**注意：**

**本项目使用了第三方库easyx 20220901版进行绘制图像。**

**提交的压缩包为vs2022的工程文件，main.cpp，存档，背景音乐在\CELL\CELL里。**

**如果运行时出现两个窗口，是windows 11 2022年末-2023年初的一次更新导致的bug，会影响键盘交互导致游戏功能无法实现！在任务栏搜索开发者设置，将终端从“让Windows决定”改为“Windows控制台主机”即可隐藏控制台，只显示第三方库easyx调用的窗口。别忘了运行完改回去😊。**

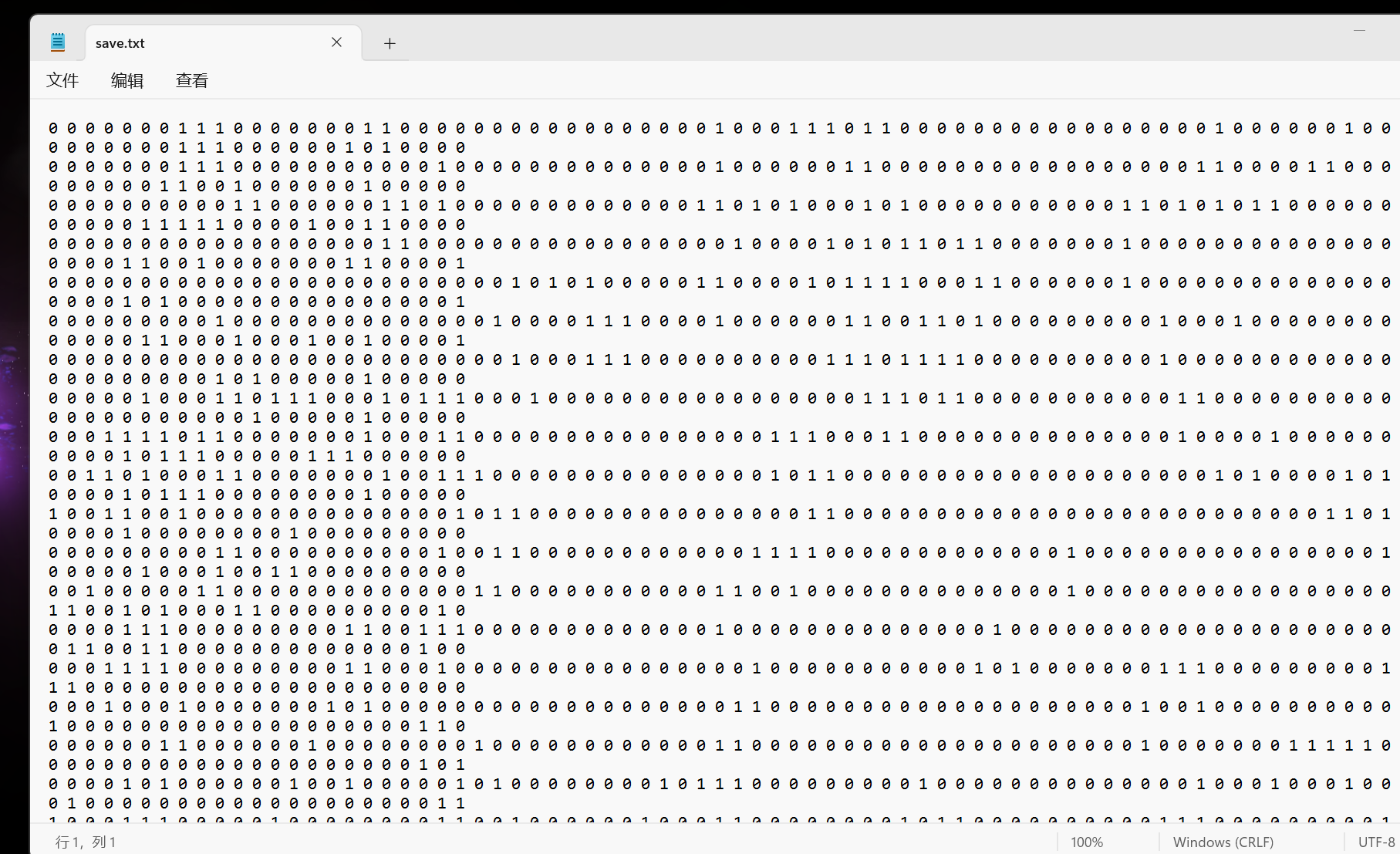
本项目实现了一个细胞自动机的游戏。游戏界面由一个矩形网格组成，每个网格单元表示一个细胞，细胞可以处于存活或死亡状态。游戏通过按键和鼠标操作来控制细胞的状态和游戏的进程。

