**消消乐实验报告**

1. **选题**
2. **游戏规则**
3. **方案设计**
4. **数据结构**
5. **函数设计**
6. **关卡设计**

**四、关键技术和难点**

**五、效果**

**六、总结收获与不足**

**宾果消消乐实验报告**

1. **选题**

宾果消消乐

1. **游戏规则（功能介绍）**

点击两个相邻的图标交换两者的位置，使得界面中有三个或三个以上的图标连成直线型、L型、T型、十字型的时候实现消除。消除部分上面的图标先掉下来填充消去的部分，剩下的部分由随机生成的图标补充。

1. **方案设计**
   * **数据结构：**map[7][7]二维数组
   * **函数：**

**各文件函数功能一览表：**

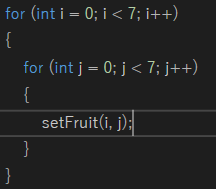
|  |  |  |  |
| --- | --- | --- | --- |
|  | **map.c** | **game\_operation.c** | **ui.c** |
| **功能** | 实现对二维数组map相关操作函数的封装 | 对游戏相关操作函数进行封装 | 游戏的总控制台 |
| **函数一** | **initmap()**:初始化二维数组，并且保证初始化的数组没有可以消除的 | **SetBoundary()**:设置边界坐标 | **Winmain()**:注册窗口类、创建窗口及接收消息循环 |
| **函数二** | **clear()**:用遍历数组的方式，消掉数组中所有可以消掉的值，置为0 | **GetBoundary()**:获取边界坐标（定义了Game\_COORD结构体，表示坐标） | **CallBack()**:处理接收的信息函数。主要处理刚创建窗口时，和接收鼠标信息时的操作 |
| **函数三** | **fill():**用遍历的方式（从后往前），将为零的值的上面的元素和下面的交换，空出来的值随机赋值， | **CreateGame()**:游戏初始化。先初始化数组，后设置游戏边界坐标 | **GamePaint()**:绘制窗口函数。主要包括绘制背景、字体、和水果 |
| **函数四** | **swap()**:交换两个点（x1，y1）和（x2,y2）的map[x1][y1]、map[x2][y2]值 |  | **ReSizeGameWnd()**:设置窗口大小的函数 |
| **函数五** | **setFruit()**:对map[i][j]随机赋值，在[1,5]之间 |  |  |

**各函数的具体实现：**

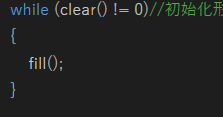
**map.c:**

1、void initmap()：采用遍历二维数组的方式，对二维数组依次赋值。赋值

调用setFruit()函数来实现，赋予[1,5]之间的值

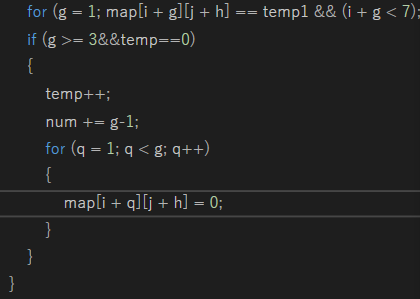


要保证新生成的图形没有可以消去的，则用clear()函数的返回值来判断，不断执行clear函数和fill函数，知道clear函数返回值为0时，生成的图形没有可以消掉的

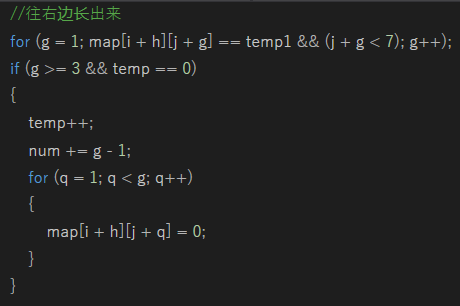
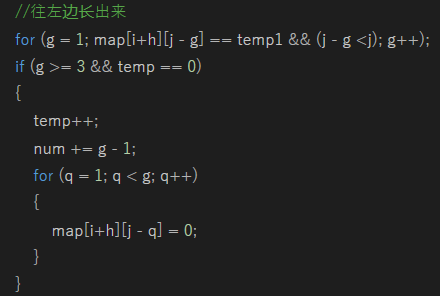
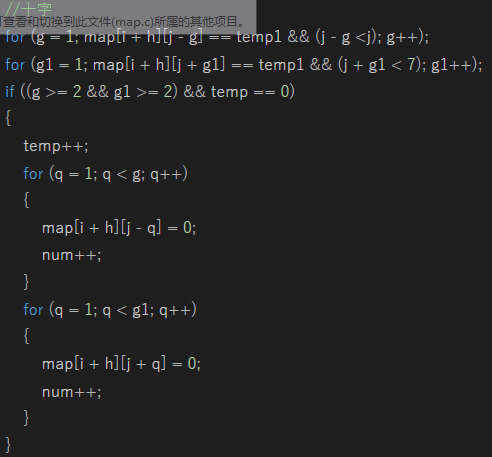


