Advance Slicing

By Friend

Pre-workshop

- Download and install Meshmixer
  - I’m using Meshmixer 3.5 for the slides, but any version should be fine
- Download the model of the rabbit with hole from Github

If there’s a word you don’t understand in the slide, let me know and I’ll add it here

- Topology: organization of vertices in a mesh
- Vertex: a point in a mesh model; a model is made up of many vertices

Note

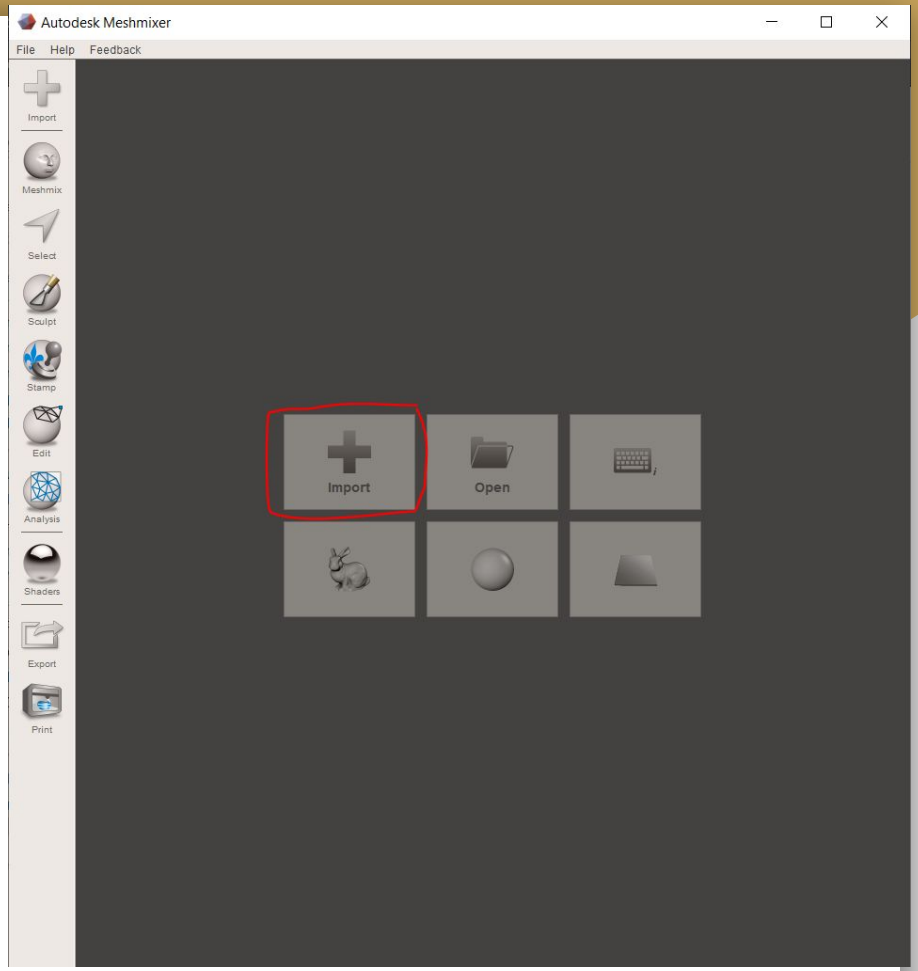
- I’ll use “” for button clicks

Agenda

- Import and export files, navigating
- Repairing holes in models
- Smoothing
- Basic Retopologizing
- Cutting models
- Tree support

Import

- You should see this screen when you first open
- Import the rabbit

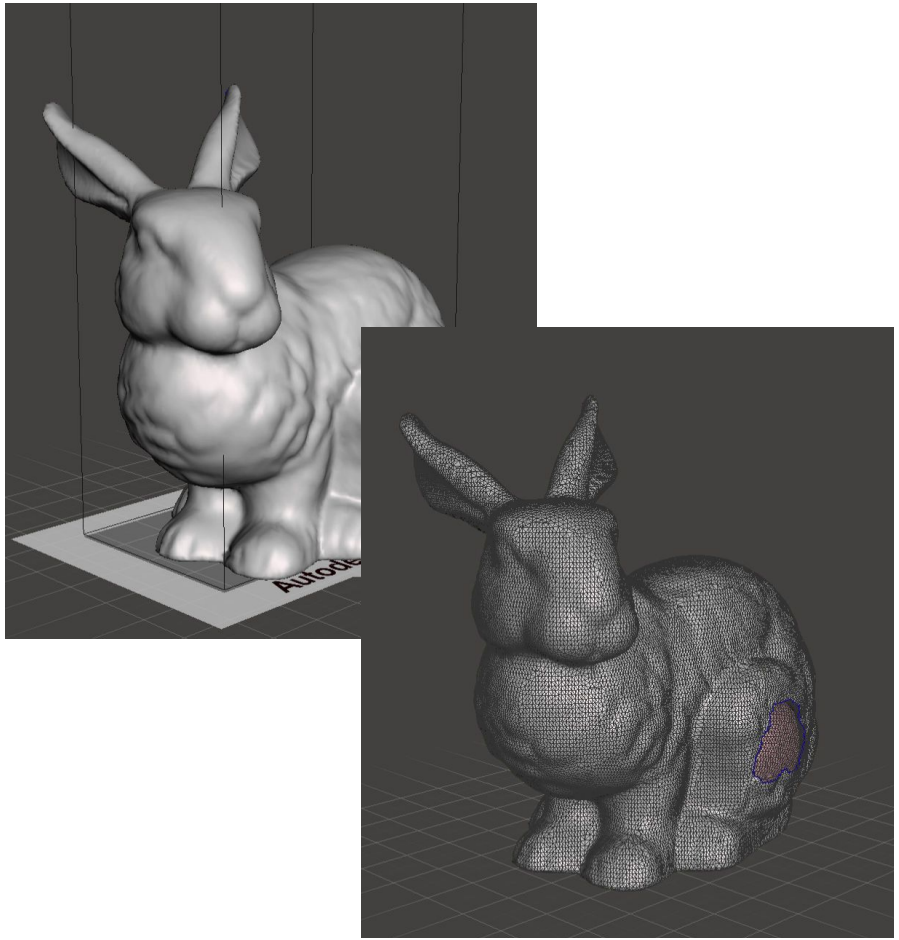


Navigating

Tumble	Alt +  or 
Pan	Alt +  or Shift +  or Alt + Shift + 
Zoom	(Maya) Alt +  or (Pen-friendly) Alt + Ctrl + ⌘ (Mac) 

