Metafont als matrixprinter

Phons Bloemen

phons@ei.ele.tue.nl

Abstract

Een verslagje van een stoeipartij met METAFONT, waarbij METAFONT wordt 'misbruikt' als matrixprinter. Door alleen de vorm van de 'printnaalden' te definieren, en een beperkt aantal plaatsen waar ze neer mogen komen, kun je heel snel een font in elkaar zetten.

1 Het Apelioteslogo

Mijn eerste ervaringen met METAFONT bestonden uit het digitaliseren van het logo van mijn studentenhockeyclub 'Apeliotes' in Nijmegen. Dit logo is in 1991 gemaakt.

Bij dit logo hoort een font 'Capital Baseball' genaamd. Dit font vind je vaak terug op de ruggen van voetballers en hockeyers. Ook Studio Sport gebruikt het. Het font moest zodanig in het logo worden verwerkt dat het moest 'rondlopen' (zie het plaatje). Daarvoor moest het geroteerd en vervormd kunnen worden. Een simple manier om een font te maken is het bepalen van een aantal referentiepunten, waartussen je dan rechte lijnen trekt. METAFONT kan heel ingewikkeld doen met splines etcetera, die zijn echter niet nodig. De vlakken tussen de lijnen vul je vervolgens gewoon op. METAFONT heeft een functie `fill(z1 -- z2 -- z3 -- cycle)` die op deze wijze een mooi driehoekje opvult. Bij het Capital Baseball font (en de vervormingen daarvan voor het logo) ben ik een dergelijke manier te werk gegaan: referentiepunten bepalen en vlakken vullen.

Een tweede font, 'Simple', is gemaakt om met het logo wat leuke 'gimmicks' uit te halen. 'Simple' was ook een manier om eens met METAFONT te oefenen: er worden 9 referentiepunten gebruikt, waartussen met de `draw` functie de lijntjes worden getekend. De 9 referentiepunten bestaan uit de 4 hoekpunten van het uiteindelijke teken en alle punten 'halverwege' die 4 hoekpunten, waarbij punt 9 dus het middelpunt van het teken is. De 'D' is op deze wijze gemaakt:

```
draw z$p1 -- z$p2 {right} .. z$p6 ..
    {left} z$p1 withweight s;
```

Hier ligt punt 1 linksonder, 2 linksboven, 3 rechtsboven, 4 rechtsonder, en 6 ligt halverwege 3 en 4. De puntjes in de 'draw' opdracht duiden op een 'spline', met 'right' kun je de richtingscoefficient bepalen waarmee de spline uit een punt 'vertrekt'. De parameter 's' bepaalt de lijndikte. Het 'Simple' font diende als eerste vingeroefening, en is gewoon een snel in elkaar geflanst ding.

2 Metafont to the limit

Het uitendelijke logo is één (nogal grote) letter geworden. De krans met 'Nijmeegse Studenten Hockeyclub' is ook één letter. De balk van het pijlsymbool kan gebruikt worden om een tekstje in te zetten: daarvoor zijn de letters van 'Simple' in diapositieve vorm aanwezig. Er waren flinke vergrotingen nodig van dit logo. Een vergroting tot 20x20 cm bleek geen probleem te zijn voor de emTEX drivers. Het gaat hier om het afdrukken van 1 teken dat 20x20cm groot is, en de *dvi* standaard specificeert slechts een maximale lettergrootte van 1x1 inch! Voor dergelijke grapjes heb je wel een 'big' METAFONT nodig. Bij het logo-font hoort ook nog een style-file met briefhoofd en pagina-opmaak voor het verenigingsblad en smoelenboek. Toen ik Nijmegen verliet, verdween daarmee ook de 'TEXpertise' uit de club. Het logo is er nog wel: in de vorm van een tot het uiterste opgeblazen PCX file.

