mged by example

An introduction to brlcad: simple examples in mged

by Paul Cobbaut

(version 0.01, 22 feb 2006)

Table of Contents

1. Chicken Politics

1.1 About Chicken Politics

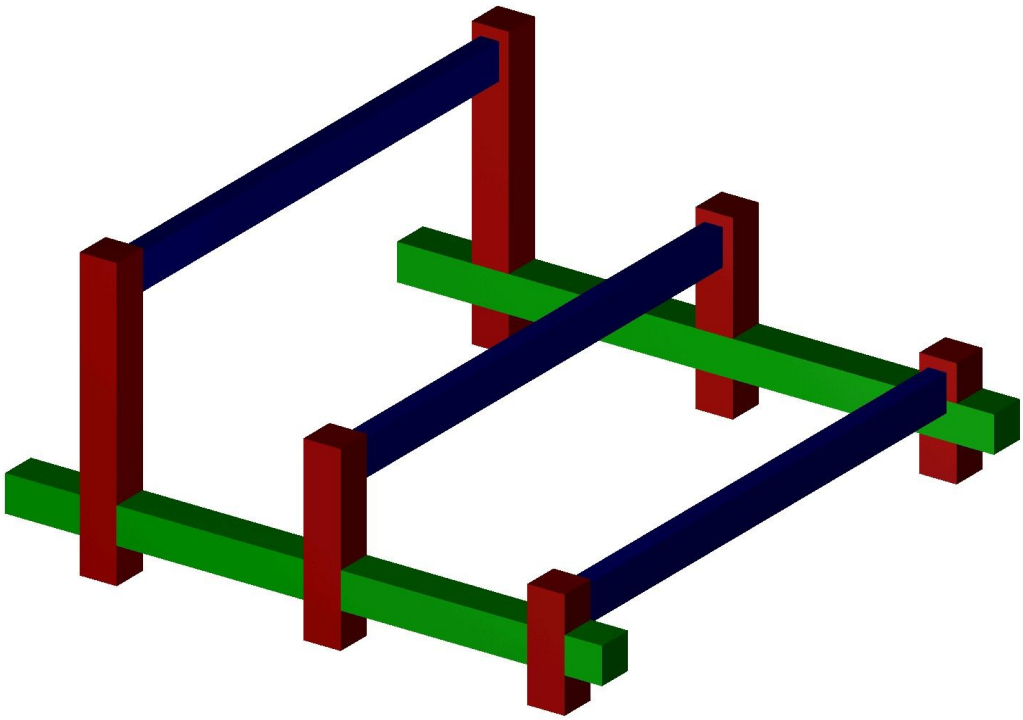
Chickens give us eggs, and for this we thank them! To give something back, lets make'm a nice place to sleep. Keep in mind that chickens have a hierarchy in the henhouse, some chickens are leaders, others are just chickens. Similar to human politics, it is common for the leaders to defecate on the lower beings.

The goal of this wooden project is to give the chickens a place to sleep and perform politics. The leaders can sleep 40cm above the ground, middle management gets 25cm and the lower beings get 15cm.

1.1.1 Technical Specs of this project

This is a simple carpenter training project, constructed entirely out of cheap wood. This example shows how to construct this project from scratch using mged. No knowledge of BRL-CAD or mged commands are needed. But it can be useful to walk trough the first chapters of “Volume II: Introduction to mged” before proceeding with these examples.

To make this project for real, you need about three meters of 50x50mm wooden beams, and about 80cm of a 25x160mm wooden beam.



The green stuff on this picture are called 'support beams', they are connected to the red 'poles' with a classical halfwood connection. The blue stuff on top (where the chickens sleep) are called 'sticks' and are connected to the poles with (wooden) cylindrical pins.

1.1.2 Prerequisites

You must have a functional copy of BRL-CAD installed on your computer. You can find instructions on installing brlcad on this website: http://www.brlcad.org/ .

This example is tested on Debian stable Sarge (Linux 2.6.8), running on an Intel Pentium 4, using brl-cad 7.6.4 (2005-11-15).

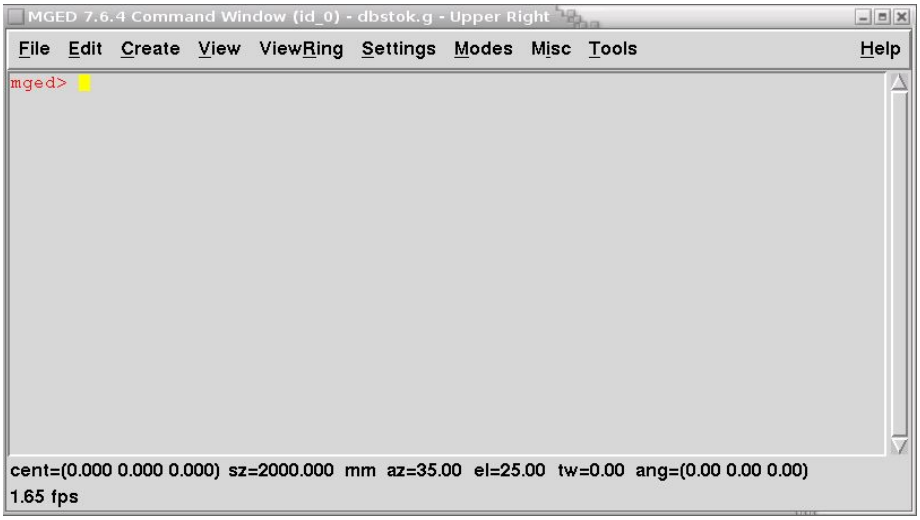
1.1.3 Starting mged

Choose a directory to host this project. On my debian i chose “/home/paul/brlcad/stok/”, but any directory where you have access to should do. Start mged from within this directory with the command “mged dbstok.g”. Besides starting mged this command will also ask you to confirm creation of a database to hold our project, as you can see in this screenshot.

This database will present itself as a file in the directory.

```
paul@barry:~/brlcad/stok$ ls -al dbstok.g
-rw-r--r--  1 paul paul  104 2006-02-20 15:34 dbstok.g
```

When you answer 'Yes' to the “Create New Database” dialog box, two windows should open. Both windows have the same title, but only one of them is big and blue-ish. The smaller window contains “mged> ” before a blinking command prompt, and should look similar to this screenshot.



