

Commodore

INFC

PRIJS f 7,95 / Bfr. 160



8 710966 001332

ONAFHANKELIJK BLAD VOOR AMIGA- EN C64- GEBRUIKERS Jaargang 7, NO.8, december '90

**Amiga '90
in Keulen**

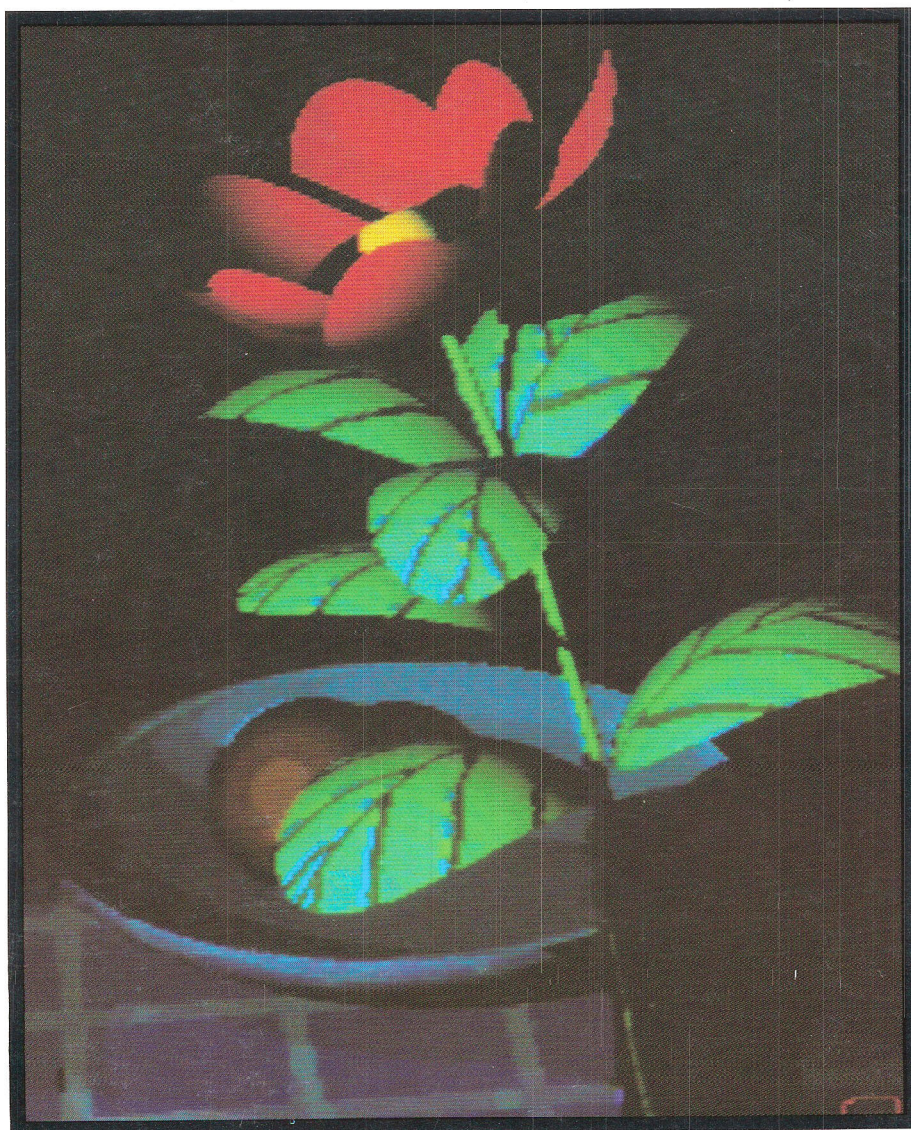
Real 3D getest

**C64-
Beeldmanipulaties**

**Vizawrite Desktop
Nederlands**

**Power PC Board
voor A500**

**De C64-
Basic Interpreter**



COLOFON

Commodore INFO is een uitgave van:
Sala Communications
Postbus 43048
1009 ZA Amsterdam

Uitgever:
V. Sharfman

Redactie:
ir. L. Sala (hoofdredacteur),
W.A. Scheer (eindredacteur),
drs. U. Schuurmans,
R. Goudriaan, B. Venema,
P. Boncz, MGCC/Johan&Johan,
Michel de Boer, Hylke Sprangers,
telefoon: 020-228871

Productie:
drs. H. Zoete,
J. Broekhuizen

Advertentie-exploitatie:
ing. V. Sala, D. van Vlijmen
ing. B. Sala, H. Bia
telefoon: 020-273198

Abonnementen en administratie:
Marjo Jansen
telefoon: 020-248006

Vragen betreffende abonnementen ontvangen wij
bij voorkeur schriftelijk, met meesturen van
het omslagetiket.

Telefonisch uitsluitend tussen
12.00 en 15.00 uur.

Vragen over artikelen:
Voor vragen en opmerkingen over artikelen of
anderszins graag een briefje aan de redactie
(zie boven).

Listingtelefoon:
02155-25162
(ma: 17.00-21.00 u)
LAATSTE MAANDAG IN JANUARI 1991
VOOR HET LAATST

Geos Gebruikers Groep:
Postbus 52,
1300 AB Almere
03240-10041

Cover:
REAL 3D

Zetwerk & druk:
NDB, Zoeterwoude

Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

© 1990 Commodore INFO
Alle rechten voorbehouden
ISSN: 0169-3085



Inhoud

Geluid en muziek op de 64 6

Synthesizer-geluiden op de 64. Wie had dat gedacht? In dit artikel gaan we eens lekker 'poken', vooral in de geluidsadressen. Het resultaat is opmerkelijk.

GEOS Info 14

Vragen over GEOS zullen er altijd wel blijven. Peter Boncz behandelt er ook nu weer een aantal. Hierin ook nog een kleine rectificatie van de Bouncing Bars-listing.

De Basic Interpreter 16

Zelf instructies maken op de 64. Het maken hiervan vereist nogal wat vaardigheden van de programmeur. Hoe het moet en hoe het niet moet.

Graphics (rectificatie) 21

Met Bouncing Ball is in nummer 7 nogal wat mis gegaan. Hier de juiste versie.

Beeldmanipulaties (4) 22

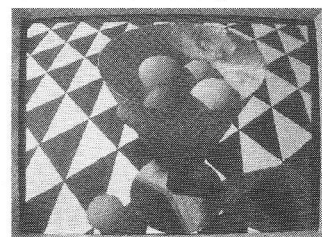
Beeldmanipulaties gaat deze keer over het scrollen. Vergezeld van de nodige voorbeelden, worden deze principes op de 64 duidelijk gemaakt.

GEOS machinetaal (slot) 26

De uitsmijter van de cursus machinetaal onder GEOS: 3dView. GEOS-meester Peter Boncz heeft deze keer zijn 64 ondersteboven gekeerd en er uitgehaald wat er in zat. En dat was nogal wat.

REAL 3D 52

Animaties en Amiga. Beter kan het niet. Naast de wat meer bekende Sculpt-versies, gaat 3Gitaal uit Amsterdam een geheel nieuw programma in Europa distribueren: Real 3D, een Fins produkt met een geheel eigen aanpak.



AMIGA '90 55

De jaarlijkse beurs in Keulen, werd dit jaar voor de tweede keer georganiseerd. Een verslag van het nieuwste.

Nieuws uit Keulen 57

Extra korte nieuwsberichten uit Keulen.

Boekbesprekingen 61

Een aantal boeken over Desktop Video deze keer, alsmede een boek over WordPerfect op de Amiga.



Kerst- en nieuwjaarskaarten 63

Met de feestdagen in het verschiet en de jaarlijkse schrijverij die daarbij schijnt te moeten horen, maken we dit jaar onze kaarten zelf. Een programma en de Amiga. Prettige feestdagen!

VizaWrite 2.02 Ned.

65

Een Nederlandse tekstverwerker die werkt volgens het WYSIWYG-principe. IFF-plaatjes inladen is mogelijk en het importeren van teksten is voor deze versie geen enkel probleem.

Desktop Video op de Amiga

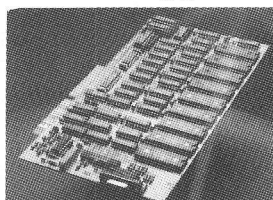
71

Voor de Amiga is Destop Video wat een speen is voor een baby. Maar voor velen blijft verborgen wat nu eigenlijk wel de mogelijkheden zijn.

Hires videokaarten

80

Monitoren met een hoge oplossing kunnen alleen maar goed functioneren met een goede videokaart.



KCS Power PC Board

83

KCS hoopt bij het verschijnen van dit nummer, versie 2.0 van de software behorende bij het Power PC Board af te hebben. Deze versie zal ook harddisks ondersteunen. Wij namen de Alfa-versie 1.54 ag onder de loep.

Monitoren

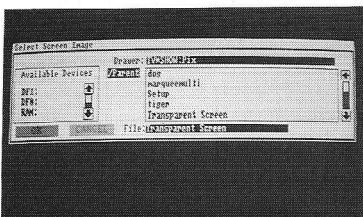
86

Welke monitor voor de Amiga en welke na- en voordelen is er bij ieder type te vinden?

TV*Text Pro en TV*Show

92

Aftitelen is een kunst apart en je kunt het in feite zo gek maken als je zelf mooi vindt. Met TV*Text Pro en TV*Show gaan we de goede kant op...



Vaste rubrieken:

Kleine advertenties 96

Listing C64:

Character edit 36

Listings Amiga:

TelephoneCalc 38

Mastermind 44

Rekenen Amiga 46

Turn-It 47

Typeles 48

Woorden raden 49

Redactioneel

De Amiga '90 in Keulen is weer geweest. Het barste er net als vorig jaar, de eerste keer in Keulen, van de bezoekers. In Duitsland lijkt het allemaal wel te kunnen wat in Nederland niet mogelijk is: veel buitenlandse bedrijven met name uit Engeland en de Verenigde Staten. Maar ook KCS uit Dordrecht was van de partij.

Ook in Engeland was een Amiga-beurs. Het publiek bleek daar van heel ander kaliber dan in Duitsland. In Engeland ligt meer de nadruk op spelletjes. De combinatie-aanbiedingen die Commodore Engeland maakt, laten dit ook duidelijk zien. Een Amiga 500 met een aantal spellen tegen een aantrekkelijke prijs.

Een extra dik nummer deze keer met veel nieuws uit Keulen in dit nummer in twee artikelen. Ook de 64-fans komen ruim aan bod. Voor hen hebben we dit keer een fiks aantal artikelen.

Op de Amiga hebben we de afgelopen tijd ook een aantal zaken bekeken. Speciale aandacht behoeft Real 3D van 3Gitaal. Dit bedrijf doet de laatste tijd goede zaken. Van Real 3D zijn ze bijvoorbeeld distributeur voor Europa geworden.

Voor de redactie van Commodore Info breekt een rustige tijd aan. Dit nummer is namelijk het laatste dat we gaan uitbrengen. Helaas hebben we door het teruglopende abonnementenaantal en de dalende advertentie-inkomsten hiertoe moeten overgaan.

Dat betekent dat de vragen die nog op de redactie liggen, zullen nog worden beantwoord.

De redactie en haar medewerkers hebben altijd met veel plezier aan Commodore Info gewerkt. Vooral omdat dit het eerste blad was, waarmee Luc Sala Sala Communications startte betreurt de redactie het des te meer dat dit het laatste nummer moet zijn. Hartelijk dank aan degenen die ons die jaren trouw zijn gebleven, ook al zat het wel eens tegen. En wellicht tot ziens in één van onze andere bladen.

Wilbert Scheer

COMMODORE INFO STOPT!!!

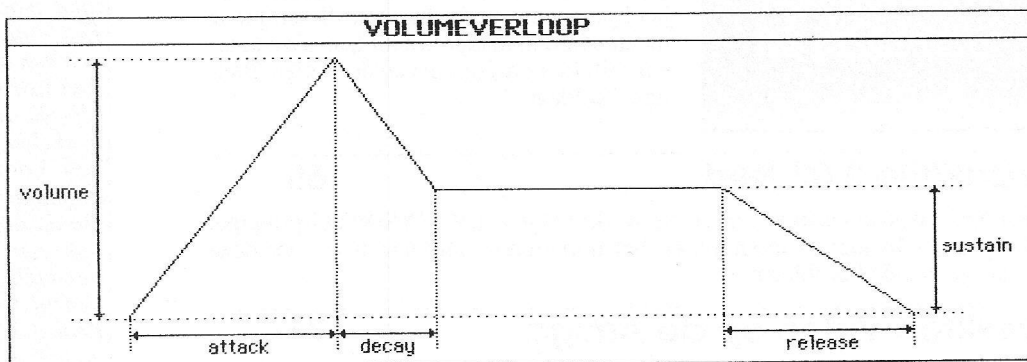
Na dit nummer zal er geen Commodore Info meer verschijnen. De listingtelefoon blijft nog tot de laatste maandag in januari 1991 dienst doen en zal daarna ook zijn einde hebben.

De abonnees die hierna nog een tegoed hebben, zullen binnen twee weken na het verschijnen van dit nummer schriftelijk bericht krijgen van de mogelijkheden.

GELUID EN MUZIEK OP DE 64

Al duizenden jaren lang houdt de mens zich bezig met het maken van muziek. Eerst door het gebruik van de stem (zingen) en handen (klappen), later door het bespelen van muziekinstrumenten. In de loop der tijd zijn er zeer veel verschillende muziekinstrumenten uitgevonden, die allen een eigen klank hebben. Grofweg zijn de muziekinstrumenten in drie categorieën in te delen: de blaasinstrumenten, de slaginstrumenten en de snaarinstrumenten. Met het ontstaan van de micro-electronica is er een nieuw instrument gemaakt, de synthesizer. Dit instrument past in geen van de drie eerder genoemde categorieën. Met de uitvinding van de synthesizer is dus tevens een nieuwe categorie instrumenten ontstaan: de digitale instrumenten. De eerste synthesizers brachten slechts een iele ziellose toon voort en de melodien die er op gespeeld werden, mogen eigenlijk niet als muziek bestempeld worden. De huidige synthesizers stellen de musicus echter in staat om ware muzikale kunstwerken te produceren. Als voorbeeld noemen wij de muziek van Jean Michel Jarre.

Ook is het mogelijk om met synthesizers geluidseffecten te creëren. Denk bijvoorbeeld aan het geluid van bomontploffingen, overvlie-



Figuur 1

gende ruimtevaartuigen of het ruisen van de zee.

Om te begrijpen hoe een synthesizer werkt moeten we eerst weten wat geluid is. Geluid is een trilling van de lucht (d.w.z. een opeenvolgingen van drukverschillen in de lucht). Deze trilling plant zich voort door de lucht. Het trommelveel in het menselijk oor kan de luchtdrukverschillen detecteren, waardoor wij geluid kunnen horen. Een synthesizer brengt nu op de volgende manier geluid voort: een chip genereert een stroom, waarvan het voltage continu verandert. Deze stroom wordt naar een luidspreker gestuurd. Door de wisselingen in het voltage trilt de luidsprekerconus, die op zijn beurt de lucht laat trillen.

Het belangrijkste onderdeel van de synthesizer is dus de chip die de stroom genereert. De Commodore 64 bevat een dergelijke chip, de Sound Interface Device (SID) chip.

Hoewel de Commodore 64 al bijna een antieke machine is, was deze computer op muzikaal gebied zijn tijd ver vooruit. De meeste computers kunnen alleen schrille pieptonen voortbrengen, die niet om aan te horen zijn. De 64 kan meer; futuristische symfonien en fantastische geluidseffecten behoren tot de mogelijkheden. In dit artikel zullen we in vogelvlucht de mogelijkheden van de SID bespreken. De voorbeeldprogramma's die we hierbij gebruiken zijn nogal sumier. De programma's zijn dan ook niet bedoeld om de SID in volle glorie te laten horen, maar om te laten zien hoe de SID vanuit een programma aangestuurd moet worden. Door met de kennis uit dit artikel te experimenteren zult u zelf wel ontdekken hoe muzikaal uw goede oude Commodore is.

De SID chip

Via de geheugenadressen 54272-54300 kan met de SID

gecommuniceerd worden. Behalve de laatste vier adressen zijn alle adressen write-only. Dit betekent dat in deze adressen alleen gePOKEd kan worden en niet gePEEKed.

De SID bevat drie onafhankelijke geluidskanalen, ook wel stemmen genoemd. Elk van deze stemmen kan een enkele toon produceren. Een stem beschikt daartoe over een 16-bits frequentie, vier verschillende golfvormen, een envelope generator, synchronisatie, ringmodulatie en een drietal filters.

Tonen

Muziek bestaat uit een opeenvolging van tonen. Om dus muziek te kunnen maken, moeten we een toon kunnen maken. Wat is nu precies een toon? Een toon wordt gekarakteriseerd door drie dingen:

- frequentie
- volumeverloop
- timbre

waarde	attack	decay/ release
0	2 ms	6 ms
1	8 ms	24 ms
2	16 ms	48 ms
3	24 ms	72 ms
4	38 ms	114 ms
5	56 ms	168 ms
6	68 ms	204 ms
7	80 ms	240 ms
8	100 ms	300 ms
9	250 ms	750 ms
10	500 ms	1.5 s
11	800 ms	2.4 s
12	1 s	3 s
13	3 s	9 s
14	5 s	15 s
15	8 s	24 s

Figuur 2

De frequentie is de snelheid waarmee de luchtdrukverschillen elkaar opvolgen. Hoe sneller de opeenvolgingen, hoe hoger de toon zal klinken.

Een toon heeft niet een constant volume, maar het volume varieert in de tijd. Een toon zwelt eerst aan tot een bepaald volume, kan vervolgens iets afzwakken tot een zachter volume. Dit volume houdt dan enige tijd aan en tot slot sterft het geluid weg. Dit wordt het volumeverloop van een toon genoemd.

Met het timbre van een toon bedoelen we de klank en de sfeer van een toon. Een toon kan warm of kil klinken, schril of zacht.

Het volumeverloop en het timbre stellen ons in staat om verschillende muziekinstrumenten van elkaar te onderscheiden.

We zullen nu laten zien hoe elk van deze drie karakteristieken met de SID kunnen worden gerealiseerd.

Frequentie

De frequentie van de toon wordt aangegeven met een 16-bits waarde. Deze waarde moet in de vorm van een lo- en hi-byte in de adressen 54272 en 54273 worden gepoked. Met de volgende formule kunt u berekenen welke waarde bij welke frequentie hoort:

waarde = frequentie/0.06097

In appendix P van het handboek staat een tabel met waarden voor een groot aantal muzieknoden.

Volumeverloop

Voor het regelen van het volumeverloop van een toon beschikt de SID over een zogenaamde envelope generator. Vier belangrijke termen die bij het volumeverloop een rol spelen, zijn: attack, decay, sustain en release. In figuur 1 zijn deze termen grafisch weergegeven. Een toon zwelt eerst binnen een bepaalde tijd aan tot een maximum volume. Deze tijd wordt aangeduid met attack. Vervolgens zwakt de toon af tot een lager volume. Dit volume wordt sustain genoemd. De term decay duidt de tijd aan die nodig is voor het afzwakken. Tot slot sterft de toon af. De snelheid van het afsterven wordt release genoemd.

Via de adressen 54277 en 54278 kunnen de attack, decay, sustain en release worden doorgegeven aan de envelope generator. De waarden van bits 0-3 van adres 54277 bepalen de decay; bits 4-7 bepalen de attack. Met de volgende instructie kunnen de attack en decay worden ingesteld:

**POKE 54277, decay +
attack * 16**

De waarden van bits 0-3 van adres 54278 bepalen de release; bits 4-7 bepalen de sustain. Het volgende commando stelt de sustain en release in:

**POKE 54278, release
+ sustain * 16**

De betekenis van de waarden voor attack, decay, sustain en release zijn in figuur 2 weergegeven.

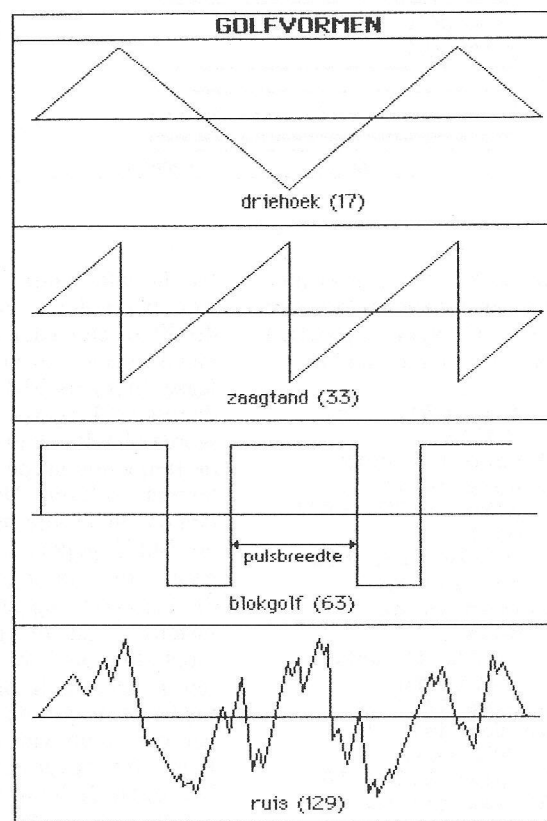
Timbre

Het timbre van een toon wordt bepaald door de zogenaamde golfvorm. De SID

beschikt over vier verschillende golfvormen: zaagtand (scherp geluid, zoals een viool of saxofoon), driehoek (helder geluid, zoals een fluit), blok golf (nasaal geluid, zoals een hobo) en ruis (geschikt voor het nabootsen van het ruisen van de zee of het gejuich van een grote menigte). In figuur 3 zijn deze golfvormen grafisch weergegeven. Om te begrijpen wat een golfvorm precies is, vergelijken we een trilling van de lucht met golven in een vijfer, die zijn ontstaan ten gevolge van een in de vijfer gevallen steen. Deze golven hebben de vorm van een wiskundige sinus. Net zoals een

adres poked, begint de SID met het genereren van de toon. Het instellen van de golfvorm moet daarom als allerlaatste gebeuren. De toon blijft klinken totdat u de waarde-1 in adres 54276 poked. Op dat moment wordt de release ingezet en sterft de toon weg.

Als u de blok golf als golfvorm kiest, dan moet ook de pulsbreedte worden ingesteld. Deze pulsbreedte wordt weergegeven door een waarde tussen 0 en 4095. Deze waarde moet in de vorm van een lo- en hi-byte in de adressen 54274 en 54275 worden gepoked.



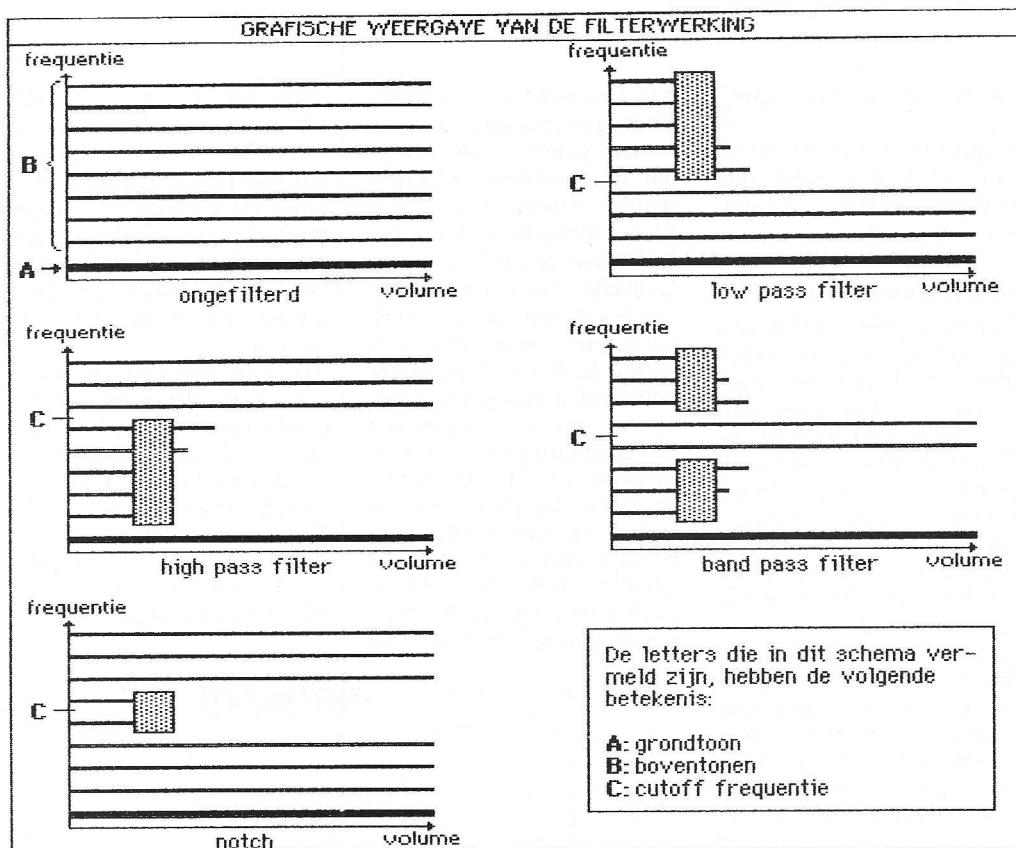
Figuur 3

watergolf een vorm heeft, zo heeft ook een geluidsgolf een vorm, de golfvorm.

We kunnen de golfvorm aan de SID doorgeven door een van de waarden die in figuur 4 tussen haakjes vermeld zijn, in adres 54276 te poken. Zodra u de waarde in dit

Volume

Tot slot moet nog het maximum volume van de toon worden ingesteld. Hiervoor dient u een waarde tussen 0 en 15 in adres 54296 te poken; hoe groter de waarde, hoe groter het volume.



Figuur 5

Met deze kennis zijn we nu in staat om een toon te produceren. Het volgende programma speelt een toonladder.

```

10 for i=54272 to 54296
20 poke i,0:next
30 poke 54296,15
40 poke 54277,2+8*16
50 poke 54278,11+10*16
60 for i=1 to 8
70 read fh,f1
80 poke 54272,f1:poke 54273,fh
90 poke 54276,17
100 for j=1 to 500:next
110 poke 54276,16
120 for j=1 to 500:next
130 next
200 data 8,180,9,196,10,247,11,158
210 data 13,10,14,162,16,10,9,17,103

```

In regels 10-20 worden alle adressen van de SID gewist, zodat eventuele oude waarde het programma niet kunnen storen (bijvoorbeeld een fil-

ter die aan staat). Regel 30 stelt het volume in. Regels 40-50 zorgen voor de juiste attack, decay, sustain en release. In regels 60-130 wordt de toonladder gespeeld. Eerst worden het lo- en hi-byte van de frequentie uit de data-statements gelezen. Deze worden dan in de adressen 54272 en 54273 gepoked. Vervolgens wordt in regel 90 de driehoeksgolf ingesteld. Het instellen van de golfvorm moet altijd als laatste gebeuren; zodra de golfvorm wordt ingesteld, begint de SID met het genereren van de toon. Regel 100 bevat een wacht-lus, zodat de toon enige tijd wordt aangehouden. In regel 110 wordt de release gestart. Regel 120 bevat een wacht-lus, zodat het wegsterven van de toon goed te horen is. Experimenteer u wat door de waarden voor de golfvorm en volumeverloop aan te passen. U kunt dan goed horen welk effect de vier golfvormen en het volumeverloop hebben op de klank van de toon.

Meerdere stemmen

Zoals eerder vermeld bevat de SID drie onafhankelijke geluidskanalen, stemmen genoemd, die elk een toon kunnen genereren. Het is dus mogelijk om tegelijkertijd drie tonen te horen. Elk van de drie stemmen bevat adressen voor het instellen van de frequentie, het volumeverloop en de golfvorm. Alleen het maximum volume is niet voor elke stem afzonderlijk in te stellen. Deze is voor alle stemmen gelijk. De adressen die we zojuist hebben behandeld zijn de adressen van stem 1. In figuur 4 staan de adressen van alle stemmen vermeld. Het genereren van een toon met stem 2 gaat op dezelfde wijze als met stem 1; u moet nu de adressen van stem 2 gebruiken i.p.v. de adressen van stem 1. Om drie verschillende tonen tegelijk te horen, moet u elk van de drie stemmen een andere toon laten genereren.

We hebben nu de basisprincipes van het maken van com-

putermuziek behandeld. We zullen nu kijken naar de meer geavanceerde mogelijkheden van de SID, zoals filters, ring modulatie, synchronisatie en modulatie.

Filters

Filters hebben invloed op het timbre van de toon. Zij kunnen een toon zwaarder en voller laten klinken of juist lichter. Om te begrijpen hoe een filter werkt, moeten we nog iets verder ingaan op de vraag wat een toon is.

U weet inmiddels dat een toon een golf is die met een bepaalde frequentie trilt. Deze beschrijving van een toon is niet compleet. Een toon bestaat namelijk niet uit een golf maar uit meerdere golven met verschillende frequenties. Er is een golf die de grondtoon wordt genoemd. De frequentie van deze golf, de grondfrequentie, bepaalt de frequentie van de hele toon. Alle andere golven worden boventonen genoemd. Deze boventonen hebben een frequentie die een veelvoud zijn van de grondfrequentie. De mate waarin de boventonen bijdragen aan de toon is bepalend voor het timbre van de toon. Door het volume van bepaalde boventonen te reduceren verandert het timbre van de toon. Dit is wat een filter doet.

De SID bevat drie filters: low pass filter, high pass filter en band pass filter. Elk van deze filters laten boventonen met een bepaalde frequentie door, de andere boventonen worden geblokkeerd. De mate waarin de boventonen geblokkeerd worden, wordt resonantie genoemd. Hoe groter de resonantie, hoe sterker het effect van de filter.

De low pass filter laat boventonen die onder een bepaalde frequentie liggen door. Deze frequentie wordt de cutoff frequentie genoemd. De low pass filter maakt een toon diep en zwaar.

De high pass filter laat boventonen boven de cutoff fre-



quentie door. Deze filter zorgt voor een heldere, lichte toon.

De bandpass filter laat boventonen door die rondom de cutoff frequentie liggen. De toon klinkt hierdoor dun.

Door de low en high pass filters te combineren is nog een vierde filter te maken. Deze filter wordt notch of band reject filter genoemd. De notch houdt de boventonen rondom de cutoff frequentie tegen. De toon die hierdoor ontstaat is nasaal van aard.

In figuur 5 is de werking van de filters grafisch weergegeven.

De filters kunnen worden ingesteld via de adressen 54293-54296. In figuur 6 zijn deze adressen beschreven. Zoals uit de figuur blijkt wordt de cutoff frequentie bepaald door een 11-bits waarde; deze waarde ligt dus tussen 0 en 2047. Deze waarde moet in een nogal bijzonder formaat in de adressen 54293 en 54294 worden gepoked. U doet dit als volgt:

```
HI=int(waarde/8):LO=waarde-8*HI
POKE 54293,LO:POKE
54294,HI
```

Uit de waarde kunt u op de volgende manier de cutoff frequentie berekenen:

```
cutoff =
(waarde*5.8)+30
```

Adres 54296 hebben we al eerder gebruikt voor het instellen van het volume. Hoe kan het dan dat dit adres ook weer gebruikt wordt voor de filters? De waarde voor het volume ligt altijd tussen 0 en 15. Dit is dus een 4-bits waarde en past dus in bits 0-3. De overige bits kunnen dus voor andere doeleinden worden gebruikt.

Voor het aan- en uitschakelen van de verschillende filters moeten losse bits aan en uit worden gezet. Normaal kunt u dit doen door met PEEK de waarde van het adres uit te lezen en vervolgens met AND en OR het

juiste bit aan of uit te zetten. Bij deze adressen kan dat echter niet, omdat de adressen write only zijn; u kunt ze dus niet uitlezen. De handigste manier om het probleem op te lossen is om de waarden die u in deze adressen poked ergens te bewaren, bijvoorbeeld in een variabele. U kunt deze variabele dan uitlezen om de juiste bits te zetten.

FILTERS	
adres	beschrijving
54293	cutoff frequentie (lo-part)
54294	cutoff frequentie (hi-part)
54295	bit 0: filter stem 1 (0=uit, 1=aan) bit 1: filter stem 2 (0=uit, 1=aan) bit 2: filter stem 3 (0=uit, 1=aan) bits 4-7: resonantie
54296	bit 4: low pass filter (0=uit, 1=aan) bit 5: band pass filter (0=uit, 1=aan) bit 6: high pass filter (0=uit, 1=aan)

Figuur 6

Complexe tonen

De vier golfvormen waarover de SID beschikt, zijn nogal eenvoudig van vorm. Golfvormen van tonen van akoestische instrumenten zijn veel complexer. Synthesizer muzikanten proberen om synthetische tonen complexer te maken, opdat het geluid rijker en voller wordt. Ring modulatie, synchronisatie en modulatie zijn drie technieken om meer complexiteit te verkrijgen.

Ring modulatie

Bij ring modulatie worden twee driehoek golven gecombineerd tot een nieuwe toon. Deze nieuwe toon heeft een metaalachtige, rinkelende klank. Ring modulatie is uitermate geschikt voor het simuleren van bellen en sirenes.

Ring modulatie berust op een eenvoudig principe. De frequenties van de twee oorspronkelijke tonen worden bij elkaar opgeteld en van elkaar afgetrokken om twee nieuwe frequenties te verkrijgen.

Deze twee frequenties worden dan hoorbaar gemaakt. Daarbij zijn ook allerlei tussenfrequenties zacht hoorbaar.

Voor ring modulatie zijn twee stemmen nodig; een stem voor het produceren van de toon en een voor de modulatie. Op de SID kunnen daarvoor de volgende stemcombinaties gebruikt worden: de toon van stem 1 kan gemoduleerd worden met

```
80 for j=0 to
100:next j,i
90 poke 54296,0
```

In regel 40 worden de frequentie en de sustain van stem 1 ingesteld. De attack, decay en release hebben de waarde 0. Regel 50 start de toon door de golfvorm in te stellen en zet tevens bit 2 aan voor ring modulatie. Er is nu een gewone toon hoorbaar. Door nu een frequentie voor stem 3 in te stellen, wordt het effect van ring modulatie hoorbaar. Dit gebeurt in de regels 60-80. Hierin wordt de frequentie van stem 3 steeds iets verhoogd.

Synchronisatie

Net zoals bij ring modulatie worden twee golven gecombineerd tot een nieuwe toon. De golfvorm van de golven is nu echter niet beperkt tot de driehoek. De klank van de toon die ontstaat door synchronisatie is voller dan de oorspronkelijke toon.

Bij synchronisatie worden de twee oorspronkelijke golven d.m.v. een logische AND met elkaar gecombineerd.

Voor synchronisatie zijn ook weer twee stemmen nodig; een voor de toon en een voor het synchroniseren. De mogelijke stemcombinaties zijn dezelfde als bij ring modulatie. Synchronisatie kan worden aangeschakeld door bit 1 van het golfvorm adres aan te zetten. Het effect van synchronisatie is weer alleen afhankelijk van de frequentie van de stem waarmee gesynchroniseerd wordt.

Door regel 50 uit het voorbeeld programma bij ring modulatie te veranderen in de volgende regel:

```
50 poke 54276,19
```

kunt u het effect van synchronisatie horen.

Modulatie

De derde techniek om tonen complexer te maken is door tijdens het klinken van een toon de frequentie, de puls-

stem 3, stem 2 met stem 1 en stem 3 met stem 2. Ring modulatie kan aangeschakeld worden door bit 2 van het adres dat dient voor het instellen van de golfvorm, aan te zetten. Om bijvoorbeeld de toon van stem 1 te moduleren, moet bit 2 van adres 54276 worden aangezet.

Het effect van ring modulatie is alleen afhankelijk van de frequentie van de stem die gebruikt wordt voor de modulatie; als u stem 1 moduleert, is dit dus stem 3. De frequentie adressen zijn dus de enige adressen die van belang zijn. Alle andere adressen van stem 3 hoeven niet te worden ingesteld.

Om dit alles te verduidelijken geven we een eenvoudig voorbeeldprogramma.

```
10 for i=54272 to
54296
20 poke i,0:next
30 poke 54296,15
40 poke
54273,50:poke
54278,15*16
50 poke 54276,21
60 for i=0 to 50
70 poke 54287,i
```




breedte (bij blok golf) of de cutoff frequentie (filters) te variëren. Deze techniek wordt modulatie genoemd. U zou bijvoorbeeld een programma op interrupt basis kunnen schrijven, dat continu de frequentie van de toon verandert volgens een bepaald patroon. Dit patroon moet voor een mooi effect de juiste vorm hebben en op de juiste snelheid de toon veranderen. Het kost zeer veel moeite om een dergelijk patroon te programmeren. De SID biedt echter twee hulpmiddelen bij het moduleren van een toon. Deze hulpmiddelen zijn de waveform follower en envelope follower. Een vereiste voor modulatie is dat de programma's in machinetaal worden geschreven. Een Basic programma is veel te traag om een toon te moduleren.

Waveform follower

De waveform follower stelt u in staat om, via adres 54299, het verloop van een golfvorm van stem 3 getalsmatig te volgen. Aan de hand van een voorbeeld wordt duidelijk wat hiermee bedoeld wordt. Stel dat we stem 3 hebben ingesteld op de zaagtand golfvorm. Op adres 54299 gebeurt nu het volgende: het adres krijgt de waarde 0. Deze waarde wordt steeds met 1 opgehoogd totdat de waarde 255 bereikt is. Vervolgens wordt de waarde weer 0 en het verhaal begint opnieuw. De snelheid van het ophogen van de waarden wordt bepaald door de frequentie van stem 3. De waarden verlopen dus volgens de grafiek van de zaagtand, zoals deze in figuur 3 is getekend. Het enige van stem 3 dat moet worden ingesteld zijn de frequentie en de golfvorm. Voor het instellen van de golfvorm kunt u de waarden uit figuur 3 met 1 verlagen. Deze verlaagde waarden zorgen ervoor dat de toon niet start; we willen namelijk alleen de golfvorm volgen,

maar de bijbehorende toon niet horen.

M.b.v. deze waarden kunt u nu de toon van een andere stem veranderen. U kunt bijvoorbeeld de waarden van adres 54299 optellen bij de frequentie van stem 1. Als u daarbij voor stem 3 een driehoek golfvorm kiest, dan zal de frequentie van stem 1 steeds op en neer gaan, waardoor een vibrerende toon ontstaat. Het vibreren kan versneld cq. vertraagd worden door de frequentie van stem 3 respectievelijk te vergroten of te verkleinen. De intensiteit van de vibrato kunt u vergroten of verkleinen door de waarde van de waveform follower eerst te vermenigvuldigen met of te delen door een vaste constante.

Op dezelfde wijze kunt u ook de pulsbreedte of de cutoff frequentie variëren.

Het volgende programma illustreert het vibrato effect. Voordat u het programma start, moet u alle adressen van de SID kennen.

LABELS MNEMONICS COMMENTAAR

```
lda #$0f ;Volume
sta $d418
lda #$07
;Frequentie
sta $d401 ;stem 1
lda #$f0 ;Sustain
sta $d406 ;stem 1
lda #$21 ;Zaagtand
sta $d404 ;stem 1
lda #$30
;Frequentie
sta $d40e ;stem 3
lda #$10 ;Driehoek
sta $d412 ;stem 3
loop lda $d41b ;Waveform
follower
sta $d400
;Moduleer frequentie
jmp loop
```

Envelope follower

De envelope follower maakt het mogelijk om, via adres 54300, het volume verloop van een toon te volgen. Net zoals de waarden van de waveform follower de grafiek van de golfvorm volgen, zo volgen de waarden van de envelope follower de grafiek van het volumeverloop (zie figuur 1).

U kunt nu echter niet volstaan met het instellen van het volume verloop. U moet

nu daadwerkelijk de toon starten door de onverlaagde waarden uit figuur 3 te gebruiken voor het instellen van de golfvorm. De attack wordt namelijk pas gestart op het moment dat de toon wordt gestart. Om er toch voor te zorgen dat de toon van stem 3 niet hoorbaar is, moet u bit 7 van adres 54296 aanzetten. Dit bit blokkeert de output van stem 3.

Door adres 54300 uit te lezen kunt u net zoals m.b.v. de waveform follower de frequentie, de pulsbreedte of de cutoff frequentie van stem 1 of 2 variëren.

We hebben nu in vogelvlucht alle mogelijkheden op muziekgebied behandeld. Daarbij zijn we niet zo diep ingegaan op programmeertechnieken om muziekprogramma's te maken. Problemen die bij het maken van muziekprogramma's een rol spelen, zijn de coördinatie tussen de drie stemmen, de opslagstructuur van een muziekstuk en de juiste timing. Met een redelijke kennis van programmeren en met dit artikel moet u echter in staat zijn om deze problemen op te lossen.

Veel succes met het componeren van uw computersymfonien.

Michel de Boer & Hylke Sprangers.



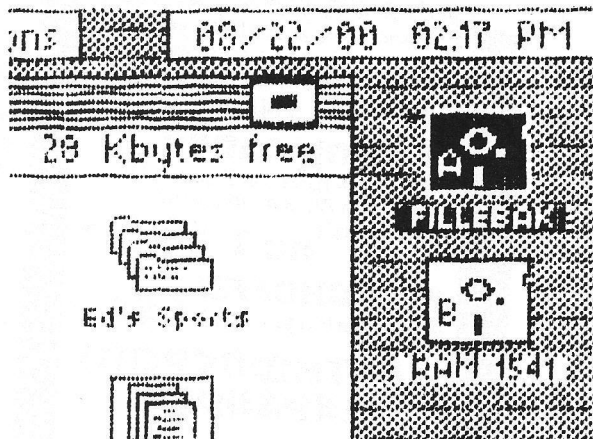
GEOS-INFO

Guillaume Kelders uit Turnhout begint de GEOS-INFO rubriek van deze maand met een forse brief vol vragen en problemen. Hij is in het bezit van een C-128 met diskdrive 1541, en gebruikt GEOS in de C-64 mode. De GEOS applicaties die hij gebruikt zijn GeoFile en GeoCalc. Kennelijk bijzonder intensief, want hij is er in geslaagd files te maken, die groter zijn dan 50K. En dan gaat het mis.

Bij het openen van zulke grote files krijgt hij steeds de melding "ERROR ACCESSING DATA DISK", en het document kan dus niet meer geopend worden. Alle GEOS applicaties vertonen naar mijn mening behoorlijk rare trekjes als het gaat om erg intensief gebruik (met grote files bijvoorbeeld). Dat kun je natuurlijk proberen goed te praten, door te zeggen dat de C-64 nu eenmaal niet bedoeld is voor zeer intensief (zakelijk) gebruik, maar naar blijft het zeker.

Disk-monitor

Vooralsnog ziet het er naar uit dat Guillaume zijn datafiles kwijt is. Wat ik zou proberen om ze toch nog te redden is het volgende schema. Benodigd daarvoor is een disk-monitor, deze hoeft niet perse onder GEOS te werken. Het idee is om allereerst twee kopieën van de data-file te maken, zeg "kopie1" en "kopie2". Zoek deze nu op met de disk-monitor in de directory. Deze begint op track 18 sector 1 (hexadecimaal \$12, \$01). De twee bytes vlak



voor de naam in de directory is het (track,sector) adres van de file op disk. In geval van GEOS datafile is dit eerste disk blok de Index Table.

Met enige moeite moeten dus de twee Index Tables van de twee kopieën op disk te vinden zijn. Een Index Table begint met op de derde byte met een opsomming van (track,sector) pointers die naar de losse records wijzen. Als een (track,sector) pointer de waarde (0,0) of (0,\$ff) heeft dan betekent dit, dat die pointer niet in gebruik is (het zijn illegale waardes). De eerste twee bytes van de Index Table dienen 0,255 te zijn. Wat gaan we doen? We gaan bij "kopie1" de eerste helft van de records behouden, en bij "kopie2" de tweede helft. Schrijf dus alle legale pointers in de Index Table van "kopie1" op. Dit zijn er zeg N, waarbij N een getal kleiner dan 128 is (meer pointer-paren passen er namelijk niet in een diskblok van 256 bytes). Maak nu de tweede helft van de pointers in de Index Table van "kopie1" (0,0), dit zijn dus de laatste 0.5N pointers. De

tweede helft van de pointers (die U nog ergens op papier hebt staan) moeten nu te beginnen bij de derde byte in de Index Table van "kopie2" ingevoegd worden. De rest van deze Index Table moet ook (0,0) gemaakt worden. Vervolgens moet de diskette vanuit de desktop gevalideerd worden. Als het goed is, bevat "kopie1" dan de eerste helft van de records, en "kopie2" de tweede helft. Beide files zijn dan plusminus 25K en kunnen weer gebruikt worden!

C-128

Guillaume vraagt zich bovendien af of er voor de C-128 ook nog andere programma's zijn die hetzelfde kunnen als GeoCalc en GeoFile. Het antwoord daarop is bevestigend: er zijn legio spreadsheets en databases voor de C-128 beschikbaar. Een kort bezoek aan de dichtbijzijnde softwarehandel zal dit bevestigen. Deze programma's werken echter niet met de user-vriendelijke gebruikersinterface. De laatste vraag van Guillaume is of er een printerdriver voor de Star

LC-10 is, waarmee in kleur geprint kan worden. Bij mijn weten bestaat er niet zo'n printerdriver.

Fontmaker

E. Van Der Toorn uit Den Haag vraagt meer informatie over het GEOS Fontmaker programma, zoals dat op de PD disk 1 is verschenen. Deze vraag is meen ik al eens aan de orde geweest, maar voor de volledigheid zal ik er nogmaals wat aandacht aan besteden. Wanneer men de font-maker start, komt men in het hoofdmenu, met 4 opties: 1) editen 2) save 3) laden en 4) conversie. De algemene werkwijze is, om eerst te beginnen met editen. Dan wordt gevraagd naar de breedte en hoogte van de letters, en men komt in een veld van bijv. 14x20 terecht. Bovenaan het scherm staat een letter, die aangeeft bij welk karakter het raster-veld hoort. Puntjes in het raster kunnen worden ingevuld door op de return toets te drukken. Het programma start bij het dan nog lege raster van het karakter "a". Met F1 gaat men naar de volgende letter, met F3 naar de vorige. F7 maakt het veld een puntje smaller, en F8 een puntje breder (maar niet breder als de aan het begin opgegeven maximale letterbreedte). Dit breder en smaller kunnen zijn van letters is een van de duidelijke kenmerken van GEOS (partial spacing). F5 gaat terug naar het hoofdmenu. Daar kunt U bijvoorbeeld het font-in-aanbouw op disk save. Het maken van een lettertype is een nogal groot karwei (256 karakters), dus U zult waarschijnlijk wel van deze optie gebruik maken. Let



wel, deze file met daarin het lettertype is niet in GEOS te gebruiken, die is alleen voor intern gebruik in de Font-maker. Met optie 3 (laden) kunt U, dat spreekt vanzelf, de zaak weer inladen, en verdergaan waar U gebleven was. De laatste optie in het hoofdmenu is conversie, aan te roepen als het lettertype klaar is. Het lettertype wordt dan op GEOS-maniër in een **USR** file gesaved. Deze file moet op de Geowrite diskette gezet worden, en kan dan gebruikt worden. Overigens is er een programma van Data Becker in omloop, dat geschreven is in machinetaal, en de eerlijkheid gebied het te zeggen, stukken beter werkt. Mijn fontmaker programma werkt ook wel, maar ik heb het al in 1988 in **BA-SIC** geschreven, beschouw het als een jeugdzonde.

Bouncing bars

Petro Kooy, een oude bekende van deze rubriek, wijst erop, dat in de listing van de Bouncing Bars iets mis is. Dit heb ik onderzocht, en het

is inderdaad waar: er zitten grote twee fouten en twee kleinere in. De kleine fouten zijn: achter `' .byte MIN_Y-1'`, in de eerste kolom van **LIST 1**, staat een komma die er niet hoort. Bovendien is in **LIST 2** het `' .seq'` statement vergeten voor de `' .psect'`. De belangrijke fouten zijn: ten eerste moet er in de routine `'rnd'`, dus tussen de labels `'rnd:'` en `'inc2_a0:'` steeds `r4L` door `a4L` en `r4H` door `a4H` vervangen worden. (Dit komt neer op 6 vervangingen.) In de tweede plaats moeten in de 6e, de 8e en de 10e regel onder `'draw_bar:'` steeds `'asl a5H'` door `'asl a5L'` vervangen worden. Petro klaagt ook over het feit dat in het begin van de listing, in de `'main:'`, de tekst nogal lastig te lezen is. Dit komt omdat het opmaakprogramma van dit blad (Ventura Publisher) de zinnen op nogal verwarrende plaatsen afkort. Daarom zal ik het stukje tussen `'jsr i_PutString'` en `'jsr i_FrameRectangle'` nogmaals afdrukken:

```
.word 220
.byte 8,24,"Bouncing Bars"
```

```
.byte 10,8,8,8,8,8,8,8
.byte 8,"-P.Boncz '90",0
```

Mocht de bovenstaande code ook weer zijn afgebroken, dan nog voor de duidelijkheid: De zinnen beginnen alleen met `' .byte'` of `' .word'`, en voor de rest moet U gewoon alles achter elkaar doortikken.

Configuratie

De heer Zwilling uit Amsterdam vraagt zich af of **GEOS 2.0**, **GeoCalc** en **GeoFile** werken met een computerconfiguratie bestaande uit een **C-64** computer met **Star-NL10** met serial impact dot matrix printerinterface. De grootste vraag is natuurlijk of het printerinterface de **Star-NL10** volledig **Commodore Compatible** maakt. Als dit zo is, d.w.z. als de interface ook grafisch compatible is met een **MPS801/803**, dan is het antwoord op de vraag zeker ja. Anders is het antwoord op de vraag: waarschijnlijk wel. Er bestaat een **Star-NL10** driver, en als die werkt kan waarschijnlijk de **Geos Gebruikers Groep** uitkomst bie-

den. Zij hebben veel ervaring met allerlei printerdrivers. Roeger Schindhelm uit Kerkrade vraagt toevalligerwijze ook naar het adres van deze groep. Hier is het dan:

stichting Geos Gebruikers Groep
Postbus 52
1300 AB Almere

Lidmaatschap kost **F35,-** en geeft ook recht op de nieuwsbrief. De groep verkoopt ook nederlandse **GEOS** boeken en handleidingen. Tevens zijn er uit de VS geïmporteerde **PD** schijven te krijgen, (met daarop programma's die overigens zeer de moeite waarde zijn). De **Geos Gebruikers Groep** houdt zich bezig met ondersteuning in de breedste zin van het woord, en zal ook in de toekomst deze rol voor de grote groep van enthousiaste **GEOS** gebruikers blijven vervullen.

Peter Boncz

AMIGA BUSWARE

Compleet assortiment Amiga PDS software voor **f 11,-** per schijf. Vraag nu een gratis catalogus aan of bestel voor **f 11,-** de speciale introductiediskette, namelijk:

de Amiga Busware Introschijf

Dit is een schijf uit het Busware assortiment, die we samen met een aantal Amiga specialisten hebben samengesteld. Daarop staat de volgende selectie:

Grafisch demopakket met workbench schermgrapjes, waaronder Wave Bench, Melt en Dropshadow
Gauge om te bepalen hoeveel geheugen er vrij is
Record Player (om zelf demonstraties te maken, alle muis en toetsbewegingen kunnen herhaald worden)
Helios Mouse (maakt venster, waar de muis is, actief)
Pins (grafisch programma)
Asteroids (ruimtespel)
DOS kwik Om meer op schijf te krijgen
Drunken Mouse (de muis gaat rare bewegingen maken op het scherm)
Backgammon spelprogramma
X-Icon Om programma's, die geen **ICON** hebben, toch te kunnen starten met **Icon**
Conman (CLI Editor)

bel: 020-273198 /02152-62343



DE BASIC INTERPRETER

Zelf instructies maken

Het hart van de Commodore 64 wordt gevormd door het Operating System, oftewel het besturingssysteem. Het besturingssysteem van de 64 bestaat uit de Basic interpreter en de kernal. De kernal zorgt voor de communicatie met de randapparatuur, het regelen van de diverse interrupts en nog veel meer. In dit artikel nemen we de Basic interpreter onder de loep. We zullen gaan kijken hoe basic programma's worden opgeslagen in het geheugen, hoe de interpreter zelf werkt. Aan het eind van dit artikel laten we dan zien hoe we zelf Basic instructies erbij kunnen maken, zoals bijvoorbeeld in Simon's Basic gebeurt. Een vereiste daarvoor is wel dat U in machinetaal kunt programmeren. Dit artikel zal echter makkelijk beginnen, zodat het begin door iedereen te lezen is.

