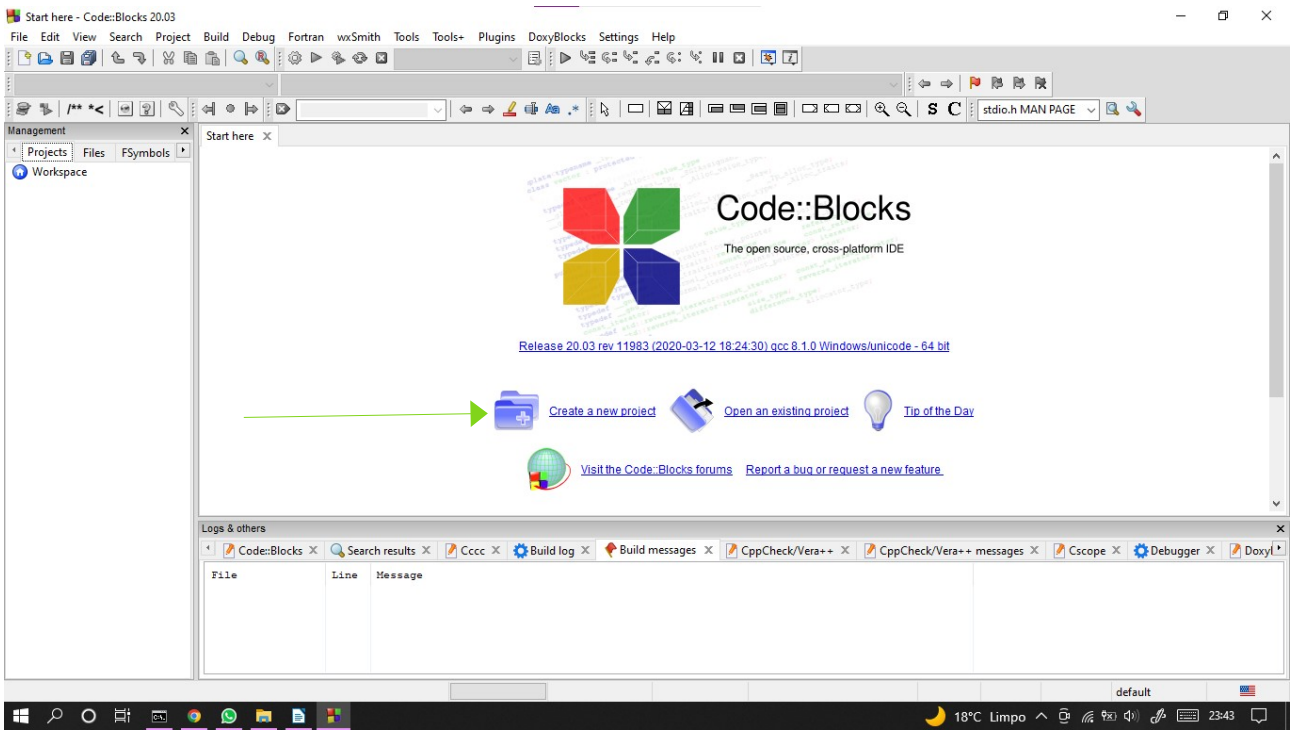
