C语言吃豆豆实验报告

**一．选题**

吃豆豆小游戏简介

吃豆豆是一款经典小游戏，玩家需通过控制吃豆人吃掉豆子同时躲避怪兽来取得胜利。

本次设计的吃豆豆小游戏着重实现了以下功能特性。

1. 吃豆人的方向通过键盘上下左右键控制。
2. 开始界面与结束界面加载位图，开始界面弹出帮助窗口。
3. 按空格键暂停与恢复。
4. 实现五只怪兽随机移动，吃豆人碰到怪兽即死亡。
5. 画地图，除怪兽是加载的位图外，吃豆人、豆子、墙壁均为笔与刷所画。
6. 实现计分，一个小豆子记一分，一个大豆子记十分。
7. 吃豆人走过的地方图像的擦除。

**二．方案**

方案设计：

此游戏分为四个文件

|  |  |
| --- | --- |
| 文件名 | 作用及功能 |
| pacman.h  头文件  实现的主要功能是定义各全局变量、声明宏定义、声明接口函数，实现函数封装。  pacman.h | 一．定义数据： |
| 1. 计时器ID的宏定义 |
| 1. 枚举游戏状态（开始/进行中/胜利/失败） |
| 1. 吃豆人状态的宏定义（胜利/死亡/错误） |
| 1. 吃豆人及怪兽方向的枚举及重命名 |
| 1. 定义表示游戏坐标的结构体 |
| 1. 定义指向各怪兽坐标的结构体指针 |
| 1. 定义暂停的标志量pause，pause=1为暂停 |
| 1. 定义吃豆子的分数score，score=178时玩家获胜 |
| 1. 定义表示地图的二维数组map[17][17] |
| 1. 定义表示吃豆人是否死亡的标志量Dead\_flag |
| 二．声明pacman.c与game-operation.c中的接口函数： |
| 1．Pacman.c中的接口函数 |
| （1）void init()  初始化地图中各物体的坐标 |
| （2）小豆子、能量豆、墙壁、吃豆人·、边界的坐标设置（set及获取（get）函数 |
| （3）int PacManMove()  控制吃豆人移动的函数 |
| （4）void SetDirction(dirction dir)  设置吃豆人方向 |
| （5）dirction GetDirction()  获得吃豆人方向 |
| （6）PGAME\_COORD MonsterMove(PGAME\_COORD Monster)控制怪兽移动的函数 |
| 2．game-operation.c中的接口函数 |
| （1）void initState()  初始化游戏状态 |
| （2）int GetScore()  得到吃豆人当前分数 |
| （3）void CreateGame(…)  创建游戏的内部数据结构和系统对象、初始化 |
| （4）void OnTimer(HWND hwnd);  计时器函数，游戏控制的一个主要流程。 |
| （5）void OnKeyDown(DWORD vk, HWND hwnd)  键盘响应函数 |
| （6）void GamePaint(HWND hwnd)  画图函数，用于地图及各物体的绘制 |
| Pacman.c  实现的主要功能是为各全局变量赋初始值，以及定义吃豆人和怪兽的移动函数，是游戏功能的主要实现部分。  Pacman.c | 一．数据赋初始值 |
| 1．吃到豆子的数量score=0 |
| 2．表示吃豆人是否死亡的标志量Dead\_flag=0（未死亡） |
| 3．为map[17][17]赋值，0、8、2、3、4分别表示小豆子、能量豆、墙壁、吃豆人、怪兽。 |
| 二．函数声明 |
| 1．全局函数 |
| （1）void init()  遍历map[17][17]数组，初始化各物体的位置；  为指向怪兽的结构体指针分配空间并赋值。 |
| （2）设置（set）小豆子、能量豆、墙壁、吃豆人及边界的坐标，以便在ui.c中遍历数组画图时将数组的行列值赋给对应的结构体坐标。 |