This project uses the metric system, and all distances use millimeters. We tell this to brlcad with the command “units”.

```
mged> units mm
```

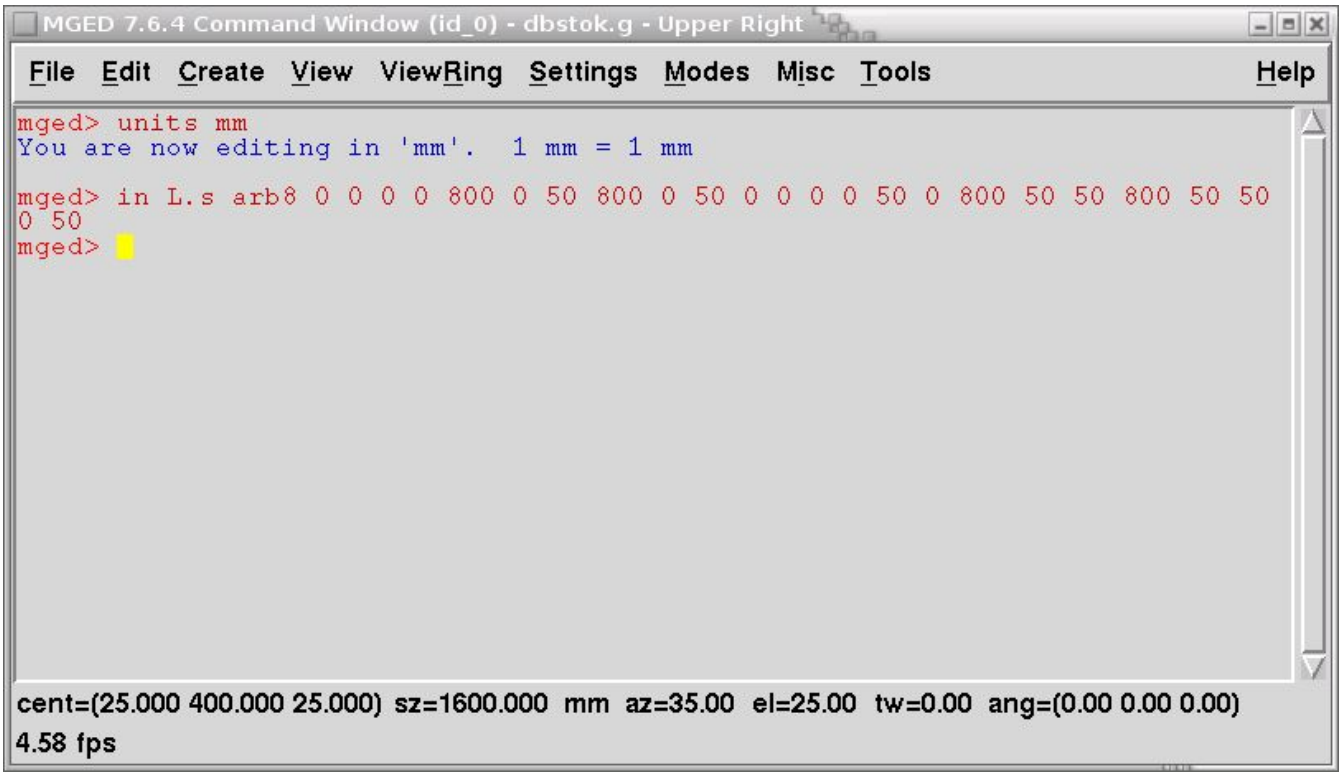
1.2 Creating the first support beam

Let's start by creating a simple (wooden) beam. We do this by typing the following command in mged (right after the “mged>” prompt).

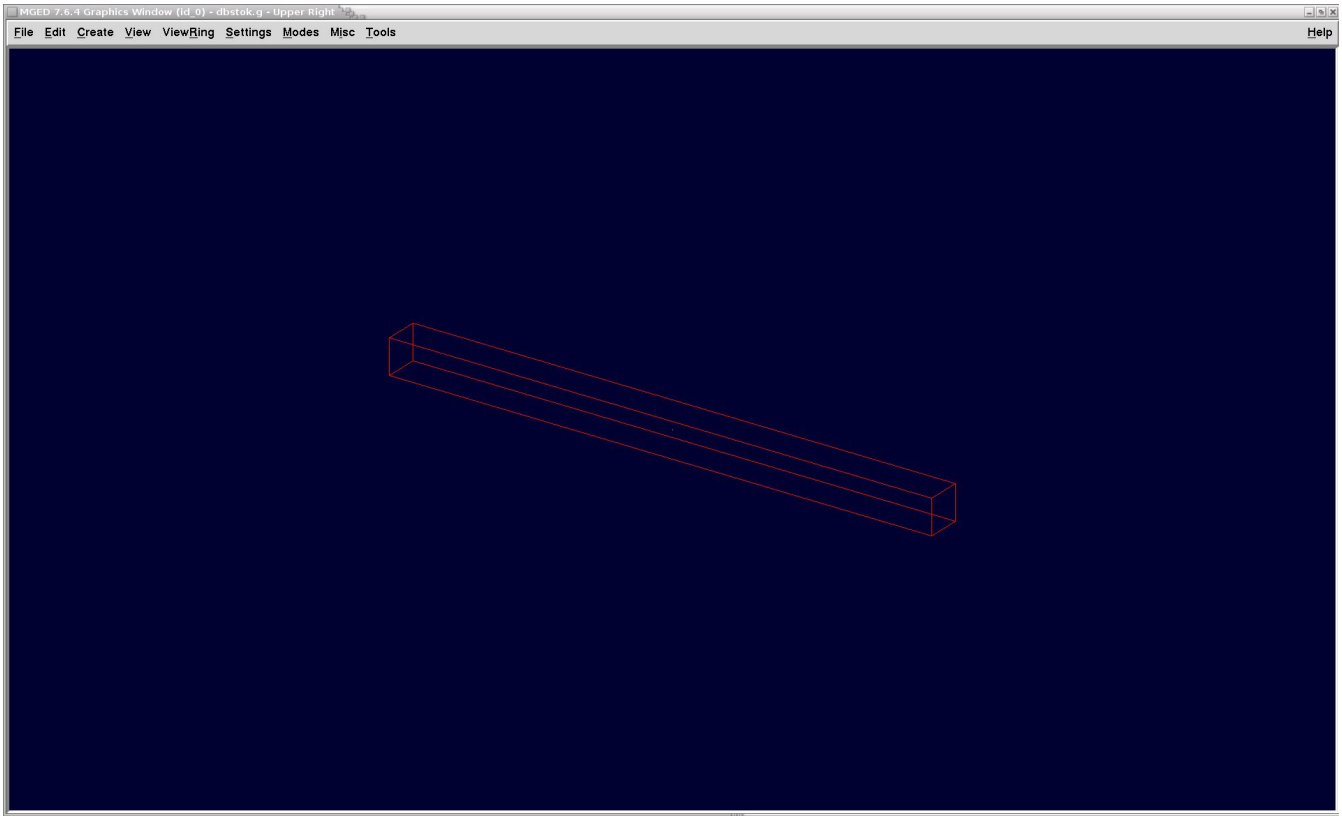
```
mged> in L.s arb8 0 0 0 0 800 0 50 800 0 50 0 0 0 0 50 0 800 50 50 800 50 50 0 50
```

