

Double-layer Timber Plate Shell

Tutorial Rhino3D and Grasshopper Part. 1

Petras Vestartas, Ph.D.

take fabrication intersection type remove shown tenon important within meaning open folder kind another used list information timber
panel angle group along bake simple ready scale mesh shell several project top work force arch part display change overall
edge hole design three plate structure place represent defined box component cut model input start even timber shell example file move set applied
first surface test introduce plugin line position form made process connected tool right run additional whole without new code pair assembly sequence boxes order know
case joint question multiple associated essentially given data structural definition slide TBOTS click result macro model shape material term plane construction insertion vector
plate timber plate wood connection direction wood wood

[Search MOOC](#)



[Video](#)



compas_wood - Installation

ibois-epfl.github.io/compas_wood/latest/installation.html

compas_wood 0.1.1

Search docs

Introduction

Description

Installation

Examples

API Reference

License

Citation

Windows

```
conda create -n wood-dev python=3.8 mpir mpfr boost-cpp eigen=3.3 cgal-cpp=5.2 pybind11 compas compas_wood
conda activate wood-dev
git clone https://github.com/petrasvestartas/compas_wood
cd compas_wood
pip install -e .
```

Mac

```
conda create -n wood-dev python=3.8 gmp mpfr boost-cpp eigen=3.3 cgal-cpp=5.2 pybind11 compas compas_wood
conda activate wood-dev
git clone https://github.com/petrasvestartas/compas_wood
cd compas_wood
pip install -e .
```

Rhino Grasshopper

Download the (Windows) or (Mac) zipped files and place them in libraries folder.
You can find this folder when you open Grasshopper.
File->Special Folder->Component Folder.
https://github.com/ibois-epfl/compas_wood/releases/tag/compas_wood_GH_1.0.0

Installation

- Windows
- Mac
- Rhino Grasshopper

Type here to search

6°C Ciel couvert 9:27 AM 2/24/2022

We will start the practical assignment for Rhino and Grasshopper and we will start from installation of the Grasshopper plugin. If you click the following link, we will get into the documentation of the code and we need to go to this GitHub page in order to download the libraries.

Notes

Summary



compas_wood - Installation x Release compas_wood_GH_1.0.0 x

github.com/ibois-epfl/compas_wood/releases/tag/compas_wood_GH_1.0.0

Search or jump to... Pull requests Issues Marketplace Explore

ibois-epfl / compas_wood Public

<> Code Issues Pull requests

Releases / compas_wood_GH_1.0.0

compas_wood

petrasvestartas released this

compas_wood Rhino Grasshopper

Assets 4

- compas_wood_release_0
- compas_wood_release_0
- Source code (zip)
- Source code (tar.gz)

hexagona_shell.3dm side_to_top.3dm 190511_VDA_Works hop_FabricationNE... Example1_3dm.3dm

Rhino is loading plug-in: Displacement

News:

- Topo Deep Dive workshop for topographic model
- Rhino.Inside.Revit Online Course, March 28-30 (M)
- Darco Rhino Basics - Online Workshop (Spanish)
- Rhino-CVC - New object library for HVAC
- Form Finding with Kangaroo Physics Webinar
- UBC Design League Designathon, February 19-20
- New version of Conveyor Rhino and Revit
- Rhino.Inside cadwork
- Open Sky/Oasis (by ARCHIOLOGICS)

Tips:

- Have you tried RefitTrim?
- SubD: Create smooth, seamless surfaces
- Have you tried MultiPipe?
- Don't want to write code to create new Rhino co
- Have you tried RibbonOffset?
- Have you tried PBR materials?
- Have you tried Named Selections?
- Have you tried EdgeContinuity?
- Have you tried Bloom?

Logged in as Petras Vestartas - petrasvestartas@gmail.com Version 7 SR12 (7.12.21313.6341, 2021-11-09) Educational Lab License

Rhinoceros® design, model, present, analyze, realize...

