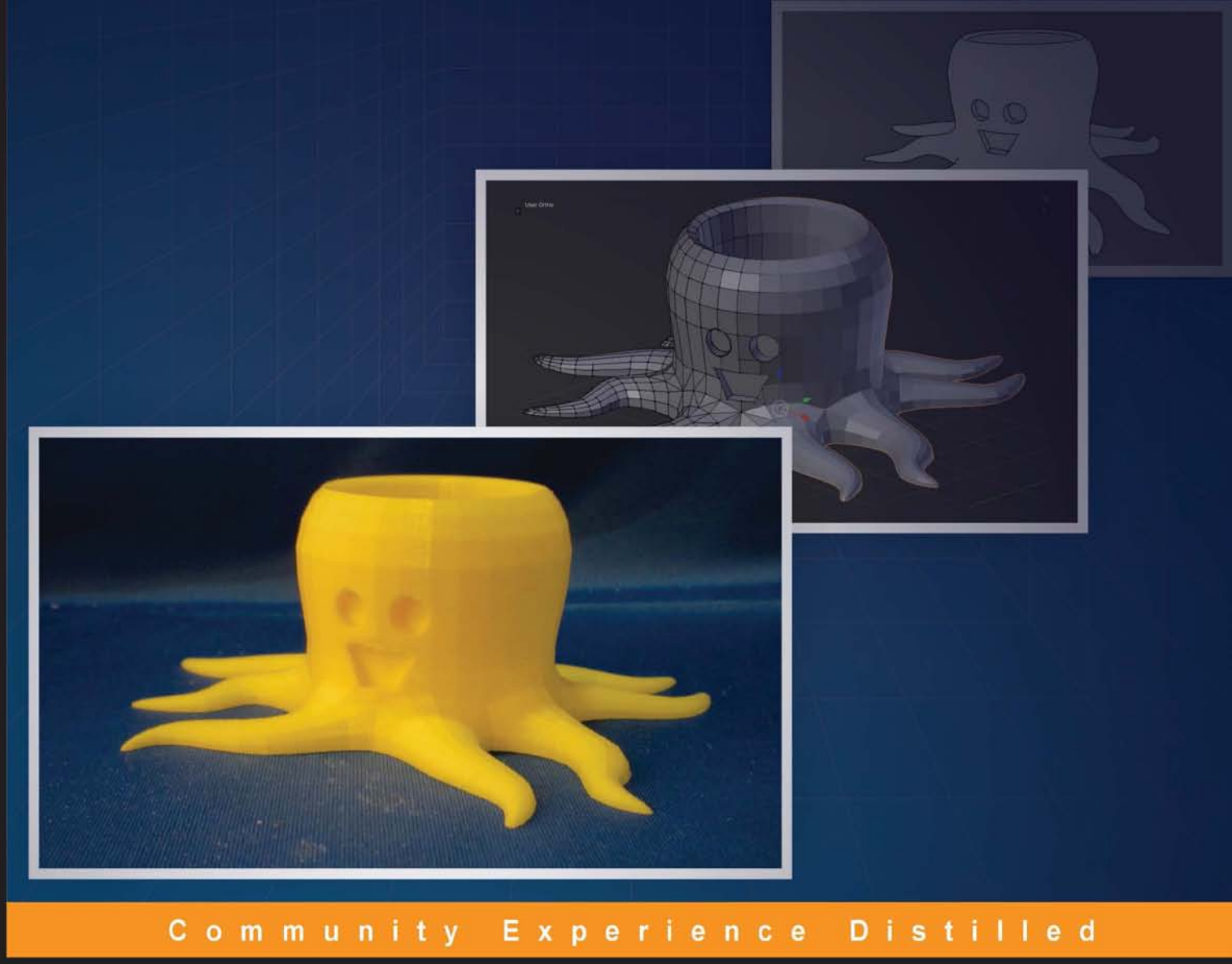


Community Experience Distilled

3D Printing Designs: Octopus Pencil Holder

Learn to design and 3D print organic and functional designs using Blender

Joe Larson

[PACKT] PUBLISHING open source* community experience distilled

3D Printing Designs: Octopus Pencil Holder

Learn to design and 3D print organic and functional designs using Blender

Joe Larson

[PACKT] PUBLISHING open source community experience distilled

BIRMINGHAM - MUMBAI

3D Printing Designs: Octopus Pencil Holder

First published: February 2016

Production reference: 1240216

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78588-517-4

www.packtpub.com

Credits

Author
Joe Larson

Reviewer
Marcus Ritland

Commissioning Editor
Edward Gordon

Acquisition Editor
Vinay Argekar

Content Development Editor
Shweta Pant

Technical Editor
Vishal Mewada

Copy Editor
Stuti Srivastava

Project Coordinator
Kinjal Bari

Proofreader
Safis Editing

Indexer
Priya Sane

Graphics
Kirk D'Penha

Production Coordinator
Shantanu N. Zagade

Cover Work
Shantanu N. Zagade

About the Author

Joe Larson is one part artist, one part mathematician, one part teacher, and one part technologist. It all started in his youth when he worked on a Commodore 64, doing BASIC programming and low-resolution digital art. As technology progressed, so did Joe's dabbling, eventually taking him to 3D modeling while in high school and college, and he temporarily pursued a degree in computer animation. He abandoned this field for the much more sensible goal of becoming a math teacher, which he accomplished when he taught 7th grade math in Colorado. He now works as an application programmer.

When Joe first heard about 3D printing, it took root to his mind, and he went back to dust off his 3D modeling skills. In 2012, he won a Makerbot Replicator 3D printer in the Tinkercad/Makerbot Chess challenge with a chess set that assembles into a robot. Since then, his designs on Thingiverse have been featured on Thingiverse, Gizmodo, Shapeways, Makezine, and other places. He currently maintains the blog `http://joesmakerbot.blogspot.in/`, documenting his adventures.

About the Reviewer

Marcus Ritland is a designer and 3D printing consultant in his small business, Denali 3D Design. Since 2008, he has been providing 3D modeling and 3D printing services as well as moderating the SketchUcation 3D printing forum (`http://sketchucation.com/`).

He has volunteered at a local makerspace, teaching SketchUp classes and leading 3D printing meetups. As an author of 3D Printing with SketchUp, he is currently on a quest to eliminate design-for-3D printing illiteracy.

Table of Contents

Preface

3D printers have arrived! Complex and beautiful objects are available at the touch of a button in our schools, libraries, or even our homes. If you have a 3D printer, learning how it works and how to design for it is the best way to be a part of this new industrial age. And the best part is that it doesn't cost a penny.

This book will teach you the things you need to know about 3D printers. Then, you will use the robust and free software, Blender, to follow step-by-step instructions through a planned project. This book is a part of a series of projects that will help you acquire the tools, techniques, and skills you need in order to make your own projects to print yourself on a 3D printer near you and share them with others online to print around the world.

What this book covers

The project in this book, octopus pencil holder, involves simple selection techniques that are unique to edit mode, modification commands in edit mode, and applying modifiers to soften and combine shapes. This technique alone can be used to create an unlimited number of cool things, once mastered.

What you need for this book

Blender's minimum system requirements are as follows:

- 32-bit dual core 2 GHz CPU with SSE2 support
- 2 GB RAM
- 24 bits 1280 × 768 display
- Mouse or trackpad
- OpenGL 2.1 compatible graphics with 512 MB RAM

Who this book is for

This book is for anyone with an interest in 3D printing and some basic computer skills. Whether you own a 3D printer or not, you can design for them. You will need Blender, a free 3D tool, and this book. With a little creativity, one day, you'll hold something designed on the computer in your hands.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "The first one is going to be the `Extrude` operator."

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Next, add a cylinder and an object to the scene. For this, add (*Shift* + *A*) a new object by navigating to **Mesh** | **Cylinder**."

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail `feedback@packtpub.com`, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from `https://www.packtpub.com/sites/default/files/downloads/3DPrintingDesigns_OctopusPencilHolder_ColorImages.pdf`.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to `https://www.packtpub.com/books/content/support` and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at `questions@packtpub.com`, and we will do our best to address the problem.

1
Octopus Pencil Holder

3D printing makes it easy to combine forms and functions. Why have just a pencil holder when you can have a pencil holder that looks like, say, a cute octopus? This project is an excellent starting project because it demonstrates a simple but versatile modeling technique that involves editing a simple mesh and smoothing it in order to go into more detail.

This project will involve simple selection techniques that are unique to the **Edit Mode**, modification commands in the edit mode, and applying modifiers to soften and combine shapes. This technique alone can be used to created unlimited cool things once mastered.

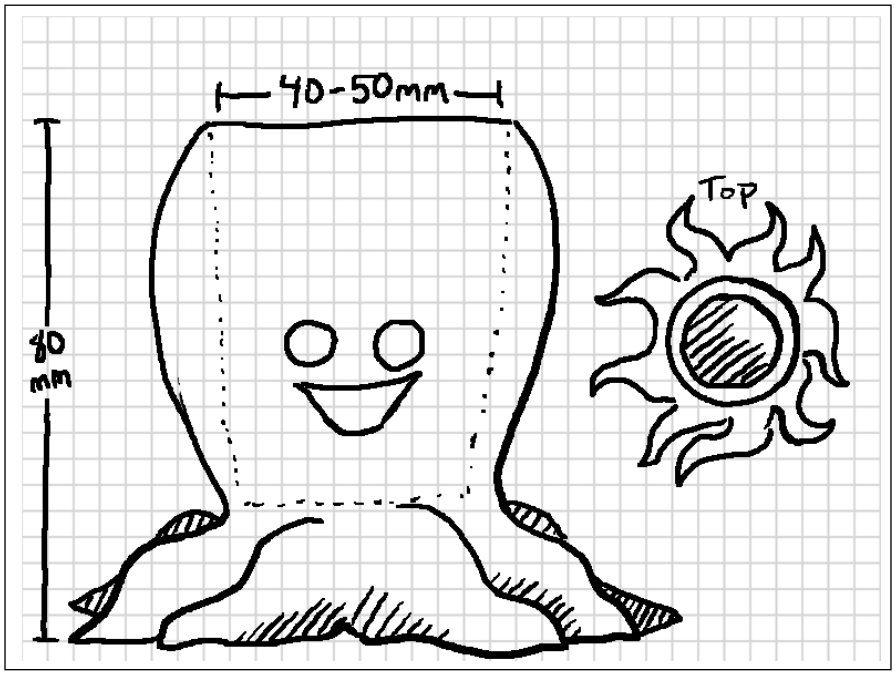
In this book, we will cover the following topics:

- Planning the project
- The first basic shape
- Smoothing the mesh with modifiers
- Bending the tentacles
- Flattening the bottom
- Renaming objects
- Adding a pencil cup
- Adding a face
- Finishing touches

Planning the project

A pencil holder is basically a cup with a sturdy base that can be used to hold objects taller than itself, such as pens, pencils, and other items that might otherwise clutter up a desk. A cute cartoony octopus' tentacles will provide the perfect base and prevent the cup from tipping over.

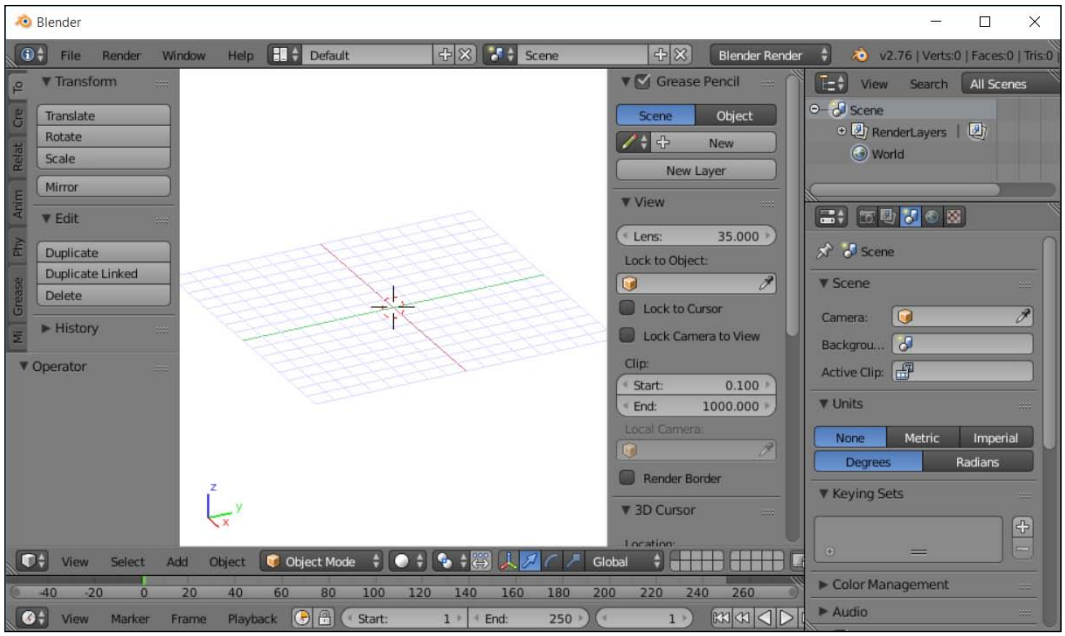
The hole in the cup should be roughly cylindrical, and about 40-50 mm wide at the base. The whole thing should stand at least 80 mm tall. And, of course, it needs a cute and friendly face; something like this:



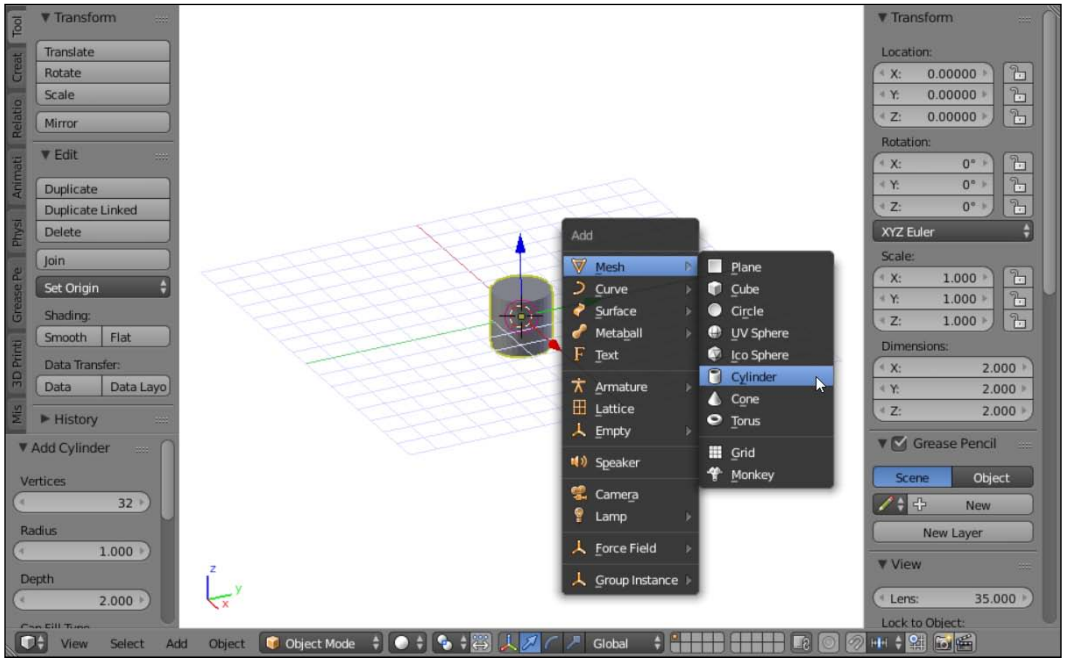
The basic shape

Let's get ready; it's time to get to work:

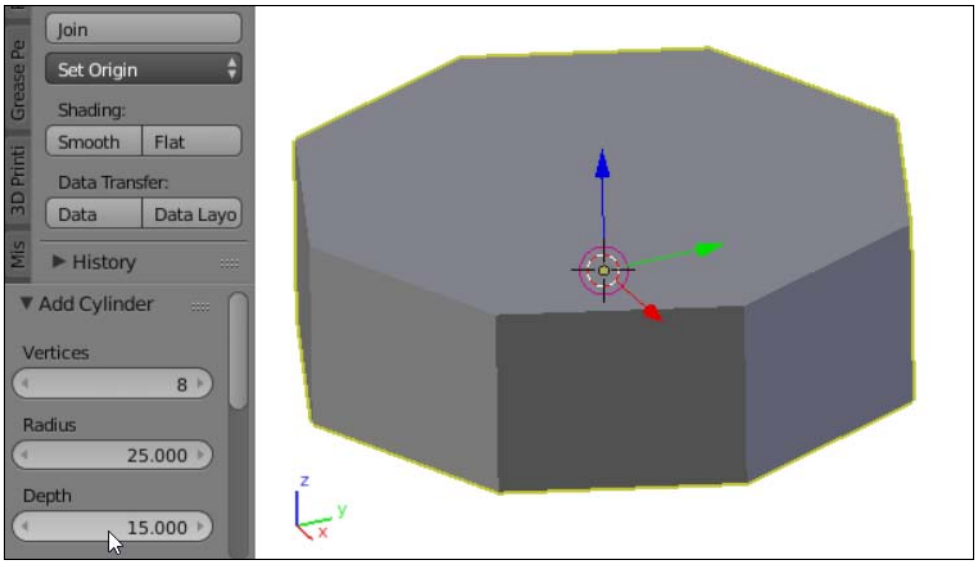
1. Open **Blender**, select everything (*A*) in the default scene and clear (*X*) the scene:



2. Next, add a cylinder and an object to the scene. For this, add (*Shift* + *A*) a new object by navigating to **Mesh** | **Cylinder**:



3. Immediately after adding the cylinder and before clicking on anything else, in the **Tool Shelf** operator, add parameters for **Add Cylinder** and change the number of **Vertices** from **32** to **8**. Change the **Radius** field to **25** and the **Depth** to **15**:

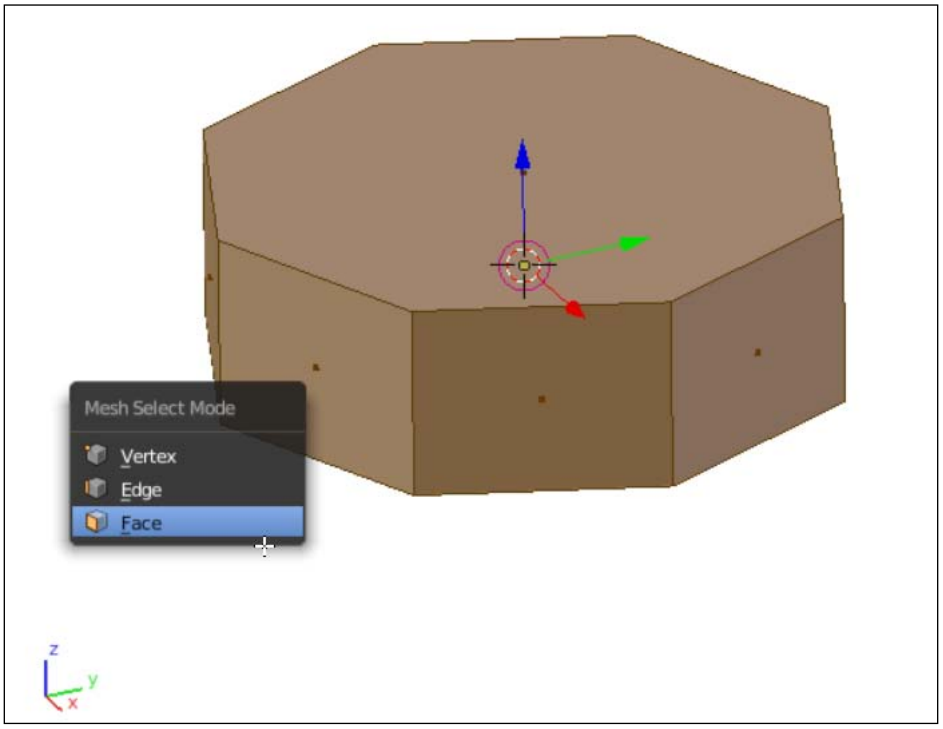


Editing the basic shape

This project is going to take advantage of several powerful editing tools that Blender provides. The first one is going to be the `Extrude` operator. Extruding takes its name from the process of creating things in real life, but in 3D modeling, extruding takes a selected part of an existing model and creates new geometry on the edge of the selected parts so that the original can be moved away but remain attached to where it came from. The result is a new shape that can then be edited.

