

实验课程名称： 软件工程基础实验

实验项目名称	结对编程--生命游戏			实验成绩	
实 验 者	张宸	专业班级	软件 2001	组 别	
同 组 者	涂典			实验日期	2022.5.10

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、实验目的及意义

- 1. 体验敏捷开发中的两人合作。
- 2. 进一步提高个人编程技巧与实践。
- 3. 体会结对编程的优势，提升与队友的默契程度。
- 4. 学习生命游戏的基本原理以及代码编写。

二、实验基本原理与方法

- 1. 选取编写生命游戏（Life Game）作为练习结对编程的实践。
- 2. 两人使用一台计算机，对生命游戏进行代码编写。
- 3. 使用 Python 进行编程。
- 4. 一人作为驾驶员，负责使用键盘对代码的编写。
- 5. 一人作为领航员，负责使用鼠标起到领航，提醒的作用。
- 6. 两人再合适的时间进行角色互换，保证效率与正确率。

三、问题描述

生命游戏没有游戏玩家各方之间的竞争，也谈不上输赢，可以把它归类为仿真游戏。事实上，也是因为它模拟和显示的图像看起来颇似生命的出生和繁衍过程而得名为“生命游戏”。在游戏进行中，杂乱无序的细胞会逐渐演化出各种精致、有形的结构；这些结构往往有很好的对称性，而且每一代都在变化形状。一些形状一经锁定就不会逐代变化。有时，一些已经成形的结构会因为一些无序细胞的“入侵”而被破坏。但是形状和秩序经常能从杂乱中产生出来。

游戏开始时，每个细胞随机地设定为“生”或“死”之一的某个状态。然后，根据某种规则，计算出下一代每个细胞的状态，画出下一代细胞的生死分布图。每个细胞迭代后的状态由该细胞及周围 8 个细胞状态所决定。

生存规则：

- (1) 当前细胞为死亡状态时，当周围有 3 个存活细胞时，则迭代后该细胞变成存活状态(模拟繁殖)；若原先为生，则保持不变。
- (2) 当前细胞为存活状态时，当周围的邻居细胞低于两个(不包含两个)存活时，该细胞变成死亡状态(模拟生命数量稀少)。
- (3) 当前细胞为存活状态时，当周围有两个或 3 个存活细胞时，该细胞保持原样。
- (4) 当前细胞为存活状态时，当周围有 3 个以上的存活细胞时，该细胞变成死亡状态(模拟生命数

量过多)。

最后根据生命游戏的生存规则来不停迭代，得到 n 次迭代下的生死分布图。

四、核心算法设计思路

数据结构设计：

本次实验利用二维数组对游戏地图进行存储。

核心算法设计：

1. 生成初始化生死分布图

在网上查阅资料时发现初始化存活概率大多为 0.2，所以本次实验也采用 0.2 作为初始存活概率。初始化游戏地图时，首先建立一个二位零矩阵作为游戏地图，遍历二维游戏地图，经过每个点时随机生成一个 1 到 10 的随机数，若数字小于等于 2 则将该点定义为活细胞即将二维数组上该店定义为 1。

2. 每次迭代对细胞生死的判断

首先建立一个新的二维零矩阵作为中间变量来存储下一次的生死分布图。遍历二维游戏地图，根据生命游戏的规则，将每个细胞的生死更新在新的二维矩阵中，在将新的二维矩阵赋值给原来的游戏地图。

五、主要仪器设备及耗材

1. 安装了 Windows 10 操作系统的 PC 机 1 台
2. PC 机系统上安装了 PyCharm 开发环境，并安装有 Python 3.9 的解释器以及 numpy, pygame 等库。

