生命游戏

设计使用说明书

1. **引言**

**1.1开发背景**

康威生命游戏，又称康威生命棋，是英国数学家约翰·何顿·康威在1970年发明的细胞自动机。 它最初于1970年10月在《科学美国人》杂志上马丁·葛登能的“数学游戏”专栏出现。在游戏的进行中，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。一些形状已经锁定，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。对于生成的形状和秩序我们称作 pattern。或者在这里，我们也把它称作 creature。“生命游戏”并不是通常意义上的游戏，它没有游戏玩家之间的竞争，也谈不上输赢，甚至可以说游戏的一开始就注定了结果。

**1.2开发目的和意义**

本游戏是小组共同开发的课程设计项目，实现了基础的康威生命游戏规则，能够模拟生命繁殖演化的基本过程，实现了必要的图形界面。

开发生命游戏，让“仿真生物”生存于计算机上，在计算机上生存、死亡，从而模拟生命的演化，通过计算机的模拟，了解生命在一定规则下，开始条件对最终结果的影响，突发事件对最终结果的影响。

1. **需求分析**

**2.1设计内容和要求**

制作用户图形界面，使得游戏在运行时，用户能在图形界面上进行操作和直观的看到演化过程与结果。

按钮，开始、暂停、重置，用来控制繁衍进程。

用户界面的要求：一个N\*N的二维格子界面和对应开始、暂停、重置的按钮，每一个格子代表一个生命，亮为生、暗为死，每一次格子的生与死都显示在屏幕上。

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。用代码实现生命游戏中的规则，通过算法控制，计算格子在每一刻的生死状态。