This command creates an object known in brlcad as an arb8. An arb8 is a volume with 8 corners defined by three-dimensional coordinates (X,Y,Z). Our arb8 represents a wooden beam measuring 50 wide(X) by 50 high(Z) and 800 long(Y).

Your mged window should look like this now:



The bigger mged window should look like the next image. When the focus is on this window, you can use the following keys to change the view: f (front), l (left), r (right), b (bottom), t (top) and 3 (same as the ae 35 25 command).



This support beam needs three halfwood holes carved out of it. We create the three holes like this:

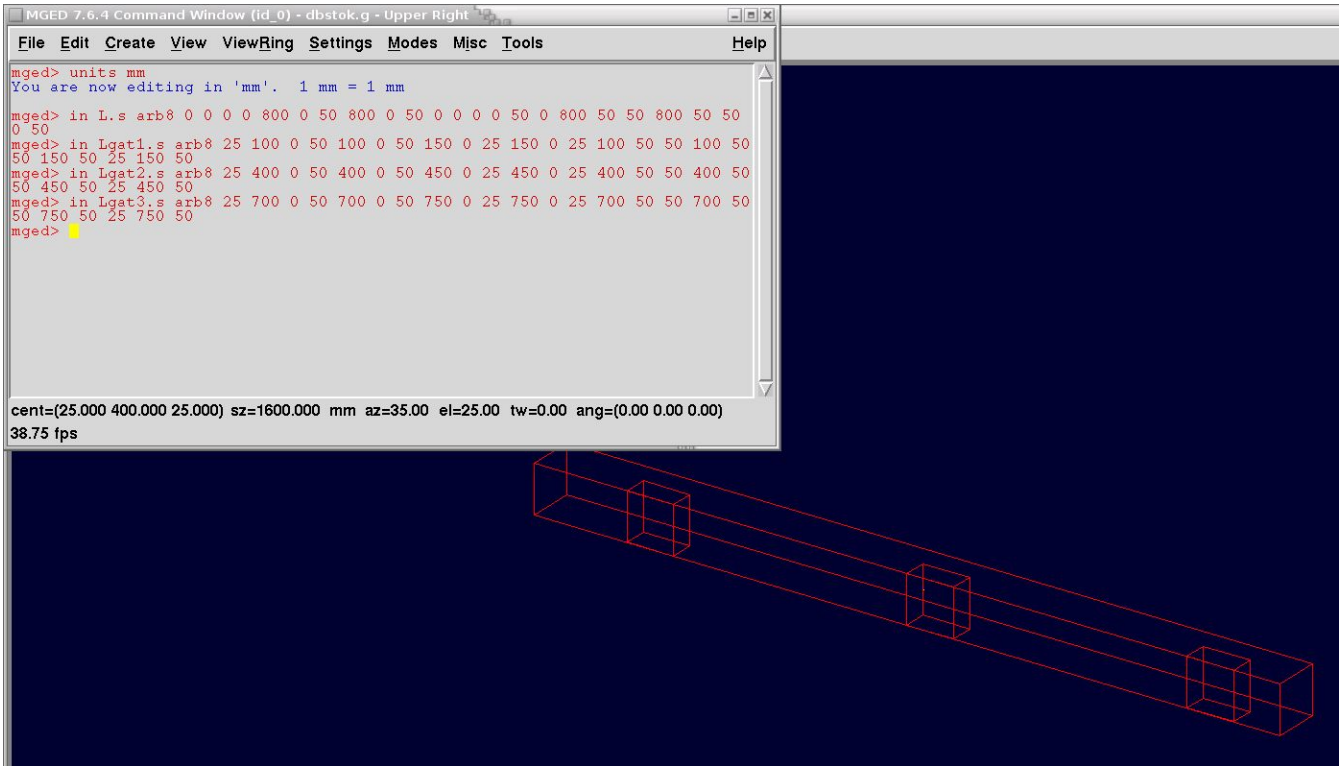
```
mged > in Lgat1.s arb8 25 100 0 50 100 0 50 150 0 25 150 0 25 100 50 50 100 50 50 150 50 25 150 50
mged > in Lgat2.s arb8 25 400 0 50 400 0 50 450 0 25 450 0 25 400 50 50 400 50 50 450 50 25 450 50
mged > in Lgat3.s arb8 25 700 0 50 700 0 50 750 0 25 750 0 25 700 50 50 700 50 50 750 50 25 750 50
```

The X-value varies between 25 and 50, cutting out a piece of the 50mm wood on this side.

The Y-value for the first hole is 100, because this hole start after 100mm. The second hole starts at 400mm, the last one at 700. The hole is 50mm, so the next values for Y are 150, 450 and 750.

The Z-value represents the heigth. All our stuff until now has the same height of 50mm.