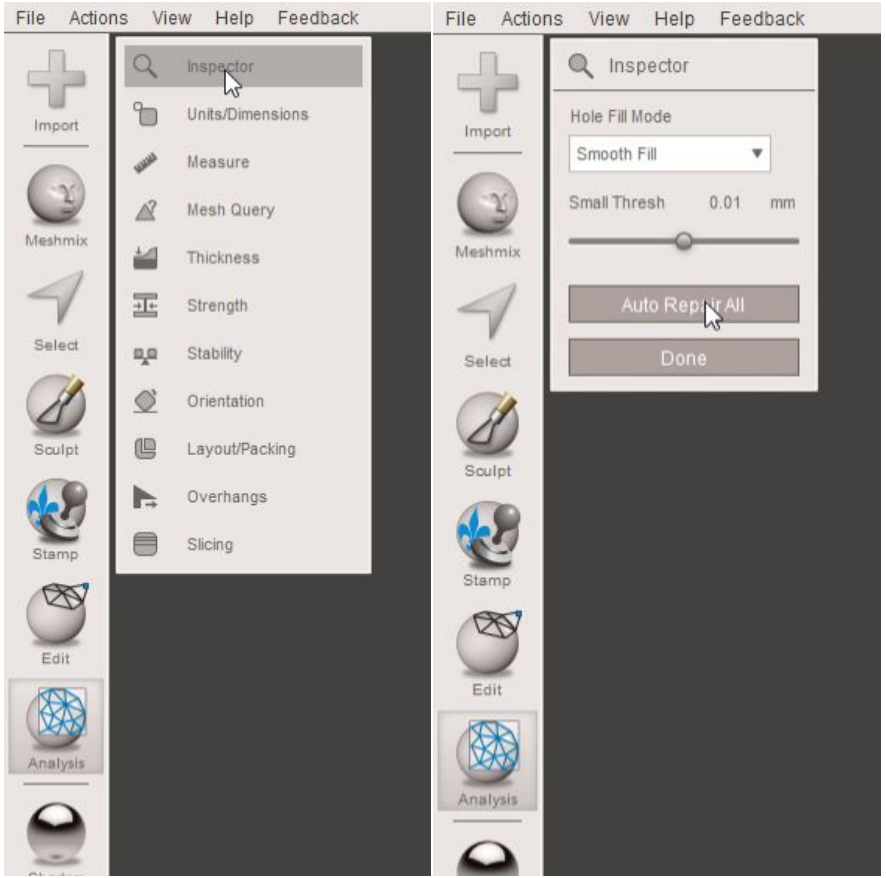
View

- On the toolbar on the top, there's "View" option
- For this WS, I recommend turning on wireframe, and turning off bed

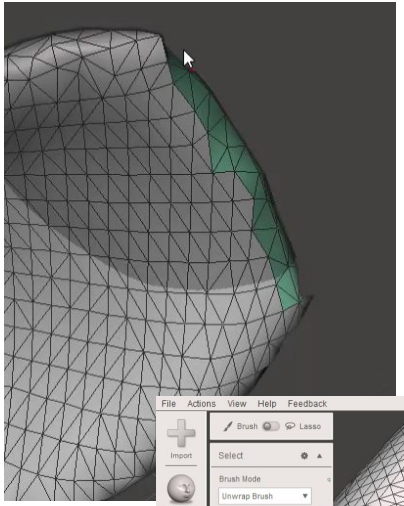


Hole repair

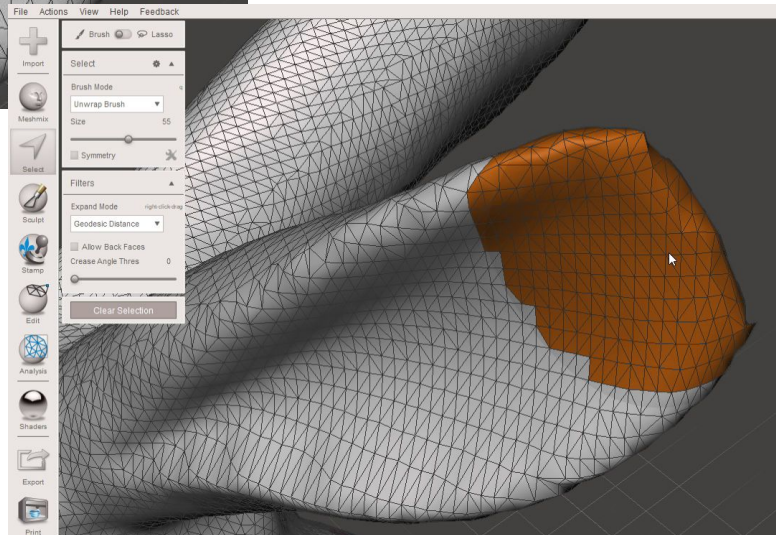
- On the left toolbar, “Analysis”->”Smooth Fill”->”Auto Repair All”
- Process is fairly automated in Meshmixer
- Different fill for different mesh
    - I used Smooth Fill since the rabbit is an organic model



