Triangle Mesh Fill

The Triangle Mesh Fill is a python script for Blender to fill closed 2D Bezier Curves with a triangle mesh of high quality. For creation of the mesh the free external Delaunay mesh generator 'Triangle' is used (for details of the fantastic program 'Triangle' see http://www.cs.cmu.edu/~quake/triangle.html from Jonathan Richard Shewchuk).

The Problem:

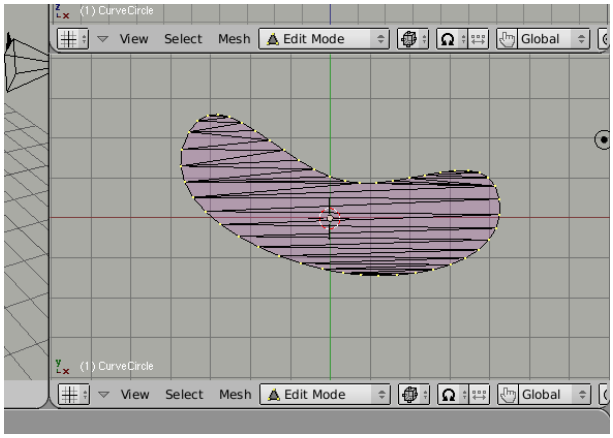
Did you ever wanted to deform a filled curve? It's nearly always a mess due to the scan-fill procedure when converting the curve to a mesh. Subdividing is nearly impossible and the very small angles of the triangles created will lead to a problem anyway.

The Solution:

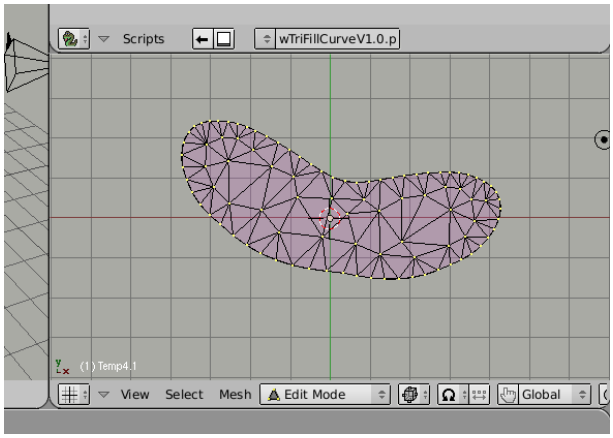
Using the external program 'Triangle' will give a very smooth triangle fill of the 2D-curve, even with very complex structures. The maximum size of the mesh (max. area value for one triangle) can be chosen and holes are possible too. The mesh created is consisting of triangles of high quality with maximum angle.

First impression of the tool

Let's have a look at a first, simple example:



Mesh created using Blender's Object -> Convert Curve to Mesh



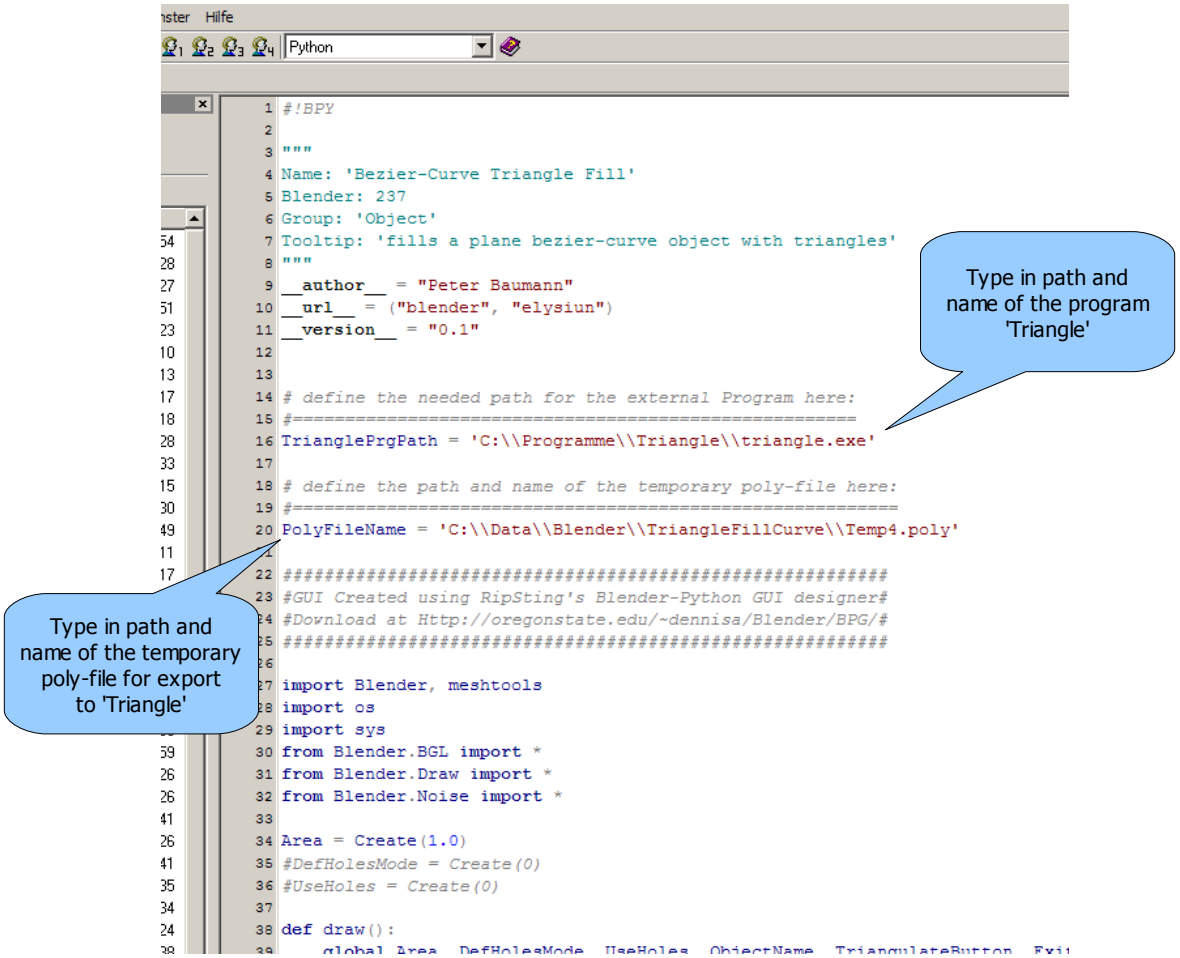
Mesh created using the 'Triangle Mesh Fill' script

Installation of the script and 'Triangle'

The script 'TriangleMeshFill.py' as well as a precompiled windows version of 'Triangle' V1.6 can be found on the BlenderArtists-Forum (http://blenderartists.org/forum/). In case you are Linux-User or just like to compile by your own, you can download the Program 'Triangle' from the link on top of this page. The source code is written in C and must be compiled with the GNU gcc compiler (was no problem on windows version ... I used gcc compiler of the Cygwin package). For the windows version the program 'Triangle.exe' and 'cygwin.dll' must be in the same directory (e.g. Create a directory
'C:\Program Files\Triangle' and copy the compiled and renamed .exe-file and the 'cygwin.dll to that directory).

Copy the python-script 'Traingle Mesh Fill' to the Blender\.blender\scripts folder and open the script with a text editor (e.g. the Blender internal one). The first lines must be edited to set the correct paths and file-names.

If you open up the script, the beginning should look like this:




```
#!BPY

"""
Name: 'Bezier-Curve Triangle Fill'
Blender: 237
Group: 'Object'
Tooltip: 'fills a plane bezier-curve object with triangles'
"""
__author__ = "Peter Baumann"
__url__ = ("blender", "elysiun")
__version__ = "0.1"


# define the needed path for the external Program here:
#=====================================================
TrianglePrgPath = 'C:\\Programme\\Triangle\\triangle.exe'

# define the path and name of the temporary poly-file here:
#=========================================================
PolyFileName = 'C:\\Data\\Blender\\TriangleFillCurve\\Temp4.poly'

############################################################
#GUI Created using RipSting's Blender-Python GUI designer#
#Download at Http://oregonstate.edu/~dennisa/Blender/BPG/#
############################################################

import Blender, meshtools
import os
import sys
from Blender.BGL import *
from Blender.Draw import *
from Blender.Noise import *

Area = Create(1.0)
#DefHolesMode = Create(0)
#UseHoles = Create(0)

def draw():
```

