

# 菜单对话框的设计

姓名：马胜楠 学号：2019020024

**摘要：**本程序设计实现的功能有：在菜单中添加消息响应框；并通过对工具栏的设计来改变画笔的颜色和粗细，并且可以选择多种绘画方式，比如画圆形、矩形、椭圆、直线、曲线等等；对于状态栏，实现了鼠标坐标的随动并在非绘画区域显示更多提示信息并且可以显示当前时间；绘制了坐标系；为程序设置了登录密码并添加了日历。

**关键字：**消息响应；画笔颜色和粗细；各种图形绘画；坐标；日历；密码器；时间显示

## 正文

通过本学期对 Visual C++的学习，制作一个单文档的菜单，尽可能的把所学融入其中。接下来，主要介绍一下程序设计的步骤。

### 一、菜单的建立

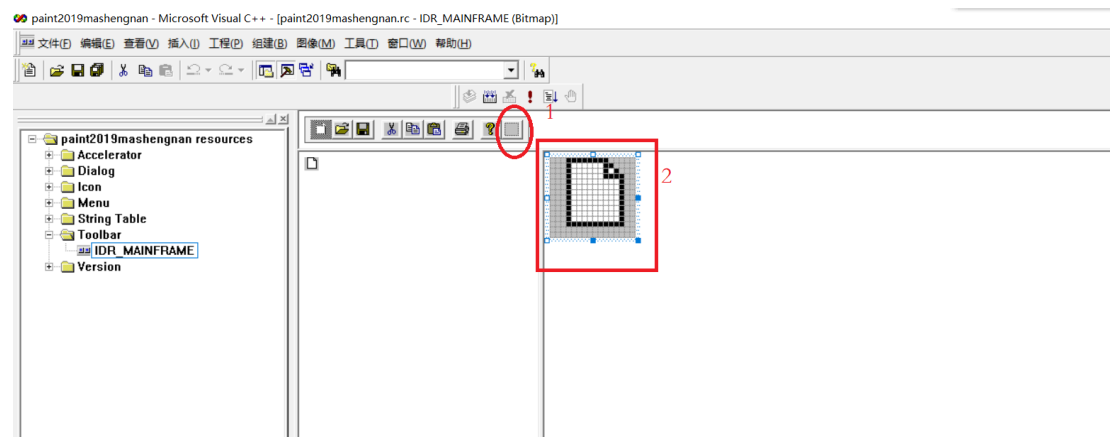
由于 win10 系统无法安装 MSDN，所以下所有步骤均为借阅课本上以及网上查阅的相关资料相结合并自己亲自码出来的。此论文的过程基本可以相当于教程，步骤非常的详细，希望老师喜欢。用户名 mashengnan，密码 123456

打开 Visual C++的软件，选择“file”，点击“新建”，就弹出如下图的对话框，选择工程中的 MFC AppWizard[exe]，输入工程名 paint2019mashengnan，其他选项默认，点击确定。接下来选择单文档，然后一直按下一步，最后点击完成。注意，我们要选择 MFC 标准风格以及最后的基类一定要是 CView 才可以。这样一个画图工程就建成了。

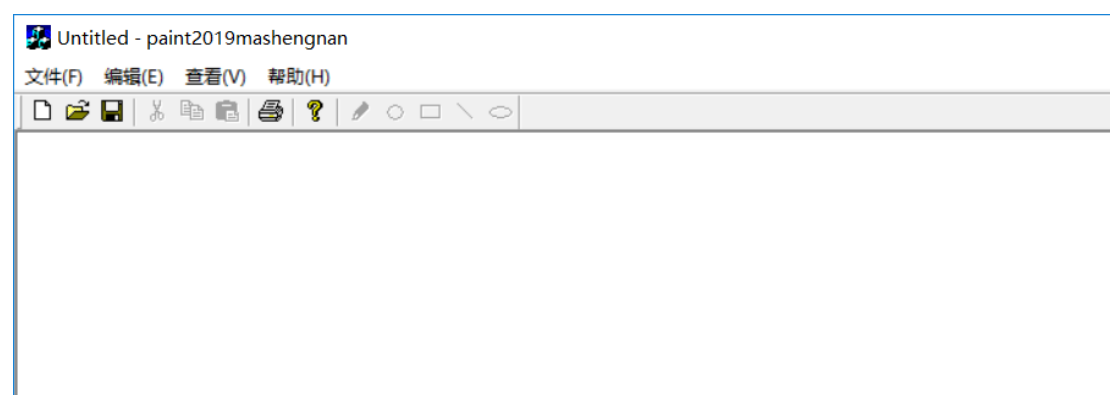


## 二、工具栏的添加和程序的编写

(1) 点开 ResourceView, 再点击 paint2019mashengnan resources 下拉菜单中的 Toolbar 选项, 双击 IDR\_MAINFRAME。

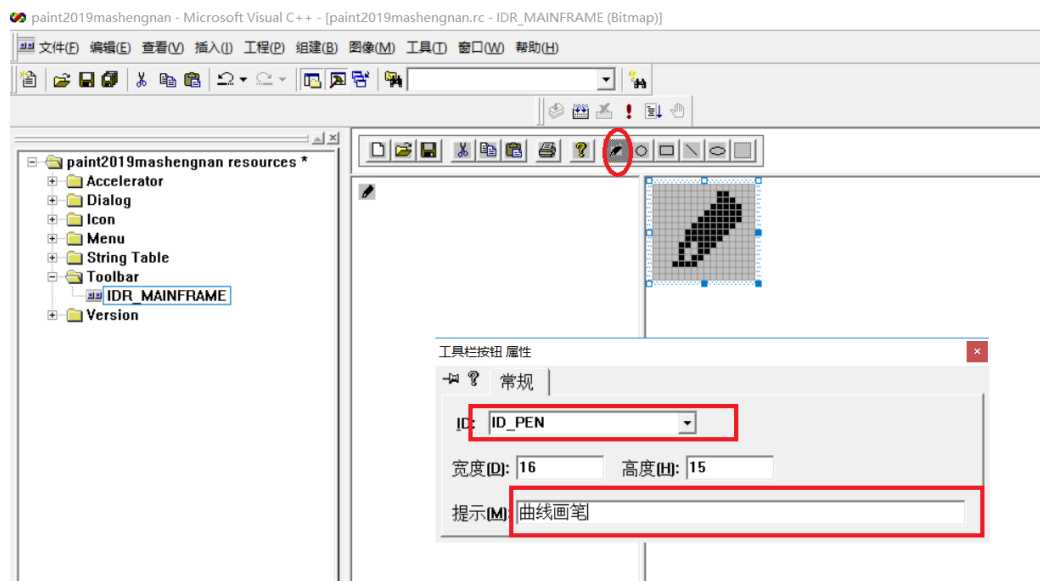


(2) 先点击上图中的圆圈部分, 创建一个新的工具栏图标, 然后在方框区域进行图标的绘制, 绘制好后我们保存运行。在这里, 我们将曲线、圆形、矩形、直线、椭圆等图标一一绘制并运行。

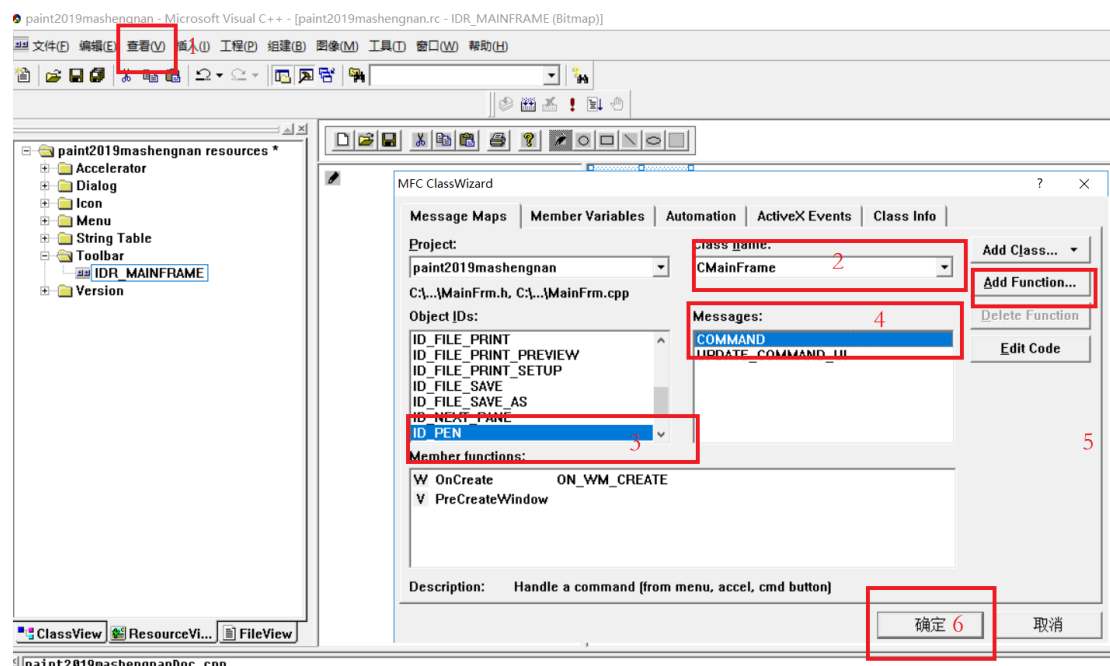


(3) 通过运行结果我们可以发现, 即使添加图标以后, 图标是灰色的, 无法进行点击, 这是因为我们并没有对其进行正规命名和 COMMAND 操作, 下面我们第一个图标, 也就是画曲线为例, 大致讲述一下添加代码的过程。

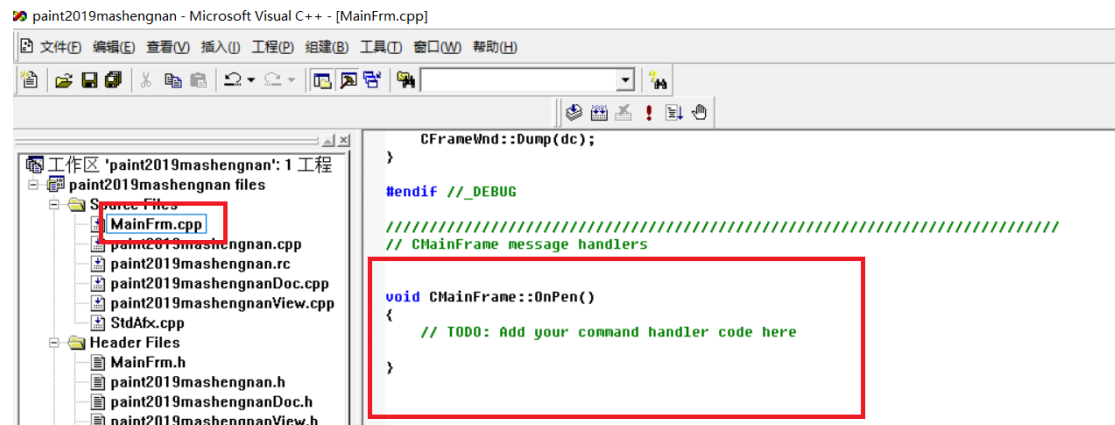
首先, 我们先双击下图中的圆圈部分, 进入设置界面, 我们将方框中的 ID 和提示分别改为 ID\_PEN 以及曲线画笔, 最后关闭窗口, 这样, 我们对这个图标的命名也就有了一个初步的定义。