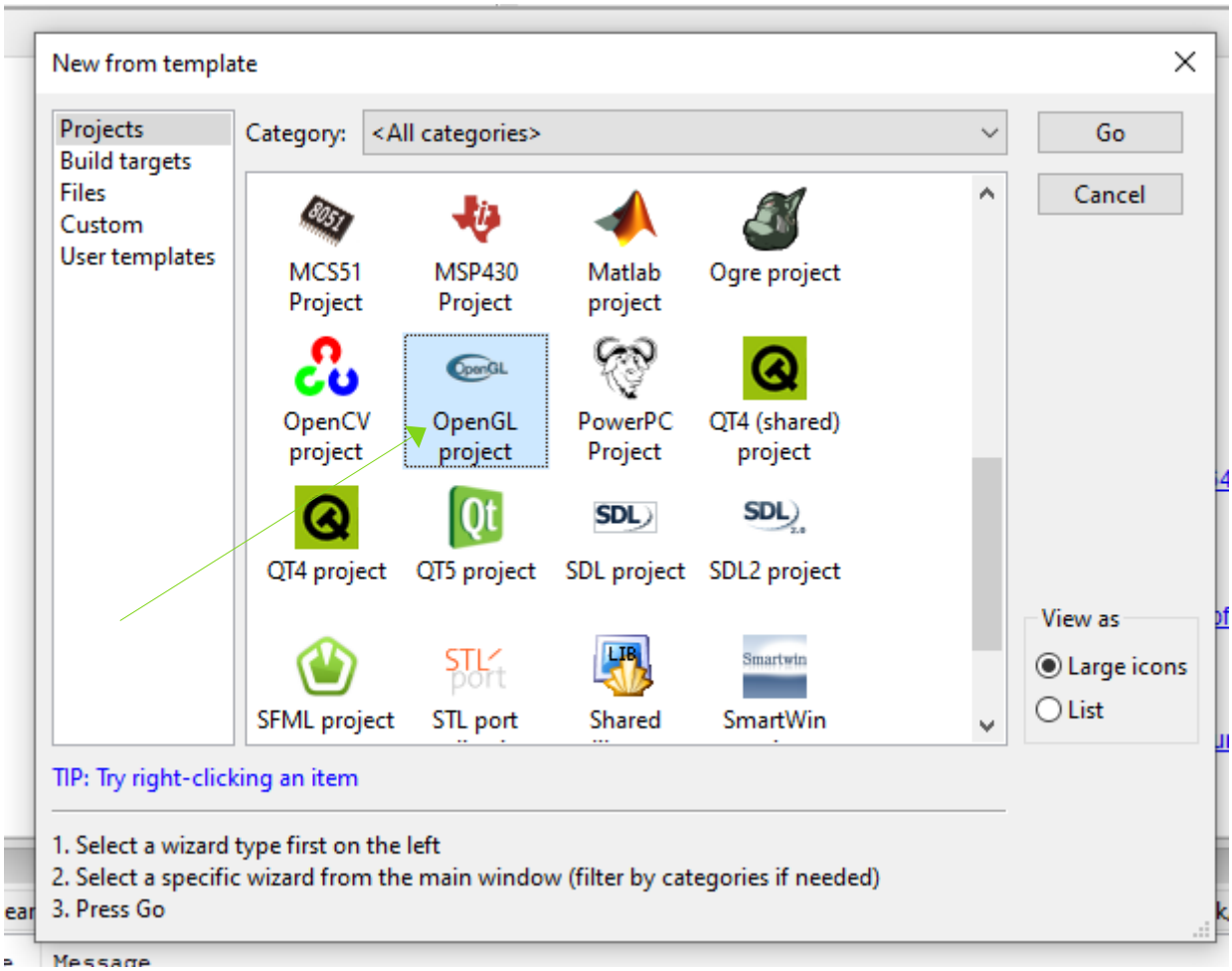
LINGUAGEM C