Copyright 1993-2021 Robert McNeel & Associates

1.47 MB 1.37 MB

Show all

compas_wood_rele...zip compas_wood_rele...zip

Type here to search

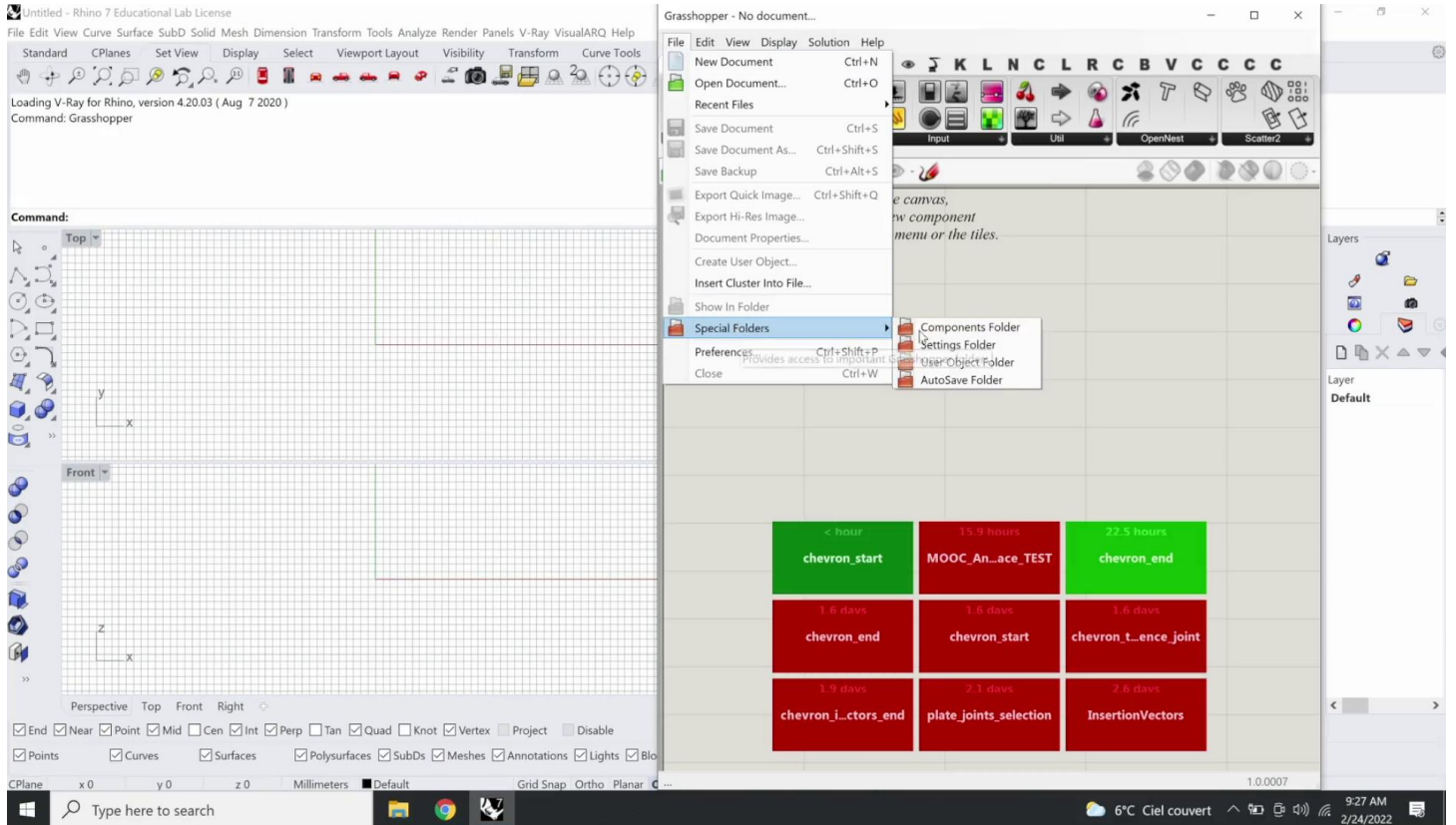
6°C Ciel couvert 9:27 AM 2/24/2022

This link goes directly to the release file and you can download the zip folder, which is the library file and then example files that we are going to use today. I'm downloading the first one and then I'm also going to download example files. Now, we can open Rhino.

Notes

Summary



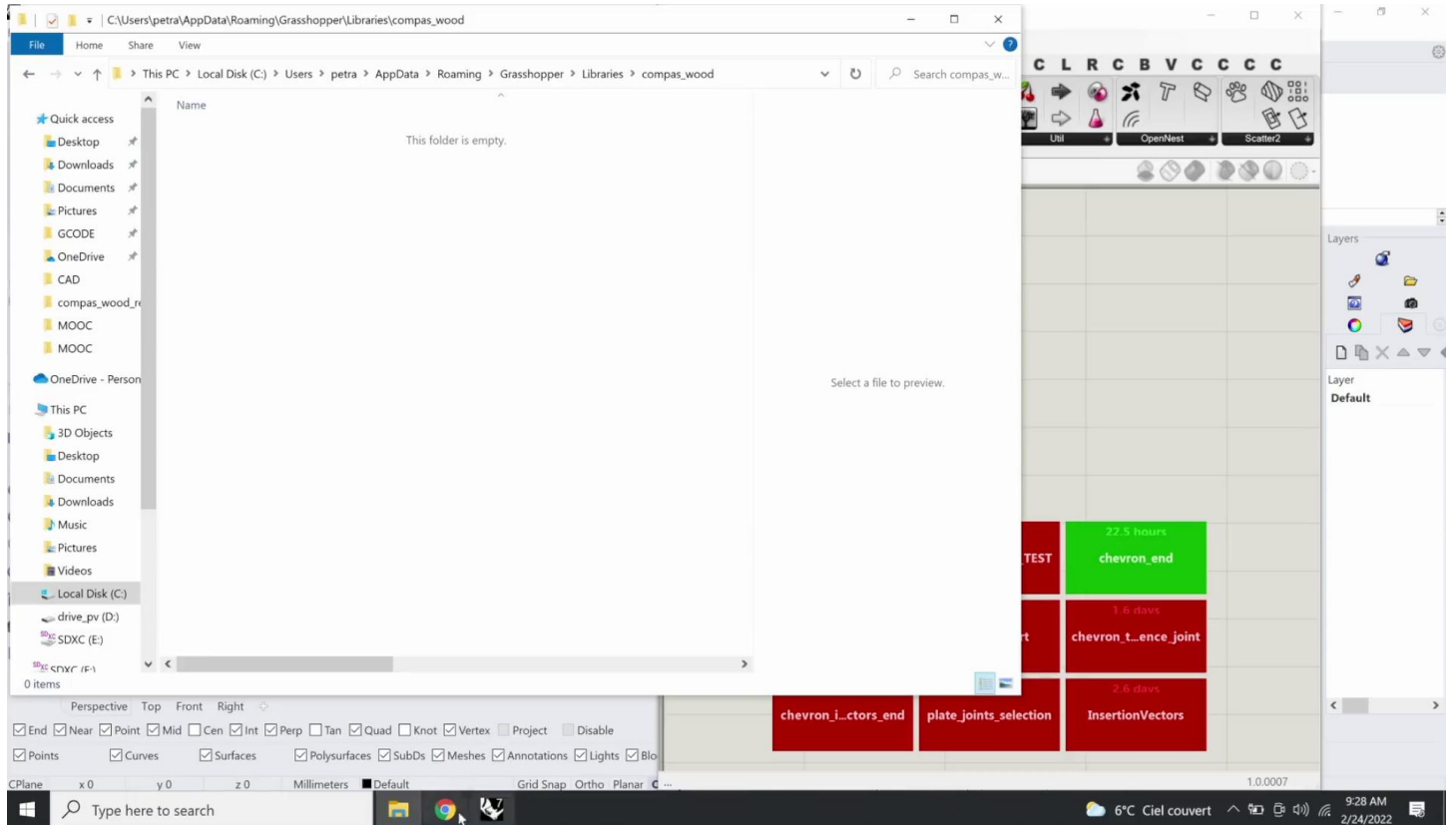


We also open Grasshopper.

Notes

Summary





If you go to file, special folders, components folder, we can create a new folder here called, compas_wood and place the necessary files here.