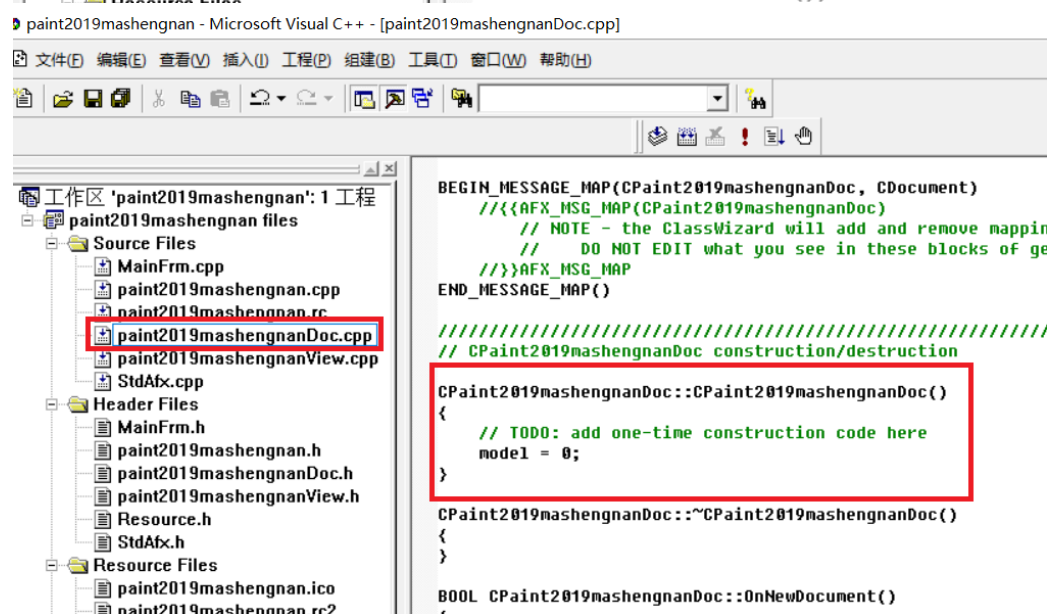
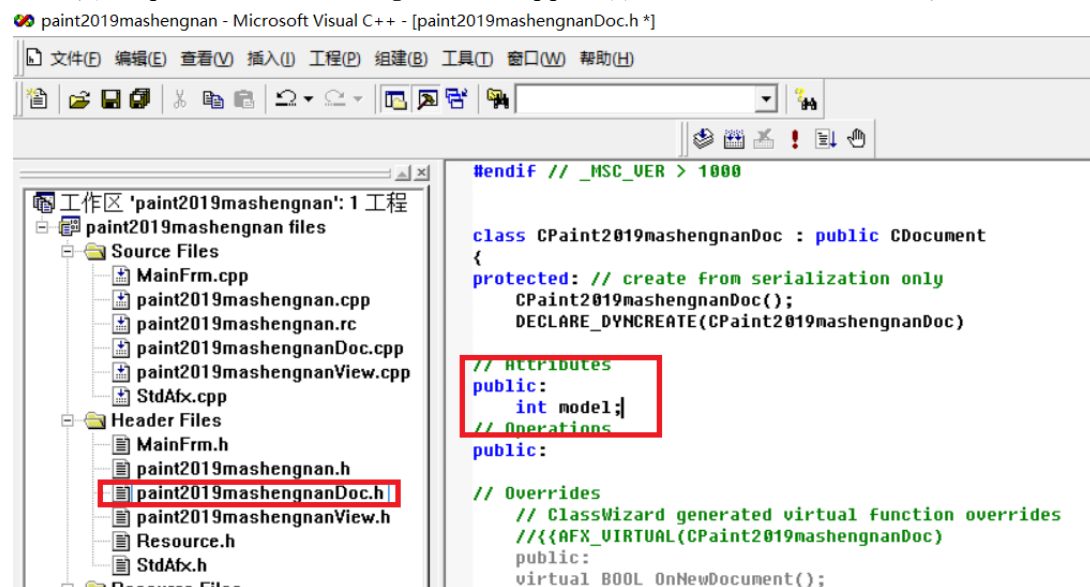
(4) 接着，我们起类向导，按照图中顺序，点击查看一>建立类向导，然后选择 CMainFrame，选择我们刚刚命名的 ID\_PEN，选择 COMMAND，点击 ADD FUNCTION，最后点击确认。



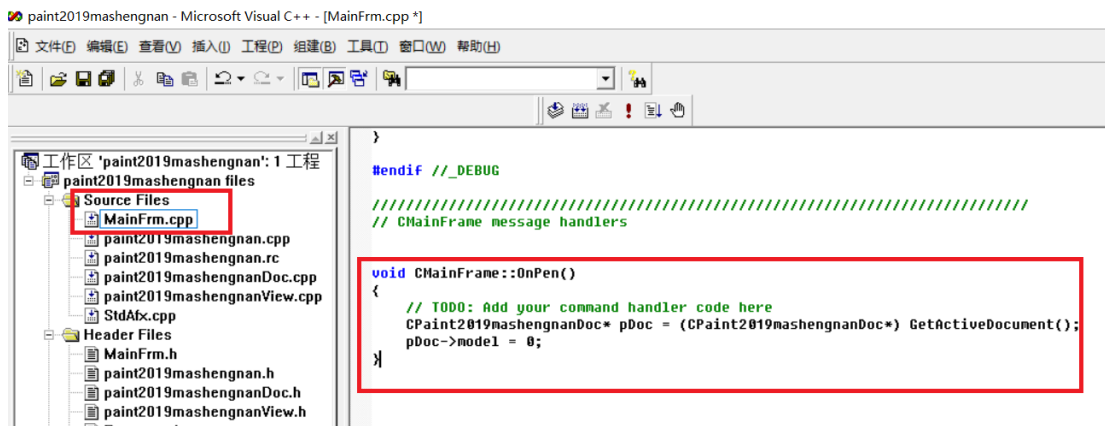
(5) 然后，在 MainFrm.cpp 中的最后，系统会自动添加一个 OnPen 函数，这个函数里面就是要添加点击画笔这个按钮后我们要做的事情。



(6) 因为我们有很多种画笔方式，那么我们为了方便定义，我们在 paint2019mashengnanDoc.h 中，新建一个 int model，用来定义当前画笔的模式，并在 paint2019mashengnanDoc.cpp 文件中，初始化定义其的值为 0。

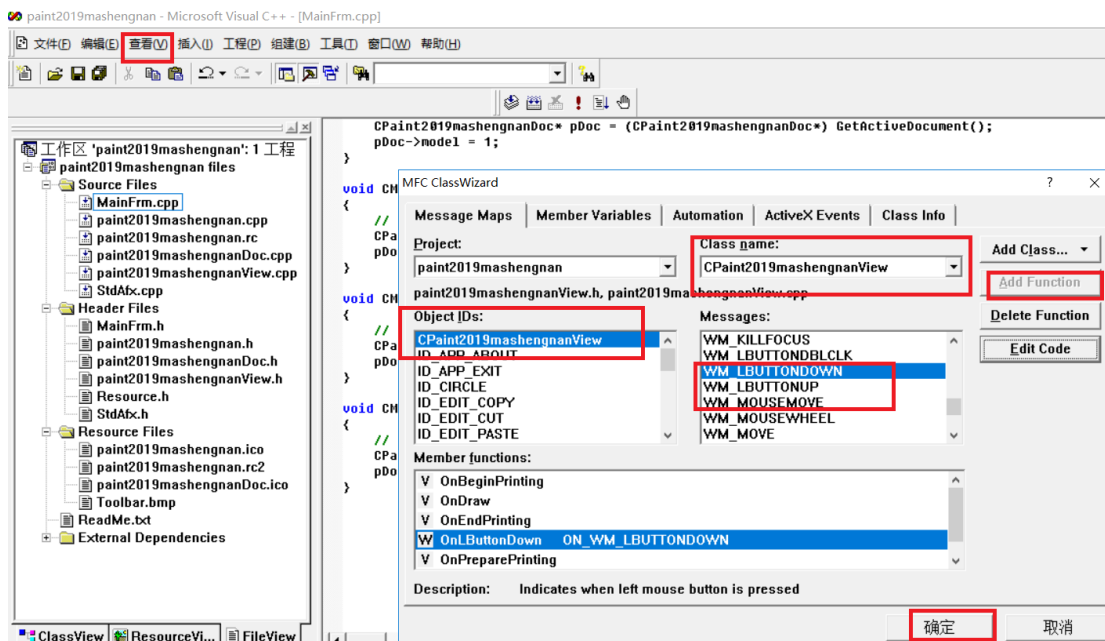


(7) 然后在 MainFrm.cpp 中的 OnPen 函数中添加如下代码，并添加#include “paint2019mashengnanDoc.h”，这样我们按钮的 COMMAND 就写好了：

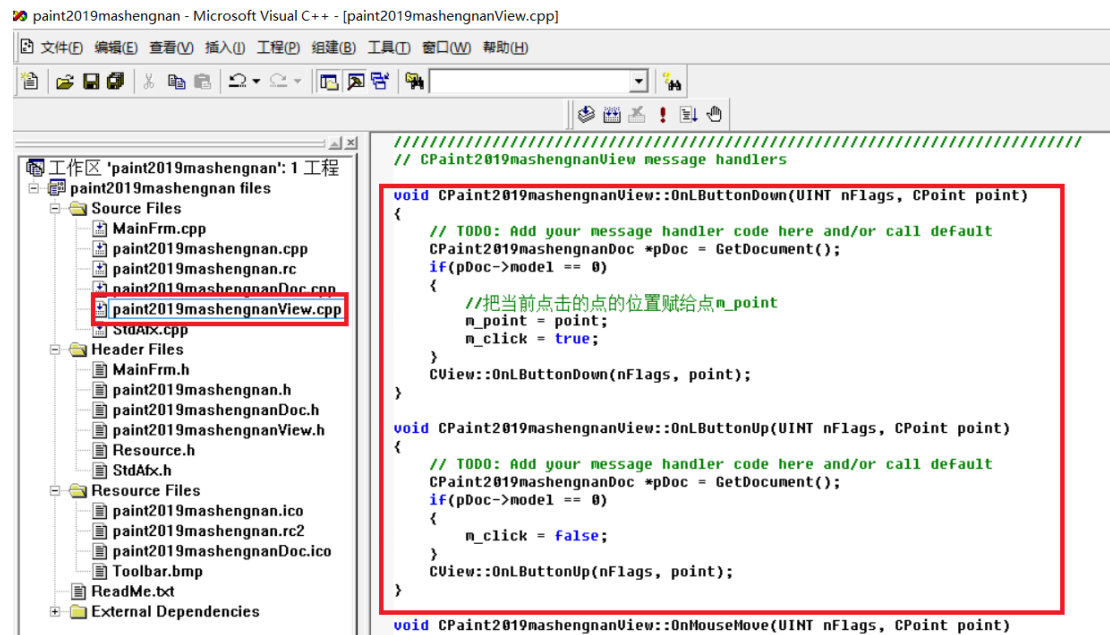


(8) 将剩下的几个按钮按照同样的方式进行赋值，分别赋值 model 为 1, 2, 3, 4..., 命名分别为 ID\_CIRCLE, ID\_TANGLE, ID\_LINE, ID\_ELIP