| （3）获得（get）小豆子、能量豆、墙壁、吃豆人及边界的坐  标，以便在吃豆人及怪兽的移动过程中作相应处理。 |
| （4）void SetDirction(dirction dir)  设置吃豆人的方向 |
| （5）dirction GetDirction()  获得当前吃豆人的方向 |
| 2．内部函数声明 |
| （1）int PacManMove()  用于吃豆人移动一步，根据下一步位置上原有的物体作出判断并返回对应值。  若下一步为墙壁，则不移动；  若下一步为豆子，则下一步的位置变为吃豆人，吃豆人原来的位置变为空地，计分器加1（小豆子）或加10（能量豆）。  【注】此处不处理下一步为怪兽的情况，因怪兽移动过程中已处理下一步为吃豆人的情况。 |
| （2）PGAME\_COORD MonsterMove(PGAME\_COORD p)  控制怪兽随机移动一步。  传入的参数和返回值都为指向怪兽坐标的结构体指针。  此函数的功能是实现：  1．只要吃豆人在身边怪兽就立即靠上去（坐标相差1），吃豆人死亡标志量Dead\_flag变为1。  2．随机生成怪兽方向，根据方向进行下一步移动。  3．移动过程中，若下一步为墙壁或怪兽则重新生成移动方向，若为其它则交换怪兽与下一步物体的位置。  【注1】由于一个位置不能存储两个物体，故怪兽走过时物体位置保持不变实际上是通过交换怪兽与物体的位置实现的。这造成了一个小小的bug，怪兽的移动是通过交换位置实现的，因此有时怪兽走过的地方的豆子会发生一步的位移。  【注2】为了增加游戏难度，此处怪兽移动方向不受限，即可以往反方向移动。 |
| game-operation.c  实现游戏过程中的各种操作控制。 | 一．全局变量定义 |
| DWORD dwTimerElapse  计时器到时间隔，决定了吃豆人的移动速度。 |
| 二．函数定义 |
| 1．int GetScore()  返回计分器积分，以便在窗口下方显示实时积分。 |
| 2．void OnTimer(HWND hwnd)  计时器处理函数。  计时器到时以后，吃豆人移动一步。  根据以后的状态，进行后续处理：  先判断Dead\_flag值。若为1则停止计时器并退出，否则判断吃豆人移动函数PacmanMove的返回值，有胜利、错误及正常移动一步三种情况。 |
| 3． void OnKeyDown(…)  键盘处理函数，通过键盘上下左右键响应SetDirection函数来控制吃豆人的移动。 |
| 4．void CreateGame（…）  创建游戏的内部数据结构和系统对象  当pause=0时，创建两个计时器，分别控制吃豆人和怪兽的移动速度，游戏开始或继续。  当pause=1时，停止两个计时器，游戏暂停 |
| ui.c  主要功能是绘图及创建窗口显示画面。 | 一．变量声明 |
| 1.宏定义 |
| （1）画图时使用的表示的圆形的直径像素点个数 |
| （2）用来绘图的颜色 |
| （3）游戏的参数的设置，如游戏界面大小等 |
| 2.各变量定义 |
| 用于画图的句柄等等 |
| 二．函数定义 |
|  |
| 1．void initState()  初始化游戏状态，给状态量gameState及暂停标志量pause赋初始值。 |
| 2．void initDC()  位图加载函数，将加载位图的语句单独封装为一个函数，只在函数创建时调用一次，避免每次画图都要调用，缩小占用的内存。 |
| 3．int WINAPI WinMain（…）  入口函数 |
| 4．void GamePaint(HWND hwnd)  画图函数  创建所需要的各种笔和刷  并根据游戏状态量gameState加载不同的场景 |
| 5．LONG CALLBACK MainWndProc（…）  消息处理回调函数  根据收到的消息不同分别进行窗口创建、绘图、键盘响应、启动计时器及销毁窗口等操作。 |

