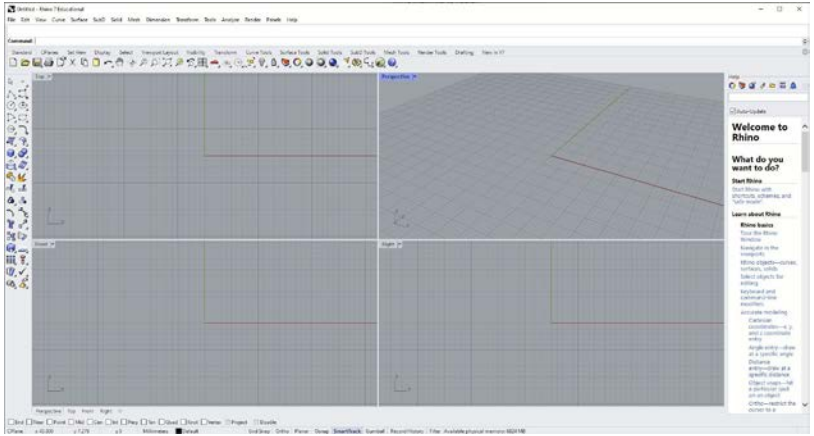
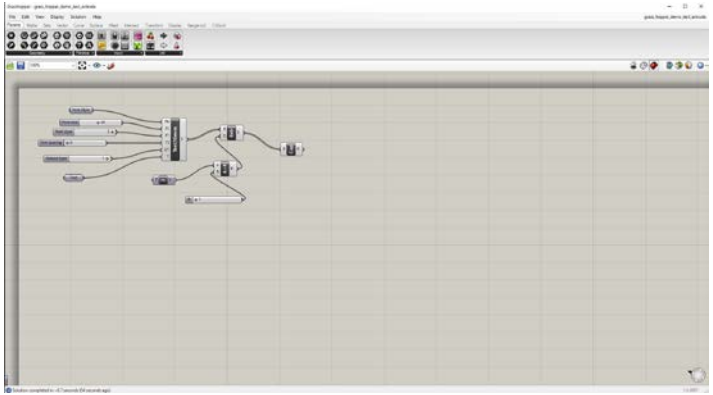
Basics

Rhinoceros (typically abbreviated **Rhino** or **Rhino3D**) is a commercial 3D computer graphics and computer-aided design (CAD) application software. Rhinoceros geometry is based on the NURBS mathematical model, which focuses on producing mathematically precise representation of curves and freeform surfaces in computer graphics (as opposed to polygon mesh-based applications). (source: https://en.wikipedia.org/wiki/Rhinoceros_3D)



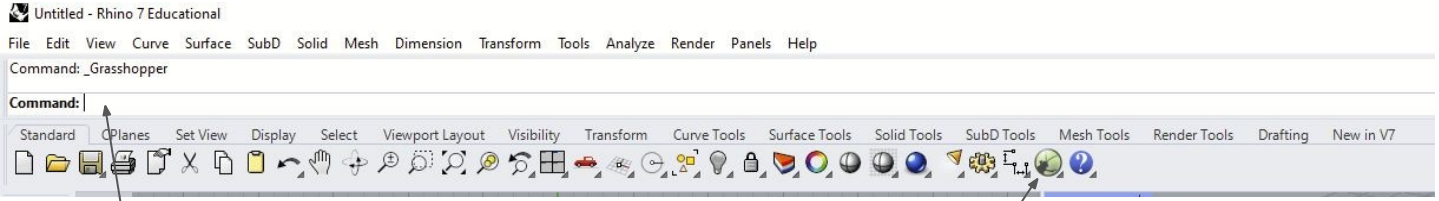
Main Window of Rhino 7

Grasshopper is a visual programming language and environment that runs within the Rhinoceros 3D computer-aided design (CAD) application. (source: https://en.wikipedia.org/wiki/Grasshopper_3D)



Main Window of Grasshopper

How do I execute Grasshopper?



click the Grasshopper icon

or type in the Rhino command line: `Grasshopper`

Requirements and the starter

- Rhino 8 64 bits for Windows
- Visual Studio 2019 with C# and .NET 4.8 (it is enough to use the community version, see here)

The skeleton code can be found in the course github repo

Name	Date modified	Type	Size
CreateSphereGrasshopper	05/06/2024 18:38	File folder	
Examples	05/06/2024 18:28	File folder	
Text2GeomViaIrit2Grasshopper	05/06/2024 18:28	File folder	

Where do I find help and the gear?

