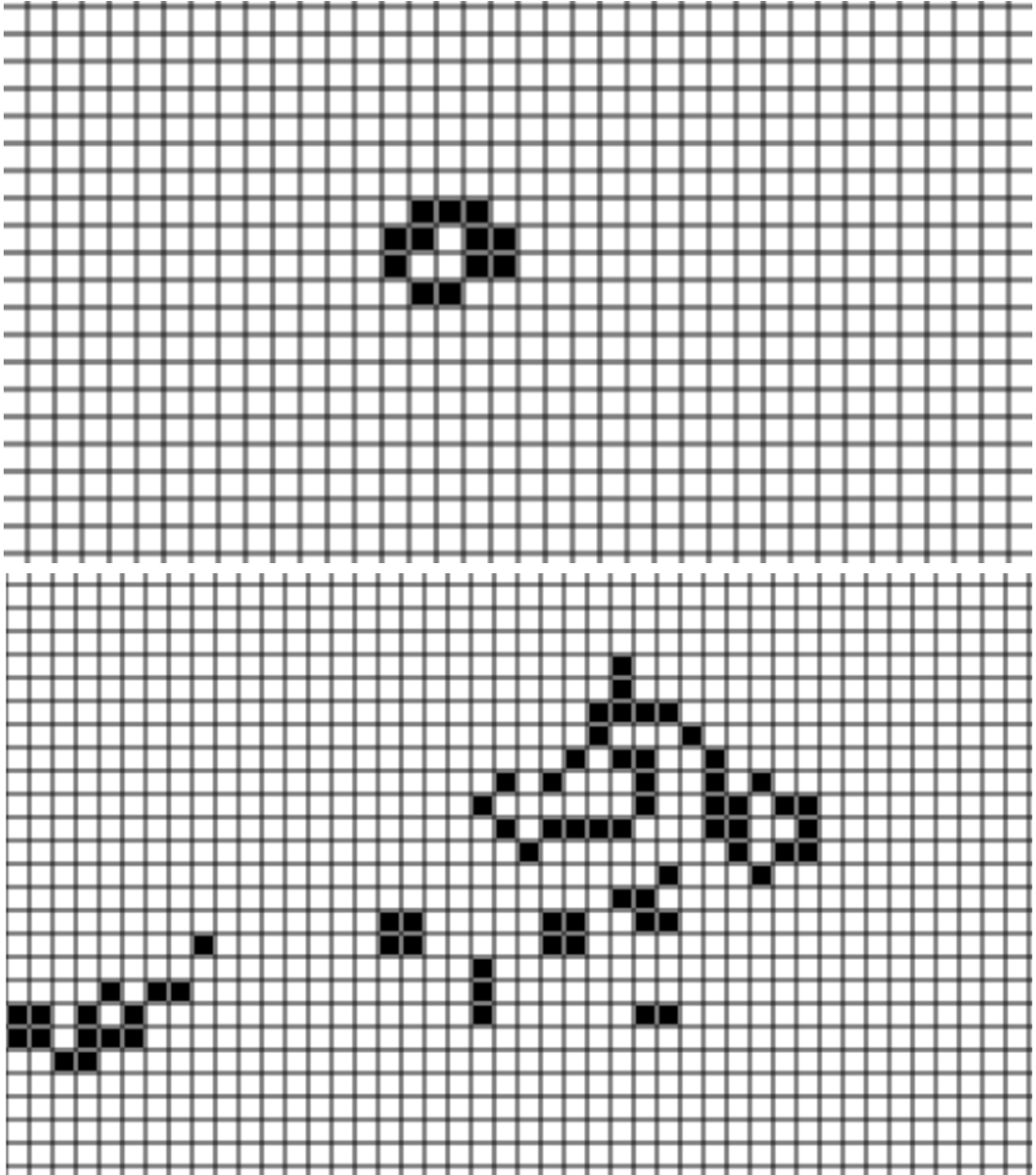
## 2 实验过程

---

1. 定义相关全局变量
  - SIZE: 单方向格子数
  - WIDTH: 格子宽度
  - GAME\_CANVAS
2. 定义主函数game
  - 初始化GAME\_CANVAS
3. 定义格子相关函数
  - generate\_cells: 用二维数组存储bool值，表示格子死活，将结果在game中实例化
  - count\_neighbors: 计算九宫格中活着的数目
  - draw\_cells: 活着则填色，死去则清空
4. 定义核心函数update
  - 画网格
  - 调用draw\_cells填充
  - 调用update计算新状态，在下次更新中画出和填充
5. 定义接口函数
  - start: 开始游戏，在定义过程中，需定义全局变量GAME\_INTERVAL，用于控制时间
  - fucking\_stop: 暂停游戏
  - step: 进行一次update
6. 定义鼠标监听函数mouselistener
  - 通过鼠标点击可以将格子设置为活的
  - 将监听函数添加到game函数中