(9) 在赋值写好后，我们接下来对画板进行操作，我们知道，我们画图需要按下左键，鼠标移动和抬起左键三个部分，那么我们对于这三个部分进行，那么我们分别对这三个动作建立类向导。具体步骤按图中所示，成功后，我们应该可以在下图中的 Member Functions 中看到这三个函数，并且在 paint2019mashengnanView.cpp 的最后看到这三个函数。



(10) 我们在函数中添加如下画直线代码：



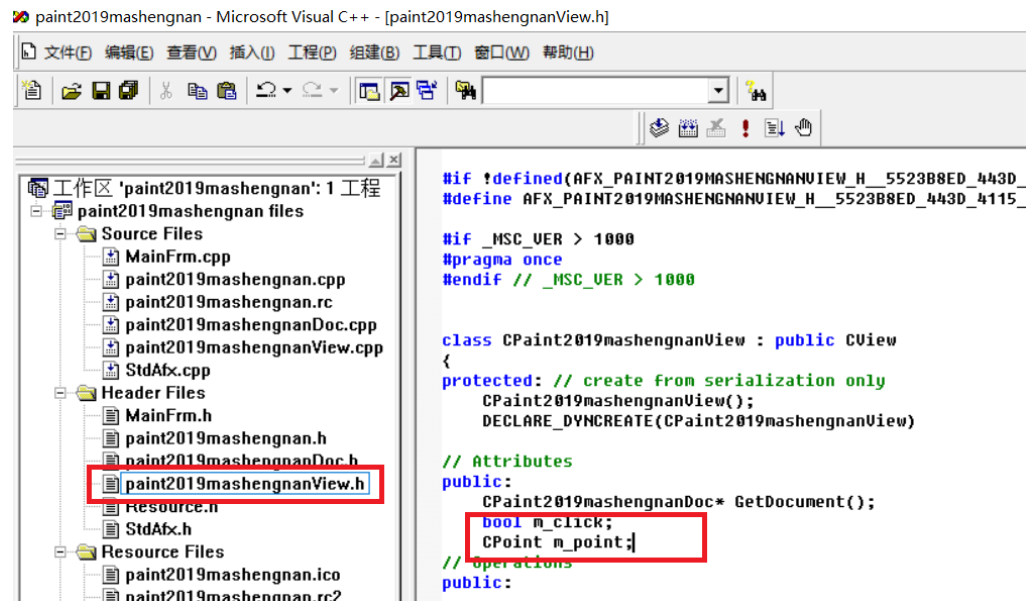
```

void CPaint2019mashengnanView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    CPaint2019mashengnanDoc *pDoc = GetDocument();
    if(pDoc->model == 0)
    {
        //画图
        CDC *p=GetDC();
        CPen pen(PS_SOLID, pDoc->thickness, pDoc->color);
        p->SelectObject(pen);

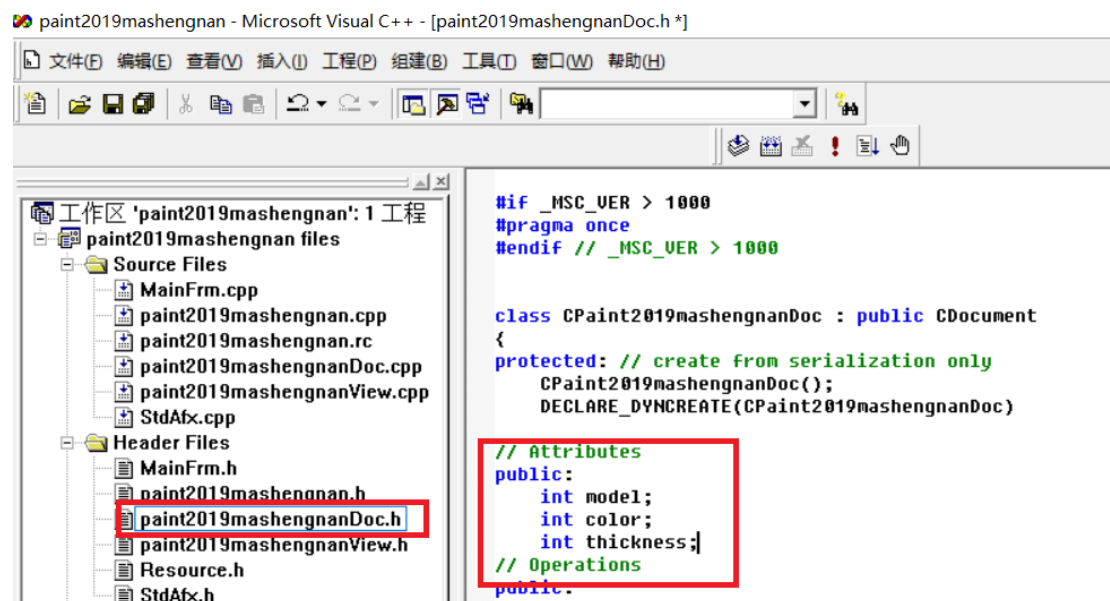
        //鼠标按下进行绘制
        if(m_click==true)
        {
            p->MoveTo(m_point);
            p->LineTo(point);
            m_point = point;
        }
    }
    CView::OnMouseMove(nFlags, point);
}

```

(11) 在 paint2019mashengnanView.h 中添加两个 public 成员变量 m\_click. 和 m\_point.



(12) 为了方便以后对于画笔的颜色和画笔的粗细进行调整，我们再看 paint2019mashengnanDoc.h 中添加两个成员变量分别是 int color 以及 int thickness. 并分别初始化为 RGB (0, 0, 0) 和 0 。



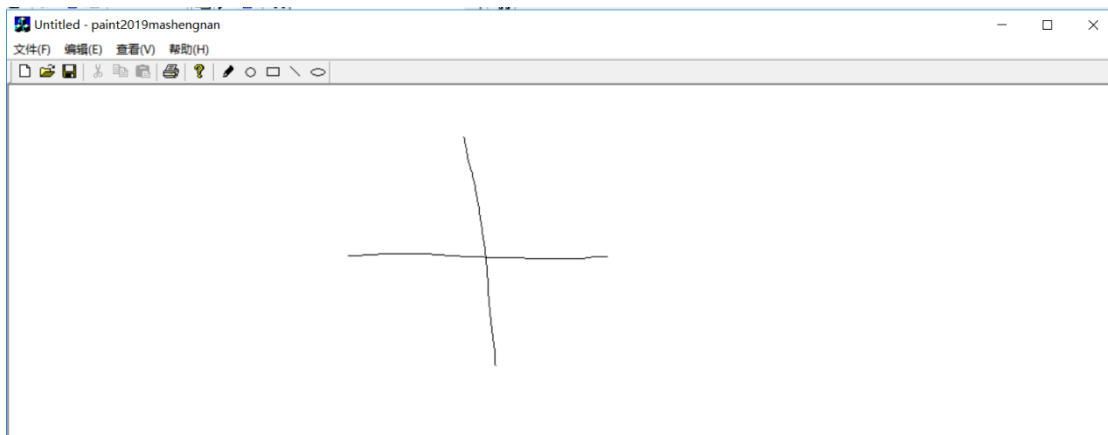
```

CPaint2019mashengnanDoc::CPaint2019mashengnanDoc()
{
    // TODO: add one-time construction code here
    model = 0;
    color = RGB(0,0,0);
    thickness = 0;
}

```

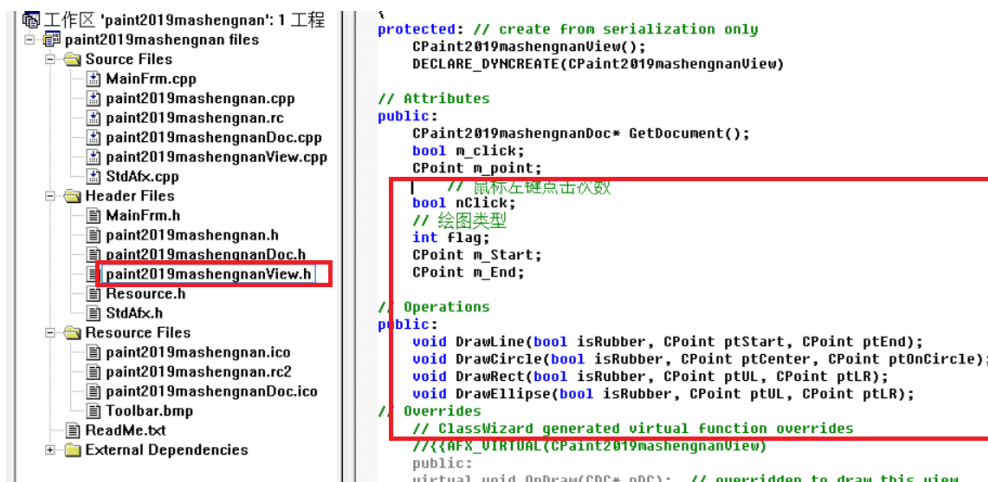


(13) 此时，我们运行一下程序，我们发现我们用黑色的画笔进行画图了。



(14) 我们将剩下的圆形、矩形、直线、椭圆分别按照这样的步骤进行编写。由于我们在画这些图形时候用到了橡皮筋效果，那么我们会多声明几个函数，分别用来画这些图形。最终的代码将一一贴出。

