

NEDERLANDS POPULAIRSTE COMPUTERBLAD

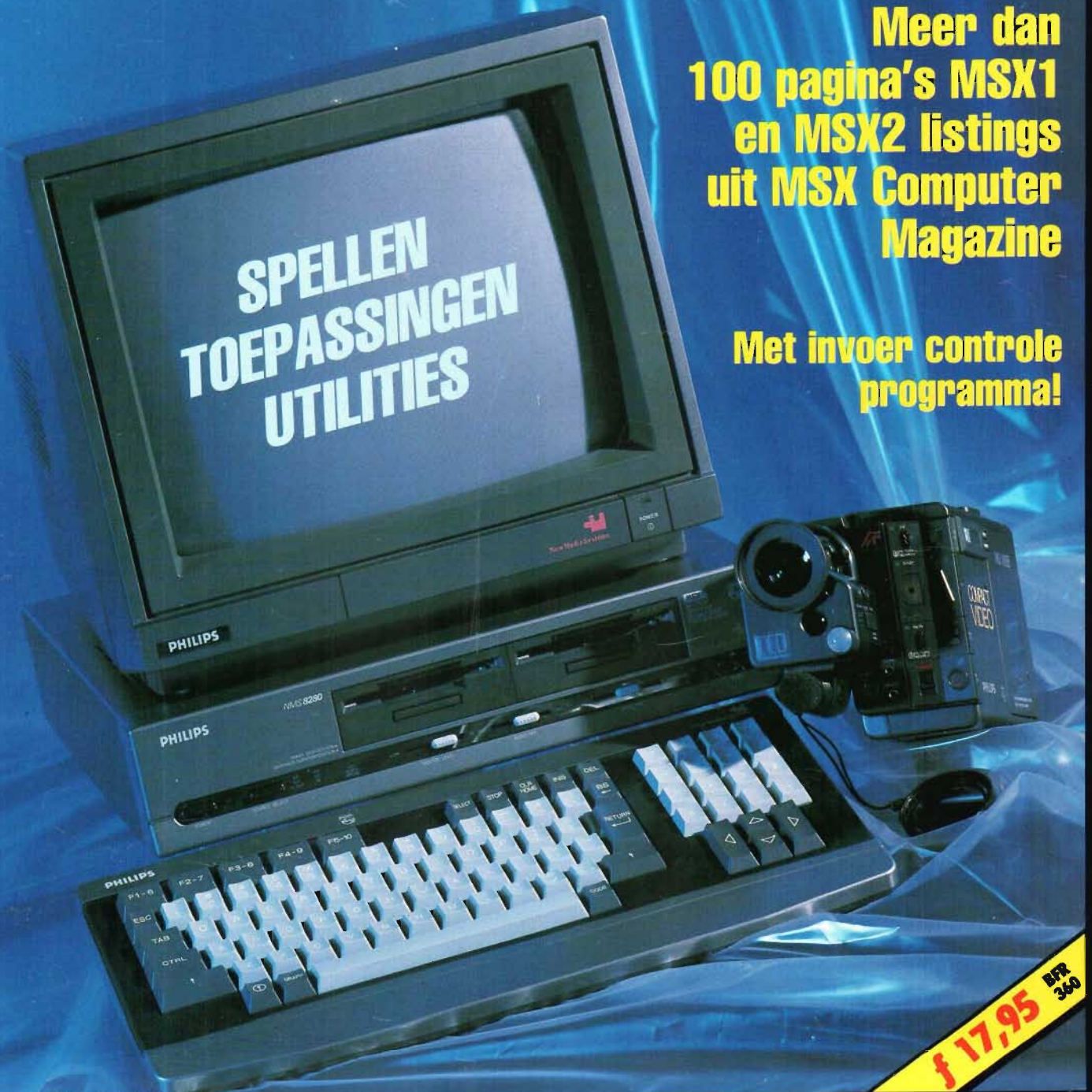
# MSX<sup>®</sup>

COMPUTER MAGAZINE

# LISTINGBOEK

Meer dan  
100 pagina's MSX1  
en MSX2 listings  
uit MSX Computer  
Magazine

Met invoer controle  
programma!



f 17,95 BFR 360



# **MSX-LISTINGBOEK NO. 2**

**een uitgave van**

**MBI** Publications bv Amsterdam

**MSX LISTINGBOEK**

is een uitgave van  
MBI Publications bv Amsterdam

**Samenstelling**

Wammes Witkop

**Uitgever**

Ronald Blankenstein

De programma's in dit boek zijn eerder verschenen in MSX Computer Magazine, het doe-blad voor MSX-gebruikers.

Dit boek is mede tot stand gekomen door de vele lezers-inzendingen.

**MSX COMPUTER MAGAZINE**

is een uitgave van  
MBI Publications bv Amsterdam

**Hoofredacteur**

Wammes Witkop

**Medewerkers**

Hans Niepoth  
Harry van Horen  
Markus The  
Hans Goddijn  
Mariëlle Mink  
Andre Knip  
Edgar Hildering  
Robbert Wethmar

**Abonnementen**

Tel. 020-657884

Abonnementen op MSX Computer Magazine kunnen elke maand ingaan.

Opzeggen abonnementen: alleen schriftelijk.

Abonnementsprijs (8 nummers) f 50,-

**Redactie**

Postbus 1392  
1000 BJ Amsterdam  
Tel.: 020-681081  
Telex: 16015 MBI NL  
Fax: 020-931263

**Programma Service**

Tel.: 020-681081, tst. 25

**Art Director**

Cock Arentsman

**Vormgeving**

Joost van Donk  
Mariëlle Mink  
Jeroen Engelberts

**Cartoons**

Jeroen Engelberts

**Distributie**

Beta Press/van Ditmar  
Burg. Krollaan 14  
5126 PT Gilze

**Toezenen materiaal**

Tenzij uitdrukkelijk anders overeengekomen heeft MSX Computer Magazine het recht om vrijelijk te beschikken over alle haar toegezonden materiaal. Terugzending van ongevraagd toegezonden materiaal zal alleen plaatsvinden als er een geadresseerde en voldoende gefrankeerde retour-enveloppe is bijgesloten.

# INHOUD

Wat is MSX .....	6	Koppie .....	62
Voorwoord .....	7	Intelligente disk-kopieerder, alleen MSX2	
Programma-Service .....	8	Teller .....	66
Invoer Controle Programma/5 .....	10	Handig hulpje als u iets moet tellen	
Onmisbaar bij het intikken		Sprite .....	68
MSXmem .....	13	Sprite-Editor van de bovenste plank	
MSX geheugenstructuur ontsluit		Joysor .....	76
HAL .....	18	Bestuur de cursor met de joystick	
Héél Apart Labyrint		Digiklk .....	79
MSXbug .....	27	Beeldscherm-klokje	
Monitor voor gevorderde programmeurs		MCM2B .....	80
Print .....	33	MSX2 database met grote capaciteit	
Snel even een uitdraaitje in kolommen		Disass .....	90
Pucky .....	36	Disassembler in machinetaal	
Pacman-kloon, maar dan een goeie!		Varln2 .....	95
Watklk .....	42	Onmisbaar hulpje voor Basic-programmeurs	
Een waterklok op uw beeldscherm?		Blast .....	97
Tstbld .....	44	Spelenderwijs leren tikken!	
Uw eigen testbeeld, net echt		Linklk .....	103
Repwek .....	46	Een speels computer-uurwerkje	
Een repeteerwekker waar muziek in zit		Jake (in the caves) .....	104
Sinterklaas .....	51	Platform-spel	
Sinterklaasgedichtjes of andere rijmelarijen		Keuken .....	111
Sculptor .....	52	De waanzinnige keuken, spel	
Een puik MSX2 tekenprogramma			
Ijsfabriek .....	56		
Boeiende economische simulatie			

# Wat is MSX?

Een van de vragen die ons het meest gesteld worden is: Wat is nou eigenlijk MSX? Natuurlijk weet u, als trouwe lezer van MSX Computer Magazine, er al alles vanaf, maar omdat er met elke nieuwe uitgave ook nieuwe lezers bijkomen zullen we deze vraag voorlopig blijven beantwoorden.

Om dat te kunnen doen moeten we eerst iets meer weten van de geschiedenis van de homecomputer. Toen rond 1977 de eerste microcomputers op de markt verschenen, was een van de problemen dat geen van de op grotere computers gebruikelijke programmeertalen op deze micro's in te zetten was. De geheugenruimte was veel te klein voor talen als Cobol, Fortran of Pascal. Er moest dus een andere oplossing gezocht worden, en die werd gevonden in de vorm van Basic (Beginners All-purpose Symbolic Instruction Code). Nu was Basic door zijn ontwerpers nooit bedoeld om als een echte werktaal te worden ingezet, Basic is oorspronkelijk gemaakt als leermiddel. Vandaar ook dat de oorspronkelijke Basic allerlei commando's miste die voor echt werk onontbeerlijk zijn. Allerlei fabrikanten van grote computers hadden deze mogelijkheden echter wel ingebouwd in hun Basic varianten, maar wel ieder op zijn eigen manier.

Basic bleek echter klein genoeg om op die eerste microcomputers te kunnen draaien, en een Amerikaans bedrijf, Microsoft, produceerde de eerste Basic vertolker, het programma dat het mogelijk maakt om in Basic te werken. Nu is men er niet met die Basic vertolker alleen, een computer heeft ook een operating system nodig. Dat is een soort van raamwerkprogramma, waarbinnen bijvoorbeeld Basic zijn werk kan doen. Een bekende firma op dit gebied is Digital, met onder meer het

toendertijd zeer populaire CP/M, wat staat voor Control Program/Microcomputers.

In de loop van de daaropvolgende jaren echter ontstond er een ware wildgroei aan Basic-dialecten en operating systemen. Iedere fabrikant bedacht zijn eigen oplossingen, die vaak zelfs voor ieder computertype van een bepaalde fabrikant verschillend waren. Ook allerlei nieuwe mogelijkheden zoals kleur en geluid werden op vele manieren geïmplementeerd. Het eind van het liedje was dat een beetje programma voor ieder type computer apart geschreven moest worden, zo groot waren de onderlinge verschillen geworden. Uiterst onhandig, en onefficient. Bovendien voor de computergebruiker een verschrikking, want voor ieder type moeten er aparte boeken, programma's en tijdschriften uitgegeven, wat de prijs alleen maar opdrijft.

Dat ging zo niet langer, er moest een wereldstandaard komen. Vooral de Japanners waren daar voorvechters van. Standaardisatie is de enige wijze om een systeem wereldwijd te kunnen verkopen. Die standaard kwam er dan ook, in de vorm van MSX. MSX, de letters staan voor MicroSoft eXtended basic, is nu door vele elektronikagiganten geaccepteerd als de nieuwe homecomputer standaard. Iedere MSX computer gebruikt hetzelfde operating system, terwijl de Basicvertolkers van de beide versies (MSX1 en MSX2) 'opwaarts compatibel' zijn. Daardoor kunnen programma's die op een Sony MSX1 geschreven zijn zonder meer op een Philips MSX2 worden gebruikt. MSX2 is namelijk een uitbreiding op MSX1, waardoor een volgens de MSX1 norm geschreven programma ook op een MSX2 kan werken. De andere kant op gaat natuurlijk niet, MSX2 kent nu een-

maal meer mogelijkheden dan MSX1.

De voordelen zijn legio, zowel voor fabrikanten als voor konsumenten. Doordat MSX zo'n grote basis heeft kunnen programma's en randapparaten in veel grotere aantallen geproduceerd worden en zullen de prijzen van deze artikelen lager liggen dan bij al die andere homecomputers. Sterker nog, juist door die grote basis is het interessant om allerlei extra's uit te brengen die bij andere computers in veel te kleine aantallen verkocht zouden worden om ze in productie te nemen.

Daarnaast komt nog dat MSX-Basic een van de beste, zo niet de beste, Basic is die er is. MSX-Basic bevat allerlei kommando's voor geluid, grafiek en kleur.

