任务4：创建游戏界面背景

2023年6月4日

21:04

这个任务包含3个部分：

（1）创建一个新的游戏界面(基于对话框的窗体)-游戏主界面，绘制相应的背景；

（2）通过主程序界面上的“基本模式”按钮，能够打开这个“游戏主界面”

（3）“休闲模式”和“关卡模式”按钮也能够打开该游戏界面，并且呈现不同的界面的标题

（4.1）创建一个新的对框框

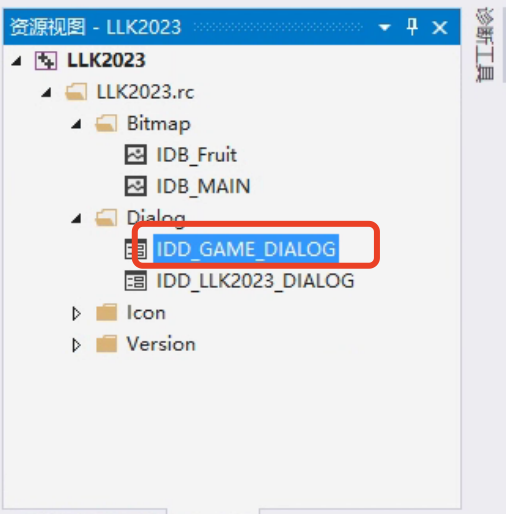
（1）在资源视图窗口，右键”Dialog“，在右键菜单中选择”添加资源“



（2）打开”添加资源“对话框，选择 ”Dialog“，点击”新建“会自动生成一个对话框

X 
0 12 
亡 盞름 
-통 도 〉 卜 > 

（3）修改对话框的ID为IDD\_GAME\_DIALOG



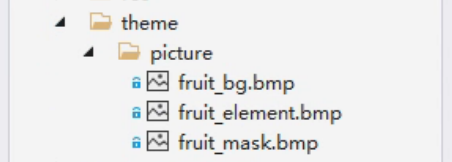
(4) 创建一个类CGameDlg和该界面IDD\_GAME\_DIALOG绑定

通过在界面上点击添加类，填写类名和绑定对应的对话框即可。

计算机生成了可选文字:
添加MFC类
类E旧
十文仁旧
灯话框IDO
IDDGAMEDIALOG
囗包括巨化支
囗包括ActiveAccessibility支
基类
CDialogEx
.cpp文仁凹

