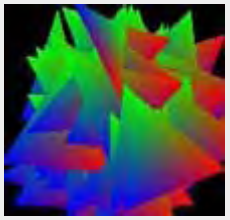
Course 3D_OpenGL: 3D-Graphics with C++ and OpenGL
Chapter 1: Moving Triangles



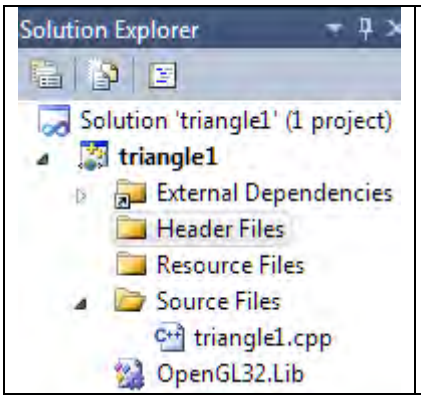
This project is the OpenGL-clone of the C#-projects:
3D-Computer Graphics with XNA, Chapter C1: Moving Triangles and of
3D-Computer Graphics with Managed DirectX, Chapter C1: Moving Triangles.

Project triangle1

Start Visual C++ 2010 Express:
`File` → `New` → `Project` → `Win32 Project` → `Name: triangle1` → `Location: C:\temp` →
uncheck `Create directory for solution` → `Win 32 Application Wizard - triangle1` →
`Application Settings` → `Application type: Windows application` →
`Additional options: Empty project` → quit with button `Finish`.

In the `Solution Explorer -` window **Right**-click `triangle1`.
A drop-down menu appears with a branch: "`Add`". Click `Add` → `Existing item...` →
`C:\Program Files\Microsoft SDKs\Windows\v7.0A\Lib\OpenGl32.Lib` →
Quit with button `Add`.
(If you don't find `OpenGl32.Lib` download opengl32.lib and gl.h.)
Check whether `OpenGl32.Lib` arrived in the Solution Explorer tree.

In the `Solution Explorer -` window click the + sign in front of `triangle1`.
A tree appears with a branch: "`Source Files`". **Right**-click `Source Files`. A drop-down menu appears.
Click `Add` → `New Item...` → `Add New item - triangle1` →
`Visual Studio installed templates: C++ File (.cpp)` →
`Name: triangle1` → `Location: c:\Temp\triangle1` → Quit with button `Add`.
Check whether `triangle1.cpp` arrived under branch `Source Files` in the Solution Explorer tree.



If the `Solution Explorer -` window is invisible,
open it via the main menu: `View` → `Solution Explorer`.