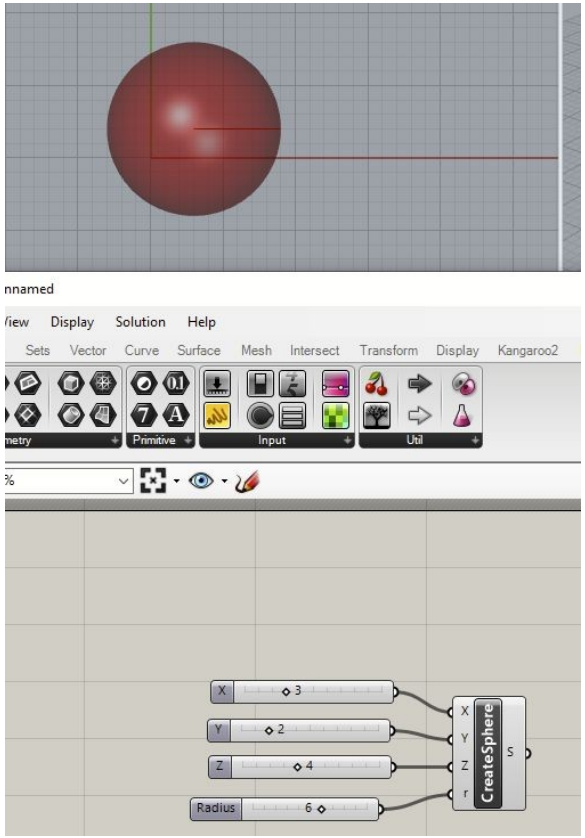
- Rhino Common API: https://developer.rhino3d.com/api/RhinoCommon/html/R_Project_RhinoCommon.htm
- Grasshopper SDK: https://developer.rhino3d.com/api/grasshopper
- Extra plugins (needs approval): https://www.food4rhino.com

`CreateSphereGrasshopperInfo` a.k.a. the basic plug-in

Defines four inputs: `X`, `Y`, `Z` coordinates of a sphere, and its `Radius`

Defines one output: BREP surface

<Interactive demo with code walk around>



The code of CreateSphereGrasshopperInfo

The main logic is located in: `CreateSphereGrasshopperComponent.cs`

The input interface is defined by overriding the method: `RegisterInputParams`

The output interface is defined by overriding the method: `RegisterOutputParams`

Finally the computing happens in the overridden method: `SolveInstance`

How to use C++ code in plugins?

We see now the plugin: `Text2GeomViaIrit2Grasshopper`

<Interactive demo with code walk around>

Font (any valid font name, e.g. Arial, Times New Roman)

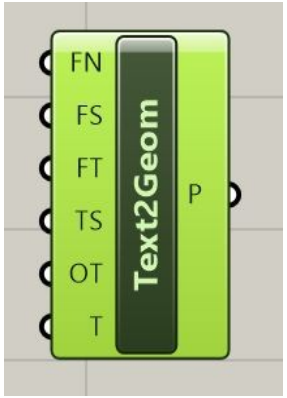
Font size

Font style: 0 - regular, 1 - italic, 2 - bold, 3 - bold-italic

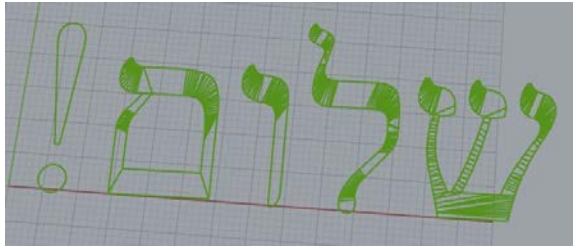
Text spacing

Output type: 0 - polygons, 1 - outline

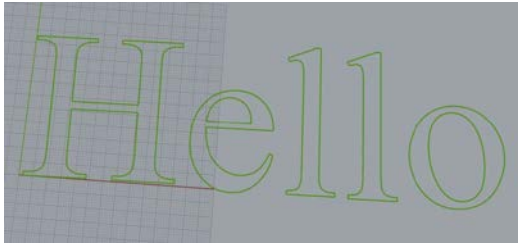
Text to convert



Output: List of polygons



Default output produced by the plugin



Outline output

The text conversion is done via IRIT library (C++), and the C# code only passes the input to it, and then performs the output conversion to the format specified by the Rhino and Grasshopper API.

How to use C++ code in plugins?

