

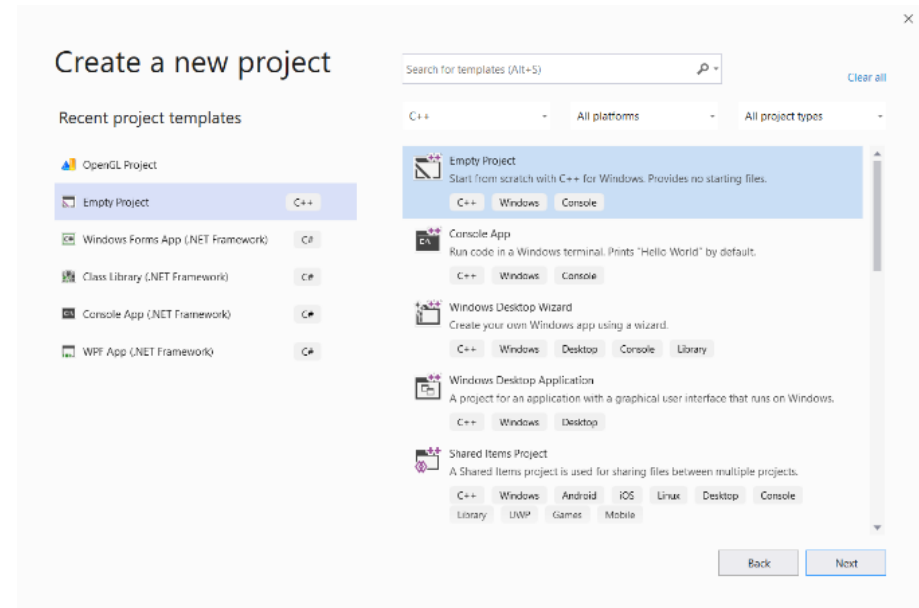
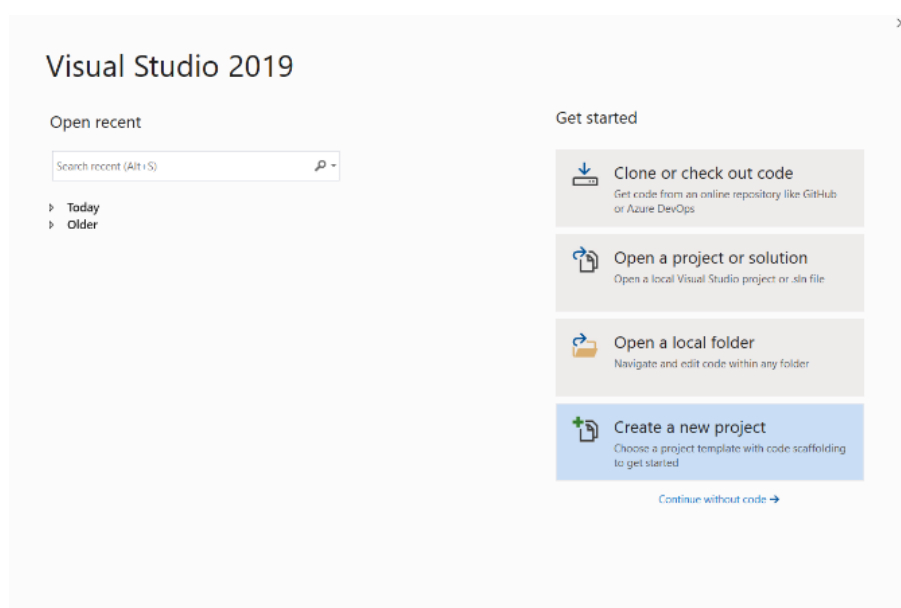
# SZÁMÍTÓGÉPES GRAFIKA