Smoothing

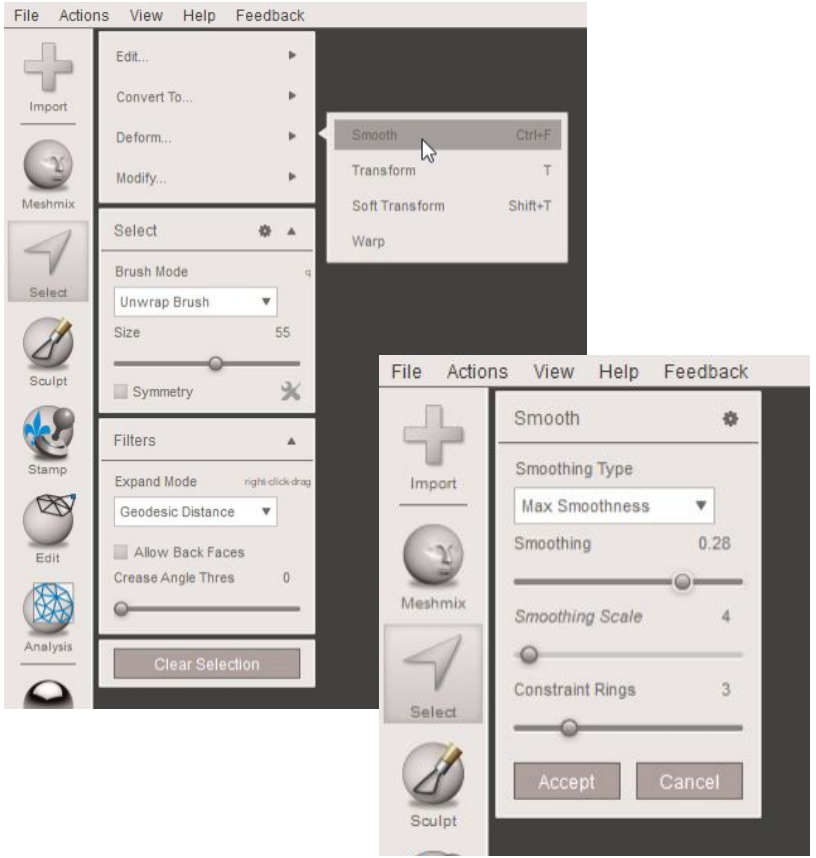


- Notice that the ear that was filled looks truncated
- We can smooth it out
- On the left toolbar, click “Select” and use the round brush to select the tip of the ear



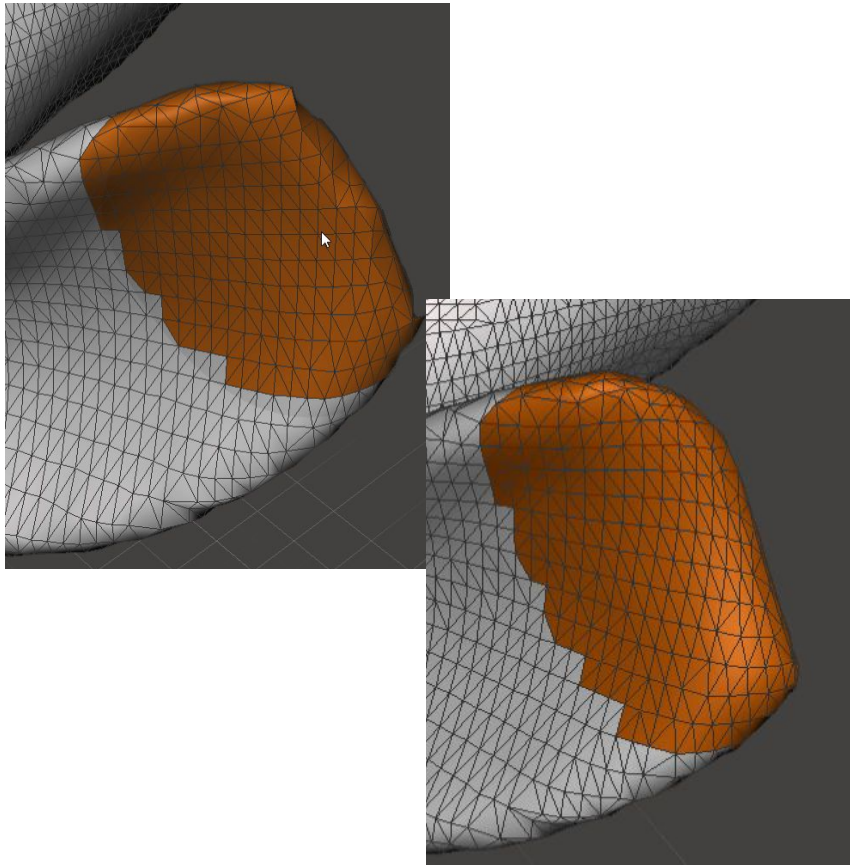
Smoothing

- Once you're done selecting, another set of buttons should pop up
- Go to "Deform…"->"Smooth"
- Play around with the settings so the ear looks nice



