课程编号： B080206030

MFC 可视化程序设计
实验报告

姓　　名	薛旗	学　　号	20155362
班　　级	软信-1503	指 导 教 师	马毅
实 验 名 称	MFC 可视化程序设计		
开 设 学 期	2016-2017 第 一 学 期		
开 设 时 间	第 10 周——第 16 周		
报 告 日 期	2017.01.01		
评 定 成 绩		评　　定　　人	马毅
		评 定 日 期	

东北大学软件学院

实验内容：

(1)模态对话框
(2)无模态对话框
(3)文件对话框
(4)消息对话框
(5)对话框类与控件交换
(6)动态创建按钮
(7)单项选择/多项选择
(8)滚动条/滑块
(9)属性表单
(10)命令响应菜单
(11)标记菜单
(12)缺省菜单
(13)图形菜单
(14)新的菜单
(15)弹出式菜单
(16)工具栏图标

实验步骤：

(1)模态对话框：

a. 步骤：创建工程-插入对话框-添加类-为对话框添加成员函数-编辑菜单项-为子菜单添加事件处理程序

b. 关键代码：

```
void CMainFrame::OnModalDlg()
{
    // TODO: 在此添加命令处理程序代码
    CFirstDlg dlg;
    dlg.DoModal();
}
```

(2)无模态对话框：

a. 步骤：插入对话框-添加类-添加函数-编辑菜单项-为子菜单添加事件处理程序

b. 关键代码：

```
void CSecondDlg::OnBnClickedOk()
{
    // TODO: 在此添加控件通知处理程序代码
    //CDialogEx::OnOK();
    DestroyWindow();
    delete this;
}
```

```
void CSecondDlg::OnBnClickedCancel()
{
    // TODO: 在此添加控件通知处理程序代码
    //CDialogEx::OnCancel();
    DestroyWindow();
    delete this;
}
```

(3)文件对话框

a.步骤：添加子菜单-为子菜单添加事件处理程序

b.关键代码：

```
void CMainFrame::OnFileDlg()
{
    // TODO: 在此添加命令处理程序代码
    /*
    CString filter;
    filter = "文本文件(*.txt)|*.txt|C++文件(*.h,*.cpp)|*.h;*.cpp||";
    CFileDialog dlg(TRUE, NULL, NULL, OFN_HIDEREADONLY, filter);
    if (dlg.DoModal() == IDOK)
    {
        CString str;
        str = dlg.GetPathName();
        AfxMessageBox(str);
    }
    */
    CString filter;
    filter = "文本文件(*.txt)|*.txt|C++文件(*.h,*.cpp)|*.h;*.cpp||";
    CFileDialog dlg(TRUE, NULL, NULL, OFN_HIDEREADONLY | OFN_ALLOWMULTISELECT, filter);
    if (dlg.DoModal() == IDOK)
    {
        CString str;
        POSITION pos;
        pos = dlg.GetStartPosition();
        while (pos != NULL)
        {
            str = dlg.GetNextPathName(pos);
            AfxMessageBox(str);
        }
    }
}
```

(4)消息对话框

a.步骤：添加子菜单-为子菜单添加事件处理程序

b.关键代码：

```
void CMainFrame::OnMessageDlg()
{
    // TODO: 在此添加命令处理程序代码
    MessageBox(_T("你幸福吗？"), NULL, MB_YESNO);
}
```

(5)对话框类与控件交换

a.步骤：在第一个对话框上添加 Button 和 Edit control-为 button 和 Edit Control 添加变量-为 button1 添加函数

b.关键代码：

```
void CFirstDlg::OnBnClickedButton1()
{
    // TODO: 在此添加控件通知处理程序代码
    UpdateData(TRUE);
    m_RelBtn.SetWindowText(m_strEdit);
}
```