Notes

Summary



compas_wood - Installation

Release compas_wood_GH_1.0.0

github.com/ibois-epfl/compas_wood/releases/tag/c

Search or jump to... Pull requests

ibois-epfl / compas_wood Public

<> Code Issues Pull requests Actions

Releases / compas_wood_GH_1.0.0

compas_wood_GH_1.0.0

petrasvestartas released this 7 days ago

compas_wood Rhino Grasshopper Release

▼ Assets 4

compas_wood_release_0_1_1.zip

1.47 MB

compas_wood_release_0_1_1_examples.zip

1.37 MB

Source code (zip)

Source code (tar.gz)

compas_wood_release_0_1_1.zip (Nicht lizenzierte Testversion)

Datei Befehle Extras Favoriten Optionen Hilfe

Hinzufügen Entp. nach Testen Anzeigen Löschen Suchen Assistent Info Virenprüfung Kommentar SFX

compas_wood_release_0_1_1.zip/compas_wood - ZIP Archiv, ungepackte Größe 2,023,424 Bytes

Name	Größe	Gepackt	Typ	Geändert	CRC32
joinery_solver_gh.gha	1,293,312	1,212,075	Grasshopper Assem...	2/23/2022 10:31...	182C0B5C
joinery_solver_net.dll	12,288	5,483	Application extension	2/23/2022 10:31...	8140F8C3
joinery_solver_pinvoke.dll	458,752	212,906	Application extension	2/23/2022 10:31...	AD008A68
rhino_util.dll	259,072	106,290	Application extension	2/23/2022 10:31...	057493FD

Ausgewählt sind 4 Dateien, 2,023,424 Bytes

Insgesamt 4 Dateien, 2,023,424 Bytes

compas_wood_rele...zip

compas_wood_rele...zip

Type here to search

6°C Ciel couvert 9:28 AM 2/24/2022

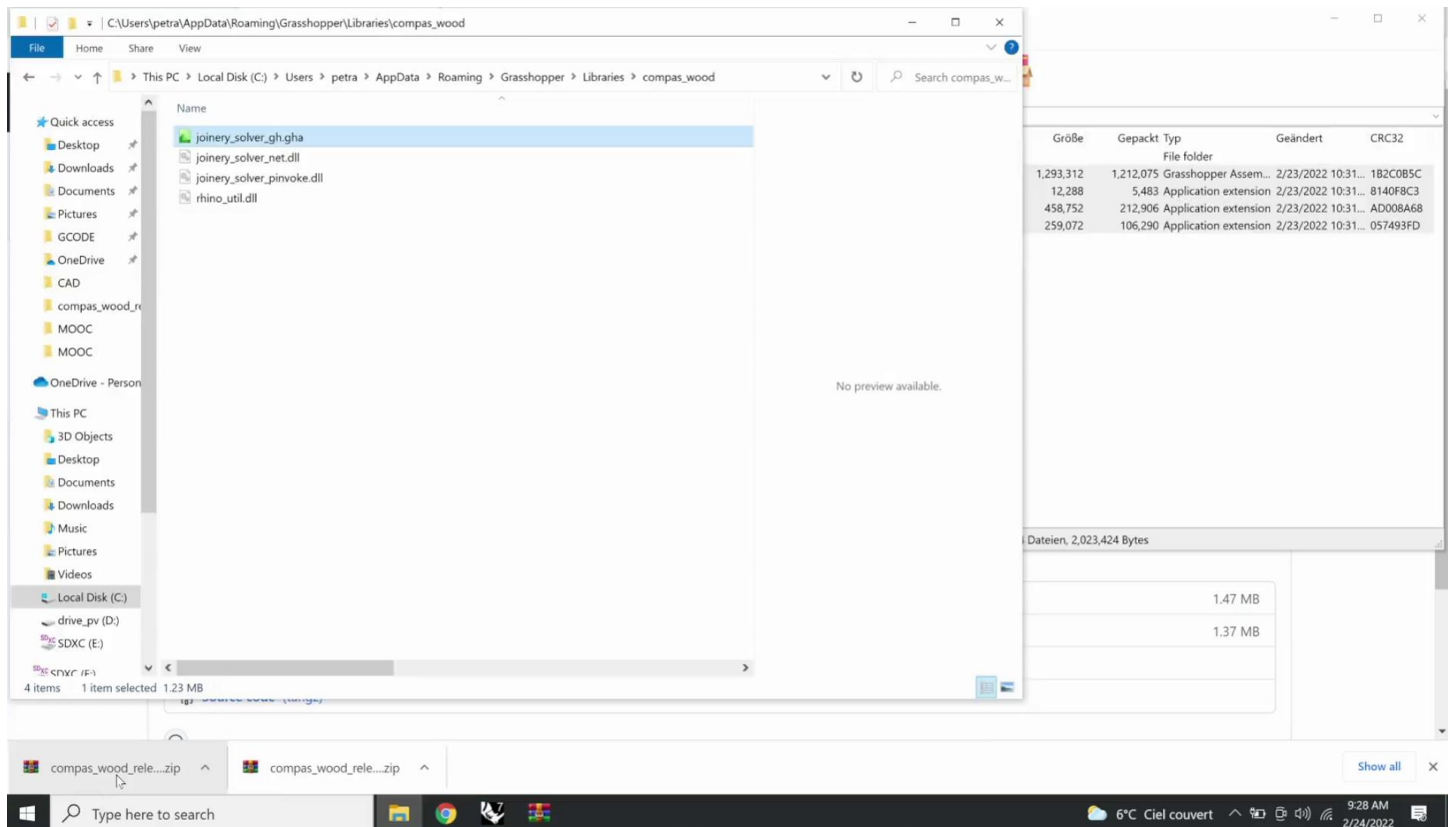
File unzip, we have a folder with all these files.

