

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management Start here

Projects Files FSymbols Workspace

Code::Blocks
The open source, cross-platform IDE

Release 20.03 rev 11983 (2020-03-12 18:24:30) gcc 8.1.0 Windows/unicode - 64 bit

Create a new project Open an existing project Tip of the Day

Visit the Code::Blocks forums Report a bug or request a new feature

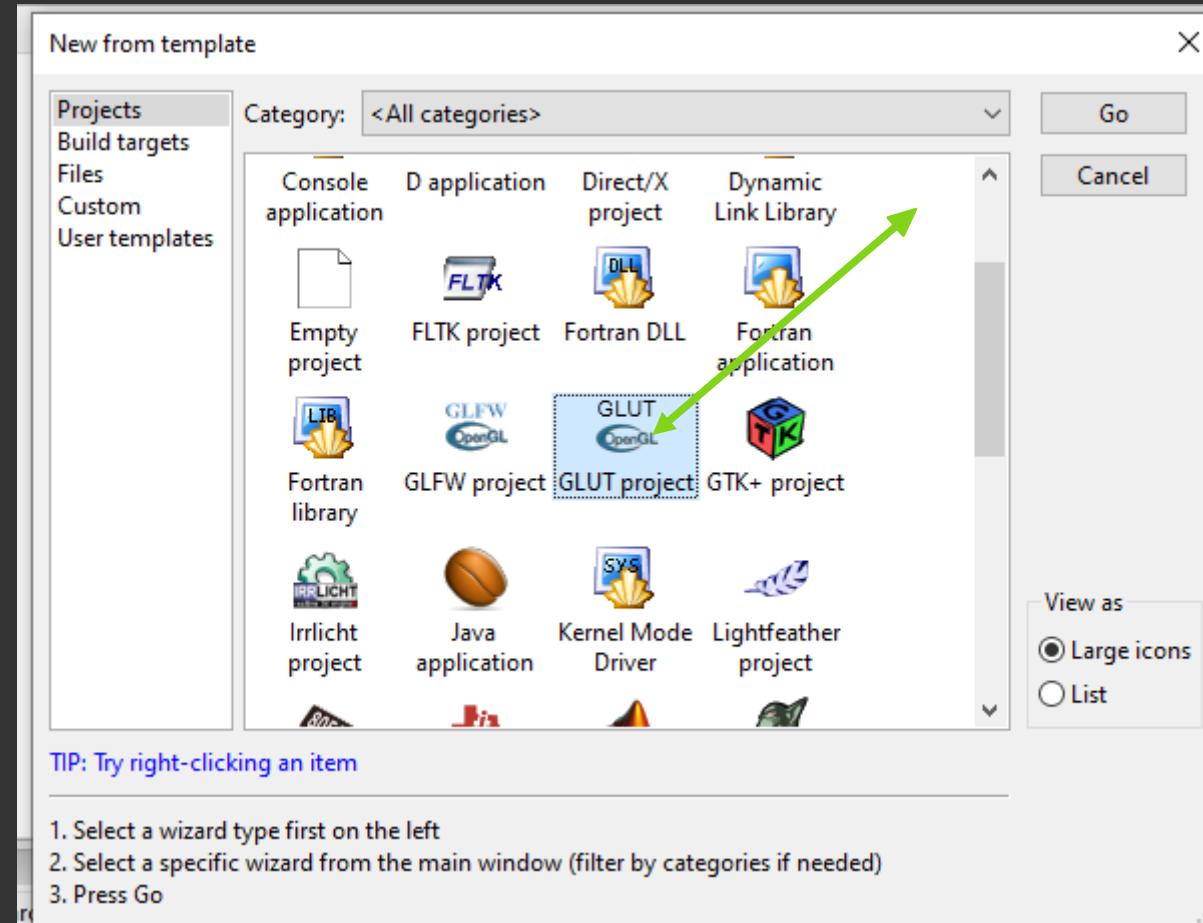
Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxy X

File	Line	Message

default

30°C Nublado



GLUT project



Welcome to the new GLUT project wizard!

This wizard will guide you to create a new project using the GLUT OpenGL extensions.

When you're ready to proceed, please click "Next"...

Skip this page next time

< Back

Next >

Cancel

GLUT project



Please select the folder where you want the new project to be created as well as its title.

Project title:



Folder to create project in:

...

Project filename:

Resulting filename:

Nome e local

< Back

Next >

Cancel

GLUT project



Please select the location of GLUT on your computer.
This is the top-level folder where GLUT was installed (unpacked).
To help you, this folder must contain the subfolders
"include" and "lib".

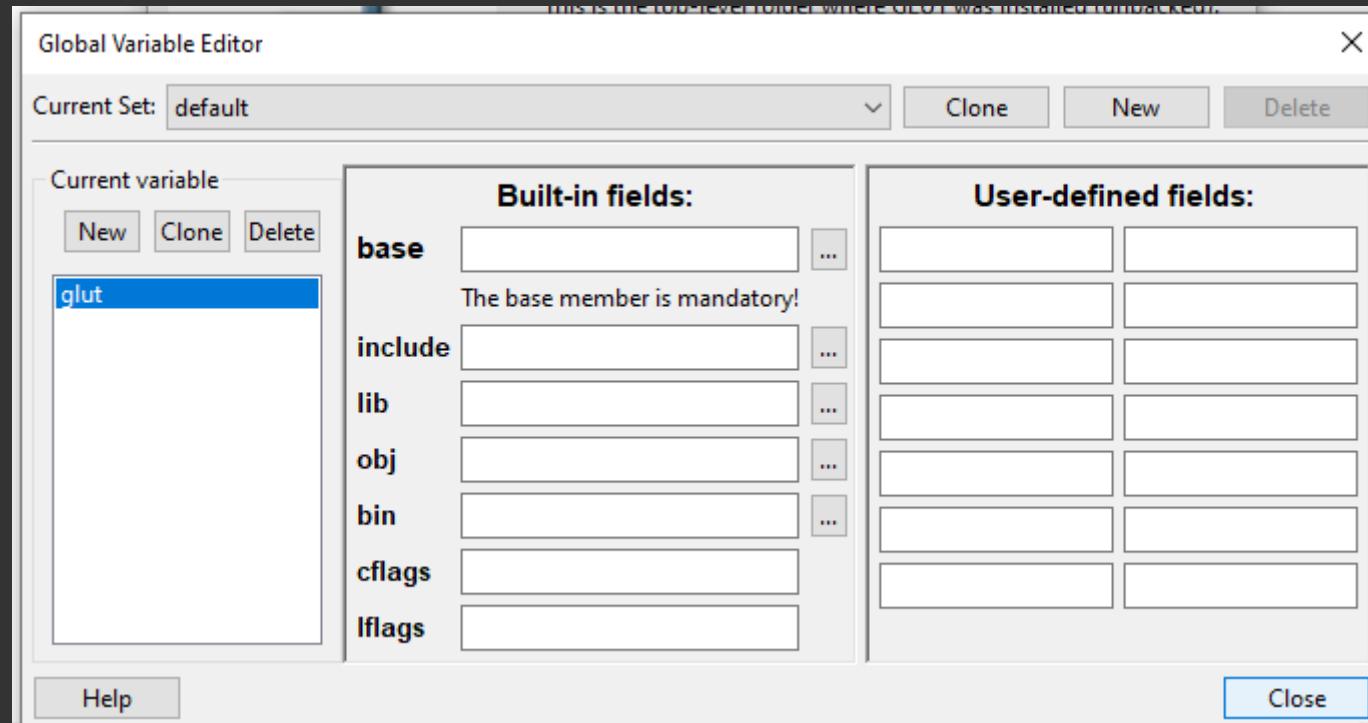
Please select GLUT's location:

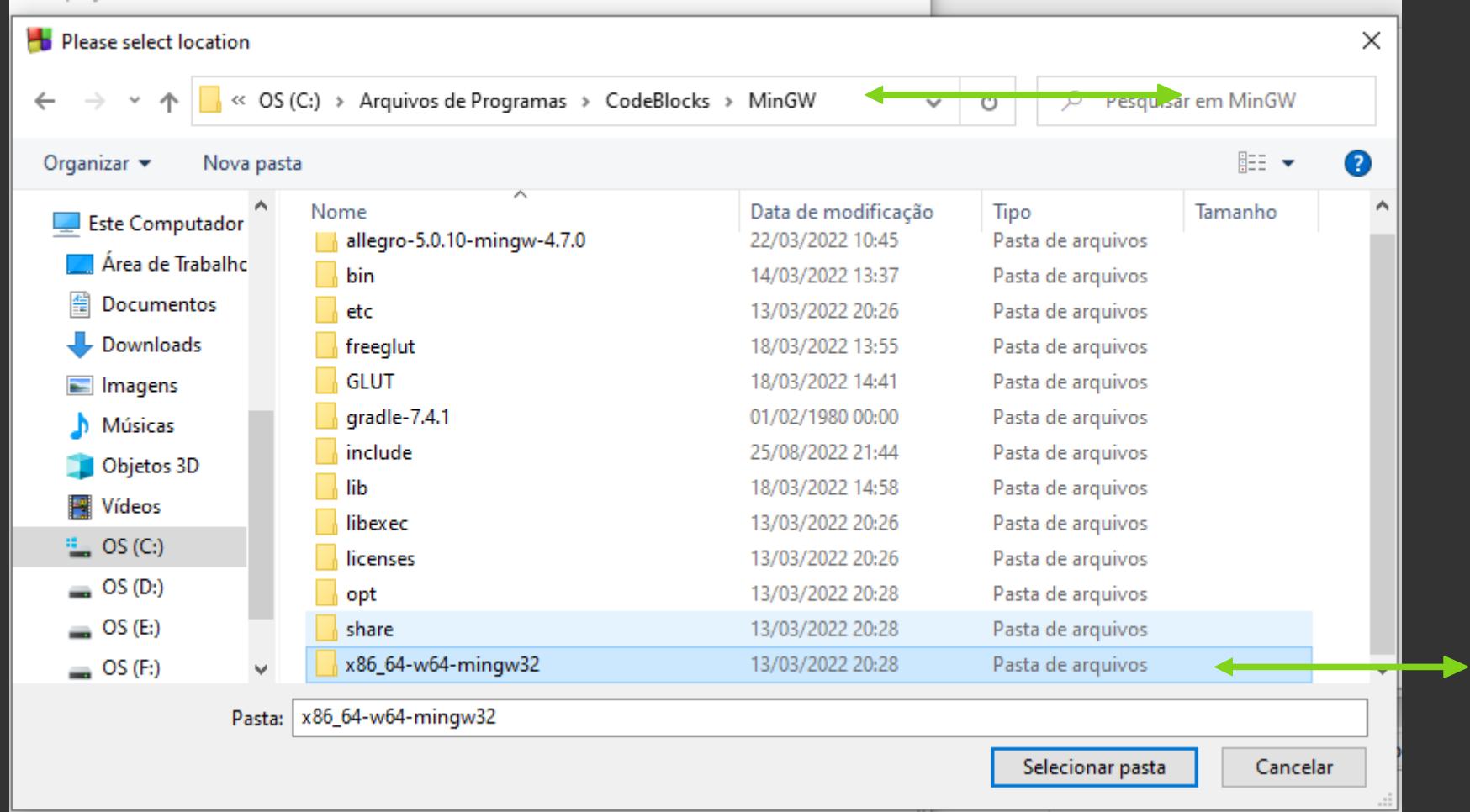


< Back

Next >

Cancel





GLUT project

X



Please select the location of GLUT on your computer.
This is the top-level folder where GLUT was installed (unpacked).
To help you, this folder must contain the subfolders
"include" and "lib".

Please select GLUT's location:

C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw



< Back

Next >

Cancel

GLUT project

X



Please select the compiler to use and which configurations you want enabled in your project.

Compiler:

GNU GCC Compiler

Create "Debug" configuration: Debug

"Debug" options

Output dir.: bin\Debug\

Objects output dir.: obj\Debug\

Create "Release" configuration: Release

"Release" options

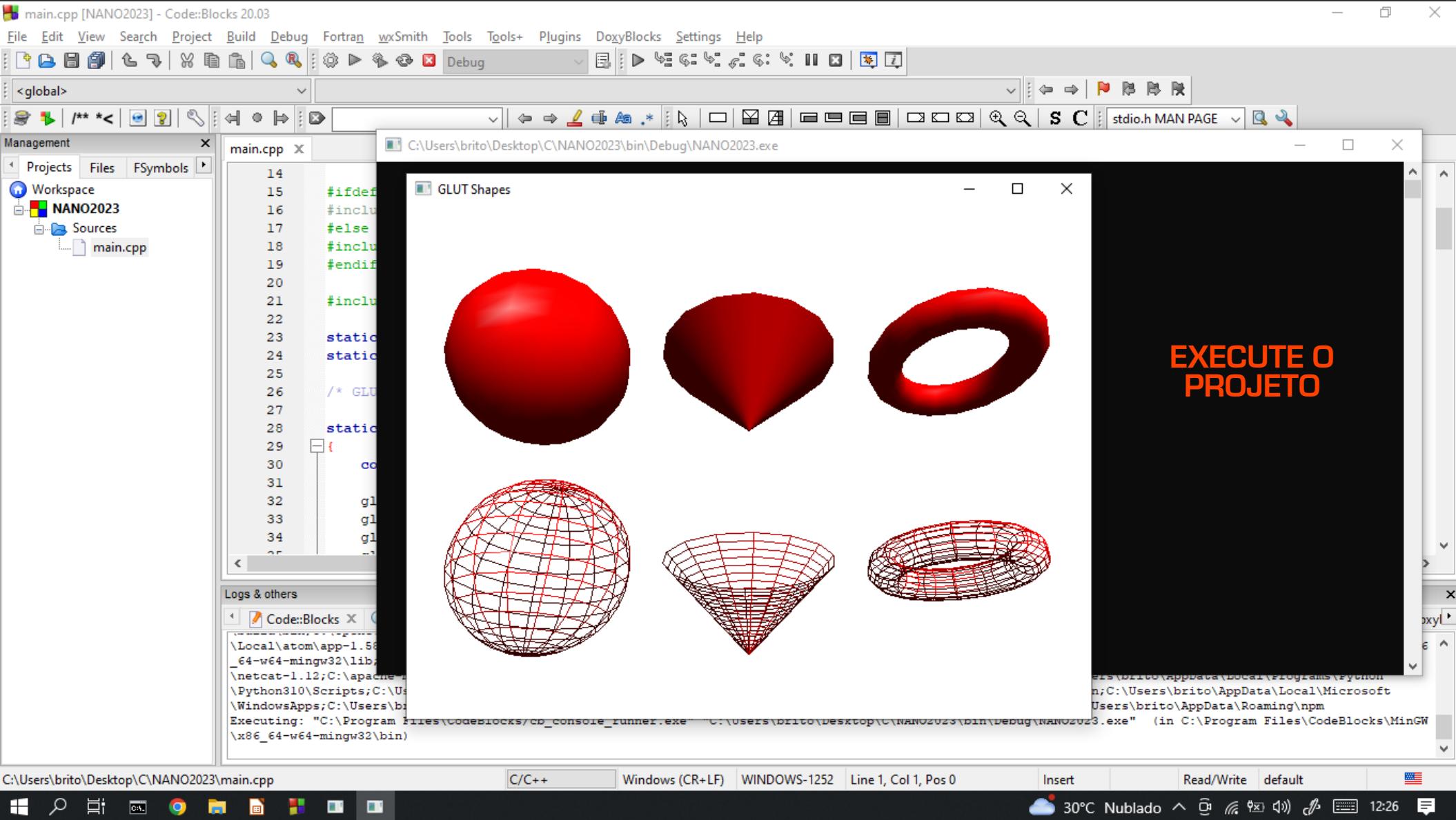
Output dir.: bin\Release\

Objects output dir.: obj\Release\

< Back

Finish

Cancel



main.cpp [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Management X

Projects Files FSymbols

Workspace NANO2023

Sources main.cpp

Save main.cpp
Close main.cpp
Open with >
Remove file from project
Format this file (AStyle)
Reparse this file

Build file
Clean file
Options >
Properties...

main.cpp X

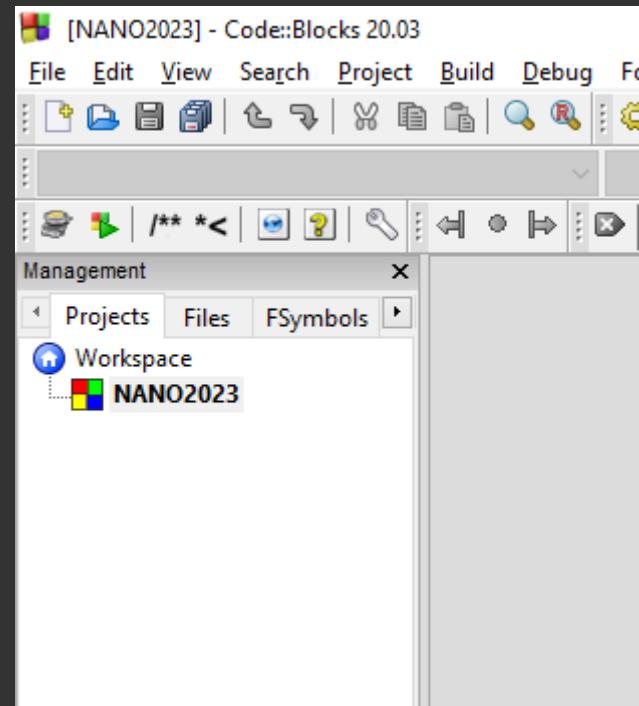
```
14
15 #ifdef __APPLE__
16 #include <GLUT/glut.h>
17 #else
18 #include <GL/glut.h>
19 f
20 ude <stdlib.h>
21 c int slices = 16;
22 c int stacks = 16;
23 UT callback Handlers */
24
25 c void resize(int width, int height)
26 const float ar = (float) width / (float) height;
27
28 LViewport(0, 0, width, height);
29 glMatrixMode(GL_PROJECTION);
30 glLoadIdentity();
31 gluPerspective(45.0, ar, 0.1, 100.0);
```

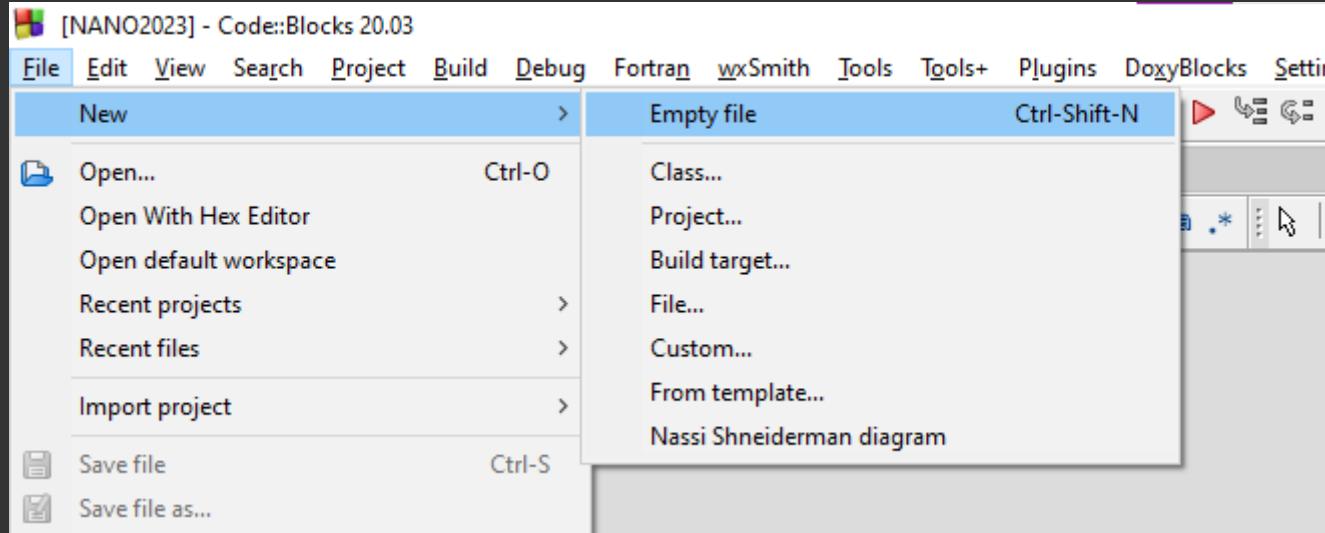
Logs & others

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera

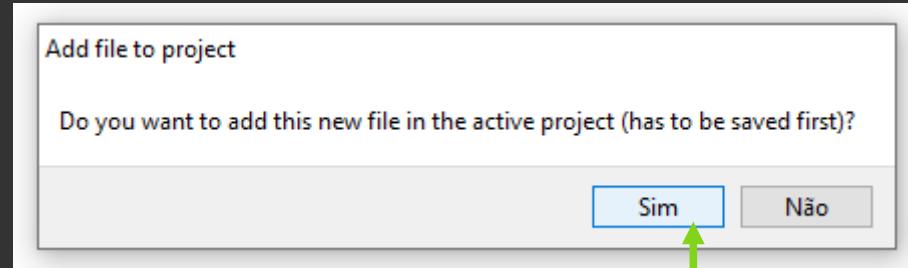
\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\My

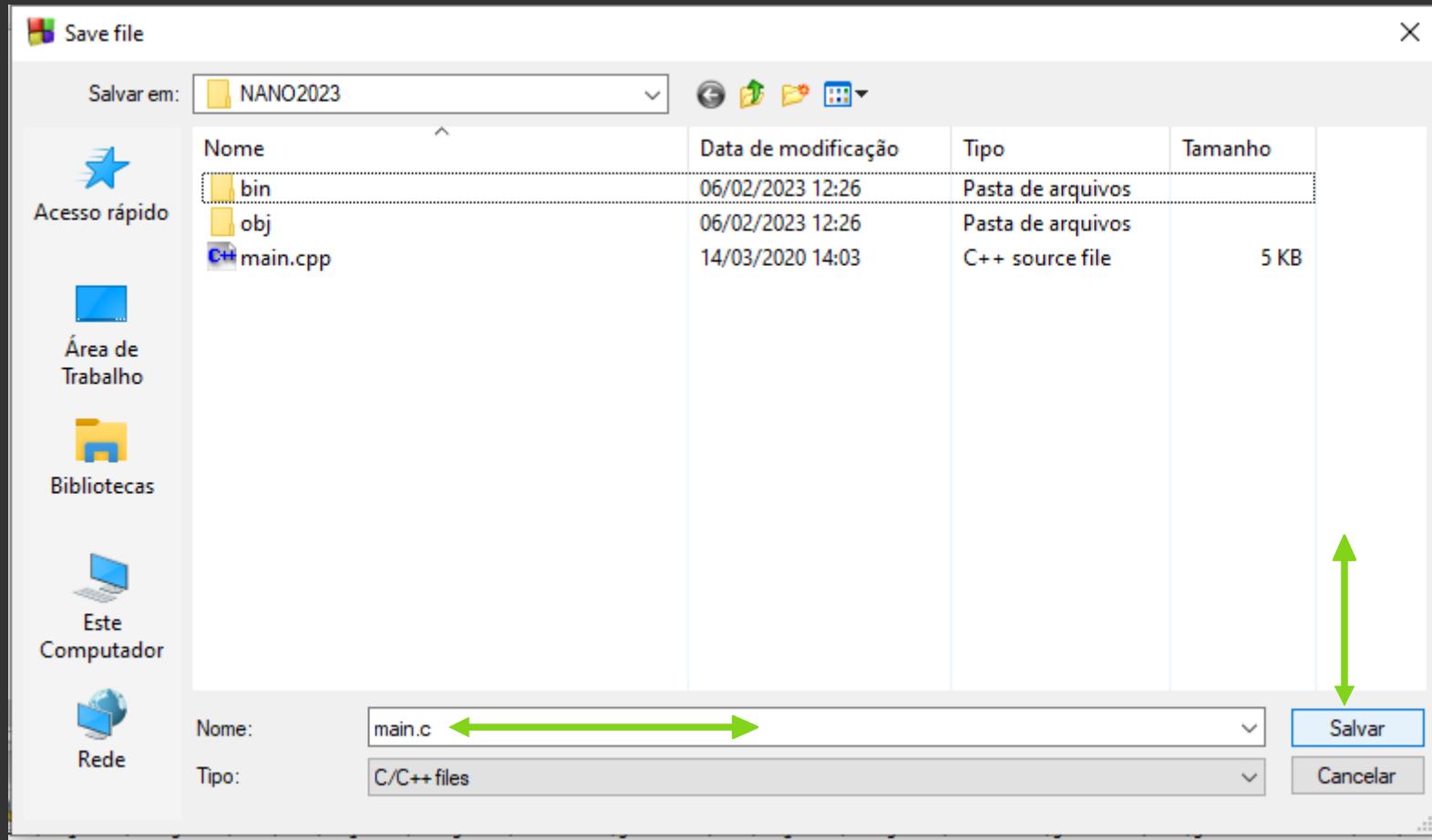
**Delete o arquivo .cpp
E crie um arquivo .c**



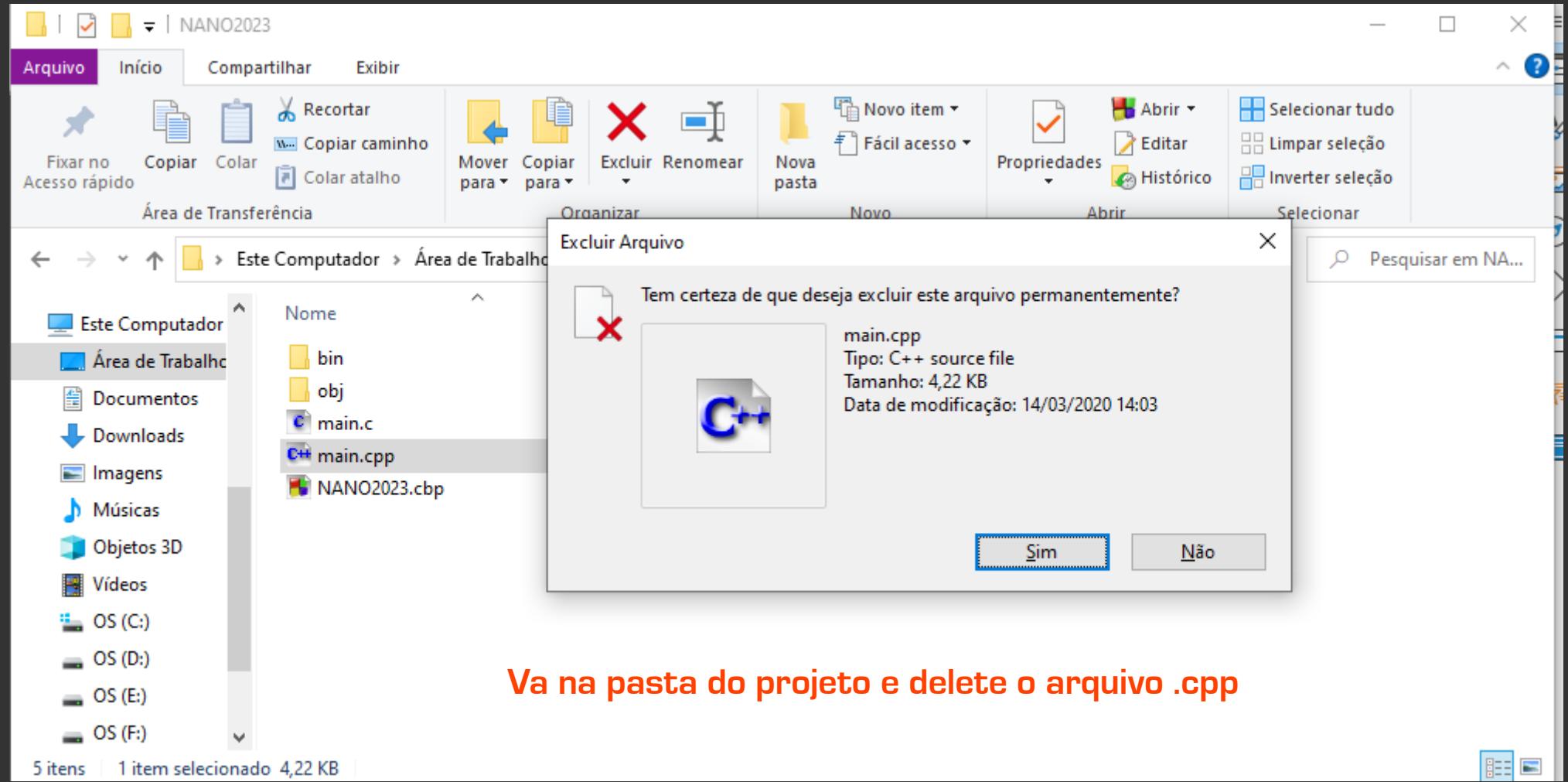


Crie o arquivo main.c





Coloque o nome de main.c e salve



NANO2023

Arquivo Início Compartilhar Exibir

Fixar no Acesso rápido Copiar Colar Recortar Copiar caminho Colar atalho Mover para Copiar para Excluir Renomear Nova pasta Novo item Fácil acesso Propriedades Abrir Histórico Selecionar tudo Limpar seleção Inverter seleção

Área de Transferência Organizar Novo Abrir Selecionar

Este Computador > Área de Trabalho > C > NANO2023

Pesquisar

Nome	Data de modificação	Tipo	Tamanho
bin	06/02/2023 12:26	Pasta de arquivos	
obj	06/02/2023 12:26	Pasta de arquivos	
main.c	06/02/2023 12:28	C source file	1 KB
NANO2023.cbp	06/02/2023 12:26	project file	2 KB

Este Computador

Área de Trabalho

Documentos

Downloads

Imagens

Músicas

Objetos 3D

Vídeos

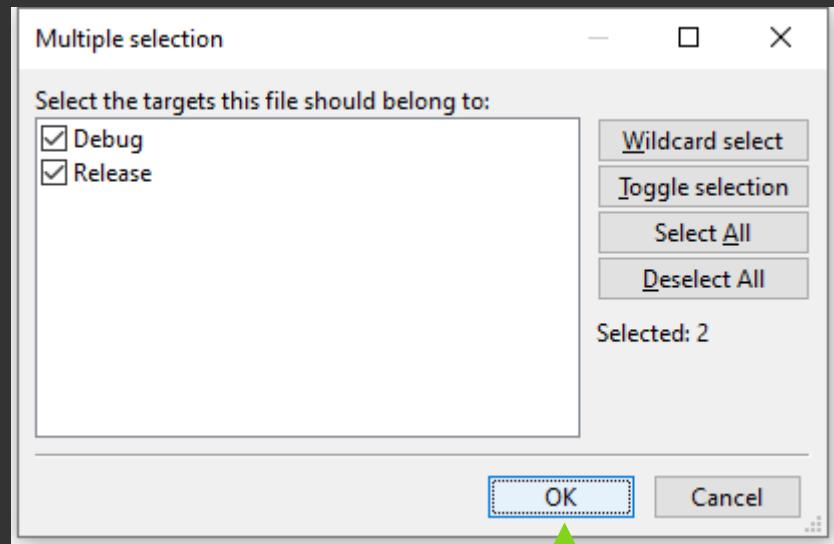
OS (C:)

OS (D:)

OS (E:)

OS (F:)

4 itens



main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management

Projects Files FSymbols

Workspace NANO2023 Sources main.c

main.c X

```
1
2
```

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxygen X

```
\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023\bin\Debug\NANO2023.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\bin)
Process terminated with status -1073741510 (0 minute(s), 14 second(s))
```

C:\Users\brito\Desktop\C\NANO2023\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/Write default 30°C Nublado 12:32

The screenshot shows the Code::Blocks 20.03 IDE interface. The title bar reads "*main.c [NANO2023] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The status bar at the bottom shows "main(): int".

The left sidebar is titled "Management" and contains tabs for Projects, Files, and FSymbols. Under "Projects", there is a "Workspace" section with a "NANO2023" project expanded, showing a "Sources" folder containing "main.c".

The main editor window displays the code for "main.c":

```
1 int main()
2 {
3     glutI
4     (●) glutGetWindow(): int
5     (●) glutHideOverlay(): void
6 }
```

A code completion dropdown menu is open at the cursor position, listing several glut functions:

- (●) glutGetWindow(): int
- (●) glutHideOverlay(): void
- (●) glutHideWindow(): void
- (●) glutIconifyWindow(): void
- (●) glutIdleFunc(): void
- (●) glutIgnoreKeyRepeat(): void
- # glutInit
- (●) glutInit(): void
- (●) glutInitDisplayMode(): void
- (●) glutInitDisplayString(): void
- (●) glutInitWindowSize(): void

The function "glutInit():" is highlighted in blue, indicating it is the selected option.

Quando digitamos glut + primeira letra da função as opções aparecem

Todas os nomes das funções começam com glut

The screenshot shows the Code::Blocks IDE interface. The title bar reads "*main.c [NANO2023] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The status bar shows "Debug". The code editor window displays the following C code:

```
int main()
{
    glutInit();
    return 0;
}
```

A tooltip or callout box is overlaid on the code editor, highlighting the `glutInit();` line. The tooltip contains the following text:

```
void glutInit(int* pargc, char** argv)
void glutInit_ATEXIT_HACK(int* argcp, char** argv)
```

Deixando o ponteiro dentro dos parentesis da função aparecem os parametros
que devemos passar para a função funcionar e seu tipo de retorno
Nesse caso temos que passar um ponteiro pra int, e um ponteiro pra ponteiro
pra char

The screenshot shows the Code::Blocks IDE interface. The title bar reads "*main.c [NANO2023] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The status bar shows "Debug". A green double-headed arrow points between the toolbar and the code editor. The code editor window contains the following C code:

```
1 int main(int C, char *V[])
2 {
3     glutInit(&C, V);
4     return 0;
5 }
6
```

The code editor has line numbers 1 through 6 on the left. The first line starts with "int main". The second line begins with an opening brace "{". The third line contains the call to "glutInit". The fourth line ends with a closing brace "}" and a return statement. A green double-headed arrow also points from the right side of the code editor towards the bottom right corner.