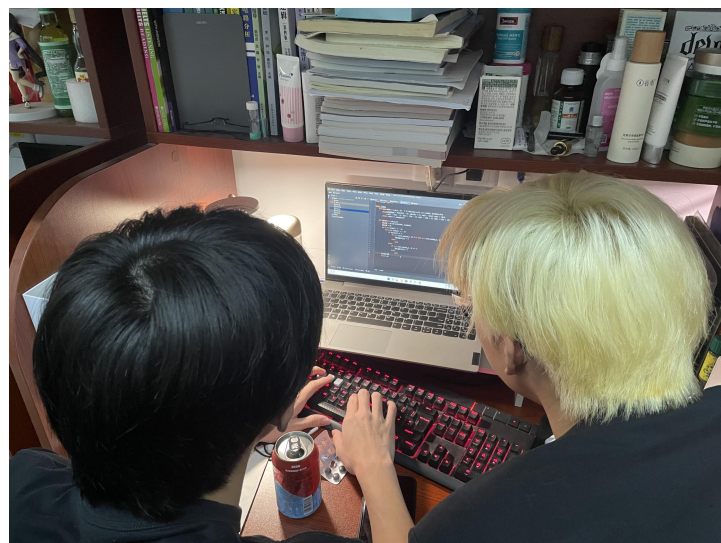
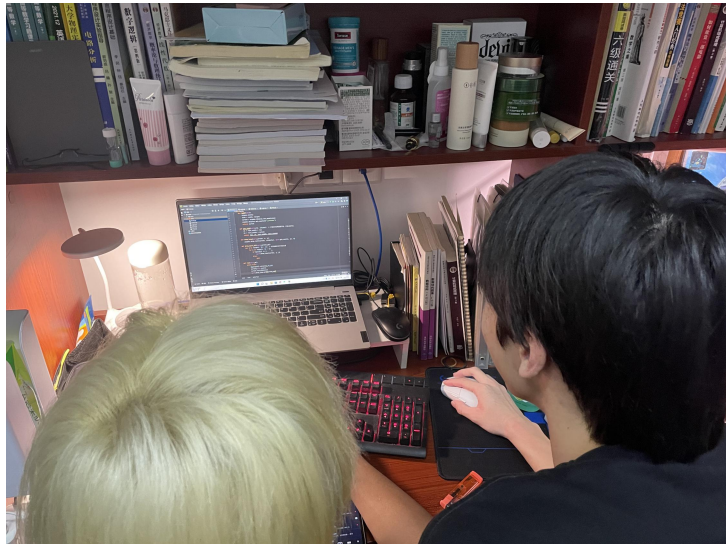
第二部分：实验过程记录（可加页）（包括实验原始数据记录，实验现象记录，实验过程发现的问题等）

一、结对编程过程

本次实验我与我的队友进行了几次交换，具体情况如下：

1. 首先我先负责编写生命游戏核心代码，队友负责口述生命游戏过程以及代码逻辑的检查。
2. 第一次交换，队友负责编写代码，使用 `pygame` 将游戏可视化，并将核心代码写入类中，实现面向对象编程，我负责领航与检查。
3. 第二次交换，我负责编写一些额外的功能例如开始、暂停、重置以及随机的功能，我的队友进行领航与检查。
4. 第三次交换，队友负责编写代码，将 `Botton` 类封装，我负责检查代码与领航。