Wat is een interpreter?

Er wordt vaak een onderscheid gemaakt tussen 'hogere' en 'lagere' programmeertalen. Een lage programmeertaal zit dicht bij de ma-

chine, zoals men dat noemt. Een voorbeeld hiervan is machinetaal of assembly code. Een hoge taal staat verder van de machine af, en kent wat meer gestructureerde instructies. Voorbeelden hiervan zijn Basic en Pascal. De programmeertaal C zou men als een tussenvorm kunnen beschouwen. U weet dat een machinetaal programma direct uit te voeren is. De computer kent deze taal. Een hogere taal als Basic, en ook C, kan echter niet gelijk uitgevoerd worden, omdat de computer deze taal niet 'kent' of 'snapt'. De instructies van zo'n taal moeten dan ook eerst vertaald worden naar machinetaal; immers, machinetaal 'kent' de computer wel. Er zijn nu twee mogelijkheden.

- Ten eerste kunnen we het geschreven programma in een keer in zijn geheel vertalen naar machinetaal en daarna de geproduceerde machinecode uitvoeren. In dit geval wordt het programma eerst gecompileerd, zoals dat heet, en daarna pas uitgevoerd.

- Als andere mogelijkheid kan het geschreven programma ook per instructie vertaald worden naar machinetaal, wat dan gelijk uitgevoerd wordt. De computer pakt in dit geval dus steeds

Token tabel instructies		
code	instructie	adres
128	end	\$a831
129	for	\$a742
130	next	\$ad1e
131	data	\$a8f8
132	input#	\$aba5
133	input	\$abbf
134	dim	\$b081
135	read	\$ac06
136	let	\$a9a5
137	goto	\$a8a0
138	run	\$a871
139	if	\$a928
140	restore	\$a81d
141	gosub	\$a883
142	return	\$a8d2
143	rem	\$a93b
144	stop	\$a82f
145	on	\$a94b
146	wait	\$b82d
147	load	\$e168
148	save	\$e156
149	verify	\$e165
150	def	\$b3b3
151	poke	\$b824
152	print#	\$aa80
153	print	\$aaa0
154	cont	\$a857
155	list	\$a69c
156	clr	\$a65e
157	cmd	\$aa86
158	sys	\$e12a
159	open	\$e1be
160	close	\$e1c7
161	get	\$ab7b
162	new	\$a642
163	tab(	
164	to	
165	fn	
166	spc(	
167	then	
168	not	
169	step	
170	+	\$b86a
171	-	\$b853
172	*	\$ba2b
173	/	\$bb12
174	^	\$bf7b
175	and	\$afe9
176	or	\$afe6
177	>	\$bfb4
178	=	\$aed4
179	<	\$b016

figuur 1a



Token tabel functies		
code	functie	adres
180	sgn	\$bc39
181	int	\$bccc
182	abs	\$bc58
183	usr	\$0310
184	fre	\$b37d
185	pos	\$b39e
186	sqr	\$bf71
187	rnd	\$e097
188	log	\$b9ea
189	exp	\$bfed
190	cos	\$e264
191	sin	\$e26b
192	tan	\$e2b4
193	atn	\$e30e
194	peek	\$b80d
195	len	\$b77c
196	str\$	\$b465
197	val	\$b7ad
198	asc	\$b78b
199	chr\$	\$b6ec
200	left\$	\$b700
201	right\$	\$b72c
202	mid\$	\$b737

figuur 1b

een instructie, vertaald deze naar machinetaal en voert dit uit. Vervolgens neemt de computer de volgende instructie enzovoort. We zeggen in dit geval dat de programmeertaal 'geïnterpreteerd' wordt, in plaats van gecompileerd.

De meeste talen (Pascal, C) worden eerst gecompileerd voordat het programma gedraaid wordt. Het Basic in de Commodore 64 wordt echter geïnterpreteerd. Voor het interpreteren van Basic, dus steeds een instructie pakken, vertalen en uitvoeren, dient de Basic interpreter. In deze aflevering gaan we kijken hoe deze werkt. Eerst gaan we echter kijken hoe een Basic programma in het geheugen wordt opgeslagen.

Opslaan van Basic

In de adressen 43 en 44 wordt het beginadres bijgehouden waar het Basic-programma begint. Als u dus een programma schrijft, of eventueel

laadt, wordt dit vanaf dit adres (dus het adres wat opgeslagen is in de adressen 43 en 44) in het geheugen gezet. Het beginadres is als volgt te berekenen:

```
print peek(43) +
      256*peek(44)
```

We zeggen ook wel dat het lo-byte in adres 43 is opgeslagen, en het hi-byte in adres 44. Samen vormen dit lo- en hi-byte het echte beginadres waar het Basic geheugen begint. Het eindadres van een programma wordt opgeslagen in de adressen 45 en 46. De berekening van dit eindadres gaat equivalent aan de berekening van het beginadres:

```
print peek(45) +
      256*peek(46)
```

We noemen adressen die zelf een adres bevatten pointers. Zo vormen de adressen 45 en 46 samen een pointer naar het eind van het Basic program-

ma dat op dat moment in de computer zit.

Het beginadres van een Basic programma is normaal 2049. Tenzij u dus de pointer naar het beginadres verandert (de adressen 43 en 44 dus), begint elk Basic programma op adres 2049 in het geheugen. We weten nu waar een Basic programma wordt opgeslagen in het geheugen. Nu moeten we nog weten hoe dat opslaan gebeurt.

Voor het begin van elk programma moet er een nul in het geheugen staan. In het normale geval moet er dus een nul in adres 2048 staan. Als deze nul er niet staat (poke er bijvoorbeeld maar eens een 1 in met poke 2048,1), geeft de computer een syntax error, als het programma gerund wordt. Verder moeten er aan het eind van elk programma twee nullen in het geheugen staan. Omdat er aan het eind van elke Basic regel ook een nul staat, zullen er in totaal drie nullen aan het eind van het programma in het geheugen staan. Hier komen we straks nog op terug. eerst gaan we kijken hoe de afzonderlijke regels in het geheugen gezet worden.

Opslaan van regels

Als u een nieuwe regel invoert vanaf het toetsenbord, wordt, nadat u op return heeft gedrukt, de hele regel in het geheugen gezet. De regel wordt echter niet precies zo in het geheugen gezet, als dat hij op het scherm staat. De regels worden namelijk gecodeerd opgeslagen. Ten eerste worden alle Basic instructies die in de regel staan gecodeerd. Dit houdt in dat er voor elke Basic instructie slechts een getalletje wordt opgeslagen. Dit getal noemen we de instructie code (ook wel token genoemd). Elke Basic instructie heeft zo een unieke instructie code. De print instructie heeft bijvoorbeeld code 153. In figuur 1 staat een tabel met de codes van alle instructies en func-

ties. De code van elke instructie is altijd groter dan 127. De eerste 128 getallen (0 - 127) worden gebruikt om gewone letters en cijfers op te slaan. Zo is het dus makkelijk voor de interpreter om gewone karakters en gecodeerde Basic instructies uit elkaar te houden. Immers, als het getal groter is dan 127, dan heeft de interpreter te maken met een Basic instructie, en anders niet. Het voordeel van het gecodeerd opslaan van instructies is dat het geheugenruimte bespaart en dat het sneller is.

Verder wordt het regelnummer niet opgeslagen in karakters. U zou misschien verwachten dat bijvoorbeeld bij het regelnummer 10 de 1 een de 0 allebij apart als een karakter (lees byte) worden opgeslagen. Dit is echter niet zo. De computer slaat alle regelnummers gelijk binair op. Bij het regelnummer 10 wordt dus het getal 10 in een byte gezet. Omdat op deze manier alleen regelnummers tot 256 mogelijk zijn, reserveert de computer voor elk regelnummer 2 bytes, een lo- en een hi-byte. Op deze manier zijn er regelnummers beschikbaar van 0 - 65535. In de praktijk zult u echter zien dat de computer regelnummers vanaf 64000 niet accepteert.

Als laatste slaat de computer voor elke regel nog het begin van de volgende regel op (ook in twee bytes). Dit vergemakkelijkt het zoeken tijdens de uitvoer van het programma. We zullen nu aan de hand van een voorbeeld de zaak nog even verduidelijken. Bekijk het volgende programma eens.

```
10 print "the artvark"
20 goto 10
```

Als met een monitor het geheugen wordt bekijken, kan het volgende gezien worden:

```
.0800 00 14 08 0a 00
      99 22 54
.0808 48 45 20 41 52 54 56 41
.0810 52 4b 22 00 1d
      08 14 00
```



.0818 89 20 31 30 00
00 00 53

Het programma begint op adres \$0801 (\$ staat voor hexadecimaal, \$0801 = 2049). Let op de nul op adres \$0800. Deze moet er dus verplicht staan. De eerste regel begint dus op \$0801. De eerste twee bytes wijzen naar het adres waar de volgende regel begint, in dit geval dus \$0814. Dit betekent dus dat regel 20 op adres \$0814 in het geheugen staat. In de volgende twee bytes is het regelnummer opgeslagen. Dit is hier dus 10 (= \$0a). Het volgende getal \$99 (=153) geeft de code van print aan. Daarna volgt de string "the artvark", die gewoon letterlijk in het geheugen staat (in ASCII). Let erop dat ook de aanhalingstekens in het geheugen staan (\$22). Aan het eind van de regel staat een nul in het geheugen, op adres \$0813. Op \$0814 begint dan de tweede regel. U kunt zelf proberen om deze te decoderen. Aan het eind van deze regel staat weer een nul (op adres \$081c). Daarna staan nog twee nullen die aangeven dat het programma afgelopen is. Samen staan er dus drie nullen aan het eind van een programma. Het programma loopt dus tot \$081f. Dit betekent dat de adressen 45 en 46 naar \$081f wijzen.

Het programma runnen

We gaan nu kijken hoe een Basic programma wordt uitgevoerd. Oftewel, hoe de Basic interpreter werkt. Als er een programma wordt gestart met RUN, worden de volgende dingen gedaan:

- Er wordt eerst een flag geset, die aangeeft dat we in de programma-mode zitten, en niet in de directe-mode. Deze flag bevindt zich op adres 157. Als hier een 0 instaat, zit de computer in de programma-mode. In de directe-mode staat er 128 in dit adres.

- Vervolgens wordt de CLR instructie uitgevoerd. Hier-

door worden alle variabelen, die er eventueel waren, gewist. De computer begint dus met een schone lei aan het uitvoeren van het nieuwe programma. Als u dit wilt vermijden, moet u het programma opstarten met de GOTO instructie, bijvoorbeeld met GOTO 10. Alle reeds bestaande variabelen blijven dan ook bestaan.

- Als er een regelnummer is meegegeven met RUN wordt nu begonnen met het uitvoeren van die regel. Als er geen regel is meegegeven, begint de computer gewoon bij de eerste regel die het programma telt.

Vervolgens wordt er steeds een instructie opgehaald en uitgevoerd. Dit werk wordt gedaan door de Basic interpreter. Hoe deze dit werk doet, staat in de volgende paragrafen.

Belangrijke adressen

Voordat we de Basic interpreter precies helemaal gaan bekijken, moeten er eerst een aantal belangrijke adressen in de computer worden uitgelegd. De adressen die hierna besproken worden, worden namelijk alle gebruikt door de Basic interpreter, om er bepaalde gegevens in op te slaan, of eruit te halen.

\$3d/\$3e:

In de adressen \$3d en \$3e worden respectievelijk het lo- en hi-byte opgeslagen van het adres waar de huidige Basic regel begint, dat de computer aan het uitvoeren is. Steeds als de computer begint met het uitvoeren van een nieuwe regel, wordt hier het adres waar die regel begint in het geheugen, opgeslagen. Als de gebruiker het CONT commando invoert na het afbreken van het programma, haalt de computer uit deze adressen het regelaadres weer op. Deze twee adressen bevatten dus als het ware een pointer naar het huidige statement, dat de computer aan het uitvoeren is of was. Als er in adres \$3e een 0

staat, wordt er op CONT geantwoord met de melding: "Can't continue error".

\$39/\$3a:

Deze twee adressen bevatten het lo- en hi-byte van het huidige regelnummer van de Basic regel die de computer aan het uitvoeren is. Als de computer bezig is met regel 10, staat dus het regelnummer 10 in deze adressen. Dit in tegenstelling tot de adressen \$3d/\$3e, waar dan het adres in zou staan, waar regel 10 begint in het geheugen.

\$7a/\$7b:

De adressen \$7a/\$7b bevatten een pointer die rechtstreeks ergens in het Basic programma wijst. De Basic interpreter gebruikt deze pointer om steeds een stukje Basic te lezen en uit te voeren. Deze pointer wijst gewoon naar een karakter. De computer kan dit karakter dan lezen, en vervolgens de pointer met 1 ophogen om het volgende karakter te lezen dat in het programma staat. Deze pointer wordt ook wel de TXTPTR (textpointer) genoemd. Deze naam zullen we verder in dit artikel aanhouden. Als het hi-byte van de TXTPTR op 02 staat, betekent dat, dat de computer in de direct-mode staat. De computer leest dan de instructies uit de Basic input buffer. Deze buffer bevindt zich op de adressen \$0200 tot \$0258.

\$73:

Op adres \$73 begint een de zogenaamde CHRGET routine. Als deze routine wordt uitgevoerd (met jsr \$0073), gebeurt er het volgende. Eerst wordt de TXTPTR (\$7a/\$7b) opgehoogd, en daarna wordt het karakter waar de TXTPTR naar wijst in de accumulator ingelezen. NB. Eerst wordt dus de TXTPTR verhoogd en daarna pas uitgelezen. De accumulator bevat dus het gelezen karakter. Het carry-bit is clear als het karakter een cijfer is (ASCII '0' tot '9'), en is

anders geset. Het zero-bit is geset als het karakter de ASCII waarde 0 heeft (statement terminator), of als het karakter de dubbele punt is ':'. Anders is het zero-bit clear. Als de CHRGET routine aangevoerd wordt op adres \$79, heet de routine CHRGET. De TXTPTR wordt dan niet eerst verhoogd, maar gelijk uitgelezen.

\$0308/\$0309:

Deze twee adressen bevatten een pointer die normaal naar \$a7e4 wijst. De routine op \$a7e4 maakt onderdeel uit van de Basic interpreter main-loop. Dit onderdeel zorgt voor de uitvoering van de volgende Basic instructie. We zullen straks zien dat de basic interpreter main-loop indirect via deze pointer naar \$a7e4 springt: jmp(\$0308). Door in deze pointer een ander adres in te voeren, kunnen we dus de Basic interpreter onderbreken.

De Basic interpreter

U heeft nu genoeg kennis om de werking van de Basic interpreter te snappen. We zullen hieronder de main-loop van de interpreter uitlijsten in assembly. Tussen de regels door wordt er commentaar bij het assembly gegeven.

```
a7ae jsr $a82c
;Op $a82c wordt
;getest of de
;stop-key
;is ingedrukt.
;Zoja, dan wordt
;daar
;het programma
;beeindigd.
a7b3 ldy $7b
a7b5 cpy #$02
;Zit de computer in
;de direct-mode?
a7b7 nop
a7b8 beq $a7be
;Zoja, spring.
a7ba sta $3d
a7bc sty $3e
;Zonee, store de
;TXTPTR.
a7be ldy #$00
a7c0 lda ($7a),y
;Check op einde
;Basic-regel,
;er staat dan een 0
;in het geheugen.
```



```

a7c2 bne $a807
;Geen einde regel,
;spring naar
;$a807.
;Op $a807 wordt
;gekeken of er een
;scheider ':' staat
;in het programma.
;Zonee, dan wordt
;er een error
;gegeven.
;Zoja, dan springt
;de computer naar
;$a7e1.
a7c4 ldy #$02
;Het eerste! byte
;van de nieuwe
;regel wordt
;overgeslagen.
;Immers,
;de TXTPTR staat op
;het laatste byte
;van de vorige
;regel.
a7c6 lda ($7a),y
;Check of er een 0
;in het tweede
;byte
;staat van de
;nieuwe regel
a7c8 clc
;Clc voor de
;routine op $a84b
a7c9 bne $a7ce
;Geen 0 in het
;tweede byte:
;spring.
a7cb jmp $a84b
;Wel een 0: het
;programma is
;afgelopen.
;$a84b termineert
;het programma.
a7ce iny
a7cf lda ($7a),y
a7d1 sta $39
;Sla het lo-byte
;van het huidige
;regelnummer op.
a7d3 iny
a7d4 lda ($7a),y
a7d6 sta $3a
;Sla het hi-byte
;van het huidige
;regelnummer op.
a7d8 tya
a7d9 adc $7a
a7db sta $7a
;Werk het lo-byte
;van de TXTPTR
;bij.
;De TXTPTR wordt op
;de eerste
;instructie gezet.
a7dd bcc $a7e1
a7df inc $7b
;Werk evt. het
;hi-byte van
;TXTPTR bij.
a7e1 jmp ($0308)
;Spring indirect
;naar de volgende

```

```

;regel ($a7e4)
a7e4 jsr $0073
;Haal de volgende
;instructie op.
a7e7 jsr $a7ed
;Voer de instructie
;uit.
a7ea jmp $a7ae
;Spring weer naar
;het begin van de
;Basic interpreter
;main-loop.

```

Hier volgen nog enige toevoegingen bij de main-loop van de basic interpreter. De main loop begint op adres \$a7ae. Deze loop leest steeds de volgende Basic instructie en voert deze daarna uit. Daarbij wordt de volgende cyclus steeds uitgevoerd:

- Test of de stop-key is ingedrukt. Zo ja, dan wordt het programma getermineerd.
- Kijk of het huidige statement afgelopen is. Zo ja, begin dan aan het volgende statement en update \$39/\$3a. Daarbij wordt ook gekeken of het programma afgelopen is, dus dat er geen volgende regel is.
- De TXTPTR wordt op de volgende instructie gezet. Deze instructie wordt vervolgens gelezen en uitgevoerd.

We hebben reeds opgemerkt dat er aan het eind van een programma drie nullen moeten staan. De eerste is om het einde van de laatste regel aan te geven, de andere twee om het einde van het programma aan te geven. Als u goed naar de bovenstaande main-loop heeft gekeken, zult u zien dat van de laatste twee nullen, alleen de laatste wordt gecheckd. Op de een na laatste nul wordt nooit gelet, dus dat hoeft in feite geen nul te zijn. Dat deze nul er wel altijd staat, komt omdat de computer deze zelf invult als er een programma wordt geschreven.

Zelf instructies maken

Zoals u aan de Basic interpreter main loop kunt zien, is deze aardig smerig geprogrammeerd. We zijn er dan ook niet zo zeker van, of u

deze sowieso begrijpt. Mocht u nog niet overtuigd zijn van het feit dat de Basic interpreter een grote bende is qua programmeerstijl, dan moet u zelf voor de grap zelf eens met een machinetaalmonitor de verscheidene subroutines bekijken, die in de main-loop worden aangeroepen: gratis gek worden, succes verzekerd. Opmerkelijk is het, dat de interpreter gemaakt is, met geheugenwinst in het achterhoofd, in tegenstelling tot snelheidswinst, wat meestal het geval is. We gaan nu kijken hoe we zelf Basic instructies kunnen maken. Daarvoor hoeft u niet precies te snappen hoe de Basic interpreter werkt.

De bekendste Basic uitbreiding is ongetwijfeld Simon's basic. Met dit programma, kan de instructie set van de Commodore een stuk worden uitgebreid. Het uitbreiden van de instructie set is in de praktijk niet zo moeilijk, als we nog even naar de interpreter main-loop kijken. Op adres \$a7e1 staat een indirecte jump via de pointer op \$0308. Deze pointer wijst normaal naar \$a7e4, dat is de instructie direct na \$a7e1. Op \$a7e1 wordt er dus indirect naar de volgende instructie gesproken, ogenschijnlijk een nutteloze gebeurtenis. Door echter de pointer op \$0308 te veranderen, kan de main-loop onderbroken worden. We kunnen bijvoorbeeld deze pointer naar \$c000 laten wijzen. Op \$c000 zetten we dan de uitbreiding van de instructie set, en aan het eind daarvan springen we naar de main-loop terug. De computer kan dan gewoon verder gaan, alsof er niets gebeurd is. We geven van dit principe een klein voorbeeldprogramma.

```

c000 lda #$0b
c002 sta $0308
;verander lo-byte
c005 lda #$c0
c007 sta $0309
;verander hi-byte
c00a rts
c00b ldy #$01

```

```

c00d lda #$41 ;ASCII
;'a'
c00f cmp ($7a),y
;vergelijk met
;TXTPTR
c011 beq $c016 ;
c013 jmp $a7e4
;verder met
;main-loop
c016 inc $d020
;verhoog border
;kleur
c019 lda #$02
c01b clc
c01c adc $7a
c01e sta $7a ;zet
;TXTPTR goed
c020 bcc $c024
c022 inc $7b
c024 jmp $a7ae ;naar
;begin main-loop

```

Het bovenstaande programma breidt de instructie set uit met de instructie 'a'. Als u een a intypt met een return, wordt de borderkleur met 1 verhoogd. Het programma moet gestart worden met SYS 49152. Op \$c000 wordt de pointer op \$0308 veranderd. Deze wijst dan naar \$c00b. Verder gebeurt er in eerste instantie niets. Als u een instructie intypt, wordt deze door de Basic interpreter behandeld. Omdat we \$0308 hebben veranderd, wordt de interpreter onderbroken, en wordt de routine op \$c00b doorlopen. Op \$c00b staat een programma dat kijkt of de instructie a is ingetypt.

Dit gebeurt door middel van de TXTPTR. Deze wijst immers naar de huidige instructie. Op \$c00f wordt de letter 'a' vergeleken met het karakter onder de TXTPTR. Daarbij moet u onthouden dat de TXTPTR bij de aanvang van een nieuwe instructie, altijd 1 karakter voor deze instructie staat (meestal op de ':'). Om het goede karakter te lezen moet de index y dus op 1 staan op \$c00f. Als de letter 'a' niet is ingetypt, wordt er verder gegaan met de main-loop. Als de letter 'a' wel is ingetypt, wordt de borderkleur verhoogd. Daarna wordt de TXTPTR aangepast. Deze moet nu met 2 worden verhoogd, in plaats van met 1! Dit omdat de



pointer altijd tussen twee instructies instaat. Er wordt daarna niet verder gegaan met de main-loop, want de instructie is al gevonden en uitgevoerd. Er wordt nu weer teruggesprongen naar het begin van de main-loop, dus naar \$a7ae.

Aan het begin van het artikel hebben we al gezegd dat elke ingebouwde instructie een instructie code (token) heeft. Nu zou het makkelijk zijn als we aan nieuwe instructies ook zo'n token konden toe-kennen. Dit is helaas niet mogelijk, omdat dan de dispatch tabel in het ROM moeten aanpassen. Er is echter geen ruimte in deze dispatch tabel overgelaten voor nieuwe instructies. De dispatchtabel bevat alle tokens met de daar-bijhorende instructies.

Voorbeeld

Als we dus langere instruc-ties willen maken, moeten we deze letter voor letter verge-lijken met de ingetypte in-structies. Hierbij kunnen we niet zomaar de TXTPTR gaan veranderen, want deze is eventueel later nog nodig in de main-loop. Alleen als er werkelijk een zelf gemaakte instructie is gevonden, moet de TXTPTR worden aange-past. Hieronder volgt een programma dat de instructies up en down toevoegt aan de instructie set. Het programma is makkelijk uitbreidbaar met andere instructies.

```
c000 lda #$0b
c002 sta $0308
      ;Verander lo-byte
c005 lda #$c0
c007 sta $0309
      ;Verander hi-byte
c00a rts
c00b ldx #$00
      ;plaats in tabel
      op $c100
c00d ldy #$01
      ;offset voor
      TXTPTR
c00f lda $c100,x
c012 beq $c02f
      ;einde van
      instructie?
c014 cmp ($7a),y
      ;vergelijk
      letters
```

```
c016 bne $c01f
      ;verkeerde
      instructie
c018 iny
c019 inx
c01a jmp $c00f
      ;volgende letter
c01d nop
c01e nop
c01f lda $c100,x
      ;zoek volgende
      instr.
c022 inx
c023 cmp #$ff
      ;laatste instr.
      gehad?
c025 beq $c047 ;ja
c027 cmp #$00
c029 bne $c01f
c02b jmp $c00d
      ;probeer
      volgende instr.
c02e nop
c02f tya ;instr.
      gevonden
c030 clc
c031 adc $7a
c033 sta $7a; zet
      TXTPTR goed
c035 bcc $c039
c037 inc $7b
      ;hi-byte van
      TXTPTR
c039 cpx #$02 ;up
      gevonden?
c03b bne $c040
c03d jmp $c04b
c040 cpx #$07 ;down
      gevonden
c042 bne $c047
c044 jmp $c052
c047 jmp $a7e4
      ;verder met
      main-loop
c04a nop
c04b inc $d020 ;up
c04e jmp $a7ae ;naar
      begin main-loop
c051 nop
c052 dec $d020 ;down
c055 jmp $a7ae ;naar
      begin main-loop

:c100 55 50 00 44 4f
      57 4e 00 ;up down
:c108 ff
```

Vanaf \$c100 zijn de instruc-ties opgeslagen in ASCII. Tussen de instructies moet er een 0 staan. Achter de laatste instructie moet eerst een nul staat en daarna \$ff.

Als er eenmaal een instructie gevonden is, wordt aan de hand van het x-register gekeken, welke instructie dit was: up of down. De rest van het programma spreekt min of meer voor zich.

Hulproutines

De volgende routines zitten standaard in de Commodore en zijn vaak handig bij het schrijven van uw eigen in-structies.

\$aefd:

Met deze routine kunt u een komma weglezen. Als voor-beeld geven we de instructie plot 100,20. De komma in het midden kunt u dan weglezen door \$aefd aan te roepen. Als de komma er dan niet staat wordt er een syntax error ge-given. Hoe de twee getallen in het voorbeeld kunnen wor-den gelezen, staat hieronder nog beschreven.

\$aefa:

Deze routine controleert op een haakje openen '('. De routine werkt analoog aan \$aefd en geeft dus ook een foutmelding als het haakje openen niet aanwezig is.

\$aef7:

Hiermee kunt u controleren of er een haakje sluiten ')' in de instructie staat.

\$a437:

Met behulp van \$a437 kan een foutmelding worden ge-given. Voor het aanroepen van deze routine moet het x-register het nummer van de gewenste foutmelding bevat-ten. Dit getal moet tussen de 1 en 30 liggen. In figuur 2 staat een tabel met getallen en de bijbehorende foutmel-dingen. Na aanroep van \$a437 wordt de foutmelding op het beeld geprint en stopt het programma met uitvoe-ren.

\$b79e:

Met deze routine kan een ge-tal-parameter worden ingele-zen. Het getal moet dan tus-sen de 0 en 255 liggen. Het gelezen getal wordt in het x-register gezet.

Als er op de huidige plaats in het Basic programma geen getal staat, wordt het pro-gramma gestopt met een foutmelding.

\$ad9e:

Deze routine leest een wille-keurige parameter in. We on-derscheiden nu de volgende gevallen.

- Als de parameter een string is, dan staat in de adressen \$22/\$23 het adres waar het eerste karakter van de string staat. de rest van de string staat in de opvolgende adres-sen. Aan het eind van de string staat een 0 in het geheu-gen.

- Als de parameter een getal is, staat dit getal in de Floa-ting Point Accumulator (FPAC). Met de volgende routines kan dit getal in de FPAC omgezet worden in een lo- en een hi-byte.

\$b1bf:

Zet een getal in de FPAC tus-sen de -32768 en 32767 om in een lo- en een hi-byte. Het hi-byte wordt in adres \$64 gezet, en het lo-byte in adres \$65. Let op, dit is precies an-dersom dan normaal.

\$b7f7:

Zet een getal in de FPAC tus-sen de 0 en 65535 om in een lo- en een hi-byte. het y-re-gister bevat na afloop het lo-byte, en de accumulator het hi-byte.

Tot zover deze aflevering over de Basic interpreter. Greetings...

Hylke Sprangers & Michel de Boer.



GRAPHICS

RECTIFICATIE

In de vorige aflevering van "Graphics op de 64" is het programma bouncing ball niet goed afgedrukt. Alle groter en kleiner dan tekens zijn in de listing weggevallen, zodat het programma niet meer werkte. In dit nummer plaatsen we de listing nogmaals. We hopen dat het plaatsen deze keer wel is gelukt. Onze excuses voor het ongemak.

```

1 rem *****
5 rem * bouncing ball *
6 rem *****
7 rem
10 for i=6400 to 6855
20 read a:poke i,a:c=c+a:next
25 if c<>50503 then print"fout in data":end
30 x=16384
40 a=-5:y0=0:x0=0:vx=3:vy=43:t=0:vs=1
50 yc=y0+vy*t+.5*a*t*t:xc=xc+vx
60 if yc<20 then gosub 100
70 if xc<0 then xc=0:vx=-vx
80 if xc>295 then xc=295:vx=-vx
90 yc=200-yc:gosub200:printxc,yc:t=t+.4:goto
  50
100 yc=20:vy=vy-vs:y0=20:t=0
110 if vy<6 then vs=-vs:ba=1
120 if ba=1 and vy=42 then 140
130 return
140 poke
  6768,x/256:poke6774,x-256*int(x/256):sys
  6678
200 by=8192+8*int(xc/8)+320*int(yc/8)+(yc
  and 7)
210 poke x,by-256*int(by/256):poke x+1,by/256
220 poke x+2,(xc and 7)/2:poke x+3,(yc and
  7):x=x+4:return
400 data 169,0,141,32,208,141,33,208
410 data 173,17,208,41,239,9,32,141
420 data 17,208,173,22,208,9,16,141
430 data 22,208,169,0,133,251,169,32
440 data 133,252,160,0,169,0,145,251
450 data 200,192,0,208,249,24,165,252
460 data 105,1,133,252,201,64,208,234
470 data 169,4,133,252,169,0,133,253
480 data 169,216,133,254,160,0,169,38
490 data 145,251,169,7,145,253,200,192
500 data 0,208,243,24,165,254,105,1
510 data 133,254,165,252,105,1,133,252
520 data 201,8,208,224,173,24,208,9
530 data 8,141,24,208,173,17,208,9
540 data 16,141,17,208,120,96,234,234
550 data 165,65,133,61,165,66,133,62
560 data 165,64,133,60,160,0,169,0
570 data 133,57,165,59,133,55,190,128
580 data 26,240,26,165,55,201,3,208
590 data 14,169,0,133,55,165,57,24
600 data 105,8,133,57,76,169,25,230
610 data 55,202,76,145,25,152,72,190
620 data 148,26,164,57,177,61,164,55
630 data 49,251,17,253,164,57,145,61
640 data 165,55,201,3,208,14,169,0
650 data 133,55,165,57,24,105,8,133
660 data 57,76,214,25,230,55,202,208
670 data 217,104,168,165,60,201,7,240
680 data 18,230,60,24,165,61,105,1
690 data 133,61,165,62,105,0,133,62
700 data 76,4,26,169,0,133,60,24
710 data 165,61,105,57,133,61,165,62
720 data 105,1,133,62,200,192,20,240
730 data 3,76,134,25,96,234,234,234
740 data 234,234,234,234,234,234,32,0
750 data 25,169,0,133,65,169,64,133
760 data 66,169,168,133,251,169,26,133
770 data 252,169,184,133,253,169,26,133
780 data 254,160,0,177,65,133,61,200
790 data 177,65,133,62,200,177,65,133
800 data 59,200,177,65,133,60,32,132
810 data 25,24,165,251,105,4,133,251
820 data 165,253,105,4,133,253,201,196
830 data 208,8,169,168,133,251,169,184
840 data 133,253,24,165,65,105,4,133
850 data 65,165,66,105,0,133,66,201
860 data 91,208,190,165,65,201,220,208
870 data 184,76,25,26,0,0,0,0
880 data 4,3,2,2,1,1,0,0
890 data 0,0,0,0,0,0,1,1
900 data 2,2,3,4,4,6,8,8
910 data 10,10,12,12,12,12,12,12
920 data 12,12,10,10,8,8,6,4
930 data 63,207,243,252,127,223,247,253
940 data 191,239,251,254,63,207,243,252
950 data 192,48,12,3,64,16,4,1
960 data 128,32,8,2,0,0,0,0

```

Beeldmanipulaties (deel 4)

Na de stevige stortvloed aan informatie van de vorige afleveringen doen we nu even een stapje terug. We gaan over op het scrollen. De meest bekende vorm van scrolling is 'smooth scrolling', dit is een vorm van scrolling die een tekst (of b.v. een plaatje) over het scherm laat bewegen zonder schokken of stooten. Buiten de interrupt kunnen we wel een scroll schokvrij te laten lopen maar kan daarnaast moeilijk nog iets anders gedaan worden. Wij gaan voor de scrolls een rasterinterrupts gebruiken.

Scroll Adressen

De VIC beschikt over enkele adressen waarmee het scherm horizontaal en vertikaal bewogen kan worden. Dit zijn de adressen \$D011 en \$D016. Deze adressen zijn al kort besproken in het overzicht van aflvering 2. Met behulp van de eerste 3 bits van \$D011 kan het scherm 8 posities omhoog of omlaag bewogen worden. Hetzelfde geldt voor de eerste 3 bits van \$D016, alleen is dit adres van toepassing op het bewegen in horizontale richting. Bij het smooth scrollen zullen we altijd alleen deze 3 bits gebruiken. Maar om ervoor te zorgen dat alles goed verloopt zetten we bij alle drie van de boven genoemde soorten van scrolling ook bit 4 van deze adressen op 0. Dit heeft bij \$D011 het gevolg dat het zichtbare scherm 1 regel korter wordt. Dit houdt in dat regel 25 nog wel steeds bestaat maar niet meer zichtbaar is. Het zelfde geldt voor

\$D016 maar hier wordt het scherm 2 karakters korter, er zijn dus nog maar 38 karakters zichtbaar in plaats van de gebruikelijke 40. Het nut hiervan zal later duidelijk worden.

Scroll Soorten

De drie meest bekende vormen van 'smooth scrolling' zijn:

De gewone 'tekst-scroll' waarmee de tekst van rechts naar links, smooth, over het scherm bewogen wordt.

Een 'omhoog-scroll' waarmee de tekst van beneden naar boven wordt bewogen. Als laatste is er dan nog een variant op de 'tekst-scroll', hierbij wordt een woord of een korte zin volgens een voorgeprogrammeerde cyclus over het scherm heen en weer bewogen.

Het principe van de scroll

Wat moet er nu eigenlijk gebeuren bij een scroll? Nou simpel gezegd moet er voor een tekst-scroll het volgende gebeuren: Verplaats een regel van het scherm 1 positie naar links. Dit heeft tot gevolg dat er 1 positie aan de rechterkant vrij komt (Tevens valt er aan de linker kant 1 karakter van het scherm). Op deze leeg gekomen positie plaatsen we een nieuwe letter uit de tekst. Als we dit achter elkaar door blijven doen hebben we een scroll. Het zelfde principe geldt voor de omhoog scroll, er moet dan alleen een aantal regels omhoog verplaatst worden en een hele nieuwe regel toegevoegd worden. De hierbeschreven methode werkt wel maar heeft tot ge-

```
10 INPUT "GEEF MIDDELPUNT:";A
20 INPUT "GEEF UITWIJKING:";B
30 INPUT "GEEF STAPGROOTTE (0.5
   AANGERADEN)";E
40 IF B>A THEN PRINT "UITWIJKING TE GROOT
   T.O.V. MIDDELPUNT":GOTO 110
50 IF B>128 THEN PRINT "UITWIJKING TE
   GROOT":GOTO 110
60 C=0
70 FOR D=0 TO 256 STEP E
80 Y=INT(A+B*SIN(D/10.19))
90 POKE 2816+C,Y
100 C=C+1:IF C<>256 THEN NEXT
110 END
```

volg dat alles nogal snel gaat en dus minder leesbaar is. Daarom maken we gebruik van de scroll adressen (\$D011 en \$D016). We kunnen aan de hand van deze adressen het scherm eerst bit voor bit of meerdere bits tegelijk laten verschijnen voor dat we de inhoud van het scherm zelf gaan verplaatsen. Uit het bovenstaande wordt ook duidelijk wat het nut van het 'verkleinen' van het scherm is. Als we de tekst nu verschuiven valt het leeg gekomen stuk op de positie die niet meer zichtbaar is. Je kunt nu dus een karakter op het scherm erbij plaatsen zonder dat hij plotseling te voorschijn klappt.

In praktijk is er nog een punt waar we rekening mee moeten houden. We mogen niet een stuk scherm verplaatsen als de computer toevallig net bezig is met het opbouwen van dat stuk scherm. Als we dit wel zouden doen krijg je schokken.

Creëren van bewegings patronen

Voordat we met het bespreken van een listing beginnen willen we nog even een ander

punt onder de aandacht brengen. In veel van de door ons geschreven routines zullen we steeds een gedeelte van het geheugen uitlezen voor een bepaald bewegings patroon. In de meeste gevallen zullen we gebruik maken van een sinus beweging. Om nu te voorkomen dat je steeds alle waarden van deze tabellen moeten over tikken hebben we in listing 1 een programma geschreven waarmee zo'n tabel vanuit BASIC gecreëerd kan worden. We raden je aan dit programma in te tikken en te bewaren, we zullen het nog vaker nodig hebben.

Als je het programma opstart vraagt het eerst om het middelpunt, de waarde die je hier verwacht wordt in te voeren zullen we altijd bij de assemblerlisting vermelden evenals de uitwijking en de stapgrootte.

De sinus die hiermee gemaakt kan worden, wordt neergezet op geheugenlokatie \$0B00. Met een machinetaalmonitor programma kan je je verschillende sinus- en wegschrijven als je dat handig zou vinden.



Listing 2

```

*= $1000
SINUS = $0B00
;MIDDELPUNT = 88
;UITWIJKING = 87
;STAPGROOTTE = 0.5
;
JSR $E544 ;CLEAR SCREEN, standaard
kernal routine
SEI
LDA #<IRQ
STA $0314
LDA #>IRQ
STA $0315
LDA #$32
STA $D012
LDA #$1B
STA $D011
LDX #$00
STX $DC0E
INX
STX $D01A

LDA #$17 ;Maak zeker dat de C64 de
kleine karakterset
STA $D018 ;gebruikt.

LDA #<TEXT ;Zorg er voor dat '($FE),y'
naar het
LDX #>TEXT ;het begin van de scroll
text wijst
STA $FE
STX $FF
LDA #$07
STA $FC ;Initialiseer $FC met begin
waarde
LDA #$01 ;SCROLL SPEED
STA $FD ;$FD bepaalt de snelheid
van de scroll

LDA #<SINUS ;'($50),y' wijst na deze
instructies naar
LDX #>SINUS ;de sinustabel in het
geheugen.
STA $50
STX $51

CLI
RTS
;
IRQ DEC $D019
LDA $52 ;Lees nieuwe $d016 waarde
voor de
STA $D016 ;sinus beweging
LDA $D012 ;Maak een wacht lus die 8
rasterlijnen
CLC ;duurt, immers een karakter
is 8 lijnen
ADC #$08 ;hoog.
CMP $D012
BNE *-3
LDA $FC ;Lees weer een nieuwe $d016
waarde
STA $D016 ;deze keer voor de gewone
scroll
LDA $D012 ;Wacht opnieuw 8 hele
rasterregels
CLC
ADC #$08
CMP $D012
BNE *-3
LDA #$08 ;En zet $d016 naar z'n
gewone waarde
STA $D016 ;dit voorkomt schudden van
het beeld
JSR SCROLL ;Scroll de tekst
JSR LR ;Maak de sinus beweging
JMP $EA31
;

```

```

SCROLL LDA $FC ;Laad de zero-page die we
voor $D016 gebruiken
SBC $FD ;Trek daar de
scroll-snelheid van af
BPL SCROLL2 ;Controleer of hij onder de
nul komt
CLC ;Als dat zo is moet er weer
8 bij opgetelt
ADC #$08 ;worden.
STA $FC
JMP SCHUIF ;En dan moet ook de tekst
verschoven worden.
SCROLL2 STA $FC ;Niet dan kan de nieuwe
waarde gewoon in $FC
RTS ;geplaatst worden.

SCHUIF LDX #$00 ;Verplaats de tekst op de
de tweede regel
LDA $0429,X ;van het scherm 1 karakter
naar links
STA $0428,X
INX
CPX #$27
BNE *-9

SCHUIF2 LDY #$00 ;Dit is nodig voor het
indirect uitlezen
LDA ($FE),Y ;Lees indirect het volgende
karakter v/d tekst
BNE SCHUIF3 ;Is het ongelijk nul ?
LDA #<TEXT ;Nee, dan moet de tekst
opnieuw beginnen
STA $FE ;dus plaatsen we het
begin-adres van de text
LDA #>TEXT ;weer in de pointer.
STA $FF
JMP SCHUIF2 ;En lees opnieuw een
karakter.
SCHUIF3 STA $044F ;Geen nul, plaats het
karakter op het scherm
INC $FE ;Verhoog de pointer naar de
tekst (Low-byte)
BNE *+4
INC $FF ;Wanneer nodig ook de
High-byte

RTS
;
TEXT .TEXT "WYTZE & ANSGAR"
.BYTE 0
;
LR LDY #0 ;Deze is weer nodig om
indirect
LDA ($50),Y ;de tabel te kunnen
uitlezen.
TAY ;Sla de waarde tijdelijk op.
AND #$07 ;Bereken de waarde voor
$D016
STA $52 ;plaats deze de $D016
' vervanger'.
TYA ;Haal de originele waarde
terug
CLC ;Deel deze door 8, dit wordt gedaan
LSR A ;door hem 3 keer door 2 te delen.
LSR A ;Immer LSR deelt feitelijk door 2.
LSR A
TAY ;Stop de verkregen waarde weer in Y
LDX #$00 ;En verplaats vervolgens de
in het
TER LDA TEXT2,X ;gehuegen staande tekst
naar het
STA $0400,Y ;scherm op positie $0400+Y.
INX
INX
CPX #$12
BNE TER
INC $50 ;En verhoog nog even de
tabel-pointer
RTS
TEXT2 .TEXT " BEELDMANIPULATIE "

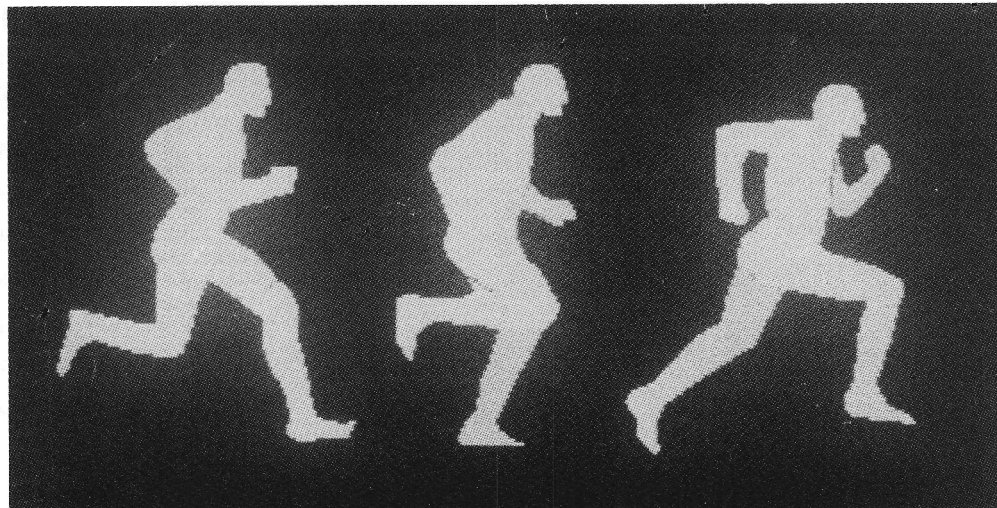
```

In de assemblerlisting van deze keer zitten twee van de eerder beschreven vormen van scrollen. We zullen nu aan de hand van listing 2 de verschillende principes uiteenzetten. Vergeet niet voordat je deze listing runt eerst een sinustabel in het geheugen aan te maken met de in de listing genoemde waarden.

Het Programma

Het programma valt in vier gedeeltes te onderscheiden. Het eerste gedeelte is het initialisatie gedeelte. Hierin initialiseren we een rasterinterrupt en er worden tevens enkele tekstpointers goedgezet. We gebruiken voor deze tekst pointers 'zero-page' adressen. Aan de hand van deze adressen kunnen we indirect tekst en bewegingen uit het geheugen lezen. Een voordeel van deze methode is dat het ophogen van het lees adres vrij simpel te realiseren is. Tevens word er nog een zero-page adres gebruikt om de snelheid van de scroll te bepalen. Je kunt hier zelf andere waarden in plaatsen (tussen de 1 en de 7, waarom?). In de initialisatie stomen we de computer dus klaar voor de overgang naar onze rasterinterrupt. Oh ja, bijna vergeten, in de initialisatie wordt ook nog de waarde #\$17 op adres \$D018 geplaatst. Dit heeft tot gevolg dat de commodore overschakelt op de kleine karakterset. Als we dit niet zouden doen krijg je vreemde tekens op het scherm (Druk na het starten van het programma maar eens op de SHIFT en COM-MODORE toets tegelijk).

Dankzij de initialisatie onstaat elke keer dat de computer rasterregel \$32 bereikt een IRQ-sigitaal. Op dat moment komen we dus binnen in onze IRQ-routine. We beginnen hier met het aanpassen van het scroll register, daartoe lezen we een zero-page adres uit en plaatsen de verkregen waarde in \$D016.



Vervolgens moeten we acht rasterregels wachten voordat we opnieuw \$D016 mogen veranderen. Immers een karakter is 8 regels hoog en als we dus eerder \$D016 zouden veranderen zou dat halverwege de karakters kunnen zijn en dan krijg je vreemde dingen. (De bovenkant van de tekst zou bijvoorbeeld 'smooth' bewegen terwijl de onderkant schokt) Na dit wachten lezen we opnieuw een zero-page adres uit en plaatsen de waarde opnieuw in \$D016. Deze keer dient de \$D016 voor de gewone scroll. De eerste \$D016 was voor de sinus beweging.

Vervolgens wachten we opnieuw 8 rasterregels en daarna zetten we de normale waarde in \$D016. Dit laatste voorkomt dat de rest van het scherm ook mee gaat bewegen. Je kunt hierdoor ook goed zien dat het scherm kleiner is op de plaats van de scrolls. Als laatste roepen we dan nog de twee subroutines aan die zorgen voor de eigenlijke scrolls.

De tekst scroll

Laten we de subroutines eens goed bekijken. We beginnen met de gewone scroll. Eerst lezen we het zero-page adres \$FC uit. Deze waarde weerspiegelt in feite de waarde van \$D016 die voor de scroll geldt. We verlagen de accumulator vervolgens met de

scroll-snelheid. Als de accumulator onder de nul uitkomt moet er weer 8 bij \$FC opgeteld worden omdat er dan immers al acht bits 'gescrollt' zijn en dat betekent dat er tekst moet worden verplaatst en dat er een nieuw karakter van de tekst bij moet worden geplaatst op het scherm. Als de accumulator niet onder nul komt kan deze gewoon weer terug geplaatst worden in \$FC en hoeft er nog niets te worden verplaatst.

Maar hoe gaat dat verplaatsen nou in zijn werk. Nou, daar zorgt het stuk na het label SCHUIF voor. Het scherm geheugen van C64 staat normaal gesproken vanaf de \$0400. Tik maar eens het volgende in de assembler in: ':0400 01', er zal dan boven in de hoek van het scherm de letter A verschijnen. Elke regel is 40 karakters lang wat betekent dat de volgende regel op adres \$0428 begint. We beginnen dus met alles wat na adres \$0428 tot aan het einde van de regel (\$044F) staat 1 karakter naar voren te verplaatsen. Hierdoor komt aan de rechter kant op adres \$044F een lege plaats. Vervolgens lezen we in de routine eerst een nieuw karakter van de tekst en we controleren of we niet toevallig aan het einde van de tekst zijn gekomen. Is dit wel het geval dan zetten we de tekstpointer weer naar

het begin van de tekst en lezen we opnieuw een letter. We plaatsen de aldus verkregen letter op de lege plaats. En dan moeten we nog even de tekst pointer ophogen want anders zouden we de volgende keer weer precies de zelfde letter op het scherm krijgen. Dit is alles wat er in principe bij een simpele tekst scroller moet gebeuren.

De sinus beweging

Dan volgt natuurlijk de andere scroll. Deze is ietwat minder simpel. Bij deze routine lezen we een waarde uit een tabel en vertalen deze vervolgens naar een positie op het scherm. Dit is dan de positie vanaf waar de tekst op het scherm zal worden geplaatst. Maar om alles 'smooth' te laten gaan moet ook de waarde van \$D016 uit de tabel-waarde worden berekend. Hoe kunnen we nu aan de hand van de tabel-waarde de positie en de \$D016 waarde berekenen. We moeten daartoe een bepaalde verdeling maken. Zoals we ondertussen weten kunnen we aan de hand van \$D016 8 bits scrollen. Daarna moeten we eerst een positie verplaatsen voor dat we \$D016 opnieuw gebruiken. Om dit makkelijk in tabel-vorm te krijgen, maken we de volgende verdeling: de waarde 0 houdt in dat er 0 bits gescrollt zijn en dat de tekst op positie 0 van het scherm begint (\$0400). De



waarde 1 betekent 1 bit scrollen maar de tekst begint nog steeds op positie 0. Bij waarde 8 worden er weer 0 bits gescrollt maar beginnen we de tekst pas op positie 1 (\$0401). Zo kunnen we dus steeds verder gaan. De waarde \$46 zou dus betekenen dat we 6 bits moeten scrollen aan de hand van \$D016 en dat we tevens pas met tekst moeten beginnen op positie 8 (\$46/8) van het scherm.

We beginnen de routine met het indirect uit lezen van de tabel. De zo verkregen waarde wordt tijdelijk in het y-register opgeslagen en de accu wordt met #\$07 ge-AND. De uitkomst hiervan wordt in \$52 geplaatst en hiermee hebben we de waarde voor \$D016 al berekent. Vervolgens maken we de accu gelijk aan het y-register om de originele waarde weer terug te

krijgen. Deze waarde moet nog door 8 gedeelt worden om de positie op het scherm te bepalen. De zo verkregen waarde plaatsen we weer in het y-register en vervolgens verplaatsen we de tekst vanuit het geheugen naar het scherm op de positie die wordt aangegeven met het y-register plus \$0400.

Als laatste moet dan nog even de pointer naar de tabel verhoogt worden. Omdat onze tabel precies 256 bytes lang is kunnen we de gewoon de hele tijd de low-byte van de pointer verhogen. Uiteindelijk is deze routine toch minder moeilijk dan hij lijkt, denken wij.

Als laatste in de IRQ routine roepen we de hier boven besproken routines aan. We doen dat op dit punt om dat we nu nog een heel scherm

aan raster-tijd beschikbaar hebben. Zouden we deze subroutines bijvoorbeeld als eerste uitvoeren dan gaat alles schokken. Want dan is de computer net bezig met het opbouwen van dat stukje scherm dat wij juist aan het veranderen zijn.

Om te kunnen zien hoeveel rastertijd er nou eigenlijk in beslag wordt door de scrolls kun je om de JSR's de border kleur veranderen. Zet b.v. voor de JSR's de instructie INC \$D020 en na de JSR's de instructie DEC \$D020 en start vervolgens het programma op. De knipperende balk geeft nu de gebruikte raster-tijd aan.

Dit is alles voor deze aflevering. We hebben nog 1 scroll soort niet besproken maar deze zal in de volgende aflevering nog aan de beurt ko-

men. Tot die tijd kunnen jullie experimenteren met de gegeven routines en zodoende de stof goed onder de knie zien te krijgen.

Wytze Westra & Ansgar Smith

NIEUW! - NIEUW! - NIEUW! - NIEUW! - NIEUW! Eindelijk uit: dé Nederlandse **GEOS 2.0-basishandleiding**

Dit boek is speciaal geschreven voor hen die werken met een Commodore 64/128 en het geavanceerde besturingssysteem GEOS 2.0. Het eerste Nederlandse boek voor de rechtgeaarde GEOS-gebruiker, die de mogelijkheden van dit pakket ten volle wil benutten.

Het boek beschrijft naast de werking van GEOS 2.0, uitvoerig het gebruik van de Desktop, GeoPaint, GeoWrite en alle desk-accessoires. Tevens komen applicaties als GeoSpell en GeoMerge aan de orde. Verder wordt

in dit boek aandacht besteed aan de andere GEOS-applicaties zoals GeoCalc, GeoFile, GeoPublish en GeoChart.

