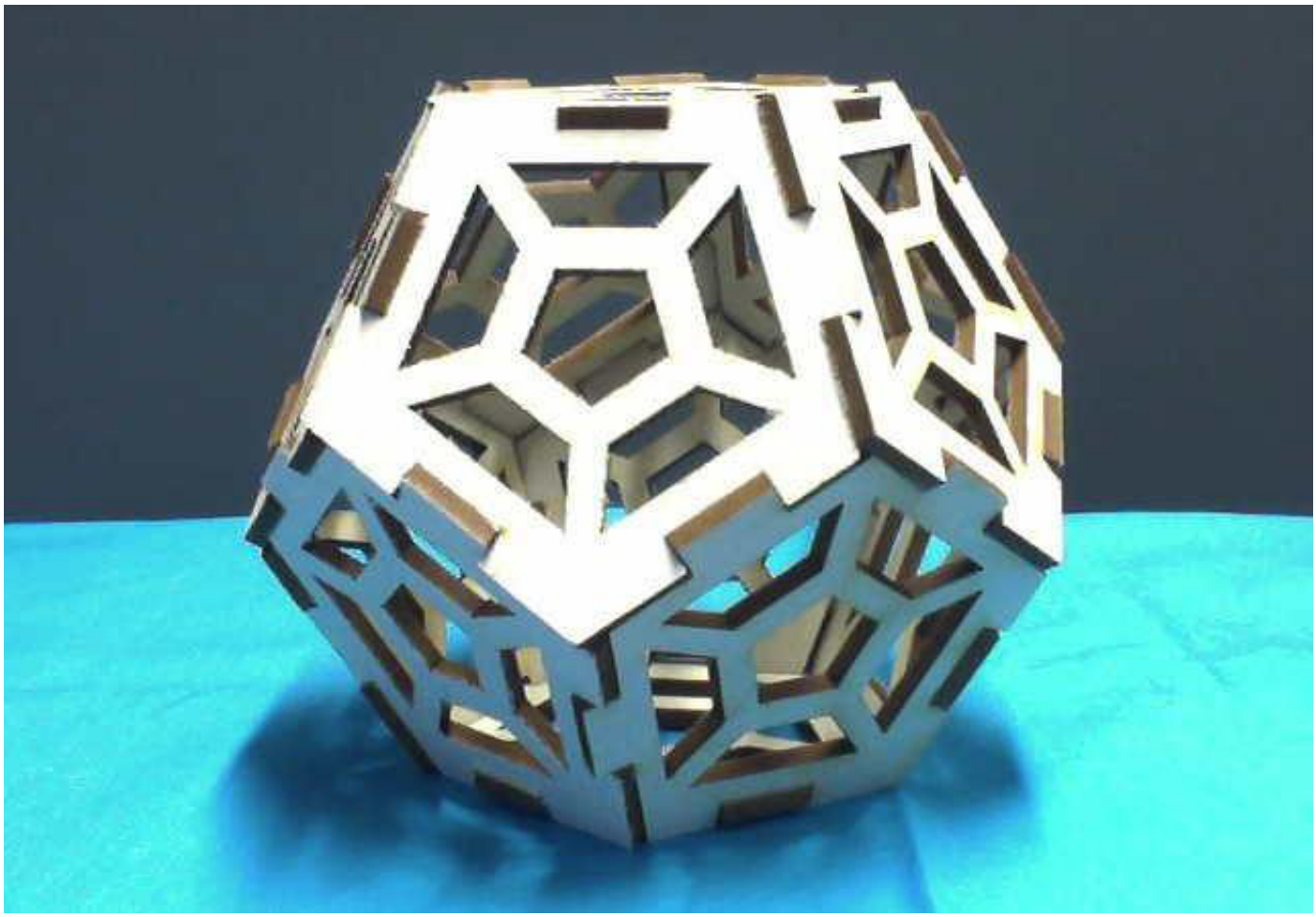


CutCAD

User Guide



Media Computing Project 2013/14

at the
Media Computing Group
RWTH Aachen University

by
Mirko Hartmann
Dennis Lewandowski
Pierre Schoonbrood
Jan Thar

Contents

1 Introduction **1**

2 Examples **3**

2.1 Building a Pyramid 4

2.2 Building a Dodecahedron 11

2.3 Building a Housing for an Electronics Device 16

3 Tool Time **23**

3.1 Select 24

3.2 Draw Rectangle 26

3.3 Draw Symmetric Polygon 27

3.4 Draw Trapezium 28

3.5 Draw Polygon 30

3.6 Connect 31

3.7 Cutout 34

3.8 Copy 36

3.9 Load SVG 37

3.10 Cut 39
3.11 STL import 43
3.11.1 Load STL 43
3.11.2 Change STL 43
3.12 Save and Load 44
3.12.1 Save 44
3.12.2 Load 44

4 Customize it **45**
4.1 Change/Add Material 46
4.2 Change the Lasercutter 48
4.3 Building Custom Tools 51

5 Behind the Scenes **55**
5.1 Align and Rotate 56
5.2 Creating the Outlines of Tenons 58
5.2.1 Problems and Limitations 59
5.3 Find Intersection for Rotation 60
5.4 Architecture overview 65

6 Future work **67**

Chapter 1

Introduction

CutCAD is the result of our work for the Media Computing Project 2014 at the RWTH Aachen. For our final project, we tried to develop our own customizable box maker to support personal fabrication. CutCAD is supposed to help you to build housings for your electronics projects, create boxes to store things, or build other crazy things with your laser cutter. We found the existing tools to be rather lackluster. To create three dimensional objects with a lasercutter, one either has to use tools such as BoxMaker, which are easy to use, but not very powerful, or create every piece of the object completely on their own in tools such as Inkscape, which are very powerful, but make it very hard to even build simple things such as a box.

One of the main issues when building three dimensional objects with a lasercutter is the connection of the parts: A common approach is to create a tenon-like structure that allows two shapes to interlock and form a stable connection. However, creating such structures is a lot of work in a vector graphics tool such as inkscape: One has to calculate the correct sizes for tenons and make sure that the sides of both shapes fit together perfectly. This is one of the main problems that CutCAD solves for you: You can simply connect shapes in CutCAD and CutCAD will take care of the calculations necessary to create stable connections between the shapes.

The basic approach of CutCAD is to build three dimensional shapes by connecting the edges of two dimensional shapes. These shapes can either be created with tools provided by CutCAD or imported from svg-files. After a connection has been made, each shape will be automatically aligned in 3D space and the corresponding tenon structure will be created on the connected edges. When all shapes are connected and the 3D-object is complete, you can directly cut these parts with the laser cutter or just export them into an SVG-file.

After the correct setup has been done, cutting the parts should be as easy as a click on an icon. CutCAD will send the necessary jobs directly to the lasercutter.

We will start this documentation by describing how to build some basic shapes as examples of how to use CutCAD. We will then go on to give you a short overview over the tools available in CutCAD. In the third section, we will explain the setup of the program for different Lasercutters (e.g. setting the IP address of the Lasercutter or changing the list of available materials). Then we will explain some of the algorithms we have come up with to create CutCAD (e.g. the algorithm for the correct alignment of shapes in 3D-space and for creating the outlines of the tenon-structure). In the last section, we will talk about the features we envisioned that did not make it into the final build of our projects to give you some ideas of what could be improved about CutCAD.

On some pages you will find comment boxes or note, which will describe some of the challenges we faced and talk about the limits of this current version of CutCAD.

If you desire to further extend the program, you are welcome to do so. As already mentioned, we describe some of our ideas in the last section of this documentation, but of course, you are free to extend it however you wish. You will also find JavaDocs together with the source code of CutCAD.

Chapter 2

Examples

In this part of the documentation, we want to show you how to build things with our software based on examples. We will start with some simple shapes and then work our way up to more complex and interesting examples.

After reading this section, you should be able to make all the crazy things you want with this program. For more detailed information about specific tools, please see the tool-section of this documentation.

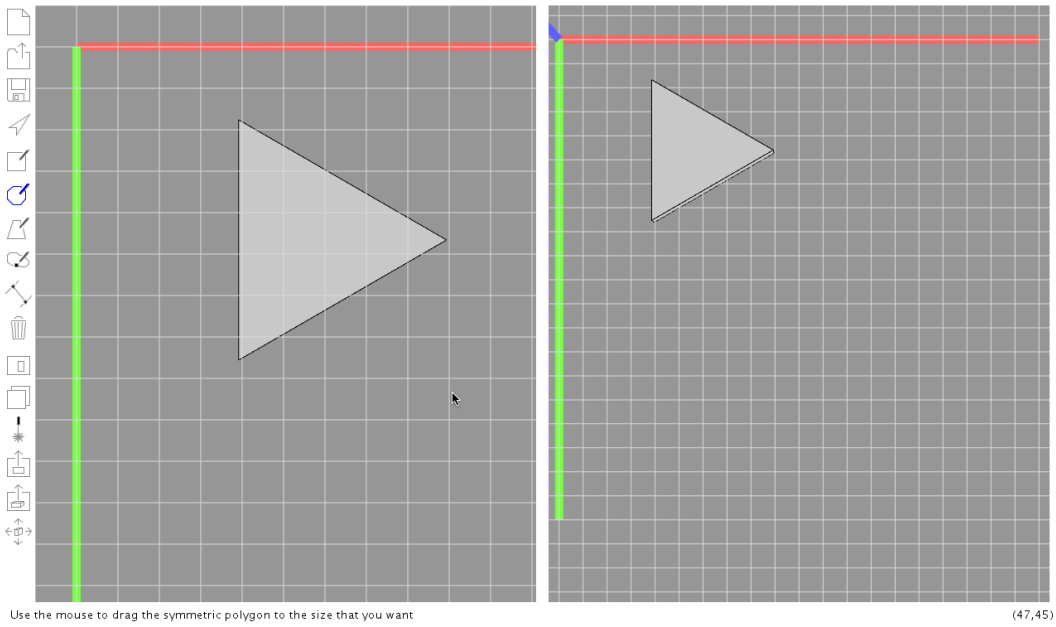


Figure 2.1: Drawing the first triangle

2.1 Building a Pyramid

To start things off, we will create one of the simplest three dimensional shapes one can make: a pyramid with a triangular bottom. For this shape, we will need four triangles with the same length and connect their edges.

Symmetric Polygon Tool



In the first step we select the symmetric polygon tool. It allows us to place shapes in the 2D-view on the left side. After selecting the tool, press the left mouse button at any position in the 2D-view and drag the mouse curser. A triangle will appear. When you drag the mouse, it will shrink or get bigger depending on the distance you dragged the mouse. When the triangle seems to have the right size, release the mouse button. Congratulations, you have created your first shape. Do not worry if it does not have the exact right size or is at the wrong position.

Select-Tool



These things can easily be fixed using the Select-Tool. After selecting the Select-Tool, move the mouse pointer over the triangle in the 2D-view (as the 3D-view is mostly for

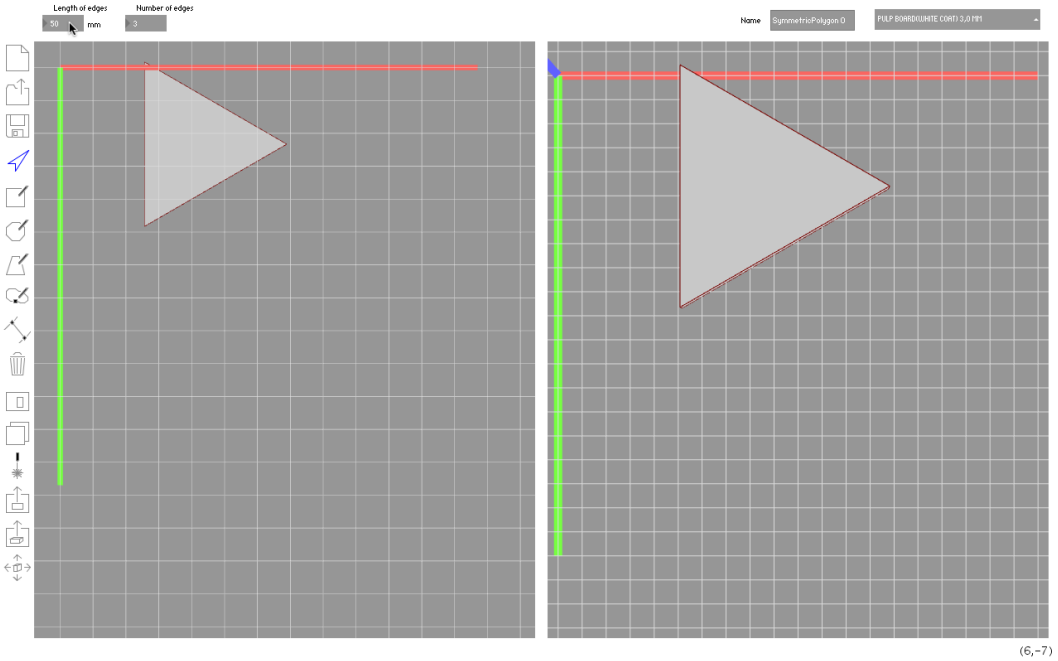


Figure 2.2: Changing the dimensions with the select tool

control and preview – the real work gets done in the 2D-view). You will notice that the triangle is now drawn with bright red border lines. This highlighting indicates that you can now select the highlighted shape by pressing the left mouse button. When you select the triangle, its border lines will be drawn in a darker shade of red, indicating that the shape is selected. You might have noticed that some text fields and a dropdown-menu appeared in the properties bar on the top of the screen. These represent the parameters of the selected object. In case of our symmetric polygon, you can change the length of its sides, the number of sides (currently three) and the name of the triangle as well as its material. For other objects, different parameters might be available, but for now let us not worry about these controls too much.

The first input field to the left can be used to change the length of the sides of the triangle (since it is a symmetric polygon, all sides will have the same length). For now, set its value to 40mm.

With the next input field you can change the number of

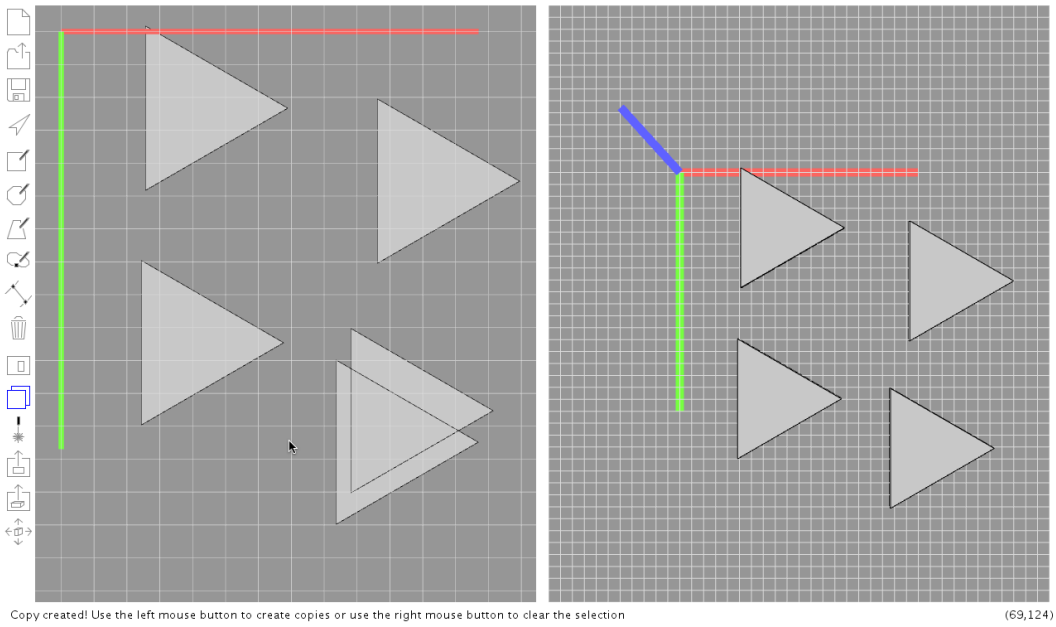


Figure 2.3: More triangles

sides of the polygon. Since we want a triangle, you do not need to change anything here, but feel free to experiment.