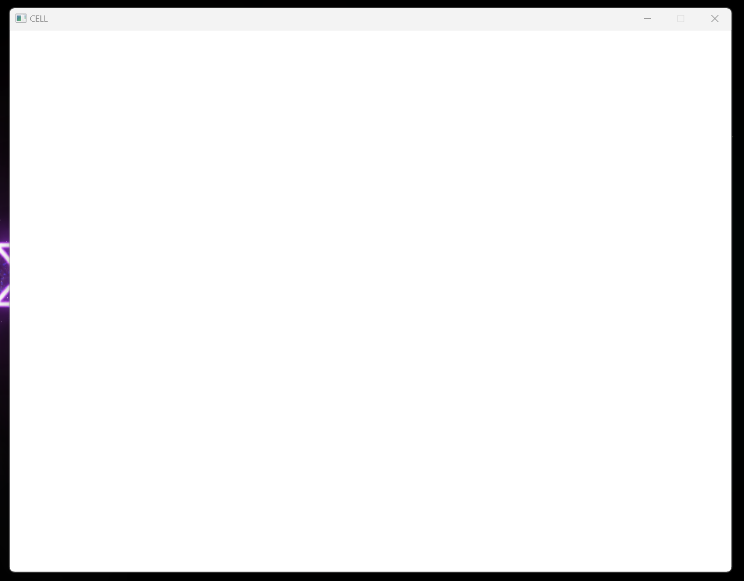
使用了new方法创建了一个三维数组(额外维度用来存储棋盘)来存储棋盘上的细胞状态，并且规避了边界屏障，使边界相互连通，模拟“无限大”的地图。

