

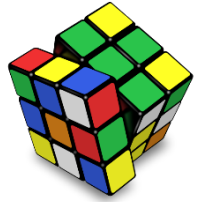
Ứng dụng MFC (Visual C++) trong mô phỏng Robot và hệ Cơ điện tử



Bài 5: Cài đặt và sử dụng OpenGL trong MFC

PHẠM MINH QUÂN

mquan.ph@gmail.com



Nội dung

1. Giới thiệu thư viện đồ họa OpenGL
2. Cài đặt OpenGL cho Visual Studio
3. Thiết lập và sử dụng OpenGL trong ứng dụng MFC
 - 3.1. Xây dựng lớp quản lý khung vẽ OpenGL*
 - 3.2. Sử dụng lớp quản lý khung vẽ OpenGL*
4. Tham khảo

1. Giới thiệu thư viện đồ họa OpenGL

- OpenGL (Open Graphics Library) là giao diện lập trình ứng dụng (API) đồ họa 3 chiều, hỗ trợ nhiều ngôn ngữ.
- OpenGL được thiết kế không phụ thuộc nền tảng phần cứng cũng như hệ điều hành (sử dụng tập hàm đồ họa chung chạy trên nhiều hệ thống khác nhau).
- Bao gồm khoảng 250 hàm. Các đối tượng hình học cơ bản là điểm, đường, đa giác ...
- Một số thư viện được xây dựng thêm để đơn giản hóa các công việc và cung cấp các hàm vẽ ở cấp cao hơn: GLU, GLUT, FreeGLUT, OpenGLUT...

2. Cài đặt thư viện OpenGL cho Visual Studio

❑ Bước 1: Tải về thư viện GLUT phiên bản mới nhất tại địa chỉ sau:
https://www.opengl.org/resources/libraries/glut/glut_downloads.php

❑ Bước 2: Sao chép các file vào thư mục cần thiết

- Copy các file “*.h” vào thư mục:

[Thư mục cài đặt Microsoft Visual Studio]\VC\include

- Copy các file “*.lib” vào thư mục:

[Thư mục cài đặt Microsoft Visual Studio]\VC\lib

- Copy các file “*.dll” vào thư mục:

