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	索键	专业班级	软件 2001	组　　别	
同 组 者	杨谨荣			实验日期	2022.5.5

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

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。

二. 意义

1）编程实现了生命游戏，在结对编程中互相学习，按时完成项目，并且保证了代码的高质量。

2）在实验中组内进行多次轮换，轮流完成领航员和驾驶员的工作。

3）实验使用了分层思想，将实验代码分成了多个模块，每个模块都能完成自己的功能，最终将它们组合在一起，实现整体功能。

4）实验中使用 python3.9，并通过 python 的第三方库 pyGame 实现生命游戏的 UI 设计，并通过定义 Button 类实现了 UI 界面的按钮设计。充分锻炼了组内成员的 python 编程能力

三. 实验基本原理与方法

本次试验中，本组选择的模拟程序为生命游戏。

1）生命游戏简介

生命游戏是英国数学家约翰·何顿·康威在 1970 年发明的细胞自动机，它包括一个二维矩形世界，这个世界中的每个方格居住着一个活着的或死亡的细胞。一个细胞在下一个时刻生死取决于相邻八个方格中活着的或死了的细胞的数量。如果相邻方格活着的细胞数量过多，这个细胞会因为资源匮乏而在下一个时刻死去；相反，如果周围活细胞过少，这个细胞会因太孤单而死去。

游戏在一个类似于围棋棋盘一样的，可以无限延伸的二维方格网中进行。例如，设想每个方格中都可放置一个生命细胞，生命细胞只有两种状态：“生”或“死”。图中用黑色的方 格表示该细胞为“死”,其它颜色表示该细胞为“生”。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 3 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 2 个细胞为生，则该细胞的生死状态保持不变；

④ 在其它情况下，该细胞为死，即该细胞若原先为生则转为死，若原先为死则保持

2）本次实验模拟方法

使用 python 对生命游戏进行模拟。使用 60x40 大小的二维数组储存某一时刻的地图，地图中活着的细胞被标记为一，死去的细胞标记为零。在点击开始游戏后，每隔 500

毫秒对当前地图进行判断。使用双层 for 循环对当前地图进行扫描，判断地图上每个细胞是否存活。遍历完成后将新生成的二维数组赋值给原数组，最后在 pygame 中调用地图，并将它打印在屏幕上。

四. 主要仪器设备与耗材

Win10 操作系统，Pycharm2021，

第二部分：实验过程记录

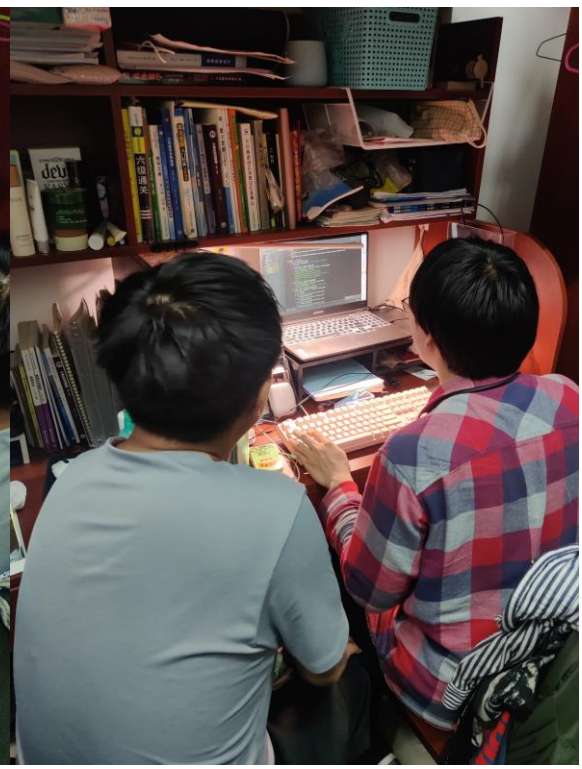
结对编程：

角色互换时间点：global.py,编程；memSet.py 编程; gameLogic.py 编程；用户 UI 界面(uiFunction.py)编程(过程中多次交换)；代码优化及调试

各自的任务：

索键：global.py, gameLogic.py 编程；用户 UI 界面(uiFunction.py)编程；代码优化；查询相关资料；实验报告

杨谨荣：memSet.py 编程；用户 UI 界面编程(uiFunction.py)；代码优化；查询相关资料；实验报告



一、 **Global.py**

该文件中储存了一些生命游戏中常用的公共变量：

棋盘宽度：Width=60

棋盘高度：Height=40

每个生命周期长度（毫秒）：Delay=500

细胞宽度（像素）：cellRound=20

除此之外还定义了一些在文件中常用的颜色，例如背景色 backgroundColor = (240, 255, 240)，活细胞颜色 liveCellColor = (0,0,0)，死细胞颜色 deadCellColor = (255,255,255)。

