How to GoDot

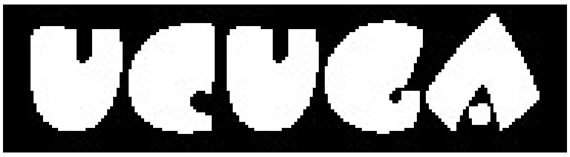
By Arndt Dettke

Working on clips (Part 4, A Sample Application)

Have you carefully examined the front page of this Digest? Did you notice this issue's main title? It's different, isn't it? If you like it, this is how to create your own newsletter main title. Come, follow me.

You need ldr.4BitGoDot for it, and the modifiers mod.ClipWorks, mod.StretchClip, mod.-PixelEdit, mod..Histogram, mod.Cartoon, and mod.Scroll. Later, you can save your results with svr.4BitGoDot, svr.PostScript or svr.GIF.

Although the text of this workshop ("UCUGA") was written by using GoDot itself I take it as given. You can download it from my site (www.godot64.de/download) as "ucugatitle.clp". It's a clip saved in GoDot's own 4Bit format.

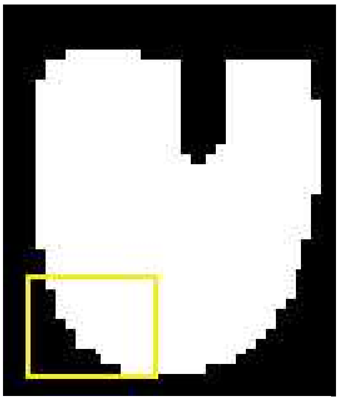


Pic #1: Written with GoDot using font "Beesknees".

If you have no access to the internet or you like to work on your own title I suggest you use any paint program, create your title text as graphics and save it in a format GoDot can read (Doodle, Koala, GIF, PCX, hires bitmap, OCP Art Studio, Paint Magic, and many more - see link on my site's main entry page). For this time I concentrate on processing "ucugatitle.clp" with you.

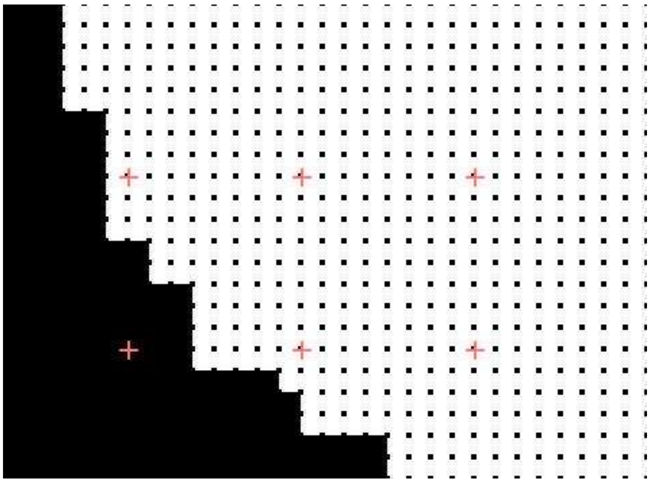
Load it to memory with **ldr.4BitGoDot** (default loader) and render the image. You'll see it's a rather small text in the upper left corner of the screen. Execute **mod.ClipWorks** to set a clip around it. Its genuine size is 23x6 cards. In the next step we will scale it up to screen width. To not distort it, the clip must have a height of 14. This is roughly the standard C=64 aspect ratio of 1.6 (23/14 is about 1.6). So your clip values have to be 0,0,23,14. Now install **mod.Stretchclip** and ***Execute***. When finished, render the image ("***Display***").

You'll see a larger text, quite jaggy, and surrounded with junk graphics remainders which we will get rid of by executing mod.ClipWorks again (the clip around the text is 0,1,38,10 this time). Click on "***ClrClp***" and select black as the color to clear with. Then hit "***Outside***" which deletes everything in memory but the clip. ***Leave*** ClrClp, "***Accept***" your settings and change the display mode to "***Hires***" now. Render again. GoDot only renders what is inside the clip and leaves the rest of the screen as it is. Since it's a smaller area this works faster than before. (If you don't like the crap that remains on screen, set "***Exec Area***" to "***Full***" for once, this will clear the display).