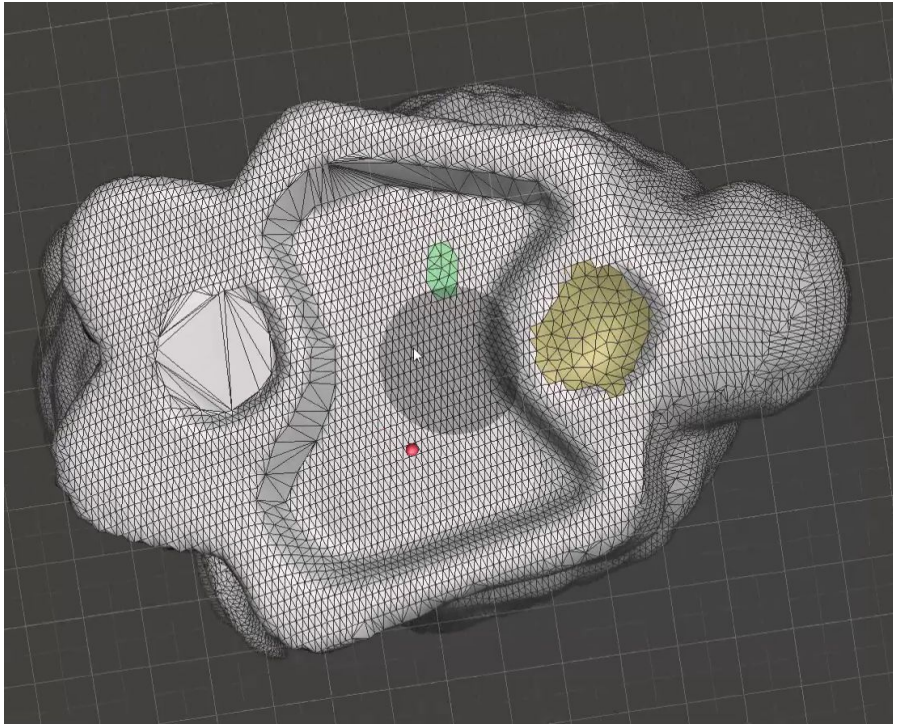
Smoothing

- Before and after



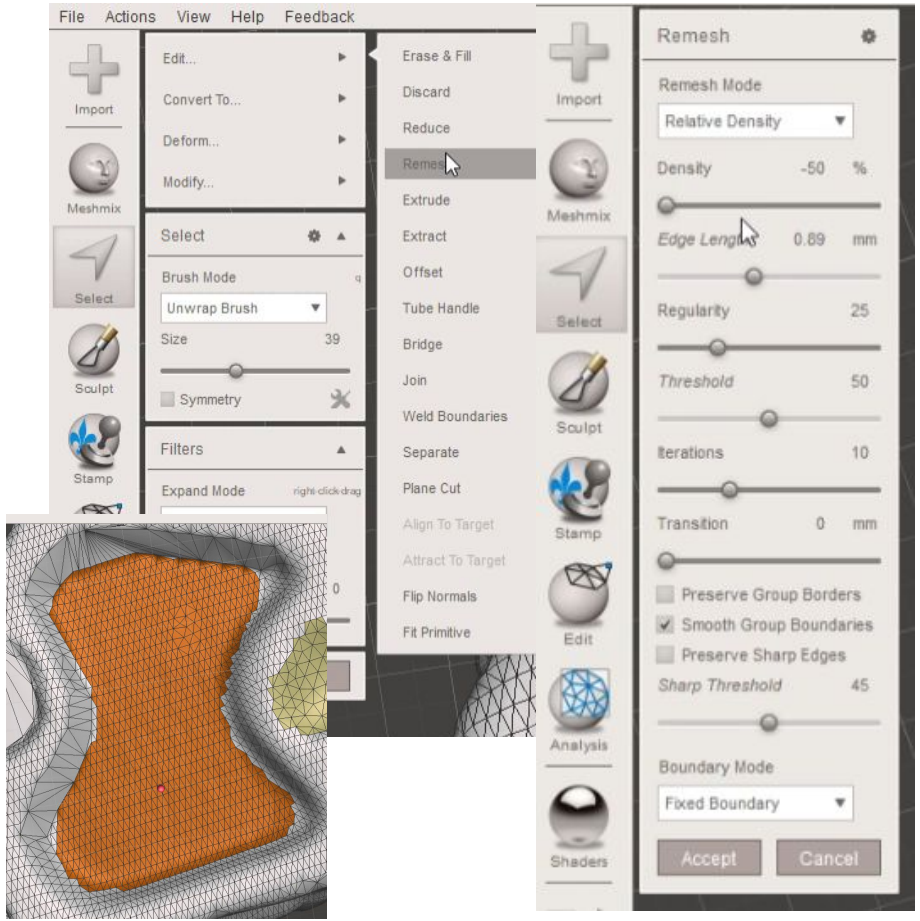
Remesh

- Rotate camera so you see the bottom
- You can see that there’s a lot of vertices for just a simple flat surface
- This makes the model larger and more complicated than it should be
- We can retopologize it (Meshmixer calls it “remesh”)



