

# Základy počítačové grafiky

Martin Němec

# Nový projekt

## Create a new project

### Recent project templates

A list of your recently accessed templates will be displayed here.

—

📄

×


🔍

[Clear all](#)

C++

Windows


All project types

**Empty Project**  
Start from scratch with C++ for Windows. Provides no starting files.

C++

Windows


Console

**Console App**  
Run code in a Windows terminal. Prints "Hello World" by default.

C++

Windows

Console

**Windows Desktop Wizard**  
Create your own Windows app using a wizard.


C++

Windows

Console

Desktop


Library

**Windows Desktop Application**  
A project for an application with a graphical user interface that runs on Windows.

C++

Windows

Desktop

**Shared Items Project**  
A Shared Items project is used for sharing files between multiple projects.

C++

Android

iOS

Linux

Windows

Console

Desktop

Games

Library

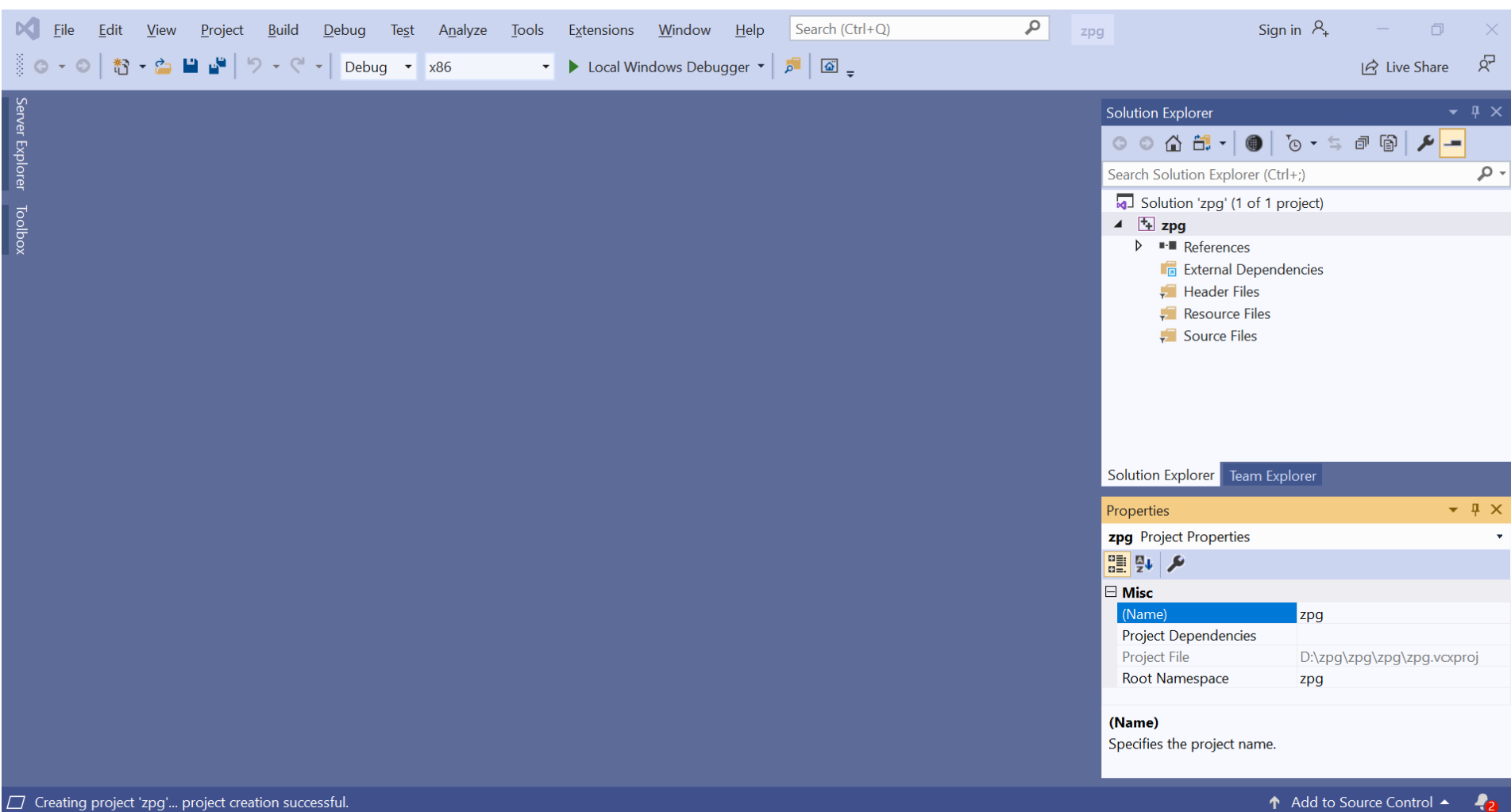
Mobile

11/11/20

Back

Next

# Nový projekt



# Hello World

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    std::cout << "Hello World!\n";
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) zpg Sign in

Process: [324] zpg.exe Lifecycle Events Thread: Stack Frame:

Source.cpp

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hello World!\n";
6     system("PAUSE");
7     return 0;
8 }
```

(Global Scope) main()

100 % No issues found Ln: 8 Ch: 2 TABS CRLF

D:\zpg\zpg\Debug\zpg.exe

Hello World!  
Press any key to continue . . .

Diagnostics Tools

Diagnostics session: 18 seconds

Events

Process Memory (MB)

Summary Events Memory Usage CPU Usage

Events

Show Events (0 of 0)