Passamos os parametros necessarios a função

The screenshot shows the Code::Blocks IDE interface. The title bar reads "*main.c [NANO2023] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The status bar at the bottom shows "main(int C, char* V[]) : int". The left sidebar is titled "Management" and contains tabs for Projects, Files, and FSymbols. Under "Projects", there is a "Workspace" section with a "NANO2023" project containing a "Sources" folder with "main.c". The main code editor window is titled "*main.c" and displays the following C code:

```
1 int main(int C, char *V[])
2 {
3     glutInit(&C,V);
4
5     // INICIALIZANDO
6     glutCreateWindow();
7     return 0;
8 }
```

A tooltip box appears over the closing brace of the main function, containing the definitions for the two glutCreateWindow functions:

```
int glutCreateWindow(const char* title)
int glutCreateWindow_ATEXIT_HACK(const char* title)
```

Olhando os parametros que devem ser passados a função temos
Um const char* title que retorna um int
Nesse caso um endereço de memoria de algum texto que vamos escrever

*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main(int C, char* V[]) : int

Management X *main.c X

Projects Files FSymbols

Workspace NANO2023 Sources main.c

```
1 int main(int C, char *V[])
2 {
3     glutInit(&C, V);
4
5     // INICIALIZANDO
6     glutCreateWindow("BASIC WINDOW OPENGL");
7     return 0;
8 }
```



main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management

Projects Files FSymbols

Workspace NANO2023 Sources main.c

main.c X

```
int main(int C, char *V[])
{
    glutInit(&C,V);

    // INICIALIZANDO
    glutCreateWindow ("BASIC WINDOW OPENGL");
    return 0;
}
```

Click em build
Repare no erro que da vamos
concerta

Logs & others

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxygen X

File	Line	Message
C:\Users\brit...		== Build: Debug in NANO2023 (compiler: GNU GCC Compiler) ==
C:\Users\brit...	3	In function 'main':
C:\Users\brit...	6	warning: implicit declaration of function 'glutInit' [-Wimplicit-function-declaration]
		warning: implicit declaration of function 'glutCreateWindow' [-Wimplicit-function-decl...]
		== Build finished: 0 error(s), 2 warning(s) (0 minute(s), 0 second(s)) ==

C:\Users\britto\Desktop\C\NANO2023\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 6, Col 45, Pos 120 Insert Read/Write default 31°C Nublado 12:54

main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Management

Projects Files FSymbols

Workspace NANO2023 Sources main.c

main.c

```
int main(int C, char *V[])
{
    glutInit(&C,V);

    // INICIALIZANDO
    glutCreateWindow ("BASIC WINDOW OPENGL");
    return 0;
}
```

stdio.h MAN PAGE

Agora de run e veja o erro

Logs & others

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxygen X

File	Line	Message
C:\Users\brit...		== Build: Debug in NANO2023 (compiler: GNU GCC Compiler) ==
C:\Users\brit...	3	In function 'main':
C:\Users\brit...	3	warning: implicit declaration of function 'glutInit' [-Wimplicit-function-declaration]
C:\Users\brit...	6	warning: implicit declaration of function 'glutCreateWindow' [-Wimplicit-function-decl...
		== Build finished: 0 error(s), 2 warning(s) (0 minute(s), 0 second(s)) ==
		== Run: Debug in NANO2023 (compiler: GNU GCC Compiler) ==

C:\Users\brito\Desktop\C\NANO2023\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 6, Col 45, Pos 120 Insert Read/Write default 31°C Nublado 12:57

A screenshot of the Code::Blocks IDE interface. The main window shows a C file named 'main.c' with code related to OpenGL window creation. A red arrow points from the text 'Agora de run e veja o erro' down to the 'Build messages' tab in the 'Logs & others' panel. Another red arrow points to the warning message in the log: 'warning: implicit declaration of function 'glutInit' [-Wimplicit-function-declaration]'.

*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management X

Projects Files FSymbols

Workspace NANO2023 Sources main.c

*main.c X

```
1 #include <GL/
2
3 int main()
4 {
5     glut
6     // i
7     glut
8     retu
9
10
11 }
```

GL/freeglut.h
GL/freeglut_ext.h
GL/freeglut_std.h
GL/gl.h
GL/glaux.h
GL/glcorearb.h
GL/glext.h
GL/glu.h
GL/glut.h
GL/glxext.h
GL/wglext.h

*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug

<global>

Management

Projects Files FSymbols

Workspace

NANO2023

Sources

main.c

```
1 #include <GL/glut.h>
2
3 int main(int C, char *V[])
4 {
5     glutInit(&C,V);
6
7     // INICIALIZANDO
8     glutCreateWindow("BASIC WINDOW OPENGL");
9     return 0;
10 }
11
```

*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins Doxy

<global>

Management X

Projects Files FSymbols

Workspace NANO2023 Sources main.c

*main.c X

```
1 #include <GL/|
2 #include <GL/freeglut.h
3 int main()
4 {
5     glut
6     GL/gl.h
7     GL/glaux.h
8     // GL/glcorearb.h
9     glut
10    GL/glext.h
11    return
12 }
```

GL/glu.h
GL/glut.h
GL/glxext.h
GL/wglext.h

*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

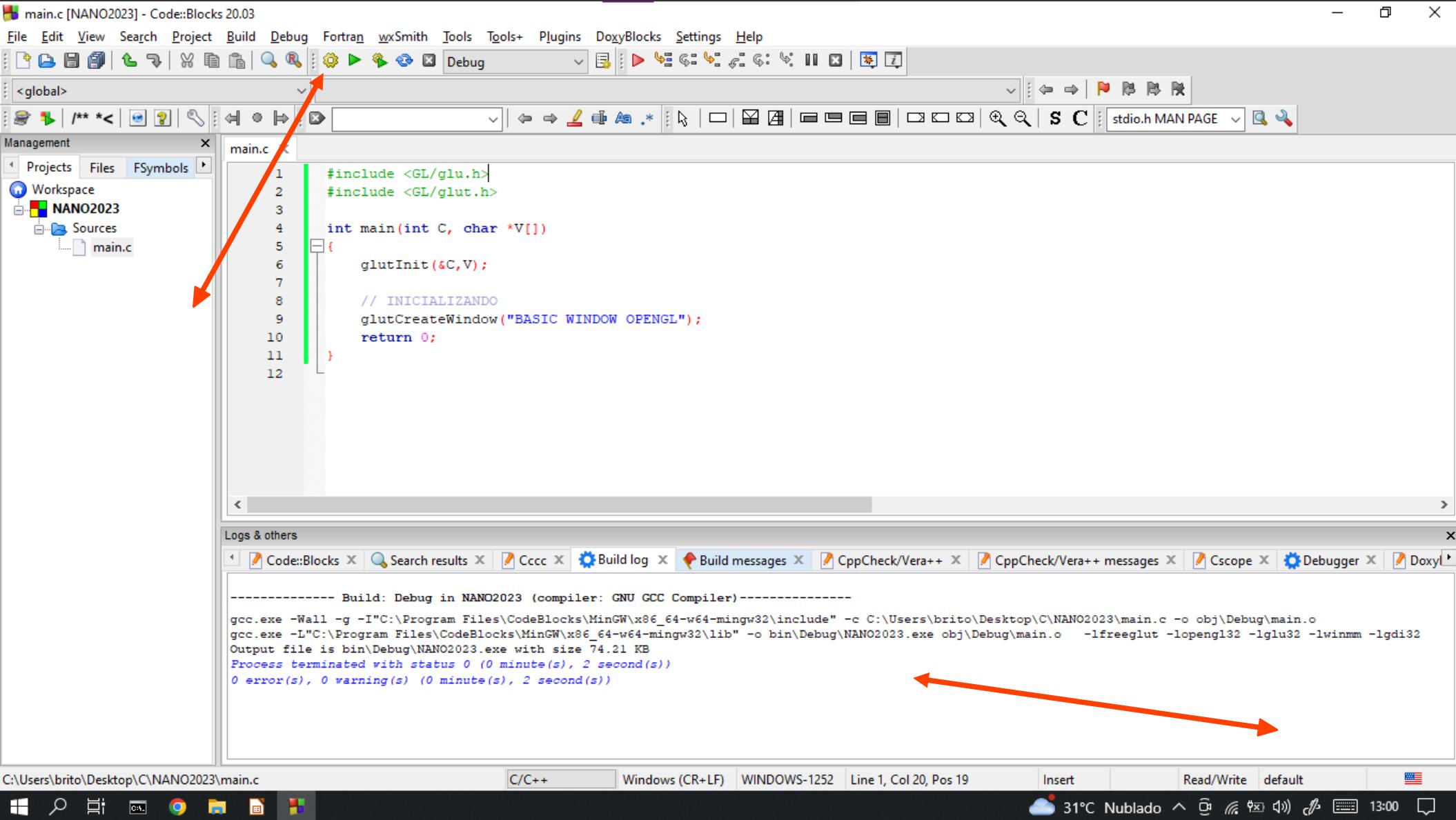
Management X

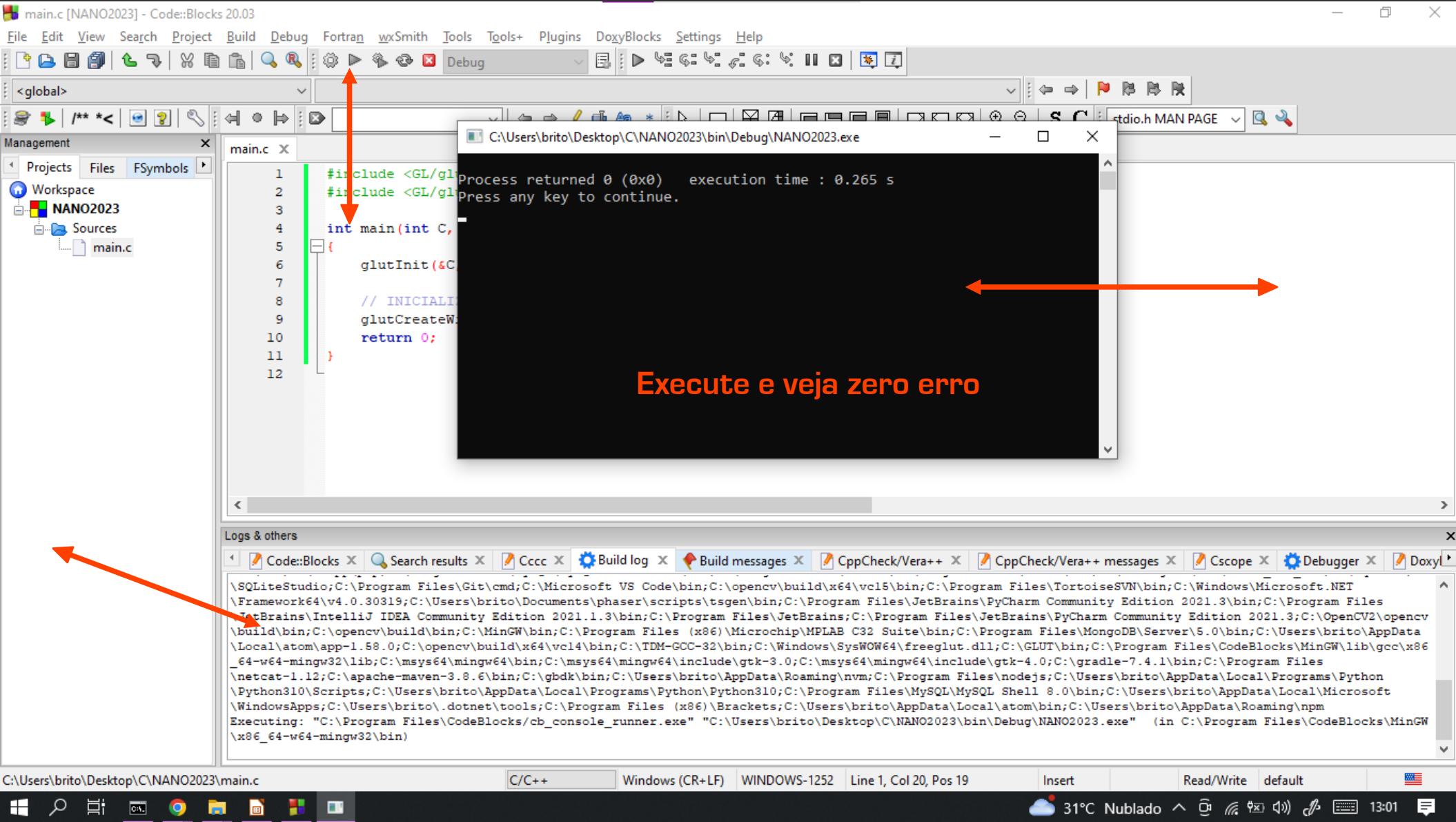
Projects Files FSymbols

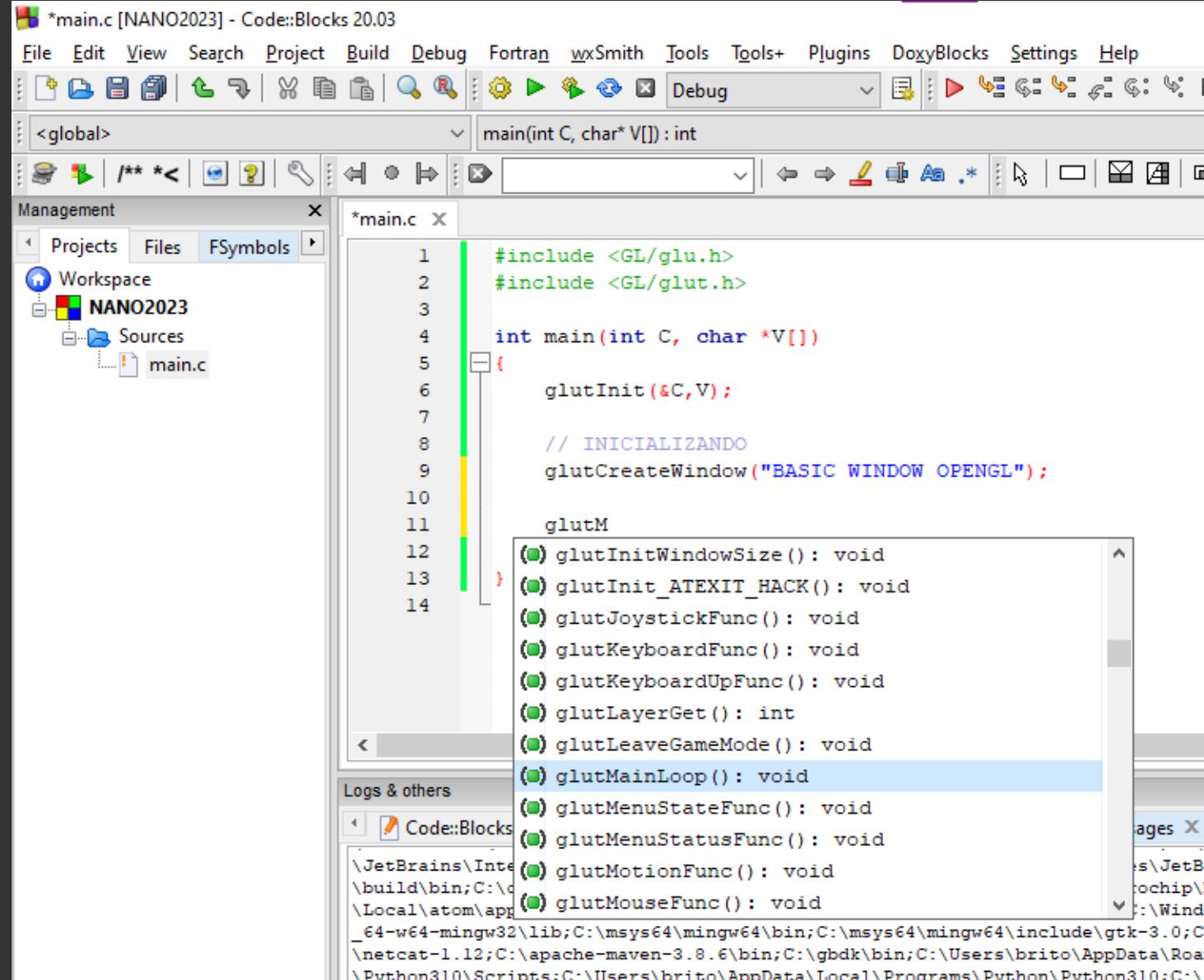
Workspace NANO2023 Sources main.c

*main.c X

```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 int main(int C, char *V[])
5 {
6     glutInit(&C, V);
7
8     // INICIALIZANDO
9     glutCreateWindow("BASIC WINDOW OPENGL");
10    return 0;
11 }
12
```







The screenshot shows the Code::Blocks 20.03 IDE interface. The title bar reads "*main.c [NANO2023] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Build. The status bar at the bottom shows "main(int C, char* V[]) : int". The left sidebar is titled "Management" and contains tabs for Projects, Files, and FSymbols. Under the Workspace tab, there is a project named "NANO2023" which contains a source file "main.c". The main editor window displays the following C code:

```
1 #include <GL/glut.h>
2
3
4 int main(int C, char *V[])
5 {
6     glutInit(&C, V);
7
8     // INICIALIZANDO
9     glutCreateWindow("BASIC WINDOW OPENGL");
10
11     glutMainLoop();
12
13 }
14
```

Função que não retorna nada e recebe nada



*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Python wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug

<global> main(int C, char* V[]) : int

Management X

Projects Files FSymbols

Workspace NANO2023 Sources main.c

```
1 #include <GL/glut.h>
2
3
4 int main(int C, char *V[])
5 {
6     glutInit(&C,V);
7
8     // INICIALIZANDO
9     glutCreateWindow("BASIC WINDOW OPENGL");
10
11     glutMainLoop();
12
13 }
14
```

```
C:\Users\brito\Desktop\C\NANO2023\bin\Debug\NANO2023.exe
freeglut (C:\Users\brito\Desktop\C\NANO2023\bin\Debug\NANO2023.exe): ERROR: No display callback registered for window 1
Process returned 1 (0x1) execution time : 0.283 s
Press any key to continue.

Vamos resolver
REGISTRANDO UMA CALLBACK
```

*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main(int C, char* V[]) : int

Management X

Projects Files FSymbols

Workspace NANO2023 Sources main.c

```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 int main(int C, char *V[])
5 {
6     glutInit(&C, V);
7
8     // INICIALIZANDO
9     glutCreateWindow("BASIC WINDOW OPENGL");
10
11     glutDis
12    	glutDisplayFunc(): void
13 }
14
15
```

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X

The screenshot shows the Code::Blocks IDE interface. The title bar reads "*main.c [NANO2023] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The status bar shows "main(int C, char* V[]) : int". The left sidebar has a "Management" tab with "Projects", "Files", and "FSymbols" options, showing a "Workspace" and "NANO2023" project with a "Sources" folder containing "main.c". The main editor window titled "*main.c" contains the following C code:

```
1 #include <GL/glut.h>
2
3
4 int main(int C, char *V[])
5 {
6     glutInit(&C,V);
7
8     // INICIALIZANDO
9     glutCreateWindow ("BASIC WINDOW OPENGL");
10
11    glutDisplayFunc();
12    glutMainLoop(); void glutDisplayFunc(void(* callback)(void))
13    return 0;
14
15 }
```

A green arrow points from the explanatory text below to the line "void glutDisplayFunc(void(* callback)(void))".

Retorna nada e recebe uma função de callback que é um ponteiro void
Ponteiros voids assumem qualquer tipo primitivo
Note que toda essa função é apenas um parametro

The screenshot shows the Code::Blocks IDE interface. The title bar reads "*main.c [NANO2023] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. A "Debug" dropdown is open. The left sidebar has a "Management" tab with "Projects", "Files", and "FSymbols" options. Under "Workspace", there is a project named "NANO2023" with a "Sources" folder containing "main.c". The main editor window displays the following C code:

```
1 #include <GL/glut.h>
2
3
4 int main(int C, char *V[])
5 {
6     glutInit(&C, V);
7
8     // INICIALIZANDO
9     glutCreateWindow("BASIC WINDOW OPENGL");
10
11    glutDisplayFunc(Draw);
12    glutMainLoop();
13 }
14
15
```

A green double-headed vertical arrow is positioned next to the closing brace of the main function, indicating where a new function definition should be placed.

Como ela recebe uma função vamos escrever uma função pra ele
nesse caso a Draw(){}
}

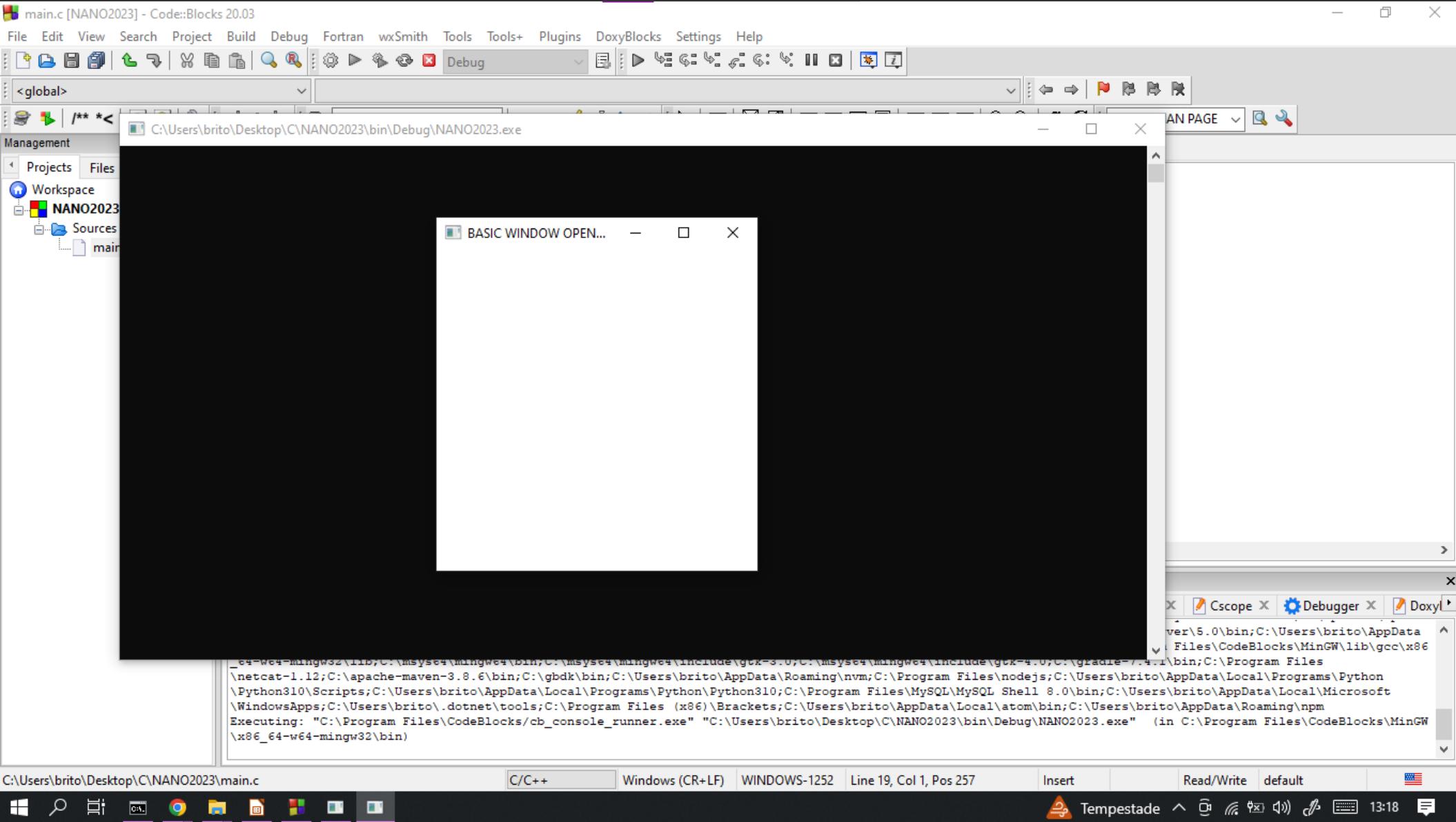
The screenshot shows the Code::Blocks IDE interface. The title bar reads "*main.c [NANO2023] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The left sidebar shows the Management view with Projects, Files, FSymbols, Workspace (NANO2023), Sources, and main.c selected. The main code editor window displays the following C code:

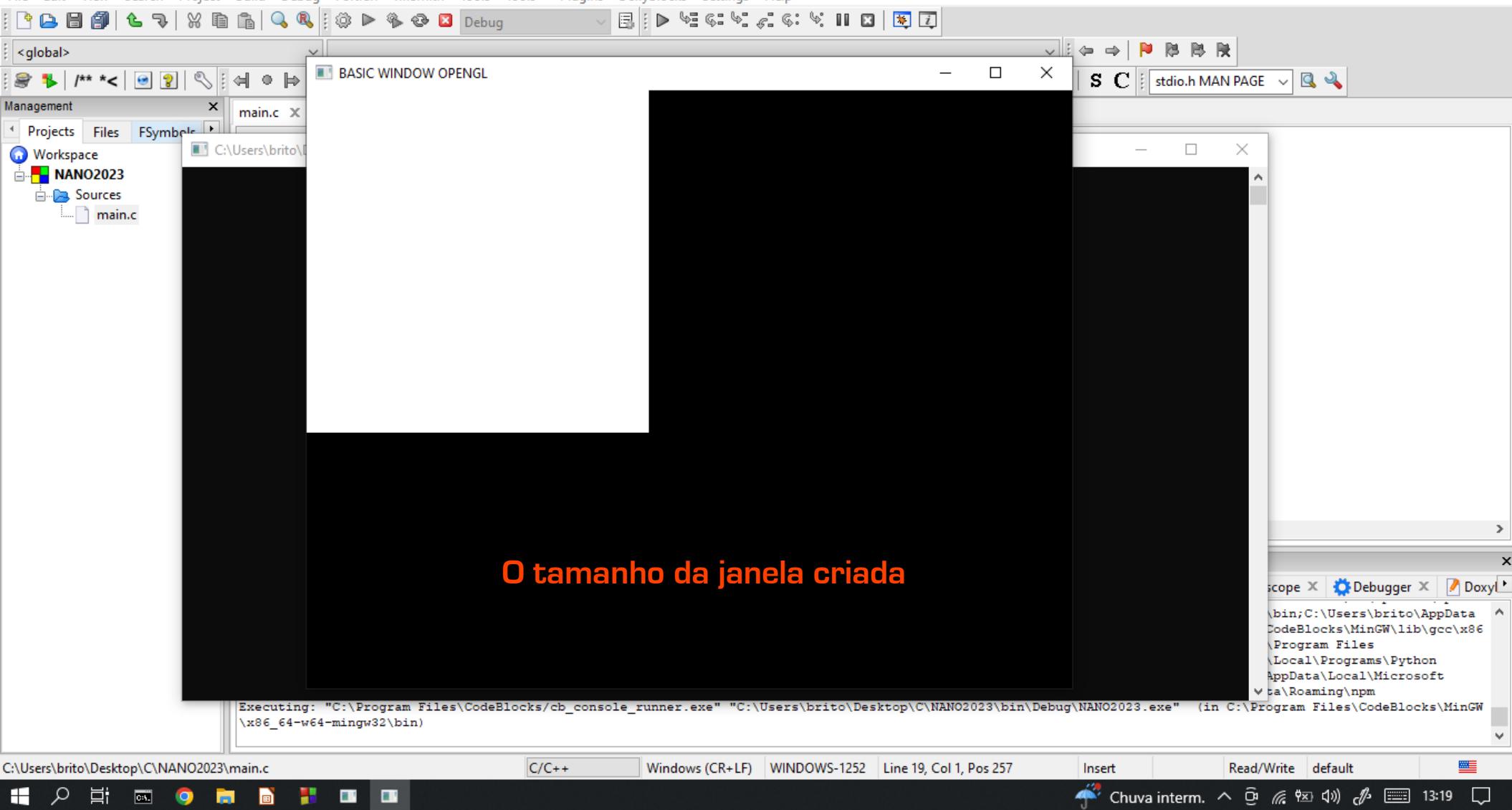
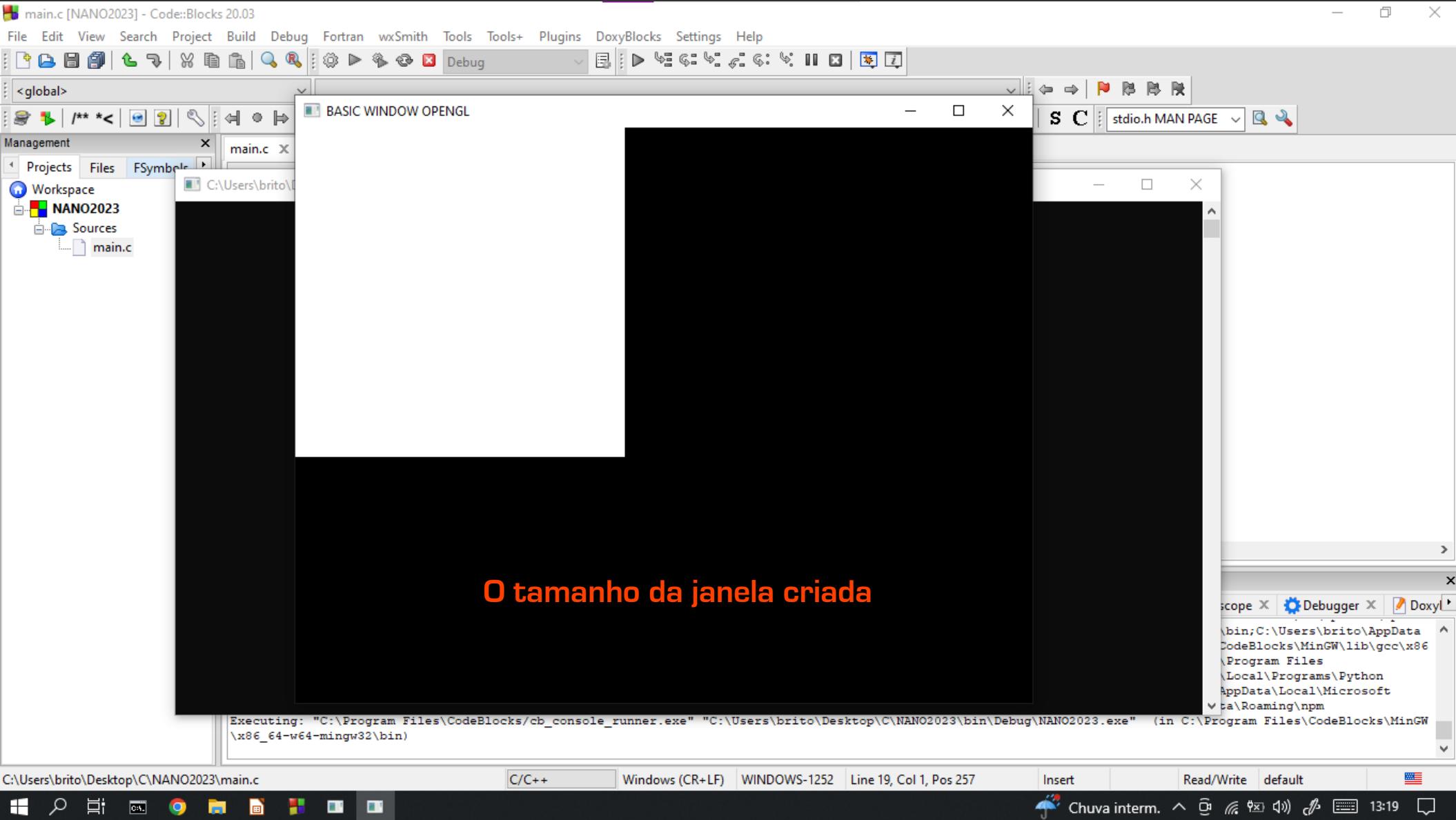
```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void Draw()
5 {
6 }
7
8 int main(int C, char *V[])
9 {
10     glutInit(&C, V);
11
12     // INICIALIZANDO
13     glutCreateWindow("BASIC WINDOW OPENGL");
14
15     glutDisplayFunc(Draw);
16     glutMainLoop();
17     return 0;
18 }
```

Three green arrows highlight specific parts of the code: one arrow points up at the top of the code editor, another points right at the opening brace of the Draw() function, and a third points down at the closing brace of the main() function.

