任务5.1：绘制地图 功能

2023年6月26日

16:11

**开始游戏**：是整个程序的开始，功能比较繁杂，需要执行以下功能：

（1）绘制游戏地图

（2）启动游戏模式

（3）接受鼠标输入，给选中的图片画边框，判断选中的两个图片是否可以消除

（4）如果有定时器的话，就启动定时器计时功能；

CGameDlg类

用来和用户进行交互，接受用户的鼠标点击事件和按钮的点击事件。

CGameControl类

用来实现业务控制操作，针对用户发起的事件比如开始游戏，暂停游戏，重排等做出具体的响应。

CGameLogic类

用来实现业务逻辑操作，设计良好的数据结构，并基于此结构完成具体的功能。

因为CGameDlg->CGameControl->CGameLogic 存在单向调用的关系，在设计时，我们采用从上到下的方式来设计，但是在实现时，需要按照从下到上的实现方式。

**（1）绘制地图功能实现**

（1.1）类CGameLogic的设计

新构建一个类CGameLogic类来实现具体的和数据结构有关的逻辑实现，供CGameControl类调用。

该类中给出具体采用的数据结构——二维数组，以及基于该结构给出的一些基本操作

计算机生成了可选文字:
AclassCGameLogic
public
//她信息
Int**GameMap《
//她行
IntrOWS
//她0
1]Itcols
//她包合的片的个数
1]ItpicNL皿
private.
stack<Vertex>verList;
p巛b112
CGameLogic0
IntGetElement(Int
所要厍到的数据结构
三==一仁三一苤过
基于该结构的相关操作
nRow
、℃IdInitMap(IntnRows，
Intncol)
nCOlItnPicNum)
int
//相当于对GameMapi*行莉始化

这里我们给出来地图的定义，用一个二维数组来保存，以及相关的一些参数，比如地图的行，列，和包含的图片的个数。同时给出了三个函数的定义。

~CGameLogic() 用来释放地图信息资源，防止内存泄漏

计算机生成了可选文字:
ACGameLogic．：CGameLogic0
卜上for(inti=0，i〈rows：
deleteC)GameMapC1],
deleteC)GameMap;

GetElement()根据输入的地图坐标，返回图片代表的ID；

计算机生成了可选文字:
刁Int
CGameLogic：GetE1ement(intnRow，
returnGameMapCnRownC01
intnC01)

InitMap()则是根据行，列和图片的个数，对GameMap进行初始化。

初始化地图需要提供3个参数，3个参数需要进行合理性检查，这里将合理性检查放在上层代码去控制

所以传入进来的三个参数，我们默认是合法参数

计算机生成了可选文字:
//这里不做参数合理性判断，判断放在前面进行
AvoidCGameLogic：InitMap(intnRows，intnC01s，intnPicNum)
nlndexl‰nC01s
闩
闩
闩
辟空间
this—>GameMa=newint*CnRows]；
for(inti：0；i〈nRows，i++)
GameMapCil
newintCnCOls
memset(GameMapCi]，NULL，sizeof(int)*nC01s)；
//用图片编号填充地图
intnRepeatNum
intnCount=0；
for(inti：0；
forint
GameMa
nCount
(nRows*nCols)/nPicNum
i〈nPicNum;i++)
·〈nReeatNum•++
nCountnColsnCount％nC01s
对地冬GameMap进行初始化
//将图片随机打散
srand((int)time(WLL))，
intnVertexNumnRows*nCols
for(inti=0；i〈nVertexNum；i++)
intnlndexl=rand0％nVertexNum；
intnIndex2=rand0％nVertexNum；
//两个下表交换
intnTmp_GameMapCnlndexl/nC01s
GameMapCnlndexl/nC01s]nlndexl‰nC01s
GameMapCnIndex2/nC01s]nIndex2‰nC01s
GameMapCnIndex2/nC01s]CnIndex2％nC01s
nTmp;