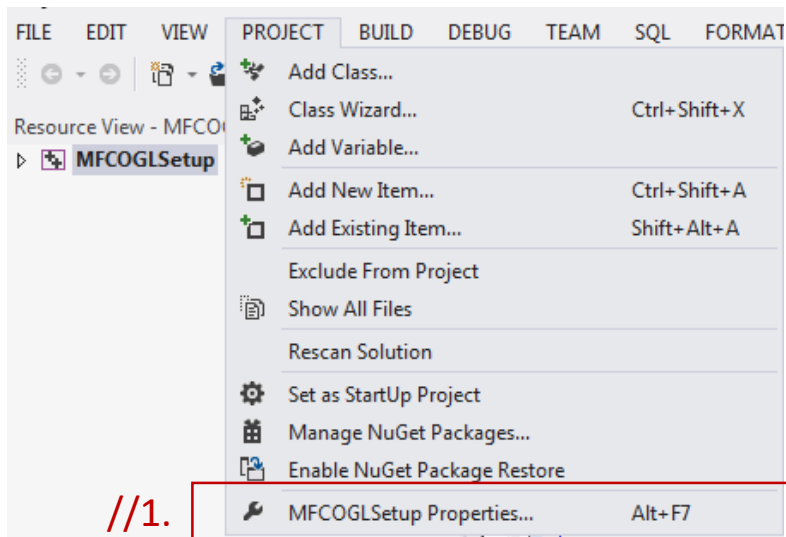
Với windows 32 bit: *Windows\System32*

Với windows 64 bit: *Windows\SysWOW64*

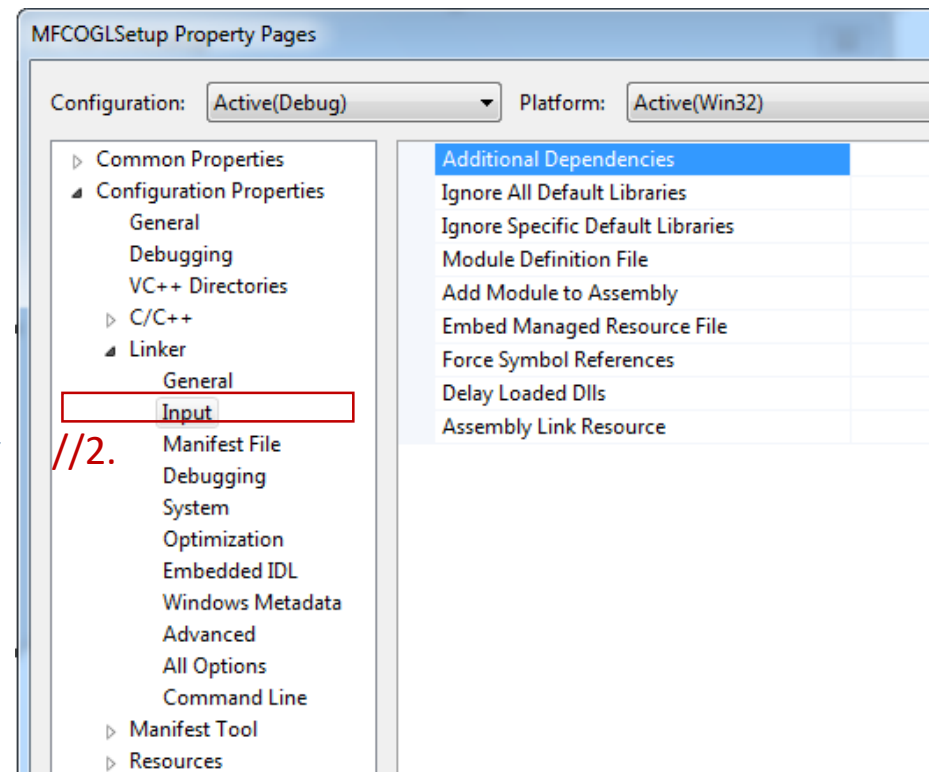
2. Cài đặt thư viện OpenGL cho Visual Studio

❑ Bước 3: Khi mở một project mới trong Visual Studio:

➤ Project -> ... Properties



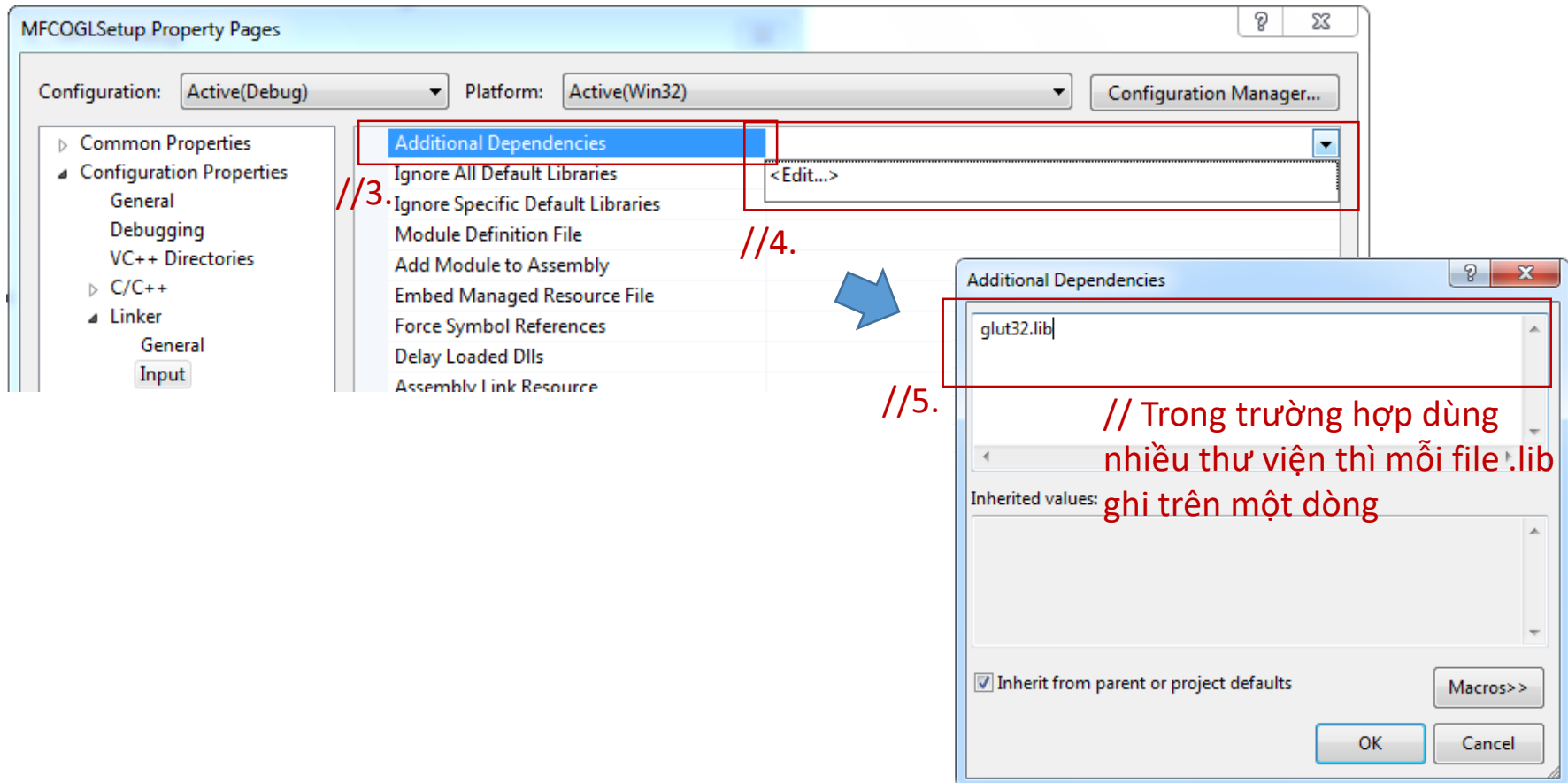
➤ Linker -> Input



2. Cài đặt thư viện OpenGL cho Visual Studio

❑ Bước 3: Khi mở một project mới trong Visual Studio:

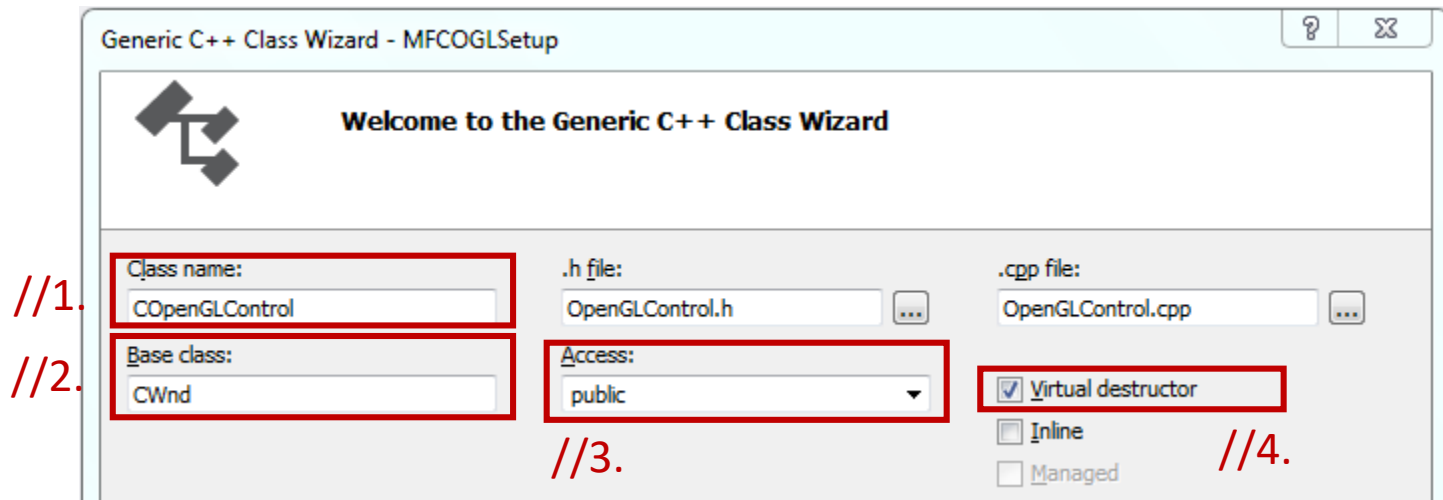
➤ Additional Dependencies -> <Edit...>



3. Thiết lập và sử dụng OpenGL trong MFC

3.1. Xây dựng lớp quản lý khung vẽ OpenGL

- ❑ Thêm lớp mới để quản lý khung vẽ OpenGL



3. Thiết lập và sử dụng OpenGL trong MFC

3.1. Xây dựng lớp quản lý khung vẽ OpenGL

□ Điền nội dung cho lớp mới

➤ Khai báo các biến

File OpenGLControl.h

```
#include "glut.h"

class COpenGLControl : public CWnd
{
public:
    // View information variables
    float    m_fLastX;
    float    m_fLastY;
    float    m_fPosX;
    float    m_fPosY;
    float    m_fZoom;
    float    m_fRotX;
    float    m_fRotY;

private:
    // Window information
    CWnd     *hWnd;
    HDC      hdc;
    HGLRC    hrc;
    int      m_nPixelFormat;
```


3. Thiết lập và sử dụng OpenGL trong MFC

3.1. Xây dựng lớp quản lý khung vẽ OpenGL

□ Điền nội dung cho lớp mới

➤ Thêm hàm oglCreate() (Hàm thiết lập khung vẽ)

File OpenGLControl.h

```
| void oglCreate(CRect rect, CWnd *parent);
```

File OpenGLControl.cpp

```
void COpenGLControl::oglCreate(CRect rect, CWnd *parent)
{
    CString className = AfxRegisterWndClass(CS_HREDRAW |
        CS_VREDRAW | CS_OWNDC, NULL,
        (HBRUSH)GetStockObject(BLACK_BRUSH), NULL);

    CreateEx(0, className, _T("OpenGL"), WS_CHILD | WS_VISIBLE |
        WS_CLIPSIBLINGS | WS_CLIPCHILDREN, rect, parent, 0);

    hWnd = parent;
}
```

3. Thiết lập và sử dụng OpenGL trong MFC

3.1. Xây dựng lớp quản lý khung vẽ OpenGL

□ Điền nội dung cho lớp mới

- Thêm hàm oglInitialize() (Hàm khởi tạo các thông số cho OpenGL)