(15) 最终，我们会在 paint2019mashengnanView.h 中再添加如下几个声明：



(16) 然后在 paint2019mashengnanView.cpp 中的所有代码如下图所示（注意，在 cpp 最前面添加#include “math.h”）：

```
void CPaint2019mashengnanView::DrawCircle(bool isRubber, CPoint ptCenter, CPoint ptOnCircle)
{
    CDC* pDC = GetDC();
    // 半径计算
    CPaint2019mashengnanDoc *pDoc = GetDocument();
    float r1 = (float)((ptCenter.x-ptOnCircle.x)*(ptCenter.x-ptOnCircle.x) + (ptCenter.y-ptOnCircle.y)*(ptCenter.y-ptOnCircle.y));
    LONG r = sqrt(r1);
    // 设置透明画刷
    CBrush *pBrush=CBrush::FromHandle((HBRUSH)GetStockObject(NULL_BRUSH));
    CBrush *pOldBrush =pDC->SelectObject(pBrush);
    if(isRubber)
    {
        pDC->SetROP2(R2_NOT);
        pDC->MoveTo(ptCenter.x,ptCenter.y);
        pDC->LineTo(ptOnCircle.x,ptOnCircle.y);
        // 半径
        pDC->Ellipse(ptCenter.x-r,ptCenter.y-r,ptCenter.x+r,ptCenter.y+r);
    }
    else
    {
        CPen pen(PS_SOLID,pDoc->thickness,pDoc->color);
        CPen *pOldPen = pDC->SelectObject(&pen);
        pDC->Ellipse(ptCenter.x-r,ptCenter.y-r,ptCenter.x+r,ptCenter.y+r);
        pDC->SelectObject(pOldPen);
        pDC->SetROP2(R2_NOT);
        pDC->MoveTo(ptCenter.x,ptCenter.y);
        pDC->LineTo(ptOnCircle.x,ptOnCircle.y);
    }
    ReleaseDC(pDC);
}
```



```

// 绘制矩形, ptUL: 左上, ptLR: 右下
void CPaint2019mashengnanView::DrawRect(bool isRubber, CPoint ptUL, CPoint ptLR)
{
    CDC* pDC = GetDC();
    CPaint2019mashengnanDoc *pDoc = GetDocument();
    // 设置透明画刷
    CBrush *pBrush=CBrush::FromHandle((HBRUSH)GetStockObject(NULL_BRUSH));
    CBrush *pOldBrush =pDC->SelectObject(pBrush);
    if(isRubber)
    {
        pDC->SetROP2(R2_NOT);
        pDC->Rectangle(ptUL.x,ptUL.y,ptLR.x,ptLR.y);
    }
    else
    {
        CPen pen(PS_SOLID,pDoc->thickness,pDoc->color);
        CPen *pOldPen = pDC->SelectObject(&pen);
        pDC->Rectangle(ptUL.x,ptUL.y,ptLR.x,ptLR.y);
    }
    ReleaseDC(pDC);
}

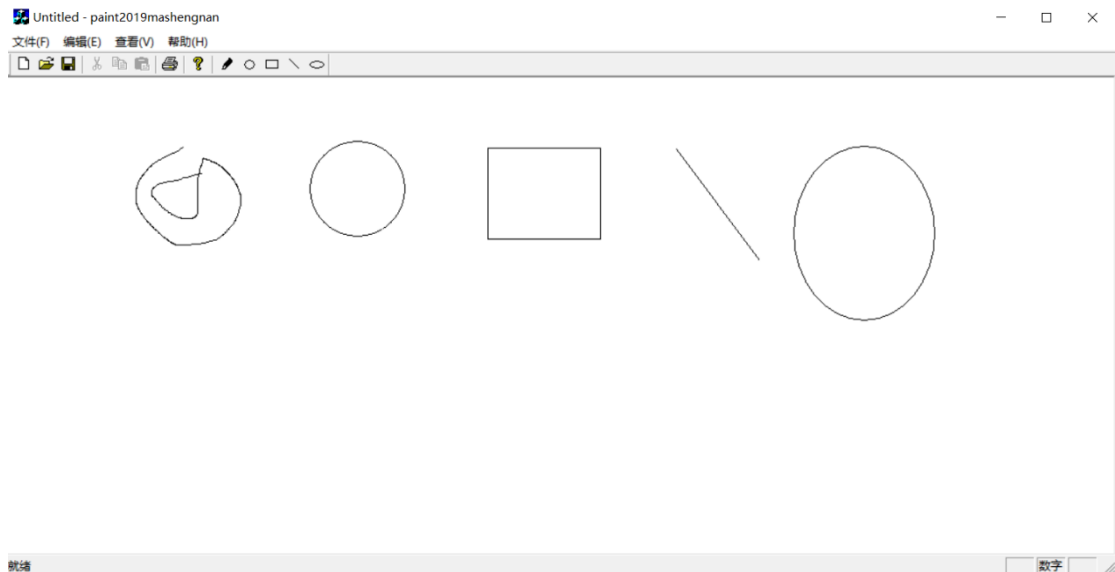
void CPaint2019mashengnanView::DrawLine(bool isRubber, CPoint ptStart, CPoint ptEnd)
{
    CDC* pDC = GetDC();
    CPaint2019mashengnanDoc *pDoc = GetDocument();
    if(isRubber)
    {
        pDC->SetROP2(R2_NOT);
    }
    CPen pen(PS_SOLID,pDoc->thickness,pDoc->color);
    CPen *pOldPen = pDC->SelectObject(&pen);
    pDC->MoveTo(ptStart.x,ptStart.y);
    pDC->LineTo(ptEnd.x,ptEnd.y);
    pDC->SelectObject(pOldPen);
    ReleaseDC(pDC);
}

void CPaint2019mashengnanView::DrawEllipse(bool isRubber, CPoint ptUL, CPoint ptLR)
{
    CDC* pDC = GetDC();
    CPaint2019mashengnanDoc *pDoc = GetDocument();
    // 设置透明画刷
    CBrush *pBrush=CBrush::FromHandle((HBRUSH)GetStockObject(NULL_BRUSH));
    CBrush *pOldBrush =pDC->SelectObject(pBrush);
    if(isRubber)
    {
        pDC->SetROP2(R2_NOT);
        pDC->Ellipse(ptUL.x,ptUL.y,ptLR.x,ptLR.y);
    }
    else
    {
        CPen pen(PS_SOLID,pDoc->thickness,pDoc->color);
        CPen *pOldPen = pDC->SelectObject(&pen);
        pDC->Ellipse(ptUL.x,ptUL.y,ptLR.x,ptLR.y);
    }
    ReleaseDC(pDC);
}

```

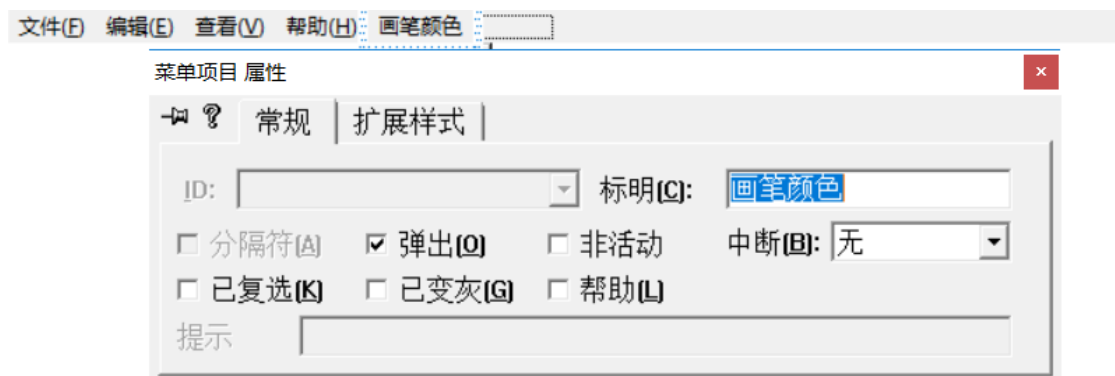
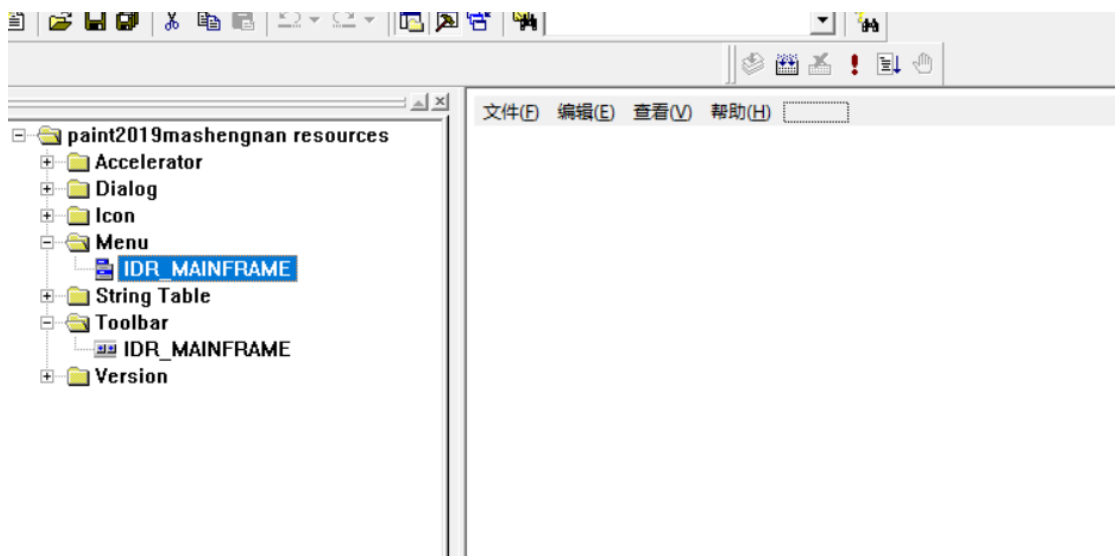
(17) 至于 LBUTTONDOWN、MOUSEMOVE 以及 LBUTTONUP 这三个函数，由于太长了，不方便放置，但是老师您有我的文件，可以在 paint2019mashengnanView.cpp 中看到具体的编码格式。

(18) 接下来我们运行这个程序，我们发现，已经可以自由的画曲线，圆形、矩形、直线、椭圆了。



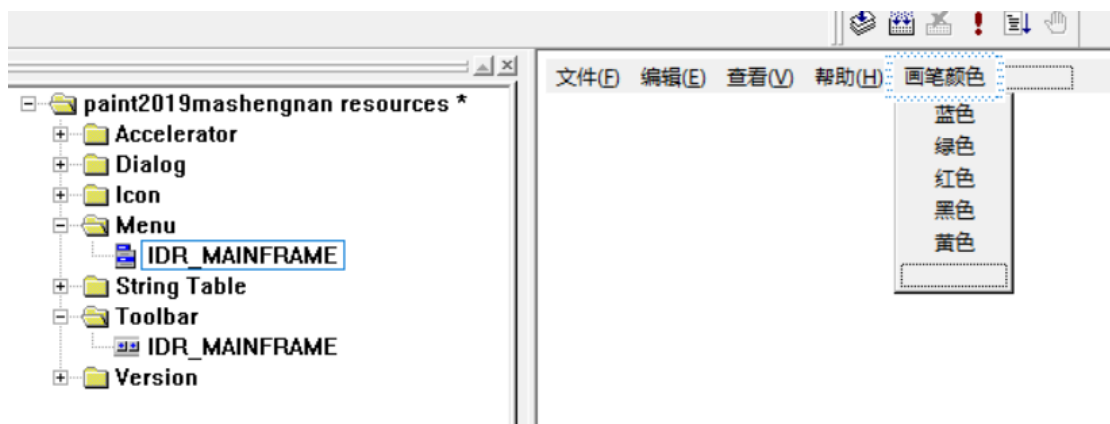
### 三、菜单栏的添加和程序的编写

(1) 点开 ResourceView，再点击 paint2019mashengnan resources 下拉菜单中的 Menu 选项，双击 IDR\_MAINFRAME。我们双击帮助后面的虚线框，会弹出相应的属性，我们将属性依次编写完毕。