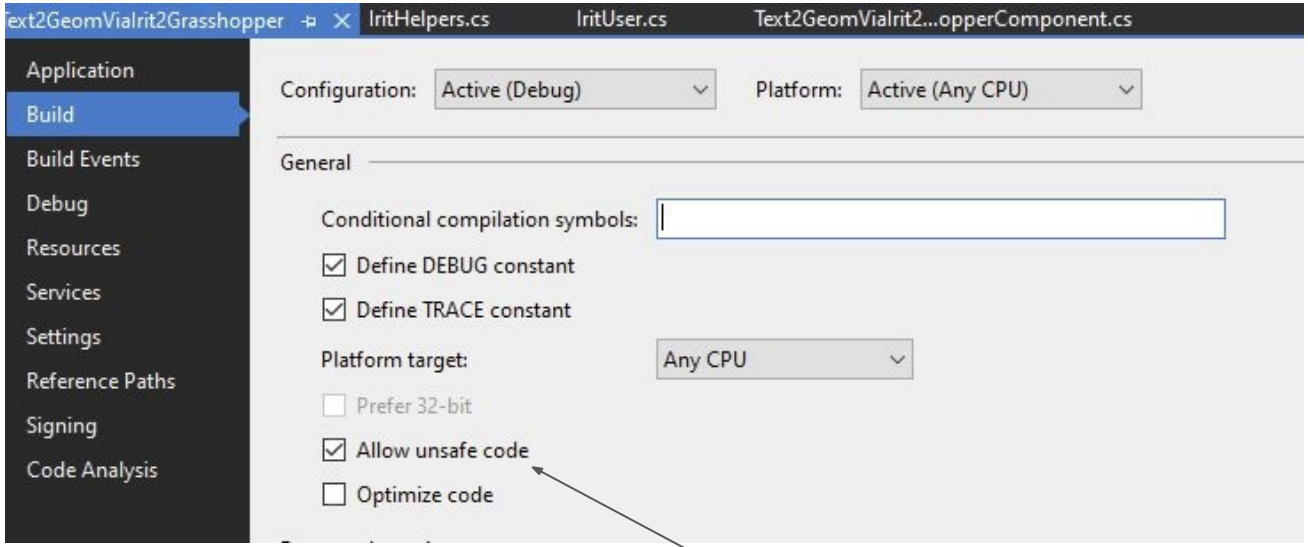
Use P/Invoke mechanise. See: https://docs.microsoft.com/en-us/dotnet/standard/native-interop/pinvoke

The method `SolveInstance` is using `IritNet.dll` library and classes defined in `IritHelpers.cs` and `IritUser.cs` in order to interact with `Irit.dll`. The library `IritNet.dll` does not provide a complete set of bindings for `Irit.dll`. Therefore some types and functions have to be additionally defined (`IritHelpers.cs` and `IritUser.cs`). Keep in mind that it is necessary to determine in `C#` the types of arguments for the functions you want to use. **Some libraries may define very complex types, which may be challenging to re-implement. Therefore, you should first check whether:**

- **Library provides bindings for `C#`**
- **The argument types are built-in or straightforward enough so you can re-implement them**

Troubleshooting

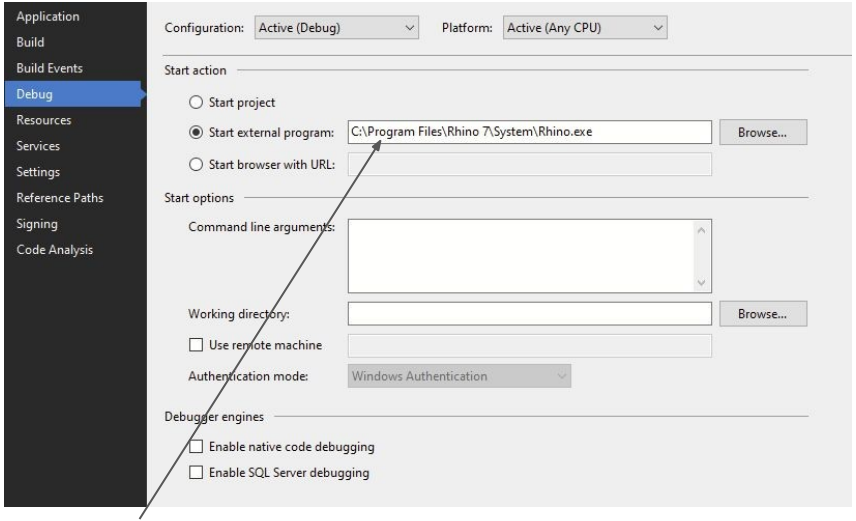
C++ library that uses pointers for functions' arguments