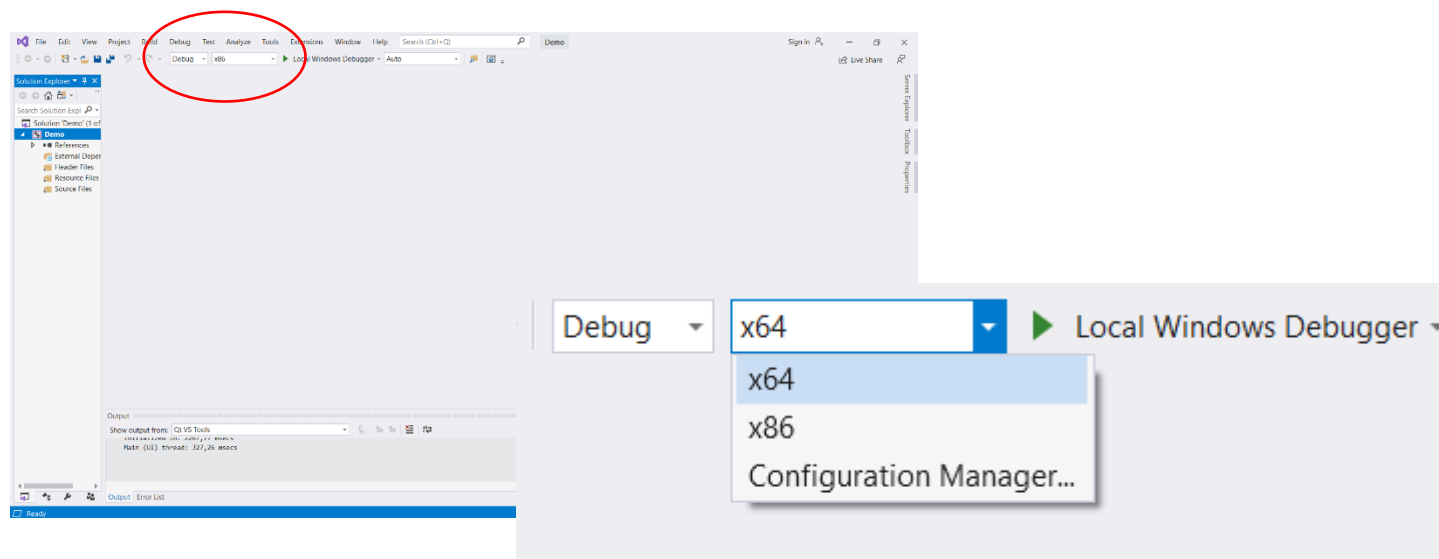
(INBMM0635-17)

2019/20/2. félév

## VISUAL STUDIO 2019 PROJECT TEMPLATE KÉSZÍTÉSE

1. Töltjük le az *OpenGLTemplate.zip* állományt a <https://drive.google.com/open?id=1c69WHreb3eoXBVXqSLeaXV0jcHFCZaU4> címről.
2. Tömörítsük ki a letöltött állományt. A letöltött mappa tartalmaz egy *include* és egy *lib* nevű könyvtárat, valamint a *glew32.dll* fájlt.
3. Indítsuk el a *Visual Studio 2019* nevű programot.
  - 3.1. Hozzunk létre egy új, üres C++ projektet.

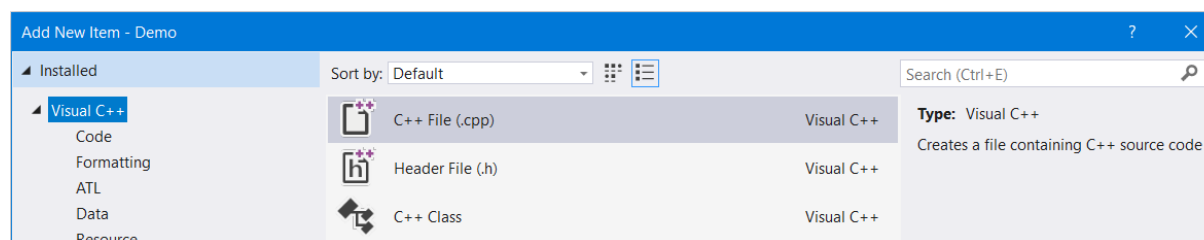




3.2. A projekt létrehozása után felül, a menüsor alatt több legördülő menüt láthatunk. A jobb oldali lehetővé teszi az **x86** és **x64** értékek közötti váltást. Válasszuk ki az **x64** értéket!

3.3. A bal oldali menüben a **Debug** és **Release** értékek között váltogathatunk. Elsőként állítsuk be a **Debug** értéket!