**三．关键技术**

1. 五只怪兽同时随机移动

这个是花了最长时间的部分了，开始时我准备定义一个数组，以下标来区分五只怪兽，然后用for循环来实现五只怪兽同时移动一步。

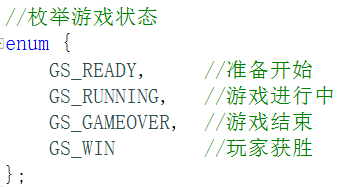
后来发现这样不行，得到的结果是同一只怪兽一次移动了五步。于是我开始单步调试，发现每次调用怪兽移动函数MonsterMove时，指向怪兽的指针变量都会变为初始值。原来是因为传入的指针每次都指向了同一块内存，且每执行一次怪兽移动函数就会调用一次初始化函数。

我的解决方案是定义五个不同的指针来表示五只怪兽的坐标，在ui.c中计时器启动时将这些指针作为参数传入怪兽移动函数，并将返回值赋给这些参数，以此实现五只怪兽坐标随时更新。

此外，我在初始化函数中为五个指针都分配了新空间，且给它们赋初始位置值，初始化函数只在游戏创建时调用一次，有效地解决了上述问题。

1. 游戏开始、游戏进行中、游戏失败、游戏胜利等界面的转换

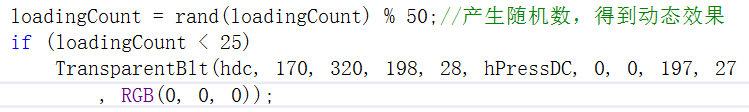
这是通过游戏状态量gameState 及相应的枚举变量



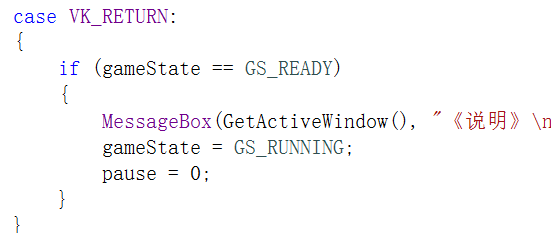
再通过与ui.c中绘图函数相互配合，使用if语句体判断gameState的值，实现游戏状态不同时，加载不同的界面，绘制不同的图像。

1. 开始界面文字的动态效果及弹出帮助窗口

使用了一个产生随机数的函数rand()，使其产生0~50之间的随机数，当产生的随机数大于25时，用设置背景透明的函数TransparentBlt显示“PREEE ENTER TO BEGIN”（按enter键开始游戏）图片。



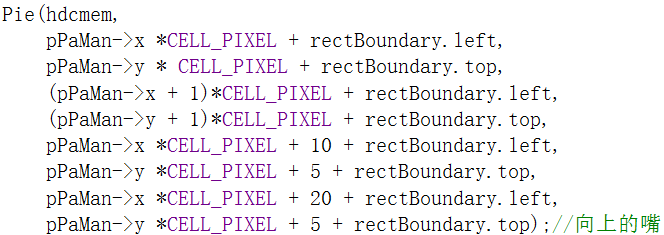
弹出帮助窗口是通过键盘响应函数实现的，当游戏状态处于开始界面且玩家按下enter键时，用MessageBox弹出窗口，显示说明。

