MENU

(https://pinshape.com/blog/how-to-generate-a-3d-printed-mold-for-an-object/)

How to generate a 3D printable mold for an object! (https://pinshape.com/blog/how-to-generate-a-3d-printed-mold-for-an-object/)

🕐
05 14 , 2015 (https://pinshape.com/blog/how-to-generate-a-3d-printed-mold-for-an-object/)
💬
24 Comments (https://pinshape.com/blog/how-to-generate-a-3d-printed-mold-for-an-object/#disqus_thread)
➡ Share

We were lucky enough to have Dan Steele (https://pinshape.com/users/11828-dan-steele) *write a post on generating a 3D printed mold for an object! He has done quite a few of these, and you can download his molds from Pinshape. But if you want to create your own, read on for Dan's tutorial!*

Introduction

I'm going to walk through the process of taking a cool character model and turning it into a mold, then casting copies from the mold.

Making a 3D printed mold can be a very easy process for some models. For instance, Robber Rex (https://pinshape.com/items/4075-3d-printed-robber-rex) was already Manifold, had no internal structures, and was low-poly so it took about 2 minutes to generate a mold w ₁ ith the reference script. You can find the mold I created for it here (https://pinshape.com/items/4996-3d-printed-robber-rex-mold).

You can find the script that I ran against the original model with the parameters I used already filled in here (https://drive.google.com/file/d/0B1w4gcTOEq2QczZFNjVqSV8zZTA/view?usp=sharing).

Bunny with an attitude was quite a bit more work.   This cool model was designed by Tanya Wiesner.

This tutorial documents how I turned this model into a mold. The exact same steps should be able to convert all but the most broken models into molds.

Making a mold and casting can be broken into three distinct parts:

- Prepping and cleaning up the model
- Making the mold
- Casting parts

Preparing and cleaning up your model

The three requirements for a model capable of generating a mold are:

- Manifoldness – A model must be manifold in order to turn it into a mold. For a model to be manifold, it must make up a volume that could exist in the real world. The most common cause of non-manifoldness are "holes" in a mesh. If the mesh isn't "water tight" and there is no clear inside and outside volume, a model is not manifold.   ₁
- No internal structures – Often when models are created, they are made by jamming multiple other models together. If these models aren't stitched together properly with their internal structures removed, then we need to fix this before we attempt to make the mold.
- Low-Poly – To use OpenSCAD to create a model, the poly count on the model must be less than 20,000. While this number may sound low for most objects, a higher poly count doesn't get you any more detail anyway.