Notes

Summary



1m 14s

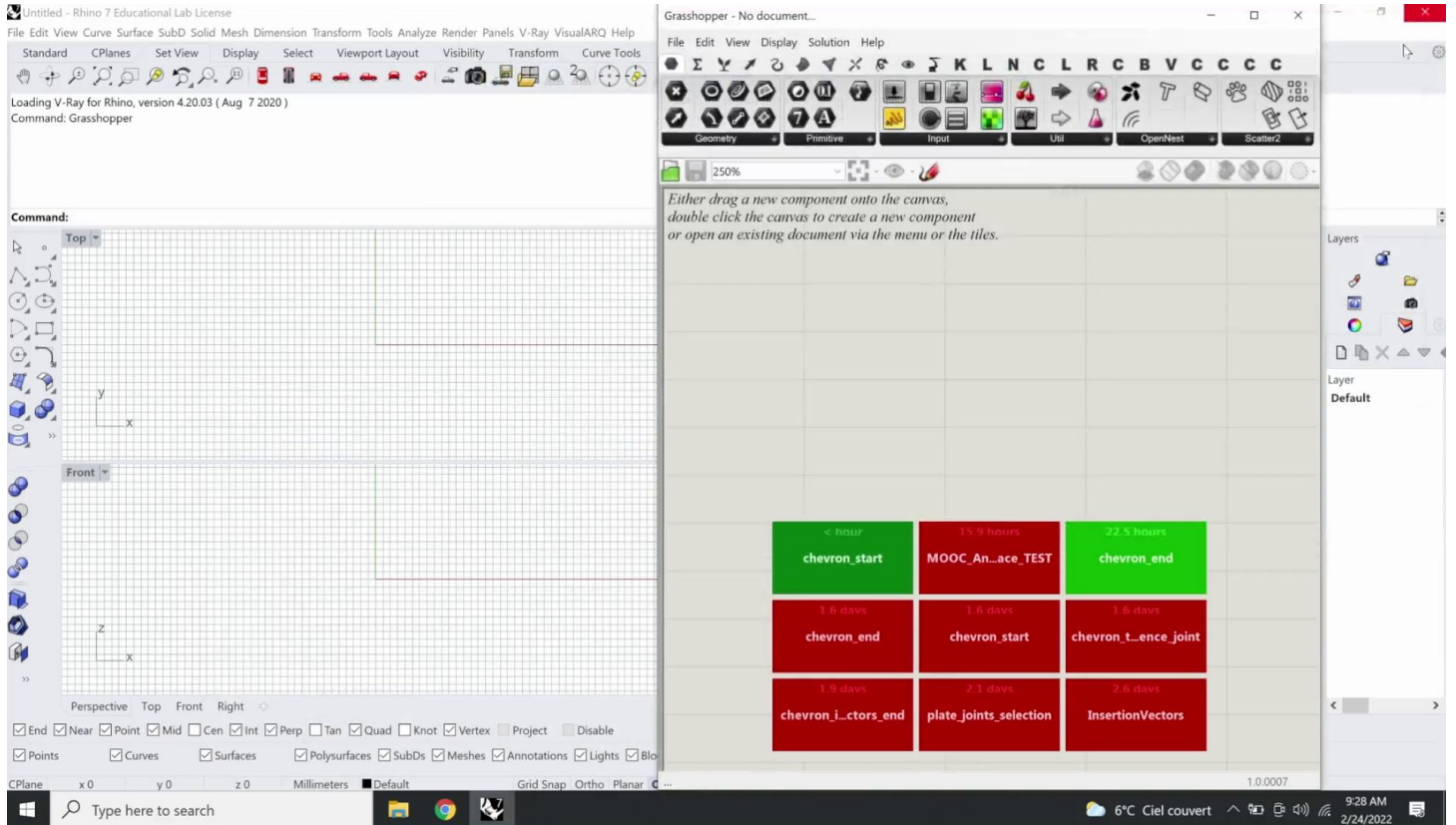


Simply drag and drop and check if the files are safe and blocked. In my case, I don't have any button here which means it's fine and we are ready to go.

