

How to GoDot #9

By Arndt Dettke

Writing with GoDot

Writing with GoDot is not like writing with geoWrite. To be true, it's a drudgery, more like taking your scissors, a pot of glue and an old newspaper and tinkering a blackmail letter. However, let me take you with me to explore this important field of image work. We'll cover a feature of **ldr.4Bit-GoDot** which was unveiled up to now: the destination gadget "*Load to*", and will heavily use **mod.Scroll** (to overlap the letters) and **mod.PixelEdit** (as a looking glass).

All the past years of Commodoring have always been a struggle against the restrictions the C64 comes along with. Only four or even only two colors per 8x8-tile? Only 320x200 pixels or even only 160x200 pixels? Only 16 fixed colors? Only this dull built-in character set? We have extended all of that today, in most cases thanks to those great ingenious demo programmers and also thanks to some ahead looking incredible productivity people.



Pic #1: A headline written and processed with GoDot. The font I used is named "Impact".

It's a good feature of poster arts to have text added to the artwork, no, better say: as a part of the artwork. I always wanted GoDot to be able to con-

tribute to this kind of artwork, too. Many of GoDot's modifiers are tools to affect geometric forms (like letters), change their view, dissolve them, melt them with their surroundings. All to give you the means to create new, never seen things on a 64. Let me give you some insight in what I mean this time and in the next issues of the UCUGA Digest. First follow me to create the text in image #1 below. And have a glimpse at all those other nice text-in-an-image examples here on the following pages.

GoDot provides four different font faces up to now which you can all download from my site (URL see end of the article), these are: *BeesKnees*, *Impact*, *Desdemona*, and *Snap* (they are derived from Windows fonts). The first of these you already know since they have been used for the UCUGA logo on its recent issues.

Get your "scissors" now! The "MetalPlate" headline in pic #1 is written in Impact, and we write onto an empty black background. All GoDot fonts are hires, white on black (except Desdemona which is white and gray on black), have a distinct baseline, and come along as individual letter clips.

One letter – one clip,
one font – one clipsize.

So, we first prepare for writing and install **mod.ClipWorks**, enter it, and clear the inside of the full image area to black (*Full, ClrClp, black, Inside, Leave*). The Impact font has a size of 5x5 cards which you can read when you first open the font disk. In ClipWorks, we now adjust width and height of a clip to this particular size of 5 both, at a position of 0, 0 (upper left corner).

We leave ClipWorks. Before we change to **ldr.4BitGoDot** we set the *screen mode* to *Hires*

and let the *Exec Area* be “Full”. After having entered the loader, you can select where the image will be loaded to. By default, this is set “to Full”, meaning that the image will start in the upper left corner of the screen. There are three more options to select: “to Clip”, “into Clip”, and “to Origin”.



Pic #2: You can select where from and where to.

“To Clip” means that the upper left corner of the image to be loaded starts at the current clip position. Portions of the image that would exceed the right and bottom screen borders get cropped. In “into Clip”, the loader attends the complete GoDot clip setting and crops everything which exceeds its limits. “To Origin” would load a clip to exactly the position where it was taken from. A standard full screen image (with suffix “.4bt”) gets loaded to full screen then, accordingly. We will use option “into Clip” to write in GoDot.

Load (in *Replace* mode) the first letter, the capital M. You’ll notice a number (“31”) in front of the clip’s filename. This number tells you the true width of a character. Instead of 40 pixels (5x8 pixels) the “M” is only 31 pixels wide (one less than four cards). To produce a balanced look of your writing your main task is to load every letter to a position near the true right margin of the letter before. It’s good to have at least a gap of two pixels between Impact letters. You shouldn’t ever undergo two pixels distance because otherwise you have a problem in multicolor mode display which is the standard display mode in GoDot.

In any case, we have to shift the image anyhow to be able to load a clip to a pixel-exact position. This is where **mod.Scroll** comes in handy. In ClipWorks we set a position for letter “e” of 5

cards farer to the right, and then we shift the whole image by 7 pixels in the same direction. When we now load the “e” it leaves a gap to the “M” of our wanted 2 pixels. After loading we have to shift back the image to its original position (7 pixels to the left).

The “e” has a width of 21 pixels. With the gain of 1 pixel from our shifting process, the right margin of the “e” is now 22 pixels east of the card it starts in and 14 pixels right of the “M”-clip (8 pixels overlap). Thus, we add 2 (cards) to the last clip column position in ClipWorks (which equals in 16 pixels distance). This would already include our 2-pixel gap. But the next letter is a “t” which doesn’t entirely start off the left margin because of the t-line. It leaves a gap of two pixels itself, and it looks better to also shift the “t” closer to the “e”.