Edit the lines indicated above and save the text-file.
In Blender change to 'Scripts Window' and use 'Scripts ⇨ Update Menus'. Now the Menu 'Scripts ⇨ Object ⇨ Bezier Curve Triangle Fill' should be available.

How to use the script

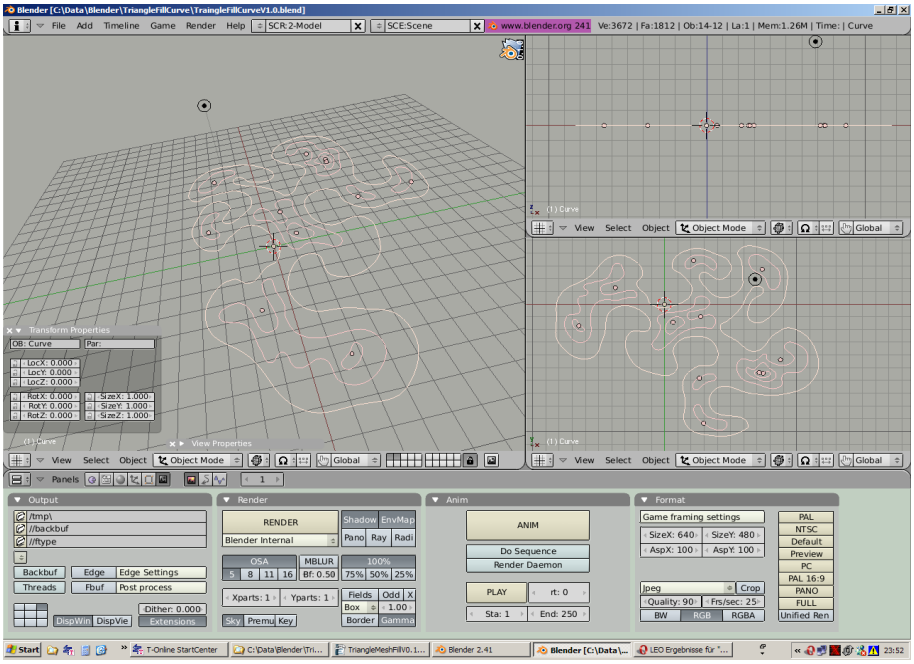
The basic steps for using this script are:

- Create one or more Bezier curves used as the outer shape (it can be serveral outer shapes simutanously)
- Create one or more Bezier curves inside these outer shapes for seams and hole definitions
  Always use a new Bezier curve object, even for holes!
- Subdivide each curve at least two or three times. Only vertices will be used and the smooth Bezier-information will be gone
- Start the script
- Select the involved curves
- Define the holes (for each hole use 3D-cursor on X/Y-plane and press 'Define Hole'-button)
- Choose the maximum area (in Blender-units)
- Create the mesh
- If result is o.k.,delete the curves (you can't see the holes, if you don't delete all used curves)

The Bezier curves to use in this script should be planar and must be located on the X/Y plane. Z-coordinates will be ignored.

Example: Building a Sculpture

At the beginning build up the Bezier curves for the outer shape of the sculpture and the holes. Please keep in mind, that you have to built a new object for each curve! This is different to the method Blender uses shapes with holes. In this case, the hole defining curve must be part of the outer curve object.
The curves can be quite complex and and you can define regions inside regions inside regions, etc. Every new curve -object inside an outer curve will be a seam of the outer object.