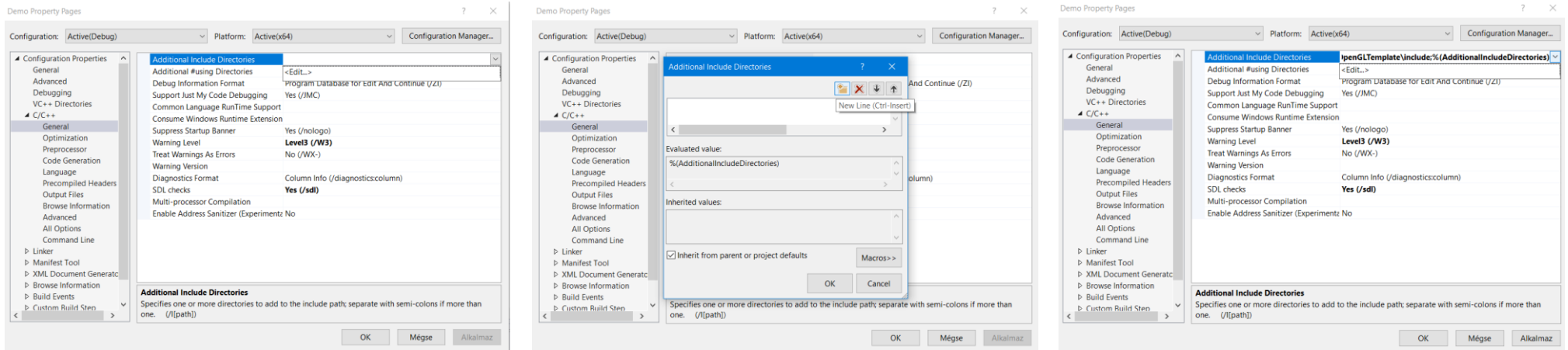
3.3.1. Debug módban keressük meg a **Solution explorer** ablakot (a fenti ábrán bal oldalon), majd kattintsunk a projektünk nevére a jobb egérgombbal és válasszuk ki az **Add / New Item...** menüpontot. Az új elem legyen egy C++ File (Az alapértelmezett Source.cpp fájlnevet nem szükséges módosítani).



3.3.2. Ismét a Solution explorer ablakban a projekt nevére kell kattintanunk a jobb egérgombbal, ekkor a legalsó,

**Properties** menüpontot válasszuk ki!

3.3.3. A bal oldali listából a **C/C++** elemhez tartozó **General** kiválasztását követően az **Additional Include Directories** értékének állítsuk be az 1. lépésben letöltött könyvtár *include* mappáját.

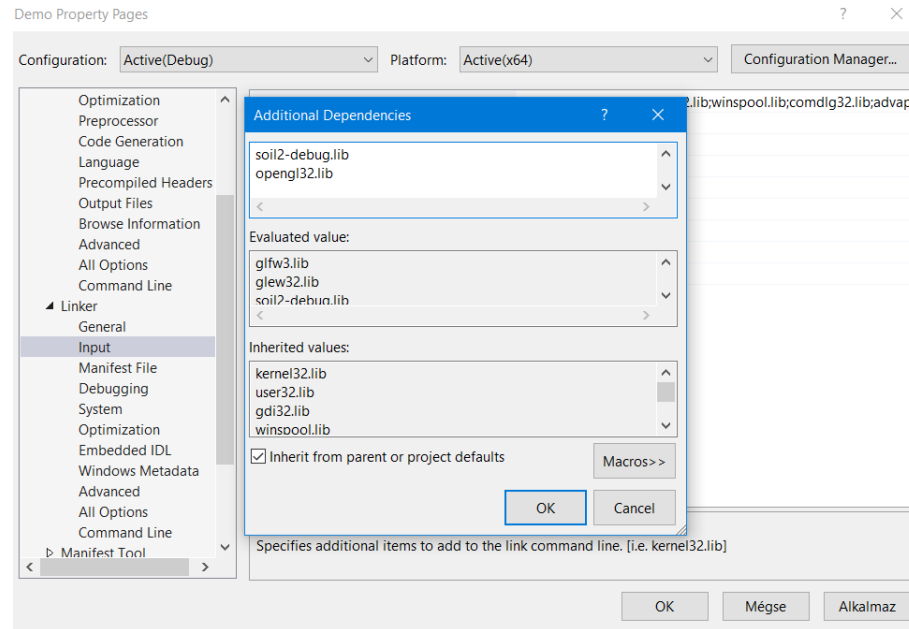


3.3.4. Kattintsunk C/C++ alatti **Linker** elemre, majd a **General / Additional Library Directories** értékének állítsuk be az 1. lépésben letöltött könyvtár *lib* mappáját.