File OpenGLControl.h

```
void oglInitialize(void);
```

File OpenGLControl.cpp

```
void COpenGLControl::oglInitialize(void)
{
    // Initial Setup:
    //
    static PIXELFORMATDESCRIPTOR pfd =
    {
        sizeof(PIXELFORMATDESCRIPTOR),
        1,
        PFD_DRAW_TO_WINDOW | PFD_SUPPORT_OPENGL | PFD_DOUBLEBUFFER,
        PFD_TYPE_RGBA,
        32, // bit depth
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        16, // z-buffer depth
        0, 0, 0, 0, 0, 0, 0,
    };

    // Get device context only once.
    hdc = GetDC()->m_hDC;
```

```
    // Pixel format.
    m_nPixelFormat = ChoosePixelFormat(hdc, &pfd);
    SetPixelFormat(hdc, m_nPixelFormat, &pfd);
    // Create the OpenGL Rendering Context.
    hrc = wglCreateContext(hdc);
    wglMakeCurrent(hdc, hrc);

    // Basic Setup:
    //
    // Set color to use when clearing the background.
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
    glClearDepth(1.0f);
    // Turn on backface culling
    glFrontFace(GL_CCW);
    glCullFace(GL_BACK);
    // Turn on depth testing
    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LEQUAL);
    // Send draw request
    OnDraw(NULL);
}
```

3. Thiết lập và sử dụng OpenGL trong MFC

3.1. Xây dựng lớp quản lý khung vẽ OpenGL

□ Điền nội dung cho lớp mới

- Thêm hàm oglDrawScene() (Hàm vẽ)

File OpenGLControl.h

```
void oglDrawScene(void);
```

File OpenGLControl.cpp

```
void COpenGLControl::oglDrawScene(void)
```

```
{
```

```
    // Clear color and depth buffer bits
```

```
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
```

```
    // Wireframe Mode
```

```
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
    glBegin(GL_QUADS);
```

```
        // Top Side
```

```
        glVertex3f( 1.0f, 1.0f,  1.0f);
```

```
        glVertex3f( 1.0f, 1.0f, -1.0f);
```

```
        glVertex3f(-1.0f, 1.0f, -1.0f);
```

```
        glVertex3f(-1.0f, 1.0f,  1.0f);
```

```
        // Bottom Side
```

```
        glVertex3f(-1.0f, -1.0f, -1.0f);
```

```
        glVertex3f( 1.0f, -1.0f, -1.0f);
```

```
        glVertex3f( 1.0f, -1.0f,  1.0f);
```

```
        glVertex3f(-1.0f, -1.0f,  1.0f);
```

```
        // Front Side
```

```
        glVertex3f( 1.0f,  1.0f,  1.0f);
```

```
        glVertex3f(-1.0f,  1.0f,  1.0f);
```

```
        glVertex3f(-1.0f, -1.0f,  1.0f);
```

```
        glVertex3f( 1.0f, -1.0f,  1.0f);
```

```
        // Back Side
```

```
        glVertex3f(-1.0f, -1.0f, -1.0f);
```

```
        glVertex3f(-1.0f,  1.0f, -1.0f);
```

```
        glVertex3f( 1.0f,  1.0f, -1.0f);
```

```
        glVertex3f( 1.0f, -1.0f, -1.0f);
```

```
        // Left Side
```

```
        glVertex3f(-1.0f, -1.0f, -1.0f);
```

```
        glVertex3f(-1.0f, -1.0f,  1.0f);
```

```
        glVertex3f(-1.0f,  1.0f,  1.0f);
```

```
        glVertex3f(-1.0f,  1.0f, -1.0f);
```

```
        // Right Side
```

```
        glVertex3f( 1.0f,  1.0f,  1.0f);
```

```
        glVertex3f( 1.0f, -1.0f,  1.0f);
```

```
        glVertex3f( 1.0f, -1.0f, -1.0f);
```

```
        glVertex3f( 1.0f,  1.0f, -1.0f);
```

```
    glEnd();
```

```
    // Swap buffers
```

```
    SwapBuffers(hdc);
```

```
}
```

3. Thiết lập và sử dụng OpenGL trong MFC

3.1. Xây dựng lớp quản lý khung vẽ OpenGL

□ Điền nội dung cho lớp mới

➤ Khai báo các hàm sự kiện (messages) *afx_msg*

File OpenGLControl.h

```
DECLARE_MESSAGE_MAP()  
afx_msg void OnPaint();  
afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);  
afx_msg void OnSize(UINT nType, int cx, int cy);  
afx_msg void OnMouseMove(UINT nFlags, CPoint point);  
afx_msg void OnDraw(CDC *pDC);
```