You can reposition any shape by clicking on it with the right mouse button and dragging the mouse. The scroll wheel of the mouse can be used to zoom in and out, if you need more space to work on. If you want to reposition the camera in the 2D-view, you can do so by clicking on an empty spot of the 2D-view with the right mouse button and dragging the mouse.

These controls may become useful very soon, since we need to add three additional triangles and might need the additional space. You could of course create those triangles by hand just like the first one. However, CutCAD includes a tool that makes this step a lot easier:

Copy-Tool



Select the Copy-Tool and click on the triangle you created. You have now selected this shape and can place copies of it where ever you want. Find an empty spot on the 2D-view and move the mouse there. A copy of the triangle you selected will follow your mouse. With a click of the left mouse

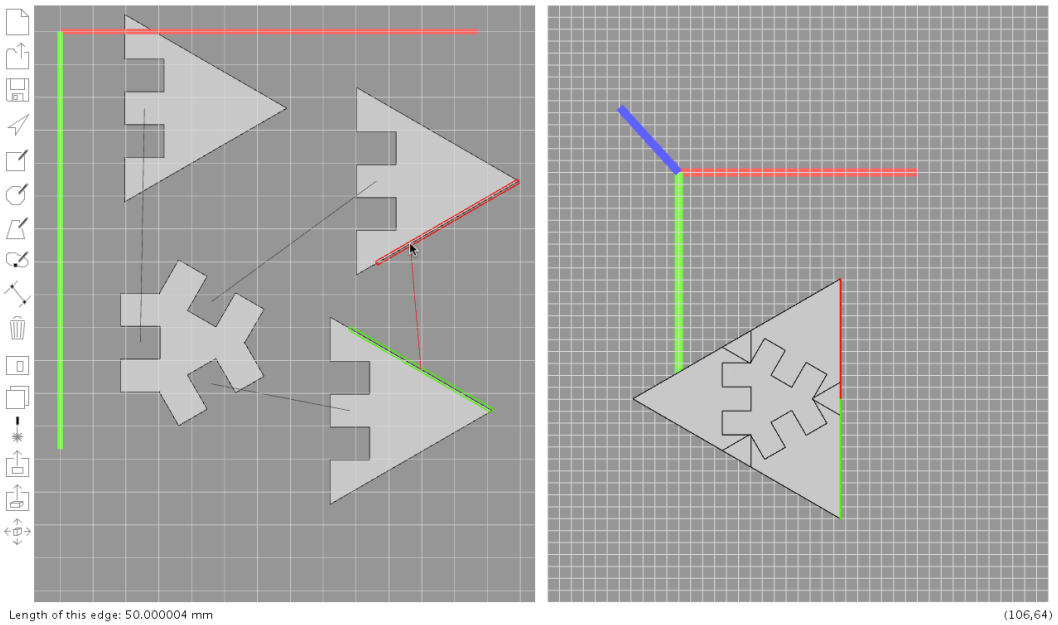
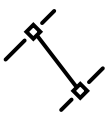


Figure 2.4: Connected a triangle on each side of the central one

button, you can place a copy of the triangle right where it currently is displayed. As long as you keep the Copy-Tool selected, you can place additional copies of the originally selected shape. Place three copies around the original one (or where ever you want to).

Now we have four triangles – not bad, but also not very impressive. We could have done the same with a number of different tools. So let us start doing something more interesting.

Connect-Tool



We will now start doing what CutCAD was made for: Connecting shapes. Click on the Connect-Tool and click on an edge of one of the triangles. You might have noticed that the edge is now highlighted in green and a line is following your mouse pointer. This line indicates the connection you are trying to connect. You have selected the first edge to be connected, but we need another edge to connect it to: Click on an edge of another triangle. The two triangles will now be aligned in the 3D-view on the right side of the screen and tenons will appear on both edges that make up the connection. Select a neighbor-edge on one of the trian-

gles and connect it with an edge of the next triangle.

Now three triangles are aligned in the 3D view. Highlight the edges by moving your mouse pointer over them until you find one of the two edges that form a line. Click on it and connect it with the other edge in that line. Now you should see the beginnings of the pyramid forming.

If you could follow along thus far without any problems: Great. If you accidentally connected the wrong sides, don't worry. Connections are reversible, and we can try that out by using the Delete-Tool.

Delete-Tool



Select the Delete-Tool and pick one of the connections. Move the mouse over the line representing the connection in the 2D-view. Click on the line and the connection will disappear – The shapes will lose their outline and will be moved back to their original position in the 3D-view. The Delete-Tool can also be used to remove shapes and cutouts.

Now recreate the connection you just delete it – we still want to build that pyramid, after all.

Just keep on connecting the other edges. If you are not sure which edges to connect, have a look at the 3D-view and see if the green and red edges fit together.

When that is done for all edges, you can admire the resulting pyramid and its tenon structure in the 3D-view.

Save-Tool



If you want to keep it for later: Click on the save tool to store somewhere on your computer so you do not lose all that work you put into building it.

But we still want to get a real thing out of this: Time to power up the laser cutter.

Cut-Tool



Click on the Cut-Tool. This will open a dialog with which you can send the shapes you just created to the lasercutter. On top of the dialog, you will see the cutting area, where you can place the shapes one by one by clicking on their names in the list on the left below the cutting area. You can move shapes placed in the cutting area by clicking on them

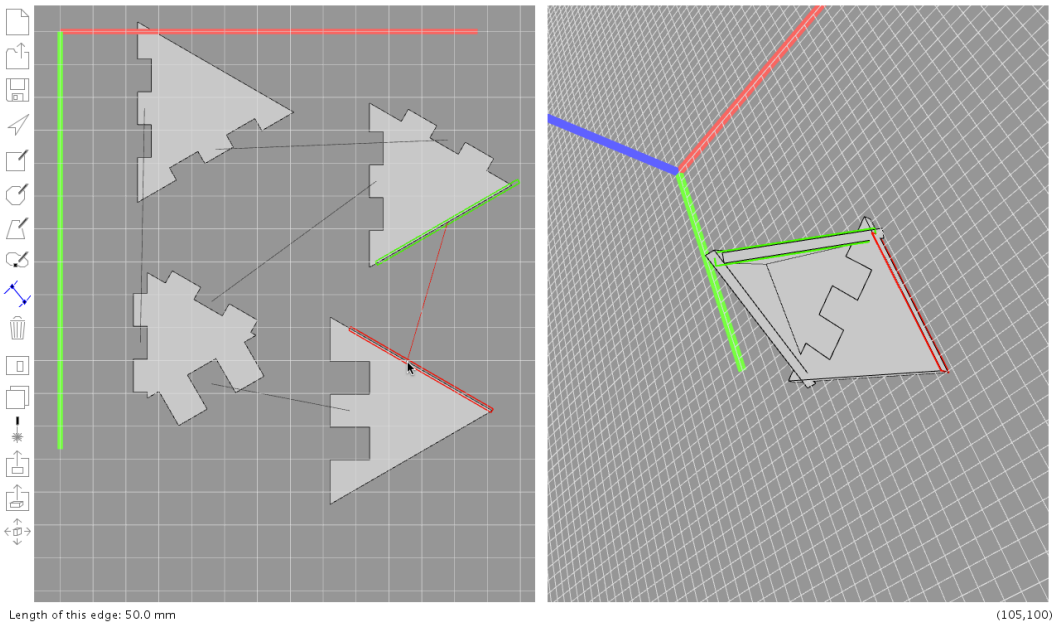


Figure 2.5: Connecting two would-be-neighbour sides

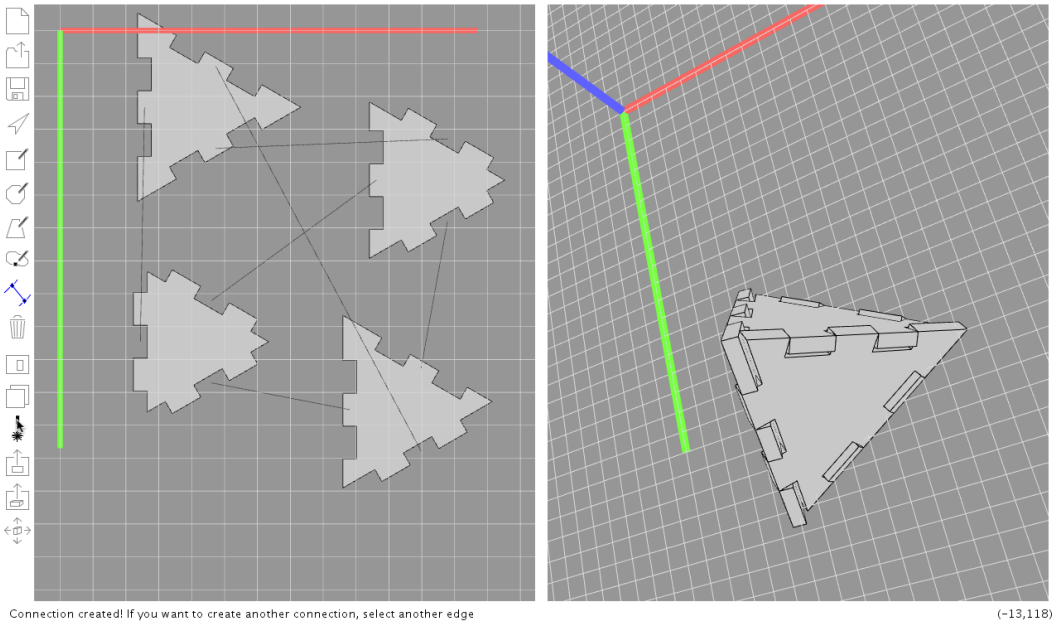


Figure 2.6: All connections done. Pyramid ready to launch

with the left mouse button, holding it down and dragging the mouse. If you had used different materials for the triangles, you would see a list of different materials on the right of the list. Each material will have its own list of objects and cutting area associated with it (you would need to cut them out of different sheets of material after all).

After you are done placing the four triangles, you can choose to store them as a SVG-file (if you do not have a lasercutter readily available) or to cut them right now. By clicking on the "Start Cutting"-Button, you can send the shapes directly to the lasercutter, which will then start to cut them out of the material you have placed inside its cutting area.

When everything is cut out, puzzle the form together and enjoy.

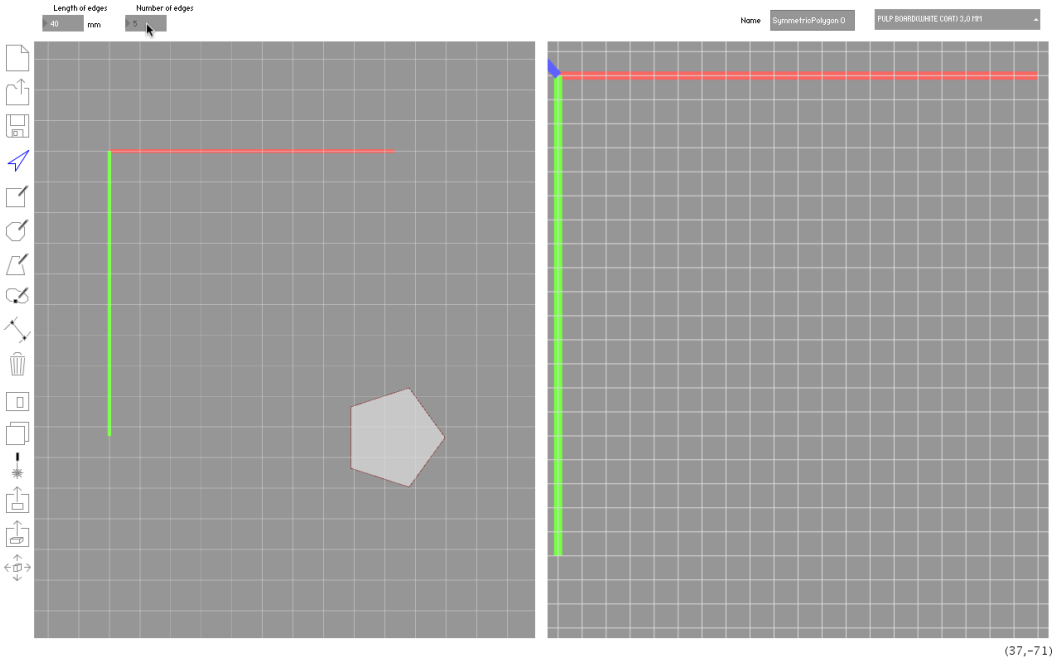


Figure 2.7: Change the number of edges of the symmetric polygon to five

2.2 Building a Dodecahedron

When you first selected a symmetric polygon (the triangle from our last example), we pointed out the parameter for the number of sides. After all, a triangle might be a symmetric polygon, but we could have named the Symmetric Polygon Tool just Triangle-Tool if you could not do more with it than creating simple triangles.

In this example, we will have a look at symmetric polygons with more sides and also learn something about cutouts.

First, select the Symmetric Polygon Tool and create a new triangle. Then select it and change the number of sides to five. The length of an edge should again be something over 40mm. Use the copy tool to make a copy, select it and make it slightly smaller – Lets say 30 mm.

Symmetric Polygon Tool



Next, you can use the Cutout-Tool to create a cutout in the bigger shape with the form of the smaller shape you just created. First, select the Cutout-Tool. Now click on

Cutout-Tool



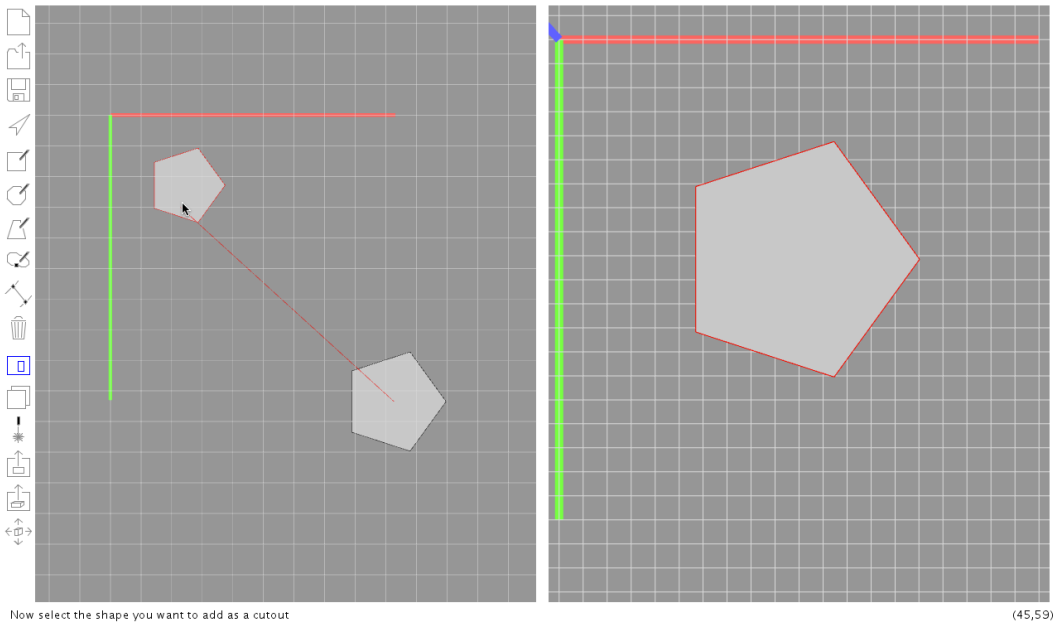


Figure 2.8: Add the smaller shape as a cut out to the bigger one

the bigger shape and after this on the smaller shape. A cutout will appear in the middle of the bigger shape, and the smaller shape will have no material and simply disappear from the 3D-view. If you do not want a cutout, but instead create an inlay from another material, you can select the smaller shape again and reassign a material. If you select the cutout with the Select-Tool – either by clicking on the cutout within the bigger shape or on the line connecting both shapes – you can move it around either by dragging the mouse or with the input fields in the properties bar. But for this example, the position is fine as it is. If you changed it, and do not want to find the middle manually, you can just use the delete tool to remove the cutout (by clicking on the line connecting both shapes) and recreate it – A new cutout will always be created in the middle of the shape it is cut from. Of course, you are not not limited to shapes of the same form – you could use any shape you like as a cutout, as long as it fits inside the bigger shape.