Bij andere computers is dat vaak veel onhandiger opgelost en blijven dergelijke mogelijkheden slechts voorbehouden aan full-time programmeurs. MSX2 steekt zelfs qua grafische mogelijkheden met kop en schouders uit boven wat er met andere homecomputers mogelijk is! Dan is het operating system van de MSX standaard het zogenaamde BIOS, dat nauw verwant is aan het tot voor kort zeer populaire CP/M. Dit CP/M was de hit op de zakelijke markt, tot men daar de (snellere) 16 en 32 bits processoren ging gebruiken, waar overigens de operating systems ook sterk op het MSX-BIOS lijken. Doordat BIOS en CP/M zo verwant zijn, was het relatief simpel om CP/M ook voor MSX computers te schrijven. Vooral voor MSX2, dat over een 80-kolomsscherm beschikt, is daardoor is er een ware schatkist aan goede, zakelijke programma's beschikbaar. Tekstverwerkers, databases, noem maar op.

Bovendien zijn de MSX-diskettes uitwisselbaar met het MS-DOS formaat zoals dat door de IBM (en compatibles) Personal Computer ge-

bruikt wordt. Dat houdt in dat men, met de juiste software, thuis verder kan werken aan bijvoorbeeld een stuk wat men op het werk geschreven heeft op een PC.

Afhankelijk van de versie - MSX1 of MSX2 - zijn er harde eisen vastgelegd, waaraan de machines moeten voldoen om het MSX-logo te mogen dragen. Dat geldt overigens ook voor randapparatuur, zoals printers en diskdrives. Dat garandeert voor de koper dat alles zonder problemen met elkaar kan functioneren.

Die eisen slaan onder meer op de te gebruiken chips, het toetsenbord, de tekenset, het geheugen etc.

Bij MSX zijn de maximale mogelijkheden echter niet vastgelegd. Een fabrikant mag een machine bouwen met wel 15 MSX slots, als hij dat wil. Of met een ingebouwde synthesizer, naast de (voorgeschreven) standaard audio chip. En om nog maar eens een dwarsstraat te noemen, een Video Disk interface, waarmee volstrekt nieuwe zaken onder bereik komen. MSX is een minimum standaard, maar het maximum is vrijwel onbeperkt. Toch kunnen al die verschillende machines in principe met elkaars programma's werken, als die programma's maar aan de standaard voldoen.

Toch is er kritiek mogelijk op de MSX standaard. De toegepaste Z80A processor is een 8 bits processor, die vaak al wat verouderd wordt gezien. Echter, voor een homecomputer is een 8-bits structuur meer dan afdoende. De grotere snelheid van zakelijke 16- of 32-bits CPU's is niet echt noodzakelijk.

Het MSX operating system lijkt sterk op het operating system wat er in die zakelijke machines zit; beide zijn geschreven door Microsoft. Zo sterk zelfs, dat een toekomstige 16- of 32-bits MSX zonder meer compatibel zal zijn met de huidige MSX1 en MSX2 machines.

Toekomstmuziek? Het hangt er maar vanaf hoe je het bekijkt. Wij gaan ervan uit dat die derde MSX generatie er zal komen, binnen het jaar. Waarschijnlijk zullen dat zelfs 32-bits machines worden.

Van de  
redactie



WAMMES WITKOP

# Voorwoord

Een computer zonder programma's, dat is als een kroeg zonder bier: geen lol aan te beleven. Vandaar dat juist de programma's in MSX Computer Magazine zo populair zijn bij de lezers, zo blijkt telkens weer uit de lezerspost. Maar om nu - als beginnende computeraar - al die oude nummers te moeten nabestellen om aan de listings te komen, dat is wel wat begroetelijk.

Vandaar dat we - voor de tweede keer alweer - de beste programma's uit MSX Computer Magazine in een apart boek gebundeld hebben, het listingboek dat u momenteel in handen houdt. Bij de vorige uitgave beweerde ik al dat u het beste listingboek dat er bestond voor u had; dit keer durf ik dat weer te stellen. Dit tweede MCM-listingboek is zeker net zo goed als het eerste, zo niet nog beter. We hebben veel bijgeleerd, in de afgelopen anderhalf jaar.

Bovendien zijn alle programma's in dit boek uitgebreid getest, niet alleen door onszelf, maar ook door de vele, vele lezers van MSX Computer Magazine, waar deze programma's eerder verschenen zijn. Mocht er al eens een enkel foutje doorheen geglipt zijn, het lezersforum heeft er ons sindsdien zeker op gewezen. Met die ervaringen in het achterhoofd hebben we op de redactie alle programma's die in dit boek staan nog eens met een fijne stofkam nagevlooid, zodat we durven te stellen dat de kans op fouten in de listings in dit boek nagenoeg gelijk aan nul is.

Als u, als lezer, nu ook nog de moeite neemt om als eerste programma ons Invoer Controle Programma/5 in te tikken en ook bij iedere listing te gebruiken, dan kan er eigenlijk niets fout gaan. Met dat ICP/5 kunt u bijna geen tikfouten meer maken, succes is verzekerd.

Toch bestaat er altijd een - klein - kansje dat u de zaak niet goed op gang krijgt. Dat kan aan vele zaken liggen, zo zal een voor MSX2 bestemd programma niet op een MSX1 willen werken. In dergelijke gevallen kunt u altijd terugvallen op onze telefonische hulpdienst, waarvan u het nummer elders in dit boek kunt vinden. Een unieke service bij een listingboek! Overigens, voor de goede orde, de eerste vraag die wij u zullen stellen, als u belt met een probleem, is of u het ICP/5 wel heeft gebruikt. Zo nee, dan is ons antwoord kort en simpel; namelijk dat u dat eerst maar eens moet proberen.

Eigenlijk is het jammer dat u in dit boek alleen maar listings - met de bijbehorende artikelen - aan zult treffen. Want zo krijgt u geen goed idee van wat MSX Computer Magazine allemaal nog meer te bieden heeft.

Iedere nummer opnieuw hebben we verhalen over alle aspecten van de MSX computer-hobby. We leggen allerlei programmatuur, die men u voor duur geld verkopen wil, op de pijnbank en nemen daarbij geen blad voor de mond als de testresultaten tegenvallen. Wie aan hardware-uitbreidingen denkt kan ook bij ons terecht; we testen printers, diskdrives, computers, noem maar op. Om over de vaste rubrieken maar te zwijgen, voor de spelfanaten is onze EHBO-rubriek een begrip, met de vele kaarten van spellen die we publiceren. Andere onderwerpen? Onze algoritme-cursus, voor diegenen die serieus willen leren programmeren en natuurlijk de al lang lopende cursus machinetaal. Dat, gecombineerd met bijvoorbeeld de BIOS-tabellen - geen ML-programmeur kan er buiten - maakt ieder nummer van MSX Computer Magazine tot een bron van informatie voor zowel de beginner als de gevorderde MSXër.

Maar goed, in dit boek beperken we ons tot listings. Wie daar wel meer van wil zien en nog geen abonnee is op MSX Computer Magazine heeft na aanschaf van dit listingboek echter een gelukje. In dit boek treft u een kaart aan, waarmee u zich met een speciale korting van tien gulden kunt abonneren!

Wammes Witkop

# BON

- |                                  |                                  |
|----------------------------------|----------------------------------|
| <input type="checkbox"/> MCM-C8  | <input type="checkbox"/> MCM-C14 |
| <input type="checkbox"/> MCM-C9  | <input type="checkbox"/> MCM-C15 |
| <input type="checkbox"/> MCM-C10 | <input type="checkbox"/> MCM-C16 |
| <input type="checkbox"/> MCM-C11 | <input type="checkbox"/> MCM-C17 |
| <input type="checkbox"/> MCM-C12 | <input type="checkbox"/> MCM-C18 |
| <input type="checkbox"/> MCM-C13 | <input type="checkbox"/> MCM-C19 |

(à f.15,-/ Bfr.300)

- |                                      |                                      |
|--------------------------------------|--------------------------------------|
| <input type="checkbox"/> MCM-D6/3.5  | <input type="checkbox"/> MCM-D13/3.5 |
| <input type="checkbox"/> MCM-D7/3.5  | <input type="checkbox"/> MCM-D14/3.5 |
| <input type="checkbox"/> MCM-D8/3.5  | <input type="checkbox"/> MCM-D15/3.5 |
| <input type="checkbox"/> MCM-D9/3.5  | <input type="checkbox"/> MCM-D16/3.5 |
| <input type="checkbox"/> MCM-D10/3.5 | <input type="checkbox"/> MCM-D17/3.5 |
| <input type="checkbox"/> MCM-D11/3.5 | <input type="checkbox"/> MCM-D18/3.5 |
| <input type="checkbox"/> MCM-D12/3.5 | <input type="checkbox"/> MCM-D19/3.5 |

(à f.30,-/Bfr.600)

- |                                       |                                       |
|---------------------------------------|---------------------------------------|
| <input type="checkbox"/> MCM-D6/5.25  | <input type="checkbox"/> MCM-D13/5.25 |
| <input type="checkbox"/> MCM-D7/5.25  | <input type="checkbox"/> MCM-D14/5.25 |
| <input type="checkbox"/> MCM-D8/5.25  | <input type="checkbox"/> MCM-D15/5.25 |
| <input type="checkbox"/> MCM-D9/5.25  | <input type="checkbox"/> MCM-D16/5.25 |
| <input type="checkbox"/> MCM-D10/5.25 | <input type="checkbox"/> MCM-D17/5.25 |
| <input type="checkbox"/> MCM-D11/5.25 | <input type="checkbox"/> MCM-D18/5.25 |
| <input type="checkbox"/> MCM-D12/5.25 | <input type="checkbox"/> MCM-D19/5.25 |

(à f.27,50/ Bfr.550)

Ja, ik maak gebruik van de Programma Service.  
Stuur mij de aangekruiste artikelen.

Invullen in blokletters

Naam: \_\_\_\_\_

Adres: \_\_\_\_\_

Postcode: \_\_\_\_\_ Woonplaats: \_\_\_\_\_

Ik heb een geldig betaalmiddel bijgesloten  
(bijvoorbeeld een betaalkaart van bank of giro)  
en krijg mijn bestelling zo snel mogelijk  
toegestuurd.

Opsturen naar: AKTU Publications bv,  
Postbus 1392, 1000 BJ Amsterdam

Alle prijzen zijn inclusief verzendkosten.

Wilt u de bon niet uitscheuren?

Maak een fotokopie van deze pagina!



# Programma Service

**Alle programma's uit dit listingboek gebruiksklaar, met een redactionele extra! De makkelijkste en snelste manier om een eigen programma-bibliotheek op te bouwen. Dat bieden we u als extra service aan met onze Programma Service. Bespaar u de moeite van het intikken van lange listings (met alle risico's van fouten), en bestel de programma's uit dit boek die u hebben wil. In totaal zijn er 18 verschillende cassettes en diskettes beschikbaar met programma's uit MSX Computer Magazine.**

## MCM-C1 met:

de MCM-database; een schuifpuzzel; een doolhofspel; een driedimensionaal tekenvoorbeeld; een logica-spel; het spel 'verlicht de stad'; een gokspel en het beeldgrapje 'Appel'.

## MCM-C2:

SpriteEditor (hoofdprijs in de eerste Sony MCM-programmeerwedstrijd); Copy en CrtDmp (utility's voor disk en plotter/printer); twee educatieve programma's (aardrijkskunde en astronomie); een disassembler; een ufo-schietspel; een variant op Galgje; Yathzee; het muziek-programma Bronksi en nog meer.

## MCM-C3:

MSX-PEN (een tekstverwerker, hoofdprijs in de tweede Sony MCM-programmeerwedstrijd); de Basic-utility Rem Space Killer; een tekenprogramma; een 'kladblok' (extra beeldscherm); de spellen Lockin' Man, Horror en Escape en tot slot het tekstadventure Mystery Town.

## MCM-C4:

Schat Duiken, prima spel; Tapdir, orde in uw cassette's; Bach, virtueuze orgelmuziek; Tips85, helpt u met uw aangifte-bijlet; Letter, de MSX-karakter editor; Reuter, een onmogelijke driehoek; Snelli, een simpel maar snel spel; Beurs, speculeer op de effectenbeurs.