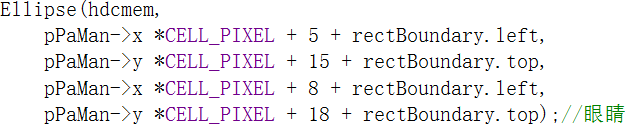


1. 吃豆人嘴和眼睛的位置随方向的不同而改变

通过switch不同方向来画扇形及在扇形上画圆实现。

画这些让我彻底理解了Pie函数、Ellipse函数及Rectangle函数的参数使用。





1. 计分器的实时显示

这个挺简单的，借鉴了老师贪吃蛇程序的代码。通过score变量及GetScore函数来获取实时分数，并计算好在窗口中显示的位置，通过画图函数显示。

**四．结果和效果**

**1．**开始界面

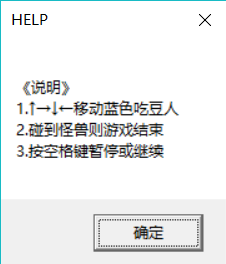


开始界面：

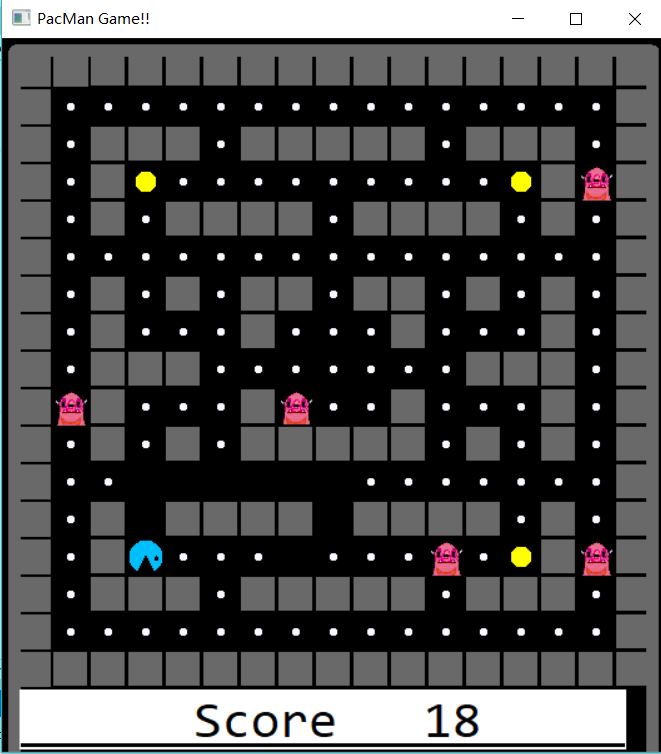
PRESS ENTER TO BEGIN图片有动态闪现效果。

当按下enter键，

弹出窗口：

****

**2．**游戏进行界面



白色的是小豆子，吃一个加一分

黄色的是能量豆，吃一个加十分

红色的是怪兽

蓝色的是吃豆人

灰色的是墙壁

窗口下方显示当前分数

**3．**游戏失败界面



失败界面使用sleep（2500）延迟，保持几秒后关闭窗口。

**4.**游戏胜利界面



胜利界面显示吃完了所有豆子，并显示最终分数177

使用sleep（2500）保持界面几秒钟。

**五．总结和不足**

**收获：**

**首先是知识上的，对窗口程序及句柄、设备环境等等都有了了解，不再只会用黑乎乎的控制台程序窗口啦。**

再就是通过本次游戏设计，了解了一个较大型程序开发的基本流程，对程序中函数的封装及功能的重用有了进一步的理解。编程能力不说有特别大的提升，但一定是进步了很多的。

除了知识上的进步之外，自己的专注力也提升了很多，能在电脑前坐很多个小时、专注在一件事情上是一件很幸福的事情。每天过的很充实，做成这个游戏很有成就感。

**不足：**

本次游戏设计还有一些想实现的功能没有实现，如吃豆人吃到能量豆时应该做的相应变化是怪兽变色，此时变为无敌模式，吃豆人可吃掉怪兽。这个功能的实现要用到倒计时，其实其它的功能如吃豆人吃到能量豆时怪兽变色等都差不多实现了，但是倒计时没有找到合适的函数实现。以后会继续完善这个功能。

本次游戏设计暴露出我的一个很大的不足是逻辑不清晰，且对计算机内部的存储方式不很了解，对数据的内存分配不理解，这也是实现五只怪兽同时随机移动花了很多时间的原因。

**总结：**

我想我以后要提高编程能力，应该多锻炼自己的抽象思维能力，解决一个问题尝试用多种算法实现。

以后还要花更多的时间编程。勤能补拙。