Copy the following code into the empty source file `triangle1.cpp`:

```
#include <windows.h>
#include <time.h>
#include <C:\Program Files\Microsoft SDKs\Windows\v7.0A\include\gl\Gl.h> //adjust path to your Gl.h !

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam);
HWND     hWnd;
HDC      hDC;
HGLRC    hRC;

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int iCmdShow)
{ WNDCLASS wc;
  wc.style         = CS_OWNDC;
  wc.style         = CS_HREDRAW | CS_VREDRAW;
  wc.lpfnWndProc   = WndProc;
  wc.cbClsExtra    = 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 = L"ogl1";
  RegisterClass( &wc );
  hWnd = CreateWindow( wc.lpszClassName, L"C++ OpenGL Chapter 1",
                       WS_CAPTION | WS_POPUPWINDOW | WS_VISIBLE,
                       0, 0, 300, 300, NULL, NULL, hInstance, NULL );
  hDC = GetDC( hWnd );
  PIXELFORMATDESCRIPTOR pfd;
  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;
  int format = ChoosePixelFormat( hDC, &pfd );
  SetPixelFormat( hDC, format, &pfd );
  hRC = wglCreateContext( hDC );
  wglMakeCurrent( hDC, hRC );
  GLfloat vertex[] = {-0.5f,-0.5f, 0.0f,  //lower left  vertex
                       0.0f, 0.5f, 0.0f,  //mid   peek  vertex
                       0.5f,-0.5f, 0.0f };//lower right vertex
  GLfloat color[]  = { 0.0f, 0.0f, 1.0f,  //blue
                       0.0f, 1.0f, 0.0f,  //green
                       1.0f, 0.0f, 0.0f };//red
  glVertexPointer( 3, GL_FLOAT, 0, vertex );
  glColorPointer ( 3, GL_FLOAT, 0, color );
  glEnableClientState( GL_VERTEX_ARRAY );
  glEnableClientState( GL_COLOR_ARRAY  );

  glClearColor( 0.0f, 0.0f, 0.0f, 0.0f );
  glClear( GL_COLOR_BUFFER_BIT );

  glLoadIdentity(); //erase all former transforms
  glDrawArrays( GL_TRIANGLES, 0, 3 ); //draw it
  SwapBuffers( hDC );                 //show it
```

```
  MSG msg;
  while ( TRUE ) //infinite message listening
  { if ( PeekMessage( &msg, NULL, 0, 0, PM_REMOVE ) )
    { if ( msg.message == WM_QUIT ) return 0; //stop infinite loop
      TranslateMessage( &msg );
      DispatchMessage ( &msg );
    }
  } //end of infinite loop
}   //end of WinMain

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)
{ switch (message)
  { case WM_CLOSE: PostQuitMessage( 0 ); return 0; //send WM_QUIT to stop infinite loop
    case WM_QUIT : wglMakeCurrent( hDC, NULL );    //disconnect hRC from hDC
                   wglDeleteContext( hRC );        //say goodbye to hRC
                   ReleaseDC( hWnd, hDC );         //say goodbye to hDC
                   DestroyWindow( hWnd ); return 0;//say goodbye to hWnd
    default      : return DefWindowProc( hWnd, message, wParam, lParam );
  }
}
```

Click `Debug` → `Start Without Debugging Ctrl F5`.

Experiments: (`Debug` → `Start Without Debugging Ctrl F5` after any change.)

1.	Change the `hWnd = CreateWindow( ... )` call by changing its window size parameters no. 6 and 7 from `300` to `600`.	The window is larger at start.
2.	In `GLfloat vertex[] = { ... }` change all `0.5f` values to `0.6f` preserving the signs.	The triangle is bigger.
3.	In `GLfloat color[] = { ... }` change the first three values from `0.0f, 0.0f, 1.0f,` = blue to `1.0f, 1.0f, 0.0f,` = yellow.	The lower left color changes.
4.	In `glClearColor( ... );` change all parameters from `0.0f` to `1.0f`.	white background

Animation

Delete the tail of the `WinMain`-procedure below `glEnableClientState( GL_COLOR_ARRAY );` and its last bracket `} //end of WinMain` and replace the lines:

```
  glClearColor( 0.0f, 0.0f, 0.0f, 0.0f );
  .
  .
  .
  } //end of infinite loop
```

by

```
  float rot = 0.0f;
  MSG msg;
  while ( TRUE ) //infinite loop
  { glClearColor( 0.0f, 0.0f, 0.0f, 0.0f );
    glClear( GL_COLOR_BUFFER_BIT );
    glLoadIdentity(); //erase all former transforms
    glRotatef( rot+=0.01f, 0.0f, 0.0f, 1.0f );
    glDrawArrays( GL_TRIANGLES, 0, 3 ); //draw it
    SwapBuffers( hDC );                 //show it

    if ( PeekMessage( &msg, NULL, 0, 0, PM_REMOVE ) )
    { if ( msg.message == WM_QUIT ) return 0; //stop infinite loop
      TranslateMessage( &msg );
      DispatchMessage ( &msg );
    }
  } //end of infinite loop
```

Experiments: (`Debug` → `Start Without Debugging Ctrl F5` after any change.)