3 Nog meer fontjes

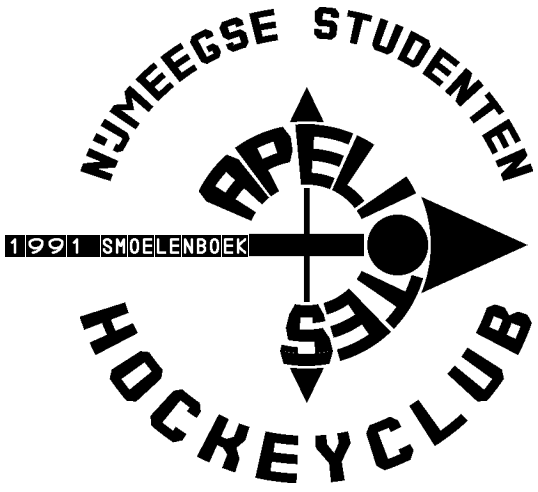
Bij het maken van dit verhaaltje voor de MAPS besloot ik er nog wat fontjes bij te maken, zie figuur 3.

De fonts voor de 'segment-displays' zijn op dezelfde manier gemaakt als 'Capital Baseball': referentiepunten bepalen en vlakken vullen. Het maken van een font voor zo'n display wordt dan heel simpel: je definieert eerst alle segmenten als macros. De letters kun je dan maken door de juiste segmenten 'aan te sturen'.

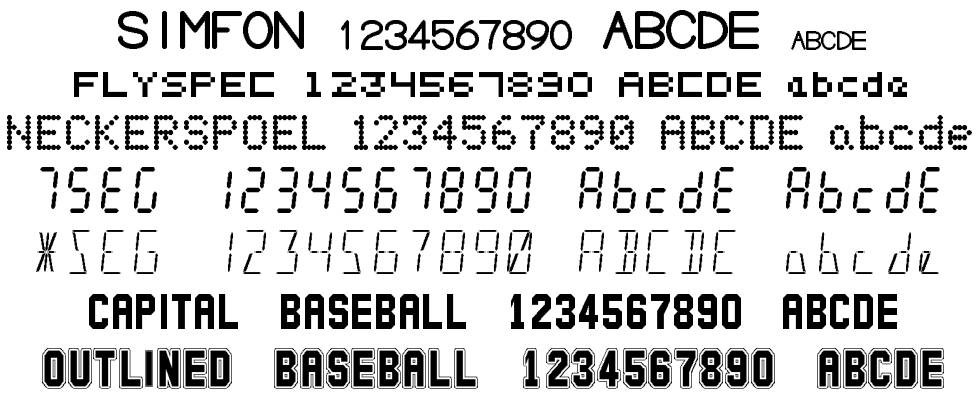
De fonts 'Flyspec' en 'Neckerspoel' benaderen de zaak iets anders: hier wordt METAFONTgebruikt als matrixprinter. Kern is de functie in figuur 4.

Met deze functie kun je allerlei soorten dot-matrix tekens maken. De vorm van de naalden van de printer kun je zelfs veranderen: gebuik gewoon een ander soort pen in het `pickup` commando. Het aanmaken van de tekens zelf is niet meer dan het opgeven van lijsten met punten.

De namen 'Neckerspoel' en 'Flyspec' lijken misschien vreemd voor deze simpele matrix-fontjes. 'Neckerspoel' komt van het futuristische busstation in Eindhoven, waar ze het gebruiken in de 'flip-dot' displays boven de bussen. Ook komt het voor op oude terminals. Het gebruikt een 5x7 matrix. 'Flyspec' komt uit de Jargon File [2], en gebruikt een 5x5 matrix.



