

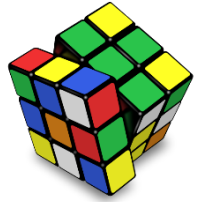
Ứng dụng MFC (Visual C++) trong mô phỏng Robot và hệ Cơ điện tử



Bài 12: Thiết kế nâng cao với ứng dụng kiểu SDI (2)

PHẠM MINH QUÂN

mquan.ph@gmail.com



Nội dung

1. Popup Dialog

1.1. Mở popup dialog

1.2. Nạp giá trị khởi tạo cho dialog

1.3. Nhận kết quả trả về từ dialog

1.4. Message Box

2. File Dialog

2.1. Tự định nghĩa một định dạng file

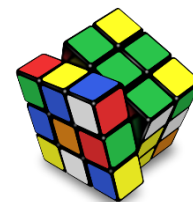
2.2. Gọi dialog để lưu file

2.3. Gọi dialog để tải file

3. Context Menu

4. Keyboard

5. Ribbon Menu



1. Popup Dialog

❑ Chuẩn bị dữ liệu

- Tạo project MFC, kiểu ứng dụng SDI, kiến trúc Doc/View. Cài đặt OpenGL vào lớp C...View như hướng dẫn trong Bài 10

- Khai báo thêm 1 thuộc tính của lớp COpenGLControl để lưu thông tin màu nền

// File OpenGLControl.h

```
| COLORREF BackgroundColor;
```

// File OpenGLControl.cpp

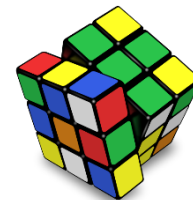
```
□ COpenGLControl::COpenGLControl(void)
| {
|     ...
|     BackgroundColor = RGB(0,0,255);
| }
```

```
□ void COpenGLControl::oglInitialize(void)
| {
|     ...
|     // Set color to use when clearing the background.
|     glClearColor((GLfloat)GetRValue(BackgroundColor)/255,
|                  (GLfloat)GetGValue(BackgroundColor)/255,
|                  (GLfloat)GetBValue(BackgroundColor)/255,
|                  0.0); //Sửa đổi lệnh glClearColor để
|     ...              sử dụng thuộc tính mới
| }
```

- Thêm 1 hàm vào lớp COpenGLControl để thay đổi màu nền khi cần

```
| void oglChangeBgColor(COLORREF BgColor);
```

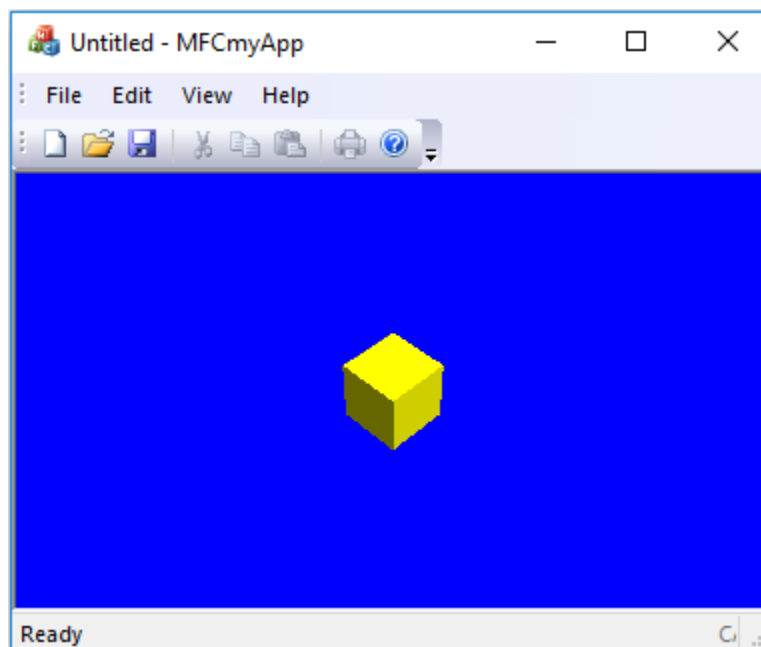
```
□ void COpenGLControl::oglChangeBgColor(COLORREF BgColor)
| {
|     BackgroundColor = BgColor;
|     glClearColor((GLfloat)GetRValue(BackgroundColor)/255,
|                  (GLfloat)GetGValue(BackgroundColor)/255,
|                  (GLfloat)GetBValue(BackgroundColor)/255,
|                  0.0);
|     oglDrawScene();
| }
```

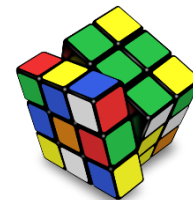


1. Popup Dialog

❑ Chuẩn bị dữ liệu

➡ *Kết quả chạy:*

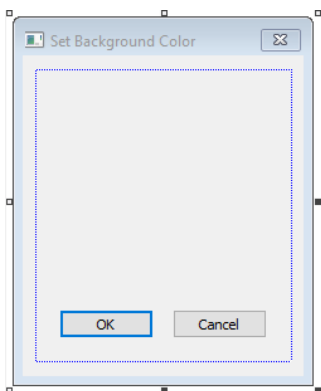




1. Popup Dialog

1.1. Mở popup dialog

- Thêm 1 dialog mới vào project, kiểu Popup



Properties:

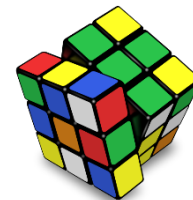
Style	Popup
ID	IDD_DIALOG_BGCOLOR

- Thêm 1 lớp mới (dẫn xuất từ CDialogEx) để quản lý dialog vừa tạo

Class name:	BgColorDlg
Base class:	CDialogEx
Dialog ID:	IDD_DIALOG_BGCOLOR

Lưu ý: Nên thêm khai báo sau trong file ...Dlg.h:

```
#include "resource.h"
```



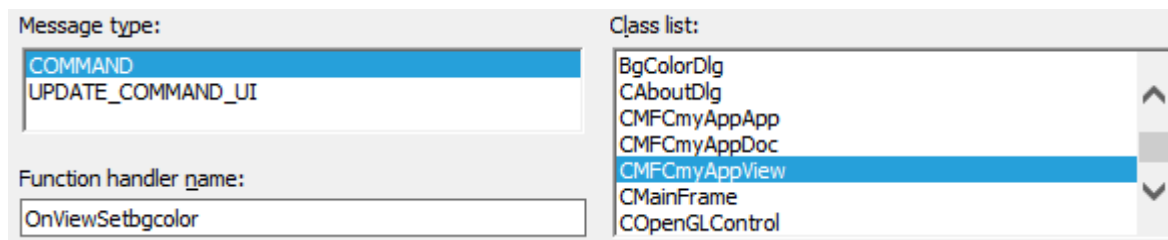
1. Popup Dialog

1.1. Mở popup dialog

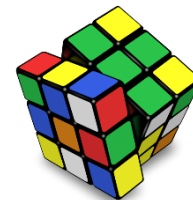
- Thêm vào menu IDR_MAINFRAME một nút lệnh mới



- Thêm hàm chức năng cho nút lệnh mới trên menu



//Chọn lớp C...View



1. Popup Dialog

1.1. Mở popup dialog

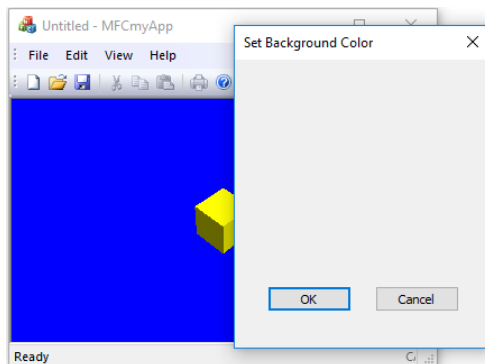
- Viết code cho hàm chức năng của nút lệnh mới

Kiểu Modal dialog

```
void CMFCmyAppView::OnViewSetbgcolor()
{
    // TODO: Add your command handler code here
    BgColorDlg m_BgcDlg;
    m_BgcDlg.DoModal();
}
```

Kết quả chạy:

- Khi click menu thì dialog mở ra
- Không thể tương tác với các cửa sổ khác (khi dialog đang mở)

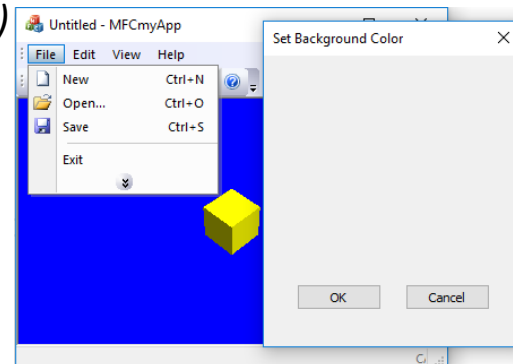


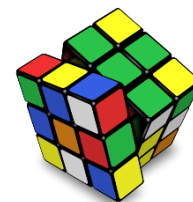
Kiểu Modeless dialog

```
void CMFCmyAppView::OnViewSetbgcolor()
{
    // TODO: Add your command handler code here
    BgColorDlg* pBgcdlg = new BgColorDlg(this);
    pBgcdlg->Create(IDD_DIALOG_BGCOLOR);
    pBgcdlg->ShowWindow(SW_SHOWNORMAL);
}
```

Kết quả chạy:

- Khi click menu thì dialog mở ra
- Vẫn tương tác được với các cửa sổ khác (khi dialog đang mở)

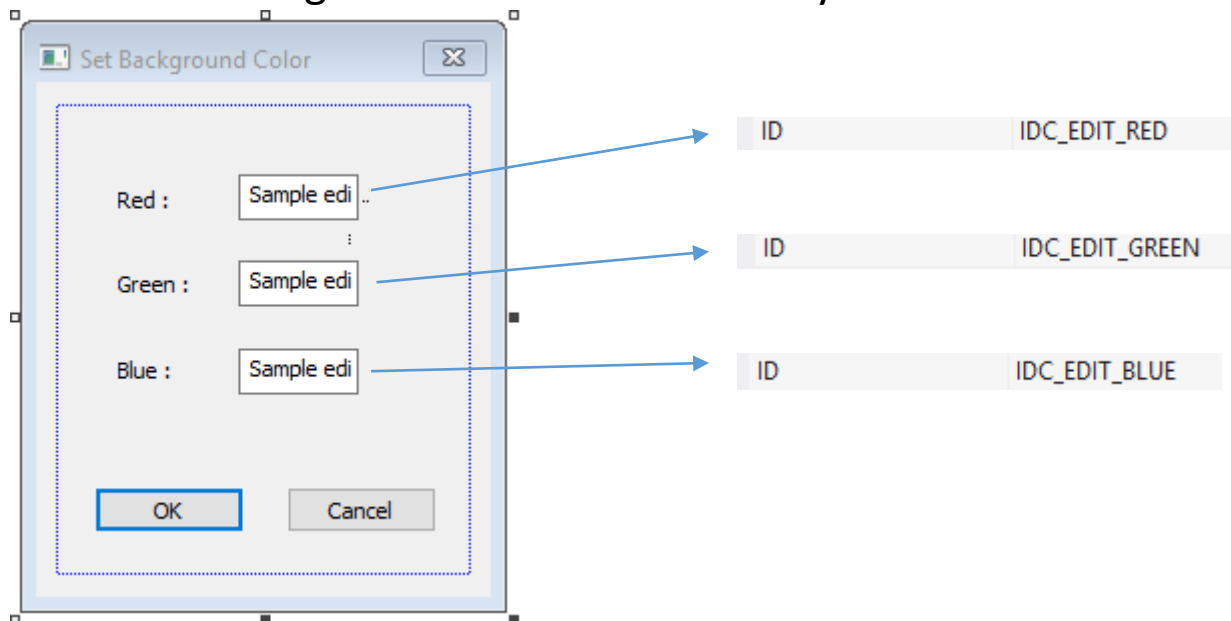




1. Popup Dialog

1.2. Nạp giá trị khởi tạo cho dialog

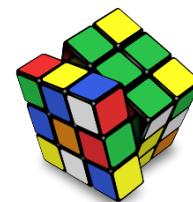
- Thêm vào dialog các editbox như dưới đây



- Thêm các biến *Value*, kiểu *int*, gắn với các ô editbox

```
public:  
    int EditR_val;  
    int EditG_val;  
    int EditB_val;
```

```
DDX_Text(pDX, IDC_EDIT_RED, EditR_val);  
DDX_Text(pDX, IDC_EDIT_GREEN, EditG_val);  
DDX_Text(pDX, IDC_EDIT_BLUE, EditB_val);
```

1. Popup Dialog

1.2. Nạp giá trị khởi tạo cho dialog

- Thêm code để nạp giá trị ban đầu cho các textbox mỗi khi mở dialog

*Cách 1: Kiểu Modal dialog

```
void CMFCmyAppView::OnViewSetbgcolor()
{
    // TODO: Add your command handler code here
    BgColorDlg m_BgcDlg;
    m_BgcDlg.EditR_val = GetRValue(m_oglWindow.BackgroundColor);
    m_BgcDlg.EditG_val = GetGValue(m_oglWindow.BackgroundColor);
    m_BgcDlg.EditB_val = GetBValue(m_oglWindow.BackgroundColor);
    m_BgcDlg.DoModal();
}
```

Kiểu Modeless dialog

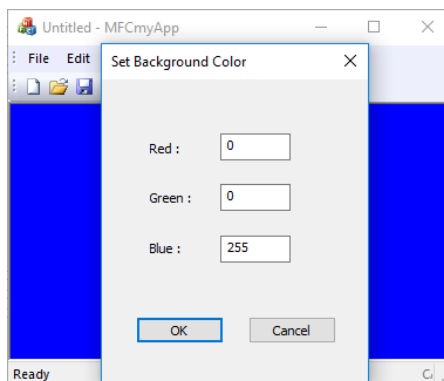
```
void CMFCmyAppView::OnViewSetbgcolor()
{
    // TODO: Add your command handler code here
    BgColorDlg* pBgcdlg = new BgColorDlg(this);
    pBgcdlg->Create(IDD_DIALOG_BGCOLOR);
    pBgcdlg->EditR_val = GetRValue(m_oglWindow.BackgroundColor);
    pBgcdlg->EditG_val = GetGValue(m_oglWindow.BackgroundColor);
    pBgcdlg->EditB_val = GetBValue(m_oglWindow.BackgroundColor);
    pBgcdlg->UpdateData(FALSE);
    pBgcdlg->ShowWindow(SW_SHOWNORMAL);
}
```

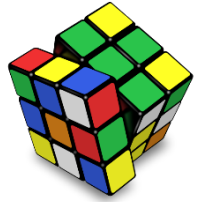
*Cách 2: Thêm hàm OnInitDialog() vào lớp quản lý hộp thoại, viết code khởi tạo giá trị trong hàm này



Kết quả chạy:

- Các giá trị màu mặc định hiển thị sẵn khi mở dialog



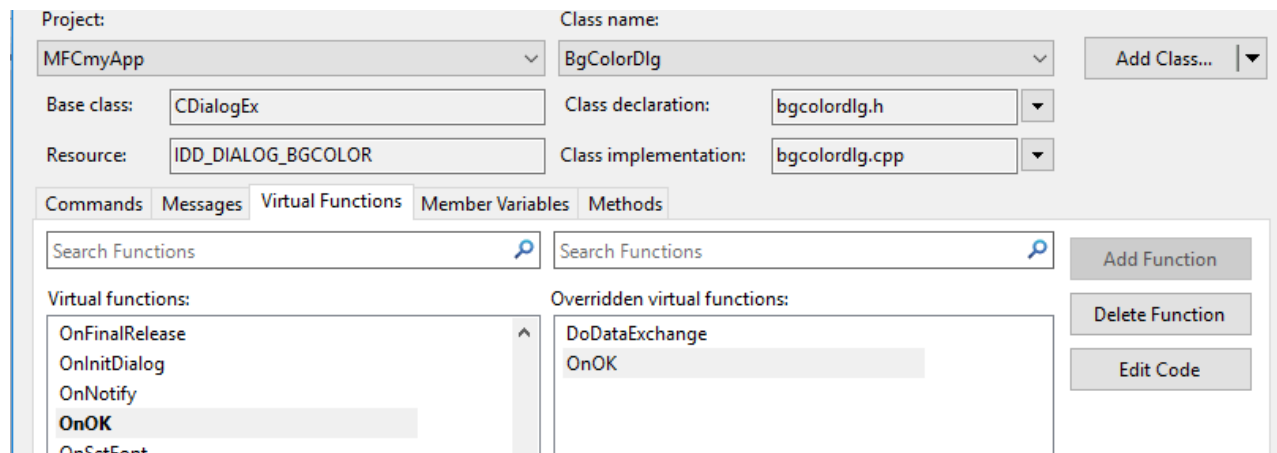


1. Popup Dialog

1.3. Nhận kết quả trả về từ dialog

❑ Cách 1: (dùng được cho cả 2 kiểu Modal và Modeless dialog)