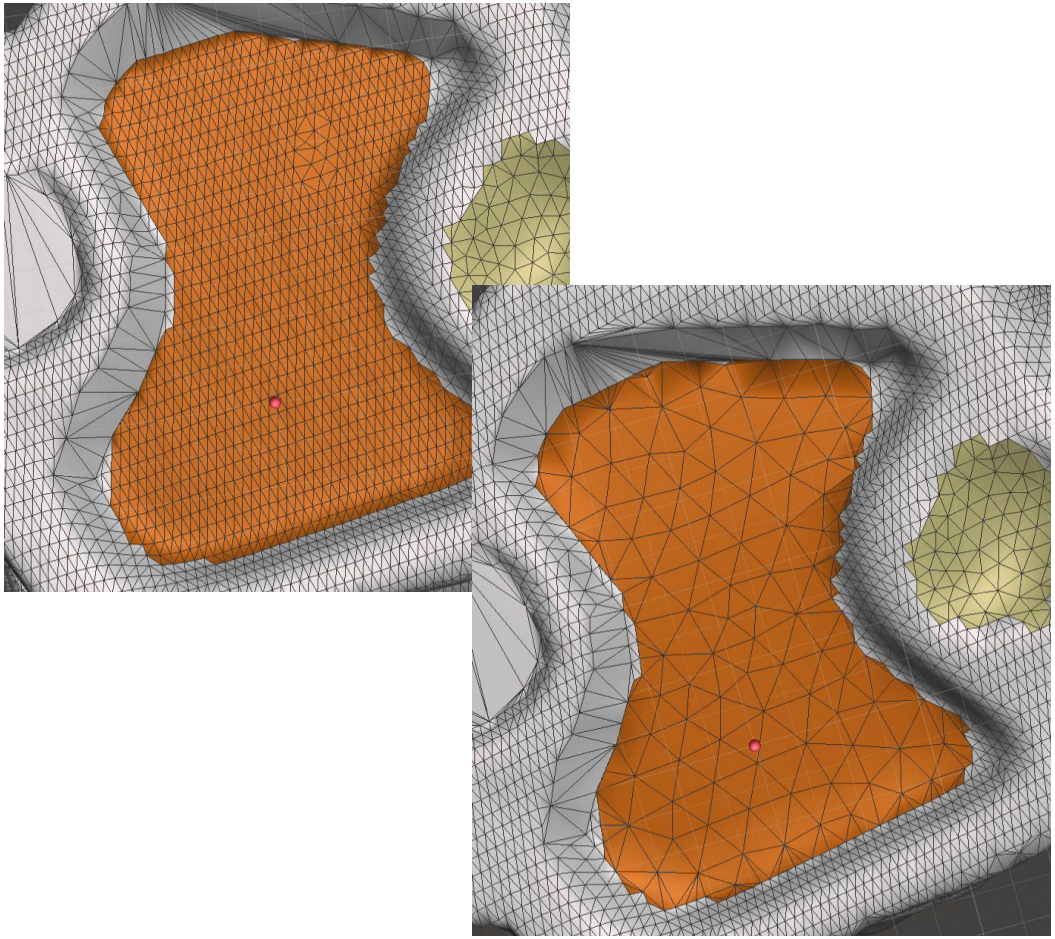
Remesh

- Click “Select” and highlight all the vertices on the flat part
- Go to “Edit”->”Remesh”
- Play with the settings to reduce the amount of vertices to the least you can