Step 1 – Reducing the polycount

Use Meshlab for this process. It's free and can be found here (http://meshlab.sourceforge.net/). Meshlab is a full-featured tool but I mostly use it for reducing polycounts. Start Meshlab up and load your STL by clicking on File->Import Mesh...
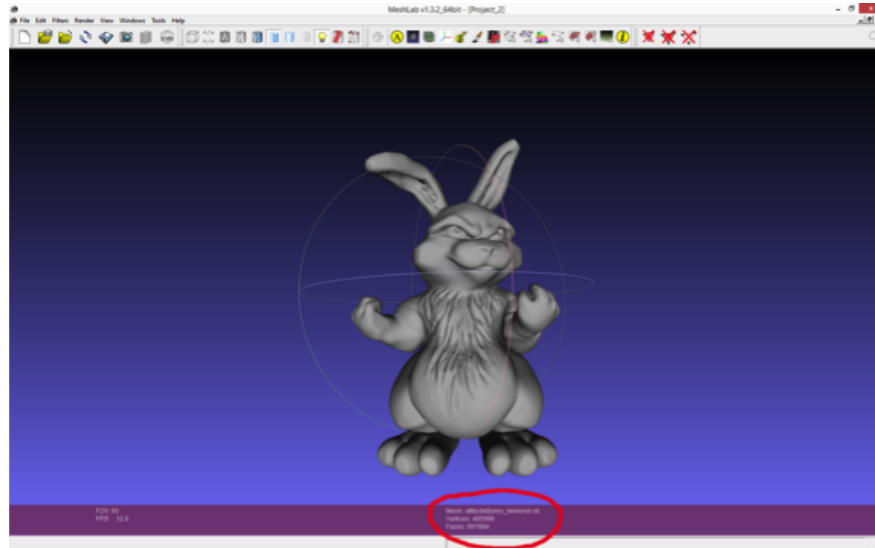
(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134534/mold-1.png)Next take a look at the face count. It's circled in red in the above picture. Right now it's at almost 1 million. I want to reduce it down to well under 100K for the first part of our process.
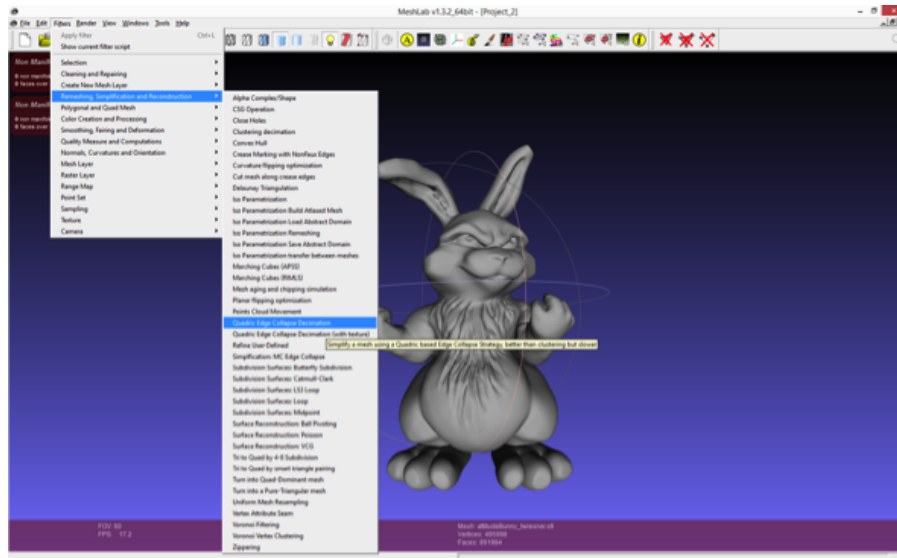
(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134528/mold-2.png)Click on:

Filters->Remeshing, Simplification, and Reconstruction->Quadratic Edge Collapse Decimation

Set the number of faces to something that still looks reasonable when you hit "Apply" but is below 100K (I chose 60K). Hit "Apply".

Your Model is now decimated down to a workable number of faces. If you skip this step, some of the software used in later steps won't load the model, or will crash when you try.
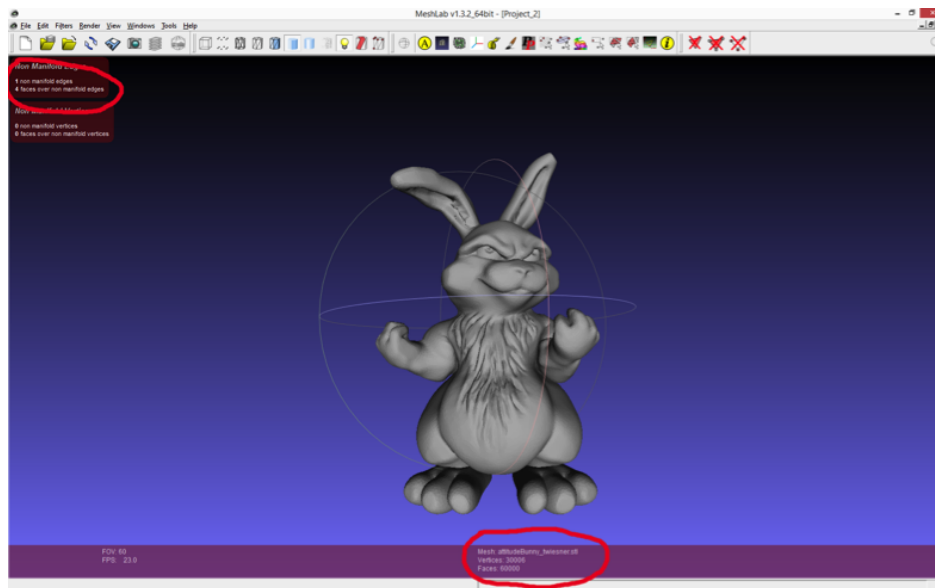
Step 2. Is it manifold?