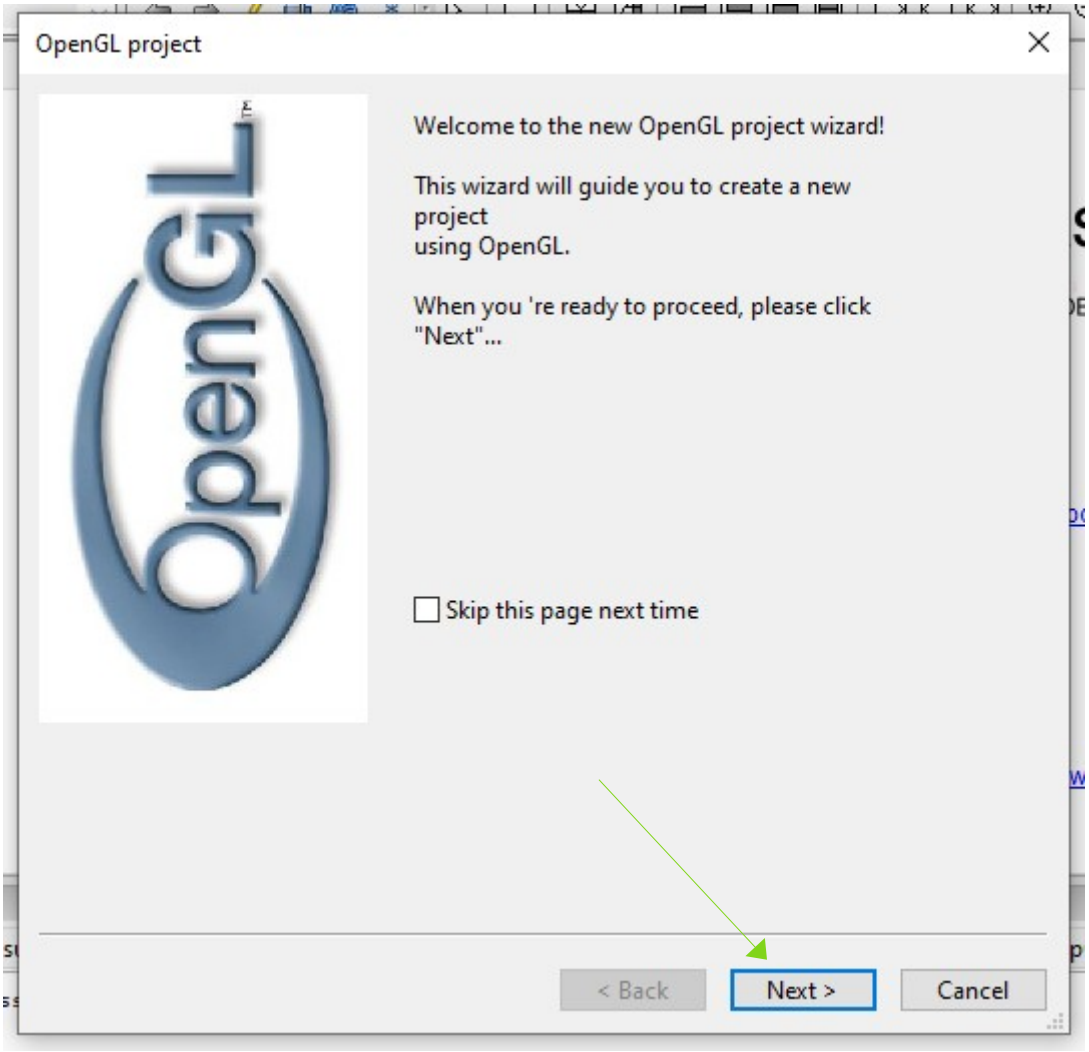
Iniciar projeto OpenGl