Copy Tool



Now select the copy tool and make eleven copies of the bigger shape. The cutout will automatically also copied (for inlays, you would also need to copy the smaller shapes,

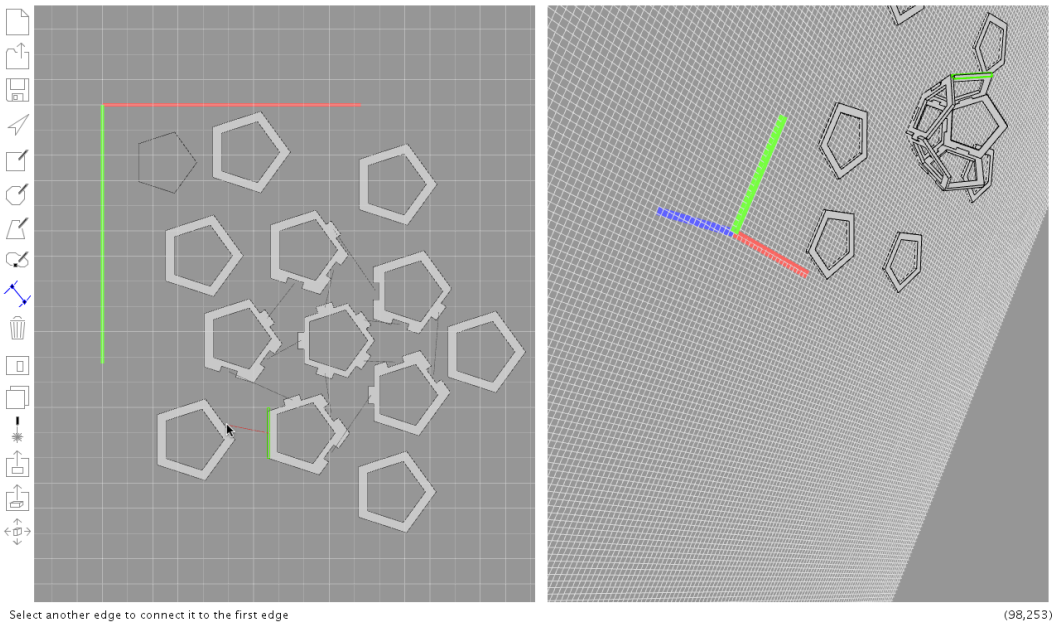
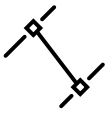


Figure 2.9: Many copys and connections

otherwise you would only have one of those to fill eleven holes – that would not work very well). A good layout to make the connections easier would be two circles of five forms around the first one, which would resemble the structure of the dodecahedron in the 2D view.

Connect-Tool



Now to the difficult part – Connect all edges. The best way to go about this is to work systematic from the center outwards. First, connect each edge of the center shape with one edge of another shape (from the inner ring). Next, select one of these shapes with one connection. Select one of the neighbor-edges of the already connected edge and find the corresponding edge on another shape (which has a common point with this edge). Select the next neighbor-edge on the latter shape and make the next connection as before. Go on until a bowl is formed and each of these five shapes has only two unconnected edges. Now connect them systematically with the next ring of five forms (keep in mind, that the edges you highlight with the mouse in the 2D-view are also highlighted in the 3D-view – this can help a lot to find your way around connecting more complicated objects). Now close the dodecahedron by connecting the fi-

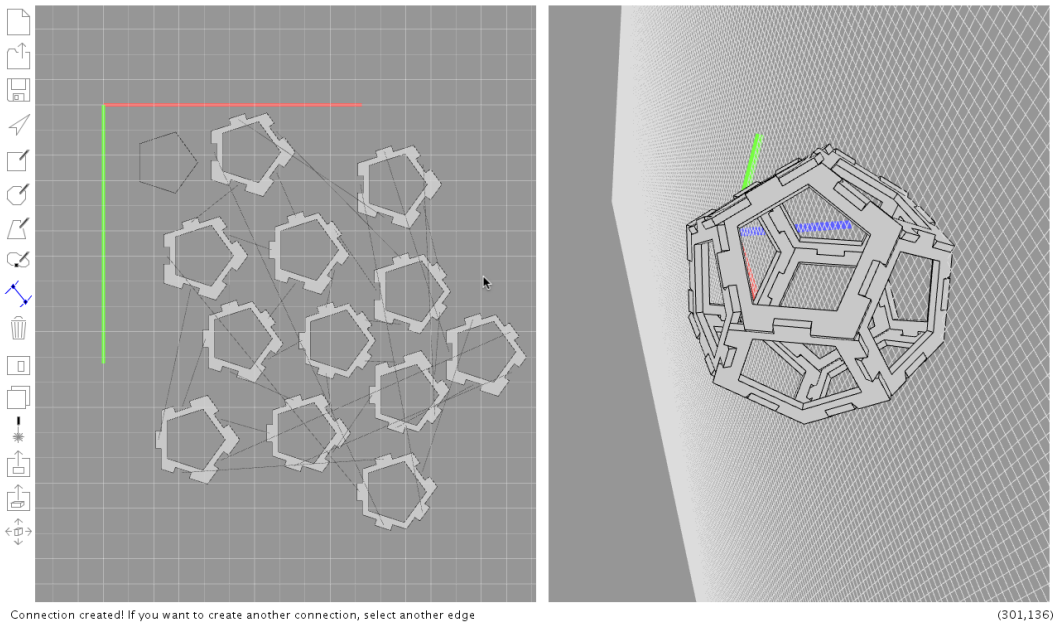


Figure 2.10: Done!

nal shape.

Lasercut it and enjoy!

Import SVGs



To create more complex cutouts and shapes you can also import SVG-files. These can be handled like normal shapes – and paths which are enclosed in another shape will be already be assigned to them as cutouts. This allows you to make cuter dodecahedrons to impress you friends.

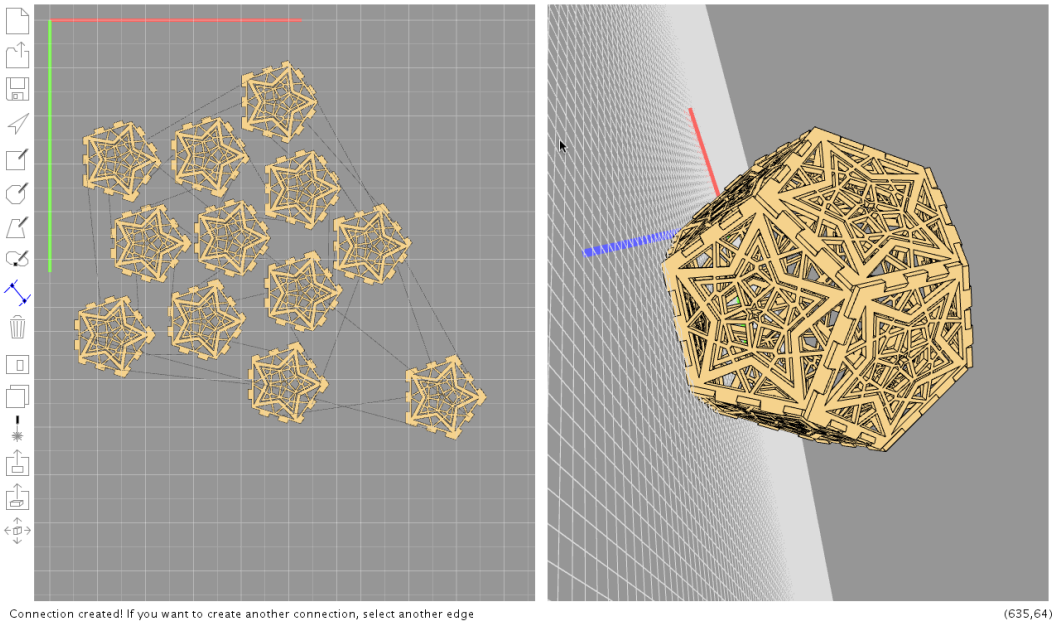


Figure 2.11: More Complex cut-outs

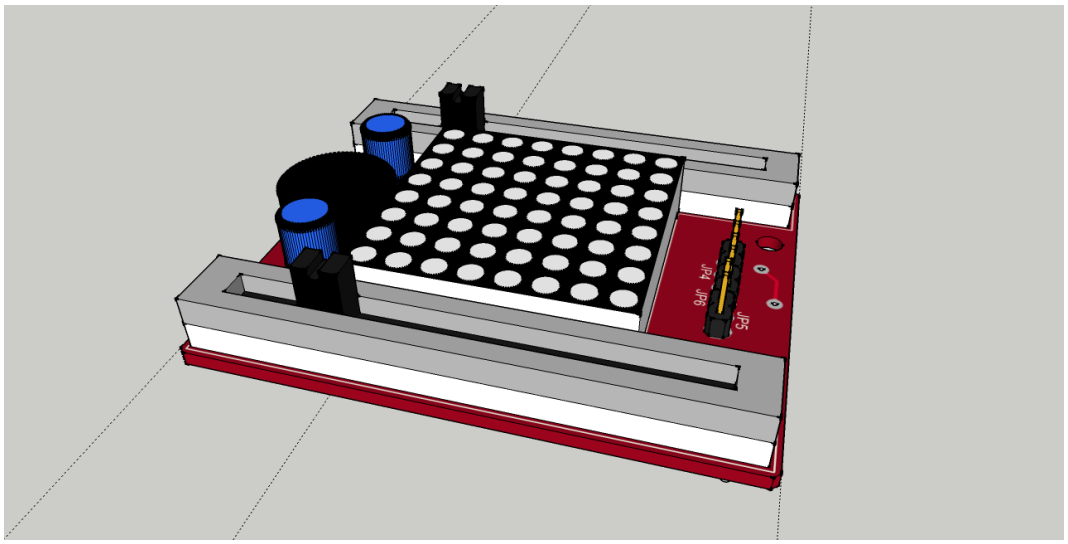


Figure 2.12: SketchUp rendering of the eagle board file after eagleUp usage

2.3 Building a Housing for an Electronics Device

In this example we want to show you how to use the STL-import function to build enclosures. There are different ways of acquiring STL-files for your project. For some devices, you may find STL-files online. Maybe you created the model youself in programs like sketchUp[1] or openSCAD[2]. Another way to obtain a STL-file of your electronics project is creating a 3D object of your PCB in eagle. Just install eagleUp[3]. EagleUp lets you create an ULP-file from your PCB. You can open this file using tools such as sketchUp.

With an extension for Sketchup to allow STL export[4] you can then transform them to STLs. If you do this, make sure to use mm as units and create a binary STL files. The resulting file can then be imported into our program.

Here, we will make a pong game we found on youtube[5].

[1] http://www.sketchup.com/de
[2] http://www.openscad.org/
[3] http://eagleup.wordpress.com/installation-and-setup/
[4] http://extensions.sketchup.com/en/content/sketchup-stl
[5] http://www.youtube.com/watch?v=xTCogmx0h1g

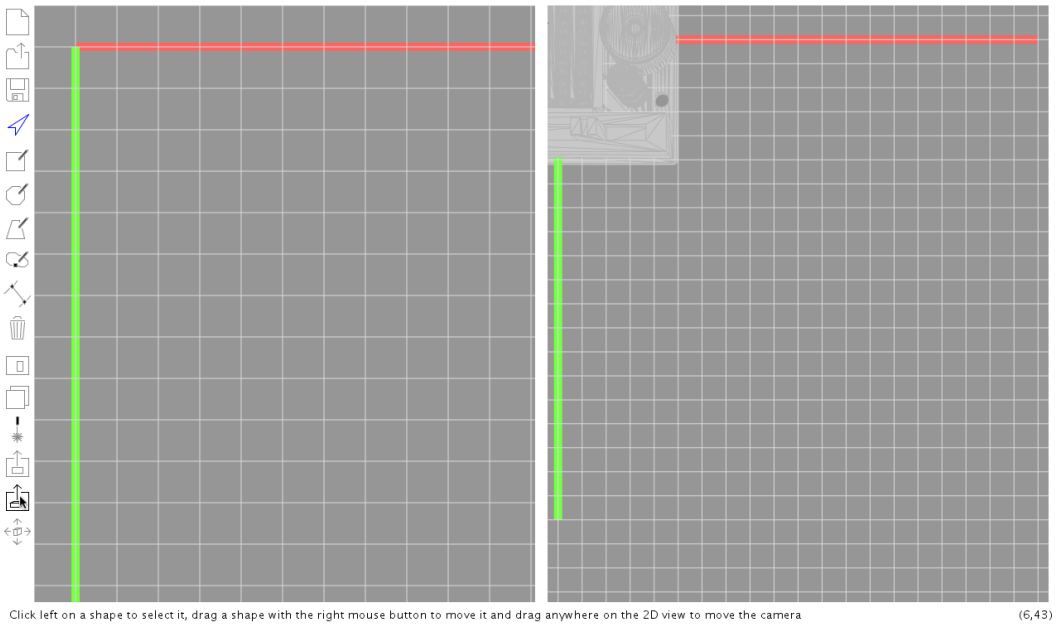


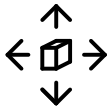
Figure 2.13: Imported STL file

Import STL-Tool



Click on the Import STL-Tool and browse to the STL-file that you want to import in the file dialog. As soon as you open it, it will appear in the 3D-view. You can not select it directly with the select tool (which works only in the 2D view, while the STL-import only appears in the 3D-view), but there is another tool for that:

Change STL-Tool



By selecting the Change STL-Tool, you get access to the properties of the STL-object, including its position and rotation in the 3D-view.

Rectangle Tool



Now that you loaded up the STL-object, it is time to build the enclosure. First, draw a simple rectangle using the Draw Rectangle-Tool. Move the STL-object such that it shares the same top left corner with the rectangle. Now you can select the rectangle and change its width and height until it has the same dimensions as the STL-object. You will have to add the material thickness to width and height, because the inner dimension will become smaller when the sides are added. Maybe add a small safety buffer as well.