使用鼠标添加或删去细胞。

添加游戏说明，方便用户使用。

**2.2技术上的可行性分析**

首先进行具体的需求分析：

**游戏功能：**

在一个二维矩形世界中，

每个方格里居住着一个活着的或死了的细胞，

每个细胞有两种状态-存活或死亡，

每个细胞与以自身为中心的周围八格细胞产生互动。

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，模拟生命繁衍。

可以通过按钮控制繁衍进程，使用鼠标添加或删去细胞。

**游戏规则：**

人口过少：当周围低于2个（不包含2个）存活细胞时， 本单元活细胞死亡。

稳定：当周围有2个或3个存活细胞时， 本单元细胞保持原样。

人口过剩：当周围有3个以上的存活细胞时，本单元活细胞死亡。

繁殖：当周围有3个存活细胞时，本单元细胞存活/活化。

**游戏界面：**

1.背景世界地图为黑色，画有灰白的网格线，作为各细胞间的分界

2.单个活细胞为矩形，红色，死亡细胞为黑色，构成背景

3.地图右侧放置按钮，开始、暂停、重置，用来控制繁衍进程

4.添加游戏说明，方便用户使用

经查阅资料并进行分析，利用python及其标准化的库，可以实现我们的设计要求

**2.3开发环境与工具使用**

操作系统：Windows

开发工具：PyCharm Community Editon 2016.1.3、Java Pydev

开发语言：Python

编码方式：UTF-8

团队管理工具：leangoo

版本控制工具：Git

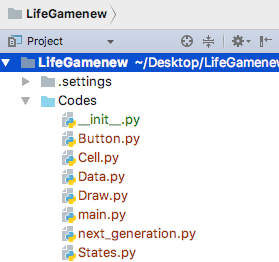
自动单元测试框架：pyunit

性能分析：profile

代码检查：pylint

1. **总体设计**

**3.1总体结构**



**3.2各功能模块描述**

a.Button :定义按钮的图像以及具体的位置

b.Cell：定义细胞类，包括细胞的大小和颜色，获取细胞位置在画布上绘制细胞

c.Data：画布，以及程序相关的具体的数据

d.Draw：将画布擦除后，画上按钮，格子以及具体的网格线，然后遍历网格数组中的细胞状态

e.next\_generation：定义细胞更新的规则：当一个细胞周围有两个或三个细胞时细胞状态为存活，当周围细胞过多或者过少时定义为细胞死亡

f.States：将整个游戏世界设定为二维数组，每个网格的状态值分为0，1两种，定义按钮功能，含开始运行、暂停、重置三个方法

开始运行：获取当前鼠标的状态和位置随时更新画布，并且根据细胞规则随时更新细胞

暂停：根据鼠标状态随时更新画布

重置：初始化整个画布，数组用0填充

g.main：程序的开始入口，设置界面的大小，定义状态机的三种状态：运行状态，暂停状态，以及重置，然后进入游戏的主循环

**3.3主要模块设计**

状态机三种状态的设置：开始，暂停，以及重置，主要在States模块中实现

States：将整个游戏世界设定为二维数组，每个网格的状态值分为0，1两种，定义按钮功能，含开始运行、暂停、重置三个方法

开始运行：获取当前鼠标的状态和位置随时更新画布，并且根据细胞规则随时更新细胞

暂停：根据鼠标状态随时更新画布

重置：初始化整个画布，数组用0填充

**3.4核心代码设计**

next\_generation：细胞规则的制定是核心代码的一部分，具体设计

遍历二维数组，更新接下来各个细胞的状态

**def** next\_generation():  
 nbrs\_count = sum(np.roll(np.roll(pygame.world, i, 0), j, 1)  
 **for** i **in** (-1, 0, 1) **for** j **in** (-1, 0, 1)  
 **if** (i != 0 **or** j != 0))  
  
 pygame.world = (nbrs\_count == 3) | ((pygame.world == 1) & (nbrs\_count == 2)).astype(**'int'**)

1. **测试分析**

**4.1游戏开始测试**

测试点击开始按钮游戏界面立即作出响应，并且可以按照细胞规则进行正确的更新

**4.2游戏重置测试**

点击重置按钮，程序立即作出响应，并且画布被擦除

**4.3使用工具进行分析测试**

自动单元测试框架：pyunit

性能分析：profile

代码检查：pylint

分析测试细节详情请见对应的工具使用文档

1. **具体的使用说明**

打开程序，出现游戏界面。

在界面上选择点击相应格子，左键添加细胞，右键擦除细胞。

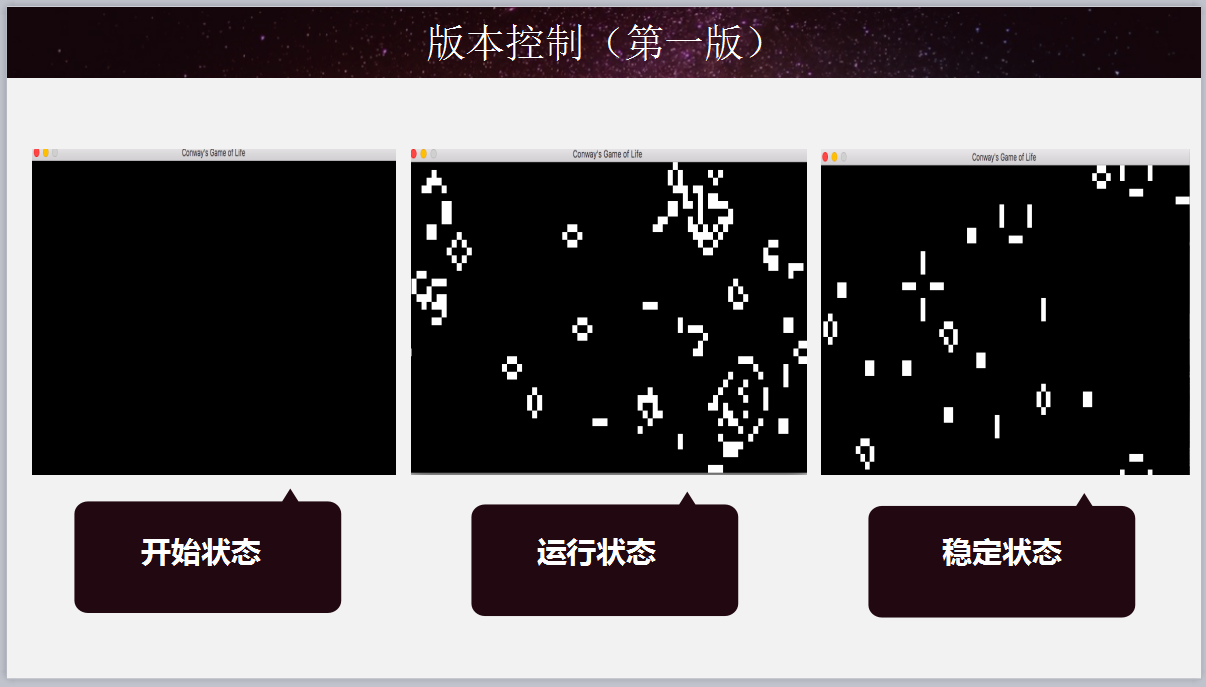
点击开始，程序自动运行，最终在界面上显示结果，或为一有各个格子组成的特殊图形，或一直杂乱的进行演化。