**PS：pdf中所有功能均已实现，并额外增加了“随机地图种子”“清除所有活细胞”和“播放背景音乐”功能**

**以下是程序运行演示:**

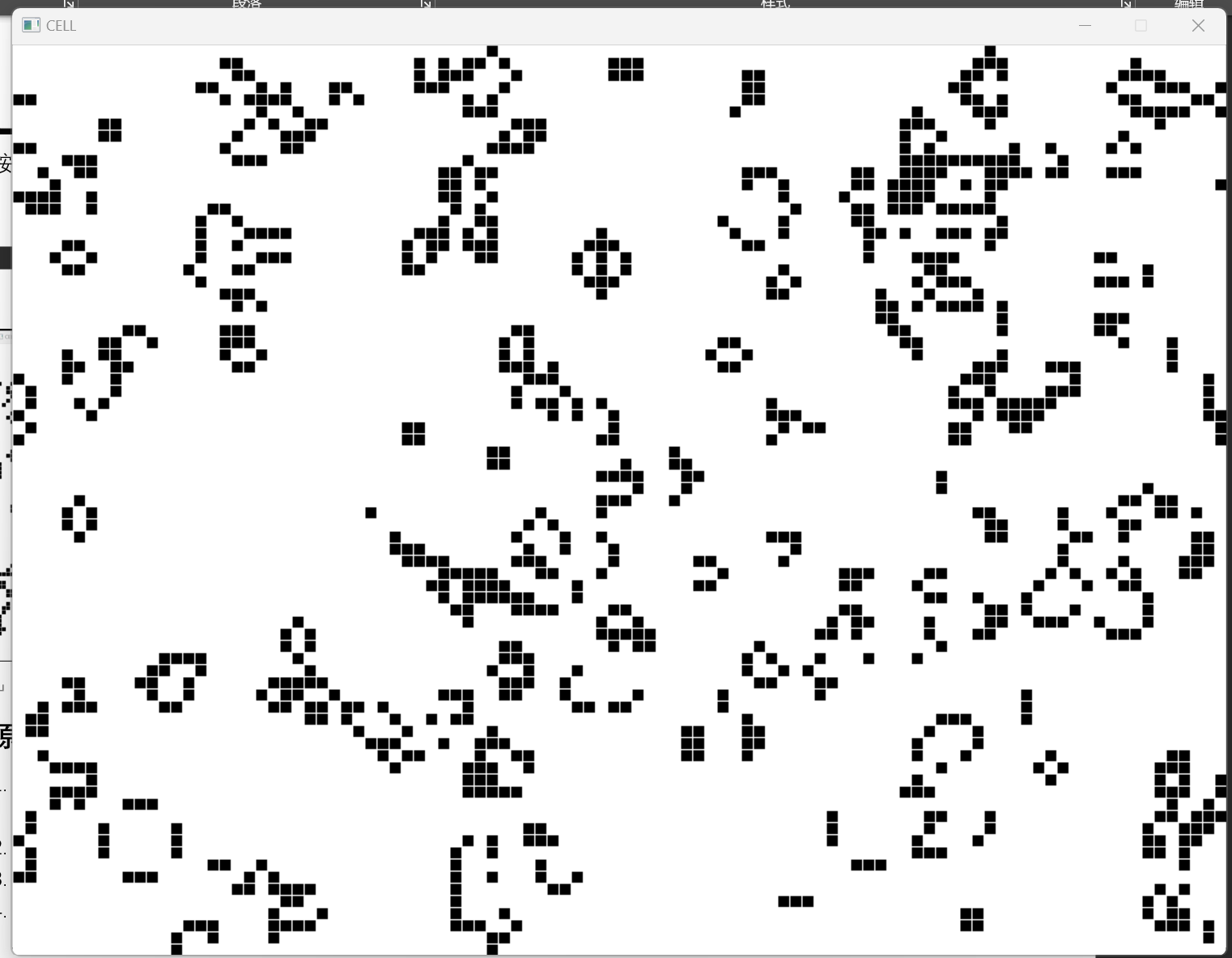
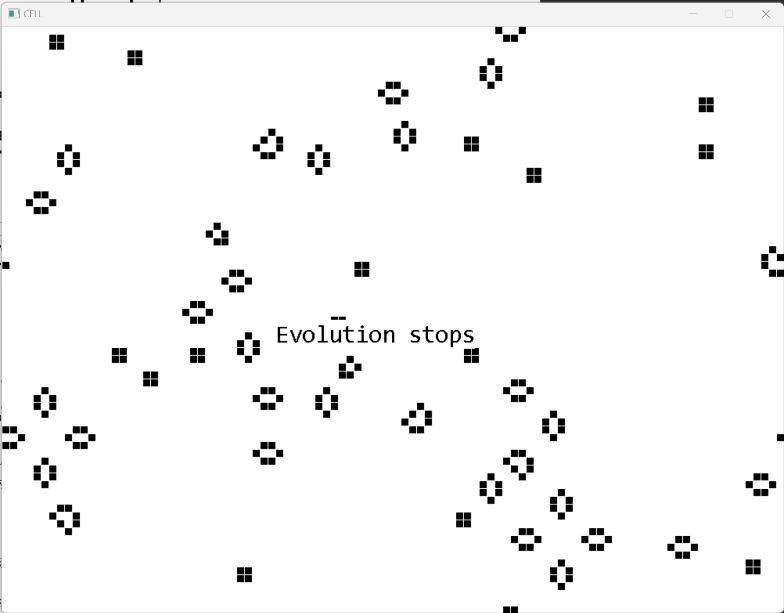
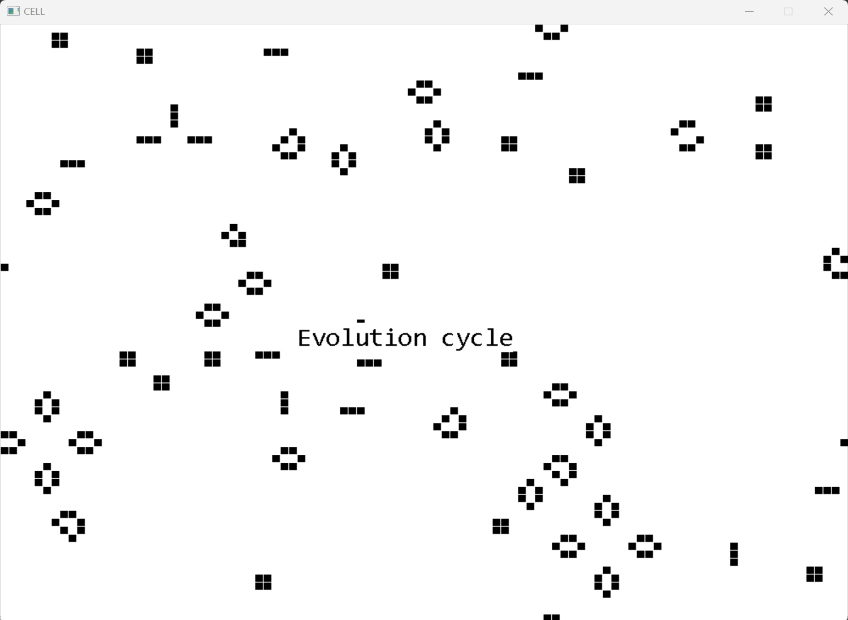
开始运行时会以时间为种子生成随机棋盘。先按空格暂停一下。