Indien u GEOS op de manier gaat gebruiken waarvoor het bedoeld is, zult u dit boek op zijn juiste waarde weten te schatten.

f 34,95

U kunt de GEOS 2.0 - basishandleiding bestellen door overmaking van het betreffende bedrag (incl. verzendkosten) op gironummer 5641219 t.n.v. SALASAN Amsterdam o.v.v. 'GEOS basishandleiding'. Dealer aanvragen welkom.



GEOS machinetaal (slot)

Het 3dVIEW programma

Deze aflevering is de slotaflevering van de cursus machinetaal programmeren met het GEOS besturingssysteem. Om de cursus een waardig einde te geven heb ik een goede duik in de C-64 genomen. Ondanks het uitvallen van mijn beide diskdrives (hoe kan het ook anders voor deze eigenlijk 13e aflevering?) staat het produkt van deze noeste arbeid bij dit artikel geLIST: het 3dVIEW programma.

3dVIEW

Met dit programma kan de gebruiker 3D-objecten ontwerpen, en deze laten roteren, uitprinten etc. De objecten zijn opgebouwd uit lijntjes. Er zijn twee soorten lijnen: rechte lijnen, en circelboogstukken (een moeilijk woord voor halve circels). De vlakken die de lijntjes vormen, worden natuurlijk niet ingekleurd: U beschikt over de rekenkracht van een C-64 en over niet die van een SPARC-station. Illustratie 1 is een screenshot van het uiteindelijke resultaat. Zoals U ziet heeft het programma een horizontaal menu met vier opties: 'exit', '3d', 'zoom' en 'data'. Hiervan is alleen 'exit' geen pulldown-menu. Deze optie beëindigt namelijk meteen het programma.

'Zoom'-menu

Deze menu-optie tovert een volgend menu tevoorschijn. U zult zien dat ook dit een

horizontaal menu is. Het bijzondere is, dat dit horizontale menu het hoofdmenu volledig afdekt. Een apart effect, niet alleen aantrekkelijk voor de verandering, maar ook omdat het geheugenruimte spaart. Dit programma gebruikt namelijk alleen de bovenste 40 pixels van het achtergrondscherf. De rest van het achtergrondscherf is in gebruik als data-gebied! Het 'zoom'-menu heeft twee opties: 'zoom in' en 'zoom out'. Zij zorgen ervoor dat er respectievelijk in en uitgezoomd wordt op het 3d-object dat op het scherm staat.

'3d'-menu

Dit menu geeft de keuze uit weer drie opties: 'read', 'write' en 'hardcopy'. De eerstgenoemde optie tekent de Dialog Box 'getDB' (in de listing 3dLst2 te vinden) op het scherm. Deze Dialog Box bevat het standaard DBGETFILES venster. DBGETFILES is DB commando 16, te vinden in de 3e regel van de getDB-tabel. Meer over Dialog Boxes is, zoals te zien is in illustratie 2, te vinden in aflevering 3 van deze cursus. Deze DBGETFILES dus, is een venster waarmee een naam geselecteerd kan worden van een van de files van een bepaald type die op disk aanwezig zijn. Read (te vinden in de listing 3dLst1) is namelijk de routine waarmee een datafile ingelezen kan worden. Zo'n datafile bevat, zoals U waarschijnlijk al snapte, een 3D-object, dat kennelijk al eerder door een gebruiker ontworpen en op

disk weggeschreven was. Dit wegschrijven op disk gebeurt met de tweede optie: 'write'. Ook deze routine maakt gebruik van een DialogBox. Deze DB, genaamd 'nameDB' en net als 'getDB' te vinden in 3dLst2, leest een string van het toetsenbord in. Deze string wordt dan vervolgens als naam gebruikt waaronder het te saven object op disk komt te staan.

Hardcopy

De derde en laatste optie van het '3d'-menu is hardcopy. Deze optie roept de routine 'hrdcpy' aan. Hrdcpy staat in 3dLst2, en heeft (nog) niet al te veel om het lijf. Hij bestaat alleen uit een jsr GotoFirstMenu en een jsr MainLoop. Wat is de bedoeling? Tussen deze twee jsr's moet een programma komen te staan dat een harcopy van het scherm maakt. Zo'n programma is al eens in deze cursus verschenen, en wel onder de naam 'hardcopy.prg' in aflevering 5. (zie illustratie 2) Het is een nogal lijvig programma en als ik het nu weer had afgedrukt was deze listing echt te groot geworden. Het idee is dus dat de 'InitRoutine' van het 'harcopy.prg' programma tussen de twee jsr's komt te staan. Let wel op, want de 'rts' waarmee de InitRoutine oorspronkelijk beëindigd werd, moet er dus niet bij. De ramsect variabelen van 'harcopy.prg' moeten gewoon bij die van het 3d-programma gestopt worden. Als U dit programma gaat intikken raad ik U wel aan om eerst deze listing kaal in te tikken,

te zorgen dat die werkt, en dan pas te proberen om ook de hardcopy optie aan de gang te krijgen.

'Data'-menu

Het laatste sub-menu, het 'data-menu', heeft 3 opties: 'ins line', 'ins arc' en 'del'. Met 'ins line' kan een rechte lijn aan het object toegevoegd lopen. Zoals hoogstwaarschijnlijk bekend is loopt een lijn tussen twee punten. Er dienen dus twee coördinaten in de driedimensionale ruimte als invoer ingegeven te worden. Deze coördinaten zijn steeds van de vorm (X,Y,Z). De X-as is in illustratie 1 de horizontale lijn, de Y-as is de schuine lijn, en de Z-as is de verticale lijn. Had U geen wiskunde in Uw lunchpakket, dan zal dit artikel sommige moeilijke punten bevatten. Zeker de rotatieformule, die nog volgt, is behoorlijk pittige stof. Maar goed, het invoeren van coördinaten moet toch nog geen probleem zijn. Dit invoeren gebeurt rechtsboven in het scherm. Eerst verschijnt er een haakje '(', en de cursor gaat aan. Dan moet een getal gevolgd door een return ingevoerd worden. Dit is de X coördinaat. Dit gedaan hebbende verschijnt er vanzelf een komma ',',. Vervolgens dienen de Y en Z coördinaat op soortgelijke wijze ingevoerd te worden. De computer sluit het geheel dan af met een sluithaak ')'. De X, Y en Z coördinaat tezamen vormen 1 coördinaat in de driedimensionale ruimte. Deze hele procedure wordt dus twee maal



uitgevoerd voor de beide coördinaten van de lijn.

In het geval van 'ins arc' moeten er drie 3d-coördinaten ingevoerd worden. Een arc is namelijk een boog en een boog heeft een begin, midden (de top), en een einde. De getallen die de coördinaten vormen mogen overigens positief en negatief gekozen worden in een bereik van -32768 tot 32767. Ze moeten natuurlijk wel geheel zijn. Het snijpunt van de drie assen is (0,0,0). Het is vanwege de schaal waarop het 3dVIEW programma bij opstarten ingesteld staat waarschijnlijk het beste om de getallen in de orde van grootte van honderden te kiezen. Dus (0,0,0) naar (800,100,0) levert een goed zichtbare lijn op. Natuurlijk kan de schaal met 'zoom in' en 'zoom out' wel naar believen aangepast worden.

Lijn-veld

De aldus ingevoerde rechte lijnen en bogen komen in het geheugen van de computer terecht. Deze lijst met lijnen is nog te bekijken tijdens dat het programma loopt. Het gedeelte rechtsboven in het scherm (waar ook de coördinaten ingevoerd worden) is hiervoor bestemd. Steeds wordt een veld van de lijnenlijst afgebeeld. Dit zichtbare veld (het zg. huidige veld) heeft een nummer, en bevat dus steeds een arc of een line. Er kan naar boven en beneden door de lijst gelopen worden m.b.v. de twee kleine pijlen rechts van het midden (zie illustratie 1). De derde menu-optie van het 'data'-menu: 'del', verwijdert het huidige veld uit de lijst. Overigens voegen 'ins line' en 'ins arc' allebei toe aan het einde van de lijst.

Rotate

Met de vier grote pijlen in het midden kan een rotatie in gang gezet worden. Er zijn twee soorten rotaties: om de X-as en om de Z-as. De ver-



tikaal gerichte pijlen verzorgen een rotatie om de X- as linksom dan wel rechtsom. De horizontale pijlen verzorgen de Z- as rotatie. Het vierkant in het midden is ook een icon, dit zet de hele beweging stop. Het is belangrijk om te begrijpen, dat het roteren weer geïmplementeerd is met een proces. Terwijl het proces aanstaat gaat het gewone 'leven' echter door: er kan met de muis bewogen worden, er kan een menu geselecteerd worden enzovoorts. Tot slot is er nog een icon onbesproken gebleven: het coördinaat- stelsel icon. Dit is het kleine frummeltje in illustratie 1, direct rechts van het menu. Met dit icon kan het coördinaat- stelsel (de drie lange rechte lijnen in illustratie 1) aan en uit worden gezet.

Rekenen

In illustratie 3 is de gang van zaken afgebeeld vanaf de ingetoetste coördinaat tot aan een punt op het beeldscherm. Dit is echter alleen van toepassing op rechte lijnen. Met bogen ligt het nog wat moeilijker. Direct na het intoetsen namelijk wordt een boog intern vertaald naar 8 losse lijnen. Van de buitenkant is van deze vertaling niets te merken. De hele hijs machinetaal vanaf label 'b0:' (in 3dLst1) tot label 'b7:' is hieraan gewijd. Met de aldus verkregen rechte lijnen wordt verder precies hetzelfde omgegaan als met de gewone rechte lijnen.

GetString

Interessant in het programma is ook het gebruik van de functie 'GetString'. Deze retourneert onmiddellijk, maar zet enige variabelen in de MainLoop om, die veroorzaken, dat zo gauw een getal is ingetoetst, naar de pointer in het adres keyVector gesprongen wordt. Parameters die vantevoren moeten worden gezet zijn: r11, r1, r2L en r0. De registers r11 en r1H bevatten resp. de X en Y coördinaat op het beeldscherm waar de cursor moet komen te staan. Register r0 moet naar de buffer wijzen waar de string in komt, r1L is doorgaans 0. De routine 'getco' in 3dLst1 maakt gebruik van GetString. De ingelezen string wordt met de routine 'string_to_dec' (ook 3dLst1) omgerekend naar een getal.

Bugs

Het ontwerpen van dit programma ging niet zo gladjes, dat kwam onder andere doordat er weer enkele bugs in de GeoAssembler bleken te zitten. In de eerste plaats moet men erg uitkijken met het plaatsen van teveel macro's achter elkaar, zonder dat er een label tussen zit. De local-label tabel van de GeoAssembler raakt dan overvol, en dan gaat het mis. Hier worden wel foutmeldingen opgegeven, maar meestal op de verkeerde pagina! Bovendien bleek de GeoAssembler bepaalde adresseringsmethoden niet te kennen, namelijk de direct,y (en de direct,x

voor enkele instructies). Omdat ik deze adresseringsmethoden toch nodig had, heb ik in de routine 'MyMult' (3dLst1) gebruik moeten maken van .byte- statements om ze toch te krijgen. In commentaarvorm staat er nu in de listing steeds voor elk .byte-statement welke instructie hij voorstelt.

Intikken

Om alles werkend te krijgen dient U maar liefst 4 files in te tikken. In de eerste plaats is dit '3dLnk', het kleine file-tje voor de GeoLinker. De hoofdlisting is vanwege de grootte opgeplitst in twee delen (de code met mijn DEBUG statements erbij was te groot voor de GeoAssembler), deze twee delen zijn '3dLst1', en '3dLst2'. Tot slot is er nog de macrofile '3dMacros'. Hierin moet U een aantal macro's uit de 'geosMac' file (die iedereen standaard heeft) overtikken. Niet alle macro's uit 'geosMac' overnemen, want dan zijn er in totaal te veel macro's en ook dat kan de GeoAssembler niet aan. Maar als U gewoon de aanwijzingen in de '3dMacros' file opvolgt, komt U er wel uit.

Het einde

Tja, en dit was hem dan, de machinetaalcursus GEOS. Als alles goed gaat zal aanvullende support voor een iets uitgebreidere versie van het 3dVIEW programma via de Geos Gebruikers Groep beschikbaar zijn. De gebruikersgroep blijft gewoon bestaan en misschien dat de GEOS artikelen nog op een ander podium een plaats kunnen vinden. Ik hoop dat U wat aan mijn cursus gehad heeft, hij heeft toch bijna alle faciliteiten van de GEOS Kernal behandeld. Een vriendelijke groet van mij en wie weet tot ziens.

Peter Boncz



```

;      3dLst1

.if Pass1
.include 3dMacros
.include geosSym
.endif
size    = 82
b_table = $6700
min_x   = 0
min_y   = 40
max_x   = 319
max_y   = 199

.ramsect
n_table:
.block 21*size

.psect
LoadB dispBufferOn,192
clrscrs 0,0,0,319,199
jsr i_PutString
.byte 6,0,32,"3dVIEW, by
P.Boncz '90",0
jsr i_FrameRectangle
.byte 2,min_y-1
.word min_x+2,max_x-2
.byte 255
LoadW r0,21*size
LoadW r1,n_table
jsr ClearRam
jsr i_FillRam
.word 54*size,b_table
.byte 255
LoadW r0,menu
jsr DoMenu
LoadW r0,icons
jsr DoIcons
jsr init_myproc
jmp MainLoop

p_table:
.word rotate,129
a_speed:.byte 0
b_speed:.byte 0
a_rot: .byte 0
b_rot: .byte 0
sin_a: .word 0
cos_a: .word 255
sin_b: .word 0
cos_b: .word 255
faktor: .word 20
stelsel:.byte 0
current:.word 0
komma: .byte ",",0
min: .byte "-",0
h_open: .byte "(",0
h_sluit:.byte ")",0
r11L_ex:.byte 0
r3_ex: .word 0
alarm: .byte 0
buffer: .word 0,0,
0,0,0,0,0,0,0,0

init_myproc:
LoadW r0,p_table

lda #1
jsr InitProcess
ldx #0
jmp RestartProc

rotate:
lda do_rotate
add #128
sta do_rotate
bne z1
z0:rts
z1:lda a_speed
bne z2
lda b_speed
beq z0
z2:cmp #2
bcc z3
jmp z6
z3:lda a_speed
beq z5
z4:Add24B 1,a_rot
jmp z8
z5:Add24B 1,b_rot
jmp z9
z6:lda a_speed
bne z7
Add24B 255,b_rot
jmp z9
z7:Add24B 255,a_rot
z8:MoveB a_rot,a9L
jsr sin
MoveW a9,sin_a
MoveB a_rot,a9L
Add24B 6,a9L
jsr sin
MoveW a9,cos_a
jmp draw_scr
z9:MoveB b_rot,a9L
jsr sin
MoveW a9,sin_b
MoveB b_rot,a9L
Add24B 6,a9L
jsr sin
MoveW a9,cos_b
jmp draw_scr
do_rotate: .byte 0

sin:LoadW a8,sin_table
asl a9L
LoadB a9H,0
AddW a9,a8
MoFrZ 0,2,a8,a9
rts

exit:
jmp EnterDeskTop

up:lda b_speed
cmp #1
beq faster
cmp #255
beq slower
LoadB a_speed,0
LoadB b_speed,1

jmp u1

down:
lda b_speed
cmp #1
beq slower
cmp #255
beq faster
LoadB a_speed,0
LoadB b_speed,255
jmp u1

faster:
CmpWI p_table+2,33
bne 1$
rts
1$:AddVW -32,p_table+2
jmp init_myproc

slower:
CmpWI p_table+2,641
bne 1$
rts
1$:AddVW 32,p_table+2
jmp init_myproc

right:
lda a_speed
cmp #1
beq faster
cmp #255
beq slower
LoadB a_speed,1
LoadB b_speed,0
jmp u1

left:
lda a_speed
cmp #1
beq slower
cmp #255
beq faster
LoadB a_speed,255
LoadB b_speed,0
jmp u1

stop:
LoadB a_speed,0
LoadB b_speed,0

u1:LoadW p_table+2,129
jsr init_myproc
uit:
jsr GotoFirstMenu
jmp MainLoop

zoom_in:
CmpWI faktor,4
beq weg
AddVW -4,faktor
weg:jsr draw_scr
jmp uit

zoom_out:

CmpWI faktor,160
beq weg
AddVW 4,faktor
jmp weg

do_stelsel:
lda stelsel
add #128
sta stelsel
jmp uit

read:
LoadB buffer,0
LoadW r7L,DATA
LoadW r5,buffer
LoadW r0,getDB
LoadW r10,0
LoadB dispBufferOn,128
jsr DoDlgBox
LoadB dispBufferOn,192
CmpBI r0L,CANCEL
beq 1$
LoadW r0,size*21
LoadW r1,n_table
jsr ClearRam
LoadW r6,buffer
LoadW r7,n_table
LoadB r0L,1
jsr GetFile
cpx #0
bne 1$
LoadW a_rot,0
LoadW sin_a,0
LoadW sin_b,0
LoadW cos_a,255
LoadW cos_b,255
LoadW current,0
jsr write_entry
1$:jsr init_table
jmp weg

write:
LoadW r0,nameDB
LoadW r5,buffer
LoadB buffer,0
LoadB dispBufferOn,128
jsr DoDlgBox
LoadB dispBufferOn,192
CmpBI r0L,CANCEL
bne 8$
jsr init_table
jmp weg
8$:LoadW file_hd,buffer
LoadW a0,n_table
ldy #0
0$:lda (a0),y
beq 1$
AddVW 21,a0
CmpWI a0,n_table+21*size
bne 0$
1$:AddVW 1,a0
MoveW a0,end_adr
MoveW a0,end_adr+2

```




```
LoadB r10L,0
2$:inc r10L
PushB r10L
LoadW r9,file_hd
jsr Save_File
PopB r10L
cpx #0
beq 3$
CmpBI r10L,7
bne 2$
3$:jsr init_table
jmp weg

init_table:
jsr i_FillRam
.word 54*size,b_table
.byte 255
LoadW a5,n_table
0$:ldy #0
lda (a5),y
beq 1$
sta a6L
MoFrZ 3,21,a0,r0
lsr a6L
PushW a5
jsr ins_b_entry
PopW a5
MoToZ 1,3,a5,entry
AddVW 21,a5
CmpWI a5,n_table+size*21
bne 0$
1$:rts

entry: .word 0
ins_b_entry:
php
LoadW a1,b_table
ldy #1
3$:lda (a1),y
cmp #255
beq 1$
AddVW 54,a1
bra 0$
1$:MoveW a1,entry
ldx #17
ldy #17
2$:lda r0,x
sta (a1),y
dex
dey
bpl 2$
plp
php
ldy #18
bcc 3$
ldy #12
3$:lda #0
sta (a1),y
iny
cpy #54
bne 3$
plp
bcc b0
rts
```

```
b0:MoveW entry,r0
MoveW entry,r1
AddVW 6,r0
AddVW 24,r1
jsr b7
AddVW 6,r0
AddVW 24,r1
jsr b7
MoveW entry,r4
MoFrZ 0,6,r4,r1
MoFrZ 12,18,r4,a7
LoadW r0,2
AddW r1,a7
ldy #r0
ldx #a7
jsr DSdiv
jsr Dnegate
AddW r2,a8
ldy #r0
ldx #a8
jsr DSdiv
jsr Dnegate
AddW r3,a9
ldy #r0
ldx #a9
jsr DSdiv
jsr Dnegate
b1:MoveW entry,r3
MoFrZ 12,18,r3,r0
MoveW a7,a3
MoveW a8,a4
MoveW a9,a5
AddW r0,a3
AddW r1,a4
AddW r2,a5
b2:MoFrZ 6,12,r3,r0
MoveW a7,a0
MoveW a8,a1
MoveW a9,a2
AddW r0,a0
AddW r1,a1
AddW r2,a2
ldx #a7
jsr Dnegate
ldx #a8
jsr Dnegate
ldx #a9
jsr Dnegate
b3:MoveW a3,r2
MoveW a4,r3
MoveW a5,r4
jsr halveer
MoveW r2,r5
MoveW r3,r6
MoveW r4,r7
jsr halveer
AddW r5,r2
AddW r6,r3
AddW r7,r4
MoveW entry,r0
AddVW 42,r0
jsr b8
jsr negatief
MoveW entry,r0
AddVW 6,r0
```

```
jsr b8
b4:MoveW a3,r2
MoveW a4,r3
MoveW a5,r4
jsr halveer
MoveW entry,r0
AddVW 36,r0
jsr b8
jsr negatief
MoveW entry,r0
AddVW 12,r0
jsr b8
jsr negatief
jsr halveer
MoveW entry,r0
AddVW 30,r0
jsr b8
jsr negatief
MoveW entry,r0
AddVW 18,r0
jsr b8
b5:LoadW r0,#169
MoveW entry,r6
MoveW entry,r5
AddVW 6,r5
AddVW 42,r6
jsr b9
LoadW r0,#222
MoveW entry,r6
MoveW entry,r5
AddVW 12,r5
AddVW 36,r6
jsr b9
b6:LoadW r0,#247
MoveW entry,r6
MoveW entry,r5
AddVW 18,r5
AddVW 30,r6
jmp b9
b7:ldy #5
1$:lda (r0),y
sta (r1),y
dey
bpl 1$
rts
b8:ldy #0
ldx #0
1$:lda a7,x
sta r9L
lda r2,x
add r9L
sta (r0),y
iny
inx
lda a7,x
sta r9L
lda r2,x
adc r9L
sta (r0),y
iny
inx
cpx #6
bne 1$
rts
b9:MoveW a0,a7
```

```
ldy #r0
ldx #a7
jsr MyMult
MoveW a1,a8
ldy #r0
ldx #a8
jsr MyMult
MoveW a2,a9
ldy #r0
ldx #a9
jsr MyMult
MoFrZ 0,6,r5,r2
MoveW r5,r0
jsr b8
MoFrZ 0,6,r6,r2
jmp b8

negatief:
ldx #r2
jsr Dnegate
ldx #r3
jsr Dnegate
ldx #r4
jmp Dnegate

halveer:
LoadW r11,2
ldy #r11
ldx #r2
jsr DSdiv
ldy #r11
ldx #r3
jsr DSdiv
ldy #r11
ldx #r4
jmp DSdiv

MyMult:
LoadB r2H,0
;lda $0001,y
.byte $b9,$01,$00
bpl 1$
lda #0
;sta $0001,y
.byte $99,$01,$00
sec
;sbcb $0000,y
.byte $f9,$00,$00
;sta $0000,y
.byte $99,$00,$00
inc r2H
1$: ;lda $0001,x
.byte $bd,$01,$00
bpl 2$
jsr Dnegate
inc r2H
2$: ;lda $0001,x
.byte $bd,$01,$00
sta r2L
lda #0
;sta $0001,x
.byte $9d,$01,$00
jsr BBMult
lda r2L
;sta $0000,x
.byte $9d,$00,$00
```



```

;lda $0001,x
.byte $bd,$01,$00
sta r2L
lda #0
;sta $0001,x
.byte $9d,$01,$00
jsr BBmult
lda r2L
clc
;adc $0000,x
.byte $7d,$00,$00
;sta $0000,x
.byte $9d,$00,$00
lda #0
;adc $0001,x
.byte $7d,$01,$00
;sta $0001,x
.byte $9d,$01,$00
lsr r2H
bcc 3$
jsr Dnegate
3$:rts

str_to_dec:
ldy #0
LoadW a1,0
LoadW a2,10
LoadB alarm,0
lda (r0),y
cmp #'-'
bne 1$
LoadB alarm,1
0$:iny
1$:lda (r0),y
beq 2$
pha
tya
pha
ldx #a1
ldy #a2
jsr DMult
pla
tay
pla
sub #'0'
add a1L
sta a1L
lda #0
adc a1H
sta a1H
bra 0$
2$:lda alarm
beq 3$
ldx #a1
jsr Dnegate
3$:rts

ins_arc:
jsr get_n_entry
lda #2
jsr ins_vars
LoadW return+1,c1
jmp get_co
c1:LoadB r1H,15
LoadW return+1,c2

```

```

jmp get_co
c2:LoadB r1H,25
LoadW return+1,c3
jmp get_co
c3:clc
jmp save_cos

ins_line:
jsr get_n_entry
lda #1
jsr ins_vars
LoadW return+1,l1
jmp get_co
l1:LoadB r1H,15
LoadW return+1,l2
jmp get_co
l2:lda #0
ldy a9L
sta (a0),y
iny
sta (a0),y
sec
jmp save_cos

ins_vars:
ldy #0
sta (a0),y
clrscr 0,172,3,315,16
clrscr 0,212,17,315,36
jsr GotoFirstMenu
LoadB a9L,3
LoadB r1H,5
ldx #0
jmp FreezeProcess

get_n_entry:
LoadW a0,n_table
ldy #0
1$:lda (a0),y
beq 2$
AddVW 21,a0
CmpWI a0,#n_table-
ble+21*size
bne 1$
pla
pla
jmp uit
2$:rts

get_co:
LoadW r11,215
AddVW 6,r1H
LoadW r0,h_open
jsr PutString
AddVW -6,r1H
LoadB a9H,0
LoadB r1L,0
LoadB r2L,16
g1:LoadW keyVector,g4
g2:LoadB buffer,0
LoadW r0,buffer
jsr GetString
jmp MainLoop
g3:LoadW r0,h_sluit

```

```

jmp g5
g4:LoadW r0,komma
g5:AddVW 6,r1H
jsr PutString
AddVW -6,r1H
LoadW r0,buffer
jsr str_to_dec
lda a1L
ldy a9L
sta (a0),y
iny
lda a1H
sta (a0),y
iny
sty a9L
inc a9H
LoadW keyVector,g3
lda a9H
cmp #2
beq g2
cmp #3
bne g1
return: jmp $1111

save_cos:
php
MoFrZ 3,21,a0,r0
plp
PushW a0
jsr ins_b_entry
PopW a0
MoToZ 1,3,a0,entry
LoadW keyVector,0
ldx #0
jsr UnFreezeProc
jmp MainLoop

del:
MoveW current,r0
LoadW r1,21
ldy #r1
ldx #r0
jsr BMult
AddVW n_table,r0
ldy #0
lda (r0),y
bne 5$
jmp uit
5$:MoFrZ 1,3,r0,r2
ldy #1
lda #255
sta (r2),y
lda #size-1
sub current
sta a9L
0$:cmp #0
beq 3$
MoveW r0,r1
AddVW 21,r1
1$:ldy #0
2$:lda (r1),y
sta (r0),y
iny
cpy #21
bne 2$

```

```

AddVW 21,r0
dec a9L
lda a9L
bra 0$
3$:ldy #0
4$:lda #0
sta (r0),y
iny
cpy #21
bne 4$
up_entry:
lda current
cmp #0
beq lb
dec current
lb:jsr write_entry
jmp uit

down_entry:
lda current
cmp #size-1
beq lb
inc current
MoveW current,a0
LoadW r0,#21
ldx #a0
ldy #r0
jsr DMult
AddVW n_table,a0
ldy #0
lda (a0),y
bne lb
dec current
jmp lb

write_entry:
clrscr 0,172,3,315,16
clrscr 0,210,17,315,36
MoveW current,r0
LoadW a0,21
ldx #a0
ldy #r0
jsr DMult
AddVW n_table,a0
LoadB r1H,12
LoadW r11,175
jsr PutKomma
LoadB a9L,3
ldy #0
lda (a0),y
beq 2$
cmp #2
php
beq 0$
LoadW r0,lineTxt
bne 1$
0$:LoadW r0,arcTxt
1$:jsr PutString
LoadB r1H,2
jsr write_co
jsr write_co
plp
bne 2$
jsr write_co
2$:rts

```



```
write_co:
  AddVW 10,r1H
  LoadB r1L,0
  LoadW r1L,215
  LoadW r0,h_open
  jsr PutString
  jsr get_r0
  jsr PutKomma
  jsr get_r0
  jsr PutKomma
  jsr get_r0
  lda #192
  jsr PutSigned
  LoadW r0,h_sluit
  jmp PutString

get_r0:
  ldy a9L
  lda (a0),y
  sta r0L
  iny
  lda (a0),y
  sta r0H
  iny
  sty a9L
  rts

PutKomma:
  lda #192
  jsr PutSigned
  LoadW r0,komma
  jmp PutString

PutSigned:
  pha
  lda r0H
  bpl 1$
  ldx #r0
  jsr Dnegate
  PushW r0
  LoadW r0,min
  jsr PutString
  PopW r0
1$:pla
  jmp PutDecimal

draw_scr:
  LoadB dispBufferOn,128
  LoadB d1+2,$a6
  ldy #25
  lda #0
  ldx #0
d1:  ;sta $a640,x
  .byte $9d,$40,$a6
  dex
  bne d1
  inc d1+2
  dey
  bne d1
  LoadW a0,n_table
  lda stelsel
  bne d2
  jsr draw_stelsel
d2:ldy #0
```

```
  lda (a0),y
  bne 0$
  jmp d3
0$:pha
  MoFrZ 1,3,a0,a2
  PushW a0
  jsr d4
  PopW a0
  pla
  cmp #1
  beq 2$
  LoadB a1H,7
1$:PushB a1H
  PushW a0
  jsr d5
  PopW a0
  PopB a1H
  dec a1H
  bne 1$
2$:AddVW 21,a0
  CmpWI a0,n_table+size*21
  beq d3
  jmp d2
d3:LoadB dispBufferOn,192
  rts
d4:LoadB alarm,0
  MoveW a2,a0
  jsr rot_n
  jsr n_to_s
  MoveW a0,a2
  AddVW 6,a2
  MoveW r3,r3_ex
  MoveB r11,r11_ex
  cpy #0
  beq d5
  LoadB alarm,1
d5:lsr a0H
  sec
  php
  MoveW a2,a0
  jsr rot_n
  jsr n_to_s
  MoveW a0,a2
  AddVW 6,a2
  MoveW r3_ex,r4
  MoveB r11L,r11L_ex
  MoveB r11L,r11L_ex
  MoveW r3,r3_ex
  ldx alarm
  LoadB alarm,0
  cpy #0
  beq d6
  inc alarm
  jmp d7
d6:cpx #0
  bne d7
  plp
  jmp DrawLine
d7:plp
  rts

draw_stelsel:
  LoadB r11L,174
```

```
  LoadB r11H,66
  LoadW r4,max_x
  LoadW r3,min_x
  sec
  jsr DrawLine
  LoadB r3L,min_y
  LoadB r3H,max_y
  LoadW r4,160
  lda #255
  jsr VerticalLine
  LoadW r3,min_x
  LoadW r4,max_x
  lda #255
  jmp HorizontalLine

rot_n:
  MoveFrZ 0,6,a0,a2
  MoveW a4,a7
  MoveW a3,a5
  MoveW sin_b,a6
  ldx #a5
  ldy #a6
  jsr MyMult
  MoveW cos_b,a9
  ldx #a3
  ldy #a9
  jsr MyMult
  MoveW sin_b,a6
  ldx #a7
  ldy #a6
  jsr MyMult
  MoveW cos_b,a9
  ldx #a4
  ldy #a9
  jsr MyMult
  SubW a7,a3
  AddW a5,a4
n6:MoveW a3,a7
  MoveW a2,a5
  MoveW sin_a,a6
  ldx #a5
  ldy #a6
  MoveW cos_a,a9
  ldx #a2
  ldy #a9
  jsr MyMult
  MoveW sin_a,a6
  ldx #a7
  ldy #a6
  jsr MyMult
  MoveW cos_a,a9
  ldx #a3
  ldy #a9
  jsr MyMult
  SubW a7,a2
  AddW a5,a3
  rts

n_to_s:
  MoveW a3,r1
  LoadW r3,#241
  ldy #r3
  ldx #r1
  jsr MyMult
  SubW r1,a2
```

```
  MoveW faktor,r1
  ldx #a2
  ldy #r1
  jsr DSdiv
  AddVW 160,a2
  LoadW r3,#86
  ldx #a3
  ldy #r3
  jsr MyMult
  SubW a4,a3
  MoveW faktor,r1
  ldy #r1
  ldx #a3
  jsr DSdiv
  AddVW 120,a3
  MoveB a3L,r11L
  MoveW a2,r3
  ldy #0
n0:jsm a3,min_y,n4
n1:jsm a2,min_x,n4
n2:jsm a2,max_x,n3
  jmp n4
n3:jsm a3,max_y,n5
n4:iny
n5:rts
```




```

; 3dMacros
;De volgende macros
;uit geosMac hier
;intikken:
;AddB,AddW,MoveW,
;MoveB,SubB,SubW,
;sub,add,PopB,PopW
;LoadB,CmpBI,CmpWI,
;LoadW,PushB,PushW
;
;en bovendien:

.macro AddVW value,dest
    clc
    lda #[value
    adc dest
    sta dest
    lda #[value
    adc dest+1
    sta dest+1
.endm

.macro jsr wrd,value,addr
    lda wrd+1
    cmp #0
    if (value>=0)&&
        (value<=32768)
        bmi jump
    else
        beq nothing
        bpl nothing
    endif
    cmp #[value

        bcs nothing
        jump:
        jmp addr
        nothing:
        .endm

        .macro Add24B value,dest
            clc
            lda dest
            adc #value
            sta dest
            bmi m_fout1
            cmp #24
            beq m_fout2
            bcc m_goed
            m_fout2:
            sec
            lda dest
            sbc #24
            sta dest
            sec
            bcs m_goed
            m_fout1:
            clc
            lda dest
            adc #24
            sta dest
            m_goed:
            .endm

            .macro clrscr co-
            lor,x1,y1,x2,y2
            lda #color
            jsr SetPattern

        jsr i_Rectangle
        .byte y1,y2
        .word x1,x2
        .endm

        .macro MoFrZ
        from,to,zp,addr
        ldy #from
        ldx #0
        macro1:
        lda (zp),y
        sta addr,x
        inx
        iny
        cpy #to
        bne macro1
        .endm

        .macro MoToZ
        from,to,zp,addr
        ldy #from
        ldx #0
        macro2:
        lda addr,x
        sta (zp),y
        inx
        iny
        cpy #to
        bne macro2
        .endm

; 3dLnk
.output 3dVIEW
.seq
.psect $0810
.ramsect $3000
3dLst1.rel
3dLst2.rel

```

GEOS

Public Domain disks

Disk #1

Deze diskette bevat o.a.:

- ◆ Geos program-maker
- ◆ **Geos fontmaker**
- ◆ **Geopaint-importer**
- ◆ **Geopaint-Catcher**
- ◆ **Geos Showladder**
- ◆ **QuickTop V1.2**

Disk #2

Deze diskette bevat o.a.:

- Iconchanger
- ◆ **Art-Importer**
- ◆ **Labyrint**
- ◆ **Kaart file**
- ◆ **Tabel file**
- ◆ **Tekst file**

Te bestellen:

door overmaking van f 12,- op giro3157656tnv Infolist Amsterdam,
o.v.v. GEOS INFO PD Disk 1 of 2

Let op !!

Programma's alleen te gebruiken in samenwerking met GEOS.



```
; 3dLst2
hrdcpy:
; jsr GotoFirstMenu
; optioneel:
; tik hier de code
; van hardcopy.prg
; in. (deel 5 v.d.
; cursus) Aan het
; einde van de
; start-routine wel
; de:
; jmp EnterDeskTop
; vervangen door:
; jmp MainLoop

menu:
; byte 5,21
; word 7,109
; byte 4
; word exitTxt
; byte 0
; word exit
; word drieDText
; byte $80
; word drieD_menu
; word zoomText
; byte $80
; word zoom_menu
; word entryText
; byte $80
; word entry_menu

drieD-menu:
; byte 5,21
; word 7,109
; byte 3
; word readTxt
; byte 0
; word read
; word writeTxt
; byte 0
; word write
; word hardcopyTxt
; byte 0
; word hrdcpy

zoom_menu:
; byte 5,21
; word 7,109
; byte 2
; word in_Txt
; byte 0
; word zoom_in
; word outTxt
; byte 0
; word zoom_out

entry_menu:
; byte 5,21
; word 7,109
; byte 3
; word i_line_Txt
; byte 0

; word ins_line
; word i_arc_Txt
; byte 0
; word ins_arc
; word delTxt
; byte 0
; word del

icons:
; byte 8
; word 100
; byte 12
; word up_icon
; byte 18,4,2,12
; word up
; word down_icon
; byte 18,26,2,12
; word down
; word left_icon
; byte 16,16,2,10
; word left
; word right_icon
; byte 20,16,2,10
; word right
; word stop_icon
; byte 18,17,2,8
; word stop
; word prev_icon
; byte 25,16,1,10
; word up_entry
; word next_icon
; byte 25,27,1,10
; word stels_icon
; byte 15,5,1,8
; word do_stelsel

up_icon:
; byte 128+24,1,128
; byte 7,224,31,248
; byte 63,252,63
; byte 252,3,192
; byte 3,192,1,128
; byte 1,128,1,128
; byte 3,192,3,192

down_icon:
; byte 128+24,3,192
; byte 3,192,1,128
; byte 1,128,1,128
; byte 3,192,3,192

left_icon:
; byte 128+20,12,0
; byte 28,0,60,0
; byte 127,135,255
; byte 255,255,255
; byte 127,135,60,0
; byte 28,0,12,0

right_icon:
; byte 128+20,0,48
; byte 0,56,0,60

; byte 225,254,255
; byte 255,255,255
; byte 225,254,0,60
; byte 0,56,0,48

prev_icon:
; byte 128+10,16,16
; byte 56,56,124
; byte 124,254,254
; byte 16,16

next_icon:
; byte 128+10,16,16
; byte 254,254,124
; byte 124,56,56,16
; byte 16

stels_icon:
; byte 128+8,8,9,10
; byte 12,127,24,40
; byte 72

nameDB:
; byte 1,42,137
; word 64,255
; byte 11,40,16
; word db1Txt
; byte 2,8,70
; byte 13,40,40
; byte $0c,16,0

getDB:
; byte 1,42,137
; word 64,255
; byte 16,6,5
; byte 11,136,16
; word db2Txt
; byte 2,17,50
; byte 5,17,70,0

file_hd:
; word buffer
; byte 3,21,128+63
; byte $ff,$ff,$ff
; byte $80,$00,$01
; byte $80,$00,$01
; byte $80,$ff,$01
; byte $80,$81,$81
; byte $81,$42,$41
; byte $81,$42,$21
; byte $82,$44,$41
; byte $82,$24,$41
; byte $84,$28,$81
; byte $84,$28,$81
; byte $88,$11,$01
; byte $8f,$f1,$01
; byte $80,$00,$01
; byte $87,$19,$19
; byte $84,$a5,$a5
; byte $84,$bd,$3d
; byte $84,$a5,$25
; byte $87,$24,$a5
; byte $80,$00,$01
; byte $ff,$ff,$ff
; byte 3,3,0

; word n_table
end_addr:
; word 0,0
; byte "3d datafile v1.0"
; byte 0,0,0,0
; byte "Peter Boncz",0
; en nog 40x .byte 0

sin_table:
; word 0,66,128,181
; word 221,247,255
; word 247,221,181
; word 128,66,0,-66
; word -128,-181
; word -221,-247
; word -255,-247
; word -221,-181
; word -128,-66,0

exitTxt:
; byte "exit",0
readTxt:
; byte "read",0
writeTxt:
; byte "write",0
hrdcpyTxt:
; byte "hardcopy",0
drieDText:
; byte "3d",0
zoomTxt:
; byte "zoom",0
entryTxt:
; byte "data",0
delTxt:
; byte "delete",0
i_arc_Txt:
; byte "arc",0
i_line_Txt:
; byte "line",0
arctTxt:
; byte "arc:",0
lineTxt:
; byte "line:",0
inTxt:
; byte "zoom in",0
outTxt:
; byte "zoom out",0
db1Txt:
; byte "please enter filena-
me",0
db2Txt:
; byte "select filename",0
```

Checksum C-64

Syntax Checksum

Het overtuiken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVet hem voordat u het programma RUNt op een diskette of cassette.

2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtuiken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...' komen dan is tot dusver alles goed. Het programma is nu in een stukje machine-taalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden (wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

De laatste tijd wordt er weer veel gebeld zodat U nogwel eens in gesprek krijgt, daarom houdt uw vraag kort, vermeld in welk blad het desbetreffende artikel stond. Heeft U veel vragen, of zis uw vraag erg uitgebreid, doe het dan schriftelijk, zodat we veel mensen op de maandag avond te woord kunnen staan.

```

1 rem *****
2 rem basic loader "SYNTAX.CHECKSUM"
3 rem na de commando's "run" en "new"
4 rem blijft dit programma in het ge-
5 rem heugen. laad het te testen pro-
6 rem gramma en tik daarna sys 49152.
7 rem *****
10 i=49152 :rem beginadres
20 reada:ifa<0then40:rem data ingelezen
30 pokei,a:i=i+1:b=b+a:goto20
40 if b<>16844thenprint"[SHIFT-CLR]fo
   ut [SPACE]in[SPACE]dataregels!":b=0:end
50 poke49184,148:poke49185,192
55 i=49300
60 read a: ifa<0then80
70 pokei,a:b=a+b:i=i+1:goto60
80 if b<>20068thenprint"[SHIFT-CLR]fo
   ut [SPACE]in[SPACE]dataregels! [SPAC
   E] (vanaf[SPACE]regel[SPACE]240)":b
   =0:end
90 print"data[SPACE]is[SPACE]weggezet"
95 print"checksum[SPACE]testen[SPACE]
   met [SPACE]sys49152"
100 data 165,43,166,44,133,163,134,164
    ,169,147
110 data 32,210,255,160,0,240,3,32,73,192
120 data 32,73,192,208,1,96,32,225,255,208
130 data 3,76,116,164,32,81,192,32,73,192
140 data 240,12,201,32,240,247,24,101,
    167,133
150 data 167,76,37,192,166,167,169,0,1
    32,168
160 data 32,205,189,169,13,32,210,255,
    164,168
170 data 76,17,192,200,208,2,230,164,1
    77,163
180 data 96,162,0,189,123,192,240,6,32,210
190 data 255,232,208,245,32,73,192,170
    ,32,73
200 data 192,132,168,32,205,189,162,3,
    169,32
210 data 32,210,255,202,208,250,169,0,
    133,167
220 data 164,168,96,82,69,71,69,76,32,0
230 data -1
240 data 165,197,201,3,240,7,201,4,240
250 data 6,76,148,192,76,34,192,169
260 data 147,32,210,255,76,161,192
270 data -1

```

** EINDE LISTING checksum 64 **

Checksum checksum 64

REGEL 1	249	REGEL 110	158
REGEL 2	84	REGEL 120	232
REGEL 3	105	REGEL 130	183
REGEL 4	2	REGEL 140	96
REGEL 5	246	REGEL 150	96
REGEL 6	152	REGEL 160	127
REGEL 7	249	REGEL 170	71
REGEL 10	157	REGEL 180	223
REGEL 20	64	REGEL 190	73
REGEL 30	38	REGEL 200	79
REGEL 40	57	REGEL 210	109
REGEL 50	14	REGEL 220	106
REGEL 55	251	REGEL 230	225
REGEL 60	192	REGEL 240	16
REGEL 70	42	REGEL 250	163
REGEL 80	244	REGEL 260	92
REGEL 90	245	REGEL 270	22
REGEL 95	237		
REGEL 100	183		

PRINT OUT C-64 met o.a. Editor

Character edit

Karakters blijven een aantrekkingskracht op de programmeurs hebben. Ook bij Wim de Wild uit Nigtevecht. Alle karakters zijn op alle mogelijke manieren te wijzigen en/of aan te passen. Het programma is volledig menu gestuurd. Let vooral eens op de bovenste regel op het scherm, alle karakters worden veranderd behalve bij de eerste regel.

```

10  rem *****
   **
20  rem *          character-editor
   *
30  rem *          _____
   *
40  rem *    programmed by wim de wild
   *
50  rem *****
   **
60  poke53281,12:poke53280,15:print"[S
   HIFT CLR][COM-8][CTRL-N]"
70  printtab(10)"please[SPACE]wait[SPA
   CE]....."
80  poke56334,0:poke1,peek(1)and251
90  fork=0to511:poke14336+k,peek(53248
   +k):next:poke1,peek(1)or4:poke5633
   4,1
100 print"[SHIFT-CLR][COM-8][CTRL-9][3
   xSPACE]character[SPACE]editor[SPAC
   E]by[SPACE]wim[SPACE]de[SPACE]wild
   [6xSPACE]"chr$(8)chr$(142)
110 fork=0to63:poke832+k,0:next:fork=0
   to7:poke832+k*3,255:next:v=53248
120 pokev+21,1:poke2040,13:pokev+39,11
   :pokev+27,1
130 fork=0to255:poke1704+k,k:poke55976
   +k,15:next
140 fork=0to39:poke1584+k,64:poke55856
   +k,1:next
150 t=0:fork=0to79:reada:poke49152+k,a
   :t=t+a:next
160 ift<>9164thenprint"data[SPACE]erro
   r[SPACE]!":end
170 sys49152
180 print"[HOME]":fork=0to7:print"[CTR
   L 9][COM-4][8xSPACE]":next
190 print"[HOME]":printtab(15)"char.nr
   .[5xSPACE][4xCRSR-LEFT]"nr
200 printtab(15)"n=next[SPACE]char"
210 printtab(15)"p=previous[SPACE]char
   "
220 printtab(15)"s=save"
230 printtab(15)"l=load[SPACE]"
240 printtab(15)"i=inverse[2xSPACE]c=c
   opy"
250 printtab(15)"w=walk[SPACE][PIJL LI
   NKS]=antiwalk"
260 printtab(15)"[home]=clear[SPACE]ch
   ar"
270 gosub630:a=55376+y*40+x:d=peek(a):
   pokea,160:gosub700
280 geta$:ifa$=""then280
290 pokea,d
300 ifa$="[CRSR-UP]"andx<>0theny=y-1
310 ifa$="[CRSR-DOWN]"andx<>7theny=y+1

```

```

320 ifa$="[CRSR-RIGHT]"andx<>7thenx=x+
   1
330 ifa$="[CRSR-LEFT]"andx<>0thenx=x-1
340 a=55376+y*40+x:d=peek(a):pokea,160
350 ifa$=chr$(13)thenpokea,15:d=15:pok
   eb+y,peek(b+y)or2[KWADRAAT PIJL](7
   -x)
360 ifa$="[SPACE]"thenpokea,11:d=11:po
   keb+y,peek(b+y)and255-2[KWADRAAT P
   IJL](7-x)
370 ifa$="n"andnr<>255thennr=nr+1:goto
   190
380 ifa$="p"andnr<>0thennr=nr-1:goto19
   0
390 ifa$="s"thengosub550:restore:goto1
   00
400 ifa$="l"thengosub600:restore:goto1
   00
410 ifa$="i"thenfork=0to1023:poke15360
   +k,255-peek(14336+k):next:goto280
420 ifa$<"c"then460
430 print"[HOME][CRSR-DOWN]copy[SPACE]
   to[SPACE](walk[SPACE]to[SPACE]nr)"
   :cr=nr:gosub720:fork=0to7
440 poke14336+nr*8+k,peek(14336+cr*8+k
   ):next:print"[HOME][CRSR-DOWN][21x
   SPACE]"
450 nr=cr:goto280
460 ifa$="[HOME]"thenfork=0to7:pokeb+k
   ,0:next:goto270
470 ifa$="w"thengosub720:goto190
480 goto280
490 rem ****    get filename    ****
500 poke214,10:print:input"enter[SPACE
   ]filename";f$
510 iflen(f$)>15then500
520 return
530 open15,8,15:input#15,a,b$:close15:
   ifb$="ok"thenreturn
540 print"[SHIFT-CLR]disk[SPACE]error[
   SPACE]:[SPACE]"a;b$:wait198,1:poke
   198,0:restore:goto100
550 rem ***** save option *****
   *
560 gosub490:open1,8,0,"`:"+f$+",p,w":
   close1:poke53265,0
570 sys57812"``:"+f$+",p,w",8,1
580 poke193,0:poke194,56:poke174,0:pok
   e175,64:sys62957:sys65511:poke5326
   5,27
590 return
600 rem ***** load option *****
   *
610 gosub490:open1,8,0,f$:get#1,z$,z$:
   gosub530:close1:ifasc(z$)<>56thenr
   eturn
620 poke53265,0:sys57812f$,8,1:poke780
   ,0:sys65493:sys65511:poke53265,27:
   return
630 b=14336+8*nr:c=55376:fork=0to7:n=p
   eek(b+k):forq=7to0step-1
640 z=sgn(nand(2[KWADRAAT PIJL]q)):pok
   ec+k*40+(7-q),11+z*4:nextq,k:retur
   n
650 data120,169,36,141,20,3,169,192,14
   1,21,3,169,127,141,13,220
660 data169,1,141,26,208,141,25,208,16
   9,130,141,18,208,169,27,141

```

- Print out 64 - Print out 64 - Print out 64 - Print out 64

```

670 data17,208,88,96,173,18,208,201,17
    0,176,18,169,21,141,24,208
680 data169,170,141,18,208,169,1,141,2
    5,208,76,49,234,169,31,141
690 data24,208,169,0,141,18,208,169,1,
    141,25,208,76,49,234,0
700 dy=int(nr/40):dx=24+(nr-40*dy)*8:p
    okev,dxand255:pokev+16,dx/256
710 pokev+1,dy*8+186:return
720 rem ***** walk command (w) *****
    **
730 geta$:ifa$=""then730
740 ifa$="[CRSR-RIGHT]"andnr<255thennr
    =nr+1
750 ifa$="[CRSR-LEFT]"andnr>0thennr=nr
    -1
760 ifa$="[CRSR-UP]"andnr>39thennr=nr-
    40
770 ifa$="[CRSR-DOWN]"andnr<240thennr=
    nr+40:ifnr>255thennr=255
780 gosub700:ifa$="[PIJL LINKS]"thenre
    turn
790 goto730

```

EINDE LISTING charedit

Checksum Character Editor

REGEL 10	207	REGEL 410	72
REGEL 20	100	REGEL 420	28
REGEL 30	179	REGEL 430	230
REGEL 40	18	REGEL 440	10
REGEL 50	207	REGEL 450	68
REGEL 60	225	REGEL 460	165
REGEL 70	13	REGEL 470	103
REGEL 80	177	REGEL 480	35
REGEL 90	47	REGEL 490	0
REGEL 100	177	REGEL 500	245
REGEL 110	7	REGEL 510	92
REGEL 120	224	REGEL 520	142
REGEL 130	27	REGEL 530	78
REGEL 140	216	REGEL 540	245
REGEL 150	167	REGEL 550	139
REGEL 160	26	REGEL 560	28
REGEL 170	163	REGEL 570	37
REGEL 180	47	REGEL 580	141
REGEL 190	103	REGEL 590	142
REGEL 200	247	REGEL 600	124
REGEL 210	55	REGEL 610	213
REGEL 220	206	REGEL 620	190
REGEL 230	184	REGEL 630	241
REGEL 240	108	REGEL 640	149
REGEL 250	201	REGEL 650	14
REGEL 260	178	REGEL 660	131
REGEL 270	41	REGEL 670	97
REGEL 280	103	REGEL 680	93
REGEL 290	72	REGEL 690	139
REGEL 300	250	REGEL 700	79
REGEL 310	128	REGEL 710	134
REGEL 320	137	REGEL 720	177
REGEL 330	3	REGEL 730	103
REGEL 340	107	REGEL 740	21
REGEL 350	122	REGEL 750	40
REGEL 360	128	REGEL 760	139
REGEL 370	84	REGEL 770	125
REGEL 380	235	REGEL 780	216
REGEL 390	33	REGEL 790	3
REGEL 400	22		

Commodore Public Domain Selectie

Infolist heeft nu, uit een wereldwijd aanbod aan Public Domain software voor de Commodore C-64 en Commodore C-128 een selectie samengesteld van utilities, spelletjes, en diverse korte programma's. Het gaat om Engelstalige Public Domain software van goede kwaliteit en zekere diepgang, enig inzicht in het gebruik van utilities is gewenst. Alleen leverbaar op 5,25" disks in verschillende uitvoeringen voor de C-64 en C-128.

Per set van 3 schijven f 26,-

- ◆ Selektie A-64 26,- incl. BTW en verzenden
- ◆ Selektie B-64 26,- incl. BTW en verzenden
- ◆ Selektie C-64 26,- incl. BTW en verzenden
- ◆ Selektie A-128 26,- incl. BTW en verzenden
- ◆ Selektie B-128 26,- incl. BTW en verzenden
- ◆ Selektie C-128 26,- incl. BTW en verzenden

Bij aankoop van 3 selekties (9 schijven) een speciale extra schijf gratis. In iedere selektie vindt u zowel utility software voor diskbeheer, printen, database, communicatie als spelletjes en korte algemene programma's.

