

《程序设计实践》实验报告

数独

学号： 201611130126

姓名： 袁宇昊

学院： 信息科学学院

专业： 计算机科学

教师： 尹乾

日期： 2018.10.12

1. 设计目的

数独游戏是在9\*9的棋盘上，填入1-9共九个数字，使1-9每个数字在每一行、每一列和每一宫中都只出现一次的一种游戏。

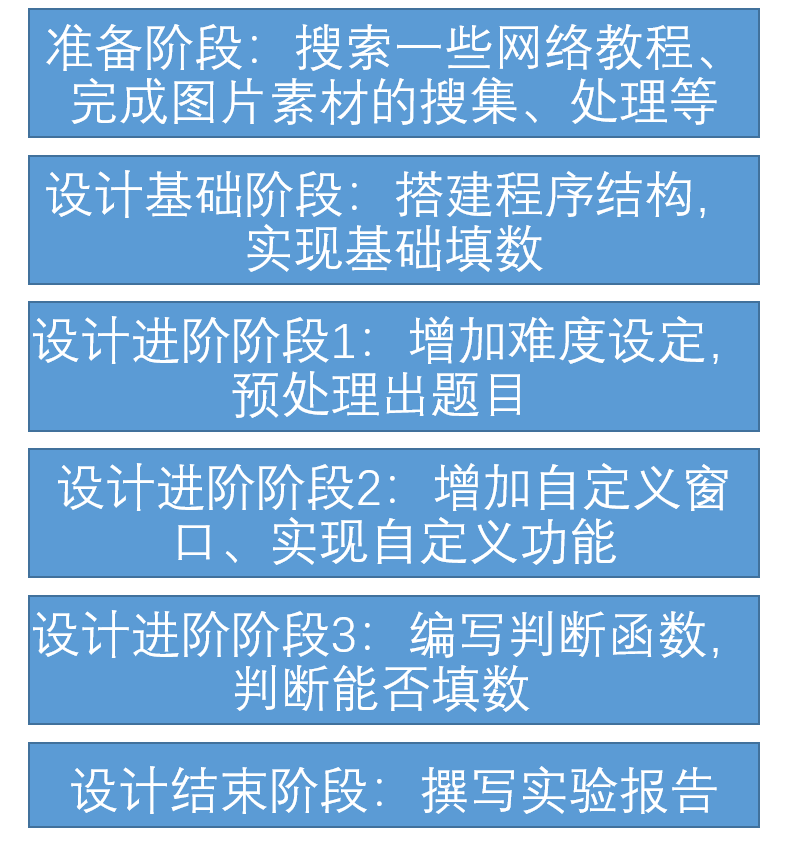
二、总体设计**（程序设计组成框图、流程图）**

整体架构是用python里的pygame库实现的。

版本是python的3.6版本和pygame的1.9.4版本。

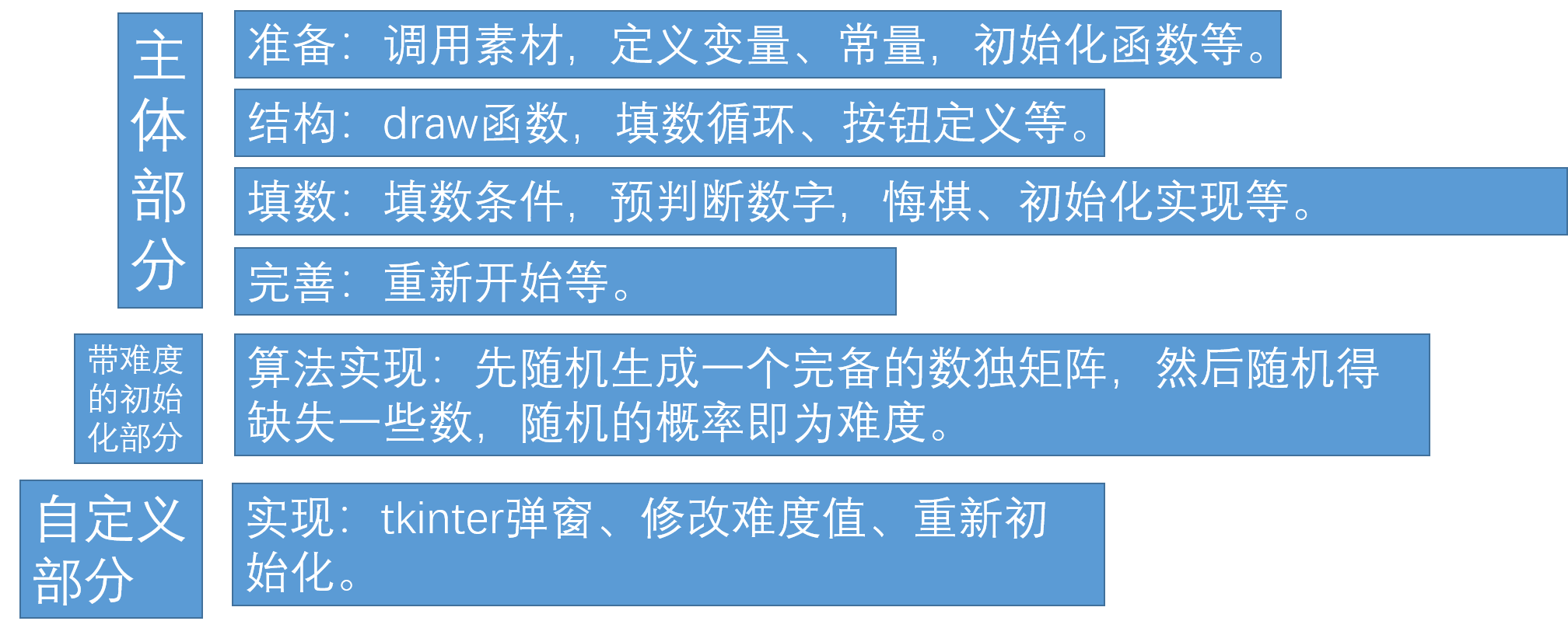
利用pygame库函数加载背景图片、棋子素材等图像材料。利用数组记录棋谱以实现悔棋功能。