以下为本次结对编程工作的照片：



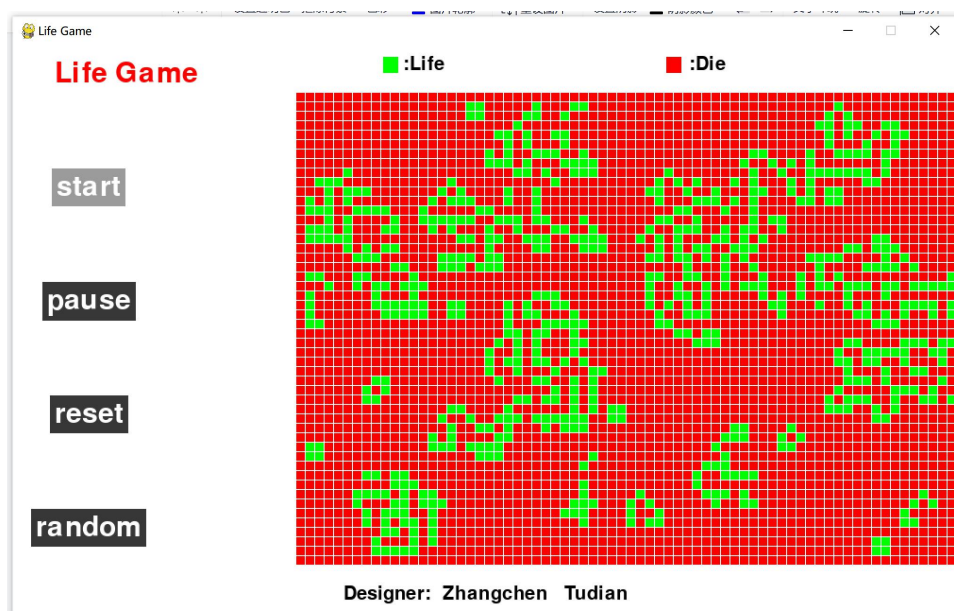
二、实验现象的记录

1.启动程序时，系统弹出生命游戏的 GUI 界面

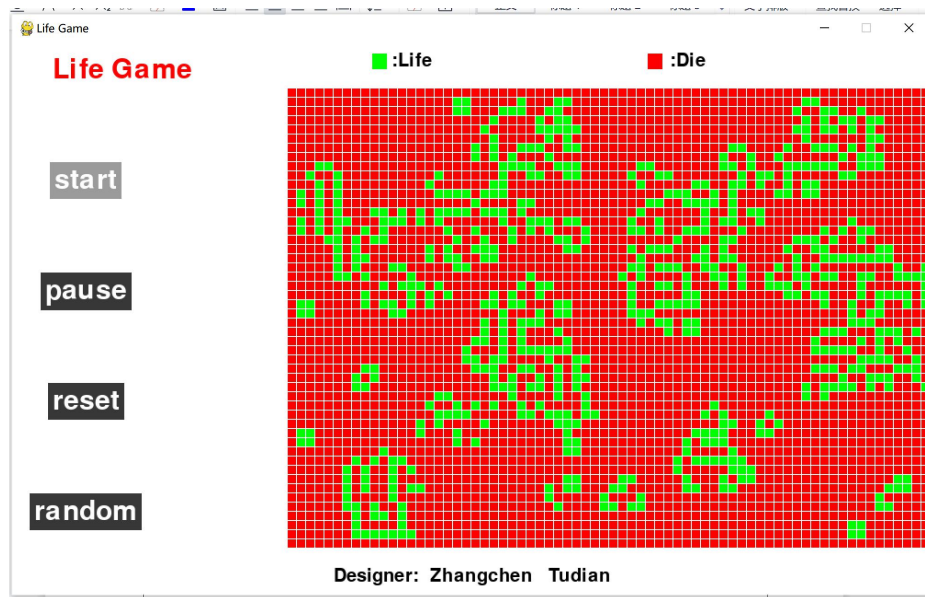


初始界面

2.点击 start 键后，生命游戏开始运行，地图先随机生成，然后对当前地图进行迭代演绎，GUI 界面中会显示每一轮的细胞生死图。（绿色为活细胞，红色为死细胞）

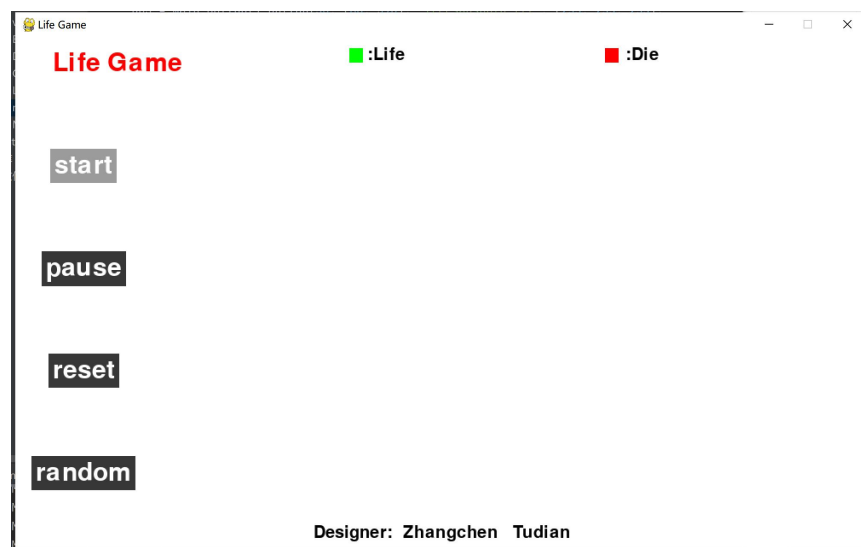


点击 start 后随机生成的初始地图



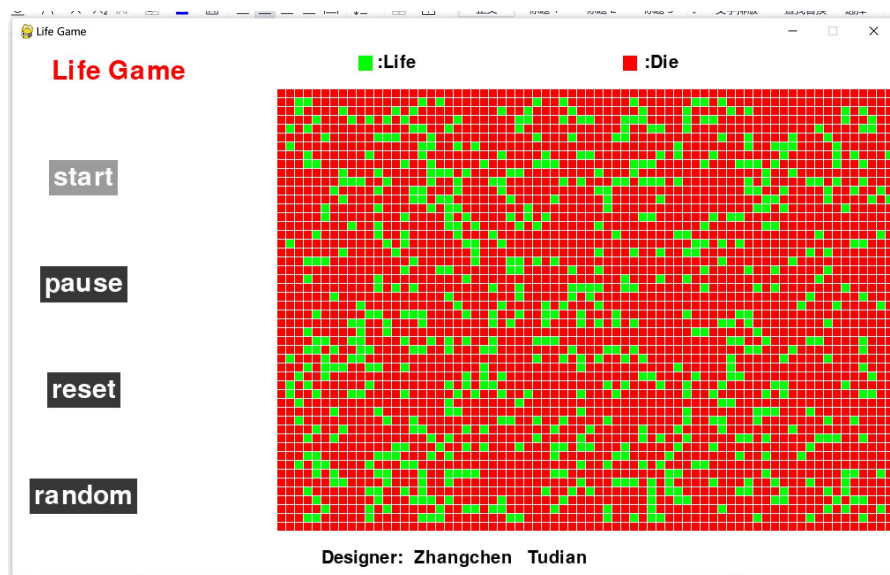
演绎一次后的游戏地图

3.按下 reset 键后，系统将原有的界面进行重置回初始界面。



点击 reset 后的界面

4.按下 random 键后，系统将游戏地图按照初始化规则进行初始化，得到一张新的游戏地图。



点击 random 后随机生成的初始化地图

三、实验过程中发现的问题

实验时发现了如下问题

1. 结对编程中两人的编程习惯问题，由于两人编程习惯上的不同，使得两人开始结对编程时不是很默契，但在实验过程中不断适应大家都产生了默契。
2. 在对实验代码进行面向对象编程时，发现对 Python 的面向对象编程并不是很熟悉。不过在网查阅资料后学会了如何使用面向对象的方式编程。
3. 在对 GUI 的设计的时候，对 pygame 等一些库的不熟悉也使得 GUI 的设计变得困难。但在请教同学和在网上学习后，学会了 GUI 的基本设计，并在不断地调试中使自己设计的 GUI 界面达到自己想要的效果。
4. 代码编写规范问题，这次实验算是一个小的工程，在编写工程文件时函数的分配以及函数的调用都有些不规范，在不断的 Debug 中将这类问题处理好了。