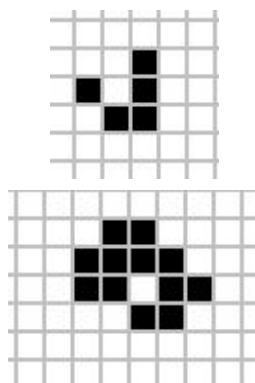
### 3 实验结果和简要分析

#### 1. 结果



#### 2. 分析

- 我尝试了几个经典的生命游戏开局，如滑翔机，轻型飞船等。观察到了生命游戏的神奇现象。加深了对Wolfram简单规则演化出复杂性的理解。



- 代码如下

```

1  const SIZE = 50;
2  const WIDTH = 10;
3  const GAME_CANVAS = document.getElementById("game");
4  let game_context = GAME_CANVAS.getContext("2d");
5  let GAME_INTERVAL = null;
6
7  let mousetlistener = function(event) {
8      let x = Math.floor(event.pageX/WIDTH) - 1;
9      let y = Math.floor(event.pageY/WIDTH) - 1;
10     GAME.cells[x][y] = +!GAME.cells[x][y];
11     draw_cells();
12 }
13
14 let game = function() {
15     GAME_CANVAS.width = WIDTH * SIZE;
16     GAME_CANVAS.height = WIDTH * SIZE;
17     this.cells = generate_cells();
18     update();
19
20     GAME_CANVAS.addEventListener("mousedown", mousetlistener,
21     false);
22 };
23
24 let generate_cells = function() {
25     let retval = new Array(SIZE);
26     for (let i = 0; i < retval.length; i++) {
27         retval[i] = new Array(SIZE);
28         for (let j = 0; j < retval[i].length; j++) {
29             retval[i][j] = 0;
30         }
31     }
32     return retval;
33 };
34
35 let draw_cells = function() {
36     for (let i = 0; i < SIZE; i++) {
37         for (let j = 0; j < SIZE; j++) {
38             if (GAME.cells[i][j])
39                 game_context.fillRect(i*WIDTH+1, j*WIDTH+1, WIDTH-
40 2, WIDTH-2);
41             else
42                 game_context.clearRect(i*WIDTH+1, j*WIDTH+1,
43 2, WIDTH-2);
44         }
45     }
46 };
47
48 let count_neighbors = function(x, y) {
49     let counter = 0;
50     for (let i = x - 1; i <= x + 1; i++) {
51         for (let j = y - 1; j <= y + 1; j++) {
52             if (i < 0 || i > SIZE-1 || j < 0 || j > SIZE - 1 || (i
53 ==x && j ==y))
54                 continue;
55             if (GAME.cells[i][j])
56                 counter += 1;
57         }
58     }
59 }

```

```

55     return counter;
56 };
57
58 let update = function() {
59     game_context.clearRect(0, 0, SIZE*WIDTH, SIZE*WIDTH);
60     for (let i = 0; i <= SIZE; i++) {
61         game_context.beginPath();
62         // horizontal
63         game_context.moveTo(0, i*WIDTH);
64         game_context.lineTo(SIZE*WIDTH, i*WIDTH);
65         // vertical
66         game_context.moveTo(i*WIDTH, 0);
67         game_context.lineTo(i*WIDTH, SIZE*WIDTH);
68         game_context.stroke();
69     }
70
71     // the first update
72     if (typeof(GAME) == "undefined")
73         return;
74
75     draw_cells();
76
77     // core logic
78     let new_cells = JSON.parse(JSON.stringify(GAME.cells)); //deep
79     copy
80     for (let i = 0; i < SIZE; i++) {
81         for (let j = 0; j < SIZE; j++) {
82             new_cells[i][j] = 0;
83             let cell = GAME.cells[i][j];
84             let nearby = count_neighbors(i, j);
85             if(nearby == 2 || nearby == 3) {
86                 if (cell || nearby == 3)
87                     new_cells[i][j] = 1;
88             }
89         }
90     }
91     GAME.cells = new_cells;
92 };
93
94 let start = function() {
95     // update per 500 ms
96     GAME_INTERVAL = setInterval(update, 500);
97 };
98 /*
99 let stop = function() {
100     console.log("stop");
101     clearInterval(GAME_INTERVAL);
102 };
103 */
104 let fucking_stop = function() {
105     console.log("fucking_stop");
106     clearInterval(GAME_INTERVAL);
107 };
108
109 let step = function() {
110     console.log("step");
111     update();

```

```
112 };  
113  
114 GAME = new game();
```

```
1 <!DOCTYPE html>  
2 <html lang = "en">  
3 <head>  
4     <meta charset = "UTF-8">  
5     <title>Title</title>  
6 </head>  
7  
8 <body>  
9     <canvas id = "game"></canvas><br>  
10    <button onclick="start()">开始</button>  
11    <button onclick="fucking_stop()">停止</button>  
12    <button onclick="step()">步进</button>  
13  
14 </body>  
15 <script src = "game.js"></script>  
16 </html>
```