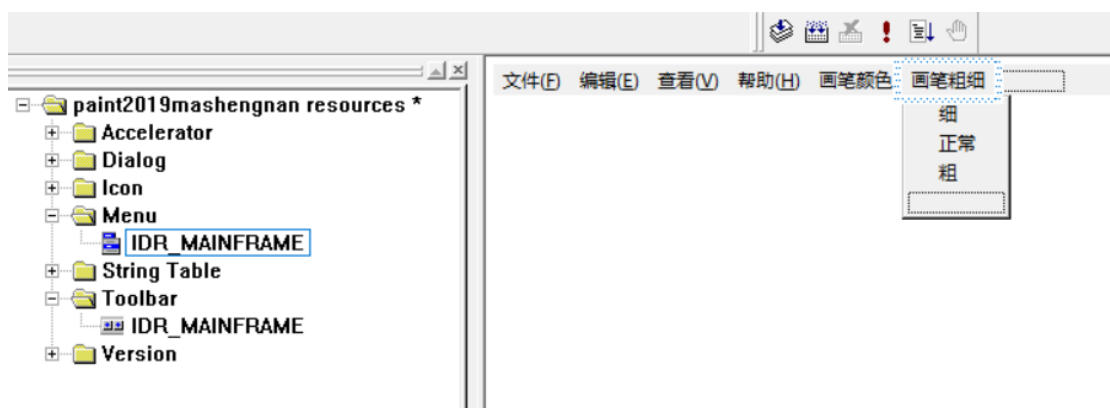




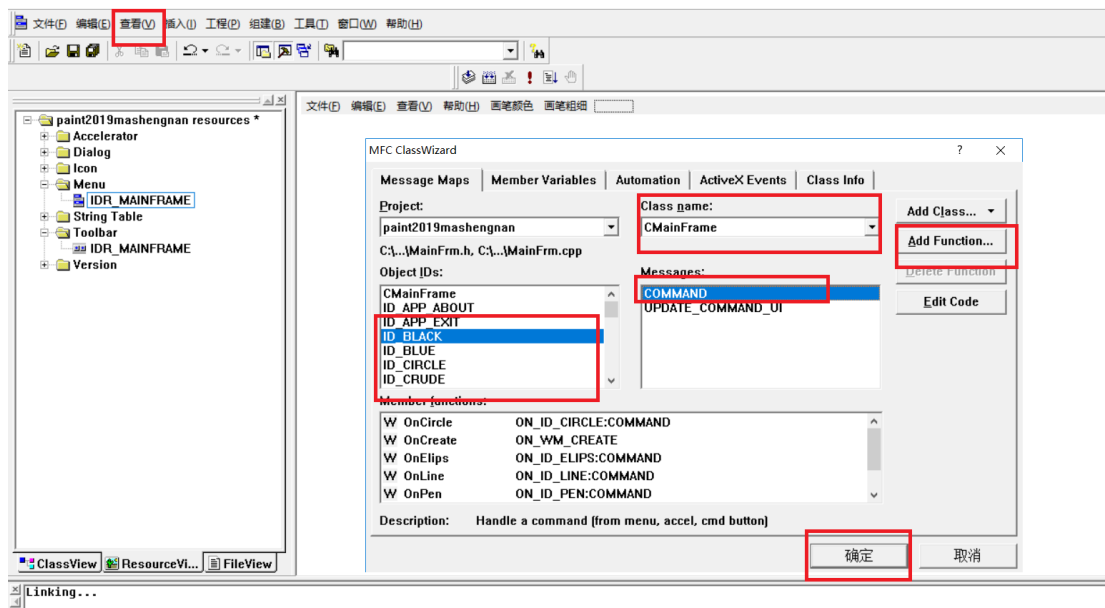
(2) 接下来我们依次命名绿色、红色、黑色、黄色，命名格式如上图，ID 分别为 ID\_GREEN, ID\_RED, ID\_BLACK, ID\_YELLOW:



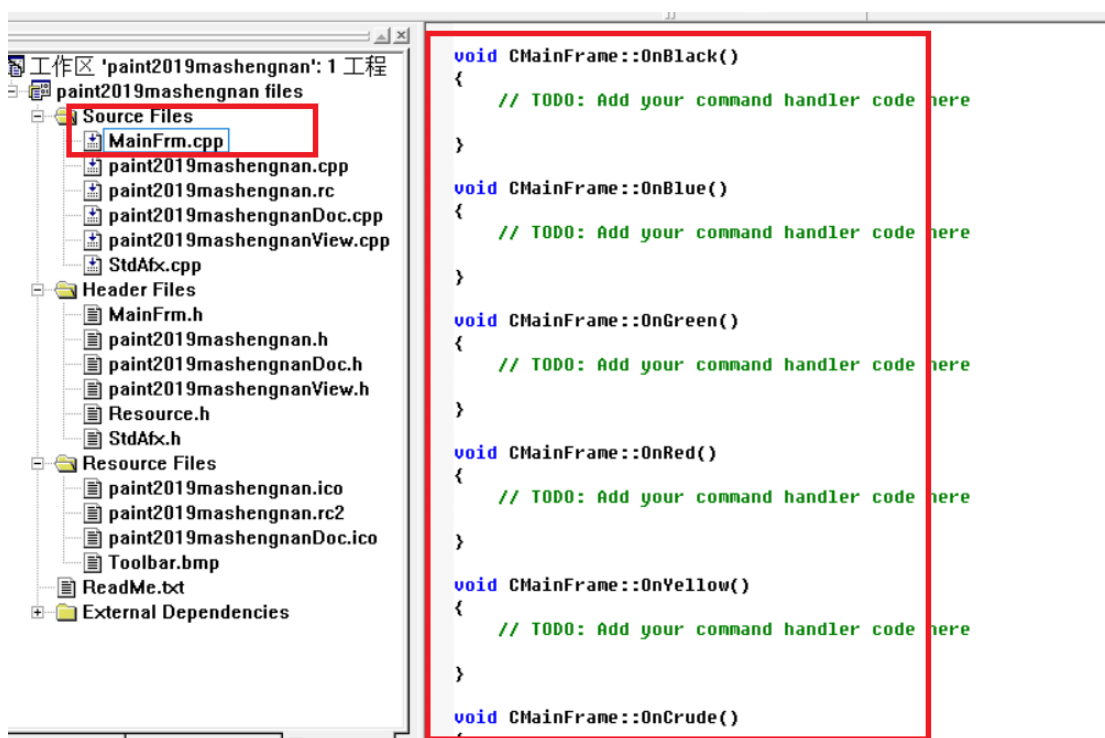
(3) 接下来我们设定画笔的粗细，分为细、正常、粗，具体方式和上面画笔颜色的方法类似 ID\_THIN, ID\_NORMAL, ID\_CRUBE。



(4) 接下来，分别建立这些颜色和粗细按钮的类向导。点击查看->类向导，选择 CMainFrame, 分别选择刚刚建立的选项的 ID, 点击 COMMAND, 点击 ADD FUNCTION, 点击确定。最后，我们应该添加了 8 个 COMMAND。



(5) 我们打开 MainFrm.cpp，我们应该可以在文件的最后找到着 8 个函数。如下图：



(6) 我们分别为这些函数添加上自己的代码：

```

void CMainFrame::OnPen()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->model = 0;
}

void CMainFrame::OnCircle()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->model = 1;
}

void CMainFrame::OnTangle()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->model = 2;
}

void CMainFrame::OnLine()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->model = 3;
}

void CMainFrame::OnElips()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->model = 4;
}

```

```

void CMainFrame::OnElips()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->model = 4;
}

void CMainFrame::OnBlack()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->color = RGB(0,0,0);
}

void CMainFrame::OnBlue()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->color = RGB(0,0,255);
}

void CMainFrame::OnGreen()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->color = RGB(0,255,0);
}

void CMainFrame::OnRed()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->color = RGB(255,0,0);
}

```

```

}

void CMainFrame::OnYellow()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->color = RGB(255,255,0);
}

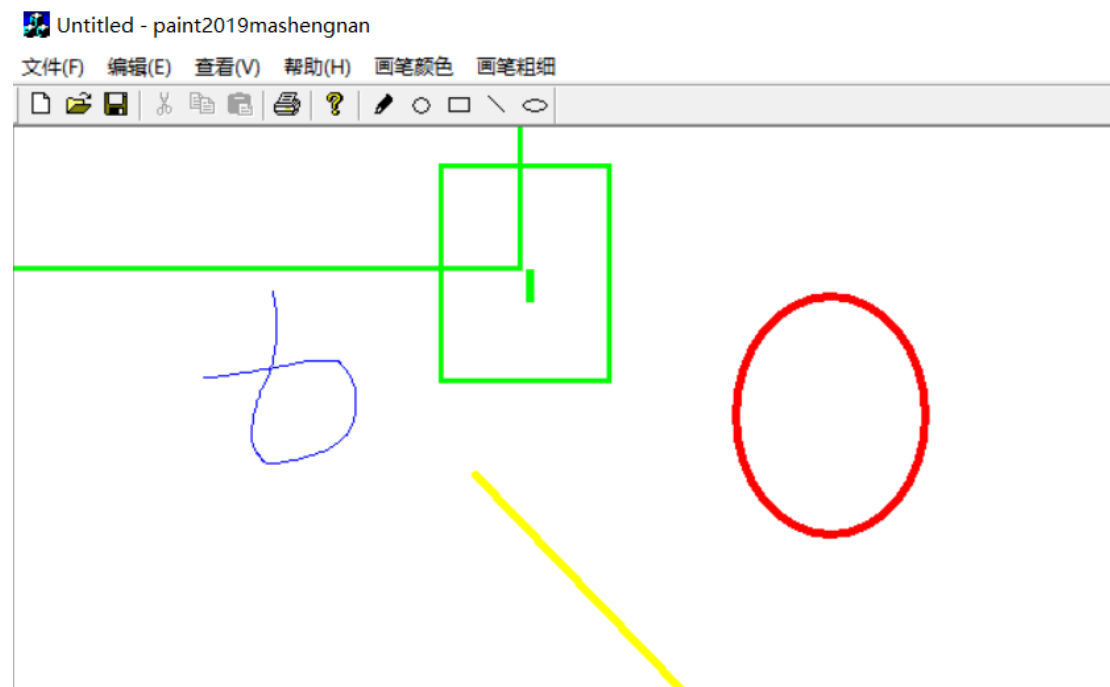
void CMainFrame::OnCrude()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->thickness = 5;
}

void CMainFrame::OnNormal()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->thickness = 3;
}

void CMainFrame::OnThin()
{
    // TODO: Add your command handler code here
    CPaint2019mashengnanDoc* pDoc = (CPaint2019mashengnanDoc*) GetActiveDocument();
    pDoc->thickness = 1;
}
}

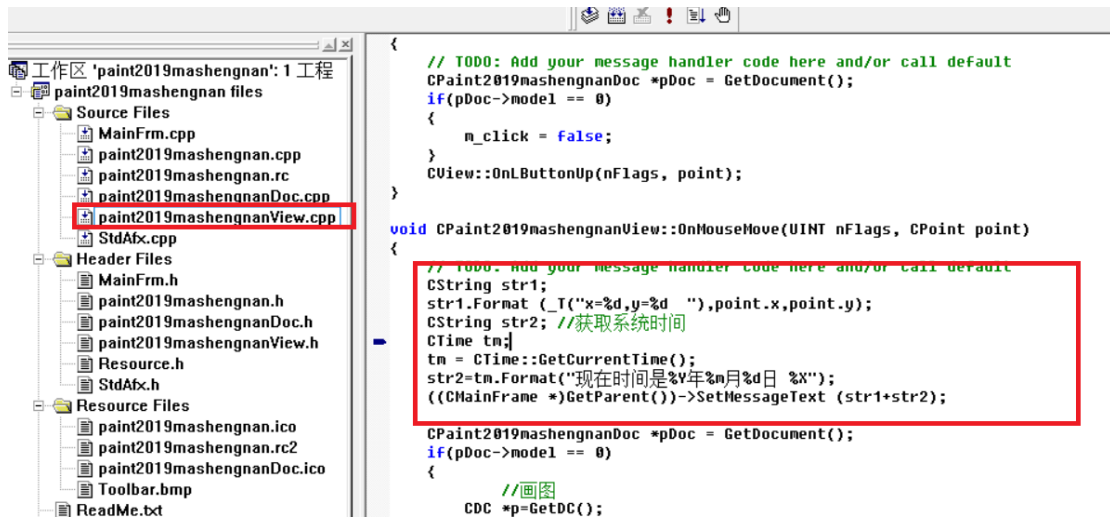
```