Uitsluitend te bestellen door overmaken van f 26,- per selektie op gironummer 3157656 van Infolist, Amsterdam. Voor België: 480 Bfr overmaken op Bank BBL nr. 310050602562, t.n.v. SAC, Amsterdam.

Let op: Geef duidelijk aan om welke selektie en computer het gaat.-

***** Infolist Software *****

PRINT OUT Amiga met o.a.

Al de Amiga listings welke we op de volgende pagina's plaatsen zijn geschreven in Amiga Basic. U kunt ze intikken nadat U AmigaBasic heeft geladen.

TelephoneCalc

Mark de Jong gebruikt veel zijn Amiga en een modem om de verschillende vrienden en bekende BBS'en te bellen. Uiteraard moet er voor deze telefoontijd betaald worden. Het probleem ontstond door het scheiden van de kosten, wat zijn 'normale' telefoon kosten en wat zijn de modem-telefoon kosten. Uitrekenen kan, maar dit is erg tijdrovend, er werd een oplossing gevonden waarbij de Amiga zelf een grote plaats inneemt. Hij maakt altijd contact door middel van het communicatie programma JrComm. Dit programma wordt door de meeste modem gebruikers erg veel gebruikt. Dit programma maakt een LOG-file aan, in deze LOG-file staan alle dingen die er tijdens het gebruik van het communicatie-Programma gedaan zijn. Door deze LOG-file te bewerken kan je er de totale kosten, totale tijd, totale up/download, kosten per contact, enz. mee berekenen.

Dit is het basis-principe waar vanuit is gegaan bij het schrijven van het hulp programma TELEPHONE Calc. Het programma laadt de gehele LOG-file in het geheugen en gaat hem daarna bewerken. Om het geheel beter te kunnen lezen is er ook een print-optie toegevoegd, hierdoor kunnen de uitkomsten worden uitgeprint.

Er zijn een aantal beperkingen om dit programma te gebruiken:

- 1)Het Programma HEEFT een MEG AAN GEHEUGEN NODIG. (Dit is noodzakelijk omdat de gehele log-file in het geheugen wordt geladen om te worden bewerkt.)
- 2)De GEBRUIKER HEEFT een (ingebouwde) KLOK NODIG. (Deze klok zit op nagenoeg alle geheugen uitbreidingen die worden verkocht.)
- 3)De GEBRUIKER moet gebruik MAKEN van het communicatie-Programma JrComm (In ieder geval JrComm versie .93-.99N) (Dit programma is een van de weinige programma's welke een log-file aan maakt welke is te gebruiken.)

De werking van TELEPHONE CALC is erg eenvoudig en door iedereen te bedienen. In het hoofdmenu zijn de opties F1 tot F10 te kiezen. Als er een optie wordt gekozen die (nog) niet mogelijk is (bv. het listen van de telefoontjes) dan gebeurt er niets. Alleen als ze na het berekenen worden gekozen dan werken ze.

Als er wordt gekozen voor bewerken dan gaat de computer de gehele LOG-file inladen en hem daarna in 5 stappen doorwerken. Als dit gedaan is krijgt de GEBRUIKER het hoofd-MENU weer voor zich. Hij kan dan uit de SCAN CALLS, LIST CALLS en PRINT CALLS kiezen. SCAN CALLS is hetzelfde als de PRINT optie, alleen wordt het dan op het beeld geprint. De LIST optie om is elk telefoontje afzonderlijk te bekijken: de online-tijd, de begintijd, de eindtijd de up- en downloads in bytes en lokaal of interlokaal. Het is ook mogelijk om een einddatum en/of begindatum in te geven die de computer uit de LOG-file moet halen.

De GEBRUIKER kan zelf de LOG-file vastzetten in het Programma door in een van de eerste regels de lg\$="df0:JrComm/JrComm.log" te veranderen in elk gewenst bron waar de LOG-file zich kan bevinden. Ook in het programma zelf kan men de LOG-file-bron veranderen, alleen is dit niet vast en moet dit elke keer dat het programma wordt gestart het weer worden aangepast.

Dan nog iets erg belangrijks voor het goed werken van het programma:

In de eerste regels van het Programma staat een dataregel:

```
DATA 2,"BBS Leeuwarden","Jan Pietersen"
```

Door nu deze twee voorbeelden te veranderen voor BBS'en of personen waarmee je - LOKAAL belt is erg belangrijk!! Het aantal BBS'en en/of personen moeten in het totaal (bv. 2) vooraan In de dataregel komen te staan. b.v.

```
DATA 3,"Johan de Vries",  
"ABC-Softview","Dirk Jansen"
```

TELEPHONE CALC berekent de kosten per telefoontje met de plaats (LOKAAL/int.lokaal), met de tijd, met de online tijd (de tijd dat je contact hebt gehad) en of de gebelde dag in een weekend valt (dan is het namelijk goedkoper!) hier voor is er een kalender in gemaakt deze werkt van het jaar 1985 TO 1999.

In de subdirectory "tarief:" worden de kosten per telefoontje berekent. Als er een LOG-file ingeladen wordt die erg groot is en waar meer dan 300 keer in is gebeld, dan moeten de DIM's in de subdirectory "Dimmen:" van 300 naar b.v. 500 gebracht worden (Alleen de DIM's van 300).

Ik heb dit op dit moment met opzet niet gedaan om toch nog een beetje GEHEUGEN te sparen. Wordt dit gedaan dan is er een mogelijkheid dat de Lader ook meer GEHEUGEN vrij moet maken. Deze ERROR wordt dan wel kenbaar gemaakt. Ook als de LOG-file groter dan 30 K is dan moet Lader meer GEHEUGEN vrij maken.

- Print out Amiga - Print out Amiga - Print out Amiga -

Het programma(tje) loader met eerst worden ingetikt en op schijf worden weggeschreven. Dit programma moet dus eerst voor het hoofdprogramma (TelephoneCalc) ingeladen worden.

In het hoofdprogramma begint de beeldopbouw in de subdirectory "opbouw:" om de berekende DATA uit te printen moeten de benodigde devices, printers e.d. wel op schijf staan.

In ieder geval veel succes met TELEPHONE Calc.

Mark de Jong, de gelukkige bezitter van een Amiga 500(1 MEG+KLOK)0!

De lader voor telephoneCalc

```
CLEAR,125000;
LOAD"TELEPHONE CALC:TelephoneCalc",r:END
```

Deze twee regels dient U eerst in te tikken en daarna op de diskette te saven. Het maakt niet uit onder welke naam als U maar onthoudt dat het programma TelephoneCalc hiermee wordt geladen. Deze loader zorgt dat er eerst voldoende geheugen voor het eigenlijke programma wordt gereserveerd.

Hierna tikt U de volgende listing letterlijk over en dit schrijft U naar diskette onder de naam : TelephoneCalc

Deze naam is wel belangrijk want deze wordt in de loader gebruikt om het programma op te starten.

```
REM * Telephone Calc, Voor Het Berekenen Van
Je Telefoonkosten*
REM * m.b.v. de JrComm-LogFile*
REM * LET OP: TC ALLEEN VOOR JRCOMM
LOG-FILES! *
```

```
GOTO opbouw
ingeving:
DIM ing$(100)
lg$="DF0:JrComm/JrComm.log":RESTORE ing: READ
totaal:FOR p=1 TO totaal:READ
ing$(p):NEXT:tlg$=lg$:RETURN
```

```
ing:
REM * Vul de BBS'en in waarmee lokaal wordt
gebelt en verander na de dateregel *
REM * het getal in het aantal lokaal te
bellen BBS'en. *
REM * LET OP: precies zoals je ze in
Jrcomm-phonebook hebt staan *
```

```
DATA 2,"BBS Leeuwarden","Jan Pietersen"
```

```
lop:
notime=1:zb=1
```

```
klop:
OPEN "I",#1,lg$
WHILE NOT EOF(1)
a=a+1:LOCATE 8,7:PRINT"Reading
Log-data":LOCATE 8,40:PRINT a
```

```
INPUT #1, a$(a):WEND:CLOSE #1:da=a:mem$="IN
USE"
```

```
begin:
e=1:y=1:FOR a=1 TO da:LOCATE 10,7:PRINT"Sort
made calls":LOCATE 10,40:PRINT a
```

```
zz:
IF MID$(a$(a),21,9)="Connected" THEN
b=b+1:plts(b)=1
GOSUB als
caller$(b)=MID$(a$(a),34,LEN(a$(a))-33)
datumb$(b)=LEFT$(a$(a),8):Tijdb$(b)=MID$(a$(a),
10,8):con=con+1:u=1:pp=1
mnd$=LEFT$(a$(a),2):mnd(b)=VAL(mnd$):ddg$=MID$(
a$(a),4,2):ddg(b)=VAL(ddg$):jrg$=MID$(a$(a),7,2)
):jrg(b)=VAL(jrg$)
IF newold=1 THEN
IF mnd(b)<=old(2) AND jrg(b)<=old(3) AND
ddg(b)<old(1) THEN b=b-1:ver=ver+1:GOTO xp
IF onlyo=1 THEN xp
IF mnd(b)>=neww(2) AND jrg(b)>=neww(3) AND
ddg(b)>neww(1) THEN b=b-1:ver=ver+1:GOTO xp
END IF
GOTO dis
END IF
```

```
ss:
IF a>da THEN verder
```

```
xp:
NEXT a
```

```
dis:
IF MID$(a$(a+u),21,6)="Z u/1:" THEN
ub$=MID$(a$(a+u),40,8):ub=VAL(ub$):ukb(b)=ukb(b)
)+ub:up(b)=up(b)+1:up=up+1:u=u+1:GOTO dis
```

```
YYYY:
IF MID$(a$(a+u),21,6)="Z d/1:" THEN
db$=MID$(a$(a+u),40,8):db=VAL(db$):dkb(b)=dkb(b)
)+db:do(b)=do(b)+1:do=do+1:u=u+1:GOTO dis
IF MID$(a$(a+u),21,13)="Disconnected." THEN
tijde$(b)=MID$(a$(a+u),10,8):datume$(b)=LEFT$(a
$(a+u),8):disc=disc+1:u=1:pp=0:GOTO ss
IF MID$(a$(a+u),21,7)="Hang up" THEN
datume$(b)=LEFT$(a$(a+u),8):tijde$(b)=MID$(a$(a
+u),10,8):disc=disc+1:GOTO ss
u=u+1
IF (a+u)>b THEN verder
GOTO dis
```

```
verder:
FOR a=1 TO b:LOCATE 12,7:PRINT"Calculate
time":LOCATE 12,40:PRINT a
tupkb=tupkb+ukb(a):tdokb=tdokb+dkb(a)
eur$(1)=LEFT$(Tijdb$(a),2):emin(1)=MID$(Tijdb
$(a),4,2):esec(1)=RIGHT$(Tijdb$(a),2)
tuur$(2)=LEFT$(tijde$(a),2):tmin(2)=MID$(tijde
$(a),4,2):tsec(2)=RIGHT$(tijde$(a),2)
eur(a)=VAL(eur$(1)):emin(a)=VAL(emin(1)):ese
c(a)=VAL(esec(1))
tuur(a)=VAL(tuur$(2)):tmin(a)=VAL(tmin(2)):tse
c(a)=VAL(tsec(2))
ntijd(1)=(eur(a)*3600)+(emin(a)*60)+esec(a):nt
ijd(2)=(tuur(a)*3600)+(tmin(a)*60)+tsec(a)
ttijd(a)=ntijd(2)-ntijd(1):IF ttijd(a)<0 THEN
ttijd(a)=-ttijd(a)
NEXT
```

```
ber:
FOR a=1 TO b:gd=0:gg=0:dg=0:dd=0:LOCATE
14,7:PRINT"Calculate tariff":LOCATE
14,40:PRINT a
IF eur(a)<8 AND tuur(a)>=8 THEN gd=1:GOTO
oshit
```

- Print out Amiga - Print out Amiga - Print out Amiga -

```

IF uur(a)<8 AND tuur(a)<8 THEN gg=1:GOTO
doort
IF uur(a)>=18 AND tuur(a)<8 THEN gg=1:GOTO
doort
IF uur(a)>=18 AND tuur(a)<=24 THEN
gg=1:GOTO doort
IF uur(a)>24 AND tuur(a)<8 THEN gg=1:GOTO
doort
IF uur(a)<18 AND tuur(a)>=18 THEN dg=1:GOTO
shit
IF uur(a)>=8 AND tuur(a)<18 THEN dd=1:GOTO
doort

io:
NEXT:GOTO tijden

shit:
dg=0:j(1)=uur(a):j(2)=tuur(a):m(1)=emin(a):m(2)
)=tmin(a):sc(1)=esec(a):sc(2)=tsec(a):z=60-m(1)

IF j(1)<17 THEN nietgoed
lz=1:GOTO door

nietgoed:
ur=17-j(1):sg(a)=(ur*3600)+(60*z)+(60-sc(1))

aa:
x=m(2):IF j(2)>18 THEN ooknietgoed
lz=2:GOTO door

ooknietgoed:
lp=18-j(2):IF lp<0 THEN lp=-lp
sd(a)=(lp*3600)+(60*x)+sc(2):GOTO io

oshit:
gd=0:j(1)=uur(a):j(2)=tuur(a):m(1)=emin(a):m(2)
)=tmin(a):sc(1)=esec(a):sc(2)=tsec(a):f=60-m(1)

IF j(1)<7 THEN bhv
lz=3:GOTO door

bhv:
dp=7-j(1):sg(a)=(3600*dp)+(60*f)+(60-sc(1))

bb:
e=m(2):IF j(2)>8 THEN zvm
lz=4:GOTO door

zvm:
ll=8-j(2):sd(a)=(3600*ll)+(60*e)+sc(2):GOTO io

door:
IF lz=4 THEN sd(a)=(e*60)+sc(2):lz=0:GOTO io
IF lz=3 THEN
sg(a)=(f*60)+(60-sc(1)):lz=0:GOTO bb
IF lz=2 THEN sd(a)=(x*60)+sc(2):lz=0:GOTO io
IF lz=1 THEN
sg(a)=(z*60)+(60-sc(1)):lz=0:GOTO aa

doort:
j(1)=uur(a):j(2)=tuur(a):m(1)=emin(a):m(2)=tmi
n(a):sc(1)=esec(a):sc(2)=tsec(a)
IF gg=1 THEN sg(a)=(ttijd(a)):sd(a)=0:GOTO io
IF dd=1 THEN sg(a)=0:sd(a)=(ttijd(a)):GOTO io
GOTO io

tijden:
FOR a=1 TO b:LOCATE 16,7:PRINT"Making
Calander":LOCATE 16,40:PRINT
a:jrg(a)=jrg(a)+1900
jaar=1984:dg=1:x=0:schr=0:feb=0:i=0:p=0:dag=0:
k=0:fk=0:v=0:z=1:o=0:'We beginnen bij 01-01-84
IF jrg(a)=2000 THEN pp

aa:
d=jrg(a)/4:IF d=INT(jrg(a)/4) OR
d=INT(-jrg(a)/4) THEN schr=2:GOSUB ui
' IF a=4 THEN STOP

pp:
v=jrg(a)-jaar

a:
GOSUB ui:p=v+1+z:IF p>7 THEN b
dag=p

b:
g=p-(x*7):IF g>7 THEN x=x+1:GOTO b

l:
dag=g:IF jrg(a)=2000 THEN dag=g+1
GOTO zienn

ui:
k=jrg(a)-jaar

gg:
k=k-4:IF k=0 THEN z=z-1:RETURN
IF k<4 THEN z=z+1:RETURN
IF k>=4 THEN z=z+1:GOTO gg

oo:
IF i<(7-dag+2) THEN i=i+(dag-1):PRINT i"Deze
datum klopt niet!":STOP

nadenken:
i=i-7:IF i>7 THEN fk=fk+1:GOTO nadenken

kll:
IF schr=2 THEN feb=30:
IF schr=0 THEN feb=28:
IF rq=1 THEN rq=0:RETURN

ie:
i=i+(dag-1):IF i>7 THEN denken
GOTO rst

denken:
i=i-7:IF i>7 THEN denken

rst:
RESTORE days:FOR zg=1 TO i:READ
wkd$(a):NEXT:IF wkd$(a)="Saturday " OR
wkd$="Sunday " THEN zazo(a)=2
NEXT:GOTO geld

days:
DATA "Sunday ","Monday ","Tuesday
","Wednesday","Thursday ","Friday
","Saturday ","Sunday "

zienn:
IF mnd(a)=1 THEN w=ddg(a)
IF mnd(a)=2 THEN w=31+ddg(a)
rq=1:GOSUB kll
RESTORE dta:FOR zg=1 TO mnd(a):READ
sp:NEXT:vw=sp:tr=feb:sp=sp+feb+ddg(a):w=sp:GOTO
zvp

dta:
DATA 0,0,31,62,92,123,153,184,215,245,276,306
REM * Er zit geen beveiliging op voor
ingeving van bv. 31 feb enz. *

zvp:
i=w:GOTO oo

geld:
FOR a=1 TO b:gldd=0:gldg=0:LOCATE
18,7:PRINT"Make cost-list":LOCATE 18,40:PRINT
a
IF plts(a)=2 AND zazo(a)=0 THEN
IF sd(a)>0 THEN
gldd=(INT(sd(a)/300)+1)*.15

```

- Print out Amiga - Print out Amiga - Print out Amiga -

```

IF sg(a)>0 THEN
gldg=(INT(sg(a)/600)+1)*.15
IF sg(a)>0 AND sd(a)>0 THEN
tar$(a)="300/600":GOTO f
IF sg(a)>0 AND sd(a)=0 THEN tar$(a)="
600 ":GOTO f
tar$(a)=" 300 "
END IF

```

```

f:
IF plts(a)=2 AND zazo(a)=2 THEN
IF sg(a)>0 THEN
gldg=(INT(sg(a)/600)+1)*.15
IF sd(a)>0 THEN
gldd=(INT(sd(a)/600)+1)*.15
tar$(a)=" 600 "
END IF

```

```

g:
IF plts(a)=1 AND zazo(a)=0 THEN
IF sd(a)>0 THEN gldd=(INT(sd(a)/47)+1)*.15
IF sg(a)>0 THEN
gldg=(INT(sg(a)/94)+1)*.15
IF sg(a)>0 AND sd(a)>0 THEN
tar$(a)="47/94 ":GOTO ij
IF sg(a)>0 AND sd(a)=0 THEN tar$(a)=" 94
":GOTO ij
tar$(a)=" 47 "
END IF

```

```

ij:
IF plts(a)=1 AND zazo(a)=2 THEN
IF sg(a)>0 THEN gldg=(INT(sg(a)/94)+1)*.15
IF sd(a)>0 THEN gldd=(INT(sd(a)/94)+1)*.15
tar$(a)=" 94 "
END IF
kosten(a)=(CINT(100*(gldg+gldd)))/100:kosten$=S
TR$(kosten(a)):kosten$=RIGHT$(kosten$,LEN(koste
n$)-1)
totkosten=totkosten+kosten(a)

```

```

zai:
IF LEFT$(kosten$,1)=". " THEN
kosten$="0"+kosten$
IF MID$(kosten$,2,1)=". " AND LEN(kosten$)=3
THEN kosten$=kosten$+"0 ":GOTO further
IF MID$(kosten$,3,1)=". " AND LEN(kosten$)=4
THEN kosten$=kosten$+"0 "

```

```

further:
kosten$=" "+kosten$:kosten$(a)=kosten$:IF
tr=1 THEN tr=0:RETURN

```

```

beeldop:
num$(a)=STR$(a):num$(a)=RIGHT$(num$(a),LEN(num$
(a))-1)
IF LEN(num$(a))=1 THEN num$(a)="000"+num$(a)
IF LEN(num$(a))=2 THEN num$(a)="00"+num$(a)
IF LEN(num$(a))=3 THEN num$(a)="0"+num$(a)
n=1:ddg$=STR$(ddg(a)):mnd$=STR$(mnd(a)):jrg$=ST
R$(jrg(a))
ddg$=RIGHT$(ddg$,LEN(ddg$)-1):mnd$=RIGHT$(mnd$,
LEN(mnd$)-1):jrg$=RIGHT$(jrg$,LEN(jrg$)-1)
IF LEN(ddg$)=1 THEN ddg$="0"+ddg$
IF LEN(mnd$)=1 THEN mnd$="0"+mnd$
datumy$(a)=ddg$+"-"+mnd$+"-"+jrg$

```

```

sa$=INKEY$:IF sa$="" THEN CLS:notime=0:GOTO
prnt
IF sa$=CHR$(27) THEN prnt
GOTO zal

```

```

totS:
CLS:tq=4:GOSUB tot
LINE (0,14)-(630,14):LINE
(0,25)-(630,25):LINE (0,158)-(630,158):LINE
(0,169)-(630,169)

```

```

LOCATE 3,22:PRINT"Survey from "olddate$
till "newdate$
LOCATE 7,14:PRINT "Total Calls Made:"b
LOCATE 8,14:PRINT "Total Time      :
"tottijd$
LOCATE 9,14:PRINT "Total Costs      :
"totkosten$
LOCATE 10,14:PRINT"Total Int.Local : "intl$
LOCATE 11,14:PRINT"Total Local      : "lc
LOCATE 12,14:PRINT"Total Uploads    : "up"
->"tupkb
LOCATE 13,14:PRINT"Total downloads : "do"
->"tdokb
GOSUB spatie:tq=0:GOTO prnt

```

```

toets:
LOCATE 21,25:PRINT"Press space for next
screen"

```

```

zgl:
ga$=INKEY$:IF ga$="" THEN j=1:GOSUB x:GOTO
vvv
IF ga$=CHR$(27) THEN prnt
GOTO zgl

```

```

tot:
tot=1:a=0:l=tottijd:ut=tottijd:GOSUB
terug:tottijd$=tjdc$(a):a=0:tr=1
totkosten=(CINT(totkosten*100))/100
totkost$(a)=STR$(totkosten):kosten$=RIGHT$(totk
ost$(a),LEN(totkost$(a))-1)
GOSUB zai:totkosten$="F "+kosten$(0):RETURN

```

```

opbouw:
CLEAR:RANDOMIZE TIMER
SCREEN 2,640,255,2,2
WINDOW 2,SPACE$(1)+"1990 Telephone Calc v1.0
by Mark de Jong", (0,11)-(631,203),16,2
PALETTE 0,0,0,.5:PALETTE 1,1,1,.13:GOSUB
blok:LOCATE 13,29:PRINT"ONE MOMENT PLEASE"
GOSUB ingeving:GOSUB
dimmen:overs$="YES":overs=0:logn$="YES":nolog=0
:dat$="YES":dat=0:tim$="YES":tim=0
:lc1$="YES":lc1=0:ilcl$="YES":ilcl=0:tf$="ALL
":tf=7:mem$="EMPTY"
ON TIMER(1) GOSUB timing
TIMER ON
BREAK ON
ON BREAK GOSUB breaking
ON ERROR GOTO errors

```

```

prnt:
IF lg$="" THEN lg$=tlg$
CLS:CLOSE #1:notime=2:CLS:GOSUB scherm:LOCATE
7,26:PRINT "[C] - Clear Memory"
LOCATE 8,26:PRINT "F 1 - Start Calculate
Costs"
LOCATE 9,26:PRINT "F 2 - Give Log-Source"
LOCATE 10,26:PRINT"F 3 - Choose on date"
LOCATE 11,26:PRINT"F 4 - Scan Calls"
LOCATE 12,26:PRINT"F 5 - List Calls
(seperate)"
LOCATE 13,26:PRINT"F 6 - Total Survey"
LOCATE 14,26:PRINT"F 7 - Print Calls"
LOCATE 15,26:PRINT"F 8 - Load Calls"
LOCATE 16,26:PRINT"F 9 - Save Calls"
LOCATE 17,26:PRINT"F10 - Exit Telephone
Calc"
LOCATE 5,2:PRINT "MAINMENU":LOCATE
3,2:PRINT"Memory: "mem$
LOCATE 5,45:PRINT"LogFile: "lg$

```

```

wm:
wm$=INKEY$:IF wm$="" THEN wm
IF wm$=CHR$(129) AND calc=0 THEN StartC
IF wm$=CHR$(130) AND calc=0 THEN
notime=2:GOTO GiveL

```


- Print out Amiga - Print out Amiga - Print out Amiga -

```

IF wm$=CHR$(131) AND calc=0 THEN
notime=2:GOTO setd
IF wm$=CHR$(132) AND calc=1 THEN
notime=2:GOTO ScanC
IF wm$=CHR$(133) AND calc=1 THEN
notime=2:GOTO ListC
IF wm$=CHR$(134) AND calc=1 THEN
notime=2:GOTO tots
IF wm$=CHR$(135) AND calc=1 THEN
notime=2:GOTO PrintC
IF wm$=CHR$(136) AND mem$<>"IN USE" THEN
notime=0:GOTO LoadC
IF wm$=CHR$(137) AND mem$<>"EMPTY" THEN
notime=0:GOTO SaveC
IF wm$=CHR$(138) THEN notime=0:GOTO ExitC
IF wm$="C" OR wm$="c" AND mem$="IN USE" THEN
CLS:GOSUB zeker:IF clr=1 THEN clr=0:GOTO
prnt
GOSUB
del:lg$=olg$:mem$="EMPTY":calc=0:ld=0:up=0:do=0
:b=0:GOTO prnt
END IF
GOTO wm

StartC:
CLS:GOSUB scherm
LOCATE 5,14:PRINT"Process":LOCATE
5,42:PRINT"N":LOCATE 3,1:PRINT"LOGFILE: "lg$
LOCATE 8,3:PRINT"1":LOCATE 10,3:PRINT
"2":LOCATE 12,3:PRINT "3":LOCATE 14,3:PRINT
"4"
LOCATE 16,3:PRINT"5":LOCATE 18,3:PRINT "6"
GOTO lop

GiveL:
CLS:notime=1:lgg$=lg$:GOSUB scherm:LINE
(110,70)-(490,70):LINE (110,130)-(490,130)
LINE (110,130)-(110,70):LINE
(490,130)-(490,70):LOCATE 5,2:PRINT
"Change Log-Source"
LOCATE 12,16:PRINT "Current Log-Source : "lg$

opn:
LOCATE 14,16:PRINT "Changing to
";:LINE INPUT lg$
IF lg$="" THEN lgg$=lgg$
GOTO prnt

ScanC:
IF calc=1 THEN x

setd:
notime=2:zb=0:GOSUB blok:LOCATE
5,2:PRINT"Starting Date":LOCATE
12,27:PRINT"Give Start Day : ";:LINE INPUT
old$(1)
LOCATE 10,27:PRINT"To Quit type / (3 times)"
LOCATE 13,27:PRINT" Month: ";:LINE
INPUT old$(2)
LOCATE 14,27:PRINT" Year : ";:LINE
INPUT old$(3)
IF old$(1)="/" OR old$(2)="/" OR old$(4)="/"
THEN prnt
IF old$(1)="" OR old$(2)="" OR old$(3)=""
THEN setd
old(1)=VAL(old$(1)):old(2)=VAL(old$(2)):old(3)=
VAL(old$(3)):IF old(1)>31 OR old(2)>12 THEN
BEEP:GOTO setd
IF LEN(old$(1))=1 THEN old$(1)="0"+old$(1)
IF LEN(old$(2))=1 THEN old$(2)="0"+old$(2)
IF LEN(old$(3))=2 THEN old$(3)="19"+old$(2)
olddate$=old$(2)+"/"+old$(1)+"/"+old$(3):LOCATE
21,17:PRINT"Do you want to give also an
Ending Date?(y/n)"

t:
t$=INKEY$:IF t$="" THEN t

IF t$="y" THEN neww
IF t$="n" THEN newold=1:onlyo=1:GOTO prnt
GOTO t

neww:
GOSUB blok
LOCATE 5,2:PRINT"Ending Date"

LOCATE 10,27:PRINT"To Quit type / (3 times)"
LOCATE 12,27:PRINT"Give End Day : ";:LINE
INPUT neww$(1)
LOCATE 13,27:PRINT" Month: ";:LINE
INPUT neww$(2)
LOCATE 14,27:PRINT" Year : ";:LINE
INPUT neww$(3)
IF neww$(1)="/" OR neww$(2)="/" OR
neww$(4)="/" THEN onlyo=0:GOTO prnt
IF neww$(1)="" OR neww$(2)="" OR neww$(3)=""
THEN neww
neww(1)=VAL(neww$(1)):neww(2)=VAL(neww$(2)):new
w(3)=VAL(neww$(3)):IF neww(1)>31 OR
neww(2)>12 THEN BEEP:GOTO neww
IF LEN(neww$(1))=1 THEN neww$(1)="0"+neww$(1)
IF LEN(neww$(2))=1 THEN neww$(2)="0"+neww$(2)
IF LEN(neww$(3))=2 THEN neww$(3)="19"+neww$(2)
newdate$=neww$(2)+"/"+neww$(1)+"/"+neww$(3):new
old=1:GOSUB spatie:notime=0:GOTO prnt

blok:
CLS:GOSUB scherm:LINE (190,70)-(410,70):LINE
(190,130)-(410,130)
LINE (190,130)-(190,70):LINE
(410,130)-(410,70):RETURN

scherm:
LOCATE 2,26:PRINT " TELEPHONE CALC V1.0 "
LINE (0,30)-(630,30):LINE (0,40)-(630,40)
LINE (0,158)-(630,158):LINE
(0,169)-(630,169):RETURN

spatie:
WINDOW OUTPUT(2)
IF notime<>2 AND tq<>4 THEN LOCATE
23,3:PRINT "Calculating Time:"trijsd sec."
LOCATE 21,25:PRINT"Press space for next
screen"

ziu:
WINDOW OUTPUT(2):WINDOW(2)
fa$=INKEY$:IF fa$="" THEN ziu
IF fa$=" " THEN RETURN
GOTO ziu

timing:
blup$=INKEY$:IF notime=4 AND blup$="" THEN
notime=0:trijsd=0:GOTO PrintC
IF notime<>2 THEN trijsd=trijsd+1:RETURN
REM LOCATE 5,59:PRINT TIME$ "DATE$:rem Als
je tijd in je menu wilt hebben haal dan de
rem helemaal vooraan weg
RETURN

dimmen:
DIM a$(1500):l=1:DIM b$(400):de=1:q=1:DIM
f$(100):DIM datumb$(300):DIM datume$(300):DIM
Tijdb$(300):DIM tijde$(300):DIM ttijd(300)
DIM uur(300):DIM tuur(300):DIM emin(300):DIM
tmin(300):DIM esec(300):DIM tsec(300):DIM
sg(300):DIM sd(300):DIM plts(300)
DIM mnd(300):DIM ddg(300):DIM jrg(300):DIM
wkd$(300):DIM zazo(250):DIM kosten$(300):DIM
num$(300)
DIM tijdc$(300):DIM plaats$(300):DIM
up$(200):DIM do$(200):DIM tar$(500):DIM
datumy$(300):DIM up(200):DIM do(200)
DIM sss(400):DIM kosten(500):DIM ukb(100):DIM
dkb(200):DIM caller$(500)

```

- Print out Amiga - Print out Amiga - Print out Amiga -

RETURN

ListC:

a=1:CLS:GOSUB scherm:notime=1:a=1

lcalls:

IF a<=0 THEN a=1

IF a>b THEN a=b

ukb\$="" : dkb\$="" : ukb\$=STR\$(ukb(a)) : dkb\$=STR\$(dkb(a)) :

IF ukb(a)=0 THEN ukb\$=" -"

IF dkb(a)=0 THEN dkb\$=" -"

LOCATE 5,2:PRINT"List Calls":LOCATE

7,17:PRINT"Number:"num\$(a):LOCATE

7,29:PRINT"(1 -"b")"

LOCATE 9,20:PRINT "Made Phonecall to

: "caller\$(a) STRING\$(15," ")

LOCATE 10,20:PRINT"Date :

"datumy\$(a) " "

LOCATE 11,20:PRINT"Day :

"wkd\$(a) STRING\$(10," ")

LOCATE 12,20:PRINT"Time Connected :

"tijdb\$(a) " "

LOCATE 13,20:PRINT"Time Disconnected :

"tijde\$(a) " "

LOCATE 14,20:PRINT"Total Online-time :

"tijdc\$(a) " "

LOCATE 15,20:PRINT"Tariff :

"tar\$(a) " "

LOCATE 16,20:PRINT"Costs this Call

: "kosten\$(a) " "

LOCATE 17,20:PRINT"Local/Int.Local :

"plaats\$(a) " "

LOCATE 18,20:PRINT"Made Uploads

: "up(a) " " : LOCATE 18,45:PRINT"Uploads in

Bytes : "ukb\$ " "SPACE\$(10)

LOCATE 19,20:PRINT"Made Downloads

: "do(a) " " : LOCATE 19,45:PRINT"Downloads in

Bytes: "dkb\$ " "SPACE\$(10)

LOCATE 21,1:PRINT"Cursor Up = Forward |

Cursor Down = Backwards | Esc = MainMenu | J

= Jump to..."

curs:

curs\$=INKEY\$:IF curs\$="" THEN curs

IF curs\$=CHR\$(27) THEN GOTO prnt

IF curs\$=CHR\$(28) THEN a=a+1:GOTO lcalls

IF curs\$=CHR\$(29) THEN a=a-1:GOTO lcalls

IF curs\$="j" THEN GOSUB jumping:GOTO lcalls

GOTO curs

jumping:

LOCATE 23,40:PRINT SPACE\$(37)

LOCATE 23,30:PRINT"Jump to N: ";:LINE INPUT

rad\$

IF rad\$="" THEN LOCATE 23,30:PRINT

STRING\$(56," ") : RETURN

gq=VAL(rad\$):IF gq>b OR gq<=0 THEN jumping

a=gq:LOCATE 23,30:PRINT STRING\$(56," ") : RETURN

breaking:

RETURN

ExitC:

GOSUB blok:LOCATE 11,32:PRINT"Are you

Sure":LOCATE 13,30:PRINT "You want to leave"

LOCATE 15,31:PRINT "Telephone Calc?":IF

calc=1 THEN LOCATE 21,29:PRINT"MEMORY IS NOT

EMPTY!!"

uitgang:

uit\$=INKEY\$:IF uit\$="y" THEN

IF clr=1 THEN clr=0:RETURN

CLEAR:CLS:SYSTEM

END IF

IF uit\$="n" THEN

IF clr=1 THEN RETURN

GOTO prnt

END IF

GOTO uitgang

PrintC:

a=1:GOTO sure

END

sure:

GOSUB blok:LOCATE 5,2:PRINT"Print

Calls":LOCATE 13,27:PRINT"Are you Sure?(y/n)"

sur:

sur\$=INKEY\$:IF sur\$="" THEN sur

IF sur\$=CHR\$(27) THEN prnt

IF sur\$="y" THEN een

IF sur\$="n" THEN prnt

GOTO sur

een:

GOSUB blok:LOCATE 13,29:PRINT"Also

Totals?(y/n)"

also:

als\$=INKEY\$:IF als\$="" THEN also

IF als\$="y" THEN totook=2:GOTO llp

IF als\$="n" THEN llp

IF als\$=CHR\$(27) THEN prnt

GOTO also

llp:

GOSUB blok

pro=2

LOCATE 13,32:PRINT"Printing..."

LOCATE 21,20:PRINT"(Press Alt-X several

times to STOP)"

notime=4:LPRINT SPACE\$(20)"Oversey over

"olddate\$" Till "newdate\$:LPRINT

nol:

IF ld=2 AND tm=1 THEN GOSUB beeldop

LPRINT "Name LogFile: "lg\$

LPRINT:LPRINT STRING\$(80,"=")

LPRINT" N. Date Day Time

loc/int Up Do Tariff Costs "

LPRINT STRING\$(80,"=")

lijstprint:

FOR a=1 TO b

pr=1:LPRINT " "num\$(a) " "datumy\$(a) "

"wkd\$(a) " "tijdc\$(a) " "plaats\$(a) "

"up\$(a) " "do\$(a) " "tar\$(a) " F

"kosten\$(a)

nx:

NEXT:LPRINT STRING\$(80,"="):LPRINT:IF

totook<>2 THEN CLOSE

#1:notime=0:totook=0:GOTO prnt

Totally:

GOSUB tot:notime=4:LPRINT " TOTAL OVERVIEW:

"olddate\$" Till "newdate\$:LPRINT

LPRINT SPACE\$(16)"Total Calls Made:"b

LPRINT SPACE\$(16)"Total Time : "tottijd\$

LPRINT SPACE\$(16)"Total Costs :

"totkosten\$

LPRINT SPACE\$(16)"Total Int.Local : "intl\$

LPRINT SPACE\$(16)"Total Local : "lc

LPRINT SPACE\$(16)"Total Uploads : "up"

->"tupkb

LPRINT SPACE\$(16)"Total downloads : "do"

->"tdokb

LPRINT:LPRINT STRING\$(80,"=")

LPRINT:CLOSE #1:notime=0:GOTO prnt

LoadC:

notime=0:tq=4:GOSUB blok:LOCATE

5,2:PRINT"Loading File"

- Print out Amiga - Print out Amiga - Print out Amiga -

```
LOCATE 12,26:PRINT"Load name:":LOCATE
13,26:PRINT">";:LINE INPUT laad$
IF laad$="" THEN prnt
laad$=laad$+".tc"
OPEN "I",#1,laad$
INPUT #1,b:FOR a=1 TO b
INPUT
#1,l,lc,tottijd,intlc,totkosten,up,do,tupkb,tdo
kb,num$(a),olddate$,newdate$,b$(a),f$(a)
INPUT
#1,datum$(a),datume$(a),Tijdb$(a),tijde$(a),tt
ijd(a)
INPUT
#1,eur$(a),tuur$(a),emin$(a),tmin$(a),esec$(a),tsec
(a),sg$(a),sd$(a),plts$(a)
INPUT
#1,mnd$(a),ddg$(a),jrg$(a),wkd$(a),zazo$(a),kosten$
(a),tijdc$(a),plaats$(a),up$(a),do$(a)
INPUT
#1,tar$(a),datumy$(a),up(a),do(a),sss(a),kosten
(a),dkb(a),ukb(a),caller$(a)
NEXT da=b:calc=1:ld=2:tm=1:CLOSE #1:LOCATE
18,25:PRINT"Loading is done."
GOSUB spatie:CLOSE
#1:olg$=lg$:lg$=laad$:mem$="IN USE":GOTO prnt
```

SaveC:

```
notime=0:tq=4:GOSUB blok:LOCATE
5,2:PRINT"Save File":LOCATE 12,26:PRINT"Save
Name:"
LOCATE 13,26:PRINT">";:LINE INPUT sav$:IF
sav$="" THEN prnt
sav$=sav$+".tc"
OPEN sav$ FOR OUTPUT AS #1
WRITE #1,b:FOR a=1 TO b
WRITE
#1,l,lc,tottijd,intlc,totkosten,up,do,tupkb,tdo
kb,num$(a),olddate$,newdate$,b$(a),f$(a),datum$
(a)
WRITE
#1,datume$(a),Tijdb$(a),tijde$(a),ttijd(a)
WRITE
#1,eur$(a),tuur$(a),emin$(a),tmin$(a),esec$(a),tsec
(a),sg$(a),sd$(a),plts$(a)
WRITE
#1,mnd$(a),ddg$(a),jrg$(a),wkd$(a),zazo$(a),kosten$
(a),tijdc$(a),plaats$(a),up$(a),do$(a)
WRITE
#1,tar$(a),datumy$(a),up(a),do(a),sss(a),kosten
(a),dkb(a),ukb(a),caller$(a)
NEXT:CLOSE #1:LOCATE 18,25:PRINT"Saving is
done.":GOSUB spatie:CLOSE #1:GOTO prnt
```

del:

```
FOR a=1 TO
b:a$(a)="" :up(a)=0:ukb(a)=0:dkb(a)=0:tupkb=0:td
okb=0:do(a)=0:caller$(a)="" :kosten$(a)="" :NEXT
b=0:da=0:a=0:intlc=0:lc=0:kosten$="" :l=0:ut=0:u
p=0:do=0:dkb=0:upkb=0:tottijd=0
totkosten=0:tijdc$(0)=""
RETURN
```

als:

```
FOR r=1 TO totaal:IF ing$(r)="" THEN r
IF RIGHT$(a$(a),LEN(ing$(r)))=ing$(r) THEN
plts(b)=2:jjj=b
```

r:

```
NEXT r:RETURN
```

zeker:

```
GOSUB blok:LOCATE 5,2:PRINT "Clear
memory":LOCATE 13,29:PRINT"Are you
sure?(y/n)":clr=1:GOTO uitgang
END
```

errors:

WINDOW(2)

```
IF (ERR=53) AND pro=2 THEN
pro=0:notime=2:WINDOW(2):LOCATE 19,25:PRINT
"Printer Stuff not on Disk":GOSUB
spatie:RESUME prt
IF (ERR=53) THEN notime=2:WINDOW(2):LOCATE
19,25:PRINT "File 'lg$' Not Found!":GOSUB
spatie:RESUME prt
IF (ERR=70) THEN notime=2:WINDOW(2):LOCATE
19,25:PRINT "Disk Write Protected!":GOSUB
spatie:RESUME prt
IF (ERR=61) THEN notime=2:WINDOW(2):LOCATE
19,25:PRINT "Disk Is Full!":GOSUB
spatie:RESUME prt
IF (ERR=57) THEN notime=2:WINDOW(2):LOCATE
19,25:PRINT "Device I/O Error!":GOSUB
spatie:RESUME prt
IF (ERR=13) THEN notime=2:WINDOW(2):LOCATE
19,25:PRINT "Can Only Load
TelCalc-files":GOSUB spatie:RESUME prt
PRINT ERR
RESUME tttt
```

tttt:
END

Mastermind

Lucien Heins combineert een oud gegeven, het traditionele denkspel mastermind, met een Amiga. Kleuren moeten worden geraden, en op de goede plaats worden weergegeven. Een uitleg wordt er in het programma gegeven.

```
REM *** Mastermind ***
REM door Lucien Heins in Augustus 1988
```

```
WINDOW CLOSE 1:RANDOMIZE TIMER
SCREEN 1,320,250,5,1:WINDOW 2," ***
Mastermind *** door Lucien Heins",,0,1
PALETTE 0,.5,.5,.5:PALETTE 1,0,0,0:PALETTE
2,1,1,1
PALETTE 3,1,0,0:PALETTE 4,0,1,0:PALETTE
5,0,0,1:PALETTE 6,1,1,0
FOR i=7 TO 22:PALETTE
i,(i-6)/16,(i-6)/16,1-(i-6)/16:NEXT
COLOR 6,0:pr 14,6,"Wilt u instructies (j/n)
?"
du$="<":dv$=">"
w:SWAP du$,dv$:LOCATE 14,33:PRINT du$
in$=INKEY$:IF in$<>"j" AND in$<>"n" THEN w
IF in$="n" THEN verder
```

```
scherm schoon
COLOR 4,0:PRINT:PRINT " Mastermind instructies"
PRINT"-----"
COLOR 5,0:PRINT
FOR i=1 TO 22:READ t$:PRINT t$:NEXT
COLOR 0,1:LOCATE 28,2:PRINT" Druk op een
toets... ":COLOR 1,0
w3:IF INKEY$="" THEN w3
```

verder:scherm schoon

```
t=8:r=1
FOR i=1 TO 125
CIRCLE (160,118),i,t,,,1
IF t=7 OR t=22 THEN r=-r
t=t+r
NEXT
FOR i=0 TO 256 STEP 32
FOR j=0 TO 31
IF j>15 THEN k=38-j ELSE k=j+7
```


- Print out Amiga - Print out Amiga - Print out Amiga -

```

LINE (i+j,0)-(320,i+j),k
LINE (320-(i+j),0)-(0,i+j),k
LINE (i+j,250)-(0,i+j),k
LINE (320-(i+j),250)-(320,i+j),k
NEXT j
NEXT i

start:LINE (80,10)-(240,226),0,bf
LINE (80,10)-(240,226),1,b
pr 6,21,"1=zwart"
pr 7,21,"2=wit"
pr 8,21,"3=rood"
pr 9,21,"4=groen"
pr 10,21,"5=blauw"
pr 11,21,"6=geel"
FOR i=1 TO 4
  ran:ck(i)=INT(RND*6)+1
  IF i<>1 THEN
    FOR j=1 TO i-1:IF ck(j)=ck(i) THEN ran
    NEXT j
  END IF
NEXT i
FOR b=1 TO 10
  be$="beurt: "+STR$(b):pr 4,21,be$
  gk=0:gp=0
  FOR i=1 TO 4:k(i)=0:NEXT
  FOR i=1 TO 4
    CIRCLE (91+(i-1)*8,27+(b-1)*16),3,1,,1
    NEXT
    pp=1
    wi:b%=b:pp%=pp:du$=" "
    MID$(du$,pp,1)="^":pr 2*b%+3,12,du$
    FOR i=1 TO 4:IF k(i)=0 THEN el
    CIRCLE (91+(i-1)*8,27+(b-1)*16),3,k(i),,,1
    PAINT (91+(i-1)*8,27+(b-1)*16),k(i),k(i)
    el:NEXT i
    wi2:in$=INKEY$:IF in$="" THEN wi2
    IF in$=CHR$(13) THEN
      FOR i=1 TO 4:IF k(i)=0 THEN BEEP:GOTO wi2
      NEXT
      FOR i=1 TO 4
        IF i<>1 THEN
          FOR j=1 TO i-1:IF k(j)=k(i) THEN BEEP:GOTO
wi2
          NEXT j
        END IF
        IF i<>4 THEN
          FOR j=i+1 TO 4:IF k(j)=k(i) THEN BEEP:GOTO
wi2
          NEXT j
        END IF
        NEXT i
        GOTO klaar
      END IF
      IF in$=CHR$(30) AND pp<>4 THEN pp=pp+1:GOTO
wi
      IF in$=CHR$(31) AND pp<>1 THEN pp=pp-1:GOTO
wi
      IF in$<"1" OR in$>"6" THEN wi2
      k(pp)=VAL(in$):GOTO wi
    klaar:
    pr 2*b%+3,12," "
    FOR i=1 TO 4
      IF k(i)=ck(i) THEN gp=gp+1
      IF i<>1 THEN
        FOR j=1 TO i-1:IF ck(j)=k(i) THEN gk=gk+1
        NEXT j
      END IF
      IF i<>4 THEN
        FOR j=i+1 TO 4:IF ck(j)=k(i) THEN gk=gk+1
        NEXT j
      END IF
      NEXT i
      IF gp=4 THEN geraden
      IF gp<>0 THEN
        FOR i=1 TO gp

```

```

CIRCLE (128+(i-1)*6,27+(b-1)*16),2,1,,,1
PAINT (128+(i-1)*6,27+(b-1)*16),1,1
NEXT
END IF
IF gk<>0 THEN
  FOR i=1 TO gk
    CIRCLE (128+gp*6+(i-1)*6,27+(b-1)*16),2,2,,,1
    PAINT (128+gp*6+(i-1)*6,27+(b-1)*16),2,2
  NEXT
END IF
NEXT b

pr 15,21,"Je hebt"
pr 16,21,"de code"
pr 17,21,"niet bin-"
pr 18,21,"nen 10"
pr 19,21,"beurten"
pr 20,21,"kunnen"
pr 21,21,"raden !"
pr 24,12,"De code was:"
FOR i=1 TO 4
CIRCLE (190+(i-1)*8,187),3,ck(i),,,1
PAINT (190+(i-1)*8,187),ck(i),ck(i)
NEXT
pr 27,12,"Wil je nog eens ?"

GOTO nogeens

geraden:
LINE (81,11)-(239,225),0,bf
FOR i%=3 TO 8:pr i%,13,"GEFELICITEERD
!!":NEXT
pr 10,12,"Je hebt de code"
te$="in"+STR$(b)+" beurten":pr 11,12,te$
pr 12,12,"geraden !"
pr 15,12,"Wil je nog eens ?"
GOTO nogeens

nogeens:
w2:in$=INKEY$:IF in$<>"j" AND in$<>"n" THEN
w2
IF in$="j" THEN start
scherm schoon
SCREEN CLOSE 1:WINDOW 1
FOR i=1 TO 1500 STEP 25
PRINT PTAB(220+200*SIN(i/180));"Tot de
volgende keer !!"
NEXT
END

SUB pr (y%,x%,t$) STATIC
LOCATE y%,x%:PRINT t$
END SUB

SUB scherm schoon STATIC
FOR i=118 TO 0 STEP -1
LINE (160-(i+42),118-i)-(202+i,118+i),3,b
LINE (160-(i+42),118-i)-(202+i,118+i),0,b
NEXT:CLS
END SUB

DATA "Bij dit spel neemt de Amiga een
geheime"
DATA "code in gedachten, die de speler
binnen"
DATA "10 beurten moet proberen te raden. De"
DATA "code bestaat uit vier verschillende"
DATA "kleuren. Er zijn 6 kleuren: zwart,wit,"
DATA "rood,groen,blauw en geel. Je kunt een"
DATA "code ingeven met de cursortoetsen en"
DATA "zes verschillende cijfers, druk op re-"
DATA "turn als je klaar bent. Als de speler"
DATA "een code ingegeven heeft, zegt de
Amiga"
DATA "wat er van die code overeenkomt met de"
DATA "geheime code. Dat doet hij door zwarte"
DATA "en witte bolletjes te tekenen. Een"

```

- Print out Amiga - Print out Amiga - Print out Amiga -

```
DATA "zwart bolletje betekent: een van de"
DATA "vier kleuren is goed en staat ook op de"
DATA "goede plaats. Een wit bolletje bete-"
DATA "kent: alleen de kleur is goed. Met deze"
DATA "informatie die je na elke beurt krijgt,"
DATA "moet je proberen de code in zo min mo-"
DATA "gelijk beurten te raden. Als je de code"
DATA "na tien beurten nog niet gevonden hebt,"
DATA "heb je verloren."
```

REKENEN AMIGA

Het rekenprogramma is gemaakt door de toen 15 jarige Viktor Vos uit Hilversum. Het is een gedeeltelijk muis gestuurd programma waarmee niet al te moeilijke sommen mee kunnen worden gemaakt. Er wordt gebruik gemaakt van verschillende toonhoogstes bij het al dan niet goed beantwoorden van een aantal opdrachten.

```
REM *****
REM ***** Amiga Rekenen *****
REM ***** Door Viktor Vos *****
REM ***** Voor Commodore info!! *****
REM *****
```

```
CLEAR,25000:SCREEN 1,320,300,5,1:WINDOW 2,"
",0,1:PALETTE 1,0,0,0:PALETTE 0,0,0,0:POKE
12574721&,254
PALETTE 4,0,.7,0:COLOR 4,0:MENU OFF
```

Start:

```
CLS
LOCATE 1,17:PRINT "Rekenen":LOCATE
2,17:PRINT "-----"
LOCATE 4,1:COLOR 8,0:PRINT
"-----"
LOCATE 5,1:COLOR 3,0:PRINT "      Klik het
rondje aan met de muis! "
LOCATE 6,1:COLOR 8,0:PRINT
"-----"
CIRCLE (7,50),5,4:PAINT (7,50),4:LOCATE
7,3:COLOR 5,0:PRINT "Optellen"
CIRCLE (7,66),5,4:PAINT (7,66),4:LOCATE
9,3:PRINT "Aftrekken"
CIRCLE (7,82),5,4:PAINT (7,82),4:LOCATE
11,3:PRINT "Vermenigvuldigen"
CIRCLE (7,98),5,4:PAINT (7,98),4:LOCATE
13,3:PRINT "Info"
CIRCLE (7,114),5,4:PAINT (7,114),4:LOCATE
15,3:PRINT "Stoppen"
LOCATE 16,1:COLOR 8,0:PRINT
"-----"
LOCATE 17,1:COLOR 0,3:PRINT "      Gemaakt
door Viktor Vos voor:      "
LOCATE 18,1:PRINT "      Commodore
Info!      "
LOCATE 19,1:COLOR 8,0:PRINT
"-----"
CIRCLE (20,168),20,9:PAINT (20,168),9:LOCATE
21,2:COLOR 0,9:PRINT "R"
CIRCLE (58,168),20,9:PAINT (58,168),9:LOCATE
21,7:PRINT "E"
CIRCLE (94,168),20,9:PAINT (94,168),9:LOCATE
21,12:PRINT "K"
```

```
CIRCLE (131,168),20,9:PAINT
(131,168),9:LOCATE 21,17:PRINT "E"
CIRCLE (168,168),20,9:PAINT
(168,168),9:LOCATE 21,21:PRINT "N"
CIRCLE (203,168),20,9:PAINT
(203,168),9:LOCATE 21,25:PRINT "E"
CIRCLE (241,168),20,9:PAINT
(241,168),9:LOCATE 21,30:PRINT "N"
CIRCLE (280,168),20,9:PAINT
(280,168),9:LOCATE 21,35:PRINT "!!":COLOR 5,0
```