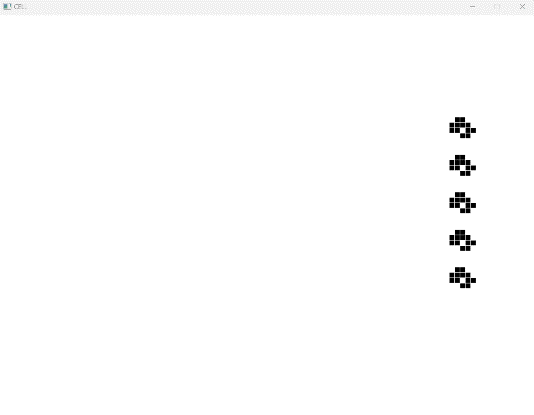
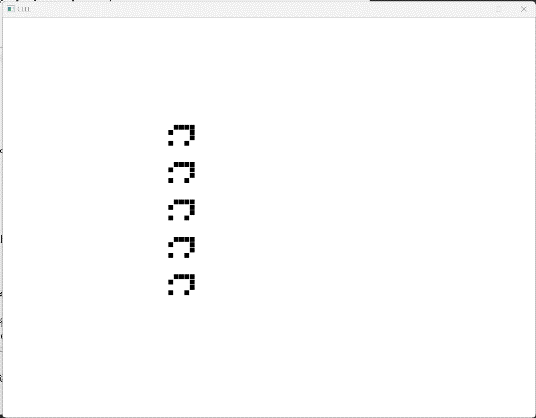
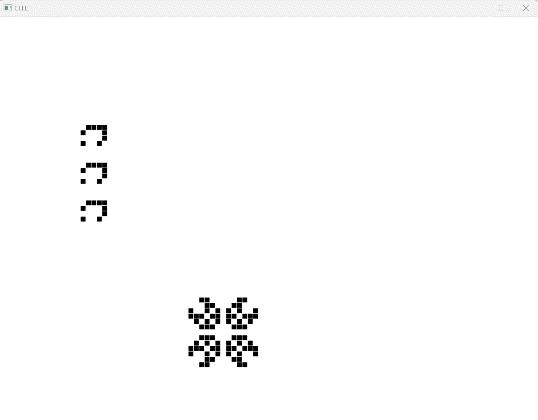
save.txt为按下“C”键保存生成的文档，以0 1记录细胞状态