(6)动态创建按钮

a.步骤：插入对话框-添加类-添加变量-编辑代码-添加子菜单-为子菜单添加事件处理程序

b.关键代码：

```
void CDlgBtnCreate::OnOK()
{
    // TODO: 在此添加专用代码和/或调用基类
    if (m_btnMine.m_hWnd == NULL)
    {
        m_btnMine.Create(L"动态创建的按钮", WS_CHILD | WS_VISIBLE | BS_PUSHBUTTON,
            CRect(20, 20, 300, 100), this, 2000);
        GetDlgItem(IDOK)->SetWindowText(L"删除动态创建的按钮");
    }
    else
    {
        m_btnMine.DestroyWindow();
        GetDlgItem(IDOK)->SetWindowText(L"运行时创建按钮");
    }
    //CDialogEx::OnOK();
}
```

(7)单项选择/多项选择

a.步骤：插入对话框-添加类-添加八个 radio buttons-添加四个 check boxes-添加 WM_INITDIALOG message 并编辑代码-添加子菜单-为子菜单添加事件处理程序

b.关键代码：

```
BOOL CDlgNetQuestionary::OnInitDialog()
```

```
{
    CDialogEx::OnInitDialog();

    // TODO:  在此添加额外的初始化
    CheckRadioButton(IDC_AGE_L18, IDC_AGE_M38, IDC_AGE_18T27);
    CheckRadioButton(IDC_CM_FTTL, IDC_CM_OTHER, IDC_CM_FTTL);
    CButton* pBtn = (CButton*)GetDlgItem(IDC_DO_POP);
    pBtn->SetCheck(1);
    return TRUE;  // return TRUE unless you set the focus to a control
                  // 异常: OCX 属性页应返回 FALSE
}
```

(8)滚动条/滑块

a.步骤：插入对话框-添加类-添加 Scroll bar 和 Slider 并分别为其添加变量-为 CBkColorDlg 添加两个私有成员-添加消息映射并编辑代码-添加子菜单-为子菜单添加事件处理程序

b.关键代码：

```
BOOL CBkColorDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // TODO:  在此添加额外的初始化
    m_scrollRed.SetScrollRange(0, 255);
    m_sliderBlue.SetRange(0, 255);
    m_sliderGreen.SetRange(0, 255);
    m_nBlue = m_nGreen = m_nRedValue = 192;
    UpdateData(FALSE);
    m_scrollRed.SetScrollPos(m_nRedValue);
    return TRUE;  // return TRUE unless you set the focus to a control
                  // 异常: OCX 属性页应返回 FALSE
}
```

(9)属性表单

a.步骤：插入对话框-添加类-添加控件-添加 OnWizardNext message 和 OnSetActive message 并编辑代码-插入对话框-添加类-添加控件，设置变量-添加 OnWizardFinish message 和 OnSetActive message 并编辑代码-添加类，基类为 CPropertySheet-编辑代码-添加子菜单-为子菜单添加事件处理程序

b.关键代码

```
void CMainFrame::OnCtlPropSheet()
{
    // TODO: 在此添加命令处理程序代码
    CPropSheetMine propSheetMine(L"工作意向表单");
    propSheetMine.SetWizardMode();
    if (ID_WIZFINISH != propSheetMine.DoModal())
```

```
        return;
    CString strTemp;
    strTemp = L"你的职业：";
    switch (propSheetMine.m_propPage1.m_nOccupation)
    {
    case 0:
        strTemp += L"程序员";
        break;
    case 1:
        strTemp += L"系统工程师";
        break;
    case 2:
        strTemp += L"项目经理";
        break;
    default:
        break;
    }
    strTemp += L"\n你的兴趣爱好：  ";
    if (propSheetMine.m_propPage2.m_bFootball)
    {
        strTemp += L"足球 ";
    }
    if (propSheetMine.m_propPage2.m_bBasketball)
    {
        strTemp += L"篮球 ";
    }
    if (propSheetMine.m_propPage2.m_bVolleyball)
    {
        strTemp += L"排球 ";
    }
    if (propSheetMine.m_propPage2.m_bSwim)
    {
        strTemp += L"游泳 ";
    }
    MessageBox(strTemp);
}
```

(10)命令响应菜单

a.步骤：添加菜单-添加子菜单-为子菜单添加事件处理程序

b.关键代码：

```
MessageBox(L"按下了命令响应菜单！");
```

(11)标记菜单

a.步骤：添加子菜单-添加成员并初始化-添加 COMMAND message 并编辑代码-添加 UPDATE_COMMAND_UI message 并编辑代码

b.关键代码：

```
void CMainFrame::OnUpdate32780(CCmdUI *pCmdUI)
{
    // TODO: 在此添加命令更新用户界面处理程序代码
    pCmdUI->Enable();
    if (m_bTag)
        pCmdUI->SetCheck();
    else
        pCmdUI->SetCheck(0);

}
```