Por fim compile e execute

Escrevemos a função nos
moldes que deveríamos
escrever não recebendo nada
e não retornando nada





<global>

Management

Projects Files FSymbols

Workspace

NANO2023

Sources

main.c

BASIC WINDOW OPENGL

S C

stdio.h MAN PAGE

C:\Users\brito\

scope Debugger Doxy

\bin;C:\Users\brito\AppData\CodeBlocks\MinGW\lib\gcc\x86\Program Files\Local\Programs\Python\Local\Microsoft\ta\Roaming\npm

Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023\bin\Debug\NANO2023.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\bin)



*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help



<global> main(int C, char* V[]) : int



Management

Projects Files FSymbols

Workspace
NANO2023
Sources
main.c

*main.c

```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void Draw()
5 {
6 }
7
8 int main(int C, char *V[])
9 {
10     glutInit(&C, V);
11     glutIni
12     # glutInit
13     (●) glutInit(): void
14     (●) glutInitDisplayMode(): void
15     (●) glutInitDisplayString(): void
16     (●) glutInitWindowPosition(): void
17     (●) glutInitWindowSize(): void
18     (●) glutInit_ATEXIT_HACK(): void
19 }
```

The screenshot shows the Code::Blocks 20.03 IDE interface. The title bar reads "*main.c [NANO2023] - Code::Blocks 20.03". The menu bar includes File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The status bar at the bottom shows "main(int C, char* V[]); int". The left sidebar is titled "Management" and contains "Projects", "Files", and "FSymbols" tabs, with "Workspace" and "NANO2023" selected. Under "Sources" in "NANO2023", there is a file named "main.c". The main code editor window displays the following C code:

```
#include <GL/glu.h>
#include <GL/glut.h>

void Draw()

int main(int C, char *V[])
{
    glutInit(&C,V);
    glutInitWindowSize();
    // INICIALIZANDO void glutInitWindowSize(int width, int height)
    glutCreateWindow("BASIC WINDOW OPENGL");

    glutDisplayFunc(Draw);
    glutMainLoop();
    return 0;
}
```

Não retorna nada e recebe dois numeros inteiros altura e largura da janela

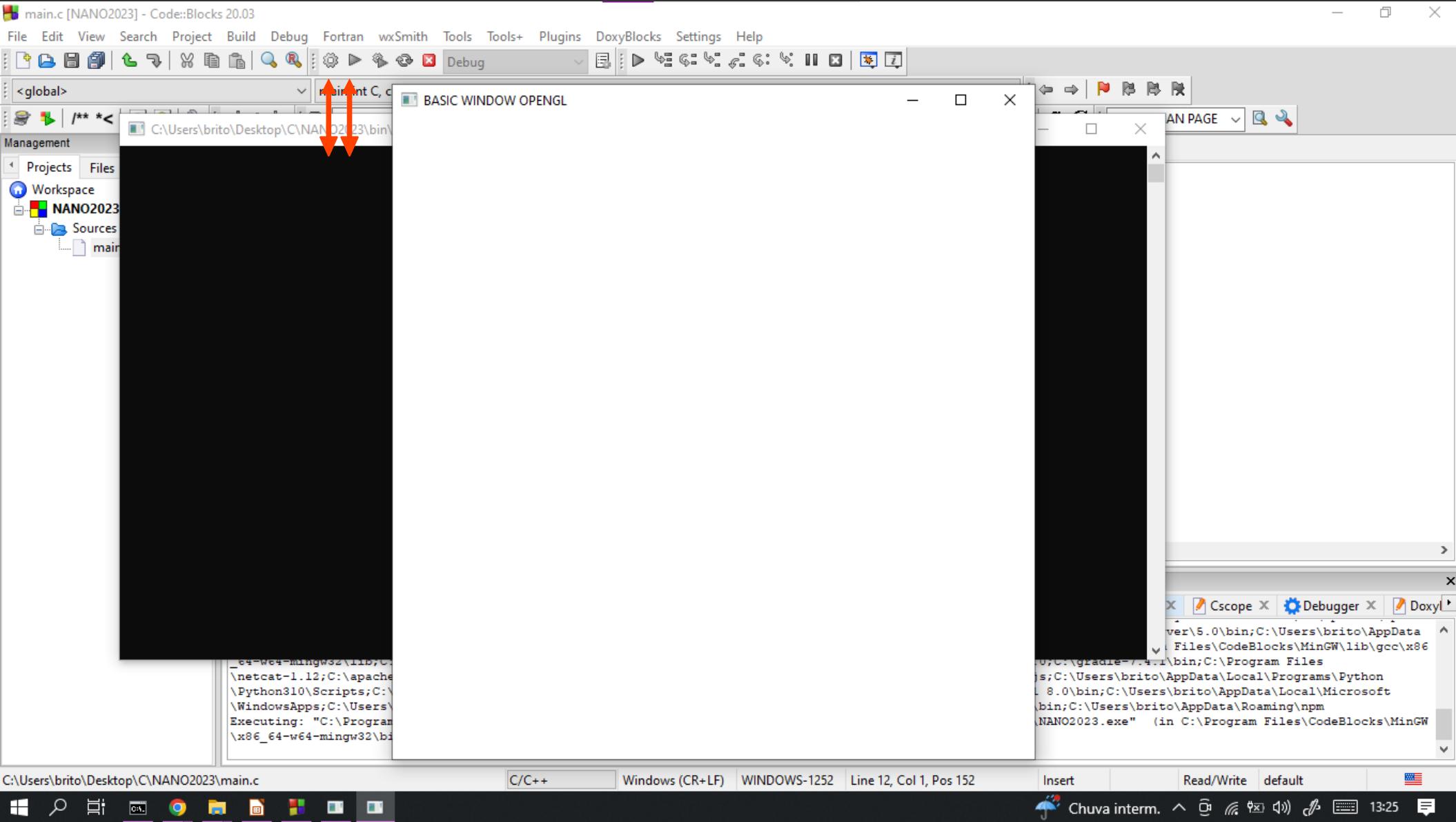
The screenshot shows the Code::Blocks IDE interface with the main.c file open in the code editor. The code implements a simple OpenGL application with a window of size 600x600.

```
#include <GL/glut.h>
#include <GL/glu.h>

void Draw()

int main(int C, char *V[])
{
    glutInit(&C, V);
    glutInitWindowSize(600, 600); ← Red arrow points here
    |
    // INICIALIZANDO
    glutCreateWindow("BASIC WINDOW OPENGL");

    glutDisplayFunc(Draw);
    glutMainLoop();
    return 0;
}
```



*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main(int C, char* V[]) : int

Management X

Projects Files FSymbols

Workspace NANO2023 Sources main.c

*main.c X

```
1 #include <GL/glut.h>
2
3
4 void Draw()
5 {
6
7 }
8 int main(int C, char *V[])
9 {
10     glutInit(&C,V);
11     glutInitWindowPosition();
12     glutIconifyWindow();
13     glutIdleFunc();
14     glutIgnoreKeyRepeat();
15     # glutInit
16     glutInitDisplayMode();
17     glutInitWindowSize();
18     glutInitDisplayString();
19     glutInitWindowPosition();
20 }
```

Logs & others

Code::Blocks X Search results X Ccc X Build Log X Build me

64-w64-mingw32\lib;C:\msys64\mingw64\bin;C:\msys64\mingw64\include\g

A screenshot of the Code::Blocks IDE interface. The central window displays a C program named 'main.c'. The cursor is positioned at the end of the line 'glutInitWindowPosition();'. A code completion dropdown menu is open, listing several member functions of the 'glutInit' class, all preceded by '(*)'. The function 'glutInitWindowPosition()' is highlighted in blue. The background of the code editor has vertical green and yellow highlighting bars.

*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug

<global>

main(int C, char* V[]); int

Management

Projects Files FSymbols

Workspace

NANO2023

Sources main.c

*main.c X

```
1 #include <GL/glut.h>
2
3
4 void Draw()
5 {
6
7 }
8 int main(int C, char *V[])
9 {
10     glutInit(&C, V);
11     glutInitWindowPosition();
12     glutInitWindowSize(600);
13
14     // INICIALIZANDO
15     glutCreateWindow("BASIC WINDOW OPENGL");
16
17     glutDisplayFunc(Draw);
18     glutMainLoop();
19     return 0;
20 }
```

Logs & others

Daqui pra frente não ficarei mostrando pois já sabemos como ler essas informações e como passa-las

*main.c [NANO2023] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug

<global>

main(int C, char* V[]) : int

Management

Projects Files FSymbols

Workspace NANO2023 Sources main.c

#include <GL/glu.h>
#include <GL/glut.h>

void Draw()
{
}

int main(int C, char *V[])
{
 glutInit(&C,V);
 glutInitWindowPosition(100,150);
 glutInitWindowSize(600,600);

 // INICIALIZANDO
 glutCreateWindow ("BASIC WINDOW OPENGL");

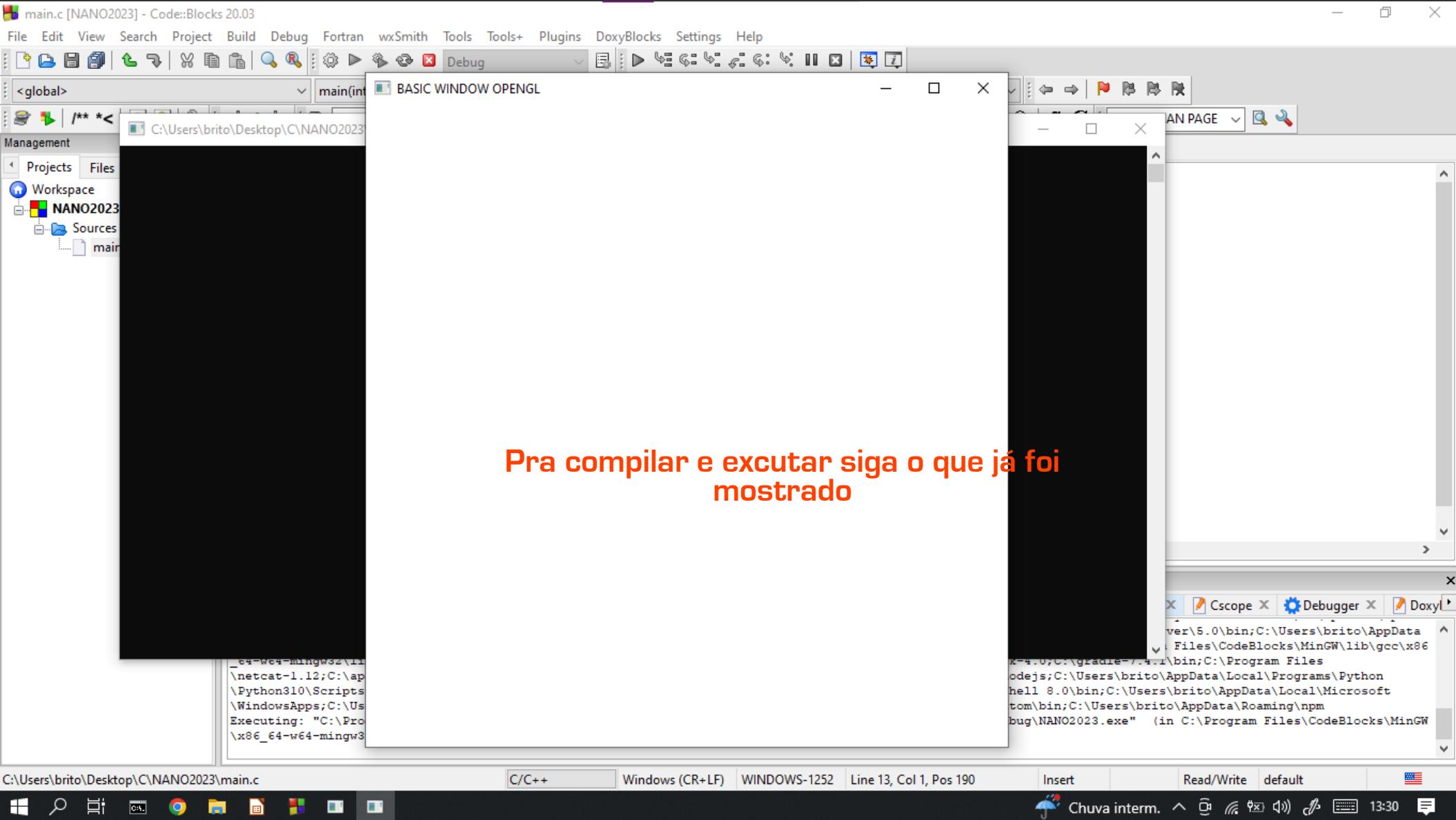
 glutDisplayFunc(Draw);
 glutMainLoop();
 return 0;
}

Logs & others

Code:Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxy X

```
_64-w64-mingw32\lib;C:\msys64\mingw64\bin;C:\msys64\mingw64\include\gtk-3.0;C:\msys64\mingw64\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm  
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023\bin\Debug\NANO2023.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\bin)  
Process terminated with status -1073741510 (0 minute(s), 3 second(s))
```

C:\Users\brito\Desktop\C\NANO2023\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 13, Col 1, Pos 190 Insert Modified Read/Write default Chuva interm. 13:30



**DEPOIS DE SAIR E VOLTAR ELE DA UM ERRO QUANDO FICA COM O LAYOUT EM PRETO
DARK**
É PRECISO RECOMPILAR TUDO DO ZERO
Apagar tudo e começar do zero mesmo

main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

```
#include <GL/glu.h>
#include <GL/glut.h>

void Draw()

int main(int C, char *V[])
{
    glutInit(&C,V);
    glutInitWindowPosition(250,50);
    glutInitWindowSize(600,600);

    // INICIALIZANDO
    glutCreateWindow("BASIC WINDOW OPENGL");

    glutDisplayFunc(Draw);
    glutMainLoop();
    return 0;
}
```

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxygen X

```
\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023-2\bin\Debug\NANO2023-2.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\bin)
Process terminated with status -1073741510 (0 minute(s), 2 second(s))
```

C:\Users\brito\Desktop\C\NANO2023-2\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 1, Col 1, Pos 0 Insert Read/Write default Chuva chegando 13:15

3-2
s
n.c

*main.c X

```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void Draw()
5 {
6
7 }
8 int main(int C, char *V[])
9 {
10     glutInit(&C, V);
11     glutInitWindowPosition(250, 50);
12     glutInitWindowSize(600, 600);
13     glutIni
14     # glutInit
15     (# glutInit(): void
16     (# glutInitDisplayMode(): void
17     (# glutInitDisplayString(): void
18     (# glutInitWindowPosition(): void
19     (# glutInitWindowSize(): void
20     (# glutInit_ATEXIT_HACK(): void
21 }
```

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera-

```
\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\U
\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bi
\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\
```

Management

Projects Files FSymbols

Workspace

NANO2023-2

Sources main.c

*main.c

```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void Draw()
5 {
6
7 }
8 int main(int C, char *V[])
9 {
10     glutInit(&C, V);
11     glutInitWindowPosition(250, 50);
12     glutInitWindowSize(600, 600);
13     glutInitDisplayMode|
14     // INICIALIZANDO
15     glutCreateWindow("BASIC WINDOW OPENGL");
16
17     glutDisplayFunc(Draw);
18     glutMainLoop();
19     return 0;
```

olhe e mantenha isso!

Note o simbolo de interrogação

OLHE TODA ESSA PARTE⁺

The screenshot shows the Code::Blocks IDE interface. The main window displays a C source file named `main.c` with the following code:

```
#include <GL/glu.h>
#include <GL/glut.h>

void Draw()
{
}

int main(int C, char *V[])
```

The `Draw()` function is highlighted with a green selection bar. A large green arrow points from the text above to the `Draw()` function in the code editor.

The bottom panel, titled "Logs & others", contains the "Build log" tab which shows the following output:

```
----- Run: Debug in NANO2023-2 (compiler: GNU GCC Compiler) -----
Checking for existence: C:\Users\brito\Desktop\C\NANO2023-2\bin\Debug\NANO2023-2.exe
Set variable: PATH=.;C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\lib;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\CodeBlocks\MinGW;C:\Program
Files\Java\jdk-17\bin;C:\Program Files\Microsoft\jdk-11.0.12.7-hotspot\bin;C:\Users\brito\AppData\Local\Programs\Python\Python39\Scripts;C:\Users\brito\AppData
\Local\Programs\Python\Python39;C:\Program Files (x86)\VMware\VMware Player\bin;C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program Files (x86)\Common
Files\Oracle\Java\javapath;C:\Windows\System32;C:\Windows;C:\Windows\System32\wbem;C:\Windows\System32\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:
\Program Files\Microsoft SQL Server\130\Tools\Binn;C:\Program Files\Java\javafx-sdk-16\bin;C:\Program Files\Java\jdk-11.0.11\bin;C:\Program Files\Java\jre1.8.0_
291\bin;C:\xampp\php;C:\Program Files\MySQL\MySQL Server 8.0\bin;C:\Program Files\dotnet;C:\Program Files\CMake\bin;C:\wxwidgets\lib\vc142_x64_dll;C:\sqlite3;C:
\SQLiteStudio;C:\Program Files\Git\cmd;C:\Microsoft VS Code\bin;C:\opencv\build\x64\vc15\bin;C:\Program Files\TortoiseSVN\bin;C:\Windows\Microsoft.NET
\Framework64\v4.0.30319;C:\Users\brito\Documents\phaser\scripts\tsgen\bin;C:\Program Files\JetBrains\PyCharm Community Edition 2021.3\bin;C:\Program Files
\JetBrains\IntelliJ IDEA Community Edition 2021.1.3\bin;C:\Program Files\JetBrains;C:\Program Files\JetBrains\PyCharm Community Edition 2021.3;C:\OpenCV2\opencv
\build\bin;C:\opencv\build\bin;C:\MinGW\bin;C:\Program Files (x86)\Microchip\MLAB C32 Suite\bin;C:\Program Files\MongoDB\Server\5.0\bin;C:\Users\brito\AppData
\Local\atom\app-1.58.0;C:\opencv\build\x64\vc14\bin;C:\TDM-GCC-32\bin;C:\Windows\SysWOW64\freeglut.dll;C:\GLUT\bin;C:\Program Files\CodeBlocks\MinGW\lib\gcc\x86
_64-w64-mingw32\lib;C:\msys64\mingw64\bin;C:\msys64\mingw64\include\gtk-3.0;C:\msys64\mingw64\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files
\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python
\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft
\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023-2\bin\Debug\NANO2023-2.exe" (in C:\Program Files\CodeBlocks
\MinGW\x86_64-w64-mingw32\bin)
Process terminated with status -1073741510 (0 minute(s), 2 second(s))
```

The status bar at the bottom indicates the file is `AN02023-2\main.c`, the editor mode is `C/C++`, the encoding is `Windows (CR+LF)`, the page size is `WINDOWS-1252`, the current position is `Line 13, Col 49, Pos 237`, and the file is in `Insert` mode with `Modified` status and `Read/Write` permissions.

The screenshot shows a software interface with a toolbar at the top and a main workspace below. The workspace includes a project management window on the left and a code editor window on the right.

Project Management:

- Projects: NANO2023-2
- Sources: main.c

Code Editor (main.c):

```
1 #include <GL/glut.h>
2 #include <GL/glu.h>
3
4 void Draw()
5 {
6
7 }
8 int main(int C, char *V[])
9 {
10     glutInit(&C, V);
11     glutInitWindowPosition(250, 50);
12     glutInitWindowSize(600, 600);
13     glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE); ←
14     // INICIALIZANDO
15     glutCreateWindow("BASIC WINDOW OPENGL");
16
17     glutDisplayFunc(Draw);
18     glutMainLoop();
19     return 0; +
20 }
21
22 // https://www.youtube.com/watch?v=wG_VaSr6a6c&t=1254s
23 // parada 20:55
24
```

The code in the editor is a C program using OpenGL and GLUT libraries. It defines a `Draw()` function and the `main()` function. The `main()` function initializes the GLUT library, creates a window titled "BASIC WINDOW OPENGL", sets the display mode to RGB and single-buffered, and starts the main loop. A green arrow points from the line `glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);` to the right margin of the editor. A green plus sign is located in the margin next to the closing brace of the `main()` function.

main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug

<global> main(int C, char* V[]) : int

Management

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

#include <GL/glut.h>
#include <GL/glu.h>

void Draw()
{
}

int main(int C, char *V[])
{
 glutInit(&C,V);
 glutInitWindowPosition(250,50);
 glutInitWindowSize(600,600);
 glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
 // INICIALIZANDO
 glutCreateWindow("BASIC WINDOW OPENGL");

 glutDisplayFunc(Draw);
}

nenhum erro ao dar build|

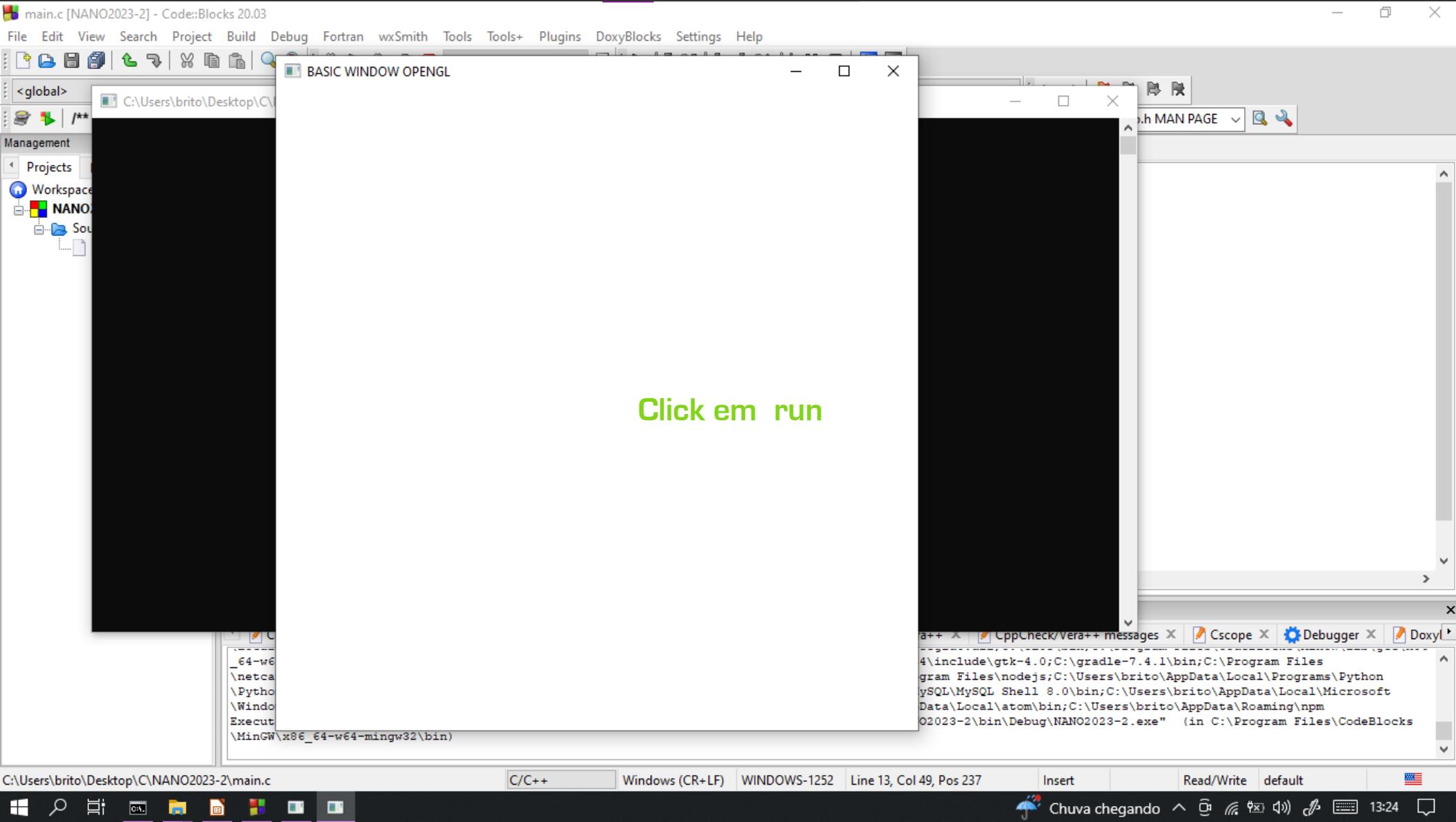
Logs & others

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxy

----- Build: Debug in NANO2023-2 (compiler: GNU GCC Compiler) -----
gcc.exe -Wall -g -I"C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\include" -c C:\Users\brito\Desktop\C\NANO2023-2\main.c -o obj\Debug\main.o
gcc.exe -L"C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\lib" -o bin\Debug\NANO2023-2.exe obj\Debug\main.o -lfreeglut -lopengl32 -lglu32 -lwinmm -lgdi32
Output file is bin\Debug\NANO2023-2.exe with size 75.55 KB
Process terminated with status 0 (0 minute(s), 1 second(s))
0 error(s), 0 warning(s) (0 minute(s), 1 second(s))

C:\Users\brito\Desktop\C\NANO2023-2\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 13, Col 49, Pos 237 Insert Read/Write default

Chuva chegando



*main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug

<global> main(int C, char* V[]) : int

Management X

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

main.c X

```
1 #include <GL/glut.h>
2
3
4 void Draw() // FUNÇÃO CR
5 {
6
7 }
8 int main(int C, char *V[])
9 {
10     glutInit(&C,V);
11     glutInitWindowPosition(250,50); // POSIÇÃO D
12     glutInitWindowSize(600,600); // TAMANHO E
13     glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE); // CORES DA
14     // INICIALIZANDO
15     glutCreateWindow("BASIC WINDOW OPENGL"); // TÍTULO DA
16
17     glc
18     (●) glCallList(): void
19     (●) glCallLists(): void
20     (●) GLclampd
21     (●) GLclampf
22 }
```

glc

- (●) glCallList(): void
- (●) glCallLists(): void
- (●) GLclampd
- (●) GLclampf
- (●) glClear(): void
- (●) glClearAccum(): void
- (●) glClearColor(): void ←
- (●) glClearDepth(): void
- (●) glClearIndex(): void
- (●) glClearStencil(): void
- (●) glClipPlane(): void
- (●) glColor3b(): void

Logs & others

Code:Blocks

\Python310\Scr... \WindowsApps;C: Executing: "C:\MinGW\x86_64-v... Process termina

Build messages C:ams\Python\Python310;C:\Program Files (x86)\Brackets;C:\Users\lner.exe" "C:\Users\brito\Desktop\main.c", 10 second(s)

```
0 int main(int c, char **v[])
1 {
2     glutInit(&C,V);
3     glutInitWindowPosition(250,50);           // POSIÇÃO DA JANELA
4     glutInitWindowSize(600,600);             // TAMANHO E LARGURA DA JANELA
5     glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE); // CORES DA JANELA A COR PADRÃO É BRANCO PODEMOS MUDAR
6     // INICIALIZANDO
7     glutCreateWindow("BASIC WINDOW OPENGL"); // TITULO DA JANELA
8
9     glClearColor()
10    glutDisplayFunc(glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha));
11    glutMainLoop();                         // FUNÇÃO OBRIGATÓRIA
12    return 0;
13
14 }
```

Recebe 3 parametros para red green e blue e um ultimo para alpha

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main(int C, char* V[]) : int

Management X

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

```
*main.c X
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc
5 {
6 }
7
8 int main(int C, char *V[])
9 {
10     glutInit(&C, V); // POSIÇÃO DA JANELA
11     glutInitWindowPosition(250,50); // TAMANHO E LARGURA DA JANELA
12     glutInitWindowSize(600,600);
13     glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE); // CORES DA JANELA A COR PADRÃO É BRANCO PODEMOS MUDAR
14     // INICIALIZANDO
15     glutCreateWindow("BASIC WINDOW OPENGL"); // TITULO DA JANELA
16
17     glClearColor(0,0,0,1); ←→
18     glutDisplayFunc(Draw);
19     glutMainLoop(); // FUNÇÃO OBRIGATORIA
20     return 0;
21 }
22
```

Logs & others

```
glutCreateWindow ("BASIC WINDOW OPENGL");           // TITULO DA JANELA  
  
glClearColor(0,0,0,1);                            // FUNÇÃO PARAS AS CORES DA JANELA  
glColor3f()  
glutDisp[void glColor3f(GLfloat red, GLfloat green, GLfloat blue)]  
glutMainLoop();          // FUNÇÃO OBRIGATÓRIA  
return 0;  
}
```

*main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug main(int C, char* V[]) : int stdio.h MAN PAGE

Management X

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

```
#include <GL/glut.h>
#include <GL/glu.h>

void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc(Draw);

int main(int C, char *V[])
{
    glutInit(&C,V);
    glutInitWindowPosition(250,50); // POSIÇÃO DA JANELA
    glutInitWindowSize(600,600); // TAMANHO E LARGURA DA JANELA
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE); // CORES DA JANELA A COR PADRÃO É BRANCO PODEMOS MUDAR
    // INICIALIZANDO
    glutCreateWindow("BASIC WINDOW OPENGL"); // TITULO DA JANELA

    glClearColor(0,0,0,1); // FUNÇÃO PARAS AS CORES DA JANELA
    glColor3f(1,0,0); // RECEBE VALORES FLOAT PARA DAR COR A JANELA
    glutDisplayFunc(Draw);
    glutMainLoop(); // FUNÇÃO OBRIGATORIA
    return 0;
}
```

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxy X

\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023-2\bin\Debug\NANO2023-2.exe" (in C:\Program Files\CodeBlocks)

*main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main(int C, char* V[]) : int

Management X

Projects Files FSymbols ▾

Workspace NANO2023-2

Sources main.c

```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc(Draw);
5 {
6 }
7
8 int main(int C, char *V[])
9 {
10     glutInit(&C,V); // POSIÇÃO DA JANELA
11     glutInitWindowPosition(250,50); // TAMANHO E LARGURA DA JANELA
12     glutInitWindowSize(600,600);
13     glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE); // CORES DA JANELA A COR PADRÃO É BRANCO PODEMOS MUDAR
14     // INICIALIZANDO
15     glutCreateWindow("BASIC WINDOW OPENGL"); // TITULO DA JANELA
16
17     MyInit(); // CRIANDO NOSSA FUNÇÃO
18     glClearColor(0,0,0,1); // FUNÇÃO PARA AS CORES DA JANELA
19     glColor3f(1,0,0); // RECEBE VALORES FLOAT PARA DAR COR A JANELA
20     glutDisplayFunc(Draw); // FUNÇÃO OBRIGATORIA
21     glutMainLoop();
22     return 0;
```

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X

```
\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\CV-NANO2023-2\bin\Debug\NANO2023-2.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\bin)
Process terminated with status -1073741510 (0 minute(s), 10 second(s))
```

```
15     glutCreateWindow("BASIC WINDOW OPENGL");      // TITULO DA JANELA
16
17     MyInit();                                     // CRIANDO NOSSA FUNÇÃO
18     glClearColor(0,0,0,1);                        // FUNÇÃO PARAS AS CORES DA JANELA
19     glColor3f(1,0,0);                            // RECEBE VALORES FLOAT PARA DAR COR A JANELA
20     glutDisplayFunc(Draw);
21     glutMainLoop();                             // FUNÇÃO OBRIGATORIA
22     return 0;
23
24 }
```

RECORDAMOS ESSAS DUAS FUNÇÕES E COLAMOS DENTRO DA NOSSA FUNÇÃO

*main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug MyInit() : void stdio.h MAN PAGE

Management Projects Files FSymbols NANO2023-2 Sources main.c

/* * < */ // FUNÇÃO PARAS AS CORES DA JANELA
// RECEBE VALORES FLOAT PARA DAR COR A JANELA
// FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc(Draw);
// POSIÇÃO DA JANELA
// TAMANHO E LARGURA DA JANELA
// CORES DA JANELA A COR PADRÃO É BRANCO PODEMOS MUDAR
// INICIALIZANDO
// TÍTULO DA JANELA
// CRIANDO NOSSA FUNÇÃO

```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void MyInit()
5 {
6     glClearColor(0,0,0,1);
7     glColor3f(1,0,0);
8 }
9 void Draw()
10 {
11 }
12
13 int main(int C, char *V[])
14 {
15     glutInit(&C,V);
16     glutInitWindowPosition(250,50);           // POSIÇÃO DA JANELA
17     glutInitWindowSize(600,600);             // TAMANHO E LARGURA DA JANELA
18     glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE); // CORES DA JANELA A COR PADRÃO É BRANCO PODEMOS MUDAR
19     // INICIALIZANDO
20     glutCreateWindow("BASIC WINDOW OPENGL"); // TÍTULO DA JANELA
21
22     MyInit();                                // CRIANDO NOSSA FUNÇÃO
```

Logs & others

Code::Blocks Search results Ccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger Doxygen

\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm Executing: "C:\Program Files\CodeBlocks\cb console runner.exe" "C:\Users\brito\Desktop\CNANO2023-2\bin\Debug\NANO2023-2.exe" (in C:\Program Files\CodeBlocks)

*main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Debug Draw() : void stdio.h MAN PAGE

<global>

Management

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

```
#include <GL/glu.h>
#include <GL/glut.h>

void MyInit()
{
    glClearColor(0,0,0,1);                                // FUNÇÃO PARA AS CORES DA JANELA
    glColor3f(1,0,0);                                    // RECEBE VALORES FLOAT PARA DAR COR A JANELA

    void Draw()                                         // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc(Draw);
    {
        glClear(GL_COLOR_BUFFER_BIT);                   // LIMPPANDO O BUFFER DA COR DE BACKGROUND
    }

    int main(int C, char *V[])
    {
        glutInit(&C,V);
        glutInitWindowPosition(250,50);                // POSIÇÃO DA JANELA
        glutInitWindowSize(600,600);                     // TAMANHO E LARGURA DA JANELA
        glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);   // CORES DA JANELA A COR PADRÃO É BRANCO PODEMOS MUDAR
        // INICIALIZANDO
        glutCreateWindow("BASIC WINDOW OPENGL");       // TITULO DA JANELA
        MyInit();                                       // CRIANDO NOSSA FUNÇÃO
        glutDisplayFunc(Draw);
        glutMainLoop();                                 // FUNÇÃO OBRIGATORIA
        return 0;
    }
}
```