通过 Global.py 的使用，可以大大缩短代码的长度，并会让阅读者更容易理解代码。

二、 **memSet.py**

在 memSet 类中定义了有关数组的操作。最主要的是创建数组 getMemSet 和初始化数组 initWork。前者的功能是创建一个合适大小的数组，这一功能是为了对代码进行封装，从而改变变量的可见性，让代码更不容易被破坏。后者是将数组进行随机化操作，通过双层 for 循环和 0 到 1 间的随机数 rand，为每个格子赋初值。若 rand 在 0.8 之上，则将数组该位置的值赋为 1，反之则设为 0，从而让初始棋盘上大约存在五分之一的活细胞。

三、 **gameLogic.py**

该类中定义了有关判断逻辑的方法。目前只有 workEvolution 一个方法。该方法用于模拟繁殖规则。对于某个死细胞，如果它周围八格中正好存在三个活细胞，该细胞死而复生。对于某个活细胞，如果它周围有二或三个活细胞，则该细胞状态保持不变。其他情况下细胞均保持死亡状态。编写该方法时，需要注意不能随着二层循环直接修改原数组。需要另外添加一个 tmp 数组用于暂存地图，在循环结束时将原地图替换为 tmp 数组。此外需要判断边界条件，否则会导致溢出。

四、 **uiFunction.py**

uiFunction 类是一个主要使用 pygame 库的类。在该类中定义了如图 4.1 中的方法。

```
4   import memSet
5
6   class uiFunction:
7       comp = []
8
9       def init(self, size, title):...
14
15      def mouseListening(self):...
22
23      def button(self, size, position, font, font_color, background,
24                 click_f_color, click_background, edge=5, width=0, title="Button"):...
29
30      def label(self, size, position, font, font_color, background=0, title="Label",
31                edge=5, width=0):...
35
36      def register_cp(self, way):...
38
39      def text_objects(self, text, font, color):...
42
43      def display(self, scr, com):...
77
78      def event_start(self, mp,screen):...
93
94      def event_pause(self, mp,screen):...
104
105     def event_reset(self, mp, screen):...
112
113     def event_random(self, mp,screen):...
125
126     def initChessBoard(self, startX,startY,screen):...
130
131     def drawCell(self, x, y, life, screen):...
136
137     def mouse(self, scr ,mp):...
167
168     def run(self, scr, background):...
```

图 4.1

该类用 pygame 创建了一个独立的 UI 窗口，该窗体的定义由 init()等方法决定，最终通过 run()方法来使用。

其中 init()是创建一个大小为 size，标题为 title 的窗体。在程序中我们创建的窗体大小为 1600x900。

mouseListening()是我们定义的一个鼠标监听器。该类赋予了程序获取鼠标方位和获取鼠标点击事件的功能，是 UI 控制界面的基础。

Button()方法的作用是对按钮进行封装。创建按钮实际上是由绘制一个大小为 size，起始位置为 position、背景颜色为 background 的矩形，以及字体颜色为 font 和 font_color 的文字框。最后的返回值为一个元组，其中的信息为以上数据，并将数据存放在开头创建的 comp。

Label()方法与 button()方法类似，是创建一个文本框，格式和内容与 button 相似，而后将传入信息封装在一个元组类型中，存放在 comp 中。

Display()方法是用于展示窗口。该方法调用在 init 中创建的窗口 scr，并将其展示。在方法内部，存在一个 for 循环，循环中通过判断 comp 中项目是按钮还是标签，并通过绘制长方形和书写文字的方式绘制按钮和标签。

而后定义了四个按钮，分别是“开始”、“暂停”、“重设”、“随机化”。开始指的是按照当前的地图数组开始无限运行。暂停指的是打断运行循环，将画面暂停。重设是将画面清空，但仍保留地图信息。随机化是重设地图，并为地图随机赋值。

在四个按钮之后定义了两个辅助方法分别为 initChessBoard()和 drawCell()这两个方法可用于辅助画图，从而让其它方法调用更加简单。

Mouse()方法的创建是用于获取鼠标的每一项事件，包括上下左右移动和点击(如图 4.2)。这个方法的创建让按钮可以被使用。具体原理是，如果鼠标点击的位置正好处于按钮图片被创建的位置，则视作点击按钮(例如图中控制台输出的 MOUSEBUTTONDOWN)。

```
[MOUSEMOTTON]: (87, 606) (0, -1) (0, 0, 0)
[MOUSEMOTTON]: (88, 605) (1, -1) (0, 0, 0)
[MOUSEMOTTON]: (88, 603) (0, -2) (0, 0, 0)
[MOUSEMOTTON]: (89, 603) (1, 0) (0, 0, 0)
[MOUSEMOTTON]: (89, 602) (0, -1) (0, 0, 0)
[MOUSEMOTTON]: (89, 601) (0, -1) (0, 0, 0)
[MOUSEMOTTON]: (90, 601) (1, 0) (0, 0, 0)
[MOUSEBUTTONDOWN]: (90, 601) 1
now reset
[MOUSEBUTTONUP]: (90, 601) 1
```