Now we need to verify our model's manifoldness. Quadratic Edge Decimation generally produces good-looking results but frequently breaks manifoldness. The attitude bunny I'm using was manifold before the decimation, but now is not. Here's how you tell:

Click Render->Show Non-Manif Edges

   then

Click Render->Show Non-Manif Vertices

(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134519/mold-3.png)Notice in the above example we now have 60,000 faces in the bottom center. Also up in the top left hand corner we also have two boxes that should have a total of 4 zeros if everything is manifold. But I have a 1 and a 4. To fix this click:

   Filter->Cleaning and Repairing-> Remove Duplicate Faces

   Filter->Cleaning and Repairing-> Remove Duplicate Vertex

   Filter->Cleaning and Repairing-> Remove Faces from Non Manif Edges

Once that's done export your mesh as an STL even if it still isn't perfectly manifold.

If after selecting all of the above your model still isn't manifold, you'll have to load it in Netfabb Basic (also free) and attempt automatic repair from there (http://www.netfabb.com/downloadcenter.php?basic=1).

If Netfabb basic won't fix it you may need to reduce the polycount further and repair it by hand in Sketchup. This is tedious and not for the faint of heart.

Speaking of Sketchup, it's a great tool for taking a look inside our mesh to see how the insides look. It's free and you can download it here (http://www.sketchup.com/).

You'll need to install an STL importer/exporter. There's one available in the Sketchup plugin library. Once you've done that, you'll be able to import your model and have a look. I highlighted the back of my model, then right-clicked and selected hide. Here's a peek at the insides:
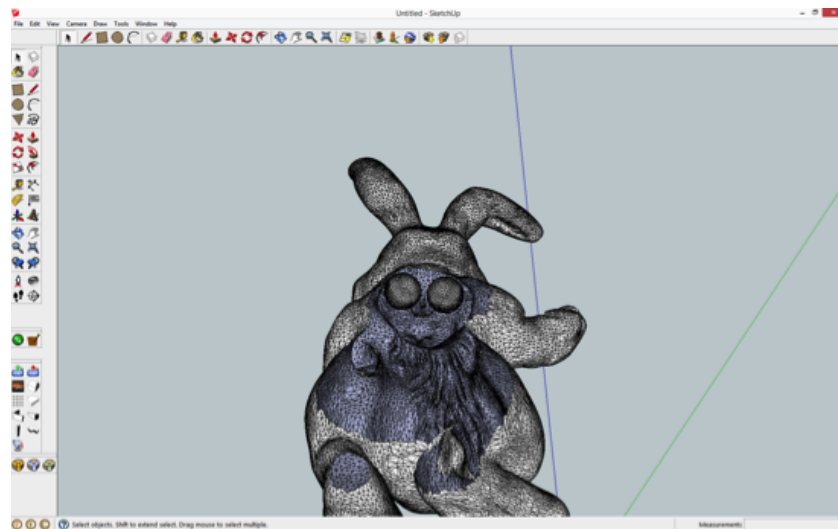
(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134503/mold-4.png)If you look carefully you'll notice two things: the spheres up in the head structure and the overlap in the legs. These internal structures need to be eliminated to make the model properly manifold.
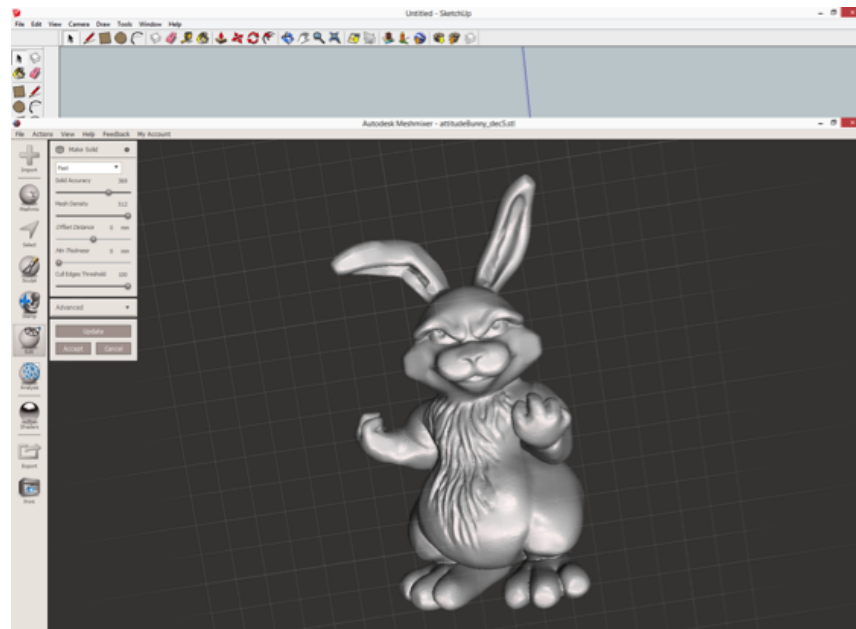
Step 3. Get rid of internal structures