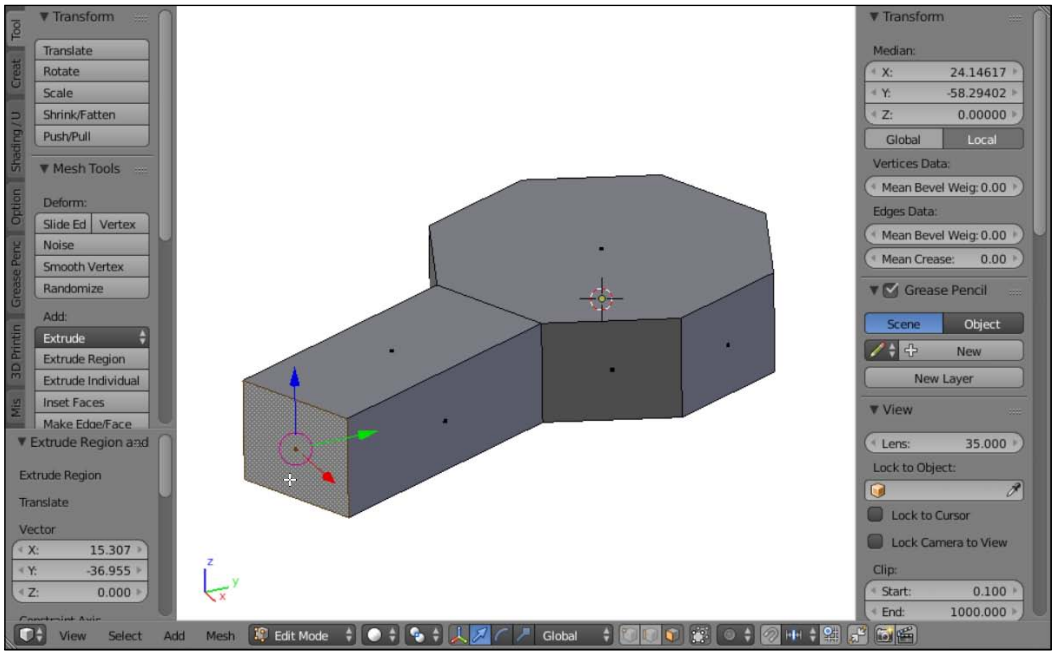
Extruding is a very powerful tool that's used to alter the shape of an object and create new faces that can be extruded themselves:

1. Enter **Edit Mode** (*Tab*) and switch to face the **Select Mode** (*Ctrl* + *Tab*):



2. Deselect all faces (*A*). Then, select one of the vertical sides of the cylinder.
3. Extrude it either by navigating to **Mesh** | **Extrude** | **Region** in the **3D View** menu or pressing *E* on the keyboard.

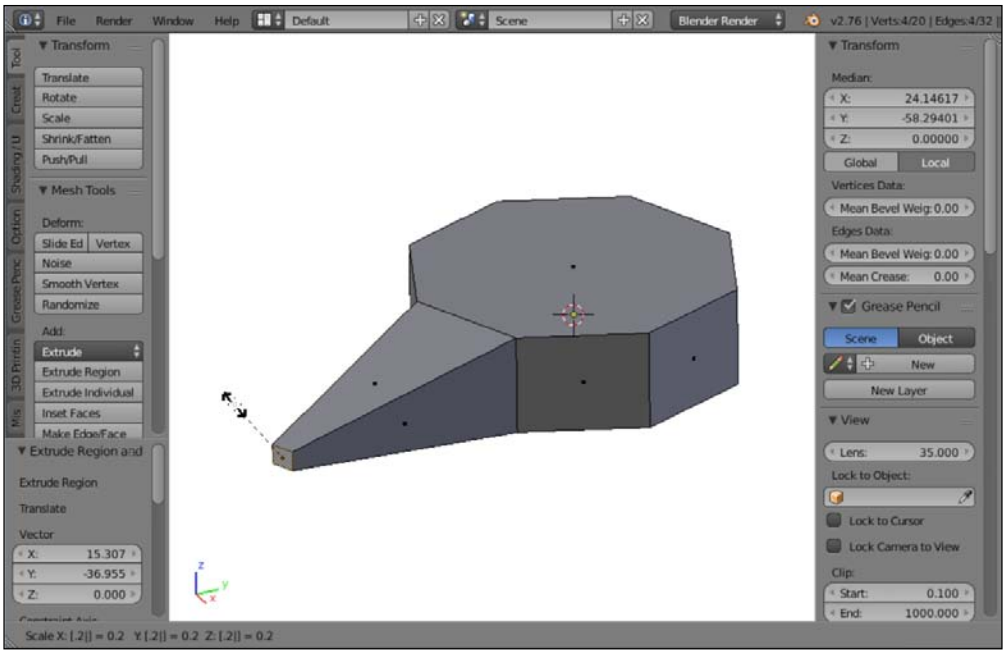
4. Extrude the face about 40 mm by moving the mouse or typing **40** on the keyboard:



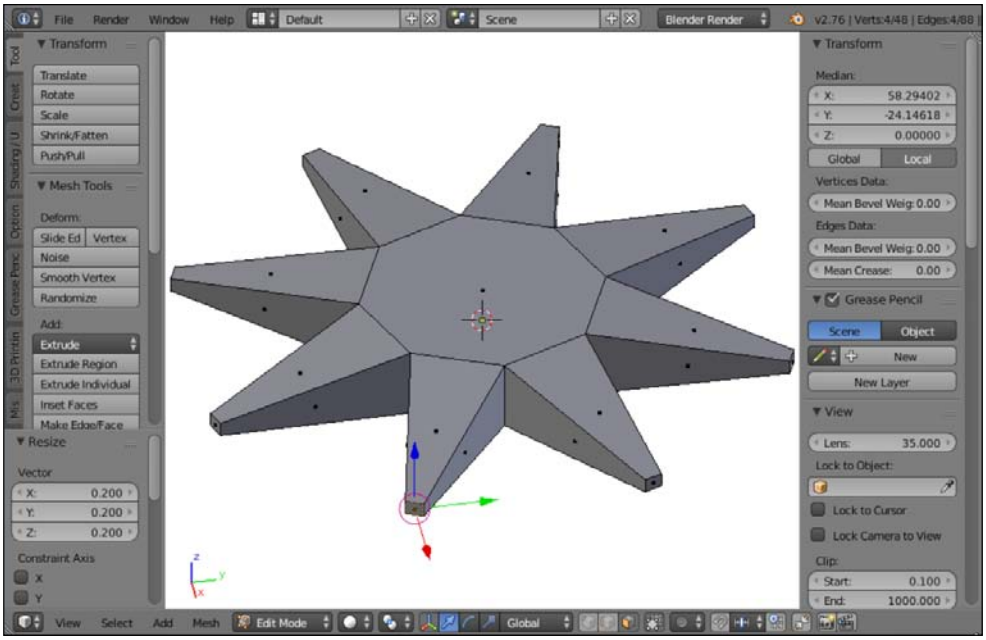
5. Press *Enter* or click on the select mouse button to complete the **Extrude** action.

Like all actions in Blender, if a mistake is made in the process of extruding, press *Esc* or click on the right mouse button to cancel the action. If a mistake is made after this, undoing the action with *Ctrl* + *Z* is always possible.

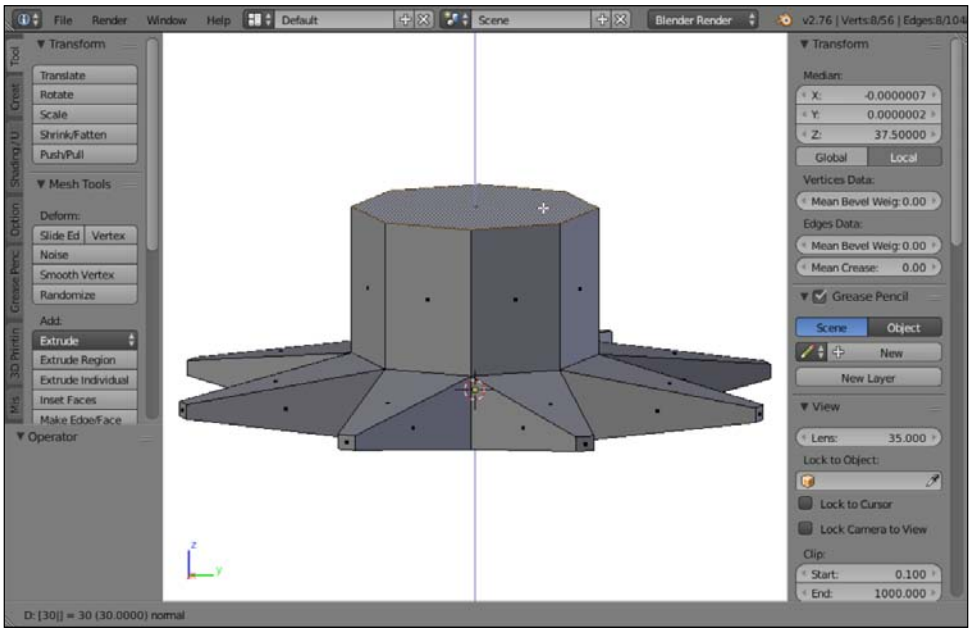
6. Then, scale the face (*S*) down to about 20% (**0.2**) in order to create a tentacle:



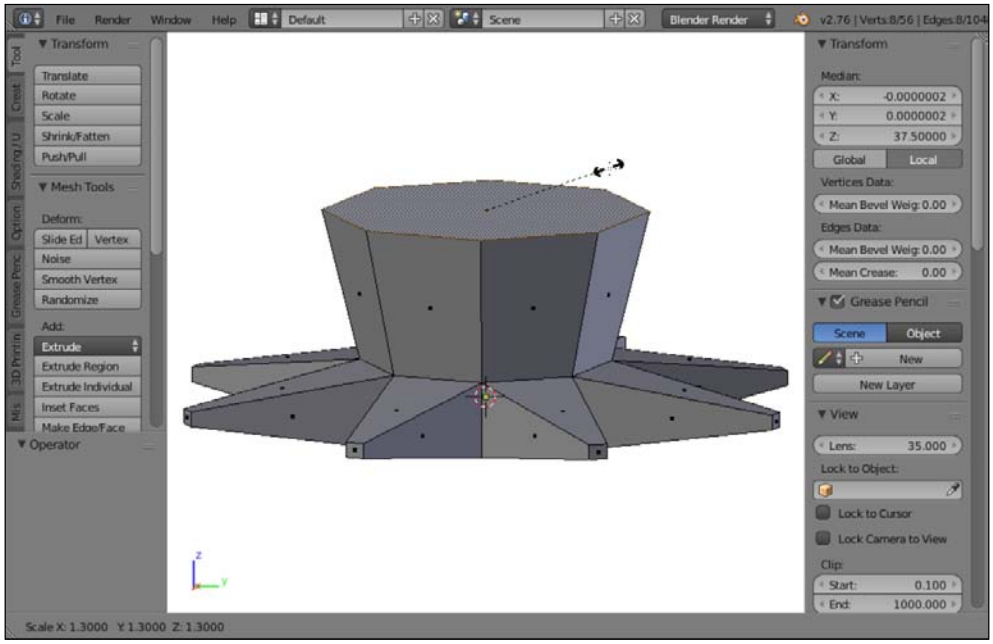
7. Repeat the extruding and scaling process with the other seven vertical faces of the cylinder to create all eight tentacles:



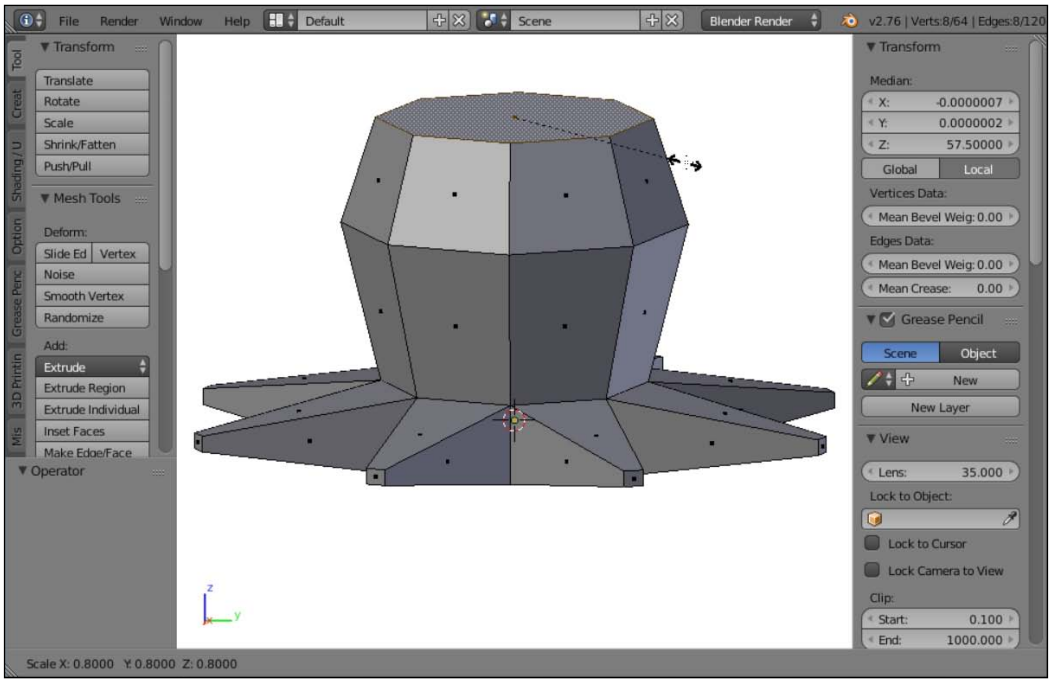
8. Select the top face of the cylinder and extrude (*E*) it about 30 mm:



9. Then, scale (*S*) it up just a little bit to make the head bulbous:



10. Extrude (*E*) the top again—this time, about 20 mm and—and scale (*S*) it in order to give the top a more rounded shape:

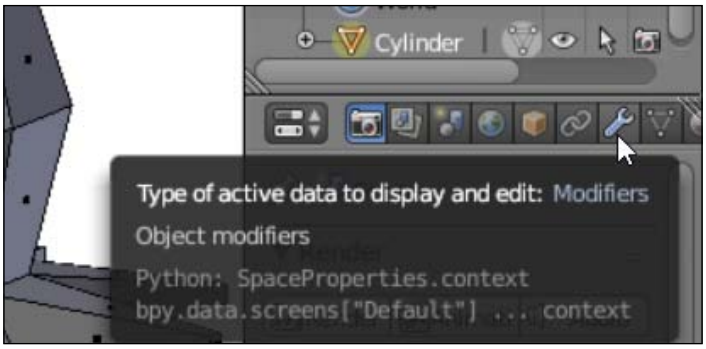


Now, the cylinder has been changed into something more of an octopus-like shape. And it was mostly accomplished with the `Extrude` command, a truly powerful tool used to modify the shape of an object.

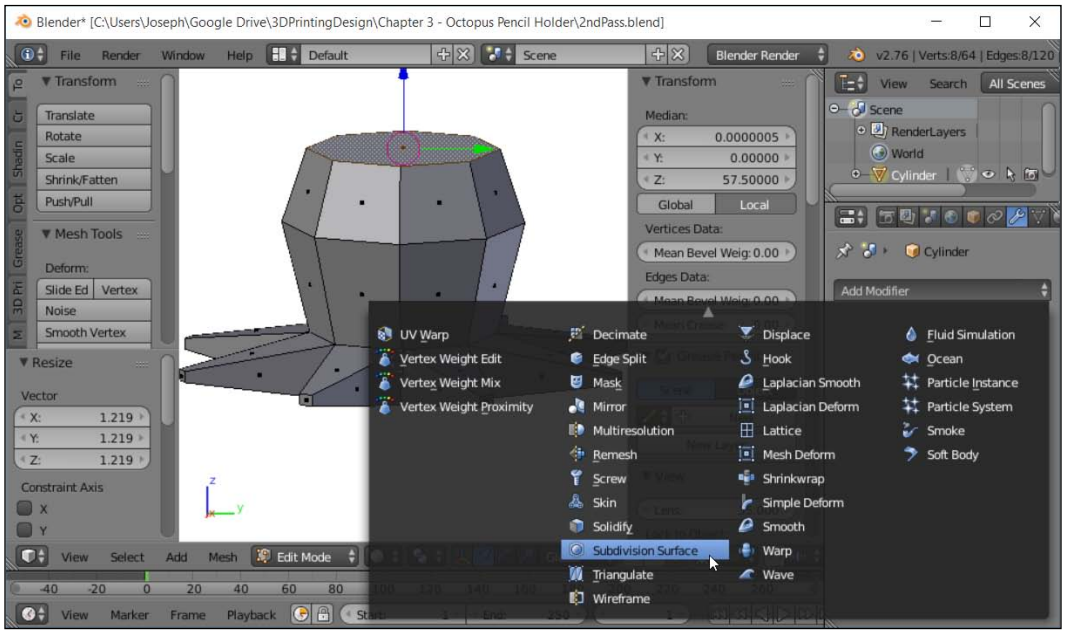
Smoothing the mesh with modifiers

This blocky octopus is fine for editing, but it needs to be much smoother for the final result. Fortunately, there's a way to increase the smoothness of the model while retaining the simple geometry, which is easy to edit. This is done by adding a **Subdivision Surface** modifier to the object:

1. To add a **Subdivision Surface** modifier, click on the **Modifiers** tab in the **Properties** panel (the one that looks like a wrench):



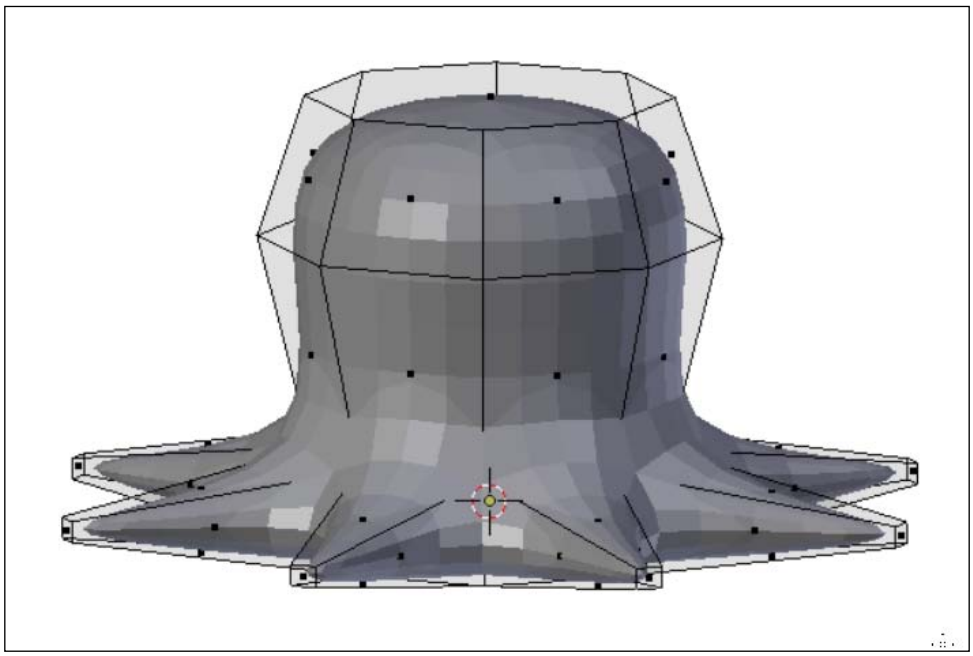
2. Click on the **Add Modifier** button, and in the menu that appears, choose **Subdivision Surface** from the list:



With the **Subdivision Surface** modifier, the shape looks much smoother. While in the edit mode, it's clear that the original geometry is still there and acts as sort of a cage that defines the shape of the smoothed mesh. As long as the modifier isn't applied, the simple geometry can be kept for editing.

Changing the **View** setting in the modifier will affect how smooth the mesh will be. Higher values will create more complex geometry that will look smoother, but they will slow down the computer more. There comes a point where increasing this setting won't have any noticeable effect. Generally, it's best to keep this setting high enough to have a good effect on the shape but low enough to not slow down the computer.

3. For now, set **View** to 2; then, the object will look like what's shown in the following screenshot:



This isn't the final form, but it will get better. If looking at the smoothed mesh while working on the simpler mesh is confusing, the modifier can be temporarily turned off by pressing the eyeball icon on the modifier.