Notes

Summary



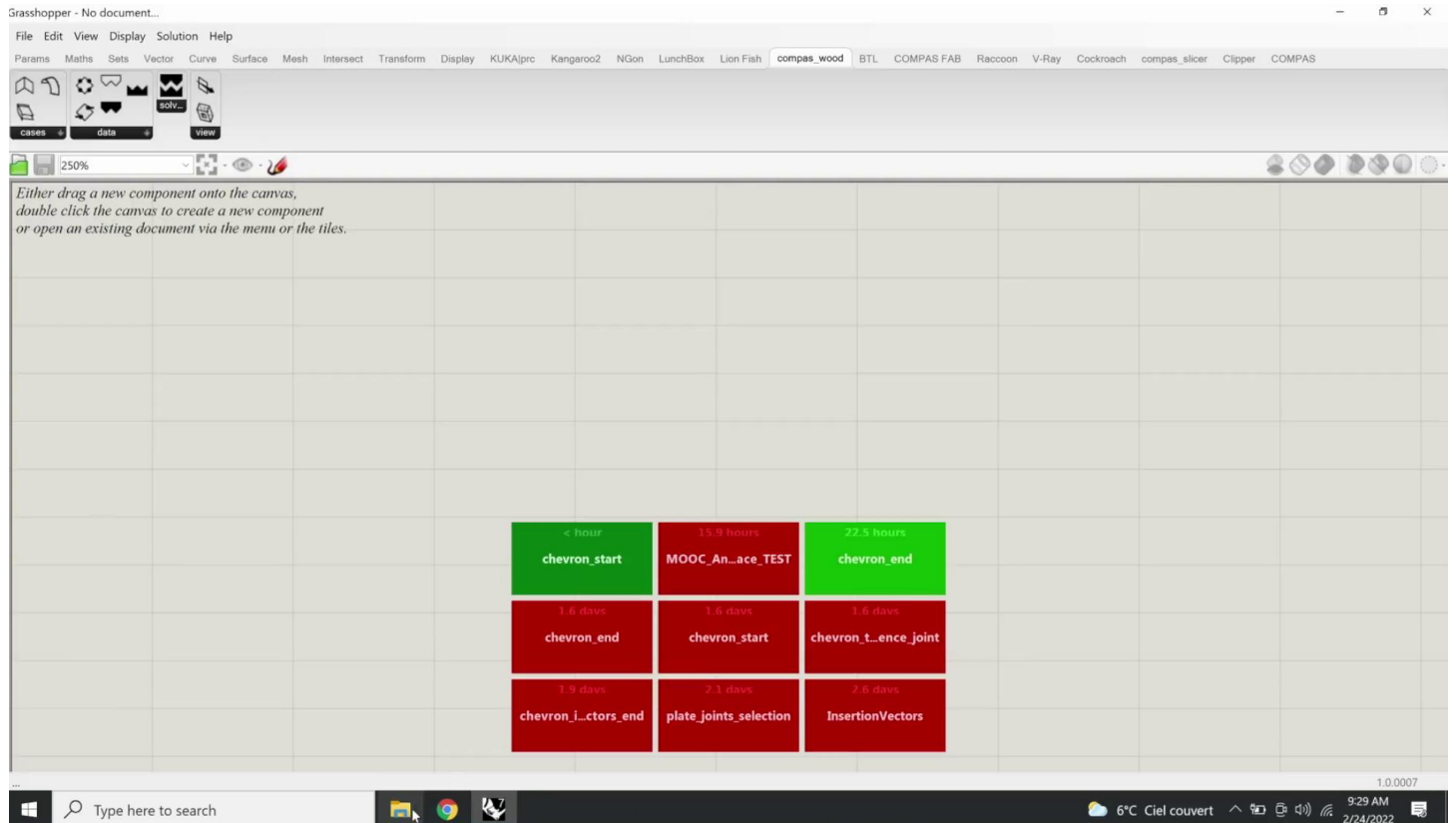


Also can unzip the example files which will be the use of today. I'll place all of these in the same folder and we can close and restart Rhino.

Notes

Summary





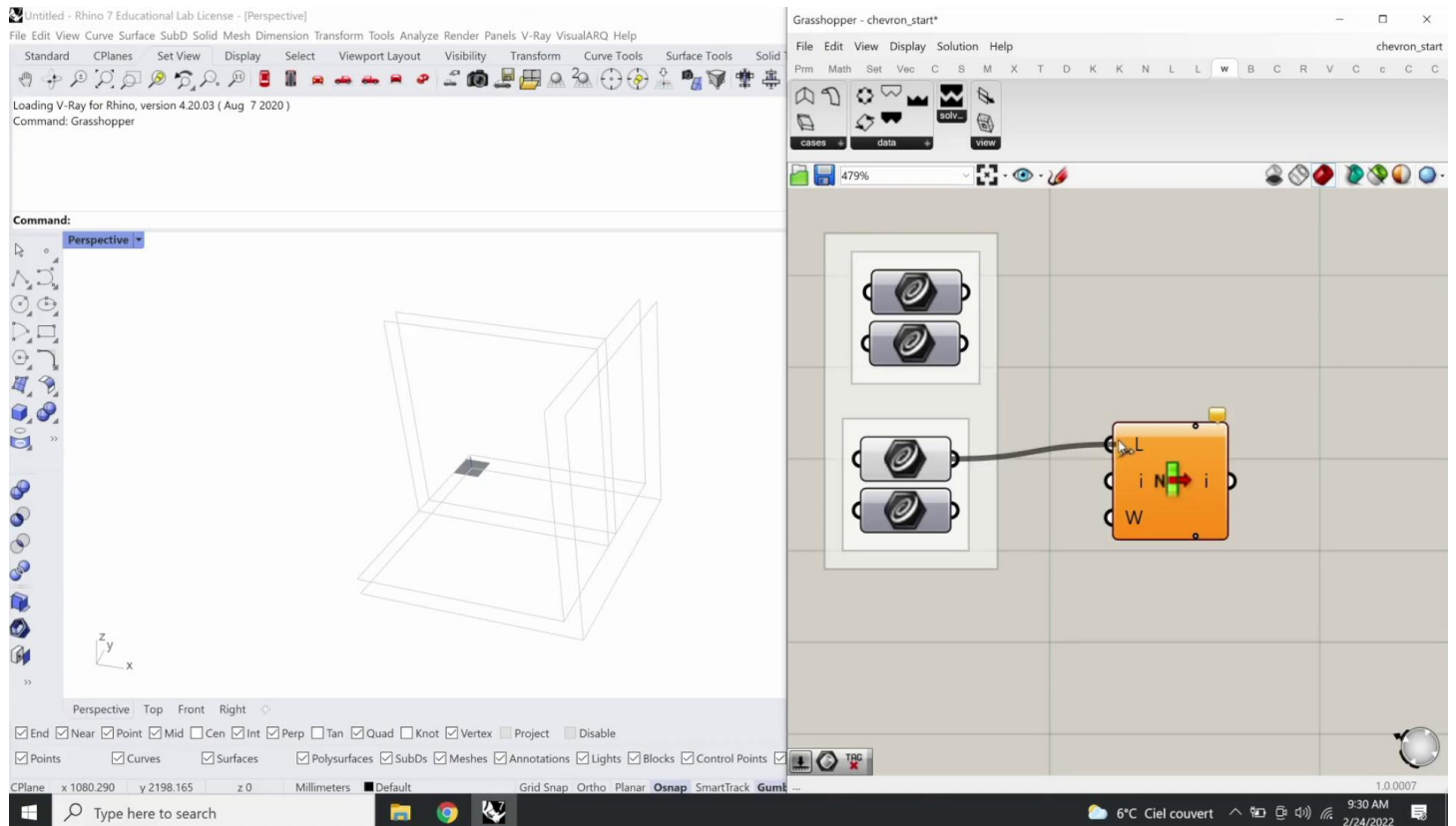
Once you open Rhino, there is already a new tab which is called compas_wood. Can be in your case that it is a text information but it is a compas_wood. Now, we need to open the example file.