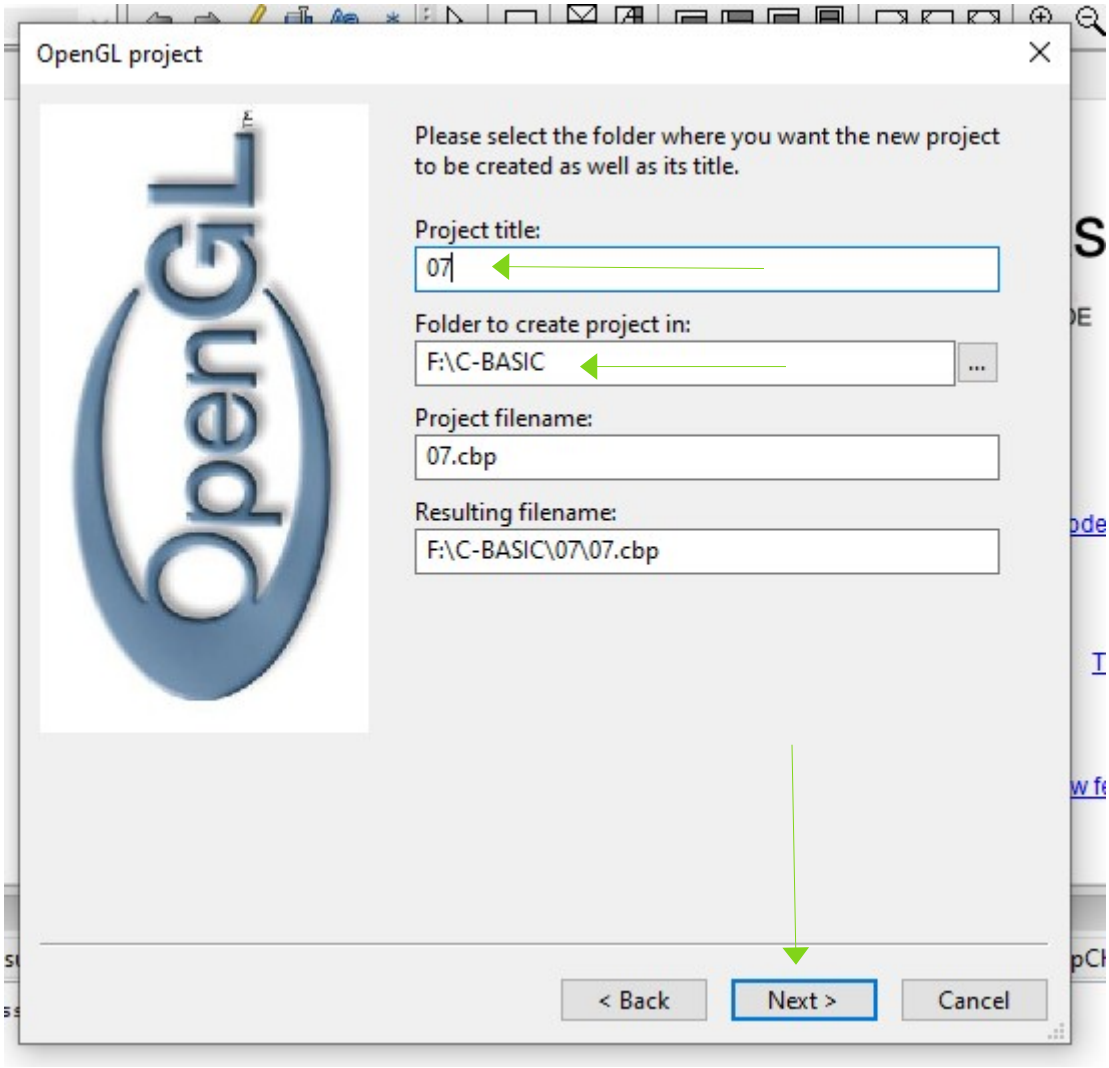
crie um novo projeto.