Checkmouse:

```
IF MOUSE (0)=0 THEN Checkmouse
IF MOUSE (1)>2 AND MOUSE (1)<12 AND MOUSE
(2)>45 AND MOUSE (2)<55 THEN p=0:GOTO
Optelsom
IF MOUSE (1)>2 AND MOUSE (1)<12 AND MOUSE
(2)>61 AND MOUSE (2)<71 THEN p=0:GOTO
Aftreksom
IF MOUSE (1)>2 AND MOUSE (1)<12 AND MOUSE
(2)>77 AND MOUSE (2)<87 THEN p=0:GOTO
Vermenigvuldigsom
IF MOUSE (1)>2 AND MOUSE (1)<12 AND MOUSE
(2)>93 AND MOUSE (2)<103 THEN Info
IF MOUSE (1)>2 AND MOUSE (1)<12 AND MOUSE
(2)>109 AND MOUSE (2)<119 THEN Stoppen
GOTO Checkmouse
```

Randomkeuze:

```
COLOR 0,0:RANDOMIZE
TIMER:a=INT(RND(1)*100):RANDOMIZE
TIMER:b=INT(RND(1)*100):c=0:RETURN
```

Fout:

```
LOCATE 10,15:PRINT "Fout!"c:p=p-1:FOR w=1
TO 5000:NEXT:RETURN
```

Goed:

```
LOCATE 10,16:PRINT "Goed! ":p=p+1:FOR w=1
TO 5000:NEXT:RETURN
```

Invoer:

```
LOCATE 8,23
PALETTE 0,0,0,0:COLOR 4,0:INPUT ab$
IF ab$="x" THEN Start
IF VAL(ab$)=c THEN GOSUB Goed
IF VAL(ab$)<>c THEN GOSUB Fout
RETURN
```

Text:

```
LOCATE 3,5:COLOR 3,0:PRINT "      Los de
volgende sommen op:
LOCATE 5,5:PRINT "(type x+return om te
stoppen.)":RETURN
```

Optelsom:

```
CLS:GOSUB Randomkeuze
GOSUB Text
LOCATE 8,13:PALETTE 0,0,0,0:COLOR 5,0:PRINT
a"+"b"="
c=a+b:LOCATE 1,14:COLOR 9,0:PRINT
"Punten:"p:LOCATE 2,14:COLOR 4,0:PRINT
"-----"
IF p=-10 THEN Tienfout
IF p=10 THEN Bonus
GOSUB Invoer
GOTO Optelsom
```

Aftreksom:

```
CLS:GOSUB Randomkeuze
GOSUB Text:IF a<b THEN Aftreksom
LOCATE 8,13:PALETTE 0,0,0,0:COLOR 5,0:PRINT
a-"b"="
c=a-b:LOCATE 1,14:COLOR 9,0:PRINT
"Punten:"p:LOCATE 2,14:COLOR 4,0:PRINT
"-----"
IF p=-10 THEN Tienfout
```

- Print out Amiga - Print out Amiga - Print out Amiga -

```
IF p=10 THEN Bonus
GOSUB Invoer
GOTO Aftreksom
```

```
Vermenigvuldigsom:
CLS:GOSUB Text:RANDOMIZE
TIMER:a=INT(RND(1)*10):RANDOMIZE
TIMER:b=INT(RND(1)*10):IF a=0 OR b=0 THEN
Vermenigvuldigsom
LOCATE 8,13:PALETTE 0,0,0,0:COLOR 5,0:PRINT
a"x"b"="
c=a*b:LOCATE 1,14:COLOR 9,0:PRINT
"Punten:"p:LOCATE 2,14:COLOR 4,0:PRINT
"-----":IF a<0 OR b<0 THEN GOSUB
Randomkeuze:GOSUB
```

```
Invoer:
IF p=-10 THEN Tienfout
IF p=10 THEN Bonus
GOSUB Invoer
GOTO Vermenigvuldigsom
```

```
Info:
CLS:COLOR 3,0
LOCATE 2,1:PRINT "Dit programma is gemaakt
door"
LOCATE 4,1:PRINT "Viktor Vos voor Commodore
Info"
LOCATE 6,1:PRINT "Gemaakt in Juni 1988."
LOCATE 8,4:PALETTE 0,0,0,0:COLOR 5,0:PRINT
"Ok":CIRCLE (7,58),5,4:PAINT (7,58),4
```

```
Check1:
IF MOUSE (0)=0 THEN Check1
IF MOUSE (1)>3 AND MOUSE (1)<12 AND MOUSE
(2)>53 AND MOUSE (2)<65 THEN Start
GOTO Check1
```

```
Stoppen:
CLS:COLOR 3,0
LOCATE 2,1:PRINT "Stoppen ?"
CIRCLE (7,26),5,4:PAINT (7,26),4:LOCATE
4,4:PALETTE 0,0,0,0:COLOR 5,0:PRINT "Ja"
CIRCLE (7,42),5,4:PAINT (7,42),4:LOCATE
6,4:PRINT "Nee"
```

```
Check2:
IF MOUSE (0)=0 THEN Check2
IF MOUSE (1)>3 AND MOUSE (1)<12 AND MOUSE
(2)>21 AND MOUSE (2)<31 THEN Einde
IF MOUSE (1)>3 AND MOUSE (1)<12 AND MOUSE
(2)>39 AND MOUSE (2)<49 THEN Start
GOTO Check2
```

```
Einde:
WINDOW CLOSE 1:WINDOW CLOSE 2:NEW
```

```
Bonus:
CLS
LOCATE 2,10:PRINT "Je hebt er 10 goed!"
FOR s=1 TO 5
SOUND 100,s
SOUND 150,s
SOUND 200,s
NEXT s
FOR s=10 TO 1 STEP -1
SOUND 200,s
SOUND 150,s
SOUND 100,s
NEXT s
FOR s=0 TO 75:SOUND 200,1,s:NEXT s
FOR s=75 TO 0 STEP -1:SOUND 200,1,s:NEXT s
GOTO Start
```

```
Tienfout:
CLS
```

```
LOCATE 2,10:PRINT "Jammer, je hebt er 10
fout!"
FOR s=150 TO 0 STEP -1:SOUND 200,1,s:NEXT s
GOTO Start
```

Turn-It

Turn-it betekent letterlijk vertaald omdraaien. Dat is het ook waar het hier allemaal omdraait. Een negental cijfers, om een vijftal witte blokjes en het geheel in een 12 tal hokjes. Dit zijn alle ingrediënten van dit spel. De bedoeling is alle cijfertjes op de juiste plaats te zetten. Dit lijkt allemaal veel makkelijker dan het is. Dit dnekspel is ontworpen en uitgewerkt door H. Vleugels uit Posterholt.

```
'Turn It <AMIGA>
'H. Vleugels, Posterholt
CLEAR ,25000&:DIM vak(4,4)
SCREEN 1,640,200,3,2
WINDOW 2,"TURN IT",(0,0)-(600,180),31,1
```

```
'vakjes tekenen
FOR x=250 TO 362 STEP 48:FOR y=55 TO 131
STEP 24
LINE (x,y)-(x+48,y+24),1,b:NEXT y,x
```

```
'aanwijzingen
LOCATE 1,1:PRINT"Zo moet het zijn!"
PRINT:PRINT" 1 2 3":PRINT:PRINT" 4
5 6":PRINT
PRINT" 7 8 9":PRINT
LOCATE 8,52:PRINT"Klikken op een gekleurd"
LOCATE 9,52:PRINT"vlak zorgt ervoor dat de"
LOCATE 10,52:PRINT"vier cijfers om dat vlak"
LOCATE 11,52:PRINT"1 keer linksom draaien."
LOCATE 12,52:PRINT"De inhoud van de vlakjes"
LOCATE 13,52:PRINT"onderaan is onzichtbaar."
LOCATE 20,2:PRINT"<ESC> = stop"
RANDOMIZE TIMER
RESTORE
```

```
GetalBepalen:
FOR x=1 TO 3:FOR y=1 TO 3
getalBepalen2:
getal=INT(RND(1)*9+1):getal(getal)=getal(getal)
+1
IF getal(getal)>1 THEN getal(getal)=1:GOTO
getalBepalen2
vak(y,x)=getal:NEXT y,x
```

```
DATA
9,34,12,34,15,34,9,40,12,40,15,40,9,46,12,46,15
,46,18,34,18,40,18,46
FOR x=288 TO 288+48 STEP 48:FOR y=74 TO
74+24 STEP 24
LINE (x,y)-(x+20,y+10),1,bf:NEXT y,x
LINE (336,122)-(356,132),1,bf
GOTO invullen
```

```
checkmouse:
LOCATE 3,35:PRINT" KLAAR "
a$=INKEY$
IF a$=CHR$(27) THEN verder
MOUSE ON
IF MOUSE(0)<>1 THEN checkmouse
x=MOUSE(1):y=MOUSE(2)
MOUSE OFF
LOCATE 3,35:PRINT"W A C H T E N"
```

```
'vakje aanklikken
```

- Print out Amiga - Print out Amiga - Print out Amiga -

```

IF x>288 AND x<308 AND y>74 AND y<84 THEN
  een
IF x>336 AND x<356 AND y>74 AND y<84 THEN
  twee
IF x>288 AND x<308 AND y>98 AND y<108 THEN
  drie
IF x>336 AND x<356 AND y>98 AND y<108 THEN
  vier
IF x>336 AND x<356 AND y>122 AND y<132 THEN
  vijf
GOTO checkmouse

'getallen verplaatsen
een:
hulp=vak(1,1):vak(1,1)=vak(1,2):vak(1,2)=vak(2,
2):vak(2,2)=vak(2,1):vak(2,1)=hulp
GOTO invullen

twee:
hulp=vak(1,2):vak(1,2)=vak(1,3):vak(1,3)=vak(2,
3):vak(2,3)=vak(2,2):vak(2,2)=hulp
GOTO invullen

drie:
hulp=vak(2,1):vak(2,1)=vak(2,2):vak(2,2)=vak(3,
2):vak(3,2)=vak(3,1):vak(3,1)=hulp
GOTO invullen

vier:
hulp=vak(2,2):vak(2,2)=vak(2,3):vak(2,3)=vak(3,
3):vak(3,3)=vak(3,2):vak(3,2)=hulp
GOTO invullen

vijf:
hulp=vak(3,2):vak(3,2)=vak(3,3):vak(3,3)=vak(4,
3):vak(4,3)=vak(4,2):vak(4,2)=hulp
GOTO invullen

verder:
LOCATE 21,33:PRINT"NOG EENS ? [j/n]"

verder2:
a$=INKEY$
IF a$="" THEN verder2
IF a$="j" THEN RUN
IF a$="n" THEN WINDOW CLOSE 2:SCREEN CLOSE
1:END
GOTO verder2

invullen:
RESTORE
FOR a=1 TO 4:FOR b=1 TO 3:READ p,q:LOCATE
p,q
IF vak(b,a)=0 THEN PRINT " ":GOTO
volgendvak
PRINT vak(b,a)
volgendvak:
NEXT b,a

'kontrole
kontrole=0
FOR a=1 TO 3:FOR b=1 TO 3
IF vak(a,b)=3*(a-1)+b THEN
kontrole=kontrole+1
NEXT b,a
IF kontrole=9 THEN verder

GOTO checkmouse

```

Type les

Types blijft de meeste programmeurs boeien. Sinds de firma RadarSoft het programma Tempo-

typen op de markt heeft gebracht zijn op dit thema al vele variaties bedacht. Ook dit programma is hier een voorbeeld van, we wilden U het echter niet onthouden. De maker hiervan is W.P.Th van de Rijke uit Den Burg.

```

"W. P. Th v.d Rijke
'Den burg texel
SCREEN 2,600,200,2,2
WINDOW 2,"TYPES",,,2
CLS:g=0:r=0:w=0:p=50:u=0:k=0 :v=0
DIM b$(20),a$(150)
FOR z=1 TO 100
  READ d$
  a$(z)=d$
NEXT z
PRINT STRING$(136,42)
LOCATE 2,30:PRINT "TYPES"
PRINT STRING$(136,42)
LOCATE 4,5:PRINT"het is de bedoeling om
het opgegeven woord in te type"
LOCATE 5,5 :PRINT"voordat het de
laserstraal raakt"
LOCATE 6,5:PRINT"Na 10 woorden volgt het
volgende level"
LOCATE 7,5:PRINT"Na 5 keer de
laserstraal te hebben geraakt volgt de
LOCATE 8,5:PRINT "uitslag hoe goed je
bent"
LOCATE 9,5:PRINT" druk op RETURN voor te
starten"

begin:
l$=INKEY$
IF l$=CHR$(13) THEN opbouwscherm
GOTO begin

opbouwscherm:
CLS
GOTO nieuwscherm

spelscherm:
CLS
LOCATE 4,5:PRINT STRING$(90,45)
LOCATE 4,6:PRINT "FINISH"
LOCATE 4,34:PRINT "START"
LOCATE 18,5:PRINT STRING$( 90,45)
FOR a=4 TO 18
  LOCATE a,9:PRINT CHR$(124)
NEXT a
RETURN

hoofddlus:
y=INT(RND*100)+1
IF g=10 THEN nieuwscherm
FOR a=1 TO LEN(a$(y))
  b$(a)=MID$(a$(y),a,1)
NEXT a

s=LEN(a$(y))
FOR a=37-(v+LEN(a$(y))) TO 0 STEP -1
  LOCATE 1,2:PRINT "Aantal goede
woorden";r
LOCATE 2,2 :PRINT"Aantal foute
woorden";w
LOCATE 2,30:PRINT"Level";u
IF a=4 THEN wlaserstraal
LOCATE 11,a:PRINT RIGHT$(a$(y),s+1);"

"
IF INKEY$=b$(k) THEN GOTO con1
FOR t=0 TO p
  IF INKEY$=b$(k) THEN GOTO con1
NEXT t

r1:  NEXT a

```


- Print out Amiga - Print out Amiga - Print out Amiga -

```

con1:
  BEEP:s=s-1:k=k+1:a=a+1
  IF s<0 THEN GOTO con2 ELSE r1

con2:
  LOCATE 11,a-1:PRINT "
":s=0:k=0:g=g+1:r=r+1:GOTO hoofdlus

nieuwscherm:
  CLS
  u=u+1:g=0
  p=p-5
  IF p=-5 THEN CLS:GOTO speleinde
  LOCATE 2,30:PRINT "level";u
  FOR t=1 TO 999:NEXT t
  v=v+2
  GOSUB spelscherm :GOTO hoofdlus

wlaserstraal:
  w=w+1:IF w=5 THEN speleinde
  LOCATE 12,3:PRINT CHR$(124)
  SOUND 6,20:SOUND 7,7:SOUND 9,14:SOUND
13,14
  FOR a=LEN(a$(y)) TO 0 STEP -1
  LOCATE 11,4:PRINT STRING$(a,42);" "
  FOR t=1 TO 50:NEXT t
  NEXT a
  SOUND 9,0
  CLS
  PRINT:PRINT:PRINT"U heeft het niet
gehaald"
  PRINT"druk op return voor het volgende
woord "

wachten:
  l$=INKEY$
  IF l$=CHR$(13) THEN GOSUB
spelscherm:k=0: GOTO hoofdlus
  GOTO wachten

speleinde:
  CLS
  LOCATE 7,3:PRINT"Aantal goede
antwoorden";r
  LOCATE 9,3:PRINT"Aantal foute
antwoorden";w
  IF w=0 THEN LOCATE 11,3:PRINT"U kunt uit
stekend type"
  IF u<3 THEN LOCATE 10,3:PRINT"U bent
verschrikkelijk slecht!"
  IF u>3 AND u<6 THEN LOCATE
10,3:PRINT"U bent slecht"
  IF u=6 OR u=7 THEN LOCATE 10,3:PRINT
"U typt al aardig"
  IF u=8 OR u=9 THEN LOCATE
10,3:PRINT"Heel redelijk"
  IF u>9 AND u<11 THEN LOCATE
10,3:PRINT"U typt snel"
  IF u=11 THEN LOCATE 10,3:PRINT "U typt
heel snel"
  LOCATE 14,4:PRINT"druk op return voor
nog eens"

wachten1:
  l$=INKEY$
  IF l$=CHR$(13) THEN
p=50:g=0:r=0:w=0:u=0:u=1:k=0:v=0:CLS:GOSUB
spelscherm:GOTO hoofdlus
  GOTO wachten1

' WOORDENSCHAT
DATA
waterval, fietsband, joystick, kachel, computer, boe
kenkast, radio, eric, gijsbert, cassette

```

```

DATA
groeien, schip, televisie, document, domino, disco, d
ialoog, deur, chequeu, citroen
DATA
appelboom, kachelpijp, machine, aanrecht, kalender,
poster, enigszins, plaat, begraven, verraden
DATA
luidspreker, station, piloot, moter, cabine, canada,
chinees, begonia, bandepesch, balkon
DATA
apotheek, gehoor, gember, video, haven, types, toets
en, inhoud, kattedak, redaktie
DATA
koken, lamp, schrift, leeuw, tijger, neus, fiets, stek
ker, zalm, vis
DATA technicus,
verlieft, suiker, autodeur, trap, bel, water, zeehond
, wolf, luifel
DATA
houtworm, bijl, poep, makelaar, examen, stabiel, stem
pel, tent, telefoon, vaandel
DATA
degelijk, hopelijk, mogelijk, tezamen, hart, melkbus
, verrasing, moord, adellijk, achttien
DATA
geschied, platform, wiskunde, vlees, fruit, tafel, so
ep, schakelaar, wekker, school
DATA
stiefvader, vuur, menu, jas, cycloon, transistor, dio
den, tabak, lepel, toinen

```

Woorden raden

Dit spel van H. Vleugels uit Posterholt laat zien dan ook een eenvoudig programma zeer verslavend kan zijn. Er moeten woorden worden gemaakt, door middel van het intikken van letters die door de computer op de juiste plaats worden gezet. Dit programma is door zijn eenvoud zeer geschikt voor de wat kleinere kinderen.

'Woorden raden <Amiga>

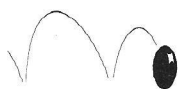
```

'H. Vleugels, Posterholt
presets:
CLS:RESTORE:qq=6:DIM a$(50,1):'max. 50
letters in een woord

tal=35:'aantal woorden in dataregels
DIM woord$(tal)
FOR q=1 TO tal:READ woord$(q):NEXT q
'woord uitzoeken in dataregels
RANDOMIZE TIMER
a$=woord$(INT(RND(1)*tal)+1):a=LEN(a$)
DIM b$(a)
b$(1)=LEFT$(a$,1):b$(LEN(a$))=RIGHT$(a$,1)
FOR x=2 TO LEN(a$)-1:b$(x)="":NEXT x
GOSUB schrijven

'letter raden
invoer:
IF c$=a$ THEN einde
LOCATE 10,20:PRINT"
"
LOCATE 8,20:PRINT"Welke letter (of woord)
invullen":LOCATE 10,20:INPUT x$
IF x$=a$ THEN einde
FOR x=2 TO a-1

```

REAL3D

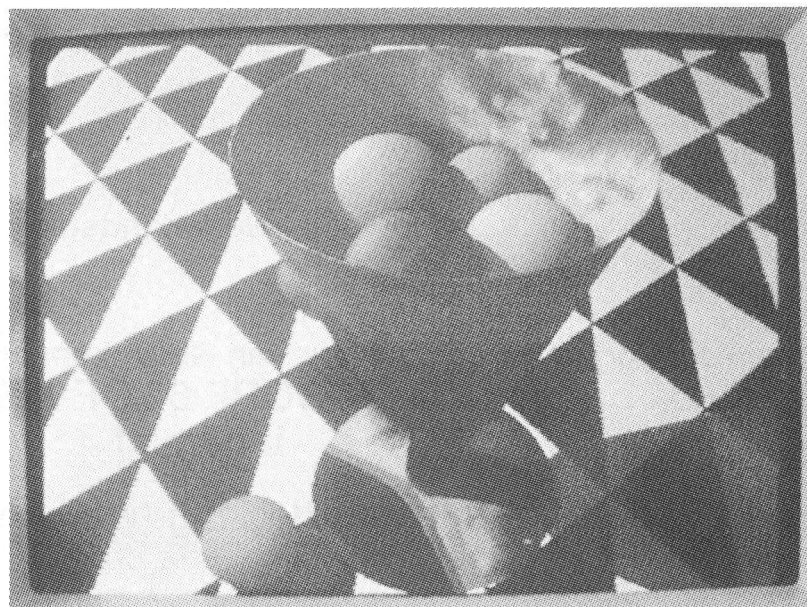
Drie-dimensionaal raytracing- en animatiepakket

Het maken van 3D grafische ontwerpen en animaties op de Amiga wordt steeds belangrijker. Desktopvideomakers, reclame-ontwerpers, zakelijke en onderwijspresentaties blijken allemaal een onverzadigbare honger voor 3D-plaatjes te hebben. Helaas blijkt het ontwerpen van dergelijke 3D raytracing graphics meestal niet zo eenvoudig als de softwareverpakking wil doen geloven. Bovendien kosten het optekenen en de animaties in de praktijk dikwijls veel tijd. Een snellere Motorola 680X0 CPU en veel RAM kunnen daarbij helpen, maar niet iedere Amigabezitter zit even ruim bij kas. Het door 3Gitaal geïmporteerde pakket REAL3D pretenteert zowel eenvoudig als redelijk geprijsd te zijn. In de eenvoudigste beginnersuitvoering kost REAL3D f 449,- en draait op elke Amiga met minimaal 1 MB aan vrij RAM in de systeemkast.

Software uit Finland zien we niet veel in Nederland. Dat men in noordelijker regionen toch aardige grafische software kan maken bewijst het pakket **REAL3D** van Activa Software Oy. REAL3D wordt aangeprezen als een drie-dimensionaal raytracing, solid modeling en animatiepakket. Deze hele mondvol aan technische termen betekent gewoon dat mogelijk zijn:

- ° Projectie van figuren en objecten op de X-, Y- en Z-as
- ° Simulatie van natuurlijke (zo als het menselijk oog de reflecties in de ruimte ziet) lichtval. De lichtbron is geheel instelbaar
- ° Naast draadmodellen kan men de vlakken vullen, solid modeling, met een bepaalde structuur, texture mapping, of gewoon egaal houden
- ° Er zijn diverse animaties van de getoonde objecten mogelijk

Dit alles is reeds mogelijk op een standaard Amiga met slechts één MegaByte aan vrij RAM. Wie meer en sneller wil zal in een krachtiger Amiga en duurdere REAL3D-versie moeten investeren.

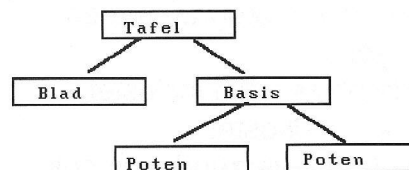


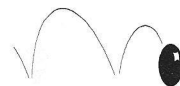
Basics

In principe is REAL3D een raytracing rendering programma. **Rendering** betekent dat het tekenpakket uit gaat van een aantal vaste vormen (cirkels, cylinders, kogels vier- en driehoeken, ookwel primitieven genoemd). Met die vaste vormen worden gecompliceerde grafische objecten opge-

bouwd die dan weer gevormd, geroteerd, anders belicht, geanimeerd e.d. kunnen worden.

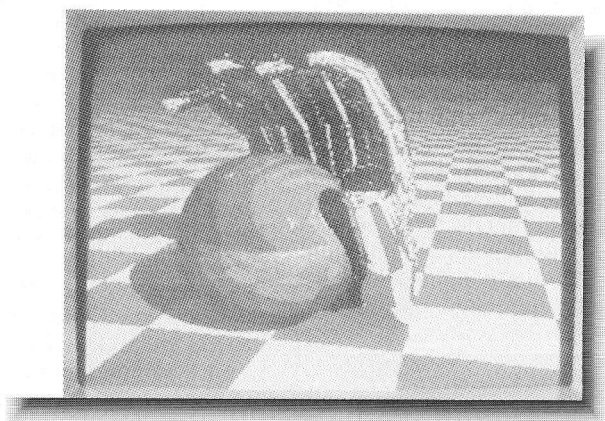
De rendering-aanpak heeft voor- en nadelen. Als belangrijkste voordeel geldt de snelle verwerking van de objecten binnen REAL3D zelf. Het opbouwen van complexe figuren gaat snel en comfortabel, terwijl tal van bewer-





kingen mogelijk zijn. Als na-
deel geldt het gebruik van
graphics uit andere tekenpak-
ketten. Het importeren van
.IFF-plaatjes wil nog wel
goed lukken, maar de bewer-
kingen blijven beperkt tot ro-
tatie, vergroten/ verkleinen
en vervormen. Punkt- en po-
lygoonmanipulaties bleken
in de door ons bekeken dem-
oversie helaas niet mogelijk.
Wellicht beschikt een vol-
gende versie wel over deze
opties.

REAL3D is volledig HAM-
compatibel. Uit het kleuren-
palet van 4096 zijn maximaal
16 kleuren bruikbaar.



Opstarten

Bij het opstarten van de vol-
ledige versie verschijnen de
ikonen voor REAL3D zelf,
de Display, de Realplay, de
.IFF-convert- en -animatie-
utilities, de animaties zelf,
materialen, textures, overige
utilities, attributen en prefe-
rences.

Door op het REAL3D-ikoon
in te klikken worden de **edi-
tor-windows** geopend. Er
zijn drie **projection wind-
ows**: voor het voor-, zij- en
bovenaanzicht. Verder zijn er
een **selectie-** en een **instruc-
tievenster** zichtbaar.

De menu's worden uitgetrok-
ken door op het instructie-
venster, bovenaan in de dis-
play, in te klikken. Dan ver-
dwijnen de muis-coördina-
ten en komen de bekende
Amiga-trekmene's daarvoor
in de plaats. REAL3D is vrij-
wel volledig muis-menuge-
stuurd en dat blijkt in de

praktijk best lekker te wer-
ken.

Hiërarchie

Het tekenen van de grafische
objecten begint in de **editor-
modus**. Met behulp van het
creationmenu wordt een be-
paalde primitief gekozen. Er
is keuze uit een rechthoek,
driehoek, cirkel, prisma, ku-
bus, pyramide, bol (sferoid),
conus, cylinder, hyperbool en
cut-hyperbool. Daarmee zijn
snel en gemakkelijk comple-
xe figuren te construeren.

Na de keuze van het gewen-
ste primitief verschijnt deze
in de root van het selectie-

venster. Zo behoudt de ge-
bruiker het zicht op het aantal
en de verschillende primitief-
typen.

De handleiding geeft een
goed voorbeeld over hoe een
tafel ontworpen kan worden.
Eerst kiest de gebruiker een
bepaalde kleur. Daarna kiest
hij uit de primitieven de ku-
bus voor het tafelblad van op-
zij gezien. De naam wordt in
het selectievenster in tafel-
blad veranderd. Nu moet het
tafelblad nog op een basis en
de tafelpoten steunen. Ook
deze objecten worden met
behulp van vervormde ku-
bussen geconstrueerd.

Alle vervorm-, verplaatsings-,
roteer-, explode, copy-, re-
name-, color-, material-, de-
lete- en paintopdrachten zijn
onder het Modify-Hiërar-
chiemenu gerangschikt.
Draadmodellen

Wireframes worden bij 3D-
pakketten gebruikt voor twee
belangrijke doeleinden:

- ° Voor het snel in real-time
kunnen bekijken van de
modellen in de X-, Y- en
Z-richting

- ° Als basis voor de latere so-
lid modeling

Het **wireframe window** be-
schikt over drie belangrijke
view-tools. Met **distance**
geeft de gebruiker aan van
hoever het object bekeken
gaat worden. Hoe dichterbij
des te groter het object en des
te sterker het perspectief

De **direction** geeft de "opna-
merichting" van het kamera-
oog aan. Binnen het scherm-
vierkant kunt u het draadmo-
del vanaf elke positie op één
van de vier zijden bekijken.
Met **screen** kan de grootte
van het plaatje veranderd
worden. Dit heeft geen (!) ef-
fect op het perspectief.

Met behulp van het wirefra-
me-window kan het model
vanuit alle richtingen, in ver-
schillende perspectief en bij
verschillende groottes beke-
ken worden. Ook worden
draadmodellen gebruikt om
de animatiebanen en het ty-
pebeweging te beschrijven.

Solid Modeling

Het feitelijke renderingpro-
ces schuilt onder **solid mode-
ling**. Vanuit het wireframe-
venster wordt op de optie
SOLID geklikt. Er verschijnt
een uitgebreid menu met een
groot aantal rendering- con-
trols.

De **baselight-control** bepaalt
hoe helder de objecten in de
schaduw zullen zijn. Hoe ho-
ger de intensiteit van deze pa-
rameter voor diffuus gere-
flecteerd licht des te minder
contrast er tussen objecten in
het licht en in de schaduw be-
staat.

De **brightness-control** regelt
de lichtbron(nen) af op een
bepaald niveau. Overigens
doet de editor dit ook al auto-
matisch en behoorlijk accu-
raat.

Met de **background-control**
worden de RGB-kleuren elk
via een afzonderlijke regelaar
ingesteld. Dezelfde RGB-re-

gelaars dienen ook voor de
kleurstelling van het base-
light.

De **overlight-control** defini-
eert wat er in de heldere
beelddelen aan overbelich-
ting (wit) plaatsvindt. Bij de
waarde 0 vindt er geen over-
belichting plaats. Hogere
waarden geven steeds meer
witte uitgebleekte vlakken.
De voornaamste functies van
deze regelaar zijn het laten
glansen van matte objecten
en speciale beeldvervreem-
dingen.

De **recursion-depth-control**
bepaalt hoe vaak de reflecties
van invallende lichtstralen in
beeld komen. Neemt u bij-
voorbeeld de waarde 5 dan
zorgt deze raytracingtechniek
er voor dat alleen de eerste
vijf weerkaatsingen van een
invallende lichtstraal zicht-
baar zijn. Een belangrijk pro-
bleem met raytracing is dat
het uitrekenen van al die na-
tuurlijke reflecties flink wat
tijd kost. Hoe meer reflecte-
rende en doorzichtige opper-
vlakten er aanwezig zijn des
te meer reflecties en des te
langer de rekentijd wordt.
Met recursion-depth valt
deze rekentijd te bekorten. Te
weinig reflecties geeft echter
een onnatuurlijk effect.

Met de **frame-control** kunt u
elk animatieframe afzonder-
lijk benaderen. Hierbij wordt
gebruik gemaakt van frame-
fields.

Onder de **mode** staan de vol-
gende instellingen:

- ° **Faste mode**; snelle een-
voudige optekening zonder
raytracing (geen spiegeling
of doorzichtigheid) en
geen oppervlakte-structuur.
Deze mode is met name
geschikt als preview-optie

- ° **Normal mode**; maximale
rendering met alle reflec-
ties, schaduwen, structuren
enz.

- ° **Mode shadowless**; snelle-
re modus zonder schadu-
wen voor o.a. animaties

- ° **Mode lampless**; een niet
geheel korrekte term want
er wordt wel één automa-
tische lichtbron gebruikt



° **Mode outline** geeft een rendering in slechts twee kleuren

Onder de options van het Solid Model vindt u verder:

° **Single**; alleen het actieve animatierframe wordt gerenderd

° **Autolight** voor automatische afregeling van het lichtniveau op de overall brightness van het object

° De **videomodi** interlace, overscan en greyscale (640 horizontale grijze pixels i.p.v 320 in kleur)

° De **dithering**

° **SAVEMEM** maakt CHIP-memory vrij voor multitasking-toepassingen

Het renderingproces wordt gewoon door inklikken op de "RENDER-toets" rechtsonder in het userinterface gestart.

Materialen

In een natuurlijke omgeving wordt de visuele intruk van solide objecten bepaald door hun helderheid, kleur, oppervlakte structuur, doorzichtigheid (en reflectie), het gedrag van het licht in het materiaal (refractie) en de lichtinval. Wat het materiaal zelf betreft hebben we met de natuurkundige principes van reflectie, absorptie en penetratie (doorvallen) te maken. Met behulp van de REAL3D-tools zijn al deze fysische eigenschappen instelbaar.

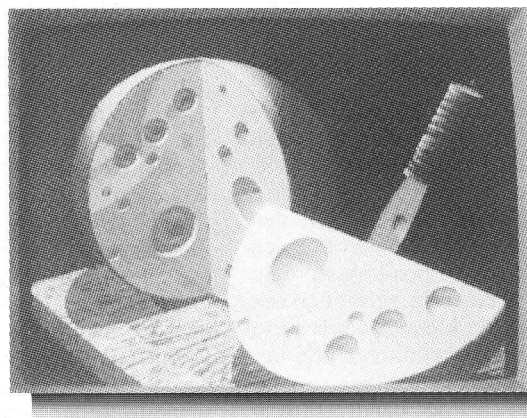
Via de **projection-optie** kan een bepaald structuuroppervlak op het object worden aangebracht. Er is keuze uit een parallelle, cilindrische, sferische en spiraalvormige projectie. Via de tilefunctie (dakpantentechniek) kunnen deze projectievormen veelvuldig op één en hetzelfde oppervlak herhaald worden. De gebruiker kan zelf met behulp van tekenprogramma's structuren voor texture mapping creëren. Het aanleggen van texture-libraries is mogelijk.

Bij elk materiaal kunnen de briljance, de doorlaatbaarheid (transparency) en licht-

snelheid in het weefsel gedefinieerd worden. Met de painting functien is een exacte positionering van de structuren mogelijk.

Logische operaties

De schitterende 3D-creaties met uitgeholde oppervlaktes, gaten en scherpe aansnedes worden meestal gemaakt met behulp van zogenaamde **logische operatoren**. Deze operatoren zijn de AND-, AND NOT-, EOR- en DEVIDE-operatoren. Met behulp van deze logische operatoren is het mogelijk gestructureerde objecten bij elkaar op te tellen, van elkaar af te trekken, met elkaar te vermenigvuldigen en door elkaar te delen. Een goed voorbeeld van een dergelijke logische operatie is de constructie van een optische lens. Maak twee gelijke bollen en laat deze elkaar snijden (intersecting). Met de tools kunnen daar dan gaten in geboord of delen uitgesneden worden.



Animaties

Bij REAL3D worden diverse roots, met elk één frame, aan elkaar gekoppeld. Elk met REAL3D ontworpen beeld is in feite een enkelbeelds-animatie. Door de frames achter elkaar af te spelen ontstaat de indruk van een vloeiende beweging.

REAL3D biedt een aantal geavanceerde en eenvoudige animatietechnieken. De eenvoudigste animatie-functies bestaan uit:

° **Orbit**; teken een baan op het scherm en de computer berekent een vloeiend bewegingspatroon voor het te animeren object

° **Rotatie**; het voorwerp roteert volgens vooraf ingestelde specificaties om zijn as

REAL3D zorgt automatisch voor het gladmaken van de ruw in de beeldvensters getrokken animatie-banen.

Versies

Het 3D raytracing-pakket REAL3D wordt in drie verschillende versies geleverd. De **beginnersversie** draait op elke Amiga 500, 1000, 2000, 2500 of 3000 met minimaal 1 MB aan boord. Deze versie kost f 449,- en ontbeert de logische Boleaanse functies en de makro-optie.

De **REAL3D professional uitvoering** biedt wel de Boleaanse functies plus de makro-opname-eenheid. Met dit makro-utility kunnen bepaalde functies opgenomen en op

Toepassingen en praktijk

REAL3D mikt op verschillende gebruikersgroepen. Bijvoorbeeld de makers van videographics, video-animaties, computergraphics, presentaties, desktopvideoproducties en interactieve animatiefilms. Voor al deze toepassingen biedt dit raytracing-pakket ruim voldoende mogelijkheden. Echt moeilijk is het niet maar u dient wel over het nodige talent te beschikken en flink te oefenen. Kwaliteit kost echt tijd! Onze onvolledige testversie bleek behoorlijk snel te zijn. De logische object-operaties bieden een schat aan realistische mogelijkheden. De realtime en interactieve animaties in overscan zullen menige dtv-er en animatiemaker aanspreken. Alle ditors en vensters werken relatief eenvoudig en overzichtelijk.

De beperkingen zitten hem voornamelijk bij de import van andere graphics. Het bewerken van .IFF-beelden is beperkt en ANIMOPT5-import zat er ook niet op. Ook konden wij geen anti-aliasing-optie ontdekken.

Wat de amateurversie betreft zijn prijs en prestatie zondermeer in evenwicht. Bij de proversie kunnen de genoemde beperkingen een bezwaar zijn. Niettemin een goed 3D raytracing-pakket.

Inlichtingen: 3Gitaal, tel.:020-970035.

U.S.

verzoek als één geheel herhaald worden. Ook worden meer diskmaterialen, structuren en voorbeelden meegeleverd. De REAL3D proversie vereist 2 MB, een harde schijf en een uitgave van f 979,-.

De **Turboversie** van f 1.269,- geeft flink gas als u over een Amiga met 68020/30 en mathematische coprocessor beschikt. Gezien de prijs meer bedoeld voor de power user van raytracing-technieken.

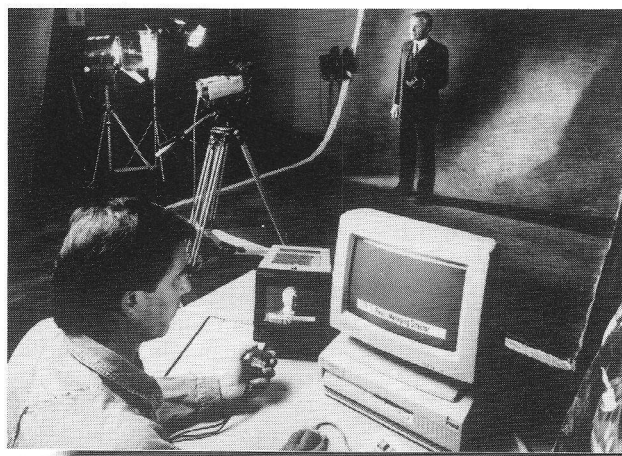
AMIGA 90

Verslag van de grote Amiga-beurs in Keulen

Is een Amiga-beurs van enige omvang in Europa nog mogelijk? Het antwoord luidt na een bezoek aan de beurshallen in Keulen ronduit ja. Kwamen er vorig jaar reeds een 35.000 bezoekers. Dit jaar werd dit aanbod ruimschoots overtroffen. Toch had de beurs niet de omvang waarop in aankondigingen gerekend werd. In 1989 betrof het vloeroppervlak alleen Halle 6. Nu waren de Halle 10 en 12 aan de beurt, maar de Halle 13 uit eerdere persaanmeldingen ontbrak. Ook waren het aantal echte nieuwigheden op de vingers van één hand te tellen. De Amiga '90 bleek meer een amateurkoop- dan een nieuwtjes- og vakbeurs te zijn. Ruim 160 deelnemende handelaren boden een breed aanbod. Alleen waren de meeste kraampjes aan de krappe kant zodat bij grootte drukte moeilijk te zien was wat er nu eigenlijk in "Sonderangebot" en "Messepreisse" verkrijgbaar was.

Amateur- of vakbeurs?

Wat de Commodore Amiga betreft proef je onder het



beursoppervlak altijd een soort scheiding tussen amateur- en professionele toepassingen. In tegenstelling tot de IBM-compatibele PC heeft de Amiga zowel een "Spelletjes"- als "professioneel grafisch"-imago. Ook Commodore zelf heeft het daar wel eens moeilijk mee. Op wie moet je nu mikken? Op professionele dtv-ers, dtp-ers, UNIX- gebruikers of die vele tienduizenden Amiga-amateurs. Ons inziens beiden, maar in Nederland wordt daar wel eens anders over gedacht.

Ook in Keulen hing deze scheiding onmiskenbaar in de lucht. Het grote publiek bestond uit de koopjesjagers terwijl de organisatoren de zaak met seminars en een heuze vakhandelarendag de

zaak professioneel probeerden aan te kleden. Tevens trachtte men met de halindelinge een scheiding aan te brengen. In hal 10.2 moest het meer professionele hardware- en softwaregebeuren plaatsvinden terwijl hal 12.2 meer op de vermaakstoer ging. Niettemin werd ook in hal 10.2 tegen stuntprijzen aan amateurs verkocht.

Nieuwigheden

Zoals reeds eerder opgemerkt werd viel het aantal nieuwigheden bar tegen. Ralf Jochem bleek eindelijk zijn **Higgraph V** grafiekkart in productie genomen te hebben. Deze kaart biedt 31,5 KHz VGA-kwaliteit bij een maximaal oplossendvermogen van 832 x 620 non-inter-



laced pixels. Het Deluxe Paint III PAL- overscanformaat van 704 x 580 pixels toont echt prima op een multisyncmonitor. Dankzij een eigen 768 KB videogeheugen zijn daadwerkelijk maximaal 4.096 kleuren instelbaar. Op de Amiga '90 ging deze kaart voor minder dan DM 600,- over de toonbank. Hopelijk komt deze veelbelovende videokaart ook voor een dergelijke prijs in Nederland beschikbaar.

Op de Commodore-stand vielen een nieuwe **inkjet-** en **laserprinter** te bewonderen. Newtek pakte op deze stand groots uit met de reeds jaren in de publiciteit zijnde **Videotoaster**. De getoonde video-effecten waren heel indrukwekkend. Kopen is er helaas nog niet bij want het gaat nog altijd om een Amerikaanse NTSC-versie!

Voor de MS-DOS-ser levert Commodore nu de **A 2286 AT-kaart** met DOS- release 4.1 uit. Beter laat dan nooit niet waar? De nieuwe **1930 VGA** en **1950 Multisync-monitoren** waren ons reeds van de Efficiency Beurs bekend.

De CD-ROM-afspeler voor de Amiga zet langzaam maar zeker door. Het aanbod aan drives en software groeit gestaag. Het Commodore Dynamic Total Vision systeem konden wij tijdens ons beurbezoek niet vinden. Het blijft zeer de vraag of CDTV dit jaar Europa nog haalt.

Verder een wat bescheiden tamtam over de de Amiga 3000 met UNIX V release 4.X.

Het moet allemaal natuurlijk steeds sneller. Een MC 6830 CPU op 50 MHz is al geen uitzondering meer. De turbo-kaarten en SCSI- harddisks waren dan ook ruimschoots vertegenwoordigd. GVP, Kupke, Supra, MAST, X-pert transputers met Parallel C-assembler en BSC ALF pakten weer ruimschoots uit. Een betrekkelijke nieuwkomer was de Stormbringer 530 68030 turbokaart voor de Amiga 500/2000. Toch ging

het ook hier weer om reeds eerder getoonde produkten. Hetzelfde gold de dtv-hardware. Alle Videocomp-, Merken-, Alcomp, Electronic Design- en Newtek-produkten waren al op de Photo-Kina '90 getoond.

Wel nieuw was de Vortex **AT-emulator ATonce** voor de Amiga 500. Deze AT-emulator draait onder MS-DOS 2.11 en een kloksnelheid van 8 MHz. Opmerkelijk is dat deze kaart het gewone 3.5 inch diskteststation van de Amiga 500 kan gebruiken.

De in Nederland onbekende fabrikant X-pert toonde het **EGS** (Extended Graphics System) met een on-board video-RAM van 2 tot 4 MB en een resolutie van 640 x 480 tot 1280 x 1024 pixels. Deze videokaart gaat in een gewoon Amiga-slot. Ook liet X-pert een nieuw cache-board, genlock en Nibble-mode RAM-kaart zien.

Multimedia en kunst

De meeste seminars gingen op een of andere manier over de Amiga als multimedia-computer en kunstenaar. Uiteraard niet ten onrechte want op deze beide terreinen slaat de Amiga alles in zijn prijsklasse.

DTV, grafische en animatie-toepassingen, DTV, MIDI, Amiga Vision en Amiga-artwork stonden flink in de belangstelling. Er was zelfs een goede tentoonstelling van Amiga-art-graphics. Verder was er in de seminars aandacht voor de virus-ellende en softwarepiraterij.

Wat **multimedia** precies inhoudt kwam nog niet echt uit de verf. Alexander Glos, zakelijk leider van AMI Shows Europe GmbH, bleef ook stecken in kreten als "De informatiebron van Morgen", "De drempel van een nieuwe informatietechnologie" en de "Amiga als meest economische en krachtigste multimedia-PC".

Wolfram Höfler van Markt & Technik had het over de in-

valshoek: "Amiga- Een produkt zonder een levenscyclus?" Daar werd weer eens benadrukt wat een professioneel en veelzijdig produkt een Amiga toch is. Door zich op de professionele markt te stabiliseren heeft de Amiga zich in Amerika, Duitsland en Europa en vaste positie voor de jaren 90 verworven. Kortom, weg met het spelletjes imago. De goede man vergat daarbij wellicht het eigen Markt & Technik Amiga Computermagazin dat toch duidelijk meer op de hobby- dan professionele markt georiënteerd is.

Netwerken

De Amiga kan zich niet meer aan het LAN-gebeuren onttrekken. Er is inmiddels keuze uit:

- ° **Amiganet 2.0** met een OSF/MOTIV achtig-interface
- ° **ACD-Amiga LAN** voor koppeling aan DEC-, Apple-, IBM- en UNIX-computers
- ° Appletalk
- ° Novell Netware
- ° TSS-net

MIDI-LAN

Deze netwerkconfiguraties zijn o.a. aantrekkelijk voor dtp- en dtvers.

Koojes-aanbod

De attractieve koopjes lagen vooral op het terrein van geheugen, harddisks, communicatie en dtv. Geheugenkaarten met 2 MB aan boord gingen voor rond de DM 500,- over de toonbank. Een kwalitatief goed 2400 baud modem werd het eigendom van een Amiga-gebruiker voor circa f 400,-.

Digiview 4.0 kostte op de Amiga '90 slecht f 240,-. Een Y/C-genlock van Electronic Design minder dan f 1250,-.

De hardcards bleken heel betaalbaar. 20 MB voor ongeveer f 650,-. 30 MB voor f 770,- en 40 MB voor rond de f 990,-.

Verder was de Highgraph V videokaart in een zeer voor-

delige aanbieding. De gemiddelde software prijzen waren in het algemeen niet veel lager dan in Nederland.

Boeken

Het aanbod aan specialistische Amiga-literatuur op de Amiga '90 viel groot te noemen. Naast DataBecker stonden daar ook de kleinere uitgeverijen zoals Gabrielle Lechner, Laterna Magica, de tijdschriftuitgevers (o.a. Markt Technik en Amiga World) en Ostermann.

Hier kon de liefhebber ruimschoots terecht voor in Nederland moeilijk verkrijgbare onderwerpen zoals desktop-video en handleidingen voor teken- of animatiepakketten. Helaas alleen in de duitse taal.

De organisatoren hoopten met de Amiga '90 tussen de 50.000 en 60.000 Amiga-fans te zullen trekken. Wij hebben hen niet kunnen tellen en zijn ook in het drukke weekend van 10 en 11 November uit de buurt gebleven. Het lijkt er in ieder geval op dat deze Europese Amiga-expo in een ruime behoefte voorziet. Daarbij denken wij dan met name aan koopjes, het vinden van moeilijk verkrijgbare Amiga-artikelen en kontakten tussen Amiga-gebruikers onderling. Een echte vakbeurs viel de Amiga '90 ondanks de seminar-aankleding nog niet te noemen. Wellicht haalt de Amiga '91 deze nominatie wel.

U.S.

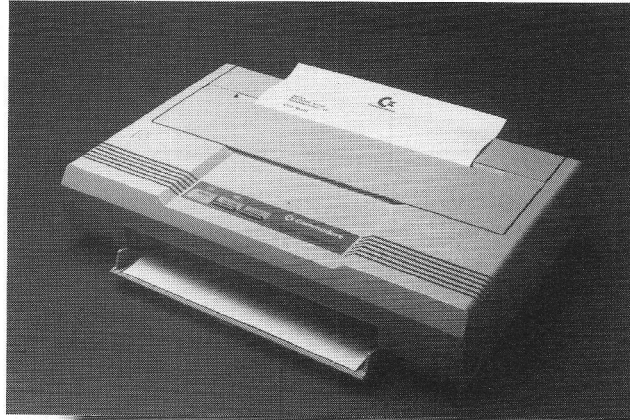
NIEUWS UIT KEULEN

Electronic Design DTV-hardware

Het Münchense dtv-bedrijf **Electronic Design** (Steffen Kramer) biedt momenteel een ruime keuze uit dtv-randapparatuur voor de Commodore Amiga. Enkele van hun genlocks en digitizers worden ook onder andere merknamen verkocht. O.a. De Hama S-590 Genlock.

Het topmodel is de Y/C-genlock met ingebouwde RGB-splitter en tevens bruikbaar als RGB-converter of PAL-encoder. Deze genlock biedt de video-amateur die met S-VHS en Hi8 werkt semi-professionele mogelijkheden op de Amiga. Titelsuperimpose, fade/in fade out met naar keuze computergraphics in de voor- of achtergrond, sleutelgateffect en digitizing in kleur zijn eenvoudig uitvoerbaar. De Y/C-kwaliteit met 5,5 MHz PAL-video en de RGB-kwaliteit van meer dan 10 MHz zijn zondermeer goed te noemen. Er zijn gescheiden draairegelaars voor helderheid, contrast, kleur, fader en computermodus (voor/achtergrond). Wat de aansluitingen betreft beschikt deze Electronic Design-genlock over zowel Hosiden Y/C- als standaard CVBS-tulp-aansluitingen. Er is een aparte mono-tulpbus voor gebruik als RGB-splitter. Deze Y/C-genlock kost in Nederland tussen de f 1350,- en f 1450,-.

Wie alleen een VHS/8mm PAL-genlock zoekt kan met



de gewone uitvoering van de hiervoor besproken Y/C-genlock volstaan. De mogelijkheden zijn hetzelfde alleen beschikt de unit nu niet over de hosiden Y/C-aansluiting. Het is alles tulp wat de videoklok slaat. De videokwaliteit van het RGB-beeld is gelijk, maar die van het gewone S-VHS natuurlijk minder. Daartegenover staat een aanmerkelijk lager prijskaartje van circa f 850,-.

Een **RGB-splitter** wordt gebruikt om in kleur te kunnen digitaliseren. Het videobeeld wordt achtereenvolgens in de kleuren Rood, Groen en blauw gedigitaliseerd. Electronic Design biedt een RGB-splitter die zowel geschikt is voor gewoon VHS (8mm) als S-VHS en Hi8. Er zijn drie regelaars voor kleur, contrast en helderheid. Voor de RGB-split zelf zit er een grote draaiknop op het kastje. Deze knop kent de standen Auto en manual R, G en B. Er zit verder een keuzeschakelaar voor de standen Amiga en Video aan de voorzijde

zodat deze Electronic Design RGB-splitter permanent aangesloten kan blijven. Ook dit apparaat is tevens bruikbaar als RGB-converter en het weergeven van S-VHS-signalen op een TV met scart-ingang. In de winkel gaat deze RGB-splitter ongeveer f 600,- kosten.

Info: Electronic-Design, Detmoldstrasse 2, 8000 München 45, Duitsland

DIGI-GEN en Videomaster

Door Peter Biet Computerdesign (PBC) werd onlangs onder de naam **DGI-GEN** en nieuw grafisch en effectsysteem voor de Commodore Amiga uitgebracht. De DIGI-GEN werd ontworpen rondom het video-chipconcept van VALVO en is in feite een combinatie van een genlock, digitale SEG, automatische RGB-splitter en S-VHS signaalprocessor. Verder beschikt de DIGI-GEN-unit over een ingebouwde blackburst-generator. Opmerkelijk is dat een groot aantal functies ook zonder het gebruik van een Amiga mogelijk zijn. De genlockfuncties worden bediend door een computer en een videofader. Daarmee zijn PURE, MIX 50% 50%, SUPERIMPOSE en INVERSIMPOSE mogelijk. Voor de effecten (128 wipes) zijn er drukknoppen aan boord. De overige functies worden via draaiknoppen bediend. Op de



beeldkwaliteit met een bandbreedte van 5,5 MHz (0,3 dB) en de signaalruisafstand van meer dan 73 dB valt weinig aan te merken. Dat is ruim voldoende voor het semi-professionele circuit. Er zijn twee DIGI-GEN-versies leverbaar. Het model zonder ingebouwde digitizer kost f 1.500,- en met f 1.800,-.