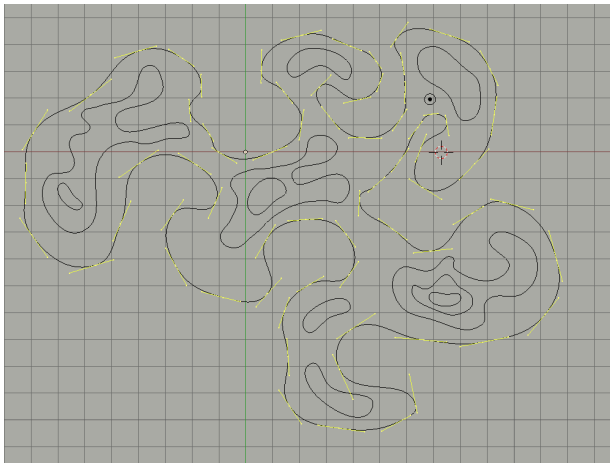


Creation of outer and inner curves of the sculpture

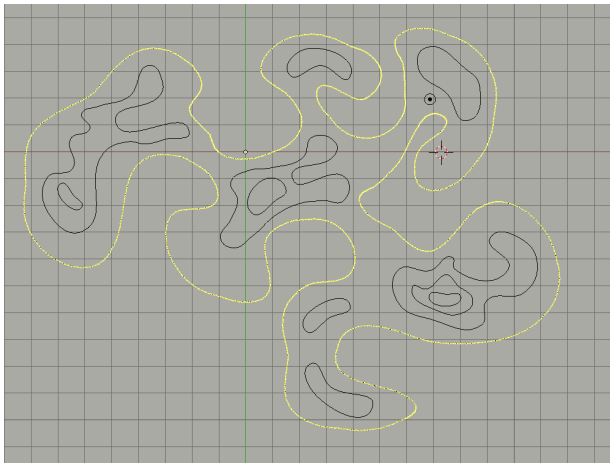
Always build up the curves on the X/Y plane. Z coordinates will be ignored. Please check, that the curve is not just rotated to be visible on the X/Y plane.

Hint: You can also import a 'SVG'-file of bezier-curves, e.g. created by 'InkScape'

Subdivide each curve. Select one curve, switch edit-mode (<TAB>), press <W> and select 'Subdivide' (first entry of the pop-up list). Use 'Subdivide' (<W>) at leat 2 or 3 times to receive smooth results of the shapes / seams. Switch to object-mode (<TAB>) and select the next curve and repeat the upper steps.



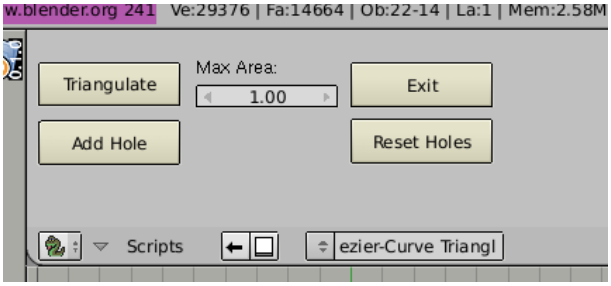
Outer curve after creation as Bezier



Outer curver after three times 'subdivide'

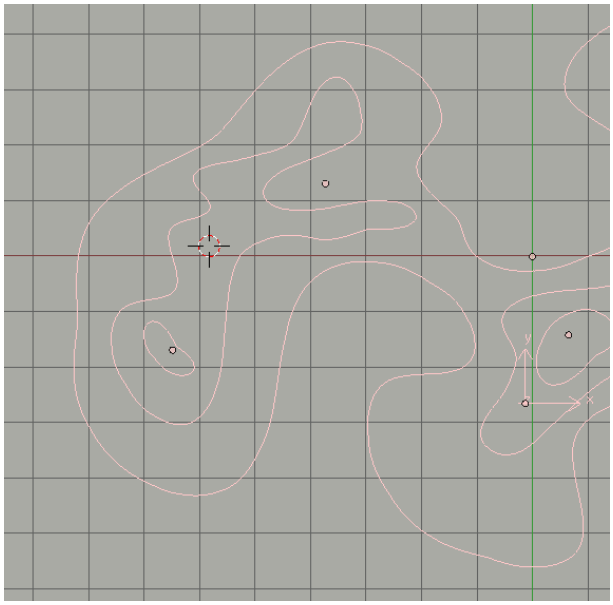
Subdividing is nessecary, because the script will only use the vertices of the curve. Thus the 'round' information of the 'Bezier-handles' will be lost and the curve will look very rough. The subdivision will create more points which are following the smooth Bezier-curve.