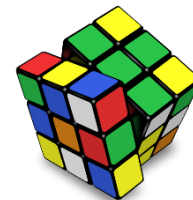
➤ Thêm hàm OnOK() vào lớp ...Dlg



➤ Sửa code hàm OnOK()

```
void BgColorDlg::OnOK()
{
    // TODO: Add your specialized code here and/or call the base class
    UpdateData(TRUE);
    CMFCMyAppView *pView = CMFCMyAppView::GetView();
    pView->m_oglWindow.oglChangeBgColor(RGB(EditR_val, EditG_val, EditB_val));

    CDialogEx::OnOK();
}
```



1. Popup Dialog

1.3. Nhận kết quả trả về từ dialog

❑ Cách 2: (chỉ dùng được cho kiểu Modal dialog)

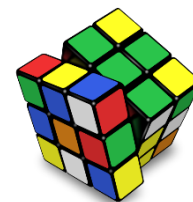
➤ Thêm hàm OnOK() vào lớp ...Dlg và sửa code như sau:

```
void BgColorDlg::OnOK()
{
    // TODO: Add your specialized code here and/or call the base class
    UpdateData(TRUE);

    CDialogEx::OnOK();
}
```

➤ Sửa code hàm chức năng của nút lệnh menu như sau

```
void CMFCmyAppView::OnViewSetbgcolor()
{
    // TODO: Add your command handler code here
    BgColorDlg m_BgcDlg;
    m_BgcDlg.EditR_val = GetRValue(m_oglWindow.BackgroundColor);
    m_BgcDlg.EditG_val = GetGValue(m_oglWindow.BackgroundColor);
    m_BgcDlg.EditB_val = GetBValue(m_oglWindow.BackgroundColor);
    if (m_BgcDlg.DoModal() == IDOK)
    {
        m_oglWindow.oglChangeBgColor(RGB(m_BgcDlg.EditR_val, m_BgcDlg.EditG_val, m_BgcDlg.EditB_val));
    }
}
```



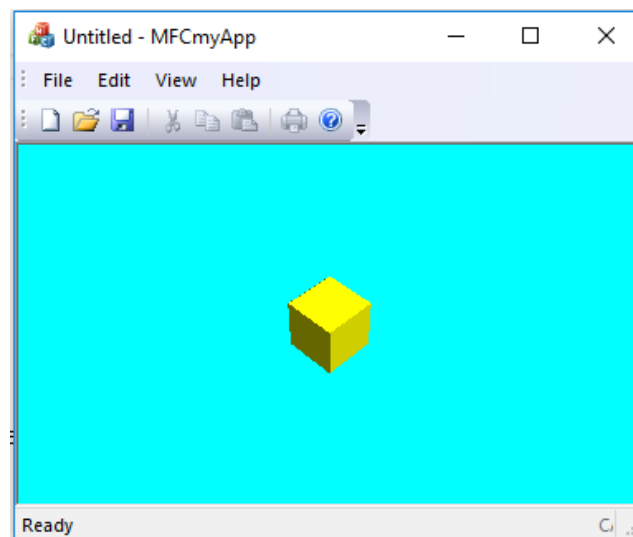
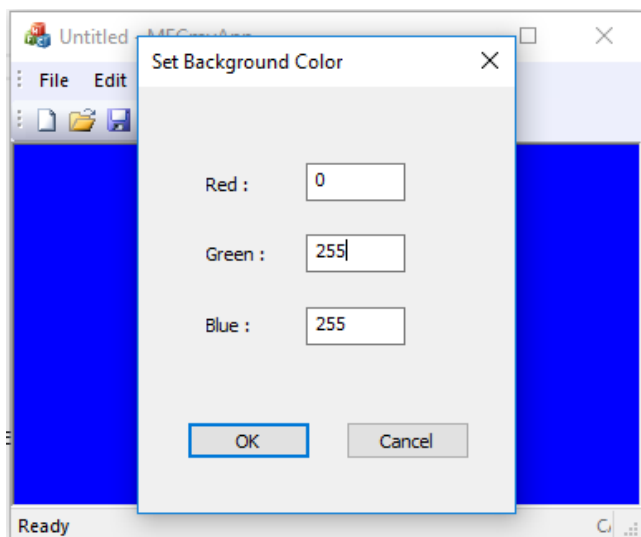
1. Popup Dialog

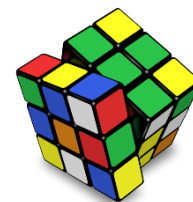
1.3. Nhận kết quả trả về từ dialog



Kết quả chạy:

- Khi nhập 1 bộ số vào các editbox và click ok, thì dialog đóng lại và khung OpenGL chuyển sang màu nền mới





1. Popup Dialog

1.4. Message Box

- Hiển thị thông báo nhanh khi giá trị màu nằm ngoài khoảng [0,255]

```
void BgColorDlg::OnOK()
{
    // TODO: Add your specialized code here and/or call the base class
    UpdateData(TRUE);
    if (EditR_val<0 || EditR_val>255 || EditG_val<0 || EditG_val>255 || EditB_val<0 || EditB_val>255)
    {
        MessageBox(_T("Values must be in the range [0,255]"), _T("Error"), MB_OK | MB_ICONERROR);
        return;
    }
    CDialogEx::OnOK();
}
```

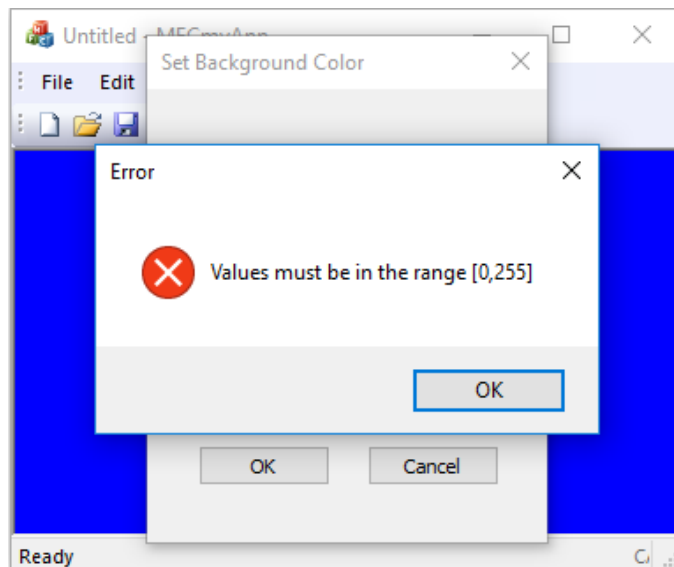
- MB_ABORTRETRYIGNORE
- MB_OK
- MB_OKCANCEL
- MB_RETRYCANCEL
- MB_YESNO
- MB_YESNOCANCEL

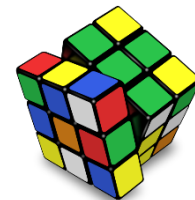
- MB_ICONHAND, MB_ICONSTOP, and MB_ICONERROR
- MB_ICONQUESTION
- MB_ICONEXCLAMATION and MB_ICONWARNING
- MB_ICONASTERISK and MB_ICONINFORMATION



Kết quả chạy:

- Khi nhập 1 số ngoài khoảng [0,255], nhấn OK thì xuất hiện thông báo lỗi. Dialog vẫn được mở để nhập lại giá trị khác





1. Popup Dialog

❑ Một số mẹo khác cho Popup Dialog:

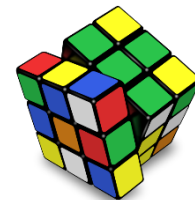
- Việc thiết kế các control trên popup dialog tương tự như với bất kỳ một dialog nào khác (tham khảo nội dung các bài ở Phần I)
- Khi mở nhiều dialog và muốn giữ 1 dialog nào đó luôn ở trên cùng thì sử dụng lệnh sau:

```
SetWindowPos(&wndTopMost, 0, 0, 0, 0, SWP_NOMOVE | SWP_NOSIZE);
```

Câu lệnh hủy bỏ thiết lập này là:

```
SetWindowPos(&wndNoTopMost, 0, 0, 0, 0, SWP_NOMOVE | SWP_NOSIZE);
```

- Modal và Modeless dialog:
 - Nên hạn chế sử dụng modeless dialog trừ khi có mục đích cụ thể.
 - Cần có cơ chế kiểm soát để tránh mở trùng dialog và xung đột dữ liệu khi dùng modeless dialog.
- MessageBox() và AfxMessageBox():
 - MessageBox() là hàm thuộc lớp CWnd và các lớp dẫn xuất của lớp này.
 - AfxMessageBox() là hàm toàn cục và có thể được gọi từ bất kỳ đâu.
 - Hai hàm này có ý nghĩa và cách sử dụng tương đương, khác nhau ở bộ tham số.



1. Popup Dialog

❑ Một số mẹo khác cho Popup Dialog:

- Việc lựa chọn màu sắc có thể được thực hiện trực quan với sự hỗ trợ của các lớp CColorDialog hoặc CMFCColorDialog.

Ví dụ: Thêm 1 button vào dialog BgColorDlg và viết code cho hàm sự kiện click nút đó như sau:

```
void BgColorDlg::OnBnClickedButtonChoosecolor()
```

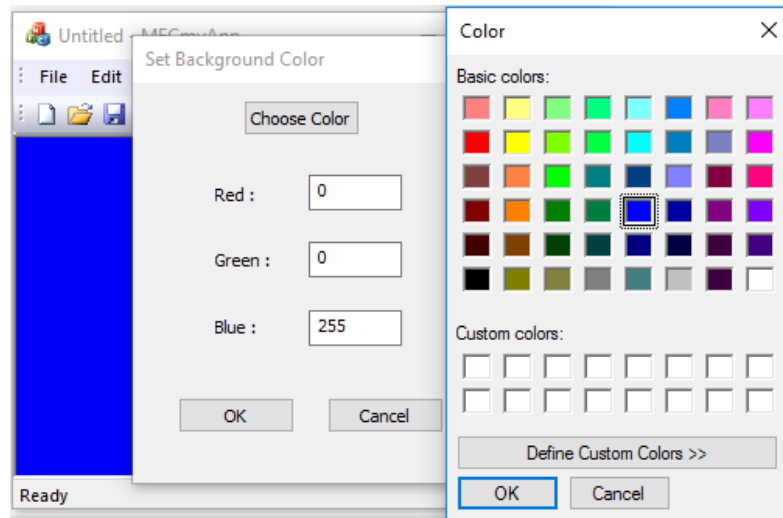
```
{  
    // TODO: Add your control notification handler code here
```

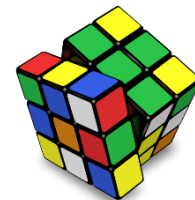
```
    CColorDialog m_ColorDlg(RGB(EditR_val, EditG_val, EditB_val)); //Thay CColorDialog bằng CMFCColorDialog để thấy  
    if(m_ColorDlg.DoModal()==IDOK)                                kết quả khác nhau
```

```
{  
    COLORREF ChosenColor = m_ColorDlg.GetColor();  
    EditR_val = GetRValue(ChosenColor);  
    EditG_val = GetGValue(ChosenColor);  
    EditB_val = GetBValue(ChosenColor);  
    UpdateData(FALSE);  
}
```



Kết quả chạy:

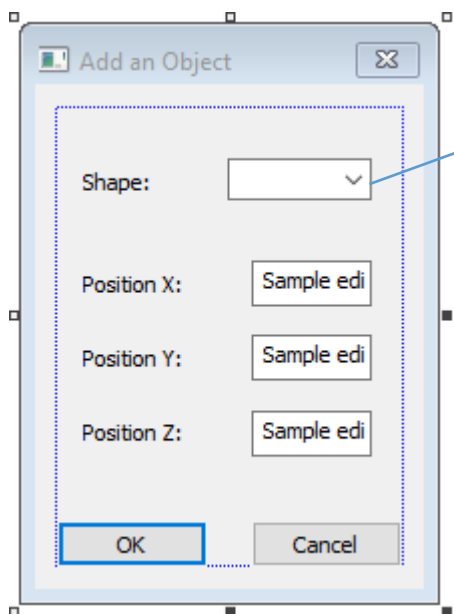




1. Popup Dialog

- ❑ Áp dụng: Thêm 1 popup dialog cho phép người dùng thêm các hình khối vào không gian đồ họa OpenGL, với hình dạng, vị trí mong muốn.

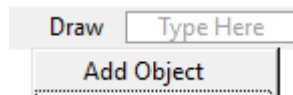
Gợi ý: • Thêm một dialog mới để thực hiện chức năng tạo thêm khối hình OpenGL. Có thể thiết kế dialog mới như sau:

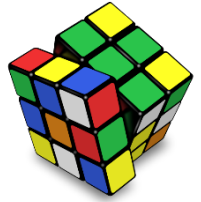


Combo box

- Biến quản lý Combobox
`CComboBox CboShape_ctrl;`
- Thêm nội dung lựa chọn vào combobox
`CboShape_ctrl.InsertString(0, _T("A"));`
- Thiết lập giá trị được chọn
`CboShape_ctrl.SetCurSel(0);`
- Lấy giá trị được chọn
`CboShape_ctrl.GetCurSel();`
- Hàm xử lý khi thay đổi lựa chọn
`OnCbnSelchangeComboShape()`

- Thêm nút lệnh menu và thêm hàm chức năng của nút lệnh (trong lớp C...View)





1. Popup Dialog

□ Áp dụng: Thêm 1 popup dialog cho phép người dùng thêm các hình khối vào không gian đồ họa OpenGL, với hình dạng, vị trí mong muốn.

- Gợi ý:
- Thêm lớp `OglObject` để quản lý các khối hình như hướng dẫn trong Bài 11
 - Sử dụng cấu trúc `std::vector<>` để lập mảng các đối tượng `OglObject`

// File OpenGLControl.h

```
#include <vector>

class COpenGLControl : public CWnd
{
    ...

    std::vector<OglObject> ObjArray;

    ...

};
```

// File OpenGLControl.cpp

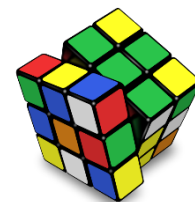
```
COpenGLControl::COpenGLControl(void)
{
    ...

    ObjArray.clear();
    ObjArray.push_back(OglObject());
}

void COpenGLControl::oglDrawScene(void)
{
    ...

    if(!ObjArray.empty())
        for(int i=0; i<ObjArray.size(); i++)
            ObjArray[i].Draw();

    ...
}
```



1. Popup Dialog

- ❑ Áp dụng: Thêm 1 popup dialog cho phép người dùng thêm các hình khối vào không gian đồ họa OpenGL, với hình dạng, vị trí mong muốn.

Gợi ý: • Thêm các hàm cần thiết vào lớp ...Dlg, sửa code để thực hiện chức năng mong muốn

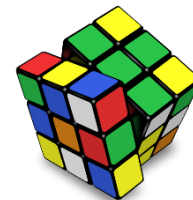
```
BOOL AddObjDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // TODO: Add extra initialization here
    CboShape_ctrl.ResetContent();
    CboShape_ctrl.InsertString(0, _T("Box"));
    CboShape_ctrl.InsertString(1, _T("Sphere"));
    CboShape_ctrl.SetCurSel(0);

    return TRUE; // return TRUE unless you set the focus to a control
    // EXCEPTION: OCX Property Pages should return FALSE
}

void AddObjDlg::OnOK()
{
    // TODO: Add your specialized code here and/or call the base class
    UpdateData(TRUE);
    CoglObject NewObj;
    NewObj.SetType(CboShape_ctrl.GetCurSel()+1);
    NewObj.SetPosition(EditPosX_val, EditPosY_val, EditPosZ_val);
    CMFCmyAppView *pView = CMFCmyAppView::GetView();
    pView->m_oglWindow.ObjArray.push_back(NewObj);
    pView->m_oglWindow oglDrawScene();

    CDialogEx::OnOK();
}
```



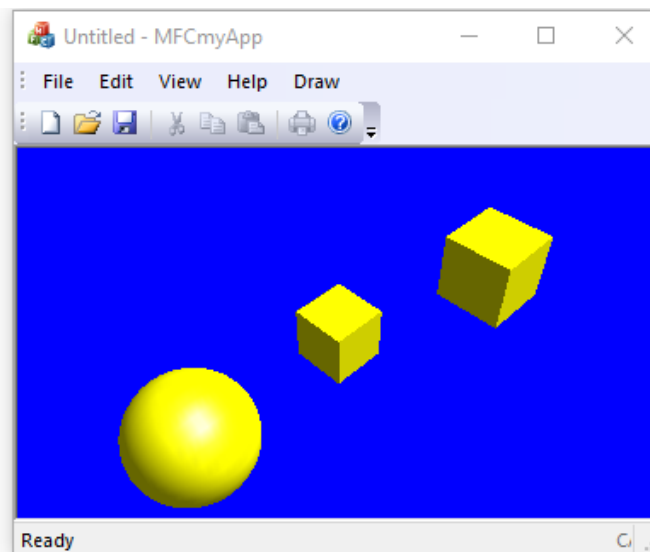
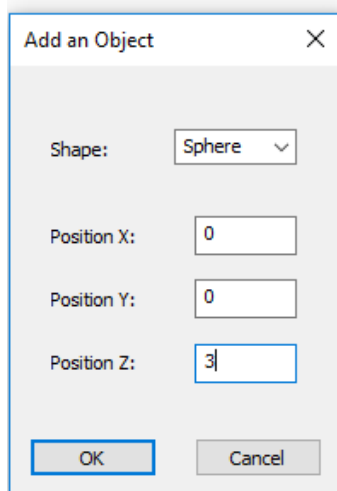
1. Popup Dialog

- ❑ Áp dụng: Thêm 1 popup dialog cho phép người dùng thêm các hình khối vào không gian đồ họa OpenGL, với hình dạng, vị trí mong muốn.