Voor de echte lowbudget professional biedt PBC de **Videomaster**. Deze "videomeester" bestaat uit een forse kast met onvermoede video-mogelijkheden. De videomaster is o.a. bruikbaar als:

- * Videosignaalprocessor voor RGB-, S-VHS Y/C- en CVBS-signalen

- * Real-time digitizer
- * Time Base Corrector
- * Signaalconverter
- * Kleurprocessor
- * Generator voor multi-pictures en stilstaand beeld

- * Beeldscherpte-versterker

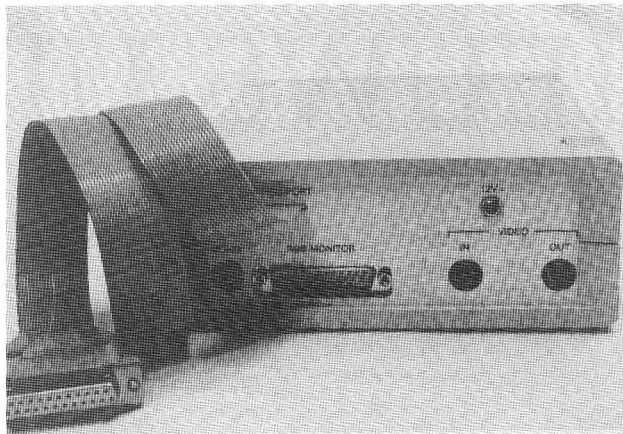
Op de basis Videomaster van circa f 4.400,- kunnen allerlei extra modules zoals een Amiga-genlock, SEGs (statische en dynamische effect-box), colorbox en VITC-editor worden aangesloten.

Voor informatie: PBC, Dieterhausenerstrasse 28, D-6409 Dipperz, Duitsland.

Deluxe View

De vermaarde Amiga-digitizer DigiView van NewTek heeft sinds twee jaar een geduchte Duitse concurrent. **Deluxe View** bestaat uit een langwerpige kastje met een lintkabel en twee tulpaanslui-

tingen. De lintkabel gaat in de printerpoort van de Amiga en de beide tulpbussen in/out zijn voor de aansluiting van



video-apparatuur. Met een bandbreedte van 4 MHz biedt Deluxe View resultaten die bij gewoon VHS en 8mm niet misstaan en ook de kwaliteit op de monitor ziet er gelijk uit. Voor digitizen in kleur is een half- of een volautomatische RGB-splitter nodig. De meegeleverde software is geheel compatibel met de snelle Motorola MC 68020 en 68030 microprocessoren. Deluxe View kost rond de f 400,-.

Inlichtingen: Electronic Design.

VidTech

Ook in de VS kan men goede dtv-produkten voor de Amiga maken. Onder de naam **VideoMaster** brengt VidTech een geavanceerde S-VHS genlock en effectgene-

rator met RGB-processor en -splitter uit. De videomaster biedt qua output zoel Y/C-, composiet- als RGB-video

aan. Er zijn aansluitingen voor een externe digitizer, videomixer, videocamera, videodisc, Amiga en meerdere monitoren. De maximale PAL-bandbreedte bedraagt 5,5 MHz. Als special effects biedt de VideoMaster diverse wipes aan. De bediening verloopt via tiptoetsen en drie schuifregelaars voor video, Amiga en de wipes. LEDs geven de actieve functie(s) aan. Vermoedelijke prijs: f 2.000,-.

De **Scanlock** van VidTech is een hoogwaardige S-VHS genlock. Ook hier vinden we tiptoetsen en faderbediening met attentie-LEDs. Een vertikale intervalschakelaar maakt het storingsvrij overschakelen tussen twee externe videobronnen mogelijk. Via de remote kan de Scanlock door een Amiga-computer of een video-editor bestuurd worden. De prijs zal waarschijnlijk omtrent de f 1.400,- komen te liggen.

Inlichtingen: VidTech International Inc. 2822 NW 79th Ave, Miami, Florida 33122, USA. Een nederlandse importeur en exacte prijzen waren op het moment van schrijven nog niet bekend.

Video & Computer

Uitgeverij Laterna Magica staat bekend om haar goede keuze in video-, foto- en dia-boeken. Ook Reiner Folgners **Video & Computer**, Destop Video mit Amiga, is weer een goede greep. In slechts 170 pocketpagina's weet deze auteur de lezer een prima indruk te geven van wat desktopvideo eigenlijk is en wat daar allemaal aan hard- & software voor te koop is.

Na de kort introductie "Amiga goes to Hollywood" over het gebruik, de mogelijkheden, beeldinstellingen en keuze van een Commodore Amiga breekt workshop-time aan. Na het bespreken van een aantal populaire titel-, animatie- en tekenpakketten gaat Folgner met de lezer aan de slag. Hoe maak ik een pakkende titel? Hoe animeer ik een titel? Wat kan ik met een digitizer? Hoe bewerk ik gedigitaliseerde beelden tot Amiga-kunst? Het komt allemaal aan bod.

Daarna zijn de genlocks met hun mogelijkheden aan de beurt. Van eenvoudig tot semi-professioneel S-VHS. Het gebruik van de kleur 0 en de maskeropties. Inverse en video/computermode. De ingebouwde RGB-splitter of kleurprocessor. Het passeert allemaal vlot leesbaar de revue. Enkele Genlocks zoals de Hama S-590 krijgen relatief veel aandacht.

Een apart hoofdstukje is gewijd aan de Amiga als VITC-editorcomputer. Kortom het monteren van videoscenes met behulp van tijdcode en de computer. Het HVS-Schnittmodul voor VHS en S-VHS en de professionele (Betacam, Umatic, Pro-S) PX editor-controllers worden uitgebreid besproken. De laatste hoofdstukken zijn gewijd aan semi-professionele videorandapparatuur, raytracing en 3/4D-software en extra Amiga-hardware om het videosysteem sneller te maken.



Video & Computer is ons inziens een dtv-boek dat sterk tot de verbeelding zal spreken. Dankzij het vlotte taalgebruik, duidelijke werkscheps en vele toepasselijke kleuren-illustraties wordt het snel duidelijk wat een Amiga allemaal kan en hoe dit thuis in zijn werk gaat. De gegeven informatie is behoorlijk up-to-date en niet belegen zoals in diverse wat oudere duitse boeken. De basis-informatie over een groot aantal softwarepakketten en optionele hardware kan helpen om de keuze te bepalen. Aanbevelen voor videomakers die de duitse taal goed beheersen. En om de prijs van circa f 30,- behoeft u het echt niet te laten.

Titel: Video & Computer

Auteur: Reiner Folgner

Aantal blz.: 167

Illustraties: Kleurenfoto's en zwartwitfoto's

Uitgeverij: Laterna Magica

ISBN:3-87467-435-5

Prijs: DM 29,90

Nieuwe Commodore Monitoren

Onlangs heeft Commodore twee nieuw kleurenmonitoren voor de Amiga- en PC-lijn geïntroduceerd. Het **Model 1930** is een eenvoudige VGA-monitor met een 14 inch beeldbuis, pitch van 0,31, horizontale frequentie van 31,5 KHz, verticale synchronisatie van 50 tot 70 Hz en een draai & kantelvoet. Het oplossendvermogen bedraagt naar videomodus 640 x 350, 640 x 400 of 640 x 480 (standaard VGA). De beeldbuis is goed ontspiegeld en geeft een rustig scherp VGA-beeld.

Het **Model 1950** is een multisync-type met eveneens een 14 inch ontspiegelde beeldbuis en een dotpitch van 0,31 mm. De horizontale synchronisatie verloopt van 15 KHz tot 35 KHz de verticale van 60 tot 80 Hz. De instelling verloopt uiteraard geheel automatisch. Qua resolutie

scoort de 1950 maximaal 800 x 600 pixels bij een videobandbreedte van 30 MHz. Ook dit model beschikt over een draai & kantelvoet.

Nieuwe Commodore Printers Ook Commodore gaat op de inkjettoer. De **MPS 1270** zet zonder lawaai een keurige laserachtige letter op het papier. Deze MPS 1270 inkjet is van een fraai slank tafeldontwerp. Het drukkertje is verrassend plat en het papier wordt er van voren in gestoken en komt er aan de bovenkant weer uit.

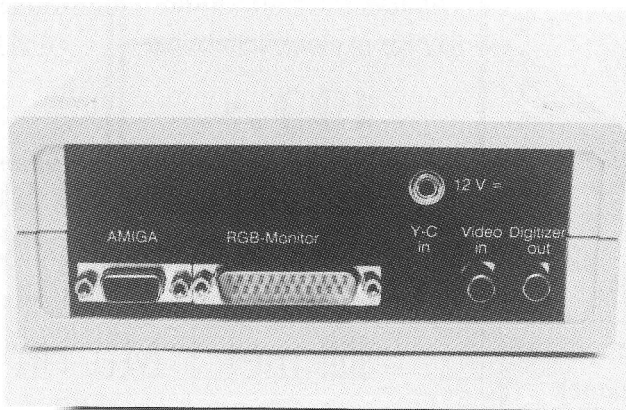
Voor het optekenen van de letters gebruikt de MPS 1270 acht pijpjes die een resolutie van 240 puntjes per inch halen. Er is een buffer van 0.7 KB aan boord. Het afdrukken van een A4-tje of overheads heet kost circa 1 minuut.

De **LPS 2000 laserafdrukker** van Commodore is een

UNIX System V versie 4

Commodore ziet grote mogelijkheden in de combinatie **Amiga 3000 en de UNIX V versie 4-release**. Vooralsnog bevindt het UNIX-gebeuren op de Amiga zich nog in een experimentele fase. Het systeem is aan een aantal universiteiten in zes verschillende landen en softwarehuizen ter beoordeling en test aangeboden. Toch denkt Commodore voorjaar 1991 de eerste UNIX-Amiga's operationeel te hebben voor de gebruikersmarkt.

Via de zogenaamde Applications Binary Interface (ABI) wordt de portabiliteit tussen de voor de Motorola 680XXX CPU geschreven UNIX-programmatuur verzekerd. Als standaard UNIX-configuratie wordt momen-



mooie kompakte (35 bij 27,6 cm) vierkante doos met twee A4-papierladen aan de voorzijde. De afdrukkwaliteit is een standaard 300 dpi bij een snelheid van zes pagina's per minuut.

Er zijn zes geïntegreerde tekensets met vijf bekende printeremulaties aan boord. Dit alles valt eenvoudig te bedienen via de druktoetsen en het 16 LED-display aan de voorzijde. Op de afdrukkwaliteit en het printergeruis valt niets aan te merken.

teel gedacht aan een 25 MHz Amiga 3000 met versie 4 van UNIX V, X-Windows en Open Look.

U.S.

BOEKBESPREKINGEN

Een oriëntatiegids op dtv-gebied is de omschrijving die uitstekend bij het boek **Amiga Desktop Video Guide** past. Deze dtv-gids werd geschreven door iemand die zijn sporen zowel bij lokale Amerikaanse TV-stations als bij het werken met Amiga's verdient heeft. En dat valt wel te merken ook want Guy Wright slaagt er prima in om de gescheiden computer- en VHS/8mm-videowereld bij elkaar te brengen.

Om de Amiga Desktop Video Guide te kunnen lezen dient u wel over enige basis-computerkennis te beschikken en de engelse taal machtig te zijn. Iedereen die de Amiga-handleiding goed gelezen heeft en het Engels op MO-niveau beheerst moet uit dit gemakkelijk lezende boek kunnen komen.

Een van de sterke kanten van Guy Wright is dat hij de technische begrippen in enkele treffende goed leesbare zinnen kan omschrijven. Waar je anders in slaap valt lees je nu met plezier over TV-normen, monitoren, resoluties Time Base Correctors, encoders, luminantie, chrominantie, Y/C en gewoon VHS/8mm-video, digitizers genlocks en softwaretoepassingen. De auteur blijkt zowel prima thuis op het gebied van video-editing als het werken met de Amiga als dtv-computer. Duidelijke tekeningen en een overzichtelijke opzet van de tekst geven zelfs de beginner direkt een goede

oriëntatie in deze toch echt niet zo makkelijke materie.

Na de basisstof over termen en definities betreffende videorecorders, camcorders en de Amiga volgen interessante en van praktische tips voorziene hoofdstukken over o.a.:

- ° Genlocks
- ° Digitizers, scanners, en framegrabbers
- ° Paint- en imaging -software
- ° Video-aftiteling
- ° Rendering en raytracing
- ° Animatiesoftware
- ° Muziek en video
- ° Special effects
- ° Video-editing

Na het lezen van de Amiga Desktop Video Guide heeft u de principes en basiskennis van dtv in de vingers. De praktische gebruikstips helpen om het maximale profijt uit hardware en software te halen. Met de kooptips kunt u een verkeerde aankoop voorkomen. Bedenk wel dat dit boek op Amerikaanse leest geschied is en sommige van de genoemde apparatuur en technieken voor NTSC-TV-doeleinden bedoeld zijn. Aanbevolen voor de beginnende en gevorderde dtv-er op de Amiga.

Titel: Amiga Desktop Video Guide

Auteur: Guy Wright

Aantal pag.: 257

Illustraties: Tekeningen en figuren

Uitgeverij: Abacus U.S.

ISBN: 1-55755-057-3

Prijs: f 44,-

Amiga de Film

Het enige (voor zo ver ons) bekende nederlandse boek over desktopvideo is **Amiga De Film**. Een echt Nederlands werkje over dtv heeft natuurlijk altijd een streepje voor. In de eigen landstaal spreken de toch al moeilijke technische videotermen eerder aan dan een Angelsaksisch of Duits jargon.

Amiga De Film geeft al aan dat dit Data Becker boek over de Amiga-animatiemogelijkheden gaat. Jammer is dat het boek grotendeels over de wat "belegen" animatiemogelijkheden bij oudere Deluxe Paint (I en II) en Deluxe Video II gaat. De versies III van deze grafische en animatiesoftware komen niet bij auteur Spanik voor. Niettemin hebben bezitters van deze al wat oudere Amiga-pakketten er toch nog een aardige gebruiksaanwijzing aan. Zeker als je nog niets van teken- en animatietechnieken weet. Spanik weet het allemaal treffend en vooral duidelijk te beschrijven en de beginner kan meteen aan de dtv-slag. Het eerste hoofdstuk geeft een overzicht van en uitleg over computeranimaties en simulaties. Ook is er aandacht voor het opzetten van een goed plan om beeld en geluid tot één vloeiende animatie te kunnen integreren. Daarna passeren zoals gezegd de pakketten Deluxe Paint I en II, en Deluxe Video II de revue. Wie het basis-kappen van de dtv-zweep wil leren kan hiermee wel uit de voeten.



De laatste hoofdstukken behandelen kort het gebruik van digitizers, genlocks en videorecorders. Nuttig is de tabel met lelijke of slecht zichtbare kleurcombinaties. Bijlage 1 biedt nog een verklaring van de technische Amiga-termen.

Amiga De Film is ons inziens alleen geschikt voor de bezitters van oudere Deluxe Paint en Video-versies. Met name de beginners kunnen hiermee de grondbeginselen van animatie-dtv leren. Voor de gevorderde gebruikers en bezitters van nieuwe releases van deze videosoftware biedt het boek te weinig. Voor alleen oriëntatie op dtv-gebied is dit boek te duur.

Titel: Amiga De Film

Auteur: Spanik

Aantal pag.: 179

Illustraties: figuren, screen-dumps en tekeningen

Uitgever: Data Bcker Nederland

ISBN: 90 229 3686 4

Prijs: f 67,50

Desktop Video Auf Dem Amiga

Dit duitse dtv-werkje voor de Amiga verscheen onlangs bij Markt & Technik. De opzet is traditioneel. Eerst komen de verschillende hardware-componenten zoals digitizers, genlocks en PAL-kaarten aan bod. Mogelijkheden, beperkingen en doel van de dtv-hardware komen ruimschoots uit de verf.

Vervolgens gaat auteur Andreas Grote op de software-toer. Deluxe Paint, Moviesetter, TV*TEXT, Aegis Video-titler, Deluxe Video, Digi-View Gold en diverse andere Amiga dtv-pakketten passeren de revue. Op deze pagina's vindt u naast een beknopte gebruiksaanwijzing van deze softwarepakketten ook de nodige tips & trucs. De beginner kan meteen met de software aan de slag. Ook de bijlagen bevatten waardevolle informatie zoals een technisch woordenboek, bronvermelding/referenties en fonts.

Naast de papieren stof biedt dit dtv-boek twee diskettes met zwartwit en kleuren beelden plus enkele animaties. Dit uiteraard ter leering en de vermaak.

Ons inziens is dit boek zowel geschikt voor de videofilmer als de computergebruiker die de grondslagen van Amiga-dtv wil leren. De enige voorwaarde is een goede beheersing van de duitse taal.

Titel: Desktop Video auf dem Amiga

Auteur: Andreas Grote

Aantal pag.: 189

Illustraties: figuren en grafdische afbeeldingen plus twee diskettes

Uitgever: Markt & Technik Verlag

Prijs: circa f 65,-.

U.S.

WordPerfect Amiga

Zij die niet het (voor)recht bezitten met een MS-DOS-computer te werken, maar bijvoorbeeld met een Commodore Amiga, kunnen wat WordPerfect betreft bijna net zoveel met deze tekstverwerker.

Het is alleen nog steeds een feit dat WP voor Amiga qua ontwikkeling nog maar bij de 4.2-versie van MS-DOS is. Maar je kunt er prima mee uit te voeren.

In twintig stappen door WP heen, dat is wat dit boek beoogt. In de inhoudsopgave vertellen klokjes hoeveel tijd er staat voor ieder hoofdstuk. Zoals bij ieder boek uit deze serie beginnen we bij de installatie van het pakket. Vermeld wordt ook hoe WP op een harddisk kan worden gezet.

Door de tekst heen staan drie soorten ikonen. Een voor Opgelet, een voor Aanwijzing en een voor Uitvoering. Deze helpen mee om het geheel nog sneller door te lopen.

Alle functies uit de menu's worden aangegeven door zowel de pull-menu's en de volgorde van de functietoetsen. Dit gebeurt heel duidelijk.

Interessant is ook het hoofdstuk Import & Export, hoewel daar een aantal opmerkingen bij zijn te maken. Om te beginnen volgt nergens uit dat het boek het Duits is vertaald, behalve dan hier. Het contact met IBM WP 4.2 en hoger wordt aangegeven door 5 tot en met 8 van de functie bewaar/opvraag. Daar kan worden gekozen om een tekst in IBM-formaat weg te schrijven. Dat het formaat op de diskette heel anders is bij een Amiga wordt vermeld en ook mogelijke (Duitse) oplossingen, als PC-handler. De meest elegante volgens ons, CrossDOS (Amerikaans) wordt niet vermeld, die met alle drives werken (in low-density) naar MS-DOS en van MS-DOS. Het toevoegen van plaatjes en in WP 4.1 voor Amiga niet mogelijk, maar met de beschreven tips komen we een heel eind in de richting. Voor wie snel WordPerfect op de Amiga wil leren, heeft een goed boek in handen.

Titel: Quickstart WordPerfect Amiga

Auteur: David Krause

Uitgever: Sybex

ISBN: 90 5160 292 8

Prijs: f 29,-

W.S.

KERST- EN NIEUW- JAARSKAARTEN

Een originele groet uit de Amiga

Tegen kerst en oud-en-nieuw verzendt Nederland grote hoeveelheden wenskaarten. Familie, vrienden en zakelijke relaties krijgen een gelukkig kerstfeest en een voorspoedig 1991 toegewenst. De meeste van deze wenskaarten zullen gewoon gekocht zijn. Een niet gering aantal goedewensers zal echter zelf kerst- en nieuwjaarskaarten vervaardigen of in opdracht originele exemplaren laten maken. Die zelfmakers kunnen zich ondermeer van de Amiga als kerst- en nieuwjaarskaart-generator bedienen. Attractieve titelletters voor de goede wensen, fraaie graphics achtergronden, gedigitaliseerde kersttafelen en



komplete slideshows, laat de eigen fantasie maar een de vrije loop.

Het zelf maken van kerst- en nieuwjaarskaarten kan op vele manieren. De een pakt een aardig wintertafreeltje (foto of tekening) en plakt daar wat pittoreske letters op. Een ander maakt zelf een grafisch ontwerp en fotografeert of digitaliseert dit. Weer een ander bouwt met behulp van takken, ballen, kunstsneeuw en kerstversiersel een komplete set. De Amiga-bezitter grijpt naar zijn of haar vertrouwde computer.

Teksten

Hoe maak ik een fraaie kerst- of nieuwjaarswens op mijn Amiga? Eigenlijk heel eenvoudig. Gewoon een goed **af-titelpakket** voor video ne-

men en daar eens lekker mee stoeien. Wij zelf gebruikten voor de hier getoonde voorbeelden **TV-Text Professional**. Dit titelpakket biedt de keuze uit een groot aantal speelse Zuma- fonts en verricht wonderen met een dithering background. Natuurlijk kunt u ook elk andere tekenpakket dat met gekleurde fonts werkt inzetten. Bijvoorbeeld Deluxe Paint III en Digipaint of een business graphics pakket. Houdt de kleuren wel een beetje in sfeer. Te knallerige of poenige wensen passen niet op kerst- of nieuwjaarskaarten. Houdt bij het maken van teksten eveneens rekening met grootte en locatie ten opzichte van eventueel te gebruiken andere afbeeldingen. Anders komt u straks misschien niet goed uit.





Figuren en achtergronden

Achtergronden voor kerst- en nieuwjaarskaarten zijn op vele manieren te maken. De kunstzinnige Amigagebruikers zullen wel raad weten met de door **paint-** en **art-work-software** geboden illustratie mogelijkheden. Als u er echt eens voor gaat zitten zijn werkelijk verbluffend originele wenskaarten mogelijk. Helaas is uw redacteur niet zo talentvol, maar het gaat hier tenslotte om de ideeën.

Voor wie het tekenwerk niet zo goed af gaat zijn er vele andere mogelijkheden. Er zijn diverse **bibliotheken** met standaard artwork of achtergronden in omloop. Met behulp van zo'n library-diskette zijn gemakkelijk zelf kersttrafieren op te bouwen. Geijkte objecten in deze zijn kaartranden (borders), klokken, kaarsen, kerstballen, versiersels, "vadertje tijd", champagneglazen, sneeuw enz. Ook een eenvoudige achtergrond doet het vaak prima.

Mogelijk drie is het binnenhalen van kerstillustraties met de videokamera of videorecorder. Met een videodigitizer als **DigiView** kunt u video beelden bevriezen of met de kamera overgenomen kerst illustraties in het Amiga grafische format opnemen. Voor opname in kleur is een RGB-splitters of -filter noodzakelijk. Het inkleuren van een zwartwit beeld kan met name bij het korrelige low

res-beeld een leuke creatie opleveren.

Hardcopies

U kunt natuurlijk moeilijk de complete Amiga op de post doen. Daarom zal er van de ontworpen en op diskette gesavede kerstkaarten een of andere vorm van een hardcopy gemaakt moeten worden. Wij bespreken in het kort twee mogelijkheden.

Uitdraaien met de **printer** is de eerste mogelijkheid. Niet iedereen beschikt echter over een kleuren-model, maar zwartwit kan ook heel mooi zijn. Afdrukken met de printer is behoorlijk flexibel. U kunt zelf het eindresultaat, binnen de margemogelijkheden, bepalen.

De twee mogelijkheid is **overnemen op fotografische materialen**. Bijvoorbeeld op dia- of negatiefilm. Dat gaat als volgt.

- ° Zet een spiegelreflex-kamera op statief voor de monitor
- ° Gebruik een objectief dat in de makrostand beeldvullend werkt
- ° Stop een ISO 100 filmemulsie (dia of negatief) in de kamera
- ° Stel de monitor normaal af
- ° Stel het objectief scherp en controleer of er geen storende objecten in de zoekeer zichtbaar zijn
- ° Verduister het vertrek
- ° Druk de kamera met een draadontspanner af

Dezelfde procedure kan ook met een direkt-klaar-film. Dan ligt het eindresultaat meteen op tafel.

Een eenmaal op negatief of dia opgenomen wenskaart kan gemakkelijk in grote hoeveelheden worden afgedrukt. Uw fotohandelaar hanteert daarvoor een voordelige meerstuks-prijs! In een aantal gevallen is het mogelijk om wenskaarttafrerelen in een standaard kerstkaartformaat van een afdrukcentrale te laten opnemen. De prijs van deze "eenheidsworsten" ligt zeer scherp en kan bij een uitgebreide kennissen- of vriendenkring zeker lonen.

Slideshow

Hebben uw goede relaties ook een Amiga dan is het maken van een persoonlijke kerst/nieuwjaars-slideshow wellicht een goed idee. Het aan elkaar koppelen van al of niet geanimeerde kerstboodschappen en illustraties omlijst met passende muziek zal hen verbazen. Denk ook eens aan vuurwerk- en andere effecten.

Een kerstvirus meesturen behoort uiteraard niet tot de goede gewoonten.

In dit artikel hebben wij u hopelijk een aantal aardige ideetjes aan de hand gedaan om een originele kerst- of nieuwjaarsgroet uit de Amiga te laten rollen. Een goed plan en wat fantasie zijn al het halve werk. Stuur ons eens uw meest geslaagde ontwerpen toe. Dan kunnen wij de redactie burelen daarmee verfraaien.

U.S.



VIZAWRITE DESKTOP 2.02

Plezierige Nederlandstalige Amiga-tekstverwerker

Een goede wordprocessor is een "joy for ever". Met een tekstverwerker moet je immers met de ogen dicht kunnen lezen en schrijven. Het pakket moet zich als het ware vanzelf wijzen om daar prettig en

meer voor de gevorderde gebruiker. Wie al dat extra tekstgeweld van het duurdere WP 5.X niet nodig heeft is met VizaWrite prima af. Ook de nieuwste **Desktop 2.0**-versie kent weer een gebruiksvriendelijke menustructuur, multitasking, im-

trekmenu's. Daar maakt VizaWrite dan ook goed gebruik van om de verschillende opmaakmodules overzichtelijk te ordenen. Na het openen van de VizaWrite-ikoon ziet u de laden voor VizaWrite zelf, de documenten, de afbeeldingen (illustraties), woordenlijsten en Print & Merge.

De door Altycos meegeleverde nederlandse gebruiksaanwijzing is zo duidelijk dat zelfs iemand die nog nooit een Amiga-computer heeft aangezet het programma toch direkt kan opstarten. Natuurlijk is wel enige achtergrondkennis over de mogelijkheden van tekstverwerken vereist. Daarvoor zijn tal van eenvoudige boekjes in het computerschap van uw boekhandel te vinden.



efficiënt documenten en brieven mee te kunnen typen. Een programma waarbij de gebruiker voortdurend in onlogische menu's en ingewikkelde toetscommando's verzeild raakt voldoet niet aan de eisen die men aan een moderne wordprocessor stelt. VizaWrite is zo'n lekker rechtlijnige tekstverwerker die bovendien voldoende extra mogelijkheden biedt.

De tekstverwerker VizaWrite wordt al jarenlang tot volle tevredenheid door bezitters van Commodore-computers gebruikt. Eerst kwamen de C-64- en later de Amiga-versies. Het pakket biedt net wat de gewone thuytper nodig heeft en ook nog nets ietsje

port van graphics, semi-dtp (voor echte desktoppublishing heeft u ook dtp-software nodig!), diverse fonts en alle mogelijkheden die men van een midden-klasse tekstverwerker mag verwachten. Ter ondersteuning is er nog het door importeur Altycos gefaciliteerde VIZA AREA FIDO-bulletinboard. De hardware-eisen zijn minimaal: 256 KB aan vrij RAM, één diskdrive en een 80-koloms monitor (standaard Amiga-monitor). VizaWrite werkt met alle Kickstart- en Workbench-versies vanaf 1.2.

Amiga-desktop

Een van de grote Amiga-voordelen is het gebruik van **desktopikonen** en duidelijke

Opstarten

U kunt VizaWrite op twee manieren opstarten:

- ° Via de **VizaWrite-ikoon**
- ° Via het **CLI** (Command Line Interface, gewoon "VizaWrite" en vervolgens ENTER in te toetsen. Met de toevoeging "-i" kan VizaWrite ook in de HIRES-interlacemodus gebruikt worden. Zoiets is echter alleen verteerbaar met een non-flikkering video-output. (zie ons artikel over grafische VGA-kaarten)

Na het inklikken op de VizaWrite-ikoon verschijnt het tekstschermb. Bovenin staan een lineaal, een aantal document- symbolen (logo's) en de menubalk. De rode cursor



knipt wachend op de eerste text-input.

Formaatfuncties

De **lineaal** geeft eenheden van circa één inch aan. Die eenheden van 2,54 cm zijn weer onderverdeeld in tienden. De maximale tekstbreedte bij VizaWrite bedraagt 18 inches. Een verstelbaar wit driehoekje geeft respectievelijk de linker en de rechter marge van de tekstpagina aan. Daartussen staan de genummerde oranje tabulatordriehoekjes. Alle veranderingen op de lineaal betreffen meteen de gehele tekst. Wilt u slechts een deel van het tekstformat veranderen dan is insert van extra linealen nodig.

Met de drie **logo's in het midden** kiest de gebruiker de regelafstanden 1x, 1,5x en 2x. Rechtsboven staan vier logo-ikonen met de regelopmaakfuncties:

- ° Links uitlijnen, rechts ongelijk
- ° Regel voor regel gecentreerd
- ° Rechts uitlijnen links ongelijk
- ° Uitvullen met links en rechts gelijk

Standaard is links uitgelijnd.

Tekstverwerken

Het wordprocessen met VizaWrite verschilt niet zoveel van andere standaard tekstverwerkingspakketten. Platte tekst wordt gewoon rechtstreeks ingetypt. Er zijn zogenaamde "snelle commando's"



voor de cursorbesturing en het aanwijzen van blokken. Bijvoorbeeld SHIFT-LINKS stuurt de cursor naar het begin van de regel en SHIFT-ONDER naar het einde van het document. Natuurlijk kunt u de cursor ook gewoon met de muis besturen. De keuze blijft aan de voorkeur van de gebruiker.

Het aanwijzen van tekstblokken gaat eveneens comfortabel met de muis. Cursor aan het begin van de tekst zetten, linker muisknop ingedrukt houden, trek de muis over het tekstblok (wordt highlighted) en vervolgens de gewenste kopieer-, verplaats/invoeg- of wisfunctie activeren. Ongedaan maken is gewoon een kwestie van nogmaals links klikken.

Opmaakopties

Tekstverwerken wordt pas echt leuk als u de tekst attractief kunt opmaken. VizaWrite biedt daartoe een viertal mogelijkheden:

- De keuze van **verschillende fonts**

- ° De **tekststijlen** gewoon plat, vet, schuin, onderstrepen, sub- en superscript
- ° Het **invoegen van plaatjes**
- ° Het **aanbrengen van kop- en staartregels**

Het instellen van kolommen kan uitsluitend door gebruik te maken van de tab-functies en dat vinden wij toch een gemis.

Alle opties kunnen met behulp van de menubalk of verkorte toetscommando's geactiveerd worden.

Het kiezen van verschillende fonts, tekststijlen en plaatjes is dermate eenvoudig dat iedereen in de korste keren geïllustreerde documenten in elkaar kan zetten. De menu's en benodigde muisklikken wijzen zich vanzelf.

Ingeladen plaatsjes kunnen middels aanklikken, er verschijnt een bewerkbaar kader, verplaatst, vergroot, verkleind of uitgerektsamengeperst worden. Met de muis kunt u de tekst er omheen rangschikken.

Er kunnen maximaal negen verschillende tekstvensters geopend worden. Daarmee zijn uitgebreide knip&plak-tekstoperaties mogelijk. Het aantal bescheidbare vensters is wel geheugen afhankelijk. Daarom raden wij minimaal 512 KB aan vrij RAM aan.

WUZIWUK

Wat U Ziet is Wat U Krijgt gaat ook bij VizaWrite slechts gedeeltelijk op. Het beeldscherm laat min-of-

meer zien hoe de tekst en bijbehorende illustraties later op het papier zullen komen. De uiteindelijke output blijft uiteraard sterk afhankelijk van de printerkwaliteit- en mogelijkheden terwijl er altijd enig verschil zal blijven bestaan tussen scherm- en printerfonts.

De afdrucken die onze HP Laserjet II en 24-naalds matrixprinter uitdraaiden zagen er echter keurig verzorgd uit. Met VizaWrite-teksten kan je zelfs zakelijk goed voor de dag komen.

Overige opties

Voor een eenvoudige tekstverwerker beschikt VizaWrite toch over een aantal nuttige extra's. Bijvoorbeeld het maken van **woordenlijsten**. In die woordenlijst worden korte tekstfrases of woorden opgenomen voor snelle inserts. De insert kan via selectie uit het woordenlijst-venster of via de naam intypen plus de Commodore-toets drukken.

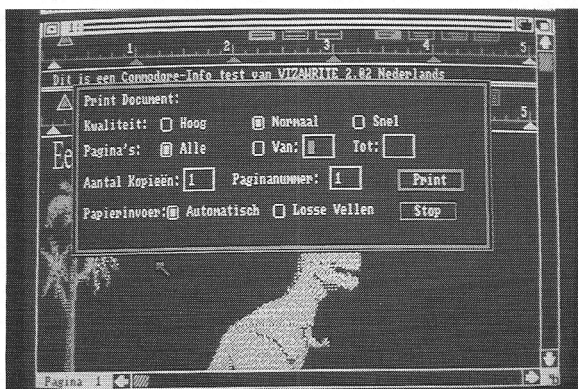
De **mail-merge-optie** is prima geschikt voor het op adres verspreiden van correspondentie. Verder beschikt VizaWrite o.a. over statusmeldingen, printerinstelling, search & replace, ga naar pagina... en een hele reeks verkorte commando's. Ook is er een digitale wekkerklok aan boord.

Conclusie

Versie 2.0 van Desktop VizaWrite is er in vergelijking met 1.0X aanmerkelijk sneller (schermfuncties) en een stuk gebruiksvriendelijker op geworden. Voor f 299,- krijgt u een semi-dtp-tekstverwerker in huis. Wie geavanceerdere tekstverwerkers met o.a. kolomfuncties wil zal meer moeten betalen.

Info: Altycos, tel.: 079-510757.

U.S.





DESKTOPVIDEO OP DE AMIGA

Mogelijkheden en techniek

Zelf professionele videotitels, flitsende animaties, logo's en artwork-achtergronden maken is met behulp van de Amiga-computer niet moeilijk meer. Kleurrijke titels kunt u in elke gewenste grootte gewoon met het toetsenbord intypen. Animaties berekent en vertoont de Amiga vlekkeloos op het beeldscherm. Logo's in 3-D ontwerpen is slechts een kwestie van talent met de muis of elektronische pen. En fraaie achtergronden kan de videomaker zelf optekenen of uit een softwarebibliotheek halen. Dit alles is mogelijk dankzij de betaalbare techniek van desktopvideo. Wat eerst het terrein van professionele video- en tv-makers leek te zijn ligt nu binnen het bereik van elke videoliefhebber met een bescheiden budget.

Wat is desktopvideo? Eigenlijk is dtv een heel ruim begrip. Dtv betekent in principe de computer als een soort creatieve montage-terminal gebruiken. In de praktijk komen de toepassingen voornamelijk neer op het maken van goede titels, titeffecten, animaties en grafische logo's of achtergronden voor video. Elk grafisch en special effect dat u met een computer kunt maken en op video overgenomen kan worden valt onder



dtv. Ook kan men het monteren van videoscenes per computer tot dtv rekenen. Het gebruik van computergestuurde tijdcodeinterfaces speelt nog slechts een bescheiden rol.

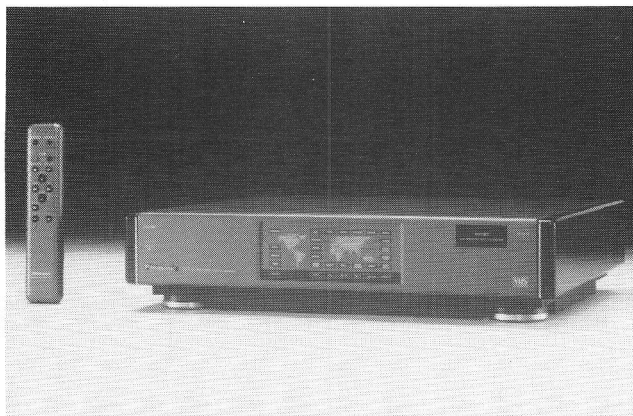
In deze Commodore-Info Special over dtv komen de verschillende mogelijkheden, gebruikte technieken en soorten video-hardware aan bod. Het eerste deel van deze special is soort verkenning binnen de Amiga-videowereld. In het tweede gedeelte wordt praktisch op de verschillende hardware, interfaces en het gebruik van de belangrijkste videosoftware ingegaan.

Geschiedenis

Een computer wordt vaak gezien als een "doods" rekeninstrument. Daar waren deze apparaten in de begin jaren '80 ook voor bedoeld. De grote verandering kwam pas toen zakelijke gebruikers rond 1983 wel eens een goed grafiekje van hun saaie cijferbrei wilden zien. In vrij korte tijd kwamen er allerlei software (=programma's) op de markt die keurige staaf-, lijn- en taartpuntgrafieken konden tekenen. De veelzijdige gebruiker wilde daar spoedig fraaie gekleurde letters, grafische symbolen en kleine animaties bij. De grafisch artiest ontdekte dat je

met een computer ook prima kon tekenen en weldra rolde er echt artwork uit de Personal Computer (PC). Reklamebureau's exploreerden de mogelijkheden van door computers ontworpen 3-D logo's, teksten en kleine animaties.

Al die door de computer gegenereerde grafische beelden trokken al snel de aandacht van film- en videoproducenten. Wat die computer op de monitor zette moest op video over te nemen zijn. Aanvankelijk ging het om peperdure apparatuur, maar dat verdien-den die producenten in bespaarde arbeidstijd snel terug. Een computer maakt ti-



tels, tekenfilms en geanimeerde logo's gewoon 4 tot 10 maal sneller aan dan volgens de conventionele knippen plaktechnieken mogelijk is.

Voor de amateur- en low-budget videomaker kwam de doorbraak in 1984 met de Amiga. Deze Commodore PC was speciaal ontworpen voor het maken van kleurrijke grafische beelden, titels en animaties bij een hoog oplos-send vermogen. Bovendien beschikte dit type PC ook nog eens over HiFi-stereogeluid. Dankzij de snel dalende prijzen en een groot scala aan videorandapparatuur werd de Amiga snel de lieveling van omroepen en kabelproducenten. Ook de video-amateur heeft de Amiga inmiddels ontdekt en uit menige huiskamer studio komen nu prima afgetitelde video- en zelfs complete tekenfilms.

Twee werelden

Uit het voorafgaande blijkt al dat computers en video langzaam naar elkaar zijn toegroeid. Qua ontwerpconcept is er echter sprake van twee gescheiden werelden. Ontwerpers van PC-video dachten in de eerste instantie alleen maar aan fraaie plaatjes op de monitor van door de computer gegenereerde graphics. Die van de videorecorder en camcorder aan een zo goed mogelijk registratie van op te nemen videobeelden. Dat leidde tot twee geheel gescheiden technieken waarbij binnen elke techniek ook nog

een groot aantal verschillende videostandaards gebruikt worden.

Bij de PC werd aanvankelijk uitsluitend digitaal (Rood, Groen, Blauw) gewerkt. Dat signaal moet voor gebruik in een monitor weer naar analoog worden omgewerkt. Bij computers met een VGA-kaart is het videosignaal wel analoog RGB maar nog niet direct geschikt voor PAL-composiet of Y/C (=Super-VHS of Hi8 video).

In plaats van in beeldlijnen drukt men bij de PC het oplos-send vermogen uit in pixels (beeldpunten). Bijvoorbeeld 800 x 600 of 1024 x 768 beeldpuntjes. Het aantal kleuren wordt beperkt door het beschikbare videogeheugen (RAM) en is RGB-gecodeerd in plaats van een composiet of Y/C-mengsignaal. Aan PC-monitoren wordt een groot aantal verschillende frequenties door verschillende videokaarten aangeboden. (zie tabel 1). Zo heeft elke videokaart zijn eigen bijbehorende monitor. En als laatste incompatibiliteits-druppel gebruiken tal van in de VS ontworpen PC-systemen de Amerikaanse NTSC tv-norm en een netfrequentie van 60 Hz.

U voelt de bui al hangen. Alle videosignalen en kabeltjes zijn incompatibel met de bestaande VHS- en 8mm videostandaards van 625 beeldlijnen en 25 beelden per seconde. Einde desktopvideo? Zeker niet! Als er ergens een signaal uitkomt moet dit in

een ander signaal kunnen worden omgezet. Bij de Amiga gebruikt men daarvoor videovertaalkastjes of insteekkaarten, de zogenaamde video-interfaces. Het video-interface bewerkt en versterkt het RGB-signaal tot een bruikbare PAL VHS/8mm- of Y/C-output. Een klein aantal recorders kan ook PAL RGB gebruiken.

Koop het juiste interface en alle hierna besproken mogelijkheden en toepassingen kunnen op videotape worden overgenomen. Welke hardware u precies nodig heeft komt in het tweede gedeelte van deze Special ter sprake.

Mogelijkheden en toepassingen

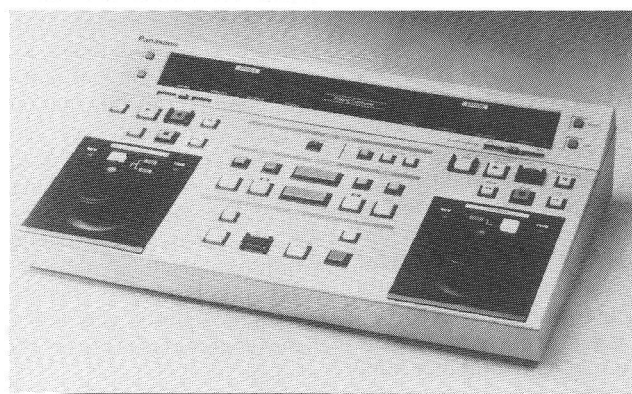
Een aantal van de door de Amiga als dtv-computer geboden mogelijkheden zijn al kort ter sprake gekomen. Centraal staan daarbij de snelheid en de veelzijdigheid van de grafische Amiga-technieken ten opzichte van het "oude handwerk". U kunt de monitor van een dtv-computer vergelijken met een grafisch palet waarop met enkele pennestreken of toetsaanslagen professionele logo's, tekeningen en titels verschijnen. Foutcorrectie, verandering van vorm, grootte en kleur, het laten uitvoeren van bewegingen, alles gaat veel sneller dan met de conventionele studiomethoden.

Omdat alles zo snel gaat kan de videomaker ook meer mogelijkheden uitproberen, variaties op één ontwerp, en

verfijningen aanbrengen. Dat laat iemand die uren heeft zitten knippen en plakken wel uit zijn of haar hoofd. Een lettertje verplaatsen of een zin centreren gaat met de PC-schaar&lijm-methode binnen 30 seconden. Kleur veranderen kost niet meer dan het aanwijzen van een andere tint in het palet.

Dankzij de hiervoor geschetste mogelijkheden is de computer heel geschikt voor de volgende montage-toepassingen:

- ° Aftiteling. Deze titels al of niet in geanimeerde vorm
- ° Het aanmaken van logo's en andere grafische symbolen
- ° Het ontwerpen van grafische achter- en voorgronden
- ° Het animeren van grafische of tekstobjecten
- ° Het maken van grafieken, stroomdiagrammen en organisatieschema's voor educatieve en zakelijke videoproducties
- ° Special (video-)Effect Generator (SEG)
- ° Videoprocessing (het bewerken van videosignalen, o.a. kleur, helderheid en contrast)
- ° Het maken van geluidseffecten
- ° Het bevriezen, digitaliseren, bewerken en afdrucken van videobeelden
- ° Tijdcode-montage van videotapes





We zullen al deze mogelijkheden eens voor een bekijken.

De titels.

Waarom titels op een aanmerkelijk duurdere Amiga-computer gaan intypen terwijl er toch goedkope titelgeneratoren voor video verkrijgbaar zijn? Iedereen die wel eens met een eenvoudig titelgeneratortje gewerkt heeft weet dat zo iets een ware crime kan zijn. Met name bij die typen waar je eerst het halve alfabet moet doorlopen om één teken op de regel te krijgen. Bij enkele korte titels valt zo iets nog te doen. Een lange aftiteling wordt bij deze apparatuur al snel monnikenwerk.

Bij een computer kunt u gewoon het schrijfmachine-toetsenbord (QWERTY) gebruiken. Eenvoudiger kan het niet. Korrektes, het verplaatsen of kopiëren van tekst vormt daarbij geen enkel probleem. Dat scheelt veel tijd en onnodige frustraties.

Een tweede beperking van de goedkopere titelgenerator vormen de kwaliteit en mogelijkheden van de ingebouwde lettertypes. De kleine letters vallen meestal wel mee. Grotere typen worden al snel hoekig of rafelig. Bij een PC met goede titelsoftware zijn de letters van tv-kwaliteit. Ook is er keuze uit een vrijwel ongelimiteerd aantal lettertypen, de fonts, en grootten. Bovendien kunt u de letters in verschillende kleuren weergeven. Hetzelfde geldt voor het toevoegen van schaduwen en stroboscopische effecten. En als klap op de professionele vuurpijl maakt het titelpakket ook nog een perfect achtergrondje aan. Dat is nog eens iets anders dan die enkele lettertypen en kleuren tegen een saai achtergrondje op een doorsnee titelgenerator.

Bovendien is een Amiga 500 goedkoper dan menige semi-professionele titelgenerator.

Tekenwerk

Kent u nog die dozen met magneetsymbooltjes waarmee je griekse tempeltjes en Eifeltorentjes kon bouwen? Zo werkt de videocomputer in principe ook maar dan gaat de tekensoftware veel verder. Met behulp van vervormbare standaard-figuren (rechthoeken, vierkantjes, cirkel, ellipsen, lijnen en bogen) worden net als bij de magneetsymbooltjes complexe figuren opgebouwd. Door spiegelen,



kopiëren, vergroten/verkleinen, uittrekken/samendrukken, inkleuren, verlopende kleurgradaties, schaduw- en 3D-effecten zijn verbluffende artwork-creaties mogelijk. Delen van tekeningen kunnen gemakkelijk uitgesneden, bewerkt en elders weer ingeplakt worden.

Wie niet over voldoende teken-talent beschikt komt met eenvoudige symbolen of "ingeblikte illustraties" (grafische bibliotheek op diskette) al een heel eind. Er zijn diverse bibliotheken met standaard-achtergronden, clipart (huisje, boompje, beestje, sport, landen, communicatie, vervoer enz.) te koop. Deze illustraties kunt u gewoon van bibliotheek-diskette in de computer laden en op het monitorscherm van teksten en andere figuren voorzien.

De meeste teken- of paint-software werkt met behulp van een muis. Er zijn echter ook lichtpennen en elektronische ezels (tekenpanelen) in omloop. Tip: Daar u de vorm en grootte van de muiscursor

kunt instellen is deze aanwijzer geschikt als explicatie-pijltje, d.w.z. de videomaker gebruikt de cursor als een aanwijzestokje bij video-overlays.

In de kantlijn van tekenpakketten staan de verschillende geometrische standaardfiguren en de teken- of bewerk-opdrachten. Kies bijvoorbeeld een cirkel-figuur en trek deze op de monitor tot de gewenste grootte uit. Daarna kan de cirkel ingekleurd, van

letters of een bepaald effect voorzien worden.

De meeste tekenpakketten laten korrektes op puntjesniveau toe. Daarvoor kunnen bepaalde delen van de tekening sterk uitvergroot worden. De mogelijkheden van dergelijke tekenpakketten zoals bijvoorbeeld Deluxe Paint III, Photon Paint en Digipaint voor de Amiga zijn bijna ongelimiteerd. De voornaamste beperkende factor vormt het eigen teken-talent.

Animaties

Behalve als titelstudio is de computer ook bruikbaar als animatiestudio. In principe kunt u alles animeren. Teksten, grafische objecten, gedigitaliseerde plaatjes en figuren. Het is zelfs mogelijk om de ene vorm gladjes in de andere te laten overgaan. In het verleden werden animatiefilms gemaakt door beeldje voor beeldje op te nemen en allerlei overlay-folies over elkaar heen te verschuiven.

Een zeer langdurig en uiterst nauwkeurig werkje. De Amiga 500, 1000, 2000, 2500 of 3000 kan dat allemaal veel sneller. Stel u wilt een gekleurd autootje over het scherm laten rijden. Daarvoor is natuurlijk eerst een autootje nodig. Dat voertuig kan de videomaker zelf tekenen of uit een plaatjes-voorraad putten.

De computer heeft de begin- en eindpositie van de beweging nodig. Alle tussenliggende vloeiende bewegingen rekent de PC voor u uit en laat vervolgens één vloeiende animatie op het beeldscherm zien. Elke verandering van richting dient afzonderlijk te worden aangegeven.

Nu is dit tekenfilmvoorbeeld nog een tamelijk doods geheel. Het autootje schuift alleen over het beeld. Leuker is het natuurlijk als ook de wielen zouden draaien en er rook uit de uitlaat kwam. Dat kan door de computer voor elk wiel de opdracht te geven dit te laten roteren en synchroon met het autootje mee te laten bewegen. Op gezette tijden komt er eventjes een flash (korte superimpose) met rookwolkje vlak achter de uitlaat.

Een dergelijke animatie maakt u natuurlijk niet in 1 minuutje. Zo iets kost enige planning en geëxperimenteer. De computer laat echter snel de resultaten zien en bijwerken gaat vlug en eenvoudig. Voor een ervaren dtv-gebruiker behoeft deze animatie niet meer dan 20 tot 30 minuten te kosten.

Logo's en business graphics

Zakelijke presentaties en dtv zijn historisch gezien nauw verbonden. In de inleiding las u al dat ontwikkeling van de computer als videopresentatiemedium sterk beïnvloed werd door zakelijke gebruikers die goede grafieken, logo's, stroomdiagrammen en organisatieschema's wisten. De software die zich met dit onderwerp bezig-



houdt wordt business graphics of- presentation graphics software genoemd.

Een business graphics-programma maakt snel en comfortabel grafieken aan met titels en symbolen ter verklaring van het getoonde cijfermateriaal. Ook zijn geavanceerde logo's en diagrammen mogelijk. Enkele pakketten kunnen de grafieken zelfs animeren. In de bijbehorende bibliotheken treft men landkaartjes en tal van clipart-symbolen aan.

Met behulp van deze software kan de videomaker gelikt ogende zakelijke en educatieve videoproducties maken. Bijvoorbeeld een bedrijfsreportage met grafieken die de groei in omzet en personeel laten zien. Ook kunt u met deze pakketten een eigen logo ontwerpen.

SEG

De computer als special effect generator is zeker geen onlogische gedachte. Dankzij bijvoorbeeld de animatiemogelijkheden met grote kleurvlakken is er keuze uit diverse wipe-effecten. Deze wipes kunnen met behulp van de software maar ook met bepaalde randapparatuur gemaakt worden. In dat laatste geval gaat het meestal om een combinatie met een genlock.

De ene vorm in de andere laten overgaan is voor de grafische Amiga-coprocessoren een simpel karweitje. Teksten die in voorwerpen veranderen en futuristische ruimteschepen de lichtsnelheid laten door breken, het is allemaal mogelijk.

Door videobeelden in de computer te voeren is het mogelijk hen met tekensoftware creatief te bewerken. Bijvoorbeeld paint- en pseudosolarisatie-effecten. Dankzij de digitale PC-techniek kan de geboden kwaliteit zeer goed zijn.

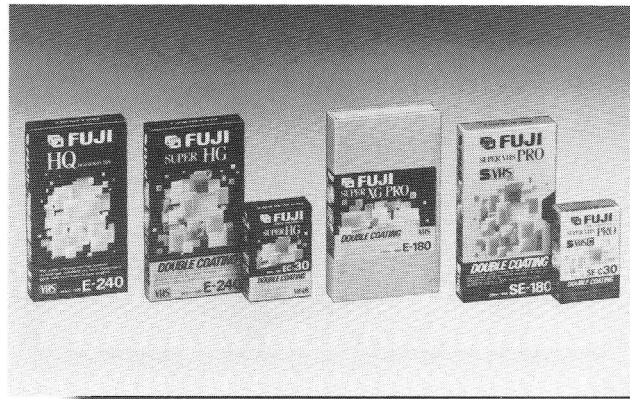
Videoprocessing

Een aantal leveranciers van dtv-hardware biedt compu-

tergestuurde videoprocessoren aan. O.a. verkrijgbaar bij Catronix (Videocomp), Michael Lamm en PCB. Gebruikelijk zijn daarbij de mogelijkheid tot kleur-, contrast-, helderheid- en scherpte-korrektie. De kwaliteit is ongeveer gelijkwaardig aan de betere gewone videoprocessor. Qua mogelijkheden, bedieningsgemak en prijs verschillen deze apparaten nauwelijks. Een aankoopcriterium kan de integratie met andere dtv-randapparatuur zijn. U slaat dan twee vliegen in één

Frame-grabbing

We hebben het tot nu toe voornamelijk gehad over grafische output van de video-PC naar de videorecorder. Het omgekeerde kan natuurlijk ook. Met behulp van camera of recorder met goed stilstaand beeld kunnen videobeelden aan een zogenaamde digitizer of framegrabber worden aangeboden. Deze framegrabber maakt van het analoge videobeeld een stilstaand digitaal computerplaatje. Dat digitale



klap (bijvoorbeeld een genlock en een processor in een) en bespaart studioruimte.