整体的程序设计流程图如下：



程序总共就分为了3大部分：主体部分、带难度的初始化部分、自定义功能部分。其中主体部分是由2个循环构成，一个是游戏界面的循环（方便自定义函数），一个是填数的循环。

程序设计组成框图如下：



**三、详细设计（模块功能说明，如函数功能、入口及出口参数说明，函数调用关系描述等）**

**结构部分：**

**主要变量：格子大小等长度，图片句柄等。**

Init ()函数：pygame库里的初始化函数。

mydraw（tx,ty,player）函数：主要界面的绘制、更新。tx,ty即当前填数的位置，player为当前填入的数字。函数逻辑大意为如果当前的格点（tx,ty）上没有数字，那么绘制一个预落子，如果当前个点上有数字，那么跳过。对于其他的格点就按照一般的处理。最后再高亮出所有当前选择的数字已经处于的位置，方便玩家观察，这一点借鉴了手机上的一款游戏。

**填数部分：**

**主要变量：now当前的选定要填的数，范围是0~9。mymatrix[][]棋盘矩阵，0为空格、>=1为数字。premymatrix[]步骤数组，记录填数步骤。**

填数功能：直接写在循环里，大致实现为判断当前格点是否有数字，没有棋子的话就调用ok()函数来判断填入的数是否合法，更新相应棋盘矩阵为now，更新步骤数组。

悔棋功能：直接写在对弈循环里，大致实现为读取步骤数组的尾步，棋盘矩阵直接置0，并且更新棋谱矩阵。

ok（tx,ty,player）函数：判断当前填数是否合法。tx,ty即当前填数的位置，player为当前要填的数字。函数逻辑大意为检查这一行、一列、一宫里是否已经有与player重复的数字。

over()函数：判断当前棋盘是否已经结束。并加了一个完成彩蛋。

选择棋子功能：在棋盘的下方有一排供选择的数字，点击相应的数字就可以更改now值为想要填写的数字。

**难度部分：**

**主要变量：hard难度参数，这个参数的意思是以hard/10的概率去选完备数独中的数字进行缺失。**

initail()函数：重置棋盘矩阵和棋谱数组等。使用了random库生成随机数来随机生成数独的题目。主要在初始化的时候和重新开始的时候调用。

**自定义部分：**

**主要变量：hard难度参数。**

call\_set()函数：在reset按钮中调用的函数：主要是定义弹窗的界面，并且处理输入的数据并进行修改。现在支持对hard难度参数的修改。有针对默认值和最大最小值的处理。

**四、调试与测试**

这个程序的大体结构与我的五子棋程序相似，需要修改的是一些函数的细节和一个新函数的编写。

使用的是spider编辑器，使用了它自带的设置断点调试的功能，也用到了它的变量表查看变量，还用到了print()函数查看进行的程序步骤。

**主要遇到的问题：**

**1数组越界：**因为用到了mymatrix[][]矩阵数组和premymatrix[]数组来存放棋盘的情况，所以要防止数组的越界。调试的报错信息会给出出现错误的行数很有帮助。

具体遇到的情况如下：

实现draw函数时搞反了x、y轴导致越界，而且在后面的更新中没有一并更改导致了越界。

解决办法：一定要加上对数组下标的限制后再进行数组的访问。同时注意程序上下文的一致性。

**2 填数时的逻辑错误：**主要体现在对ok()函数的编写上，因为要实现判断的功能，所以要对列、行、宫都要进行检查，有的地方就造成遗漏或者重复，体现出来有一下几点：

弄反x轴和y轴，找错了宫等。

解决方法：主要是采用print()办法查出来出错的地方后再更改，如此往复。

**五、源程序清单和执行结果**

只有一个主程序main.py，同时有十七张图片的素材在resource文件夹里。

执行的结果可以参考下面的图片：



难度为8的棋盘



自定义窗口



通过自定义窗口自定义难度为4的棋盘