This used to be a long and tedious process but now there's an easy and free solution. Autodesk's Meshmixer tool is free and can take care of this for us. Download it here (http://www.123dapp.com/meshmixer).

Load Meshmixer and choose import to import the latest version of your mold.

(https://blog-assets.pinshape.com/blog/wp-
content/uploads/2015/05/02134457/mold-5.png)

Click Edit->Make Solid

You'll need to make some adjustments here to make things come out right. Adjust the values of "Solid Accuracy" and "Mesh Dens[1] ity" upwards until your model looks nice and clean when you hit "Update". Once you're happy with the results hit "Accept". Now go to File->Export and export it as an STL. This will produce an amazingly large file (mine was 1.9GB and a couple million polys). You'll need to bring this back into MeshLab and decimate it down to less than 20,000 polys (I used 11,000). Follow the steps above to clean up any final manifold errors.

Your model is now ready to be turned into a mold! ☺

Making the mold

For the next part you'll need to download OpenSCAD. It is another free modeling tool and you can find it here (http://www.openscad.org/).

You'll also need my mold generation scad which can be found here (https://drive.google.com/file/d/0B1w4gcTOEq2QYjhkYWZmT18xQUE/view?
usp=sharing).

Put the Mold Generator script in the same directory as your model and load the script in OpenSCAD. The code window will look something like this:

```
/***************************************************

Original Author

Parametric two-part mold generator
Author: Jason Webb
Website: http://jason-webb.info

Modified by

Dan Steele (rocketboy on thingiverse)

Changed one of the keys to a cube so mold can't be put together backwards.
Center poor hole option for plaster.

A parametric two-part mold generator that constructs two-
part molds with registration marks based on STL files.

***************************************************/
// Model parameters
model_filename = "dbdec11.stl";
model_rotate = [90,0,90];
model_translate = [-35,-25,-11.8];
model_scale = .9;

// Mold parameters
mold_width = 78;           // Measured along X axis
mold_height = 47;          // Measured along Y axis
mold_depth = 23;           // Measured along Z axis
mold_spacing = 3;          // Space between mold halves
rounded_corners = true;    // Rounded the corners to lessen warping
edge_radius = 8;           // Radius of rounded corners

// Key parameters (registration marks)
key_size = 3;        // Radius of spherical keys
key_fettle = 0.4;          // Size difference between keys and corresponding holes
key_margin = 7;  // Distance from outside edge of mold

// Pour hole parameters
pour_hole = 0; //0 for false 1 for true.
location = 0; //0 for now


side_by_side();

/*******************************************
 Rotate and place both halves side by side
 along the X axis for easy single-plate printing
*******************************************/
module side_by_side() {
    // Scoot the left half over a bit
    translate([mold_width/2 + mold_spacing/2, 0, mold_depth/2])
        bottom_half();

    // Rotate the top half, then scoot it over a bit
    translate([-mold_width/2 - mold_spacing/2, 0, mold_depth*3/2])
        rotate([0, 180, 0])
            top_half();
}

/*******************************************
 Bottom half of the mold
*******************************************/
module bottom_half() {
    // Create the mold form with negative keys
    difference() {

        // Create the basic mold form by subtracting the STL from a cube half it's size
        difference() {
            difference() {
                if(rounded_corners)
```

(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134444/mold-6.png)