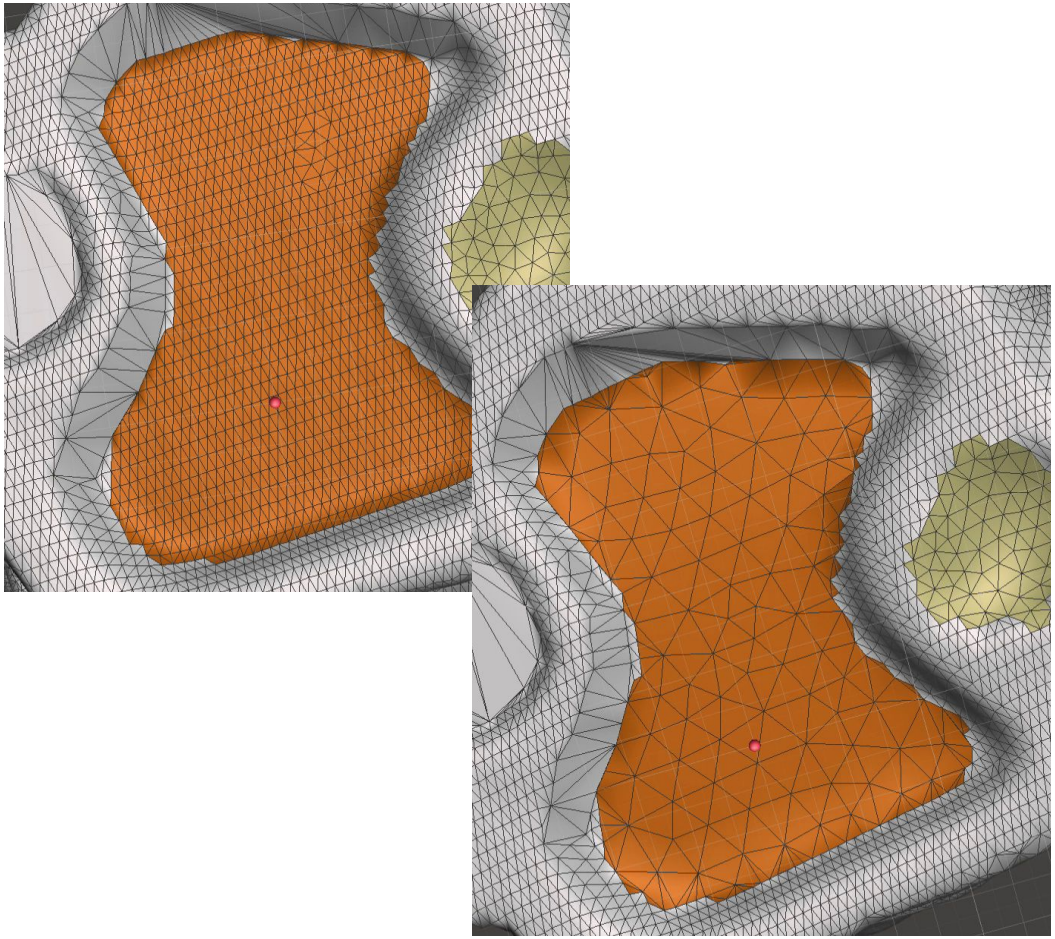
Remesh

- Before and after



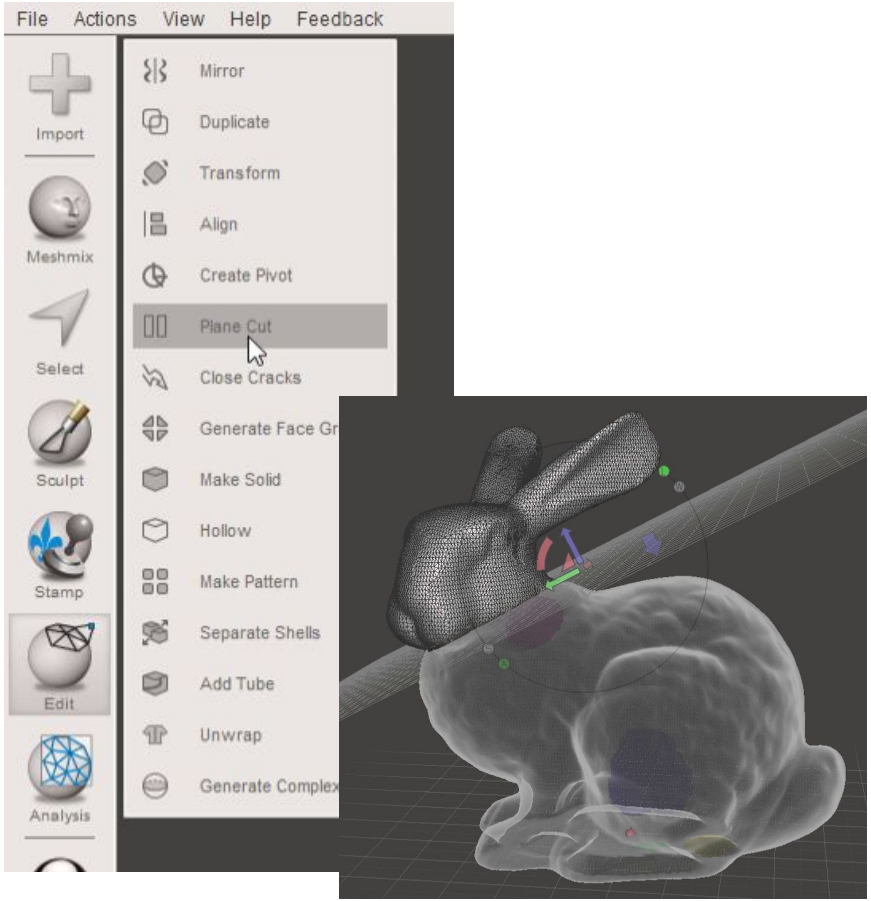
Remesh

- Before and after



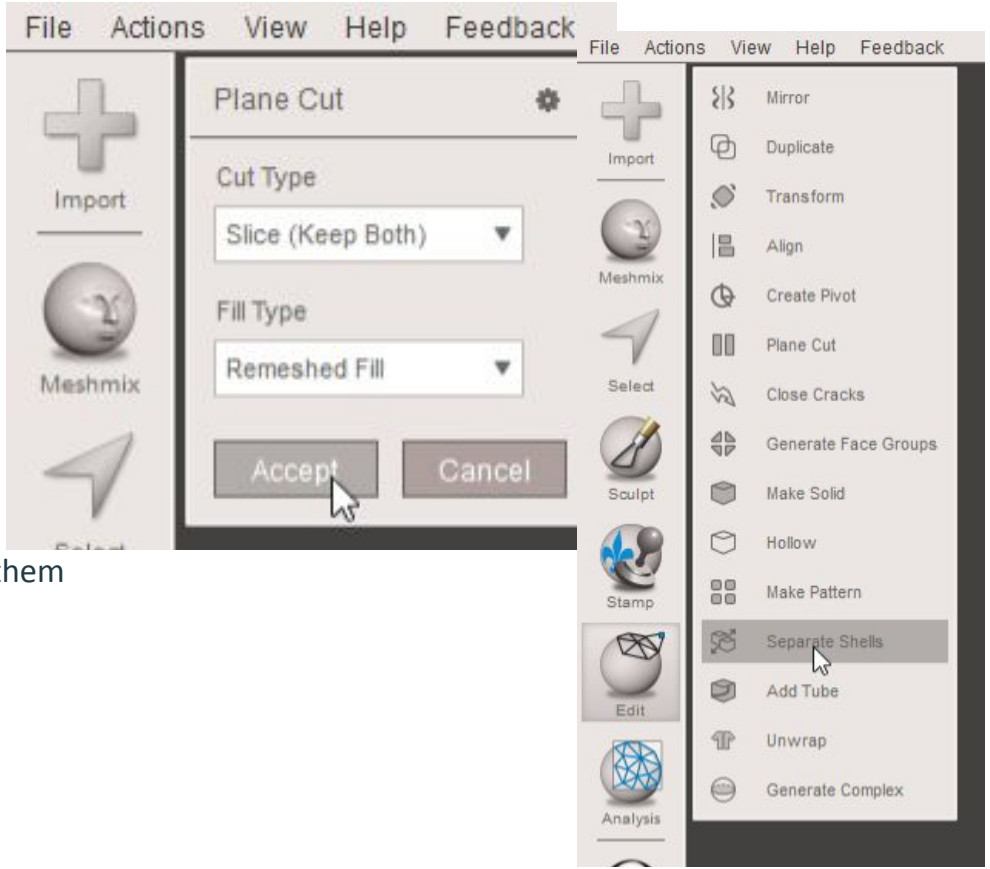
Plane cut

- Sometimes you want to cut a model into more printable chucks
  - Too big for printer, overhang wasting supports, get rid of overhang altogether, etc.
- Use plane cut by going to “Edit”->”Plane Cut”
- Use arrows to translate the plane, and the quarter-circles to rotate the plane



