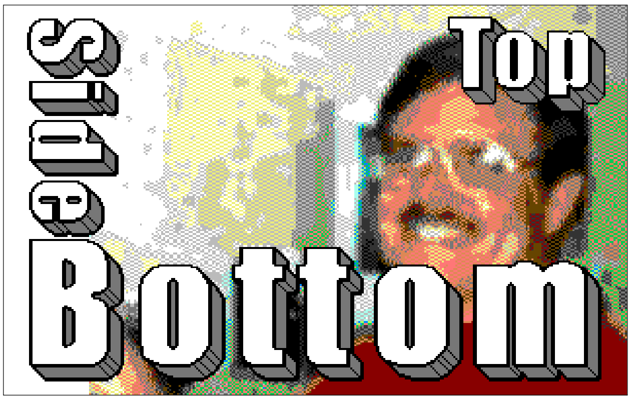
How to GoDot #7

By Arndt Dettke

Lettering Images (part 2)

What have we achieved so far? We have made text shine through an image in the last issue with option “*Mix*” of **ldr.4BitGoDot**. We also applied 3D-looking captions to our graphics, again by use of this versatile function “Mix”. Finally we learned how to “bump” letters onto our images, thus letting them look like they were coined into them.

This issue will cover how to extrude objects, e.g. letters, and let them truly look like boxes or ingots. On the cover of this pamphlet you have an example for what I mean. For this task we need again ldr.4BitGoDot’s option “Mix”, and some modifiers to complete the set of tools: **mod.-.Histogram**, **mod.Scroll**, **mod.QuickMask**, **mod.-DrawMask**, **ldr.4Bit&Mask**, and finally **mod.PixelEdit**. Nothing new.

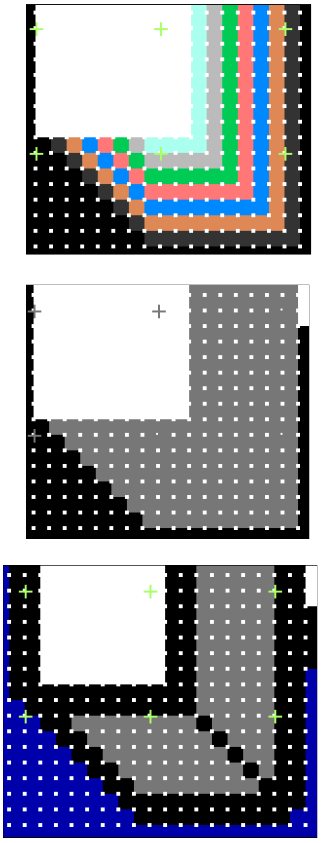


Pic #1: Sorry, Dale, I **had** to do this! ;-) These are letters extruded by 45 degrees to southeast.

To create Pic #1 we just have to repeatedly apply the same method as in the Klimt example last issue. Only difference is that Mix’s application mode is set to “*FGr*” here to avoid rimming on top-left of the letters. So, we load the image with the writing in it (“sidebottom.4bt”), displace it by 1 pixel northwest with **mod.Scroll**, reload the image in *Compose Mix FGr 25%* mode of **ldr.4Bit-GoDot**, and do this seven times all in all. After that, all colors which are not black (the background) or white (the letters) become joined into the color the extrusion should have (mid gray in this case).

In the next step we’re outlining the letters like we did in the October 2003 issue (swapping black to blue in **mod.-.Histogram** and then applying **mod.Cartoon**). And finally we have to fine-tune the image by painting the little diagonal lines at the edges and curves with **mod.Pixel-Edit**. Don’t forget to shift the writing back to a balanced position (six pixels down and four right).

At last we combine the writing with image “dale2000.4bt” by using **mod.QuickMask** (where we set blue to transparent) and **ldr.4Bit&Mask** (see UCUGA’s November 2003 issue). Finished.

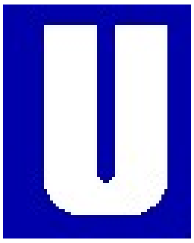


Pic #2: Metamorphosis, 7 times mixed (top), joined (mid), and cartooned (bottom).

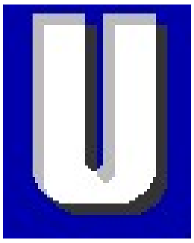
You can of course extrude to every direction you like. You just displace the image to the counter direction and finally shift it back to where it came from. Also, other degree values for the extrusion than 45 are possible when you displace vertically with a different value than horizontally or vice versa (e.g. two up, one left or so).

How to produce ingot-like extrusions? Well, basically it’s the same method as reported above but only with option Mix set to “*all*”. This leads to rims on either side, in the direction you move the extrusion and also in the counter direction (as seen in the Klimt image in UCUGA’s December 2003 issue). Unfortunately, this method lets the area of white in the letters shrink, so it isn’t too much applicable for small letters. Also, fonts which are mainly rounded (not many edges in them) aren’t really favorable (you can’t really apply the ingot “bar” look to them). And third, you need to apply two shades of gray to have it look realistic. This all forces us to spend a little more effort in the result. But ok, let’s simply go for an example.