Memory Usage

Take Snapshot

Autos

Search (Ctrl+E) Search Depth:

Name	Value	Type
------	-------	------

Autos Locals Watch 1

Call Stack

Name	Lang
------	------

Call Stack Breakpoints Exception Settings Command Window Immediate Window Output

# OpenGL – main.cpp

<https://blender.vsb.cz/download/zpg/main.cpp>

```
//Include GLFW
```

```
#include <GLFW/glfw3.h>
```

```
//Include GLM
```

```
#include <glm/vec3.hpp> // glm::vec3
```

```
#include <glm/vec4.hpp> // glm::vec4
```

```
#include <glm/mat4x4.hpp> // glm::mat4
```

```
#include <glm/gtc/matrix_transform.hpp> //
```

```
glm::translate, glm::rotate, glm::scale,
```

```
glm::perspective
```

```
#include <glm/gtc/type_ptr.hpp> // glm::value_ptr
```

```
//Include the standard C++ headers
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

# OpenGL – main.cpp

The screenshot displays the Visual Studio Code interface with the following components:

- Source.cpp\* (zpg):** The main C++ file containing the following code:

```
1 //Include GLFW
2 #include <GLFW/glfw3.h>
3
4 //Include GLM
5 #include <glm/vec3.hpp> // glm::vec3
6 #include <glm/vec4.hpp> // glm::vec4
7 #include <glm/mat4x4.hpp> // glm::mat4
8 #include <glm/gtc/matrix_transform.hpp> // glm::translate, glm::rotate, glm::scale, glm::perspective
9 #include <glm/gtc/type_ptr.hpp> // glm::value_ptr
10
11 //Include the standard C++ headers
12 #include <stdlib.h>
13 #include <stdio.h>
14
15
16
17 static void error_callback(int error, const char* description) { fputs(description, stderr); }
18
19 static void key_callback(GLFWwindow* window, int key, int scancode, int action, int mode)
```