## MCM-C5 omvat:

Edit (een bestands-editor); Memmon voor nieuwsgierige aagjes; Colors voor de MSX2; 3D-Des, tekenen in 3 dimensies; Figrek (educatief); Snake3, een leuk spelletje en nog het een en ander.

## MCM-C6 omvat ondermeer:

Alien, schieten maar; Dsktyp, ontrafel uw diskette; Typles, leren typen op de MSX; Linlst en Varlst, helpen u om Basic programma's te doorgronden; Vissen, een prima spel en natuurlijk de prachtige Philips MSX2 Basic demo-programma's.

## MCM-C7 omvat:

Space, een winnaar van een spel, grotendeels in ML; Dskidx, bekijk en begrijp track 0 van uw diskette's; Digkik, een grappig computerklokje; CTRL-P, een machinetaal-screendumper in twee versies; Snabar, de enige echte MSX-snackbar, educatief spel; de diverse KORT & KRACHTIG programmaatjes; de listingkjes uit onze ML-kursus en, als extra, het uitstekende KUUB'ERT spel!

## MCM-C8 omvat:

Supdir, een prachtig hulpprogramma voor diskebruikers; MSXPRT, een machinetaal-programma dat van elke printer een MSX-printer maakt; Varln2, de ML-versie van de Basic-hulp-programma's Varlst en Linlst samen; Topografie, vlieg met een heli over Nederland, prijswinnaar; Tellen en Tafels, educatieve programma's; Trein, reis per trein door Nederland; en - als extra - het gedigitaliseerde portret van uw hoofdredakteur (alleen op disk en slechts voor MSX2!)

## MCM-C9 omvat:

Drum, een fraaie MSX drum-machine; Reflst, een handig hulpje voor programmeurs; Break, een dijk van een doolhofspel, winnaar!; Linklk, een fraaie klok; Repwek, een MSX-repeteer-wekker; Watklk, een computer-waterklok; maar liefst 7 MSX2 Kort & Krachtigjes en de listingkjes van de Z80-cursus.

## MCM-C10 omvat:

Joysor, een handige ML-utility; Sprite, uitstekende sprite-editor; Keuken, snel spel; Strklis, een Sinterklaas-surprise; Sciptr, een fraaie MSX2 scherm-tekenaar.

## MCM-C11 omvat:

MSXMEM, het antwoord op al uw geheugen-vragen; Teller, een handig hulpje voor al uw telwerk; Pucky, een dijk van een Pacman-spel; Tstblid, nu ook een testbeeld zonder zender; Begadr, zoek de ML-adressen op disk; Kerst, een fraaie MSX kerstkaart en de listingkjes van de Z80-cursus.

## MCM-C12 met:

Jake in the Caves, een uitstekend platform-spel; Print, afdrukken in kolommen; Salber, reken uw salaris na en, als extra, alleen op cassette en diskette; Belast, een uitgebreid belasting-programma.

## MCM-C13 bevat:

Ijsfabriek, een leerzaam spel; Viper, bestuur een slang; de listings behorende bij de programmeer-cursus en de listingkjes uit de machinetaal-cursus.

## Op MCM-C14 vindt u:

MCMPRT, de aanvulling op MCMBASE; DRPASC, statistiek in beeld; DEMOMUIS, BLOKMUIS en TEXTMUIS, kleine voorbeeldjes van muis-programmering; DRAWMUIS, een muisgestuurd tekenprogrammaatje; FILEMUIS, een handig muisgestuurde bestands-

hulpprogramma en de programma's uit de algoritme-cursus: random-generatoren en random-testprogramma's.

## MCM-C15 biedt u:

MCMBCD, Het MSX Computer Magazine's BASICODE-3 programma; MSXBUG, een dijk van een (machinetaal) monitor, onmisbaar voor wie de MSX echt wil doorgronden. Ook voor MSX2 slotstructuren; OTHELL, een lastig denkspel met de computer als tegenstander; maar liefst bstienes KORT & KRACHTIGjes; de listings uit onze Z80-cursus en de programma's uit de algoritme-cursus. Bovendien hebben we - alleen op de diskette - wat gedigitaliseerde plaatjes gezet, beelden van de MCM-redactie aan het werk!

## MCM-C16 omvat:

HAL, het Heel Apart Labyrint, een razend lastig doolhofspel met vele schermen; de listings uit onze Z80-cursus, disk-programmeren vanuit ML dus en alweer wat gedigitaliseerde prenten voor MSX2. Die beelden staan echter alleen op de diskette, voor cassette zijn ze te lang.

## Op MCM-C17 vindt u:

MCM2B, onze database de luxe voor MSX2 computers, maakt gebruik van de Memory-Mapper; KOPPIE, een intelligente disk-kopieerder die in slechts vier keer wisselen een 720K disk kan overzetten, alleen geschikt voor MSX2; DISASS, een hele slimme disassembler, werkt samen met MSXBUG; De listings uit onze Z80-cursus en MCM's Invoer Controle Programma nummer 5.

## Op MCM-C18 staan:

BLAST, een spel dat u snel leert tikken; Schaak, een fraaie schaak-leermeester, dat echter niet zelf kan spelen; Een drietal mooie kerstidjes; De listings uit onze algoritme-cursus en MCM's Invoer Controle Programma nummer 5.

## Ook op diskette

Hebt u een disk-drive? U kunt de programma-verzamelingen ook op diskette bestellen, met de bestelcode D1/3.5 (of bijvoorbeeld D12/3.5) voor Sony, Philips etc. of Dx/5.25 (o.a. AVT). Zie de bestelbon.

Lezers in België kunnen eveneens profiteren van de Programma Service. De prijzen in Belgische Francs: cassette Bfr. 300, diskette 3.5 Bfr. 600, diskette 5.25 Bfr. 550.

## Hoe bestelt u?

1. Gireer het juiste bedrag (met vermelding van de juiste bestelcodes, zie de bon) naar postgiro-nummer 3008971 t.n.v. AKTU Publications bv, Amsterdam. Uw overschrijving zegt ons precies wat we waarheen moeten zenden.

2. Of bestel schriftelijk: gebruik de bestelbon (kruis de juiste hokjes aan), en sluit een geldig betaalmiddel bij. Opsturen naar: AKTU Publications bv, Postbus 1392, 1000 BJ Amsterdam. U krijgt uw bestelling zo snel mogelijk thuisgestuurd.

Schrijf uw naam en adres - en uw bestelling - in duidelijke blokletters.

Voor vragen over de Programma Service kunt u terecht op het telefoonnummer: 020-681081, toestel 25.

# INVOER CONTROLE PROGRAMMA 5

**Zelfs de meest zorgvuldig geproduceerde en gedrukte listings sluiten niet uit dat er toch een fout kan worden gemaakt bij het intikken. Verwisselde cijfers of verkeerde leestekens leiden in het beste geval tot een foutmelding. Erger nog is het als een programma slechts schijnbaar goed, althans zonder fouten die de computer zelf kan bespeuren, werkt. Om u te helpen dit soort problemen te voorkomen publiceert MSX Computer Magazine bij alle listings een controlegetal oftewel checksum per programmaregel. Achter iedere programmaregel staat een checksum, een waarde tussen de 0 en de 255.**

**Om deze te vergelijken met uw zelf ingetikte programma dient het bijgaande Invoer Controle Programma versie 5, kortweg ICP/5.**

## GEBRUIKSAANWIJZING ICP/5

ICP/5 berekent voor iedere ingetikte programmaregel een checksum, zodra u op de *enter* of *return* drukt. Deze checksum verschijnt dan links onder op uw beeldscherm, op de positie waar anders de bij de F1 behorende tekst -COLOR- staat. Deze waarde moet overeenkomen met het getal dat in de listing bij de betreffende regel is afgedrukt, als dit niet het geval is heeft u een foutje gemaakt bij het intikken.

In dat geval moet u de betreffende regel meteen verbeteren met behulp van de normale edit-mogelijkheden van uw MSX computer. U hoeft de regel dus niet opnieuw in te tikken, ICP/5 kijkt altijd naar de hele programmaregel zoals die op het scherm staat, niet alleen naar wat er echt ingetikt wordt. Daardoor kunt u ook al eerder ingetikte programma-regels makkelijk controleren. Gewoon de regel listen, dan de cursor weer omhoog te brengen tot deze zich ergens in de te checken programmaregel

bevindt en op *return* of *enter* drukken.

ICP/5 maakt natuurlijk onderscheid tussen hoofd- en kleine letters en dat kan soms problemen opleveren. Bij het intikken van een programma zult u meestal de Basic woorden in kleine letters intikken, maar bij het listen van een regel verschijnen ze juist wel in hoofdletters. ICP/5 gaat er van uit dat Basic woorden met hoofdletters geschreven moeten worden, net zoals ze in de listings staan. Als u dus een regel heeft ingetikt met de Basic termen in kleine letters en dan op *return* of *enter* drukt, dan zult u een verkeerde checksum te zien krijgen. Gelukkig is dit echter simpel te omzeilen, door voor u met intikken begint de Caps-lock in te drukken, waarna alle letters als hoofdletter op het scherm verschijnen. Alleen als er ergens kleine letters in een programma voorkomen moet u dan de Caps-lock even uitschakelen. Wat natuurlijk ook kan is de regel intikken, op *return* druk-

ken, dan de zojuist ingevoerde regel opnieuw listen met LIST., de cursor weer in die regel plaatsen en nogmaals op *return* drukken. De tweede keer kijkt ICP/5 naar de geliste regel en daar heeft uw MSX keurig alle Basic woorden in hoofdletters vertaald.

Voor REM-regels (die ook met het ' teken aangegeven kunnen worden) wordt de checksum op nul gesteld. Slechts als de REM of het 'tekeningje niet meteen na het regelnummer staan tellen deze regels wel mee.

ICP/5 is in feite een machinetaal-programma. De Basic-listing zet deze ML - die in de DATA staat - op de juiste plek in het geheugen, waarna het zichzelf weer uitwist. Als u na het runnen van ICP/5 een LIST-kommando geeft, dan zult u zien dat het programma schijnbaar helemaal verdwenen is. De machinetaal echter staat ergens hoog in het geheugen voor u klaar, zoals u kunt zien aan de getallen links-onder.

Die ML kunt u eventueel uitzetten door de F1 in te druk-

ken, en weer aanzetten middels het kommando:

A=USR(0)

Om de ML helemaal te verwijderen moet u de computer even resetten, of even uitschakelen.

Lange listings intikken kost vaak meer dan een zitting. Ook dat is geen enkel probleem, als u stopt moet u datgene wat u tot dan toe ingetikt heeft gewoon eventjes saven, zoals u altijd een Basic-programma wegschrijft. Alleen de Basic-tekst wordt dan bewaard, niet het ICP/5.

Om de volgende keer weer door te gaan dient u eerst ICP/5 weer te laden en te runnen, waarna u het programma waar u aan bezig was weer kunt laden. Een kind kan de was doen!

Met MSX Computer Magazine is het Invoer Controle Programma/5 het laatste programma dat u zonder hulp hoeft in te tikken.

## BELANGRIJK

Test een zojuist ingetikt programma nooit meteen uit. Save het eerst, voordat u RUN intikt. Sommige programma's zouden, als er fouten in schuilen, de computer op slot kunnen zetten. En dan is de enige mogelijkheid om zelf weer de controle over de machine te krijgen een reset, of mogelijk zelfs uit en aanzetten. In beide gevallen bent u uw programma kwijt, waarvoor u zojuist een hele tijd had gespandeerd met intikken. Ook het uitproberen van nog niet helemaal ingetikte programma's is uit den boze!