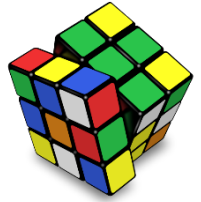


Kết quả chạy:

- Mở dialog “Add Object”, điền thông tin và nhấn OK, một hình khối mới xuất hiện trong khung đồ họa.
- Có thể thêm bao nhiêu hình tùy ý.



2. File Dialog



2. File Dialog

2.1. Tự định nghĩa một định dạng file

- Viết hàm ghi dữ liệu ra file.

```
bool WriteOglFile(wchar_t* filename, std::vector<OglObject> ObjArray);
```

// File GlobalFunctions.h

```
bool WriteOglFile(wchar_t* filename, std::vector<OglObject> ObjArray)
```

// File GlobalFunctions.cpp

```
{  
    FILE *f1;  
    _wfopen_s(&f1, filename, _T("w"));  
    if(f1== NULL) return FALSE;  
  
    fprintf(f1, "# List of Objects.\n");  
    int NoObj = ObjArray.size();  
    float posx, posy, posz;  
  
    fprintf(f1, "\nNumber of Objects\t%d\n", NoObj);  
  
    for(int i=0; i<NoObj; i++)  
    {  
        fprintf(f1, "\tShape\t%d\n", ObjArray[i].GetType());  
        ObjArray[i].GetPosition(posx, posy, posz);  
        fprintf(f1, "\tPosition\t%e\t%e\t%e\n", posx, posy, posz);  
    }  
    fprintf(f1, "\n# End of File.");  
    fclose(f1);  
    return TRUE;  
}
```



2. File Dialog

2.1. Tự định nghĩa một định dạng file

- Viết hàm đọc dữ liệu từ file.

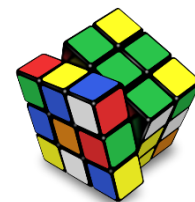
```
bool ReadOglFile(wchar_t* filename, std::vector<OglObject>& ObjArray);
```

// File GlobalFunctions.h

```
bool ReadOglFile(wchar_t* filename, std::vector<OglObject>& ObjArray)
```

// File GlobalFunctions.cpp

```
{  
    FILE *f1;  
    _wfopen_s(&f1, filename, _T("r"));  
    if(f1== NULL) return FALSE;  
    float tmp1,tmp2,tmp3;  
    int NoObj = 0;  
    int type;  
    OglObject Object;  
    ObjArray.clear();  
  
    while(fgetc(f1)!= '\n'){  
        fscanf_s(f1, "%s%s%s%d", &NoObj);  
  
        for(int i=0; i<NoObj; i++)  
        {  
            fscanf_s(f1, "%s%d", &type);  
            Object.SetType(type);  
            fscanf_s(f1, "%s%f%f%f", &tmp1, &tmp2, &tmp3);  
            Object.SetPosition(tmp1, tmp2, tmp3);  
            ObjArray.push_back(Object);  
        }  
        fclose(f1);  
        return TRUE;  
    }  
}
```



2. File Dialog

2.2. Gọi dialog để lưu file

- Thêm 1 lệnh mới vào menu và viết code hàm chức năng của nút lệnh này như sau

```
void CMFCMyAppView::OnDrawSaveobjects()
{
    // TODO: Add your command handler code here
    char CurrentPath[_MAX_PATH];
    LPTSTR WorkingDir = (LPTSTR)CurrentPath;
    GetCurrentDirectory(_MAX_PATH, WorkingDir);

    CFileDialog m_fileDlg(FALSE, _T("ogl"), NULL, OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT, // Mở dialog để SAVE file
        _T("List of Objects (*.ogl)|*.ogl|"));

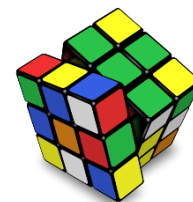
    m_fileDlg.m_pOFN->lpstrTitle=_T("Save a list of objects");
    m_fileDlg.m_pOFN->lpstrInitialDir = WorkingDir;

    if(m_fileDlg.DoModal()==IDOK)
    {
        wchar_t * FilePath=0;
        CString m_strPath = m_fileDlg.GetPathName();
        FilePath = m_strPath.GetBuffer(m_strPath.GetLength());
        m_strPath.ReleaseBuffer();

        if(!WriteOglFile(FilePath, m_oglWindow.ObjArray))
        {
            MessageBox(_T("Can not Write to File"), _T("Save File Error"), MB_OK);
            return;
        }
    }
}
```



Kết quả chạy: Khi click menu, chương trình mở hộp thoại Save và thực hiện chức năng như mong muốn



2. File Dialog

2.3. Gọi dialog để tải file

- Thêm 1 lệnh mới vào menu và viết code hàm chức năng của nút lệnh này như sau

```
void CMFCmyAppView::OnDrawLoadobjects()
{
    // TODO: Add your command handler code here
    char CurrentPath[_MAX_PATH];
    LPTSTR WorkingDir = (LPTSTR)CurrentPath;
    GetCurrentDirectory(_MAX_PATH, WorkingDir);

    CFileDialog m_fileDlg(TRUE, _T("ogl"), NULL, OFN_HIDEREADONLY|OFN_FILEMUSTEXIST, // Mở dialog để OPEN_file
        _T("List of Objects (*.ogl)|*.ogl|"), this);

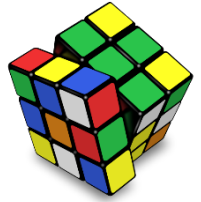
    m_fileDlg.m_pOFN->lpstrTitle=_T("Load a list of objects");
    m_fileDlg.m_pOFN->lpstrInitialDir = WorkingDir;

    if(m_fileDlg.DoModal()==IDOK)
    {
        wchar_t * FilePath=0;
        CString m_strPath = m_fileDlg.GetPathName();
        FilePath = m_strPath.GetBuffer(m_strPath.GetLength());
        m_strPath.ReleaseBuffer();

        if(!ReadOglFile(FilePath, m_oglWindow.ObjArray))
        {
            MessageBox(_T("File Not Found or Invalid File Name"), _T("File Not Found"), MB_OK);
            return;
        }
        m_oglWindow.oglDrawScene();
    }
}
```



Kết quả chạy: Khi click menu, chương trình mở hộp thoại Open và thực hiện chức năng như mong muốn



2. File Dialog

- ❑ Trường hợp dialog dùng để lưu nhiều định dạng file khác nhau.
Ví dụ: *.abc, *.xyz

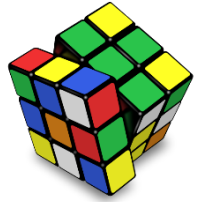
```
void CMFCMyAppView::OnDrawSaveobjects()
{
    ...

    CFileDialog fileDlg(FALSE, _T("abc"), NULL, OFN_HIDEREADONLY|OFN_OVERWRITEPROMPT,
        _T("The first file type (*.abc)|*.abc|The second file type (*.xyz)|*.xyz|"));

    ...

    if(fileDlg.DoModal()==IDOK)
    {
        ...

        CString m_strExt = fileDlg.GetFileExt();
        CString ext1("abc"), ext2("xyz");
        if(!m_strExt.Compare(ext1))
        {
            ...
        }
        else if(!m_strExt.Compare(ext2))
        {
            ...
        }
    }
}
```



2. File Dialog

- ❑ Trường hợp dialog dùng để mở nhiều định dạng file khác nhau.

Ví dụ: *.abc, *.xyz

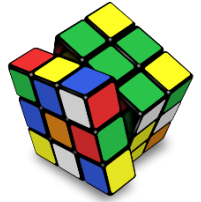
```
void CMFCmyAppView::OnDrawLoadobjects()
{
    ...

    CFileDialog fileDlg(TRUE, _T("abc"), NULL, OFN_HIDEREADONLY|OFN_FILEMUSTEXIST,
        _T("The first file type (*.abc)|*.abc|The second file type (*.xyz)|*.xyz|"), this);

    ...

    if(fileDlg.DoModal()==IDOK)
    {
        ...

        CString m_strExt = fileDlg.GetFileExt();
        CString ext1("abc"), ext2("xyz")
        if(!m_strExt.Compare(ext1))
        {
            ...
        }
        else if(!m_strExt.Compare(ext2))
        {
            ...
        }
        ...
    }
}
```



3. Context Menu

❑ Chuẩn bị dữ liệu

- Thêm tính năng chọn khối hình bằng click chuột như hướng dẫn trong Bài 11
- So với nội dung Bài 11, cần thay đổi code của 1 số hàm như sau:

```
void COpenGLControl::DrawInMode(GLint mode)
{
    glPushMatrix();
    GLfloat mat_color_normal[] = { 1.0, 1.0, 0.0 };
    GLfloat mat_color_selected[] = { 1.0, 0.0, 0.0 };
    GLfloat mat_specular[] = { 1.0, 1.0, 1.0 };
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glMaterialf(GL_FRONT, GL_SHININESS, 30);

    for(int i=0; i<ObjArray.size(); i++)
    {
        if (SelectedItem==(i+1)) glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mat_color_selected);
        else glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, mat_color_normal);
        if (mode == GL_SELECT) glLoadName(i+1);
        ObjArray[i].Draw();
    }

    glPopMatrix();
}
```



3. Context Menu

❑ Chuẩn bị dữ liệu

- Thêm tính năng chọn khối hình bằng click chuột như hướng dẫn trong Bài 11
- So với nội dung Bài 11, cần thay đổi code của 1 số hàm như sau:

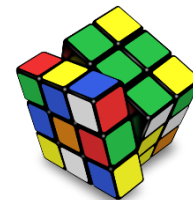
```
void COpenGLControl::OnMouseMove(UINT nFlags, CPoint point)
{
    ...

    else
    {
        double MousePosX, MousePosY, MousePosZ;
        GetMousePositionInOgl(point.x, point.y, 0, m_LastMouseZDepth, MousePosX, MousePosY, MousePosZ);

        ObjArray[SelectedItem-1].SetX(ObjArray[SelectedItem-1].GetX() + MousePosX-m_LastMousePosX);
        ObjArray[SelectedItem-1].SetY(ObjArray[SelectedItem-1].GetY() + MousePosY-m_LastMousePosY);
        ObjArray[SelectedItem-1].SetZ(ObjArray[SelectedItem-1].GetZ() + MousePosZ-m_LastMousePosZ);

        m_LastMousePosX = MousePosX;
        m_LastMousePosY = MousePosY;
        m_LastMousePosZ = MousePosZ;
    }

    ...
}
```



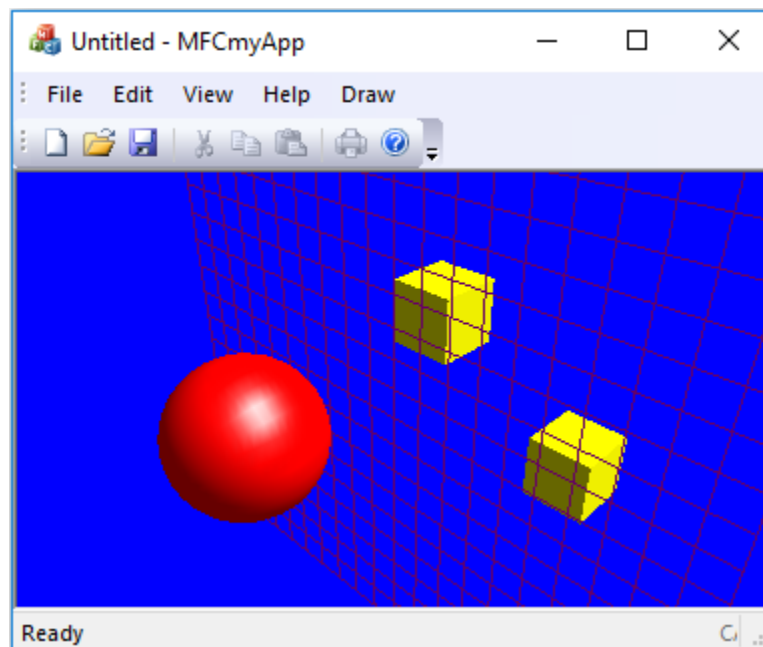
3. Context Menu

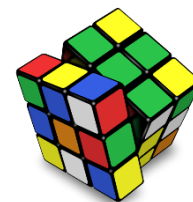
□ Chuẩn bị dữ liệu



Kết quả chạy:

- Dùng dialog “Add Object” để thêm các khối hình mới vào không gian đồ họa.
- Khi click chuột vào khối hình nào thì khối hình đó chuyển sang màu đỏ và có thể di chuyển bằng cách nhấn giữ chuột trái.





3. Context Menu

❑ Chuẩn bị dữ liệu

- Trả lại nội dung ban đầu cho hàm OnContextMenu() của lớp C...View

```
void CMFCMyAppView::OnRButtonUp(UINT /* nFlags */, CPoint point)
{
    ClientToScreen(&point);
    OnContextMenu(this, point);
}

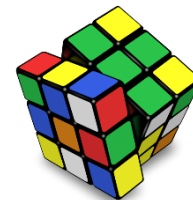
void CMFCMyAppView::OnContextMenu(CWnd* /* pWnd */, CPoint point)
{
    #ifndef SHARED_HANDLERS
        theApp.GetContextMenuManager()->ShowPopupMenu(IDR_POPUP_EDIT, point.x, point.y, this, TRUE);
    #endif
}
```

➡ Kết quả chạy: Khi click chuột phải thì menu context mặc định hiện ra. Tuy nhiên, thao tác này làm sai lệch các thao tác Rotate, Zoom, Pan. Vì vậy, cần có các thay đổi sau.