(12)缺省菜单

a.步骤：添加子菜单-设置缺省菜单-添加 UPDATE_COMMAND_UI 消息并编辑

b.关键代码：

```
GetMenu()->GetSubMenu()->SetDefaultItem(IDM_MENU_DEFAULT);
```

(13)图形菜单

a.步骤：添加子菜单-添加数据成员-加载图片-添加 UPDATE_COMMAND_UI 消息并编辑

b.关键代码：

```
m_bitmap.LoadBitmap(IDB_BITMAP1);
    GetMenu()->GetSubMenu(5)
        ->SetMenuItemBitmaps(3, MF_BYPOSITION, &m_bitmap, &m_bitmap);
```

(14)新的菜单

a.步骤：插入新菜单-创建一个新的菜单项-添加数据成员-分别为新菜单项和加载菜单项添加 COMMAND 消息映射

b.关键代码：

```
void CMainFrame::OnMenuReturn()
{
    // TODO: 在此添加命令处理程序代码
    if (m_menuMine.m_hMenu)
        m_menuMine.DestroyMenu();
    m_menuMine.LoadMenu(IDR_MAINFRAME);

    SetMenu(&m_menuMine);
    GetMenu()->GetSubMenu(5)->SetDefaultItem(IDM_MENU_DEFAULT);
    GetMenu()->GetSubMenu(5)->SetMenuItemBitmaps
    (3, MF_BYPOSITION, &m_bitmap, &m_bitmap);

}
```

(15)弹出式菜单

a.步骤：添加 WM_CONTEXTMENU message-编辑代码

b.关键代码：

```
void CMainFrame::OnContextMenu(CWnd* pWnd, CPoint point)
{
    // TODO: 在此处添加消息处理程序代码
    CMenu* pSysMenu = GetMenu();
    int nCount = pSysMenu->GetMenuItemCount();
    if (nCount>1)
    {
        pSysMenu->GetSubMenu(nCount - 1)
            ->TrackPopupMenu(TPM_LEFTALIGN | TPM_RIGHTBUTTON,point.x, point.y, this);
    }
}
```

(16)工具栏图标

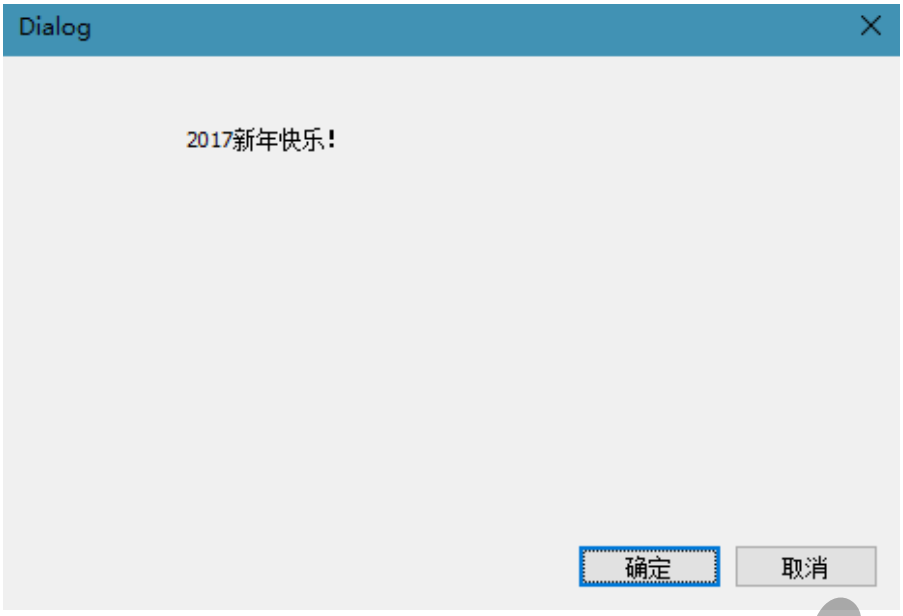
a.步骤：打开 IDR_MAINFRAME-添加两个 toolbar 元素-加载 Toolbar.bmp-修改对应的 ID-新建一个 toolbar-添加三个元素-加载 mainfram.bmp-修改 ID-添加成员并初始化-编辑代码