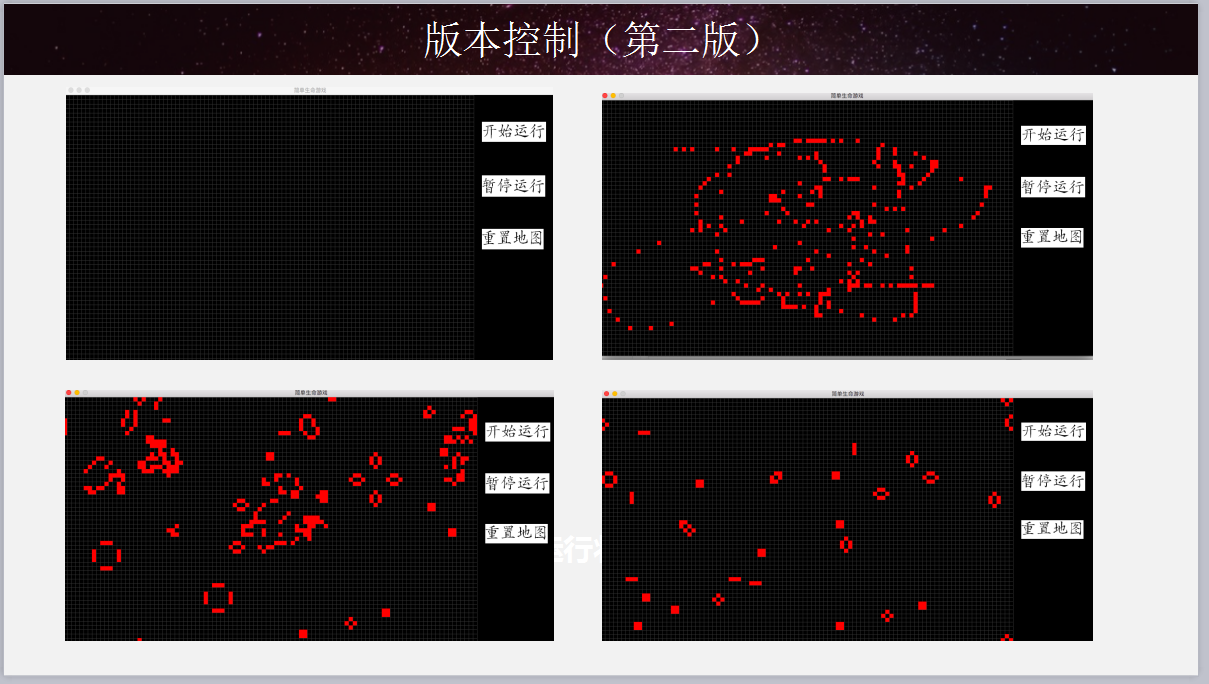
也可在任意时刻点击暂停按钮，查看到这一时刻生命游戏在一定规则下生成的结果。

玩家可以随时在画布上添加或者擦除细胞。

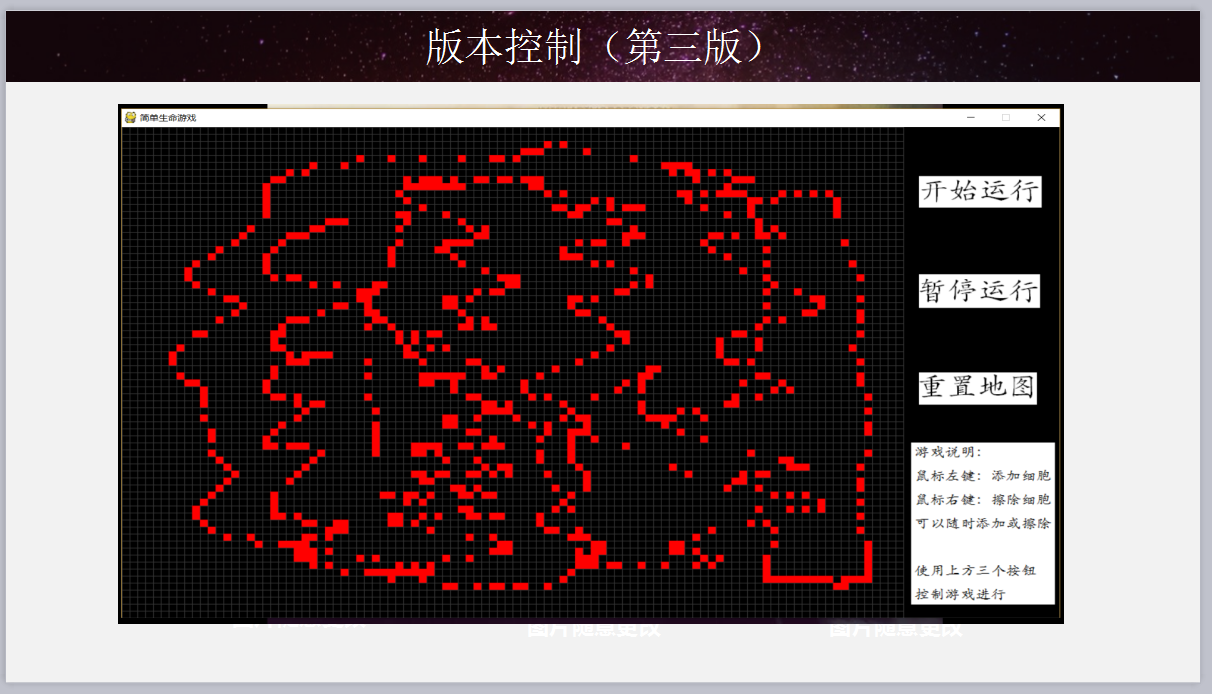