Your screen should look similar to this:



You can play with the different views now, you will see our beam, and the three holes that fit perfectly inside the beam. We can create a region of this so we can address our beam-with-holes as one object.

```
mged> r L.r u L.s - Lgat1.s - Lgat2.s – Lgat3.s
```

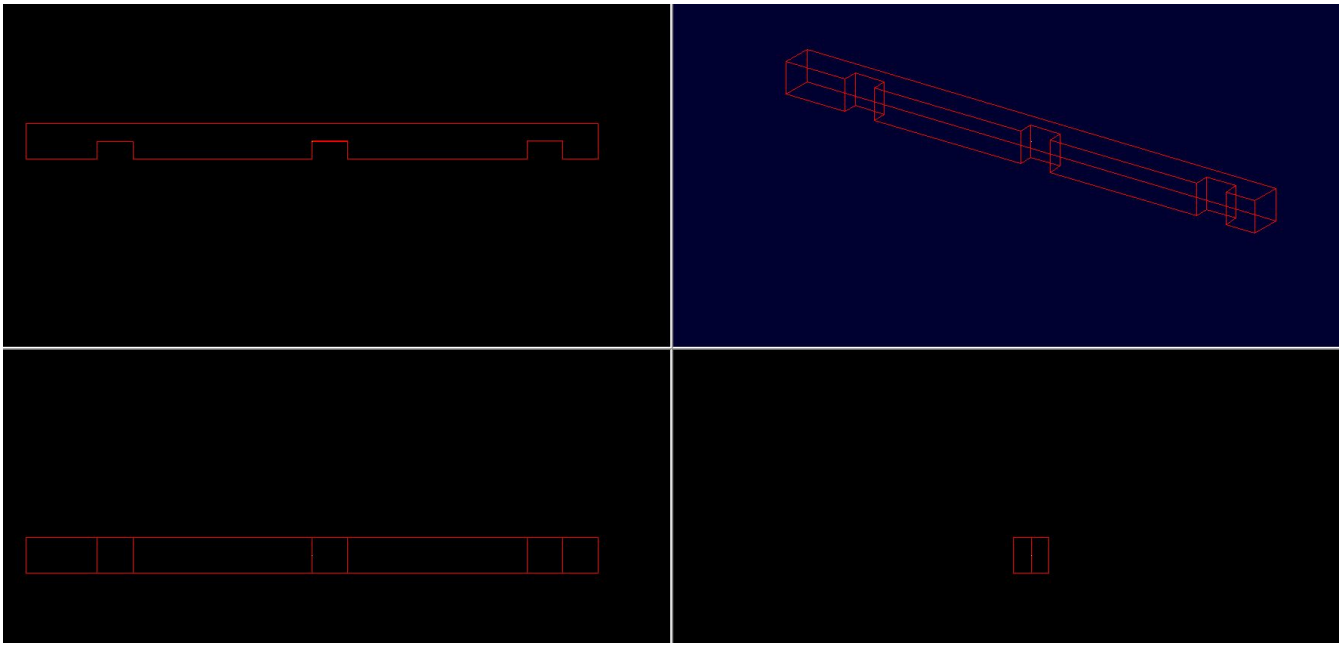
The previous lines defines a region L.r that consists of the union(u) of our beam, with the three holes substracted(-) from it. Now put the region on the screen with the Blast command.

```
mged> B L.r
```

Surprised that you see no difference ? All views still show all lines, even the ones that don't exist. You can solve this problem by creating a bev of your region.

```
mged> bev L.bev L.r
mged> B L.bev
```

Now there is a difference. When looking at the top,left,front etc views, the drawing now is very similar to a technical drawing of our beam. Below is a screenshot of this, with multipane active.



Next lets move our beam in 3D space. To do this, we have to select the beam, which you can do with the mouse...or with the 'sed' command. Moving is done with the 'tra' command, followed by three coordinates. In this example we move the beam 350mm on the X-axis, and 50mm higher.

```
mged> sed L.bev
mged> tra 350 0 50
mged> accept
```

Accept is the opposite of the preceding sed command; it deselects the L.bev object.

Since we don't need the primitive shapes anymore, we can issue the kill command to remove them from our database.

```
mged> kill L.r L.s Lgat1.s Lgat2.s Lgat3.s
```

Congratulations, we now have our beam. We can use the B (from Blast) command to erase the big view and only show our beam.

```
mged> B L.bev
```

1.3 Creating the other support beam

The other support beam is very similar to the first one. We could use the following commands to create it:

```
mged> in R.s arb8 0 0 0 0 800 0 50 800 0 50 0 0 0 0 50 0 800 50 50 800 50 50 0 50
mged> in Rgat1.s arb8 0 100 0 25 100 0 25 150 0 0 150 0 0 100 50 25 100 50 25 150 50 0 150 50
mged> in Rgat2.s arb8 0 400 0 25 400 0 25 450 0 0 450 0 0 400 50 25 400 50 25 450 50 0 450 50
mged> in Rgat3.s arb8 0 700 0 25 700 0 25 750 0 0 750 0 0 700 50 25 700 50 25 750 50 0 750 50
mged> r R.r u R.s - Rgat1.s - Rgat2.s - Rgat3.s
mged> bev R.bev R.r
mged> sed R.bev
mged> tra -400 0 50
mged> accept
mged> kill R.r R.s Rgat1.s Rgat2.s Rgat3.s
```

The holes are positioned on the other side, and the 'tra' moves the beam in the other direction on the X-axis. The rest is the same as before.

There is is shorter way to accomplish this! We can make a copy of L.bev, rotate it 180 degrees along the Y-axis, and then move the rotated beam to its location. Something like this:

```
mged> cp L.bev R.bev
mged> sed R.bev
mged> rot 0 180 0
mged> tra -700 0 50
mged> accept
```

Because of the rotate, you will see that the second beam is 50mm lower, hence the 50 on the Z-axis, to level it with L.bev again.