(7) 运行程序，我们发现，我们已经可以通过菜单栏选择来选择画笔的颜色和粗细了。



## 四、坐标系的添加和状态栏鼠标位置和时间显示的程序编写

(1) 为了显示鼠标的是实时位置，我们在 paint2019mashengnanView.cpp 中的 MOUSEMOVE 函数的开头添加如下的代码：



(2) 注意要添加头文件#include “MainFrm.h”，此时我们运行程序，可以发现，这时候的状态栏已经可以显示当前的鼠标位置和时间了。但是在画图框之外，他会自动显示就绪，我们修改这句话。

(3) 点开 ResourceView，再点击 paint2019mashengnan resources 下拉菜单中的 String Table 选项，双击字符串表。我们双击 AFX\_IDS\_IDLEMESSAGE，会弹出相应的属性，我们将提示依次编写完毕。



(4) 此时，我们运行程序得到的结果如下图：

x=334,y=174 现在是2019年11月17日 09:52:41

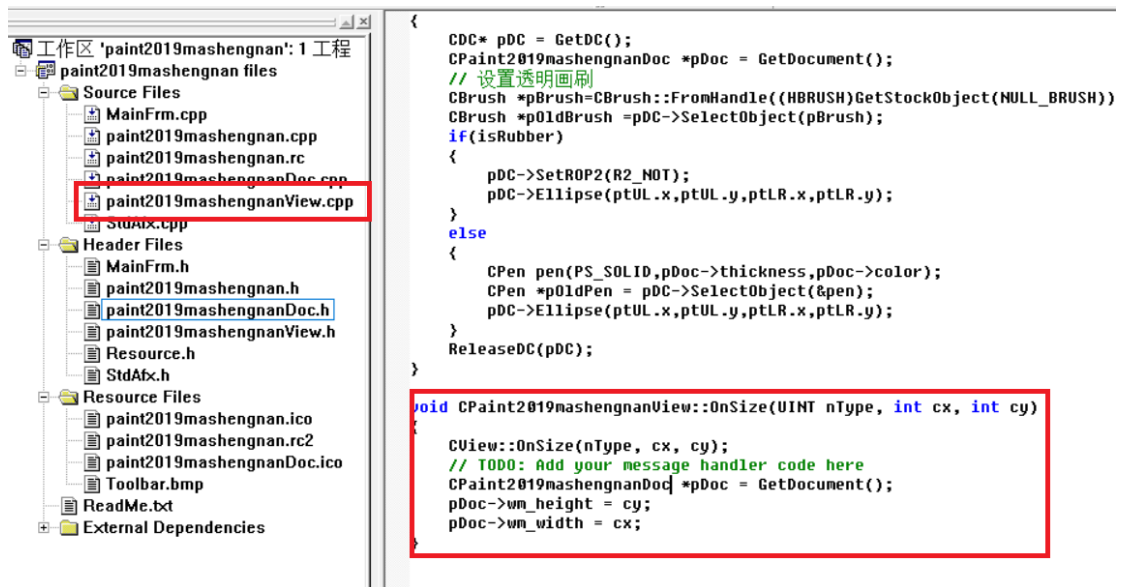
(5) 如果我们的鼠标不在画图区域内部，那么显示如下：

鼠标在画图区域外部，无法显示当前鼠标位置

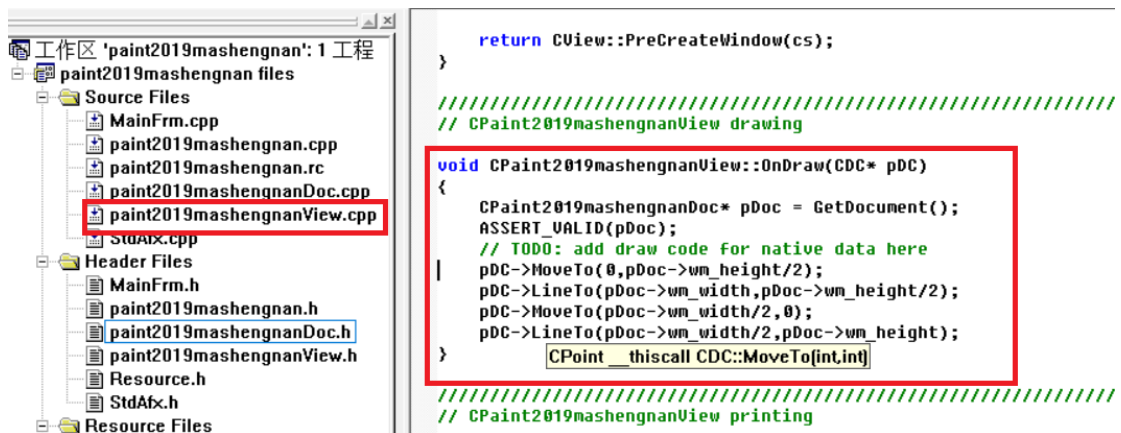


The screenshot shows the Visual Studio IDE with the 'Solution Explorer' on the left and the 'Code' window on the right. The 'Solution Explorer' displays the project structure for 'paint2019mashengnan'. The 'Source Files' folder contains: MainFrm.cpp, paint2019mashengnan.cpp, paint2019mashengnan.rc, paint2019mashengnanDoc.cpp, paint2019mashengnanView.cpp, and StdAfx.cpp. The 'Header Files' folder contains: MainFrm.h, paint2019mashengnan.h, paint2019mashengnanDoc.h (highlighted with a red box), paint2019mashengnanview.h, Resource.h, and StdAfx.h. The 'Code' window shows the content of paint2019mashengnanDoc.h, which includes a class declaration for CPaint2019mashengnanDoc, a public attribute section with variables model, color, thickness, wm\_width, and wm\_height (the latter two are highlighted with a red box), and a public override section for OnNewDocument.

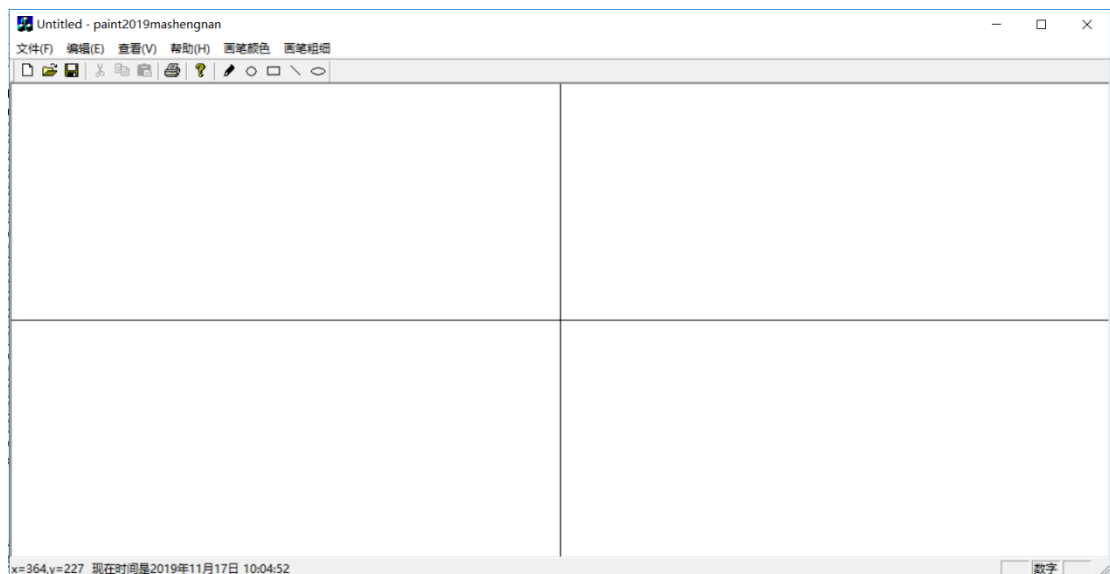
(8) 我们在 OnSize 函数中添加如下代码:



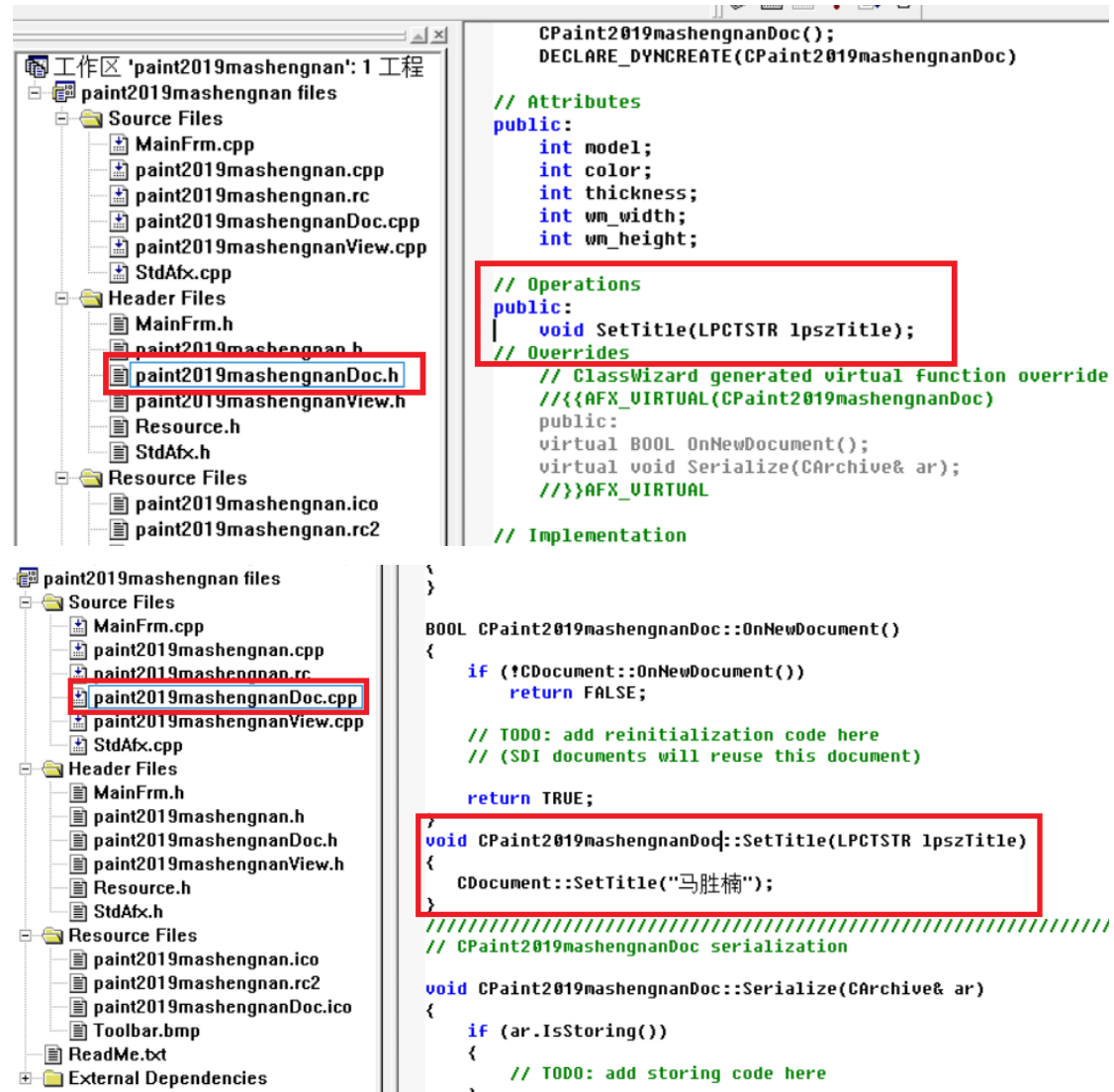
(9) 接着，再在 paint2019mashengnanView.cpp 中找到 OnDraw 函数，在该函数内添加如下代码：



(10) 这样我们运行的结果中，画图界面就会添加了相应的坐标系



(11) 此时我们不妨再关注一下细节，我们发现这个运行窗口的名称为 Untitled—paint2019mashengnan，者应看起来未免有些尴尬，我们将其修改一下，首先，修改 Untitled。首先我们打开 paint2019mashengnanDoc.h，添加一个成员函数 SetTitle，然后在 paint2019mashengnanDoc.cpp 补充相应的函数，如下图：

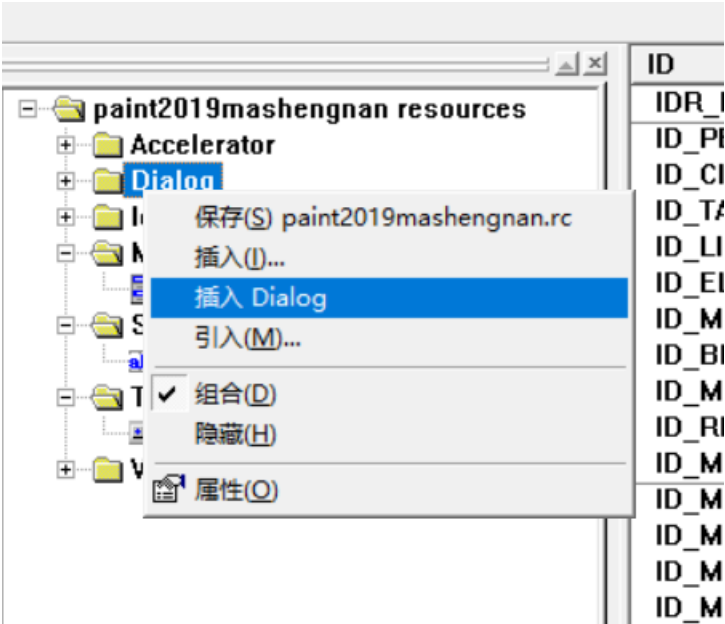


(12) 接着修改 paint2019mashengnan，点开 ResourceView，再点击 paint2019mashengnan resources 下拉菜单中的 String Table 选项，双击字符串表。我们双击 IDR\_MAINFRAME，会弹出相应的属性，我们将标题填写为画图软件，这样我们再运行程序，就会发现标题已经改好了

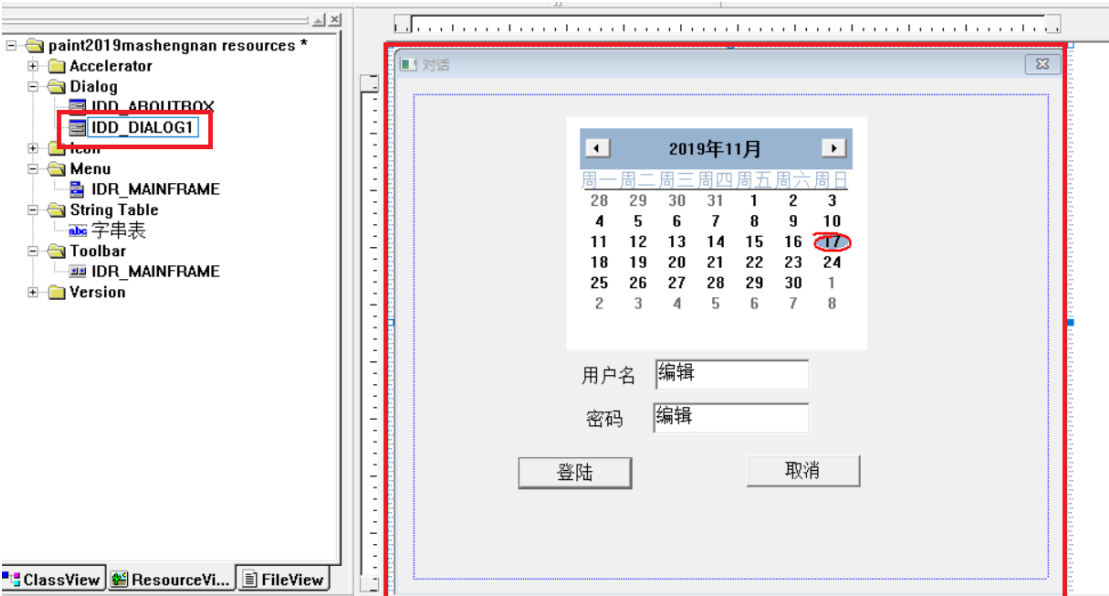


## 五、登陆界面的程序编写

(1) 根据上面的步骤，我们已经将基本的画图软件完成了，那么接下来就是建立一个登陆界面。首先我们点开 ResourceView，右键 dialog，点击插入 Dialog



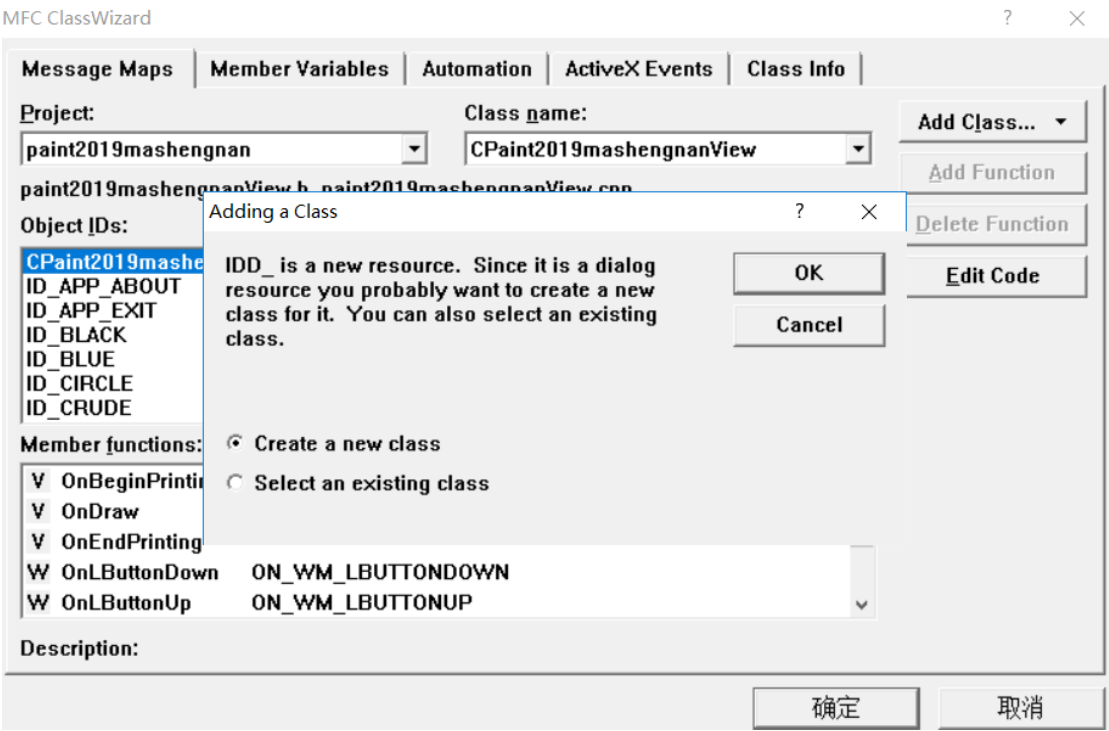
(2) 我们点击这个新建的 DIALOG，将界面设计成如下图所示，分别有一个日历，两个静态文本框，两个编辑框，两个按钮，在密码一行对应的编辑框中，右键->属性->样式中，将密码一栏勾选上，然后分别将这两个编辑框的 ID 改为 IDC\_USERNAME 和 IDC\_PASSWORD。（一定要修改）