- Thêm các hàm sự kiện WM_MBUTTONDOWN, WM_RBUTTONDOWN vào lớp COpenGLControl
- Thêm các dòng lệnh sau vào cả 3 hàm: OnLButtonDown(), OnMButtonDown(), OnRButtonDown() của lớp COpenGLControl

```
m_fLastX = (float)point.x;
m_fLastY = (float)point.y;
```

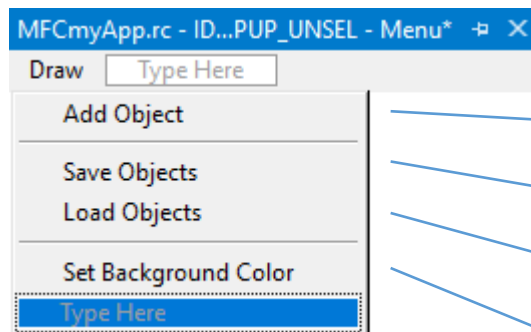
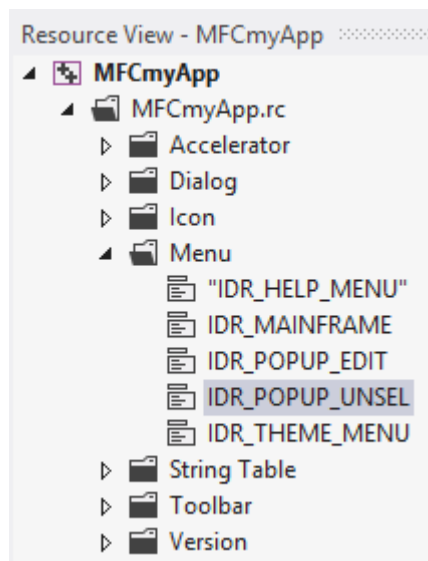
➡ Kết quả chạy: Khắc phục ảnh hưởng của việc mở context menu đến các thao tác chuột khác



3. Context Menu

❑ Thiết kế menu

- Thêm 1 menu mới vào project và thiết kế các nút lệnh như sau:



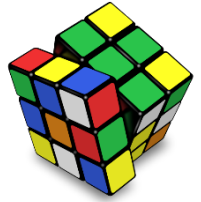
//Đặt ID trùng với ID của các nút lệnh tương ứng trên menu IDR_MAINFRAME

ID	ID_DRAW_ADDOBJ
ID	ID_DRAW_SAVEOBJECTS
ID	ID_DRAW_LOADOBJECTS
ID	ID_VIEW_SETBGCOLOR

- Thêm chuỗi ký tự vào String Table để chú thích về menu mới và các lệnh của nó

IDS_MENU_UNSEL	101	Show Right-button menu when no item is selected
ID_VIEW_SETBGCOLOR	32776	Set background color
ID_DRAW_ADDOBJ	32778	Add an Object
ID_DRAW_SAVEOBJECTS	32779	Save objects to file
ID_DRAW_LOADOBJECTS	32780	Load objects from file

//Nếu đã thêm chú thích khi thiết kế menu IDR_MAINFRAME thì không cần tạo lại



3. Context Menu

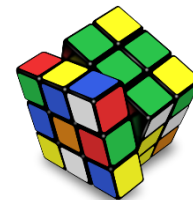
❑ Tích hợp menu

- Sửa code hàm C...App::PreLoadState()

```
void CMFCMyAppApp::PreLoadState() // File MFC<tên app>.cpp
{
    BOOL bNameValid;
    CString strName;
    bNameValid = strName.LoadString(IDS_MENU_UNSEL);
    ASSERT(bNameValid);
    GetContextMenuManager()->AddMenu(strName, IDR_POPUP_UNSEL);
}
```

- Sửa code hàm C...View::OnContextMenu()

```
void CMFCMyAppView::OnContextMenu(CWnd* /* pWnd */, CPoint point) // File MFC...View.cpp
{
#ifdef SHARED_HANDLERS
    theApp.GetContextMenuManager()->ShowPopupMenu(IDR_POPUP_UNSEL, point.x, point.y, this, TRUE);
#endif
}
```

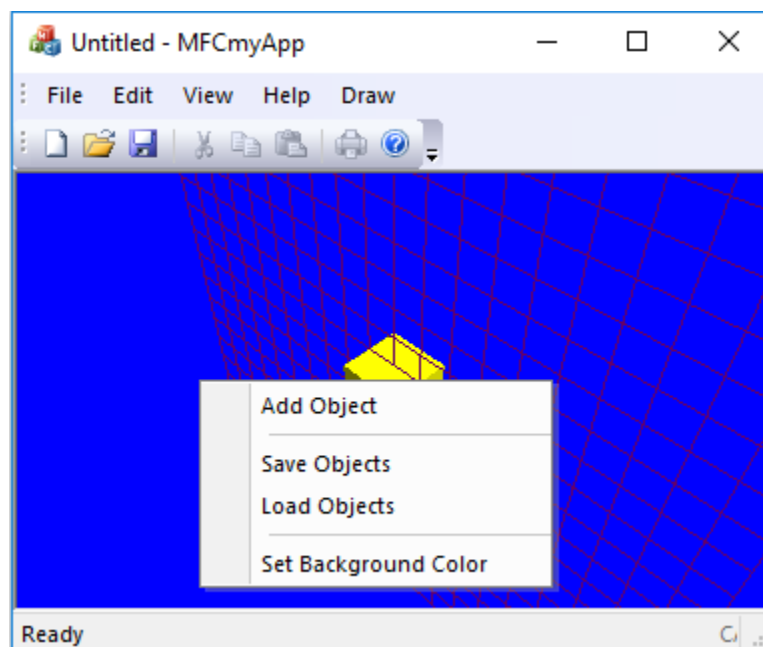
3. Context Menu

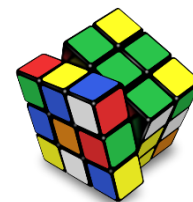
☐ Tích hợp menu



Kết quả chạy:

- Click chuột phải trong khung View, menu ngữ cảnh hiện ra như đã thiết kế.
- Click vào các nút lệnh, thực hiện chức năng tương tự như trên menu chính.

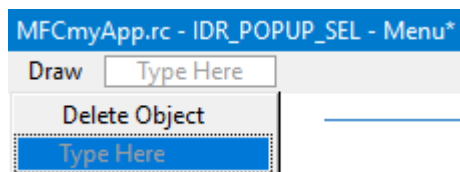
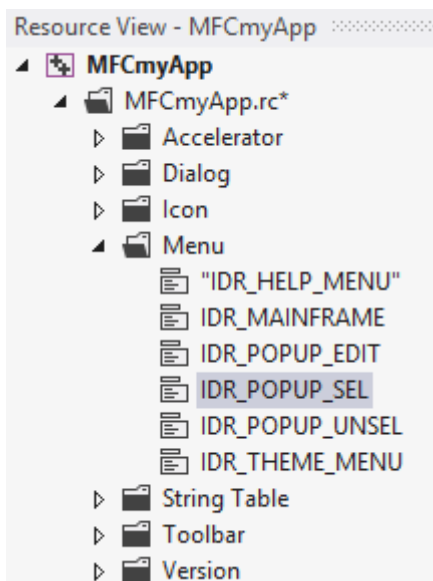




3. Context Menu

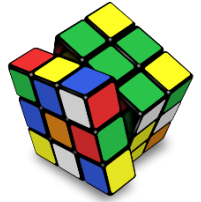
□ Tương tự, thêm 1 menu ngữ cảnh mới

➤ Thêm 1 menu mới vào project và thiết kế các nút lệnh như sau:



ID	ID_DRAW_DELOBJ
----	----------------

IDS_MENU_SEL	102	Show Right-button menu when one item is selected
ID_DRAW_DELOBJ	32789	Delete the selected object



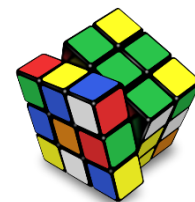
3. Context Menu

□ Tương tự, thêm 1 menu ngữ cảnh mới

- Chuyển code trong hàm `COpenGLControl::OnLButtonDown()` sang một hàm mới.

```
void SelectItembyMouse(CPoint point);           // OpenGLControl.h
void COpenGLControl::SelectItembyMouse(CPoint point) // OpenGLControl.cpp
{
    GLuint selectBuf[BUFSIZE];
    GLint hits;
    GLint viewport[4];
    glGetIntegerv (GL_VIEWPORT, viewport);
    glSelectBuffer (BUFSIZE, selectBuf);
    glRenderMode (GL_SELECT);
    glInitNames();
    glPushName(0);
    glMatrixMode(GL_PROJECTION);
    glPushMatrix ();
    glLoadIdentity();
    gluPickMatrix ((GLfloat) point.x, (GLfloat) (viewport[3]- point.y), 1.0, 1.0, viewport);
    gluPerspective(FoVY, (float)viewport[2]/(float)viewport[3], Znear, Zfar);
    glMatrixMode(GL_MODELVIEW);
    DrawInMode(GL_SELECT);
    glMatrixMode(GL_PROJECTION);
    glPopMatrix ();
    hits = glRenderMode (GL_RENDER);
    ProcessHits (hits, selectBuf);
    glMatrixMode(GL_MODELVIEW);
    oglDrawScene();

    GetMousePositionInOgl(point.x, point.y, 1, m_LastMouseZDepth, m_LastMousePosX, m_LastMousePosY, m_LastMousePosZ);
}
```



3. Context Menu

□ Tương tự, thêm 1 menu ngữ cảnh mới

➤ Gọi tới hàm mới này trong các hàm sự kiện OnLButtonDown() và OnRButtonDown().

```
void COpenGLControl::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    m_fLastX = (float)point.x;
    m_fLastY = (float)point.y;

    SelectItembyMouse(point);

    CWnd::OnLButtonDown(nFlags, point);
}

void COpenGLControl::OnRButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    m_fLastX = (float)point.x;
    m_fLastY = (float)point.y;

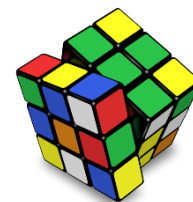
    SelectItembyMouse(point);

    CWnd::OnRButtonDown(nFlags, point);
}
```



Kết quả chạy:

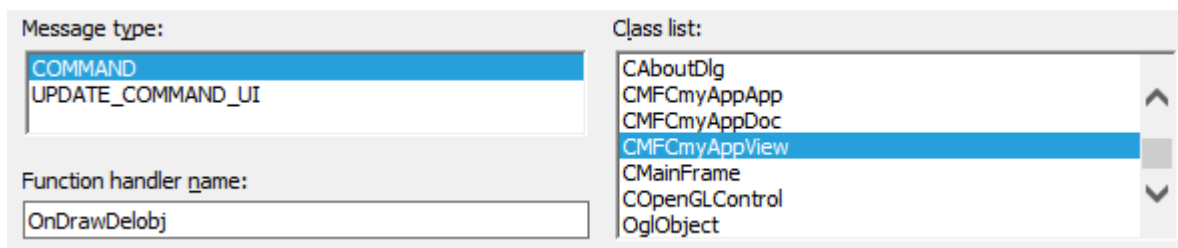
- Có thể chọn khối hình bằng cả 2 thao tác click chuột trái hoặc click chuột phải.



3. Context Menu

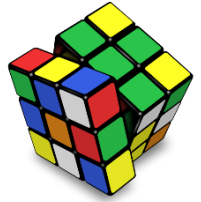
❑ Tương tự, thêm 1 menu ngữ cảnh mới

➤ Thêm hàm chức năng cho nút lệnh của menu mới.



```
void CMFCmyAppView::OnDrawDelobj()
{
    // TODO: Add your command handler code here
    if (!m_oglWindow.SelectedItem) return;
    for(int i=m_oglWindow.SelectedItem-1; i<m_oglWindow.ObjArray.size()-1; i++)
        m_oglWindow.ObjArray[i] = m_oglWindow.ObjArray[i+1];
    m_oglWindow.ObjArray.pop_back();

    m_oglWindow.SelectedItem = 0;
    m_oglWindow.oglDrawScene();
}
```



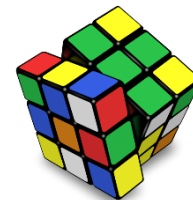
3. Context Menu

□ Tương tự, thêm 1 menu ngữ cảnh mới

➤ Cài đặt menu mới

```
void CMFCMyAppApp::PreLoadState() // File MFC<tên app>.cpp
{
    BOOL bNameValid;
    CString strName;
    bNameValid = strName.LoadString(IDS_MENU_UNSEL);
    ASSERT(bNameValid);
    GetContextMenuManager()->AddMenu(strName, IDR_POPUP_UNSEL);
    bNameValid = strName.LoadString(IDS_MENU_SEL);
    ASSERT(bNameValid);
    GetContextMenuManager()->AddMenu(strName, IDR_POPUP_SEL);
}
```

```
void CMFCMyAppView::OnContextMenu(CWnd* /* pWnd */, CPoint point) // File MFC...View.cpp
{
    #ifndef SHARED_HANDLERS
        if (!m_oglWindow.SelectedItem)
            theApp.GetContextMenuManager()->ShowPopupMenu(IDR_POPUP_UNSEL, point.x, point.y, this, TRUE);
        else
            theApp.GetContextMenuManager()->ShowPopupMenu(IDR_POPUP_SEL, point.x, point.y, this, TRUE);
    #endif
}
```



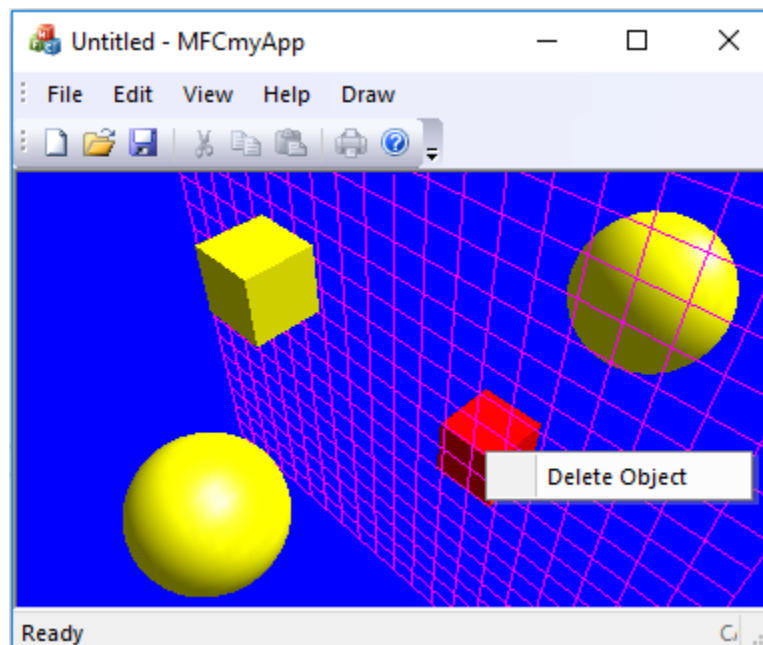
3. Context Menu

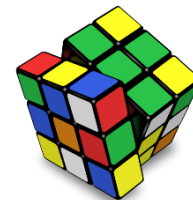
☐ Tương tự, thêm 1 menu ngữ cảnh mới



Kết quả chạy:

- Click chuột phải vào không gian trống thì xuất hiện menu `IDR_POPUP_UNSEL`
- Click chuột phải vào 1 khối hình thì khối hình đó được chọn và xuất hiện menu `IDR_POPUP_SEL`.
Chọn lệnh `Delete Object` thì khối hình đang được chọn sẽ bị xóa.

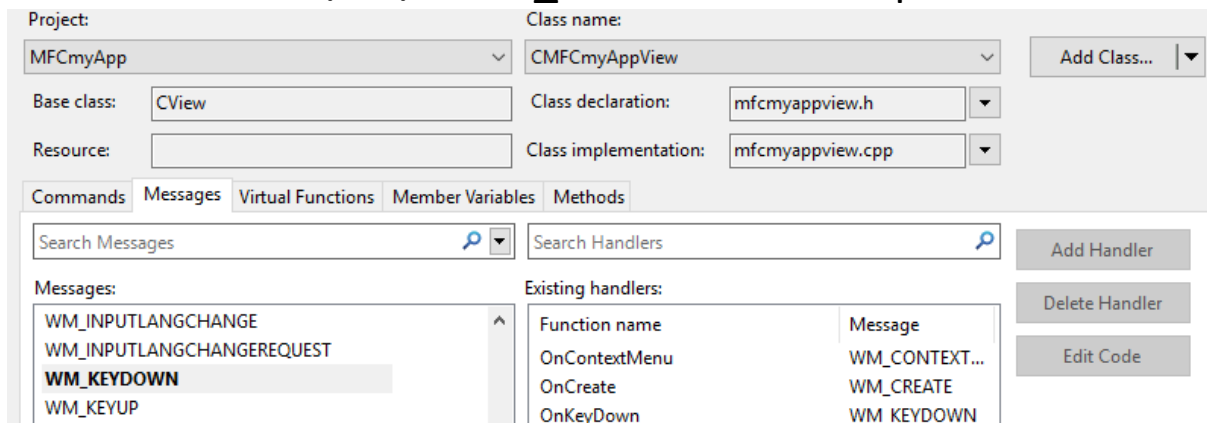




4. Keyboard

❑ Hàm OnKeyDown()

- Thêm hàm bắt sự kiện WM_KEYDOWN vào lớp C...View



- Viết code cho hàm OnKeyDown() của lớp C...View

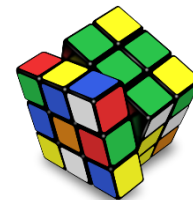
```
void CMFCMyAppView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    // TODO: Add your message handler code here and/or call default
    if (nChar == VK_DELETE)
        OnDrawDelobj();

    CView::OnKeyDown(nChar, nRepCnt, nFlags);
}
```



Kết quả chạy:

- Chọn 1 khối hình và nhấn phím Delete, khối hình biến mất.
- Khi không có khối hình nào được chọn thì phím Delete không có tác dụng.