Notes

Summary



1m 56s



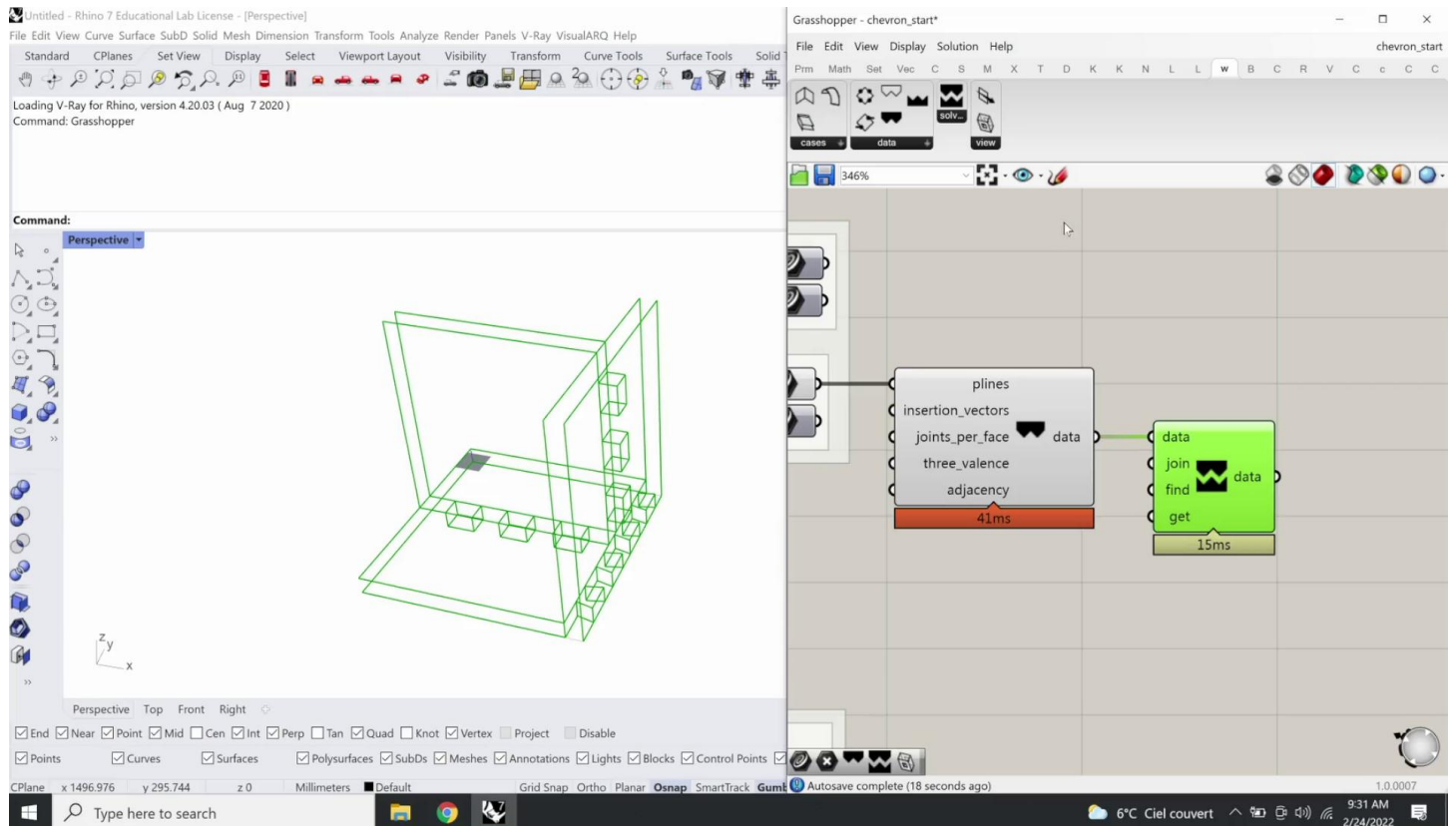
We can start from using chevron_start which will be our starting point of all the examples I'm going to show today. Simply drag and drop to the canvas and these are the data sets as a default once. We would not need to draw anything, just explaining really the core process. If I would zoom in to those polylines, you would see that there's a group of polylines and there are in total six polylines that represents each timber plate and if I go one by one, this first one polyline, second, third, fourth and so on.