（4.2）背景资源导入

（1）将包含主题图片的文件夹，加入到项目中



（2）定义成员属性

在游戏主界面中，我们需要设置两个成员属性：

一个依旧是CDC m\_dcMem 用于表示内存，

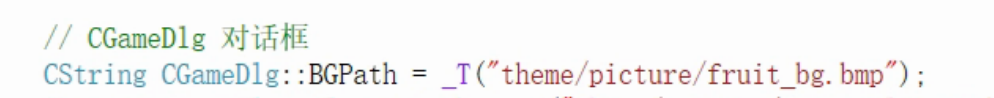
另一个成员属性是CDC m\_dcBG 用于表示背景

两者的区别在于：m\_dcBG用于保存当前加载的具体背景，而m\_dcMem表示内存，可以接受m\_dcBG的内容，通过调用BitBlt()函数去动态的刷新背景

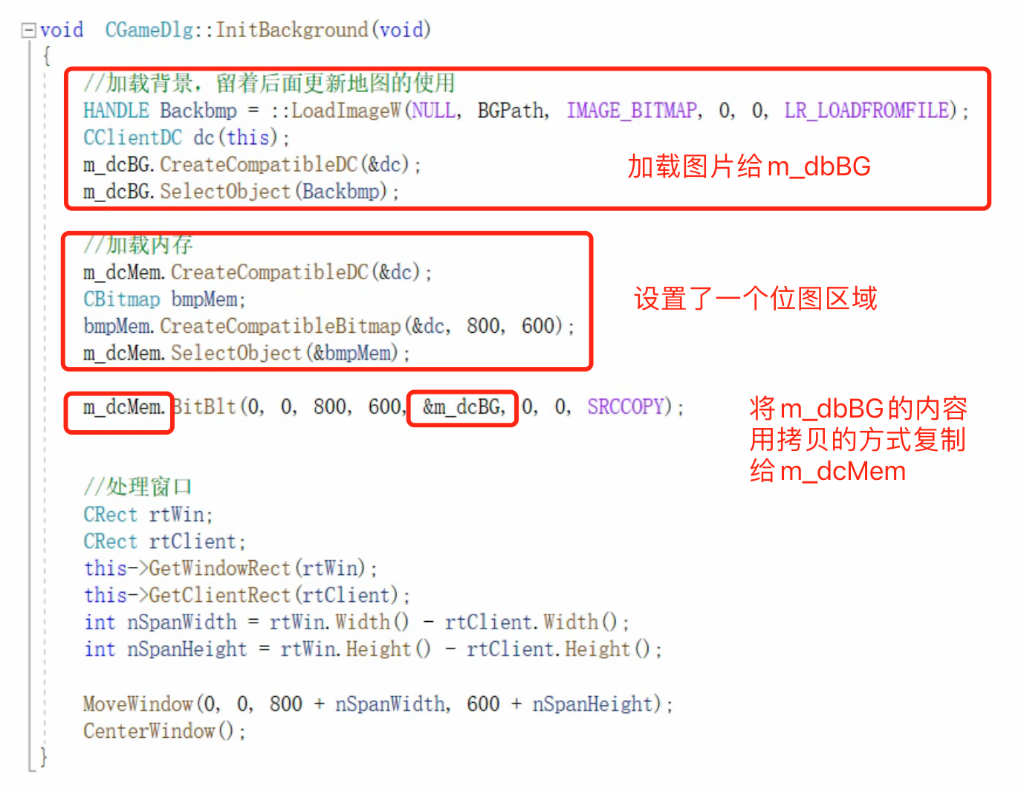


参考任务3的方法，编写CGameDlg的InitBackground函数，此时是动态加载背景

（A）首先给BGPath赋值



（B）然后进行InitBackground函数的编写



注意这里和任务3中 加载图片的区别

任务3中：现将位图导入到资源中，通过位图资源的ID来加载图片

具体代码如下：

声明一个Cbitmap对象bmpMain，然后这个对象去掉用LoadBitmapW()方法，这个方法的参数是一个已经定义好的位图资源IDB\_MAIN.

Cbitmap bmpMain;

bmpMain.LoadBitmapW(IDB\_MAIN);

CClientDC dc(this);

m\_dcMem.CreateCompatibleDC(&dc);

m\_dcMem.SelectObject(bmpMain);

而在任务4中：通过动态加载图片来实现

具体代码如下：

声明一个HANDLE对象bmpGame,然后调用系统的LoadImageW()方法来动态加载图片，其中第二个参数是加载图片的具体地址

HANDLE bmpGame = ::LoadImageW(NULL, BGPath, IMAGE\_BITMAP, 0, 0, LR\_LOADFROMFILE);