4. Keyboard

❑ Hàm OnKeyDown()

- Sửa code hàm OnKeyDown() để thêm phím chức năng

```
void CMFCmyAppView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    // TODO: Add your message handler code here and/or call default
    if (nChar == VK_DELETE)
        OnDrawDelobj();
    else if ((nChar == 0x41) && (GetKeyState(VK_SHIFT) & 0x8000))
        OnDrawAddobj();
    else if ((nChar == 0x44) && (GetKeyState(VK_SHIFT) & 0x8000))
        OnDrawDelobj();

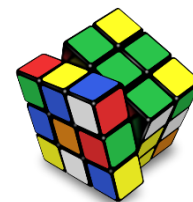
    CView::OnKeyDown(nChar, nRepCnt, nFlags);
}
```

// Tham khảo bảng các **Virtual-Key Codes** trên trang MSDN tại địa chỉ sau:
<https://msdn.microsoft.com/en-us/library/windows/desktop/dd375731.aspx>



Kết quả chạy:

- Nhấn tổ hợp phím Shift và A, hộp thoại Add Object hiện ra.
- Nhấn tổ hợp phím Shift và D, khối hình đang chọn sẽ bị xóa.

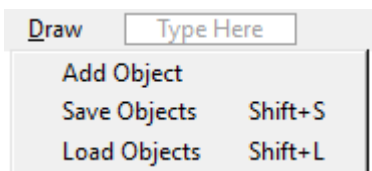


4. Keyboard

☐ Phím tắt cho Menu

➤ Với các lệnh có trong menu, ta có thể cài đặt phím tắt như sau:

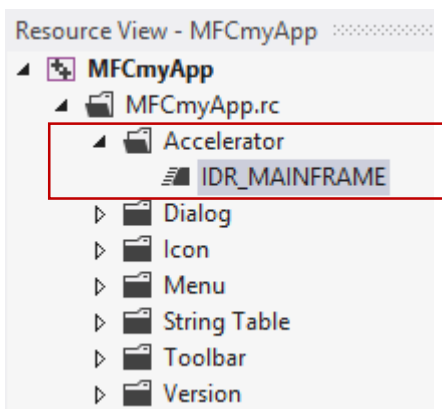
- Thêm phần hướng dẫn phím tắt vào caption



Properties :

Caption	Save Objects\tShift+S
Caption	Load Objects\tShift+L

- Khai báo phím tắt trong bảng Accelerator

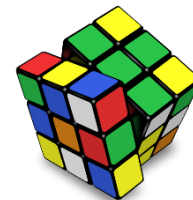


ID	Modifier	Key	Type
ID_DRAW_SAVEOBJECTS	Shift	S	VIRTKEY
ID_DRAW_LOADOBJECTS	Shift	L	VIRTKEY

//Chú ý: - Không khai báo trùng tổ hợp phím tắt giữa các lệnh khác nhau.

Kiểm tra bằng cách: Mở menu IDR_MAINFRAME -> Chuột phải -> Check Mnemonics.

- Tránh cài đặt trùng lặp hoặc mâu thuẫn giữa bảng Accelerator và hàm OnKeyDown()



4. Keyboard

☐ Phím tắt cho Menu



Kết quả chạy:

- Nhấn tổ hợp phím Shift và S, hộp thoại Save Objects hiện ra.
- Nhấn tổ hợp phím Shift và L, hộp thoại Load Objects hiện ra.

//Lưu ý: Các cài đặt mới với accelerator và menu có thể chưa được cập nhật tức thì.

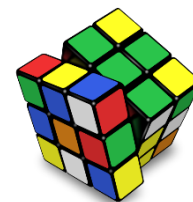
- Nguyên nhân là do chương trình tự động lưu một số thiết lập như: menu, trạng thái cửa sổ... vào Registry và đọc ra dùng cho các lần chạy sau.
- Vì vậy, để kiểm tra ngay các thay đổi cần thêm 1 dòng lệnh sau:

```
BOOL CMFCmyAppApp::InitInstance()
{
    ...

    SetRegistryKey(_T("Local AppWizard-Generated Applications"));
    LoadStdProfileSettings(4); // Load standard INI file options (including MRU)

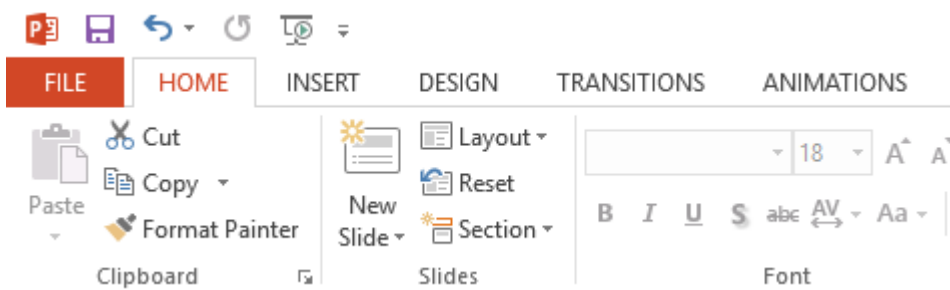
    // Add the Unregister() after LoadStdProfileSetting() to prevent the app
    // from saving state to and loading state from Registry
    Unregister();

    ...
}
```

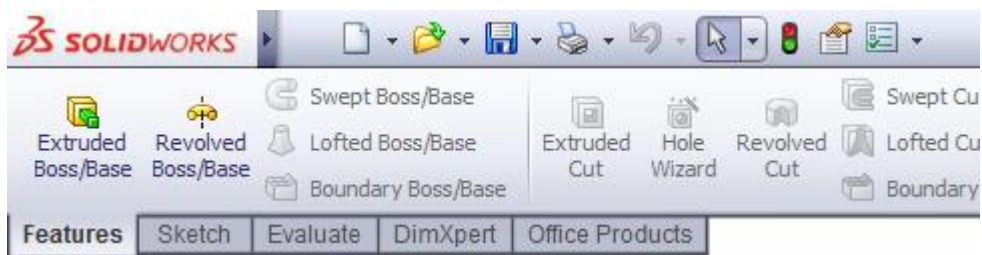


5. Ribbon Menu

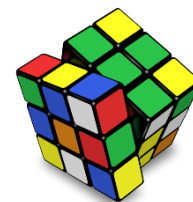
☐ Menu kiểu ribbon



Powerpoint 2013



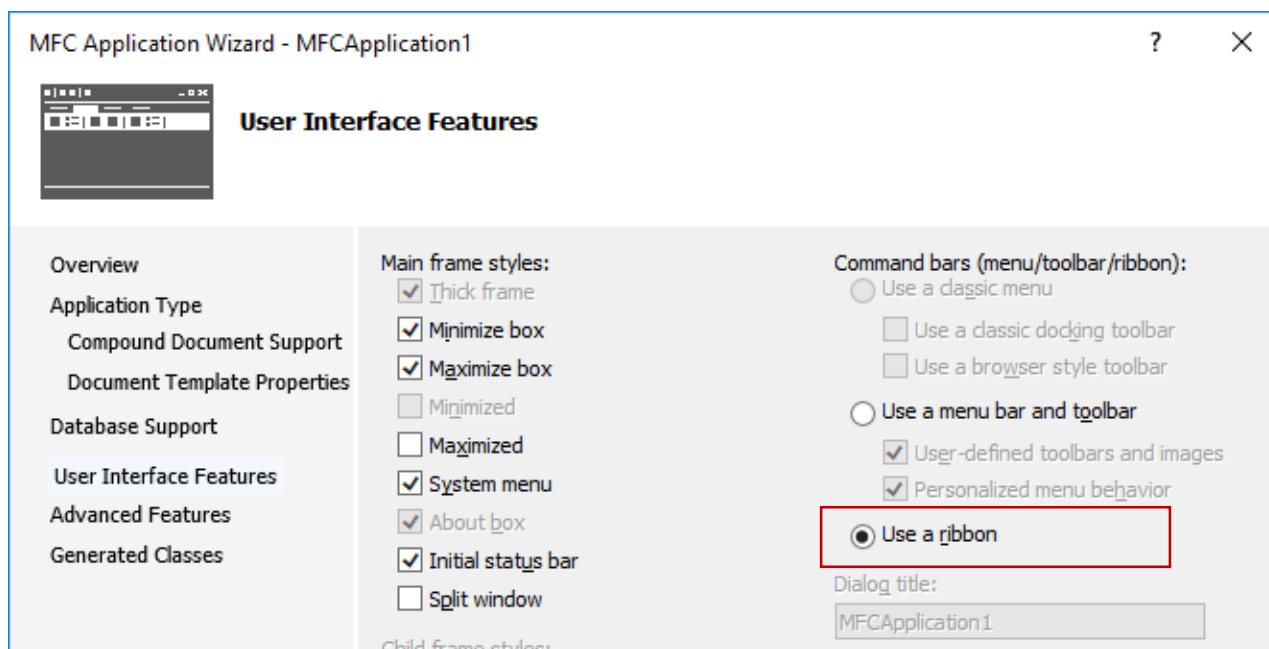
Solidworks 2013

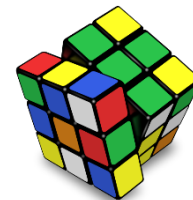


5. Ribbon Menu

❑ Cài đặt ribbon menu cho project mới

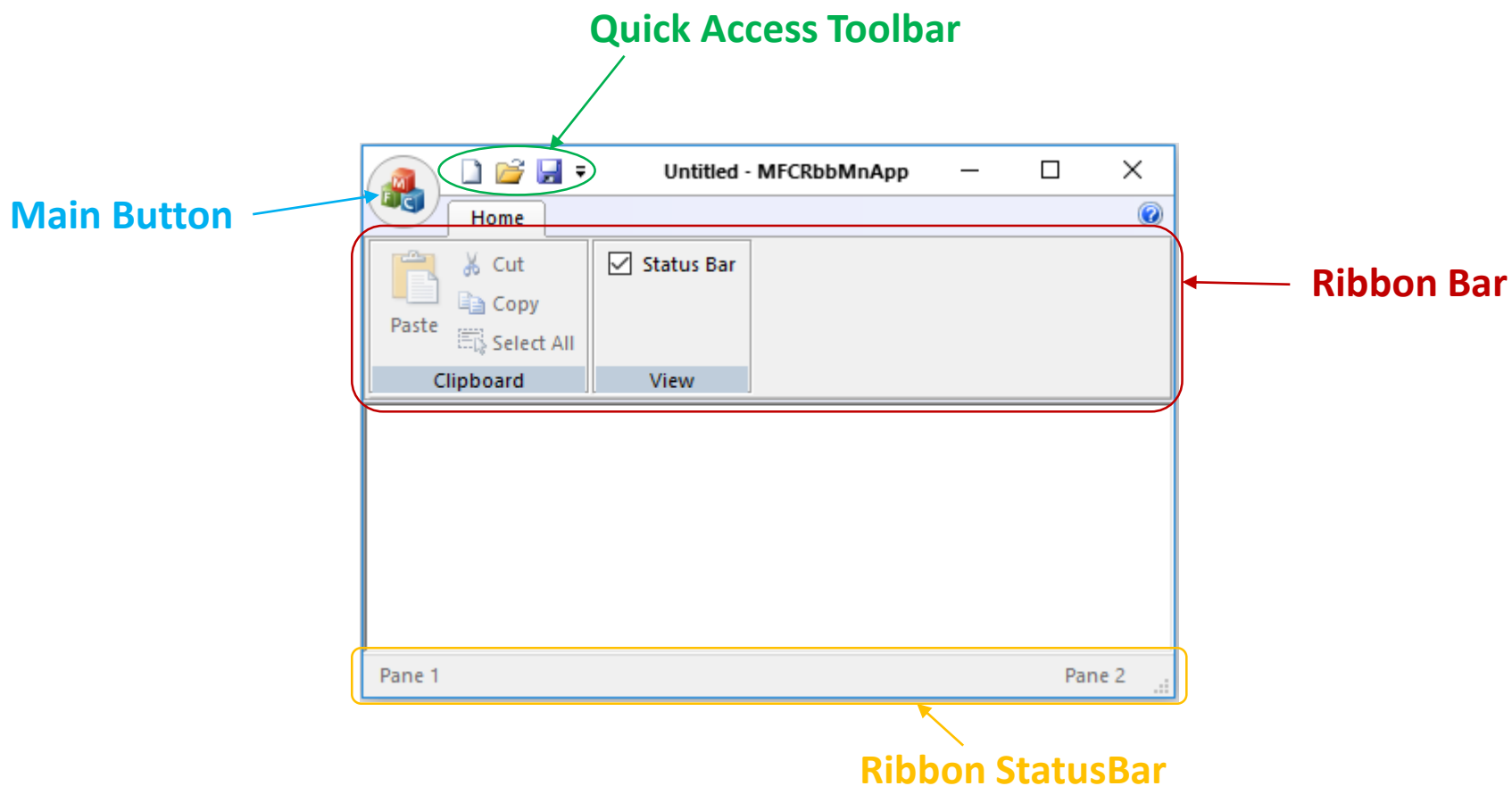
- Tạo project mới và cài đặt ribbon menu ngay từ đầu bằng cách đánh dấu vào lựa chọn dưới đây:

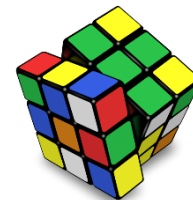




5. Ribbon Menu

- ❑ Cài đặt ribbon menu cho project mới
 - Quan sát kết quả chạy





5. Ribbon Menu

❑ Cài đặt ribbon menu cho project mới

➤ Quan sát lớp MainFrame

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CFrameWndEx::OnCreate(lpCreateStruct) == -1)
        return -1;

    BOOL bNameValid;

    m_wndRibbonBar.Create(this);
    m_wndRibbonBar.LoadFromResource(IDR_RIBBON);

    if (!m_wndStatusBar.Create(this))
    {
        TRACE0("Failed to create status bar\n");
        return -1;    // fail to create
    }