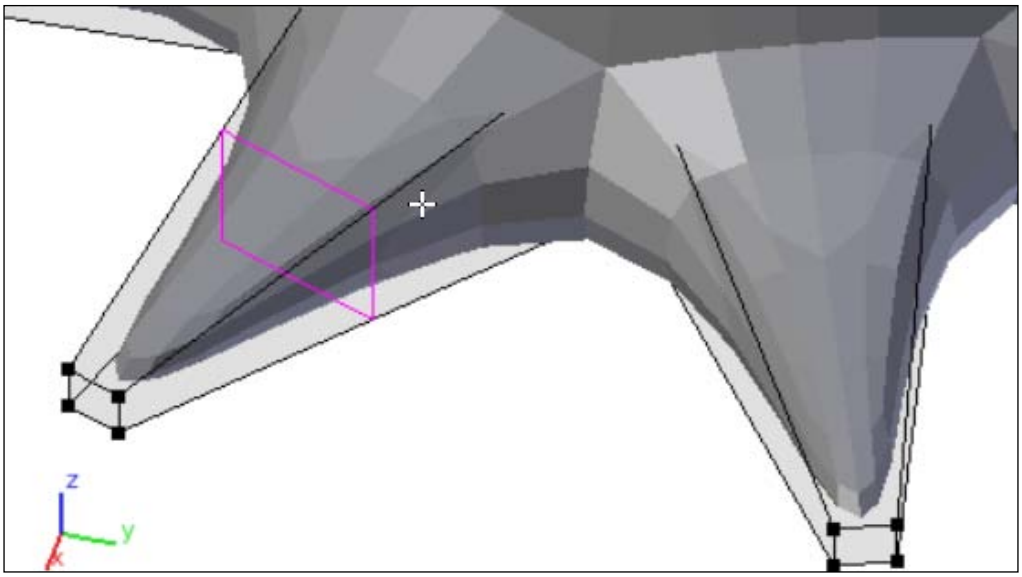
Bending the tentacles

Now, it's time to begin adding some details to the tentacles and give the model some personality.

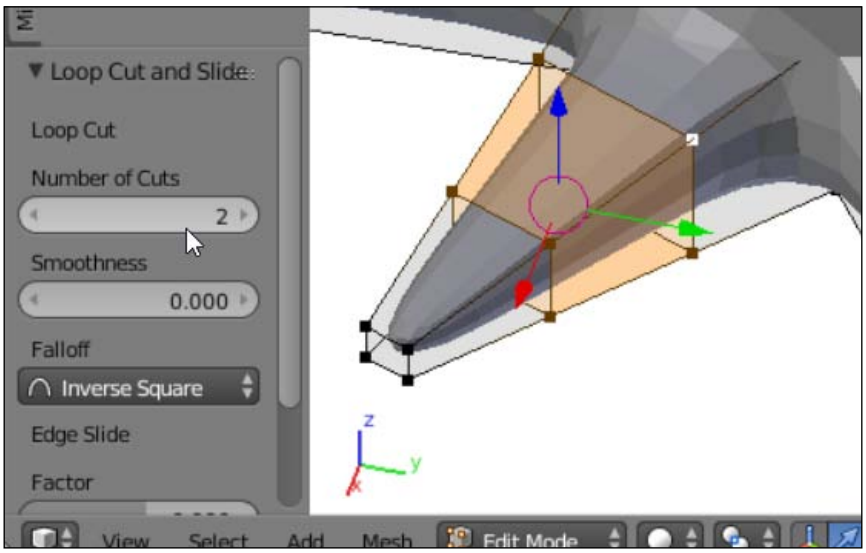
Before moving ahead, let's take a look at another powerful tool when editing meshes, which is called **loop cut** or **loop subdivide**, which will be used while bending the tentacles. Loop cut adds points in the middle of an edge and all around a portion of the geometry. These points can then be transformed. Loop cut only works with edges or vertexes' select mode and will automatically switch modes when using them.

To perform a loop cut, execute the following steps:

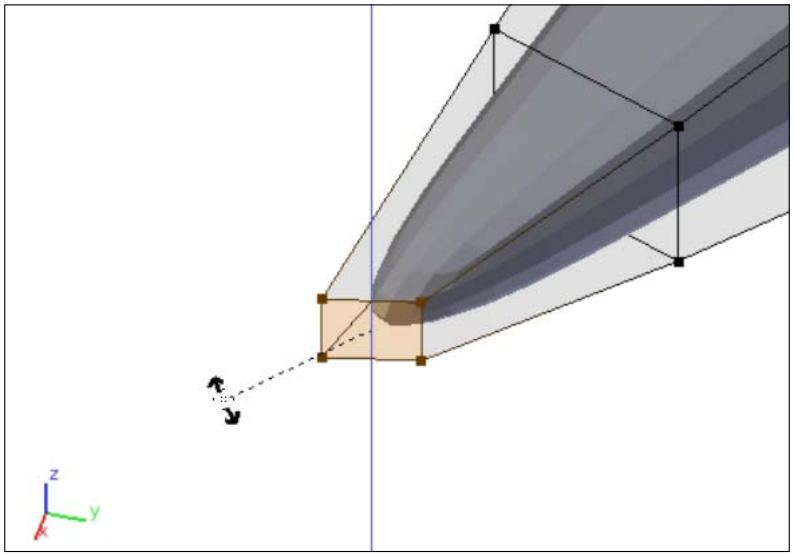
1. Navigate to **Mesh** | **Edges** | **Loop Subdivide** from the **3D View** menu, or press *Ctrl* + *R*.
2. Then, move the mouse pointer near an edge, and the loop cut suggestion will be suggested.
3. When the loop cut is where desired, click on the select mouse button or press *Enter*. At this point, the loop isn't set and can still be slid back and forth in order to decide an exact location.
4. When it is where desired, press *Enter* or the select mouse button again in order to finish the operation. At any time, the operation can be canceled by pressing the right mouse button or pressing the *Esc* key.
5. Now that we know how to perform a loop cut, let's go for why we are here, that is, bending the tentacles.
6. While still in the edit mode, switch to the vertex edit mode (*Ctrl* + *Tab*). Add a loop cut (*Ctrl* + *R*) to one of the tentacles. Place the cut at the default location in the middle of the tentacle:



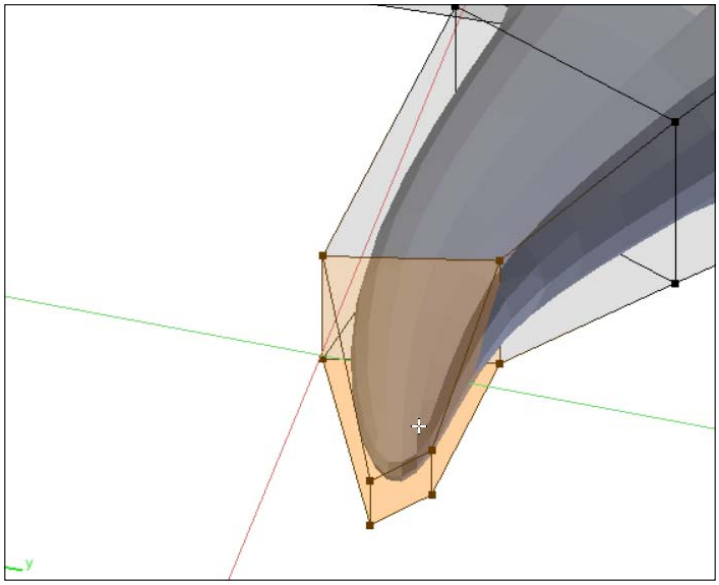
7. After setting the loop cut, change the operator settings at the bottom of **Tool Box** and then change **Number of Cuts** to **2**:



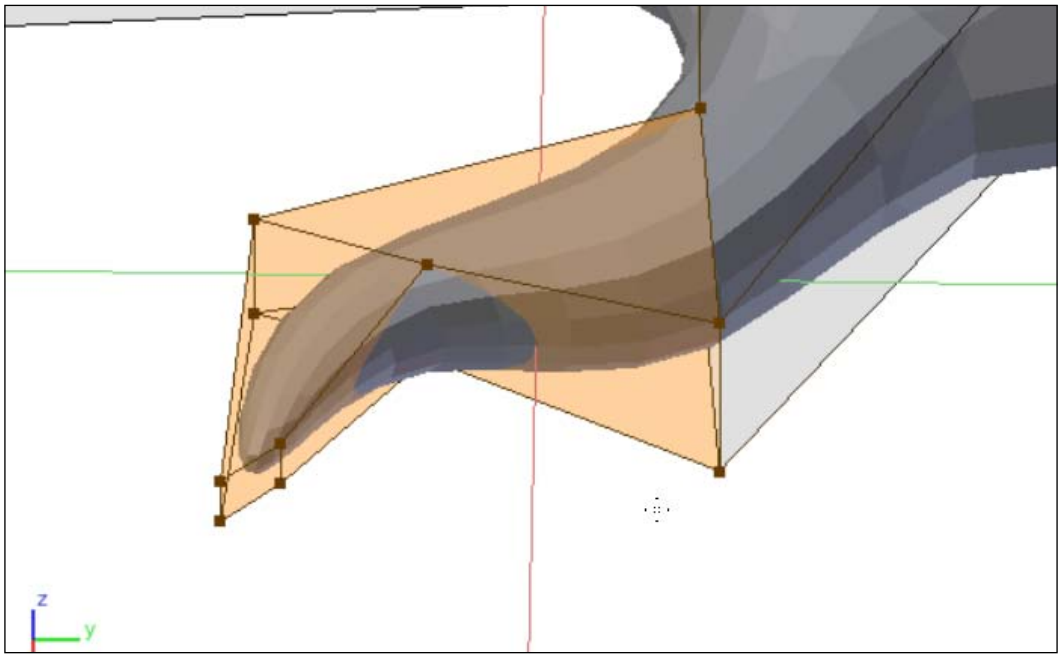
8. Deselect all points (*A*) and select the points at the end of the tentacle. Rotate (*R*) the points around the *z* axis (*Z*) and move (*G*) them along the *x* and *y* axes (*Shift* + *Z*) to bend the tentacle a bit:



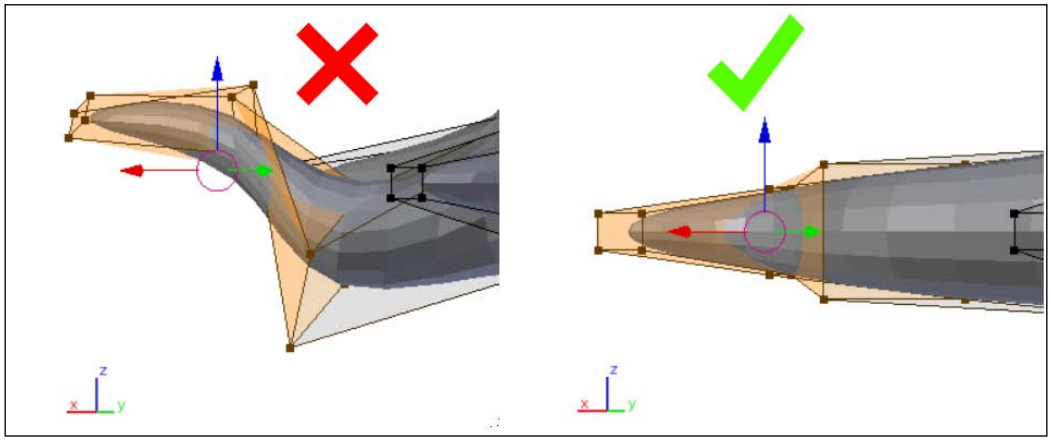
9. Expand the selection (*Ctrl* + NumPad + the (**+**) sign). Again, rotate (*R*) the selection around the *z* axis (*Z*) and move (*G*) it along the *x* and *y* axes (*Shift* + *Z*) to bend the tentacle a bit more:



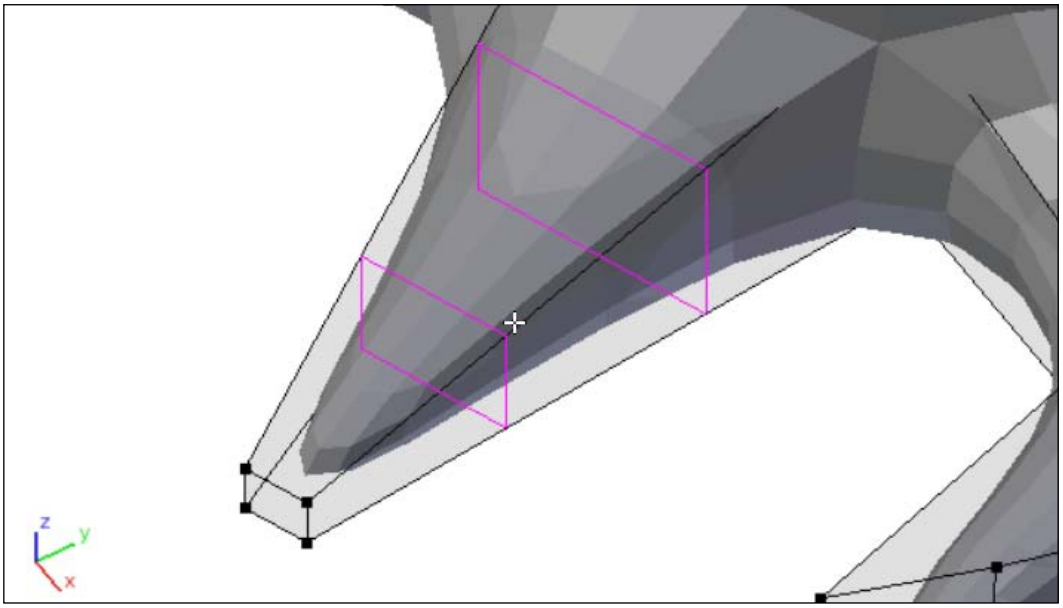
10. Expand the selection again. Rotate (*R*) and move (*G*) the tentacle, being careful to constrain the movement:



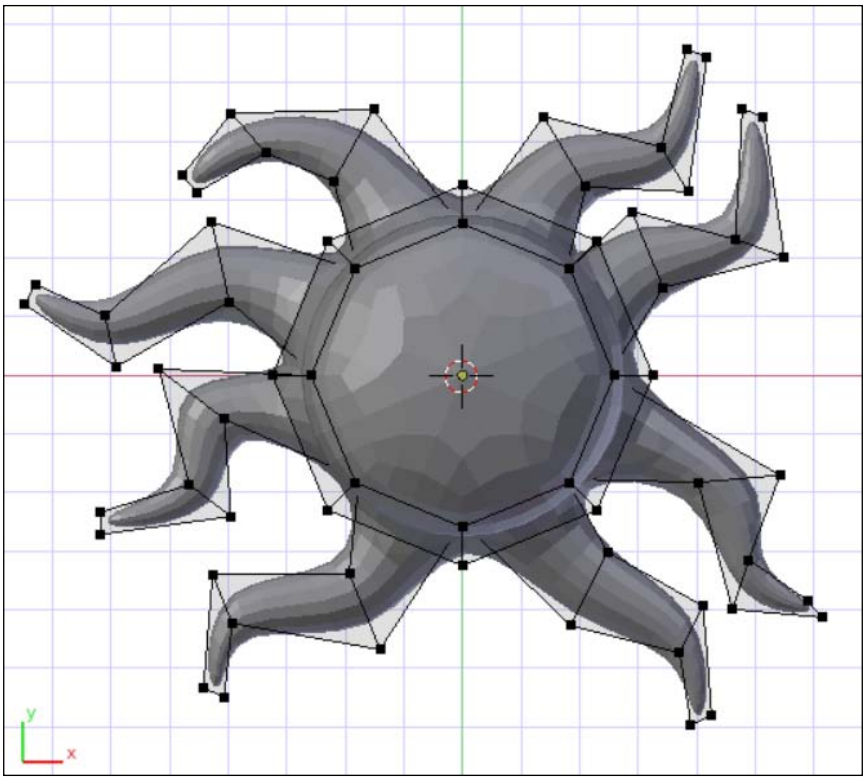
11. Check your tentacle. Adjust your view and make sure that from the front or side views, the twisty tentacle is still on the same level as the other tentacles and not twisted up and down. If it isn't, then your movements weren't constrained properly:



12. If not correct, then undo (*Ctrl* + *Z*) your movements and try again. There's a way to ensure that the movements are constrained properly. You can do this easily by doing all your transformation while in the top view (NumPad *7*). It may be necessary to select points in the **Wireframe view** (*Z*) to get all the points and not just the ones on top.
13. Move to the next tentacle and cut it through loops. This time, while the loop location is being displayed, before clicking on the mouse or pressing *Enter* the first time, try to press the *2* key to quickly change the number of cuts to **2**. This is a shortcut method that can be used optionally instead of changing the parameters after the cut:



14. Work around, cutting, selecting, moving, and rotating each tentacle. Give each one a different twist, being careful not to overlap them. If the tentacles overlap, the model won't print properly when exported:



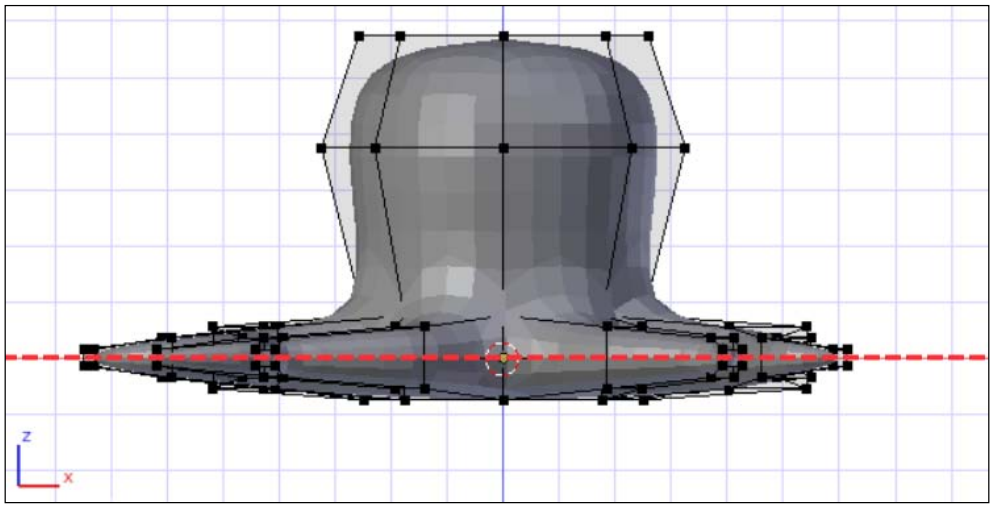
Your octopus doesn't need to look exactly like this. Make it your own. When all the tentacles have been detailed, it should look much more like an octopus.

Some 3D printers won't be able to print this model if it's too big, so keep those tentacles tucked in and don't let them fly too far. The smaller 3D printer build areas are about 150 mm or 6 inches across. If the total width of your model is more than that, you'll need to think about the printer you're designing this for.

Flattening the bottom

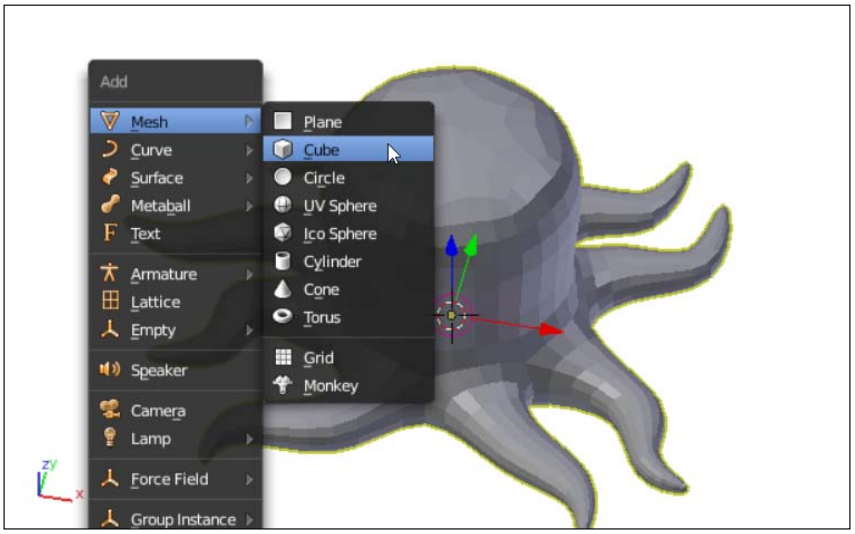
A friendly 3D print needs to have a solid, flat base. There is more than one way the bottom of a model can be flattened. For one, the geometry can be edited to be flat. Alternatively, a floor object can be created and cut out of the object using a Boolean modifier. Since this project is using the `subdivision surface` modifier, editing the geometry can be complicated. So, the `floor` method will be used in this project.