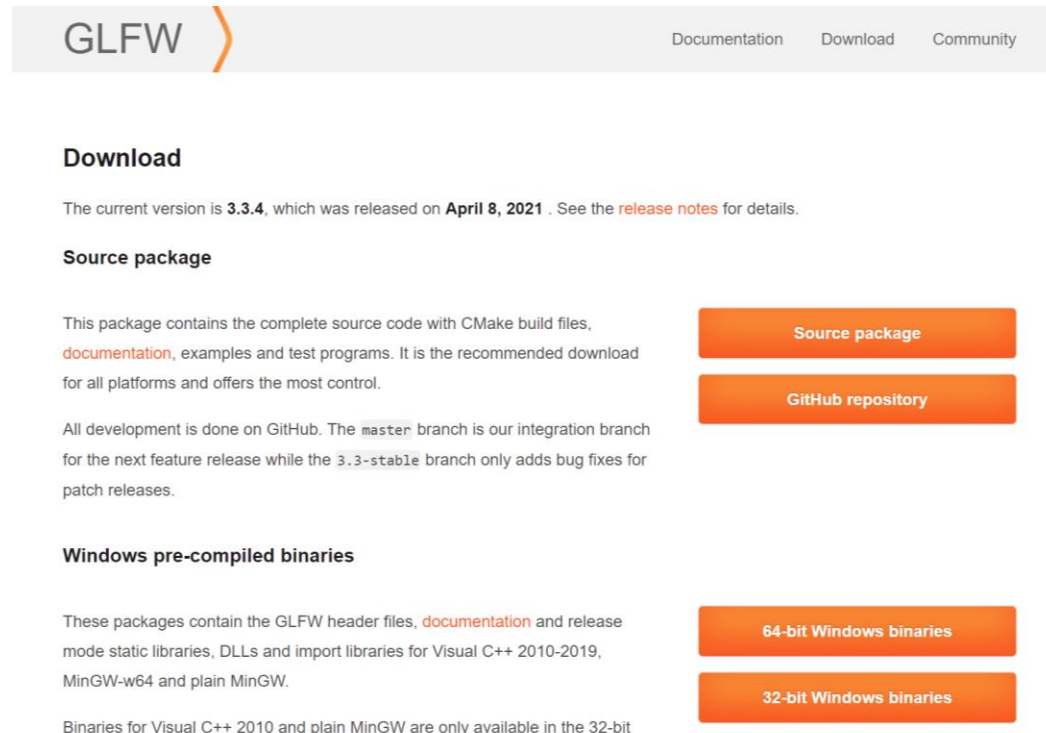
The **Output** window shows the following debug output:

```
zpg.exe (Win32): Loaded 'C:\Windows\System32\user32.dll'.
'zpg.exe' (Win32): Loaded 'C:\Windows\System32\advapi32.dll'.
'zpg.exe' (Win32): Loaded 'C:\Windows\System32\msvcrt.dll'.
'zpg.exe' (Win32): Loaded 'C:\Windows\System32\kernel.appcore.dll'.
The thread 0xa8c has exited with code 0 (0x0).
The thread 0x670 has exited with code 0 (0x0).
The program '[324] zpg.exe' has exited with code 0 (0x0).
```

The **Solution Explorer** on the right shows the project structure for 'zpg' (1 of 1 project), including Source Files and Source.cpp.

# Budeme potřebovat knihovny

- <https://www.glfw.org/download>
- Source vs. binaries?
- Dodržet verze (32bit vs. 64bit)



The screenshot shows the GLFW website's 'Download' page. At the top, there is a navigation bar with the GLFW logo and links for 'Documentation', 'Download', and 'Community'. The main heading is 'Download'. Below it, a paragraph states: 'The current version is **3.3.4**, which was released on **April 8, 2021**. See the [release notes](#) for details.' The section 'Source package' follows, describing the source code and providing two orange buttons: 'Source package' and 'GitHub repository'. The 'Windows pre-compiled binaries' section describes pre-compiled binaries for Windows and provides two orange buttons: '64-bit Windows binaries' and '32-bit Windows binaries'. The page is clean with a white background and orange accents.

GLFW

Documentation Download Community

## Download

The current version is **3.3.4**, which was released on **April 8, 2021**. See the [release notes](#) for details.

### Source package

This package contains the complete source code with CMake build files, [documentation](#), examples and test programs. It is the recommended download for all platforms and offers the most control.

All development is done on GitHub. The `master` branch is our integration branch for the next feature release while the `3.3-stable` branch only adds bug fixes for patch releases.

**Source package**

**GitHub repository**

### Windows pre-compiled binaries

These packages contain the GLFW header files, [documentation](#) and release mode static libraries, DLLs and import libraries for Visual C++ 2010-2019, MinGW-w64 and plain MinGW.