3.3.5. Szintén a Linker-nél válasszuk ki az **Input** elemet. Itt az **Additional Dependencies** listához kell új értékeket hozzáadnunk: a szerkesztés itt is az <Edit> kiválasztásával lehetséges.

3.3.6. A felugró ablak felső, üres szövegmezőjébe a következő állományneveket kell felsorolnunk, egymás alá:

- **glfw3.lib**
- **glew32.lib**
- **soil2-debug.lib**
- **opengl32.lib**



3.4. Az OK gombbal zárjuk be a Properties ablakot!

3.5.A Debug módot követően válasszuk ki a **Release** módot a fenti, legördülő menüből.

3.5.1. A 3.3.2-ben leírtak szerint nyissuk meg ismét a Properties ablakot (ezúttal Release konfiguráció mellett)

3.5.2. Ismételjük meg a 3.3.3, 3.3.4, 3.3.5 és a 3.3.6 lépéseket, majd a 3.4-es lépést!

3.6. Exportáljuk az elkészült template-t: a menüsorból válasszuk ki a **Project / Export template...** menüpontot.

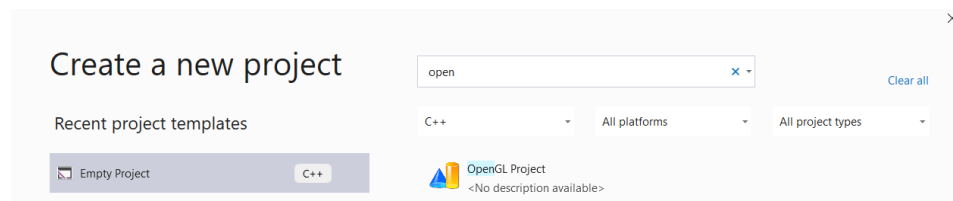
3.6.1. A Template type maradjon az alapértelmezett Project template értéken. Next gomb.

3.6.2. Adjunk valamilyen beszédes nevet a template számára, például **OpenGL Project**. Finish.

(Az elkészült template a Dokumentumok / Visual Studio 2019 / My Exported Templates mappába kerül zip állományként.)

## OPENGL PROJEKT LÉTREHOZÁSA

1. A Visual Studio-ban hozzunk létre egy új **OpenGL Project** típusú projektet: **File / New Project...**



2. Keressük meg ismét a **Solution explorer** ablakot, majd a projektünk Source files mappáját. Az itt található Source.cpp (a név eltérhet, ha az előző rész 3.3.1. pontjában más nevet adtunk az állománynak) tartalma egyezzen meg az alábbi kóddal:  
<https://drive.google.com/open?id=1Ktdq-lg1AAQOK6crjGXR3fK6zaoH9E00>
3. Ezután Debug módban (és x64 szerepeljen a másik legördülő menüben!) kattintsunk a **Local Windows Debugger** gombra (szintén a menüsor alatt van). Elsőre hibát tapasztalunk, viszont a projektünk mappájában ezzel létrehoztunk egy Debug nevű mappát, amelyre hamarosan szükségünk lesz.
4. Ismételjük meg a fenti lépéseket Release módban is (plusz x64 változatlanul). Itt is hibát tapasztalhatunk, de csak a Release mappa generálása fontos számunkra jelenleg.
5. Kattintsunk jobb gombbal a projekt nevére a **Solution explorer** ablakban, majd **Open Folder in File Explorer**. Itt található a C++ állományunk és később ide kerülnek majd az úgynevezett shader fájlok is. Ha egy szinttel feljebb lépünk a mappa hierarchiában, láthatunk egy x64 nevű mappát, amelyben megtekinthetjük a 3. és 4. lépésekben generálódott Debug és Release mappákat! Mindkettő mappába másoljuk bele a munkánk legelején letöltött **glew32.dll** állományt!
6. Visszatérve a Visual Studio-ba (Debug és x64 mellett) ismét megnyomhatjuk a **Local Windows Debugger** gombot. Ekkor az első grafikus alkalmazásunknak sikeresen el kellene indulnia!