Repeat this for the sides: Copy the rectangle (one side has

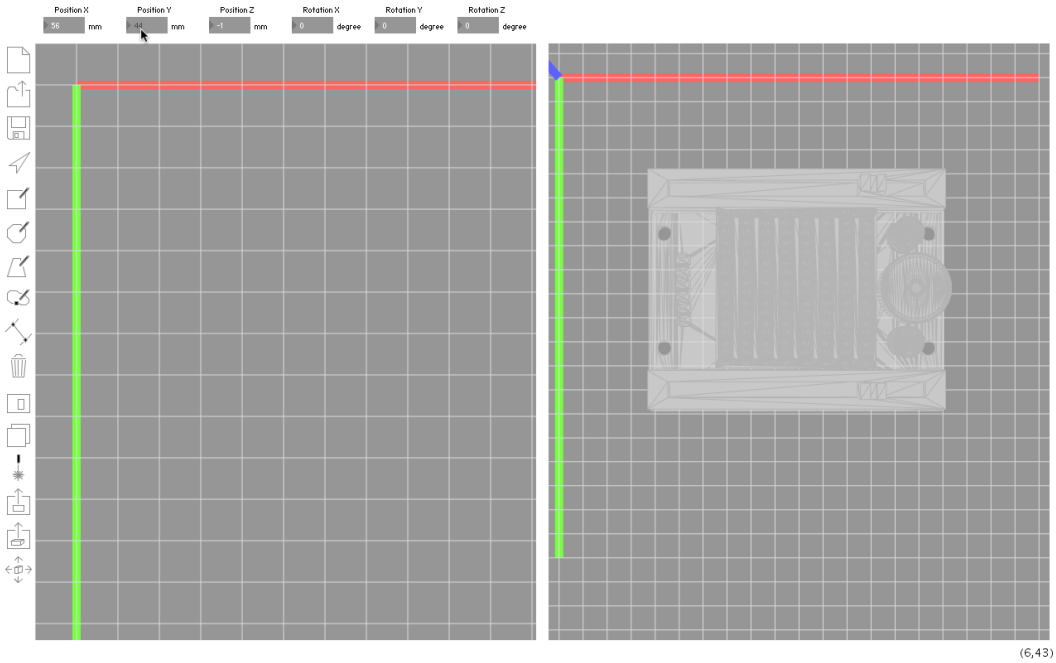


Figure 2.14: Moved STL file

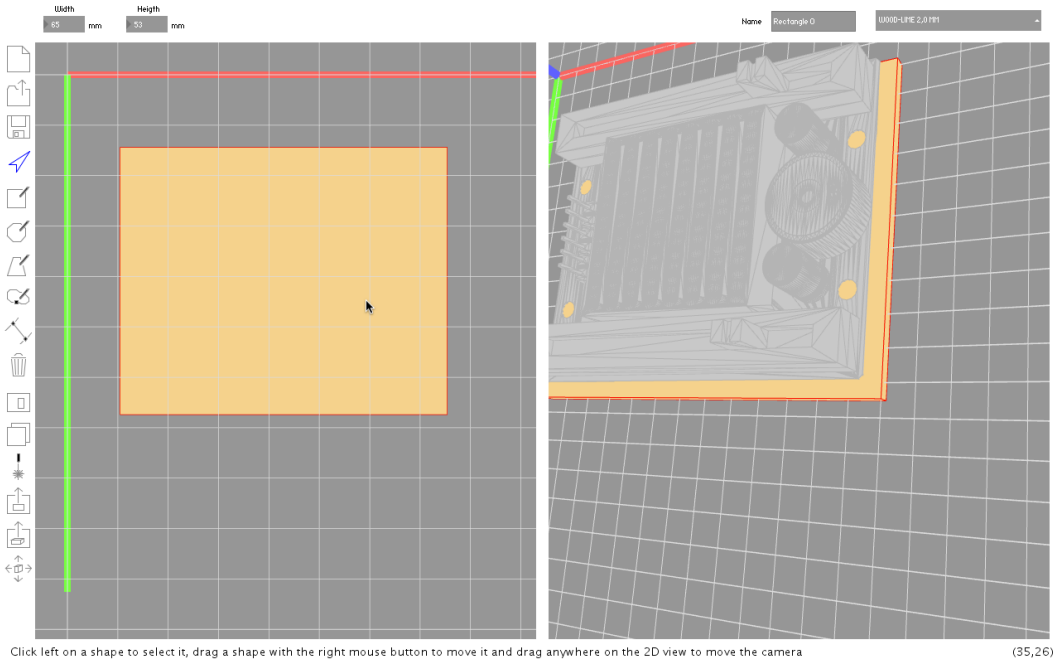


Figure 2.15: And a Rectangle...

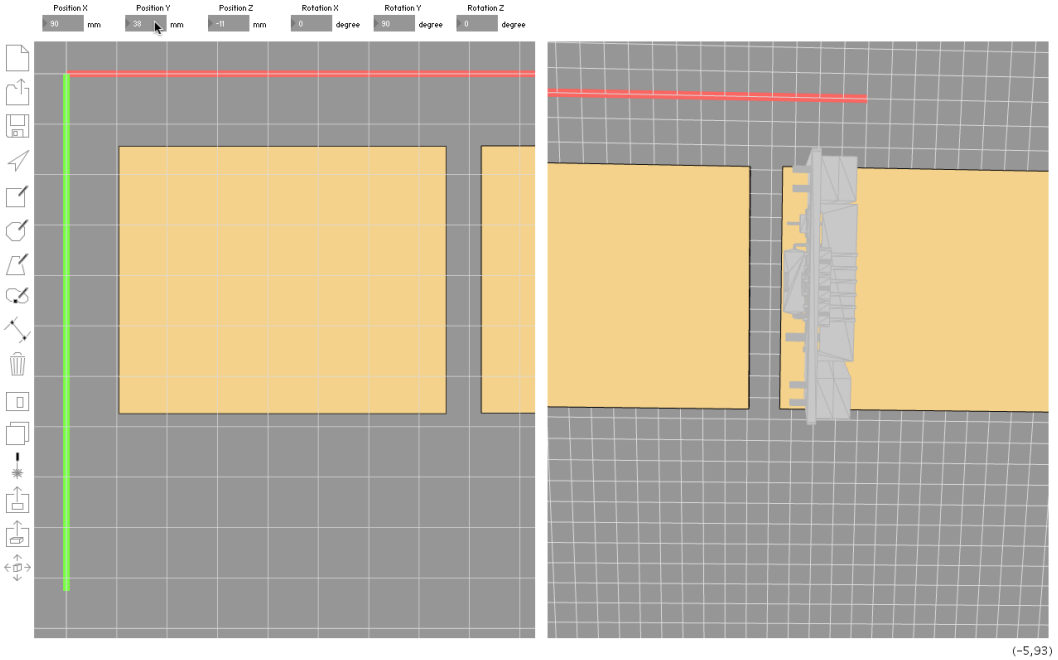


Figure 2.16: ...and another one

already the right size) and rotate and move the STL object above it. It might be useful to change the z-coordinates such that the biggest parts intersect with the rectangle. This will make it easier to find the right dimensions. Since we want to include a battery, we make the enclosure a bit higher.

Now we can copy the side and create the rest of them. Connect them to form a box and move the STL-object inside it to check if it actually fits.

Add a copy of the bottom rectangle as a cover and move the STL-object upwards such that you can see the components for which cutouts are needed. In this case we need three cutouts - two identical ones for the slider and another one for the LED matrix in the middle. Just add two additional rectangles and use the Cutout-Tool to add them as cutouts to the cover. Select the cutouts on the cover to move them on the right position. If you want to change the dimensions of the cutout, you can select the original shape (now only shown as a transparent shape in the 2D view). Now just use the Cutout-Tool a second time on the same rectangle

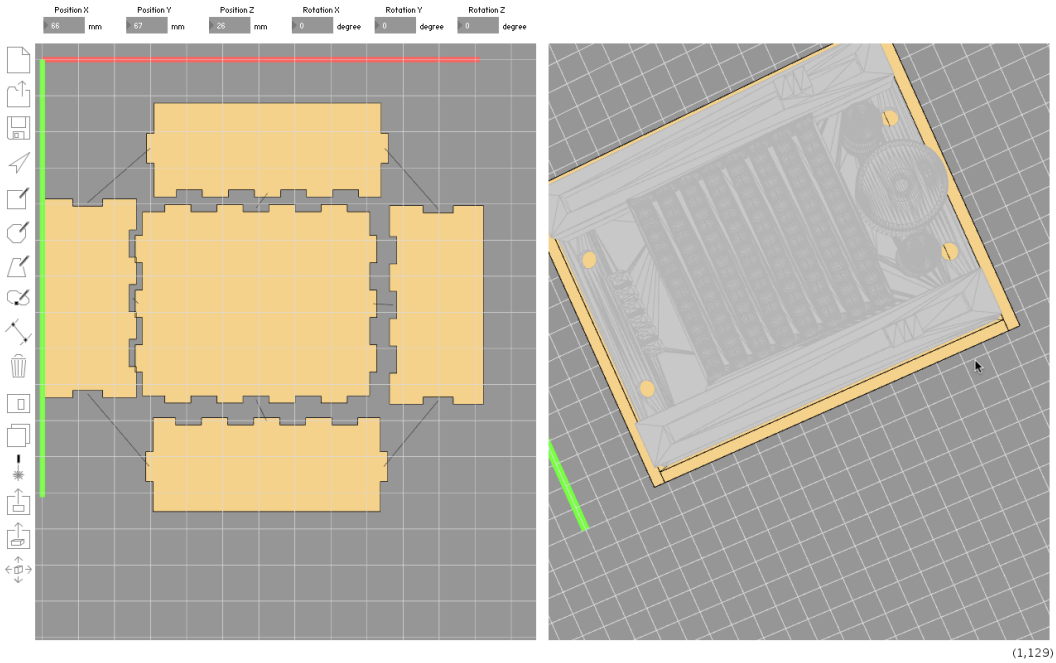


Figure 2.17: PCB inside a box

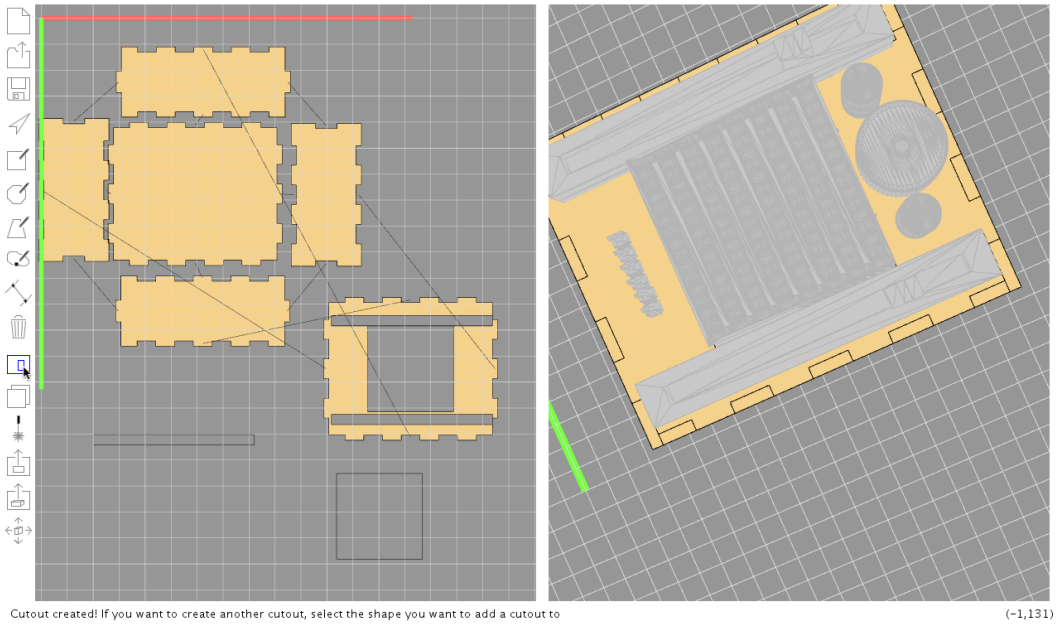


Figure 2.18: Cut-outs for slider and LEDs

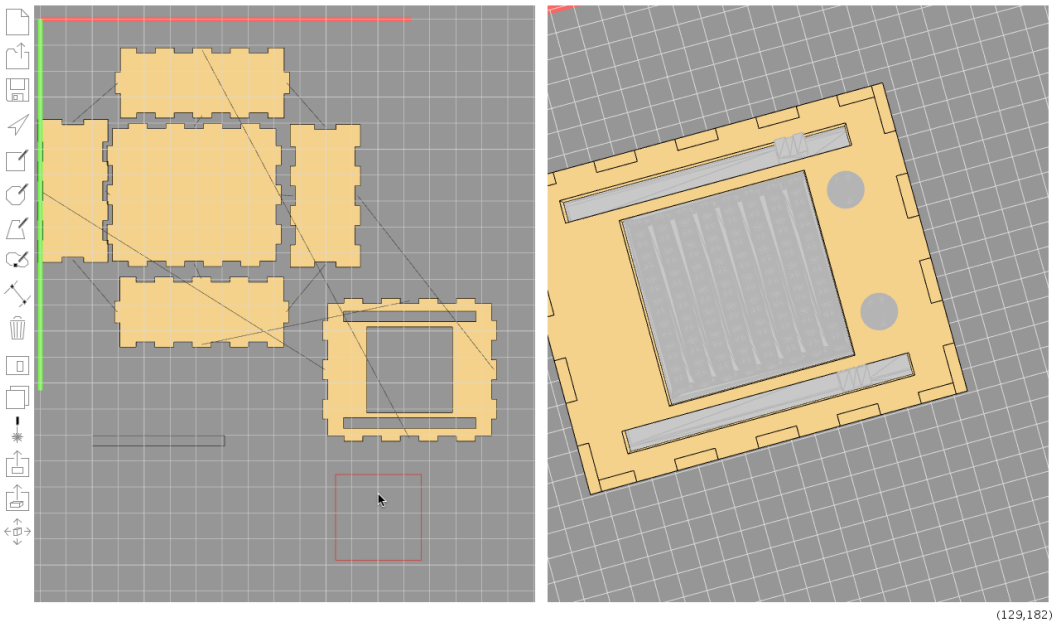


Figure 2.19: Fine tuning

you used for the slider cutout. You could of course just add another rectangle, but since you already made sure that the cutout has the correct dimensions for the slider, you might as well use it again. Also, this way you can correct the dimensions for both slider cutouts at the same time if needed.

You might also want to add holes for screws (e.g use symmetric polygons with a high number of edges), and either directly cut it or export it as an SVG-file in the cutter dialog to add some engravings (like text) with Inkscape.

FAB SCAN:
A great source for stl-files for your own electronic devices is the FabScan: Just make a 3D scan of the object, convert it to an STL-file and import it into our program. No more making measurements where holes for button and leds belong.

Fab scan

Chapter 3

Tool Time

In this section we will describe the different tools you can find in the toolbar on the left side of the screen. These tools are the main way to interact with this program. However, there are two things you can do without selecting a specific tool:

- Zooming: Simply move the mouse over either the 2D- or 3D-view and scroll the mouse-wheel. This will result in zooming in/out of the corresponding view.
- Cameramovement in 3D-view: Whenever your mouse pointer is located over the 3D-view, you can move the camera. Pressing the left mouse button and moving the mouse will rotate the camera, while pressing the right mouse button and moving your mouse will pan the camera.



Select

The Select-Tool can be used to select objects in the 2D-view and manipulate them. When placing the mouse pointer over any object in the 2D-view, the object will be highlighted in red. Pressing the left mouse button will select the highlighted object, which will show its selection status with a deeper red color.

To change the properties of the selected object, use the so-called properties bar at the top of the screen.

Shape

Every shape will give you the option to select its material as well as the thickness of the material from a drop-down menu at the right side of the properties bar. Selecting a different material or thickness will update the color and thickness in the 2D- and 3D-view. Even the materials of already connected shapes can be changed. In this case, the tenons of the connected shapes might be recalculated depending on the new thickness of the material.

Other properties, mostly the length of edges will be displayed as number input field. These inputs will only work for unconnected shapes, since changing the length of the edges might disturb the connections. These field will nevertheless appear for connected shapes, which allows you to check the dimensions and reuse these values for further shapes.

Right clicking and holding the mouse button allows you to drag the object to another position in the 2D view. If no object is selected, you can move the viewpoint in the same way.

Connection

Selecting a connection will give you the option to manually rotate one shape around the other. The angle can be controlled with a number input field. Normally, the program will rotate shapes on its own to find a way to make a connection possible, but this manual control can be used to determine special positions for the shapes without relying

on the automatic rotation. For example, one could create a free form kind of bird wing out of plane pieces, where are no constraints for which angles are used between the planes.

Cutout

Selecting a cutout of a shape will allow you to move or rotate the cutout.