**64-bit Windows binaries**

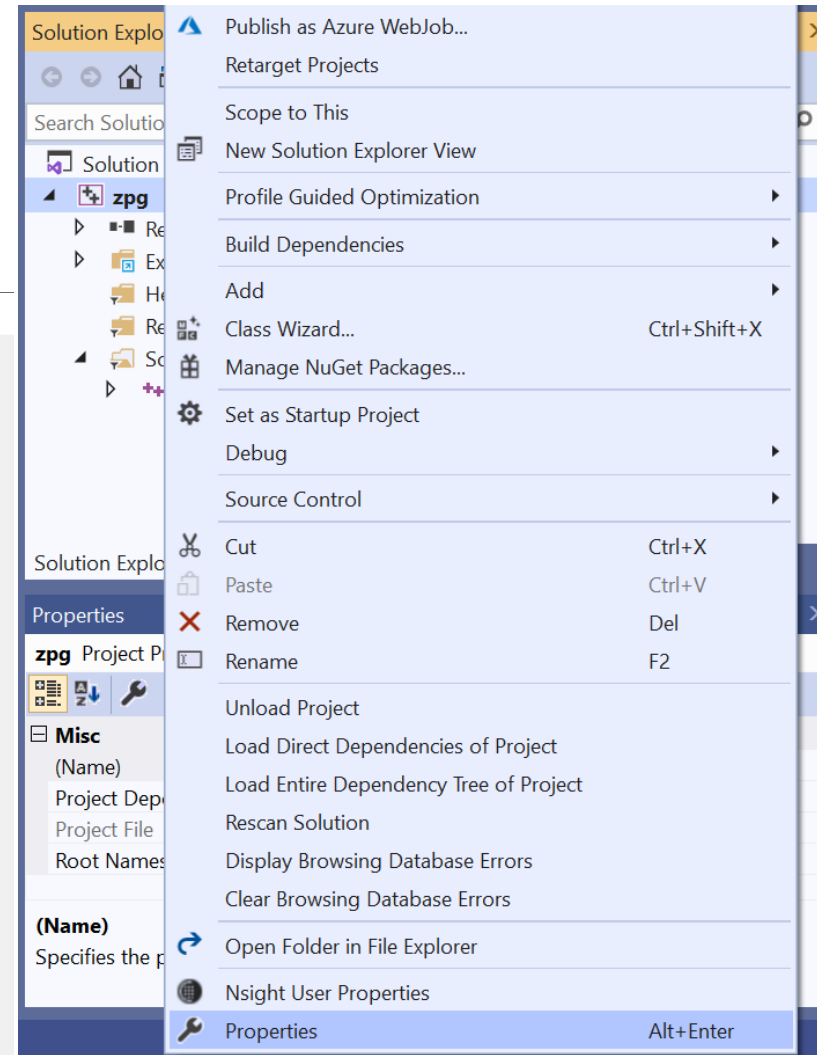
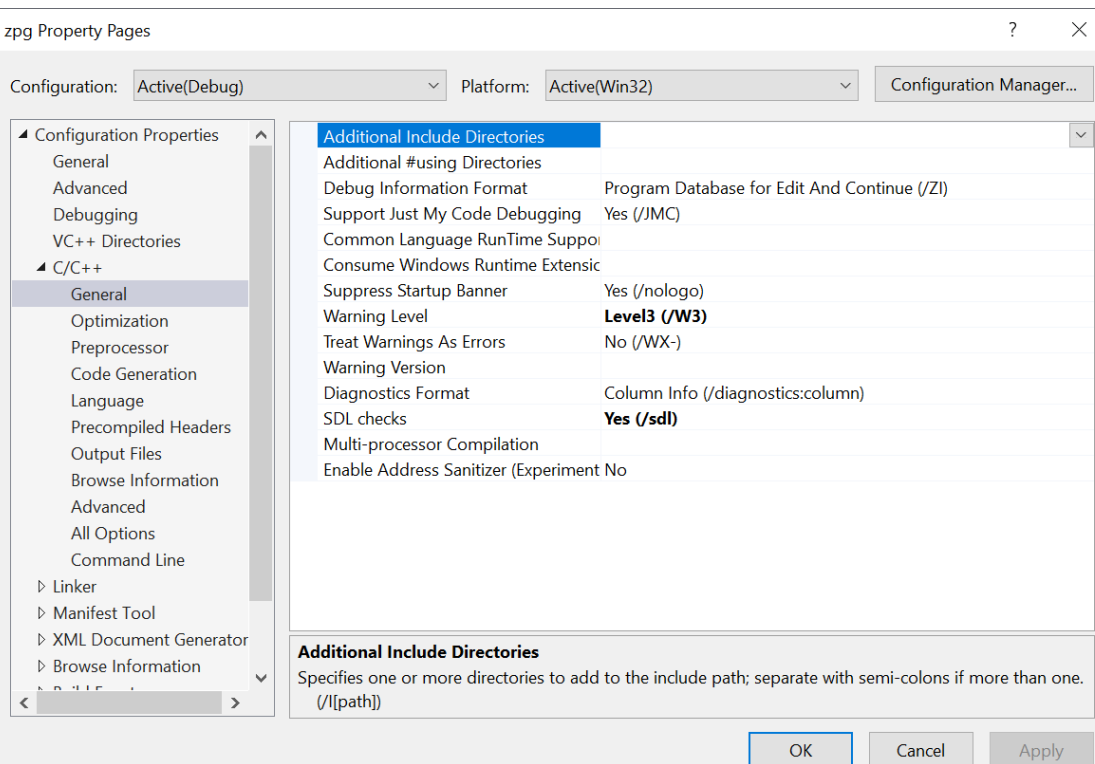
**32-bit Windows binaries**