Figuur 1: *Het Apeliotes-logo, ietwat aangepast*



Figuur 2: *Samples van de fonts, in 20 punten*

Figuur 3: *De 'entry' van Flyspec 3 in de Jargon File, gezet in Flyspec 3* `:-)`

4 Een gereedschapskist toe

Geinspireerd door de zuurstokletters in [1] heb ik een toolkitje gemaakt voor het 'aanpassen' van METAFONT fonts. De functie `endchar` wordt in `plain.mf` gedefiniëerd. Hij wordt aangeroepen wanneer een teken al helemaal is gedefinieerd, en het alleen nog maar 'afgedrukt' hoeft te worden. Op dat moment kun je nog transformaties op het hele teken loslaten. Met een aagepaste `endchar` kun je dus leuke dingen gaan doen. De toolkit `mftools.mf` bevat functies die je kunt gebruiken in je aangepaste `endchar`. Hij bevat allerlei transformaties als spiegelingen, rotaties, het in diapositief zetten, maar ook 'outline' en 'double outline' (zie het Capital Baseball voorbeeld). Ook de matrix-printer komt hier weer terug: hij wordt gebruikt als patroongenerator. Met de matrix-print functie wordt een patroon (bijvoorbeeld een arcering) op het gehele karakterveld gezet. Het al gegenereerde teken wordt nu gebruikt als stans, om het resultaat uit dit patroon te snijden. De zuurstokletters zijn hiermee eenvoudig te maken. Ook deze 'buggy' versie van `cmtt10` komt uit deze stal.

Je hoeft er dus niet eens de originele `cmtt10.mf` file voor aan te passen. Ook een 'grayed font' is geen probleem De functie `otilepic` stanst het al gegenereerde teken uit in het 'papier' met bugjes (`picture groundpattern`), en voegt tevens een 'outline' van het teken toe. Er is ook een functie `tilepic`, die de outline niet toevoegt.

```
%%% This draws the dot matrix (a generic function, can
%%% be used for many dot matrix characters).
%%% Parameters of 'flychar':
%%%     c : character number (decimal)
%%%     k : # dots horizontal ('width')          0  1  2  3
%%%     l : # dots vertical ('height')           4  5  6  7
%%%     m : # dots descender ('depth')           8  9 10 11
%%%     t : sequence of dot numbers to set.     12 13 14 15
%%%         Dot numbering of a 4 x 5 'flychar':  16 17 18 19
def flychar(expr c,k,l,m)(text t) =
beginchar((char c), (k+1)*uu#, l*uu#, m*uu#);
   clearit;
   pickup pensquare xscaled (w/(k+1)) yscaled((h+d)/(l+m));
   for a = t :
      drawdot((0.5+(a mod k))*w/(k+1),h-((0.5+floor(a/k))*(h+d)/(l+m)));
   endfor;
endchar;
enddef;
%%%
......
%%% And this prints the number 0:
flychar(48,5,5,0)(1,2,3,5,9,10,14,15,19,21,22,23);
%%%
```

Figuur 4: *The matrix printhead in* METAFONT

```
% Computer Modern Bugs            %%% Example of 'mftools'
if unknown mag: mag := 1; fi;
input mftools;                    %%% get the toolkit
uu# := 0.1pt#;                    %%% define dot size of bugs
picture groundpattern;            %%% create 'clean' paper
gentile(0.01,13,14,2.5uu#,1.7uu#,0.8uu#,pensquare,groundpattern)(
1,2,6,7, 14,15,19,20, 28,29,31,32, 40,43,46, 52,56,60, 65,69,73, 78,82,86,
91,95,99, 105,108,111, 119,120,121,122,123, 58, 85, 96, 67, 106);
                                  %%% fill it with bugs
clearit;
def endchar =                     %%% modify the 'endchar' of plain.mf
 scantokens extra_endchar;
 reversevideo;                    %%% try some tools from 'mftools'
 otilepic(groundpattern,6uu#);    %%% cut character out of bug paper
 xmirror;                         %%%
 if proofing>0: makebox(proofrule); fi
 chardx:=w;     % desired width of the character in pixels
 shipit;
 if displaying>0: makebox(screenrule); showit; fi
 endgroup enddef;
input cmtt10;                     %%% pick any font you like :-)
```

Figuur 5: *Voorbeeld van een vervormd font*

5 Verkrijgbaarheid

Het hele pakketje draagt de naam 'Capital Baseball'. Het staat helaas nog niet op de CD-ROM (de extra fontjes zijn vrij recent). Op CTAN zal het in de directory `/fonts/capbas` worden geplaatst.

Happy METAFONTing...

Referenties

[1] Alan Hoenig. When TEX and METAFONT work together. *MaPS*, 10:124, 1993.

[2] Eric S. Raymond. *The New Hacker's Dictionary*. MIT Press, London, GB, 1991. aka `Jargon File v2.9.6`.