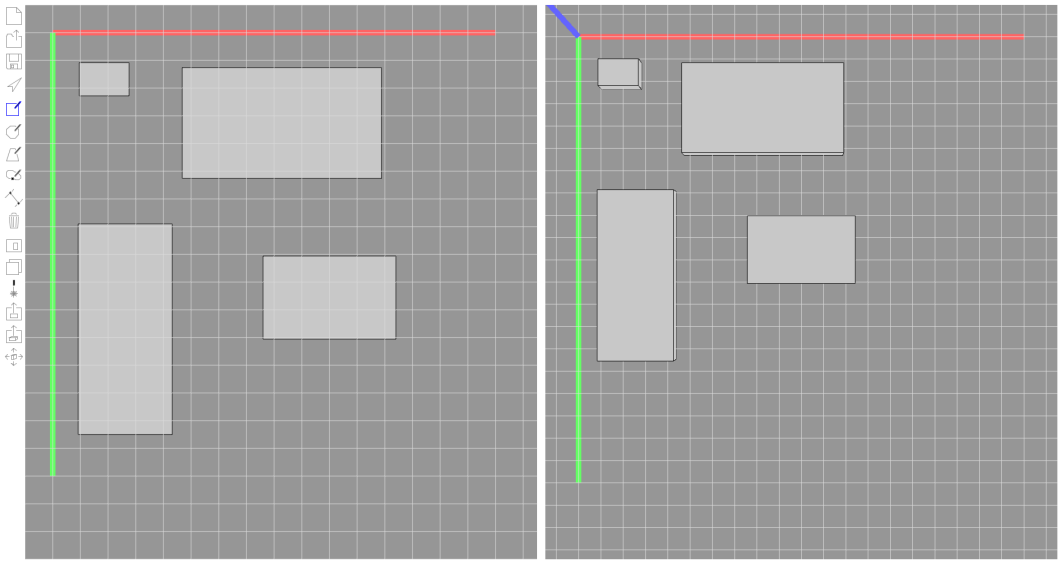


Figure 3.1: The "Draw Rectangle"-Tool can be used to create a simple rectangle.



Draw Rectangle

To create a rectangle, simply click on any free space of the 2D-view and hold the mouse button down. You can then drag the mouse to determine the size of the rectangle. As soon as you are happy with the size of the rectangle, let go of the mouse button. After you released the mouse button, the new rectangle will be created.

If you want to change the width or height of the rectangle later on, you can select it with the Select-Tool and then change its parameters in the properties bar on top of the screen.

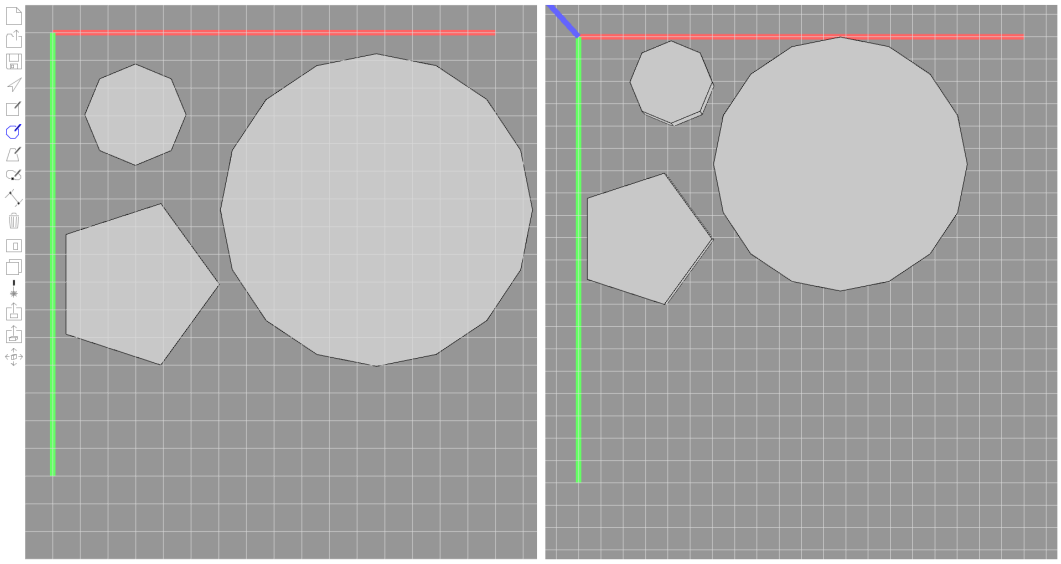


Figure 3.2: The ”Draw Symmetric Polygon”-Tool can be used to create symmetric, concave polygons.



Draw Symmetric Polygon

It basically works like the ”Draw Rectangle”-Tool. First, you have to click somewhere on the 2D-view. A triangle will appear at the position of the mouse pointer. By holding the mouse button down and dragging, you can change the size of the triangle. After you released the mouse button, the new polygon will be created.

You can change the number of sides as well as the length of each side just like with the ”Draw Rectangle”-Tool: Select the polygon with the Select-Tool and change the parameters of the polygon in the properties bar on top of the screen.

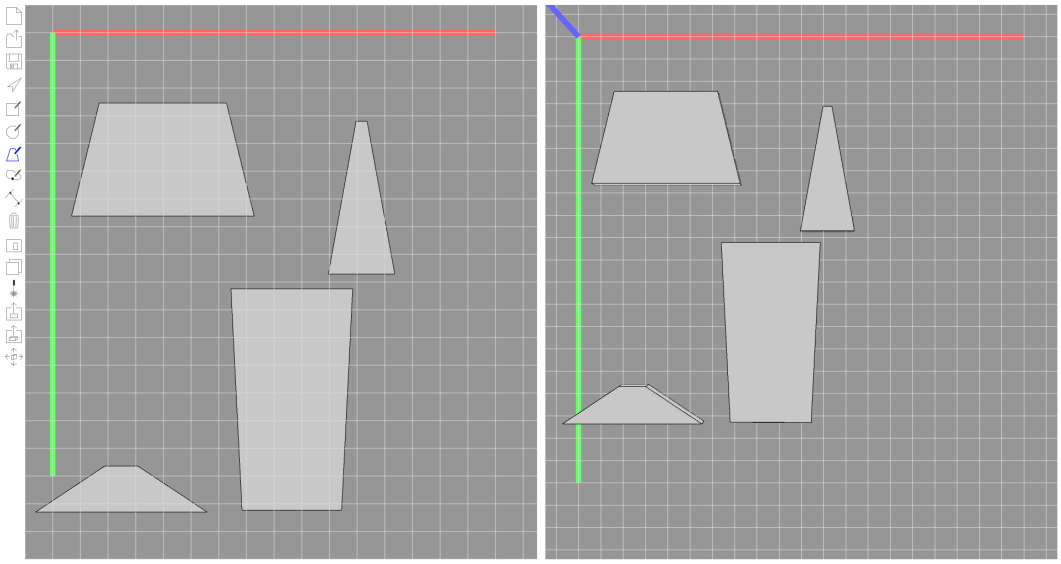


Figure 3.3: The "Draw Trapezium"-Tool can be used to create a trapezium.



Draw Trapezium

Again, creating the trapezium works basically like the other draw tools: You first click on an empty space in the 2D-view, drag the trapezium to the size you want it and let go of the mouse button. At first, it will have the shape of a simple rectangle. To change this, select the rectangle with the Select-Tool. You will notice that you now have more parameters in the properties bar than for a simple rectangle:

- Top: The length of the top-edge of the trapezium
- Bottom: The length of the bottom-edge of the trapezium
- Side: The length of the two sides connecting the top and the bottom of the trapezium

By adjusting these parameters, you can create trapeziums that you can still easily use to connect them with other shapes.

Beware of impossible Values

There are some combinations of values for these parameters that will result in impossible shapes. If for example the length of the top plus the length of the length of both sides is smaller than the bottom, the resulting trapezium will have a height of zero. If you happen to come across such a configuration, just change the parameters back to more realistic values to obtain a normal trapezium again.

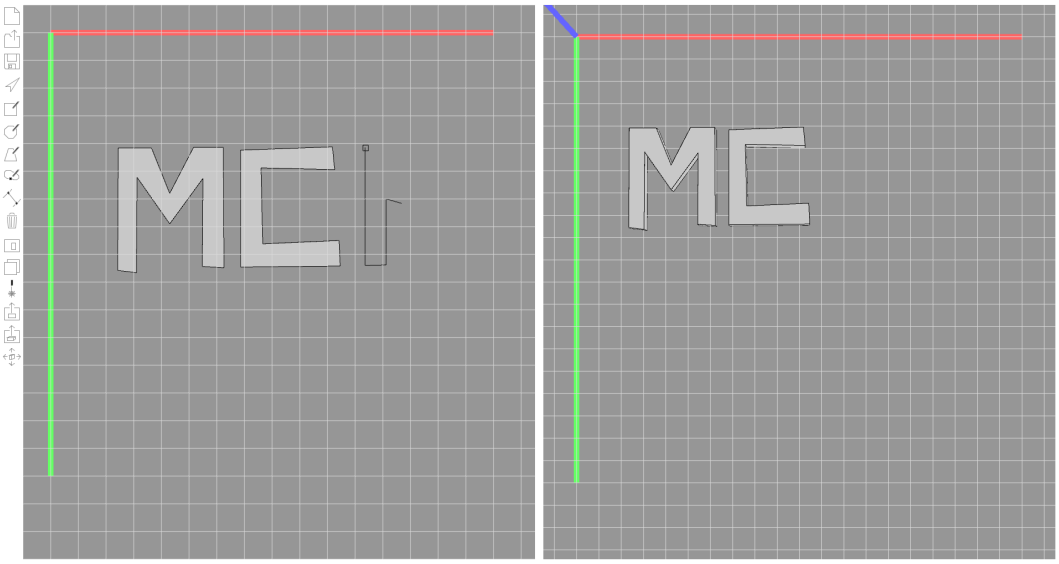


Figure 3.4: The "Draw Polygon"-Tool allows you to create custom shapes by simply drawing their outline.



Draw Polygon

If you want to create a custom shape that is more complex than the standard shapes, but not complex enough to justify opening up inkscape and importing it as a svg-file, the "Draw Polygon"-Tool is for you. You can use it to create any shape by drawing a path.

After selecting the tool, you can start your shape by clicking on any free space on the 2D-view. You will see a small rectangle appear where you click. This is the starting point of your shape. You can now click on another spot in the 2D-view to draw a path from the first rectangle to the point you clicked on. Everything that is within the path you drew will be part of the newly created shape. To finalize the shape, simply close the path by clicking on the first rectangle.

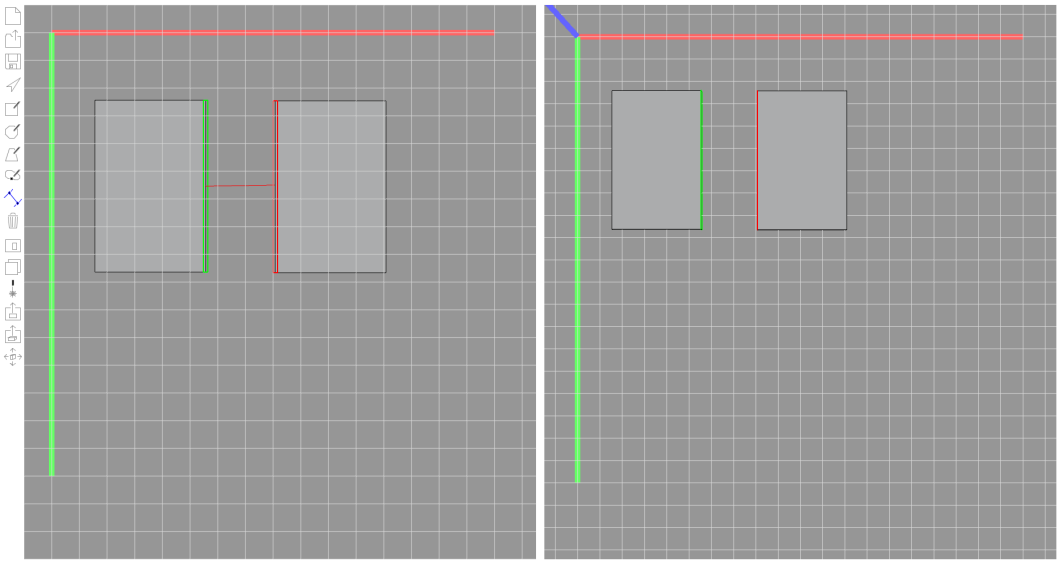
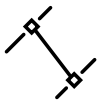


Figure 3.5: The Connect-Tool before the final click: The edge highlighted in green was selected first, the edge highlighted in red is the second edge to be selected



Connect

The connect tool is used to connect one edge of a shape with an edge of another shape. Only edges of the same length can be connected to each other.

First Edge

The first step after selecting the connect-tool is to select the first edge: When placing the mouse pointer over an edge, a red box will appear around the edge both in the 2D- and 3D-view. When clicking on the red box, the first edge is selected. As soon as the first edge is connected, the box surrounding the edge will turn green and will stay like this as a reminder which edge was selected for the first edge of the connection.

Second Edge

In the next step, the second edge to be connected to the first edge has to be selected. In order to do this, move the mouse pointer over another edge. If this edge can be connected to the first edge (i.e. if it has the same length), it will again be highlighted with a red box. In this case, the connection can

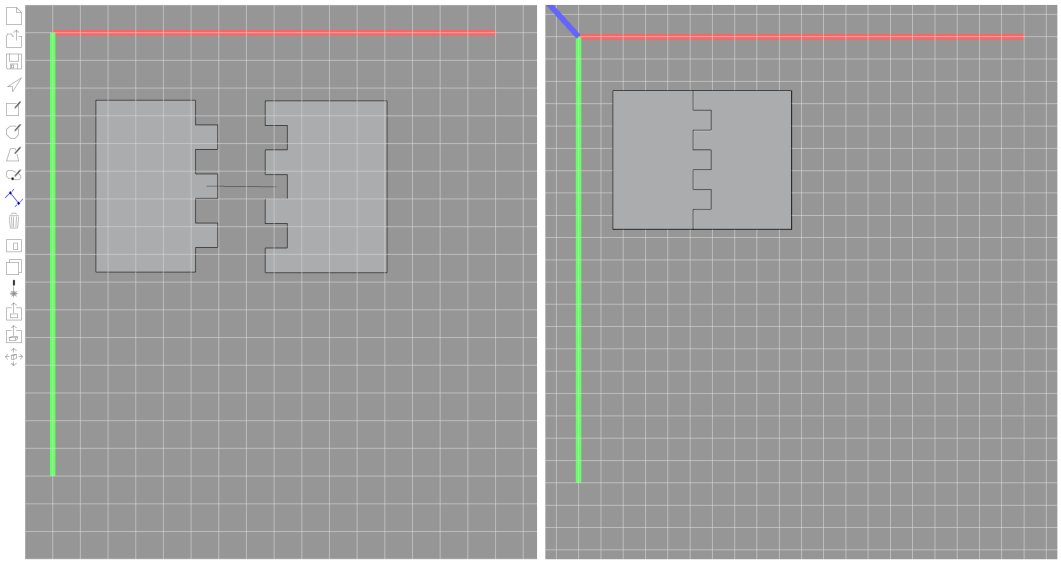


Figure 3.6: The Connect-Tool after the final click: Tenons for both edges have been created and the shapes were aligned.

be made by simply clicking on the currently highlighted edge. If the edge you moved the mouse pointer over does not get highlighted, please make sure that both edges have the same length.

If both edges were not yet connected to any other edges, have the same length and are not part of the same shape, a new connection should be created between them. However, there are some special cases in which it might be impossible to create a connection even though all these conditions are fulfilled.

If there are other connections that connect the shapes the selected edges belong to to other shapes, these connections might impose restrictions on the movement and rotation of the shapes and thus limit the options of connecting the edges. In those cases, our program will try to find a possible configuration in which the connection can still be made. If the program fails to find such a configuration, it will display an error-message in the status bar. Generally, this should only occur in cases where it is actually physically impossible to make a connection between the two selected edges while still keeping all previous connections intact.

We have however observed a few situations in which it was possible to make a connection, but the program failed to do so. In such cases, it can help to manually rotate the shapes such that the edges you want to connect are as close together as possible and then retry to make the connection. Sadly, we do not yet know when exactly this happens or how it can be helped.

TROUBLESHOOTING:
If connecting two edges does not work as expected, try the following things:

- Look at the 3D-view and make sure that the correct edges are highlighted. Sometimes, the position of the shapes in the 2D-view can be irritating.
- Try to make the connection the other way around.
- Try to manually rotate the shapes in the correct position and retry making the connection.

Troubleshooting

If the outline of a shape is incorrect (e.g. it should start with an intruded tenon instead of a extruded one), you can correct this by deleting the connection of this edge and making a new one. When more than one connections of the shape remain, it will stay at the same position, and there will not be any problems reconnecting the shapes. If both shapes are connected or both are unconnected, the first edge will usually start with an intruded tenon, while the second edge will start with extruded tenon. If only one of them is connected, the unconnected edge will start with the extruded tenon.