Let us confine ourselves to only one letter. I suggest the uppercase “U” as it has both edges and curves (Pic #3). What we want to apply to it is a thick rim (of 4 pixels) all around the letter. After loading the original “U”, we displace it by 4 pixels northwest (**mod.Scroll**) and compose it onto itself using option *Mix*, with *25%*, set to *“all”* (Pic #4). This setting of Mix applies also to the next operation. We move the image down four pixels and then again mix it with itself (Pic #5). Don’t forget to move the image right four pixels to get it back where it first came from.

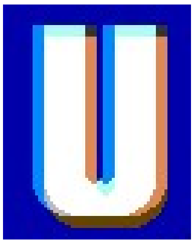


Pic #3: An uppercase U



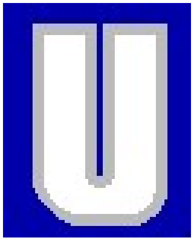
Pic #4: After the first step

Now we have surrounded the U with a four pixel banding of different colors. All these colors will be turned to only one single color using **mod..Histogram** (Option *“Join”*). I took light gray as the *target* color (Pic #6). The curve of the U had to be corrected a bit (mod.PixelEdit).



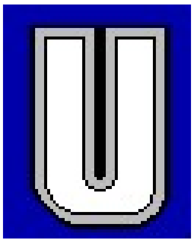
Pic #5: After second displacement

In the next step we apply the black borders to the color areas using **mod.Cartoon** (Pic #7).



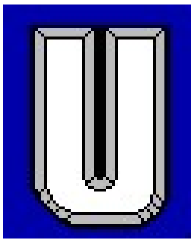
Pic #6: The rim turned to only one color

To achieve a more realistic 3D view we now have to enter those little black lines with **mod.PixelEdit** (Pic #8). Now it already looks like an ingot. A really perfect result would show a darker shadow left or right. For this purpose we apply mid gray only to the left and upper sides of the ingot bars. In the November 2003 issue of the UCUGA Digest you can read how to do that. In short: Light gray must be masked (with **mod.QuickMask** (*Generate/Invert*). Then those portions of the mask which determine the right and bottom sides must be erased with **mod.DrawMask** or **mod.MaskEd**. Finally, you must save the current image, and clear the whole memory to mid gray. Then apply the saved image to the color mask with **ldr.4Bit&Mask**. If you performed well, everything results in Pic #9.



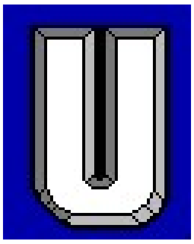
Pic #7: Cartoo-

It’s a bit of a work, especially when you have more than one letter and perhaps many curved letters like “o” or “c”. But it’s worth the effort!



Pic #8: The 3D lines must be painted manually

Before applying these letters to the final graphics, you can of course recolor your letters. The letters are of four colors, what you see here is black, mid gray, light gray, and white.



Pic #9: Correct shadowing, finished!

In the next issue we will talk about applying patterns and textures to our letters and graphics. Having done that, I will cover how to “write” with GoDot, so that you can try all what we performed up to now with your own GoDot generated texts.

Enough for today, it was quite a lot! ;-)

Command history

For Pic #1:

```
Load Replace “sidebottom.4bt”
Inst: Scroll

Repeat 7 times:
        Execute
        Set Amount: 1
        Direction: Left
        Execute
        Direction: Up
        Execute
        Leave
        Load Compose
        Mix: FGr 25%
        Mix "sidebottom.4bt"
End Repeat

Inst: .Histogram (Execute)
Join gr1, ora, lbl, lrd, grn, cya
Target: gr3
Swap blk, blu
Exit
Inst: Cartoon (Execute)
Inst: PixelEdit
(…Edit…)
Exit

Inst: Scroll (Execute)
Set Amount: 4
Direction: right
Execute
Set Amount: 6
Direction: down
Execute
Leave

Inst: QuickMask (Execute)
(Select:) blue
Generate
Leave
Load: 4Bit&Mask
Get 4Bit "dale2000,4bt"
Leave
Display
```

For Pic #9:

```
Load: 4BitGoDot
Load Replace "U.4bt"
Inst: Scroll (Execute)
Set Amount: 4
Direction: left
Execute
Direction: up
Execute
Leave
Load Compose
Mix: all 25%
Mix "U.4bt"
Execute
Direction: down
Execute
Leave
Load Compose
Mix: "U.clp"
Execute
Direction: right
Execute
Leave
Inst: .Histogram (Execute)
Join bwn, gr1, ora, lbl, cya
Target: gr3
Swap blk, blu
Exit

Inst: PixelEdit (Execute)
(…Edit…)

Inst: Cartoon (Execute)
Inst: PixelEdit (Execute and Edit)
Exit
Display

Inst: QuickMask (Execute)
(Select:) light gray
Generate
Invert
Leave
Inst: DrawMask (Execute)
Brushsize: 4
Mode: Clear
Draw Mask
(…Edit…)
Leave
(Save: 4BitGoDot)
Save "Undo 4Bit" (to Unit RAM if REU)
Inst: ClipWorks (Execute)
Full
ClrClp
Select: mid gray
Inside
Leave
Accept
Load: 4Bit&Mask
Load
(Get 4Bit from: Undo - if REU)
Get 4Bit "Undo 4Bit"
Leave
Display
```

Have fun using GoDot!