You'll need to change model_filename to the name of your STL. You can hit F5 to render a view of what the script will do right now. If you are very lucky it will be perfect right away, but chances are it won't be. Note that this script creates a negative of your model and pushes it into two halves of your mold. A negative space is difficult to see and position so there is a small change you can make to the script to see the model you are trying to place in the mold halves.

```
/*******************************************
 Bottom half of the mold
*******************************************/
module bottom_half() {
    // Create the mold form with negative keys
//  difference() {

        // Create the basic mold form by subtracting the STL from a cube half it's size
//      difference() {
//          difference() {
                if(rounded_corners)
                    roundedBox([mold_width, mold_height, mold_depth], edge_radius, true);
                else
                    cube(size = [mold_width, mold_height, mold_depth], center = true);

                scale(model_scale)
                    translate(model_translate)
                        rotate(model_rotate)
                            import(model_filename);
//          }
            if (pour_hole) {
                translate(model_translate)
                    translate([mold_depth,0,0])
                        cylinder(h = mold_depth*2, r1 = mold_depth*2, r2 = mold_depth, center = true);
            }
//      }
//  }
        }
```

(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134431/mold-7.png)Notice the six '//' I added on the left-hand side. Add those and hit F5 and all of a sudden you will see your model in the preview. You can now move the model around and put it halfway into the right-hand side of your mold. You move your model around by tweaking these variables

```
*************************************************/
// Model parameters
model_filename = "abdec11.stl";
model_rotate = [90,0,90];
model_translate = [-35,-25,-11.8];
model_scale = .9;

// Mold parameters
mold_width = 78;          // Measured along X axis
mold_height = 47;            // Measured along Y axis
mold_depth = 23;          // Measured along Z axis
mold_spacing = 3;             // Space between mold halves
```

(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134418/mold-8.png)

For model_rotate and model_translate the 3 variables are X, Y, and Z respectively. model_scale is a multiplier applied to the size of your original model.

Adjust the mold_width, height and depth as needed so that your model sits nicely in the middle of your mold. Don't forget to hit F5 after you make an adjustment to see the effect. When you have all the variables set right, your preview should look something like this:

(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134408/mold-9.png)
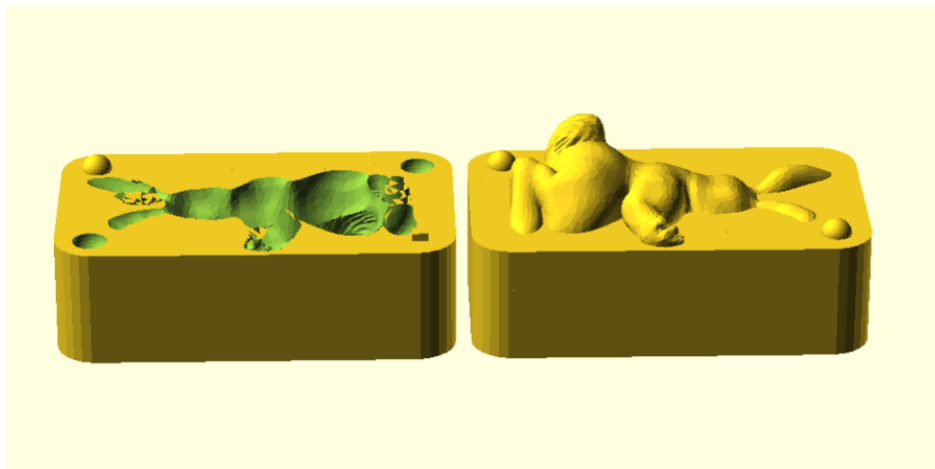
Once it looks right, you'll need to remove the 6 '//' comment markings we added above and hit F5 again. It should now look like this:

(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134356/mold-10.png)
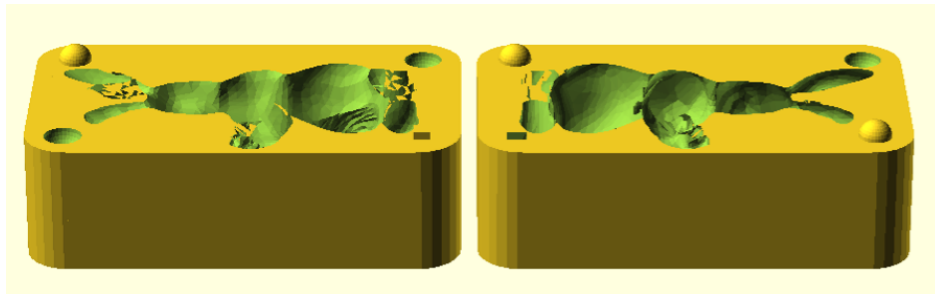
Your last step is to do a full render and export the STL. To do a render hit F6. This will take much longer than when you hit F5. Once the render is done, the model in the preview will look much clearer:

(https://blog-assets.pinshape.com/blog/wp-content/uploads/2015/05/02134347/mold-11.png)
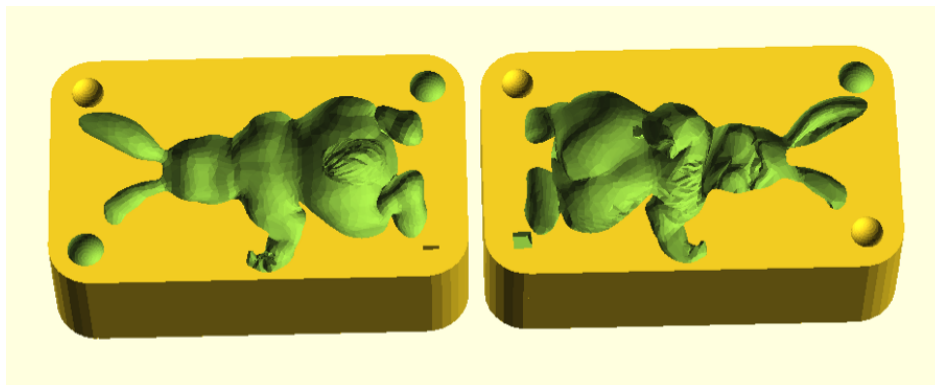
You can now select File->Export and Export as an STL.

Casting with Oogoo

The original instructions for making 1 Oogoo can be found here (http://www.instructables.com/id/How-To-Make-Your-Own-Sugru-Substitute/).

I've got a couple of tips for using Oogoo so that it will work well with these molds.

- Add essential oils. Just add a few drops of vanilla (or whatever other odour strikes your fancy) essential oil to the silicon before adding the corn starch . It will replace the yucky vinegar smell much quicker.
- Add a bit of oil-based paint even if you don't care about color. The extra oil from the paint and essential oil makes the Oogoo a bit thinner and delays the curing process a bit.

Here are my steps to making the mold:

1. Add the essential oils into the silicon and stir in the paint
2. Stir until paint is mixed evenly
3. Mix in corn starch. Add the same amount of cornstarch as silicon by volume. I did this one teaspoon at a time, stirring until it was well blended.
4. Smoosh the Oogoo into the mold. Add more than you think you'd need.

5. Squeeze the mold halves together as tightly as you can. Oogoo should push out from all four sides if you've added enough. The harder you squeeze and clamp the halves, the less flashing you'll have to clean up later.

In this picture, I'd already added the essential oil and was ~~adding the~~ paint into the silicon.

Leave your mess sitting around on the counter so you can poke at the leftover Oogoo every once in a while. When it's hard and rubbery, the stuff in your mold will be as well. Pull the mold apart carefully and enjoy your new minion. Make a thousand. Start an army....take over the world!

We'd love to hear from you!

Questions? Comments? Leave them for Dan below!

Related

(https://pinshape.com/blog/top-10-most-popular-3d-printing-articles-of-2015-on-pinshape/)
Top 10 Most Popular 3D Printing Articles of 2015 on Pinshape!
(https://pinshape.com/blog/top-10-most-popular-3d-printing-articles-of-2015-on-pinshape/)
January 4, 2016
In "Industry News"

(https://pinshape.com/blog/low-cost-injection-molding-using-3d-printing/)
Low-Cost Injection Molding Using 3D Printing
(https://pinshape.com/blog/low-cost-injection-molding-using-3d-printing/)
November 2, 2017
In "Pinshape News"

https://pinshape.com/blog/what-is-3d-printing/)
What is 3D Printing?
(https://pinshape.com/blog/what-is-3d-printing/)
July 14, 2015
In "Guides"

**PINSHAPE
(HTTPS://PINSHAPE.COM/BLOG/AUTHOR/LETMETELLYOUABOUT3DPRINTING/)**

Pinshape is a 3D printing community and marketplace where makers from all over the world can find and share their next great 3D print and help each other get the best results from their 3D printers.