Wrong Outline

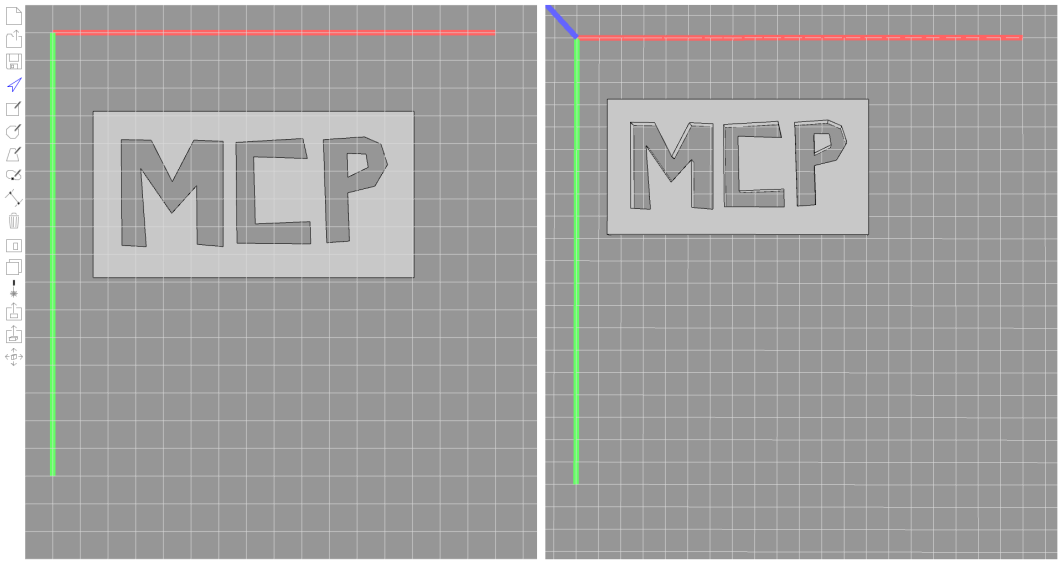


Figure 3.7: The Cutout-Tool can be used to add cutouts to existing shapes



Cutout

First make sure that you have already created the shape you want to add a cutout to as well as the shape you want to use as a cutout. As soon as you have those two shapes set up, you can start using the Cutout-Tool:

First, select the shape that you want to add the cutout to. In the second step, select the shape you want to use as a cutout. You should now see a cutout appear in the middle of the shape you selected first. You can use the Select-Tool to move the cutout within the first shape either by dragging the cutout while holding the right mouse button or by selecting it with the left mouse button and changing the values for its X- and Y-position in the properties bar. You can also set the angle of the cutout in the properties bar after selecting it.

Inlays

A newly created cutout will have the material "Nothing" assigned to it. This means that it will not be a part of the final product. In this case, it will only appear as a cutout,

but not as a shape of its own. It will also not appear in the 3D-view.

However, it is also possible to assign a different material to the cutout. This will create a shape of the selected material that you can use as an inlay for the cutout.



Copy

The Copy-Tool can be used to create as many copies of a shape as you like.

After selecting the Copy-Tool, click on the shape you want to copy. You will notice that a copy of the selected shape is now following your mouse pointer. You can place it anywhere in the 2D-view by clicking on an empty space in the 2D-view. This can be repeated as often as necessary to place any number of copies in the 2D-view.

By pressing the right mouse button, the currently selected shape is deselected, such that another shape to be copied can be selected.

The copy of the shape will contain all the cutouts of the original, but will not keep the same connections or outline. If you want to use the cutouts as inlays, remember to make a copy of the cutout as well, since otherwise you will not have enough inlays in the end.

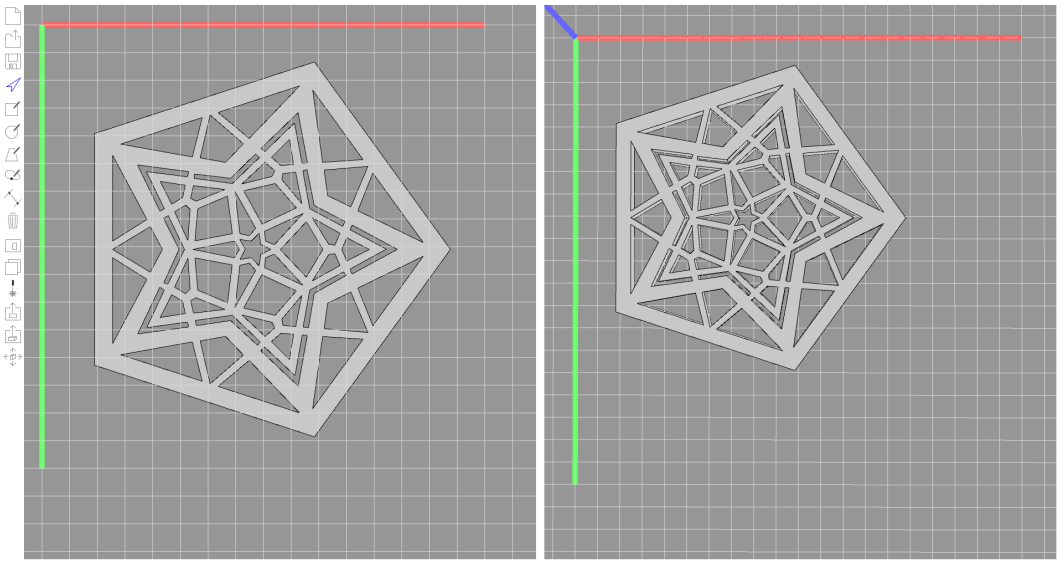


Figure 3.8: Import complex shapes with the SVG-import functionality of CutCAD



Load SVG

To create simple shapes, the tools CutCAD provides should be enough. If you however want to create more complex shapes, you might want to use a full-blown vector graphics editor (e.g. inkscape or illustrator). For this reason, CutCAD supports the import of SVG-files. To import a SVG-file into CutCAD, simply click on the "Load SVG"-Icon and select the file containing the shapes you want to import. You can import files containing multiple shapes and every shape will be imported individually. Cutouts will also be detected and created.

DPI issues

For several reasons, importing shapes from SVG-files is a bit tricky. The most important reason is that different Tools assume SVG-files to use different DPI-values. However, the DPI-value used is not saved with the file, such that the shapes in a SVG-file created and saved with Illustrator will have a different size if they are opened with Inkscape. We did not find a clean solution for DPI-detection, thus we decided to stick with the freely available tool Inkscape. There-

fore, we are using the same DPI-value as Inkscape, meaning that files imported from Illustrator will result in incorrect sizes of shapes.

Things to know about SVG

THINGS TO KNOW ABOUT SVG:

- As mentioned above: Use inkscape SVGs to get the right DPI scaling.
- Convert everything to path before importing.
- We use the logical positions of vertices. The contour in inkscape has usual a dimension, too. This is included in the displayed dimensions of a shape in inkscape. The imported shape is therefor smaller. To circumvent this: Use either a grid to align the vertices, or use no contour and filling instead.



Figure 3.9: The Cut-Tool will open a dialog to send the created shapes to a lasercutter or export them as SVG-files



Cut

When you are happy with the thing you constructed in CutCAD, it is time to make it become a reality.

Clicking on the Cut-Tool will open a dialog in which you can go through the last few steps necessary to get to the final product. You can either export the shapes you created into SVG-files to take them with you and cut them at a later point in time (e.g. if you do not have a lasercutter available at the moment) or you can directly send them to the lasercutter. Material settings and everything should be already be determined correctly with the selected material (make sure that you have customized the material list and lasercutter settings to your needs).

Place objects

But before we can start with the cutting, you first you need to place the objects for the lasercutter. The grey space in the upper left section of the dialog (1. in figure 3.10) represents

the cutting area of your selected lasercutter. Its size directly depends on the information CutCAD has about your lasercutter, so as long as you make sure that you have material sheets of the same size, it is going to be very easy to place the shapes correctly.

Before we start placing objects, have a look at the section below the cutting area: You will find a list of objects you can place at the left side of the dialog (2. in figure 3.10) and a list of jobs for the lasercutter directly to the right of it (3. in figure 3.10). For every material you chose during the construction of your objects in CutCAD, a separate job will be created. The list on the left will only show you the shapes made of the material for the selected job. All you have to do to place the objects is select each job (the button of the selected job will be painted black, see 4. in figure 3.10) and then click on the objects on the list on the left. As soon as you click on an object in the list, a representation of the object will appear in the cutting area. By holding down the left mouse button, you can drag the shape to another position in the cutting area. Right-clicking on a shape will remove the shape from the cutting area and add it back to the list.

Add additional sheets

If you happen to run out of space on the cutting area for a job, click on the "Add extra job"-Button in the lower left corner of the dialog (5. in figure 3.10). This will add another job of the same kind of material of the actual selected job from which you can place the shapes that did not fit into the cutting area of the original job. If you later on realize that you actually did not need this job (maybe you found a clever way to place all objects on the first job), do not worry: Empty jobs will not be send to the lasercutter and also will not be exported as SVG-files. After adding another job, this job will be the active one, and the layout of this job (objects placed in this job) will be shown.

After you have placed all the objects for all jobs, it is time to cut them. You can now choose to either export them as SVG-files or to directly send them to the lasercutter. If you missed a shape and did not place it or have overlapping shapes, a warning will be displayed.

Export svg

Clicking on the ”Export as SVG”-Button will save the shapes you created in CutCAD as separate SVG-files to your harddrive. There will be a SVG-file for each job, named after the material of the shapes that are saved inside the file, so it should be very easy to later on cut them on a lasercutter.

Send to laser cutter

If you have a lasercutter readily available, the easier option is to send the jobs you just created directly to the lasercutter. Just insert the sheet of material of the job you currently have selected into the lasercutter and click on ”Start Cutting”. If you need to change the material, you will get a hint reminding you to do so.

After everything is cut, do not quit the program - you can use the 3D-view as an assembly guide, just compare the form of each virtual object with the real one and assemble them correspondingly.

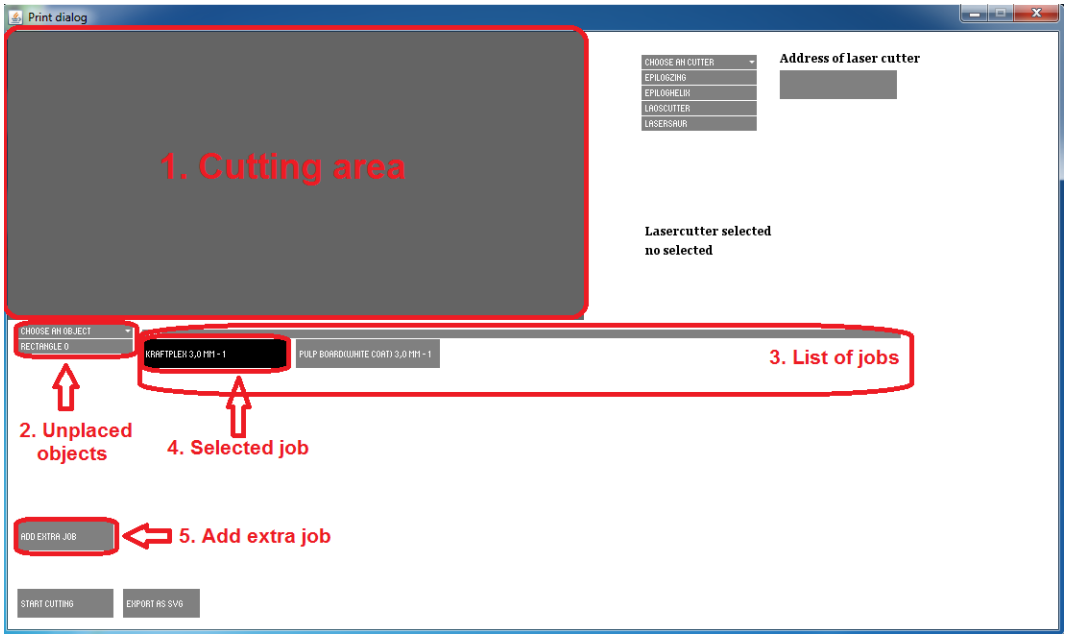


Figure 3.10: The interface of the cut dialog.
1. The cutting area.
2. The unplaced objects of the material corresponding to the selected job.
3. List of all cutting jobs.
4. Laser job selected to be edited.
5. Button to add an extra job.

STL import

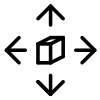


Load STL

The ”Load STL”-Tool can be used to import one STL-file into the 3D-view of your sketch. The idea behind this feature is to use the imported 3D-object as an orientation for dimensions and positions while building whatever it is you want to build: Maybe you want to build an enclosure for your arduino-based electronics project and are not sure where exactly to put Cutouts for the USB-connector or you want to make sure that your enclosure has the correct size? Just import a stl-file containing a three dimensional model of the arduino and place it inside your enclosure to find out. You can change the position and rotation of the the imported STL-model using the ”Change STL”-Tool.

Importing multiple STL Files

We have limited the import of STL-files to one single file. However, it is possible to import multiple three dimensional shapes by composing them together into one STL-file in an external program and loading this one file.



Change STL

This tool is used to relocate a loaded stl-object in the 3D-view. For this purpose, for six different input areas (slider and number field for each of them) exist in the properties bar at the top of the screen. They allow the movement of the object along the three axis, as well as rotating it in each direction.

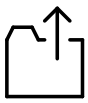
This allows you to place the object corresponding to your design out of 2D-shapes in the 3D-view and control its dimensions. It also allows a relative easy way of position e.g. cut-outs at the right positions.

Save and Load



Save

The Save-Tool allows you to save a sketch you created with this program to load it back up again later on. Saving a sketch will result in all shapes, connections and cutouts you created being written to the file you specified. You can restore your work later by clicking on the Load-Icon.



Load

The Load-Tool allows you to load a file to return to a sketch you saved earlier using the Save-Tool. Simply click on the Load-Icon to open a File-Selection-Dialog. In the File-Selection-Dialog, select the file containing the saved sketch and click on ”Open” to load the sketch.

Chapter 4

Customize it

Some parts of CutCAD depend on information about what tools and materials are available to you (e.g. which lasercutter you will be using and what settings this lasercutter needs to cut a certain type of material). In order to make CutCAD usable for anyone and not just people working with the particular lasercutter-model we have been working with, we have encoded this information in configuration-files outside of CutCAD itself or made sure that the necessary data can be entered inside of CutCAD. This part of the documentation will explain what files will need to be changed to use CutCAD with your own setup.

We will also describe how to add your own drawing tool. If, for example, you need a specific complex shape, which you want to use again and again with some basic parameters, you will learn how to write the code to implement a tool that does this job for you.

4.1 Change/Add Material

If you want to change the settings for the different materials or add a new material, you can change the corresponding xml files or create a new one. The program loads every file in the materials folder and uses the following structure to identify a material:

```
<material>
<identity red="246" green="209" blue="148"
alpha="255">MusterMaterial</identity>
<thickness value="100">
<cut>
<speed>100</speed>
<power>50</power>
<frequency>5000</frequency>
<focus>0.0</focus>
</cut>
</thickness>
<thickness value="200">
...
</thickness>
</material>
```

The outer `<material>...</material>` just determines that it is a material, keep it as it is.

Name

This is followed by the identity `<identity red="246" green="209" blue="148" alpha="255">MaterialName</identity>`, where you can set the color a shape of this material should have in the 2D- and 3D-view. CutCAD expects values between 0 and 255 for each of the colors. The alpha value is used for transparent materials (a material with alpha=”0” is fully transparent, a shape with alpha=”255” is not transparent at all). You can also insert a name for the material which will be shown in the drop-down menu where materials can be selected (replace the MusterMaterial with a more suited string).