File OpenGLControl.cpp

```
BEGIN_MESSAGE_MAP(COpenGLControl, CWnd)  
    ON_WM_PAINT()  
    ON_WM_CREATE()  
    ON_WM_SIZE()  
    ON_WM_MOUSEMOVE()  
END_MESSAGE_MAP()
```

* Lưu ý: chỉ riêng hàm OnDraw là phải khai báo thủ công, 4 hàm còn lại có thể được thêm tự động bằng Class Wizard (như hướng dẫn trong Bài 1)

3. Thiết lập và sử dụng OpenGL trong MFC

3.1. Xây dựng lớp quản lý khung vẽ OpenGL

□ Điền nội dung cho lớp mới

- Định nghĩa các hàm sự kiện *afx_msg*

File OpenGLControl.cpp

```
...  
void COpenGLControl::OnPaint()  
{  
    //CPaintDC dc(this); // device context for painting  
    ValidateRect(NULL);  
}  
  
int COpenGLControl::OnCreate(LPCREATESTRUCT lpCreateStruct)  
{  
    if (CWnd::OnCreate(lpCreateStruct) == -1)  
        return -1;  
  
    oglInitialize();  
  
    return 0;  
}  
...
```

```
...  
void COpenGLControl::OnSize(UINT nType, int cx, int cy)  
{  
    CWnd::OnSize(nType, cx, cy);  
  
    if (0 >= cx || 0 >= cy || nType == SIZE_MINIMIZED) return;  
  
    // Map the OpenGL coordinates.  
    glViewport(0, 0, cx, cy);  
  
    // Projection view  
    glMatrixMode(GL_PROJECTION);  
  
    glLoadIdentity();  
  
    // Set our current view perspective  
    gluPerspective(35.0f, (float)cx / (float)cy, 0.01f, 2000.0f);  
  
    // Model view  
    glMatrixMode(GL_MODELVIEW);  
}  
...
```

3. Thiết lập và sử dụng OpenGL trong MFC

3.1. Xây dựng lớp quản lý khung vẽ OpenGL

□ Điền nội dung cho lớp mới

- Định nghĩa các hàm sự kiện *afx_msg*

File OpenGLControl.cpp

```
...  
void COpenGLControl::OnMouseMove(UINT nFlags, CPoint point)  
{  
    int diffX = (int)(point.x - m_fLastX);  
    int diffY = (int)(point.y - m_fLastY);  
    m_fLastX = (float)point.x;  
    m_fLastY = (float)point.y;  
  
    // Left mouse button  
    if (nFlags & MK_LBUTTON)  
    {  
        m_fRotX += (float)0.5f * diffY;  
        if ((m_fRotX > 360.0f) || (m_fRotX < -360.0f)) m_fRotX = 0.0f;  
  
        m_fRotY += (float)0.5f * diffX;  
        if ((m_fRotY > 360.0f) || (m_fRotY < -360.0f)) m_fRotY = 0.0f;  
    }  
  
    // Right mouse button  
    else if (nFlags & MK_RBUTTON)  
    {  
        m_fZoom -= (float)0.1f * diffY;  
    }  
}
```

```
...  
  
// Middle mouse button  
else if (nFlags & MK_MBUTTON)  
{  
    m_fPosX += (float)0.05f * diffX;  
    m_fPosY -= (float)0.05f * diffY;  
}  
  
if (nFlags)  
{  
    OnDraw(NULL);  
    oglDrawScene();  
}  
  
CWnd::OnMouseMove(nFlags, point);  
}
```