As the next step start the 'TriangleMeshFill' script, so that we can begin to define the holes.

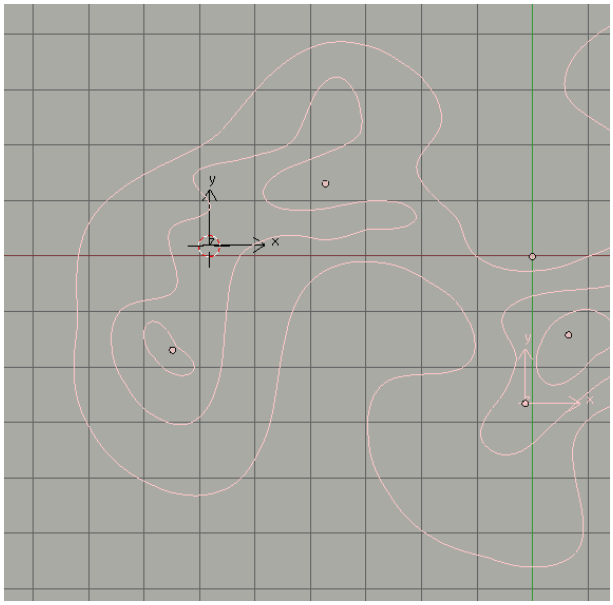


Layout of the script

For 'Triangle' holes are defined as single spots inside a border (seam). From this paticular point all triangles will be 'eaten' until a defined border will be reached. These borders are the curve-objects inside the outer curve-object, that we defined before.
Switch to the X/Y-Plane view and place the 3D-cursor inside an inner curve. Press the button 'Add Hole' to mark this point for hole-creation. Pressing the button will actually create an empty objectt sign the place of the hole (the name of the empty object will start with 'hole' for later recognition.
In case you want to restart the definition of the holes simply press 'Reset Holes'. Single holes can also be deleted by deleting the concerning empty object manually. When using 'Exit' all created holes (empty objects) will be removed.

