

实验课程名称： 软件工程基础实验

实验项目名称	结对编程			实验成绩	
实 验 者	贺秋程	专业班级	软件 1804	组 别	
同 组 者	罗治			实验日期	2020.5.14

第一部分：实验预习报告（包括实验目的、意义，实验基本原理与方法，主要仪器设备及耗材，实验方案与技术路线等）

一. 实验内容及要求

1. 选择一个程序实例，练习结对编程实践
2. 要求学生两人一组，自由闭合。每组使用一台计算机，二人共同编码，完成实验要求
3. 要求在结对编程工作期间，两人的角色至少切换四次
4. 编程语言不限

二. 问题描述

生命游戏：

游戏在一个类似围棋棋盘，可以无限延伸的二维方格网进行。每方格中都放置一个生命细胞，细胞只有“生”与“死”两种状态。以黑色代表“生”，白色代表“死”。游戏开始时，每个细胞可以随机地（或给定地）被设定为“生”或“死”之一的某个状态，然后再根据如下生存定律计算下一代每个细胞的状态：

- 1) 每个细胞的状态由该细胞及其周围 8 个细胞上一次的状态所决定；
- 2) 如果一个细胞周围有三个细胞为生，则该细胞为生，即该细胞若原先为死则转为生，若原先为生则保持不变；
- 3) 如果一个细胞周围有两个细胞为生，则该细胞生死状态保持不变；
- 4) 在其他情况下，该细胞为死，即该细胞原先为生则转为死，若原先为死则保持不变

第二部分：实验过程记录（可加页）（包括实验原始数据记录，实验现象记录，实验过程发现的问题等）

三. 算法设计思路

分两个类实现功能，一个为 class Cell，另一个为 class GUI

1. class Cell

1) 定义一个数组代表棋盘，每一个数据代表一个细胞，0 代表死，1 代表生。

2) 实现棋盘初始化，遍历数组，将棋盘所有细胞赋予“死”的状态，数组的行数列数取为设定长度再加上 2。边界判断不能越界，所以每个边界要多申请一格。

3) 定义方法用于确定某个细胞周围细胞的数量，将循环范围限制到给定坐标加减一个单位：

```
private int getNeighborCount(int i1, int j1)//参数用于定位该细胞
{
    int count = 0;
    for (int i = i1 - 1; i <= i1 + 1; i++)
        for (int j = j1 - 1; j <= j1 + 1; j++)
            count += grid[i][j]; //如果周围细胞还活着，周围细胞数便会+1
    count -= grid[i1][j1]; //每个细胞不是自己的周围细胞，则如果细胞还活着，周围细胞数-1.
    return count;
}
```

4) 定义方法实现细胞繁衍功能，新建一个数组用于存储新一代细胞，遍历该新数组，调用 getNeighborCount 方法，根据返回值确定新的细胞状态：

```
public void update() {
    int[][] newGrid = new int[maxLength + 2][maxWidth + 2];
    for (int i = 1; i <= maxLength; i++)
        for (int j = 1; j <= maxWidth; j++)
            switch (getNeighborCount(i, j)) {
                case 2:
                    newGrid[i][j] = grid[i][j]; //周围有 2 个细胞，细胞状态保持不变
                    break;
                case 3:
                    newGrid[i][j] = 1; // 周围有 3 个细胞，细胞存活
                    break;
                default:
                    newGrid[i][j] = 0; // 其他情况，细胞死亡
            }
    for (int i = 1; i <= maxLength; i++)
        for (int j = 1; j <= maxWidth; j++)
            grid[i][j] = newGrid[i][j]; //把新一代细胞的情况赋值给原数组
    nowGeneration++;
}
```

5) 定义方法实现其他功能:随机初始化细胞, 遍历数组将每个元素赋值为 0 或 1 中任意一个; 细胞清零, 遍历数组将所有元素置零; 代数清零, 将当前代数置零;

