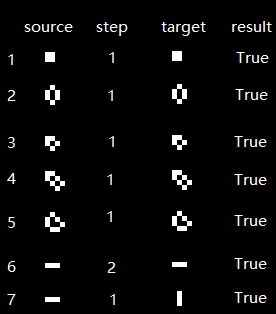
# 生命游戏说明文档

向首兴

2014013421

[xsx1996@163.com](mailto:xsx1996@163.com)

* 程序框架：
  + index.html是游戏显示的基本框架；
  + life\_game\_logic.js中描述了生命游戏的逻辑规则
    - calculate函数用于计算下一时刻的所有单位的存活情况
    - update函数用于依据calculate计算结果进行更新
  + life\_game\_render.js中描述了游戏的绘制、控制，包括画布等各个标签的建立与事件响应。
    - render函数在canvas绘制当前时刻的生命分布情况。
    - timer以固定频率不断调用，模拟生命的周期变化。
    - random\_button.onclick会在游戏停止状态的时候对所有生命随机赋予状态，存活状态可能性为50%。
    - clear\_button.onclick会在游戏停止状态的时候杀死所有生命。
    - game\_switch.onclick更改游戏状态，在停止和运行间切换。
    - canvas.onclick会在游戏停止状态的时候更改点击单元的生命状态。
* 技术细节：
  + 逻辑：
    - 两个二维数组，一个存储当前点阵，另一个用于计算下一周期的点阵。
    - 计算过程是简单遍历所有单元，考虑其周围8个单元的情况，复杂度为：8\*单元数。
  + 控制：
    - 随机布局：利用Math的随机数对每一个单元的生命状态随机赋予。
    - 点击自主布局：通过画布的点击坐标确定所点击的单元，更改其生命状态。
    - 通过setTimeout实现时钟。
* 单元测试：
  + 环境：
    - 操作系统Windows 10 教育版
    - Node 6.6.0
    - 浏览器：Chrome
    - Npm：3.10.3
    - Node包：
      * Mocha：3.0.2
      * Chai：3.5.0
  + 代码测试：
    - get\_value：测试是一个两个参数的函数，以及其参数的边界判断，测试通过。
    - calculate和update：测试是一个无参数的函数。
  + 游戏逻辑测试用例与结果：



共7个测试案例，source是案例的初始图形，step是经历的变化回合数，target是案例的正确结果，result是该案例的通过情况。

左图可见，七个案例都通过了测试。

* + 运行测试的方法：
    - 双击运行life\_game\_test.html即可查看测试结果。
    - 双击运行index.html，打开审查元素，在命令行运行show\_example()函数即可看到七个测试案例的可视化图形。

感谢您的阅读