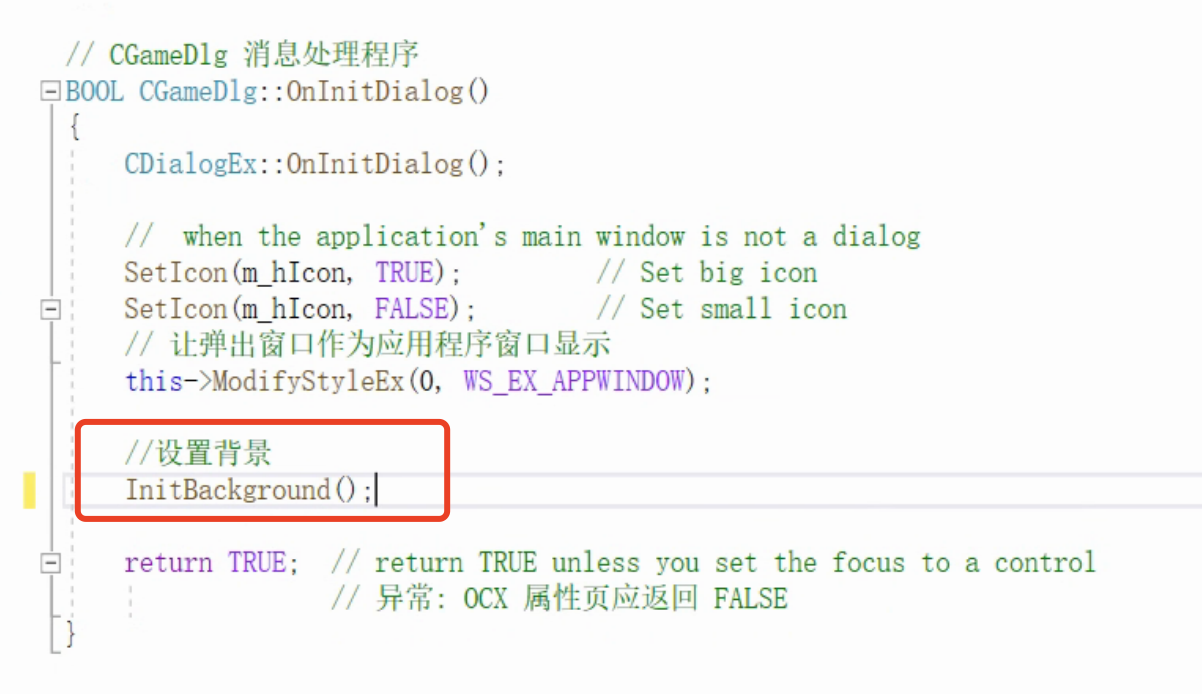
CClientDC dc(this);

m\_dcMem.CreateCompatibleDC(&dc);

m\_dcMem.SelectObject(bmpGame);

（4.3）给CGameDlg类添加OnInitDialog()

参考任务3的方法，打开类向导，添加OnInitDialog()函数，OnInitDialog() 函数内容

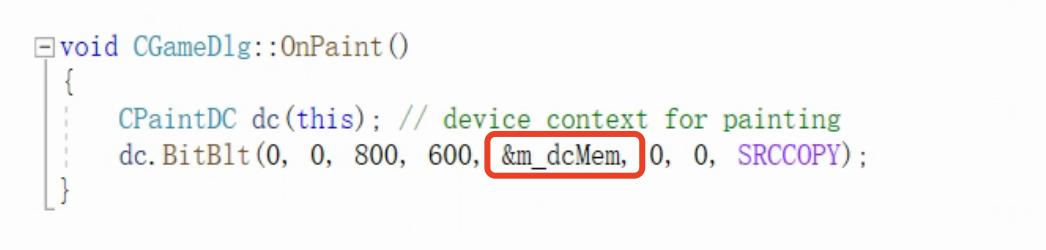


（4.4）给CGameDlg类添加添加OnPaint()函数

在消息标签下，找到WM\_PAINT，添加OnPaint()函数

LLQ023 
CDia log Ex 
IDD GAME DIALOG 
WM NCXBUTTONDOWN 
WM NCXBUTTONUP 
WM NEXTMENU 
WM NOTIFYFORMAT 
WM PAINT 
WM PAINTCLIPBOARD 
WM PALETTECHANGED 
WM PALETTEISCHANGING 
WM PARENTNOTIFY 
WM POWERBROADCAS 
WM QUERYDRAGICON 
WM QUERYENDSESSION 
WM QUERYNEWPALETTE 
WM QUERYOPEN 
CGa meDlg 
On?aint 
CGameDlg.h 
CGa meDlg .cpp 
WM PAINT 