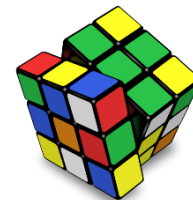
    CString strTitlePane1;
    CString strTitlePane2;
    bNameValid = strTitlePane1.LoadString(IDS_STATUS_PANE1);
    ASSERT(bNameValid);
    bNameValid = strTitlePane2.LoadString(IDS_STATUS_PANE2);
    ASSERT(bNameValid);
    m_wndStatusBar.AddElement(new CMFCRibbonStatusBarPane(ID_STATUSBAR_PANE1, strTitlePane1, TRUE), strTitlePane1);
    m_wndStatusBar.AddExtendedElement(new CMFCRibbonStatusBarPane(ID_STATUSBAR_PANE2, strTitlePane2, TRUE), strTitlePane2);
    ...
}
```

ID của Ribbon Bar

// Khởi tạo Ribbon Bar

// Khởi tạo Ribbon StatusBar

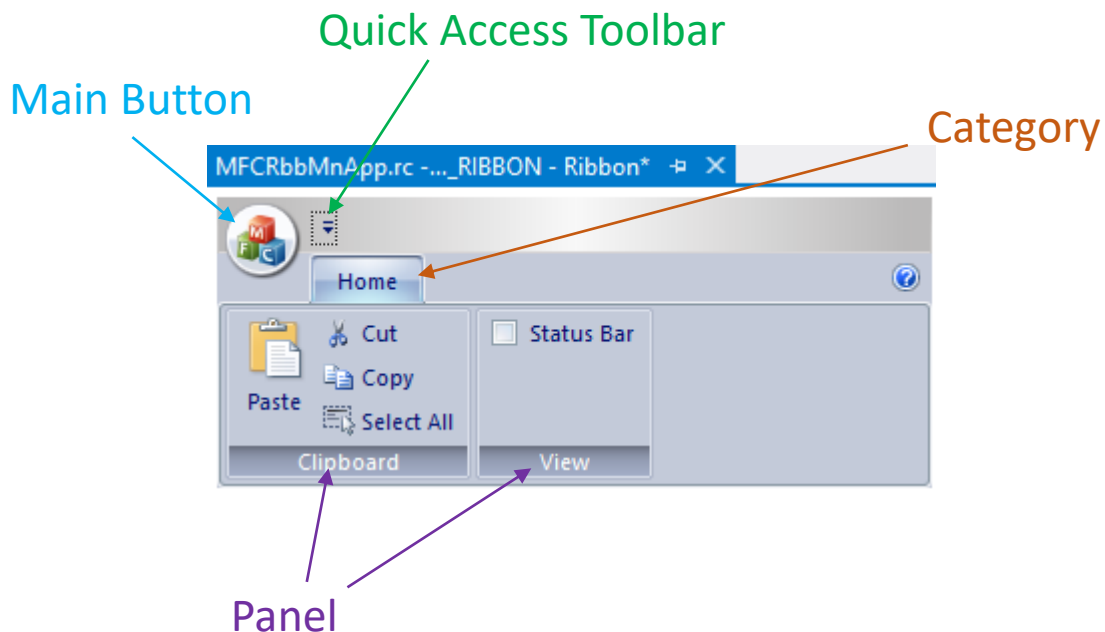
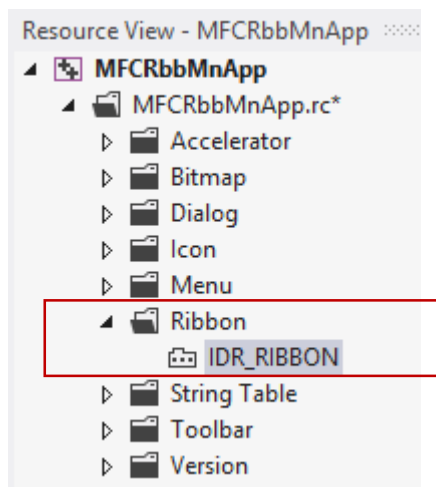
// Thêm 2 Status Pane vào Ribbon StatusBar

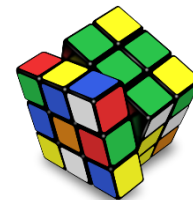


5. Ribbon Menu

❑ Thiết kế ribbon menu

- Mở Ribbon Menu (Resource View -> Ribbon -> IDR_RIBBON)

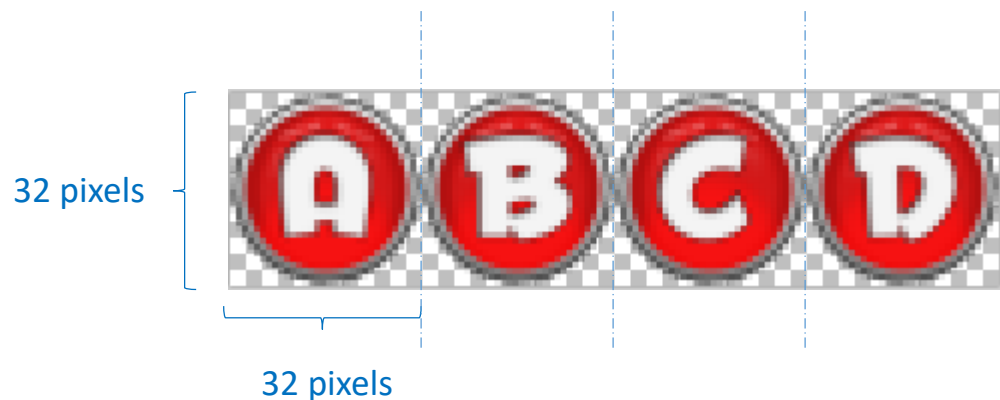




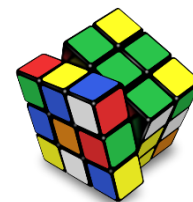
5. Ribbon Menu

❑ Thiết kế ribbon menu

- Chuẩn bị icon cho các lệnh trên ribbon menu.
 - Tạo 2 file bitmap (.bmp) có kích thước như sau:



Bit depth: 32 bit
(24 bit màu + 8 bit alpha)
↓
(độ trong suốt)

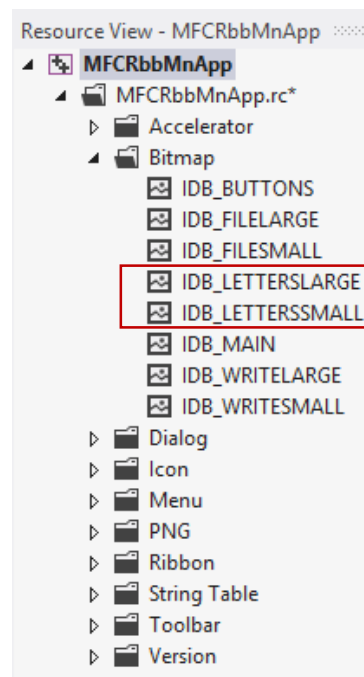
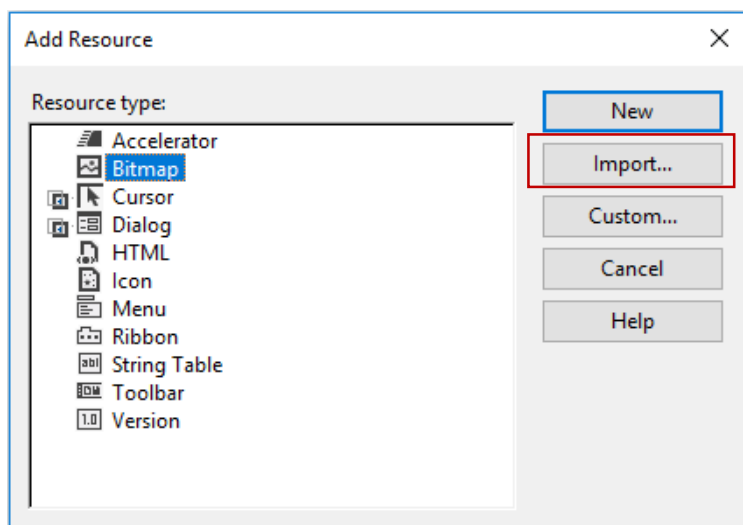


5. Ribbon Menu

❑ Thiết kế ribbon menu

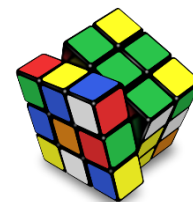
➤ Chuẩn bị icon cho các lệnh trên ribbon menu.

- Copy các file bitmap vào thư mục: <tên project>\<tên project>\res
- Thêm các file bitmap vào project bằng Add Resource -> Bitmap -> Import



Properties:

Filename	res\LettersLarge.bmp
ID	IDB_LETTERS LARGE
Filename	res\LettersSmall.bmp
ID	IDB_LETTERS SMALL



5. Ribbon Menu

❑ Thiết kế ribbon menu

- Thêm Category vào menu bằng cách kéo thả từ Toolbox vào Ribbon Bar

The screenshot illustrates the steps to add a new category to a ribbon menu. On the left, the **Toolbox** window shows the **Category** item selected. A red arrow indicates this item being dragged to the **Letters** tab in the **MFCRbbMnApp.rc - ..._RIBBON - Ribbon*** window. A second red arrow points to the **Properties** window, which displays the configuration for the **Letters** category.

Properties

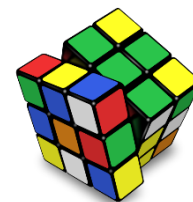
Category IRibbonCategoryEditor

Appearance

Caption	Letters
Keys	L

Images

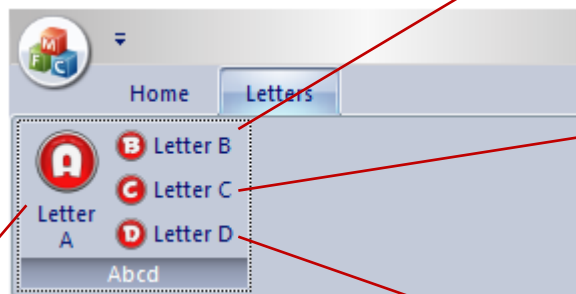
Large Images	IDB_LETTERS LARGE
Small Images	IDB_LETTERS SMALL



5. Ribbon Menu

❑ Thiết kế ribbon menu

- Thêm Panel, Button... vào Category vừa tạo



Caption	Letter B
Image Index	1
Large Image Index	-1
ID	ID_BUTTON_B
Prompt	Do command B\nButton B

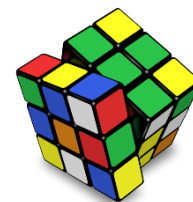
Caption	Letter C
Image Index	2
Large Image Index	-1
ID	ID_BUTTON_C
Prompt	Do command C\nButton C

Caption	Letter D
Image Index	3
Large Image Index	-1
ID	ID_BUTTON_D
Prompt	Do command D\nButton D

Appearance	
Always Large	False
Caption	Letter A
Image Index	0
Keys	
Large Image Index	0
Split Mode	True
Behavior	
Menu Items	Empty
Misc	
(Name)	ID_BUTTON_A (Button)
ID	ID_BUTTON_A
Prompt	Do command A\nButton A

Image Collection

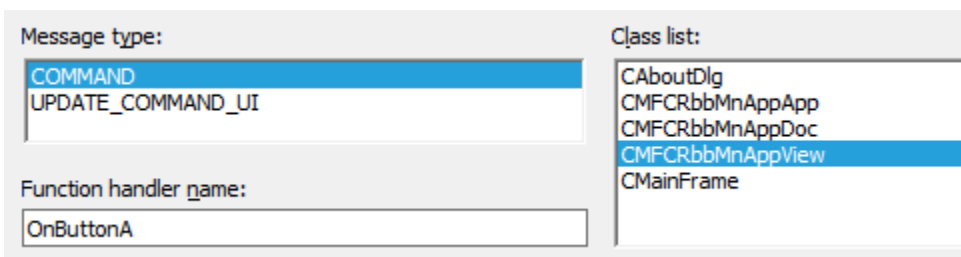
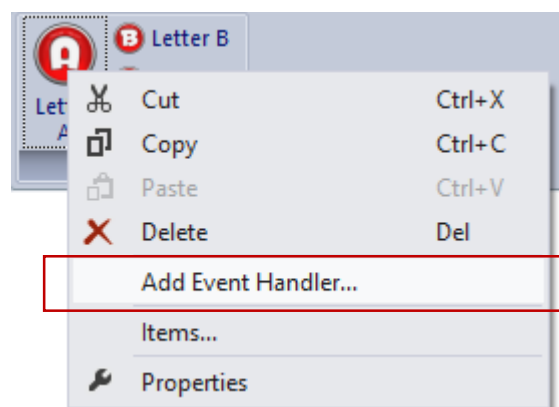




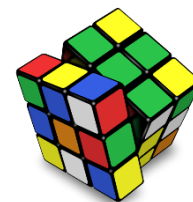
5. Ribbon Menu

❑ Thiết kế ribbon menu

- Thêm hàm sự kiện cho các nút lệnh



```
void CMFCRbbMnAppView::OnButtonA()
{
    MessageBox(_T("This is letter A"),_T("Information"),MB_OK+MB_ICONINFORMATION);
}
void CMFCRbbMnAppView::OnButtonB()
{
    MessageBox(_T("This is letter B"),_T("Information"),MB_OK+MB_ICONINFORMATION);
}
void CMFCRbbMnAppView::OnButtonC()
{
    MessageBox(_T("This is letter C"),_T("Information"),MB_OK+MB_ICONINFORMATION);
}
void CMFCRbbMnAppView::OnButtonD()
{
    MessageBox(_T("This is letter D"),_T("Information"),MB_OK+MB_ICONINFORMATION);
}
```



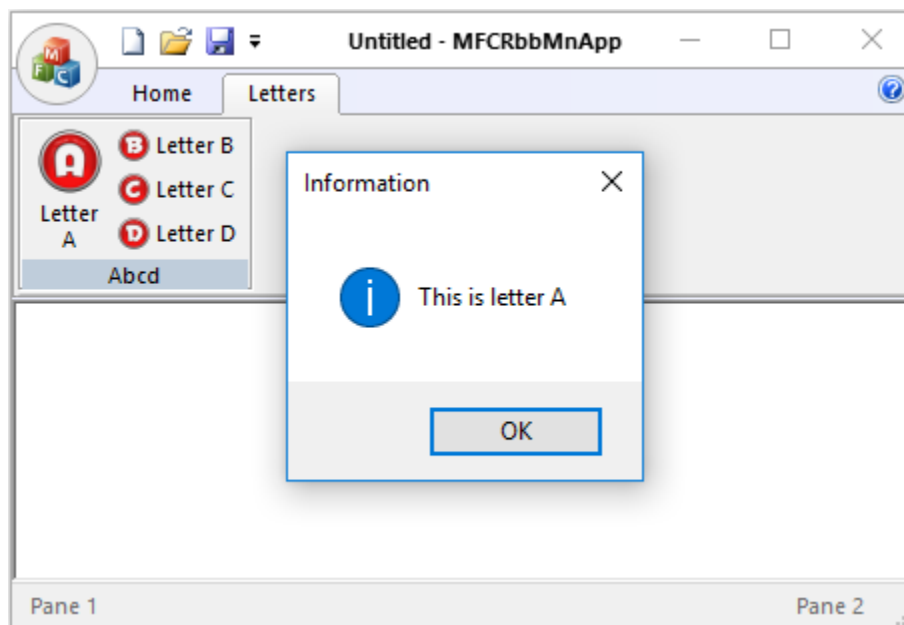
5. Ribbon Menu

❑ Thiết kế ribbon menu

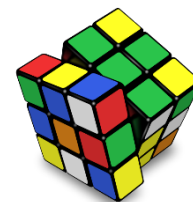


Kết quả chạy:

- Ribbon menu hiển thị như thiết kế. Khi đưa chuột qua các button thì xuất hiện Tooltip.
- Click vào các nút lệnh thì hàm chức năng được thực hiện như thiết kế.



//Lưu ý: Trong 1 số trường hợp các cập nhật với menu chưa thể hiện ngay thì cần thêm lệnh `Unregister()` như hướng dẫn ở phần trước



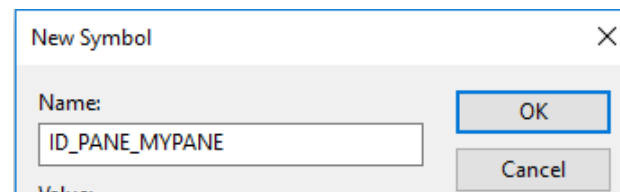
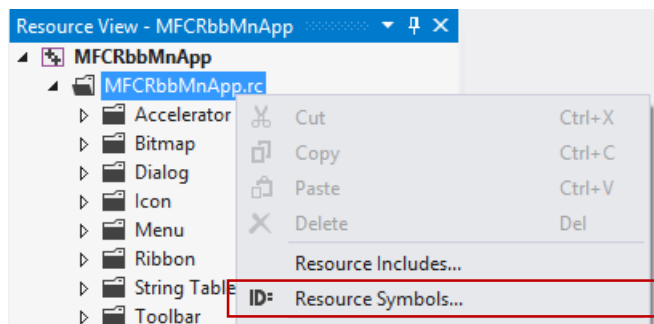
5. Ribbon Menu

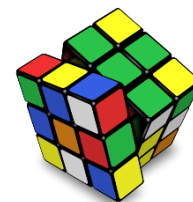
❑ Thiết kế ribbon menu

- Thêm vào lớp MainFrame một Dockable Pane như sau.