Looking at the model from the front orthographic view (NumPad *1* and NumPad *5*), it's clear that the parts of the tentacles that sit below the *x* and *y* axes (the red or green line depending on the view) is where the model needs to be cut off in order to create a flat, printable base:

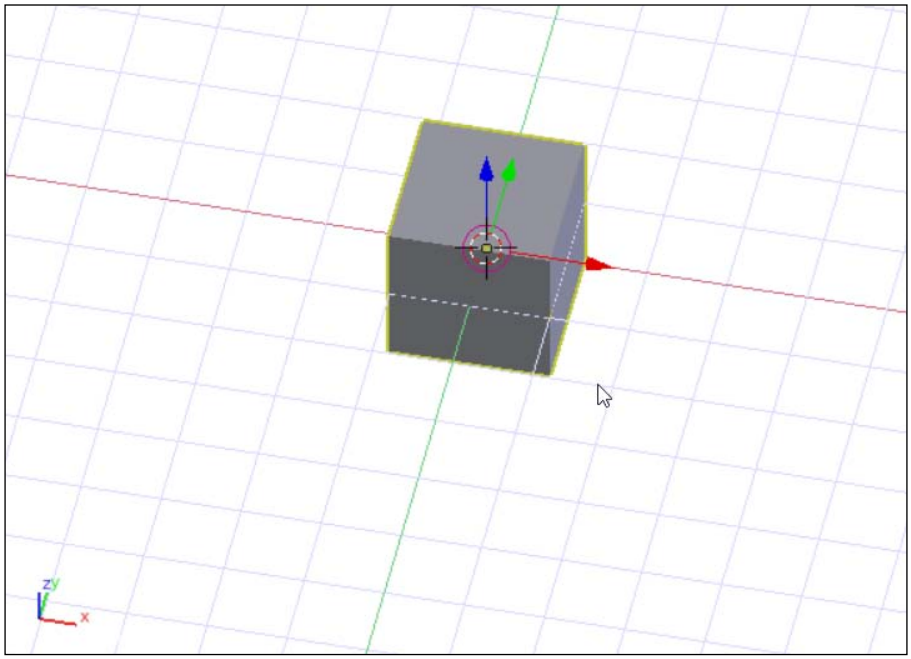


So, let's get started:

1. First, exit the **Edit Mode** (*Tab*).
2. Ensure that the 3D cursor is at the **3D View** origin point (*Shift* + *C*). Then, create a cube (*Shift* + *A*) by navigating to **Mesh** | **Cube**:

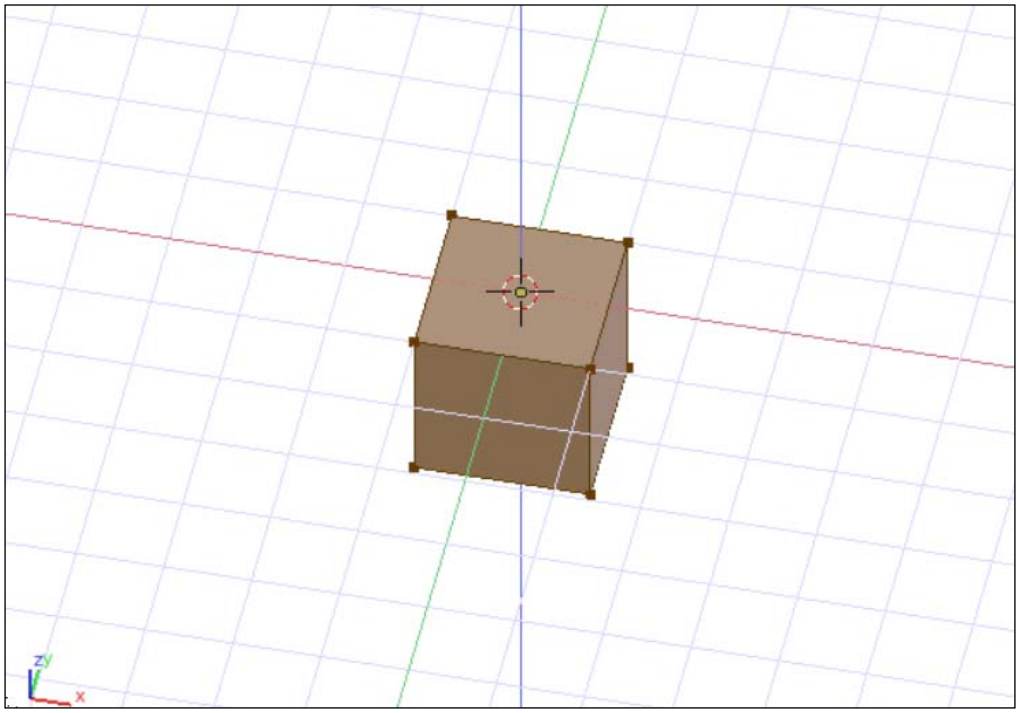


3. The cube cannot be seen in the solid view because it's completely inside the octopus, so with the cube still selected, switch to the local view (NumPad /) to view the cube by itself:

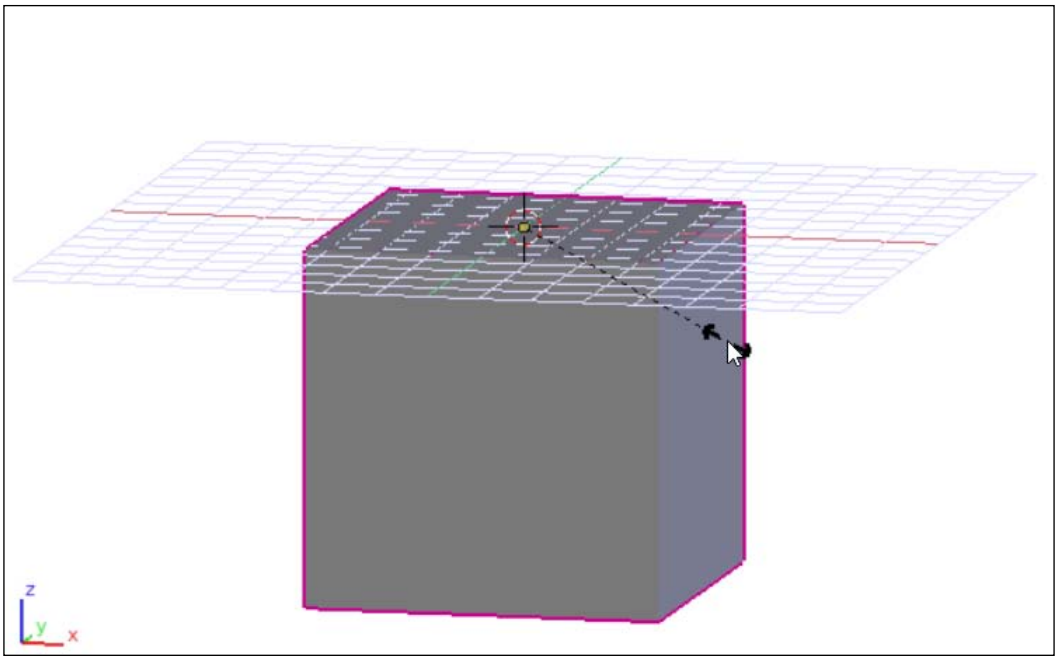


The cube is half above and half below the origin. Fortunately, there's an easy way to create a floor and ensure that no matter how it's scaled, the top remains on the *XY* plane.

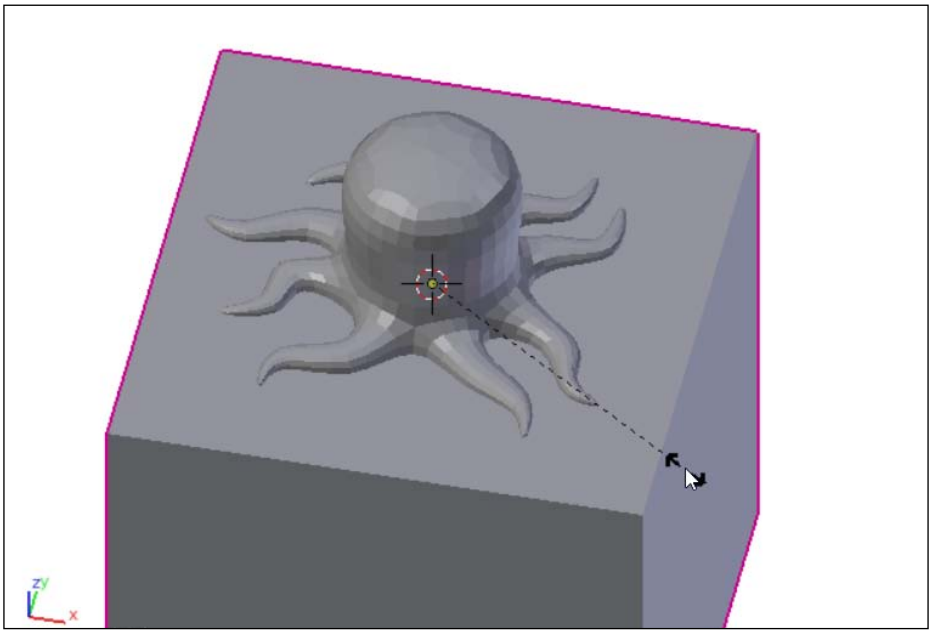
4. In the **Edit Mode** (*Tab*), with all points selected (which should be selected by default; *A* is the hotkey if they're not), move (*G*) along the *z* axis (*Z*) by `1` unit (*-1*).



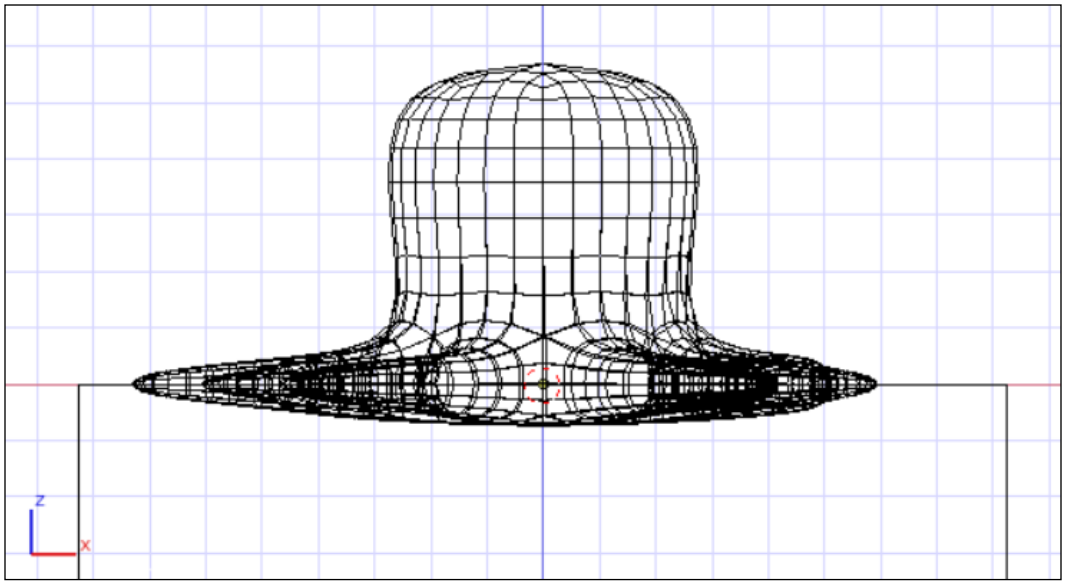
5. Then, exit the **Edit Mode** (*Tab*) and scale (*S*) the cube. Note that the top remains on the *XY* plane. This is because object transformations are made in relation to the object's origin. When the points were all moved in the **Edit Mode**, the origin wasn't affected. So now, when scaling the top, because it's in line with the cube's origin, it remains on the same plane:



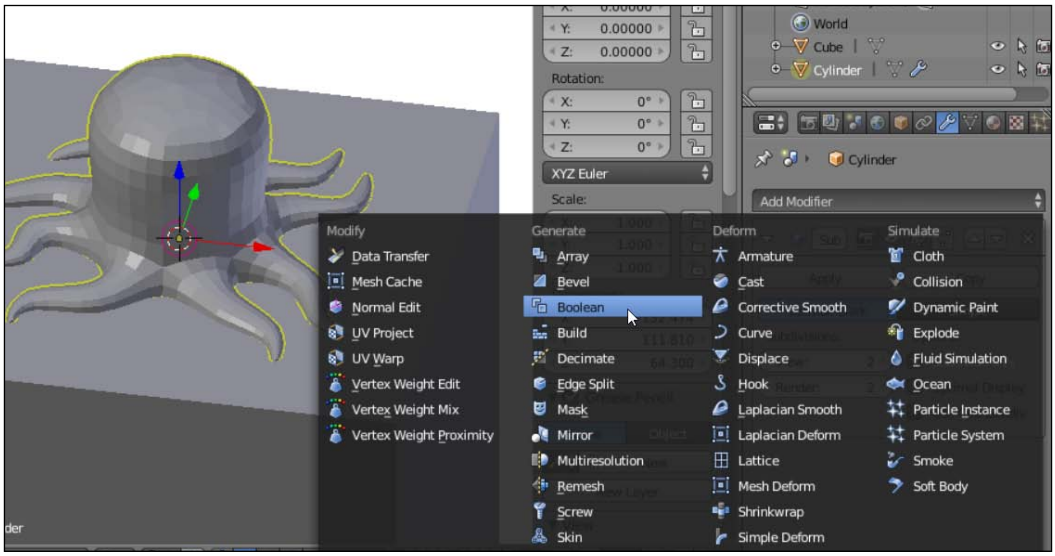
6. Exit the local view (NumPad /) and scale (*S*) the cube until it covers the bottom of the octopus body:



In the **Solid** view (Z), it may look like the octopus has a flat bottom, but in the **Wireframe View**, (Z), it is clear that the cube is only hiding the bottom part. For the final model, the bottom needs to actually be flat:

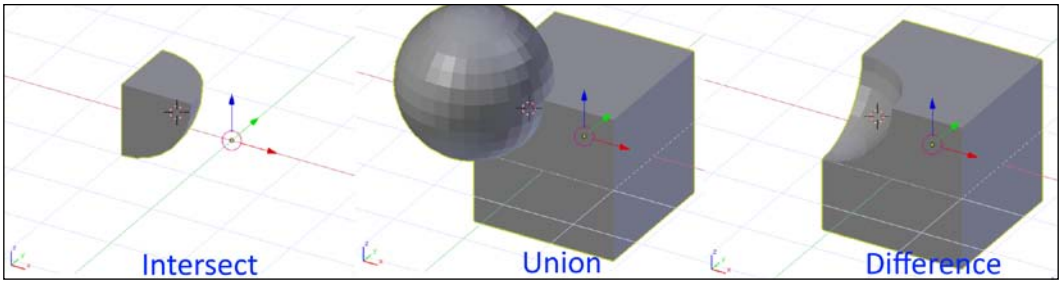


7. Select the octopus body again, and in the **Modifier** tab, add a Boolean modifier:

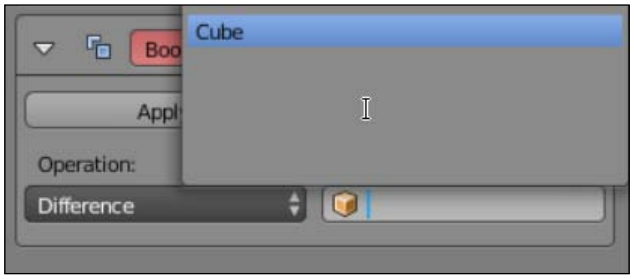


The Boolean modifier has the option of combining the selected object with another object in a number of ways; **Intersect**, **Union**, and **Difference**. **Union** joins the two object, so that they become one. **Difference** cuts the second object out of the first, and **Intersect** leaves only the part where the two objects overlapped. Boolean is a powerful tool and it's good to be aware of how to use it.

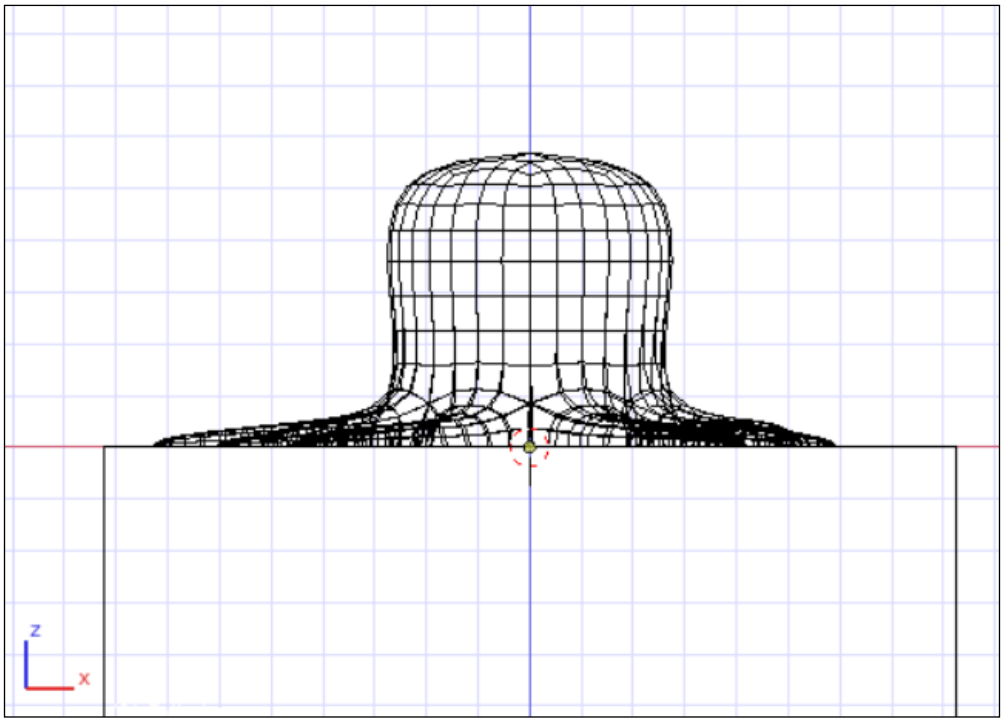
The following screenshot shows the three ways to combine selected objects with other objects:



1. Change the **Operation** value of the Boolean modifier to **Difference**. Click on the **Object** text field and choose **Cube** from the list that pops up:

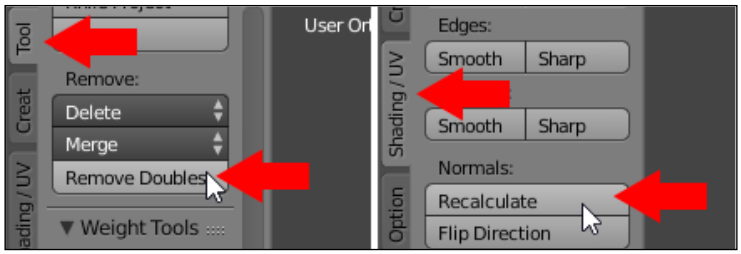


2. Now the octopus body actually has a flat bottom that's suitable for 3D printing. This can be confirmed in the **Wireframe** view (*Z*):



The Boolean modifier is one of the most important tools in Blender that's used to create 3D printable objects. However, it can also be one of the most frustrating to use. The Boolean modifier demands that a mesh be clean and free from holes or problems. Even things that aren't obvious while looking at the mesh, such as duplicate points or flipped faces, can cause problems with the Boolean operation. If your Boolean operation isn't working, here are a few steps you can perform in order to try to fix it:

1. In the **Edit Mode**, select all the points (*A*), and in **Tool Shelf** (*T*), locate the **Remove Doubles** button in the **Tools** tab and click on it.
2. With all points still selected, switch to the **Shading/UV** tab, locate the **Recalculate** button, and click on it:

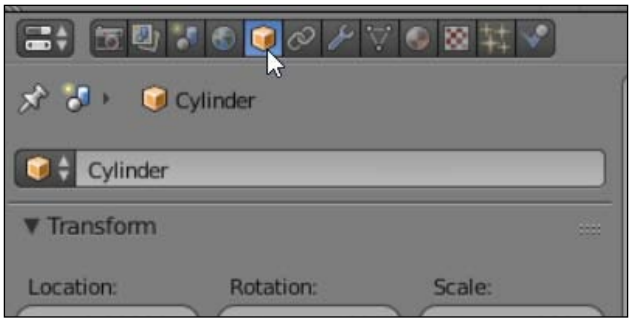


3. Now, exit the **Edit Mode**, and the Boolean modifier may work in many cases. If these fixes don't make the Boolean modifier work, then the best option may be to start over again, making sure that you perform all the steps correctly.