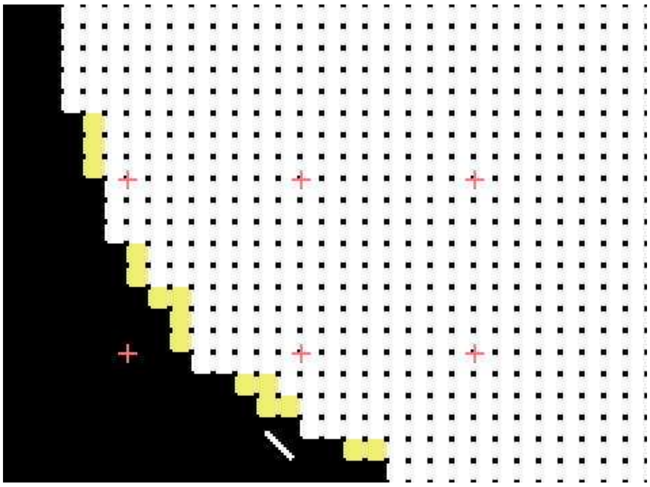
Pic #2: The middle U of UCUGA, many jaggies. The framed area is magnified in the next images.

Yes! The text is jaggy! What to do now? It doesn't look too impressive this way… Ok, we have to edit the image and paint the jaggies off. Install **mod.PixelEdit** and execute it. To "unjag" the text you look for pixel steps of at least two bits width at the borders of the letters. Click the "***O***" in the upper right box. GoDot will give you an overview then, and you can click in it and thus move the magnification window to this location.



Pic #3: Many two bit steps which must be filled.

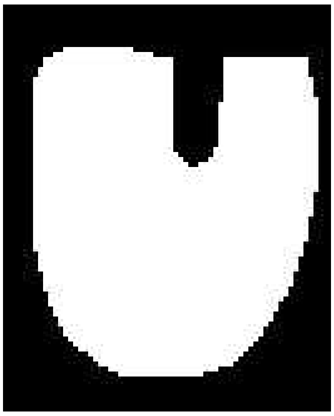
You fill the jaggies with white ink, starting in the black angle, moving the pen along the longer side of a step until you reach half the length of it (one more than half if the number of pixels is odd). You perform this with every two bit angle around the letter, vertically or horizontally. When finished, leave PixelEdit and render again. If you notice little peaks or even humps, go into it again and clear them away in black. Proceed with the next letter until everything is finished. It will take some time, don't forget the background music! ;-)



Pic #4: The highlighted pixels are filling the angles to let the letter finally look smoother.

This was the drudgery, now we can lean back and let GoDot do the work. Next step is setting the beautiful rim around the letters. For that we install **mod..Histogram** and execute it. It should show only two colors, black and white. ***Swap*** black with blue and ***exit*** Histogram.

Install **mod.Cartoon**. This modifier draws a black border around every colored area in the image (this is the reason why we had to exchange black by blue: otherwise the border would have been invisible). Execute it. This modifier doesn't open an own window but indicates being busy by a white bar wandering down the preview window (once). When finished, render the image (full screen). A nice white writing on blue with a black rim.

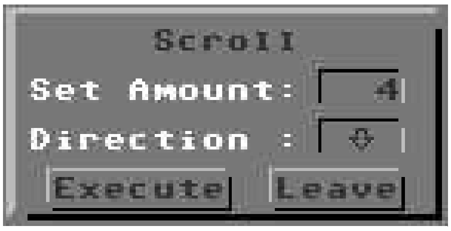


Pic #5: Unjagged.

But this Digest issue's title is colorful, how to avchieve that? Easy: install again mod..Histogram. It will now show three colors, black, much blue, and white. ***Swap*** black and yellow (now the rim is yellow), ***swap*** white and green (now the letters are green), and ***swap*** blue and black. ***Exit*** Histogram and render. Green letters with yellow borders on black.

Before we perform the last step – the black shadow – we must save the file as it is now. Owners of an REU save it to unit RAM as "***Undo 4Bit***" (every saver which opens the directory dialogue is applicable here), everyone else cannot but save it to disk (using **svr.4BitGoDot**).

The shadow is nothing else but the title itself. Only it is just black, and displaced by some pixels. So, what we do is install **mod.Scroll** (which we use for the first time) and execute it. We set an amount (of pixels to shift the image), let the direction by "***Right***" and ***Execute*** Scroll. The little white bar appears and goes again. Now, we click the direction to "***Down***" and execute again. We have shifted the entire image four pixels south-east now.



Scroll

Set Amount: 4

Direction : ⇩

Pic #6: The highest amount is 64 pixels at a time.

Next step (and then the finals): change colors with mod..Histogram. Install and execute it. ***Swap*** black and white (the background is white now). And now something new, we ***Join*** yellow and green with ***Target*** black. As a result we have a black shape of our letters on a white backing, verify this by rendering the image.