3. Thiết lập và sử dụng OpenGL trong MFC

3.1. Xây dựng lớp quản lý khung vẽ OpenGL

□ Điền nội dung cho lớp mới

➤ Định nghĩa các hàm sự kiện *afx_msg*

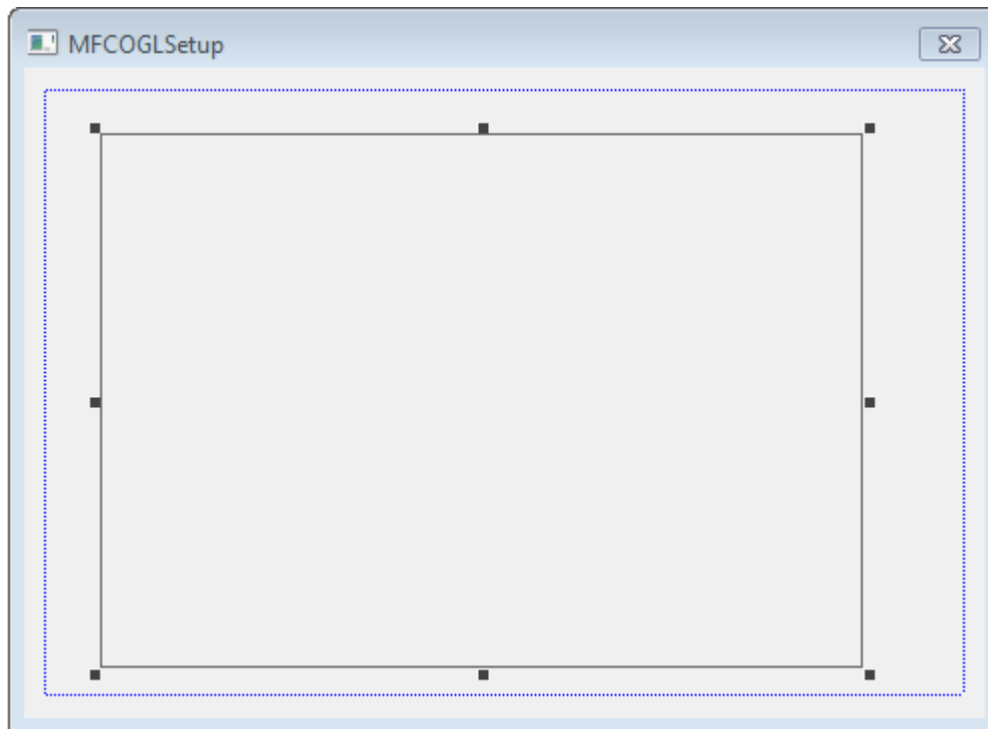
File OpenGLControl.cpp

```
...  
  
void COpenGLControl::OnDraw(CDC *pDC)  
{  
    // TODO: Camera controls.  
    glLoadIdentity();  
    glTranslatef(0.0f, 0.0f, -m_fZoom);  
    glTranslatef(m_fPosX, m_fPosY, 0.0f);  
    glRotatef(m_fRotX, 1.0f, 0.0f, 0.0f);  
    glRotatef(m_fRotY, 0.0f, 1.0f, 0.0f);  
}
```

3. Thiết lập và sử dụng OpenGL trong MFC

3.2. Sử dụng lớp quản lý khung vẽ OpenGL

- Thêm một điều khiển Picture Control vào giao diện dialog (kéo thả từ Toolbox)
- Thiết lập properties cho điều khiển này với các thuộc tính: *ID: IDC_OPENGL*
Visible: False



Behavior	
Accept Files	False
Disabled	False
Help ID	False
Visible	False
Misc	
(Name)	IDC_OPENGL2 (Picture Cont
Group	False
ID	IDC_OPENGL
Image	
Tabstop	False
Type	Frame