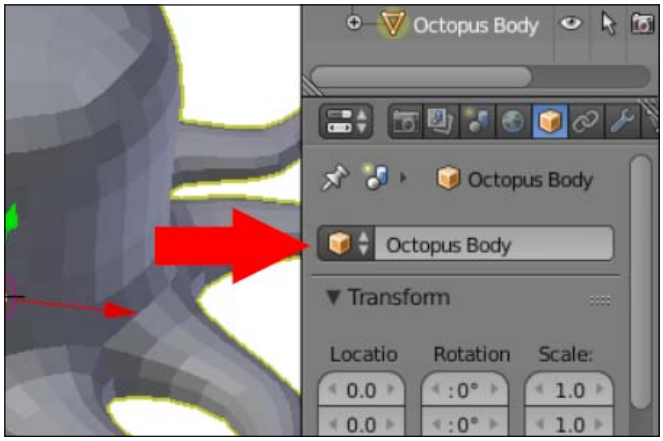
Renaming objects

Now that there is more than one object in the scene, leaving the objects' names as the names of the basic shapes they started as can be confusing, especially when other objects enter the scene. It's best practice to name objects as something more descriptive. Let's see how to rename an object.

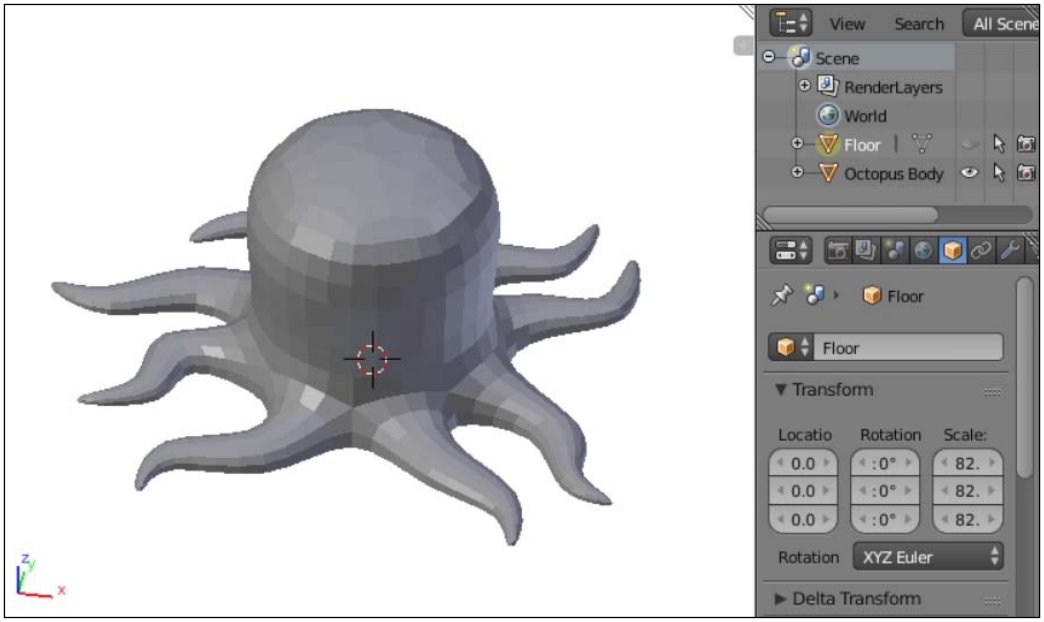
Objects can be renamed in the **Object** tab in the **Properties** panel, the one marked with an orange cube icon, as shown in the following screenshot:



1. Select **Octopus Body** in **3D View** (or **Cylinder** in the **Outliner** panel). In the **Objects** menu, click on the **Name**, currently **Cylinder**, and change it to `Octopus Body`:



2. Now, select the cube, and in the **Objects** menu, change its name to `Floor`.
3. Since the floor has served its purpose, in order to avoid it getting in the way or getting accidently transformed, hide it from the view by selecting it, and then, in the **3D View** menu, navigating to **Object** | **Show/Hide** | **Hide** or pressing *H*. The object is still in the outliner view but hidden in the 3D view:

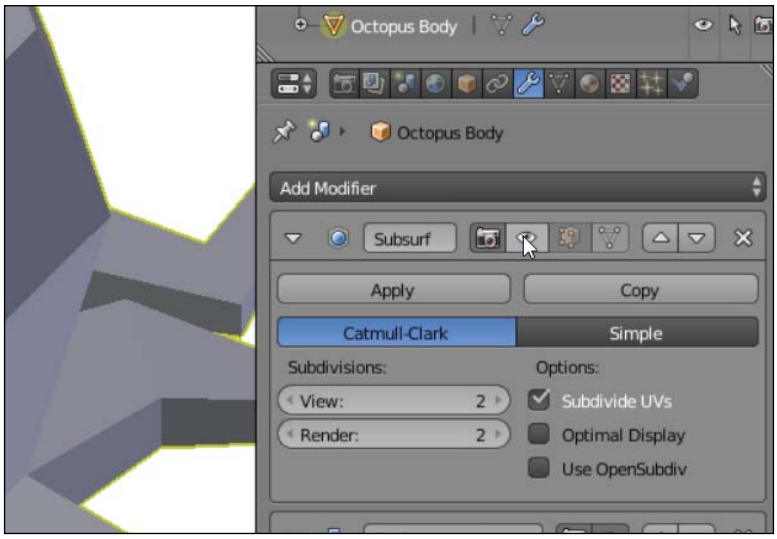


Hidden objects can be unhidden by navigating to **Object** | **Show/Hide** | **Show All**, pressing *Alt* + *H*, or pressing the eye icon next to the object in the **Outliner** panel.

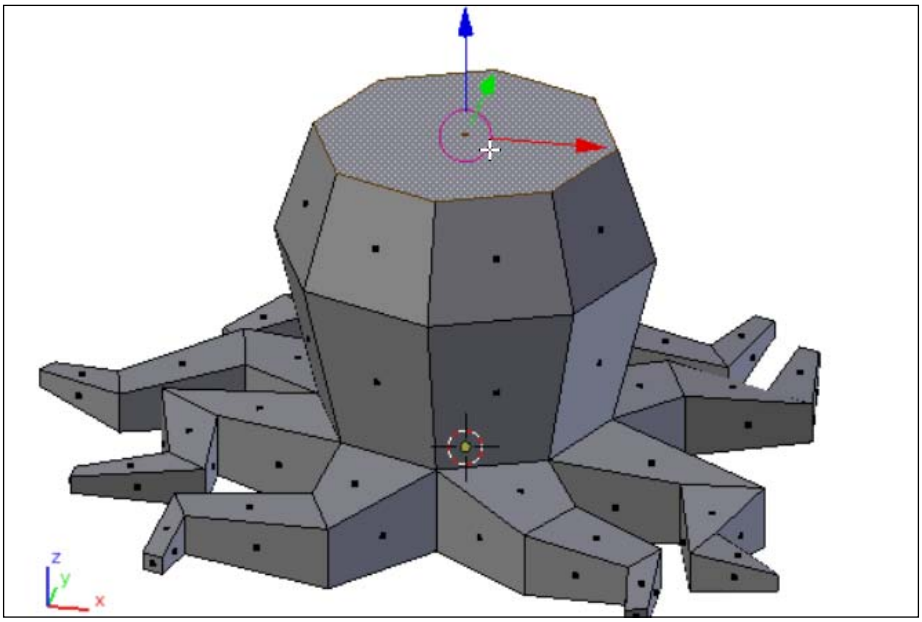
Adding a pencil cup

The octopus model looks appealing so far, but it can be functional as well. The plan for this project was a cup holder, so it needs to have the shape changed so that things can be put inside it:

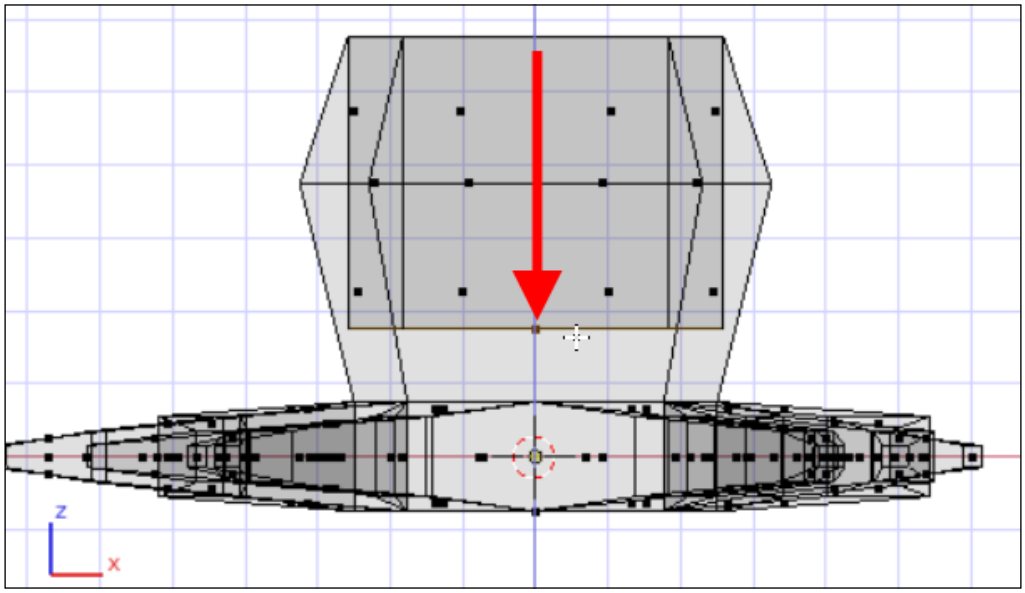
1. To start, temporarily turn off the subdivision surface modifier by locating the **Subsurf** modifier in the **Modifiers** tab and clicking the eye icon in it. Now, the simplified geometry is easier to work with:



2. Go to the **Face Select** mode (*Ctrl* + *Tab*) in the **Edit Mode** (*Tab*). Now, select the topmost face of the octopus:

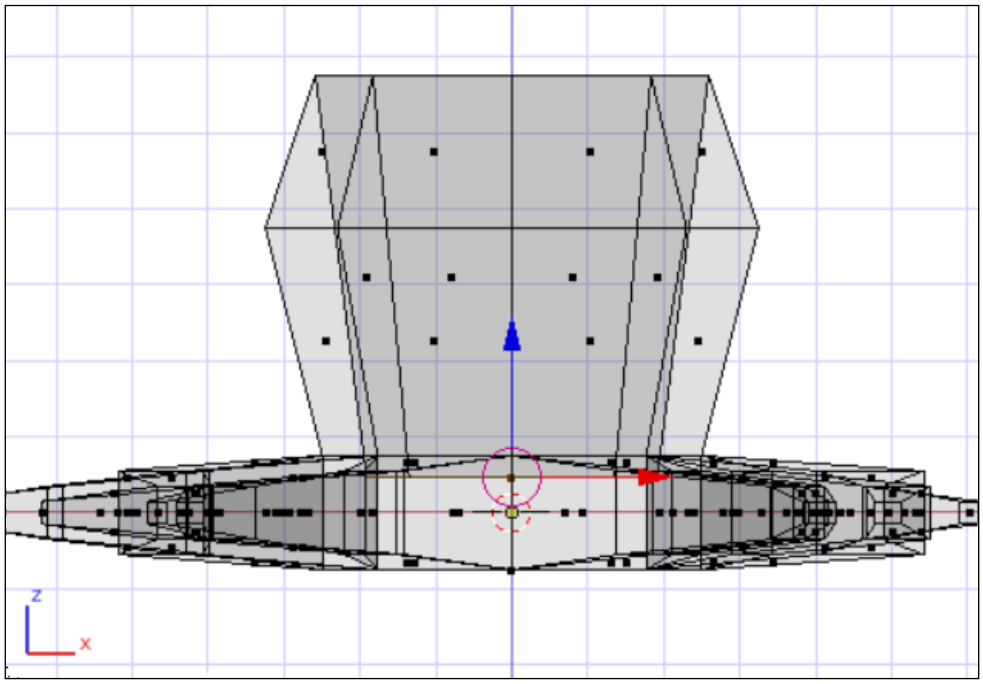


3. Switch to **Wireframe** (*Z*) and the **Front Ortho** view (NumPad *1*) and extrude (*E*) the top face into the body. Stop just a little above the red *x* axis line:

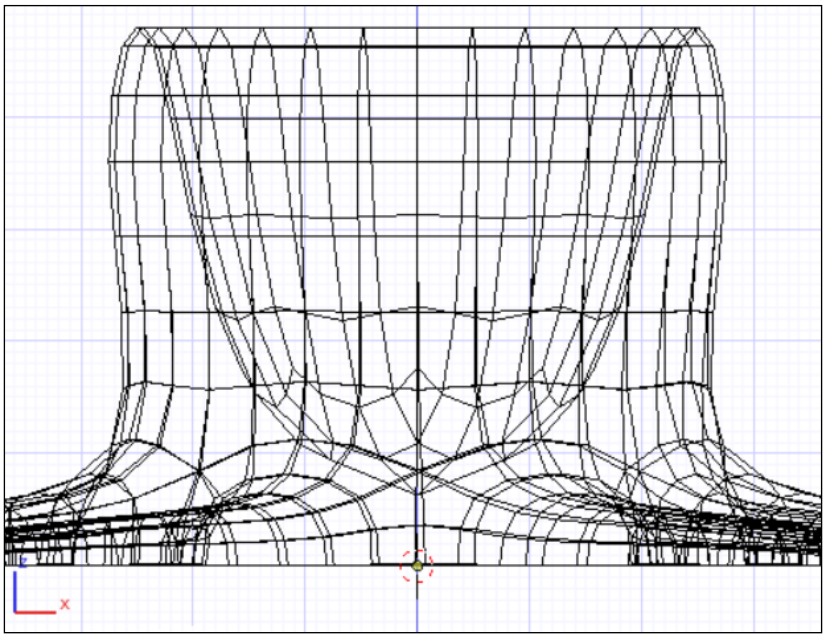


It may be necessary to scale (*S*) the bottom of the cup a little so that it fits inside the body.

After scaling the body will look this:

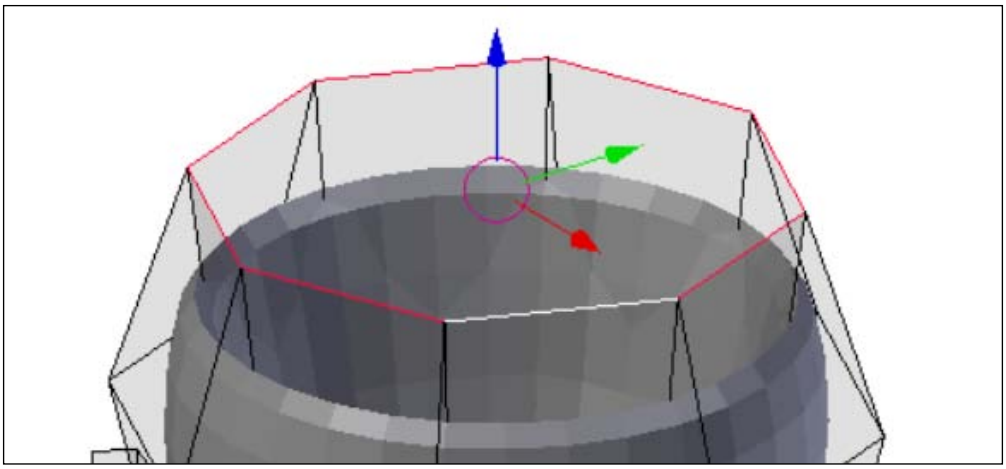


4. Now, the shape is generally right for a pencil holder. Turn the subdivision surface modifier back on by clicking on the eye icon in the **Subsurf** modifier in the **Modifiers** tab. You'll notice that the cup bottom is a bit too round to be a space-efficient cup:

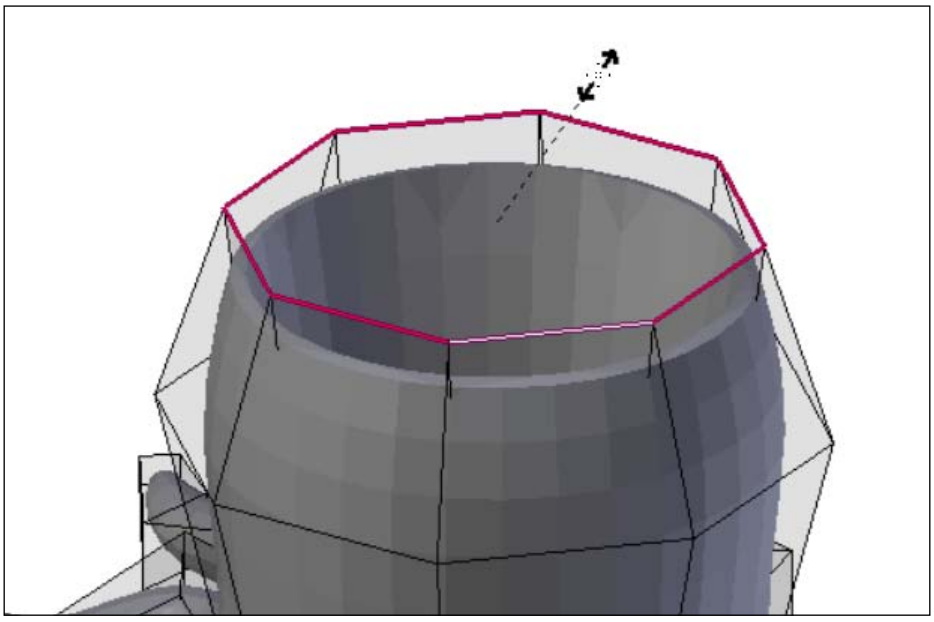


In the **Edit Mode**, edges can be marked with **Crease** to indicate to the **Subsurf** modifier that they should be sharpened.

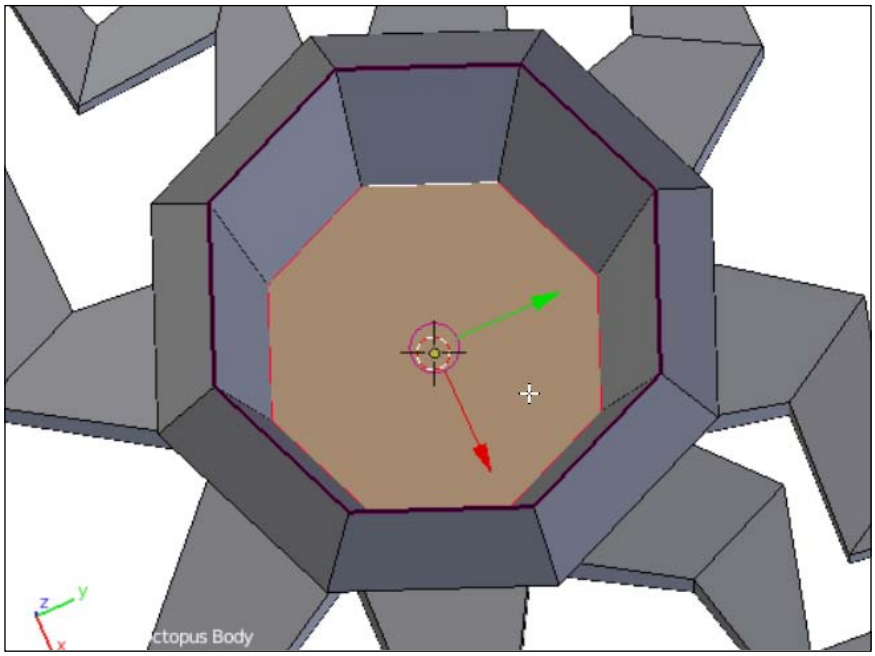
5. With no points selected (*A*) in edge or point, select the mode (*Ctrl* + *Tab*), hold down *Alt* on the keyboard, and select and click on one of the edges around the top lip of the cup in order to select all the points in a loop around the top of the mug:



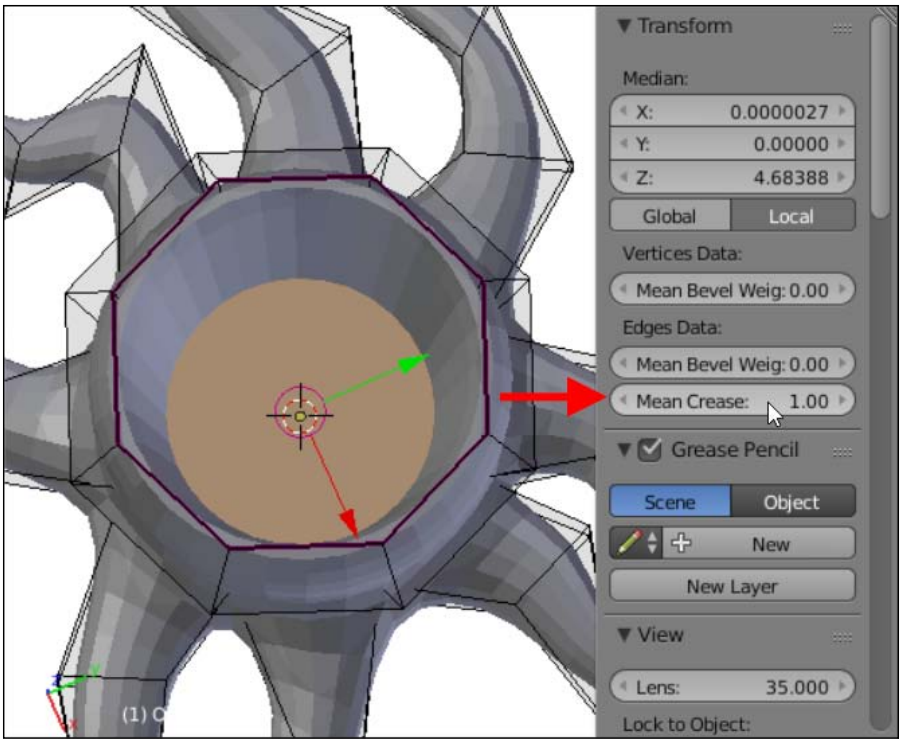
6. From the **3D View** menu, navigate to **Mesh | Edges | Edge Crease** or press *Ctrl + E* on the keyboard. Move the mouse pointer up or down in order to increase or decrease the amount of crease applied to the edge until it looks good:



7. Hold *Alt* and select and click on one of the edges at the bottom of the cup to loop and select all the points around the bottom of the cup. It may help to switch to the **Wireframe** mode (*Z*) or temporarily turn off the **Subsuf** modifier:



8. This time, set the crease value by finding the **Mean Crease** setting in **Properties** (*N*). This setting can be any decimal number between **0.0** (off) and **1.0** (maximum). Click on the setting and enter `1`, and then press *Enter* to set the value. Now, the bottom of the cup is flat:

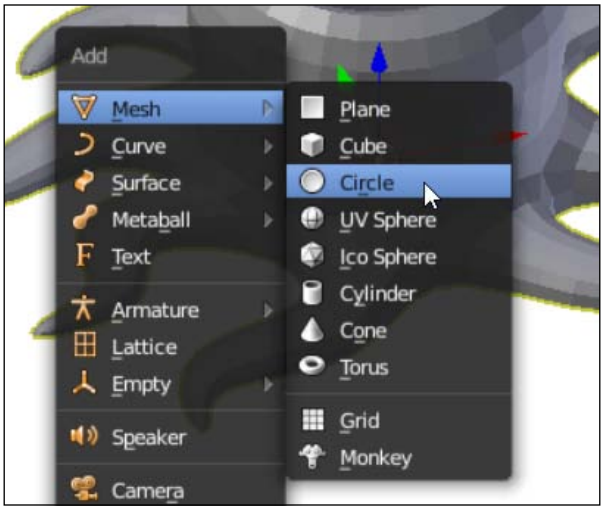


This sort of edge creasing is a powerful way to control the **Subsurf** modifier. Note that the bottom of the cup is still circular even though the original mesh is an octagon. **Subsuf** smoothens out the other edges even if it's told to crease others. It's very smart!