图 4.2

最后有一个 run()方法，会将先前绘制的窗口中展现出来。方法中有一个 while 循环，用于让窗口可以保持存在。首先调用 mouse 类，而后将 display 中绘制的按钮和对话框加入窗口。最后添加鼠标监视器，并刷新页面。重复以上操作就能让窗体稳定存在。

五、 **__init__.py**

最终添加了一个__init__文件，用于充当主类(如图 4.3)。主函数中先将界面中的按钮进行添加，最后再运行上文中的 run()方法，即可完成代码的运行。

```
23  if __name__ == '__main__':
24      ui.register_cp(la1)
25      ui.register_cp(bu1)
26      ui.register_cp(bu2)
27      ui.register_cp(bu3)
28      ui.register_cp(bu4)
29      ui.run(scr, (0, 0, 0))
```

图 4.3

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）

1）.初始界面

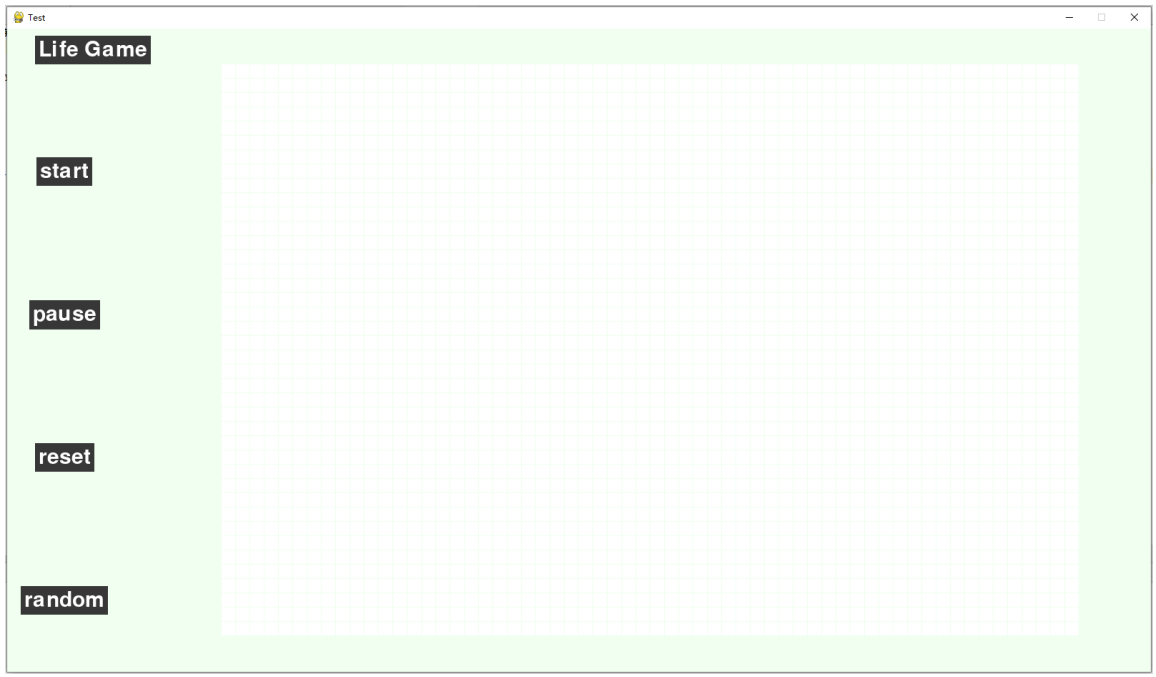


图 1

2）.开始

为了方便截图，我使用了暂停功能，图二图三是两个连续的图案，图案变更时间为 500ms。

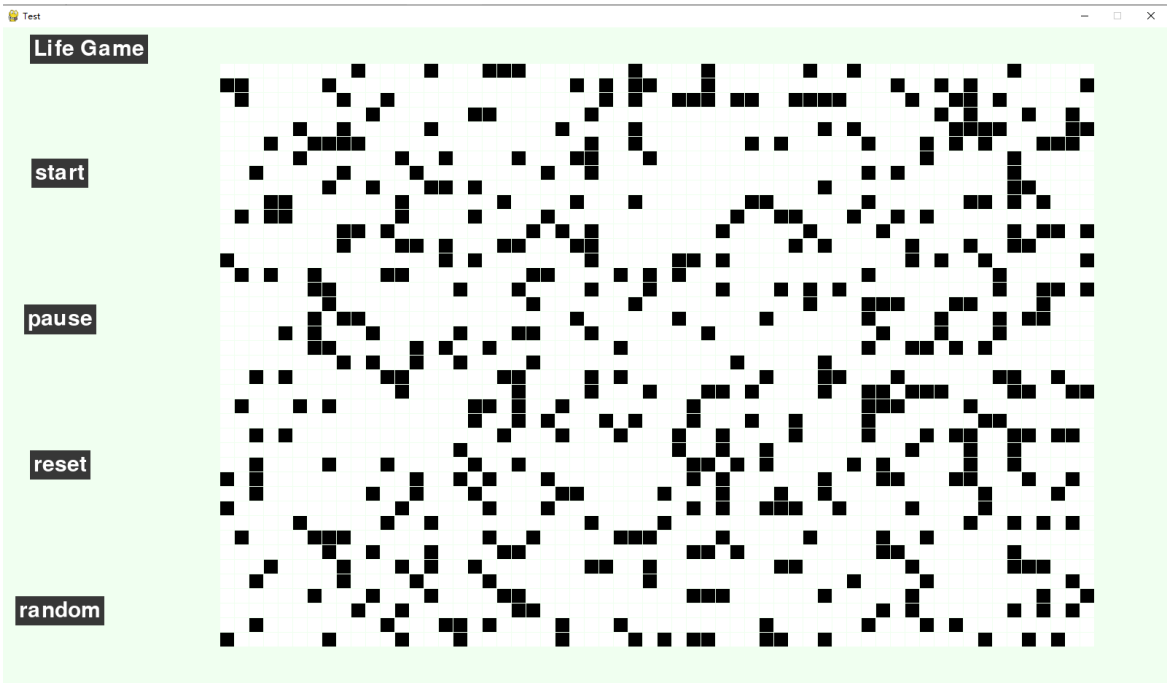


图 2

图 3

3）.暂停

为证明实现暂停功能，暂停后我截取了两个时间点，证明图案确实不会改变。

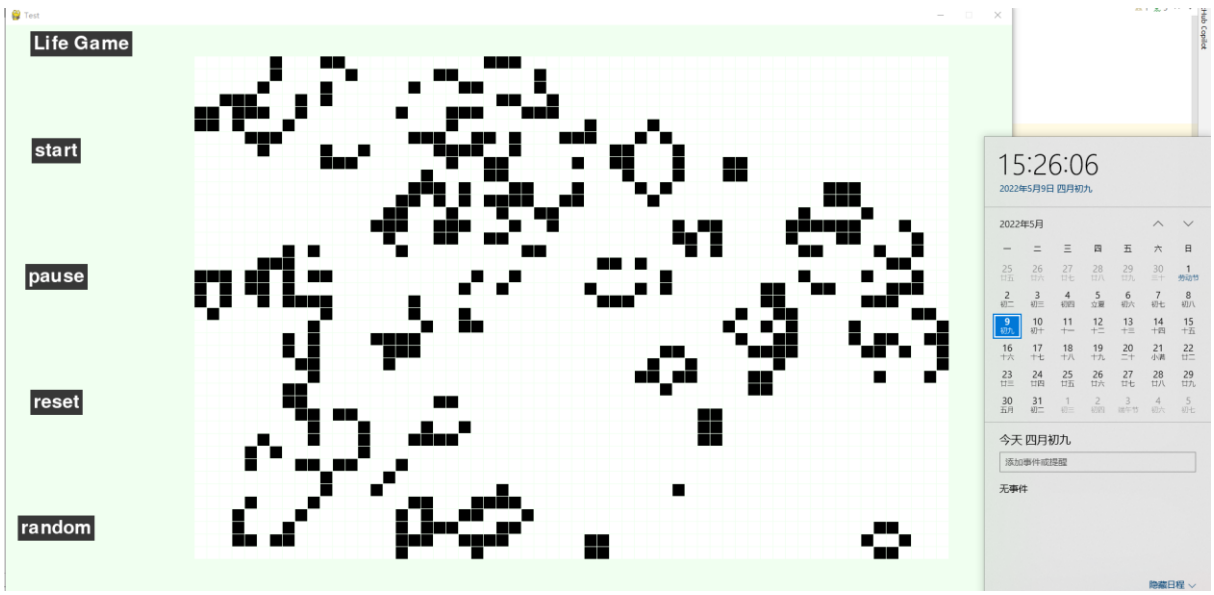


图 4

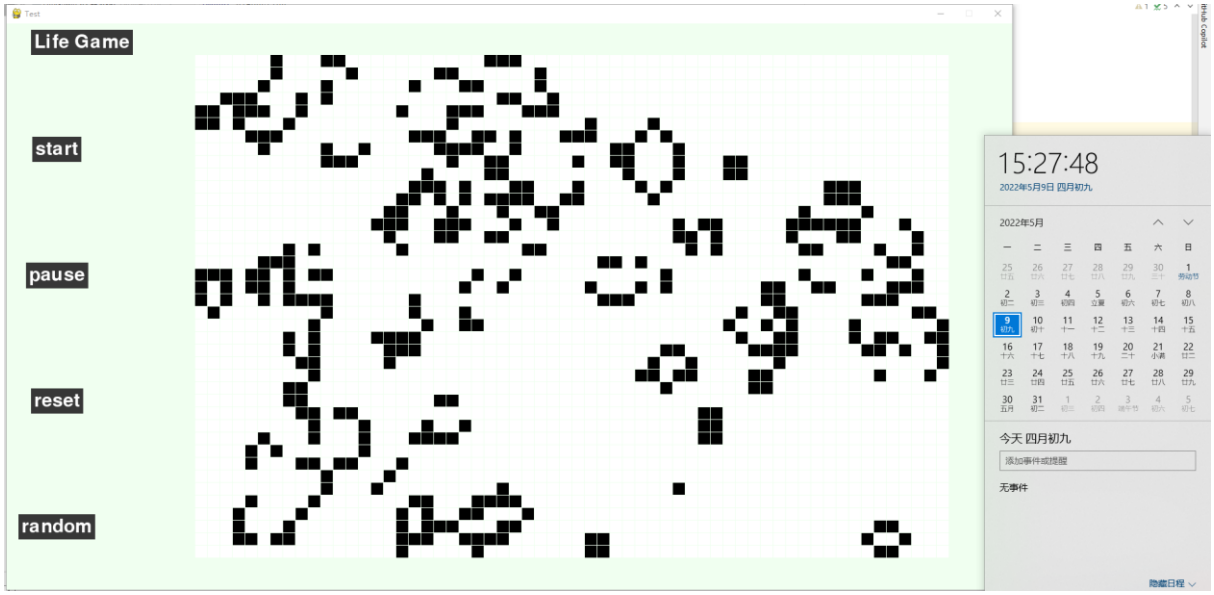