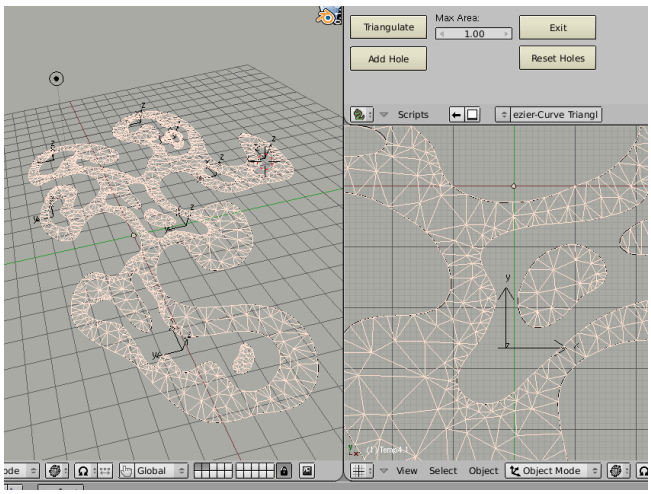


Place 3D-cursor to dedicated hole

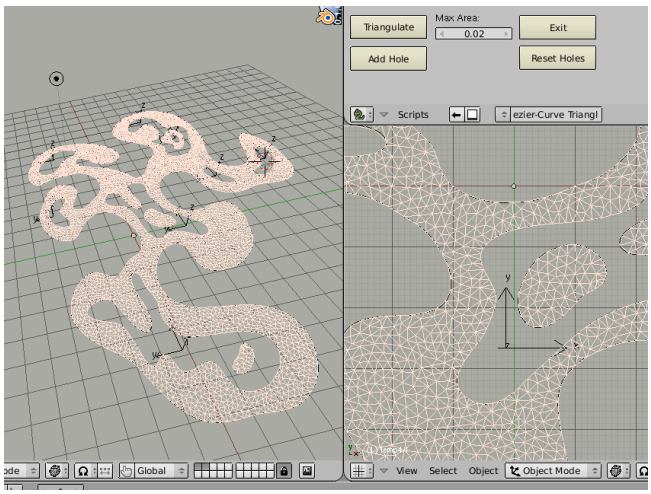


Press button 'Add Hole' and position will be signed

Before pressing 'Triangulate' to create the mesh, choose a proper maximum area size for the triangles, which will be created. The scale is given in Blender units. In case of the default value of 1.00 the area of one triangle would not overcome the area of one square seen on the 3D-floor of your scene (in case the view is not on one of the main planes).
The minimum size of the triangles is related to the density of your curve points.
Below you will find two examples for the maximum area size. The first on eis the standard value of 1.00, the second one a very small value of 0.02 to produce a very dense mesh (with a high memory consumption).

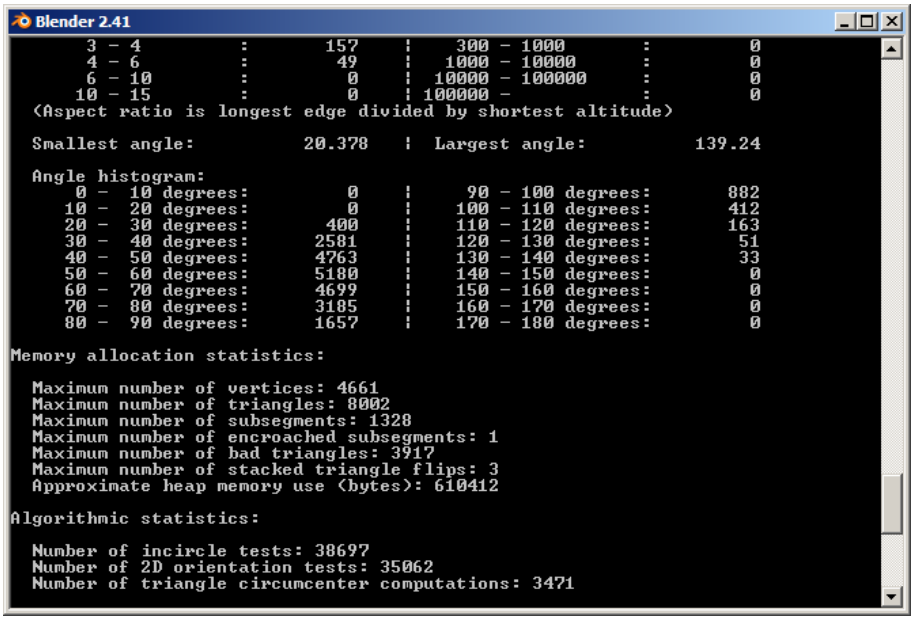


Max Area = 1.00 (default)



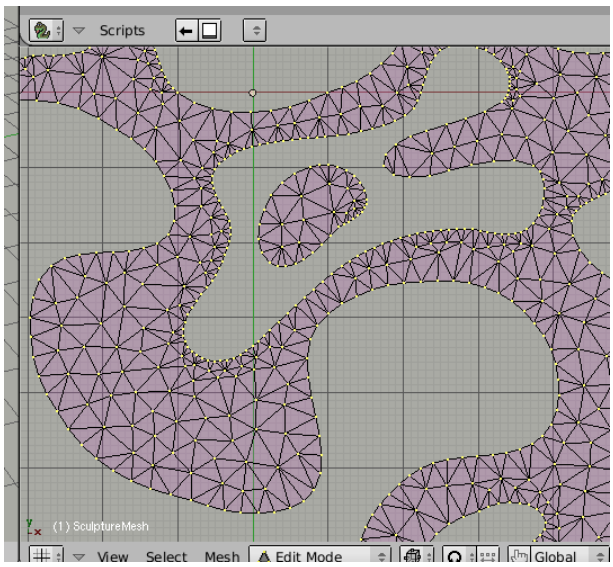
Max Area = 0.02

A detailed information about the statistics of the created mesh can be found in the Blender console.