Logs & others

Code::Blocks X Search results X Cccccc X Build log X Build messages X ConCheck/Vera++ X ConCheck/Vera++ messages X Csccccc X Debugger X Doxygen X

```
9     void Draw()
10    {
11        glClear(GL_COLOR_BUFFER_BIT);
12        glF
13    } (●) glFeedbackBuffer(): void
14    in (●) glFinish(): void
15    {
16        GLfloat
17        (●) glFlush(): void
18        (●) glFogf(): void
19        (●) glFogfv(): void
20        (●) glFogi(): void
21        (●) glFogiv(): void
22        (●) glFrontFace(): void
23        (●) glFrustum(): void
24
| GLUT_SINGLE | "OPENGL");
others
```

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Draw() : void

Management

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

```
#include <GL/glu.h>
#include <GL/glut.h>

void MyInit()
{
    glClearColor(0,0,0,1); // FUNÇÃO PARA AS CORES DA JANELA
    glColor3f(1,0,0); // RECEBE VALORES FLOAT PARA DAR COR A JANELA

}

void Draw()
{
    glClear(GL_COLOR_BUFFER_BIT); // LIMPANDO O BUFFER DA COR DE BACKGROUND
    glFlush(); // ATUALIZA OS PIXELS

}

int main(int C, char *V[])
{
    glutInit(&C,V);
    glutInitWindowPosition(250,50); // POSIÇÃO DA JANELA
    glutInitWindowSize(600,600); // TAMANHO E LARGURA DA JANELA
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE); // CORES DA JANELA A COR PADRÃO É BRANCO PODEMOS MUDAR
    // INICIALIZANDO
    glutCreateWindow("BASIC WINDOW OPENGL"); // TÍTULO DA JANELA
    MyInit(); // CRIANDO NOSSA FUNÇÃO
    glutDisplayFunc(Draw);
    glutMainLoop(); // FUNÇÃO OBRIGATÓRIA
}
```

Logs & others

CodeBlocks Search results Ccc Build log Build messages CppCheck/Vera CppCheck/Vera + messages CcConsole Debugger Devil

```
9   void Draw()                                // FUNÇÃO CRIADA PARA ATENDER
10  {
11      glClear(GL_COLOR_BUFFER_BIT);           // LIMPPANDO O BUFFER DA COR
12
13      /** DESENHANDO UM RETANGULO **/
14      glVertex2d();
15      glFlush(); void glVertex2d(GLdouble x, GLdouble y) //ATUALIZA OS PIXELS
16  }
17  int main(int C, char *V[])
18  {
19      glutInit(&C,V);
20      glutInitWindowPosition(250,50);          // POSIÇÃO DA JANELA
21      glutCreateWindow("500x500");             // TITULO E TAMANHO DA TELA
```

```
10  {
11      glClear(GL_COLOR_BUFFER_BIT);           // LIMPPANDO O BUFFER DA COR DE BACKGROUND
12
13      /** DESENHANDO UM RETANGULO **/
14      glVertex2d(-0.5, 0.5);                // CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO PLANO CARTESIANO
15      glVertex2d( 0.5, 0.5);
16      glVertex2d( 0.5,-0.5);
17      glVertex2d(-0.5,-0.5);
18      glFlush();                           // ATUALIZA OS PIXELS
19  }
20  int main(int C, char *V[])
21  {
22      glutInit(&C,V);
23      glutInitWindowPosition(250,50);        // POSIÇÃO DA JANELA
```

```
11     glClear(GL_COLOR_BUFFER_BIT);           // LIMPFANDO O BUFFER DA COR DE BACKGROUND
12
13     /** DESENHANDO UM RETANGULO **/
14     glBegin();                           // INICIA O RETANGULO
15     glVertex2d(-0.5, 0.5);              // CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO PLANO CARTESIANO
16     glVertex2d( 0.5, 0.5);
17     glVertex2d( 0.5,-0.5);
18     glVertex2d(-0.5,-0.5);
19     glFlush();                          // ATUALIZA OS PIXELS
20 }
21 int main(int C, char *V[])
22 {
23     glutInit(&C,V);
24     glutInitWindowPosition(250,50);      // POSIÇÃO DA JANELA
```

```
12
13     /** DESENHANDO UM RETANGULO ***/
14     glBegin();                                // INICIA O RETANGULO
15     glVertex2d(-0.5, 0.5);                    // CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO PLANO CARTESIANO
16     glVertex2d( 0.5, 0.5);
17     glVertex2d( 0.5,-0.5);
18     glVertex2d(-0.5,-0.5);
19     glEnd();                                  // TERMINA O RETANGULO
20     glFlush();                                // ATUALIZA OS PIXELS
21 }
22 int main(int C, char *V[])
23 {
```



```
/** DESENHANDO UM RETANGULO ***/
glBegin(GL_POLYGON);
glVertex3f(0.0, 0.0, 0.0);
glVertex3f(1.0, 0.0, 0.0);
glVertex3f(1.0, 1.0, 0.0);
glVertex3f(0.0, 1.0, 0.0);
glEnd();
glFlush();
}

int main(int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutCreateWindow("Retângulo");
    glutDisplayFunc(display);
    glutMainLoop();
}
```

GL_PIXEL_MAP_I_TO_R_SIZE
GL_PIXEL_MAP_R_TO_R
GL_PIXEL_MAP_R_TO_R_SIZE
GL_PIXEL_MAP_S_TO_S
GL_PIXEL_MAP_S_TO_S_SIZE
GL_PIXEL_MODE_BIT
GL_POINT
GL_POINTS
GL_POINT_BIT
GL_POINT_SIZE
GL_POINT_SIZE_GRANULARITY
GL_POINT_SIZE_RANGE

D:\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\

*main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

Draw() : void

include <GL/glut.h>

#include <GL/glut.h>

void MyInit()

{

glClearColor(0,0,0,1);
 glColor3f(1,0,0);

}

void Draw()

{

glClear(GL_COLOR_BUFFER_BIT);

 /* DESENHANDO UM RETANGULO */
 glBegin(GL_POINTS);
 glVertex2d(-0.5, 0.5);
 glVertex2d(0.5, 0.5);
 glVertex2d(0.5,-0.5);
 glVertex2d(-0.5,-0.5);
 glEnd();
 glFlush();

int main(int C, char *V[])

{

glutInit(&C,V);

// FUNÇÃO PARAS AS CORES DA JANELA
 // RECEBE VALORES FLOAT PARA DAR COR A JANELA

// FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc(Draw);

// LIMPANDO O BUFFER DA COR DE BACKGROUND

// INICIA O RETANGULO
 // CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO PLANO CARTESIANO

// TERMINA O RETANGULO
 // ATUALIZA OS PIXELS

POR FIM COMPILE E EXECUTE

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxy

\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023-2\bin\Debug\NANO2023-2.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\bin)
Process terminated with status -1073741510 (0 minute(s) 10 second(s))

C:\Users\brito\Desktop\C\NANO2023-2\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 14, Col 50, Pos 570 Insert Modified Read/Write default Chuva chegando

main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

C:\Users\brito\Desktop\C\NANO2023-2\main.c

BASIC WINDOW OPENGL

stdio.h MAN PAGE

CONTRAPESOS

CONFIGURAMOS UMA JANELA DE COR PRETA

Chuva chegando

Windows (CR+LF) WINDOWS-1252 Line 14, Col 50, Pos 565 Insert Read/Write default

13:55

The screenshot shows the Code::Blocks IDE interface. On the left is the project tree with a single file 'main.c' selected. The main editor window is titled 'BASIC WINDOW OPENGL' and contains a blank black screen. To the right of the editor is a terminal window showing the command-line interface with various environment variables and paths listed. The status bar at the bottom indicates the current file is 'main.c' and the line number is 'Line 14, Col 50, Pos 565'. A large green watermark text 'CONFIGURAMOS UMA JANELA DE COR PRETA' is overlaid in the center of the black window area.

```
/** DESENHANDO UM RETANGULO */
glPoi
(●) glPassThrough(): void
(●) glPixelMapfv(): void
(●) glPixelMapuiv(): void
(●) glPixelMapusv(): void
(●) glPixelStoref(): void
(●) glPixelStorei(): void
(●) glPixelTransferf(): void
(●) glPixelTransferi(): void
(●) glPixelZoom(): void
(●) glPointSize(): void
(●) glPolygonMode(): void
(●) glPolygonOffset(): void
```

Adicionando o tamanho dos pontos

The screenshot shows the Code::Blocks IDE interface with the following details:

- Title Bar:** *main.c [NANO2023-2] - Code::Blocks 20.03
- Menu Bar:** File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help
- Toolbar:** Includes icons for file operations, search, and build.
- Project Explorer:** Management window showing Projects, Files, FSymbols, Workspace, and NANO2023-2 project with Sources and main.c.
- Code Editor:** The main window displays the code for main.c:

```
#include <GL/glu.h>
#include <GL/glut.h>

void MyInit()
{
    glClearColor(0,0,0,1);
    glColor3f(1,0,0);
}

void Draw()
{
    glClear(GL_COLOR_BUFFER_BIT);

    /* DESENHANDO UM RETANGULO */
    glPointSize(5);
    glBegin(GL_POINTS);
    glVertex2d(-0.5, 0.5);
    glVertex2d( 0.5, 0.5);
    glVertex2d( 0.5,-0.5);
    glVertex2d(-0.5,-0.5);
    glEnd();
    glFlush();
}

// FUNÇÃO PARA AS CORES DA JANELA
// RECEBE VALORES FLOAT PARA DAR COR A JANELA

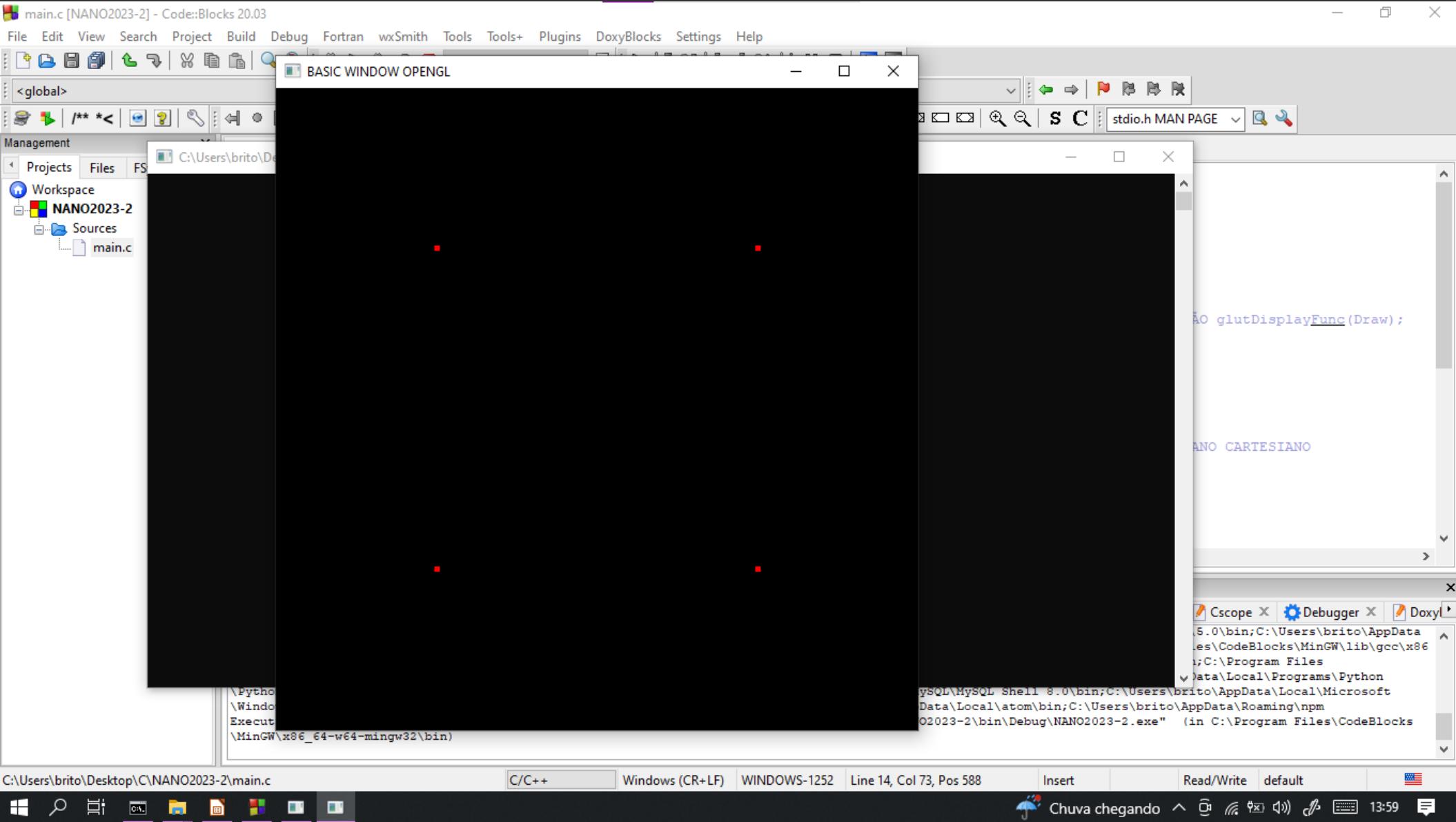
// FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc(Draw);

// LIMPPANDO O BUFFER DA COR DE BACKGROUND

// TAMANHO DO RETANGULO
// INICIA O RETANGULO
// CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO PLANO CARTESIANO

// TERMINA O RETANGULO
// ATUALIZA OS PIXELS
```
- Logs & others:** Shows the build log with the following output:

```
_64-w64-mingw32\lib;C:\msys64\mingw64\bin;C:\msys64\mingw64\include\gtk-3.0;C:\msys64\mingw64\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023-2\bin\Debug\NANO2023-2.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\bin)
Process terminated with status -1073741510 (0 minute(s), 6 second(s))
```



```
8 }  
9     void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayF  
10 {  
11     glClear(GL_COLOR_BUFFER_BIT); // LIMPANDO O BUFFER DA COR DE BACKGROUND  
12  
13     /** DESENHANDO UM RETANGULO **/  
14     //glPointSize(5); // TAMANHO DO RETANGULO  
15     glBegin(GL_LINES); // INICIA O RETANGULO  
16     glVertex2d(-0.5, 0.5); // CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO PLANO CARTESIANO  
17     glVertex2d( 0.5, 0.5);  
18     glVertex2d( 0.5,-0.5);  
19     glVertex2d(-0.5,-0.5);  
20     glEnd(); // TERMINA O RETANGULO  
21     glFlush(); // ATUALIZA OS PIXELS
```

... & others

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debug

COMENTE OU EXCLUA A LINHA DE CIMA E VAMOS DESENHAR LINES

*main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Draw j: void

Management

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

#include <GL/glut.h>

void MyInit()

{

glClearColor(0,0,0,1);
 glColor3f(1,0,0);

}

void Draw()

{

glClear(GL_COLOR_BUFFER_BIT);

 /* DESENHANDO UM RETANGULO */
 //glPointSize(5);
 glBegin(GL_LINES);
 glVertex2d(-0.5, 0.5);
 glVertex2d(0.5, 0.5);
 glVertex2d(0.5,-0.5);
 glVertex2d(-0.5,-0.5);
 glEnd();
 glFlush();

}

int main(int C, char *V[])

{

glutInit(&C,V);
 glutInitWindowPosition(250,50);

// POSTURA DA JANELA

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxy

_64-w64-mingw32\lib;C:\msys64\mingw64\bin;C:\msys64\mingw64\include\gtk-3.0;C:\msys64\mingw64\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Program Files (x86)\Brackets;C:\Users\brito\AppData\Local\atom\bin;C:\Users\brito\AppData\Roaming\npm

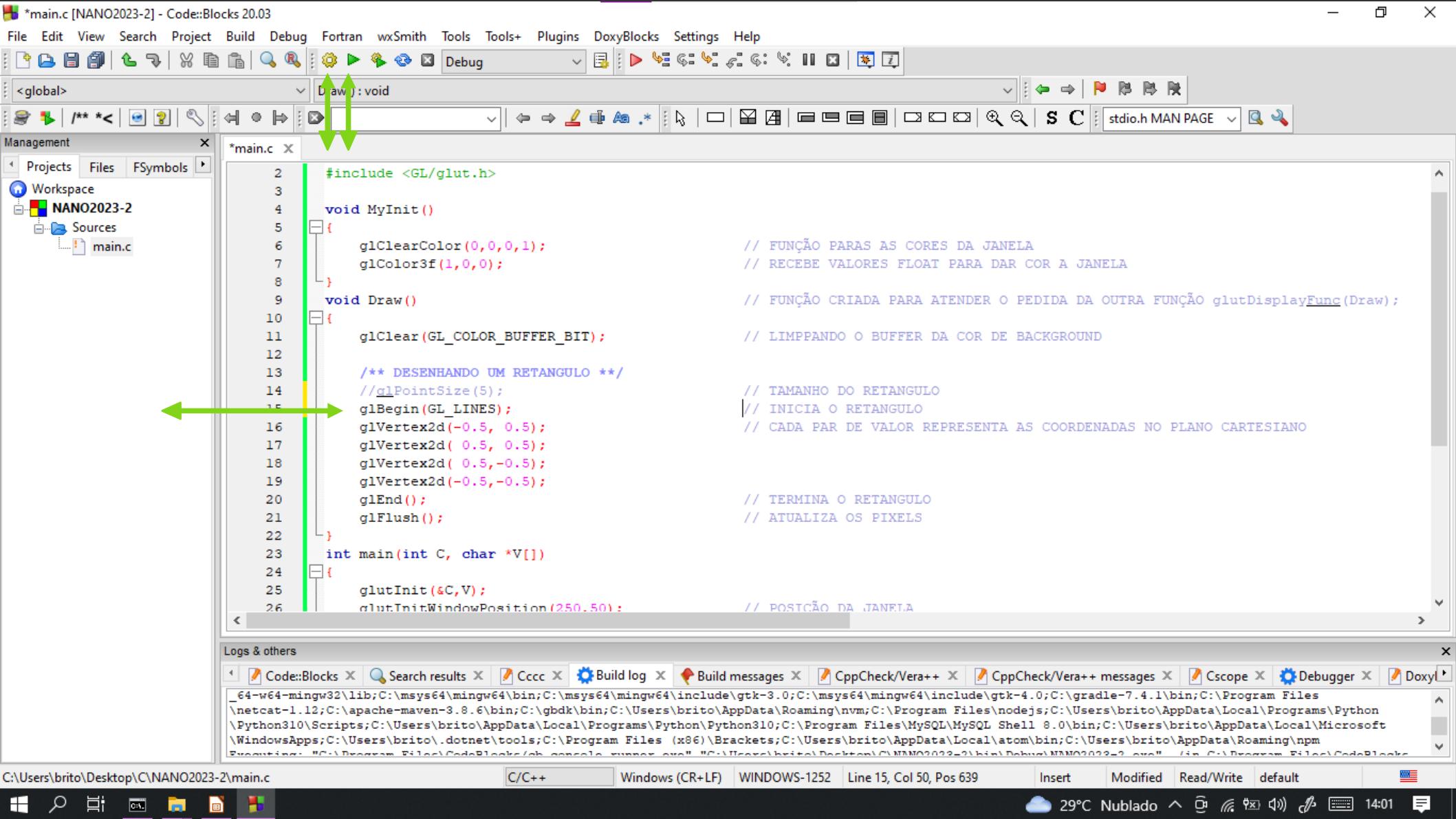
Executing: "C:\Program Files\Code::Blocks\cb_console_runner.exe" "C:\Users\brito\Desktop\NANO2023-2\main.c" (in C:\Users\brito\Desktop\NANO2023-2)

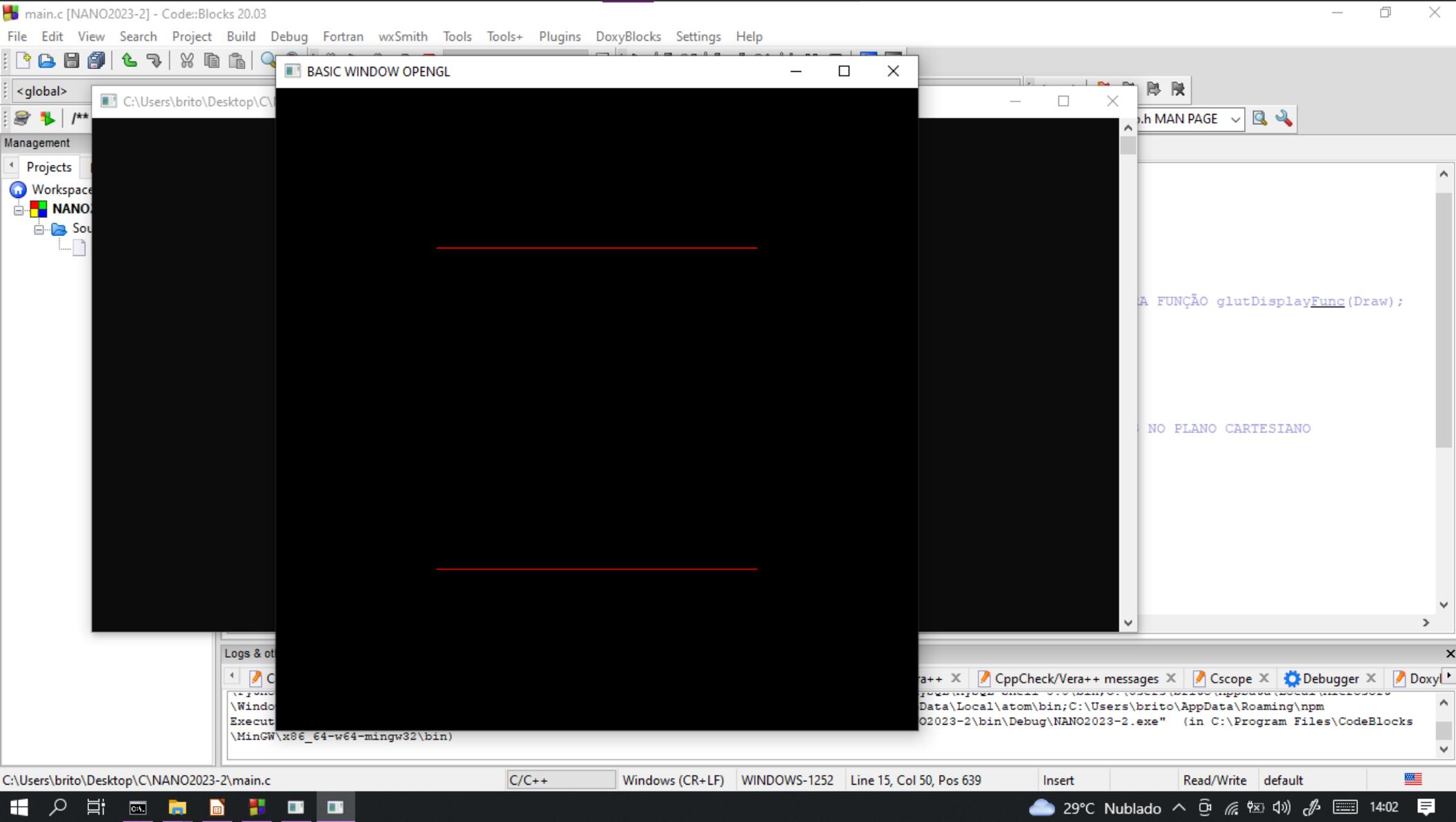
C:\Users\brito\Desktop\C\NANO2023-2\main.c

C/C++ Windows (CR+LF) WINDOWS-1252 Line 15, Col 50, Pos 639 Insert Modified Read/Write default

29°C Nublado

14:01





```
12  
13     /** DESENHANDO UM RETANGULO **/  
14     //glPointSize(5);  
15     glBegin(GL_LINE_ );  
16     glVertex # GL_LINE_BIT  
17     glVertex # GL_LINE_LOOP  
18     glVertex # GL_LINE_RESET_TOKEN  
19     glVertex # GL_LINE_SMOOTH  
20     glEnd # GL_LINE_SMOOTH_HINT  
21     glFlush # GL_LINE_STIPPLE  
22 }
```

*main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

raw void

Management

Projects Files FSymbols

Workspace NANO2023-2 Sources main.c

#include <GL/glut.h>

void MyInit()

{

glClearColor(0,0,0,1);
 glColor3f(1,0,0);

}

void Draw()

{

glClear(GL_COLOR_BUFFER_BIT);

 /* DESENHANDO UM RETANGULO */
 //glPointSize(5);
 glBegin(GL_LINE_LOOP);
 glVertex2d(-0.5, 0.5);
 glVertex2d(0.5, 0.5);
 glVertex2d(0.5,-0.5);
 glVertex2d(-0.5,-0.5);
 glEnd();
 glFlush();

}

int main(int C, char *V[])

{

glutInit(&C,V);
 glutInitWindowPosition(250,50);

// FUNÇÃO PARAS AS CORES DA JANELA
// RECEBE VALORES FLOAT PARA DAR COR A JANELA

// FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc(Draw);

// LIMPANDO O BUFFER DA COR DE BACKGROUND

// TAMANHO DO RETANGULO
// INICIA O RETANGULO
// CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO PLANO CARTESIANO

// TERMINA O RETANGULO
// ATUALIZA OS PIXELS

// POSTOÃO DA JANELA

Logs & others

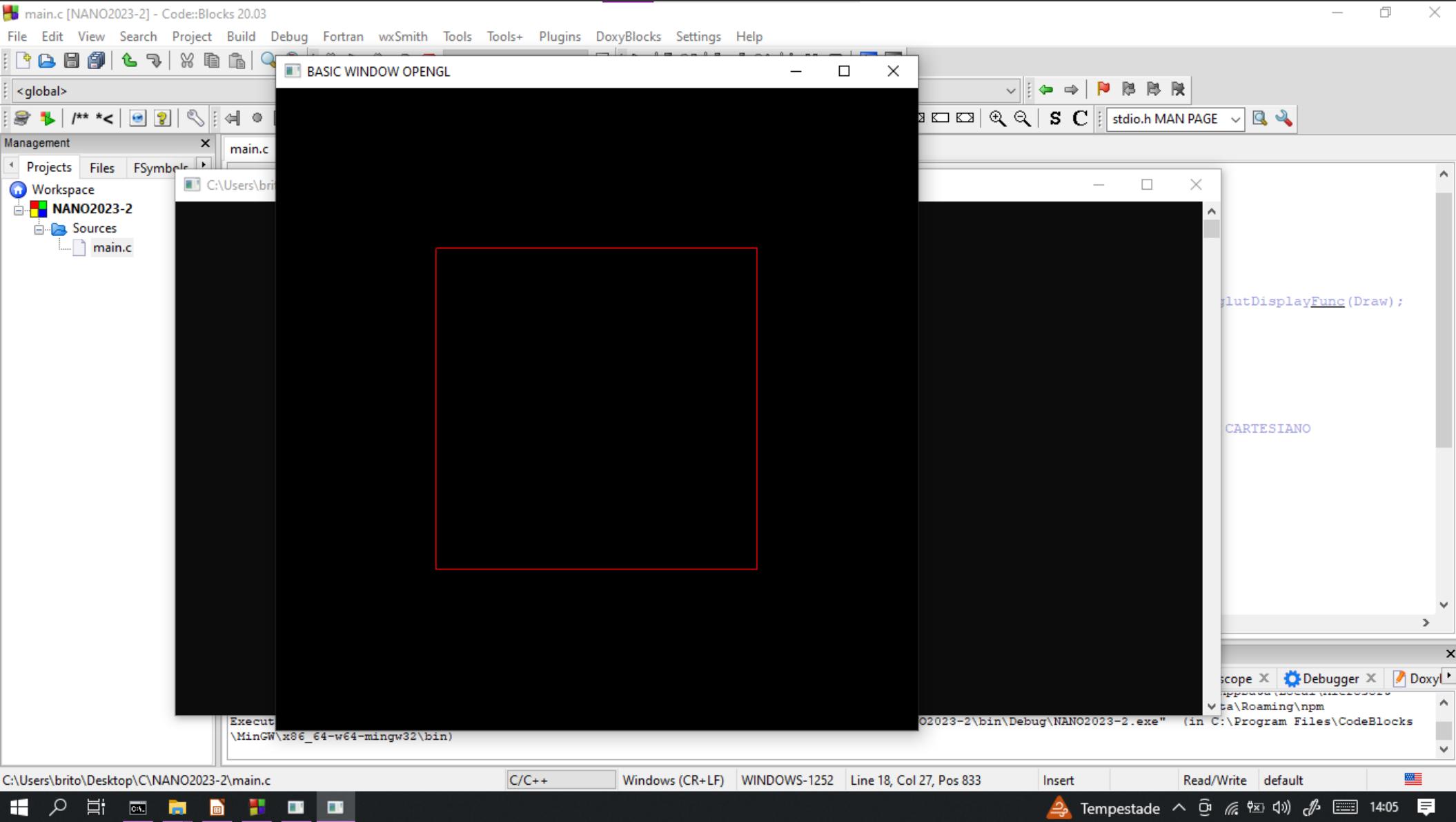
Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxygen X

\MinGW\x86_64-w64-mingw32\bin)
Process terminated with status -1073741510 (0 minute(s), 5 second(s))

C:\Users\brito\Desktop\C\NANO2023-2\main.c

C/C++ Windows (CR+LF) WINDOWS-1252 Line 18, Col 27, Pos 833 Insert Modified Read/Write default

Tempestade



Desenhando um retangulo sólido

*main.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Draw(); void

Management Projects Files FSymbols NANO2023-2 Sources main.c

```
#include <GL/glut.h>

void MyInit()
{
    glClearColor(0,0,0,1);
    glColor3f(1,0,0);
}

void Draw()
{
    glClear(GL_COLOR_BUFFER_BIT);

    /* DESENHANDO UM RETANGULO */
    //glPointSize(5);
    glBegin(GL_POLYGON);
    glVertex2d(-0.5, 0.5);
    glVertex2d( 0.5, 0.5);
    glVertex2d( 0.5,-0.5);
    glVertex2d(-0.5,-0.5);
    glEnd();
    glFlush();
}

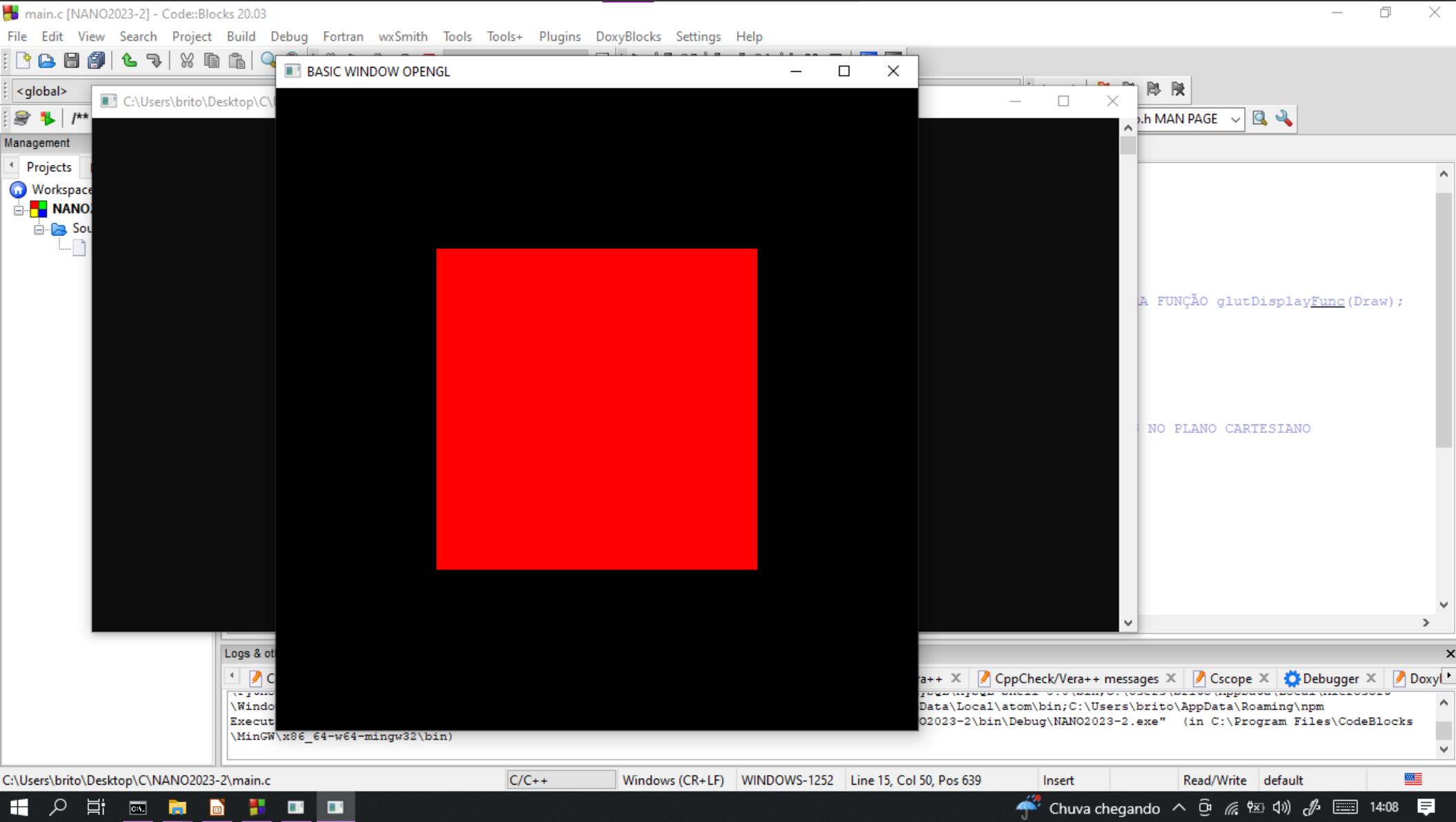
int main(int C, char *V[])
{
    glutInit(&C,V);
    glutInitWindowPosition(250,50);
    // POSIÇÃO DA JANELA
}
```

