/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Includes

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <windows.h>

#include <gl/gl.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Function Declarations

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LRESULT CALLBACK WndProc (HWND hWnd, UINT message,

WPARAM wParam, LPARAM lParam);

void EnableOpenGL (HWND hWnd, HDC \*hDC, HGLRC \*hRC);

void DisableOpenGL (HWND hWnd, HDC hDC, HGLRC hRC);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* WinMain

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int WINAPI WinMain (HINSTANCE hInstance,

HINSTANCE hPrevInstance,

LPSTR lpCmdLine,

int iCmdShow)

{

WNDCLASS wc;

HWND hWnd;

HDC hDC;

HGLRC hRC;

MSG msg;

BOOL bQuit = FALSE;

float theta = 0.0f;

/\* register window class \*/

wc.style = CS\_OWNDC;

wc.lpfnWndProc = WndProc;

wc.cbClsExtra = 0;

wc.cbWndExtra = 0;

wc.hInstance = hInstance;

wc.hIcon = LoadIcon (NULL, IDI\_APPLICATION);

wc.hCursor = LoadCursor (NULL, IDC\_ARROW);

wc.hbrBackground = (HBRUSH) GetStockObject (BLACK\_BRUSH);

wc.lpszMenuName = NULL;

wc.lpszClassName = "GLSample";

RegisterClass (&wc);

/\* create main window \*/

hWnd = CreateWindow (

"GLSample", "OpenGL Sample",

WS\_CAPTION | WS\_POPUPWINDOW | WS\_VISIBLE,

0, 0, 256, 256,

NULL, NULL, hInstance, NULL);

/\* enable OpenGL for the window \*/

EnableOpenGL (hWnd, &hDC, &hRC);

/\* program main loop \*/

while (!bQuit)

{

/\* check for messages \*/

if (PeekMessage (&msg, NULL, 0, 0, PM\_REMOVE))

{

/\* handle or dispatch messages \*/

if (msg.message == WM\_QUIT)

{

bQuit = TRUE;

}

else

{

TranslateMessage (&msg);

DispatchMessage (&msg);

}

}

else

{

/\* OpenGL animation code goes here \*/

glClearColor (0.0f, 0.0f, 0.0f, 0.0f);

glClear (GL\_COLOR\_BUFFER\_BIT);

glPushMatrix ();

glRotatef (theta, 0.0f, 0.0f, 1.0f);

glBegin (GL\_TRIANGLES);

glColor3f (1.0f, 0.0f, 0.0f); glVertex2f (0.0f, 1.0f);

glColor3f (0.0f, 1.0f, 0.0f); glVertex2f (0.87f, -0.5f);

glColor3f (0.0f, 0.0f, 1.0f); glVertex2f (-0.87f, -0.5f);

glEnd ();

glPopMatrix ();

SwapBuffers (hDC);

theta += 1.0f;

Sleep (1);

}

}

/\* shutdown OpenGL \*/

DisableOpenGL (hWnd, hDC, hRC);

/\* destroy the window explicitly \*/

DestroyWindow (hWnd);

return msg.wParam;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Window Procedure

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LRESULT CALLBACK WndProc (HWND hWnd, UINT message,

WPARAM wParam, LPARAM lParam)

{

switch (message)

{

case WM\_CREATE:

return 0;

case WM\_CLOSE:

PostQuitMessage (0);

return 0;

case WM\_DESTROY:

return 0;

case WM\_KEYDOWN:

switch (wParam)

{

case VK\_ESCAPE:

PostQuitMessage(0);

return 0;

}

return 0;

default:

return DefWindowProc (hWnd, message, wParam, lParam);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Enable OpenGL

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void EnableOpenGL (HWND hWnd, HDC \*hDC, HGLRC \*hRC)

{

PIXELFORMATDESCRIPTOR pfd;

int iFormat;

/\* get the device context (DC) \*/

\*hDC = GetDC (hWnd);

/\* set the pixel format for the DC \*/

ZeroMemory (&pfd, sizeof (pfd));

pfd.nSize = sizeof (pfd);

pfd.nVersion = 1;

pfd.dwFlags = PFD\_DRAW\_TO\_WINDOW |

PFD\_SUPPORT\_OPENGL | PFD\_DOUBLEBUFFER;

pfd.iPixelType = PFD\_TYPE\_RGBA;

pfd.cColorBits = 24;

pfd.cDepthBits = 16;

pfd.iLayerType = PFD\_MAIN\_PLANE;

iFormat = ChoosePixelFormat (\*hDC, &pfd);

SetPixelFormat (\*hDC, iFormat, &pfd);

/\* create and enable the render context (RC) \*/

\*hRC = wglCreateContext( \*hDC );

wglMakeCurrent( \*hDC, \*hRC );

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Disable OpenGL

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void DisableOpenGL (HWND hWnd, HDC hDC, HGLRC hRC)

{

wglMakeCurrent (NULL, NULL);

wglDeleteContext (hRC);

ReleaseDC (hWnd, hDC);

}