```
| CDockablePane m_MyPane; // File MainFrm.h
|
| int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct) // File MainFrm.cpp
| {
|     ...
|     if (!m_MyPane.Create(_T("My Pane"), this, CRect(0, 0, 300, 600), TRUE, ID_PANE_MYPANE,
|         WS_CHILD | WS_VISIBLE | WS_CLIPSIBLINGS | WS_CLIPCHILDREN | CBRS_LEFT | CBRS_FLOAT_MULTI))
|     {
|         TRACE0("Failed to create My Pane\n");
|         return FALSE; // failed to create
|     }
|     m_MyPane.EnableDocking(CBRS_LEFT|CBRS_RIGHT);
|     DockPane(&m_MyPane);
|     ...
| }
```

//Để tạo ID mới, ta mở Resource view -> Chuột phải -> Resource Symbols -> New...

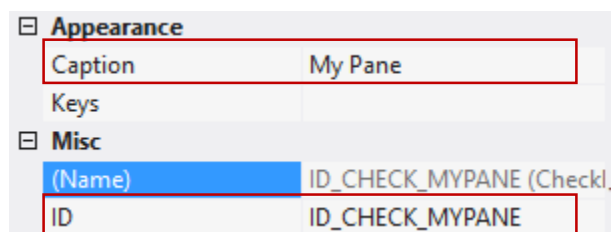
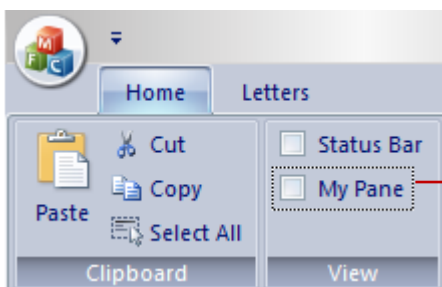




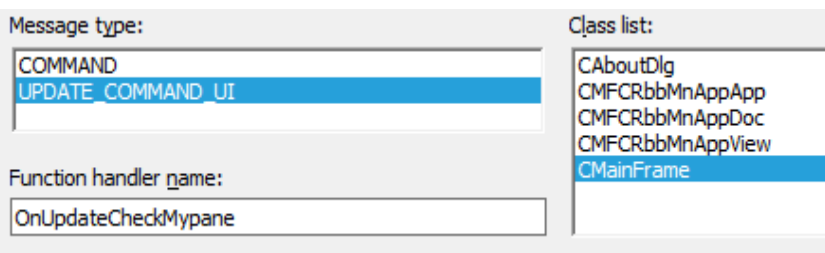
5. Ribbon Menu

❑ Thiết kế ribbon menu

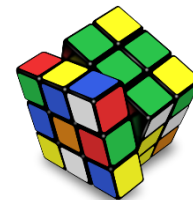
- Thêm vào Ribbon bar một Checkbox và viết code cho các hàm sự kiện của nó



```
void CMainFrame::OnCheckMypane()  
{  
    // TODO: Add your command handler code here  
    if (m_MyPane.IsPaneVisible()) m_MyPane.ShowPane(FALSE, FALSE, FALSE);  
    else  
    {  
        m_MyPane.ShowPane(TRUE, FALSE, FALSE);  
        m_MyPane.SetAutoHideMode(FALSE, CBRS_LEFT);  
    }  
}
```



```
void CMainFrame::OnUpdateCheckMypane(CCmdUI *pCmdUI)  
{  
    // TODO: Add your command update UI handler code here  
    pCmdUI->SetCheck(m_MyPane.IsPaneVisible());  
}
```

5. Ribbon Menu

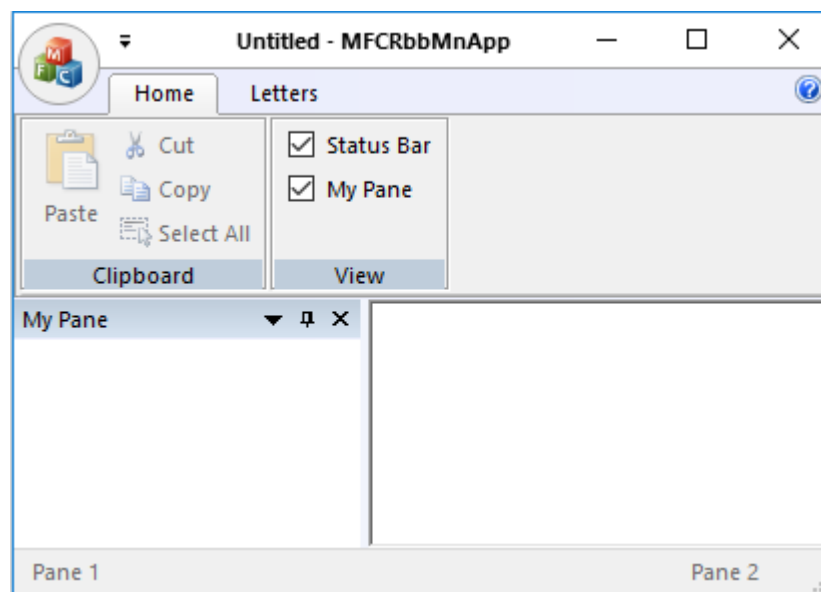
❑ Thiết kế ribbon menu

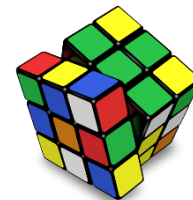
- Thêm vào Ribbon bar một Checkbox và viết code cho các hàm sự kiện của nó



Kết quả chạy:

- Khi chọn hay bỏ chọn ở checkbox My Pane thì sẽ tương ứng hiện hoặc ẩn ô cửa MyPane.



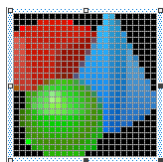


5. Ribbon Menu

❑ Thiết kế ribbon menu

➤ Chỉnh sửa Main Button

- Tạo file bitmap (24x24 pixels, 32 bit) và thêm vào project để làm hình ảnh cho Main Button

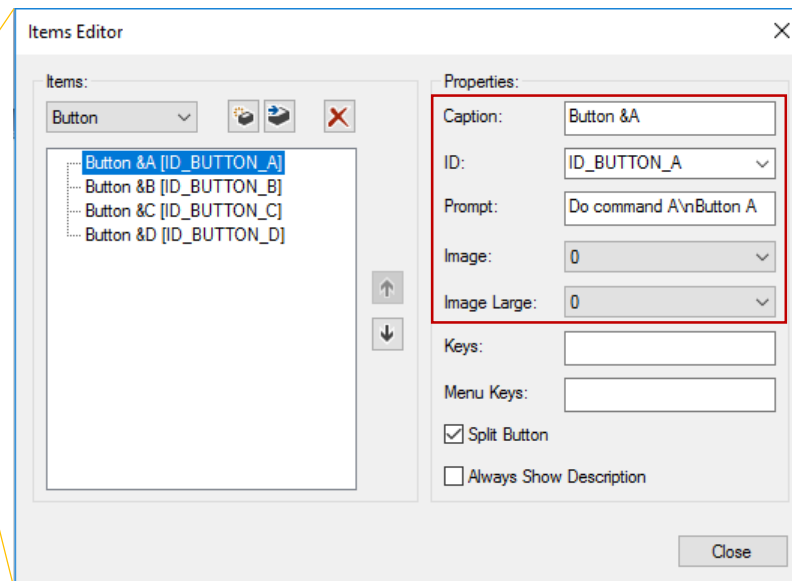


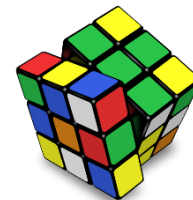
Filename	res\MainButton.bmp
ID	IDB_MAINBUTTON

- Thay đổi Properties của Main Button



Large Images	IDB_LETTERS LARGE
Small Images	IDB_LETTERS SMALL
Behavior	
Buttons	Collection
Main Items	Collection
Images	
HDPI Image	
Image	IDB_MAINBUTTON
Misc	
(Name)	IDC_STATIC (Application But
ID	IDC_STATIC
Prompt	
Recent List	
MRU Caption	Recent Documents
Use	False
Width	300





5. Ribbon Menu

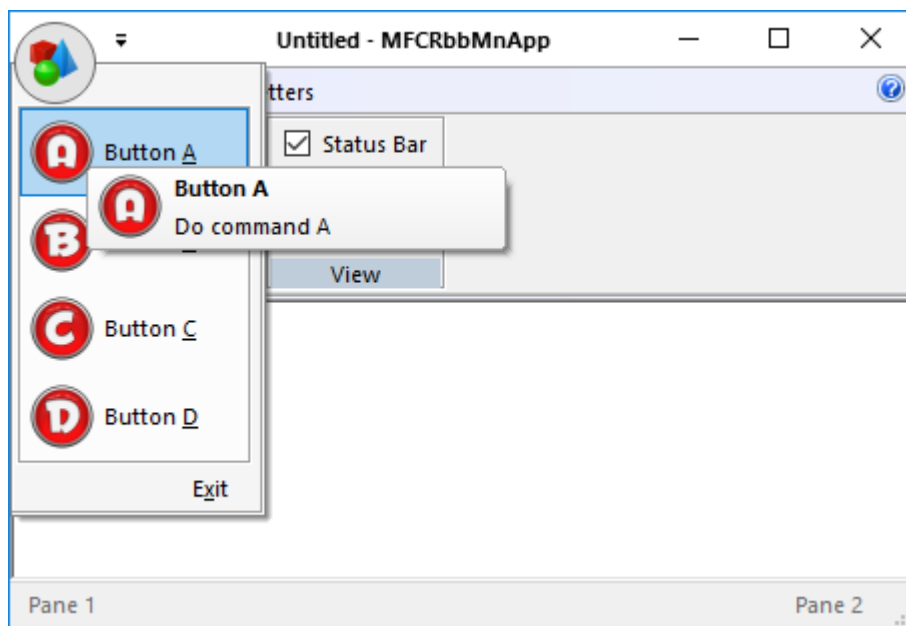
❑ Thiết kế ribbon menu

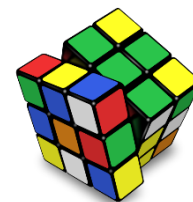
- Chỉnh sửa Main Button



Kết quả chạy:

- Nút Main Button thay đổi giao diện và chức năng như thiết kế.

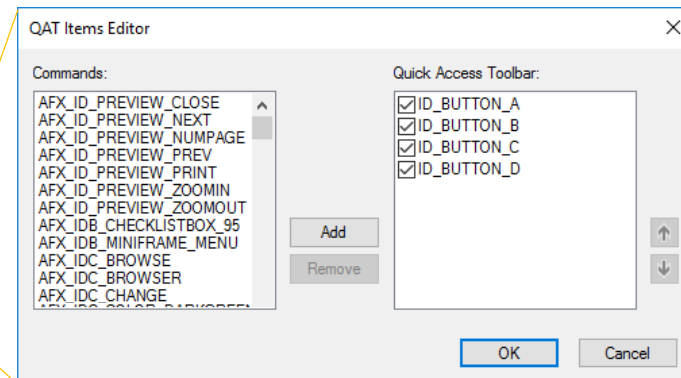
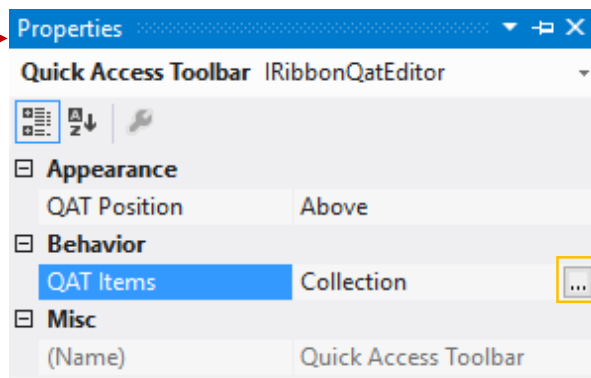
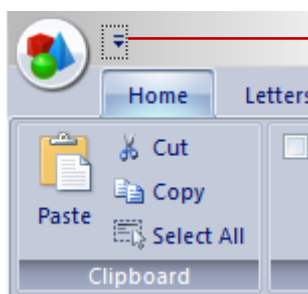




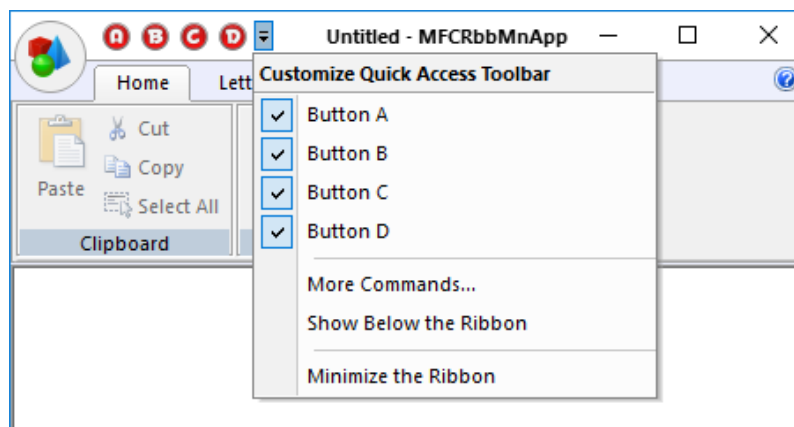
5. Ribbon Menu

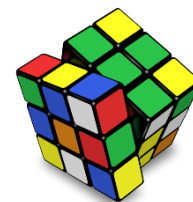
❑ Thiết kế ribbon menu

➤ Chỉnh sửa Quick Access Toolbar



➡ Kết quả chạy:





5. Ribbon Menu

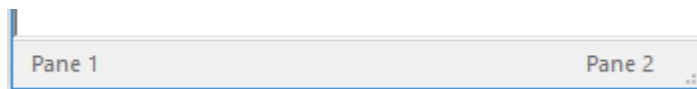
❑ Ribbon Status bar

SLIDE 58 OF 65 ENGLISH (UNITED STATES) NOTES COMMENTS 46%

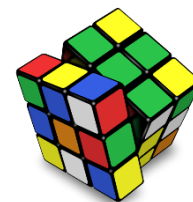
Ribbon Status bar của PowerPoint 2013

- Thêm Element vào Ribbon Status bar
 - Quan sát các lệnh thêm phần tử vào status bar

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ...
    m_wndStatusBar.AddElement(new CMFCRibbonStatusBarPane(ID_STATUSBAR_PANE1, strTitlePane1, TRUE), strTitlePane1);
    m_wndStatusBar.AddExtendedElement(new CMFCRibbonStatusBarPane(ID_STATUSBAR_PANE2, strTitlePane2, TRUE), strTitlePane2);
    ...
}
```



//Tương tự, có thể thêm vào Status bar các phần tử Ribbon khác như: button, slider...



5. Ribbon Menu

❑ Ribbon Status bar

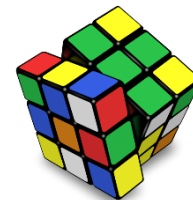
➤ Hiển thị thông tin ra Ribbon Status bar

- Thêm hàm có chức năng thay đổi nội dung hiển thị của Ribbon Status bar

```
void SetStatusbarContents(int status, double ProgressInPercents = 0); // File MainFrm.h
```

```
void CMainFrame::SetStatusbarContents(int status, double ProgInPerc) // File MainFrm.cpp
{
    CMFCRibbonStatusBarPane* pStatusPane2 = (CMFCRibbonStatusBarPane*) m_wndStatusBar.GetExElement(0);
    if (status) // Running
    {
        m_wndStatusBar.SetInformation(_T("Working"));
        CString strPercentage;
        strPercentage.Format(_T("%.2f %%"), ProgInPerc);
        pStatusPane2->SetText(LPCTSTR(strPercentage));
        pStatusPane2->Redraw();
    }
    else // Stop
    {
        m_wndStatusBar.SetInformation(_T("Ready"));
        pStatusPane2->SetText(_T(""));
        pStatusPane2->Redraw();
    }
}
```

// Truy cập tới phần tử mở rộng thứ nhất (ExElement[0]),
nếu có các phần tử khác thì chỉ số sẽ tăng dần



5. Ribbon Menu

❑ Ribbon Status bar

- Hiển thị thông tin ra Ribbon Status bar
 - Sửa code để thiết lập nội dung hiển thị ban đầu trên Statusbar

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    ...

    m_wndStatusBar.AddElement(new CMFCRibbonStatusBarPane(ID_STATUSBAR_PANE1, strTitlePane1, TRUE), strTitlePane1);

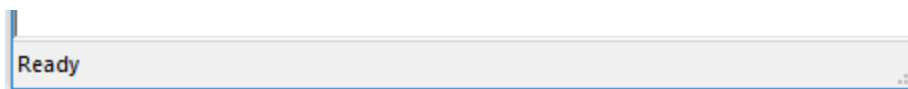
    CMFCRibbonStatusBarPane* pPane2 = new CMFCRibbonStatusBarPane(ID_STATUSBAR_PANE2, strTitlePane2, TRUE);
    pPane2->SetAlmostLargeText(_T("000.00 %"));
    m_wndStatusBar.AddExtendedElement(pPane2, strTitlePane2);

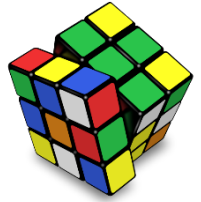
    SetStatusBarContents(0);

    ...
}
```



Kết quả chạy:





5. Ribbon Menu

❑ Ribbon Status bar

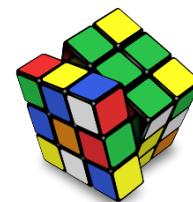
- Hiển thị thông tin ra Ribbon Status bar
 - Thêm hàm sự kiện OnTimer() vào lớp C...View và viết code như sau

// File C...View.h

```
double finished, total;  
bool isWorking;  
int timerID;
```

// File C...View.cpp

```
CMFCRbbMnAppView::CMFCRbbMnAppView()  
{  
    // TODO: add construction code here  
    finished = 0;  
    total = 20;  
    isWorking = false;  
    timerID = 1;  
}  
  
void CMFCRbbMnAppView::OnTimer(UINT_PTR nIDEvent)  
{  
    // TODO: Add your message handler code here and/or call default  
    CMainFrame* pFrame = (CMainFrame*) AfxGetMainWnd();  
    if (isWorking && finished < total)  
    {  
        pFrame->SetStatusBarContents(1, finished/total*100);  
        finished += 0.5;  
    }  
    else  
    {  
        finished = 0;  
        pFrame->SetStatusBarContents(0);  
        KillTimer(timerID);  
        isWorking = false;  
    }  
    CView::OnTimer(nIDEvent);  
}
```

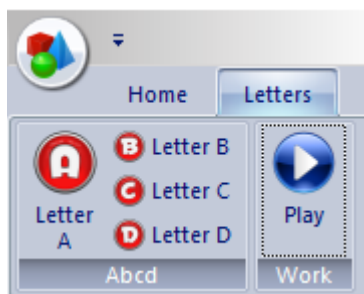



5. Ribbon Menu

❑ Ribbon Status bar

➤ Hiển thị thông tin ra Ribbon Status bar

- Thêm vào Ribbon bar một nút lệnh và viết code cho các hàm chức năng của nó như sau:



Message type:	Class list:
COMMAND	CAboutDlg
UPDATE_COMMAND_UI	CMFCRbbMnAppApp
	CMFCRbbMnAppDoc
	CMFCRbbMnAppView
	CMainFrame

Function handler name:

OnButtonPlay