按“P”键清空棋盘，再按下“V”键可成功读取save.txt文件。

按下“N”键可手动单次演变(图为演变十次结果)

按下“M”键可手动返回上次的演变

继续演变，当进入循环或停止状态时，会自动检测并在窗口中间输出“Evolution cycle”或“Evolution stops”

再次清屏，用鼠标左键点击或按住拖动，鼠标右键点击或按住拖动，设置细胞死活，绘制小飞机联队和脉冲星😊

(提交的文件里自带了这个存档，对这几个图像感兴趣的话可以按“V”键读取一下看看，(如果没有被覆盖的话))

最后到了背景音乐环节，程序运行时会自动读取根目录的该文件并播放

**代码中的按键汇总如下：**

- 空格键：暂停或继续游戏，切换游戏的暂停状态。

- "C" 键：保存当前游戏板面上所有细胞的状态到根目录，以save.txt形式保存。

- "V" 键：从文件中加载细胞的状态，更新游戏板面。

- "P" 键：清空游戏板面，将所有细胞状态设置为死亡。

- "N" 键：进化一代，手动进行游戏进化，会生成lastevolve.txt 临时文件存储数据。

- "M" 键：恢复到上一代，加载最近存储的游戏板面状态（最多只能返回一代）。

- 鼠标左键点击或按住拖动：设置细胞状态为存活。

- 鼠标右键点击或按住拖动：设置细胞状态为死亡。

**代码中使用了以下额外头文件：**

- graphics.h：用于图形绘制

- conio.h：用于控制台输入

- windows.h：用于调用Windows API

- vector：使用向量容器

- fstream：文件读写

- mmsystem.h：播放游戏背景音乐

**代码中的主要成员函数和其功能如下：**

- Automaton::Cell: GetNeighborCount:(): 获取指定细胞周围活细胞的数目

- Automaton::Cell: Display:(): 绘图函数

- Automaton::Automaton(): 构造函数，初始化游戏板面并分配内存。

- Automaton::~Automaton(): 析构函数，释放内存。

- Automaton::Init(): 初始化游戏板面，随机生成细胞的存活状态。

- Automaton::run(): 更新游戏板面，根据细胞的存活状态和规则进行更新。

- Automaton::Display(): 绘制游戏板面，用图形库在图形窗口中绘制网格，并根据细胞的存活状态填充相应的颜色。

- Automaton::TogglePause(): 暂停或继续游戏，切换游戏的暂停状态。

- Automaton::SetCell(): 设置指定位置细胞的状态，可以设置为存活或死亡。

- Automaton::getAllCellStates(): 获取当前游戏板面上所有细胞的状态，以二维向量的形式返回。

- Automaton::saveToFile(): 将当前游戏板面上所有细胞的状态保存到文件中。

- Automaton::loadFromFile(): 从文件中读取细胞的状态，更新游戏板面。

- Automaton::clearBoard(): 清空游戏板面，将所有细胞状态设置为死亡。

- Automaton::storeRecentStates(): 存储最近的几个游戏板面状态，用于检测游戏是否进入停滞或周期状态。

- Automaton::isRecentStatesEqual(): 比较两个存储的游戏板面状态是否相等。

- Automaton::handleMouseActions(): 处理鼠标操作，根据鼠标左键和右键的点击或移动设置细胞的状态。

- Automaton::Evolutionstops(): 在图形窗口中显示"Evolution stops"的消息。

- Automaton::Evolutioncycle(): 在图形窗口中显示"Evolution cycle"的消息。