Tijdens het intikken is het eveneens verstandig om, zeker als het om langere listings gaat, zo nu en dan een kopie te saven op cassette of disk. Spanningspieken in het lichtnet kunnen er ook oorzaak van zijn dat uw computer zijn programma 'vergeet'. Of er struikelt iemand over het netsnoer, waardoor de stekker uit het stopcontact getrokken wordt. Beter tien maal onnodig saven, dan een keer te weinig.

```

10 ' MSX Computer Magazine checker/5      0
20 '                                       0
30 ' copyright MBI Publications B.V.      0
1985                                       0
40 '*****                               0
50 ' PAS TOP OF BASIC MEMORY AAN ****    0
60 CLEAR 200,(PEEK(&HFC4A)+256*PEEK(&
HFC4B))-207                                42
70 B=PEEK(&HFC4A)+256*PEEK(&HFC4B): D
EFUSR0=B+77                                17
80 ' STEL SCHERM IN *****              0
90 SCREEN 0: WIDTH 37: COLOR 15,4,4      142
100 ' GEEF INSTRUCTIE OP SCHERM *****  0
110 LOCATE 8,0: PRINT "MSX COMPUTER M
AGAZINE"                                    15
120 LOCATE 5,2: PRINT "INVOER CONTROL
E PROGRAMMA/5"                              15
130 LOCATE 0,5: PRINT "Dit programma
maakt het mogelijk om de listings ui
t dit blad foutloos in te voeren."        242
140 PRINT "Bij het intikken van progr
amma-regelsverschijnt, nadat u op 'RE
TURN' of 'ENTER' gedrukt heeft een
getal linksop de onderste regel."        208
150 PRINT "Dit getal moet gelijk zijn
aan de bijde listing afgedrukte chec
ksumwaarde.Als dit niet zo is, dan is
er een fout gemaakt bij het intik
ken."                                       33
160 PRINT "Let op, Basic-woorden moet
en met hoofdletters geschreven wo
rden!"                                     221
170 ' INSTALLEER MACHINECODE *****    0
180 FOR R=0 TO 206                          141
190 READ A$                                  8
200 CS=CS+(ASC(LEFT$(A$,1))+ASC(RIGHT
$(A$,1))*2)*R                              182
210 IF LEFT$(A$,1)<>"*" THEN POKE B+R
,VAL("&H"+A$): GOTO 250                    65
220 IF A$="*" THEN READ A$: AB=B+VAL
("&H"+A$): POKE B+R,AB-(INT(AB/256)*2
56): R=R+1: POKE B+R,INT(AB/256): GOT
O 250                                       217
230 IF A$="*1" THEN READ A$: AB=B+VAL
("&H"+A$): POKE B+R,AB-(INT(AB/256)*2
56): GOTO 250                              192
240 IF A$="*2" THEN POKE B+R,INT(AB/2
56): GOTO 250                              117
250 NEXT R                                   44
260 ' KONTOLEER DATA-WAARDES *****    0
270 IF CS<>3363620# THEN CLS: PRINT "
U heeft een fout gemaakt in de data-
regels!": PRINT: PRINT "Eerst verbete
ren!": STOP                                110
280 ' ZET CHECKSUMROUTINE AAN *****    0
290 A=USR0(0)                                33
300 PRINT: PRINT "Begint u maar met i
ntikken"                                    210
310 NEW                                       42
320 ' MACHINECODE *****                0
330 DATA 21,5E,F5,7E,23,FE,20,20,FA,7
E,23,FE,20,28,FA,FE,27,28,D,FE,52,20,
D,7E,FE,45,C0,23,7E,FE,4D,C0,E1,C3,**
,9F,FE,72,C0,7E                            59
340 DATA FE,65,C0,23,7E,FE,60,28,EF,C
9,36,27,1,6,0,21,**,47,11,7F,F8,ED,B0
,3E,C9,32,DB,FD,C3,**,CB,63,6F,6C,6F,

```

```

72,20,1,6,0                                213
350 DATA 21,7F,F8,11,**,47,ED,B0,21,7
F,F8,36,27,21,83,F8,36,27,23,36,D,21,
DB,FD,36,C3,23,36,*1,71,23,36,*2,C9,1
E,0,CD,**,00,21,5E                          4
360 DATA F5,6,1,7E,4F,FE,27,28,B1,18,
1,4E,3E,0,B9,28,16,C5,16,7,CB,39,30,1
,80,15,28,4,CB,20,18,F4,83,5F,C1,4,23
,18,E4,6B                                    45
370 DATA 11,80,F8,26,0,1,64,0,CD,**,B
9,1,A,0,CD,**,B9,1,1,0,CD,**,B9,18,12
,37,3F,3E,0,ED,42,FA,**,C5,3C,18,F8,C
6,30,12                                       55
380 DATA 13,9,C9,CD,C9,0,C9                10

```

## HOE DE LISTINGS IN TE TIKKEN

MSX Computer Magazine publiceert alleen programma's die door de redactie uitgebreid getest zijn op hun deugdelijkheid. Om te voorkomen dat er bij het zetten alsnog fouten insluipen wordt fotografisch zetwerk, van listings die rechtstreeks van dit geteste programma gemaakt zijn, gebruikt. Deze listings zijn van een speciaal formaat, dat ontworpen is om fouten tijdens het intikken zoveel mogelijk te voorkomen.

In programma's is iedere letter, ieder cijfer en elk lees-teken van belang. De kleinste vergissing bij het intikken kan desastreuze gevolgen hebben.

Om verwarring tussen de hoofdletter 'O' en het cijfer '0' te vermijden is de nul altijd doorgestreept.

De kolommen bevatten 37 tekens, programmaregels die langer zijn worden na het 37ste teken afgebroken, net zoals dit op het beeldscherm van uw MSX1 computer gebeurt.

Programma's die alleen voor MSX2 geschikt zijn worden echter met een breedte van 80 tekens per regel afgedrukt, de standaard scherm breedte van het MSX2 tekstscherf.

De getallen die in een aparte kolom rechts naast de eigenlijke listing staan moet u *niet* intikken, dit zijn de controlegetallen die samen met het Invoer Controle Programma/5 u het mogelijk maken om een listing in een keer foutloos in te tikken.

Een veel voorkomende fout tijdens het intikken is het vergeten van de RETURN, die na *iedere* programmaregel moet worden ingetikt. Ook als de vorige regel precies 37 - of 80, bij MSX2 - tekens lang is, zodat de cursor al vooraan de volgende regel staat, is dit *absoluut* noodzakelijk. Een voorbeeld hiervan kunt u in het ICP/5 zelf zien; regel 170 is precies 37 tekens lang. Als u nu de RETURN vergeet, dan zal het lijken of regel 180 gewoon in het programma staat. Bij het listen verschijnt deze regel normaal.

U kunt 180 dan echter niet apart listen, want volgens de computer maakt die regel gewoon deel uit van regel 170, een kommentaar-regel. Dit leidt uiteindelijk tot een foutmelding:

NEXT WITHOUT FOR IN 250

Want pas in regel 250 ontdekt de computer dat er een FOR-kommando ontbreekt, omdat regel 180 niet als zelfstandige regel in de computer is ingevoerd.

# SPECIALE AANBIEDING

## Een abonnement op MSX Computer Magazine met tien gulden korting

Mogelijk smaakt de inhoud van dit grote listingboek naar meer. Wilt u vaker dergelijke eersteklas

programma's voor uw MSX computer onder ogen krijgen. Dat kan natuurlijk, door een abonnement te nemen op MSX Computer Magazine. In ieder nummer van MSX Computer Magazine staan vele pagina's *listings*, van dezelfde onovertreffelijke kwaliteit als in dit listingboek.

Maar behalve die prima listings staat er nog veel meer in MSX Computer Magazine.

Zo nemen we altijd de nieuwste MSX-*hardware* onder de loep, waarbij we naast de sterke punten ook de zwakheden duidelijk naar voren brengen, zodat onze lezers niet blindelings in het diepe hoeven te springen bij de aankoop van bijvoorbeeld een printer.

Onze *software-tests* zijn berucht, althans bij de leveranciers van slechte programma's, want onze recensenten nemen geen blad voor de mond. Natuurlijk testen we niet alleen spelletjes, ook serieuze programma's komen uitgebreid aan bod.

Verder hebben we allerlei interessante *achtergrondverhalen*, waarin de meest uiteenlopende aspecten van de computerhobby worden aangesneden. Of het nu de *telesoftware* uit Viditel is of een verhaal over een

van de grote *computer-beurzen*, u leest het in MSX Computer Magazine.

In de *brievenrubriek* komen allerlei problemen aan bod waar onze lezers mee kampen. Hardware, software, noem maar op. Wie snel beantwoord wil worden, die kan iedere week gebruik van ons telefonisch vragenuurtje, een bijzondere service voor onze lezers.

### Voor ieder wat wils

Zo bevat MSX Computer Magazine ieder nummer weer opnieuw onderwerpen die zowel de beginner als de gevorderde computeraar aanspreken. En, natuurlijk, onze eersteklas listings, maar die kent u nu al.

### Korting

Speciaal voor de kopers van dit boek hebben we een bijzonder aanbod in petto. Met de speciale kaart, die u in dit boek vinden kan, krijgt u een namelijk een korting van maar liefst *tien gulden* op onze normale abonnements-prijs van f 50,-. Voor slechts f 40,- ontvangt u een jaar lang MSX Computer Magazine. Acht nummers volgepakt met programma's, tests, achtergrondverhalen en wetenswaardigheden.

Kortom, stuur in die bon. Als u tenminste het onderste uit uw MSX-computer wilt kunnen halen.

Categorie: utility's

# MSXmem

Met enige trots publiceren wij een wel heel speciaal stukje MSX-gereedschap, MSXmem. Een hulpprogramma dat een einde moet maken aan al het gekraak rond de MSX-geheugen perikelen.

Wat het doet is - althans op het eerste gezicht - vrij simpel. MSXmem laat namelijk keurig op het scherm zien *hoe* nu precies de inwendige architectuur van een MSX-computer in elkaar steekt. Waar wat zit, als het ware.

Daarbij weet MSXmem bijvoorbeeld *al het geheugen* op te sporen, zowel op MSX1 als op MSX2. Ook een eventuele memory-mapper ontsnapt niet aan het onderzoek dat MSXmem door de slots van een MSX-computer heen uitvoert. Net zomin als allerlei andere extra's, trouwens.

## Voorbehoud

Daarbij willen we echter een voorbehoud maken. Tijdens het ontwerpen en testen van MSXmem hebben we *alle* computers waar we de hand op konden leggen - en dat zijn er heel wat geweest - betrokken, juist omdat MSXmem natuurlijk *wel* op alle machines moet kunnen draaien. We denken dan ook dat MSXmem (bijna) alle voorkomende situaties aan moet kunnen. Er is echter ook nu al een computer waar het niet op werkt en waar het programma ook niet voor aan te passen bleek.

Mogelijk worden er in de toekomst meer van dergelijke 'vreemde' modellen uitgebracht, zodat we het niet aandurven om de goede werking van MSXmem te garanderen. Maar een fabrikant moet ze wel heel bruin bakken, om MSXmem te misleiden!

## Funktie

Wat MSXmem precies doet is eigenlijk heel simpel; het programma probeert namelijk na te gaan hoe de slot-structuur van een MSX-computer nu precies in elkaar steekt. Zoals u in het artikel over MSX-geheugenstructuren elders in dit nummer kunt lezen kan dat na-

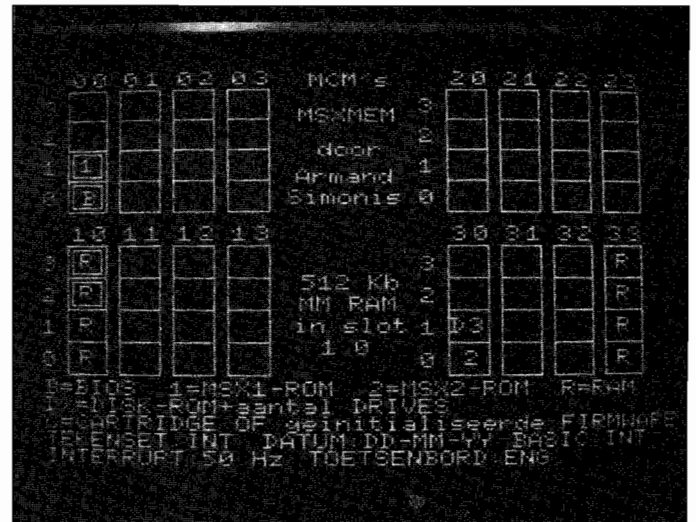
melijk nogal verschillend zijn. Zo zal de ene computer een minimaal stukje geheugen van 16K bevatten in slot 2, pagina 3, terwijl een andere machine met een massaal blok RAM in een memory-mapper kan zijn uitgerust. Die memory-mappers kunnen per slot van rekening wel 4 Megabyte aan geheugen aan, per stuk dan wel te verstaan. Meer dan een enkele memory-mapper mag namelijk ook nog...