1.4 Creating the six poles

To name the poles, i use L for the Left side (matching the L.bev support beam) and R for the other side. The heights are 400(G), 250(M) and 150(K), they all measure 50x50mm.

```
mged> in LG.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 400 0 50 400 50 50 400 50 0 400
mged> in RG.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 400 0 50 400 50 50 400 50 0 400
mged> in LM.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 250 0 50 250 50 50 250 50 0 250
mged> in RM.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 250 0 50 250 50 50 250 50 0 250
mged> in LK.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 150 0 50 150 50 50 150 50 0 150
mged> in RK.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 150 0 50 150 50 50 150 50 0 150
```

The halfwood holes:

```
mged> in Lgat.s arb8 0 0 50 25 0 50 25 50 50 0 50 50 0 0 100 25 0 100 25 50 100 0 50 100
mged> in Rgat.s arb8 25 0 50 50 0 50 50 50 50 25 50 50 25 0 100 50 0 100 50 50 100 25 50 100
```

Until now we have been using 'arb8' for everything, so it's time for a new object, the cylinder (rcc). We want to create cylindrical holes in the poles, later they will hold cylindrical pins. The first three numbers for an rcc are its position in X,Y,Z coordinates. The next three define the orientation. The last one is the radius. For our project we want all cylinders aligned with the X-axis, with a length of 30mm and a radius of 10mm. We create these holes 40mm below the top of each pole (400-40=360, 250-40=210 and 150-40=110). This gives:

```
mged> in penK.s rcc 0 25 110 30 0 0 10
mged> in penM.s rcc 0 25 210 30 0 0 10
mged> in penG.s rcc 0 25 360 30 0 0 10
```

Same as before, we create regions where the holes are substracted from the poles, and make a bev of each region. We start with the left side poles:

```
mged> r LK.r u LK.s - Lgat.s - penK.s
mged> r LM.r u LM.s - Lgat.s - penM.s
mged> r LG.r u LG.s - Lgat.s - penG.s
mged> bev LG.bev LG.r
mged> bev LM.bev LM.r
mged> bev LK.bev LK.r
```

Let's move all the pins 20mm on the X-axis, so they are correctly positioned for the right side poles:

```
mged> sed penK.s
mged> tra 20 0 0
mged> accept
mged> sed penM.s
mged> tra 20 0 0
mged> accept
mged> sed penG.s
mged> tra 20 0 0
mged> accept
```

And create regions where the holes are substracted from the right-side poles, and make a bev of each region:

```
mged> r RK.r u RK.s - Rgat.s - penK.s
mged> r RM.r u RM.s - Rgat.s - penM.s
mged> r RG.r u RG.s - Rgat.s - penG.s
```

```
mged> bev RG.bev RG.r
mged> bev RM.bev RM.r
mged> bev RK.bev RK.r
```

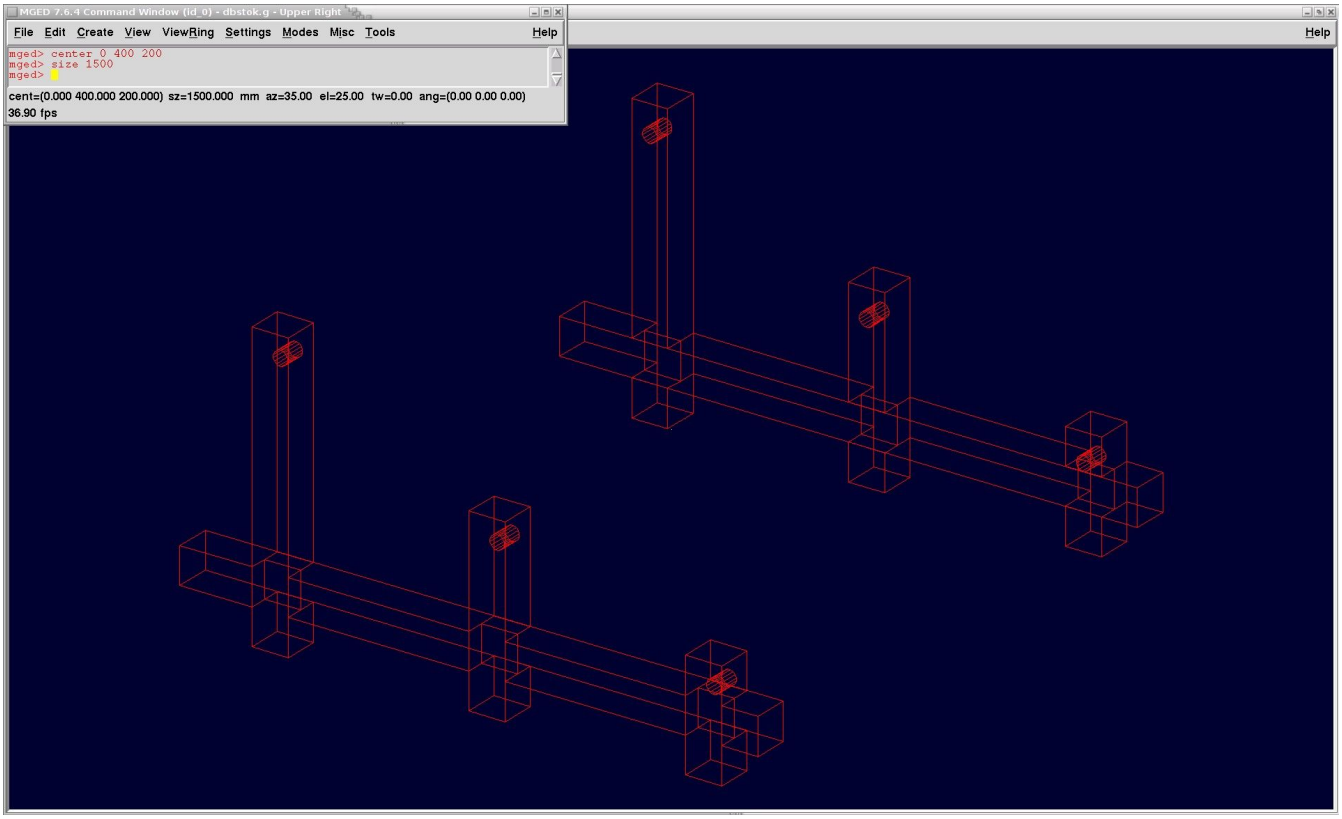
All poles created, now let's move all six of them to their correct position:

```
mged> sed LG.bev
mged> tra 350 100 0
mged> accept
mged> sed LM.bev
mged> tra 350 400 0
mged> accept
mged> sed LK.bev
mged> tra 350 700 0
mged> accept
mged> sed RG.bev
mged> tra -400 100 0
mged> accept
mged> sed RM.bev
mged> tra -400 400 0
mged> accept
mged> sed RK.bev
mged> tra -400 700 0
mged> accept
```