This has to be checked

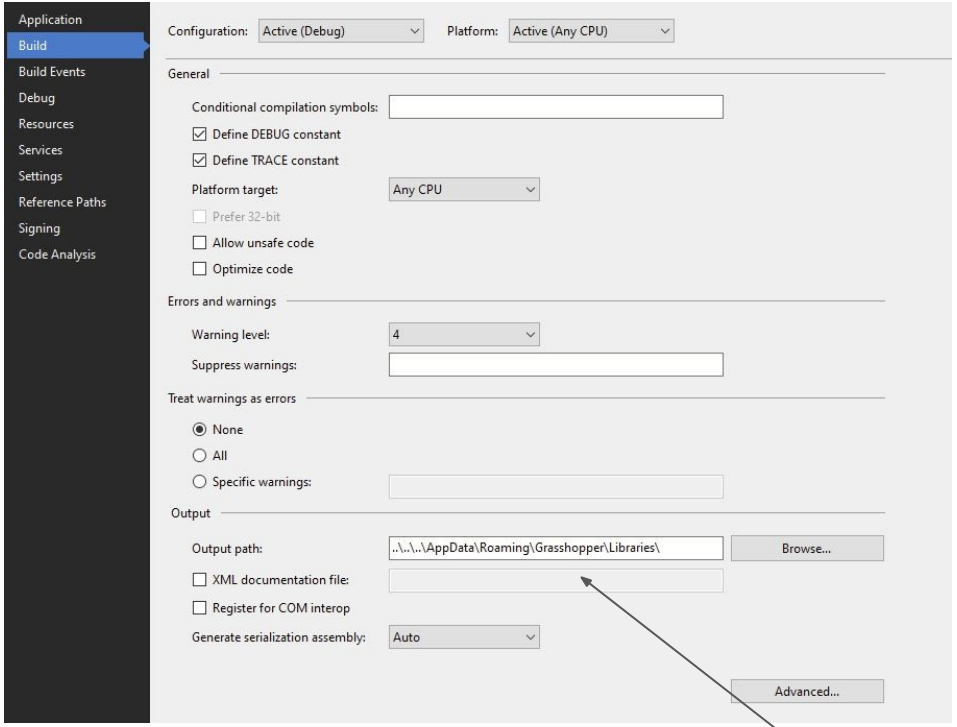
Troubleshooting

Integration with Rhino



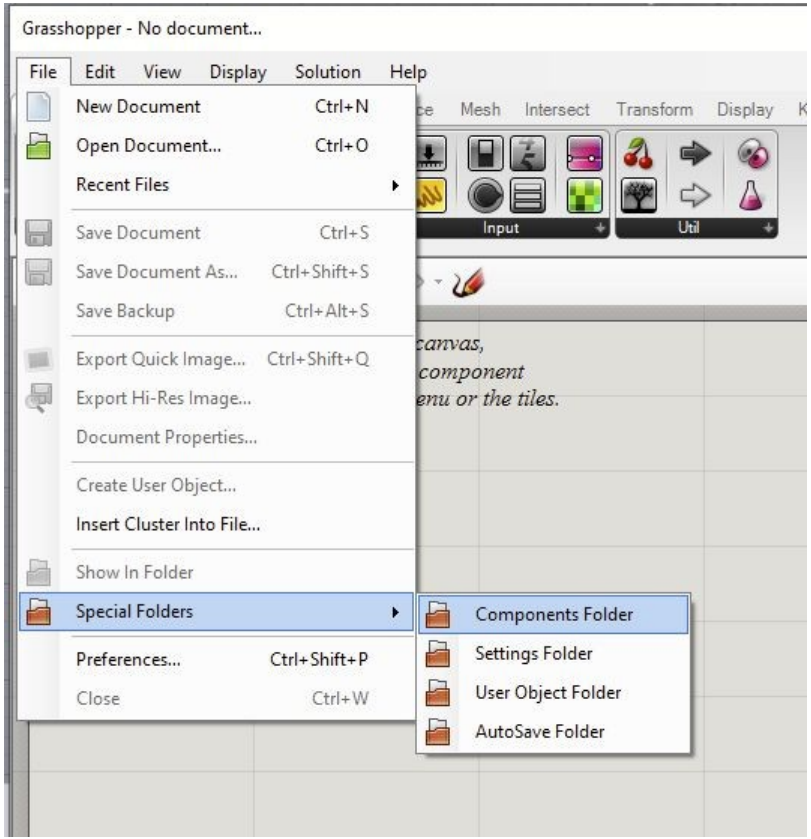
Set it to the Rhino.exe file

The plugin is not placed in the Grasshopper components



To find the Components Folder on your machine see the next slide

Where is the Grasshopper Components Folder?



Once clicked a new `Explorer` window will open displaying the component folder (you need to copy the path and set it in Visual Studio, see the previous slide)

Troubleshooting

I installed a Grasshopper plugin but it does not work

First, to install plugins you need to place them in the `Components Folder`. A plugin can be blocked for security reasons. To unlock it, go to the file properties and and click **unblock. You will need to do this for each file used by a plugin.**

See: https://wiki.mcneel.com/rhino/unblockplugin
and https://www.giancadm.com/plugins-install/