Notes

Summary



2m 14s

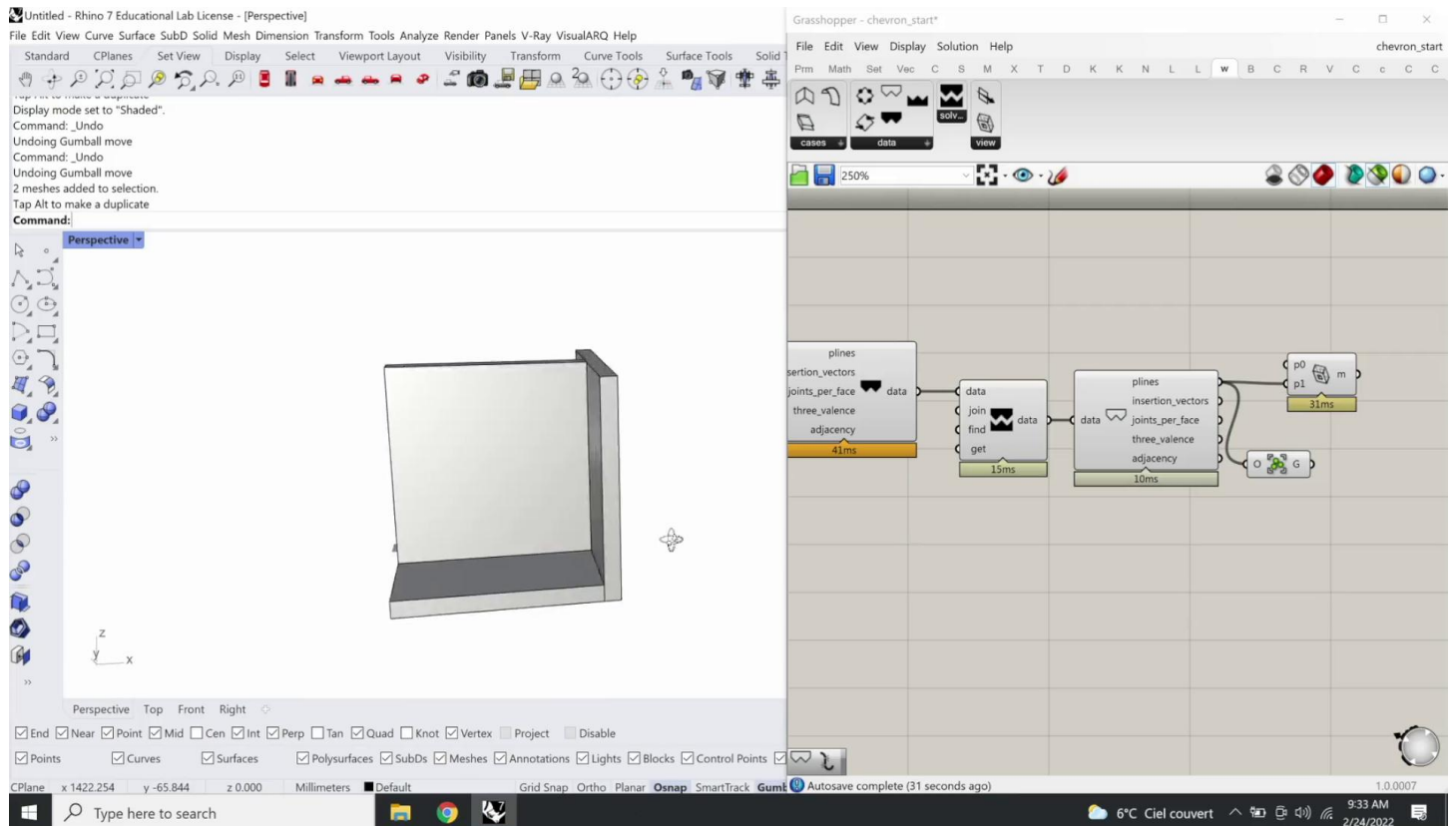


Meaning that the list of polyline must be an equal to a pair of polylines, where each pair represents a timber plate. Now the question, how we can generate a list of joints within this corner? As we saw in the segmented timber shell, we can first try to set up the data of the input of the plugin and then we'll try to compute the joints. Very first component that you need to use, it is called input_set. You can find it under compas_wood data. This component essentially has multiple properties, but at this moment, we will just place the polylines to a list of clients. This computed already some information that can be ready for the solver to compute the joints. Now, we can use the solver that actually computes the timber joints that requires the data which is essentially the list of polylines. It asks additional information but at current stuff, we can just use data as input and output and you see that it directly computed a list of joints. This information is made like that because we need also to output the data in. But the data you'll see that for instance, this input parameter and this input parameter is the same so we need somehow get the data structure back into Rhino.

Notes

Summary



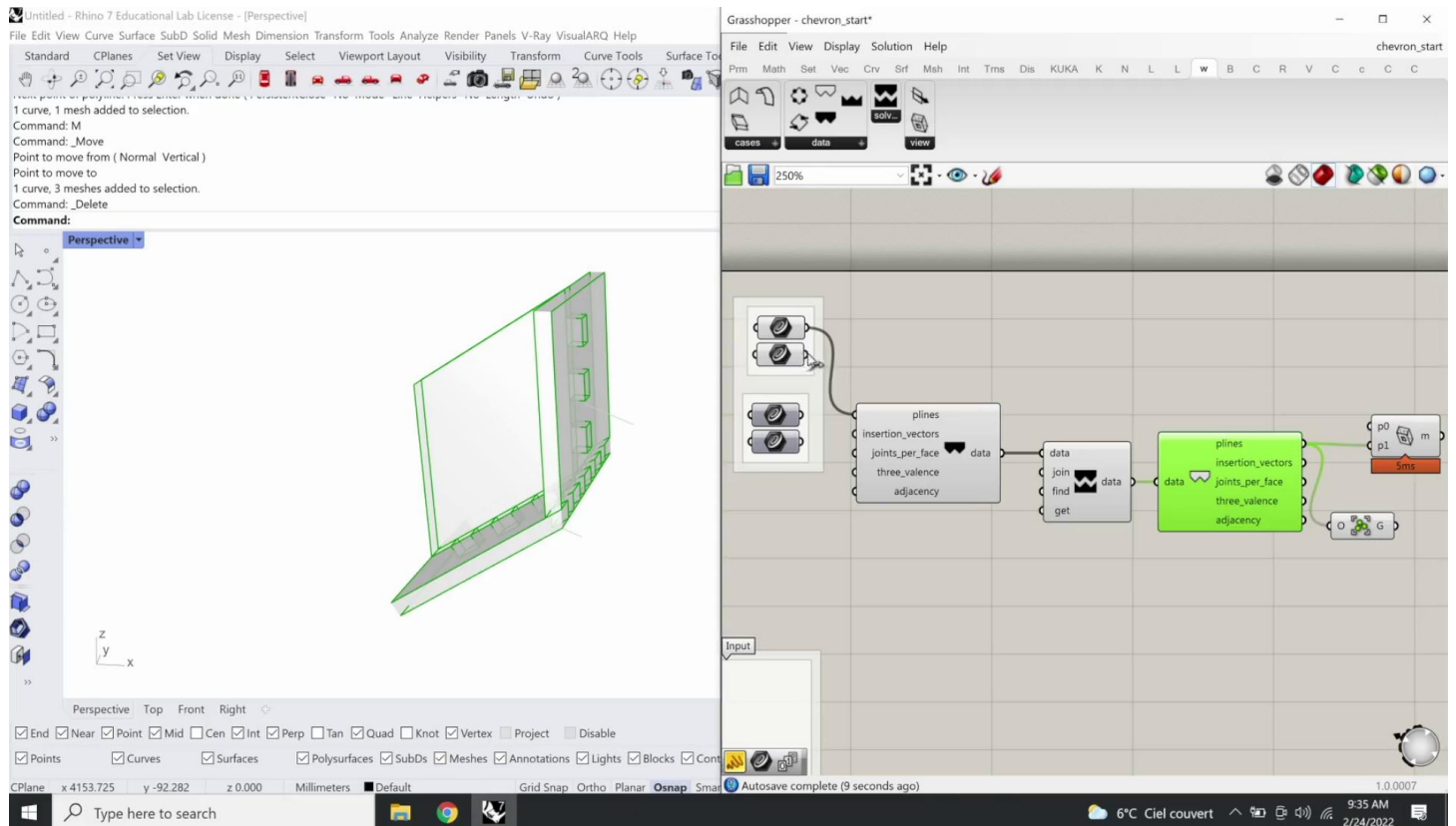


We can see this one is not really readable by anything. For this reason, there is another component called `input_get` so we ungroup the data set. You see in this case, we can have a series of outlines so these outlines represents each plate per branch so this is a branch ID. Then the following are the plate polylines so if I would bake the information. For instance, if I would try to remove it or remove the plates, you'd see that for instance, some of the plates already have two polylines because there are these timber joints. Another ones have two polylines including the holes and the same one for the next neighbour. Another question, how we can display this information not as polylines but a series of solid objects? We can use a component called `loft` so this simply loft the polylines including the holes. If I would bake this information again, you would see that I would have a series of solids that represents the timber plate. The problem right now is that this kind of joint works for orthogonal cases like we can assemble or disassemble those ductile joints. But when geometry gets curled, when it's not planner for the case of segmented timber shell, we need to insert this polyline in an angle.