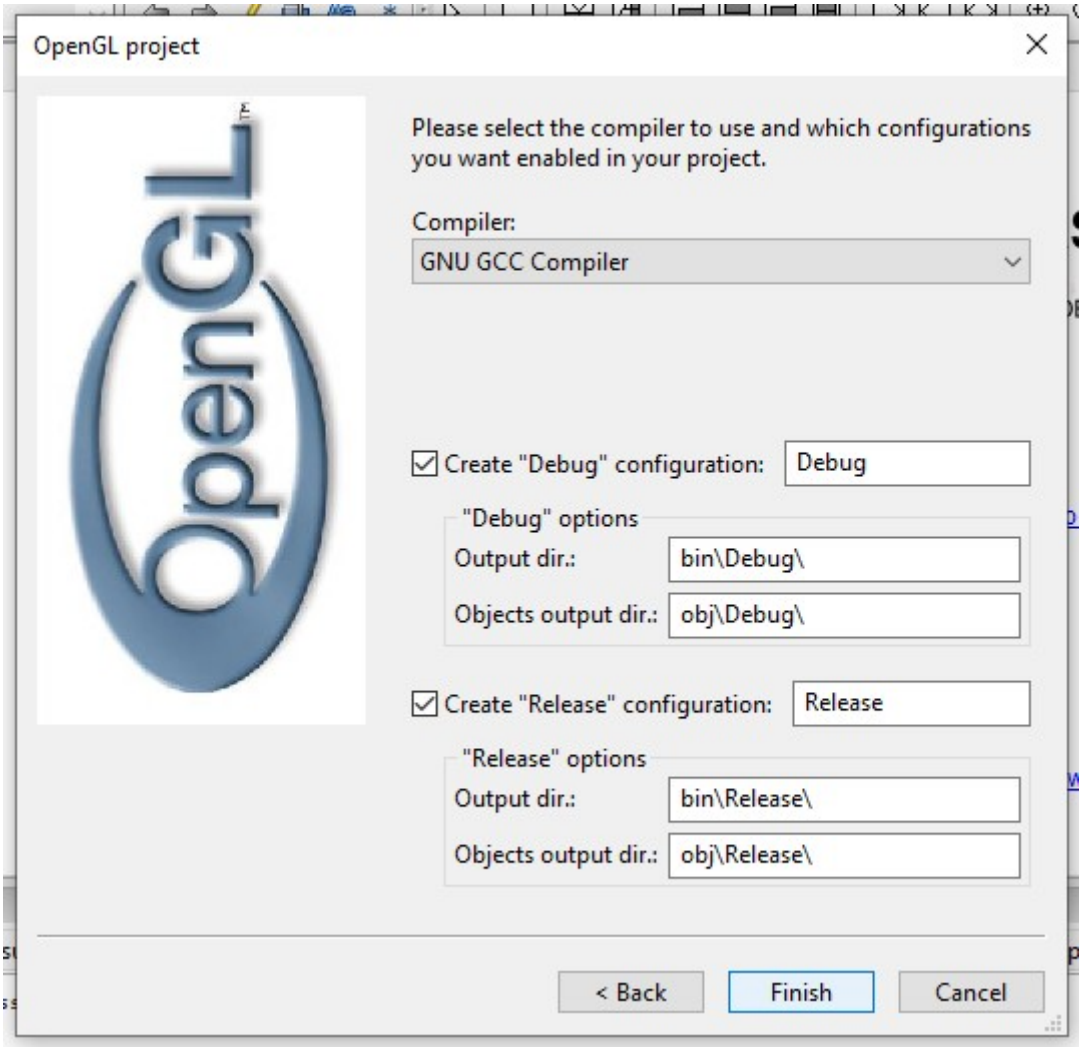
Do tipo OpenGl project.


OpenGL project
Welcome to the new OpenGL project wizard!
This wizard will guide you to create a new project
using OpenGL.
When you 're ready to proceed, please click "Next"...
Skip this page next time
< Back
Next >
Cancel

NEXT



Escolha o caminho e a pasta para salvar de nome ao projeto e click em next.



Finish

```
#include <windows.h>
#include <gl/gl.h>

LRESULT CALLBACK WindowProc(HWND, UINT, WPARAM, LPARAM);
void EnableOpenGL(HWND hwnd, HDC*, HGLRC*);
void DisableOpenGL(HWND, HDC, HGLRC);


int WINAPI WinMain(HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPSTR lpCmdLine,
                   int nCmdShow)
{
    WNDCLASSEX wcex;
    HWND hwnd;
    HDC hDC;
    HGLRC hRC;
    MSG msg;
    BOOL bQuit = FALSE;
    float theta = 0.0f;
```