1.	Set comment slashes `//` in front of `glClear( GL_COLOR_BUFFER_BIT );`.	Circular color move.
2.	Set comment slashes `//` in front of `glLoadIdentity();`	Big angular jumps.
3.	`glRotatef( rot+=0.01f, 0.0f, 0.0f, 1.0f );` change to `glRotatef( rot, 0.0f, 0.0f, 1.0f );`.	No rotation at all.
4.	`glRotatef( rot+=0.01f, 0.0f, 0.0f, 1.0f );` change to `glRotatef( rot+=0.01f, 1.0f, 0.0f, 0.0f );`.	Rotation around the x-axis.
5.	`glRotatef( rot+=0.01f, 0.0f, 0.0f, 1.0f );` change to `glRotatef( rot+=0.01f, 0.0f, 1.0f, 0.0f );`.	Rotation around the y-axis.
6.	`glRotatef( rot+=0.01f, 0.0f, 0.0f, 1.0f );` change to `glRotatef( rot+=0.01f, 1.0f, 1.0f, 1.0f );`.	Rotation around the x-, y- and z-axis.
7.	Change the angle increment in `glRotatef( rot+=0.1f, 0.0f, 0.0f, 1.0f );` from `rot+=0.1f` to `rot+=0.5f` and to `rot+=0.05f` and to `rot-=0.1f`.	Faster and slower rotation and clockwise rotation.
8.	Change the starting angle `float rot = 0.0f;` from `0.0f` to `90.0f` and to `-90.0f`.	The rotation starts from 90 or -90 degrees.
9.	Set comment slashes `//` in front of `SwapBuffers( hDC );`	The back buffer is never promoted to serve as visible front buffer.

Three Triangles

Change the `while ( TRUE ) //infinite loop` until il looks like this:

```
while ( TRUE ) //infinite loop
{ glClearColor( 0.0f, 0.0f, 0.0f, 0.0f );
  glClear( GL_COLOR_BUFFER_BIT );
  glLoadIdentity(); //erase all former transforms
  glRotatef( rot+=0.1f, 0.0f, 0.0f, 1.0f );
  glDrawArrays( GL_TRIANGLES, 0, 3 ); //draw it in the middle
  glScalef    ( 0.7f, 0.7f, 0.7f );   //zoom it down
  glTranslatef( -1.0f, 0.0f, 0.0f );  //shift it to the left side
  glDrawArrays( GL_TRIANGLES, 0, 3 ); //draw it again
  glTranslatef( 2.0f, 0.0f, 0.0f );   //shift it to the right side
  glDrawArrays( GL_TRIANGLES, 0, 3 ); //draw it again
  SwapBuffers( hDC );                 //show it

  if ( PeekMessage( &msg, NULL, 0, 0, PM_REMOVE ) )
  { if ( msg.message == WM_QUIT ) return 0; //stop infinite loop
    TranslateMessage( &msg );
    DispatchMessage ( &msg );
  }
} //end of infinite loop
```

Click `Debug` → `Start Without Debugging Ctrl F5`.
Experiments: (Recover the change after any experiment.)

1.	Change the zoom-factor of the small triangles in `glScalef( 0.7f, 0.7f, 0.7f );` from `0.7f` to `0.2f` and to `1.2f`.	Smaller / larger triangles.
2.	Change the left shift of the left small triangle in `glTranslatef( -1.0f, 0.0f, 0.0f );` from `-1.0f` to `-1.5f` and to `-0.5f`.	The left triangle is horizontally shifted along the x-axis.
3.	Change the left shift of the left small triangle in `glTranslatef( -1.0f, 0.0f, 0.0f );` from `(-1.0f,0.0f,0.0f)` to `(-1.0f,1.0f,0.0f)` and to `(-1.0f,-1.0f,0.0f)`.	The left triangle is vertically shifted.
4.	Change the left shift of the left small triangle in `glTranslatef( -1.0f, 0.0f, 0.0f );` from `(-1.0f,0.0f,0.0f)` to `(-1.0f,0.0f,-2.0f)`.	The left triangle is shifted back and seems smaller.
5.	Copy a new line `glRotatef( rot, 0.0f, 0.0f, 1.0f );` below the line `glDrawArrays( GL_TRIANGLES, 0, 3 ); //draw it in the middle`.	The small triangles make their own rotations.

