

小作业 1 报告 — 编写生命游戏程序

王永赫¹ - 2014013406 - jlmhkwang@gmail.com

Abstract

小作业 1 要求编写一个生命游戏程序（细胞自动机），要求使用 JavaScript 单元测试框架 mocha (<http://mochajs.org/>) 进行单元测试，技术文档要求说明程序实现的技术细节以及单元测试方案（包括单元测试环境、测试用例设计、测试结果以及运行测试的方法等）。

KeyWords

javascript - mocha - eslint

¹ 清华大学软件学院 - 软件 41 班

目录

1	发布与运行方式	1
1.1	线上 (github-pages)	1
1.2	运行方式	1
2	实现技术细节	1
2.1	模块划分	1
2.2	全局变量	1
2.3	外部库引用	1
3	单元测试方案	1
3.1	环境	1
3.2	方案	1
3.3	周围细胞计数	1
3.4	数据更新	2
3.5	语法静态分析	2

1. 发布与运行方式

1.1 线上 (github-pages)

发布链接：点我
或复制到浏览器地址栏：
<http://cyclops-thss.github.io/LifeGame/>

1.2 运行方式

在 src 子目录中，到终端键入 `npm install`，后打开 `test.html`、`index.html` 即可。

2. 实现技术细节

2.1 模块划分

- 1、绘图显示模块 (render.js)
负责提供向屏幕上的画布绘制图形的接口、控制 `<canvas>` 的边长等。
- 2、游戏逻辑模块 (logic.js)
负责存储进行中数据和进行数据更新，使用绘图显示的接口。

另外直观验证逻辑可以在图形界面上测试，对于任何输入一定会终止计时器（所有全死）或完全稳定（界面不动）或周期变化且周期很小。

3、定时器模块 (timer.js)

负责处理计时和动画相关的事情，使用逻辑接口。

4、界面交互模块 (ui.js)

负责与用户交互和控制游戏，使用其他接口。

5、测试模块 (test.js)

负责开发期间测试各个接口和内部函数，保证其功能稳定。

2.2 全局变量

`g_renderer`: 绘图模块的代表

`g_controller`: 逻辑模块的代表

`g_timer`: 计时器的代表

由于闭包内部函数外部不可见，它们都带有 `test` 成员用来把接口暴露给测试用。

2.3 外部库引用

jQuery、MaterializeCSS

3. 单元测试方案

3.1 环境

mocha (使用 chai 的 `assert`)、代码静态分析使用 `eslint`。测试请打开 `test.html`，运行 `npm test` 会报错（没有导入那些 js）。

3.2 方案

主要测试各个函数和参数（绘图的模块还是需要浏览器上测试），游戏逻辑的测试更详细，如下两个例子：

3.3 周围细胞计数

由于画布无限，数组需要周而复始的访问，这方面设计了测试用例 `[[1, 2, 3], [2, 3, 4], [3, 4, 5]]` 测试几个关键点，比如左上角周围应该返回 26，结果正确，这个步骤保证后续的计算不出问题；

3.4 数据更新

为了验证更新算法是否正确，取一个测试用例 [[0, 1, 0, 0], [1, 1, 1, 0], [0, 1, 0, 0], [0, 0, 0, 0]]，运行一次 update 函数，与标准结果 [[1, 1, 1, 0], [1, 0, 1, 0], [1, 1, 1, 0], [0, 0, 0, 0]] 对比进行验证，结果正确。

3.5 语法静态分析

使用 eslint 保证基本的例如 tab 对齐的统一等不出问题。

Acknowledgments

感谢《软件工程 (3)》的老师 and 助教团队！