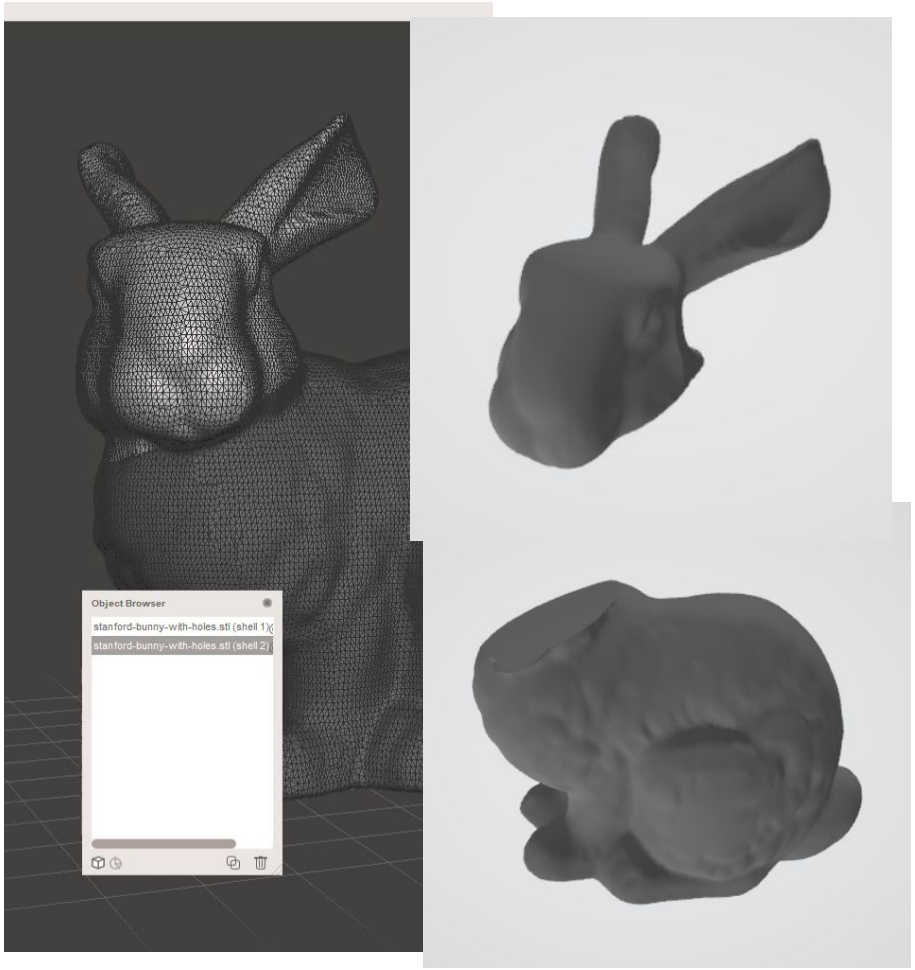
Plane cut

- Once you're happy with plane location, change Cut Type to "Slice (Keep Both)"
- Click "Accept" to make the cut
- Go to "Edit"->"Separate Shells" to make them separate models



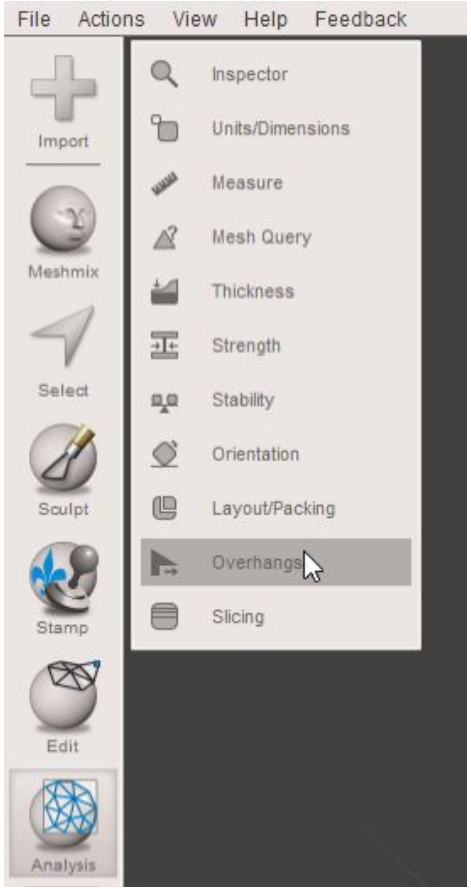
Plane cut

- You can select the model you want and export the models individually
- Go to the top toolbar, “File”->”Export” to export the selected model



Tree Support

- You've seen grid support in previous WS
- They take up quite a bit of plastic, and are difficult to remove, aren't they
- Tree support is generally less wasteful and easier to remove
- On the left toolbar, "Analysis"->"Overhangs"

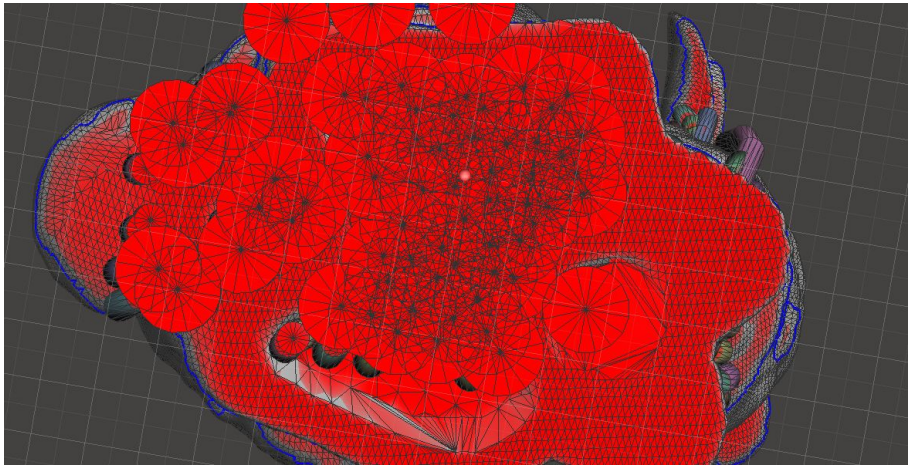


Tree Support

- You can change Angle Thresh to 45deg
- I'll leave other settings for you to explore
- Click "Done" to generate tree support
- Try generating and ctrl-z-ing a few times to get an idea of what each setting does to the support

Oh No

- Rotate to the bottom
- You'll see that there's a bunch of support generated for the groove on the bottom (the bump that we retopologized)
- Do we really need that groove? Can you get rid of it altogether?