Binaries for Visual C++ 2010 and plain MinGW are only available in the 32-bit

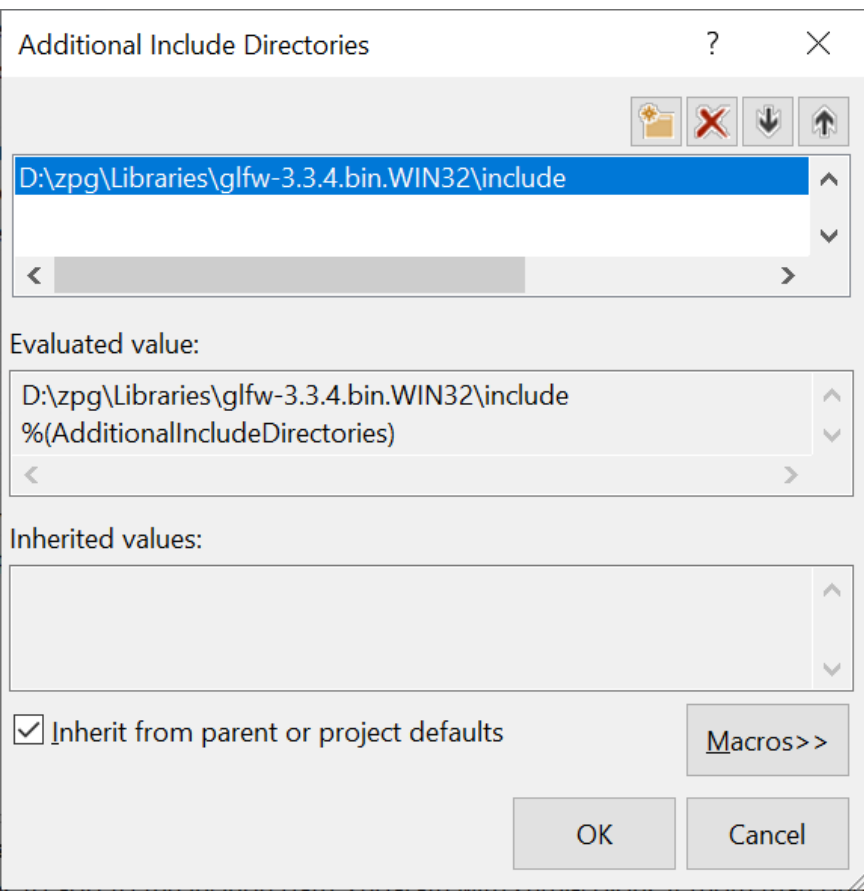


# GLFW

- Nastavení projektu



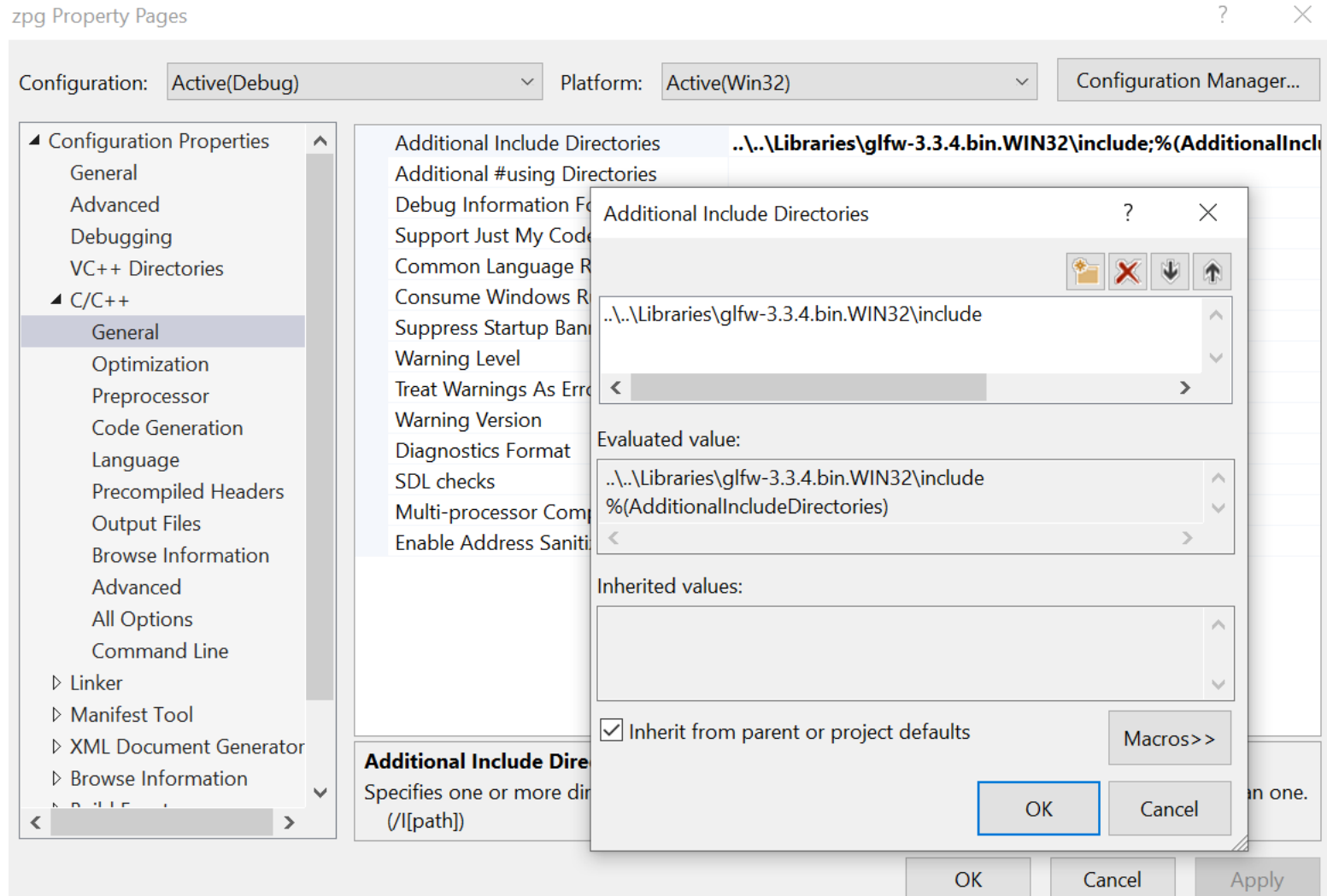
# Statická vs. dynamická část



```
1 //Include GLFW
2 #include <GLFW/glfw3.h>
3
4 //Include GLM
5 #include <glm/vec3.hpp> // glm::vec3
6 #include <glm/vec4.hpp> // glm::vec4
7 #include <glm/mat4x4.hpp> // glm::mat4
8 #include <glm/gtc/matrix_transform.hpp> // glm::tr
9 #include <glm/gtc/type_ptr.hpp> // glm::value_ptr
10
11 //Include the standard C++ headers
12 #include <stdlib.h>
13 #include <stdio.h>
```

```
1 //Include GLFW
2 #include <GLFW/glfw3.h>
3
4 //Include GLM
5 #include <glm/vec3.hpp> // glm::vec3
6 #include <glm/vec4.hpp> // glm::vec4
7 #include <glm/mat4x4.hpp> // glm::mat4
8 #include <glm/gtc/matrix_transform.hpp> // glm::tr
9 #include <glm/gtc/type_ptr.hpp> // glm::value_ptr
10
11 //Include the standard C++ headers
12 #include <stdlib.h>
13 #include <stdio.h>
14
```