Logs & others

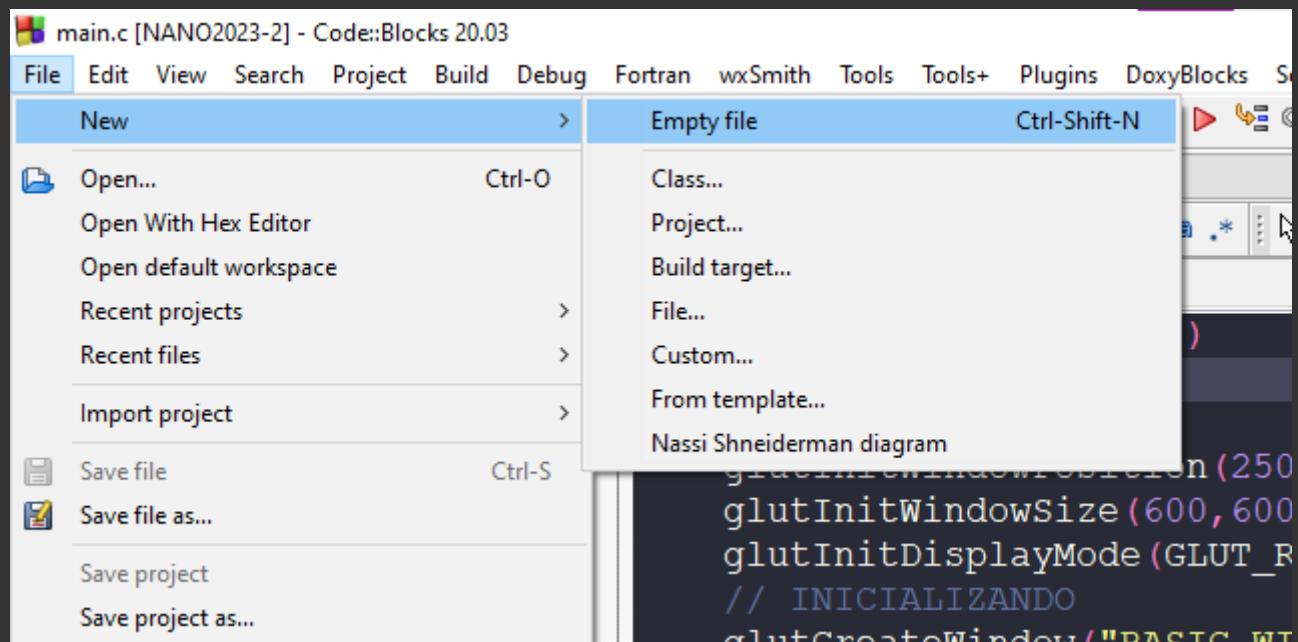
Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxygen X

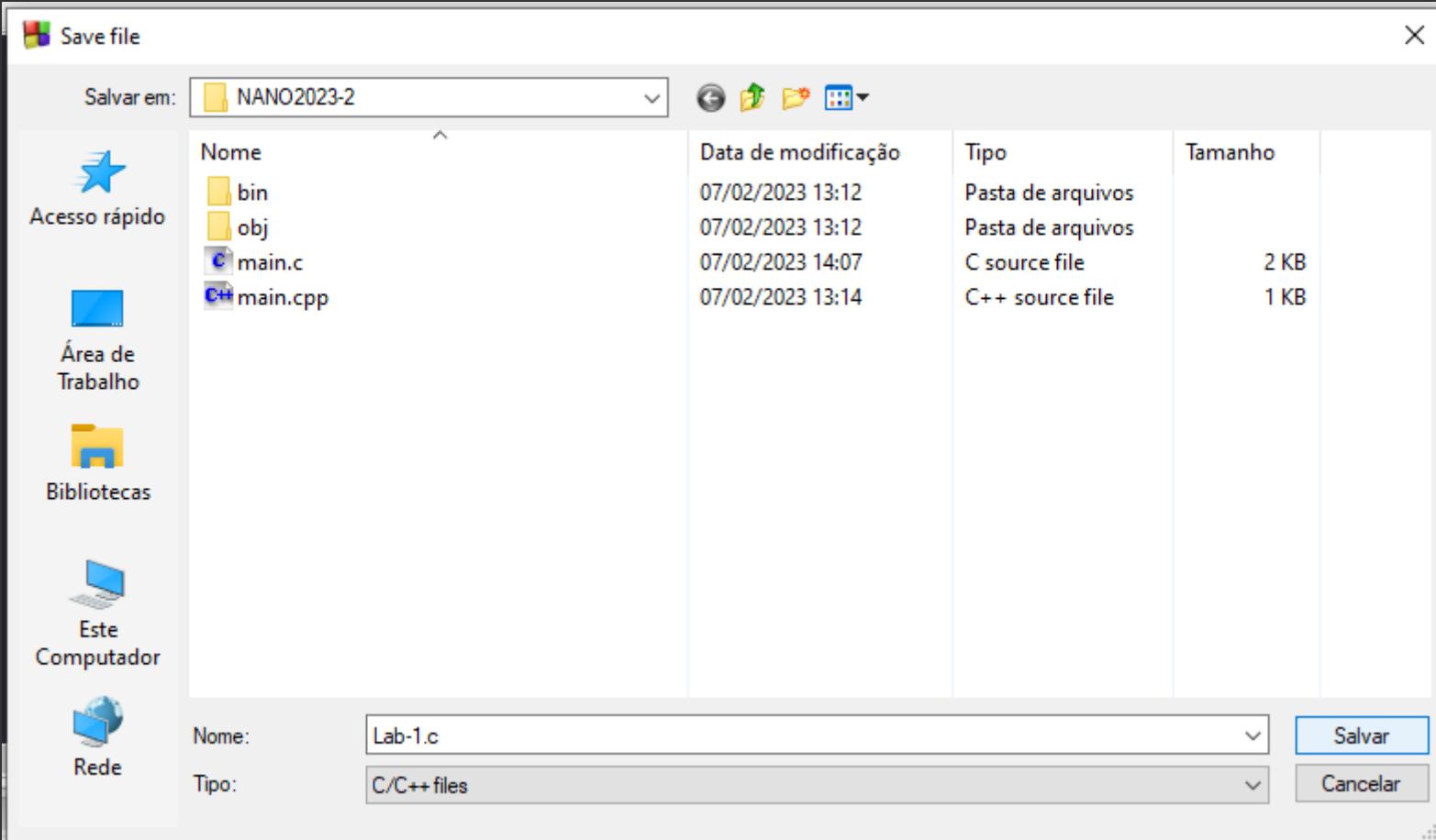
\MinGW\x86_64-w64-mingw32\bin) Process terminated with status -1073741510 (0 minute(s), 5 second(s))

C:\Users\brito\Desktop\C\NANO2023-2\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 15, Col 50, Pos 639 Insert Modified Read/Write default Chuva chegando 14:07



DRAW CUBE
COM OPENGL





Multiple selection

Select the targets this file should belong to:

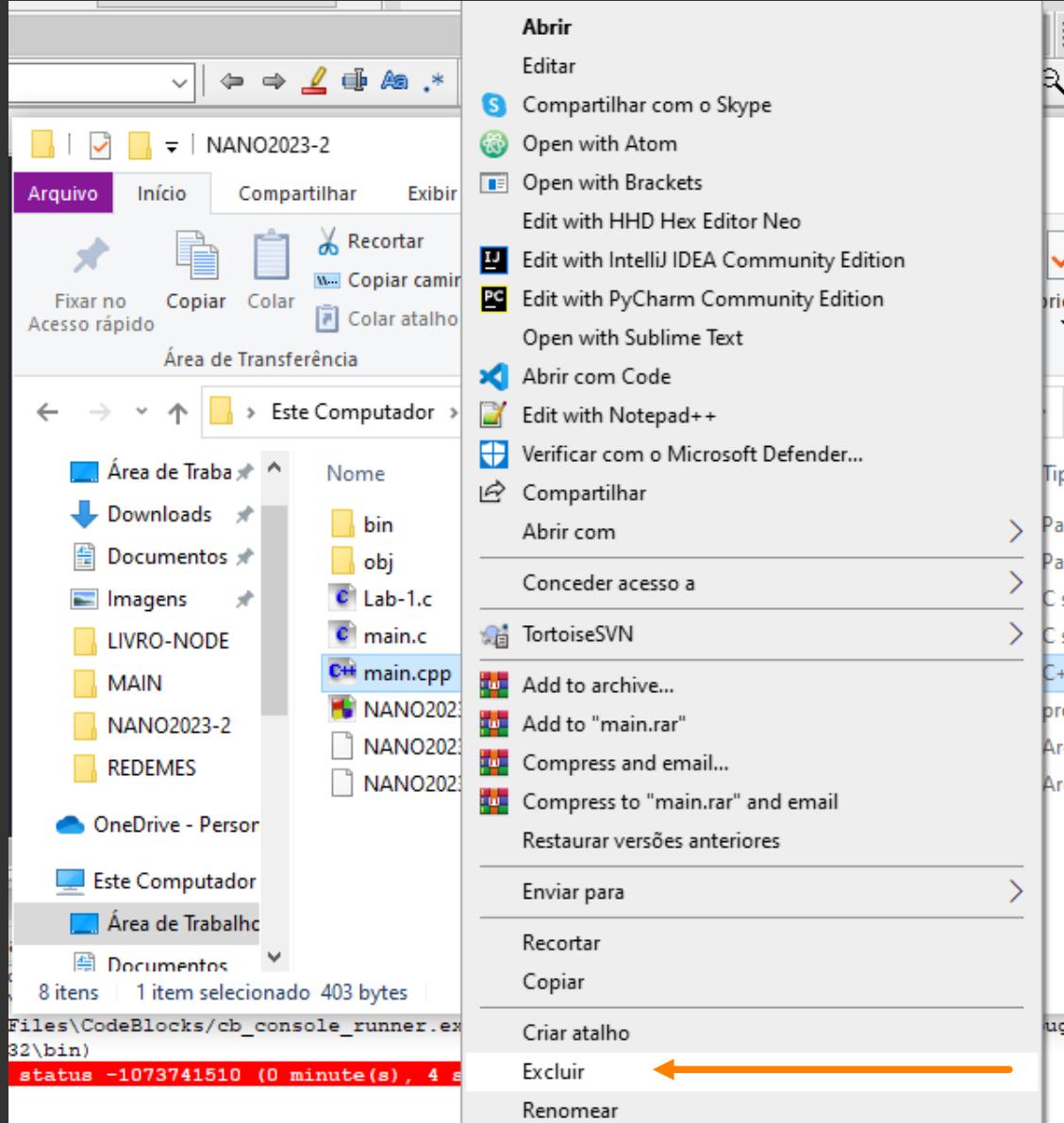
- Debug
- Release

[Wildcard select](#)
[Toggle selection](#)
[Select All](#)
[Deselect All](#)

Selected: 2

OK

Cancel



*Lab-1.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management X

Projects Files FSymbols

Workspace NANO2023-2 Sources Lab-1.c main.c

main.c *Lab-1.c X

```
8     }
9     void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA
10    {
11        glClear(GL_COLOR_BUFFER_BIT); // LIMPANDO O BUFFER DA COR DE BACKGROUND
12
13        /** DESENHANDO UM RETANGULO **/
14        //glPointSize(5); // TAMANHO DO RETANGULO
15        glBegin(GL_POLYGON); // INICIA O RETANGULO
16        glVertex2d(-0.5, 0.5); // CADA PAR DE VALOR REPRESENTA AS COORDENADAS
17        glVertex2d( 0.5, 0.5);
18        glVertex2d( 0.5,-0.5);
19        glVertex2d(-0.5,-0.5);
20        glEnd(); // TERMINA O RETANGULO
21        glFlush(); // ATUALIZA OS PIXELS
22    }
23    int main(int argc, char *argv[])
24    {
25        glutInit(&argc, argv);
```

Copie o programa anterior e cole

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X Doxygen X

```
\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Users\brito\AppData\Local\atom\bin
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023-2\bin\Debug\NANO2023-2.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-mingw32\bin)
Process terminated with status -1073741510 (0 minute(s), 4 second(s))
```

C:\Users\brito\Desktop\C\NANO2023-2\Lab-1.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 39, Col 1, Pos 1687 Insert Modified Read/Write default URGENTE 11:58

```
main.c *Lab-1.c
20         glEnd();                                     // TERMINA O RETANGULO
21         glFlush();                                    // ATUALIZA OS PIXELS
22     }
23     int main(int C, char *V[])
24     {
25         glutInit(&C, V);
26         glutInitWindowPosition(250, 50);                // POSIÇÃO DA JANELA
27         glutInitWindowSize(600, 600);                   // TAMANHO E LARGURA DA JANELA
28         glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE); // CORES DA JANELA A COR PADRÃO É BRANCA
29         // INICIALIZANDO
30         glutCreateWindow("BASIC WINDOW");             // TITULO DA JANELA
31         MyInit();                                     // CRIANDO NOSSA FUNÇÃO
32         glutDisplayFunc(Draw);
33         glutMainLoop();                                // FUNÇÃO OBRIGATORIA
34     return 0;
35 }
36
37 // https://www.youtube.com/watch?v=WC_VaSr6z6c&t=1254s
```

A primeira modificação é no buffer

```
21     glutMainLoop(); // ALIMENTA OS TIPOS
22 }
23 int main(int C, char *V[])
24 {
25     glutInit(&C, V);
26     glutInitWindowPosition(250, 50); // POSIÇÃO DA JANELA
27     glutInitWindowSize(600, 600); // TAMANHO E LARGURA DA
28     glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE); // CORES DA JANELA A COF
29 // INICIALIZANDO
30     glutCreateWindow("BASIC WINDOW OPENGL"); // TITULO DA JANELA
31     MyInit(); // CRIANDO NOSSA FUNÇÃO
32     glutDisplayFunc(Draw);
33     glutMainLoop(); // FUNÇÃO OBRIGATORIA
34     return 0;
35 }
36
37
```

```
25 glutInit(&C, V);
26 glutInitWindowPosition(250, 50); // POSIÇÃO DA JANELA
27 glutInitWindowSize(600, 600); // TAMANHO E LARGURA DA JANELA
28 glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_D ); // CORES DA JANELA A COR PADRÃO É BRANCA
29 // INICIALIZANDO
30 glutCreateWindow("BASIC WINDOW OPENGL");
31 MyInit();
32 glutDisplayFunc(Draw);
33 glutMainLoop();
34 return 0;
35 }
36 // https://www.youtube.com/watch?v=Uc_VaSpn6as
```

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck
C:\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin
et\tools;C:\Users\brito\AppData\Local\atom\bin
ing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\I
\x86_64-w64-mingw32\bin)
s terminated with status -1073741510 (0 minute(s), 4 second(s))

```
// POSIÇÃO DA JANELA
// TAMANHO E LARGURA DA JANELA
BLE | GLUT_DEPTH); // CORES DA JANELA
#define GLUT_DEPTH 0x0010
// TÍTULO DA JANELA
// CRIANDO NOSSA FUNÇÃO
```

Não encontrou uma resposta? Ask in Stack Overflow in English. [X](#)

Duas dessas constantes, `GLUT_DOUBLE` e `GLUT_DEPTH`, cada uma contém um bit definido:

4

```
GLUT_DOUBLE = 0x0002 = 0b0000 0000 0000 0010  
GLUT_DEPTH  = 0x0010 = 0b0000 0000 0001 0000
```



A combinação dessas constantes com um OR bit a bit cria um novo valor com ambos os bits definidos e é possível verificar se esses bits estão definidos no valor resultante usando, por exemplo

```
if ((display_mode & GLUT_DOUBLE) != 0) { ... }
```

`GLUT_RGB` é zero. Inclusive não tem efeito no resultado; Suspeito que represente uma configuração verdadeira por padrão.

Os números específicos usados não têm nenhum significado mais

problematica
matéria?

Tábuas lascadas
meias

いいじゃない

Primeira história
década de 19
bomba atômica

Por que todos
aumentar o te
argumentam
ignorá-lo?

Como forçar
exatamente a

Não liste o di
subdiretórios

Como se pro

Como a comp

Lat/Long extr
UTM

*Lab-1.c [NANO2023-2] - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main(int C, char* V[]) : int

Management Projects Files NANO2023-2 Sources Lab-1.c main.c

```
20     glEnd();                                // TERMINA O RETANGULO
21     glFlush();                               // ATUALIZA OS PIXELS
22 }
23 int main(int C, char *V[])
24 {
25     glutInit(&C, V);
26     glutInitWindowPosition(250, 50);           // POSIÇÃO DA JANELA
27     glutInitWindowSize(600, 600);              // TAMANHO E LARGURA DA JANELA
28     glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH); // A COMBINAÇÃO DESSAS DUAS CONSTANTES INICIALIZANDO
29
30     glutCreateWindow("LAB Program - 3 : CUBE IN SPIN"); // TÍTULO DA JANELA
31     MyInit();                                 // CRIANDO NOSSA FUNÇÃO
32     glutDisplayFunc(Draw);
33     glutMainLoop();                          // FUNÇÃO OBRIGATÓRIA
34     return 0;
35 }
36
37 // https://www.youtube.com/watch?v=wG_VaSr6z6ct-1254s
```

Logs & others

Code::Blocks X Search results X Ccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debugger X DoxyBlocks X

\Users\brito\AppData\Local\Programs\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Users\brito\AppData\Local\atom\bin

Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\NANO2023-2\bin\Debug\NANO2023-2.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\bin)

Process terminated with status -1073741510 (0 minute(s), 4 second(s))

C:\Users\brito\Desktop\C\NANO2023-2\Lab-1.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 30, Col 63, Pos 1424 Insert Modified Read/Write default USD/BRL +1,2... 12:11

```
    }
    void Draw() // FUNÇÃO CRIADA P
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH); // LIMPA
    /** DESENHANDO UM RETANGULO
    //glPointSize(5);
    glBegin(GL_POLYGON);
    glVertex2d(-0.5, 0.5);
    glVertex2d( 0.5, 0.5);
    glVertex2d( 0.5,-0.5);
    glVertex2d(-0.5,-0.5);
    glEnd();
}
```

GL_DEPTH
GL_DEPTH_BIAS
GL_DEPTH_BITS
GL_DEPTH_BUFFER_BIT ←
GL_DEPTH_CLEAR_VALUE
GL_DEPTH_COMPONENT
GL_DEPTH_FUNC
GL_DEPTH_RANGE
GL_DEPTH_SCALE
GL_DEPTH_TEST
GL_DEPTH_WRITEMASK

others

Code::Blocks X Search results X Cccc X Build log X Build
s\brito\AppData\Local\Programs\Python\Python310\Scripts;C:\Us
o\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\
ting: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:
4-mingw32\bin"

```
6     glColor4f(0,0,0,1), // FUNÇÃO PARA AS CORES DA JANELA
7     glColor3f(1,0,0); // RECEBE VALORES FLOAT PARA DAR COR A
8 }
9 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA
10 {
11     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPPANDO O BUFFER DA COR
12
13     /** DESENHANDO UM RETANGULO **/
14     //glPointSize(5); // TAMANHO DO RETANGULO
15     glBegin(GL_POLYGON); // INICIA O RETANGULO
16     glVertex2d(-0.5, 0.5); // CADA PAR DE VALOR REPRESENTA AS COO
17     glVertex2d( 0.5, 0.5);
18     glVertex2d( 0.5,-0.5);
19     glVertex2d(-0.5,-0.5);
20     glEnd(); // TERMINA O RETANGULO
```

```
    glVertex2d(-0.5,-0.5);
    glEnd();
    glutSwap(); // TERA
}
in glutStrokeCharacter(): void
{
    glutStrokeLength(): int
    glutStrokeLengthf(): GLfloat
    glutStrokeWidth(): int
    glutStrokeWidthf(): GLfloat
    glutSwapBuffers(): void // ATUA
    glutTabletButtonFunc(): void
    glutTabletMotionFunc(): void
    glutTimerFunc(): void
    glutUseLayer(): void
    glutVideoPan(): void
    glutVideoResize(): void
```

Blocks X

so\AppDat

Data\Loca

"C:\Prog

gw32\bin

minated

ab-1.c

e 21, Col 1

```
11     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
12
13     /** DESENHANDO UM RETANGULO ***/
14     //glPointSize(5);
15     glBegin(GL_POLYGON);
16     glVertex2d(-0.5, 0.5);
17     glVertex2d( 0.5, 0.5);
18     glVertex2d( 0.5,-0.5);
19     glVertex2d(-0.5,-0.5);
20     glEnd();
21     glutSwapBuffers();
22 }
23 int main(int C, char *V[])
24 {
25     glutInit(&C,V);
26     glutInitWindowPosition(250,50);
27     glutInitWindowSize(600,600);
```

```
13     /** DESENHANDO UM RETANGULO **/
14     //glPointSize(5);
15     glBegin(GL_POLYGON);
16     glVertex3d(-0.5, 0.5);
17     (■) glVertex3d(): void
18     (■) glVertex3dv(): void
19     (■) glVertex3f(): void
20     (■) glVertex3fv(): void
21     (■) glVertex3i(): void
22     } in (■) glVertex3iv(): void
23     (■) glVertex3s(): void
24     (■) glVertex3sv(): void
25
26     250, 50);
27     glutInitWindowSize(600, 600);
```

```
0  {
1      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
2
3      /** DESENHANDO UM RETANGULO **/
4      //glPointSize(5);                                // TAMANHO
5      glBegin(GL_POLYGON);                          // INICIA
6      glVertex3f(-0.5, 0.5);                      // CADA P
7      glVertex3f( 0.5, 0.5);
8      glVertex3f( 0.5,-0.5);
9      glVertex3f(-0.5,-0.5);
0      glEnd();                                    // TERMINA
1      glutSwapBuffers();                           // ATUALIZA
2
3  int main(int C, char *V[])
4  {
5      glutInit(&C,V);
6      glutInitWindowPosition(250,50);              // POSIÇÃO
7      glutInitWindowSize(600,600);                 // TAMANHO
```

```
9 void Draw()
10 {
11     GLfl
12     (■) glFeedbackBuffer(): void    | GL_DEP
13     (■) glFinish(): void
14     ■■ GLfloat
15     (■) glFlush(): void
16     (■) glFogf(): void
17     (■) glFogfv(): void
18     (■) glFogi(): void
19     (■) glFogiv(): void
20     (■) glFrontFace(): void
21     (■) glFrustum(): void
22
23 }
24
25
```

```
main.c x *Lab-1.c x
8 }
9 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA F
10 {
11     GLfloat Vertices[8][3] = {
12         {
13             {
14                 {
15                     }
16         glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPPANDO O BUFFER DA COR DE BACKGROU
17
18     /** DESENHANDO UM RETANGULO **/
19     //glPointSize(5); // TAMANHO DO RETANGULO
20     glBegin(GL_POLYGON); // INICIA O RETANGULO
21     glVertex3f(-0.5, 0.5); // CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO
22     glVertex3f( 0.5, 0.5);
23     glVertex3f( 0.5,-0.5);
24     glVertex3f(-0.5,-0.5);
25     glEnd(); // TERMINA O RETANGULO
```

```
*Lab-1.c x
8    }
9    void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDI
0    {
1        GLfloat Vertices[8][3] = {
2            {-0.5, 0.5, 0.5}, // PONTOS DOS RETANGULOS
3            { 0.5, 0.5, 0.5},
4            { 0.5,-0.5, 0.5},
5            {-0.5,-0.5, 0.5},
6        };
7        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPPANDO O BUFFER DA C
8
9        /** DESENHANDO UM RETANGULO **/
0        //glPointSize(5); // TAMANHO DO RETANGULO
1        glBegin(GL_POLYGON); // INICIA O RETANGULO
2        glVertex3f(-0.5, 0.5); // CADA PAR DE VALOR REPRESENTA AS COORDENADAS
3        glVertex3f( 0.5, 0.5);
4        glVertex3f( 0.5,-0.5);
5        glVertex3f(-0.5,-0.5).
```

AS COORDENADAS SE REFEREM AO PLANO CARTESIANO

```
*Lab-1.c x
8    }
9    void Draw() // FUNÇÃO CRIADA PARA ATENDER O
10   {
11     GLfloat Vertices[8][3] = {
12         {-0.5, 0.5, 0.5}, // PONTOS DOS RETANGULOS
13         { 0.5, 0.5, 0.5},
14         { 0.5,-0.5, 0.5},
15         {-0.5,-0.5, 0.5},
16
17         {-0.5, 0.5,-0.5}, // PONTOS DOS RETANGULOS
18         { 0.5, 0.5,-0.5},
19         { 0.5,-0.5,-0.5},
20         {-0.5,-0.5,-0.5},
21     };
22     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPANDO O BUFFER
23
24     /** DESENHANDO UM RETANGULO **/
25     //glPointSize(5); // TAMANHO DO RETANGULO
```

```
void Draw() // FUNÇÃO CRIADA PELA MÍDIA
{
    GLfloat Vertices[8][3] = {
        {-0.5, 0.5, 0.5}, /* Top Left */
        { 0.5, 0.5, 0.5}, /* Top Right */
        { 0.5,-0.5, 0.5}, /* Bottom Right */
        {-0.5,-0.5, 0.5}, /* Bottom Left */
        {-0.5, 0.5,-0.5}, /* Top Left */
        { 0.5, 0.5,-0.5}, /* Top Right */
        { 0.5,-0.5,-0.5}, /* Bottom Right */
        {-0.5,-0.5,-0.5} /* Bottom Left */
    };
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // Limpa o buffer de cor e profundidade

    /** DESENHANDO UM RETÂNGULO **/
    //glPointSize(5); // Tamanho do ponto
```

```
*Lab-1.c X
8 }
9 void Draw() // FUNÇÃO DE DIBUJO
10 {
11     GLfloat V[8][3] = {
12         {-0.5, 0.5, 0.5},
13         { 0.5, 0.5, 0.5},
14         { 0.5,-0.5, 0.5},
15         {-0.5,-0.5, 0.5},
16         {-0.5, 0.5,-0.5},
17         { 0.5, 0.5,-0.5},
18         { 0.5,-0.5,-0.5},
19         {-0.5,-0.5,-0.5}
20     };
21     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
```

TROQUE O NOME PRA V

```
    };  
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIM  
    Cube(V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]); // ←  
    /** DESENHANDO UM RETANGULO **/  
    //glPointSize(5); // TAMANHO DO RETANGULO  
    glBegin(GL_POLYGON); // INICIA O RETANGULO
```

```
7     glColor3f(1, 0, 0);  
8 }  
9  
10 void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[], GLfloat V5[], GLfloat V6[]);  
11 void Draw();  
12 {  
13     GLfloat V[8][3] = {
```

```
9     , GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[], GLfloat V5[], GLfloat V6[], GLfloat V7[]);  
10 }  
11  
12 }
```

```
6     glClearColor(0,0,0,1);           // FUNÇÃO PARA AS CORES DA JANELA
7     glColor3f(1,0,0);              // RECEBE VALORES FLOAT PARA DAR COR A JANELA
8 }
9
10 void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat V4[],GLfloat V5[],GLfloat V6
11 {
12     |
13 }
14
15 void Draw()                      // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA F
16 {
17     GLfloat V[8][3] = {           // PONTOS DOS RETÂNGULOS COORDENADAS
18         -0.5, 0.5, 0.5,
```

```
8 }  
9  
0 void Cube(GLfloat v0[], GLfloat v1[], GLfloat v2[], GLfloat v3[],  
1 {  
2     //  
3 }  
4  
5 }
```

```
3 }
4
5     void Cube(GLfloat v0[], GLfloat v1[], GLfloat v2[], GLfloat v3[], GLfloat v4[], GLfloat v5[], GLfloat v6[])
6     {
7         Face();
8     }
9
10    void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA F
11    {
12        GLfloat v[8][3] = {
```

```
main.c *Lab-1.c
6     glClearColor(0,0,0,1);           // FUNÇÃO PARAS AS CORES I
7     glColor3f(1,0,0);              // RECEBE VALORES FLOAT PA
8 }
9
10 void Face()
11 {
12 }
13
14
15 void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[],
16 {
17     Face();
18 }
19
20 void Draw();                         // FUNÇÃO CRIADA PARA ATEN
21 {
22     GLfloat V[8][3] = {             // PONTOS DOS F
23         {-0.5, 0.5, 0.5},
```

```
33     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);      // LIMPANDO O BUFFER DA COR DE FONTE
34
35     Cube(V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]);
36     /** DESENHANDO UM RETANGULO **/
37     //glPointSize(5);                                     // TAMANHO DO RETANGULO
38     glBegin(GL_POLYGON);                                // INICIA O RETANGULO
39     glVertex3f(-0.5, 0.5);                            // CADA PAR DE VALOR REPRESENTA AS COORDENADAS
40     glVertex3f( 0.5, 0.5);
41     glVertex3f( 0.5,-0.5);
42     glVertex3f(-0.5,-0.5);
43     glEnd();                                         // TERMINA O RETANGULO
44     glutSwapBuffers();                                // ATUALIZA OS PIXELS
45 }
46 int main(int c, char *v[])
47 {
```

CUT ESSA PARTE

```
main.c *Lab-1.c X
8 }
9
10 void Face()
11 {
12     glBegin(GL_POLYGON); // INICIA O RETANGULO
13     glVertex3f(-0.5, 0.5); // CADA PAR DE VALOR REPRESENTA AS COO
14     glVertex3f( 0.5, 0.5);
15     glVertex3f( 0.5,-0.5);
16     glVertex3f(-0.5,-0.5);
17     glEnd();
18 }
19
20 void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat V4[],GLfloat V5[],GLF
21 {
22     Face();
23 }
24
25 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA
```

COLE

```
6     glClearColor(0,0,0,1);                                // FUNÇÃO PARAS AS CORES DA JANELA
7     glColor3f(1,0,0);                                    // RECEBE VALORES FLOAT PARA DAR CORES
8 }
9
10 void Face(GLfloat A[],GLfloat B[],GLfloat C[],GLfloat D[])
11 {
12     glBegin(GL_POLYGON);                                // INICIA O RETANGULO
13         glVertex3f(-0.5, 0.5);                         // CADA PAR DE VALOR REPRESENTA UMA VERTICE
14         glVertex3f( 0.5, 0.5);
15         glVertex3f( 0.5,-0.5);
16         glVertex3f(-0.5,-0.5);
17     glEnd();
18 }
19
20 void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat V4[],GLfloat V5[])
21 {
22     Face();
23 }
```

```
void Face(GLfloat A[], GLfloat B[], GLfloat C[], GLfloat D[])
{
    glBegin(GL_POLYGON);                                // INICIA O RETANGULO
        glVertex3f(A[0],A[1],A[2]);                      // CADA PAR DE VALOR REP
        glVertex3fv();
```

(■) glVertex3fv(): void

```
    glEnd();
```

```
void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[])
{
    Face();
```

```
    glColor3f(1,0,0);                                // RECEBE VALORES FLOAT
}
}

void Face(GLfloat A[],GLfloat B[],GLfloat C[],GLfloat D[])
{
    glBegin(GL_POLYGON);                            // INICIA O RETANGULO
    glVertex3f(A[0],A[1],A[2]);                    // CADA PAR DE VALOR REFERE
    glVertex3fv(B);
    glVertex3fv(C);
    glVertex3fv(D);
    glEnd();
}

void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat V4[
{
    Face();
}
```

```
0 void Face(GLfloat A[],GLfloat B[],GLfloat C[],GLfloat D[])
1 {
2     glBegin(GL_POLYGON);                                // INICIA O RETANGUL
3         glVertex3fv(A);                               // CADA PAR DE VALOR REPRESENTA
4         glVertex3fv(B);
5         glVertex3fv(C);
6         glVertex3fv(D);
7     glEnd();
8 }
9
10 void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat
11 {
12     Face(V0,V1,V2,V3);
13 }
```

```
18 }
19
20 void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[], GLfloat V5[],
21 {
22     Face(V0, V1, V2, V3);
23 } void Face(GLfloat A[], GLfloat B[], GLfloat C[], GLfloat D[])
24
25 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA
26 {
27     GLfloat V[8][3] = {
28         {-0.5, 0.5, 0.5}, // PONTOS DOS RETÂNGULOS COC
29         {0.5, 0.5, 0.5},
```

```
19
20 void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[], GLfloat V5[], GLfloat V6[]
21 {
22     Face(V0, V1, V2, V3);
23     Face(V5, V5, V6, V7); ←
24 }
25
26 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA
27 {
28     GLfloat V[8][3] = {
29         {-0.5, 0.5, 0.5}, // PONTOS DOS RETANGULOS COORDENADAS
30         { 0.5, 0.5, 0.5},
31         { 0.5, -0.5, 0.5},
32         {-0.5, -0.5, 0.5}
```



```
*Lab-1.c x
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void MyInit()
5 {
6     glClearColor(0,0,0,1);           // FUNÇÃO PARAS AS CORES DA JANELA
7     glColor3f(1,0,0);              // RECEBE VALORES FLOAT PARA DAR COR A JANELA
8 }
9
10 void Face(GLfloat A[],GLfloat B[],GLfloat C[],GLfloat D[])
11 {
12     glBegin(GL_POLYGON);          // INICIA O RETANGULO
13     glVertex3fv(A);              // CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO
14     glVertex3fv(B);
15     glVertex3fv(C);
16     glVertex3fv(D);
17     glEnd();
```

Draw() : void

The screenshot shows the Code::Blocks IDE interface. The main window displays a C++ source code file named "main.c". The code defines a function `Cube` that takes eight arrays of GLfloats as parameters and contains two calls to the `Face` function. It also defines a `Draw` function that initializes an array `V` with vertex coordinates. The code is annotated with comments in Portuguese. The IDE has a dark theme with various toolbars and panels visible at the top and bottom.

```
main.c *Lab-1.c
17     glEnd();
18 }
19
20 void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[], GLfloat V5[], GLfloat V6[])
21 {
22     Face(V0, V1, V2, V3);
23     Face(V5, V4, V6, V7);
24 }
25
26 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA
27 {
28     GLfloat V[8][3] = {
29         {-0.5, 0.5, 0.5}, // PONTOS DOS RETANGULOS COORDENADAS
30         { 0.5, 0.5, 0.5},
31         { 0.5,-0.5, 0.5},
32         {-0.5,-0.5, 0.5},
33         {-0.5, 0.5, -0.5}, // PONTOS DOS RETANGULOS COORDENADAS
34         { 0.5, 0.5, -0.5},
35         { 0.5,-0.5, -0.5},
36         {-0.5,-0.5, -0.5}
37 }
```

Logs & others

Code::Blocks Search results Ccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger Doxygen

The screenshot shows a dark-themed integrated development environment (IDE) window. At the top is a toolbar with various icons. Below the toolbar, two tabs are visible: "main.c" and "*Lab-1.c". The code editor displays the following C code:

```
32             {-0.5,-0.5, 0.5},  
33             {-0.5, 0.5,-0.5},           // PONTOS DOS RETANGULOS COORDENADAS  
34             { 0.5, 0.5,-0.5},  
35             { 0.5,-0.5,-0.5},  
36             {-0.5,-0.5,-0.5}  
37         };  
38     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);    // LIMPANDO O BUFFER DA COR DE BACKGRO  
39  
40     Cube (V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]);  
41     /** DESENHA float Draw::V[] */  
42     //glPointSize(5);                                // TAMANHO DO RETANGULO  
43     //termina o retangulo  
44     glutSwapBuffers();                            // ATUALIZA OS PIXELS  
45  
46 }  
47 int main(int C, char *V[])  
48 {  
49     glutInit (&C, V);
```

The code uses OpenGL functions like `glClear`, `Cube`, and `glutSwapBuffers`. It also includes comments in Portuguese explaining the purpose of certain lines. A yellow vertical bar on the left margin highlights line 42.