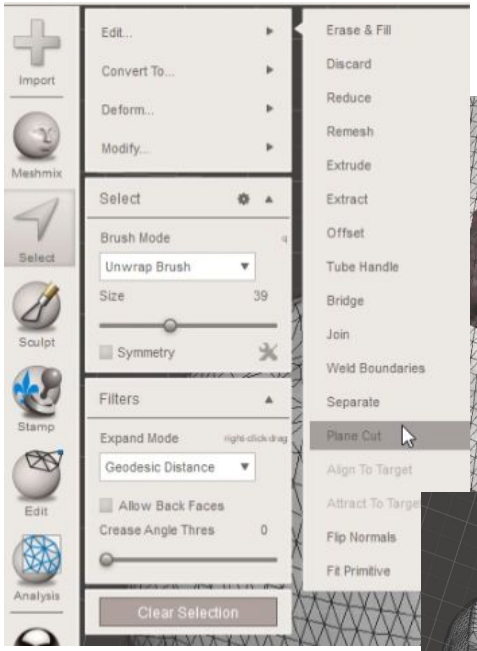


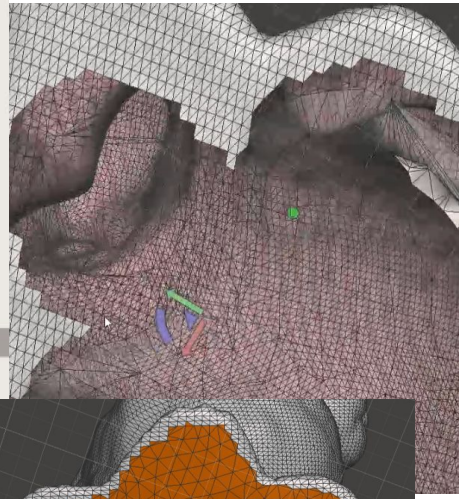
Exercise

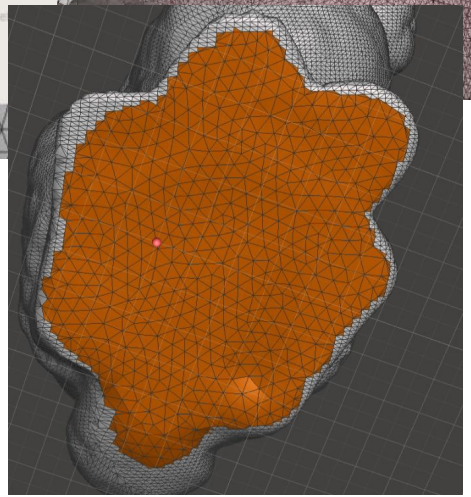
- As an exercise, I'll let you figure out a way using the skills we learned previously in this WS
- I did it by exploring the software a bit and breaking the problem down into steps:

1. Select the problematic grooves
2. Do a plane cut on them
3. Fill the giant hole created
4. Remesh it to make it efficient

- You don't have to do it this way

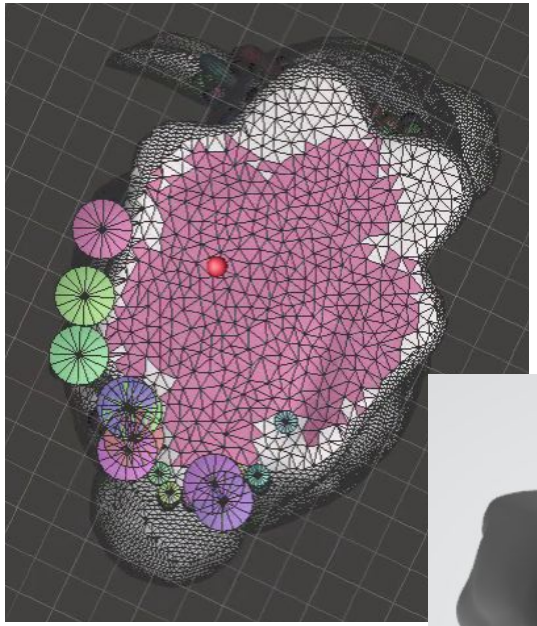






Exercise

- Generate tree support again and rotate to bottom
- You should see that there’s no wasteful supports generated now
- Export it



Note about exporting

- You should see two options when exporting: binary stl vs ASCII stl
- Pick binary stl for smaller file size, pick ascii stl for readability*

*in-depth explanation in the next slide, read it if you're interested

Footnote about exporting

- ASCII stl stores information in ASCII
- Each ASCII character is 1 byte, so if you open ASCII stl in Notepad++ and do character count, you'll see that character count matches size in bytes
- Since ASCII is essentially a table that converts binary to human characters, you need the same amount of binary digits to store any character
- EX: number 7 in binary is "111"*, whereas in ASCII it's "0110111"**
- That's a lot of wasted digits…

*I know 111 can also be two's complement -1 so "0111" is a better way to say 7, but I'm using the shortest form of a number to illustrate the point here **ASCII is actually 7 bits, but due to alignment, it's rounded up to 8 bits, hence a byte

Footnote about exporting

- Binary stl stores information directly in binary
- For comparison, a vector stored in binary is 3 numbers of 4 bytes (32 bits) each
- The range of a 32 bit number is −2,147,483,648 ~ 2,147,483,647, so we're not losing any precision
- For comparison, 2147483647 has 10 characters, so it'll take 10 bytes to store just one number! (that's wasteful isn't it)
- For a vector and 3 coordinates, a facet of ASCII stl could be as large as 120 bytes (10 * 3 * 4) (or even larger if your software (like MeshMixer) exports numbers with more than 10 characters), whereas the binary equivalent is only 50 bytes (4 * 3 * 4 + 2, for alignment)
- Scaling this saving up to 60k, 100k, 200k facets, and you'll see why binary stl is smaller than ASCII stl
- If you want to know the structure of binary stl files, there's a link on the next slide

Footnote about exporting

- This link is a no-nonsense reference for what ASCII stl is: https://people.sc.fsu.edu/~jburkardt/data/stla/stla.html
- Likewise, a no-nonsense binary stl reference: https://people.sc.fsu.edu/~jburkardt/data/stlb/stlb.html
- I've also put two stl files (a mesh with 1 facet) in this workshop's folder for comparison
- 312 bytes in ASCII vs 134 bytes in binary
- 312 chars = 312 bytes vs 50 + 80 + 4 = 134 bytes

Homework

- Distributed in WS

Some further readings

- These are where I got WS material from, and there's a lot more content I didn't cover in them:
  - http://www.diva-portal.org/smash/get/diva2:1222615/FULLTEXT02.pdf
  - https://blog.prusaprinters.org/cut-stl-models-3d-printing-meshmixer/
- If you want to get ahead, Basic Mesh Modeling WS uses Blender which does everything that Meshmixer does, but much more
- Cura also does tree support, although it is less customizable; I highly recommend you look at it