Daarnaast kan de fabrikant ook nog eens allerlei extra ROM-software in een MSX inbouwen, de zogenaamde firmware. Een trend die de laatste tijd - bij de nieuwste MSX2 modellen - steeds meer opgaat. Ook die firmware zal MSXmem vinden, als ze tenminste geïntialiseerd is. Een diskdrive-interface wordt natuurlijk ook ontdekt en aangegeven, compleet met het aantal drives wat er op aangesloten is.

Kortom, met MSXmem verschijnen 'de diepste geheimen' van uw computer op het scherm. Maar om die 'diepste geheimen' te kunnen begrijpen moet u wel wat kennis hebben van de manier waarop de MSX met zijn geheugen omgaat. Vandaar dan ook dat we MSXmem in hetzelfde nummer hebben opgenomen als ons diepgravende artikel over dat MSX-geheugen.

## Gebruik

Het gebruik van MSXmem is de eenvoud zelf. Gewoon laden en runnen, verder niet. Om het programma na afloop



weer te stoppen dient u de Control-Stop toetscombinatie te gebruiken, er is niet voorzien in een extra kommando voor dat stoppen.

Als MSXmem eenmaal gestart is, dan zal het om te beginnen een grafisch plaatje op het scherm zetten dat de interne geheugen-structuur van een MSX moet voorstellen. De vier MSX-slots worden ieder apart aangegeven in een van de vier windstreken van het scherm.

Daar ieder slot weer uit vier subslots kan bestaan worden ook die aangegeven, door verticale balkjes. Boven die balken staan nummers, die de slot/subslot indeling aangeven. Zo vinden we links-boven op het scherm de vakken 00, 01, 02 en 03, die achtereenvolgens staan voor de subslots 0 tot en met 3 van slot 0. Op die manier is ieder van de in totaal maximaal mogelijke 16 sub-slots aangegeven.

Elk subslot kan in principe maar liefst 64K aan adresruimte omvatten, die weer onderverdeeld is in 4 pages van elk 16K. Die pagina's staan ook voor ieder subslot aangegeven, doordat de verticale vlakken - die de subslots voorstellen - in vier stukken verdeeld zijn. Naast iedere pagina staat het pagina-nummer, dat tussen de 0 en de 3 kan liggen. In totaal worden er dus 64 pages op het scherm afgebeeld, 4 in ieder van de 16 sub-slots.

Nu kan in principe iedere van die 64 pages in gebruik zijn voor het een of ander, maar ze kunnen natuurlijk niet alle 64

tegelijktijd binnen het Z80 adresbereik vallen. Die Z80 processor kan maximaal 64K aan vier pages dus.

Daarbij geldt bovendien dat de *plaats* van die pages vastgelegd is, een page 0 kan nooit als page 3 worden ingeschakeld. Met andere woorden, de MSX moet een keuze maken uit de beschikbare pages; er mag slechts een van de zestien mogelijkheden voor page 0 in het adresbereik geschakeld zijn, net zoals er maar een enkele page 1, 2 of 3 kan worden aangewezen op een bepaald moment. Voor ieder van die vier pages is er dan ook een keuze uit maximaal zestien mogelijkheden.

## Invulling

Nu is het natuurlijk niet zinnig om willekeurig uit die pages te kiezen. De meeste ervan zullen zelfs geheel 'leeg' zijn, dat wil zeggen dat ze noch ROM noch RAM bevatten. Dergelijke pages zal de MSX nooit selecteren onder het opstarten, als er zinniger alternatieven aanwezig zijn tenminste.

Wat er in welke page te vinden is geeft MSXmem aan door in het vakje van de betreffende pagina een kode te zetten. Daarbij vinden we de volgende mogelijkheden:

- B BIOS-ROM;
  - 1 MSX1-ROM;
  - 2 MSX2-ROM;
  - R RAM;
  - C Cartridge of geïntialiseerde firmware;
  - D Disk-ROM, gevolgd door aantal logische drives wat aan deze interface is aangesloten.
- Dat laatste verdient wat extra

uitleg. Als een MSX met een diskdrive namelijk wordt aangezet, dan zal de computer die ene feitelijke diskdrive toch als twee logische drives initialiseren. Mocht u met een diskkommando die tweede (B) drive aanspreken, dan wordt u via het scherm vriendelijk verzocht om de diskette voor die B-drive in de disk-eenheid te steken. Dat maakt kopiëren etcetera wel wat simpeler maar kost ook extra geheugen, voor de buffers.

Maar als u de machine aanzet met de Control-toets ingedrukt, dan wordt er maar ruimte voor een enkele logische drive gereserveerd, en zal MSXmem ook dat keurig melden.

Op het scherm laat MSXmem keurig zien wat er nu precies in welke page van ieder subslot aanwezig is. Daarbij worden de op dat moment geselecteerde pages aangegeven door er een extra vierkantje in te tekenen. Men ziet dus in een oogopslag welke pages op dat moment deel uitmaken van het Z80 adresbereik, maar ook wat er nog meer in de computer aanwezig is. Bovendien wordt een eventuele memory-mapper nog eens apart gemeld, midden op het scherm.

Een eventuele tweede memory-mapper gaat de capaciteiten van MSXmem te boven, die zal niet meer op het scherm verschijnen.

### Verdere wetenswaardigheden

Ook wat andere kleine wetenswaardigheden over uw MSX worden in beeld gebracht, op de onderste twee schermregels. Daar kunt u bijvoorbeeld zien welke tekenset uw machine heeft - Japans, Internationaal of Koreaans - en welk datum-formaat er gehanteerd wordt. Verder de Basic-versie, die echter wel altijd internationaal zal wezen. De speciale Japanse Basic is wat zeldzaam in Europa.

De interrupt-frekwentie kan 50 of 60 Hertz bedragen, het toetsenbord bestaat in vele versies. Meestal zal het overigens een Engels toetsenbord zijn, maar de mogelijkheden zijn momenteel:

Japans, internationaal, Frans, Engels, DIN, Koreaans, Deens, Noors, Zweeds, Spaans en Russisch.

### Problemen

Op een paar MSX-computers zal MSXmem onoverkomelijke problemen geven. Ten eer-

ste zijn dat alle computers met minder dan 32K RAM, omdat het programma zelf minimaal dat geheugen nodig heeft.

Verbazender was echter om te ontdekken dat het op de Sony HB-F75p tot een reset leidde. Onderzoek en ruggespraak met de importeur wees uit dat deze MSX - onder andere door de ingebouwde database - een iets andere ROM gebruikt, waardoor MSXmem in problemen komt.

### Voorbeeld

Tenslotte nog een voorbeeldje van wat MSXmem allemaal weet uit te vinden. Op de schermfoto bij dit verhaal ziet u de resultaten op een Sony HB-F700P, een van huis uit al behoorlijk uitgebouwde MSX2.

Om het helemaal ingewikkeld te maken hadden we deze computer bovendien nog eens voorzien van een losse memory-mapper van 512K, die in een van de slots gestoken was. Die memory-mapper was dan ook nog eens als RAM-disk geïnitieerd.

Deze losse memory-mapper is overigens een prototype waarvan we ook nog niet weten

wanneer deze in de verkoop zal komen.

Dat alles wist MSXmem keurig te ontdekken. We zien zelfs dat er in slot 3-0, page 1, een disk-interface met drie drives zit, de RAM-disk wordt ook meegeteld. Slechts de ingebouwde memory-mapper van de HB-F700p - 256K RAM in slot 3-3 - ontsnapt, daar MSXmem aanneemt dat er slechts een enkele memory-mapper aanwezig zal zijn. Mogelijk zullen we het programma in de toekomst, als dergelijke losse memory-mappers op de markt komen, nog eens voor aanpassen.

### Tenslotte

MSXmem bestaat voor een zo groot mogelijk deel uit normale Basic, slechts waar het niet anders kon hebben we naar machinetaal gegrepen. Die Basic-listing is behoorlijk van commentaar voorzien, zodat voor geïnteresseerden een en ander hopelijk wel te volgen is.

Daarnaast hebben we de machinetaal-sourcecode ook opgenomen. Alweer voorzien van rijkelijk veel commentaar, zodat ML-programmeurs onze truuks desgewenst kunnen gebruiken.

#### Slotselect-register uitlees routine \*\*\*\*\*

```

CD,38,01 : CALL RSLREG :geeft de primaire slot situatie
32,FF,AF : LD (AFFF),A :schrijft het weg voor de BASIC
FB : EI :zet de interrupt weer aan
C9 : RET :terug naar BASIC-routine

```

#### Page 3 Machine-code routine \*\*\*\*\*

```

F3 : DI :zet de interrupt uit
ED,73,0,B8 : LD (B800),SP :bewaart de oude stack-pointer
31,00,AF : LS SP,AF00 :verandert de stack-pointer
CD,38,01 : CALL RSLREG :leest primair slot register
F5 : PUSH AF :bewaren voor terug naar BASIC
21,00,D0 : LD H&H442C000 :page 3 nummer
E5 : PUSH HL :ook bewaren
3E,00 : LD A,(slot) :slot nummer in BASIC-loop
CD,24,00 : CALL ENASLT :(voor)-selecteert slot
E1 : POP HL :ophalen van HL
7E : LD A,(HL) :leest wat er op D000 staat
2F : CPL :complement van accumulator
77 : LD A,(HL) :schrijft de nieuwe waarde terug
AE : XOR (HL) :A wordt nul als er RAM staat
CA,24,B0 : JP Z,true :RAM ? Y=>true ;N verder
F1 : POP AF :ophalen voor terug naar BASIC
CD,38,01 : CALL WSLREG :schrijft primair slot register
C3,32,B0 : JP terug :gaat naar terug keer routine
3A,11,B0 : true: LD A,(B011) :leest het slotnummer met RAM in
32,FF,AF : LD (AFFF),A :schrijft het weg voor BASIC
F1 : POP AF :ophalen voor terug naar BASIC
CD,38,01 : CALL WSLREG :schrijft primair slot register

```

```

0,0,0,0  : NOP's           :bladvulling
21,00,D0  : terug: LD HL,D000    :page 3 nummer
3E,00     : LD A,data         :wordt gevult door BASIC-loop
CD,24,00  : CALL ENASLT     :(voor)-selecteerd slot
ED,7B,0,B8: LD SP,(B800)       :zet stack-pointer weer terug
FB        : EI              :de interrupt weer aan..
C9        : RET          :terug naar BASIC-routine

```

Page 1-2-3 Machine-code routine \*\*\*\*\*

```

21,FF,3F  : LD HL,3FFF    :page nummer door BASIC bepaald
3E,00     : LD A,data     :slot nummer door BASIC bepaald
F5        : PUSH AF       :bewaren voor terug naar BASIC
CD,0C,00  : CALL RDSLT   :lezen van data in slot
2F        : CPL          :complement van accumulator
5F        : LD E,A      :nodig voor schrijven met WRSLT
F1        : POP AF      :slot nummer ook natuurlijk..
F5        : PUSH AF     :wel even bewaren voor later
CD,14,00  : CALL WRSLT  :schrijf E in slot A op HL
F1        : POP AF     :slot nummer weer ophalen
F5        : PUSH AF     :maar ook weer bewaren
D5        : PUSH DE     :samen met E
CD,0C,00  : CALL RDSLT  :lezen uit slot A ,page HL
D1        : POP DE     :E is weer terug..
AB        : XOR E       :E en A gelijk ?
CA,1E,B0  : JP Z,true    :Y,dan RAM=> true
F1        : POP AF     :ophalen voor terug naar BASIC
FB        : EI         :interrupt weer aan
C9        : RET        :terug naar BASIC
F1        : true: POP AF :ophalen...
32,FF,AF  : LD(AFFF),A  :slotnummer voor BASIC-routine
FB        : EI         :interrupt weer aan
C9        : RET        :terug naar BASIC
C9        : RET        :terugkeer naar BASIC