Geluidseffecten

Dat de Amiga-computer ook HiFi stereo geluidseffecten kan voortbrengen lijkt op het eerste gezicht voor de videomaker niet zo interessant. Geluiden kan je immers ook ingeblikt kopen of zelf opnemen.

Anders wordt bij het maken van tekenfilms op de computer. Synchroon aan de animatie kan een soundtrack worden toegevoegd. Daarmee zijn zeer realistische effecten mogelijk. Een auto-ongeluk met gierende banden, dreunende klap en langsratelende onderdelen is de PC wel tevertrouwd. Verder kan de AV-computer geluidsfragmenten digitaliseren, componeren en creatief (vervormen, echo) bewerken.

plaatje kan dan weer verder met tekensoftware bewerkt worden. Desgewenst kan het bewerkte plaatje weer terug naar de videorecorder worden geschreven.

Bij het opnemen in kleur is een RGB-splitter of drie-fasen-opname (door de opeenvolgende filters Rood, Groen en Blauw) nodig. Dit om het gemengde videosignaal in de drie basiscomputerkleuren te splitsen.

Praktische interface- en softwaretechniek

Na het ruiken aan de mogelijkheden van dtv op de Amiga zijn we nu bij het praktische toepassings- en hardwaredeel aangekomen. Dit tweede Special-deel gaat nader in op de verschillende interfacetypen, de beperkingen, de kosten en het praktisch werken met videosoftware. Alleen door een verstandige hard- en software-

keuze zijn optimale resultaten te behalen.

De Amiga zomaar met wat listig gesoldeerde kabeltjes op de videorecorder aansluiten heeft geen zin. In het eerste deel kwam al duidelijk ter sprake dat het om twee geheel gescheiden werelden gaat. Computer en video groeien weliswaar snel naar elkaar toe. Toch is er altijd een of ander elektronisch kastje of interfacekaartje nodig om alles aan elkaar te koppelen.

Welke Amiga?

Vaak wordt de vraag gesteld: "Welke Amiga is nu het best geschikt voor dtv?" Het antwoord hangt grotendeels van het beoogde doel en het beschikbare budget af. Met een Amiga 500 kunt u al heel veel voor weinig geld. Het voornaamste bezwaar van de Amiga 500 is echter het gebrek aan interne uitbreidingsmogelijkheden. Een andere videokaart er in is bijvoorbeeld niet mogelijk en de uitwisseling van coprocessoren lastig.

De Amiga 2000 en 2500 zijn duurder dan de 500. Daar tegenover staan de grote uitbreidingsmogelijkheden met snellere processoren, betere videokaarten, interne genlocks, meer RAM en zwaardere drives. Die uitbreiding kan geleidelijk aan gebeuren en behoeft het hobbybudget niet al te zeer te belasten.

Wie alles in één koop wil en/of multimedia wil bedrijven is met de aankoop van een Model 3000 meteen uit de brand. Deze zeer complete videomachine geeft voor rond de f 9.000,- alles wat de professionele videostudio zoekt.

De aankoop van een genlock biedt bij alle Amiga-typen zowel een goed PAL-sig-naal als bruikbare video-overlays (superimpose) voor bijvoorbeeld aftiteling.

RGB in, PAL uit

De Amiga-kleurencomputer werkt met RGB-signalen die



de conventionele VHS- en 8mm-recorders niet kunnen verwerken. Omzetting in een bruikbaar PAL composiet- of Y/C-videosignaal is noodzakelijk. Het interface dat deze omzetting doet heet PAL-encoder of -generator. De encoder bestaat uit een insteekkaart of kastje met Digitaal/Analoog (D/A)-converter, versterker en ondersteunende video-electronica.

Aan de ene kant gaat er digitaal RGB in. Aan de andere kant komt er analoog composiet (tulpplug) of Y/C (hosidenplug) uit. Dit videosignaal kan gewoon op een geschikte conventionele video-recorder worden opgenomen. Videokaarten voor de Amiga zijn relatief goedkoop. Voor rond de f 400,- verbouwt u een standaard model 2000 B tot PAL-machine. Ook voor de Amiga zijn professionele, en dus dure, PAL-encoders te koop. De geboden kwaliteitsverschillen zijn eigenlijk alleen voor de exploitanten van kabelkranten interessant.

Voor de Amiga zijn diverse zogenaamde HIRES (= hoge resolutie) VGA en daarvan afgeleide videokaarten te koop. De video-output van deze kaarten is bedoeld voor PC VGA- of multisync-monitoren en niet voor video-opnamedoeleinden. De PAL-kwaliteit blijft dus gelijk. Alleen de PC-monitordisplay wordt er scherper op. Een tweede reden om een HIRES VGA-kaart te kopen kan het elimineren van de storende beeldflikkering op de monitor zijn. Een beeldfrequentie van 25 Hz werkt irriterend bij stilstaande beelden. De bekende MicroWay AGA 2000 Flickerfixer (o.a. bij Amigis) levert al 704 x 510 pixels bij 50 Hz vertikaal en 31.5 KHz VGA. Bij 60 tot 70 Hz wordt het stilstaande beeld nog veel stabiel. Voor de 25 beelden motion video zijn dergelijke kaarten natuurlijk niet nodig. Praktisch gezien heeft de gebruiker niet zoveel aan alleen een PAL-output. Tenzij u gewoon titels tussen de videobeelden wilt plakken, uitslui-

tend animaties opneemt of een kabelkrant uitzendt. Overlays zijn bijvoorbeeld niet mogelijk.

De genlock

Twee videobeelden over elkaar heen leggen biedt een scala aan professionele mogelijkheden. Als eerste mogelijkheid het leggen van titels over bestaande videobeelden, een superimpose. De tweede belangrijke moge-



lijkheid is het combineren van videobeelden met een computer voor- of achtergrond. En als derde hier genoemde mogelijkheid het gebruik van de PC als SEG, bijvoorbeeld voor wipe- en titelcombinaties.

Om deze mogelijkheden en effecten te realiseren is een zogenaamde genlock nodig. Praktisch gezien doet dit randapparaat niets anders dan een inkomend video- en computersignaal met elkaar te combineren en er voor te zorgen dat de gekombineerde video-informatie precies op de juiste plaats en in dezelfde tijd terecht komt. Verder beschikt elke genlock over een ingebouwde PAL-encoder om het RGB-signaal in gewoon VHS en 8 mm (tulp) of Y/C (hosiden) om te zetten. Een aantal genlocks biedt zowel gewone als super-video-kwaliteit.

Genlocks kunnen qua prijs en uitvoering sterk variëren. Van gemiddeld f 900,- voor gewoon VHS/8mm video tot

circa f 1.500,- voor een Y/C-genlock.

Vrijwel alle genlocks bieden via de moderegelaar de keuze uit:

- ° Het computerbeeld als achtergrond
- ° Het computerbeeld als voorgrond. O.a. voor titels en sleutelgateffecten
- ° Alleen het computerbeeld of inkomende videobeeld wordt weergegeven

Als extra opties zitten er nog eens kleur-, helderheids- en fader-regelaars op. De faders worden in de eerste plaats gebruikt voor het traploos in en uitvloeien van titels of grafische objecten. Desgewenst kan de gebruiker er ook het videobeeld van camcorder of videorecorder mee in of uitfaden. Een aantal exemplaren beschikt tevens over ingebouwde RGB-splitters of zelfs decoders voor tegen kopiëren beschermde videotapes. Een voorbeeld van een dergelijke veelzijdige S-VHS (Hi8) genlock is de S509 van Hama.

Over de beeldkwaliteit van genlocks doen vele verhalen de ronde. In het verleden waren er notoire flikkeraars, strepers en ongewenste dubbelbeelden in de handel. De moderne genlock biedt een redelijk tot goed beeld. Dat geldt zeker voor de S-VHS- en Hi8-typen. Bij goedkope gewone VHS/8mm-modellen is het toch oppassen geblazen. Koop in ieder ge-

val nooit een genlock zonder deze eerst op de eigen video-recorder te proberen! Goede VHS/8mm-genlocks worden in Nederland o.a. door Commodore, Amigis en Catronix (Rendale) geleverd.

Tegenvaller

De consumenten-videostandaard (S-)VHS- en (Hi)8mm-video een compromis zijn tussen de oude zwartwit tv-norm en de daar later bovenop geperste PAL-kleurencodering. Met veel kunstenvliegwerk wordt een goede beeldscherpte en kleurweergave bewerkstelligd. Toch kan deze kwaliteit niet aan de RGB-display van een VGA-kaart tippen. De afzonderlijke aansturing en hoge coderingsruimte van de kleuren Rood, Groen en Blauw geeft een veel strakkere kleurweergave. Ook de scherpte ligt vaak hoger. Er bestaat een flink bandbreedteverschil van bijvoorbeeld 5 MHz bij S-VHS tot meer dan 10 MHz bij PC-video. Let bij de aankoop van dtv-randapparatuur goed op het bereik van de bandbreedte en het aantal beeldlijnen. Die moeten in overeenstemming zijn met de kwaliteit van de opname-apparatuur.

De beginnende dtv-er denkt dikwijls dat al die Amigakleurenpracht en scherpte ook zo op de videotape komt. Helaas niet! De PAL-opname blijkt veelal slechts een mager aftreksel van het RGB-origineel. Felle kleuren blijken te vlammen in plaats van strak te zijn, er treedt verloop op tussen kleurvlakken (bijvoorbeeld tussen letterbody en -schaduw) en de scherpte oogt aanmerkelijk minder. Deze terugval is doorgaans niet de schuld van uw genlock maar die van de genoemde PAL-beperkingen. Door rekening te houden bij de kleurstelling kunnen de zwakke PAL-punten groten-deels omzeild worden.

Bij het kopiëren van banden met genlockopnamen wordt het kwaliteitsverlies steeds



storender. S-VHS en Hi8 zijn hier dankzij de Y/C-techniek beter tegen bestand.

Ondanks deze kritische kanttekeningen geeft de gemiddelde genlock en professioneel ogend PAL-resultaat op videotape. Menige tekstgenerator en videoprocessor brengt het er slechter vanaf.

Digitizers

Een framegrabber of digitizer is een randapparaat dat videobeelden in digitale computerplaatjes omzet. Veel is daar eigenlijk niet voor nodig. Een digitizer-interface, bijvoorbeeld de Digiview 4.0, lijkt op een fors doosje lucifers met een tulpaansluiting. De Digiview digitizer kost inclusief software en RGB-filters circa f 400,-.

Via de AV-uit van uw camcorder of videorecorder (met stilstaand beeld!) kan een zwartwit beeld gedigitaliseerd worden. Voor kleur zijn drie opeenvolgende Rood, Groen- en Blauw-opnamen nodig. Daarvoor kunnen filters of een elektronische kleurscheider, de RGB-splitter (randapparaat van circa f 350,- tot f 400,-, gebruikt worden.

De verkregen resultaten zijn sterk afhankelijk van:

- ° De ingestelde resolutie; Hoe meer puntjes des te scherper het stilstaande beeld
- ° Het aantal gekozen kleuren; Doorgaans is er keuze uit min of meer natuurlijke kleuren of vervreemdings-effecten
- ° De instelling van het contrast, helderheid en shading

De meeste instellingen van digitizers gaan softwarematig. Met de muis wijst de gebruiker de gewenste instellingen op het menu aan.

In het algemeen geldt hoe hoger de resolutie en hoe meer kleuren er ingesteld zijn des te langer het digitizingproces duurt. Normaal is enkele minuten. Bij zeer hoge resoluties kan het wel eens een minuut of 10 worden.



Digitizers worden o.a. gebruikt voor het aanleggen van beeldarchieven, computerdiashows en videovervreemdingen. U kunt het via de Amiga bewerkte still videobeeld altijd weer via PAL-encoder of genlock naar de videorecorder terugspelen.

Gewone digitizers doen lang over het digitaliseren. De veel duurdere real-time digitizers (bijvoorbeeld de VD-4, Amigis) doen de conversie in enkele seconden of zelfs tienden van seconden. Sommige van deze realtimers kunnen ook een bewegende beeldenreeks digitaliseren en weer geanimeerd weergeven.

Voor het opslaan en vloeiend weergeven van gedigitaliseerde beelden dient de Amiga over minimaal 2 MB aan geheugen te beschikken. Leuk meegenomen is dat een door de PC gedigitaliseerd beeld met de printer kan worden afgedrukt. Zo kunt u voor weinig geld redelijke kleuren en zwartwitprintjes van video-opnamen maken.

Gekombineerde videohardware

Is het in uw videostudio ook al een oerwoud van apparatuur? Dtv doet daar nog eens een schepje bovenop. Genlock, videoprocessor, RGB-splitter en digitizer en er staan zo weer vier kastjes extra op het bureau. Een aantal fabrikanten heeft dit ruimten kabelprobleem ook ingezien en biedt de dtv-er een volledig geïntegreerde stu-

dio. Als voorbeeld noem ik de VES_ONE studio voor de Amiga. In één relatief klein kastje zitten een video mixer (genlock) een SEG met drie verschillende wipes, digitizer, RGB-splitter en signaalprocessor (helderheid, contrast en kleurverzadiging). Het titelpakket Videopage met scrollende titels wordt gratis meegeleverd. Dit alles voor iets minder dan f 3.000,-. In Nederland wordt dit apparaat verkocht door CAT&KORSH International (=Catronix). Een gelijkwaardig systeem is de Alladyne VGS van PCB/Lamm met zes verschillende wipes.

Diverse andere dtv-leveranciers werken met modulaire eenheden of videoracks. U begint met één unit of module en stapelt het volgende daar gewoon bovenop. Dergelijke semiprofessionele racks worden o.a. gefabriceerd door het Duitse Video-Comp en Peter Biet Computerdesign (PBC).

Tijdcode-interfaces

Menig videohobbyist droomt er van het gehele draaiboek achter zijn of haar Amigaatje op de monitor in te typen. Staat alles eenmaal op de computer dan doen het tijdcode-interface en de recorders de rest. Na een uurtje koffiedrinken zit daar een perfect gemonteerd tape in de masterrecorder.

Tot op heden is het zover nog niet. De voor amateurdoeleinden ontworpen tijdcode-interfaces voldoen nog niet

echt aan de eisen en zijn tamelijk duur.

Zijn videoscenes eenmaal door een tijdcode, bijvoorbeeld het VITC-systeem, gemarkeerd dan kan een computer de recorderbesturing overnemen en de scenes automatisch in de gewenste volgorde achter elkaar monteren. Voor semiprofessionele doeleinden zijn reeds enkele VITC-montagepanelen, o.a. de Panasonic VW-EC10 en de Hama Videocut, leverbaar. Op zich zijn deze apparaten eigenlijk al kleine computers compleet met programmeerbare microprocessor en IR-afstandsbesturing van de VCR. Dat kunt u tevens merken aan het prijskaartje van meer dan f 3.000,-.

Bij menige computerbezitter zal de vraag rijzen of de Amiga ook met een tijdcodeinterface kan monteren. Dat kan inderdaad, maar is helaas wel een prijzige aangelegenheid. Er zijn diverse mogelijkheden. Het is relatief eenvoudig om een bestaand montagepaneel via een serieel-interface aan de PC te koppelen. Dan is er verder een softwarepakket nodig dat de beschikbare montageopdrachten kan aansturen, bijvoorbeeld via ASCII-codes.

Een andere mogelijkheid is om primair van de Amiga uit te gaan. De Amiga dient dan over een interface te beschikken dat tijdcodesignalen kan traceren en videorecorderbedieningssignalen kan leren. Werd er niet met tijdcode opgenomen dan zal een tijdcodegenerator deze moeten aanbrengen. Uw videorecorder moet dan ook in staat zijn om het tijdcode signaal apart vast te leggen anders moet er een kopieband met tijdcode gemaakt worden!

Tijdcode-interfaces voor Amiga's zijn sinds kort verkrijgbaar. Het gaat hierbij met name om professionele videomontage-apparatuur en besturingssystemen voor grote diashows met meerdere projectoren. De meeste van deze interfaces brengen zelf



hun tijdcode (op een compatibele montagerecorder) aan. De videoproducent geeft op het monitor-draaiboek aan waar wat aan elkaar geplakt moet worden. Het montageprogramma neemt daarna het cut & paste-proces over.

Tot op heden hebben wij nog geen amateur-timecodeunit gezien die voor een reëel bedrag goede resultaten biedt. Het gaat nog voornamelijk om experimentele modellen of peperdure uitvoeringen. Zodra hier meer nieuws over is komen wij op dit onderwerp terug.

Titels in de praktijk

Voor het maken van goede titels op de computer heeft u een videotitelpakket nodig. Deze software maakt het mogelijk om standaardlettertypen tot een professionele video-aftiteling met perfect ogende achtergronden om te "toveren".

Een titel begint altijd met de keuze van het lettertype, de typeface. Vrijwel alle titelsoftware beschikt over voorbeelden (letterstalen) van typefaces zodat u vooraf een keuze kunt maken. Daarna volgt de fontkeuze. Een font is een bewerking van het lettertype. Die bewerking kan zowel in grootte (bijvoorbeeld een 24- of 48-punts letter), stand (schuin of italics) als dikte (vet of bold) zijn.

Fonts worden aangeleverd op diskette. Met het titelpakket kiest u het gewenste font uit en gaat dan naar het kleuren-

palet. Behalve de kleur van de letters is het bij de meeste titelsoftware tevens mogelijk om schaduw (drop of cast)- en stoboscopische effecten in verschillende kleuren toe te voegen. Verder biedt de titelsoftware de gebruiker de mogelijkheid tot het verplaatsen, centreren en uitlijnen van de tekst. Enkele titelpakketten beschikken ook over een beperkte tekenoptie voor kaders, rechthoeken en cirkels. Bij de achtergrond kan de titelmaker uit diverse kleuren (van puur tot mengtinten), gradaties en structuren kiezen. Professionele pakketten zoals TV*Text Professional en Aegis Videotitler bieden alle titel- en achtergrondmogelijkheden die u zich maar kunt wensen.

De mogelijkheden voor scrollen en wipes zijn sterk pakket gebonden. Bij de een is het nodig om daarvoor een extra softwaremodule te kopen. Bij de ander zijn deze scroll- en wipe-opties reeds ingebouwd. Voor goede titelsoftware betaalt u gemiddeld f 150,- tot f 350,-.

Al die geavanceerde titelmogelijkheden dagen uit tot experimenteren en gewoon elke optie eens uit te proberen. In de praktijk blijken er echter toch een aantal titelregels te gelden om tot professionele resultaten te komen. Deze titelregels zijn in het kort:

- ° Hoe groter de letters des te scherper, duidelijker en rustiger de tekst

- ° Pas op met knal blauwe, rode en groene achtergronden. Die gaan snel vlammen. Een neutrale lichte mengkleur doet uw letters veel beter uitkomen

Veel gebruikte letterkleuren zijn wit, geel en rood. Rode letters neigen snel tot onrustig vlammen en uitlopen in de omgeving. Wit en geel zijn veel rustiger. Pas op met sterke contrasten zoals knal gele of rode letters tegen een knal blauwe achtergrond. Dergelijke combinaties vragen het uiterste van uw genlock of PAL-kaart

- ° Wees zuinig met schaduwen en stoboscopische effecten

Voor titels geldt het al oude gezegde "In de beperking kent men de meester". Eenvoudige letters doen het vaak veel beter dan protserige titels.

Bij de Amiga-display is het veelal niet nodig dat het gehele monitorscherm wordt uitgevuld. Bij het overnemen van complete titelschermen echter wel, want anders ontstaan er later zwarte randen bij de video-opname. Gebruik daarom titelsoftware uitsluitend in de PAL overscanmodus, 704 x 564 pixels, waarbij het gehele beeld bedekt is.

Video- en animatiesoftware

Onder videosoftware verstaat men in het algemeen animatiesoftware die via een of andere scriptvorm werkt. In

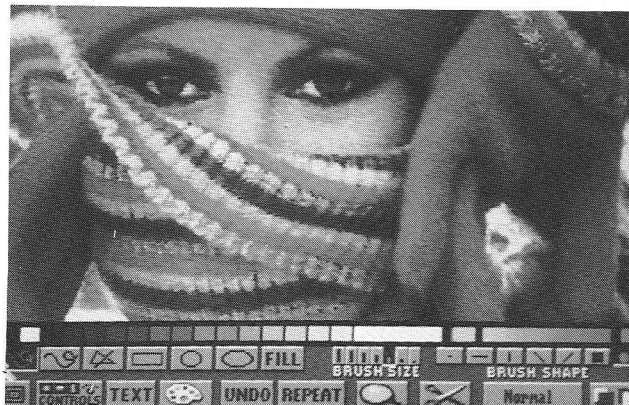
principe maakt videosoftware een soort storyboard met tijdbalken aan. Elke tijdbalk wordt een spoor of track genoemd. Er zijn tracks voor titels, achtergronden, animaties en geluid.

De effecten en gebeurtenissen worden in de vorm van blokjes of andere markers op de tijdbalk gezet. Bijvoorbeeld een fade in, gevolgd door een move to (beweeg naar) en disappear (verdwijn). Een object komt zo op tijdstip x het beeld binnen. Beweegt vervolgens naar een opgegeven lokatie. En verdwijnt tenslotte weer.

Door meerdere tracks met elkaar te combineren kunnen complete animatiefilms met meerdere bewegende objecten, wisselende achtergronden, verklarende titels en begeleidende muziek plus geluidseffecten gemaakt worden.

Uiteraard kost het maken van animaties met videosoftware de nodige tijd en planning. Objecten moeten eerst getekend of verzameld worden. Hetzelfde geldt voor de titels en het geluid. Een passend draaiboek is onmisbaar.

Videosoftware, zoals bijvoorbeeld Deluxe Video III en The Director, wordt nogal eens gebruikt voor het maken van interactieve videoproducties. Dankzij tijdslussen kunnen er vragen gesteld en na enige tijd de bijbehorende antwoorden gegeven worden. Eventueel kunt u met deze pakketten ook automatisch afspelende diashows maken.





Een gemiddeld videosoftware-pakket kost tussen de f 250,- en f 350,-. Voor een eenvoudiger animatiepakket zoals bijvoorbeeld, Fantavision, dient de hobbyist 150 gulden neer te tellen. Professionele animatiesoftware voor imponerende logo- en andere effecten beginnen rond de f 225,-. Bijvoorbeeld Animagic voor de Amiga. Voor software met 3D-effecten betaalt men doorgaans meer dan 500 gulden.

Paintsoftware

De tekensoftware op computers is functioneel op vele dtv-gebieden. Er worden steeds meer opties ingebouwd. Tot de toepassingen van paintsoftware behoren o.a.:

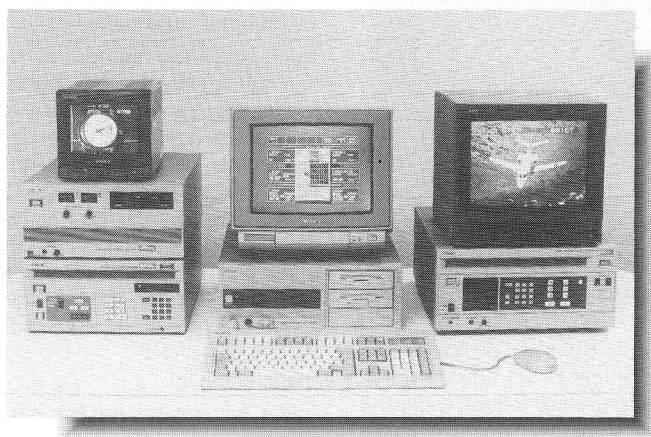
- ° Het ontwerpen van figuren, achtergronden, logo's en artwork
- ° Het bewerken van bestaande PC-illustraties en gedigitaliseerde beelden
- ° Het maken van creatieve titels. Er zijn minder mogelijkheden dan bij echte titel software, maar u komt er een aardig eindje mee
- ° Beperkte animaties

Paintsoftware zoals Deluxe Paint III en Digipaint voor de Amiga beschikt over al deze genoemde opties en kost circa f 300,-. Eenvoudiger tekensoftware kost ongeveer honderd tot honderdenvijftig gulden.

Het tekenen is niet zo gemakkelijk als de beginner denkt.

Behalve een flinke dosis artistiek talent is er een leertijd van enkele dagen nodig om alles onder de tekenknijp te

meegeleverd. Eventueel kan de videomaker ook leentjebuur spelen bij illustratiepakketten voor kranten en



krijgen.

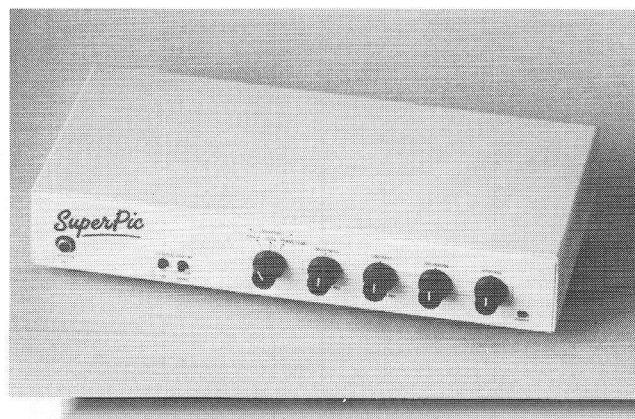
Bibliotheken

Voor dtv-toepassingen zijn diverse symbool-, achtergrond-, clipart en fontbibliotheken te koop. Een bibliotheek heeft altijd een verwerkend programma nodig. Bijvoorbeeld teken- of videosoftware. Anders kunnen de illustraties of letters niet worden ingeladen. De keuze is groot en de prijzen variëren van enkele gulden tot enkele tientjes per afbeelding. Het is verstandig om de inhoud van de bibliotheken eerst te bekijken alvorens een diskette te kopen. De illustratie-inhoud zou wel eens niet aan uw verwachtingen kunnen voldoen. Bij de meeste tekensoftware wordt een beperkte symbool- en/of clipart-bibliotheek

nieuwsbrieven.

In deze Commodore-Info Special over desktopvideo konden wij u slechts een blikje gunnen in deze veelzijdige en creatieve videokeuken. De Amiga kan videofilms echt afmaken door het toevoegen van professionele titels, illustraties en animaties. Een ware verruiming van de creatieve montage mogelijkheden. De kosten zijn in verhouding tot andere professionele aftitel- en animatie-apparatuur gering, het gaat allemaal veel sneller en het bedieningsgemak blijkt vaak groter. Bij aankoop is het verstandig om goed op de compatibiliteit met de bestaande VHS-, 8mm- of Y/C-opname-apparatuur en het type Amiga rekening te houden. Dat voorkomt onnodige teleurstelling. Laat u bij voorkeur deskundig adviseren door een goede computer-shop, Commodore Nederland of de importeur van videorandapparatuur. Zie verder de lijstjes met deskundige leveranciers en dtv-boeken.

U.S.



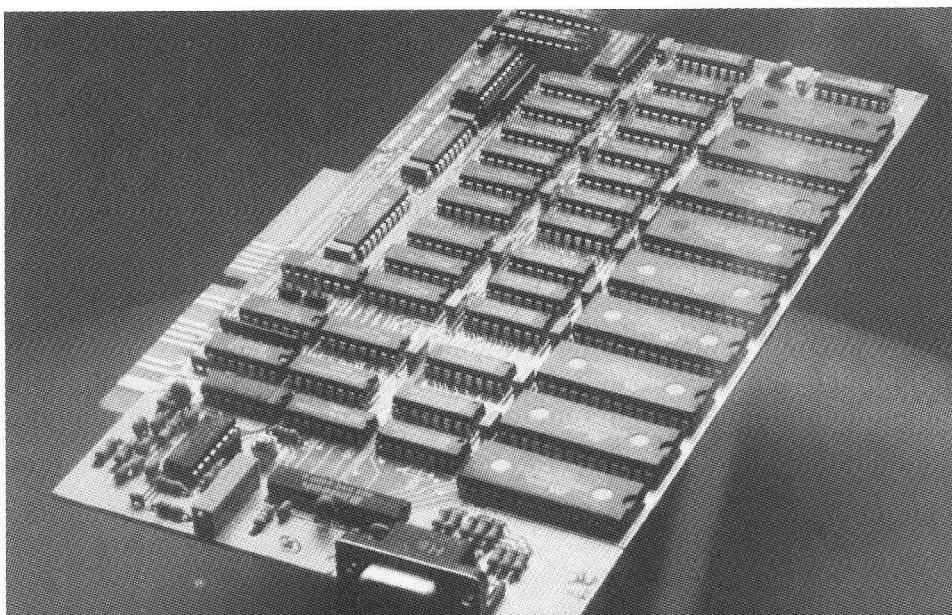
HIRES VIDEOKAARTEN

Betere scherpte en einde van de flikkering

De videonorm van de Amiga 500 en 2000/2500 stamt eigenlijk nog uit het 1000-tijdperk. Bij de introductie van het Model 1000 was een oplossendvermogen van 640 x 256 pixels al heel wat. Later kwamen daar nog enkele hogere resoluties zoals de HIRES Interlace met 640 x 400 pixels en overscan 704 x 512 pixels bij. En daarmee begon het geflikker want bij een dubbele beeldlijnoptekening op 25 Hz is het beeld wel scherp, maar allerm minst stabiel. Indien u een Amiga 3000- of de PC VGA-standaard uit de 2000/2500 wilt halen zit er niets anders op dan daar een speciale Amiga VGA- kaart in te steken. Wij bekeken een drietal van dergelijke AmigaVGA-kaarten voor u.

De Commodore Amiga staat bekend om zijn fraaie kleurrijke graphics. De veelkleurige displays zijn meestal van een lage resolutie. LORES meet niet meer dan 320 x 200 pixels. Wie meer beeldlijnen wil ziet het aantal beschikbare kleuren snel zakken en de neiging tot flikkering toene men.

Alleen de Commodore Amiga 3000 is van huis uit in staat om een hogere beeldresolutie dan 704 x 512 (PAL overscan) pixels te halen. Daarvoor zorgen de Enhanced Chipset en een ingebouwde Flikkerfixer-chip. Hetzelfde kan ook bij de Amiga 2000/2500 via uit-



AGA 2000 flickerFixer

breiding met een HIRES videokaart. In de praktijk gaat het om een VGA-kaart met anti-flikkering-hardware en een eigen video-RAM aan boord. Daarmee kan de kwaliteit van Enhanced VGA bij de MS-DOS compatibele PC geëvenaard of zelf overtroffen worden.

Lastig geflikker

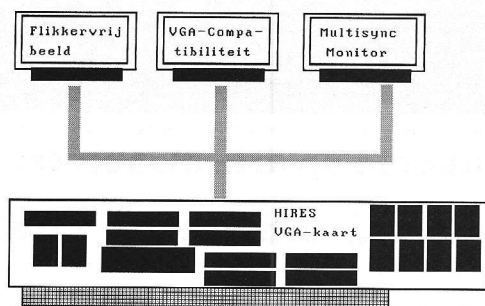
In Europa is een monitorbeeldfrequentie van 50 beelden per seconde standaard. Deze beeldfrequentie komt overeen met de netstroomfrequentie van 50 Hz. Nu blijkt deze frequentie voor het menselijk oog niet zo rustig als het beeld op het eerste ge-

zicht lijkt. Na enige tijd treedt een vermoeidheid op die aan de beeldflikkering op 50 Hz kan worden toegeschreven. Bij bewegende beelden valt het allemaal nogal mee. Bij stilstaande gedetailleerde graphics en tekst slaat de vermoeidheid al snel toe. Aanbevolen wordt een monitor met een beeldfrequentie van 60-70 Hz. Daar is dan echter wel een geschikte videokaart voor nodig.

Een complicerende factor kan de Amerikaanse 60 Hz netfrequentie zijn. Amerikaanse computers die van NTSC naar PAL gemodificeerd werden lijden dikwijls aan een versterkte flikkering

door het verschil in netfrequentie. Het beeld bleekt op de lange duur te onrustig om plezierig naar te kijken.

De Amiga werkt op de lagere resoluties met zogenaamde **non-interlace video**. Dat betekent dat de beelden als één geheel op de monitor getrokken (met de kathodestraal) worden. In het standaard Amiga-beeldformaat worden daarbij bijvoorbeeld 640 x 256 pixels in 16 kleuren opgetekend. Om de verticale resolutie tot 512 lijnen te verdubbelen wordt via de **interlace-techniek** eerst een frame met oneven beeldlijnen en daarna een frame met de even beeldlijnen geschreven. De beide frames geven bij el-



kaar opgeteld een scherper beeld dan bij de non-interlaced videodisplay. Helaas moet de gebruiker hiervoor een stevige prijs betalen. Daar het fosfor van het monitorscherm niet lang genoeg nagloeit, de deeltjes doven al licht uit voordat het tweede beeld is opgetekend, gaat de display flikkeren. Bovendien neemt het beschikbare kleurenpalet af. Bij overname op VHS of 8mm video verdwijnt de flikkering weer, omdat de PAL norm met 25 beelden per seconde werkt en het flikkereffect, met behoud van scherpste, weer te niet doet. Daar heeft overigens alleen de dtv-er iets aan en niet een Amiga-gebruiker normaal lang naar het beeldscherm moet kijken.

Wat doen Amiga HIRES VGA-kaarten?

Alle drie hier besproken VGA-kaarten doen in feite vier dingen:

- ° Bieden en **hogere VGA-compatibele resolutie** tot aan Amiga 3000-kwaliteit toe
- ° **Elimineren de interlaced beeldflikkering**
- ° Bieden een **eigen extra video-RAM**
- ° Maken het **gebruik van een PC VGA-monitor mogelijk**. In de praktijk verdient een goed afregelbaar multisync-type de voorkeur

Toch kunnen de kaarten qua oplossend vermogen en verschillende schermmodi nog verschillen. daarom bespreken wij hen elk afzonderlijk.

De Flickerfixer

Microway was de eerste firma die een eind aan het interlaced geflikker maakte. Met de inmiddels beroemde **AGA 2000 Flickerfixer**-uitbreidingskaart kon de Amiga-bezitter eindelijk flikkervrij van een dubbele verticale beeldresolutie genieten.

De Flickerfixer voorziet de Amiga 2000/2500 van een VGA-compatibele display met een horizontale scanfrequentie van 31,5 KHz. De beeldfrequentie zelf bedraagt een acceptabele 50 Hz. In de PAL-versie geeft de AGA 2000 een maximale beeldresolutie van 704 x 510 pixels. Daarbij blijft de Flickerfixer iets onder de E-VGA-norm van 800 x 600 pixels en blijft de verticale overscan licht in gebreke.

De Flickerfixer biedt natuurlijk de gebruikelijke grafische modi LORES, HIRES, Extra Halfbrite en HAM. Voor gebruik als VGA-kaart wordt een multisyncmonitor aangeraden. De aansluiting is een standaard 9-pins RGB-connector. Optionele uitbreiding voor genlock-toepassing is mogelijk.

De antiflicker-truc die de Flickerfixer toepast is die van het beeldlijngeheugen. De oneven lijnen worden in het video-RAM bewaard totdat de even lijnen er aan komen. Vervolgens wordt één beeld in zijn geheel in plaats van twee opeenvolgende frames op de monitor weergegeven. Daarmee vervallen de donkere lijnen en de uitdovende fosforpartikeltjes die de sto-

rende beeldflikkering veroorzaken. Omdat het video-RAM net te krap is is helaas geen optimale verticale overscan mogelijk.

In de praktijk voldeet de Flickerfixer redelijk tot goed. Weg flikkering, betere beeldscherpte en contrast. Snelle animaties toonden soms enigszins hoekig en schokkerig en de verticale overscan is zoals gezegd te krap.

De prijs van deze Amiga VGA-kaart ligt rond de f 1.250,-. O.a. verkrijgbaar bij Amigis, tel.: 01180-25632.

Highgraph V

In Duitsland vindt men tal van kleine Amiga-hardware-fabrikanten zoals de in ons land tamelijk onbekende firma Ralf Jochheim Computer Tuning. De nieuw ontworpen (release November 1990)

Highgraph V haalt met 832 x 620 pixels ruim de E-VGA-norm. Dankzij het 768 KB meten on board video-RAM slaagt deze kaart er net in de verticale overscan te halen. Wat de monitor betreft heeft u weer een 31,5 MHz VGA-model nodig. Een multisync-type verdient weer de voorkeur hoewel de kaart via twee potentiometer goed op een E-VGA-monitor in te regelen valt.

In de praktijk geeft de Highgraph V een goed scherp en kontrastrijk beeld. De beeldflikkering wordt door deze 50 Hz kaart naar tevredenheid gereduceerd.

De Highgraph V kost circa f 1.000,-. Een nederlandse importeur is ons onbekend. Leverancier: Ralf Jocheim Computer Tuning, Osnabrückerstrasse 96, 4802 Halle, Duitsland.

X-tension Pro-Video

Van professionele allure is de **X-tension Pro-Video**-kaart van de duitse firma IOAG. Deze kaart onderscheidt zich van zijn beide voorgangers door de hogere beeldwisselfrequentie van 60 tot 70 Hz!. Daarmee slaagt de X-tension er zeer goed in om de storende beeldflikkering terug te

brenge. De kaart slaagt er zelfs in de Amber Flickerfixer-chip van de Amiga 3000 te kloppen. Helaas werkt zo iets alleen met peperdure monitoren die deze hoge beeldfrequentie aankunnen.

Een beperking van deze kaart is die van de maximale Amiga-schermgrootte, 640 x 512 pixels. Overscan zit er dus niet bij en dat is treurig voor de dtv-er. Een tweede bezwaar vormt de hoge prijs van circa f 2.000,-. Leverancier: IOAG, Bubenreuther Strasse 23, 8523 Beiersdorf-Iglesdorf, Duitsland.

De keuze

Welke VGA-kaart nu voor uw Amiga 2000/2500? Gaat het met name om een doorsnee stabiel VGA-beeld dan komen zowel de AGA 2000 Flickerfixer als de Highgraph V in aanmerking. De laatstgenoemde is vanwege de optimale overscan beter geschikt voor dtv toepassingen. Wie een optimaal flikkervrij beeld bij dtp zoekt kiest voor de X-tension Pro-Video. Door het gebrek aan overscan is deze kaart minder geschikt voor desktopvideo-toepassingen.

U.S.



KCS POWER PC BOARD

Zin of onzin op de 500

Met gratis MS-DOS-emulator, stond er vaak in de advertenties te lezen. Sommige Amiga 500's werden verkocht met een software-emulator die MS-DOS-programma's kan laten lopen op een Amiga. In de praktijk bleek dat behoorlijk tegen te vallen. De emulator, Transformer genaamd, bleek onbetrouwbaar en te weinig flexibel. Lang niet alle software voor MS-DOS was te runnen en de snelheid was vergelijkbaar met die van een slak.

Voor de grotere broers van de 500 waren er al snel mogelijkheden in de vorm van insteekkaarten, waar de 500 niets aan had. Als eerste op

de wereld (JA ECHT) heeft de Nederlandse firma KCS (Kolff Computer Supplies) uit Dordrecht een heuse MS-DOS-emulator op de markt gebracht die de concurrentie met de bestaande XT- en AT-

kaarten aardig aan kan. Wij testten de Alfa-versie 1.50 ag.

Inclusief DOS

Wie het KCS PC Power Board koopt, krijgt een perfect uitziend pakket met daarin DOS 4.01, het Board met de software en de nodige Nederlandstalige handleidingen. In tegenstelling tot normale prints, die meestal groen zijn, is het KCS Board rood. Op zich maakt dit hem al speciaal. Dat KCS standaard MS-DOS meelevert, weegt mee in de prijs. Maar zonder DOS ben je nergens. En het daarna nog apart kopen, kost zeker zoveel. Installatie is zo gebeurt. Het Board past namelijk precies in de onderkant van de Amiga 500, waar normaal een geheugenuitbreiding in gaat. Zij die dus daar al een andere

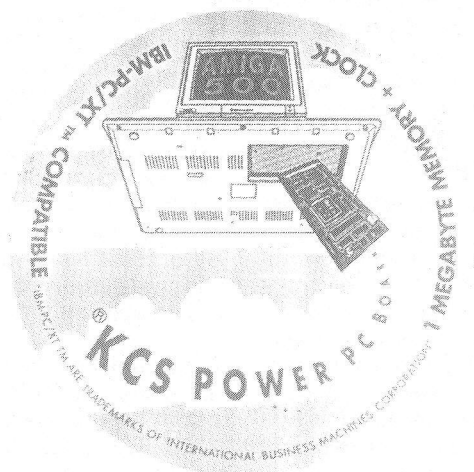
kaart hebben zitten, moeten deze verwijderen.

De kaart is niet voorzien van een aan/uit-schakelaar, maar de handleiding voorziet wel in een paar aanwijzingen om dit simpel voor elkaar te krijgen.

Diskette

Het Board werkt niet vanzelf. Zet je de Amiga na installatie aan, dan verschijnt de normale 'Amiga-hand'. Om de emulator te activeren, moet er worden opgestart met een Amiga-schijf die is meegeleverd. Daarop staat de software voor het opstarten van de kaart. Na enige ogenblikken verschijnt dan een MS-DOS-schermdie vraagt om een systeemdisk, terwijl hij net als een echte MS-DOS-computer het geheugen controleert. In die tussentijd is er dus tijd van schijf te wisselen. Na de geheugencheck wordt het MS-DOS besturingssysteem geladen en enkele ogenblikken later is de Amiga veranderd in een MS-DOS-computer.

Het verschil met de situatie bij de Amiga 1000 met Sidecar en de 2000 met XT- of AT-kaart is dat de Amiga 500 in dit geval niet multitasking is. Althans in die zin dat hij niet tegelijkertijd in MS-DOS- en in Amiga-modus kan draaien. Dat houdt dus in dat bestanden niet zonder meer uitwisselbaar zijn.



```
C>si
SI-System Information, Advanced Edition 4.50, (C) Copr 1987-88, Peter Norton

Computer Name: IBM PC
Operating System: DOS 4.00
Built-in BIOS dated: Thursday, January 18, 1990
Main Processor: NEC V30
Co-Processor: None
Serial Ports: 2
Parallel Ports: 1
Video Display Adapter: Monochrome (MDA)
Current Video Mode: Text, 80 x 25 Monochrome
Available Disk Drives: 4, A: - D:

DOS reports 784 K-bytes of memory:
 92 K-bytes used by DOS and resident programs
612 K-bytes available for application programs
A search for active memory finds:
640 K-bytes main memory (at hex 0000-A000)
128 K-bytes display memory (at hex A000-C000)

Computing Index (CI), relative to IBM/XT: 3.3
Disk Index (DI), relative to IBM/XT: Not computed. No drive specified.
Performance Index (PI), relative to IBM/XT: Not computed.

C>
```

De oude Norton index...



```
C>si
SI-System Information, Advanced Edition 4.50, (C) Copr 1987-88, Peter Norton

Computer Name: IBM PC
Operating System: DOS 4.00
Built-in BIOS dated: Thursday, January 18, 1990
Main Processor: NEC V30
Serial Ports: 2
Co-Processor: None
Parallel Ports: 1
Video Display Adapter: Monochrome (MDA)
Current Video Mode: Text, 80 x 25 Monochrome
Available Disk Drives: A, B, C, D:

DOS reports 784 K-bytes of memory:
  92 K-bytes used by DOS and resident programs
  612 K-bytes available for application programs
A search for active memory finds:
  640 K-bytes main memory (at hex 0000-A000)
  128 K-bytes display memory (at hex A000-C000)

Computing Index (CI), relative to IBM/XT: 4.2
Disk Index (DI), relative to IBM/XT: Not computed. No drive specified.
Performance Index (PI), relative to IBM/XT: Not computed.
C>
```

De nieuwe...

Extra geheugen

Tja, daar zit je dan met een PC Power Board en zonder geheugenuitbreiding, want die moest je net verwijderen. Mooi niet! De emulator heeft namelijk 1 MB aan RAM. Dat betekent dat je intern 1 MB aan geheugen (fast RAM) hebt en nog eens 512 KB in de vorm van een ram-disk (KCSRAM:). Een klok is ook aanwezig, maar die is niet compatible met het 'set clock-commando' van de WorkBench-diskette. Daarvoor levert KCS een ander commando mee. Vanaf de Amiga-opstartdisk kunnen diskette worden gecheckt op de aanwezigheid van het oude commando 'SetClock opt load'. De software zet er een ';' voor, zodat dit commando niet meer herkent wordt. Het nieuwe commando 'PCCLOCK' vindt elders in de startup-sequence een plek. De klok kan overigens ook vanaf de PC-kant worden aangeroepen. Het geheugen voor de PC bedraagt 704 KB. De rest gaat op aan de emulatie.

Kleuren

De Transformer kende maar twee kleuren. MS-DOS-programma's die kleur ondersteunden, kwamen daarmee maar matig over. De kaart van KCS emuleert MGA en CGA. MGA is vergelijkbaar met Hercules (monochrome grafische kaart). CGA onderscheidt acht of zestien kleuren (in de nieuwe versie 1.50 ag) afhankelijk van het aantal dat de gebruiker wil. Normaal gesproken kent CGA

zestien kleuren, maar om het Board sneller te maken kan er ook voor acht kleuren worden gekozen. Bij de pakketten, zoals Lotus, wordt er automatisch omgeschakeld, mocht dit nodig blijken.

Configuratie

Om de instellingen van het Powerboard te veranderen kan vanaf de Amiga-diskette een menu zichtbaar worden gemaakt. Daarvoor moet een muisknop bij het starten worden ingedrukt, zodat de emulatie niet wordt opgestart, maar er een menu verschijnt met een aantal opties, waaronder de instellingen aan de MS-DOS kant. Vooreerst kan hier de taal worden ingesteld, die Nederlands, Duits, Engels of Frans kan zijn. De nieuwste versie kent nu ook Portugees. Dit maakt het Powerboard al heel erg internationaal. Verder kunnen hier de diskdrives worden ingesteld (vier in totaal), de video-modus (MGA/CGA), de printer, het geluid, de muis, de RS-232-poort, de joystick en het toetsenbord. Helaas kende de Alfa-versie nog geen harddisk-ondersteuning. De instellingen hebben over het algemeen met de poorten te maken. Zo kan bijvoorbeeld worden aangegeven, aan welke poort de emulatie de muis moet zoeken.

Het opstarten van MS-DOS gaat razensnel. Natuurlijk kunnen hier config.sys en autoexec.bat naar believen met commando's worden gevuld.

Voor iemand die de Amiga-muis wil herkennen, moet over een muis-driver van 'Mouse Systems' beschikken; deze wordt niet meegeleverd. In de instellingen staat standaard ook een keuze voor een Microsoft muis. Maar wie deze drive laadt, kan met de Amiga-muis weinig doen. Dat die mogelijkheid wordt geboden, komt doordat KCS hier wel aan werkt.

Menu onder MS-DOS

Een aantal instellingen kan ook in MS-DOS-mode worden opgeroepen. Linker-Amiga-toets en Help tegelijk drukken tovert een Amiga-menu te voorschijn, waarin bijvoorbeeld de kleuren kunnen worden aangepast mochten die tegenvallen tijdens het draaien van een programma. Dit menu wordt trouwens gewoon over het draaiende programma heen gelegd (hoezo geen multitasking?)

Het geluid wordt ten dele ondersteunt. Bij spelletjes kregen we wel eens rare toestanden. Het originele geluid werd dan wel eens vervormd. Maar voor de zakelijke pakketten geen enkele probleem.

Het beeld wordt qua grootte prima ondersteunt. KCS heeft kans gezien om de lampjes van de 'Lock-toetsen' linksonderin beeld te laten verschijnen. De nieuwste versie heeft verder een mogelijkheid om rechtsonderin beeld een trackdisplay te laten zien. Daar staat dan bijvoorbeeld te lezen: 'A 00 B 02 C 00 D —'. Drive A en C staan in dit geval in track 0, drive B staat in track 2 en drive D werd tot nog toe niet aangesproken, maar is wel actief.

Het Power PC Board is in feite een complete PC. Hij is opgebouwd rond een NEC V30 processor. Via Norton SI geeft het board een snelheid van 3.3 aan. De nieuwe versie kwam al op 4.4 en het

eindresultaat van de allernieuwste versie, die ook harddisks zal ondersteunen, zal 5.0 aan moeten geven, volgens KCS.

Pagemaker

Je kunt je afvragen wat je aan dit Powerboard hebt. Wat kun je er mee? Het enige dat deze kaart nadelig maakt, is dat het geen harddisk ondersteunt (straks dus wel). Want de meeste MS-DOS-pakketten vragen tegenwoordig een harddisk minimaal, want anders werkt het niet. Zit de software er eenmaal in, dan loopt het als een trein, mits er voldoende geheugen voorhanden is. En dat was bij ons wel het geval. Je moet natuurlijk niet met pakketten als Pagemaker proberen of dat misschien op het Power PC Board werkt, want daarmee wordt een open deur ingetrapt. Het gaat bij de KCS-kaart alleen maar om standaardgeheugen. Dus zodra we het gaan hebben over expanded en extended geheugen, dan moet-ie afhaken.

Iemand die alleen zo nu en dan een paar files moet overzetten naar MS-DOS hoeft deze emulator niet aan te schaffen. Daarvoor zijn goedkopere oplossingen te bedenken, zoals de softwarepakketten DOS-2-DOS en CrossDOS, waarbij een Amiga-drive tot (MS)-DOS-drive wordt omgedoopt.

Maar moet er regelmatig dBase of Lotus worden gedraaid, dan kan het Powerboard goede diensten verrichten. Wie daarbij in het achterhoofd houdt, dat versie 1.50 ag in de definitieve versie ook harddisks ondersteunt, doet een goede koop met een oneindige update-mogelijkheid. Want KCS vraagt voortdurend om nieuwe ideeën voor deze emulator, die bij voldoende belangstelling in de software worden verwerkt.



AMIGA VGA- EN MULTISYNCMONITOREN

Optimale beeldscherpte en rustige display

In ons artikel over VGA-kaarten met een hoge resolutie kwam al uitvoerig ter sprake dat er voor een HIRES non-flickering VGA-beeld op de standaard-Amiga een optionele kaart en monitor nodig zijn. De Amiga 3000 beschikt al standaard over een FlickerFixer-chip en HIRES VGA-achtige resoluties. In beide gevallen kunt u kiezen uit een groot aanbod in VGA- en Multisyncmonitoren. Dit artikel gaat in op de monitortechniek en verschillende monitortypen. Ook worden de gangbare technische monitorspecificaties kort uitgelegd.

Een goede monitor is goud waard. Groot of klein, in kleur of monochroom in alle gevallen bepaalt deze beeldbuis uw Amiga- kijk- en werkgenot.

De aanschaf van een slechte monitor is zonde van alle Amiga- investeringen. Beter een goede koop in eens dan veel ergernis achteraf. Na het lezen van deze Amiga monitorenspecial kunt u een verantwoorde keuze maken.

Verschillen in Amiga-video

Wat voor type monitor u op de Amiga kunt aansluiten hangt van het desbetreffende model af. In het eenvoudigste geval is het mogelijk om een Amiga 500 of 2000 op een



gewone TV aan te sluiten. Daarvoor zit dan een tulpvideo-aansluiting aan de achterzijde. Indien de machine niet over een PAL-kaart/chip beschikt dan zal het beeld monochroom zijn. Dat is te verhelpen door een TV-modulator (wordt los door Commodore geleverd) tussen de Amiga en TV-set te schakelen. Dan krijgt u een normaal PAL TV- beeld dat voldoet voor spelletjes. Bij tekstverwerking en HIRES- graphics krijgt de gebruiker daar echter al snel hoofdpijn van. De kleurstelling is niet zo fraai, de scherpte valt dik tegen en de beelden flikkeren onrustig.