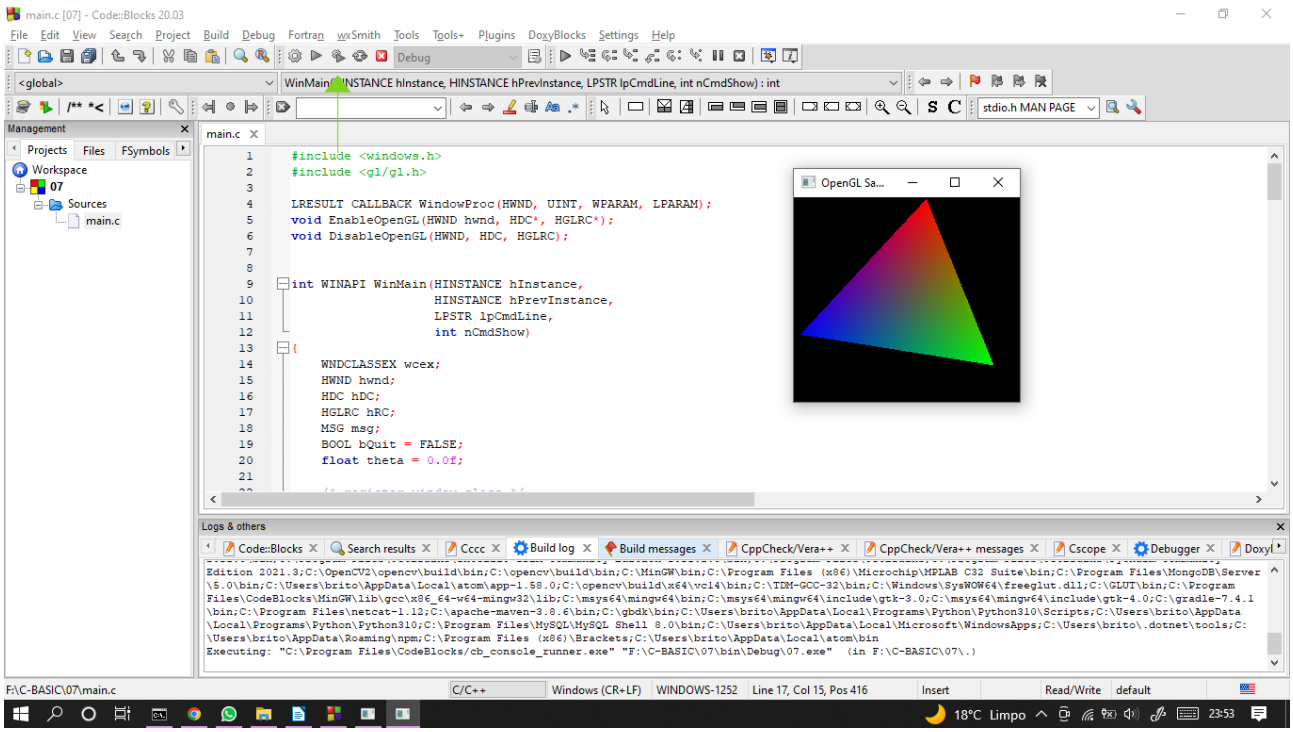
```
/* register window class */
wcex.cbSize = sizeof(WNDCLASSEX);
wcex.style = CS_OWNDC;
wcex.lpfnWndProc = WindowProc;
wcex.cbClsExtra = 0;
wcex.cbWndExtra = 0;
wcex.hInstance = hInstance;
wcex.hIcon = LoadIcon(NULL, IDI_APPLICATION);
wcex.hCursor = LoadCursor(NULL, IDC_ARROW);
wcex.hbrBackground = (HBRUSH)GetStockObject(BLACK_BRUSH);
wcex.lpszMenuName = NULL;
wcex.lpszClassName = "GLSample";
wcex.hIconSm = LoadIcon(NULL, IDI_APPLICATION);;


if (!RegisterClassEx(&wcex))
    return 0;

/* create main window */
hwnd = CreateWindowEx(0,
                      "GLSample",
                      "OpenGL Sample",
                      WS_OVERLAPPEDWINDOW,
                      CW_USEDEFAULT,
                      CW_USEDEFAULT,
                      256,
                      256,
                      NULL,
                      NULL,
                      hInstance,
                      NULL);

ShowWindow(hwnd, nCmdShow);

/* 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;
}

LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam)
{
    switch (uMsg)
    {
        case WM_CLOSE:
            PostQuitMessage(0);
        break;
```

```
        case WM_DESTROY:
            return 0;

        case WM_KEYDOWN:
        {
            switch (wParam)
            {
                case VK_ESCAPE:
                    PostQuitMessage(0);
                break;
            }
        }
        break;

        default:
            return DefWindowProc(hwnd, uMsg, wParam, lParam);
    }

    return 0;
}

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);
}
```

```
void DisableOpenGL (HWND hwnd, HDC hDC, HGLRC hRC)
{
    wglMakeCurrent(NULL, NULL);
    wglDeleteContext(hRC);
    ReleaseDC(hwnd, hDC);
}
```

Uma janela básica é criada e é nela que mudaremos suas configurações para criar nossos futuros projetos.



Click em buiild and run e o projeto deve iniciar. Com essa pequena animação desse triangulo.

Com isso temos uma janela básica e moldaremos nossos games ou qualquer outro projeto que use opengl.

É nessa janela básica que podemos fazer games entre outras coisas como reconhecimento facial.

DEUS E FIEL