So, we shift the image 2 pixels right, and then load the “t”. This time we have to use option “Compose – Foreground” because otherwise portions of the “e” would be overwritten and vanish. Don’t forget to shift back left 2 pixels. It’s a good idea to frequently have a look at the writing with a “looking glass” (**mod.PixelEdit**) to better value the correct distance between the letters (instead of computing it like we did above).



Pic #3: This is how the letters overlap. See our table for the appropriate values.

Well, this is how you write in GoDot. Not too comfortable, but nevertheless efficient. Use our table to complete our “MetalPlate” headline on your own.

The next workshop issues will cover how to scale up and down, move clips, turn clips, and how to apply effects to the writing like can be seen in our little gallery on the following pages and in pic #1.

Table 1

Letter	M	e	t	a	l	P	l	a	t	e
Column	0	5	7	9	12	14	17	18	21	23
Scroll Amount	-	7	2	1	2	6	6	2	5	5
Replace/Compose	r	r	c	r	r	r	r	r	c	c

Command history

Preparation:

(Inst: ClipWorks)
 Execute
 Full
 ClrClp
 (Select:) black
 Inside
 Leave
 Clip: Wid 5, Hgt 5
 Accept
 (Screen Mode:) Hires
 Exec Area: Full

First letter:

(Load: 4BitGoDot)
 Load
 Load: into Clip
 Replace "31-M.clp"
 Display

Next letters:

Inst: ClipWorks - Execute
 Last
 Clip: Col 5 (*and so on, see table*)
 Show
 Accept
 Exec Area: Full
 Inst: Scroll - Execute
 Set Amount: 7 (*see table for further values*)
 Direction: right
 Execute - Leave
 Load Replace "21-e.clp" (*and so on*)
 (*or: Load Compose Foreground "file"*)
 Execute (*of Scroll*)

Direction: left

Execute - Leave

Display (*STOP to break rendering*)

Inst: PixelEdit – Execute

(*look at right margin, note distance to next card*)

Exit

Repeat section "Next letters" until all letters are written.

Save headline:

Inst: ClipWorks – Execute

Clip: 0, 0, 25, 5 (*the complete headline*)

Save: 4BitClip

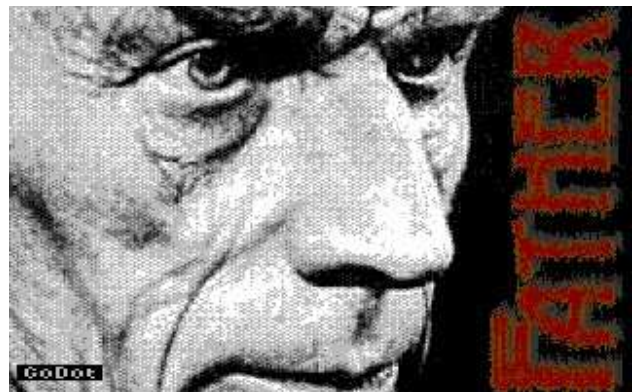
Save "metalplate.clp"

I have uploaded the four font disks as zipped d64 files to my download site at <http://www.godot64.de/download>. Also to be found there are all the images used in the gallery in their original stage (without text in them). Look forward to the next UCUGA issues to learn how to apply the text to them processed as seen here.

Have fun using GoDot!



Pic #4: Font „Snap“, scaled up and processed with a texture.



Pic #5: „Desdemona“, scaled down, turned and heavily processed.



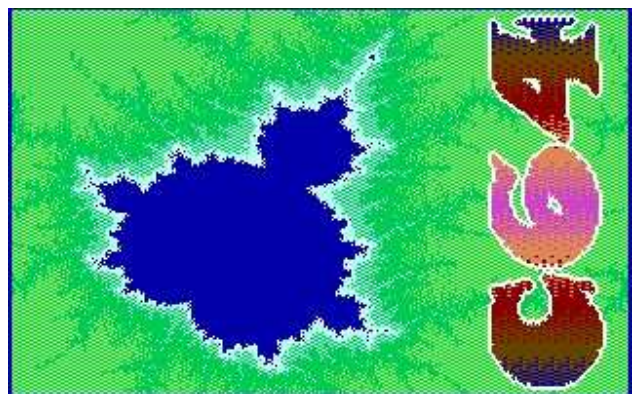
Pic #6: This one with processed font “Desdemona”.



Pic #7: Just font „BeesKnees“ inserted here with no further processing.



Pic #8: „BeesKnees“ too, but this time processed and scaled up.



Pic #9: Font „Snap“ processed with a gradient.