图 5

4）.重设

重设仅仅是将目前画面设置为空，再次点击开始后会将原图案保留并打印下来。

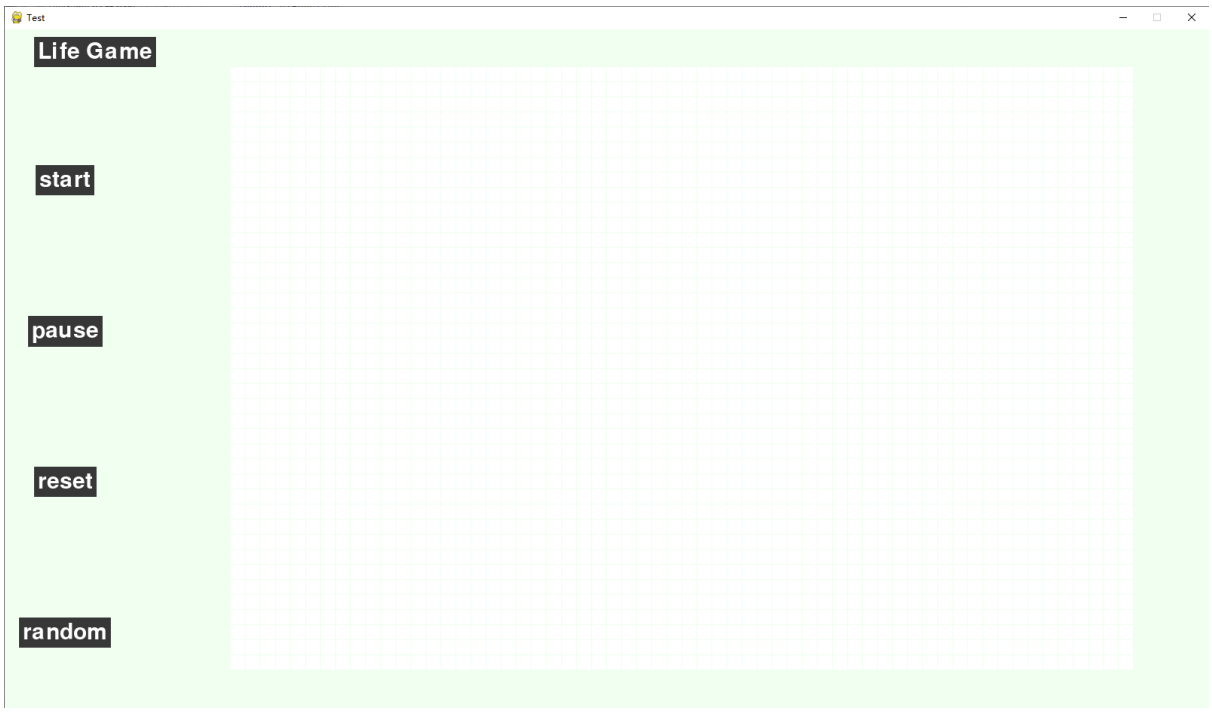


图 6

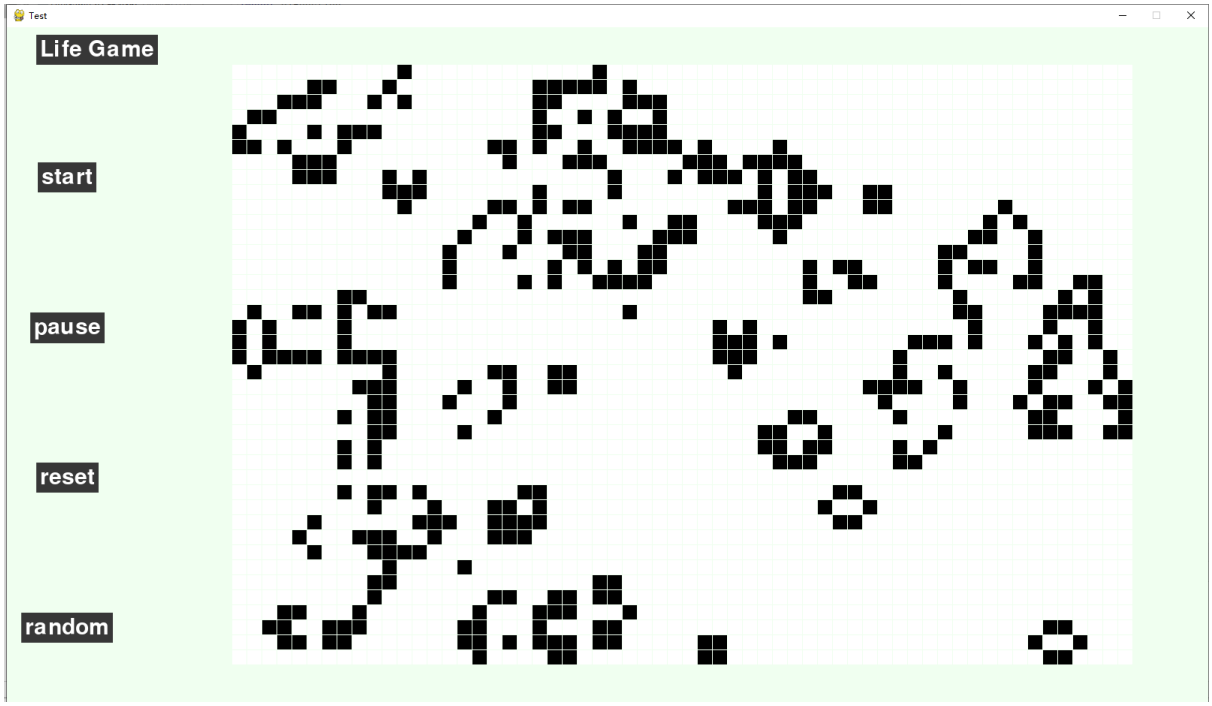


图 7

5）.随机化

点击随机化后，将会再次使用初始化方法，从而随机展现一组图案。在任何一张图中均可使用开始功能，并从当前图案进行演化。（图 8、图 9 为点击随机化后出现的图案，图 10 为图 9 的演化图案）。