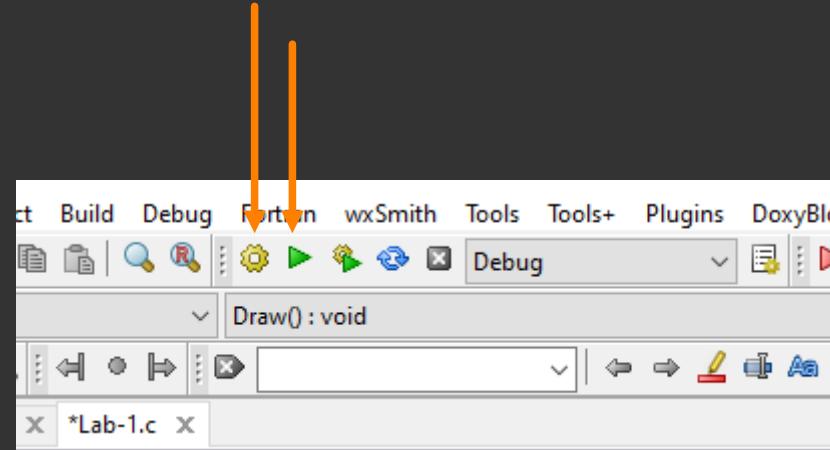
main.c X *Lab-1.c X

```
43     //glPointSize(5);                                // TAMANHO DO RETANGULO
44                                         // TERMINA O RETANGULO
45     glutSwapBuffers();                            // ATUALIZA OS PIXELS
46 }
47 int main(int C, char *V[])
48 {
49     glutInit(&C,V);
50     glutInitWindowPosition(250,50);                // POSIÇÃO DA JANELA
51     glutInitWindowSize(600,600);                   // TAMANHO E LARGURA DA JANELA
52     glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH); // A COMBINAÇÃO DESSAS DUAS CONSTANTES
53 // INICIALIZANDO
54     glutCreateWindow("LAB Program - 3 : CUBE IN SPIN"); // TITULO DA JANELA
55     MyInit();                                     // CRIANDO NOSSA FUNÇÃO
56     glutDisplayFunc(Draw);
57     glutMainLoop();                             // FUNÇÃO OBRIGATÓRIA
58     return 0;
59 }
60 }
```

main.c x

*Lab-1.c x

```
48 {  
49     glutInit(&C, V);  
50     glutInitWindowPosition(250, 50);           // POSIÇÃO DA JANELA  
51     glutInitWindowSize(600, 600);             // TAMANHO E LARGURA DA JANELA  
52     glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH); // A COMBINAÇÃO DESSAS DUAS CONSTANTES  
53     // INICIALIZANDO  
54     glutCreateWindow("LAB Program - 3 : CUBE IN SPIN");      // TÍTULO DA JANELA  
55     MyInit();                                         // CRIANDO NOSSA FUNÇÃO  
56     glutDisplayFunc(Draw);  
57     glutMainLoop();                                // FUNÇÃO OBRIGATÓRIA  
58     return 0;  
59 }  
60  
61 // https://www.youtube.com/watch?v=wG\_VaSr6a6c&t=1254s  
62 // parada 20:55  
63  
64
```



Compile e depois rode

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins Doxygen Settings Help

<global>

Management

Projects Files FSymbols

Workspace LAB-1 Sources Lab-1.c

Lab-1.c X

```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void MyInit()
5 {
6     glClearColor(0,0,0,1);                                // FUNÇÃO PARA AS CORES DA JANELA
7     glColor3f(1,0,0);                                    // RECEBE VALORES FLOAT PARA DAR COR A JANELA
8 }
9
10 void Face(GLfloat A[],GLfloat B[],GLfloat C[],GLfloat D[])
11 {
12     glBegin(GL_POLYGON);                                // INICIA O RETANGULO
13     glVertex3fv(A);                                    // CADA PAR DE VALOR REPRESENTA AS COORDENADAS NO PLANO CARTESIANO
14     glVertex3fv(B);
15     glVertex3fv(C);
16     glVertex3fv(D);
17     glEnd();
18 }
19
20 void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat V4[],GLfloat V5[],GLfloat V6[],GLfloat V7[])
21 {
22     Face(V0,V1,V2,V3);
23     Face(V1,V2,V3,V4);
24     Face(V2,V3,V4,V5);
25     Face(V3,V4,V5,V6);
26     Face(V4,V5,V6,V7);
27     Face(V5,V6,V7,V0);
28 }
```

Logs & others

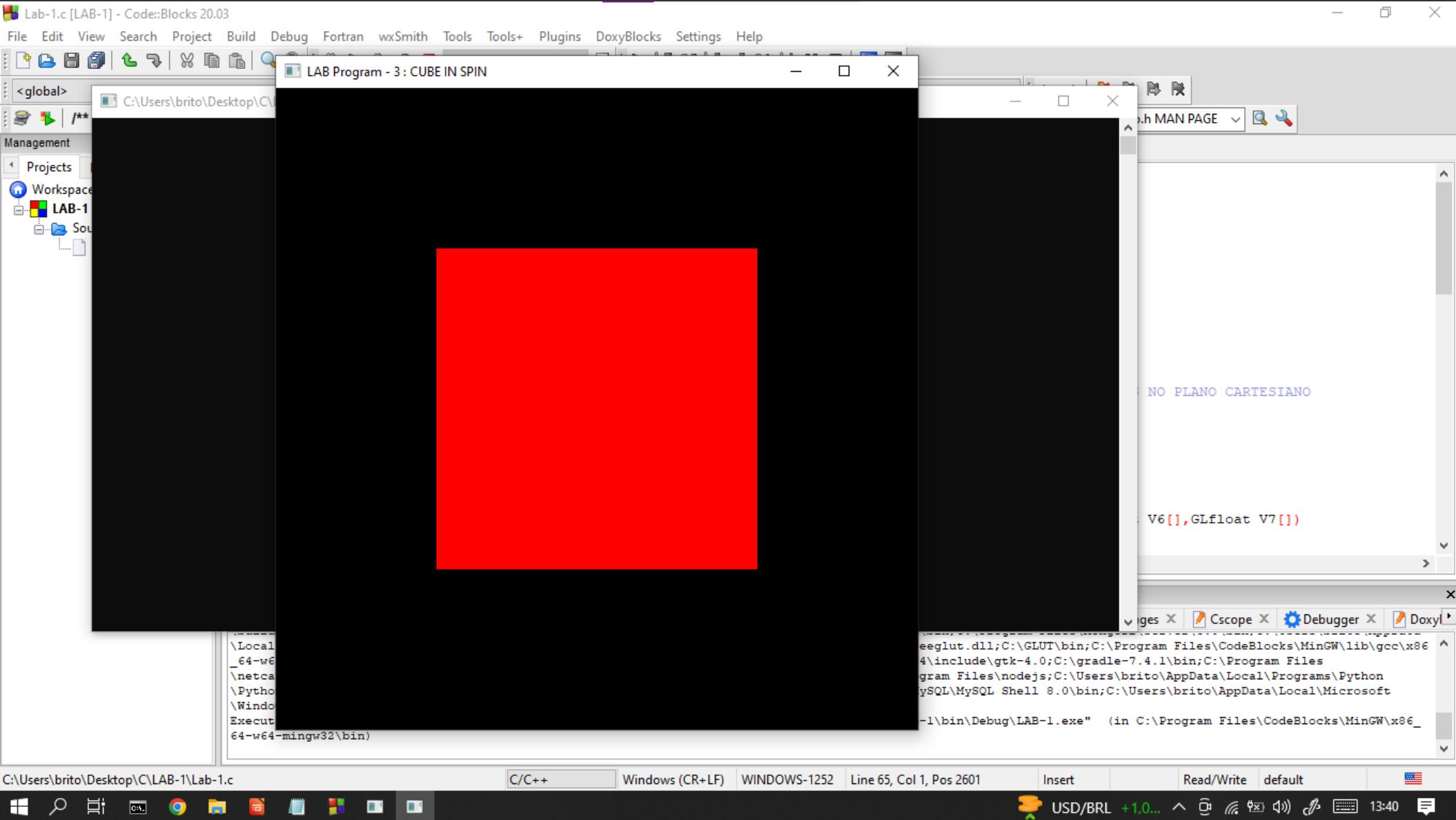
Code::Blocks Search results Ccc C Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger Doxygen

```
\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\WindowsApps;C:\Users\brito\.dotnet\tools;C:\Users\brito\AppData\Local\atom\bin
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\LAB-1\bin\Debug\LAB-1.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64-w64-mingw32\bin)
Process terminated with status -1073741510 (0 minute(s), 9 second(s))
```

Para rodar

- iniciei um novo projeto
- rodei primeiro em c++
- Deletei a pasta do c++
- criei o projeto Lab-1.c
 - buil (compilei)
 - run (rodei)

Assim deu certo



```
55     MyInit();                                // CRIANDO NOSSA FUNÇÃO
56     glutDisplayFunc (Draw);
57     glutId
58     (void) glutGameModeString(); void
59     (void) glutGet(); int
60     (void) glutGetColor(); GLfloat
61     (void) glutGetMenu(); int
62     (void) glutGetModifiers(); int
63     (void) glutGetWindow(); int
64     (void) glutHideOverlay(); void
65     (void) glutHideWindow(); void
66     (void) glutIconifyWindow(); void
67     (void) glutIdleFunc(); void
68     (void) glutIgnoreKeyRepeat(); void
69     #include <GL/glut.h>
70
71     glutInit(&argc, &argv);
```

Logs & others

Code::Blocks

```
\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Windows Apps;C:\Users\brito\.dotnet\tools;C:\Users\brito\AppData\Local\atom\bin
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\LAB-1\bin\Debug\LAB-1
```

```
54     glutCreateWindow("LAB Program - 3 : CUBE IN SPIN");           // TITULO D
55     MyInit();                                         // CRIANDO NOSSA FUNÇÃO
56     glutDisplayFunc(Draw);
57     glutIdleFunc();
58     glutMainLoop(void glutIdleFunc(void(* callback)(void))) OBRIGATORIA
59     return 0;
60 }
61
62 // https://www.youtube.com/watch?v=wG\_VaSr6a6c&t=1254s
63 // https://www.youtube.com/watch?v=wG_VaSr6a6c&t=1254s
```

```
47 int main(int C, char *V[])
48 {
49     glutInit(&C,V);
50     glutInitWindowPosition(250,50);           // POSIÇÃO DA JANELA
51     glutInitWindowSize(600,600);             // TAMANHO E LARGURA DA JANELA
52     glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_DEPTH); // A COMBINAÇÃO DES
53     // INICIALIZANDO
54     glutCreateWindow("LAB Program - 3 : CUBE IN SPIN");      // TITULO DA JANELA
55     MyInit();                                         // CRIANDO NOSSA FUNÇÃO
56     glutDisplayFunc(Draw);
57     glutIdleFunc(Spin); ←
58     glutMainLoop();                                // FUNÇÃO OBRIGATORIA
59     return 0;
60 }
61
62 // https://www.youtube.com/watch?v=wG_VaSr6a6c&t=1254s
63 // parada 20:55
64
65
66
67
```

The screenshot shows a C IDE interface with a code editor window titled '*Lab-1.c'. The code is a simple OpenGL application using glut.h. It includes headers, defines two functions (Spin and MyInit), and implements a Face function that draws a rectangle.

```
1 #include <GL/glu.h>
2 #include <GL/glut.h>
3
4 void Spin()
5 {
6
7 }
8
9 void MyInit()
10 {
11     glClearColor(0,0,0,1);           // FUNÇÃO PARA AS CORES DA JANELA
12     glColor3f(1,0,0);              // RECEBE VALORES FLOAT PARA DAR CO
13 }
14
15 void Face(GLfloat A[],GLfloat B[],GLfloat C[],GLfloat D[])
16 {
17     glBegin(GL_POLYGON);          // INICIA O RETANGULO
18     glVertex3fv(A);              // CADA PAR DE VALOR REPRESENTA AS
19     glVertex3fv(B);
20     glVertex3fv(C);
21     glVertex3fv(D);
22 }
```



```
41                         { 0.5,-0.5,-0.5},  
42                         {-0.5,-0.5,-0.5}  
43                     };  
44         glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPANDO O BUFFER DA COR DE BACKGROUND  
45  
46         glRotatef(0)  
47         Cube(V[0] void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)  
48         /** DESENHANDO UM RETANGULO **/  
49         //glPointSize(5); // TAMANHO DO RETANGULO  
50         //TERMINA O RETANGULO  
51         //ATUALIZA OS DADOS
```

& others

```
38
39                     {-0.5, 0.5,-0.5},           // PONTOS DOS RETANGULOS COORDENADAS
40                     { 0.5, 0.5,-0.5},
41                     { 0.5,-0.5,-0.5},
42                     {-0.5,-0.5,-0.5}
43             };
44     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);    // LIMPANDO O BUFFER DA COR DE BACKGROUND
45
46     glRotatef(90,0,1,0);
47     Cube(V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]);
48     /** DESENHANDO UM RETANGULO **/
49     //glPointSize(5);                                // TAMAÑO DO RETANGULO
50     //termina o retangulo                           // TERMINA O RETANGULO
51     //atualiza os buffers
```

gs & others

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X Cscope X Debug

netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Mi

```
--          ( 0.0, 0.0, 0.0 ),
42             {-0.5,-0.5,-0.5}
43         };
44     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPANDO O BUFFER DA COR DE BACKGROUND
45
46     glRotatef(90,0,1,0); // ANGULOS DE ROTAÇÃO ←
47     Cube (V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]);
48     /** DESENHANDO UM RETANGULO **/
49     //glPointSize(5); // Tamanho do Retangulo
50     //glBegin(GL_QUADS); // Termina o Retangulo
51     //glEnd(); // Atualiza os pixels
```

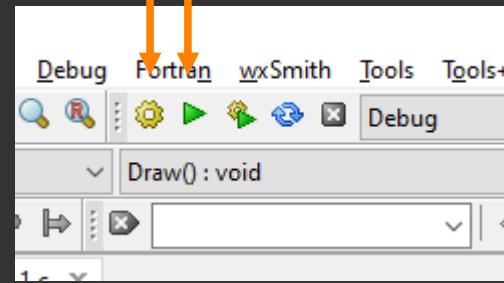
x others

```
39             {-0.5, 0.5,-0.5}, // PONTOS DOS RETANGULOS COORDENADAS
40             { 0.5, 0.5,-0.5},
41             { 0.5,-0.5,-0.5},
42             {-0.5,-0.5,-0.5}
43         };
44         glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPPANDO O BUFFER DA COR DE
45         GL1c
46         (●) glLighti(): void // ANGULOS DE ROTAÇÃO
47         (●) glLightiv(): void
48         (●) glLightModelf(): void
49         (●) glLightModelfv(): void
50         (●) glLightModeli(): void
51         (●) glLightModeliv(): void
52     }
53     in(●) glLineStipple(): void
54     (●) glLineWidth(): void
55     (●) glListBase(): void
56     (●) glLoadIdentity(): void
57     (●) glLoadMatrixd(): void
58     netcat-1.12;C:(●) glLoadMatrixf(): void
59     Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL\MySQL Shell 8.0\bin;
60     WindowsApps;C:\Users\brito\.dotnet\tools;C:\Users\brito\AppData\Local\atom\bin
61     executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "C:\Users\brito\Desktop\C\LAB-1\bin\Debug\LAB-1.exe"
62     -w64-mingw32\bin)
63 process terminated with status -1073741510 (0 minute(s), 35 second(s))
```

```
41                         { 0.5,-0.5,-0.5},  
42                         {-0.5,-0.5,-0.5}  
43                     };  
44         glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPPANDO O BUFFER DA CO  
45         glLoadIdentity(); // ANGULOS DE ROTAÇÃO  
46         glRotatef(90, void glLoadIdentity(void)  
47         Cube(V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]); // DESENHANDO UM RETANGULO  
48         /** DESENHANDO UM RETANGULO **/  
49         //glPointSize(5); // TAMANHO DO RETANGULO  
50         //glutSwapBuffers(); // TERMINA O RETANGULO  
51         glutSwapBuffers(); // ATUALIZA OS PIXELS  
52     }
```

```
1          { 0.5,-0.5,-0.5},  
2          {-0.5,-0.5,-0.5}  
3      };  
4  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPPANDO O BUFFER DA COR DE BACKGROUND  
5  glLoadIdentity();  
6  glRotatef(90,0,1,0); // ANGULOS DE ROTAÇÃO  
7  Cube(V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]);  
8  /** DESENHANDO UM RETANGULO **/  
9  //glPointSize(5); // TAMANHO DO RETANGULO  
0  //termina o retangulo // TERMINA O RETANGULO  
1  glutSwapBuffers(); // ATUALIZA OS PIXELS  
2  
3  int main(int C, char *V[])  
4 ,
```

mers



File C:\Users\brito\Desktop\CLAB-1\bin\Debug\LAB-1.exe

LAB Program - 3 : CUBE IN SPIN

RDENADAS

RDENADAS

DE BACKGROUND

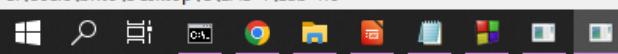
Logs & other outputs

\Local
_64-w64
\netca
\Python
\Windo
Execut
64-w64-mingw32\bin)

a++ x CppCheck/Vera++ messages x Cscope x Debugger x Doxygen x

eeglut.dll;C:\GLUT\bin;C:\Program Files\CodeBlocks\MinGW\lib\gcc\x86_64\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files\gram Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\ySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft-1\bin\Debug\LAB-1.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_64\bin)

C:\Users\brito\Desktop\CLAB-1\Lab-1.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 45, Col 22, Pos 1623 Insert Read/Write default

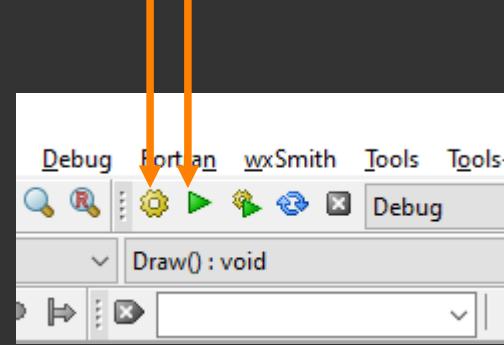


```
40                     { 0.5, 0.5,-0.5},  
41                     { 0.5,-0.5,-0.5},  
42                     {-0.5,-0.5,-0.5}  
43             );  
44     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); // LIMPANDO O BUFFER DA COR DE BACKGROUND  
45     glLoadIdentity();  
46     glRotatef(45,0,1,0); ← // ANGULOS DE ROTAÇÃO  
47     Cube (V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]);  
48     /** DESENHANDO UM RETANGULO **/  
49     //glPointSize(5); // TAMANHO DO RETANGULO  
50     //glRectf(100,100,300,200); // TERMINA O RETANGULO  
51     glutSwapBuffers(); // ATUALIZA OS PIXELS  
52 }  
53 int main(int C, char *V[]){
```

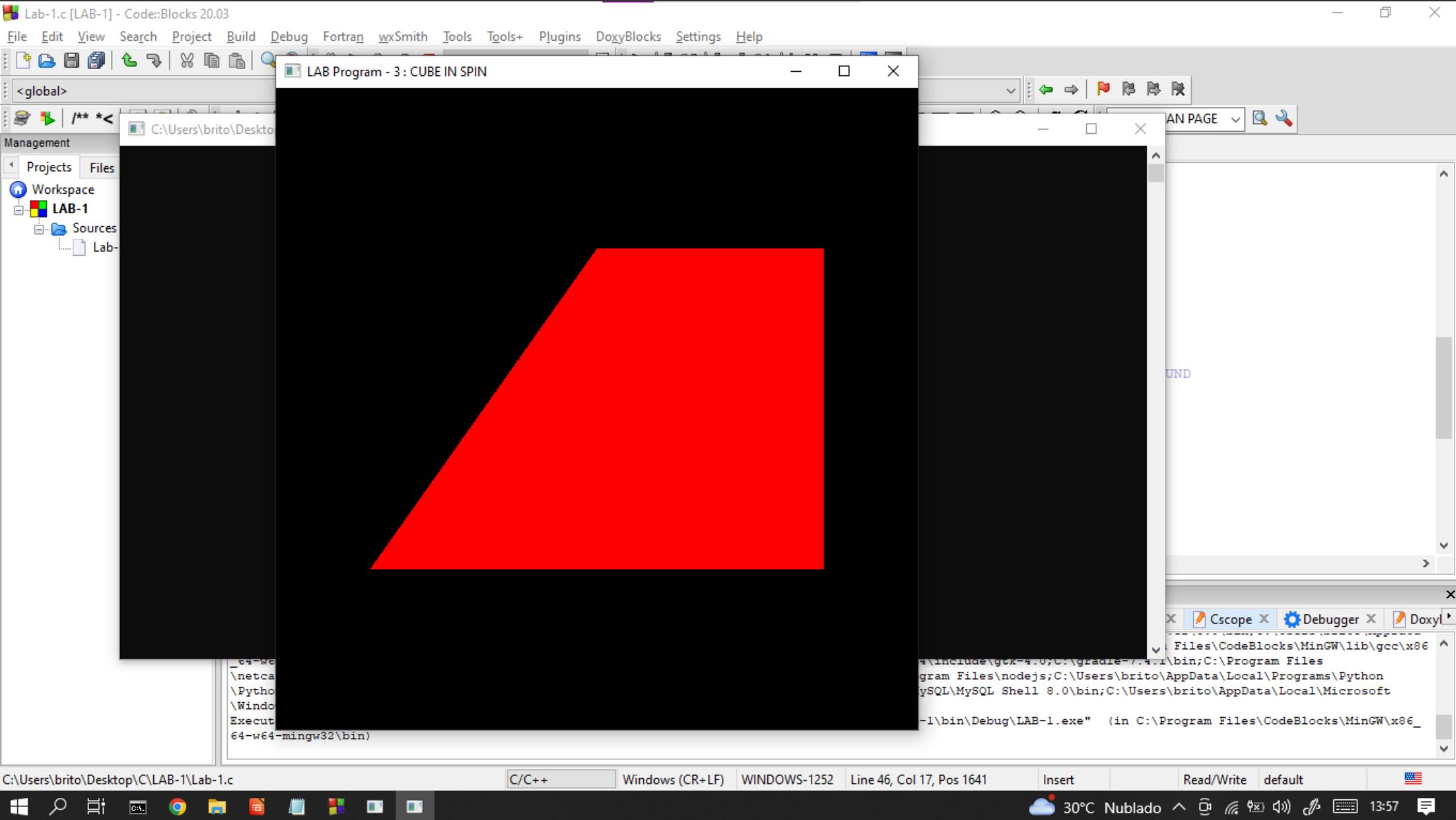
& others

Code::Blocks X Search results X Cccc X Build log X Build messages X CppCheck/Vera++ X CppCheck/Vera++ messages X C

Mude o ângulo de rotação para 45



Por praticidade vou me referir a construa e rode



```
24
25     void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[], GLfloat V5[])
26 {
27     glCo
28     (●) glClearDepth(): void
29     (●) glClearIndex(): void
30     (●) glClearStencil(): void
31     (●) glClipPlane(): void
32     void() glColor3b(): void // FUNÇÃO CRIADA PARA ATEN
33     (●) glColor3bv(): void
34     (●) glColor3d(): void
35     (●) glColor3dv(): void
36     (●) glColor3f(): void // PONTOS DOS RETANGULOS
37     (●) glColor3fv(): void
38     (●) glColor3i(): void
39     (●) glColor3iv(): void
```

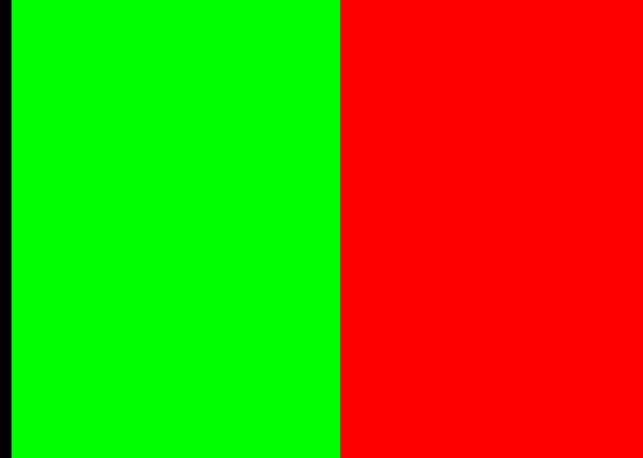
```
22     glEnd();
23 }
24
25 void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat V4[],GLfloat V5[],GLfloat V6[],GL
26 {
27     glColor3f(1,0,0);
28     Face(V0,V1,V2,V3);
29     Face(V4,V5,V6,V7);
30 }
31
32 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA C
33 {
```

```
21         glVertex3fv(D);
22     glEnd();
23 }
24
25 void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[], GLfloat V5[], GLfloat V6[], GLfloat V7[])
26 {
27     glColor3f(1,0,0);
28     Face(V0,V1,V2,V3);
29
30     glColor3f(0,1,0);
31     Face(V4,V5,V6,V7);
32 }
33
34 void Draw() // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc
```

BUILD E DEPOIS RUN

File C:\Users\brito\Desktop\CLAB-1\bin\Debug\LAB-1.exe

LAB Program - 3 : CUBE IN SPIN



stdio.h MAN PAGE

DENADAS NO PLANO CARTESIANO

GLfloat V6[], GLfloat V7[])

R O PEDIDA DA OUTRA FUNÇÃO glutDisplayFunc

DENADAS

Logs & other

C:\Program Files\CodeBlocks\MinGW\bin\Debug\LAB-1.exe

eeglut.dll;C:\GLUT\bin;C:\Program Files\CodeBlocks\MinGW\lib\gcc\x86_4\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files\gram Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\ySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\1\bin\Debug\LAB-1.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_

```
45                         {-0.5,-0.5,-0.5}
46                         );
47 glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);      // LIMPANDO O BUFFER DA COR DE BACKGROUND
48
49 glLoadIdentity();
50 glRotatef(T,0,1,0); ←—————                         // ANGULOS DE ROTAÇÃO
51
52 Cube (V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]);
53 /** DESENHANDO UM RETANGULO **/
54 //glPointSize(5);                                     // TAMAÑHO DO RETANGULO
55                                         // TERMINA O RETANGULO
56 glutSwapBuffers();                                  // ATUALIZA OS PIXELS
57 }
58 int main(int C, char *V[])
```

```
5     void Spin()
6 {
7     T = T + 1;
8     if(T > 360)
9         T = 0;
10    glutE
11 }
12
13 void
14 {
15     glutLeaveGameMode(): void
16     glutMainLoop(): void
17     glutMenuStateFunc(): void
18     glutMenuStatusFunc(): void
19     glutMotionFunc(): void
20     glutMouseFunc(): void
21     glutOverlayDisplayFunc(): void
22     glutPassiveMotionFunc(): void
23     glutPopWindow(): void
24     glutPositionWindow(): void
25     glutPostOverlayRedisplay(): void
26     glutPostRedisplay(): void
```

```
    FUNÇÃO PARAS AS CORE  
    RECEBE VALORES FLOAT
```

INICIA O RETANGULO

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck/Vera++

```
\netcat-1.12;C:\apache-maven-3.8.6\bin;C:\gbdk\bin;C:\Users\brito\AppData\Roaming\nvm;C:\Program  
Python310\Scripts;C:\Users\brito\AppData\Local\Programs\Python\Python310;C:\Program Files\MySQL
```

```
3     glutInit( &argc, &argv ); // CRIAÇÃO
4
5     void Spin()
6     {
7         T = T + 1;
8         if(T > 360)
9             T = 0;
10        glutPostRedisplay();
11    }
12
```

```
void Spin()
{
    T = T + 1;
    if(T > 360)
        T = 0;
    glutPostRedisplay();
}

void MyInit()
```

```
void glutPostRedisplay(void)
```

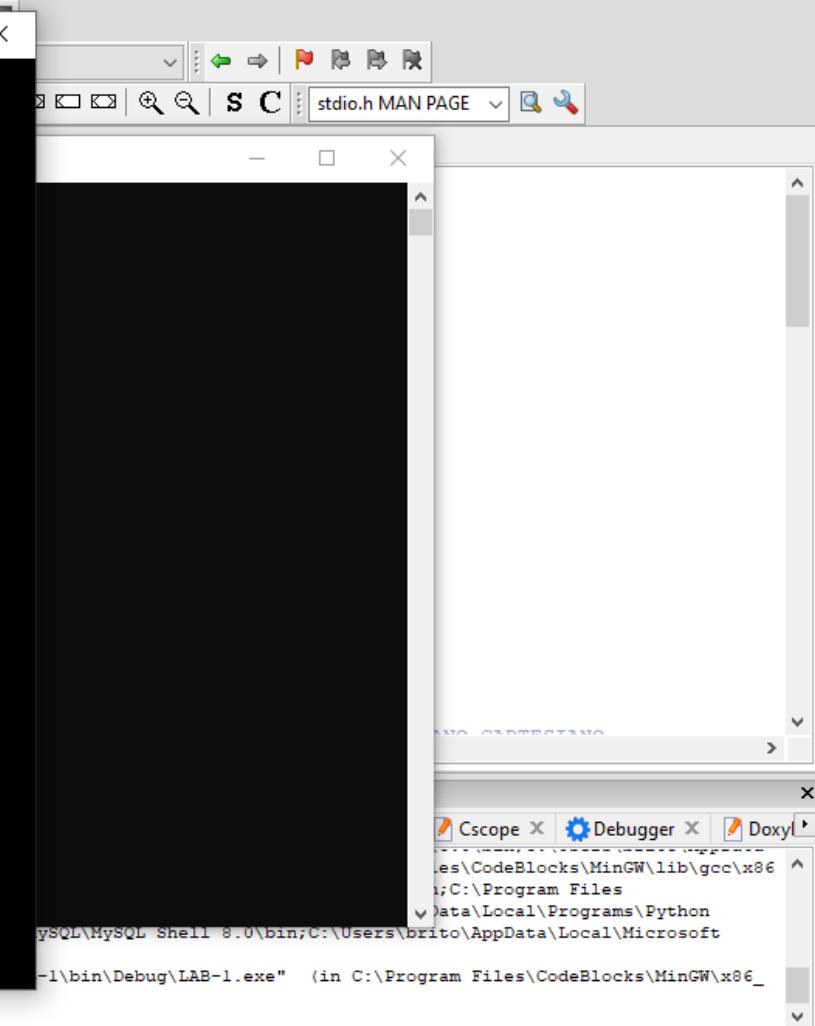
```
5     GLfloat T = 0;                                // CRIAMOS
6
7 void Spin()
8 {
9     T = T + 1;
10    if(T > 360)
11        T = 0;
12    glutPostRedisplay();
```

```
// |
```

