|  |
| --- |
|  |
| Game of Life 实验 |
|  |
| 说明文档 |

2014-9-30

作者: 董亮 杨志强

Game of Life 实验

说明文档

### 一．实现情况：

* 1. 实验人员：
     + 1. 董亮 2012013305
       2. 杨志强 2012013328
  2. 按照作业内容的要求完成，满足细胞游戏的四个演化条件。
  3. 围棋格子的规格为40x40，每一个小细胞采用div表示，生死的不同状态由不同的css样式表示。
  4. 在对细胞游戏每一轮演化进行计算时，没有采用暴力求解的方式，利用三个数组进行了动态求解，第一个数组为细胞的状态数组cells，第二个数组为temp\_a保存对应cells周围的细胞存活情况（由第一个数组的一次遍历得到）。第三个数组temp\_b由temp\_a复制得到。每进行一次演化时，遍历一次temp\_a，刷新cells的同时也能刷新temp\_b，计算复杂度降低。

### 二．部署情况：

1. 文件上传到github上，游戏页面运行 生命游戏.html文件。
2. 测试文件testRunner.html，可在网页上显示。

### 三．单元测试运行方法：

1. 点击testRunner.html文件。

### 四．分工情况

1. 编写程序：按照要求结对编程，一同完成实验的程序编写部分，一人编写，一人协助，两人轮流交换。
2. 文档编写（董亮）
3. 上传与发布（杨志强）

### 五．结对编程的感受

1. 董亮：
   * 1. 好处：
        1. 编程之前，两个人的思考和讨论有利于将问题分析地清晰，考虑的更细致。
        2. 编程中，协助者可以及时发现编程者的失误，并给予及时的提醒和改正。
        3. 编程中出现问题时，两个人讨论容易解决问题，而一个人往往陷入自己的惯性思维很难自拔。
        4. 遇到bug时，协助者能够较快地帮助编程者找到bug所在或可能的出错方向。
        5. 两个人交替工作，利于减轻疲劳感，并活跃思维。
     2. 坏处：
        1. 结对编程需要建立在熟练掌握基础知识和有一段时间的工作基础上，这样两个人的合作可以形成的较好的互补，但是在目前大量的bug和语法还需要自己查询，js的某些函数的运行机制没有弄清楚的情况下，结对编程容易延长编程的时间，降低了自我学习的效率，很多同学普遍因此放弃转为分工独立编程，对比下速度加快很多。这可能不是结对编程的缺点，但也反映出我们编程基础能力还是有待加强。
2. 杨志强：
3. 好处：
4. 对程序的分析、解决方法、编程思想的交流补充，更有利于编程任务的高质量完成。
5. 双方在互相交流中学习、进步，特别是换种编程的环境下，当有个人在旁协助，可以更有动力提高自己的编程能力，同时发现一些长期积累的不良习惯。
6. 坏处：
7. 结对编程不可避免的会在一些情况下浪费时间，毕竟编写的代码在大多数情况下是不需要有人协助观看，除了一些关键节点，。
8. 心理素质要求提高了，一方面有人在旁的编程感受多少回让人紧张，另一方面编程中的沟通不当会导致编程人员心理受到打击等。

### 六．对单元测试的感受

1. 董亮：
2. 单元测试旨在对于编写程序的最小可测试块进行测试，可以完成一个小函数后立刻进行测试，降低了修改的难度，避免对一堆代码进行测试，出错后查找bug非常麻烦的问题，大大提高了编码效率。同时要求我们在编写代码时就要有大化小的思想，将问题分解，编写多个小函数分步分块解决，还可以降低程序的耦合性，提高代码的复用性，感觉这种编程思想会对于日后的大型工程的部署和解决有很大帮助。
3. 杨志强：
4. 单元测试不仅是看起来的用来在程序后期查询bug、消除bug，更加值得关注的是其实单元测试指引了程序的需求分析、程序结构、编程思想，从这个角度看，单元测试贯穿在了程序编写的全过程。举例来说，程序员要编生命游戏的程序，如果前期不考虑单元测试的话，后期再，至少几乎不可避免的是，程序员会发现自己的代码不适合实现单元测试，原因是耦合太高、代码结构复杂等。