

# 生命游戏 02 测试文档

李南星 2013013286

## 测试环境

本次测试通过 mocha，覆盖率测试通过 istanbul 完成。

mocha 版本为 2.3.3。

Istanbul 版本为 0.3.22。

系统为 Windows 8.1，64 位。

Python 版本为 3.3.2。

## 测试用例

测试文件为 scripts 目录下的 test.js 文件

本次测试主要对逻辑层进行了测试，有 4 个模块。

### 1. 获取临近活细胞数量的 getNeighbour 函数

- 1) 在给定的地图上进行计算并挑选特殊点进行验证
- 2) 在随机生成的地图上进行计算并对中央点进行验证

```
describe('Testing getNeighbour function', function () {
  it('in a given 5x5 map', function () {
    main.initializeMap(5, 5, 0);
    for (var i = 0; i < 5; i++) {
      (main.map)[i][0] = (main.map)[i - 1][0];
    }
    for (var i = 0; i < 5; i++) {
      for (var j = 1; j < 5; j++) {
        (main.map)[i][j] = ((main.map)[i][j - 1]);
      }
    }
    main.getNeighbour(0, 0).should.equal(0);
    main.getNeighbour(1, 1).should.equal(0);
    main.getNeighbour(0, 1).should.equal(0);
    main.getNeighbour(2, 2).should.equal(0);
    main.getNeighbour(2, 3).should.equal(0);
    main.getNeighbour(4, 4).should.equal(0);
    main.getNeighbour(1, 4).should.equal(0);
  });
  it('in a randomly generated 5x5 map', function () {
    for (var i = 0; i < 10; i++) {
      main.initializeMap(5, 5, 0.5);
      num = (main.map)[0][2] + (main.map)[1][2] + (main.map)[3][2] + (main.map)[4][2] + (main.map)[2][0] + (main.map)[2][1] + (main.map)[2][3] + (main.map)[2][4];
      main.getNeighbour(2, 2).should.equal(num);
    }
  });
});
```

### 2. 进行生长计算的 grow 函数

- 1) 给定地图，进行一次 grow 操作，验证相应位置的细胞死活

```
describe('Testing grow function', function () {
  it('in a given 5x5 map', function () {
    main.initializeMap(5, 5, 0);
    for(var i = 1; i < 5; i++){
      (main.map)[i][0] = !(main.map)[i - 1][0];
    }
    for(var i = 0; i < 4; i++){
      for(var j = 1; j < 5; j++){
        (main.map)[i][j] = !(main.map)[i][j - 1];
      }
    }
    main.grow();
    (main.map)[2][3].should.equal(true);
    (main.map)[2][4].should.equal(0);
    (main.map)[2][2].should.equal(0);
    (main.map)[2][1].should.equal(true);
    (main.map)[1][3].should.equal(0);
    (main.map)[0][3].should.equal(true);
    (main.map)[4][3].should.equal(0);
  });
});
```

### 3. 进行重置操作的 reset 函数

1) 随机生成地图，进行 reset 操作，验证地图是否为空

```
describe('Testing reset function', function () {
  it('after resetting a randomly generated 100x100 map', function () {
    main.initializeMap(100, 100, 0.5);
    main.reset();
    for(var i = 0; i < 100; i++)
      for(var j = 0; j < 100; j++)
        (main.map)[2][3].should.equal(0);
  });
});
```

### 4. 进行生成的 initializeMap 函数

1) 验证函数能否对相应参数进行正确修改

2) 验证函数的随机生成功能是否准确（随机设定密度，进行多次随机生成并统计验证、重复多次）

```
main.initializeMap(63, 23, 0.33);
main.returnWidth().should.equal(63);
main.returnHeight().should.equal(23);
main.returnRate().should.equal(0.33);
});
it('testing random rate generation effectivity', function () {
  for(var i = 0; i < 10; i++){
    rate = Math.random();
    var num = 0;
    for(var j = 0; j < 10; j++){
      main.initializeMap(100, 100, rate);
      num += main.returnLives();
    }
    ((num / (100 * 100 * 10)) / rate).should.be.within(0.98, 1.02);
  }
});
});
```

## 测试结果

mocha 测试结果全部通过。

```
Testing getNeighbour function
  ✓ in a given 5x5 map
  ✓ in a randomly generated 5x5 map

Testing grow function
  ✓ in a given 5x5 map

Testing reset function
  ✓ after resetting a randomly generated 100x100 map

Testing initializeMap function
  ✓ testing setting effectivity
  ✓ testing random rate generation effectivity <1650ms>

6 passing (2s)
```

istanbul 测试结果如下：

```
===== Coverage summary =====
Statements : 100% < 160/160 >
Branches   : 93.33% < 28/30 >
Functions  : 100% < 18/18 >
Lines      : 100% < 160/160 >
=====
```

## 测试运行方法

在 test.js 所在目录下运行 mocha test.js 即可进行单元测试。

如图：

```
C:\Users\Asthen\Downloads\eeeexpi\scripts>mocha test.js

Testing getNeighbour function
  ✓ in a given 5x5 map
  ✓ in a randomly generated 5x5 map

Testing grow function
  ✓ in a given 5x5 map

Testing reset function
  ✓ after resetting a randomly generated 100x100 map

Testing initializeMap function
  ✓ testing setting effectivity
  ✓ testing random rate generation effectivity

6 passing (76ms)
```

由于环境问题，在进行覆盖率测试时，我不得不找到 mocha 的绝对路径进行 istanbul 测试（同样在 test.js 所在目录运行命令），如图：

```
C:\Users\Asthen\Downloads\eeeexp1\scripts>istanbul cover C:\Users\Asthen\node_modules\mocha\bin\_mocha -- -R spec test.js

Testing getNeighbour function
  ✓ in a given 5x5 map
  ✓ in a randomly generated 5x5 map
```

如果环境没问题，运行 `istanbul cover _mocha test.js` 应当能够正常进行覆盖率测试。