```

Memory mapper Machine-code routine \*\*\*\*\*

```

3E,00     : LD A,data     :slot nummer door BASIC bepaald
21,00,40  : LD HL,4000    :page nummer=1
CD,24,00  : CALL ENASLT  :(voor)-selecteert slot
3E,00     : LD A,data     :slot nummer door BASIC bepaald
CD,3B,01  : CALL WSLREG   :primair slotselect
3E,02     : LD A,02      :
D3,FD     : OUT (FD),A    :page 1 in memory-blok 2
21,00,40  : LD HL,4000    :willekeurig adres uit page 1
36,76     : LD (HL),76   :laadt willekeurig byte ('HALT')
3C        : INC A        :a=a+1 in memory-blok 2
D3,FD     : OUT (FD),A    :volgend memory-blok..
3A,00,40  : LD A,(4000)  :leest weer de inhoud
EE,76     : XOR 76      :als uitkomst 0 dan toch geen MM
C2,25,B0  : JP NZ,verder :
AF        : XOR A        :laadt A met 0
C3,3B,B0  : JP terug     :naar terugkeer-routine
3E,06     : verder:LD A,6  :kijkt 64 K verder
D3,FD     : loop:OUT(FD),A :schakelt in
21,00,40  : LD HL,4000
4E        : LD C,(HL)    :leest weer in
F5        : PUSH AF     :bewaart memory-blok nummer
79        : LD A,C      :geeft door aan accumulator
EE,76     : XOR 76      :als uitkomst 0 dan klaar
28,05     : JP Z,klaar   :tellen van blokken gereed
F1        : POP AF     :ophalen blok-nummer
C6,04     : ADD A,4      :telt er vier bij op
18,EF     : JR loop     :opnieuw testen
F1        : klaar:POP AF  :ophalen aantal memory blokken
D6,02     : SUB 2        :corrigeren..
32,FF,AF  : terug:LD (AFFF),A :wegschrijven voor BASIC-routine
3E,02     : LD A,2      :page 1 weer terug in oude stand
D3,FD     : OUT (FD),A  :..wat hier plaats vindt..
3A,C1,FC  : LD A,(FC1)  :ROM-problemen voorkomen
21,00,40  : LD HL,4000  :page 1
CD,24,00  : CALL ENASLT  :(voor)-selecteren
3E,00     : LD A,00     :slot nummer door BASIC bepaalt
CD,3B,01  : CALL WSLREG   :schakelt primair slotregister
FB        : EI         :interrupt aan

```

```

10 REM MSXmem 0
20 REM 0
30 REM MSX Computer Magazine 0
40 REM 0
50 REM Een programma dat laat zien 0
60 REM hoe uw MSX-computer het ge- 0
70 REM heugen in zich heeft. Inklus- 0
80 REM sief cartridges en disk-drives 0
90 REM 0
100 REM Met de gegevens van MSX-TYPE 0
110 REM 0
120 DEFINT A-Z: COLOR 15,4,4: SCREEN 105
2: OPEN "GRP:" AS #1
130 DIM F1(16) 44
140 ' Funkties scherm-opbouw ***** 0
150 DEF FN X(A,B)=20-140*(A>=2)+B*20 244
160 DEF FN Y(A,B)=10-76*(((A=1)+(A=3) 84
)<>0)
170 DEF FN XX(M)=FN X(M AND 3,(M AND 5
12)/4)+5
180 DEF FN YY(M)=FN Y(M AND 3,(M AND 8
12)/4)+4
190 ' Bouw scherm ***** 0
200 FOR T=0 TO 3 164
210 FOR E=0 TO 3 29
220 PRESET (FN X(T,0)-10,5+15*E+FN 58
N Y(T,U))
230 PRINT #1,RIGHT$(STR$(3-E),1) 1
240 NEXT E 80
250 FOR U=0 TO 3 213
260 A=FN X(T,U):B=FN Y(T,U) 129
270 PRESET (A+2,B-9): PRINT #1,RI 141
GHT$(STR$(T),1);RIGHT$(STR$(U),1)
280 DRAW "BM=A;:B;R15D15NL15D15N 114
L15D15NL15D15L15U60"
290 NEXT U 26
300 NEXT T 55
310 GOSUB 1590: 'MSX-type gegevens op 142
scherm zetten
320 ' BIOS-ROM ***** 0
330 PRESET (FN X(0,0)+6, FN Y(0,0)+3*1 0
5+5): 'BIOS altijd in slot 0-0, page
340 PRINT #1,"B" 156
350 ' MSX1-ROM ***** 173
360 M=PEEK(&HFCC1): ' bevat slot adre 0
s MSX1-ROM 10
370 PRESET (FN X(0,(M AND 12)/4)+5, FN 64
Y(0,(M AND 12)/4)+2*15+4)
380 PRINT #1,"1" 182
390 ' MSX2-ROM ***** 0
400 IF PEEK(&H2D)=0 THEN 460: ' Bij M 83
SX1 geen MSX2-ROM
410 M=PEEK(&HFAF8): ' bevat slot adre 181
s MSX2-ROM 38
420 IF M=0 THEN 460
430 PRESET (FN X(M AND 3,(M AND 12) 9
/4)+6, FN Y(M AND 3,(M AND 12)/4)+3*15
+4) 100
440 PRINT #1,"2" 0
450 ' Disk(s) ROM ***** 215
460 IF PEEK(&HFA7)=&HC9 THEN 540 114
470 FOR T=&HFB22 TO &HFB28 STEP &H2 165
480 DN=DN+1 160
490 IF PEEK(T)=0 THEN 530
500 M=PEEK(T): ' bevat slot adres 255
disk-ROM
510 PRESET (FN XX(M)-4, FN YY(M)+ 53
2*15)
520 PRINT #1,"D";: DRAW "BL9": P 169
RINT #1,PEEK(T-1): ' Bevat aantal dri 146
ves
530 NEXT T 0
540 ' Slotselect register *****

```

```

550 RESTORE 1790 20
560 FOR T=&HB000 TO &HB007 82
570 READ A$ 43
580 POKE T,VAL("&H"+A$) 56
590 NEXT T 158
600 DEFUSR=&HB000: DU=USR(0): 'Lees h 155
et primaire slotregister 201
610 SL=PEEK(&HAFFF): T=128: Y=0: P=SL 131
620 IF T<1 THEN 660 47
630 Q=P\T
640 IF Q>=1 THEN A(Y)=1: P=P-T ELSE 77
A(Y)=0
650 Y=Y+1: T=T/2: GOTO 620 56
660 FOR P=0 TO 3 144
670 S=A(P*2)*2+A(P*2+1) 185
680 IF PEEK(&HFCC1+S)<>&H80 THEN E= 138
0: GOTO 700: 'Is slot geëxpandeerd
690 E=(PEEK(&HFCC5+S) AND (4^(3-P)* 168
3))/(4^(3-P)): 'Zo ja, lees dan regis 169
ter
700 A1=FN X(S,E)+2: A2=FN Y(S,E)+P* 64
15+2 23
710 LINE (A1,A2)~(A1+11,A2+11),,B: 158
'Geef de ingeschakelde pages aan 0
720 NEXT P 157
730 P=0: S=0: E=0 96
740 ' RAM page 3 ***** 172
750 RESTORE 1800 21
760 FOR T=&HB000 TO &HB040 77
770 READ A$ 246
780 POKE T,VAL("&H"+A$) 85
790 NEXT T 131
800 DEFUSR=&HB000 230
810 PA=(SL AND &HC0)/64
820 IF PEEK(&HFCC1+PA)=0 THEN 840
830 PA=PA+(PEEK(&HFCC5+PA) AND &HC0 54
)/16+&H80 219
840 POKE &HB036,PA: 'Stel slot page 3 25
in voor ENASLT 139
850 FOR T=0 TO 15: 'Slots 0-0 t/m 3-3 28
in theorie 70
860 POKE &HAFFF,&HFF
870 POKE &HB011,PEEK(&HFCC1+(T AND 25
3))+T: 'Slot van page 3 waar naar RAM
gezocht wordt
880 DU=USR(0)
890 IF PEEK(&HAFFF)=255 THEN 940: 'Is
het RAM?
900 M=PEEK(&HAFFF): 'Zo ja, lees sl 0
otnummer 73
910 IF PEEK(&HFCC1+(T AND 3))=0 THE 242
N M=M AND 3: 'Is het een expanded slo 30
t? 241
920 PRESET (FN XX(M), FN YY(M)) 69
930 PRINT #1,"R" 0
940 NEXT T 129
950 ' RAM page 0-1-2 ***** 128
960 Z=1: RESTORE 1820 176
970 FOR T=&HB000 TO &HB023 25
980 READ A$ 200
990 POKE T,VAL("&H"+A$) 24
1000 NEXT T 130
1010 DEFUSR=&HB000
1020 FOR Y=1 TO 3: 'De pages 0 t/m 2 73
1030 FOR T=0 TO 15: 'De slots 0-0 t 64
/m 3-3 in theorie
1040 POKE &HB002,Y*64-1: 'Geeft p 52
age voor RDSLt en WRSLT 185
1050 POKE &HAFFF,&HFF: POKE &HB00 168
4,PEEK(&HFCC1+(T AND 3))+T: 'Geef slo
t
1060 DU=USR(0)
1070 IF PEEK(&HAFFF)=255 THEN 113
0: 'Is het RAM?

```



```

1080 M=PEEK(&HAFFF): 'Zo ja, lees
slotnummer 249
1090 IF PEEK(&HFCC1+(T AND 3))=0
THEN M=M AND 3: 'Is slot geexpandeed
? 172
1100 IF Y=2 AND FI(Z-1)<>M THEN F
1(Z)=M: Z=Z+1: 'Bewaren voor memory-m
apper routine 153
1110 PRESET (FN XX(M),FN YY(M)+(4
-Y)*15) 127
1120 PRINT #1,"R" 10
1130 NEXT T 217
1140 NEXT Y 13
1150 ' Cartridge of geinitialiseerde
firmware ***** 0
1160 FOR S=0 TO 3 107
1170 FOR E=0 TO 12 STEP 4 52
1180 FOR P=0 TO 3 123
1190 IF PEEK(S*16+E+P+&HFCC9)=0
THEN 1260: ' bevat info over elke pag
e 183
1200 M=S+E 249
1210 FOR T=0 TO 5: 'Is deze p
age al in gebruik? 42
1220 IF POINT(FN XX(M)+T,FN
YY(M)+(3-P)*15+4)<>PEEK(&HF3EA) THEN
1260 64
1230 NEXT T 237
1240 PRESET(FN XX(M),FN YY(M)+(
3-P)*15): PRINT#1,"C" 125
1250 PRINT #1,"C" 241
1260 NEXT P 46
1270 NEXT E 36
1280 NEXT S 217
1290 ' Memory-Mapper ***** 0
1300 IF PEEK(&H2D)=0 THEN 1510: ' is h
et geen MSX1 ? 67
1310 OUT &HFD,2: OUT &HFC,3: IF INP(&
HFD)=INP(&HFC) THEN 1510: ' Zit er we
l een MM in ? 105
1320 POKE &HAFFF,0: RESTORE 1830 107
1330 FOR T=&HB000 TO &HB051 113
1340 READ DS 146
1350 POKE T,VAL("&H"+D$) 193
1360 NEXT T 224
1370 DEFUSR=&HB000 48
1380 FOR T=1 TO Z-1: 'De pages 1 met
AM bekijken 249
1390 POKE &HB001,F1(T) 225
1400 POKE &HB009,SL+((F1(T) AND 3)*
4) 207
1410 POKE &HB04C,SL: 'Page 1 om naa
r Basic terug te kunnen keren 140
1420 DU=USR(0): 'Is dit een Memory-
Mapper? 48
1430 AB=PEEK(&HAFFF): 'Lees het aan
tal blokken RAM 223
1440 IF AB<>0 THEN SM=F1+T): GOTO 1
470: 'Blokken gevonden? Zo ja, klaar
1450 NEXT T 223
1460 ' Meldt Memory-Mapper ***** 0
1470 TX$=STR$(AB*16)+" Kb" 187
1480 XT=95: YT=100: GOSUB 1720: TX$="
MM RAM": XT=100: YT=110: GOSUB 1720
1490 TX$="in slot": XT=98: YT=120: GO
SUB 1720: TX$=STR$(SM AND 3)+STR$(-(S
M AND &HC)/4): XT=102: YT=130: GOSUB
1720 69
1500 ' Programma-gevans ***** 0
1510 TX$="MCM's": XT=104: YT=2: GOSUB
1720 240
1520 TX$="MSXMEM": XT=100: YT=20: GOS
UB 1720 8
1530 TX$="door": XT=107: YT=37: GOSUB