In de meeste gevallen zal een koper van een Amiga Model 500 of 2000 daar een Commodore Monitor bij aanschaffen. Het gaat dan om een standaard RGB kleurenmonitor, bijvoorbeeld een 1084S, met 640 x 512 pixels (standaard Amiga HIRES gebruikt 640 x 256 beeldpunten) en een lijnfrequentie van 15,625 KHz. Deze monitor neemt kwalitatief gezien een soort middenpositie tussen een CGA- en een goede EGA-display in. De rasterfrequentie is 50/60Hz met een redelijk stabiel beeld. In de interlacemodus wordt het echter een storend geflikker.



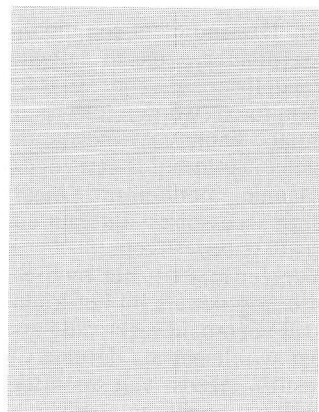
De Amiga 2500 is het eerste model dat met een Flicker-Fixer is uitgerust. Deze videokaart geeft een display die tussen gewoon VGA 640 x 480 pixels en Enhanced VGA 800 x 600 pixels in ligt. In de praktijk gaat het om 704 x 510 pixels. De rasterfrequentie bedraagt 50 Hz en de lijnfrequentie op circa die van VGA 31,5 KHz. Om deze beeldkwaliteit ook op het beeldscherm te krijgen heeft de Amiga-bezitter een VGA- en liefst een multisync-monitor nodig.

Als laatste het multimedia Amiga Model 3000. Deze Amiga-telg stuurt zowel PAL (15,75 KHz) als een soort Commodore Super-VGA uit. De maximale schermresolutie bedraagt 1280 x 512 pixels (4 kleuren display). Verder levert de nieuwe Denise- met FlickerFixer-video chip ook diverse VGA-modussen waaronder 640 x 480 non-interlaced. De display Enhancer geeft een horizontale lijnfrequentie van 31,5 KHz en u heeft dus weer een goede VGA- of multisync-monitor nodig. N.B.: Een multisyncmonitor is geschikt voor meerdere horizontale lijnfrequenties. De monitor regelt zich zelf automatisch op het aangeboden videosignaal af. Commodore zelf levert onder het eigen merk zowel VGA- als multisyncmonitors.

Inmiddels heeft u nu een indruk van welk type monitor bij de eigen Amiga past. Dan wordt het nu tijd om eens naar de verschillende uitvoeringen, typen en de technische specificaties te kijken.

Kleur of monochroom?

Op de Amiga voert de kleurdisplay de boventoon. Toch zijn er applicaties zoals desktoppublishing en tekstverwerken te bedenken waarbij een monochroom beeld meer rust en scherpheid geeft. Ook bij databases en spreadsheets is niet altijd een kleurdisplay nodig.



Voor de Amiga met een VGA-kaart zijn er diverse kwalitatief goede monochrome VGA-monitors te koop. Bijvoorbeeld van Wyse, Samsung, Philipsen en NEC. Ook Commodore zelf biedt in de vorm van de A2024 een prima monochrome beeldbuis aan.

Een voordeel van de zwart-wit-monitor kan ook de aanzienlijk lagere prijs ten opzichte van een kleurentype met dezelfde beeldkwaliteit zijn.

Een van de belangrijkste argumenten om toch met kleur in plaats van monochroom te werken is het verhogen van de subjectieve zichtbaarheid. Weergave in verschillende kleuren valt nu eenmaal eerder op dan het gebruik van 16 of meer grijstinten(/trappen). Bovendien gaat de grafische userinterface (Werkbank) uit van een gebruiksvriendelijke kleurendisplay. Hetzelfde geldt voor de gebruikersinterfaces van de meeste Amiga-software.

Bij het maken van technische of artistieke ontwerpen, business graphics en video-animaties is kleur gewoon en must en maakt het meer details zichtbaar. Door het selectief aanbrengen van kleur wordt het oog geattendeerd op bepaalde details of ontwikkelingen. Idem dito bij diverse grafische software-interfaces waarbij de ontwerpers het gebruik van de diverse functies, boodschappen, waarschuwingen e.d. door middel van kleur accentueren. Bijvoorbeeld geel betekent "optie actief", een rode

boodschap "Pas op er dreigt een fout!".

Grote of kleine buis?

De grootte van het monitorscherm door de fabrikant opgegeven als de **beelddiagonaal** in inches. Bijvoorbeeld 12-, 14-, 16- en 19/20- inch. Slechts een enkele maal wordt het bruikbare beeldoppervlak vermeld. De beelddiagonaal zegt alleen iets over de afstand linker onderhoek tot rechterbovenhoek van het monitorscherm. Hoe groter die diagonaal des te groter het schermoppervlak in lengte en breedte uitpakt.

Het **bruikbare beeldoppervlak** zegt hoeveel van die beeldbuis u ook daadwerkelijk kunt benutten. Bij dure topmodellen zal dat altijd bijna de gehele beeldbuis zijn. In het geval van een goedkopere monitor blijkt nogal eens dat men het bruikbare beeld wat kleiner dient in te stellen. Een te krappe voeding of vervormingsproblematiek (overscan) verpesten dan het beeld bij het gebruik van het volledige scherm. Grote schermen zijn aantrekkelijk voor gebruikers die veel op één scherm tegelijk willen zien. Bijvoorbeeld desktoppublishers, grafische ontwerpers, CAD/CAM-ingenieurs, ontwerpers van elektronische prints, bij image processing & analysing en gebruikers van grote spreadsheets. Grote schermen hebben in het algemeen drie nadelen:

- ° Zij zijn extreem duur
- ° De scherpheid oogt minder dan bij kleinere beeldbuisen
- ° De beeldbuis is bij grote maten moeilijk vlak te houden en dat geeft vervorming van rechthoekige graphics. O.a. lastig bij beeldbuisfotografie

Een vierde bezwaar was de grote omvang en het gewicht van de monitor. Gelukkig wordt bij de moderne monitortechniek dit probleem steeds beter ondervangen.

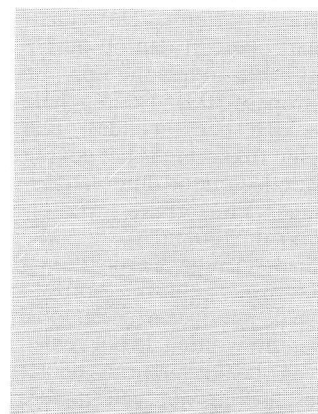
Toch blijft een 19/20 inch scherm een grote bak.

Goedkopere Amiga-systemen worden om prijs-attractieve redenen vaak standaard met kleine 14 inch kleurenmonitors geleverd. Wie groter wil zal fors in de beurs moeten tasten.

De scherpheid

Wanneer is een monitor nu echt scherp? Dat hangt van tal van factoren af. Als eerste het **oplossend vermogen in pixels** (komt overeen met de snijpunten van het aantal horizontale en verticale lijnen) per beeldoppervlak. De videokaart geeft een x aantal pixels dat het beeldscherm vult. Bij grote schermen wordt de onderlinge pixelafstand groter en toont het beeld minder scherp. Daarom gebruikt men bij voorkeur HIREs-videokaarten bij grote monitors.

Als tweede factor de **(be)kijkafstand**. Iedereen die wel eens bovenop een tv heeft gezeten is bekend met de onscherpte en trillende indruk van de korte kijkafstand. Een metertje verder gaan staan maakt het beeld al een stuk scherper. Nu meet de beelddiagonaal van de doorsnee huiskamer-kleurentv 55 cm of meer en is daarmee aanmerkelijk groter dan de monitorbeeldbuis. De 12/14 inch PC-monitor wordt van circa 30 tot 50 cm afstand bekeken en oogt dan redelijk tot goed scherp. Bij een grote monitorbuis kan de gebruiker beter een wat gro-





tere bekijfstand in acht nemen.

N.B.: Een goede PAL TV kan maximaal 625 beeldlijnen bij een beeldfrequentie van 25 weergeven. De kabel levert echter slechts 300 horizontale lijnen aan. Een Super-VHS/Hi8 (Y/C-uitgang) camcorder of videorecorder leveren, afhankelijk van het prijskaartje, 320-600 beeldlijnen aan. Een VGA-monitor met bijbehorende videokaart kunnen dus een betere beeldkwaliteit dan de huiskamer-tv neerzetten.

Het **contrast** is de derde belangrijke scherptefactor. Hoe beter de aan- en uitstaande beeldpunten tegen elkaar contrasteren des te scherper het beeld oogt. Samen met een optimale contrastweergave hangt het wegvangen van strooistraling middels een schaduwmasker.

Reflecties maken het beeld slecht leesbaar. Anti-glarevoorzieningen van de beeldbuis voorkomen weerspiegelingen en geven zo een betere scherpte-indruk.

De invloed van de **dotpitch** komt bij het technische hoofdstukje over de resolutie uitgebreid ter sprake.

Als laatste scherpte beïnvloedende factor het **geflikker**. Een monitor zet een 25 tot 50 beeldjes per seconde op het scherm en hoopt dat het menselijk oog deze beeldsequentie als één stilstaand beeld interpreteert. Helaas blijken Amerikaanse tv-normen (NTSC, 525 beeldlijnen, 60Hz) hier nogal eens roet in het eten te gooien omdat daar een iets andere netfrequentie gebruikt wordt. De videokaart opereert dan op 60 Hz halfbeeld-frequentie i.p.v. de Europese 50 Hz (= 25 gehele beelden per seconde). Een lichte flikkering is het gevolg die alleen door dure aanpassingen, FlickerFixers, verholpen kan worden.

Bij de Amiga wordt, indien gewenst, gebruik gemaakt van de interlace-videomodus. Daarbij worden de oneven en even beeldlijnen na elkaar geschreven om een hogere

beeldscherpte te creëren. Daar in de tijd tussen het schrijven van de beide lijnenseries het pixelfosfor al wat begint uit te doven ontstaat een hinderlijk geflikker.

Kleurweergave

De gewenste kleurtint van monitor is een persoonlijke kwestie. Er zijn warme getinte, natuurgetrouwe en koele kleurenmonitoren in omloop. De beeldbuiscoating, gebruikte rasters/maskers en het type elektronenkanonnen spelen daarbij een rol.

De monitor zelf mag nauwelijks tot geen invloed uitoefenen op het aantal kleuren in de display. Daar speelt immers de videokaart of -chipset de hoofdrol. De gebruikte videostandaard (CGA t/m Super-VGA), de RGB-aansturing (TTL geeft minder beschikbare kleuren dan analoog) en de hoeveelheid video-RAM op de kaart bepalen het beschikbare palet en de hoeveelheid kleuren die bij de gekozen resolutie op het scherm kunnen. En verder doet natuurlijk de software een duit in het kleurenzakje.

Ergonomie

In een **verworpen houding** voor het beeldscherm zitten geeft snel lichamelijke klach-

ten. Daarom dient een monitor draaibaar en in hoogte verstelbaar te zijn. Goede monitoren worden met een draai- kantelvoet geleverd. Helemaal ideaal is een zogenaamde monitorarm, maar deze accessoires zijn duur en lastig te bevestigen.

Vergeet niet dat het **menselijk oog** net als bij het lezen snel **vermoeid** raakt. Nog erger wordt het als de datatypist(e) steeds van monitor naar het papier en visa versa moet kijken. Een papier standaard kan hier de belasting verlichten. Neem verder minimaal eens in de twee uur een kwartier pauze of wissel beeldschermwerk met ander werk af.

Over de **RF- en zachte X-straling** van monitoren en andere tv-toestellen doen vele waarheden en onwaarheden de ronde. Het enige duidelijke is dat er momenteel veel onduidelijkheid over de echt schadelijke effecten bestaat. Genoemd worden stress, cataract van het oog, bloedafwijkingen, miskramen, psychische afwijkingen en chromosoomveranderingen. Wie het zekere voor het onzekere wil nemen kan alvast een stralingschermpje voor de beeldbuis plaatsen. Bijvoorbeeld dat

van Pheriphlex Nederland, tel. 02975- 68533. Langer dan twee uur aan één stuk achter het beeldschermzitten wordt ten stelligste afgeraden!. Neem tijdig een pauze en streef naar een zo flikkervrij mogelijke monitor.

Ook van belang zijn de **beeldinstellingen**. Contrast, helderheid, scherpte, beeldcentering (vertical size en horizontal hold), kleur en overschakeling tussen de diverse videomodussen dienen comfortabel en onafhankelijk regelbaar te zijn.

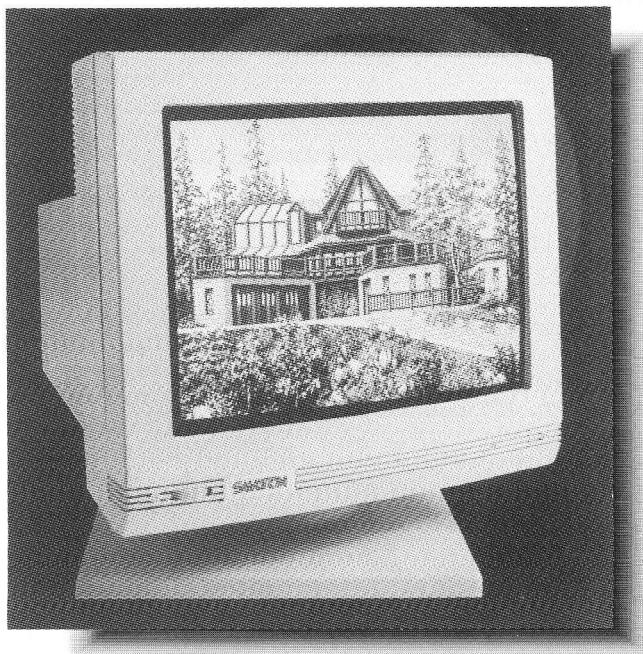
Aansluitingen

Voor het aansluiten van de monitor op de videokaart heeft u een videokabeltje nodig. Bij een aantal monitoren zit dit kabeltje vast aan het toestel. Andere leveranciers geven het videokabeltje los mee.

Op de Amiga treft u afhankelijk van het type een brede platte RGB-stekker (standaard) of een TTL **9-polige monochroom/RGB- videoconnector** (VGA- en/of Flickerfixerkaart) aan. Een variant daarop is de **8- polig RGB-DIN-connector**, eveneens Transistor to Transistor Logic, die op een aantal Amiga-monitoren zit. De verloopkabel wordt al standaard meegeleverd.

Daarnaast bestaan nog de composiet-video-aansluitingen voor PAL- compatibele systemen. Daarbij gebruikt men vaak tulp (cinch-) of BNC-pluggen. Let bij gebruik van de PC als videobron voor VHS op het verschil gewoon- en Super-VHS (Y/C-uitgang!).

Meestal geven videokabeltjes qua verkrijgbaarheid weinig problemen. Soms zijn verloop- of verlengkabels (bijvoorbeeld bij een Towercase onder het bureau) nodig. Wat de **voedingskabel** betreft is er keuze uit losse netstekkers en drie-aderige doorsluitconnectoren die de netspanning uit de PC-systeemkast trekken. Dit laatste type komt op de Amiga niet





voor. Wel bij andere PC-ty-
pen.

Technische termen

De argeloze koper die op zoek gaat naar een geschikte Amiga-monitor wordt geconfronteerd met een hele waslijst aan technische termen. "Stijlbloempjes" als scanfrequenties, bandbreedte, pixels, tridot pitch, resolutie, spot size, digitaal/analoo, time variance, misconvergence, voedingsstabilisatie (voltage regulation), schermcoatings, interlacing en overscan slaan je in de prospectussen om de oren. Het folder- en advertentiepapier is immers geduldig.

Als u deze tv-terminologie helemaal precies wilt weten raden wij u dringend een handboek over tv- en videotechneek aan. Commodore-Info behandelt hier alleen de essentie van de terminologie in zoverre deze van belang is bij de aankoop van een geschikte Amiga monochroom- of kleurenmonitor.

Scanfrequenties

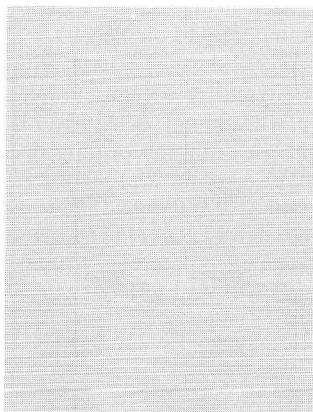
Een tv-beeld is een dynamisch spel van lijnen optekenen. Beeldlijnen worden opgebouwd en vervolgens weer afgebroken. Dit "verversproces" wordt **scannen** genoemd. De **scanfrequentie** is een maat voor de snelheid waarmee de elektronenbundel over de beeldbuis schrijft. Er zijn twee soorten scanfrequenties:

• De **vertikale scanfrequentie** is de snelheid waarmee de opeenvolgende video-beeldjes (frames) op het scherm verschijnen. Een gewone PAL-televisie heeft een verticale scanfrequentie van 25 Hz (50 halve beelden is 25 hele beelden per seconde). Dat maakt het beeld van dichtbij onrustig en minder geschikt voor het vertonen van teksten en andere stilstaande beelden. Pas bij verticale scanfrequenties van 60 Hz en meer wordt het beeld voor het menselijk oog aangenaam rustig.

Normaliter liggen de verticale scanfrequenties tussen de 45 tot 80 KHz. Dat is ruim voldoende. In sommige gevallen is meer nodig en levert de fabrikant een speciale video-adaptor met monitor.

• De **horizontale scanfrequentie** definieert de scanrate (schrijfsnelheid) waarmee de elektronenbundel de beeldlijnen optekent. Daar het hier om afzonderlijke beeldlijnen gaat liggen deze frequenties in het KiloHertz (KHz met K=1000)-bereik. Hoe hoger de scanfrequentie des te beter wordt de beeldkwaliteit en de hardware duurder.

De verschillende videostandaards hebben elk hun eigen verticale scanfrequentie waarmee de gebruikte monitor compatibel dient te zijn. Een CGA-kaart levert een verticale scanfrequentie van 15.6 KHz aan, Pal zit op 15,7 KHz, de EGA-kaart heeft een frequentie van 21.8 KHz en de standaard VGA-kaart scoort in deze 31.5 KHz. Voor Enhanced en Super-VGA komt u op monitoren met 35 KHz en meer. Hier ligt ook het belangrijke verschil tussen de **CGA-, EGA- en Multiscan-monitoren**. Een CGA-monitor kan alleen met een verticale scanfrequentie van 15.6 KHz uit de voeten, een EGA monitor werkt zowel met 15.6 KHz als 21.8 KHz en de moderne Multiscanmonitor kan alle genoemde verticale scanfrequenties de baas.



Voor Enhanced (800 x 600 pixels) en Super-VGA-kaarten met 1024 x 768 pixels of meer is beslist een monitor met een voldoende hoge horizontale scanfrequentie nodig. Om deze hoge beeldscherpte daadwerkelijk op het Amiga-scherm te krijgen heeft u een horizontale scanrate van meer dan 35 KHz en een verticale frequentie van 60 Hz nodig. Monitoren met deze specificaties zijn behoorlijk duur! In zo'n geval een gewone 640 x 480 pixels VGA-monitor kopen is weer zonde van de capaciteiten van de Enhanced- of Super-VGA-kaart. FlickerFixer-kaart of Amiga 3000.

De keuze van het type monitor vindt meestal plaats op grond van de video-adaptor in de Amiga met de daarbij behorende emulaties van de verschillende videostandaards en de software. Amiga-gebruikers die ook onder MS-DOS willen draaien dienen rekening te houden met PC-monitoremulaties. **Moderne MS-DOS-software** draait zowel onder de Hercules, EGA- als VGA-standaard. Voor oude releases is soms nog een CGA-emulatie nodig. Dat is dan weer wel zonde van de kwaliteit van het VGA-systeem.

De bandbreedte

Een elektronenkanon (N.B. diverse kleurenmonitoren gebruiken meerdere, meestal 3 in de kleuren Rood-Groen-Blauw, elektronenkanonnen) wordt voortdurend aan en uitgeschakeld. De **bandbreedte** geeft aan hoe vaak een elektronenkanon tijdens het scannen aan en uitgeschakeld kan worden. Daarmee definieert de bandbreedte hoeveel puntjes (pixels) de monitor per tijdseenheid op één beeldlijn kan zetten. Hoe hoger de bandbreedte in MegaHertz (MHz) des te hoger is het oplossend vermogen van de monitor. Elke videostandaard kent een **minimum bandbreedte**, maar de meeste met de desbetreffende

standaard compatibele monitoren scoren daar ver boven. Normaal zijn bandbreedtes van 35-70 MHz. Voor de VGA-standaard is minimaal 34 MHz nodig. Koop dus geen Multiscanmodel dat onder deze grens scoort.

Voor hen die met VHS werken de volgende gegevens: Gewoon VHS heeft een bandbreedte van 3,1 MHz nodig, S-VHS/Hi8 5,1 MHz tot 8 MHz voor taperegistraties en S-VHS naar RGB-monitor 4 MHz. Denk er aan dat de PAL-kwaliteit altijd achterblijft bij de beeldkwaliteit van Amiga-VGA!

De resolutie

Het **oplossend vermogen** van een monitor is reeds een aantal malen ter sprake gekomen. Daarbij is gebleken dat het om een zeer gecompliceerde eenheid met vele beïnvloedende factoren gaat. De verticale en horizontale scanfrequentie, de bandbreedte maar ook de kwaliteit van de schermcoating annex fosforlaag spelen in deze een belangrijke rol.

Technisch gezien is de **resolutie** het maximum aantal pixels dat op de horizontale en verticale beeldlijnen gaat. **Pixels** zijn beeldpuntjes die op de kruispunten van het uit de verticale en horizontale beeldlijnen opgebouwde elektronische rasterwerk (beeldmatrix) zitten. Dit elektronische rasterwerk moet op de beeldbuis zichtbaar gemaakt worden, daartoe heeft de monitorfabrikant de beeldbuis bekleed met fosfordeeltjes die opglowen



zodra zij door de electronenbundel worden geraakt. Bij **monochroom** gaat het om enkelvoudige fosfordeeltjes. Bij **kleur** is er sprake van zogenaamde **RGB-triaden**, gecombineerde Rood-Groen-Blauw fosforpartikeltjes. De fosfordeeltjes worden meestal uniform over de beeldbuis verdeeld. Een enkele fabrikant brengt daar een eigen alternerend lijnenpatroon al of niet in combinatie met een speciaal filterraster voor de electronenbundels op aan.

De afstand tussen de fosfordeeltjes, de **tridot pitch**, bepaalt hoeveel pixels er op het scherm kunnen. Hoe kleiner die afstand des te fijner zijn de fosforpartikeltjes en des te groter wordt de schermresolutie met de daaraan gekoppelde beeldscherpte. In het algemeen ligt de tridot pitch tussen 0.25 en 0.3 millimeter. Boven de 0.4 neemt de kwaliteit duidelijk af. Een aantal HIRES zwartwit monitoren scoort in deze door het gebruik van enkelvoudige kleinere fosfordeeltjes (0.18 mm) of nog beter.

Worden de fosfordeeltjes door electronen getroffen dan gaan zij opgloeien. De getroffen plaatsen geven dus licht (pixel on), de rest van het scherm blijft donker (pixel off). Natuurlijk gaan de partikeltjes niet direct weer uit, maar blijven zij even nagloeien. Men spreekt van een **nagloeitijd**. Die nagloeitijd mag niet te kort, maar ook weer niet te lang zijn. Te kort is vervelend voor het oog. Te lang maakt het gebruik van lichtpenningen en andere schermcontrols moeilijk of onmogelijk. De meeste fabrikanten gaan uit van een gemiddelde nagloeitijd. Deze nagloeitijd is doorgaans niet geschikt voor de **Amiga-interlacemodus** zonder FlickerFixer.

Een andere absolute maat voor de resolutie is de **spot size**. De spot size is de minimum deeltjes grootte die door de electronenbundel verlicht kan worden. Deze maat geeft een goede indruk



van de gedetailleerdheid van het getoonde videobeeld. In de praktijk ligt de spotsize tussen de 0.4 en 0.8 millimeter. Dat betekent een lagere resolutie dan de tridot pitch! Helaas blijkt de spotsize geen konstante grootte. Met name bij goedkopere monitoren neemt de spotsize naar de randen van het beeldscherm toe en daarmee de resolutie af. Ook is de spotsize gevoelig voor de helderheid van het beeld. Hoe helderder het beeld des te meer de randen van de spotsize beginnen uit te dijen.

Naast de technische resolutie krijgt de monitorgebruiker ook met een **subjectieve resolutie** te maken. Bij de schermgrootte kwam al ter sprake dat grotere monitoren minder scherp ogen dan kleinere beeldbuisen. Dat komt omdat er een zelfde aantal horizontale en verticale beeldlijnen op een groter schermoppervlak gaan. De onderlinge afstand tussen lijnen en pixels neemt dienovereenkomstig toe en het aantal puntjes per oppervlakte eenheid neemt af. Het oog beoordeelt dit als "minder scherp" en heeft relatief gezien ook gelijk. Gelukkig neemt de scherpte van grote buizen dankzij sterk verbe-

terde beeldbuisstechnieken snel toe.

Behalve de schermgrootte spelen ook andere factoren zoals het contrast, kleurverzadiging en de antireflectiecoating mee. Kontrastrijke, niet spiegelende monitoren met sterke verzadigde kleuren ogen scherper dan schermen met dezelfde pixelresolutie die wel spiegelen, minder kontrastrijk en kleurverzadigd zijn.

Veel gebruikte technieken om de effecten van het omringende licht op de beeldkwaliteit te elimineren zijn:

- ° Het **etsen** van de beeldbuis. De etstechniek verzacht en verspreidt het opvallend licht. Kenmerken voor deze beeldschermen zijn de reflecties met zachte randen. Een nadeel van etsen is de lichte vervorming van de pixels.

- ° Het **tinten** (meestal een donkere kleur laag aanbrengen) van het glas. Hierbij treedt geen vervorming op. Wel kan de kleurechtheid onder de tintkleur lijden.

Gepoleiste beeldschermen geven geen pixelvervorming, maar wel veel reflecties.

Time variance

De afregeling van de electronenbundel is in de tijd gezien niet konstant. Er treedt altijd enig "gewiebel" (fluctuaties) op. Een maat voor de fluctuaties in de positie van de electronenbundel per tijdseenheid is de zogenaamde **time variance**. Er kunnen drie verschillende time variance-typen onderscheiden worden:

- ° De **jitter**, een hoogfrequente "wiebel" gemeten op intervallen van 0.5 seconden of kleiner.

- ° De **swim**, een soort "zwembeweging" bij gemiddelde frequenties gemeten over intervallen van 10 seconden.

- ° De **drift**, een laag frequentie fluctuatie die per minuut gemeten wordt.

Hoe minder time variance-fluctuaties optreden des te rustiger en minder vermoeiend de monitor werkt. In de praktijk is er altijd wel sprake van licht wiebelende of in elkaar overgaande lijnen. De ene monitor doet het daarin beter dan de andere.

De voeding

Goedkope monitoren lijden wel eens onder een te krap bemeten voeding en slechte regulatie, de **voltage regulation**. De daaruit voortvloeiende vervormings- en bibber-effecten hangen sterk van het gebruik af. Normaliter mogen schermlijnen of displayde vormen niet lijden onder hetgeen elders op het scherm plaatsvindt. Toch blijkt dat bij het vertonen van grote heldere vlakken (vraagt veel energie) de schermranden gaan vervormen en lijnen zich verplaatsen. Hoe beter de interne spanningsregulatie van de monitor des te minder invloed hebben grote heldere of juist zwarte vlakken op de rest van het schermbeeld.

Sommige fabrikanten vangen hun beperkte voedingsmogelijkheid en voltregulatie slim op door gewoon de grens voor de maximale helderheid te verlagen. Dan krijg je ge-



woon geen problemen. Echt puur wit is dan uiteraard niet mogelijk. Het omgekeerde geldt echter ook. Monitoren met een hoge heldersheidslimiet doen het bij een lagere schermhelderheid aanzienlijk beter dan modellen met een kleinere limiet.

Schermafwijkingen door onvoldoende voltage regulation worden opgegeven in procenten. Zeer goed is minder dan 1%. Slecht meer dan 10%.

Misconvergence

De elektronenbundel en fosfordeeltjes moeten goed op elkaar worden afgeregeld. Dit geldt zowel voor de horizontale als digitale lijnen. Bij een slechte afregeling ziet u allerlei lelijke randjes of kleurweempjes rond de teksttekens. Ook komt een na-ijleffect voor. Een maat voor deze afregeling is de **misconvergence**.

De misconvergence, mate van slechte afstemming, neemt toe van het beeldcentrum naar de beeldranden. Goede monitoren laten daarbij slechts een geringe toename zien. Bij goedkope modellen kan de misconvergence naar de randen flink oplopen.

Analoog/digitaal

Het videosignaal kan digitaal en analoog aan de monitor worden aangeboden. Aan beide signaaltypen kleven voor en nadelen en het hangt sterk van het toekomstige gebruik af welk type monitor

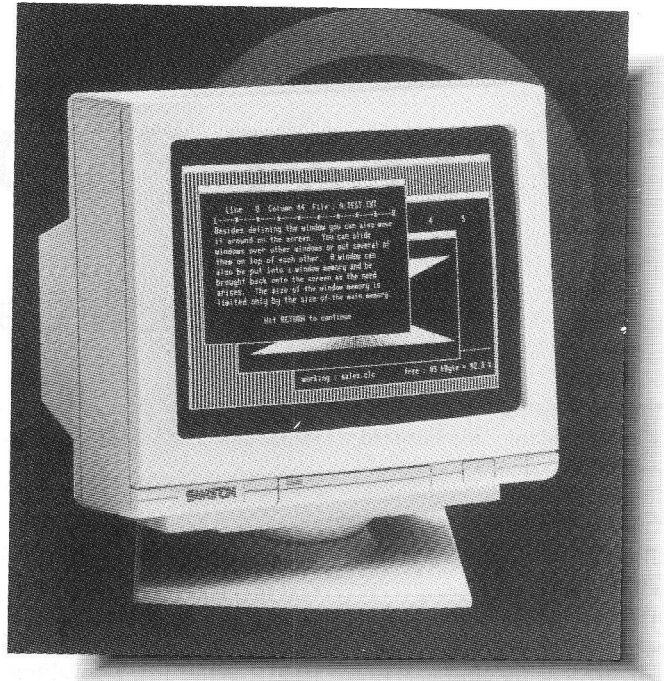
met bijbehorende videoadapter uiteindelijk de voorkeur verdient.

Een **digitaal signaal** maakt puntsgewijze aansturing (elke pixel in de beeldmatrix), een hoge resolutie en een veelzijdige beeldbewerking c.q. correctie mogelijk. De monitor spreekt dezelfde digitale taal als de computer. Echte geheel digitale monitoren zijn zeldzaam, omdat de aansturing van de elektronenkanonnen bij de meeste modellen nog wel analoog verloopt. De meest bekende vorm van de digitale aansturing is de **TTL**, Transistor-Transistor-Logic. Het principe waarmee vrijwel alle monochrome monitoren werken. Een belangrijk bezwaar van het digitale signaal is dat de kleurmogelijkheden sterk beperkt worden door de eigenschappen van de videokaart. Voor het mixen van RGB-kleuren zijn vier te coderen grootheden -de signalen Rood, Groen, Blauw en de intensiteit- nodig. Dat beperkt de mogelijkheden bij een CGA-adapter tot 16 kleuren en bij EGA in het eenvoudige geval tot 64 en bij een aantal EGA-superkaarten tot 256 kleuren. Bij de Amiga worden allerlei truuks zoals HAM gebruikt om deze kleurbependingen te omzeilen.

Een **analoge video-adapter** is van huis uit ook digitaal. De kaart procest de beeldinformatie en kleuren digitaal maar stuurt via een on-board D/A-converter een analoog RGB-signaal uit. Het grote voordeel is dat kleurmenging nu een variabele signaal is in plaats van een reeks digitale codes. Daarmee zijn nu in de praktijk bij de VGA-adapter 256.000 (de theoretische grens ligt bij 16.000.000) kleuren mogelijk.

Interlacing

Een goedkope oplossing voor HIREs monitoren is de zogenaamde **interlace-techniek**. Bij interlacing worden de verticale beeldlijnen niet in



één keer maar in twee opeenvolgende keren ververs. Het verversen gaat alternerend. Bijvoorbeeld eerst de oneven en dan de even lijnen.

De kosten besparing vloeit voort uit een lagere verticale scanfrequentie (25 in plaats van 50 Hz) die technisch gezien goedkoper is. Behalve de lagere prijs wordt de consument echter ook getrokken op een hogere beeldflickering. Een en ander valt wel op te vangen met truukjes zoals een langere nagloei-tijd van het fosfor, maar het blijft natuurlijk allemaal een surrogaatoplossing.

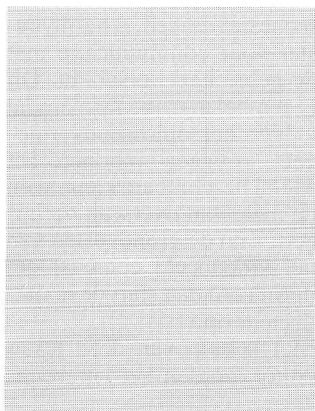
Uitvalpercentage
Monitoren hebben niet het eeuwige leven. Sommigen geven al na het uit de doos halen de geest. Fabrikanten beweren dat zij in de garantietermijn minder dan 1% uitval hebben. Ons inziens kan dat met name bij de goedkopere typen wel eens tegenvallen. Niettemin zijn de moderne VGA- en multisyncmonitoren van een goede kwaliteit. Een ander punt vormt het warmlopen en inbranden. Sommige monitoren lopen onnodig warm en gaan zelfs stinken. In de afgelopen hete zomer heeft uw testredacteur daar de nodige ervaring mee

gehad. Zoiets mag eigenlijk niet, want die stank is afkomstig van giftige brandwerende materialen in het monitorhuis.

Inbranden treedt ook sneller op dan ons lief is. Bij veel grafisch werk en een hoge beeldhelderheid is inbranden binnen drie maanden geen uitzondering. Ook komt het voor dat fosfordeeltjes gewoon uit de beeldlijnen duikeken. Koop daarom monitoren van een goed merk en met minimaal één jaar garantie.

Bij de Amiga heeft u de keuze uit vele monitortypen. De keuze wordt in de eerste plaats bepaald door de videooutput en beoogde toepassing. Daarna komen de technische specificaties. Een goede monitor garandeert meer werkplezier en betere resultaten. Dat is die extra investering meer dan waard.

U.S.



TV*TEXT PRO en TV*SHOW

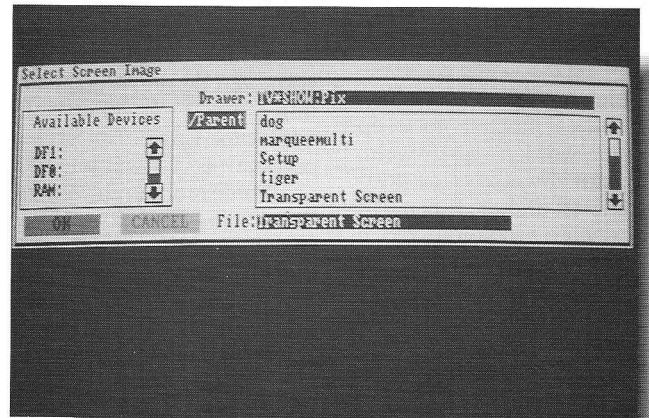
Hogeschool der videotitelatuur

De Amiga is reeds jarenlang ingeburgerd als professionele af-titelaar van videoproducties. Keurige gekleurde fonts in alle soorten en maten, fraaie schaduw- en stoboscopische effecten, kaders, cirkels en perfecte achtergronden kunnen zo op de monitor gecomponeerd worden. Nog even de genlock of PAL-encoder tussen Amiga en videorecorder aansluiten en de titels staan op video. Nu zijn er natuurlijk titelpakketten en titelpakketten. Zuma heeft haar TV*TEXT de extensie Professional aangemeten en dat is niet voor niets. Het titelpakket presteert aanmerkelijk beter dan tal van eenvoudiger aftitelaars. Er zijn echter geen animatie-effecten en wipes aan boord. Daarvoor heeft u het optionele pakket TV*SHOW nodig. Tezamen bieden deze Zuma-videopakketten als wat de

dtv-er aan titels en bijbehorende animaties nodig heeft.

TV*TEXT Professional is de pro-uitvoering van het oudere pakket TV*TEXT. Ook TV*TEXT was al tot heel wat in staat. TV*TEXT Professional doet daar, tegen een stevige meerprijs, nog een paar echte TV-kwaliteit schepjes bovenop. Een van die extra schepjes zijn de Zuma fontdiskettes 1 t/m 3 die al een deel van de prijs van circa f 350,- rechtvaardigen.

TV*SHOW is de animatiecollega van TV*TEXT. Dit pakket kan zowel voor het animeren van videotitels als het creëren van complete tekenfilms worden ingezet. Zeer interessant zijn de zogenaamde transition effects waarmee diverse schermovergangen en wipe-effecten meegemaakt kunnen worden. Voor TV*SHOW betaalt u in de winkel circa f 300,-.



Titels

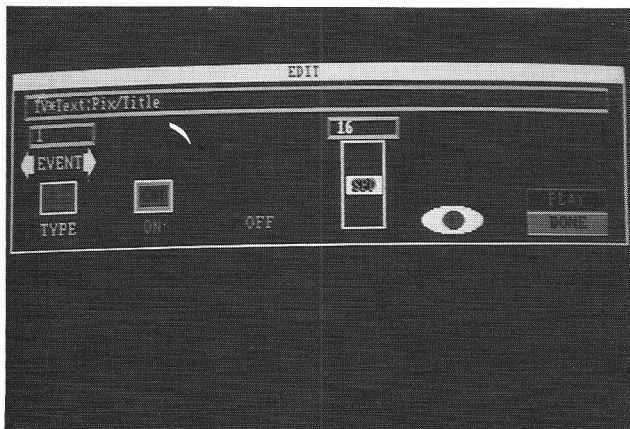
De met TV*TEXT te componeren zijn van een zeer hoog grafisch gehalte. Dankzij de grote keuze aan fonts (lettertypen en groottes), kleuren, effecten en achtergronden kunt u thuis achter de Amiga menige professionele TV- en videoproducent naar de kroon steken. Het ziet er allemaal prima uit en de enige grenzen worden in feite door het eigen artistieke inzicht gesteld. Behalve voor videotitels kunt u TV*TEXT Pro natuurlijk ook voor desktoppresentaties (PC-slideshows) en het maken van titeldia's gebruiken.

TV*TEXT Pro is net als zijn niet-professionele titelcollega een **rendering-programma**. Bij rendering geeft de gebruiker een beschrijving van hetgeen de computer op het scherm dient te zetten. De Amiga tekent deze titels en/of objecten op gebruikmakend van reeds voorgedefinieerde vormen (bijvoorbeeld standaard-fonts) of wiskundige formules. Het renderpro-

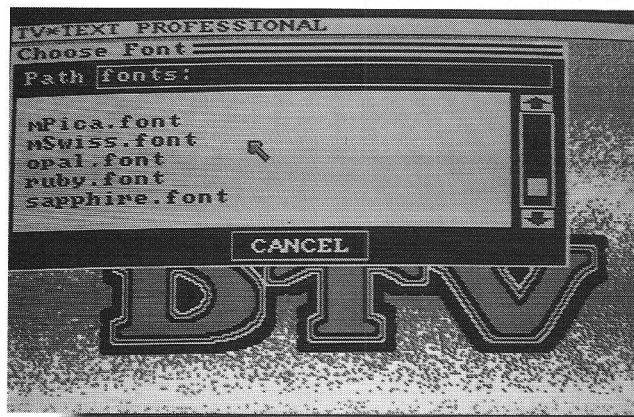
ces kost enige tijd en gaat wat minder handig dan het zelf met de muis figuren of letters tekenen. De verkregen resultaten zijn echter meer dan de moeite van het wachten en schuiven met objecten waard. Rendering-software behoort tot de beste aftitel- en animatiepakketten die er voor de Commodore Amiga verkrijgbaar zijn.

De aanmaakprocedure voor titels is bij TV*TEXT Pro relatief simpel. Start het programma met behulp van de ikoon (grafisch TV*TEXT-symbool) op en de **screen-format-requester** (N.B.: een requester is een schermvenster voor communicatie met de gebruiker) verschijnt. Dit is een belangrijke fase omdat u hier het oplossendvermogen, wel/geen interlace, het aantal beschikbare kleuren en de mate van overscan definieert. Zijn de grondvesten van het videobeeld eenmaal gedefinieerd dan start de textredacteur op.

Met behulp van deze teksteditor worden het lettertype,



de tekststijl (vet, schuin e.d.), de kleur-, schaduw- en stroboscopische effecten plus de wijze van uitlijnen gekozen. Nu is het de bedoeling dat de gebruiker een tekstregel in typt in het venster bovenaan het scherm. Daar kan eventueel nog gekorrigeerd of de laatst gebruikte tekstregel teruggehaald worden. Is de tekstregel naar de zin dan klikt de gebruiker met de muis de renderknop aan en TV*TEXT slaat aan het rekenen. Op het scherm verschijnen diverse meldingen van



hoever, in percentages, het met het optekenen van de tekst en bijbehorende effecten staat. Is de titel gereed dan verschijnt de tekstregel links bovenaan in beeld.

Vervolgens is het de bedoeling dat u de tekst gaat positioneren, d.w.z. schuiven met de regel tot deze op de gewenste plaats staat. op het scherm verschijnt naast de tekst ook een drukknop of een vuilnisbakje plus een horizontale en verticale instellijn. Die lijnen functioneren als een soort draadkruis. Met de muis (linkerknop ingedrukt) trekt u de titel naar de gewenste schermlocatie. Is de drukknop-aanwijzer (muiscursor) zichtbaar dan is een druk op de rechter muis-knop voldoende om de tekst vast te plakken. Stond er een vuilnisbak-cursor op het scherm dan wordt de titel uitgevlakt.

Desgewenst zijn er meerdere titels op één tekstscherm te

plakken. Daarvoor kunt u hetzelfde of verschillende fonts in diverse kleuren gebruiken. Eenmaal aangemaakte titels worden voor later gebruik naar de diskette of harde schijf weggeschreven.

Achtergronden

Standaard is een zwarte achtergrond met gekleurde of witte titelletters. Wie zoiets maar een doods achtergrondje vindt kan bij TV*TEXT Pro uit tal van creatieve mogelijkheden kiezen. Bijvoorbeeld:

° Het wijzigen van de achtergrondkleur via het **kleurenpalet**

tergrondkleur via het **kleurenpalet**

° **Color flooding**; Daarvoor zijn meerdere instelbare kleurpatronen aanwezig. De gebruiker kan de kleuren Rood, Groen, Blauw, de verzadiging (S), helderheid (V) en tint (H) instellen

° Onder **dithering** verstaat men het creëren van verlopende patronen van kleurpuntjes. Er zijn diverse verloopmogelijkheden. Bijvoorbeeld van boven naar beneden of diagonaal

° Via de **Tile**-optie kan een bepaald voorgedefinieerd patroon over het gehele scherm heen "gestempeld" worden. Neem hiervoor niet te kleine objecten, want dan kan het volstem-pelen weleens een langdurige zaak worden

° Met behulp van de **Wallpaper**-optie is het mogelijk om het scherm met een

uitgesneden tekst of object te behangen. U plakt als het ware vellen "behangselpapier" op het scherm

° Met behulp van **Grid** kan de titelmaker snel rasters of een ruitjespapier-achtergrond samenstellen

Andere beschikbare achtergrondmogelijkheden zijn nog reverse, stencil (houdt de opgegeven objecten in de voorgrond) en lock background (beschermde achtergrond tegen veranderingen). En natuurlijk kan de gebruiker ook bestaande grafische achtergronden, bijvoorbeeld die uit een tekenpakket, inladen.

Outlines en objects

Via de preferenties kan de titelmaker vier belangrijke titelparameters instellen:

° Het **Face** of te wel het lichaam van de letter

° De **outline**, het randje om de letter heen

° De **schaduw**, de diepte die de letter krijgt (tevens 3D-effect)

° De **Light Direction** stelt de lichtinval in

Al deze opties zijn via het Prefs-menu bereikbaar.

De **outline-requester** is één van de uitgebreidste onder de voor de Amiga beschikbare videosoftware. Het is hier onmogelijk om alle instellingen en mogelijkheden te bespreken. Via de keuze "Outline" uit de Create/Edit Prefs-requester komt u bij een venster waar instelbaar zijn de diepte, de kleur, de uitvulling van de letter zelf (8 fill-mo-

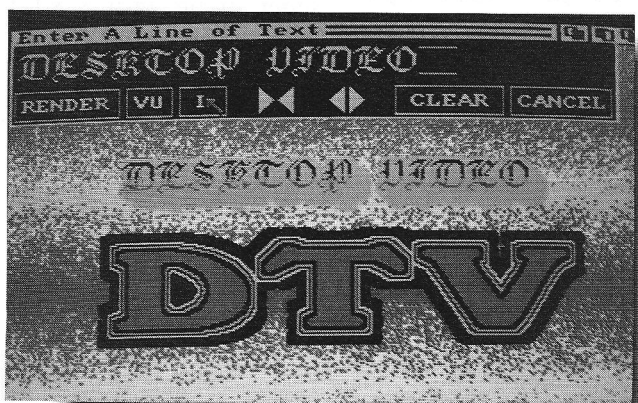
gelijkheden) en de vorm de outline. Een gewone outline omringt de gehele letter. Een Edge-outline geeft alleen een outline in de richting van de ingestelde lichtval. De XTRD (=extruded) outline geeft een outline op de plaatsen die het verst van de lichtbron verwijderd zijn.

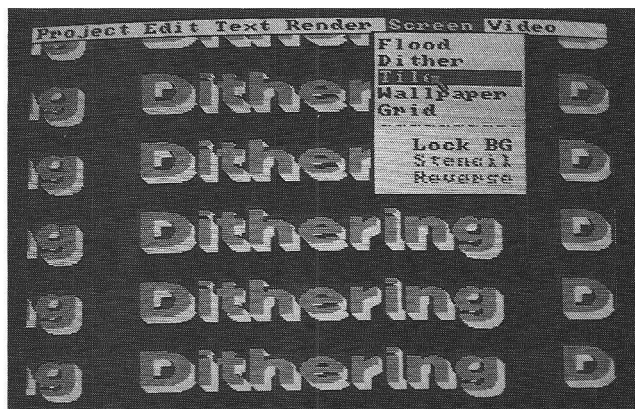
Het **objectmenu** biedt tal van creatieve mogelijkheden. In de eerste plaats kunt u er een aantal standaardfiguren (shapes) mee tekenen. De rechtehoek, cirkel en ellips kunnen naar keuze vervormd, ingekleurd, uitgesneden en aan elkaar geplakt worden. Verder is het mogelijk om bestaande objecten en brushes in TV*TEXT Pro te halen.

Verder kan de gebruiker via het EDIT-menu titel/objectstukken uitknippen, elders in plakken en verwijderen. Het pakket is uiteraard geheel genlock-compatibel.

Animatie en Presentatie

De nieuwe TV*SHOW 2.01-versie plaatst zich in de range van geavanceerde dtv-slide-show- en animatiepakketten. TV*SHOW is zowel geschikt voor het laten bewegen van videotitels, als het creëren van Special-Effects, PC-diashows en interactieve lesprogramma's. Kortom een compleet presentatiepakket. De bezitter van TV*TEXT Pro zal TV*SHOW er graag bij willen omdat de eerstgenoemde ZUMA-aftitelaar niet over transition-mogelijkheden en Special-Effects beschikt. TV*SHOW biedt deze mogelijkheden in ruime





mate. Daarnaast is TV*SHOW ook geheel zelfstandig bruikbaar. Het pakket kan met andere pakketten aangemaakte tekeningen, titels, verfkwasen (brushes), gedigitaliseerde geluiden en zelfs complete animaties importeren en presenteren.

Scripts

Net als bij Deluxe Video en echte videoproducties maakt TV*SHOW gebruik van een soort storyboard of script. In dit script worden alle gebeurtenissen, effecten en het begeleidend geluid in de tijd aangegeven. Zit het script eenmaal in elkaar dan kan de computer de gehele show automatisch afspelen, de playback. De scripts kunnen per frame (single), een vooraf opgegeven bereik (range) en in hun geheel worden afgespeeld.

Eerst dient de showmaker het script een naam te geven. Daarna maakt hij of zij een keuze uit één van de zes events: het scherm (achtergrond), het object (tekst, logo, vorm/shape of ander grafisch object), de animatie, de color cycle, de lus en de interactieve toets. Per event kunnen via het EDIT-venster de volgende scriptparameters worden aangegeven:

- ° Het **Type** voor het definiëren van één 42 standaard presentatie- en/of effectmogelijkheden (transitions). Er is keuze uit een groot aantal wipe-effecten, scrolls (rollen over het scherm), fades, pushes, reveals e.d.

- ° De **Cut** (beeld stil, volgend effect)
- ° De **Snelheid van de transition**
- ° De **Dwell** staantijd van de event in seconden
- ° **Sound**, het begeleidend geluid
- ° **Off**, object aan het einde van de transition van het scherm. De achtergrond blijft staan
- ° **Cycle** voor kleurveranderings effecten

In feite is de gebruiker van TX*SHOW bezig met een soort presentatie-legodoos. De gehele show wordt steen voor steen opgebouwd en aan elkaar gekoppeld. Via de editmodus is het mogelijk om events tussen te voegen of te verwijderen, uitsnijden & verplaatsen (cut & paste), objecten te positioneren en de geluidseffecten (ook de computerspraak) aan te passen.

Loops, keys en merge

Een **loop-event** is een automatische herhaling van het script. Bij presentaties is het vaak van belang dat een bepaalde beeldsequentie voor de langswandelende bezoekers voortdurend herhaald wordt. Daarvoor biedt TV*SHOW de loop-optie om een niet-eindigende presentatielus te creëren.

Een **Key-event** is een soort verwijzer naar een bepaald deel van het script. Degeen die naar de show kijkt kan dan behulp van een key op het toetsenbord een bepaalde selectie maken. Bijvoorbeeld de keuze uit een aantal ant-

woorden bij een cursusprogramma. Keuze van de "goede antwoord-toets" doet het script naar het volgende item gaan. Een "verkeerde toets" geeft een foutmelding of een explicatie waarom het gegeven antwoord niet goed was. Met **merge** kunt u meerdere scripts tot één nieuw groter script verenigen. Dat kan via aan elkaar plakken (append) en invoegen (insert).

Animaties

Voor het invoegen van animaties in een script moet eerst een ANIM-event geactiveerd worden. TV*SHOW kan alleen werken met looping animaties volgens de gangbare Amiga IFF Anim 5-standaard. Anders gebeurt er niets. Om misverstanden te voorkomen: **TV*SHOW gebruikt hier door andere pakketten zoals Deluxe Paint III gemaakte animaties** en animeert dus niet zelf! Alleen de lus- en vertragingstijd voor het laatste beeld worden door TV*SHOW geregeld. Met de Loops-regelaar wordt het aantal herhalingen van de animatie(s) ingesteld. Met Delay kan het laatste beeld van de animatie een bepaalde tijd worden vastgehouden.

Van huisuit beschikt TV*SHOW allleen over de animaties Roll (scroll), Fly (een brush verplaatsen), Push, color cycling (neonreklame-effect) en Wipe. De functie van dit presentatiepakket beperkt zich hier verder tot het via de scripts kop-

pelen van animaties, achtergronden, objecten en geluid.

Praktijk

Zowel TV*TEXT als TV*SHOW bieden uitgebreide een professioneel ogende mogelijkheden voor de Amiga-dtv-er. Titels, Special-Effects en slideshows gaat deze combinatie prima af. Door het groot aantal mogelijkheden en de niet zo beste index van de engelstalige manuals neemt het leren van alle beschikbare opties flink wat tijd in beslag. De resultaten zijn er dan wel naar. Wie alleen goed ogende titels wil is met TV*TEXT meteen uit de brand. Alleen is het jammer dat dit titelpakket een scroll en goede tekenoptie ontbeert. TV*SHOW vormt een welkome aanvulling als SEG voor het titelpakket, maar kan ook als stand-alone voor PC-shows gebruikt worden. Een gemis vormen de echte animatiemogelijkheden zoals we die van bijvoorbeeld Deluxe Video gewend zijn.

Inlichtingen o.a.: Amigis en het Computer Collectief.

U.S.