# Relativní vs. absolutní cesta



# unresolved external symbol

The screenshot shows the Visual Studio IDE with a C++ project named 'zpg'. The main window displays the source code for 'Source.cpp', which includes GLM headers and a main function. The error list at the bottom shows a single error: 'LNK2019 unresolved external symbol \_imp\_glBegin@4 referenced in function \_main'. The error list is filtered by 'Entire Solution' and shows 30 errors, 0 warnings, and 0 messages. The error details indicate that the symbol is referenced in the function '\_main' in the file 'Source.obj' at line 1. The Solution Explorer on the right shows the project structure, including 'References', 'External Dependencies', 'Header Files', 'Resource Files', 'Source Files', and 'Source.cpp'. The Properties window at the bottom right shows the 'zpg' project properties, including the project name, dependencies, project file, and root namespace.

```
Source.cpp
zpg (Global Scope)

40
41
42
43 //GLM test
44
45 // Projection matrix : 45° Field of View, 4:3 ratio, display range : 0.1 unit <-> 100
46 glm::mat4 Projection = glm::perspective(45.0f, 4.0f / 3.0f, 0.01f, 100.0f);
47
48 // Camera matrix
49 glm::mat4 View = glm::lookAt(
50     glm::vec3(10, 10, 10), // Camera is at (4,3,-3), in World Space
51     glm::vec3(0, 0, 0), // and looks at the origin
52     glm::vec3(0, 1, 0) // Head is up (set to 0,-1,0 to look upside-down)
53 );
54 // Model matrix : an identity matrix (model will be at the origin)
55 glm::mat4 Model = glm::mat4(1.0f);
56
57
58 int main(void)
```

Error List

Entire Solution 30 Errors 0 Warnings 0 Messages Build + IntelliSense

Search Error List

Code	Description	Project	File	Line	Suppression State
LNK2019	unresolved external symbol _imp_glBegin@4 referenced in function _main	zpg	Source.obj	1	

Properties

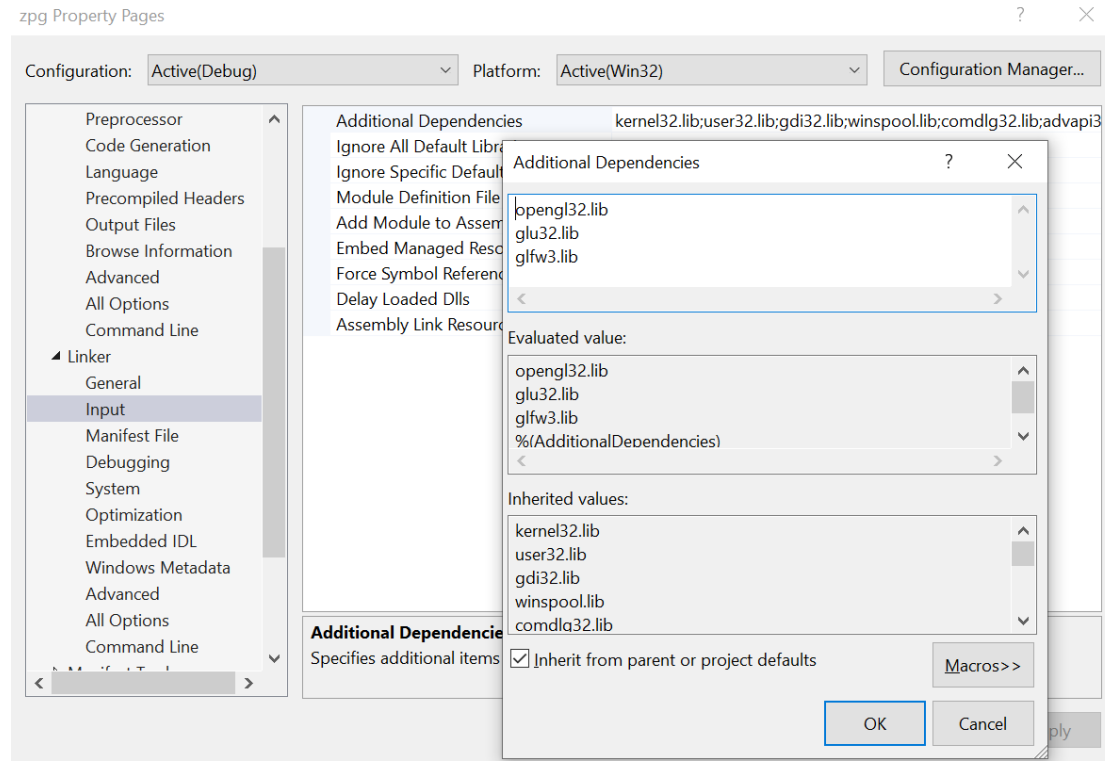
zpg Project Properties

Misc

(Name)	zpg
Project Dependencies	
Project File	D:\zpg\zpg\zpg.vcxproj
Root Namespace	zpg

(Name)  
Specifies the project name.