Thickness

The next thing you might want to change is the thickness of the material. You can add several different thicknesses for

each material – each with their own settings for the laser-cutter. CutCAD assumes that you entered the thickness in one hundredth of a millimeter (thus, the first thickness of the material in the example above is 1mm, the second is 2mm). Each thickness will get its own entry (together with the name) in the material drop down menu.

Settings for Cutting

You need to specify the lasercutter parameters for cutting the material for each thickness. You can do this by entering the same parameters you would enter into visicut for this material and thickness in the cut-tag (`<cut>...</cut>`).

Visicut Material import

VISICUT MATERIAL IMPORT:
While we could have added an option for the import of visicut settings, we decided against it. The main reason was that we would also need the color and could not be sure that every setting in visicut already defined the thickness of the material (some settings did not specify the thickness). Thus, we decided to write our own, more structured definition. This also simplifies the experience for the user for the rest of the program, since only the person setting up the program the first time will have to deal with the material customization. As soon as the first setup is done, all the settings are hidden from the user.

Engraving

ENGRAVING:
At the moment, our program supports only cutting. For simple marks one might be able to use the same settings but with reduced power. If future versions will allow the user to also engrave shapes, the described structure will just have be to extended with an `<engrave>...</engrave>`block that holds the correct lasercutter settings for engraving each material.

4.2 Change the Lasercutter

Type

Before a job can be sent to a lasercutter using the cut dialog, some settings have to be applied. First of all, a lasercutter type should be chosen. The box highlighted in figure 4.1 shows the supported lasercutters. By clicking on one of them, it is selected as the lasercutter to be used. After a lasercutter has been chosen, the chosen lasercutter will be shown (1. in figure 4.1), and the width and height of the cutting area will be adjusted accordingly to the dimensions of the selected lasercutter type. Note that the settings for the material types should be set accordingly to those of the lasercutter selected. How to do this is described in section 4.1.

Resolution

After a lasercutter type has been chosen, a list of supported DPI settings will be displayed. This box is not displayed, before a lasercutter type has been chosen, as the possible DPI settings depends on the lasercutter type which was selected. Before starting to lasercut, a DPI setting has to be chosen, using the box displayed with 2 in figure 4.2. When a DPI setting has been chosen it will be displayed (3. in figure 4.2).

Address

Lastly an IP address or port should be entered in a corresponding field. This field is denoted by 4. in figure 4.2. For the Epilog series this is usually an IP address. This can differ for other types of lasercutters. The address can be a port number as well.



Figure 4.1: The box where the type of the lasercutter can be selected is red highlighted.

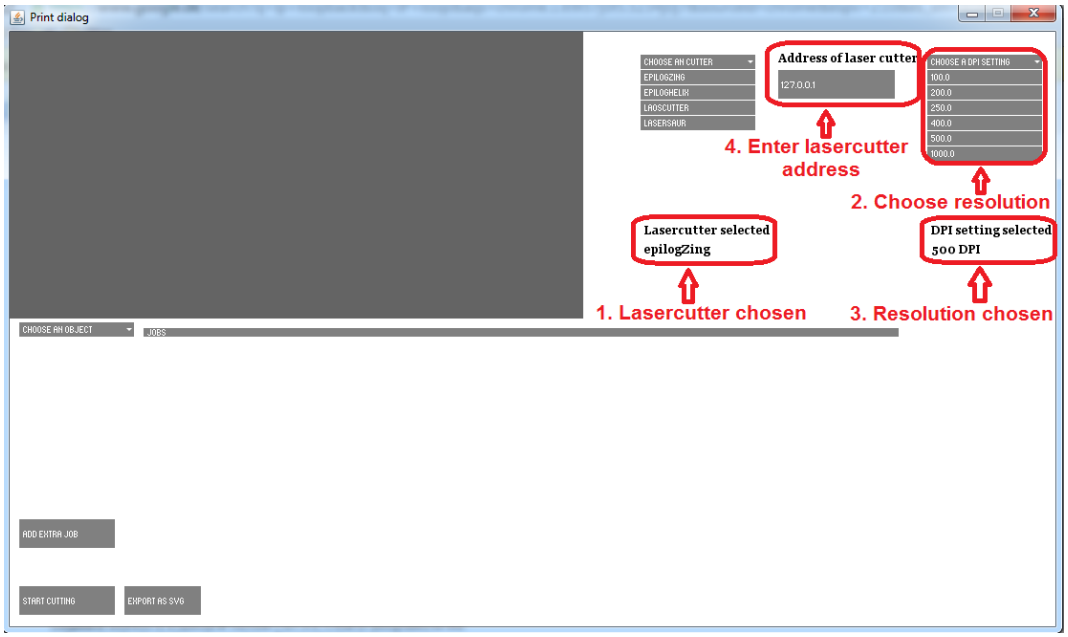


Figure 4.2: The settings which need to be set, after choosing a lasercutter type.
1. Displays the lasercutter type chosen.
2. Box for the selection of one of the possible resolutions.
3. Displays the resolution chosen.
4. Field to enter the address of the lasercutter.

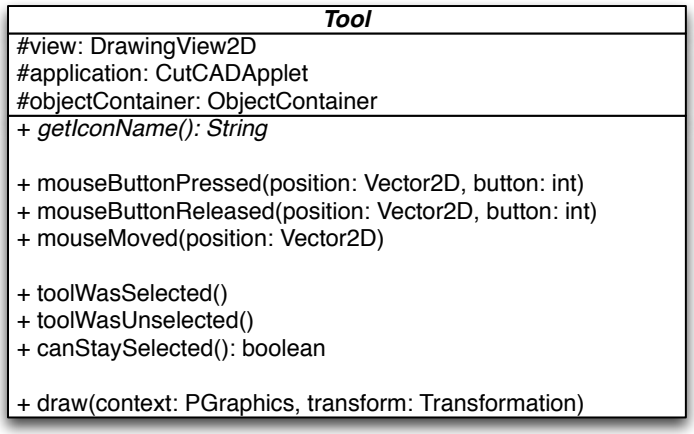


Figure 4.3:

4.3 Building Custom Tools

New tools are always implemented as a subclass of the `de.mcp.cutcad.application.Tool` class. Tools are event driven, this means that you will be informed about the actions of the user through methods that are called by the application. By overriding the method for a certain event, your tool can react to the users actions. Tools are also included in the drawing process of the application, which means that whenever the applications elements are redrawn, the tool also gets notified about the redrawing and has a chance to draw its own content into the applications views.

y now, tools are only able to draw into and interact with the 2D drawing view of the application.

Tool API

The tool API provides methods that must be overridden in order to achieve certain behavior.

Methods that **must** be overridden:

- `getIconName()`: This method must return the name of the SVG file that holds the icon of the tool.When

the toolbar is created, the icon is automatically loaded from this file. The file needs to be placed inside the `icons` folder inside the project folder.

Methods that *can* be overridden:

- `mouseButtonPressed(position: Vector2D, button: int)`: This method will be called whenever the user presses a button. Override this method in order to intercept button clicks and have your tool react to it. The mouse cursors position is provided through the `position` argument, the button that was pressed is given in the `button` argument.

- `mouseButtonReleased(position: Vector2D, button: int)`: This method will be called whenever the user releases a button. Override this method in order to intercept button releases and have your tool react to it. The mouse cursors position is provided through the `position` argument, the button that was released is given in the `button` argument.

- `mouseMoved(position: Vector2D)`: This method will be called whenever the user moves the mouse. Override this method in order to intercept button clicks and have your tool react to it. The mouse cursors position is provided through the `position` argument.

- `toolWasSelected()`: This method will be called when the tool is selected. Override it if your tool needs to do additional set up or in order to show dialogues or the like to the user.

- `toolWasUnselected()`: This method will be called when another tool was selected. Override it if you need to perform additional clean up tasks on your tool.

SUPER!:
The methods `toolWasSelected()` and `toolWasUnselected()` both have default implementations that do important stuff, so please never forget to call the `super.toolWasSelected()` or `super.toolWasUnselected()` methods.

Super!

- `canStaySelected()`: This method defines whether your tool can stay selected (like the Select tool or any draw tool) or if it is just a click-and-forget tool (such as the save tool or load tool). The default return value for this method is `true`. If your tool is a click-and-forget tool, override this method to return `false`.
- `draw(context: PGraphics, transform: Transformation)`: This method allows you to draw into the 2D drawing view in order to visualize actions performed by your tool, such as a preview of the object the user is going to draw with your tool. Keep in mind that this method is called by the rendering runloop and therefore you should not perform expensive calculations here, since this will drastically decrease the applications performance.

Add to Toolbar

Once you finished implementing your tool, you need to add it to the toolbar. In order to do this, open the `de.mcp.cutcad.application.Toolbar` class. There you will find a factory method `createDefaultToolbar(CutCADApplication application)`. In this method you will find an array `Tool[] tools` that already is configured with the default tools of the application. Instantiate an object of your tool class here and it will be added to the application.

Chapter 5

Behind the Scenes

In this section we want to present a basic overview over the algorithms used in CutCAD for those interested in extending or reusing our code. We will try to briefly explain the ideas behind the algorithms, while not going into too much detail. For more detailed information, please have a look at the source code as well as the javadoc-files.

5.1 Align and Rotate

In order to show the user a graphical representation of the assembled final product in the 3D view, the shapes the user created in the 2D-view have to be aligned and rotated such that they fit together in the three dimensional space whenever they are connected. In this section, we will describe our solution to this problem. A connection is always established between two shapes: The master, which will not be moved, and the slave which will be aligned with the master and rotated to match the angle necessary for the particular connection. Furthermore, a connection is made between two edges of those two shapes. We will refer to those edges as the masteredge and the slaveedge. An edge has two defining points on each of its ends.

There are five steps taken in order to align the slave with the master correctly:

1. The slaveedge is aligned to the masteredge
2. The slave is moved to the origin
3. The slave is rotated to align it with the master
4. The slave is rotated around the slaveedge to match the specified angle
5. The slave is moved to the master

In the following, we will explain each step in a bit more detail:

In the first step, the slaveedge is aligned to the masteredge. First, vectors that point from the origin into the same direction as the corresponding edge are calculated for each edge. Then we calculate the angle between those two vectors. The crossproduct of the two vectors is used to find a normal vector for them. This normal vector is then used as the rotationaxis to rotate the slave around by the angle between the two calculated vectors. This will align both edges in the three dimensional space. Note, however, that while the slave and the master are now aligned with respect

to the connected edges, there might still be a difference in rotation around those edges for both shapes. This rotation will be fixed in later steps.

In the second step, the slave is moved to the origin. This step simply makes it easier to rotate the slave around the slaveedge with the library for vector-calculations that we are using (Toxiclibs).

In the third step, the slave is rotated around the slaveedge to match the rotation of the master. To do this, we calculate the normal vector for both shapes by calculating the crossproduct between two edges of each shape. Since the shapes are flat, the crossproduct of any two edges will give us a correct normal vector. After finding the normal vectors, we calculate the angle between the two and rotate the slave around the slaveedge by this angle in order to align the two shapes perfectly.

Now we only need to rotate the slave by necessary angle for this connection (either specified by the user or calculated by our algorithm for finding intersections). This is done just like in the last step: The slave is rotated around the slaveedge by the specified angle.

In the last step, we move the slave from the origin to the master. This is simply done by means of vectoraddition for each vertex of the shape. To find out which vector to add, we look at the coordinates of the endpoint of the masteredge that will be matched to the endpoint of the slaveedge which is currently located at the origin. Adding those coordinates to all vertices of the slave will move the slave over to the master with perfectly aligned master- and slaveedge. Since we already took care of the correct rotation for the slave, the two shapes are now aligned just like they were supposed to be.

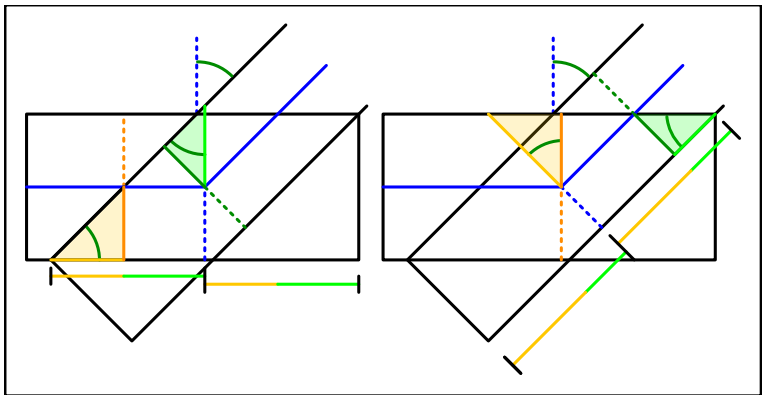


Figure 5.1: Triangle calculations to determine the outline dependent on angle between the shapes and their thicknesses

5.2 Creating the Outlines of Tenons

In order to connect two shapes, they both need a outline of tenons which will interlock as flawlessly as possible. In this section, we will explain how we went about generating such outlines.

In the first step, an edge is divided into several parts, each with the length of a single tenon. To make sure that the connection will look good in the final product, the length of a tenon depends on the thickness of both shapes that are going to be connected.

Every tenon will be extruded or intruded perpendicular to the direction of the edge. How far it is extruded or intruded depends on the angle between the shapes and the thickness of the shapes. Therefore two triangle calculations are made for the intrusion (which has the equal length as the extrusion) of each shape as shown in the picture.

In case such an outline has already been created for an adjacent edge of the shape, only a correction of the corner positions is necessary. To do this, the directly adjacent lines of the other edges outline are taken and enlarged. Now, the intersection between both lines can be computed and

the lines are shortened or enlarged until both end exactly at this intersection point.

5.2.1 Problems and Limitations

There are some scenarios in which this algorithm does not work or can produce problems:

If the angle between two shapes is too small, the intrusion can be very long, which will look weird and will lead to problems in stability. Therefore, the intrusion is limited to a certain length (depending on the thickness of the materials used). Also, the corners might need sandpapering, since the linear correction of corners is not correct for some shapes. Another limitation is the ratio of object size to thickness. If one would try to make a connection between two objects of 5mm thickness and 1mm side length, there is no way to find a correct tenon structure that would allow the user to connect the two shapes. In such cases, the algorithm might give unexpected results.

5.3 Find Intersection for Rotation

If the user wants to connect two shapes that are not connected with any other shapes yet, it is easy to make the connection: We have already explained how the tenon-outlines are created and how the shapes are aligned and rotated in the 3D-view to fit together.

However, if the shapes the user selected to make a connection are already connected to other shapes, there are limitations on how they can be moved and rotated – the previously made connections must not be lost in the process.

If one of the shapes is connected to more than one other shape, it is locked in place. There is no way to rotate this shape individually without breaking one of the previously made connections. But if a shape only has one previously established connection to another shape, it can be rotated around the edge it now shares with the shape it is already connected to. In this case, the goal is to find a rotation around this edge such that the two shapes the user selected can be aligned and connected.

In the simplest case, the two shapes the user wants to connect already share a common end point on an edge due to other, previously made connections. A common example of this is the construction of a simple cube: One would start by connecting the sides of the cube to the bottom. Afterwards, the sides can be connected with their neighbours, in which case they will share an endpoint of their edges that connect to the bottom.

Since both shapes are already connected to another shape, they both can only rotate around one axis (defined by the edge that they are connected to). Each shape is only allowed to rotate around this one axis. It will thus describe a circle of possible positions for the edge. If those circles do not have an intersection, a connection cannot be made. If they however do share an intersection point, the shapes can be rotated in a way such that the two selected edges can be connected.