Adding a face

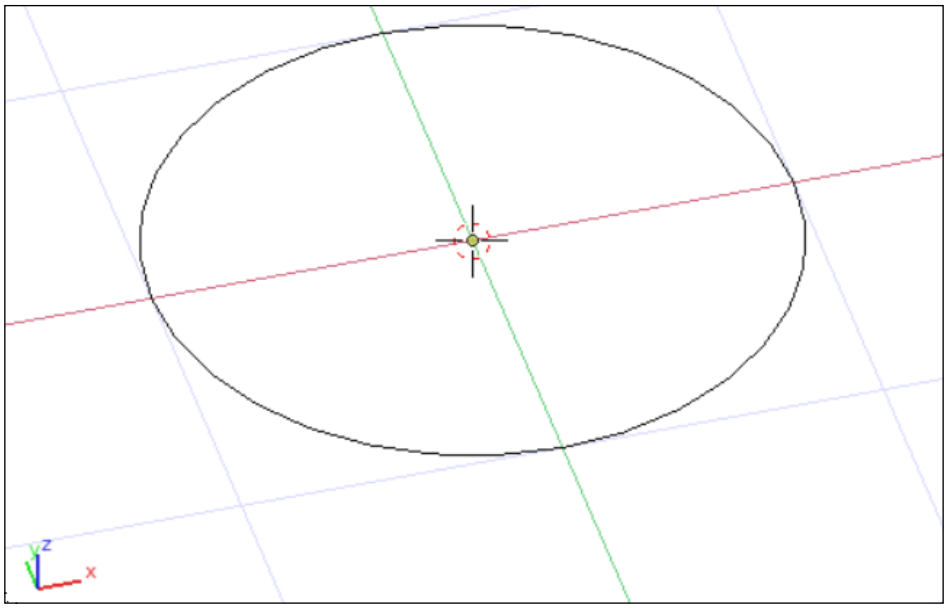
Functionally, the design is complete, but it's the little details that really make it. This little octopus would be much cuter with a face:

1. To start making a cute face, make sure the **Edit Mode** is off (*Tab*) and make sure the 3D cursor is at the origin (*Shift* + *C*). Now, create a circle (*Shift* + *A*) by navigating to **Mesh** | **Circle**:

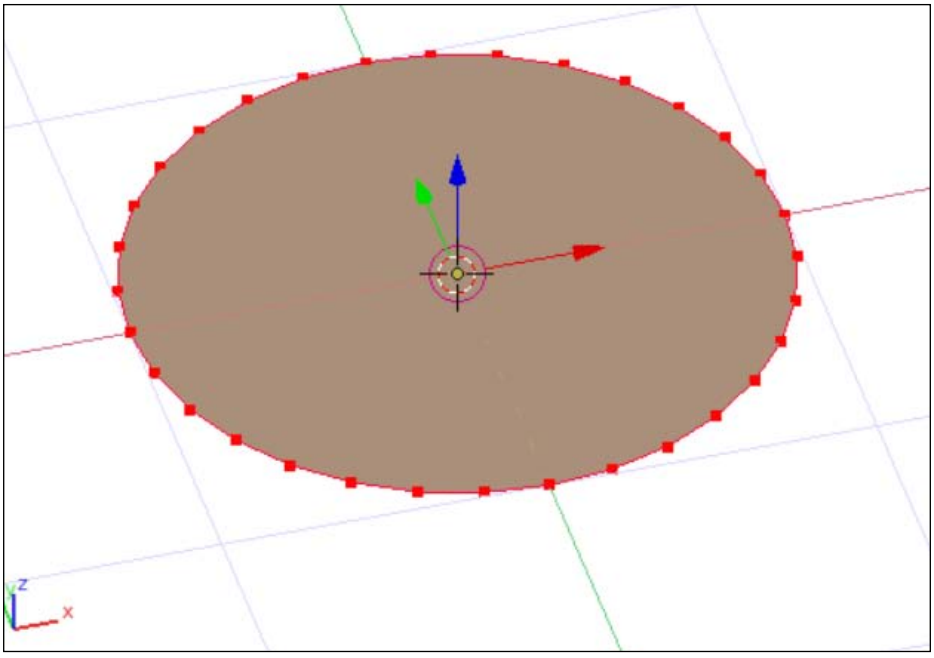


2. The circle is hidden inside the octopus, so with the circle still selected, switch to the local view (Numpad /). Since this circle will become the face, it's good to name it `Face` using the same method to rename the object as the one used earlier.

   The circle is a flat object that has no face and is just a ring of points and lines. That's easy enough to fix:

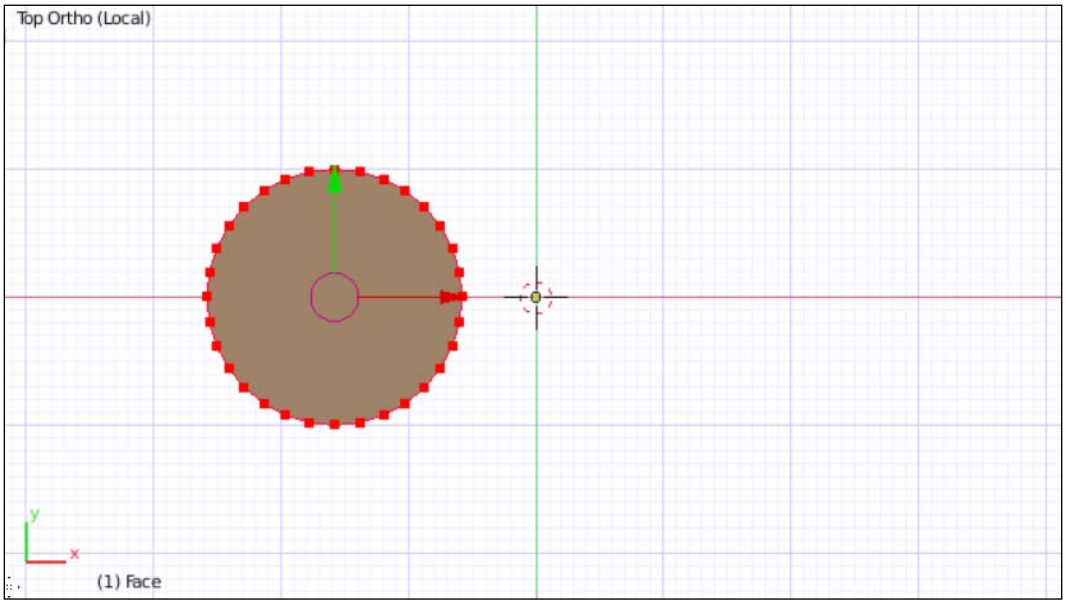


3. Switch to the **Edit Mode** (*Tab*). Now, with all the points or edges selected (*A*) from the **3D View** menu, navigate to **Mesh | Faces | Make Edge/Face** or press *F* on the keyboard:



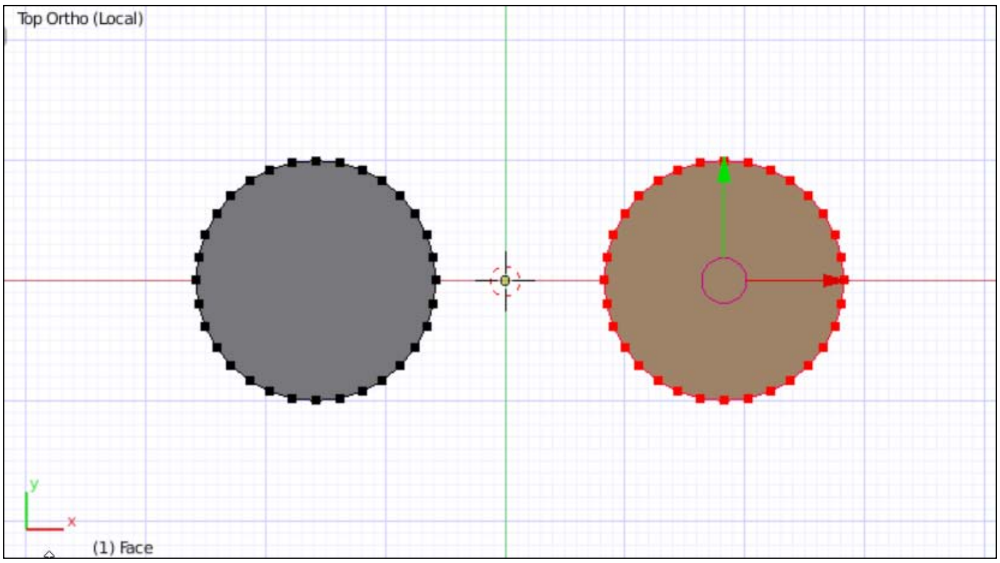
The `Make Edge/Face` command attempts to make a connected face from the selected vertices or lines. It can be quite intelligent, but if the points are not all flat, the edges may twist unexpectedly.

4. While still in the **Edit Mode**, switch to the top view (Numpad *7*), and with all the points still selected, move (*G*) them to the left of the green *y* axis:



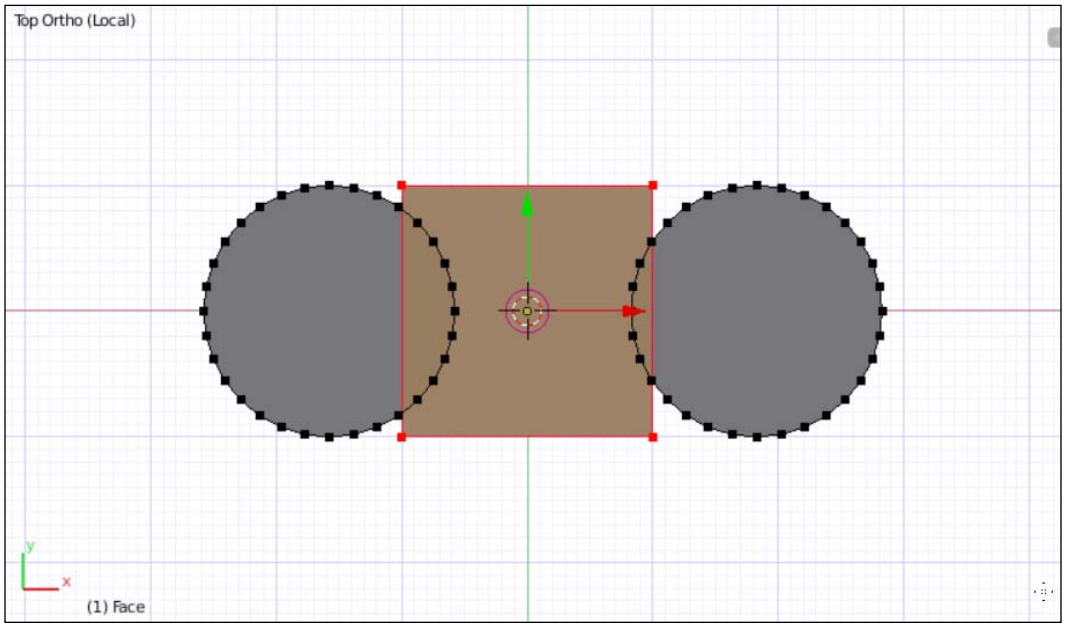
Duplicating an object outside of the **Edit Mode** creates a new object. However, duplicating points in the **Edit Mode** doesn't technically create a new object. Whatever is selected is duplicated while remaining a part of the same object. Objects in Blender can have many parts to them, all inside the same object.

5. Duplicate (*Shift* + *D*) all the points and place the duplicated points on the opposite side of the green *y* axis line:

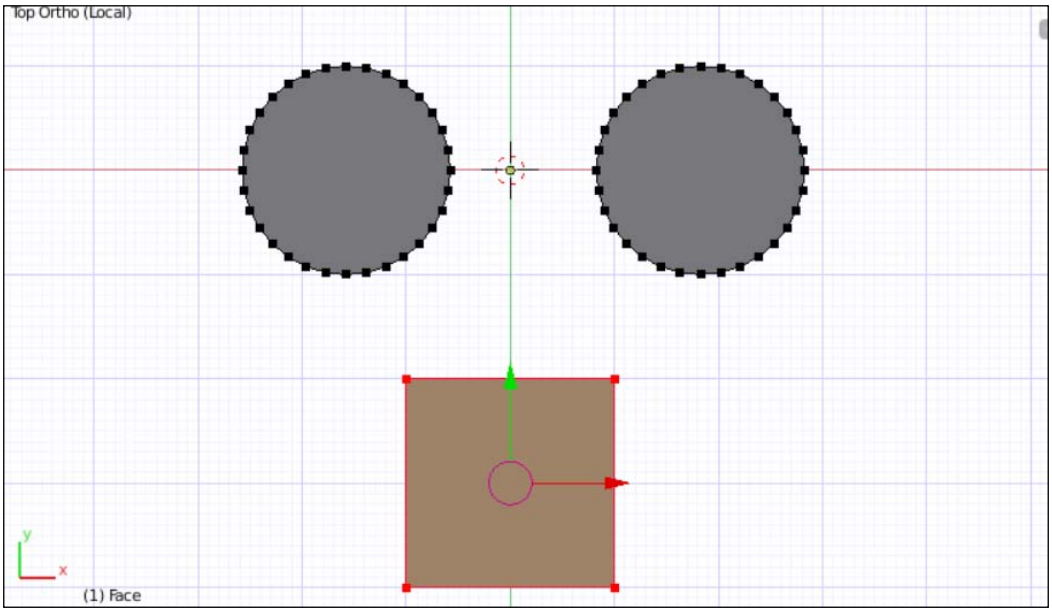


In this way, objects created while in the **Edit Mode** are just points, lines, and faces added to the existing object, even if they're not connected.

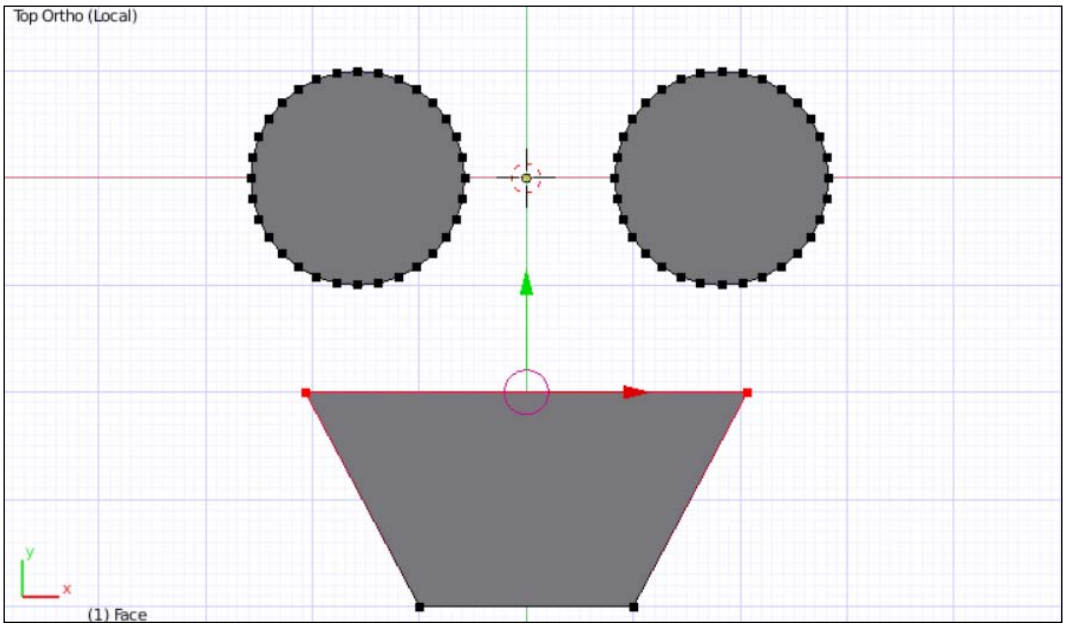
6. While still in the **Edit Mode** (making sure the 3D cursor is still at the origin), add (*Shift* + *A*) a plane to the face object by navigating to **Mesh** | **Plane**:



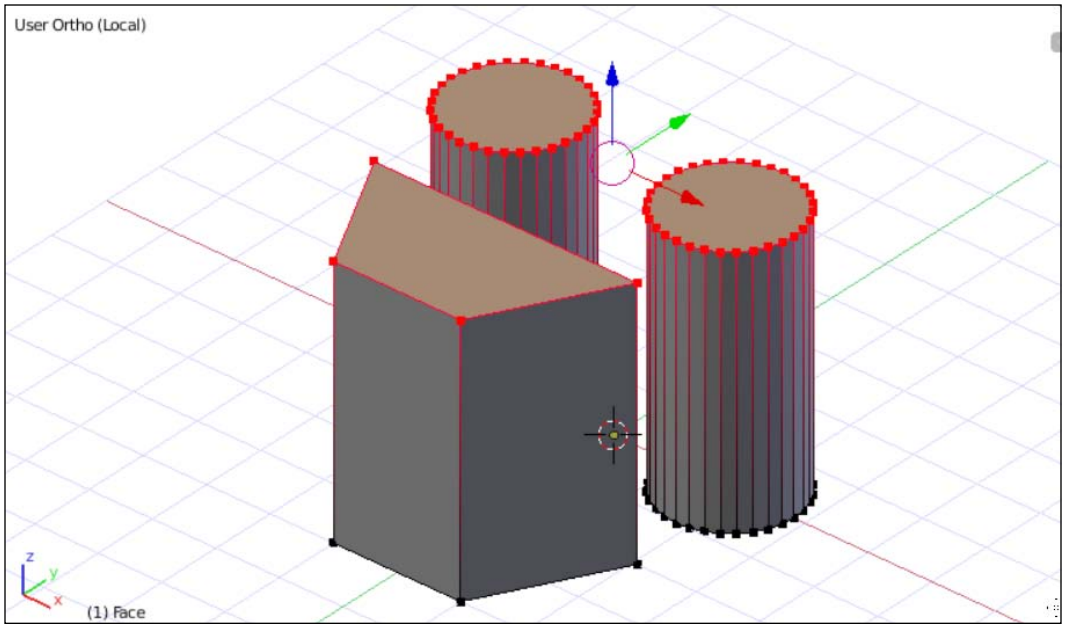
7. Move (*G*) the plane down along the *y* axis (*Y*) about **3** units:



8. Deselect all the points (*A*) and select the two points at the top of the plane. Then, scale (*S*) them up about twice in order to create a smiling mouth:

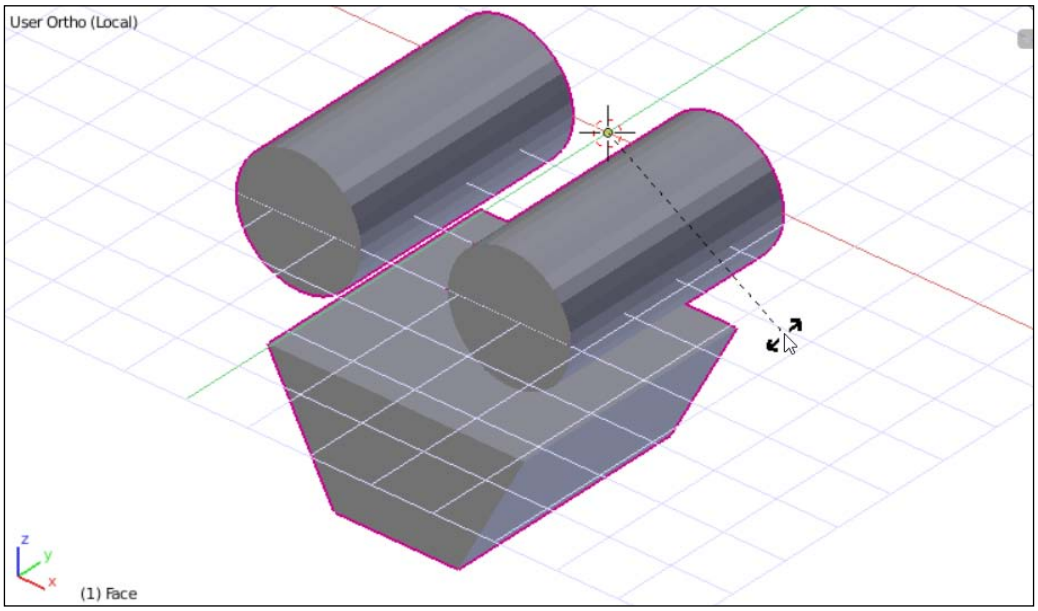


9. The shapes now look like a face but only in two dimensions. To give it the additional dimension required, select all the points (*A* twice) and extrude them (*E*) about **4** or **5** units.