Using the ls command you can now see all the objects that exist in our database. If the clutter is bothering you, you could remove unnecessary objects:

```
mged> kill LG.s RG.s LM.s RM.s LK.s RK.s Lgat.s Rgat.s penK.s penM.s penG.s
mged> kill LG.r RG.r LM.r RM.r LK.r RK.r
```

The result should be similar to this screenshot:



1.5 Creating the three top sticks

There is no new theory here. We start by creating one stick with its two cylindrical pins:

```
mged> in stok.s arb8 0 0 0 700 0 0 700 25 0 0 25 0 0 0 50 700 0 50 700 25 50 0 25 50
mged> in stokpen.s rcc 700 12 25 30 0 0 10
mged> in stokpen2.s rcc -30 12 25 30 0 0 10

mged> r stok.r u stok.s u stokpen.s u stokpen2.s
mged> bev stokG.bev stok.r
```

Then we make two identical copies:

```
mged> cp stokG.bev stokM.bev
mged> cp stokG.bev stokK.bev
```

And we move them all to their correct position. We created the pins at a position of 25mm with a radius of 10mm, so we should be putting the sticks at 12.5mm. Instead of 12 and a halve, i put them at 12 here; 100+12, 400+12 and 700+12. Same for the height, counting the stick's height and the difference, we substract 65 from every height.

```
mged> sed stokG.bev
mged> tra -350 112 335
mged> accept
mged> sed stokM.bev
mged> tra -350 412 185
mged> accept
mged> sed stokK.bev
mged> tra -350 712 85
mged> accept
mged> kill stok.s stokpen.s stokpen2.s stok.r
```

Congratulations, all the parts are now in our database.

1.6 Coloring the view

Let's use some colors to make the drawing a little nicer. To do this, first create a 'comb', then apply a 'mater' (material) to it. I use the default(plastic) and very simple RGB colors.

Make the bottom support beams green:

```
mged> comb bodem.c u L.bev u R.bev
mged> mater bodem.c 0 0 255 0 0
```

Make the six poles red:

```
mged> comb palen.c u LG.bev u RG.bev u LM.bev u RM.bev u LK.bev u RK.bev
mged> mater palen.c 0 255 0 0 0
```

Make the three sticks blue:

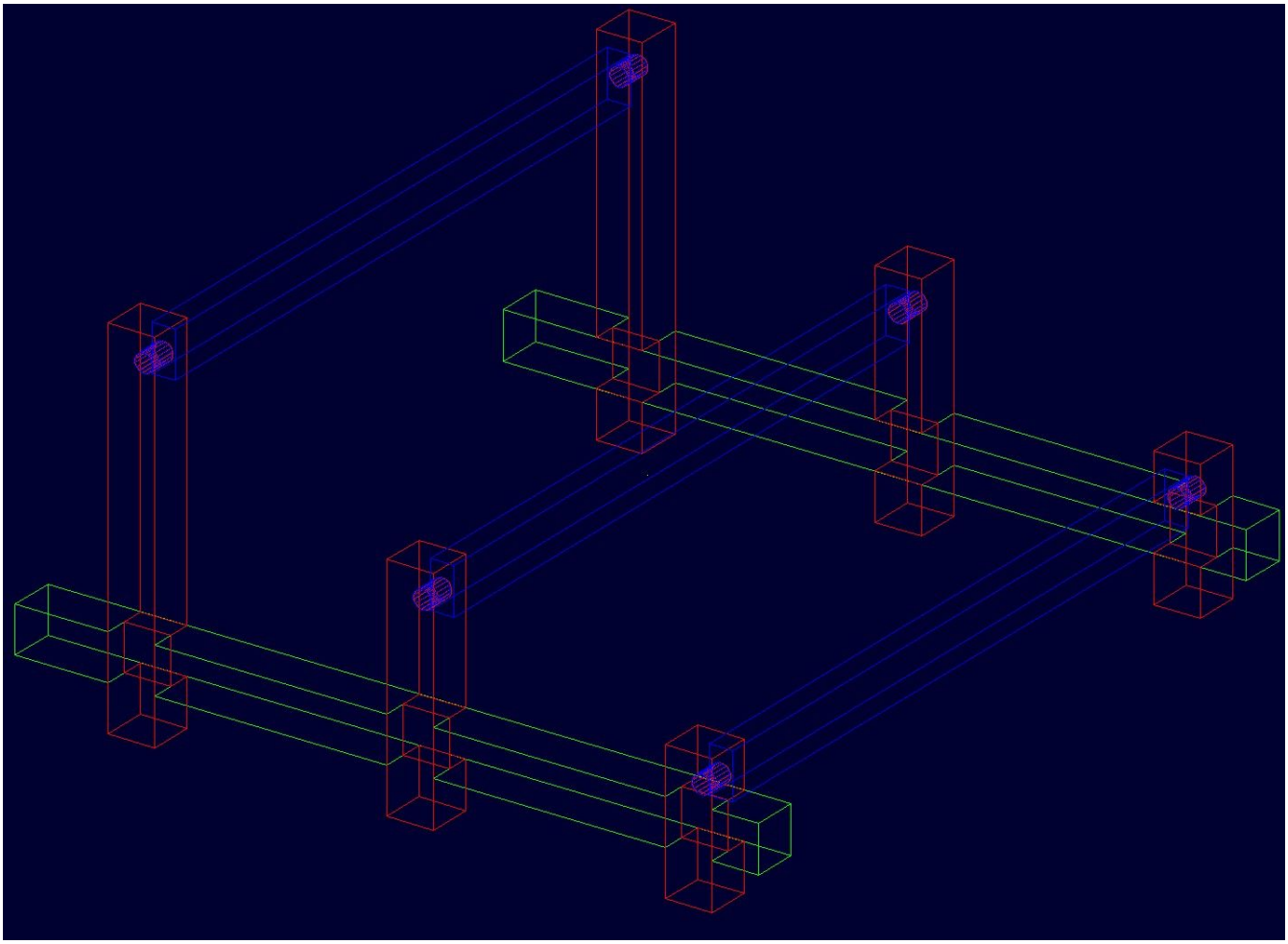
```
mged> comb stokken.c u stokG.bev u stokM.bev u stokK.bev
mged> mater stokken.c 0 0 0 255 0
```