```
void CMFCRbbMnAppView::OnButtonPlay()
{
    // TODO: Add your command handler code here
    isWorking = true;
    SetTimer(timerID, 100, NULL);
}
```

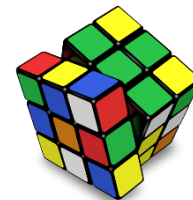
Message type:	Class list:
COMMAND	CAboutDlg
UPDATE_COMMAND_UI	CMFCRbbMnAppApp
	CMFCRbbMnAppDoc
	CMFCRbbMnAppView
	CMainFrame

Function handler name:

OnUpdateButtonPlay



```
void CMFCRbbMnAppView::OnUpdateButtonPlay(CCmdUI *pCmdUI)
{
    // TODO: Add your command update UI handler code here
    pCmdUI->Enable(!isWorking);
}
```



5. Ribbon Menu

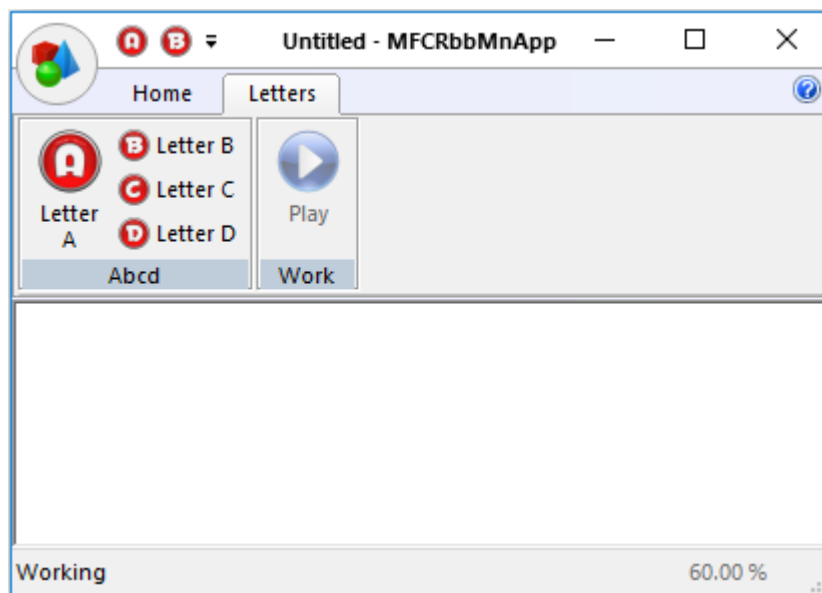
❑ Ribbon Status bar

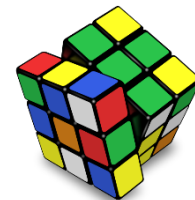
- Hiển thị thông tin ra Ribbon Status bar



Kết quả chạy:

- Ban đầu nút Play được hiển thị rõ ràng, Status bar hiển thị trạng thái Ready.
- Sau khi click vào nút Play, nút này bị mờ đi, Status bar hiển thị trạng thái Working và tỉ lệ phần trăm công việc tăng dần.
- Sau khi hoàn thành 100% công việc thì trạng thái nút Play và Status bar trở lại như ban đầu.





5. Ribbon Menu

❑ Cài đặt ribbon menu cho project sẵn có

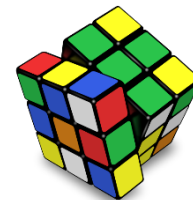
- Với các project không cài đặt ribbon menu từ đầu, ta có thể thay đổi từ menu cũ sang ribbon menu bằng những chỉnh sửa như sau:
 - Thay thế các đối tượng quản lý Menu, Toolbar, Statusbar cũ bằng các đối tượng mới

//File MainFrm.h

...

```
protected: // control bar embedded members
    //CMFCMenuBar      m_wndMenuBar;
    //CMFCToolBar      m_wndToolBar;
    //CMFCStatusBar    m_wndStatusBar;
    //CMFCToolBarImages m_UserImages;
    CMFCRibbonBar      m_wndRibbonBar;
    CMFCRibbonStatusBar m_wndStatusBar;
```

...



5. Ribbon Menu

❑ Cài đặt ribbon menu cho project sẵn có

➤ Với các project không cài đặt ribbon menu từ đầu, ta có thể thay đổi từ menu cũ sang ribbon menu bằng những chỉnh sửa như sau:

- Thay thế code trong hàm `MainFrame::OnCreate()` bằng đoạn code mới

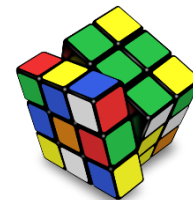
```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)    //File MainFrm.cpp
{
    if (CFrameWndEx::OnCreate(lpCreateStruct) == -1)
        return -1;
    BOOL bNameValid;
    m_wndRibbonBar.Create(this);
    m_wndRibbonBar.LoadFromResource(IDR_RIBBON);
    if (!m_wndStatusBar.Create(this))
    {
        TRACE0("Failed to create status bar\n");
        return -1;    // fail to create
    }

    CString strTitlePanel1;
    CString strTitlePanel2;
    bNameValid = strTitlePanel1.LoadString(IDS_STATUS_PANE1);
    ASSERT(bNameValid);
    bNameValid = strTitlePanel2.LoadString(IDS_STATUS_PANE2);
    ASSERT(bNameValid);
    m_wndStatusBar.AddElement(new CMFCRibbonStatusBarPane(ID_STATUSBAR_PANE1, strTitlePanel1, TRUE), strTitlePanel1);
    m_wndStatusBar.AddExtendedElement(new CMFCRibbonStatusBarPane(ID_STATUSBAR_PANE2, strTitlePanel2, TRUE), strTitlePanel2);
    ...
    // enable Visual Studio 2005 style docking window behavior
    CDockingManager::SetDockingMode(DT_SMART);
    // enable Visual Studio 2005 style docking window auto-hide behavior
    EnableAutoHidePanes(CBRS_ALIGN_ANY);
    // set the visual manager used to draw all user interface elements
    CMFCVisualManager::SetDefaultManager(RUNTIME_CLASS(CMFCVisualManagerVS2008));
    return 0;
}
```

→ ID của Ribbon Bar cần tạo thêm

→ // Có thể copy hàm `MainFrame::OnCreate()` từ một project cài đặt sẵn ribbon menu từ đầu

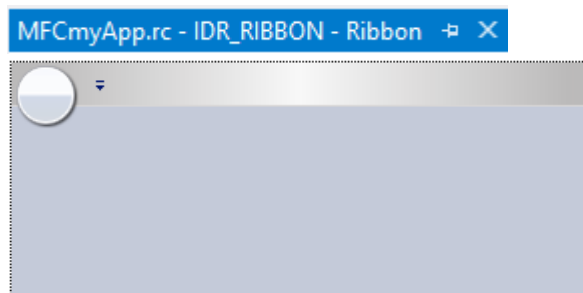
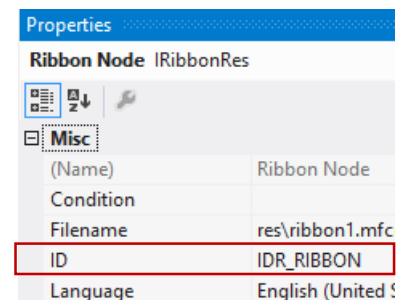
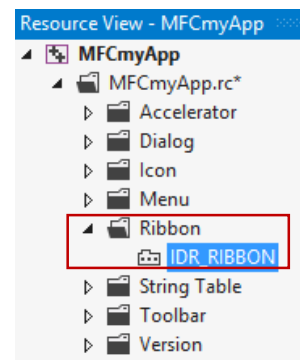
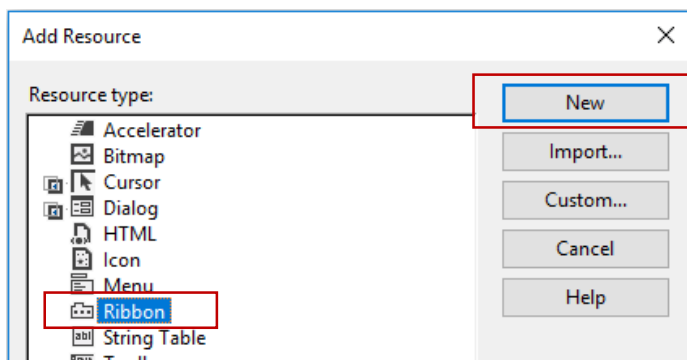
→ //Thêm code vào đây khi cần khởi tạo các ô cửa mới

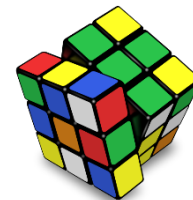


5. Ribbon Menu

❑ Cài đặt ribbon menu cho project sẵn có

- Với các project không cài đặt ribbon menu từ đầu, ta có thể thay đổi từ menu cũ sang ribbon menu bằng những chỉnh sửa như sau:
 - Thêm Ribbon bar vào project bằng: Add Resource -> Ribbon -> New...



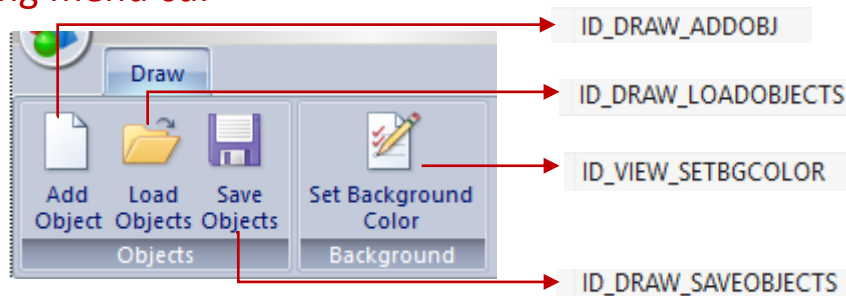


5. Ribbon Menu

❑ Cài đặt ribbon menu cho project sẵn có

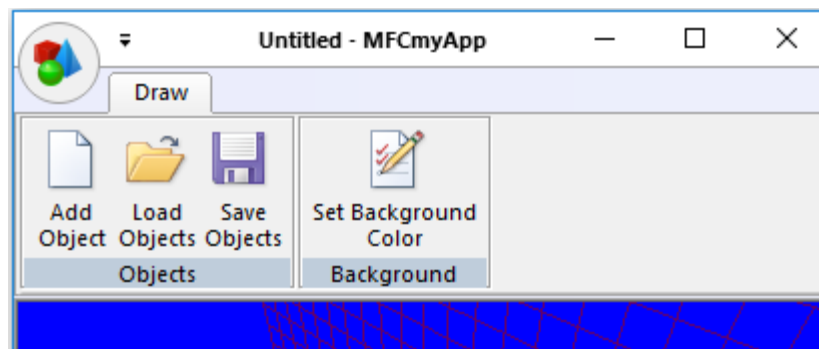
- Với các project không cài đặt ribbon menu từ đầu, ta có thể thay đổi từ menu cũ sang ribbon menu bằng những chỉnh sửa như sau:
 - Thiết kế Ribbon menu như đã hướng dẫn ở phần trước.

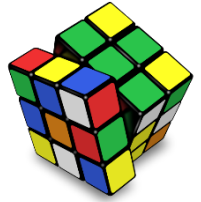
Lưu ý: Khi thiết lập ID cho các nút lệnh, chọn ID trùng với ID của các nút lệnh tương đương đã xây dựng trong menu cũ.



Kết quả chạy:

- Các nút lệnh trên Ribbon menu mới thực hiện chức năng tương tự như các nút lệnh của menu cũ

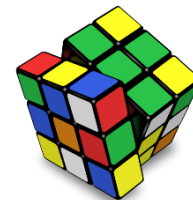




5. Ribbon Menu

□ Áp dụng

- Kết hợp những kiến thức đã được trình bày để xây dựng một ứng dụng tùy ý, dùng Ribbon menu (tìm hiểu và sử dụng thêm các control khác như editbox, combobox, slider...)



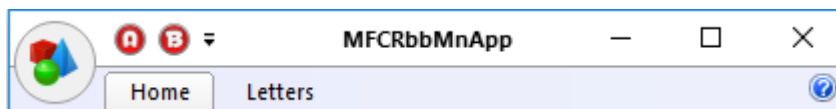
Mở rộng

❑ Title của ứng dụng

- Để bỏ “Untitled” khỏi tiêu đề, ta thêm dòng code sau

```
BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
    if( !CFrameWndEx::PreCreateWindow(cs) )
        return FALSE;
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs
    cs.style &= ~(LONG) FWS_ADDTOTITLE;

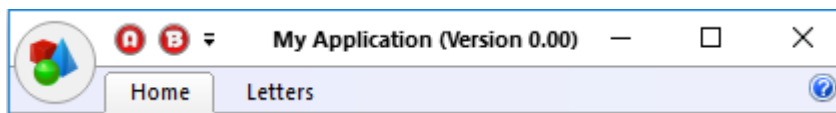
    return TRUE;
}
```

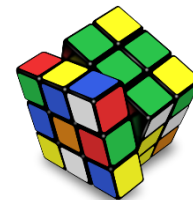


- Để thay hoàn toàn tiêu đề bằng một cụm từ cố định khác, ta sửa code như sau

```
BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
    if( !CFrameWndEx::PreCreateWindow(cs) )
        return FALSE;
    // TODO: Modify the Window class or styles here by modifying
    // the CREATESTRUCT cs
    cs.style = WS_OVERLAPPED | WS_CAPTION
        | WS_THICKFRAME | WS_MINIMIZEBOX | WS_MAXIMIZEBOX | WS_MAXIMIZE | WS_SYSMENU;
    cs.lpszName = _T("My Application (Version 0.00)");

    return TRUE;
}
```



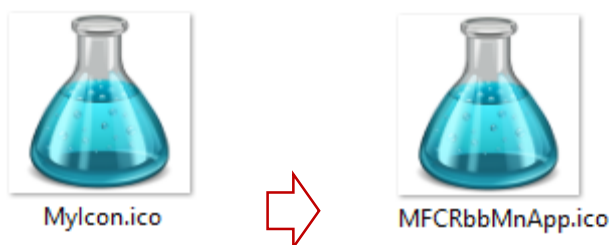


Mở rộng

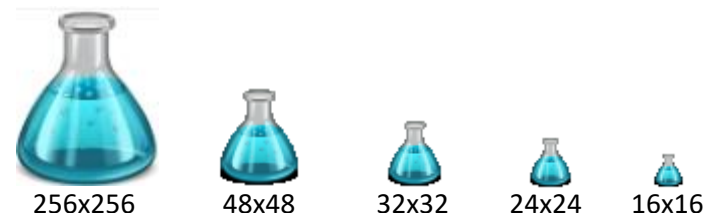
❑ Icon của ứng dụng

➤ Để thay đổi icon của ứng dụng ta thực hiện lần lượt các bước:

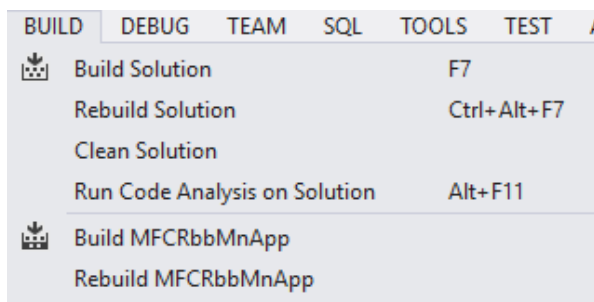
- Tạo 1 file icon mới (.ico), copy vào thư mục <tên project>\<tên project>\res và đổi tên để thay thế icon cũ của ứng dụng



//Lưu ý: file .ico cần phải chứa đủ các phiên bản kích thước cần thiết:

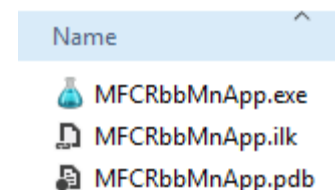


- Dựng lại Project hoặc Solution bằng cách chọn menu: Build -> Rebuild...



Kết quả chạy:

- Icon mới xuất hiện trên thanh Taskbar khi mở chương trình.
- Icon mới trở thành icon của file .exe trong thư mục Debug.



Hết Bài 12