（1.2）类CGameControl的设计

CGameControl类包含1个属性，一个CGameLogic类对象，用来进行具体的逻辑实现；

同时定义个StartGame()函数表示开始游戏功能和GetElement()函数表示获取地图中的具体图片，这两个函数都供CGameDlg进行调用，具体定义如下图所示。

计算机生成了可选文字:
Once
*include"CGameLogic.h"
classCGameContr01
public:
CGameLogicmGameLogic;
public:
//实现具体的逻辑
CGameContr010；//带参数的构造函数
intGetElement(intnROW,
intnC01)，
//获取地图中的数据
intpicNum)；//开始游戏，供CGameD1g类调用
boolStartGame(introws,
intcols，

GetElement()函数

计算机生成了可选文字:
闩int
CGameControl:GetE1ement(intnRow，intnC01)
returnmGameLogic.GetElement(nRownC01)；
通过EECGameLogic的GetElement0函数实

StartGame()函数

计算机生成了可选文字:
@boolCGameControl：StartGame(introws,
intcols，intpicNum)
@//首先判断3个参数是否合理，不合理，返回false
//进行地图初始化，填充m_pGameMap返回true
if（(rows*cols)‰(picNum*2）！=0）
returnfalse;
else
汶里进行了参台理性判断
mGameLogic.InitMap(rows，cols，picNum)；
returntrue；

（1.3）类CGameDlg的设计

CGameDlg类主要来进行具体地图的展示，其主要步骤分为下面3步：

Step1：需要将图片和图片掩码读入到系统，编写InitElement()进行该操作，并且在OnInitDialog()函数中进行调用；

Step2: 需要对地图进行初始化，通过调用CGameControl的StartGame()函数实现，该操作主要是得到初始化的地图；

Step3：编写UpdateMap()，将地图展示在CGameDlg界面上；

类CGameDlg的定义

计算机生成了可选文字:
CGameDlg
publlcCDialogEx
DECLAREDYNAMIC(CGameDlg)
p巛b112
CGameDlg(CWnd*pParent
nullptr
virtual、CGameDlg0
//对诟框数据
E#ifdefAFXDESIGNTIME
ni-皿[IDD=IDD_GAIE_DIALOG}，
//析准造函数
#endif
private.
1]ItGameMode
protected：
/标识渫戏模式
virtual、℃idDODataExchange(CDataExchange*PDX)
//DDX/DDV0；三
HICONmhlcon;
内存DC
CDCmdcMem;
CDCmdcEle
staticCStringBGPath《
staticCStringElementPath．
staticCStrinMaskPath．
CGameControl象
新增的属性
背到片路径
揎码片
CGameControlmGameC
DECLARE
public
virtual
afxmsg
afxmsg
afxmsg
afxmsg
afxmsg
MESSAGEMAP()
BOOLOnlnitDialog0，
voidOnPaint0
voidOnC1ickedBtnStart0
voidOnclickedBtnStop0
voidOnclickedBtnReset0
voidOnclickedBtnPrompt0
p巛b112
VOld
private.
VOld
VOld
VOld
VOld
VOld
SetGameMode(Intmode)
InitMode(int
mode)
InitBackground(void)-
三=一三=
新增的方法
SetButton(boolstart，b001stop,b001以ompt,b001reset)
InitElement(CStringElementPath，CStringMaskPath)
UpdateMap（、℃Id），

（1）加载图片和图片掩码

定义了两个CDC属性 m\_dcEle和m\_dcMask分别用来存放图片和图片掩码的信息，通过InitElement()函数来进行加载

计算机生成了可选文字:
到、℃IdCGameDlg：：(CStringElementPath，CStringMaskPath)
CCIientDCdc(thls)，
HANDLEhBmp=：：LoadlmageW(NULL,ElementPath，IIIAGE_BITMAP，0，0，LR_LOADFROMFILE)
mdcE1e.CreateCompatib1eDC(&dc)
[〔0婴0《《]Select()bject(hBmp)，
冬片加载到mdcELe中
：Loadlmagew(WLL,[氬氬《]IILAGE_BITMAP,0，LRLOADFROlvFILE)，
HANDLEhMask
mdcMask.CreateCompatib1eDC(&dc)
[叵§4§下《SelectObJect(hMask)，
冬片掩码加载到mdcMask中