点击重置按钮可以清空画布，重新来过。

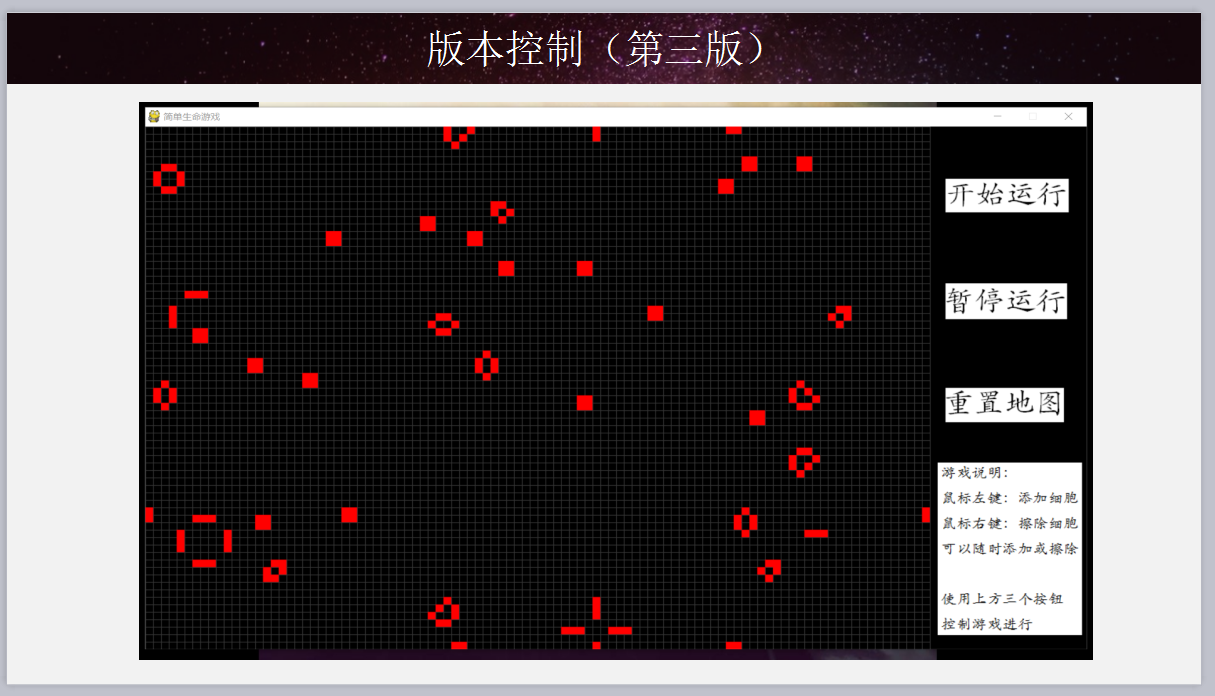
**运行截图演示：**

****

****

****

****

****