Notes

Summary



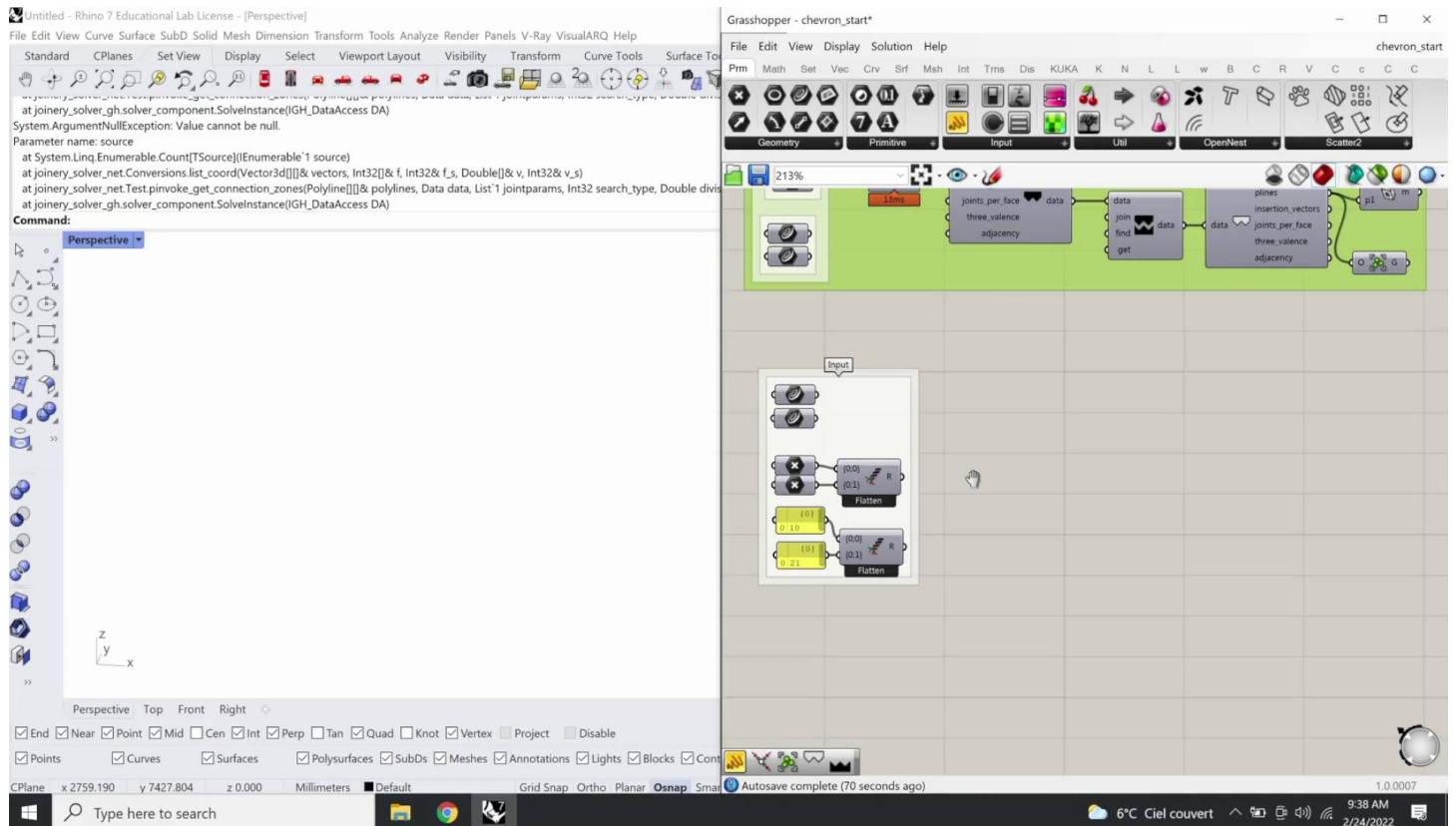


For instance, we need to insert, let's say according to one axis and if I would do it like this, it will not be physically possible because there is one insertion vector that goes down and one insertion vector that goes to the right. The question, how to do this? There is a second example that we're going to use in this case we can delete all the necessary information. We go to the next sample that is a case coming from the actual timber shell when there is an angle. I'll introduce how we can use the same definition in order to compute timber joints and change the direction of the timber joints. For this special case, we have another input information which is a series of lines that gives a direction of insertion and what is important to mention that these lines has a starting point at the plate edge and this is how computationally, those directions are computed based on the starting point the polyline face finds the insertion direction. If I would take again this component and place these polylines as these polylines currently, you see, that the edges were computed perpendicular to the face. Now, we can input also insertion vectors which angled and one another important part is that we need to partition the list in order to get a data tree.

Notes

Summary





For this step, we need to have a special component called `closest_lines`. We import a polylines here and then input a series of lines here. Then these polylines get split into a series of peers and then we place plines here and then we also place insertion vectors here. Now, this definition is working what this component did is an assignment of vectors because these are just two polylines and we don't know where to sell those polylines. If I would go into the output of V, you would see that the first plate, which is the top plate assigned a vector to the edge, and then assign another vector to another edge. The way it is structured that the first index is actually the top plate and bottom face and then the rest 2, 3, 4, 5 is actually this face, this face, this face, and this face. That's why we have actually six faces because each plate in this case, have six faces. This was the first example. Now, we are going to have a bit more a different example, showing the case of the box component. Let's hide all the information by wide selecting the full definition and we can go to solution and preview off and we can actually move the data so that we know that this is a simple case.

Notes

Summary



8m 36s