该加载操作应该放在对话框初始化的时候进行，因此同步修改OnInitDialog()函数

计算机生成了可选文字:
//CGameD1g泽息处畦程序
闩BOOLCGameDlg：OnlnitDialog0
CDialogEx:OnlnitDialog0，
//whentheapplicationSmalnwindowisnotadialog
三
SetIcon(mhlcon，TRUE)
SetIcon(mhlcon，FALSE)
//Setbig《20n
//Setsmall《20n
让弹出窗囗作为应用程序窗囗显示
th1s¯>ModifyStyleEx（0，WS_EXAPPWINDOW)
InitBackground0
//设置模式
InitMode(this—>GameMode)
设置渫戏窗囗按钮的初始状态
SetButton(TRUEFALSE，FALSE，FALSE)，
增加了汶个部分
一0二一层0
InitElement(ElementPath,MaskPath)
returnTRUE：/／returnTRUEunlessyousetthefocustoacontrol
OCX属性页应返回
FALSE

（2）对地图进行初始化

通过调用CGameControl的StartGame()函数，得到初始化的地图。该地图实际上是存在CGameLogic的GameMap属性中，因此该操作实际上是调用了CGameLogic的InitMap()函数。

修改OnClickedBtnStart（）函数实现地图初始化功能：

计算机生成了可选文字:
//开始渫戏按钮
、℃IdCGameD1g：OnclickedBtnStart0
b001status=mGameC.StartGame(ROWS,Cols，PicNum)
三
Ifstatus//口
初始化操作
/设置状态
paylng_true
firstSelect
0礻三
TRUE)，
UpdateMap0，/／展示她信息
=：一新丁二酽．0三三舌二飞
AfxMessageBox(L"
，MBOKCANCEL)，
InvalidateRect(FALSE)

（3）地图刷新

UpdateMap() 用于在对话框窗体中显示当前地图，其主要过程包括2步骤：

A)通过BitBlt方法先在地图的合适位置，用SRCPAINT模式画图片的掩码，注意掩码黑的部分都是用0表示；

B)通过BitBlt方法在地图的合适位置，用SRCAND模式画图片，图片的背景部分和掩码进行与运算后，会被清除掉，显示去掉背景的效果；

BitBlt(int x, int y, int width, int height, CDC \*p, int xSRC, int ySRC, DWORD dwp )

表示从\*p中的(xSRC,ySRC)的位置读取图片，用dwp的方式，画在 左上角坐标为(x,y)，大小为（width,height）的位置。

具体操作如下。

计算机生成了可选文字:
、℃IdCGameDlg：UpdateMap0
读取她的信息，并且在界面上显示
Int
Int
Int
Int
for
nTop=PTOP，
nLeft
_MAPLEFT，
nElemW
PICWIDTH
nElemH=PICHEIGHT．
(inti
i〈Rows，i++)
for(intj
j〈C01j++)
氵意：汶里是级联调厍，最终是在GameLogi（中调用GetE》ement从地图中获取对应内容
ItnElemVal=mGameC.GetElement（．
mdcMem.BitBlt(nLeft+j*nEl旧，nTop+inElemH，
mdcMem.BitBlt(nLeft+j*nEl旧，nTop+inElemH，
nElemW，
nElemW，
nEl巳n旧，
nEl巳n旧，
&mdcMask，0，nE1emVa1冰nE1emH，SRCPAINT)
&mdcE1e,0，nE1emVa1*nE1，SRCAND)，

（4）运行程序

所有的代码修改已经完毕，编译运行，点击 开始游戏，出现地图，如图所示

计算机生成了可选文字:
00巛》00孬“00、丿00丿
0。0巛．琴巛0砌·“0飞是丿0
智00飞丿00
过、、丿：丿少一0u000晷河@