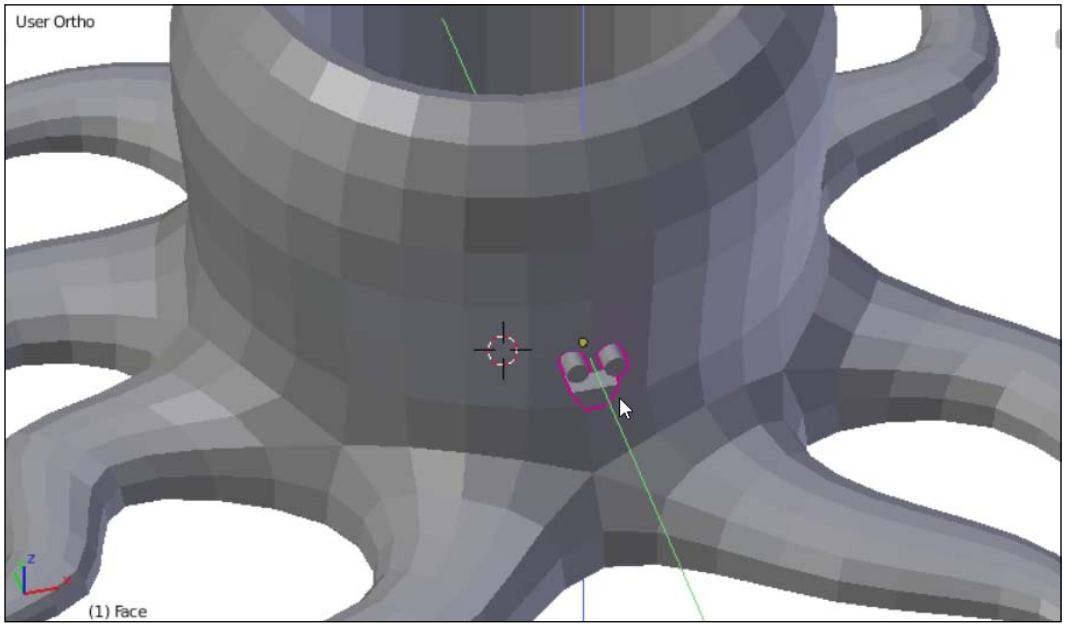


10. Exit the **Edit Mode** (*Tab*). The face template is now complete, and it just needs to be scaled and positioned.

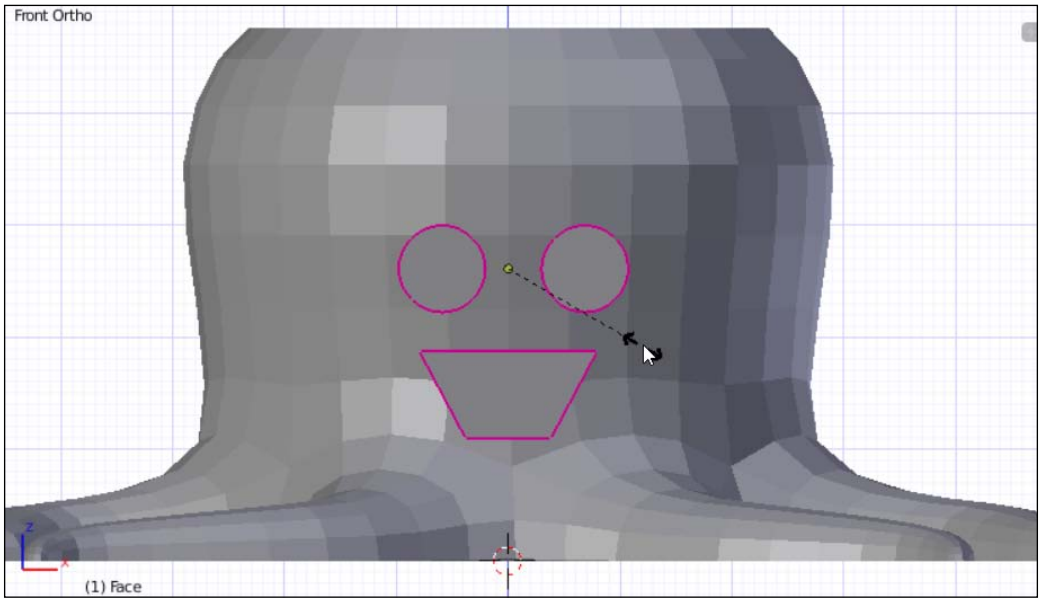
11. Rotate (*R*) the face object around the *x* axis (*X*) by 90 degrees (*90*):



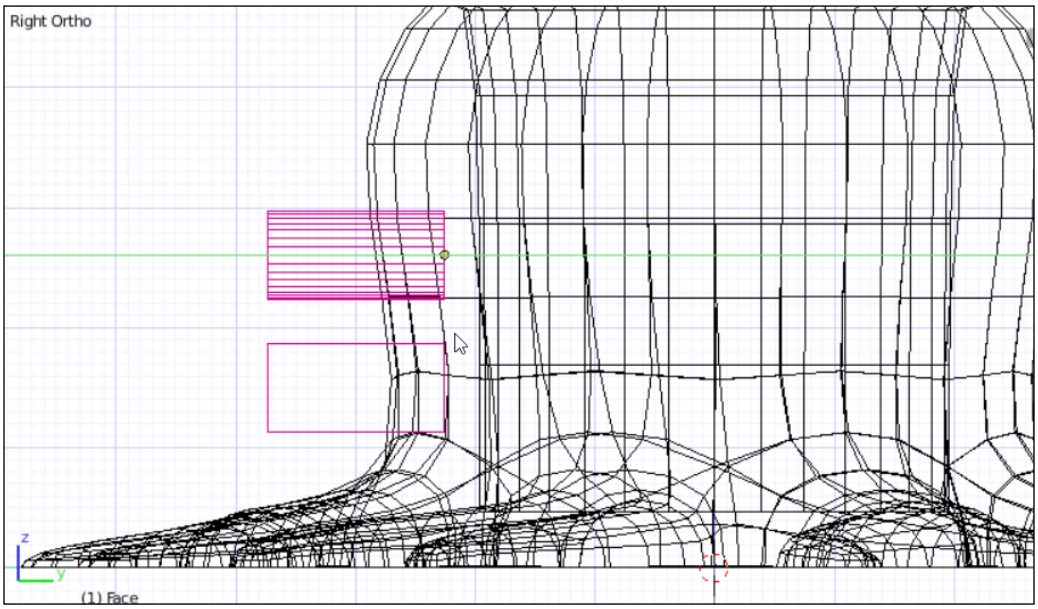
12. Exit the local view (Numpad */*). The face is hidden inside the body, so move (*G*) it in the *z* and *y* axes (*Shift* + *X*) until it's outside the octopus body:



13. In front of the **Ortho** view (NumPad *1*), scale the face (*S*) and move (*G*) it in the *z* axis (*Z*) until it's positioned properly:



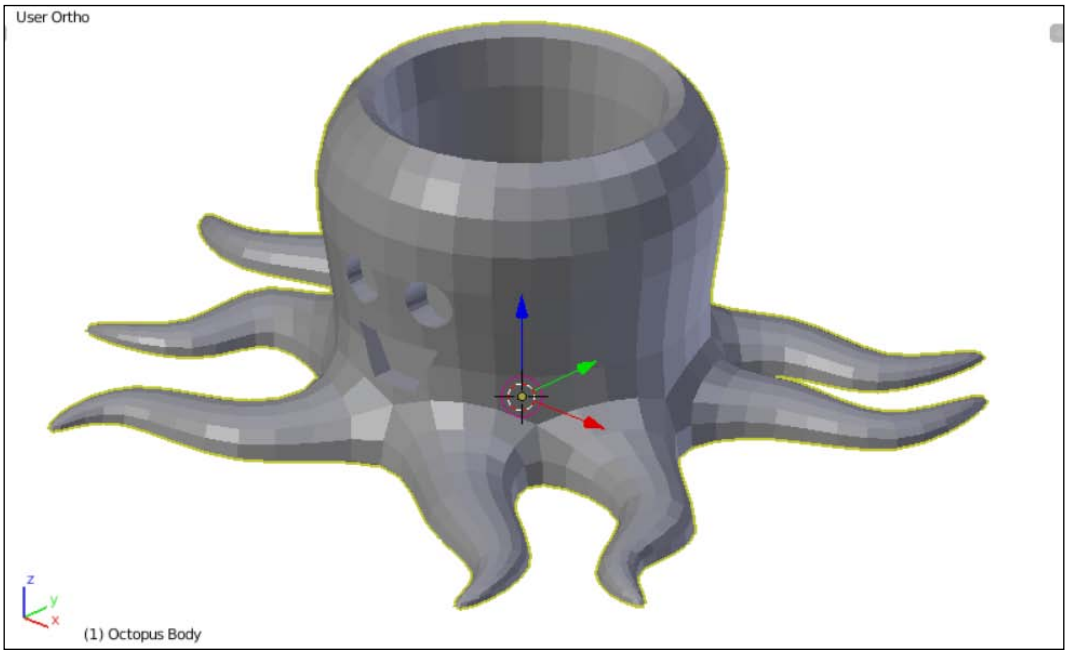
14. In the side **Ortho** view (Numpad *3*), move (*G*) the face along the *y* axis (*Y*) until it's partway into the body:



It may help to use the **Wireframe** view to ensure that the face is deep enough but not too deep. About four or five small squares (or 0.4 or 0.5 real-world millimeters) is perfect. If the face is too deep, it may cut all the way through into the cup space and may create an overhang problem. The top of the mouth will rely on some bridging during printing, and that's okay.

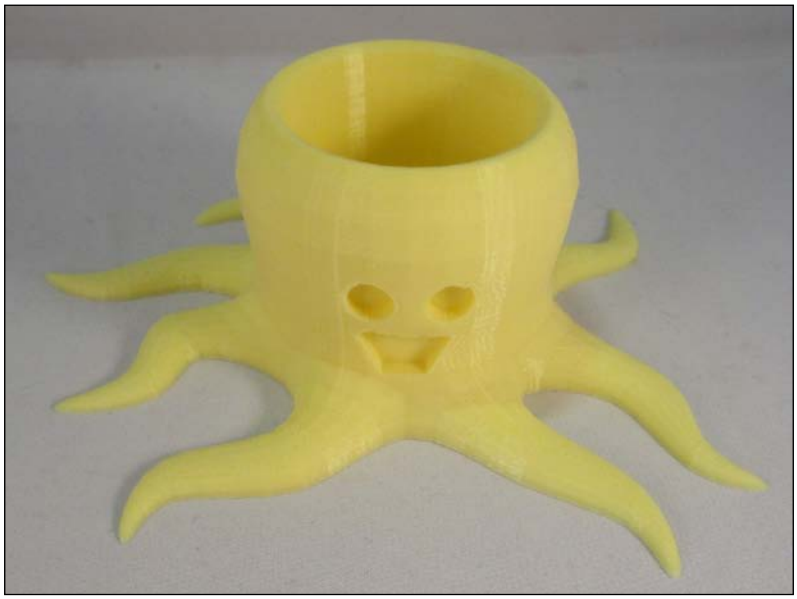
15. Now, select the body, and in the **Modifier** tab, add a **Boolean** modifier. Change the settings of the Boolean modifier to be a **Difference** operation with the **Face Object**.

16. The effect of this new modifier won't be visible immediately, so select the face object and hide it (*H*):



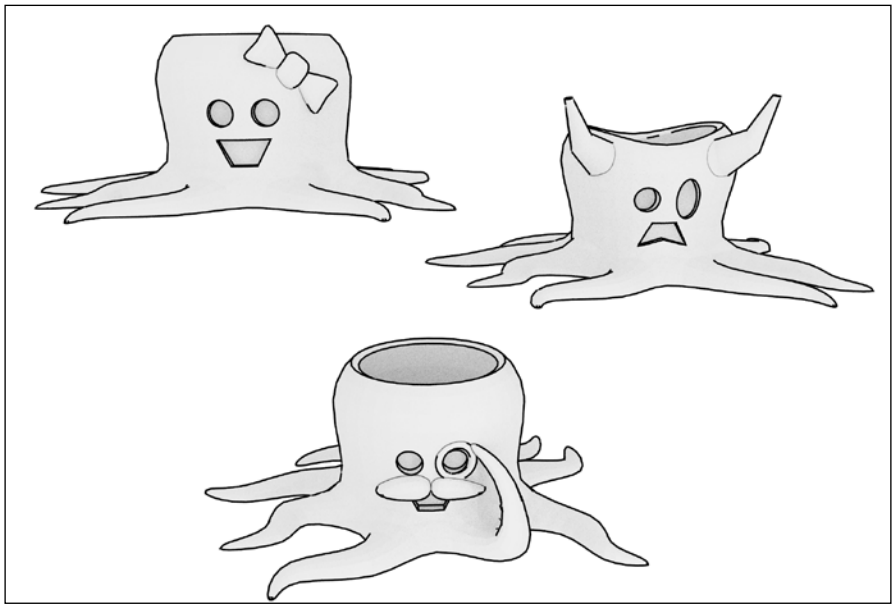
Your octopus pencil holder is now complete.

17. Select the body, and in the **Info** panel menu (at the top of the screen), navigate to **File** | **Export** | **Stl (.stl)** to view the 3D-print-ready octopus holder. Ensure that you choose a directory where you can find the file later in order to send it to the printer:



Finishing touches

Now, the octopus is done, but don't stop here. Use your imagination to make this design your own. Add accessories and personality, change the base shape, and come up with your own design to make this something you'll want to share. Just remember the rules about overhangs in your design and the sky is the limit:



Summary

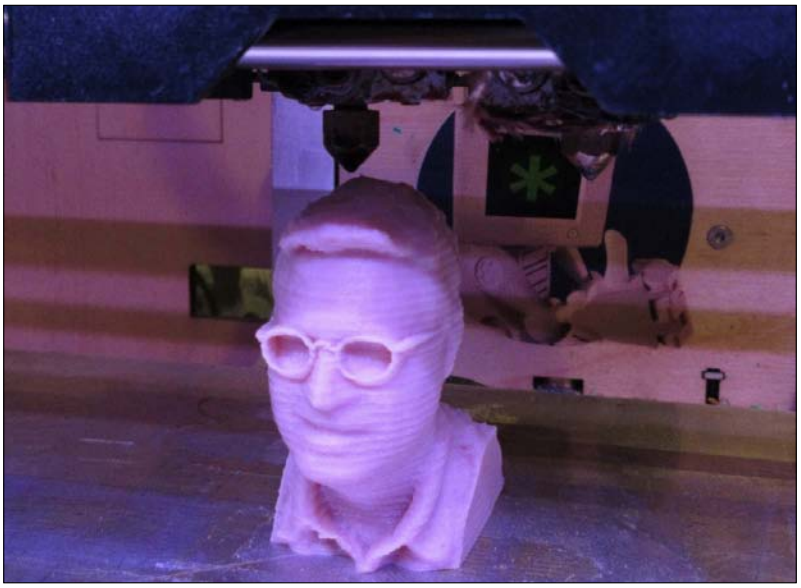
This project was functional but also cute. The techniques of vertex editing basic shapes in the **Edit Mode** with extrusion and loop cuts and then using the `mesh smooth` modifier and Boolean modifier in combination to further modify them can be used to create endless projects of any type and shape. Not all designs need to be cute, but they also don't all need to be purely practical. Combine a practical design with an aesthetic element in order to add personality and really take advantage of what 3D printing can offer.

Happy designing!

2
Prologue

3D printing basics

3D printing has been called a limitless technology. Still, that doesn't mean that it can create anything without any limitations. 3D printing can create things that no other manufacturing method can, but it has its rules that need to be followed in order to ensure success:



In this prologue, we'll discuss the following:

- What is 3D printing?
- Support-less 3D Printing and YHT (`http://joes3dworkbench.blogspot.in/2014/05/3d-printing-tip-designing-with-supports.html`)
- Wall thickness

What is 3D printing?

3D printing is cool. It seems like not a day passes without another mention of 3D printing in the news and media online. Everyone is excited about 3D printing. But when you look deeper, it seems like everything is being printed in 3D, and anything could be. Does 3D printing something make it better? What exactly is 3D printing?

In many ways, 3D printers are just tools, the same as any tool that you'd find in a wood shop or a garage. These tools make cool things, but they don't do that on their own, and just because something is made with, say, an electric drill press, doesn't automatically make it better than something that isn't. It's the things that people like you are doing with these tools that make them cool.

This is not to say that by itself, 3D printing isn't cool. It allow you create things, test them, change the design, and try something new quickly until you get them right. It creates things of incredible complexity and, because it's additive manufacturing, it generates comparatively little waste. Cheaper and faster 3D printers imply that there's a chance that there's a 3D printer near you.

The right software

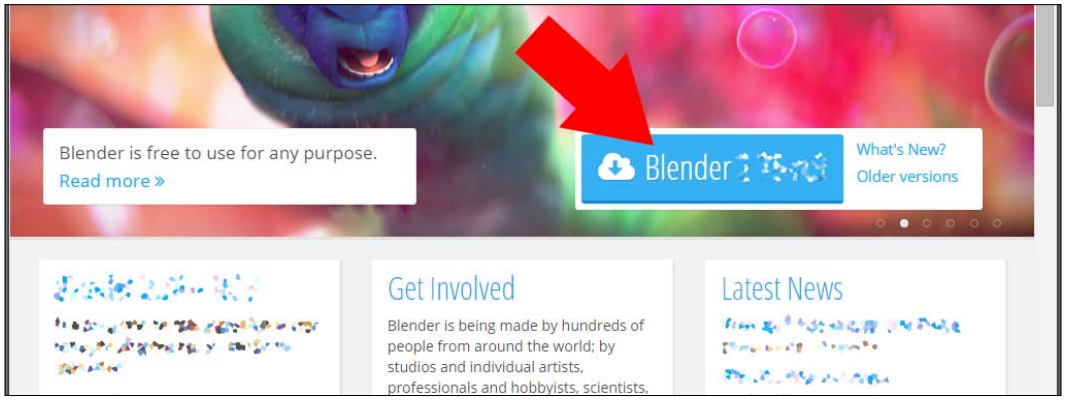
Creating 3D files requires 3D modeling software. There are many to choose from, with many new options appearing all the time, each with their own strengths and weaknesses. For this book, our software of choice is Blender because it's free, one of the most commonly chosen options for new 3D printing designers, and easily the most comprehensive and versatile option available. However, Blender requires some practice. The best practice is to use it to create cool things.