Show the colored stuff:

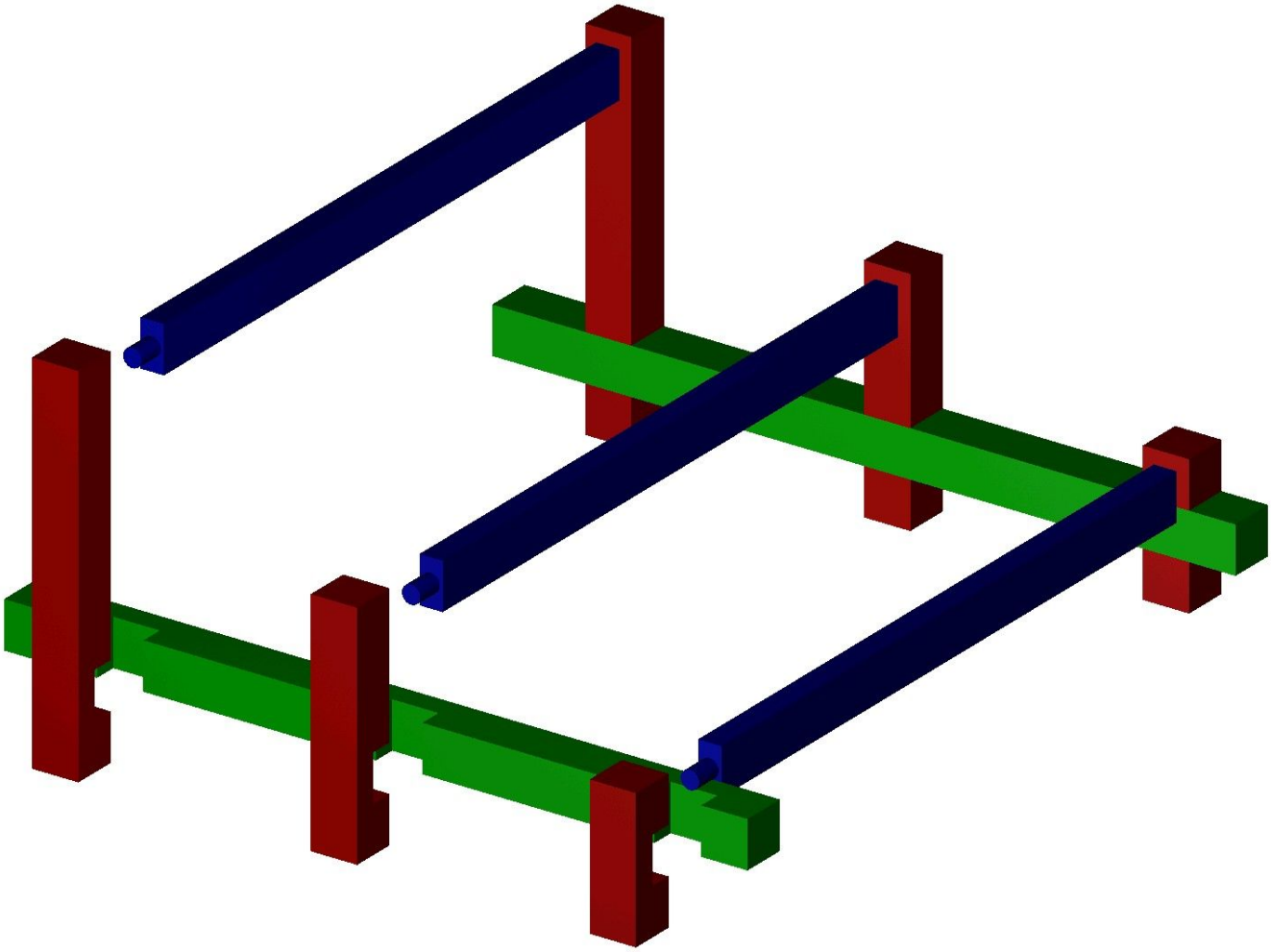
```
mged> B bodem.c palen.c stokken.c
```

Make sure we can see everything on the screen:

```
mged> size 1500
mged> center 0 400 180
```



Now you can raytrace this with a white background to obtain the picture from the beginning of this example. Or you could used the 'sed' and 'tra' commands to move the left poles 100mm away from the center. Raytracing this result will show you the halfwood connections and pins:



2. Complete Kitchen

todo ;-)