b。关键代码：

```
if (!m_wndToolBarMine.CreateEx(this, TBSTYLE_FLAT, WS_CHILD
        | WS_VISIBLE | CBRS_TOP
        | CBRS_GRIPPER | CBRS_TOOLTIPS
        | CBRS_FLYBY | CBRS_SIZE_DYNAMIC) ||
        !m_wndToolBarMine.LoadToolBar(IDR_TOOLBAR_NEW))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;      // fail to create
    }
    m_wndToolBarMine.EnableDocking(CBRS_ALIGN_ANY);
    //DockControlBar(&m_wndToolBarMine);
    ShowControlBar(&m_wndToolBarMine, FALSE, FALSE);
```

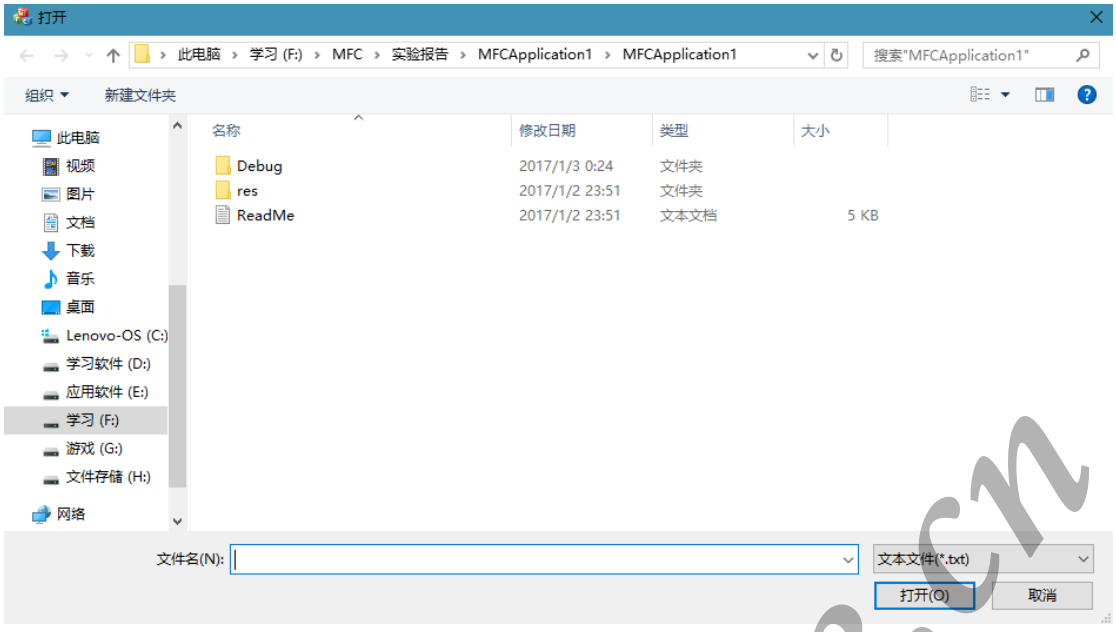
实验结果：

(1)模态对话框



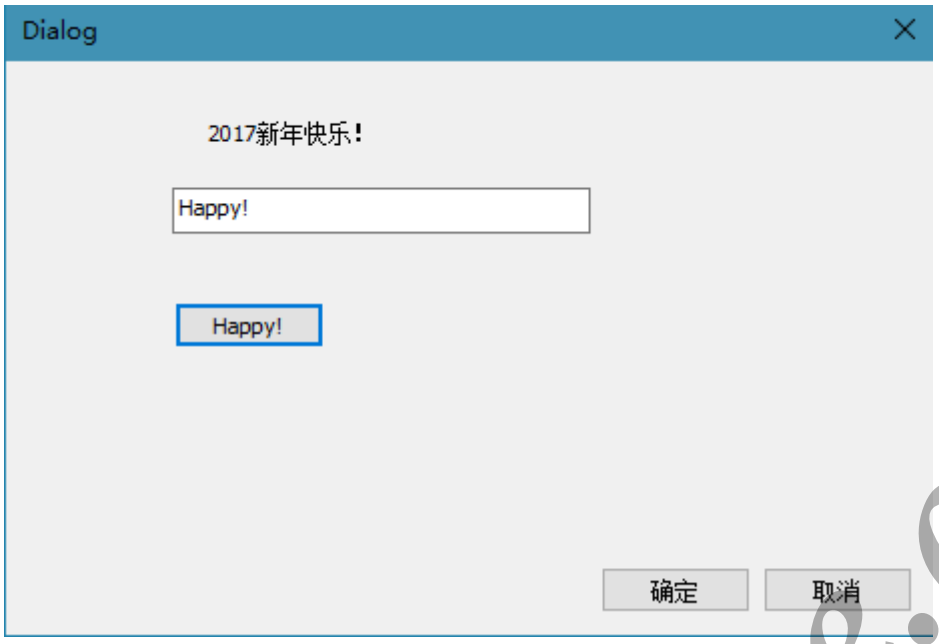
(2)无模态对话框

(3)文件对话框