3. Thiết lập và sử dụng OpenGL trong MFC

3.2. Sử dụng lớp quản lý khung vẽ OpenGL

- Khai báo và sử dụng đối tượng của lớp COpenGLControl vừa xây dựng

File ...Dlg.h

```
#include "OpenGLControl.h"

COpenGLControl m_oglWindow;
```

File ...Dlg.cpp

// Hàm OnInitDialog()

```
...

// TODO: Add extra initialization here
CRect rect;
// Get size and position of the picture control
GetDlgItem(IDC_OPENGL)->GetWindowRect(rect);
// Convert screen coordinates to client coordinates
ScreenToClient(rect);
// Create OpenGL Control window
m_oglWindow.oglCreate(rect, this);
```

// Hàm OnPaint()

```
...

else
{
    m_oglWindow.oglDrawScene();
    CDialogEx::OnPaint();
}

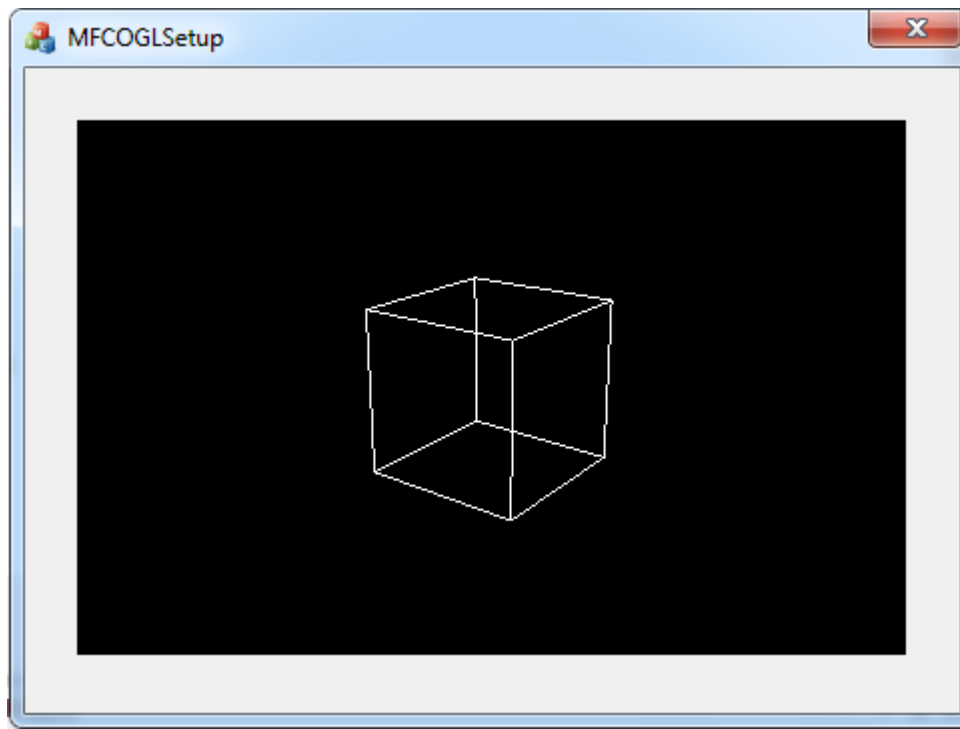
...
```

3. Thiết lập và sử dụng OpenGL trong MFC

3.2. Sử dụng lớp quản lý khung vẽ OpenGL

➡ Kết quả chạy:

- Khung hình đồ họa OpenGL xuất hiện trên dialog.
- Có thể sử dụng chuột Trái, Giữa, Phải, để thực hiện các thao tác xoay, di chuyển, thu phóng hình.



4. Tham khảo

- “OpenGL API Documentation”

<https://www.opengl.org/documentation/>

- “OpenGL Programming Guide” (The red book)

<http://www.glprogramming.com/red/>

- Brett Fowle, “Setting Up OpenGL in an MFC Control”

http://www.codeguru.com/cpp/cpp/cpp_mfc/tutorials/article.php/c10975/Setting-Up-OpenGL-in-an-MFC-Control.htm

Hết Bài 5