The finals! Enter ldr.4BitGoDot (is it still installed?) and ***Compose*** the image we saved even just now (as "Undo 4Bit") as ***Foreground***. Render the full image, and be happy, this is our main title!

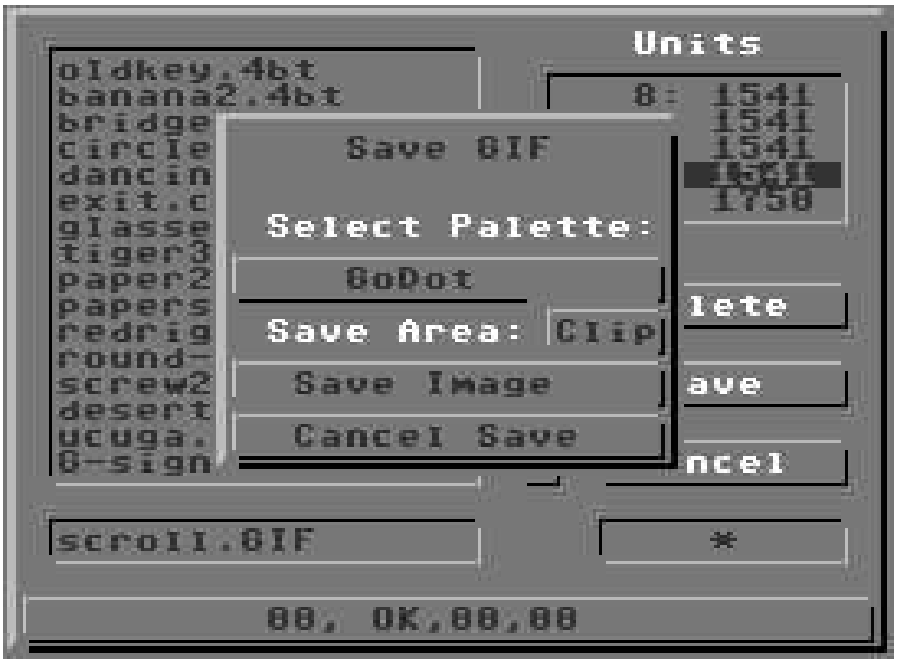
If you like to save it as an EPS clip first set a clip around the letters (0,1,38,10) and save it with **svr.PostScript** where its "***Save Area***" is set to "***Clip***". You find the same dialogue in **svr.GIF** (see picture #7). So, here we are!

Mod.Scroll – Module to shift an image to any of the directions Up, Down, Right, or Left. The highest amount of pixels at a time is 64. If you want to shift an image by 96 pixels right (12 cards) you enter 48, and execute it twice.

Mod.Cartoon – This modifier draws a black rim around every colored area in the image thus giving the impression of a cartoon painting.

Command history
to create a main title headline

(Load: 4BitGoDot)
Load Replace "ucugatitle.clp"
(Inst: Clipworks)
Execute
Clip: 0,0,23,14
Accept
Inst: StretchClip
Execute
Exec Area: Full
(Screenmode:) Hires
Display
Inst: ClipWorks
Clip: Wid 38, Hgt 10, Row 0, Col 1
ClrClp
(Select:) black
Outside
Leave
Accept
Inst: PixelEdit
Execute
(Edit…Exit…Display…Execute…Edit…)
Exit
Inst: .Histogram
Execute
Swap blk, blu
Exit
Inst: Cartoon
Execute
(Screenmode:) Multi
Exec Area: Full
Display
Inst: .Histogram
Execute
Swap blk, yel
Swap blu, blk
Swap wht, grn
Exit
Display
(Save: 4BitGoDot)
Save
Unit: RAM (or any drive)
Save "Undo 4Bit"



Save GIF
Select Palette: GoDot
Save Area: Clip
Save Image
Cancel Save

Pic #7: Save the result as a GIF image. The PostScript saver has the same UI. The filename suffix (.GIF or .EPS) gets attached automatically after saving.

Inst: Scroll
Execute
Set Amount: 4
(Direction: Right)
Execute
Direction: Down
Execute
Leave
Inst: .Histogram
Execute
Swap blk, wht
Join yel, grn

```
Target: blk
Exit
- if an REU was used:
        Load
        Source: Undo
        Compose Foreground
- if a drive was used:
        Load
        Compose Foreground “Undo 4Bit”
Cancel Load
Display

Save: GIF (or PostScript)
Save “ucugatitle”
Save Area: Clip
Save Image
```

Mod.PixelEdit has been updated for this issue of the UCUGA digest. Download it from my site (www.godot64.de).

Have fun using GoDot!