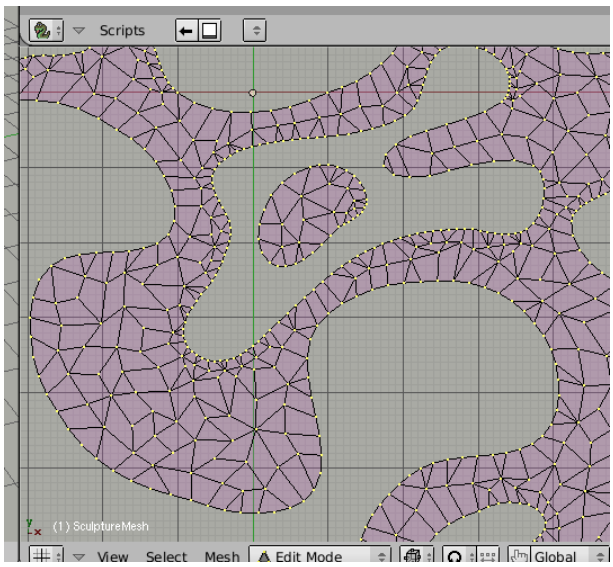


For the example a medium value of 0.10 for the maximum area was used.
In case the mesh does not fit your impressions (e.g. you would need a more dense mesh), simply delete the created mesh (the new mesh is the only selected object now, so just press the <DEL> key and confirm the object delete by 'OK'). Change the settings to your needs, select all curves once more and press the button 'Triangle' again.
If you are satisfied with the result, delete all curves you needed to create the triangle mesh. If these curves are not deleted (or moved to a different layer), they might have bad influence on the rendered images (no holes visible, crostalk of the two planes at the same location)
The easiest method is to select the curves in the outliner-view and to delete them in the 3D-view. At this point it is always wise to rename the new created object for more easy reference later on.

For those who don't like the triangles very much, you can easily convert (where possible), the triangles into quads by using the Blender function 'Mesh ⇨ Faces ⇨ Convert Triangles to Quads' in edit-mode.

