

Se realiza el movimiento de un cuadrado. Con devc y opengl. (Los alumnos ya deben saber abrir un programa basico con devc en multimedia y opengl)



Debemos quitar la opcion de repetir con while, y pondremos un for.

```
/* 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 */
//desplazamiento en x solamente
```

quitar o comentar

```

        glColor3f (1.0f, 1.0f, 1.0f);
        Sleep (20);
        SwapBuffers (hDC);

        //x+=0.01f;
        // y+=0.01f;
    }

    // Sleep (2000);
    //}
// }

/* shutdown OpenGL */
DisableOpenGL (hWnd, hDC, hRC);

```

Quitar tambien

Solo se coloco la parte que se editò. Despues, Abajo esta el programa

```

/*****
 * 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
 *
 *****/
float x,y;
int WINAPI WinMain (HINSTANCE hInstance,
                    HINSTANCE hPrevInstance,
                    LPSTR lpCmdLine,
                    int iCmdShow)

```

completo

```

/* 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 */
//desplazamiento en x solamente
for(x=-1;x<1; x+=0.01f)
{
    glClearColor (0.0f, 0.0f, 0.0f, 0.0f);
    glClear (GL_COLOR_BUFFER_BIT);

    glPushMatrix ();
    glRotatef (theta, 0.0f, 0.0f, 1.0f);
    y=0.2;
    glBegin (GL_QUADS);
    glColor3f (1.0f, 0.0f, 0.0f);
    glVertex2f (x, y);
    glVertex2f (x-1, y);
    glVertex2f (x-1,y-1);
    glVertex2f(x,y-1);
    glEnd ();
    glPopMatrix ();
    glColor3f (1.0f, 1.0f, 1.0f);
    Sleep (20);
    SwapBuffers (hDC);

    //x+=0.01f;
    // y+=0.01f;
}

    // Sleep (2000);
// }
// }

```

```
/******
```

```
* 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
```

```
*
```

```
*****/
```

```
float x,y;

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 */

//desplazamiento en x solamente
for(x=-1;x<1; x+=0.01f)
{
    glClearColor (0.0f, 0.0f, 0.0f, 0.0f);
    glClear (GL_COLOR_BUFFER_BIT);

    glPushMatrix ();
    glRotatef (theta, 0.0f, 0.0f, 1.0f);
    y=0.2;
    glBegin (GL_QUADS);
    glColor3f (1.0f, 0.0f, 0.0f);
    glVertex2f (x, y);
    glVertex2f (x-1, y);
    glVertex2f (x-1,y-1);
    glVertex2f(x,y-1);
    glEnd ();
    glPopMatrix ();
    glColor3f (1.0f, 1.0f, 1.0f);
    Sleep (20);
    SwapBuffers (hDC);

    //x+=0.01f;
    // y+=0.01f;
```

```
}  
  
    // Sleep (2000);  
  
    //}  
  
//}  
  
/* 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.iPixelFormat = 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);
```

```
}
```