Downloading and installing Blender

The first thing that needs to be done is downloading and installing Blender:

1. On your PC or Mac computer, open a browser and go to `http://www.blender.org`.

2. Locate the download button on the main page for the latest version of Blender and click on the download Blender button, as shown in the following screenshot:



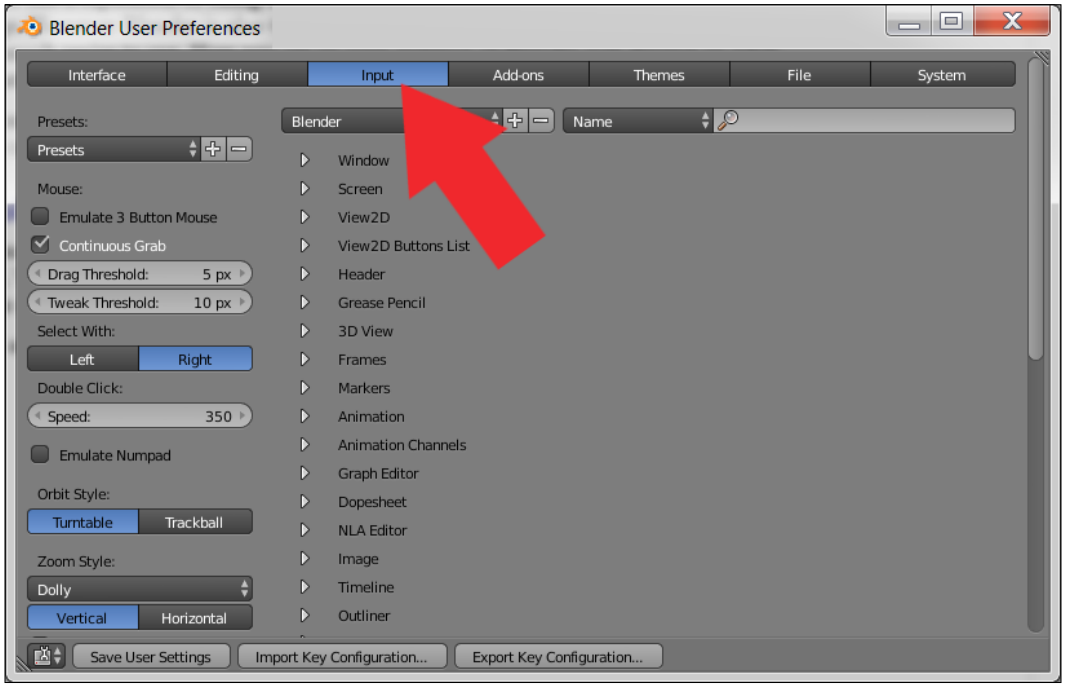
3. Scroll down and find the download appropriate for your system. If you're on a Windows system and are not sure what to do, just choose the MSI package option.
4. When the installer finishes downloading, run it.
5. Follow the prompts to install Blender on your system.
6. When the installer is finished installing Blender, run the program. Click anywhere to close the splash screen.

Blender is now installed and ready for use.

Setting up Blender

The default settings for Blender can be unintuitive. However, Blender is also very easily adjusted with a few settings and can be made much easier to use. How you set up your instance of Blender will depend on what your setup is like.

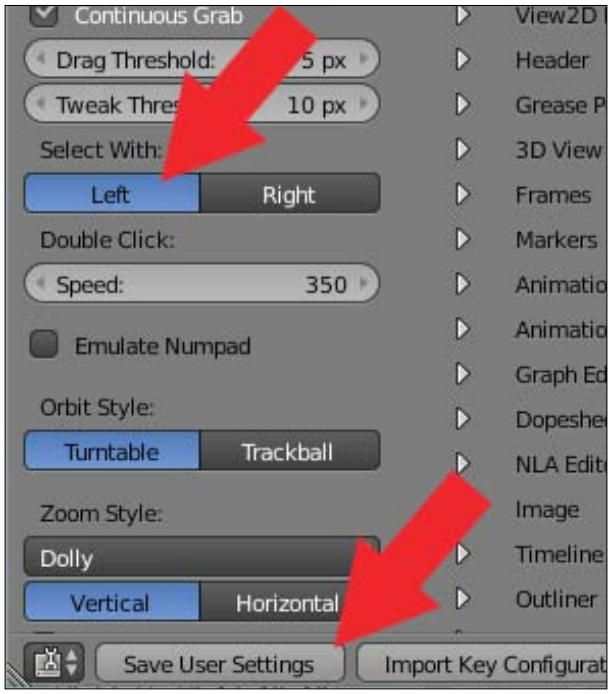
To access the settings in the top menu, navigate to **File** | **User Preferences**. In the **Blender User Preferences** panel that comes up, select the **Input** tab button, as shown in the following screenshot:



Scrolling the mouse wheel and number pad

The recommended setup for Blender is to use your keyboard and a mouse with a scroll wheel. In this case, there is only one setting that is recommended to be changed from the default:

1. Click on the button that says **Left** under the **Select With** section.
2. Click on the button that says **Save User Settings**, as shown in the following screenshot:



Changing this one setting will make Blender much more intuitive to use.

Because the default is for the right mouse button to be the select button, and some may prefer to keep this the default, we will refer to whatever option is chosen as the select mouse button and the other mouse button as the right mouse button. While still a bit confusing, it will help users who miss this section.

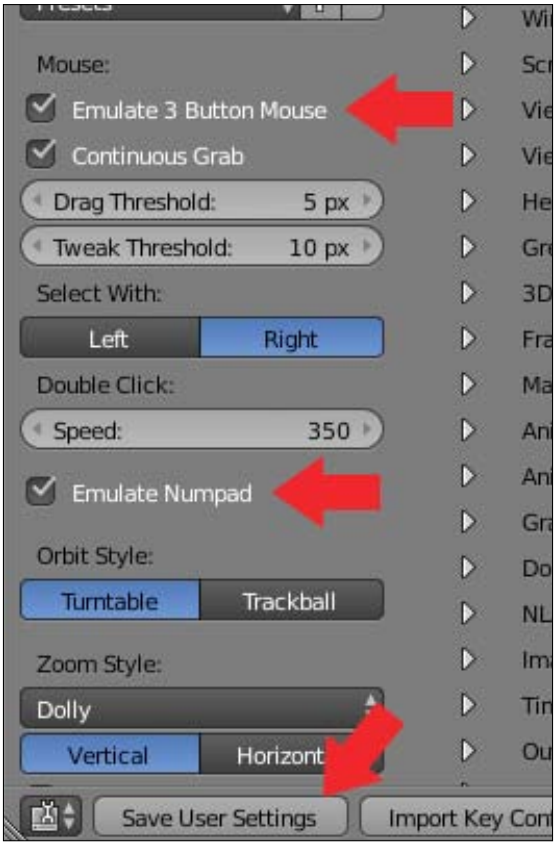
With a scroll wheel mouse, the scroll wheel can be used as a middle mouse button. Blender uses this middle mouse button to manipulate the view.

If you have a NumPad but no scroll wheel mouse, it may still be recommended that you use this setting because the interface will be much more intuitive, though you will be sacrificing some functionality that the middle mouse click offers. The choice is yours as to whether it's worth having a more intuitive interface.

Laptops with a touchpad but no number pad

On a laptop with a touchpad (with no middle click) and no number pad, both very important to navigate the view, select the following settings:

1. Check the **Emulate 3 Button Mouse** and **Emulate Numpad** checkboxes.
2. Click on the button that says **Save User Settings**, as shown in the following screenshot:



With this setup, most of Blender's functionality will be available to laptop users, though these settings are less intuitive. With these settings, you will need to use the right mouse button to select objects, press *Ctrl* and the right mouse button as a middle mouse button to change the view, and the number keys across the top to perform the functions of the number pad on a regular keyboard.

With Blender set up, it's time to start learning how to use it. Close the **User Preferences** window.

What to design for?

It is the best practice to always design according to the strengths and weaknesses of the medium you'll be using. The projects in this series of books will focus on designing for Fused Filament Fabrication 3D printers because they're inexpensive and more readily available than the other methods. And the parts from FFF 3D printers are suitable for a wide variety of functional uses. Also, many of the techniques for FFF design transfer to the other types of 3D printing. But because FFF 3D printers have limitations, there will be some things you need to know first.

Overhangs and supports

FFF 3D printers have to worry about overhang. Overhang is when a part of the design, when it prints, does not have anything between it and the build platform. To compensate for this, the 3D printer can build a lattice of support material up to the overhanging part. After the print, the support material will have to be removed. But since, for most FFF 3D printers, the support material is made of the same material as the object, it can rarely be removed without a trace, which is sometimes difficult to clean up completely and can leave a mess in more complex prints:

Because of the troubles with supports, it's a good idea to design for support-less 3D printing.

Supportless 3D printing

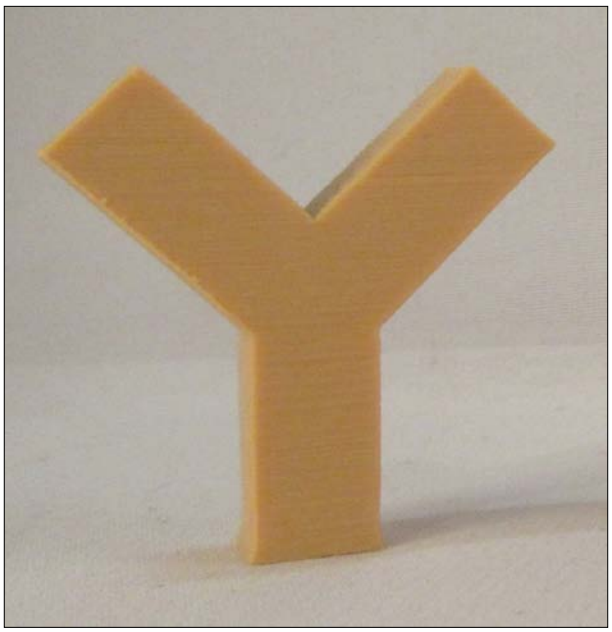
Think of building a snowman or sand castle. There's a lot that can be done with the medium of sand or snow, but try to get too fancy with the design, and it will fall apart. As long as every part is sitting on top of something, chances are it will hold together. You could even slope gently outwards as long as you don't push it too far.

It's the same way with 3D prints. Because they print in layers, each layer needs to have something to lay down on. If a design is made so that a part has nothing underneath it, dangling in the air, then the printer will still extrude some plastic to try to print the part, but with nothing to print on, the plastic will just drool from the extruder until it gets wiped off on some other part, making an ugly mess and ruining the print.

As long as you put some thought into it, you can make designs that will succeed in most cases. There are a few rules that can help, and these rules can be illustrated with the letters Y, H, and T.

Y – gentle overhangs

Think about 3D printing a capital letter Y, standing up on the build platform; something like this:

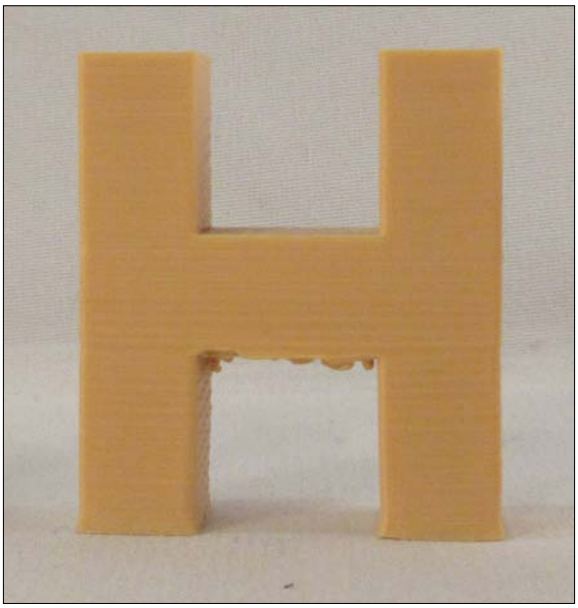


As the print gets to the part where the arms of the Y branch out, the change is gradual. It is possible to have the current layer slightly larger than the previous one, provided the overhang is gentle. Generally, a 45-degree overhang is safe. Hence, a shape such as the letter Y will successfully print standing up.

However, if the overhang is too great or too abrupt, the new layer will droop, causing a print to fail. Some 3D printer owners pride themselves in pushing their overhang and have seen success with angles as steep as 80 degrees, but to be safe, keep your angles no more than 45 degrees.

H – bridging

If a part of the print has nothing above it but has something supporting it on either side, like a capital letter H standing up, then when printing it, we may be able to bridge the gap:



Use caution when bridging. The printer makes no special effort when making bridges; they are drawn like any other layer: outline first, then infill. As long as the outline has something to attach to on both sides, it should be fine. But if that outline is too complex or contains parts that will print in mid-air, it may not succeed. Being aware of bridges in the design and keeping them simple is the key to successful bridging. Even with a simple bridge, some 3D printers need a little bit more calibration to print it well.

Hence, a shape like the capital letter H will successfully print most of the time because of bridging.

T – orientation

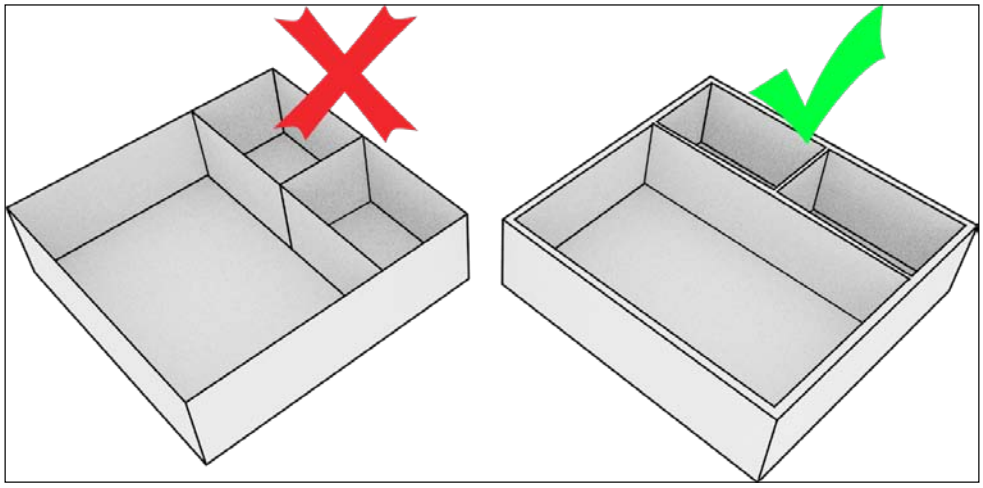
If you were to try to print a capital letter T standing up on the build platform, you would surely run into problems:

The top arms have far too much overhang to print successfully. Of course, the solution to this is simple; when designing, flip the T over or lay it down. In fact, every letter of the alphabet will print successfully if laid on its back, but the letter T illustrates this best. Sometimes, when designing a part for 3D printing, it's good to turn it around and orient it so it prints well. Not every print needs to be printed in the same way it's going to be used.

Wall thickness

There is a minimum size for what a 3D printer can print. It's best to err on the side of caution. While a 0.8 mm wall will be okay on most FFF 3D printers, a 2 mm wall is thick enough for slicers to use one or two outlines without conflict and still have room for a little infill, no matter what the nozzle diameter is. This will make solid prints that will succeed in almost all cases, and 2 mm is still fairly thin, allowing considerable detail.

In the same vein, a wall by itself, without an inside or outside, isn't printable because a 2D wall has no thickness and doesn't describe a shape that can exist in real life. 3D prints must be a part of a three-dimensional shape with a thickness:



Summary

3D printing is cool and allows the creation of fantastic and detailed objects without requiring much interaction with people after the design is done. But designing for 3D printing is a lot like designing for any other type of manufacturing.

Fused Filament Fabrication 3D printing, or FFF for short, is one of the oldest, most mature, and cheapest forms of 3D printing, so this series will focus on designing for it. It involves melting a plastic filament and drawing the object layer by layer, with each layer sitting on top of the one below it.

Designing for most effective FFF printing means thinking about overhangs and supports and about the parts of the prints that don't have anything underneath them when they print. To avoid requiring supports when printing, it can help to remember the letters Y, H, and T when designing in order to remember to consider gradual overhangs, bridging, and orientation. In addition, it's important to remember that details should generally be about 2 mm thick.

This was, of course, the most basic of overviews of designing for 3D printing and setting up Blender. For more detailed information, let's start with our first project: octopus pencil holder.

Index

Symbols

B

F

O

P

R

Thank you for buying
3D Printing Designs: Octopus Pencil Holder

About Packt Publishing

Packt, pronounced 'packed', published its first book, *Mastering phpMyAdmin for Effective MySQL Management*, in April 2004, and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution-based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern yet unique publishing company that focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website at `www.packtpub.com`.

About Packt Open Source

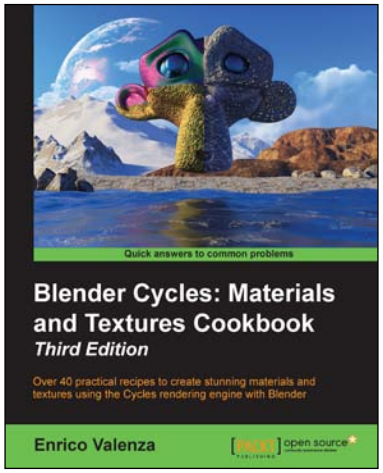
In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around open source licenses, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each open source project about whose software a book is sold.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to `author@packtpub.com`. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, then please contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

[PACKT] open source
community experience distilled
PUBLISHING



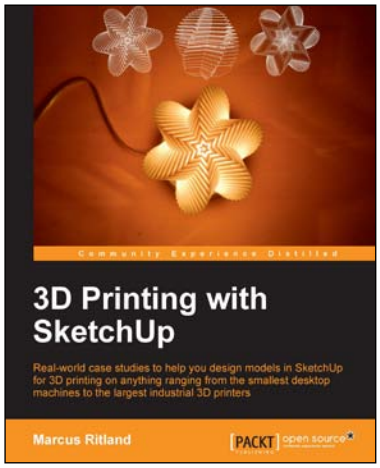
Blender Cycles: Materials and Textures Cookbook

Third Edition

ISBN: 978-1-78439-993-1 Paperback: 400 pages

Over 40 practical recipes to create stunning materials and textures using the Cycles rendering engine with Blender

1. Create realistic material shaders by understanding the fundamentals of material creation in Cycles.
2. Quickly make impressive projects production-ready using the Blender rendering engine.

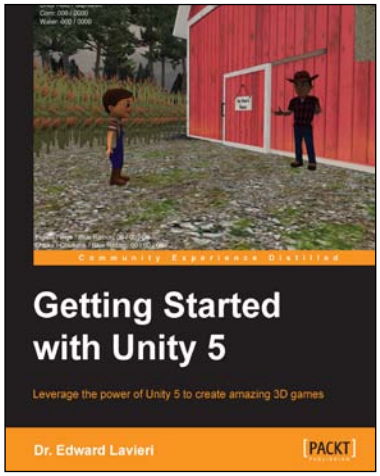


3D Printing with SketchUp

ISBN: 978-1-78328-457-3 Paperback: 136 pages

Real-world case studies to help you design models in SketchUp for 3D printing on anything ranging from the smallest desktop machines to the largest industrial 3D printers

1. Learn how to design beautiful architectural models that will print on any 3D printer.
2. Packed with clearly illustrated examples to show you just how to design for 3D printing.
3. Discover the essential extensions and companion programs for 3D printing your models.



Getting Started with Unity 5

ISBN: 978-1-78439-831-6 Paperback: 184 pages

Leverage the power of Unity 5 to create amazing 3D games

1. Learn to create interactive games with the Unity 5 game engine.
2. Explore advanced features of Unity 5 to help make your games more appealing and successful.
3. A step-by-step guide giving you the perfect start to developing games with Unity 5.

Building a Game with Unity and Blender

ISBN: 978-1-78528-214-0 Paperback: 250 pages

Learn how to build a complete 3D game using the industry-leading Unity game development engine and Blender, the graphics software that gives life to your ideas

1. Learn the fundamentals of two powerful tools and put the concepts into practice.
2. Find out how to design and build all the core elements required for a great game - from characters to environments, to props.
3. Learn how to integrate Artificial Intelligence (AI) into your game for sophisticated and engaging gameplay.