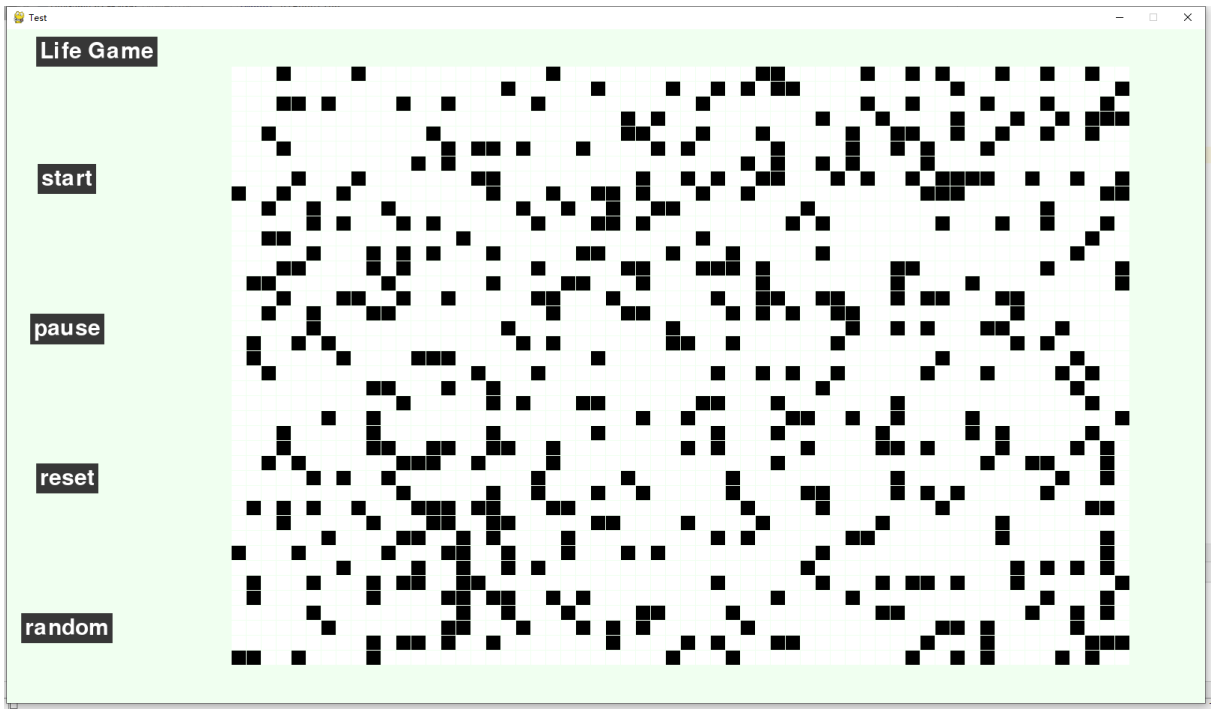


图 8

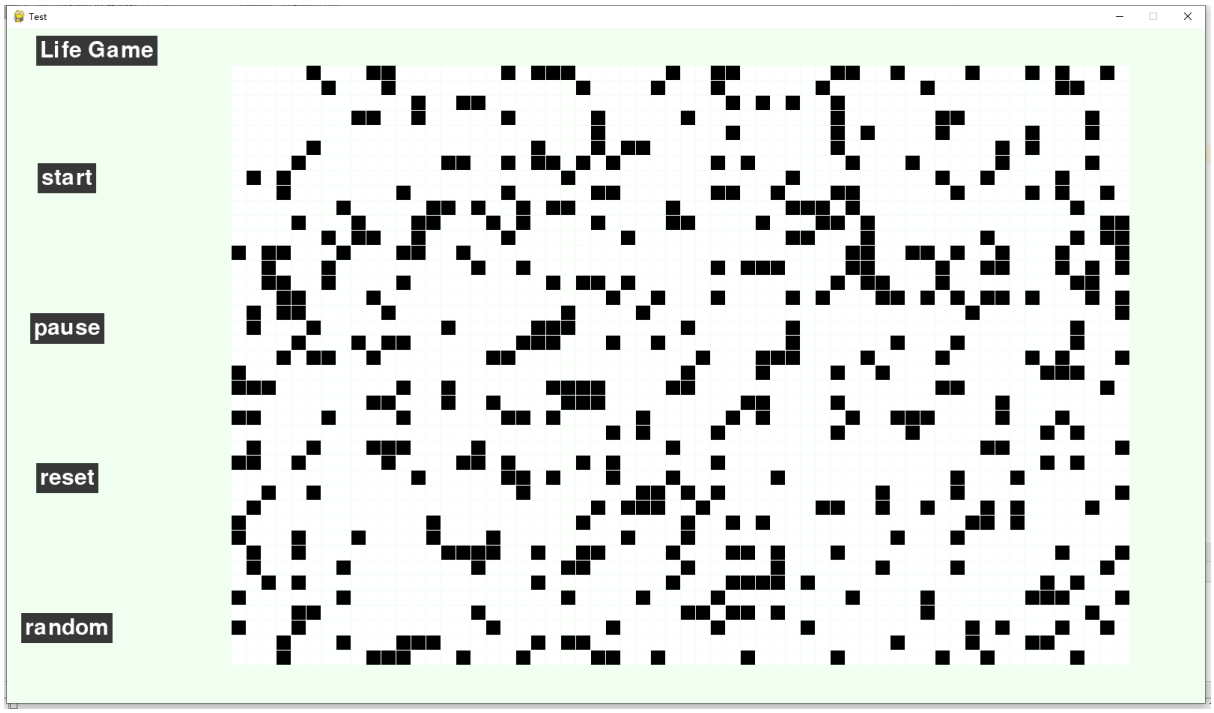


图 9

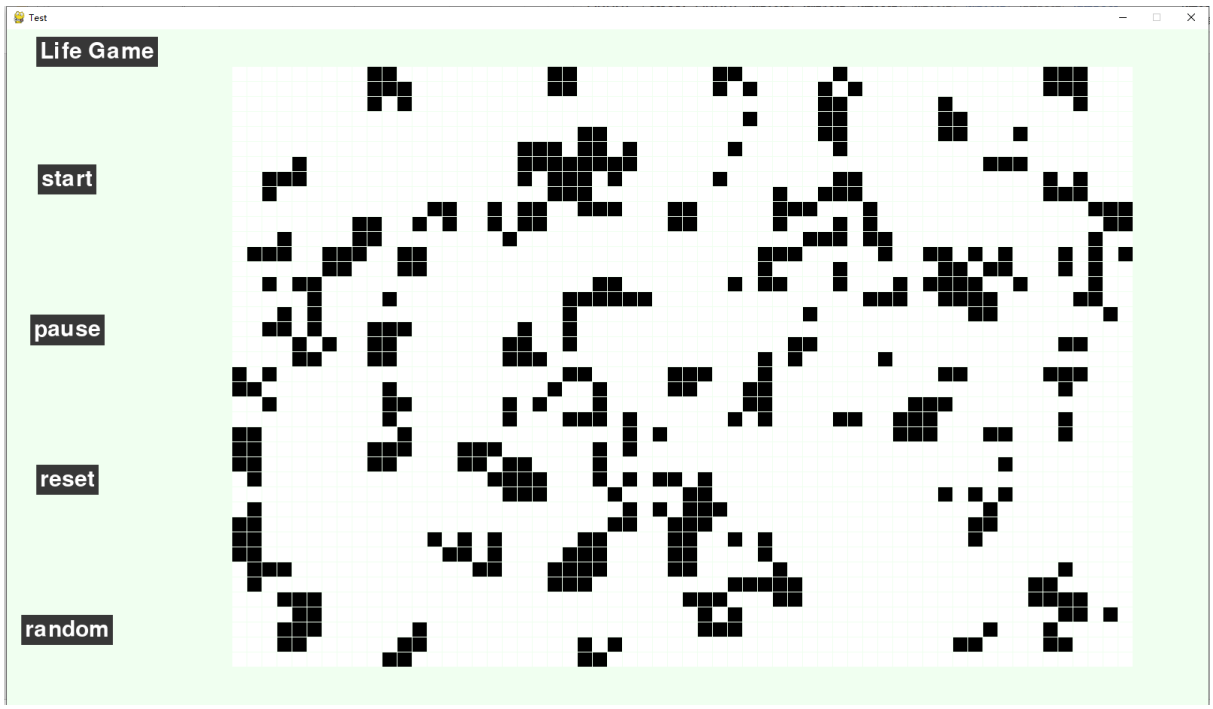


图 10

二、实验小结及体会

在这次实验中，我们使用 python 开发了具有基础功能和图形界面的生命游戏。在开发过程中，遇到了很多困难，在解决困难的过程中也对 python 的知识进行了一定程度的复习。况且在结对编程的过程中，可以通过讨论的方式让复杂问题简单化，特别是在查 bug 环节。在将项目面向对象化的过程中，我经常需要对比方法的输入与输出，并需要对项目的结构反复修改。在这个过程中会出现

许多运行异常，这些异常仅凭我一个人很难查出来。而在队友的帮助下可以相对轻松的查出问题所在。

这次实验的过程中也有很多不足的地方。我和索键对 python 都不算很熟悉，特别是其中的 pygame 库。因此在编写过程中，查询了很多语法相关的知识，也做了一些重复工作。例如 pygame 中的按钮，我们的实现方式相对比较原始，而且也不方便修改，为后期改为面向对象编程造成麻烦。但是在完成工程后，又发现了一个更简单的写法，我们也不得不重新修改。

总之，这次实验还是很成功的。首先，我对面向对象编程有了更深的认识，明白了应该如何对项目进行分层、分块编写。其次，我了解到结对编程的优越性，两个人可以发挥出 1+1>2 的实力，在编程过程中可以相互帮助，各司其职。在对软件开发效率拥有越来越高要求的今天，这样的方法能够使得开发能够更加明确，且能够加强开发的速度，达到事半功倍的效果。

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	