```

```

1720 126
1540 TX$="Armand": XT=99: YT=50: GOSU
B 1720 206
1550 TX$="Simonis": XT=96: YT=60: GOS
UB 1720 250
1560 BEEP 4
1570 GOTO 1570 28
1580 ' MSX type gegevens ***** 0
1590 RESTORE 1770 0
1600 FOR T1=0 TO 2 249
1610 READ FX$ 83
1620 FOR T=1 TO LEN(TX$) 108
1630 'PRESET (5+T*6,149+T1*8): PRI
NT#1,MID$(TX$,T,1) 122
1640 NEXT T 230
1650 NEXT T1 47
1660 I=PEEK(&H2B) AND 15: TX$="TEKENS
ET:"+MID$("JAPINTKOR",1+I*3,3): XT=5:
YT=173: GOSUB 1720 64
1670 I=(PEEK(&H2B)AND112)/16: TX$="DA
TUM:"+MID$("YY-MM-DDMM-DD-YYDD-MM-YY"
,1+I*8,8): XT=87: YT=173: GOSUB 1720
219
1680 I=(PEEK(&H2C)AND240)/16: TX$="BA
SIC:"+MID$("JAPINT",1+I*3,3): XT=178:
YT=173: GOSUB 1720 115
1690 I=(PEEK(&H2B)AND128)/128: TX$="I
NTERRUPT:"+MID$("0050",1+I*2,2)+" Hz"
: XT=5: YT=181: GOSUB 1720 5
1700 I=PEEK(&H2C)AND15: TX$="TOETSENB
ORD:"+MID$("JAPINTFRAENGDKORDENNORZ
WESPARUS",1+I*3,3): XT=103: YT=181: G
OSUB 1720: ' Vanaf KOR niet geheel ze
ker 211
1710 RETURN 140
1720 FOR T=1 TO LEN(TX$) 118
1730 PRESET (XT+6*T,YT): PRINT#1, M
ID$(TX$,T,1) 200
1740 NEXT T 226
1750 RETURN 152
1760 ' DATA ***** 0
1770 DATA B=BIOS 1=MSX1-ROM 2=MSX2-
ROM R=RAM, Dx=DISK-ROM+aantal DRIVES
, C=CARTRIDGE OF geinitialiseerde FIR
MWARE 104
1780 ' Machine code ***** 0
1790 DATA CD,38,01,32,FF,AF,FB,C9 190
1800 DATA F3,ED,73,0,B8,31,0,AF,CD,38
,1,F5,21,0,D0,E5,3E,8F,CD,24,0,E1,7E,
2F,77,AE,CA,24,B0,F1,CD,3B,1 55
1810 DATA C3,32,B0,3A,11,B0,32,FF,AF,
F1,CD,3B,1,0,0,0,21,0,D0,3E,0,CD,24
,0,ED,7B,0,B8,FB,C9,FF 201
1820 DATA 21,FF,3F,3E,83,F5,CD,C,0,2F
,5F,F1,F5,CD,14,0,F1,F5,D5,CD,C,0,D1,
AB,CA,1E,B0,F1,FB,C9,F1,32,FF,AF,FB,C
9 166
1830 DATA 3E,3,21,0,40,CD,24,0,3E,FC,
CD,3B,1,3E,2,D3,FD,21,0,40,36,58,3C,D
3,FD,3A,0,40,EE,58,C2,25,B0,AF,C3,3B,
B0,3E,06 43
1840 DATA D3,FD,21,0,40,4E,F5,79,EE,5
8,28,5,F1,C6,4,18,EF,F1,D6,2,32,FF,AF
,3E,2,D3,FD,3A,C1,FC,21,0,40,CD,24,0,
3E,F0,CD,3B,1,FB,C9 254

```

## CATEGORIE: SPELLEN

# HAL

Bijna iedere computer-hobbyist speelt — tussen alle ernstiger zaken door — wel eens een spelletje op de machine. Om over de echte spel-fanaten nog maar te zwijgen; die doen niets anders. Begrijpelijk overigens, want de mogelijkheden van de computer als spel-apparaat zijn werkelijk legio.

Er bestaan zoveel soorten computerspellen, dat het bijna onbegonnen werk is om ze allemaal in een bepaald type in te delen. Platform-spelen, bord-spelen, schietspelletjes, noem maar op.

In het verleden hebben we al heel wat spel-listings geplaatst, maar één bepaalde spelsoort bent u tot nog toe nog niet in MSX Computer Magazine tegengekomen als listing. Namelijk het doolhofspel, een razend populaire spelsoort onder de vele liefhebbers.

### Labyrint

Aan dat gemis is nu echter een einde gekomen, met HAL — de naam staat voor 'Heel Apart Labyrint' — van Ab Reitsma uit Enschede. HAL is namelijk een joekel van een doolhofspel, met maar liefst twintig schermvullende velden. En in ieder van die velden is het maar al te gemakkelijk om de dood te vinden en zo n van de zes levens te verspelen.

Allerlei ongemak, vanaf roterende sterren tot en met heen en weer racende auto'tjes, maken het uw mannetje lastig. Iedere aanraking betekent dat u het weer eens opnieuw mag proberen! Behalve de bewegende gevaren zijn er ook heel wat stilstaande zaken die u absoluut dient te vermijden. Op allerlei plekken zult u obstakels tegenkomen, zoals scherpe rechtopstaande punten, die

subiet een einde aan het plezier maken. Zes levens is alles behalve een vetpot in HAL!

### Goudstaven

Natuurlijk valt er **meer** in het Heel Apart Labyrint te beleven dan alleen maar een groot aantal manieren om levens te verliezen. Waar het eigenlijk om draait zijn — voorspelbaar — de *goudstaven*, die zo hier en daar verspreid liggen. In totaal zijn het er twintig, in ieder veld is er eentje.

Deze staven dient u op te rapen, door er met uw mannetje overheen te lopen.

Bovenin beeld wordt bijgehouden, in een horizontaal venster, hoeveel staven u al verzameld heeft. Pas als dat venster helemaal vol is, heeft u de totale buit binnen. Maar dat kan wel even duren, voor het zover is.

### Demo-mode

HAL heeft, net zoals vele professionele spellen, een ingebouwde demonstratiemode. Als u het spel na het starten niet meteen gaat spelen, krijgt u vanzelf een demo te zien. Er zijn een paar van die demo's in het spel, tel-

kens verschijnt er weer een andere.

Zo'n demo is ideaal om eerst eens de kat uit de boom te kijken. Tijdens de demonstratie laat de computer zelf een aantal truuks zien, die u broodnodig zult hebben om HAL uit te kunnen spelen. Techniek spreekt zeer zeker een woordje mee, om in HAL alle obstakels te kunnen overwinnen!

### Hefbomen

Sommige velden lijken op het eerste gezicht totaal onoplosbaar. Dat is echter niet zo, als de speler maar eenmaal doorheeft dat HAL-velden 'omschakelbaar' zijn. Op allerlei plaatsen komt men hefbomen tegen, die omgezet kunnen worden door er gewoon overheen te lopen. Als zo'n hefboom eenmaal de andere kant opstaat, blijkt het veld opeens wat anders in elkaar te zitten. Barrières gaan plotseling open — of dicht! — en verschuiven van plaats.

Ieder veld is daardoor oplosbaar, maar het hoeft niet eenvoudig te zijn. We hebben u gewaarschuwd!

### Bediening

HAL kan zowel met het toetsenbord als met de joystick — in poort 1 — gespeeld worden, door één speler. Op het startscherm kan men de bedieningswijze kiezen, door op de F1 te drukken.

Die funktietoets dient als een soort schakelaar; als de bediening op de cursortoetsen is ingesteld, dan zal de F1 daar juist joystick van maken, en omgekeerd.

Met de F2 kan men op dit intro scherm de — korte — gebruiksaanwijzing krijgen, terwijl de F3 het spel laat beginnen.

Op het introscherm treffen we ook een high-score aan.

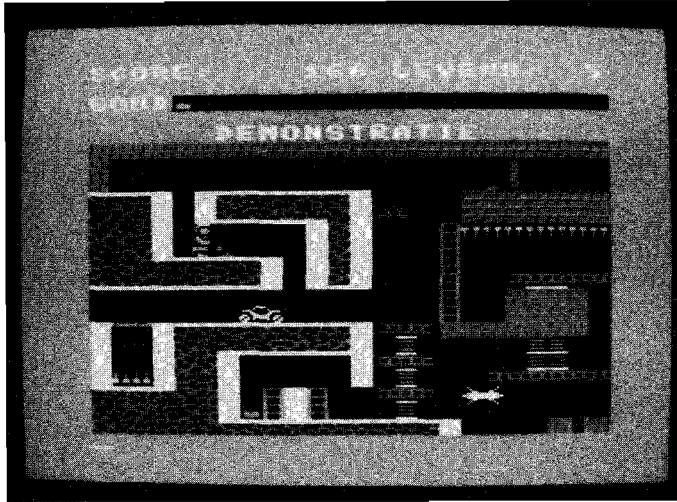
Tijdens het spel kan men de cursor-toetsen gebruiken om links en rechts te gaan; de spatiebalk is goed om mee te springen. Naar boven en beneden gaat niet met de cursor-toetsen, klimmen doet men door — alweer met de spatiebalk — gewoon wat langer te springen. Afdalen is al helemaal simpel, gewoon van een randje afstapen en u gaat naar beneden. Kijk alleen wel even uit of er onderaan de afdaling niet iets dodelijks op u ligt te wachten.

Wie met de joystick speelt gebruikt daar ook alleen maar de linker- en rechterstand van. Springen moet met de vuurknop.

### Vast

Het kan best gebeuren dat men tijdens het spelen helemaal vastloopt. Sommige velden zijn ronduit onspeelbaar, als men vanuit de verkeerde hoek binnenkomt. Een prima reden om zo snel mogelijk een kaart te maken





van het Heel Apart Labyrint, want zonder zo'n kaart is het wel erg lastig. Om nu te voorkomen dat men in zo'n vastgelopen situatie alle levens domweg moet opofferen om weer naar het startveld terug te komen, kan men de F1 en de F2 gebruiken. Met de F1 verliest u een leven en begint weer opnieuw in het huidige veld; met de F2 gaat u zelfs helemaal naar het startveld terug. Met het verlies van een leven natuurlijk, anders wordt het te gemakkelijk.

**Knap**

Naar de mening van de MCM redactie is HAL een razend knap programma, dat geheel in Basic geschreven is. Door de toepassing van allerlei slimme technieken heeft Ab Reitsma kans gezien een behoorlijk snel en zeer gecompliceerd spel in elkaar te zetten.

Met karakter-georiënteerde kleuregraphics op screen 1 is heel wat te doen, zo blijkt. Zeker als er daarnaast ook nog sprites gebruikt worden.