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）

本次实验逻辑上就是一个模拟细胞生死的过程，输出数据，数据的表现都集中在了 GUI 界面中的游戏地图中，经过自己的随机抽样验证，经检验所得结果是与生存规则手动演绎的一致。大体上可以判断结果是正确的。

二、实验小结及体会

本次实验与队友体验了结对编程，体验到了结对编程的优势，在效率上，结对编程可以最大化的结合两个人的能力实现互补，因为领航员的存在也能让程序员在高效率的情况下，保证代码的正确性，也能提升编写速度。但是结对编程需要两个队员的不断磨合与适应，克服两人在代码编写习惯与个性上的差异。但是结对编程可以将两人的优势互补，在各自擅长的领域进行编写。在这次实验中明显可以感受到结对编程带来的优势。

本次实验中也遇到了许多问题，因为 Python 也不是很熟练，对 Python 的库和方法都不是特别熟悉，也是在网上不断地查阅资料的同时学习代码的编写以及库和方法的使用。另外本次实验要求使用面向对象的编程，所以在学习如何使用 Python 编写面向对象编程时也遇到和很多困难，在克服了这些困难后，对核心代码的编写其实就是一个模拟的过程，十分简单明了。结对编程的优势在本次实验中十分的明显，同时两人的氛围下可以让这次实验边的更轻松。本次实验粗略的学习到了 Python 的一些库，例如 numpy, pygame, random。这些库能帮助我以最简单的方式写出我需要的特殊功能，极大的提升了代码的可读性以及程序的完整性，同时也将我的程序的结果以 GUI 形式显示在控制台。

本次也算是一次完整的工程，让我对敏捷开发，结对编程以及面向对象编程有了更深的理解。在本次实验中也有许多的语法上的小问题，主要是对 Python 语言的不了解。以后可以通过多多练习和多多学习，来加强自己 Python 编写能力以及编写技巧。

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	