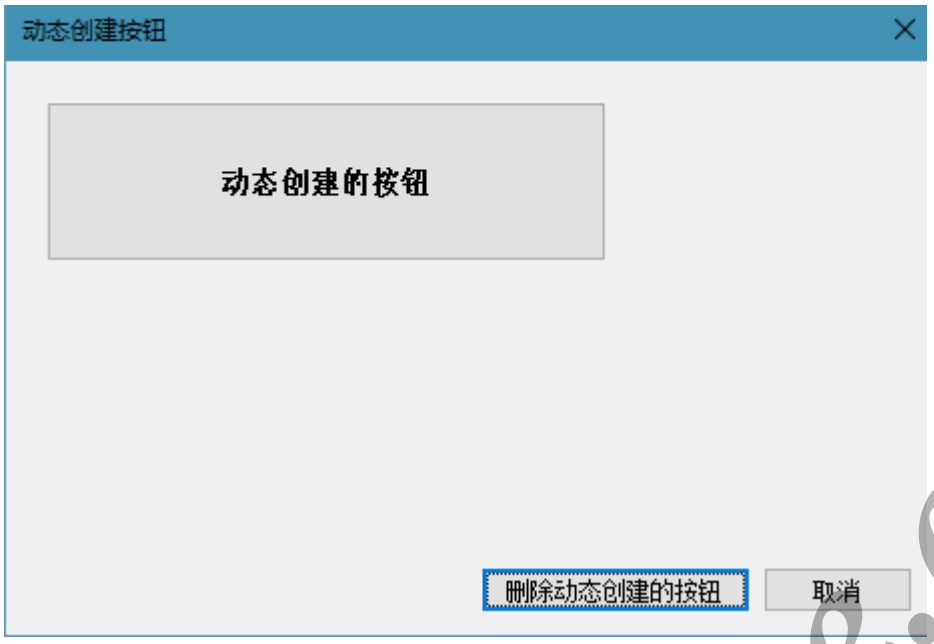


(4)消息对话框

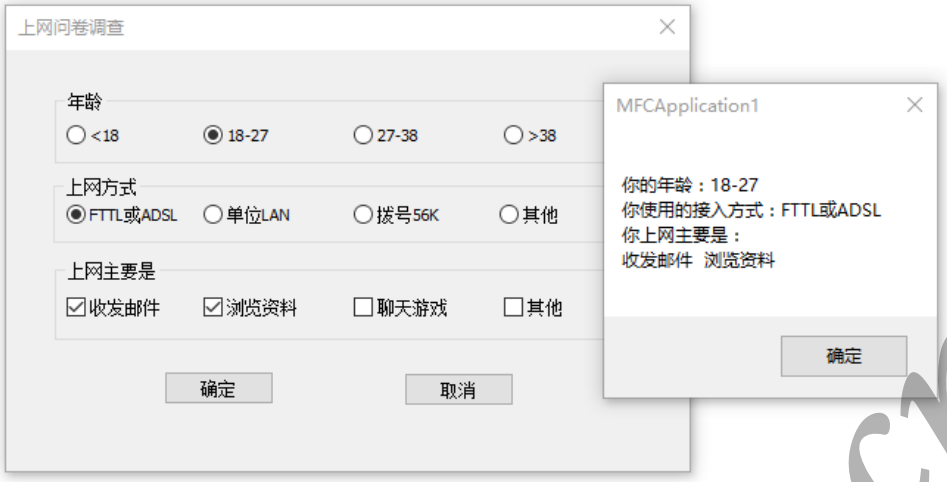
(5)对话框类与控件交换



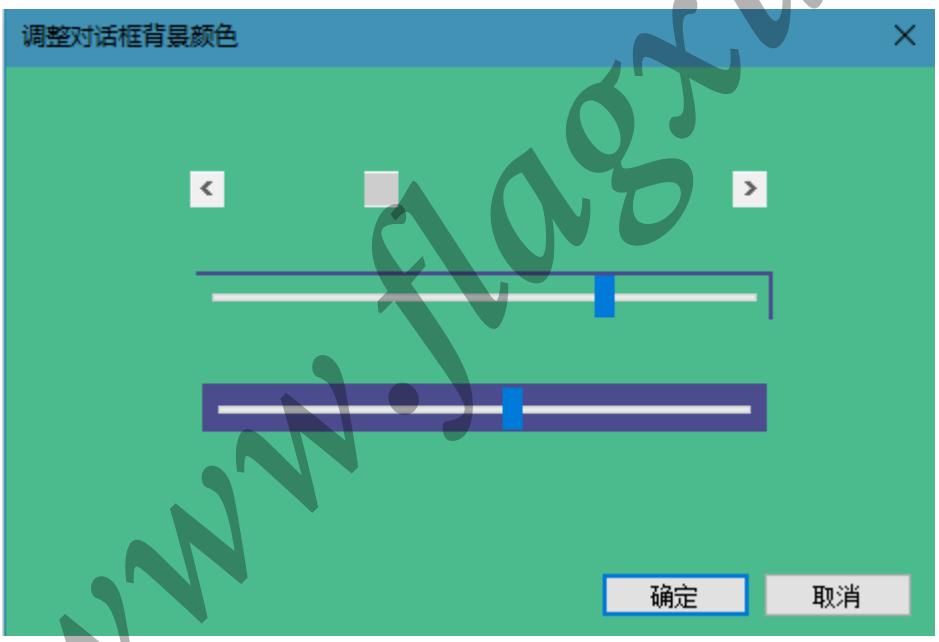
(6)动态创建按钮



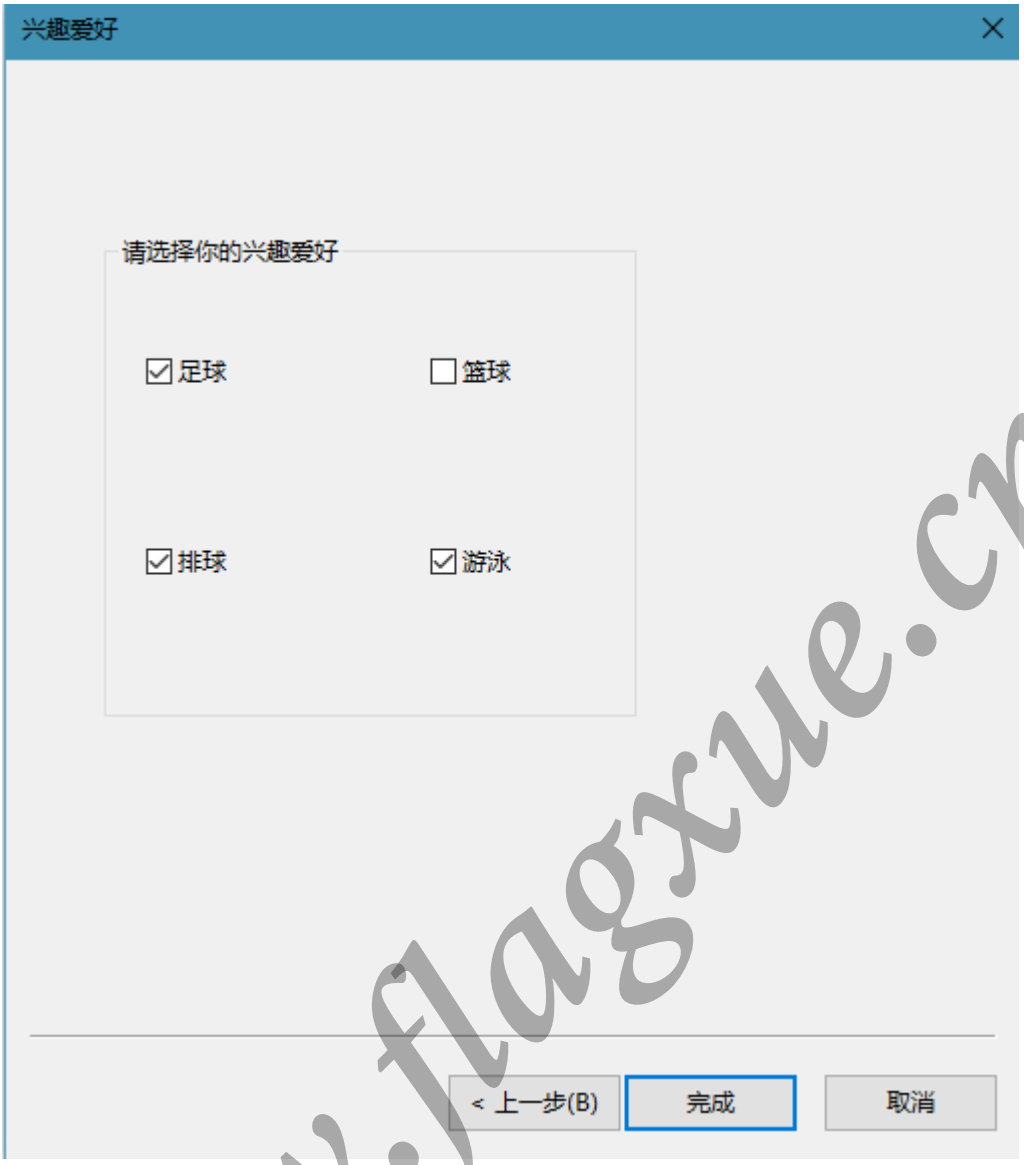
(7)单项选择/多项选择



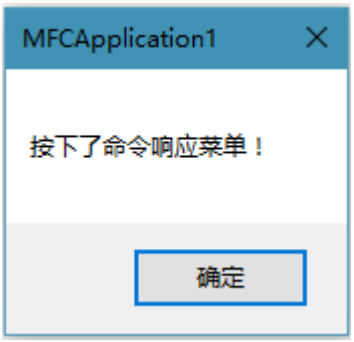
(8)滚动条/滑块



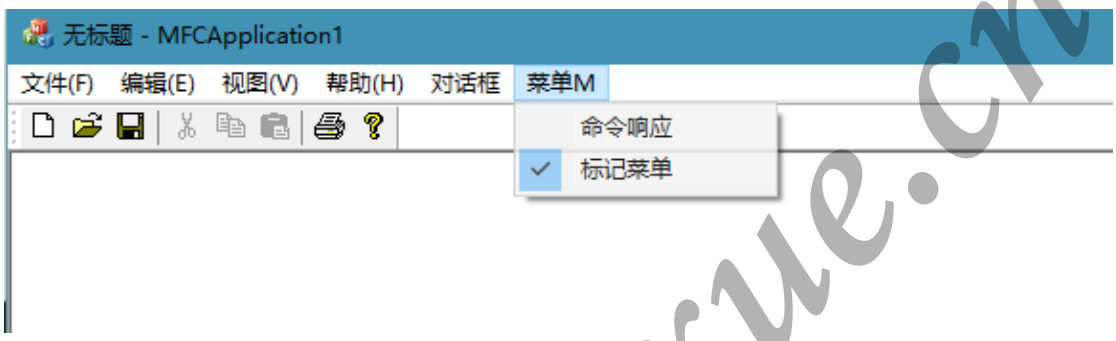