**COMPILE DEPOIS
RUN**



O cube deve girar

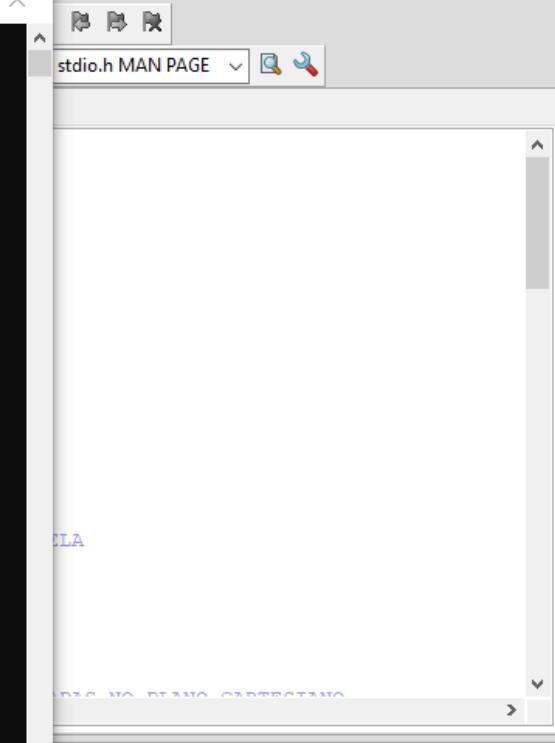
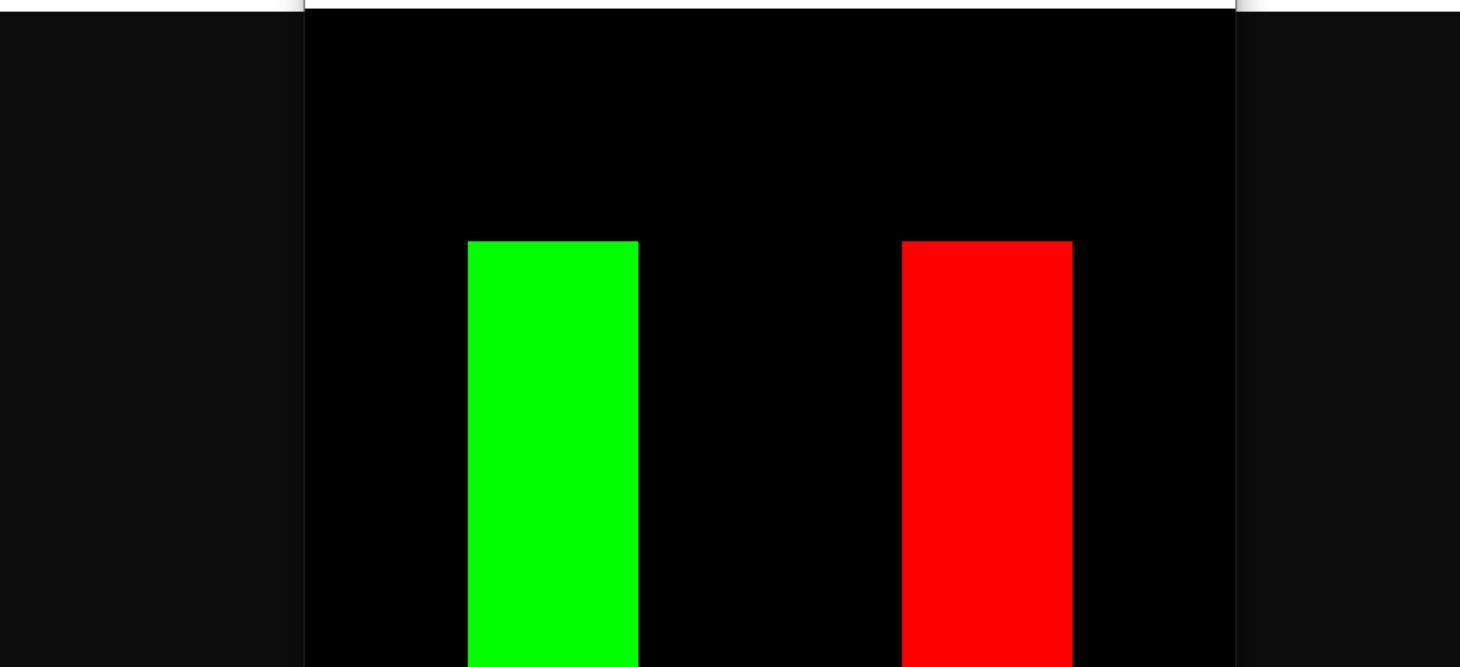


```
3     GLfloat T = 0;                                // CRIAMOS UMA VARIÁVEL GLOBAL T
4
5     void Spin()
6     {
7         T = T + 0.5;                               // VELOCIDADE DE ROTAÇÃO
8         if(T > 360)
9             T = 0;
10        glutPostRedisplay();                     //
11    }
12
13    void MyInit()
```

**ALTERE A VELOCIDADE PARA GIRAR MAIS LENTO
DEPOIS BUILD E DEPOIS RUN**

C:\Users\brito\Desktop\C\LAB-1

LAB Program - 3 : CUBE IN SPIN

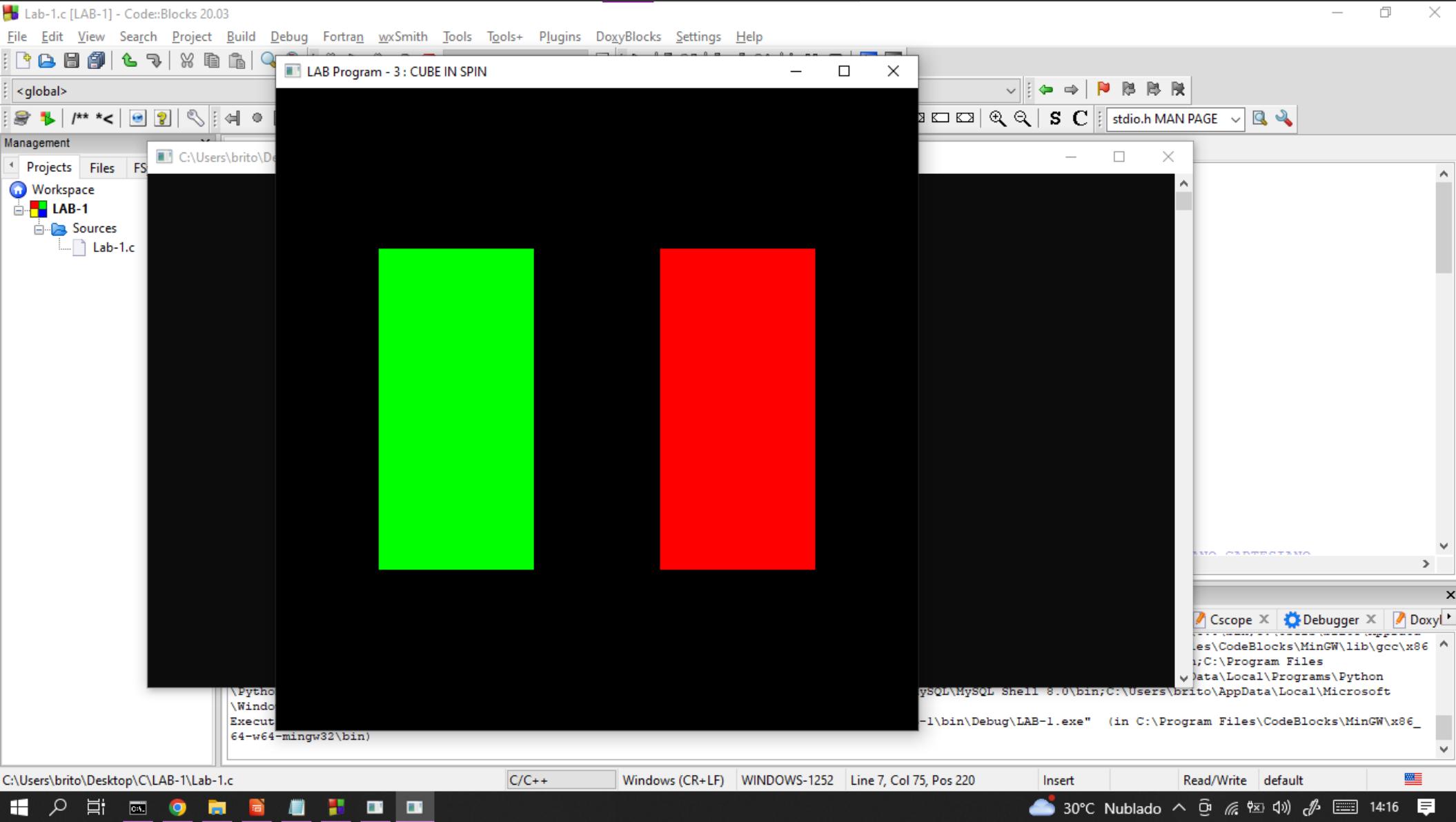


```
a++ x CppCheck/Vera++ messages x Cscope x Debugger x Doxy x
eeglut.dll;C:\GLUT\bin;C:\Program Files\CodeBlocks\MinGW\lib\gcc\x86_4\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files\gram Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\ySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft-1\bin\Debug\LAB-1.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_
```

```
\Local
_64-w6
\netca
\Pytho
\Windo
\Windo
Execut
64-w64-mingw32\bin)
```

```
3     GLfloat T = 0;                                // CRIAMOS UMA VARIÁVEL GLOBAL T
4
5     void Spin()
6     {
7         T = T + 0.25;                            // VELOCIDADE DE ROTAÇÃO
8         if(T > 360)
9             T = 0;
10        glutPostRedisplay();                   //
11    }
12
```

COMPILE DEPOIS RUN



```
4
5     void Spin()
6 {
7     T = T + 0.05;                                // VELOCIDADE DE ROTAÇÃO
8     if(T > 360)
9         T = 0;
10    glutPostRedisplay();                         //
11 }
12
```

COMPILE DEPOIS RUN

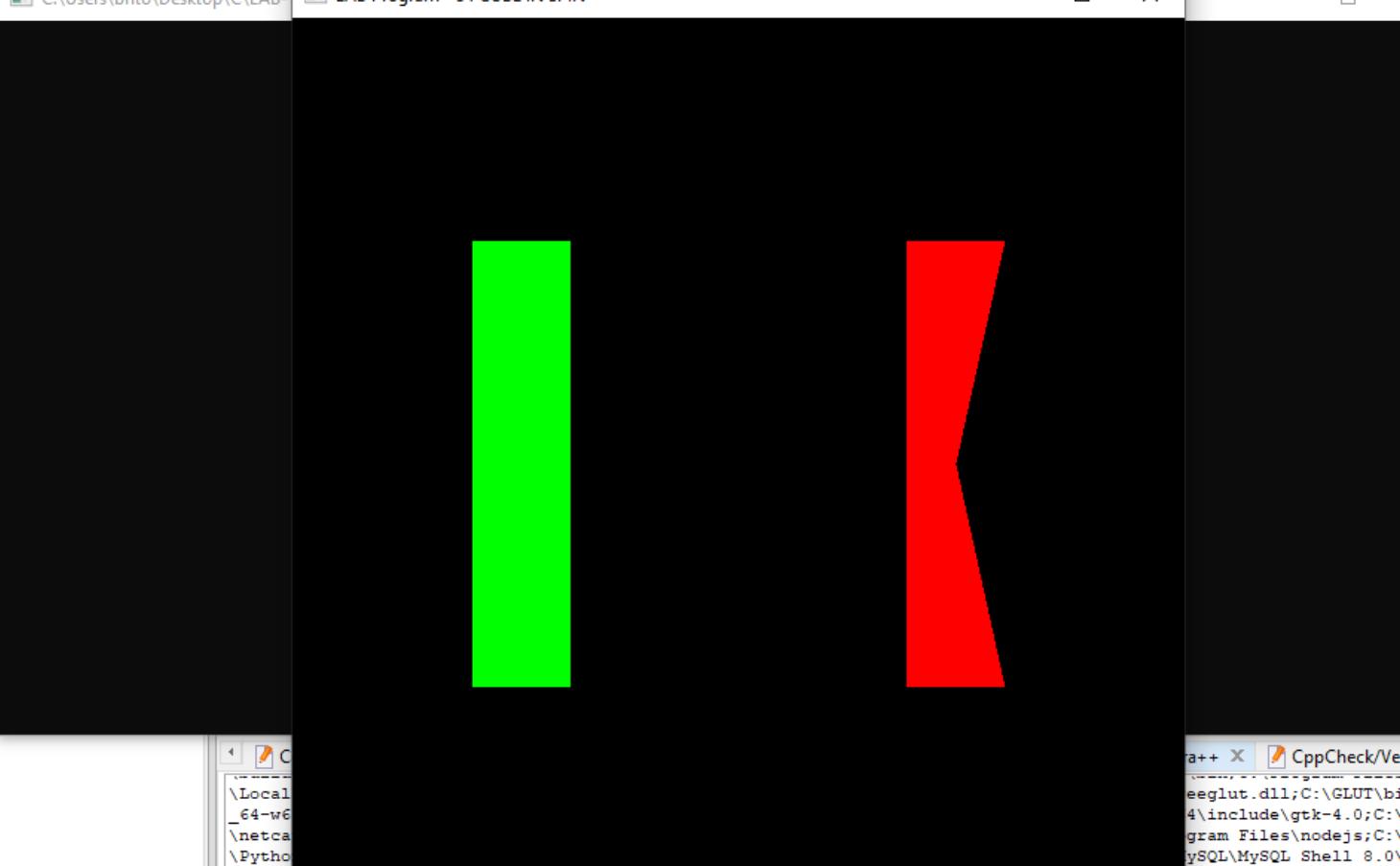
```
26     glEnd();
27 }
28
29 void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat V4[],GLfloat V5[],GLfloat V6[],GLfloat V7[])
30 {
31     glColor3f(1,0,0);
32     Face(V0,V1,V3,V2); |←
33
34     glColor3f(0,1,0);
35     Face(V4,V5,V6,V7);
36 }
37
```

COMPILE DEPOIS RUN



C:\Users\brito\Desktop\CLAB-1\Lab-1.c

LAB Program - 3 : CUBE IN SPIN



stdio.h MAN PAGE

```
ADAS NO PLANO CARTESIANO

loat V6[], GLfloat V7[])

PEDIA DA OUTRA FUNÇÃO glutDisplayFunc

ADAS
```

CppCheck/Vera++ messages Cscope Debugger Doxy

```
eeglut.dll;C:\GLUT\bin;C:\Program Files\CodeBlocks\MinGW\lib\gcc\x86_4\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\ySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\1\bin\Debug\LAB-1.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_
```

```
    .
    .
void Cube(GLfloat V0[], GLfloat V1[],
{
    glColor3f(1,0,0);
    Face(V0,V1,V2,V3);
```

VOLTE OS VALORES

```
    }

    void Cube(GLfloat V0[], GLfloat V1[], GLfloat V2[], GLfloat V3[], GLfloat V4[], GLfloat V5[], GLfloat V6[], GLfloat V7[])
    {
        glColor3f(1,0,0);
        Face(V0,V1,V2,V3); // 

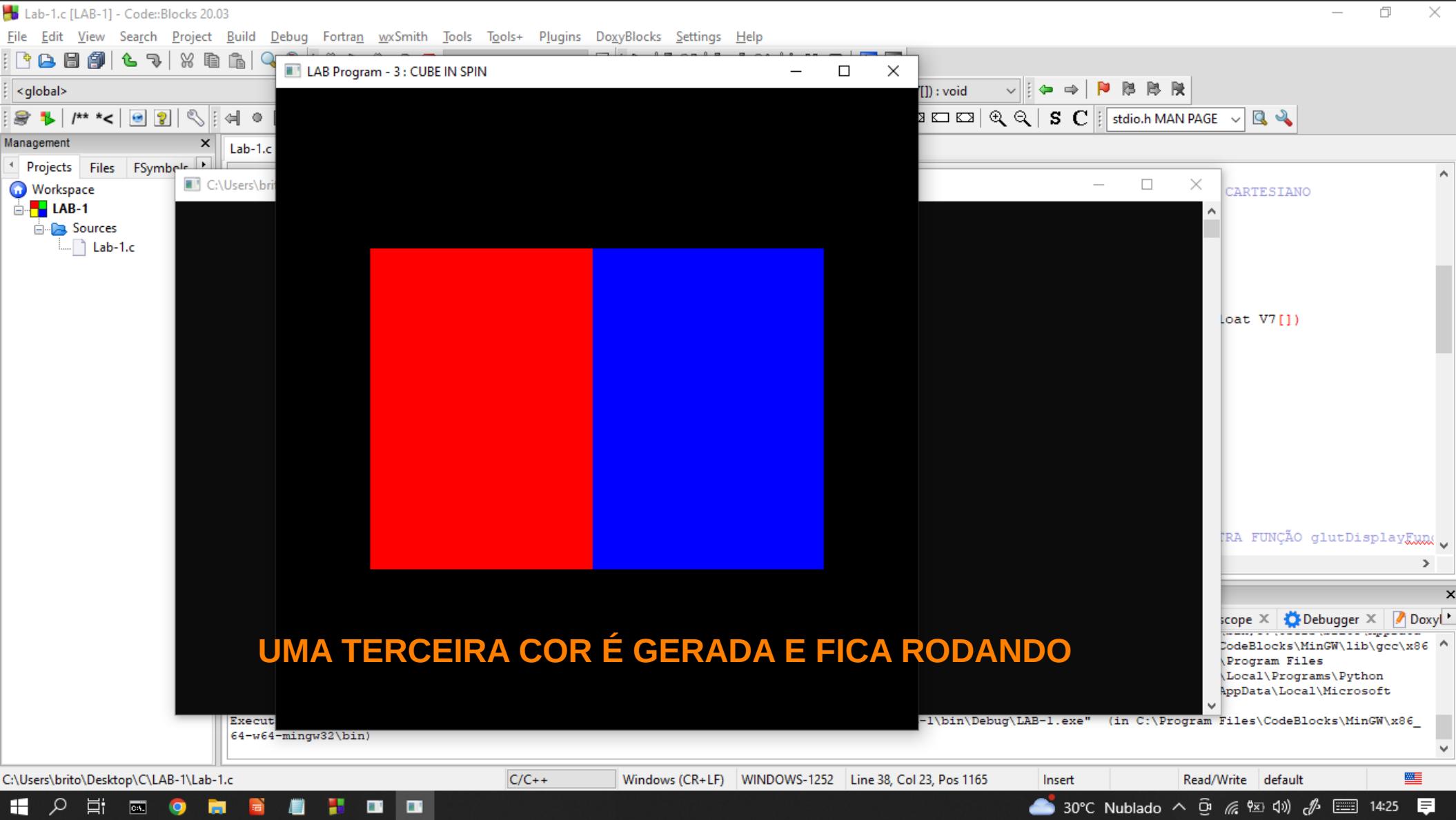
        glColor3f(0,1,0);
        Face(V4,V5,V6,V7);

        glColor3f(0,0,1);
        Face(V0,V3,V7,V4); ←

    }

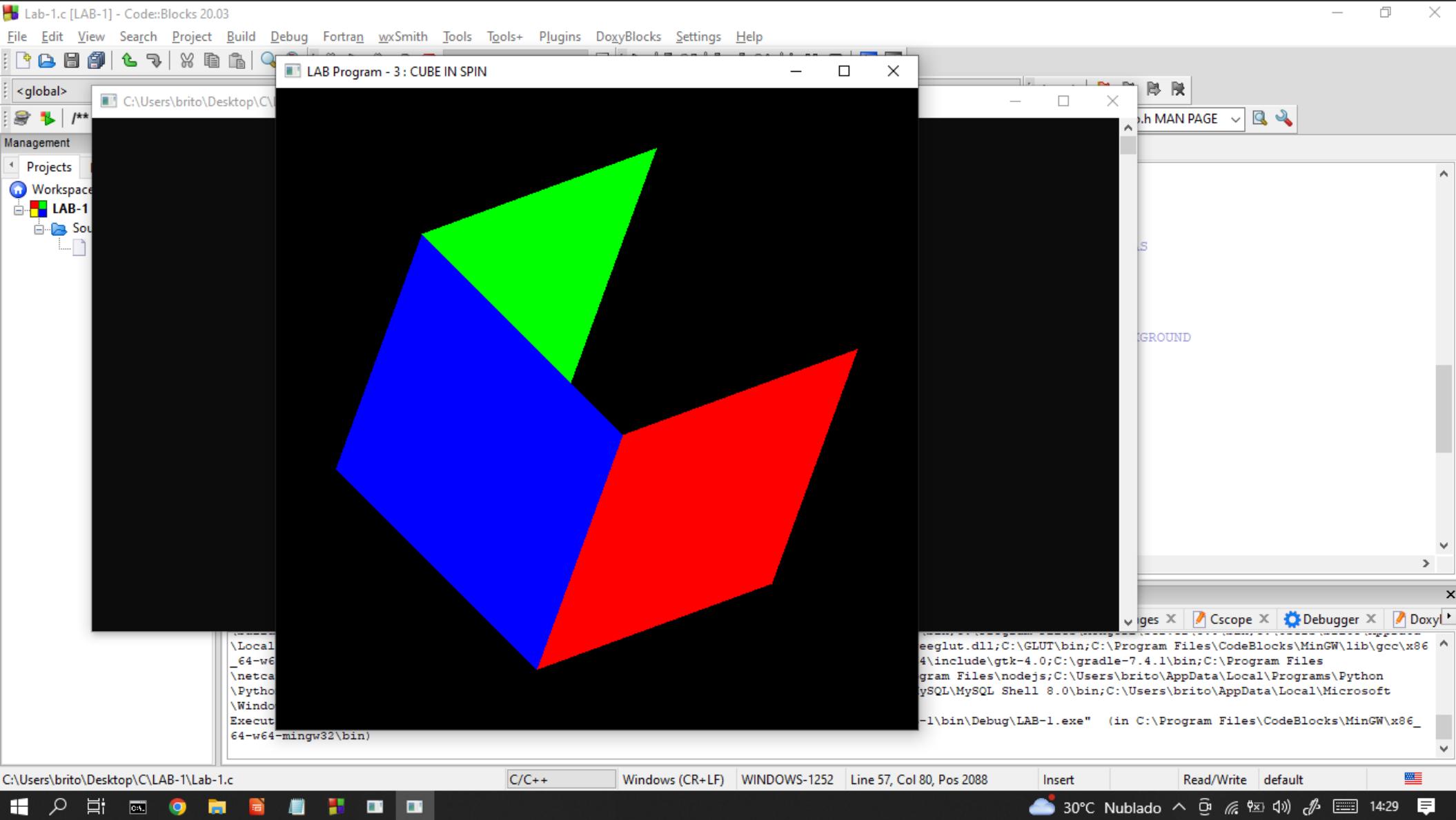
    void Draw()
    {
        // FUNÇÃO CRIADA PARA ATENDER O PEDIDA DA OUTRA FUN
    }
}
```

COMPILE E DEPOIS RUN



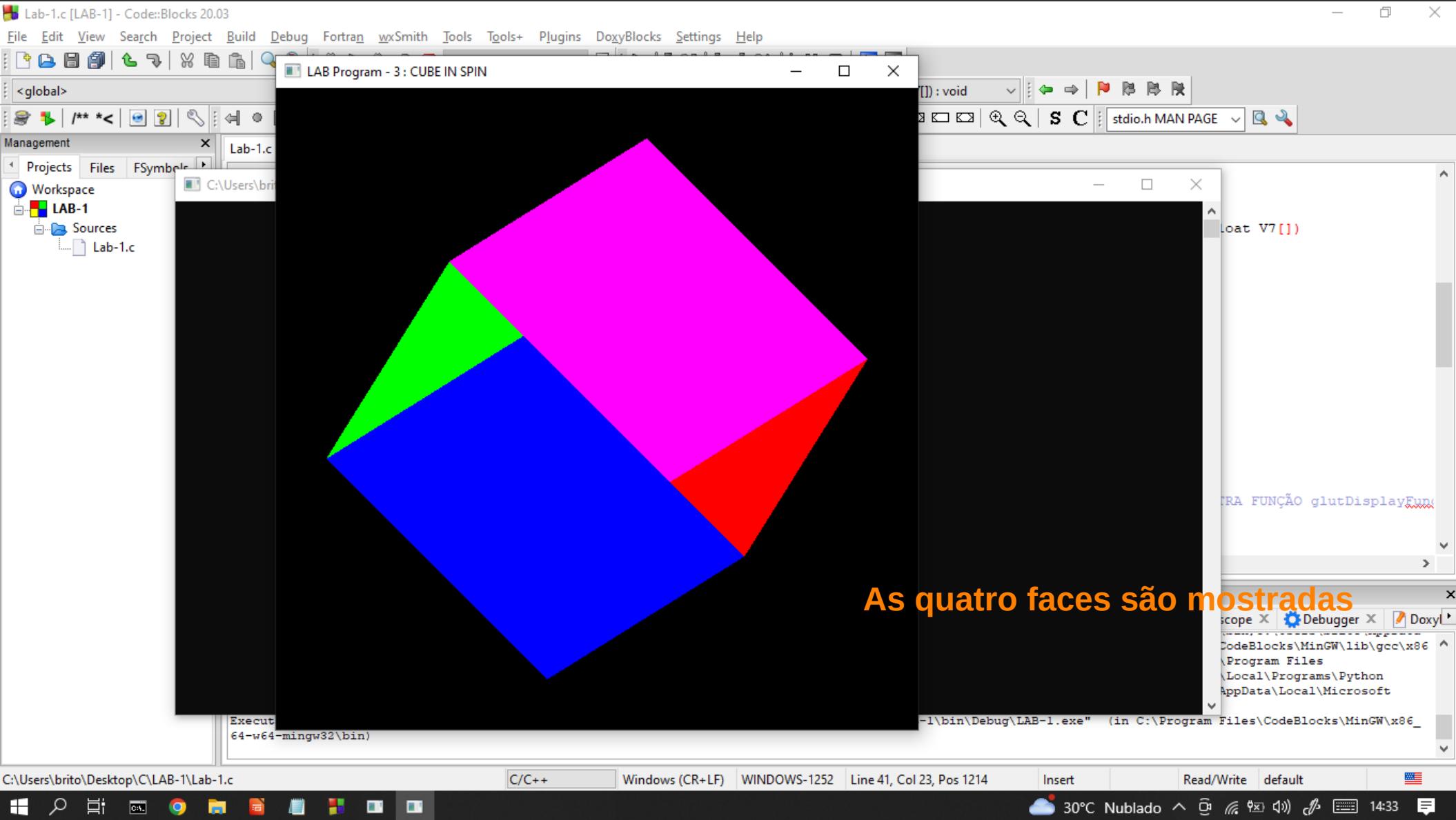
```
        (-0.5,-0.5,-0.5);
    };
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);      // LIMPPANDO O BUFFER DA COR
glLoadIdentity();
glRotatef(T,1,1,0); ←                                // ANGULOS DE ROTAÇÃO
Cube (V[0],V[1],V[2],V[3],V[4],V[5],V[6],V[7]);
/** DESENHANDO UM RETANGULO */
//glPointSize(5);                                     // TAMAÑHO DO RETANGULO
// TERMINA O RETANGULO
```

COMPILE E DEPOIS RUN



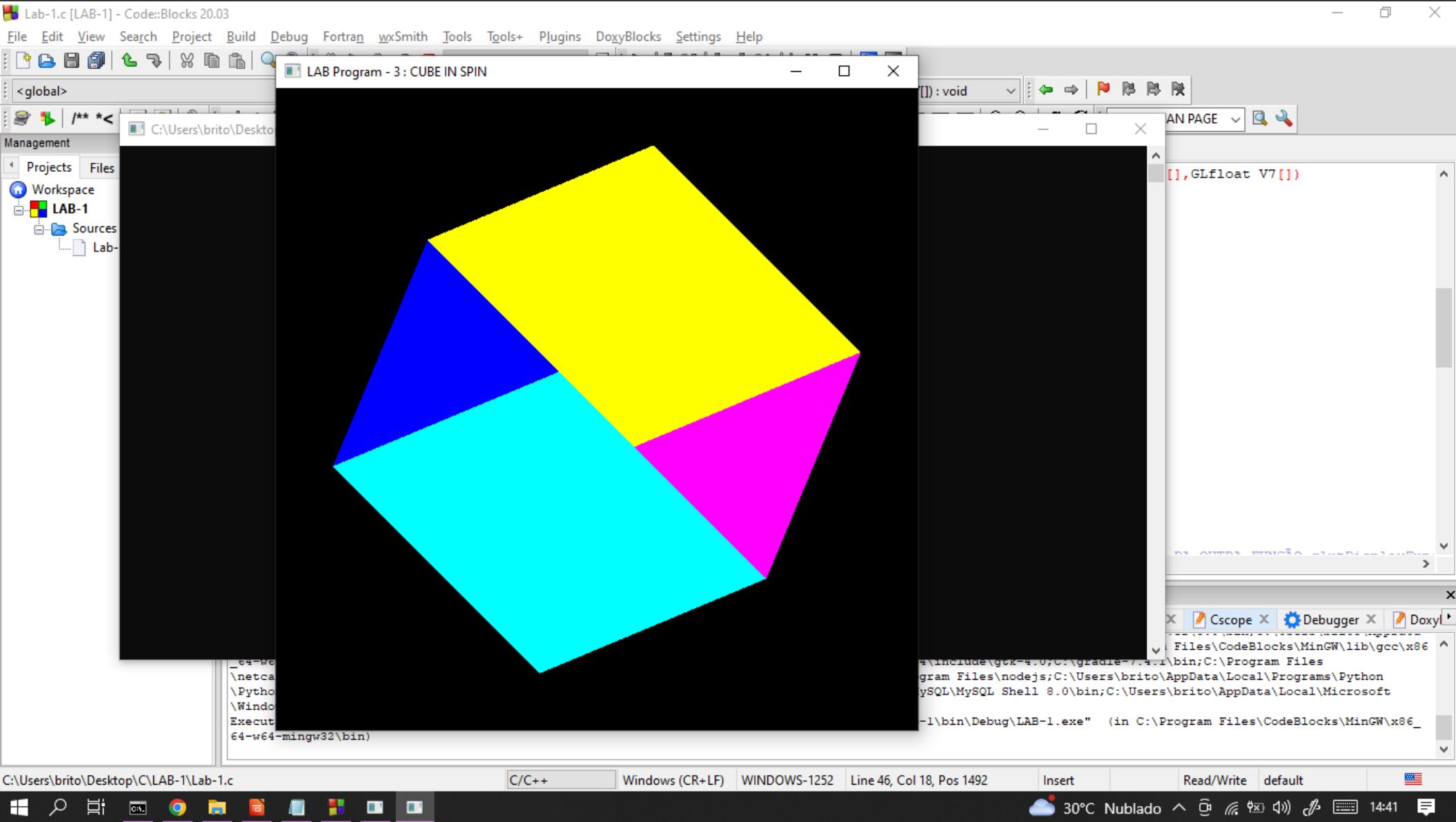
```
27 }  
28  
29     void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat V4[],GLfloat V5[  
30 {  
31     glColor3f(1,0,0);  
32     Face(V0,V1,V2,V3); //  
33  
34     glColor3f(0,1,0);  
35     Face(V4,V5,V6,V7);  
36  
37     glColor3f(0,0,1);  
38     Face(V0,V3,V7,V4);  
39  
40     glColor3f(1,0,1);  
41     Face(V1,V2,V6,V5);  
42 }  
43  
44     void Draw() // FUNÇÃO CRIADA PARA ATEN  
45 {  
46     GLfloat V[8][3] = {
```

Compile e depois run



```
*Lab-1.c X
29     void Cube(GLfloat V0[],GLfloat V1[],GLfloat V2[],GLfloat V3[],GLfloat V4[],GLfloat V5[])
30     {
31         glColor3f(1,0,0);
32         Face(V0,V1,V2,V3); // FACE FRONT
33
34         glColor3f(0,1,0); // FACE BACK
35         Face(V4,V5,V6,V7);
36
37         glColor3f(0,0,1);
38         Face(V0,V3,V7,V4); // FACE LEFT
39
40         glColor3f(1,0,1);
41         Face(V1,V2,V6,V5); // FACE RIGHT
42
43         glColor3f(1,1,0);
44         Face(V0,V1,V5,V4); // FACE TOP
45
46         glColor3f(0,1,1);
47         Face(V3,V2,V6,V7); // FACE BOTTOM
48     }
49
50     .....
51 }
```