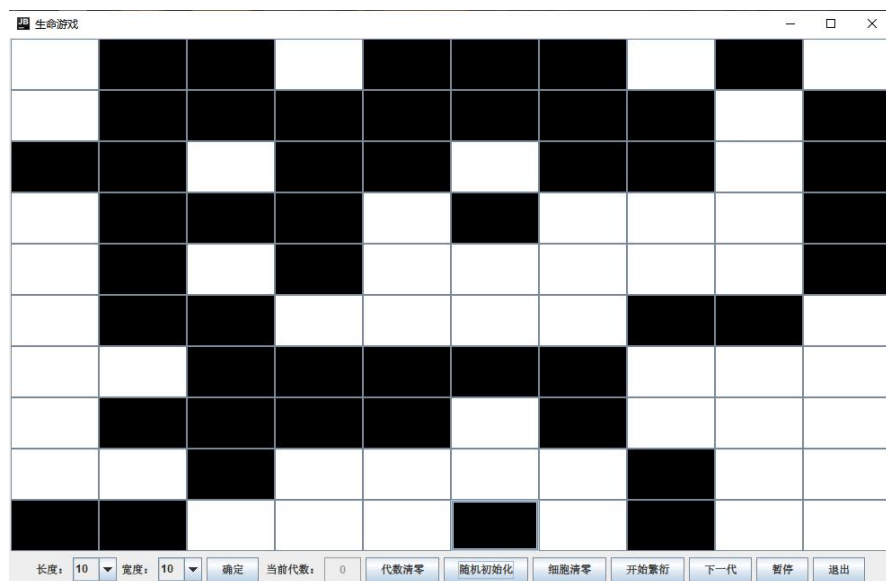
2. class GUI

1) 设置窗体各个组件: , 标签, 按钮, 下拉框等等, 并为他们设置长宽, 可视化以及对于位置。采用边界布局将棋盘设置在中间位置, 把按钮功能部分设置在棋盘下方。棋盘中的各个格子都设置为按钮形式, 将初始颜色都设置为白色, 代表“死”状态。按钮分别有“代数清零”“随机初始化”“细胞清零”“开始繁衍”“暂停”“退出”。

2) 定义方法 showCell, 遍历数组根据数值展示细胞存活状态: 若数值为 1 则将对应该格子的背景颜色设置为黑色; 若为 0 则设置为白色。

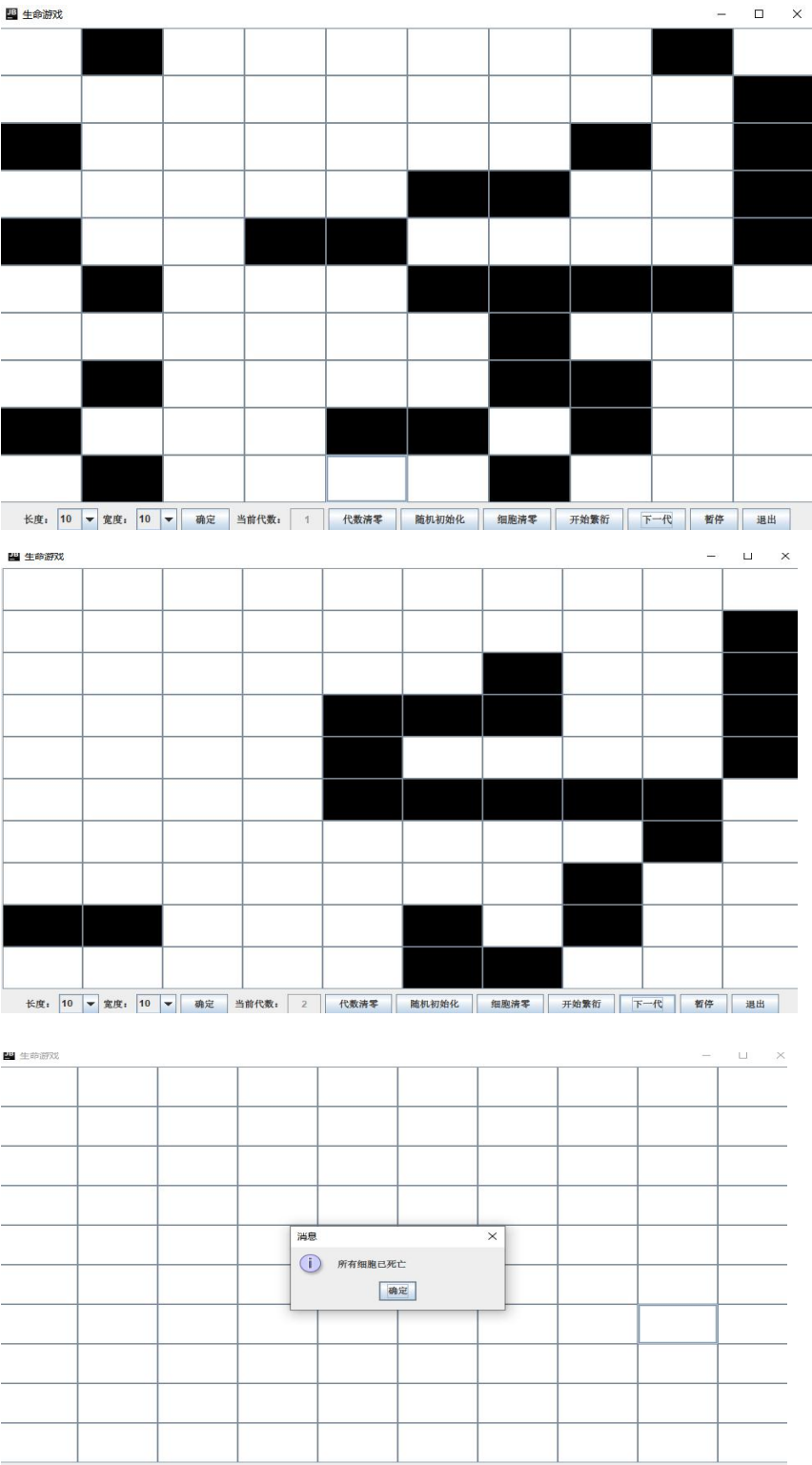
3) 为所有按钮组件设置监听事件, 根据 e.getSource 判断鼠标点击的是那个按钮从而取执行对应的事件。“代数清零”: 将当前代数置零并在按钮后边的文本框中显示出来; “随机初始化”: 点击之后棋盘上会自动出现随机数量黑格; “细胞清零”: 将棋盘上的所有格子置为白色; “开始繁衍”: 点击后程序根据棋盘上已有的细胞不断调用方法 update 和 showcell, 并将结果实时显示, 在此需要设置线程, 体现出细胞更新换代的过程, 使用函数 Thread.Sleep, 并且每次更新都进行一次判断, 如果棋盘上所有细胞均死亡则返回提示; 棋盘本身也是按钮, 用户不需要初始化直接点击棋盘也可以初始化初始棋盘中细胞的存活, 也可以在细胞更新换代的过程中点击, 有着同样的效果。; “退出”点击之后程序直接停止并退出。