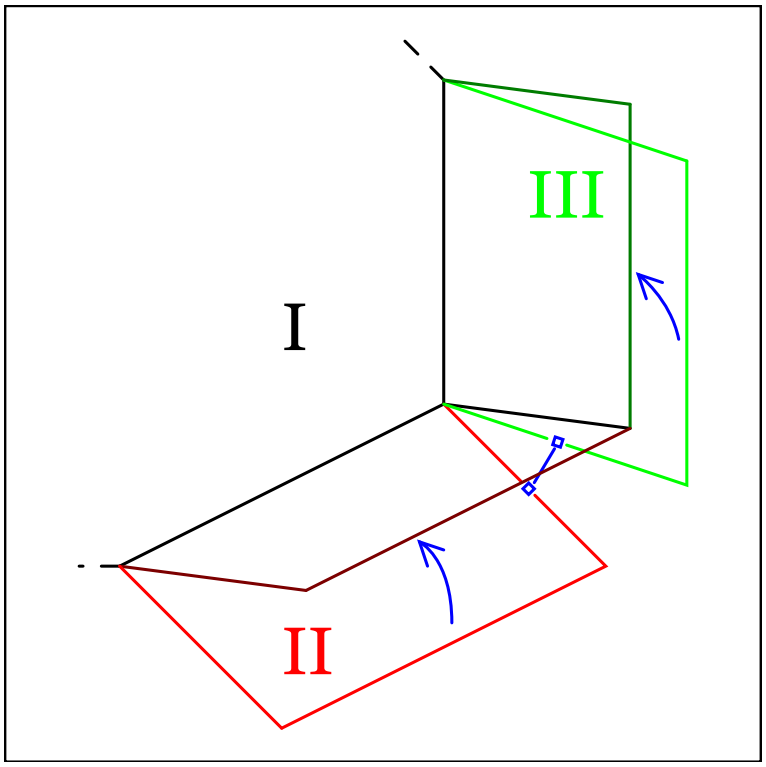


Figure 5.2: Connect two shapes II and III which are already connected with another shape (I)

To simplify, we first calculate the projection of the circles in one plane. First, we align both shapes in this plane. The construction of a virtual shape of the two edges (in this example the edges between (I,II) and (I,III)) allows us to reuse the already implemented algorithm for aligning and rotating.

It is not necessary that both shapes II and III are actually connected to the same common shape. The only constraint is that the previously connected edges and the edges we want to connect now share a common point. A 180 degree rotation of both shapes around the corresponding rotation axis (e.g. the edge between shape I and II for shape II) gives the second endpoint for the projection. This is essentially a mirroring of the edges along the rotating axis in the two dimensional, planar view. Drawing a line between the original and the rotated endpoint gives us the projection of both

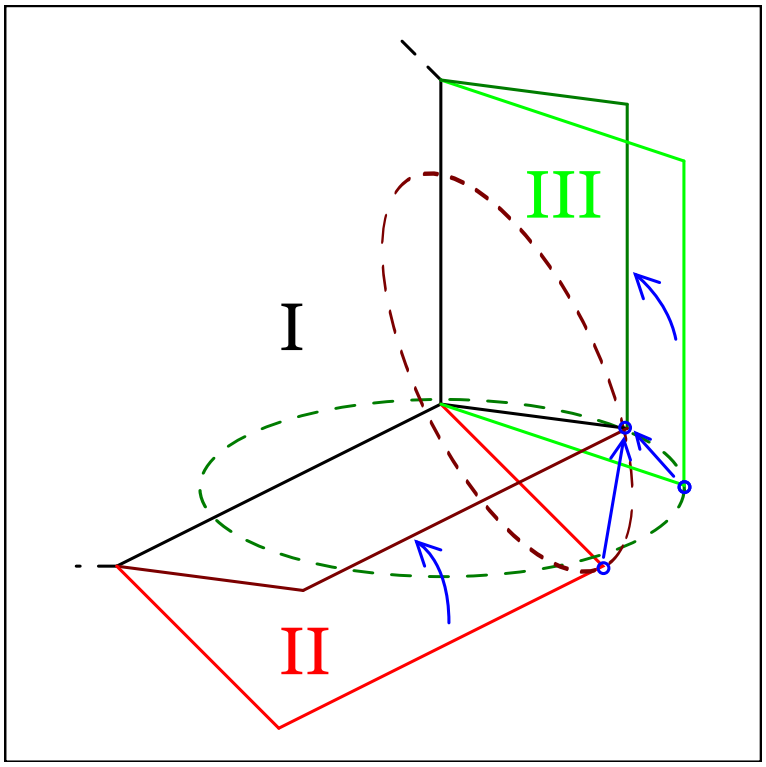


Figure 5.3: The rotation of the endpoints of the edges describes two circles around the already connected edge. We are looking for the intersection of the circles, as that will be the point where both shapes can be aligned

circles in this two dimensional view. Now we just need to calculate the intersection between both projections, which gives us the projection of the intersection point of the two circles on the plane.

In the last step, we calculate the rotations of each of the two shapes necessary to reach the intersection point. The nearest (perpendicular) distance of both the intersection point in the plane and the original end point of the edge can be used to calculate the angle. First, lets note that the latter will stay the same independent from the angle. We also know that the line between the intersection point and its projection has to be perpendicular to the projection plane. Thus, a simple (co-)sine triangle calculation can be used to calculate the necessary angle.

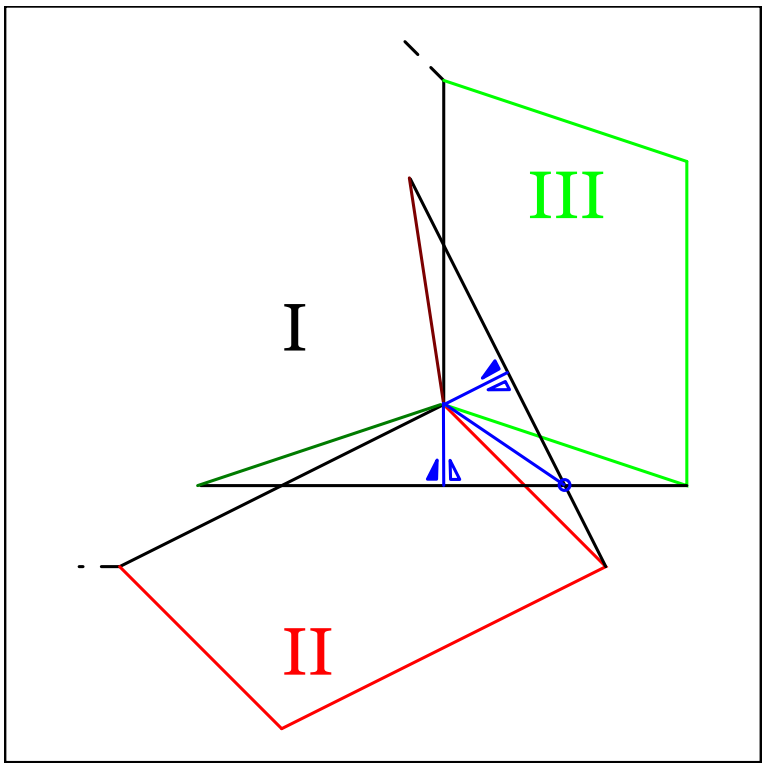


Figure 5.4: Projection of the rotational circles of the endpoints on a plane

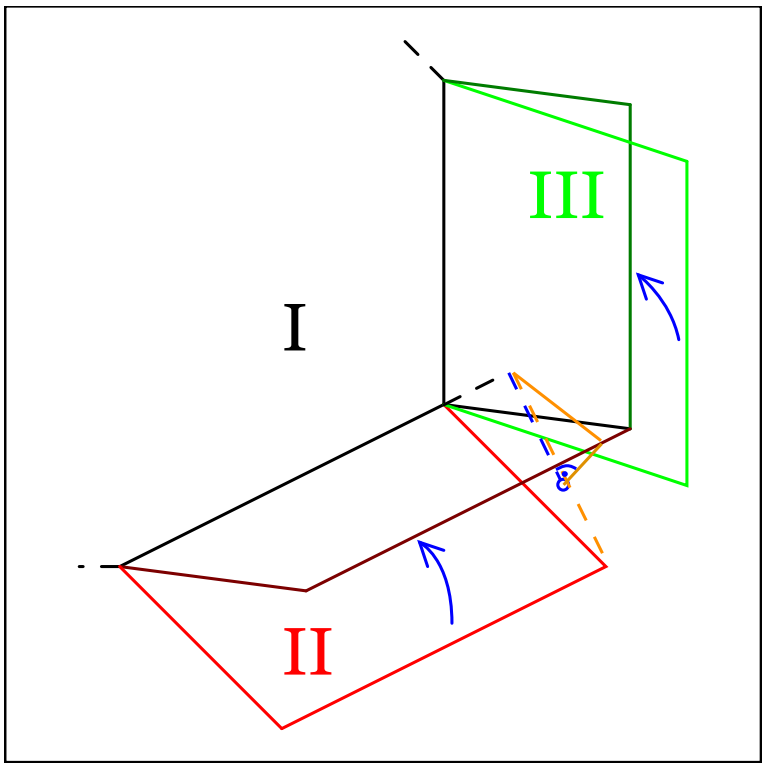


Figure 5.5: Use the perpendicular distance from intersection point and edge end point to the already connected edge to calculate the angle

5.4 Architecture overview

On the next side you will find an overview over the main architecture that is driving the application. The starting point is the `CutCADApplet`, which manages the drawing and event handling for the application and dispatches drawing, mouse and keyboard events to the tools or UI elements. It is also responsible for creating the UI, which consists of the `Properties` bar, the `Statusbar`, the `Toolbar` and the drawing views (`DrawingView2D/3D`).

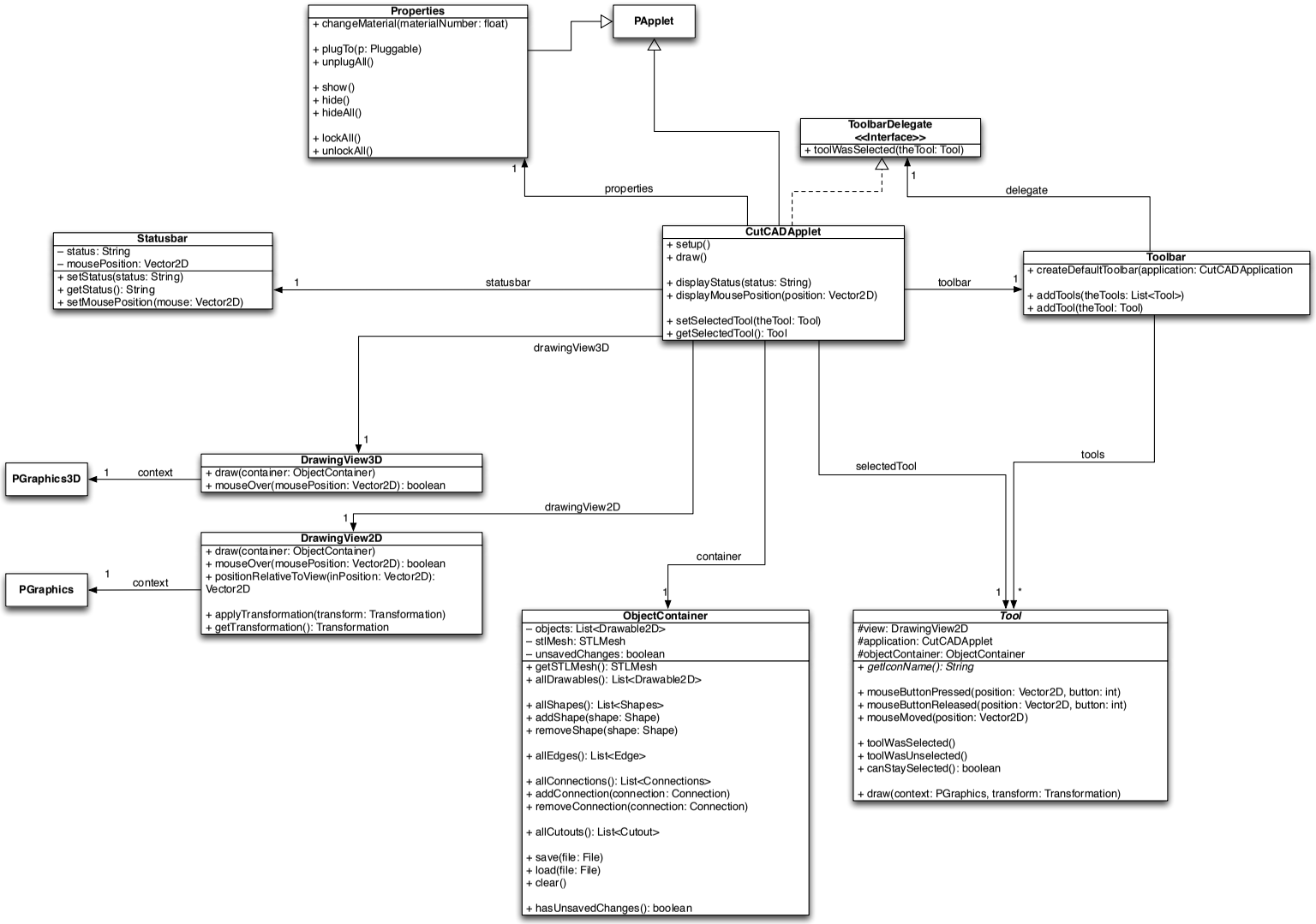


Figure 5.6: Architecture overview

Chapter 6

Future work

While working on CutCAD, we thought of many improvements of existing features or new features that would make CutCAD even better. However, we did not find the time to realize all of them. In this section, we are going to go over possible future improvements of CutCAD and explain the thought process behind them.

Engraving

We already give the user the option to add cutouts to shapes in order to customize the appearence of the finished product. Another step in this direction would be to let the user add engravings to shapes and extend the LaserCut-Dialog with the capability of sending the engravings on an extra-layer to the lasercutter. At the moment, engravings can only be added outside of our application by first exporting the final results into svg-format and then adding the additional engraving-layer in an external program such as inkscape.

Flexible material

To allow the user to create round shapes, the option to use flexible materials that can be bend could be added to the program together with the interactions necessary to let the user bend the shape into the desired form. To extend on this idea, even relatively inflexible materials such as wood could be altered by using diamond-shaped cut-outs to make them more flexible and use them in this way.

GUI Improvements

While we are happy with most parts of the user interface,

the properties bar in particular might not be as intuitive as it could be. A relatively simple improvement would be to include icons instead of descriptions for the parameters. The part of the object that gets altered by the corresponding slider could be highlighted in the icon to give the user a faster way of understanding what the particular control does.

Align-Tool

We had many ideas for other possible tools, but this one stood out the most: An align-tool that lets the user change edges of existing shapes in order to connect them with another edge would certainly be a big improvement in usability, but also brings many complications with it.

Our idea was that the user would first select the edge he wants to connect (just like with the connection-tool) and then select another edge of a different length that he wants to connect to the first selected edge. The second edge would then be changed to be the same size as the first edge by recalculating the shape. Of course, recalculating the shape is not necessarily easy: In certain cases, it can be (e.g. with simple rectangles), but with more complex shapes, it becomes very hard to decide how exactly to change the size of one edge. Depending on the decision which vertex of the shape to move, the operation might have very different outcomes that completely change the form of the shape. It might be possible to first let the user mark parts of the shape that should not be changed and then use such a tool, but so far we are not sure about the specifics of the process.

STL to shapes

After building more complex objects with the prototypes of our software, we realized that it might be possible to use CutCAD to construct three-dimensional shapes from imported STL-files. Instead of rebuilding an object in CutCAD by hand, the user could import an STL-file and the program could first break down the triangle mesh to a level of complexity that is not too high for the software to handle and then replace the triangles of the mesh by triangular shapes, add tenons to them and let the user cut them out.

Lasercutter Table

When constructing more complex forms (i.e. anything that does not use rectangular angles), some manual post-production is needed to create the best possible endresult.

The reason for this lies in the fact that the laser-cutter can only cut vertical. Slanted edges can be simulated by engraving a relief, but this is not a clean solution. Also, the process of engraving such reliefs is very slow.

While thinking about another solution we had the idea of using a support table with movable columns. By extending the columns on one side of the support table and lowering the columns on the other side, a non-rectangular angle between laser beam and table can be produced. This would allow the laser to cut in the necessary angle. Since the laser moves in only one direction, the possible allowed angle for the table would be very limited. However, this could be countered by rotating the object on the table sucht that the table just needs to rotate parallel to the axis on which the laser beam moves.