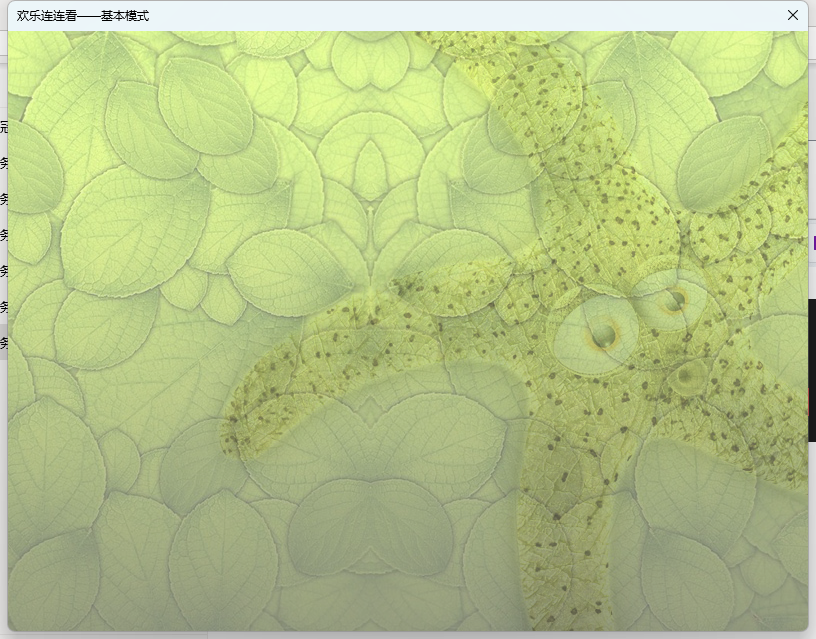
OnPaint()函数内容



注意：在OnPaint()函数中，是将m\_dcMem里面的内容，通过拷贝的方式放入到设备dc中，进行页面刷新，因此，在背景变换之前需要，更新m\_dcMem的内容，修改其内容也是通过BitBlt()函数的形式。

（4.5）运行程序

游戏界面就能够正常展示显示



（4.6）根据不同的“模式”展示不同的游戏界面

解决思路：对话框类中，设置一个参数来表示不同的模式，不同模式按钮点击时，就设置不同的参数，然后对话框根据对应的参数进行出初始化。

MFC中，对话框的构造函数和OnInitDialog()函数

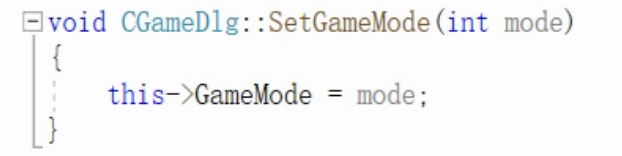
构造函数是在Dialog类对象构造时调用，这是类对象的构造，而不是对话框的构造

对话框的初始化函数是OnInitDialog()，该函数在对话框构造完成之后调用的，它是Windows消息WM\_INITDIALOG的响应函数。在对话框将要显示之前，Windows会向对话框发送WM\_INITDIALOG消息，OnInitDialog()函数就是响应这个消息的映射函数

（1）设置一个参数来表示模式



并且给出设置该属性的方法



（2）增加一个模式设置方法，用于对话框的界面进行个性化设置

计算机生成了可选文字:
]、℃IdCGameDlg：：InitMode(Intmode)
if(mode
this->SetWindowTextW()T（"次千莲莲看一一基本模式
elseIf00d巳
this->SetWindowTextW()T（"次千莲莲看一一休禾模式
this->SetWindowTextW()T（"次千莲莲看一一关卡模式

（3）修改OnInitDialog()函数，调用InitMode（mode）来进行模式设置



（4）修改主界面的三个模式按钮的点击函数

回到LLK2023Dlg.cpp代码窗口，找到OnBnClickedBtnBasic（），OnClickedBtnRelax（）和OnClickedBtnLevel（）函数，将其填充完整。

OnBnClickedBtnBasic（）函数例子如下所示：

计算机生成了可选文字:
//基本模式按钮
闩、℃IdCLLK2023D1g．：OnBnC1ickedBtnBasic0
//0藏三界面
this—>ShowWindow（HIDE)
//显示渫就界面，河0亍
0是休闲
CGameDlgdig,
1是基
dlg.SetGameMode(l)
其他是关卡
dlgDoModa10
//层示三界面
this—>ShowWindow（SHOW)

注意：OnInitDialog()函数会自动调用，根据设置的模式进行对话框的初始化操作，他将在DoModal()函数执行之前进行调用。

运行程序，展示主界面，点击 “休闲模式”按钮，展示如下界面

计算机生成了可选文字:
苁乐过一一

代码完成了各个不同模式的切换展示