- Ve Windows mají statické knihovny obvykle příponu .lib (static library) a dynamické knihovny mají příponu .dll (dynamic linked library).



Configuration: Active(Debug)

Platform: Active(Win32)

Configuration Manager...

## Configuration Properties

General

Advanced

Debugging

VC++ Directories

C/C++

Linker

General

Input

Manifest File

Debugging

System

Optimization

Embedded IDL

Windows Metadata

Advanced

All Options

Command Line

Manifest Tool

XML Document Generator

Browse Information

Output File \$(OutDir)\$(TargetName)\$(TargetExt)

Show Progress Not Set

Version

Enable Incremental Linking **Yes (/INCREMENTAL)**

Suppress Startup Banner Yes (/NOLOGO)

Ignore Import Library No

Register Output No

Per-user Redirection No

Additional Library Directories **D:\zpg\Libraries\**

Link Library Dependencies Yes

Use Library Dependency Inputs No

Link Status

Prevent DLL Binding

Treat Linker Warning As Errors

Force File Output

Create Hot Patchable Image

Specify Section Attributes

## Additional Library Directories

Allows the user to override the environmental library path.

## Additional Library Directories

D:\zpg\Libraries\glfw-3.3.4.bin.WIN32\lib-vc2019

Evaluated value:

D:\zpg\Libraries\glfw-3.3.4.bin.WIN32\lib-vc2019  
%(AdditionalLibraryDirectories)

Inherited values:

☒ Inherit from parent or project defaults

Macros&gt;&gt;

OK

Cancel

OK

Cancel

Apply

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) zpg Sign in

Process: [11828] zpg.exe Lifecycle Events Thread: Stack Frame:

Source.cpp x zpg (Global Scope)

```
40
41
42 //GLM te cursor_callback
43 //GLM te cursor_callback
44 //GLM te cursor_callback
45 // Proj cursor_callback
46 glm::mat cursor_callback
47 // Camer cursor_callback
48 glm::mat cursor_callback
49
50 glm:
51 glm:
52 glm:
53 );
54 // Model
55 glm::mat
56
```

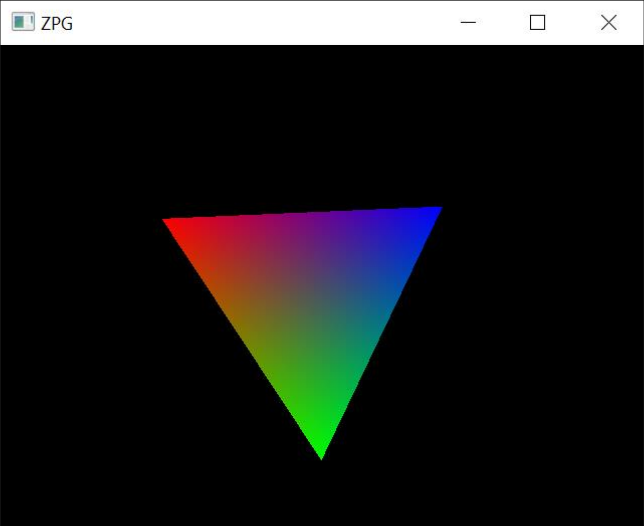
100 % No issues

Autos Search (Ctrl+E) Name

Autos Locals Watch 1

D:\zpg\zpg\Debug\zpg.exe

ZPG



Diagnostic Tools

6 seconds

10s

MB)

28

Memory Usage CPU Usage

of 0)

Lang

Window Output

# Volitelné úkoly na cvičení

- Seznamte se s kódem, podívejte se na základní příkazy, najděte si je v dokumentaci.
- Vytvořte čtverec, kde poslední vrchol bude mít žlutou barvu.
- Přidejte možnost ovládání rotace (směr, rychlost atd.)