

生命游戏2 测试文档

软件32班 王昭

学号: 2013013330

2015 年 10 月 9 日

1 测试环境

mocha: 2.3.3

istanbul: 0.3.21

2 测试模块

2.1 数据结构

Grid类中有三个成员变量: width, height, grid`[][]`。本次测试中测试了grid`[][]`的初始化是否使得其中的所有元素都为DEAD, 测试结果正常。

2.2 接口测试

首先是对setRandomGrid方法进行了测试, 该方法是随机生成满足条件: 活细胞数量与总细胞数量比值属于区间 $[LIVE_DENSE-0.01, LIVE_DENSE+0.01]$ 的地图。本次测试中, 将比值从0.15到0.85每隔0.1进行一次测试, 测试结果正常。

下一个是对get_neighbor_count方法进行了测试, 该方法可以获取地图中某一个非墙壁元素周围扩展四联通区域的活细胞邻居个数, 如果大于等于4, 则返回4。本次测试中, 从地图的四边挑出四个点以及地图中的某一点进行了测试, 结果正常。

接下来测试了update方法, 该方法即按照游戏规则对地图中每个非墙壁元素进行生死状态转换。本次测试中对上一个测试点中挑出的五个点进行了测试, 结果正常。

3 覆盖率测试

本次测试中使用istanbul对测试覆盖率进行检查。检查结果如下：

File	Statements	Branches	Functions	Lines
js/	93.75% (75 / 80)	95.45% (21 / 22)	50% (4 / 8)	93.75% (75 / 80)

其中js中被测试文件有grid.js以及index.js。可以看到，分支覆盖率为95.45%，说明测试还是比较全面的，唯一没有被覆盖的分支是随机生成地图时，如果活细胞与所有细胞个数的比值不满足条件时，那么重新生成，该分支是不一定被覆盖的。没有被测试到的函数是无法或无需测试的函数，因为它们就是一个setInterval 或者clearInterval 等类似函数，因为它们调用的函数是测试过了的，所以它们就不用测试了。

4 测试方法

4.1 mocha

在cmd中转至工程根目录下，输入mocha，即可开始测试。

```
F:\study\大学\大三上\软件工程(3)\workspace\homework2_extended_javascript_game>mocha

Test grid.js
  Test Grid(width, height)
    ✓ after init grid, all elements should be DEAD

Test logic.js
  Test setRandomGrid(dense)
    ✓ live cell number should belongs to [cell number * (dense - 0.005), cell number * (dense + 0.005)]
  Test get_neighbor_count(x, y)
    ✓ should get correct neighbor count, btw, if live neighbor > 4, then return 4
  Test update()
    ✓ should get correct status (live or dead)

4 passing (63ms)
```

在同样的目录下，输入

istanbul cover %APPDATA%\npm\node_modules\mocha\bin_mocha，即可开始覆盖率测试。

```
F:\study\大学\大三上\软件工程 (3) \workspace\homework2_extended_javascript_game>istanbul cover %APPDATA%\npm\node_modules\mocha\bin\_mocha

Test grid.js
  Test Grid(width, height)
    ✓ after init grid, all elements should be DEAD

Test logic.js
  Test setRandomGrid(dense)
    ✓ live cell number should belongs to [cell number * (dense - 0.005), cell number * (dense + 0.005)] (231ms)
  Test get_neighbor_count(x, y)
    ✓ should get correct neighbor count, btw, if live neighbor > 4, then return 4
  Test update()
    ✓ should get correct status (live or dead) (459ms)

4 passing (797ms)

=====
Writing coverage object [F:\study\大学\大三上\软件工程 (3) \workspace\homework2_extended_javascript_game\coverage\coverage.json]
Writing coverage reports at [F:\study\大学\大三上\软件工程 (3) \workspace\homework2_extended_javascript_game\coverage]
=====

===== Coverage summary =====
Statements : 93.75% ( 75/80 )
Branches   : 95.45% ( 21/22 )
Functions  : 50% ( 4/8 )
Lines      : 93.75% ( 75/80 )
=====
```

5 出现的问题以及解决办法

一开始在测试中出现了问题，是在测试网页运行时，自动生成的地图中width和height均为NaN，这是因为在游戏网页中，这两项参数是根据网页中滑动条来获得的，然而在测试网页中并没有，于是导致了错误。解决办法：取消游戏网页打开时自动生成地图，这样就可以避免测试出现问题了。