Hundred Triangles

Add the following declarations just below the line `int WINAPI WinMain( ... )`.
The head of `int WINAPI WinMain( ... )` should look like this:

```
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine,
int iCmdShow)
{ const int nTriangles = 100;
  float dx[nTriangles];
  float dy[nTriangles];
  float dz[nTriangles];
  float ax[nTriangles];
  float ay[nTriangles];
  float az[nTriangles];
  srand( (unsigned)time(NULL) ); //initialize random number generator
  for ( int i=0; i < nTriangles; i++ )
  { dx[i] = -0.5f + (float)rand() / RAND_MAX; //something between -0.5 and +0.5
    dy[i] = -0.5f + (float)rand() / RAND_MAX; //something between -0.5 and +0.5
    dz[i] = -0.5f + (float)rand() / RAND_MAX; //something between -0.5 and +0.5
    ax[i] = 90.0f * (float)rand() / RAND_MAX; //something between  0.0 and 90.0
    ay[i] = 90.0f * (float)rand() / RAND_MAX; //something between  0.0 and 90.0
    az[i] = 90.0f * (float)rand() / RAND_MAX; //something between  0.0 and 90.0
  }
  WNDCLASS wc;
  .
  .
```

Change the `while ( TRUE ) //infinite loop` as follows:

```
  while ( TRUE ) //infinite loop
  { glClearColor( 0.0f, 0.0f, 0.0f, 0.0f );
    glClear( GL_COLOR_BUFFER_BIT );
    for ( int i=0; i < nTriangles; i++ )
    { glLoadIdentity(); //erase all former transforms
      glScalef( 0.7f, 0.7f, 0.7f );
      glRotatef( ax[i]+=0.1f, 1.0f, 0.0f, 0.0f ); //around x-axis
      glRotatef( ay[i]+=0.1f, 0.0f, 1.0f, 0.0f ); //around y-axis
      glRotatef( az[i]+=0.1f, 0.0f, 0.0f, 1.0f ); //around z-axis
      glTranslatef( dx[i], dy[i], dz[i] );         //shifts
      glDrawArrays( GL_TRIANGLES, 0, 3 );          //draw it
    }
    SwapBuffers( hDC ); //show all
    MSG msg;            //listen to what happens outside the loop
    if ( PeekMessage( &msg, NULL, 0, 0, PM_REMOVE ) )
    { if ( msg.message == WM_QUIT ) return 0; //stop infinite loop
      TranslateMessage( &msg );
      DispatchMessage ( &msg );
    }
  } //end of infinite loop
```

Click `Debug` → `Start Without Debugging Ctrl F5`.
Hundred small triangles will dance as flying paper sheets do.

Experiments: (Recover the change after any experiment.)

1.	Put comment slashes `//` in front of both `glClear( GL_COLOR_BUFFER_BIT );`.	Old triangles are never erased.
2.	Vary the constant `const Int32 nTriangles = 100;` between `10` and `1000`.	Less or more triangles.
3.	Vary the random initial rotation angles `ax[i], ay[i], az[i]` in the head of `WinMain` by changing the factor `90.0f` to `30.0f` and to `10.0f`.	Different levels of disorder.
4.	Vary the angular increments `ax[i]+=0.1f`, `ay[i]+=0.1f` and `az[i]+=0.1f` inside the three `glRotatef( ... )`-statements from `0.1f` to `0.01f` and `1.0f`.	Rotation velocities change.
5.	Insert the new line `glTranslatef( dx[i], dy[i], dz[i] );` just below `glScalef( 0.7f, 0.7f, 0.7f );`.	The bevy expands.