2、int clear(): 采用正向遍历二维数组的方式，对可消去的情况进行分类

之后发现，若对某一map[i][j]进行横向判断之后，就不用进行判断纵向判断。

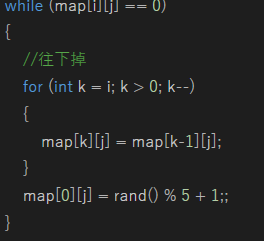
横向判断：对map[i][j]发现在水平方向有三个以上的值相同，则水平方向可消。在对每一个相同的值进行纵向判断，若纵向有超过三个相同的则可消成特效情况L

纵向判断：纵向的情况相对较多。若map[i][j]在纵向能有三个以上的值相同，则对除了第一个元素之外的每一个相同值看，向左或向右能否有三个以上的值相同构成L型特效，或者左右加上中间的能否有超过三个相同构成十字型特效



3、void fill():消消乐是从下往上消，则逆序遍历数组。若遇到为0的值

就是消去的，则把上面的值和下面的值交换，空出来的那一个随机赋值



**game\_operation.c:**

1. PGAME\_COORD GetBoundary()和void SetBoundary(int,int)两个函数是直接copy的老师代码。
2. void CreateGame():调用initmap()初始化数组；

调用SetBoundary()设置边界坐标

**ui.c:**

1. 程序入口WinMain()和ReSizeGameWnd()函数是直接copy的老师代码，对一些小数据稍作改动
2. GamePaint():画背景：贴背景位图

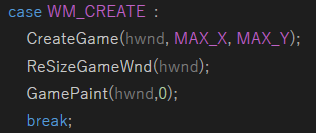
画水果：遍历数组，对不同的值贴不同的水果位图

画文字：分别画出移动步数和消灭个数的值。

都运用GDI相关函数

1. CallBack()：当接收到WM\_CREATE信息时，要完成初始化游戏，设置游戏窗

的过程



当接收到鼠标信息时，分别点击相邻两个图标，交换两者位置若能消去，则消除；若不能消除，则把两个位置交换为原来的样子。用两个全局变量point1、point2记录两次点击的位置，然后计算转换为map[i][j]的i，j。

* **关卡设置：**每一关卡有限制的交换次数和消除数量。在交换次数之内达到消除数量则进入下一关，一共有三关。

1. **关键技术和难点：**

**关键技术：**

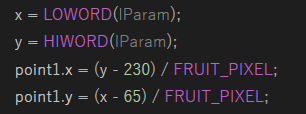
1、GDI的应用：这是之前没有接触过的新知识，是实现游戏界面化的关键

技术。一些GDI的函数在代码中起到了至关重要的作用

**2、**函数的封装：实现良好的函数封装对设计思路和算法的实现都有着至关重要的作用

难点：

1. 接收鼠标信息的处理过程：要保存两个鼠标点击的坐标，并且将其转化

为相应二维数组的i,j值。

1. 消去函数clear():因为二维数组的存储在计算机中是转化为类似一维数

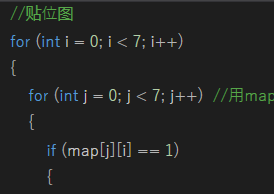
组的，所以在横向判断时，要加上j+k<7,j-g<j





1. 贴位图时，因为计算机是从左往右再从上往下贴图，与平时有一定的差

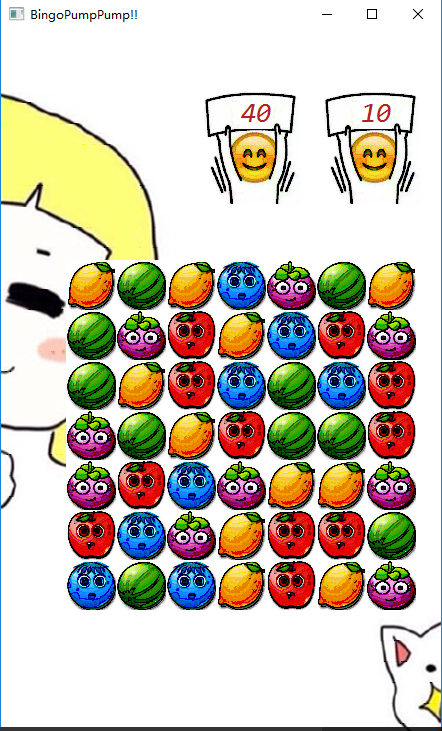
所以改成map[j][i]



**五、结果和效果：**

基本达成预期效果：

静态效果



动态效果课上已展示

**六、总结**

**收获：**1、对于计算机专业的同学来说，知识更新的速度非常快，很多东西需要我们自学。在小学期这段时间，有很多没接触过的知识都是靠自己在网上自学，所以在这方面有很大进步

2、在不断修改bug的过程中，提高了自己的编程能力，把编程的基础打 得更牢靠了

3、这段时间的学习，让我非常深刻的意识到何为百折不挠，永不言败，我认为这也是程序员很重要的一个品质。也明白了合作对于一个程序的完成的重要性，学会合作是对每个程序员的要求。

**不足：**1、在效率方面有很大的提升空间，效率太低完成的进度太慢

2、在精益求精方面还要对自己要求更高一些

3、编程的基础确实不牢固。