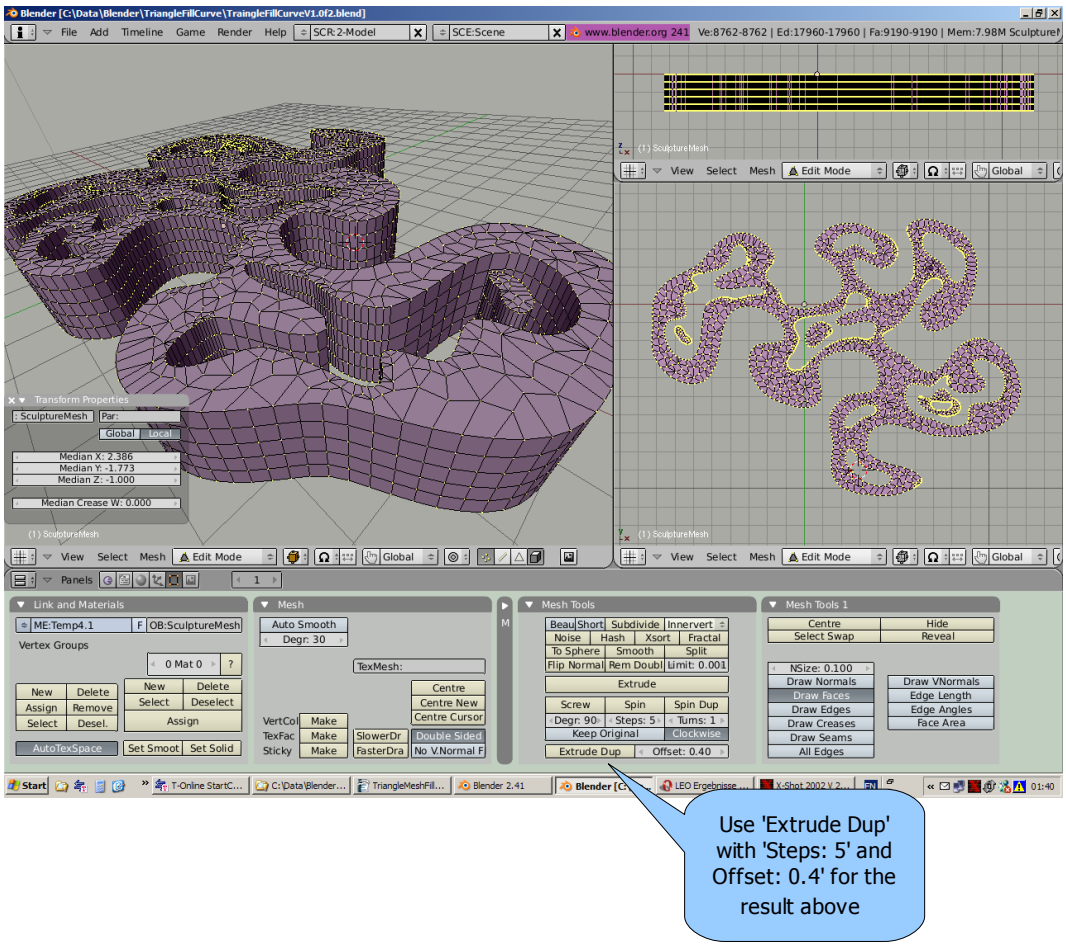


Original created mesh

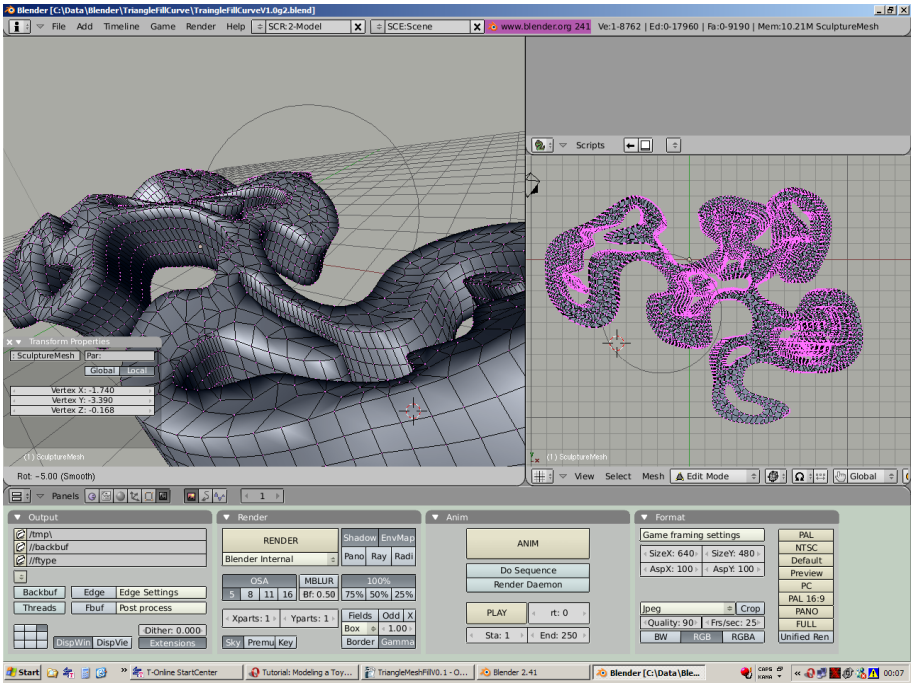


After using 'Convert Triangles to Quads'

The mesh can be extruded into the z-direction by using the 'Mesh Tools' in edit-mode. The size in z-direction of the single steps should be in the same order of size like the triangles / quads on the border of the mesh (in this example approx. 0.4).



After having formed a solid block, you can start to modify the new solid object to your needs. In our example we will start to rotate single points wich are connected via 'smooth falloff' to all other vertices. Please make sure to take a diameter of the 'smooth falloff'-influence to get a nice result.



As a last step I copied the new sculpture object to the nice 'glassfruits' scene well-known from the elysiun-pages http://blenderartists.org/forum/showthread?t=42605 (thanks to 'salted' by the way) and rendered the image with YafRay.

Peter Baumann, 19. April 2006