(9)属性表单



(10)命令响应菜单

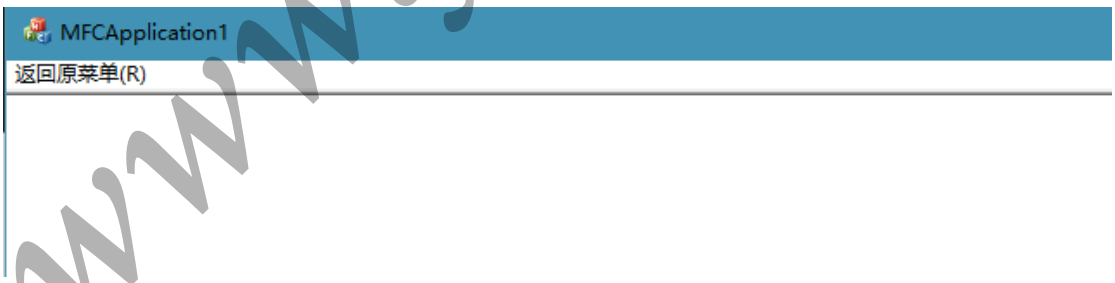


(11)标记菜单



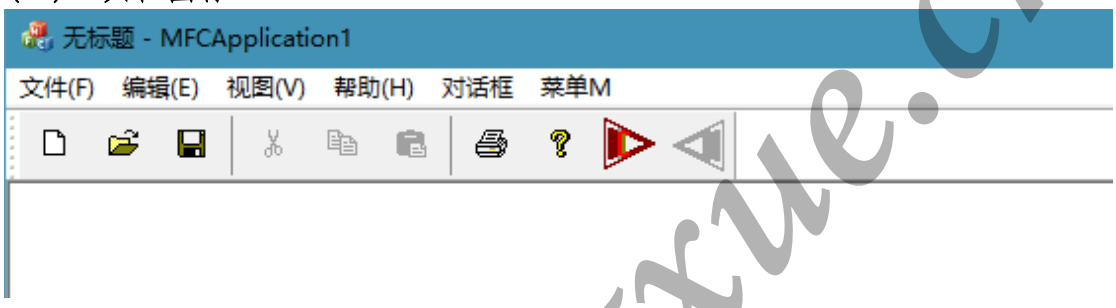
(12)(13)缺省菜单/图形菜单

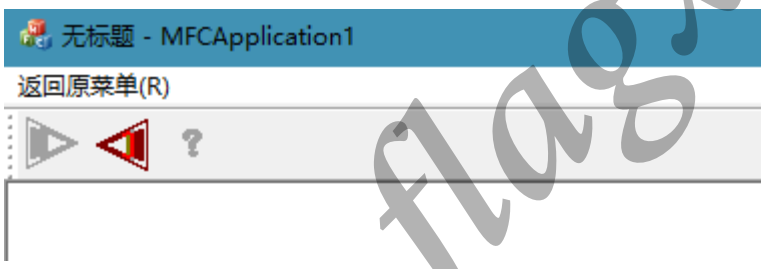
(14)新的菜单



(15)弹出式菜单

(16)工具栏图标





实验总结：

通过学习，我了解到MFC编程的实现是通过建立消息映射表来实现各个函数的调用，知道了一些类的基本使用规则。每种对话框的操作要通过先建立相应的CDialog类来具体进行。通过本次实验，使我更具体的了解了其中的各种类、函数和工具，让我初步接触了如何通过MFC实现基本的可视化编程，获益匪浅。

教师评语或评价表格：（任课教师可根据实际情况，做适当调整）

评语及评价表格的字体颜色为红色

评价表格示例：（考核标准与教学大纲中的实验考核标准一致）

考核标准	得分
（1）正确理解和掌握实验所涉及的概念和原理（20%）；	
（2）能设计测试用例，运行结果正确（20%）；	
（3）认真记录实验数据，原理及实验结果分析准确（20%）；	

（4）所做实验具有一定的创新性（10%）；	
（5）实验报告规范（30%）。	
总分	