# 生命游戏说明文档

一、技术细节

1、细胞的表示：在这个游戏中，我用div块来显示细胞，一个div块对应于一个细胞。为了能够让用户一眼就看出哪些是活细胞哪些是死细胞，我将死细胞设置成黑色，活细胞设置成了红色，用户可以很明显的看出来。此外，用户在点击细胞(即div块)的时候可以改变细胞的状态，即将死细胞变为活细胞，将活细胞变为死细胞，我用onclick来触发此效果。

2、细胞信息的存储：我用一个二维数组(cells)来存储对应的细胞，每一个数组元素对应于相关id的div块，数组中的元素为1说明该细胞为活的，元素为0说明该细胞为死的。如此，在计算下一次整个地图的细胞状态的时候就可以简单地用数组来计算了，在计算的时候对于第i行第j列的格子，我们只需要对cells[i-1][j-1]、cells[i-1][j]……cells[i+1][j]、cells[i+1][j+1]这八个数组元素进行求和就可以知道cells[i][j]周围的活细胞数量，从而决定该细胞下个状态是死是活，这是非常方便的。

3、高度自由化的设计：在我的程序中，用户可以指定初始时候的横纵格子的数量和初始化的活细胞数目以及演变的速率。对于横纵格子数量，我只是在初始化的时候根据这些数量修改了div块的长和宽，并控制了随机函数中初始化的细胞数量。而对于演变，我写了一个函数cellschange，专门用于进行演变，没进行一次演变就需要调用一次这个函数，而演变的速率则通过setInterval中的时间参数来控制，当活细胞数为零时，不能够继续演变。

4、人性化的附属功能：我在游戏控制里面增加了速度控制、运行和暂停、下一次演变和清空界面按钮。速度控制可以让用户在不满意演变的速度时对演变速度进行修改，他只是简单地修改了一下setInterval的时间参数，为了安全，只能够在演变暂停的时候调控速率。运行和暂停按钮很简单，只是调用了setInterval和clearInterval。下一次演变也是调用了一下演变函数cellschange。至于清空界面按钮，就是将屏幕上所有div块的背景色设置成黑色，然后将记录活细胞数的living参数修改为0就可以了。此外，我还在页面中专门开辟了一小块界面用于显示程序当前的状态，包括已经演变的次数、当前的活细胞数和演变速度等状态都可以显示出来，让用户可以直观地了解当前程序。

二、单元测试方案

1、单元测试环境：mocha 3.0.0

2、测试用例：在本次测试中，我的测试设置是这样的：32行、50列，初始细胞数300个，演变速率240次/分。

3、测试结果：

①对于init函数，我只是让它简单地画出有的div块，而暂时不生成活细胞(这在initcell函数里面进行)，所以我需要检查的就是它是否生产了足够多的div块并且没有产生活细胞。通过测试，我发现它生成了1600个div块，与32\*50相等，并且没有生成活细胞，因此这个函数是符合要求的。

②对于initcell函数，是在init函数已经生成了所有div块的基础下生成细胞。由于init函数可能失败，因此在失败的时候不能调用initcell函数，并且initcell函数应该生成对应的活细胞个数并且当要求的活细胞数量太多时报错。

在测试的时候，当init函数初始化失败的时候，initcell中的有效内容也不会被执行，这是符合要求的。而对于活细胞的数量，我则是通过遍历所有div块，将处于活细胞状态细胞数量统计一下，为300个，与目标要求一致。在测试活细胞数量太多时，我直接将目标活细胞数量设置成了1600，显然正常的程序应当不允许这种现象存在（程序中我的限制是活细胞数量不能超过总细胞数量的90%），而程序也确实报错并且没有执行活细胞初始化，因此此函数的实现应当是正确的。

③对于getid函数，它的作用有两个，一个是对于给定的行和列，获取对应位置的div块的id，方便修改div块的颜色；另一个是根据div块的id，获取对应的行和列，方便进行数组操作。我对其要求是，输入两个数字参数时，能够返回一个字符串变量，为对应的div块id；而输入一个字符串变量时，能够返回一个含有两个参数的数组，为div块对应的行和列；当输入其他变量时，应该不执行特殊操作并且返回0。在实测中其是完全符合要求的。

④对于clear函数，其作用是清除图上所有的活细胞，也就是在不删除div块的作用下，将所有的活细胞变成死细胞。在调用了这个函数以后，我检查了所有的div块，发现他们均已被设置成了黑色背景，这说明clear函数被正确地执行并发挥作用了。

⑤对于cellchange函数，其作用是进行一次演变，即根据此时所有细胞的状态来决定下一次所有细胞的状态。由于细胞的变化较难检测，我只检查其运行状态是否正确。即在程序处于不正常状态（失败状态）或者细胞数为0的时候其是否进入了对应的运行状态，而在细胞数不为零的时候其是否进行了真正的演变，通过检查函数的返回值，我得到了这个函数正常运行的结论。

三、总结

通过自己的代码设计与检测，这次的程序正常运行。