Appendix A

MGED script to create Chicken Politics

```
# Below is the complete script to create this project.
# You can download this for free from http://cobbaut.be/stok.brl.txt
# Four steps to run it:
# step One: Enter the directory where dbstok.g is located
# step Two: Save this script as stok.brl
# step Three: Start mged with this command “mged dbstok.g”
# step Four: Inside mged type “source stok.brl”

# Title the database
title Kippenstok, cursus hout 2005/2006

# We use metric millimeters
units mm

# Erase leftovers from the previous run.
# Kill * would be nice, but doesn't work :(

# remove both the support beams
kill L.bev R.bev
kill LG.bev RG.bev LM.bev RM.bev LK.bev RK.bev
kill stokG.bev stokM.bev stokK.bev
# End this killing! Now we start constructing...something.

#####
# Creating the bottomleft support beam
# We start with a full beam(arb8), measuring 50 wide(X) by 50 high(Z) and 800 long(Y)
in L.s arb8 0 0 0 0 800 0 50 800 0 50 0 0 0 0 50 0 800 50 50 800 50 50 0 50
# We create three little parts (the halfwood holes in the beam)
in Lgat1.s arb8 25 100 0 50 100 0 50 150 0 25 150 0 25 100 50 50 100 50 50 150 50 25 150 50
in Lgat2.s arb8 25 400 0 50 400 0 50 450 0 25 450 0 25 400 50 50 400 50 50 450 50 25 450 50
in Lgat3.s arb8 25 700 0 50 700 0 50 750 0 25 750 0 25 700 50 50 700 50 50 750 50 25 750 50
# We cut the three holes out of the full beam
r L.r u L.s - Lgat1.s - Lgat2.s - Lgat3.s
# We create a bev of this, so front/left/top-view look more like a technical drawing
bev L.bev L.r
# We select our beam...
sed L.bev
# ...and move it to the desired location
tra 350 0 50
# and deselect
accept
# We don't need these shapes anymore, so we remove them.
kill L.r L.s Lgat1.s Lgat2.s Lgat3.s
# We now have a support beam (L.bev)
#####

#####
# Create the bottomright support
in R.s arb8 0 0 0 0 800 0 50 800 0 50 0 0 0 0 50 0 800 50 50 800 50 50 0 50
in Rgat1.s arb8 0 100 0 25 100 0 25 150 0 0 150 0 0 100 50 25 100 50 25 150 50 0 150 50
in Rgat2.s arb8 0 400 0 25 400 0 25 450 0 0 450 0 0 400 50 25 400 50 25 450 50 0 450 50
in Rgat3.s arb8 0 700 0 25 700 0 25 750 0 0 750 0 0 700 50 25 700 50 25 750 50 0 750 50
r R.r u R.s - Rgat1.s - Rgat2.s - Rgat3.s
bev R.bev R.r
sed R.bev
tra -400 0 50
accept
kill R.r R.s Rgat1.s Rgat2.s Rgat3.s
# We now have the second support beam (R.bev)
#####

#####
# Create all the poles (L=left,R=right,G=big,M=middle,K=small)
# They all measure 50(X) by 50(Z)
# Their height is 400(LG & RG), 250(LM and RM) or 150(LK & RK)
in LG.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 400 0 50 400 50 50 400 50 0 400
```

```
in RG.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 400 0 50 400 50 50 400 50 0 400
in LM.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 250 0 50 250 50 50 250 50 0 250
in RM.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 250 0 50 250 50 50 250 50 0 250
in LK.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 150 0 50 150 50 50 150 50 0 150
in RK.s arb8 0 0 0 0 50 0 50 50 0 50 0 0 0 0 150 0 50 150 50 50 150 50 0 150

# halfwood holes (same position for all poles)
in Lgat.s arb8 0 0 50 25 0 50 25 50 50 0 50 50 0 0 100 25 0 100 25 50 100 0 50 100
in Rgat.s arb8 25 0 50 50 0 50 50 50 50 25 50 50 25 0 100 50 0 100 50 50 100 25 50 100

# pins (position relative to the height of the poles)
# 400-40=360 250-40=210 150-40=110
in penK.s rcc 0 25 110 30 0 0 10
in penM.s rcc 0 25 210 30 0 0 10
in penG.s rcc 0 25 360 30 0 0 10

# substract the halfwood hole and the correct pin from each pole
r LK.r u LK.s - Lgat.s - penK.s
r LM.r u LM.s - Lgat.s - penM.s
r LG.r u LG.s - Lgat.s - penG.s
bev LG.bev LG.r
bev LM.bev LM.r
bev LK.bev LK.r

# now we move the three pins so we can use 'm again
sed penK.s
tra 20 0 0
accept

sed penM.s
tra 20 0 0
accept

sed penG.s
tra 20 0 0
accept

# substract the halfwood hole and the correct pin from each pole
r RK.r u RK.s - Rgat.s - penK.s
r RM.r u RM.s - Rgat.s - penM.s
r RG.r u RG.s - Rgat.s - penG.s
bev RG.bev RG.r
bev RM.bev RM.r
bev RK.bev RK.r

# move the six poles to their correct position
sed LG.bev
tra 350 100 0
accept
sed LM.bev
tra 350 400 0
accept
sed LK.bev
tra 350 700 0
accept
sed RG.bev
tra -400 100 0
accept
sed RM.bev
tra -400 400 0
accept
sed RK.bev
tra -400 700 0
accept

kill LG.s RG.s LM.s RM.s LK.s RK.s Lgat.s Rgat.s penK.s penM.s penG.s
kill LG.r RG.r LM.r RM.r LK.r RK.r
# We now have six poles (LG.bev LM.bev LK.bev RG.bev RM.bev RK.bev)
#####

#####
# create the three sticks
in stok.s arb8 0 0 0 700 0 0 700 25 0 0 25 0 0 0 50 700 0 50 700 25 50 0 25 50
in stokpen.s rcc 700 12 25 30 0 0 10
```

```
in stokpen2.s rcc -30 12 25 30 0 0 10

r stok.r u stok.s u stokpen.s u stokpen2.s
bev stokG.bev stok.r

# copy stokG to stokM and stokK
cp stokG.bev stokM.bev
cp stokG.bev stokK.bev

# move the sticks to their location
# X is the same for all three
# Y is 100 400 and 700, each + 12 (half of 25 ;-)
# Z is 400, 250 and 150, each - 65
sed stokG.bev
tra -350 112 335
accept

sed stokM.bev
tra -350 412 185
accept

sed stokK.bev
tra -350 712 85
accept

kill stok.s stokpen.s stokpen2.s stok.r
# We now have three sticks
#####

# make the bottom support beams green
comb bodem.c u L.bev u R.bev
mater bodem.c 0 0 255 0 0

# make the six poles red
comb palen.c u LG.bev u RG.bev u LM.bev u RM.bev u LK.bev u RK.bev
mater palen.c 0 255 0 0 0

# make the three sticks blue
comb stokken.c u stokG.bev u stokM.bev u stokK.bev
mater stokken.c 0 0 0 255 0

# show the colored stuff
B bodem.c palen.c stokken.c

# make sure we can see everything on the screen
size 1500
center 0 400 180

# when this script runs, you can use the following keys to change the view:
# b f r l t 3 4
```