24 Comments   **Pinshape Blog**   ● **Login** ⌄

♡ Recommend 2   ↱   Sort by Best ⌄
                          Share

👤 | Join the discussion… |

LOG IN WITH   OR SIGN UP WITH DISQUS ⑦

                         | Name |

Andrew Patton • 10 months ago
Any advice would be appreciated. I get an error at Rendering... Also can't find Line:326.

Rendering Polygon Mesh using CGAL...
ERROR: CGAL error in CGAL_Nef_polyhedron3(): CGAL ERROR: assertion violation! Expr: e->incident_sface() != SFace_const_handle() File: /opt/mxe/usr/x86_64-w64-mingw32.static/include/CGAL/Nef_S2/SM_const_decorator.h Line: 326
1
∧ | ∨   • Reply • Share ›

        Andrew Patton ➔ Andrew Patton • 10 months ago
        Fixed. Took awhile but I had to repair my model with multiple programs.
        ∧ | ∨ • Reply • Share ›

Gabriel Martins Raphael • a year ago
cool.. doesn't work. when press f6 the mold get only the outside block surface.. and don't the model negative.
1
∧ | ∨   • Reply • Share ›

        saeed1922 ➔ Gabriel Martins Raphael • 3 months ago
        same problem :(
        ∧ | ∨ • Reply • Share ›

Ultra Force • 3 years ago
Then you can use your two part mold to rotational cast it hollow with a RotoMAAK

SEARCH

search..   Su

PINSHAPE

Pinshape is a 3D printing community and marketplace where makers from all over the world can find and share their next great 3D print and help each other get the best results from their 3D printers.

NEWSLETTER

Sign up now to get the best 3D printing news & coolest new designs each week!

Your email address

Sign up

FOLLOW US!

(http://facebook.com/pinshape)(https://twitter.com/pinshape)(https://pinterest.com/pinshape)(http://instagram.com/pinshape3d/)

POPULAR POSTS

5 Interesting 3D Printer Controller Boards to Check Out! (https://pinshape.com/blog/3-interesting-3d-printer-controller-boards/)

FDM vs SLA: How does 3D Printing Technology Work? (https://pinshape.com/blog/fdm-vs-sla-how-does-3d-printer-tech-work/)

Low-Cost Injection Molding Using 3D Printing (https://pinshape.com/blog/low-cost-injection-molding-using-3d-printing/)

Learn How to Make Low-Cost Injection Molds With 3D Printing

Learn How To 3D Print With Nylon

How to 3D Print with Nylon (https://pinshape.com/blog/how-to-3d-print-with-nylon/)

Learn What Resolution Means in 3D Printing

4 Things You Need to Know About 3D Printing Resolution (https://pinshape.com/blog/4-things-you-need-to-know-about-3d-printing-resolution/)

RECENT POSTS

PINSHAPE FEATURED
Our Favourite Designs Of the Week
Design and Image by Louise Driggers

Pinshape Featured Designs - December 15th (https://pinshape.com/blog/pinshape-featured-designs-december-15th/)

Announcing the Winners of the Create to Educate Lesson Plan Contest
BROUGHT TO YOU BY
Ultimaker pinshape MatterHackers

Announcing the Create to Educate Lesson Plan Contest Winners! (https://pinshape.com/blog/announcing-create-educate-lesson-plan-contest-winners/)

PINSHAPE FEATURED
Our Favourite Designs Of the Week
Design and Image by jobsmolders

Pinshape Featured Designs - December 8th (https://pinshape.com/blog/pinshape-featured-designs-december-8th/)

Pinshape's Official 2017 Holiday Gift Guide (https://pinshape.com/blog/pinshapes-official-2017-holiday-gift-guide/)

Pinshape's 2017
Holiday Gift Guide

Design and Image by Print That Thing

Pinshape Featured Designs - December 1st
(https://pinshape.com/blog/pinshape-featured-designs-december-1st/)

PINSHAPE FEATURED

Our Favourite Designs
Of the Week

Design and Image by Adafruit

(https://pinshape.com/blog/3d-printing-for-charity-with-miniworld/)

(https://pinshape.timelapse-tuesdays-boneheads-raven-skull/)

(http://facebook/(twitter/(youtube/(pinterest/(google+/(instagram).com/pinshape3d/)