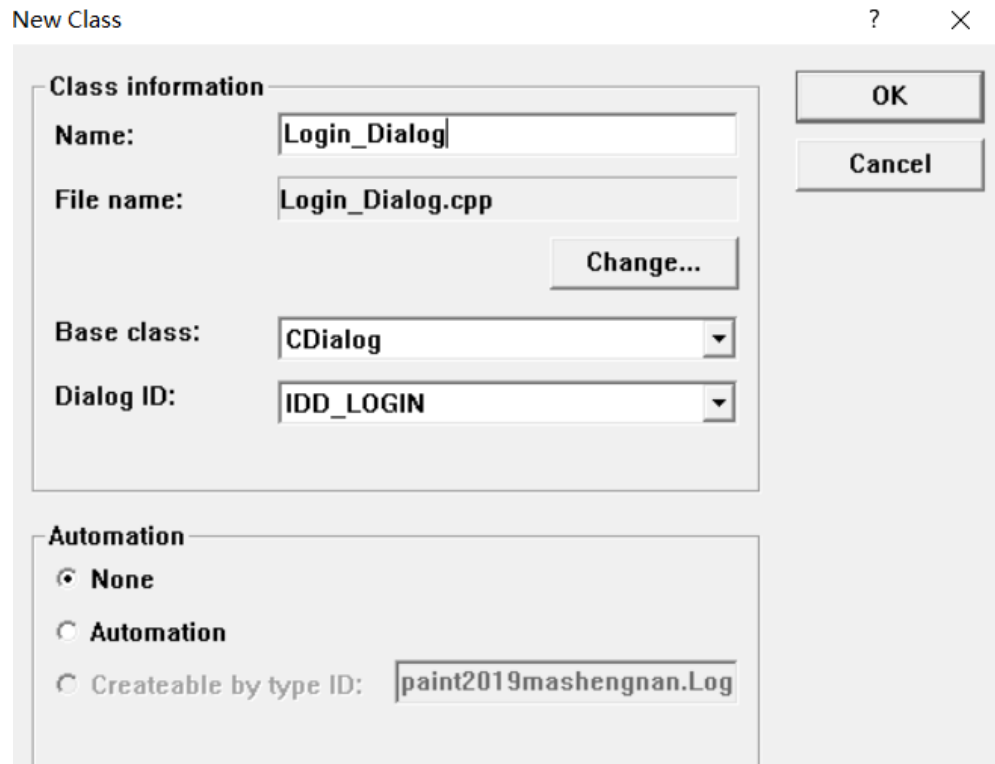
(3) 然后我们右键对话框的空白处，点击属性，将属性设置如下后推出属性设置：



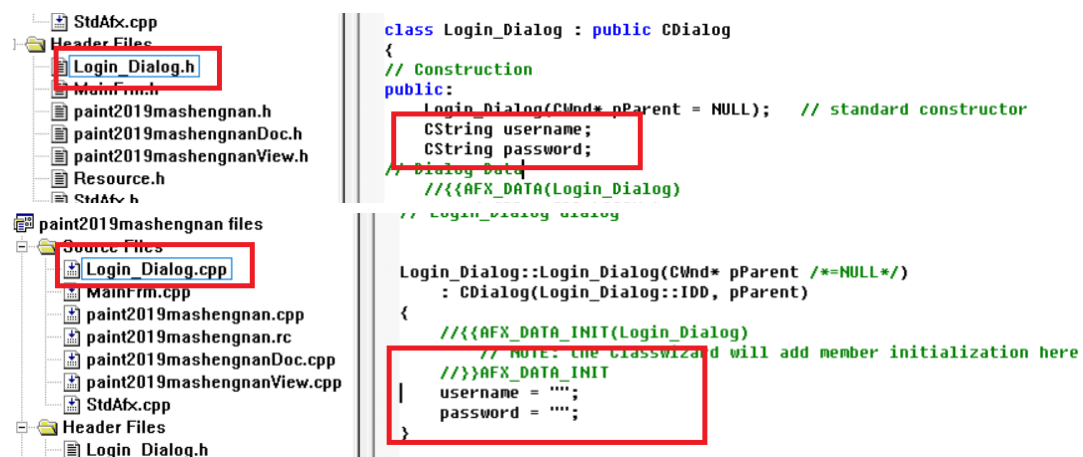
(4) 此时，双击对话框空白位置（一定要双击），会弹出如下框图：



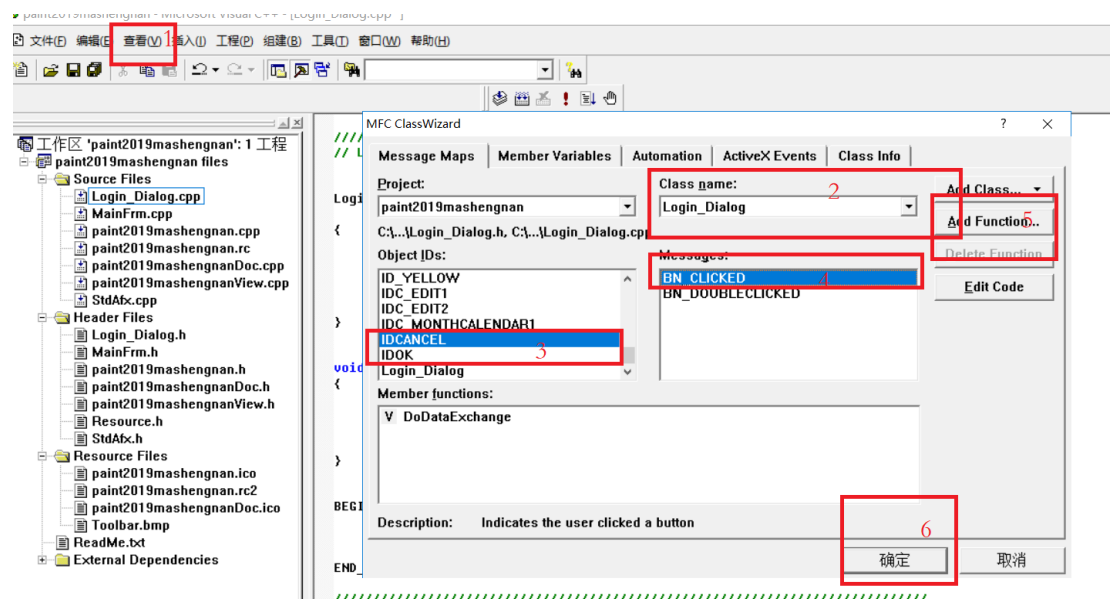
(5) 选择 create a new class，点击 OK。然后在弹出界面新建一个类，类名为 Login\_Dialog, Base Class 一栏一定一定要选择 Dialog，否则将会弹出失败，最后点击 OK 即可。之后系统会自动生成一个 Login\_Dialog.h 和 Login\_Dialog.cpp。



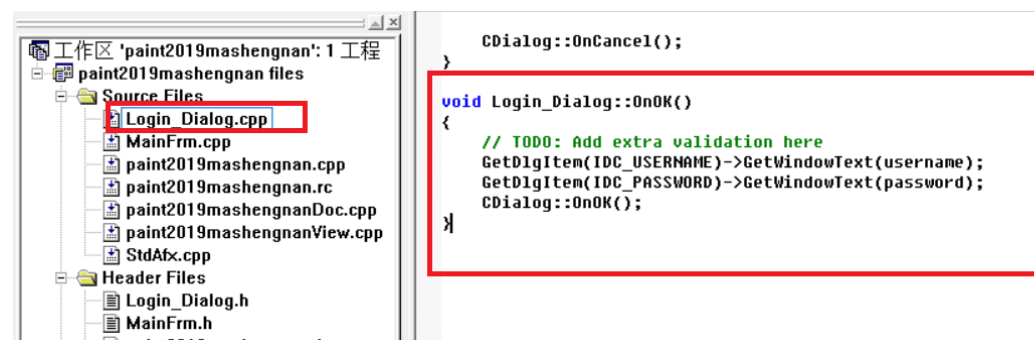
(6) 我们点击 Login\_Dialog.h，添加两个共有的成员变量 CString username 和 CString password，并赋初值为空。



(7) 随后我们建立类向导，按照图中步骤，分别建立 IDCANCEL 和 IDOK 的关于 ON\_CLICKED 的函数，随后我们会在 Login\_Dialog.cpp 的最后几行看到这个函数。



(8) 我们在函数中添加如下代码：注意一定要修改两个编辑框的 ID 否则会出错



(9) 然后，我们打开 paint2019mashengnan.cpp, 找到 InitInstance() 函数在前面函数最前面添加如下代码，注意，要在前面添加#include “Login\_Dialog.h”：

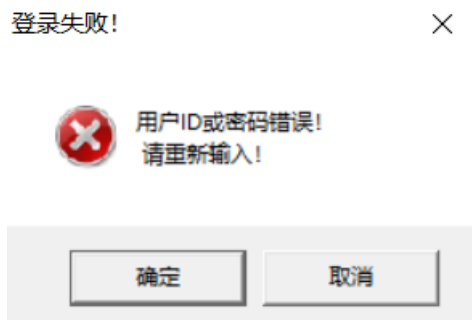


(10) 接着我们运行程序，就会发现，程序就会自动先跳出登陆界面，成功后再进入画图界面。用户名 mashengnan，密码 123456

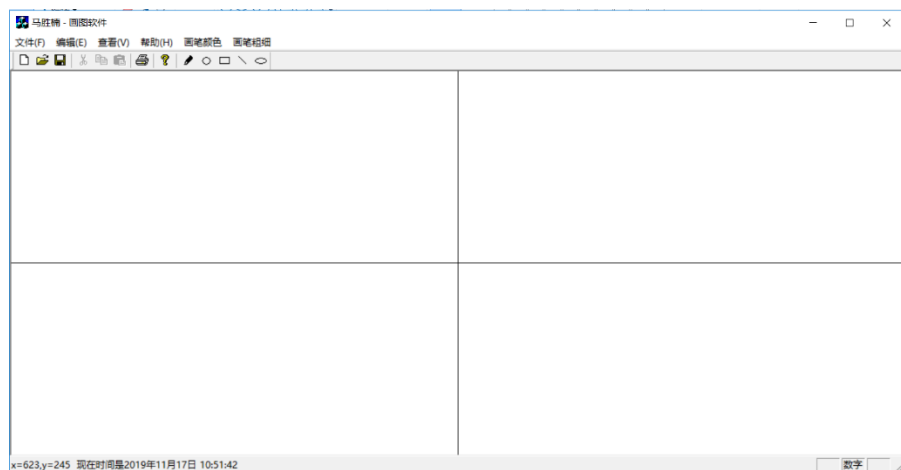




(11) 如果密码错误会提示如下框图：



(12) 如果登陆密码正确，进入画图软件：



(13) 至此，所有程序全部完成。

## 五、心得体会

通过此次编程设计，对于 MFC 中 Doc, Mainfrm, View 等类的相互作用和关系有了更直接的认识，对于各种按钮的交互以及类向导的创建有了更熟悉的操作，尤其是对于画圆形、矩形、椭圆形之类的时候所应用到的橡皮筋效果有了更深层次的认识，对于多个 Dialog 之间的交互关系也有了更明确的认识和理解，不仅如此，我对于 Visual C++MFC 的功能有了更深刻的了解，在实际编程的过程中，也加深了对 C++理论知识的理解，学到了一些实用的功能。在学习过程中，一定要坚持理论与实际相结合。在实际操作中，既需要理论知识的引导也能加深对理论知识的理解。