四. 实验结果分析

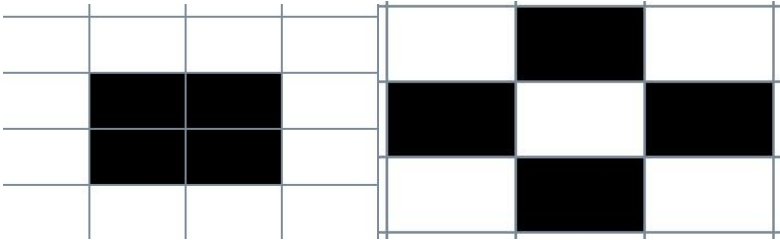


第三部分 结果与讨论（可加页）

一、实验结果分析（包括数据处理、实验现象分析、影响因素讨论、综合分析和结论等）



并且发现，当细胞之间存在某些组合并且周围不在产生新的细胞时，他们就会一直存活：

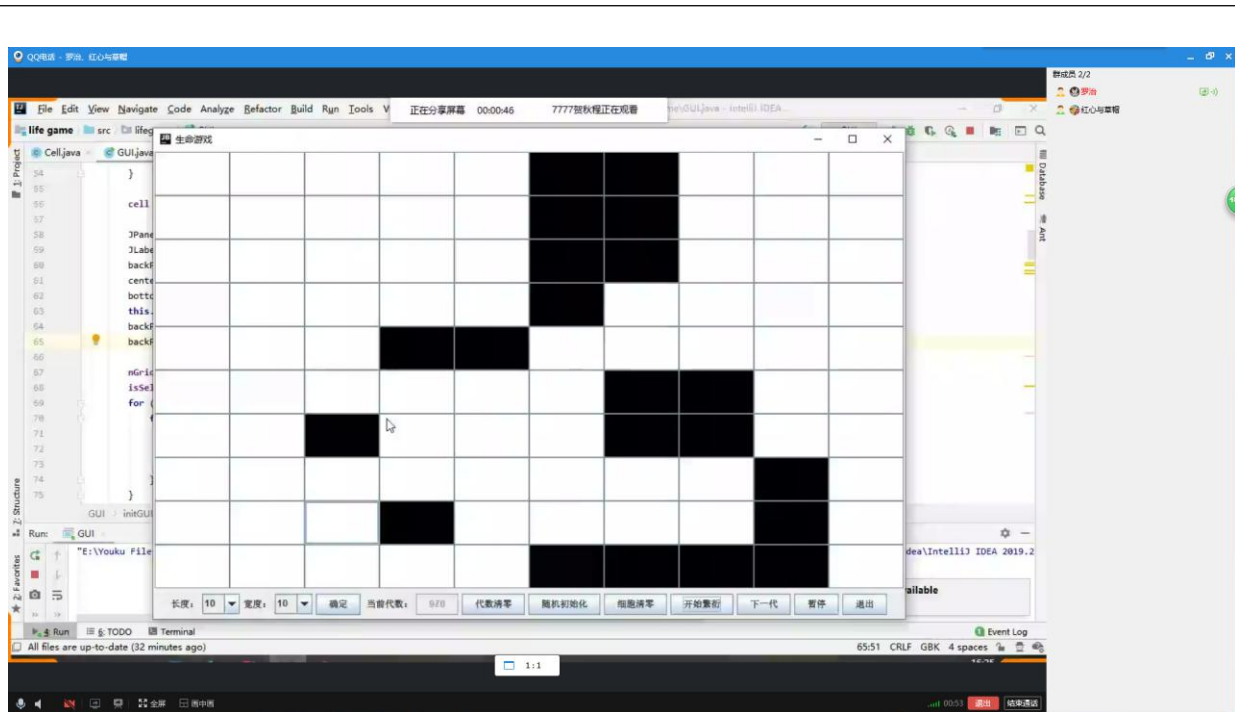


时间	“驾驶员”	“领航员”	具体分工
2020.05.14 20: 30-22: 30	罗治	贺秋程	罗治负责 cell 类实现，贺秋程查找相关算法
2020.05.16 14: 30-16: 30	贺秋程	罗治	贺秋程负责 cell 字符界面的查错和修改，罗治查找 GUI 相关函数
2020.05.17 18: 30-20: 30	罗治	贺秋程	罗治负责 GUI 的界面编程，贺秋程查找相关算法以及和字符界面的对接
2020.05.18 9: 30-11: 30	贺秋程	罗治	对代码进行测试，并对错误进行修改

```

70 for (int j = 0; j < maxLength; j++) {
71     nGrid[i][j] = new JButton( text: "" ); // 数组内每置空以表示颜色
72     nGrid[i][j].setBackground( Color.WHITE ); // 初始时所有按钮均为白色
73     centerPanel.add( nGrid[i][j] );
74 }
75
76
77 jLength = new JLabel( text: "长度: " );
78 lengthList = new JComboBox();
79 for (int i = 3; i <= 100; i++)
80     lengthList.addItem( String.valueOf( i ) );
81 lengthList.setSelectedIndex( maxLength - 3 ); // 设置后初始最小值为3，最大值3，下同
82
83 jWidth = new JLabel( text: "宽度: " );
84 widthList = new JComboBox();
85 for (int i = 3; i <= 100; i++)
86     widthList.addItem( String.valueOf( i ) );
87 widthList.setSelectedIndex( maxWidth - 3 );
88
89 ok = new JButton( text: "确定" );
90 jHowGeneration = new JLabel( text: "当前代数: " );
91 jHowGeneration = new JButton( text: "" ); // 调用 cell.getHowGeneration() // 调用 cell.getHowGeneration() 方法
92 jHowGeneration.addActionListener( new ActionListener() {
93     public void actionPerformed() {
94         GUI.run();
95     }
96 } );

```



二、实验小结及体会

- 1.通过本次实验回顾了 JAVA 相关的语法以及算法。
- 2.体会到了结队编程的好处以及优点。
- 3.体会到了结队编程和自己编程的区别。
- 4.虽然前期的进度比较的慢，但通过逐渐的磨合之后进度进展的十分之快。
- 5.结队编程能让编程变得不那么枯燥。

成绩评定表：

序号	评分项目	满分	实得分
1	实验报告格式规范	2	
2	实验报告过程清晰，内容详实	4	
3	实验报告结果正确性	2	
4	实验分析与总结详尽	2	
	总得分	10	