COMPILE E DEPOIS RUN



```
2
3     void MyInit()
4     {
5         glClearColor(0,0,0,1);
6         glColor3f(1,0,0);
7         glEnable(GL_DEPTH_TEST);
8     }
9
10    void MyDisplay()
11    {
12        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
13        glLoadIdentity();
14        gluLookAt(0,0,0, 10,0,0, 0,1,0);
15        glBegin(GL_TRIANGLES);
16        glVertex3f(-1,-1,0);
17        glVertex3f(1,-1,0);
18        glVertex3f(0,1,0);
19        glEnd();
20    }
21
22    void MyReshape(int width, int height)
23    {
24        glViewport(0,0,width,height);
25        glMatrixMode(GL_PROJECTION);
26        glLoadIdentity();
27        gluPerspective(45, (double)width/(double)height, 1, 100);
28        glMatrixMode(GL_MODELVIEW);
29    }
30
31    void MyIdle()
32    {
33        glutPostRedisplay();
34    }
35
36    void MyKeyboard(unsigned char key, int x, int y)
37    {
38        if(key == 'q') exit(0);
39    }
40
41    void MyMouse(int button, int state, int x, int y)
42    {
43        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
44            glutPostRedisplay();
45    }
46
47    void MyMotion(int x, int y)
48    {
49        if(y < 0) y = 0;
50        if(y > 100) y = 100;
51        glutPostRedisplay();
52    }
53
54    void MyTimer(int value)
55    {
56        glutPostRedisplay();
57    }
58
59    void MyInit()
60    {
61        glClearColor(0,0,0,1);
62        glColor3f(1,0,0);
63        glEnable(GL_DEPTH_TEST);
64    }
65
66    void MyDisplay()
67    {
68        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
69        glLoadIdentity();
70        gluLookAt(0,0,0, 10,0,0, 0,1,0);
71        glBegin(GL_TRIANGLES);
72        glVertex3f(-1,-1,0);
73        glVertex3f(1,-1,0);
74        glVertex3f(0,1,0);
75        glEnd();
76    }
77
78    void MyReshape(int width, int height)
79    {
80        glViewport(0,0,width,height);
81        glMatrixMode(GL_PROJECTION);
82        glLoadIdentity();
83        gluPerspective(45, (double)width/(double)height, 1, 100);
84        glMatrixMode(GL_MODELVIEW);
85    }
86
87    void MyIdle()
88    {
89        glutPostRedisplay();
90    }
91
92    void MyKeyboard(unsigned char key, int x, int y)
93    {
94        if(key == 'q') exit(0);
95    }
96
97    void MyMouse(int button, int state, int x, int y)
98    {
99        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
100            glutPostRedisplay();
101    }
102
103    void MyMotion(int x, int y)
104    {
105        if(y < 0) y = 0;
106        if(y > 100) y = 100;
107        glutPostRedisplay();
108    }
109
110    void MyTimer(int value)
111    {
112        glutPostRedisplay();
113    }
114
115    void MyInit()
116    {
117        glClearColor(0,0,0,1);
118        glColor3f(1,0,0);
119        glEnable(GL_DEPTH_TEST);
120    }
121
122    void MyDisplay()
123    {
124        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
125        glLoadIdentity();
126        gluLookAt(0,0,0, 10,0,0, 0,1,0);
127        glBegin(GL_TRIANGLES);
128        glVertex3f(-1,-1,0);
129        glVertex3f(1,-1,0);
130        glVertex3f(0,1,0);
131        glEnd();
132    }
133
134    void MyReshape(int width, int height)
135    {
136        glViewport(0,0,width,height);
137        glMatrixMode(GL_PROJECTION);
138        glLoadIdentity();
139        gluPerspective(45, (double)width/(double)height, 1, 100);
140        glMatrixMode(GL_MODELVIEW);
141    }
142
143    void MyIdle()
144    {
145        glutPostRedisplay();
146    }
147
148    void MyKeyboard(unsigned char key, int x, int y)
149    {
150        if(key == 'q') exit(0);
151    }
152
153    void MyMouse(int button, int state, int x, int y)
154    {
155        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
156            glutPostRedisplay();
157    }
158
159    void MyMotion(int x, int y)
160    {
161        if(y < 0) y = 0;
162        if(y > 100) y = 100;
163        glutPostRedisplay();
164    }
165
166    void MyTimer(int value)
167    {
168        glutPostRedisplay();
169    }
170
171    void MyInit()
172    {
173        glClearColor(0,0,0,1);
174        glColor3f(1,0,0);
175        glEnable(GL_DEPTH_TEST);
176    }
177
178    void MyDisplay()
179    {
180        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
181        glLoadIdentity();
182        gluLookAt(0,0,0, 10,0,0, 0,1,0);
183        glBegin(GL_TRIANGLES);
184        glVertex3f(-1,-1,0);
185        glVertex3f(1,-1,0);
186        glVertex3f(0,1,0);
187        glEnd();
188    }
189
190    void MyReshape(int width, int height)
191    {
192        glViewport(0,0,width,height);
193        glMatrixMode(GL_PROJECTION);
194        glLoadIdentity();
195        gluPerspective(45, (double)width/(double)height, 1, 100);
196        glMatrixMode(GL_MODELVIEW);
197    }
198
199    void MyIdle()
200    {
201        glutPostRedisplay();
202    }
203
204    void MyKeyboard(unsigned char key, int x, int y)
205    {
206        if(key == 'q') exit(0);
207    }
208
209    void MyMouse(int button, int state, int x, int y)
210    {
211        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
212            glutPostRedisplay();
213    }
214
215    void MyMotion(int x, int y)
216    {
217        if(y < 0) y = 0;
218        if(y > 100) y = 100;
219        glutPostRedisplay();
220    }
221
222    void MyTimer(int value)
223    {
224        glutPostRedisplay();
225    }
226
227    void MyInit()
228    {
229        glClearColor(0,0,0,1);
230        glColor3f(1,0,0);
231        glEnable(GL_DEPTH_TEST);
232    }
233
234    void MyDisplay()
235    {
236        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
237        glLoadIdentity();
238        gluLookAt(0,0,0, 10,0,0, 0,1,0);
239        glBegin(GL_TRIANGLES);
240        glVertex3f(-1,-1,0);
241        glVertex3f(1,-1,0);
242        glVertex3f(0,1,0);
243        glEnd();
244    }
245
246    void MyReshape(int width, int height)
247    {
248        glViewport(0,0,width,height);
249        glMatrixMode(GL_PROJECTION);
250        glLoadIdentity();
251        gluPerspective(45, (double)width/(double)height, 1, 100);
252        glMatrixMode(GL_MODELVIEW);
253    }
254
255    void MyIdle()
256    {
257        glutPostRedisplay();
258    }
259
260    void MyKeyboard(unsigned char key, int x, int y)
261    {
262        if(key == 'q') exit(0);
263    }
264
265    void MyMouse(int button, int state, int x, int y)
266    {
267        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
268            glutPostRedisplay();
269    }
270
271    void MyMotion(int x, int y)
272    {
273        if(y < 0) y = 0;
274        if(y > 100) y = 100;
275        glutPostRedisplay();
276    }
277
278    void MyTimer(int value)
279    {
280        glutPostRedisplay();
281    }
282
283    void MyInit()
284    {
285        glClearColor(0,0,0,1);
286        glColor3f(1,0,0);
287        glEnable(GL_DEPTH_TEST);
288    }
289
290    void MyDisplay()
291    {
292        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
293        glLoadIdentity();
294        gluLookAt(0,0,0, 10,0,0, 0,1,0);
295        glBegin(GL_TRIANGLES);
296        glVertex3f(-1,-1,0);
297        glVertex3f(1,-1,0);
298        glVertex3f(0,1,0);
299        glEnd();
300    }
301
302    void MyReshape(int width, int height)
303    {
304        glViewport(0,0,width,height);
305        glMatrixMode(GL_PROJECTION);
306        glLoadIdentity();
307        gluPerspective(45, (double)width/(double)height, 1, 100);
308        glMatrixMode(GL_MODELVIEW);
309    }
310
311    void MyIdle()
312    {
313        glutPostRedisplay();
314    }
315
316    void MyKeyboard(unsigned char key, int x, int y)
317    {
318        if(key == 'q') exit(0);
319    }
320
321    void MyMouse(int button, int state, int x, int y)
322    {
323        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
324            glutPostRedisplay();
325    }
326
327    void MyMotion(int x, int y)
328    {
329        if(y < 0) y = 0;
330        if(y > 100) y = 100;
331        glutPostRedisplay();
332    }
333
334    void MyTimer(int value)
335    {
336        glutPostRedisplay();
337    }
338
339    void MyInit()
340    {
341        glClearColor(0,0,0,1);
342        glColor3f(1,0,0);
343        glEnable(GL_DEPTH_TEST);
344    }
345
346    void MyDisplay()
347    {
348        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
349        glLoadIdentity();
350        gluLookAt(0,0,0, 10,0,0, 0,1,0);
351        glBegin(GL_TRIANGLES);
352        glVertex3f(-1,-1,0);
353        glVertex3f(1,-1,0);
354        glVertex3f(0,1,0);
355        glEnd();
356    }
357
358    void MyReshape(int width, int height)
359    {
360        glViewport(0,0,width,height);
361        glMatrixMode(GL_PROJECTION);
362        glLoadIdentity();
363        gluPerspective(45, (double)width/(double)height, 1, 100);
364        glMatrixMode(GL_MODELVIEW);
365    }
366
367    void MyIdle()
368    {
369        glutPostRedisplay();
370    }
371
372    void MyKeyboard(unsigned char key, int x, int y)
373    {
374        if(key == 'q') exit(0);
375    }
376
377    void MyMouse(int button, int state, int x, int y)
378    {
379        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
380            glutPostRedisplay();
381    }
382
383    void MyMotion(int x, int y)
384    {
385        if(y < 0) y = 0;
386        if(y > 100) y = 100;
387        glutPostRedisplay();
388    }
389
390    void MyTimer(int value)
391    {
392        glutPostRedisplay();
393    }
394
395    void MyInit()
396    {
397        glClearColor(0,0,0,1);
398        glColor3f(1,0,0);
399        glEnable(GL_DEPTH_TEST);
400    }
401
402    void MyDisplay()
403    {
404        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
405        glLoadIdentity();
406        gluLookAt(0,0,0, 10,0,0, 0,1,0);
407        glBegin(GL_TRIANGLES);
408        glVertex3f(-1,-1,0);
409        glVertex3f(1,-1,0);
410        glVertex3f(0,1,0);
411        glEnd();
412    }
413
414    void MyReshape(int width, int height)
415    {
416        glViewport(0,0,width,height);
417        glMatrixMode(GL_PROJECTION);
418        glLoadIdentity();
419        gluPerspective(45, (double)width/(double)height, 1, 100);
420        glMatrixMode(GL_MODELVIEW);
421    }
422
423    void MyIdle()
424    {
425        glutPostRedisplay();
426    }
427
428    void MyKeyboard(unsigned char key, int x, int y)
429    {
430        if(key == 'q') exit(0);
431    }
432
433    void MyMouse(int button, int state, int x, int y)
434    {
435        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
436            glutPostRedisplay();
437    }
438
439    void MyMotion(int x, int y)
440    {
441        if(y < 0) y = 0;
442        if(y > 100) y = 100;
443        glutPostRedisplay();
444    }
445
446    void MyTimer(int value)
447    {
448        glutPostRedisplay();
449    }
450
451    void MyInit()
452    {
453        glClearColor(0,0,0,1);
454        glColor3f(1,0,0);
455        glEnable(GL_DEPTH_TEST);
456    }
457
458    void MyDisplay()
459    {
459        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
460        glLoadIdentity();
461        gluLookAt(0,0,0, 10,0,0, 0,1,0);
462        glBegin(GL_TRIANGLES);
463        glVertex3f(-1,-1,0);
464        glVertex3f(1,-1,0);
465        glVertex3f(0,1,0);
466        glEnd();
467    }
468
469    void MyReshape(int width, int height)
470    {
471        glViewport(0,0,width,height);
472        glMatrixMode(GL_PROJECTION);
473        glLoadIdentity();
474        gluPerspective(45, (double)width/(double)height, 1, 100);
475        glMatrixMode(GL_MODELVIEW);
476    }
477
478    void MyIdle()
479    {
480        glutPostRedisplay();
481    }
482
483    void MyKeyboard(unsigned char key, int x, int y)
484    {
485        if(key == 'q') exit(0);
486    }
487
488    void MyMouse(int button, int state, int x, int y)
489    {
490        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
491            glutPostRedisplay();
492    }
493
494    void MyMotion(int x, int y)
495    {
496        if(y < 0) y = 0;
497        if(y > 100) y = 100;
498        glutPostRedisplay();
499    }
500
501    void MyTimer(int value)
502    {
503        glutPostRedisplay();
504    }
505
506    void MyInit()
507    {
508        glClearColor(0,0,0,1);
509        glColor3f(1,0,0);
510        glEnable(GL_DEPTH_TEST);
511    }
512
513    void MyDisplay()
514    {
514        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
515        glLoadIdentity();
516        gluLookAt(0,0,0, 10,0,0, 0,1,0);
517        glBegin(GL_TRIANGLES);
518        glVertex3f(-1,-1,0);
519        glVertex3f(1,-1,0);
520        glVertex3f(0,1,0);
521        glEnd();
522    }
523
524    void MyReshape(int width, int height)
525    {
526        glViewport(0,0,width,height);
527        glMatrixMode(GL_PROJECTION);
528        glLoadIdentity();
529        gluPerspective(45, (double)width/(double)height, 1, 100);
530        glMatrixMode(GL_MODELVIEW);
531    }
532
533    void MyIdle()
534    {
535        glutPostRedisplay();
536    }
537
538    void MyKeyboard(unsigned char key, int x, int y)
539    {
540        if(key == 'q') exit(0);
541    }
542
543    void MyMouse(int button, int state, int x, int y)
544    {
545        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
546            glutPostRedisplay();
547    }
548
549    void MyMotion(int x, int y)
550    {
551        if(y < 0) y = 0;
552        if(y > 100) y = 100;
553        glutPostRedisplay();
554    }
555
556    void MyTimer(int value)
557    {
558        glutPostRedisplay();
559    }
560
561    void MyInit()
562    {
563        glClearColor(0,0,0,1);
564        glColor3f(1,0,0);
565        glEnable(GL_DEPTH_TEST);
566    }
567
568    void MyDisplay()
569    {
569        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
570        glLoadIdentity();
571        gluLookAt(0,0,0, 10,0,0, 0,1,0);
572        glBegin(GL_TRIANGLES);
573        glVertex3f(-1,-1,0);
574        glVertex3f(1,-1,0);
575        glVertex3f(0,1,0);
576        glEnd();
577    }
578
579    void MyReshape(int width, int height)
580    {
581        glViewport(0,0,width,height);
582        glMatrixMode(GL_PROJECTION);
583        glLoadIdentity();
584        gluPerspective(45, (double)width/(double)height, 1, 100);
585        glMatrixMode(GL_MODELVIEW);
586    }
587
588    void MyIdle()
589    {
590        glutPostRedisplay();
591    }
592
593    void MyKeyboard(unsigned char key, int x, int y)
594    {
595        if(key == 'q') exit(0);
596    }
597
598    void MyMouse(int button, int state, int x, int y)
599    {
600        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
601            glutPostRedisplay();
602    }
603
604    void MyMotion(int x, int y)
605    {
606        if(y < 0) y = 0;
607        if(y > 100) y = 100;
608        glutPostRedisplay();
609    }
610
611    void MyTimer(int value)
612    {
613        glutPostRedisplay();
614    }
615
616    void MyInit()
617    {
618        glClearColor(0,0,0,1);
619        glColor3f(1,0,0);
620        glEnable(GL_DEPTH_TEST);
621    }
622
623    void MyDisplay()
624    {
624        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
625        glLoadIdentity();
626        gluLookAt(0,0,0, 10,0,0, 0,1,0);
627        glBegin(GL_TRIANGLES);
628        glVertex3f(-1,-1,0);
629        glVertex3f(1,-1,0);
630        glVertex3f(0,1,0);
631        glEnd();
632    }
633
634    void MyReshape(int width, int height)
635    {
636        glViewport(0,0,width,height);
637        glMatrixMode(GL_PROJECTION);
638        glLoadIdentity();
639        gluPerspective(45, (double)width/(double)height, 1, 100);
640        glMatrixMode(GL_MODELVIEW);
641    }
642
643    void MyIdle()
644    {
645        glutPostRedisplay();
646    }
647
648    void MyKeyboard(unsigned char key, int x, int y)
649    {
650        if(key == 'q') exit(0);
651    }
652
653    void MyMouse(int button, int state, int x, int y)
654    {
655        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
656            glutPostRedisplay();
657    }
658
659    void MyMotion(int x, int y)
660    {
661        if(y < 0) y = 0;
662        if(y > 100) y = 100;
663        glutPostRedisplay();
664    }
665
666    void MyTimer(int value)
667    {
668        glutPostRedisplay();
669    }
670
671    void MyInit()
672    {
673        glClearColor(0,0,0,1);
674        glColor3f(1,0,0);
675        glEnable(GL_DEPTH_TEST);
676    }
677
678    void MyDisplay()
679    {
679        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
680        glLoadIdentity();
681        gluLookAt(0,0,0, 10,0,0, 0,1,0);
682        glBegin(GL_TRIANGLES);
683        glVertex3f(-1,-1,0);
684        glVertex3f(1,-1,0);
685        glVertex3f(0,1,0);
686        glEnd();
687    }
688
689    void MyReshape(int width, int height)
690    {
691        glViewport(0,0,width,height);
692        glMatrixMode(GL_PROJECTION);
693        glLoadIdentity();
694        gluPerspective(45, (double)width/(double)height, 1, 100);
695        glMatrixMode(GL_MODELVIEW);
696    }
697
698    void MyIdle()
699    {
700        glutPostRedisplay();
701    }
702
703    void MyKeyboard(unsigned char key, int x, int y)
704    {
705        if(key == 'q') exit(0);
706    }
707
708    void MyMouse(int button, int state, int x, int y)
709    {
710        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
711            glutPostRedisplay();
712    }
713
714    void MyMotion(int x, int y)
715    {
716        if(y < 0) y = 0;
717        if(y > 100) y = 100;
718        glutPostRedisplay();
719    }
720
721    void MyTimer(int value)
722    {
723        glutPostRedisplay();
724    }
725
726    void MyInit()
727    {
728        glClearColor(0,0,0,1);
729        glColor3f(1,0,0);
730        glEnable(GL_DEPTH_TEST);
731    }
732
733    void MyDisplay()
734    {
734        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
735        glLoadIdentity();
736        gluLookAt(0,0,0, 10,0,0, 0,1,0);
737        glBegin(GL_TRIANGLES);
738        glVertex3f(-1,-1,0);
739        glVertex3f(1,-1,0);
740        glVertex3f(0,1,0);
741        glEnd();
742    }
743
744    void MyReshape(int width, int height)
745    {
746        glViewport(0,0,width,height);
747        glMatrixMode(GL_PROJECTION);
748        glLoadIdentity();
749        gluPerspective(45, (double)width/(double)height, 1, 100);
750        glMatrixMode(GL_MODELVIEW);
751    }
752
753    void MyIdle()
754    {
755        glutPostRedisplay();
756    }
757
758    void MyKeyboard(unsigned char key, int x, int y)
759    {
760        if(key == 'q') exit(0);
761    }
762
763    void MyMouse(int button, int state, int x, int y)
764    {
765        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
766            glutPostRedisplay();
767    }
768
769    void MyMotion(int x, int y)
770    {
771        if(y < 0) y = 0;
772        if(y > 100) y = 100;
773        glutPostRedisplay();
774    }
775
776    void MyTimer(int value)
777    {
778        glutPostRedisplay();
779    }
780
781    void MyInit()
782    {
783        glClearColor(0,0,0,1);
784        glColor3f(1,0,0);
785        glEnable(GL_DEPTH_TEST);
786    }
787
788    void MyDisplay()
789    {
789        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
790        glLoadIdentity();
791        gluLookAt(0,0,0, 10,0,0, 0,1,0);
792        glBegin(GL_TRIANGLES);
793        glVertex3f(-1,-1,0);
794        glVertex3f(1,-1,0);
795        glVertex3f(0,1,0);
796        glEnd();
797    }
798
799    void MyReshape(int width, int height)
800    {
801        glViewport(0,0,width,height);
802        glMatrixMode(GL_PROJECTION);
803        glLoadIdentity();
804        gluPerspective(45, (double)width/(double)height, 1, 100);
805        glMatrixMode(GL_MODELVIEW);
806    }
807
808    void MyIdle()
809    {
810        glutPostRedisplay();
811    }
812
813    void MyKeyboard(unsigned char key, int x, int y)
814    {
815        if(key == 'q') exit(0);
816    }
817
818    void MyMouse(int button, int state, int x, int y)
819    {
820        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
821            glutPostRedisplay();
822    }
823
824    void MyMotion(int x, int y)
825    {
826        if(y < 0) y = 0;
827        if(y > 100) y = 100;
828        glutPostRedisplay();
829    }
830
831    void MyTimer(int value)
832    {
833        glutPostRedisplay();
834    }
835
836    void MyInit()
837    {
838        glClearColor(0,0,0,1);
839        glColor3f(1,0,0);
840        glEnable(GL_DEPTH_TEST);
841    }
842
843    void MyDisplay()
844    {
844        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
845        glLoadIdentity();
846        gluLookAt(0,0,0, 10,0,0, 0,1,0);
847        glBegin(GL_TRIANGLES);
848        glVertex3f(-1,-1,0);
849        glVertex3f(1,-1,0);
850        glVertex3f(0,1,0);
851        glEnd();
852    }
853
854    void MyReshape(int width, int height)
855    {
856        glViewport(0,0,width,height);
857        glMatrixMode(GL_PROJECTION);
858        glLoadIdentity();
859        gluPerspective(45, (double)width/(double)height, 1, 100);
860        glMatrixMode(GL_MODELVIEW);
861    }
862
863    void MyIdle()
864    {
865        glutPostRedisplay();
866    }
867
868    void MyKeyboard(unsigned char key, int x, int y)
869    {
870        if(key == 'q') exit(0);
871    }
872
873    void MyMouse(int button, int state, int x, int y)
874    {
875        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
876            glutPostRedisplay();
877    }
878
879    void MyMotion(int x, int y)
880    {
881        if(y < 0) y = 0;
882        if(y > 100) y = 100;
883        glutPostRedisplay();
884    }
885
886    void MyTimer(int value)
887    {
888        glutPostRedisplay();
889    }
890
891    void MyInit()
892    {
893        glClearColor(0,0,0,1);
894        glColor3f(1,0,0);
895        glEnable(GL_DEPTH_TEST);
896    }
897
898    void MyDisplay()
899    {
899        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
900        glLoadIdentity();
901        gluLookAt(0,0,0, 10,0,0, 0,1,0);
902        glBegin(GL_TRIANGLES);
903        glVertex3f(-1,-1,0);
904        glVertex3f(1,-1,0);
905        glVertex3f(0,1,0);
906        glEnd();
907    }
908
909    void MyReshape(int width, int height)
910    {
911        glViewport(0,0,width,height);
912        glMatrixMode(GL_PROJECTION);
913        glLoadIdentity();
914        gluPerspective(45, (double)width/(double)height, 1, 100);
915        glMatrixMode(GL_MODELVIEW);
916    }
917
918    void MyIdle()
919    {
920        glutPostRedisplay();
921    }
922
923    void MyKeyboard(unsigned char key, int x, int y)
924    {
925        if(key == 'q') exit(0);
926    }
927
928    void MyMouse(int button, int state, int x, int y)
929    {
930        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
931            glutPostRedisplay();
932    }
933
934    void MyMotion(int x, int y)
935    {
936        if(y < 0) y = 0;
937        if(y > 100) y = 100;
938        glutPostRedisplay();
939    }
940
941    void MyTimer(int value)
942    {
943        glutPostRedisplay();
944    }
945
946    void MyInit()
947    {
948        glClearColor(0,0,0,1);
949        glColor3f(1,0,0);
950        glEnable(GL_DEPTH_TEST);
951    }
952
953    void MyDisplay()
954    {
954        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
955        glLoadIdentity();
956        gluLookAt(0,0,0, 10,0,0, 0,1,0);
957        glBegin(GL_TRIANGLES);
958        glVertex3f(-1,-1,0);
959        glVertex3f(1,-1,0);
960        glVertex3f(0,1,0);
961        glEnd();
962    }
963
964    void MyReshape(int width, int height)
965    {
966        glViewport(0,0,width,height);
967        glMatrixMode(GL_PROJECTION);
968        glLoadIdentity();
969        gluPerspective(45, (double)width/(double)height, 1, 100);
970        glMatrixMode(GL_MODELVIEW);
971    }
972
973    void MyIdle()
974    {
975        glutPostRedisplay();
976    }
977
978    void MyKeyboard(unsigned char key, int x, int y)
979    {
980        if(key == 'q') exit(0);
981    }
982
983    void MyMouse(int button, int state, int x, int y)
984    {
985        if(button == GLUT_LEFT_BUTTON & state == GLUT_DOWN)
986            glutPostRedisplay();
987    }
988
989    void MyMotion(int x, int y)
990    {
991        if(y < 0) y = 0;
992        if(y > 100) y = 100;
993        glutPostRedisplay();
994    }
995
996    void MyTimer(int value)
997    {
998        glutPostRedisplay();
999    }
1000}
```



```
    }

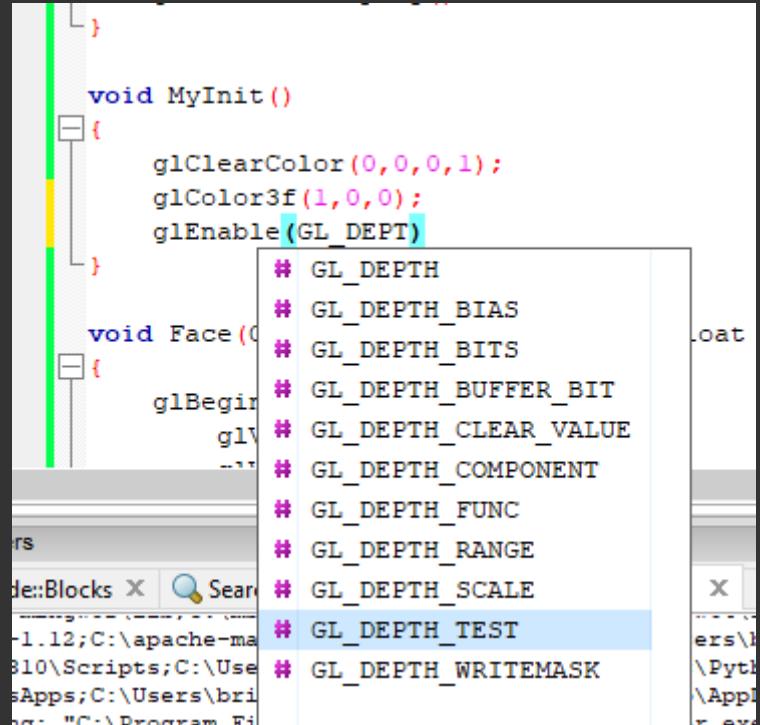
    void MyInit()
    {
        glClearColor(0,0,0,1);
        glColor3f(1,0,0);
        glEnable(GL_DEPTH)

    }

    void Face()
    {
        glBegin(GL_TRIANGLES);
        glVertex3f(-0.5,-0.5,0);
        glVertex3f(0.5,-0.5,0);
        glVertex3f(0,0.5,0);
    }

    void draw()
    {
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glLoadIdentity();
        glTranslatef(0,0,-5);
        glRotatef(45,1,0,0);
        Face();
        glFlush();
    }
}

int main()
{
    glutInit(&argc,&argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
    glutCreateWindow("My OpenGL Window");
    glutResiz...
```



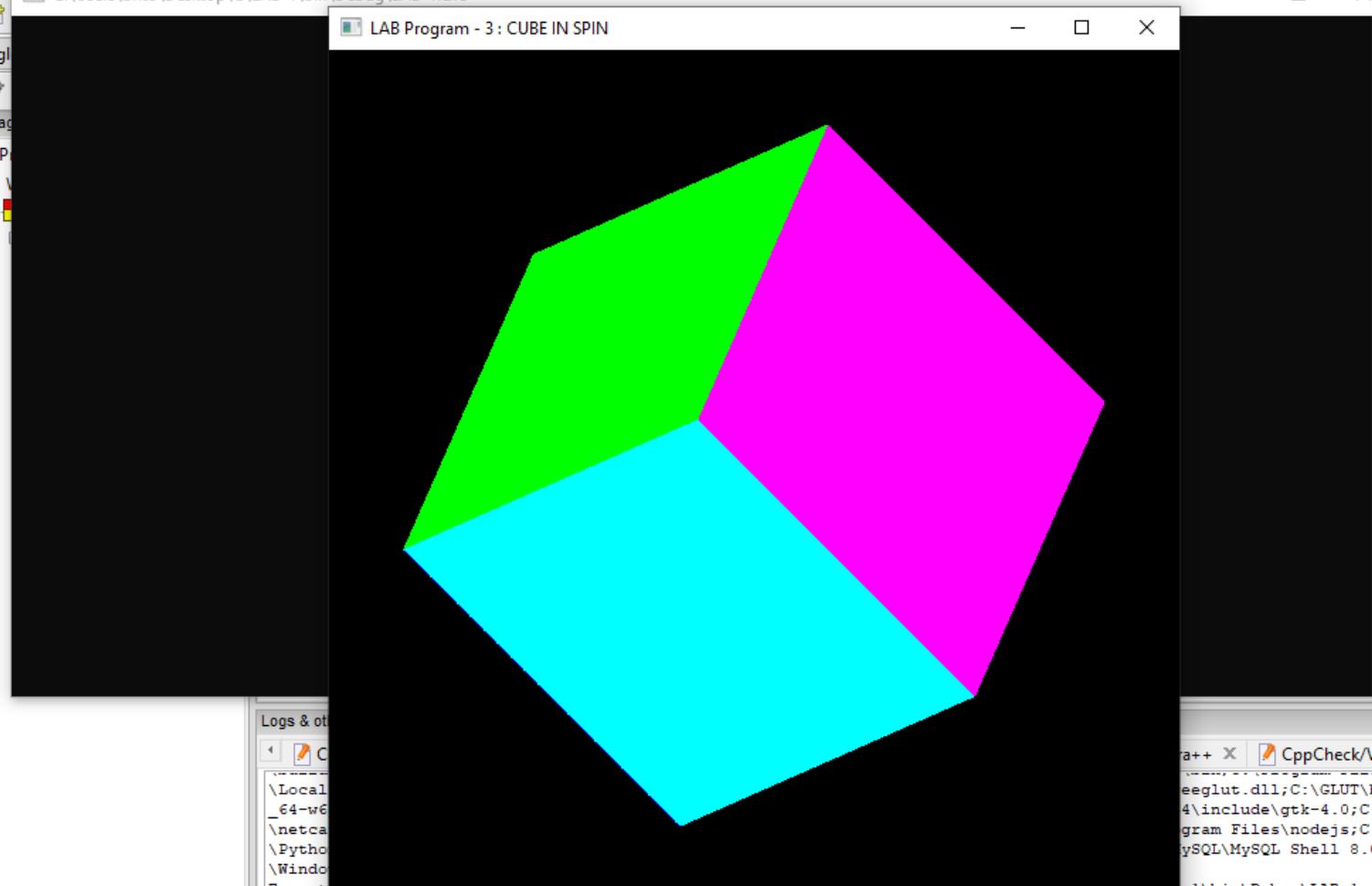
The screenshot shows a code editor with an open tooltip menu. The menu lists several OpenGL constants related to depth testing and buffer management. The item '# GL_DEPTH_TEST' is highlighted with a blue background, indicating it is the current selection or being hovered over.

- # GL_DEPTH
- # GL_DEPTH_BIAS
- # GL_DEPTH_BITS
- # GL_DEPTH_BUFFER_BIT
- # GL_DEPTH_CLEAR_VALUE
- # GL_DEPTH_COMPONENT
- # GL_DEPTH_FUNC
- # GL_DEPTH_RANGE
- # GL_DEPTH_SCALE
- # GL_DEPTH_TEST
- # GL_DEPTH_WRITEMASK

```
1  - }
2
3  void MyInit()
4  {
5      glClearColor(0,0,0,1);           // FUNÇÃO PARAS AS CORES DA JANELA
6      glColor3f(1,0,0);             // RECEBE VALORES FLOAT PARA DAR COR A JANELA
7      glEnable(GL_DEPTH_TEST);       // |
8
9 }
```

COMPILE E DEPOIS RUN

C:\Users\brito\Desktop\CLAB-1\bin\Debug\LAB-1.exe



stdio.h MAN PAGE

JANELA

DENADAS NO PLANO CARTESIANO

a++ x CppCheck/Vera++ messages x Cscope x Debugger x Doxygen x

```
eeglut.dll;C:\GLUT\bin;C:\Program Files\CodeBlocks\MinGW\lib\gcc\x86_4\include\gtk-4.0;C:\gradle-7.4.1\bin;C:\Program Files\nodejs;C:\Users\brito\AppData\Local\Programs\Python\ySQL\MySQL Shell 8.0\bin;C:\Users\brito\AppData\Local\Microsoft\1\bin\Debug\LAB-1.exe" (in C:\Program Files\CodeBlocks\MinGW\x86_
```