De keerzijde van de medaille is echter wel dat de HAL-listing ook knap lang is. Zo lang, dat MSX2 diskdrive-bezitters zullen moeten opstarten met de Control-toets ingedrukt, anders is er te weinig geheugen beschikbaar op hun machines. In die omstandigheden is dan zo'n 1K vrij geheugen over.

Dat was dan ook te weinig voor ons om er onze gebruikelijke interne checksums in onder te brengen. De datablokken worden niet door het programma zelf gecontroleerd. Natuurlijk kan – en moet, eigenlijk – ICP/5 wel gebruikt worden, als u het maar eerst weer helemaal verwijderd voordat u HAL gaat spelen. Oftewel, even de computer uitzetten, nadat u HAL gesaved heeft.

**Klus**

HAL is de langste Basic-listing ooit in MSX Computer Magazine gepubliceerd. Het intikken zal, temeer daar er – wegens ruimtegebrek – vrijwel geen spaties in HAL staan, een flink stuk werk betekenen. Neem daar gerust een paar avonden de tijd voor; en probeer vooral niet om HAL eventjes snel in te tikken. Dat leidt met zekerheid tot tikfoutjes, waarna het programma het niet zal doen.

Een dergelijke listing intikken zonder daarbij gebruik te maken van het Invoer Controle Programma – voorin dit blad – is bij voorbaat een hopeloos karwei. Daar zullen altijd foutjes bij insluipen.

Voor diegenen tenslotte die zo'n stuk tikwerk echt niet zien zitten, HAL staat natuurlijk ook – foutloos – op de bij dit nummer van MSX Computer Magazine verkrijgbare cassette/diskette.

```

10 REM HAL
20 REM
30 REM MSX Computer Magazine
40 REM
50 REM door: A. Reitsma
60 REM
70 KEYOFF:SCREEN1:WIDTH26:FORX=1T026:
LOCATEX,10:PRINTMID$(" (C) MSX Compute
r Magazine",X,1):PLAY"t255164o4b", "t2
55164o3b", "T255L6402B":FORY=0T099:IFP
LAY(3)THENNEXTELSENEXTX:LOCATE6,15:PR
INT"even geduld a.u.b":FORX=1T03000:N
EXT:GOTO850
80 GOSUB140:A=VPEEK(P):B=VPEEK(P-32):
IFA<100ORB<100THENIFA=32ORB=32THEN440
ELSE800ELSEIFA>122THENONA-122GOTO90,1
00,120:RETURNELSERETURN
90 VPOKEP,122:OB=OB+1:IFOB=20THEN3270
ELSEC=SC+25-(2000>TIME)*(2000-TIME)/
10:O(V)=0:LOCATEOB+3,2:PRINT"{" :LOCAT
E7,0:PRINTUSING"#####";SC:PLAY"v1004
CDEFGAB", "v1105CDEFGAB":FORX=0T01STEP
0:IFPLAY(2)ORPLAY(3)THENNEXTELSETIME=
0:RETURN
100 IFST=100THENST=101:FORX=0T04096ST
EP2048:VPOKEX+995,3:VPOKEX+996,3:VPOK
EX+997,4:VPOKEX+800,0:VPOKEX+801,0:NE
XT:RETURN
110 ST=100:FORX=0T04096STEP2048:VPOKE
X+995,192:VPOKEX+996,192:VPOKEX+997,3
2:VPOKEX+800,255:VPOKEX+801,255:NEXT:
RETURN
120 IFBM=106THENBM=108:BP=120:FORX=0T
04096STEP2048:VPOKEX+1003,3:VPOKEX+10
04,3:VPOKEX+1005,4:FORY=1T014:VPOKEY+
X+848,0:VPOKEX+952+Y,60:NEXT:NEXT:RET
URN
130 BM=106:BP=118:FORX=0T04096STEP204
8:VPOKEX+1003,192:VPOKEX+1004,192:VPO
KEX+1005,32:FORY=1T014:VPOKEY+X+848,6
0:VPOKEX+952+Y,0:NEXT:NEXT:RETURN
140 ONSXGOTO150,180,210,240
150 X1=X1+S1:D1=D1XOR1:IFX1>B1ORX1<A1
THENS1=-S1:X1=X1+S1:PUTSPRITE1,(X1,Y1
),,D1ELSEPUTSPRITE1,(X1,Y1),,D1
160 X2=X2+S2:D2=D2XOR1:IFX2>B2ORX2<A2
THENS2=-S2:X2=X2+S2:PUTSPRITE2,(X2,Y2
),,D2ELSEPUTSPRITE2,(X2,Y2),,D2
170 X3=X3+S3:D3=D3XOR1:IFX3>B3ORX3<A3
THENS3=-S3:X3=X3+S3:PUTSPRITE3,(X3,Y3
),,D3:RETURNELSEPUTSPRITE3,(X3,Y3),,D
3:RETURN
180 X1=X1+S1:D1=D1XOR1:IFX1>B1ORX1<A1
THENS1=-S1:X1=X1+S1:PUTSPRITE1,(X1,Y1
),,D1ELSEPUTSPRITE1,(X1,Y1),,D1
190 X2=X2+S2:D2=D2XOR1:IFX2>B2ORX2<A2
THENS2=-S2:X2=X2+S2:PUTSPRITE2,(X2,Y2
),,D2ELSEPUTSPRITE2,(X2,Y2),,D2
200 Y3=Y3+S3:D3=D3XOR1:IFY3>B3ORY3<A3
THENS3=-S3:Y3=Y3+S3:PUTSPRITE3,(X3,Y3
),,D3:RETURNELSEPUTSPRITE3,(X3,Y3),,D
3:RETURN
210 X1=X1+S1:D1=D1XOR1:IFX1>B1ORX1<A1
THENS1=-S1:X1=X1+S1:PUTSPRITE1,(X1,Y1
),,D1ELSEPUTSPRITE1,(X1,Y1),,D1

```

0  
0  
0  
0  
0  
22  
248  
211  
215  
218  
162  
105  
16  
237  
82  
162  
243  
88  
66  
230

```

220 Y2=Y2+S2:D2=D2XOR1:IFY2>B2ORY2<A2
THENS2=-S2:Y2=Y2+S2:PUTSPRITE2,(X2,Y2)
,,D2ELSEPUTSPRITE2,(X2,Y2),,D2
230 Y3=Y3+S3:D3=D3XOR1:IFY3>B3ORY3<A3
THENS3=-S3:Y3=Y3+S3:PUTSPRITE3,(X3,Y3)
,,D3:RETURNELSEPUTSPRITE3,(X3,Y3),,D
3:RETURN
240 Y1=Y1+S1:D1=D1XOR1:IFY1>B1ORY1<A1
THENS1=-S1:Y1=Y1+S1:PUTSPRITE1,(X1,Y1)
,,D1ELSEPUTSPRITE1,(X1,Y1),,D1
250 Y2=Y2+S2:D2=D2XOR1:IFY2>B2ORY2<A2
THENS2=-S2:Y2=Y2+S2:PUTSPRITE2,(X2,Y2)
,,D2ELSEPUTSPRITE2,(X2,Y2),,D2
260 Y3=Y3+S3:D3=D3XOR1:IFY3>B3ORY3<A3
THENS3=-S3:Y3=Y3+S3:PUTSPRITE3,(X3,Y3)
,,D3:RETURNELSEPUTSPRITE3,(X3,Y3),,D
3:RETURN
270 PLAY"v8o6b":YP=YP+8:PUTSPRITE0,(X
P,YP-4),,W:P=P+32:GOSUB80:PUTSPRITE0,
(XP,YP):GOTO420
280 A=VPEEK(P-1):IFA<BMORA>BPTHENPLAY
"v7o2c":XP=XP-8:PUTSPRITE0,(XP+4,YP),
,S+1:S=-S+2:P=P-1:GOSUB80:PUTSPRITE0,
(XP,YP),,S:W=5
290 GOTO420
300 A=VPEEK(P+1):IFA<BMORA>BPTHENPLAY
"v7o2c":XP=XP+8:PUTSPRITE0,(XP-4,YP),
,S+7:S=-S+2:P=P+1:GOSUB80:PUTSPRITE0,
(XP,YP),,S+6:W=11
310 GOTO420
320 A=VPEEK(P-64):IFA<BMORA>BPTHENPLA
Y"v8o6b":YP=YP-8:PUTSPRITE0,(XP,YP+4)
,,W:P=P-32:GOSUB80:PUTSPRITE0,(XP,YP)
:A=VPEEK(P+32):IFA>=ST ANDA<=BMTHEN42
0
330 IFJS=0THENJS=1:GOTO320ELSE420
340 W=5:A=VPEEK(P-65):B=VPEEK(P-33):I
FA<BMORA>BPTHENIFB>=BMANDB<=BPTHEN320
ELSEPLAY"v8o6b":XP=XP-8:YP=YP-8:PUTSP
RITE0,(XP+4,YP+4),,4:P=P-33:GOSUB80:P
UTSPRITE0,(XP,YP):A=VPEEK(P+32):IFA>=
ST ANDA<=BPTHENPUTSPRITE0,,,W:GOTO420
ELSEIFJS=1THENGOSUB140
350 IFJS=0THENJS=1:GOSUB140:GOTO340EL
SEA=VPEEK(P+31):IFA<ST ORA>BPTHENJS=1
ELSEPUTSPRITE0,,,W:GOTO420
360 A=VPEEK(P+31):IFA<BMORA>BPTHENPLA
Y"o6b":XP=XP-8:YP=YP+8:PUTSPRITE0,(XP
+4,YP-4):P=P+31:GOSUB80:PUTSPRITE0,(X
P,YP):A=VPEEK(P+32):IFA>=ST ANDA<=BPT
HENPUTSPRITE0,,,W:GOTO420ELSESELEPUTS
PRITE0,,,W:GOTO420
370 IFJS=1THENJS=0:GOSUB140:GOTO360EL
SEPUTSPRITE0,,,W:GOTO420
380 W=11:A=VPEEK(P-63):B=VPEEK(P-31):
IFA<BMORA>BPTHENIFB>=BMANDB<=BPTHEN32
0ELSEPLAY"v8o6b":XP=XP+8:YP=YP-8:PUTS
PRITE0,(XP-4,YP+4),,10:P=P-31:GOSUB80
:PUTSPRITE0,(XP,YP):A=VPEEK(P+32):IFA
>=ST ANDA<=BPTHENPUTSPRITE0,,,W:GOTO4
20ELSEIFJS=1THENGOSUB140
390 IFJS=0THENJS=1:GOSUB140:GOTO380EL
SEA=VPEEK(P+33):IFA<ST ORA>BPTHENJS=1
ELSEPUTSPRITE0,,,W:GOTO420
400 A=VPEEK(P+33):IFA<BMORA>BPTHENPLA

```

248  
72  
153  
254  
78  
194  
128  
141  
145  
126  
85  
35  
221  
21  
54  
143  
170  
11

```

Y"o6b":XP=XP+8:YP=YP+8:PUTSPRITE0,(XP
-4,YP-4):P=P+33:GOSUB80:PUTSPRITE0,(X
P,YP):A=VPEEK(P+32):IFA>=ST ANDA<=BPT
HENPUTSPRITE0,,,W:GOTO420ELSESELEPUTS
PRITE0,,,W:GOTO420
410 IFJS=1THENJS=0:GOSUB140:GOTO400EL
SEA420
420 IFDMTHEN430ELSEIFDTHENGOSUB800ELS
EGOSUB140:A=VPEEK(P+32):IFA<ST ORA>12
0THEN270ELSEIFSTRIG(J)THENJS=0:ONR(ST
ICK(J))+1GOTO320,340,380ELSEONR(STICK
(J))GOTO280,300:PUTSPRITE0,,,W:GOTO42
0
430 IFDTHENGOSUB800ELSEGOSUB140:A=VPE
EK(P+32):IFA<ST ORA>120THEN270ELSEDB=
ASC(MID$(D$(DS),DT,1)):DT=DT+1:IFDB>7
2THENJS=0:ONR(DB-73)+1GOTO320,340,380
ELSEONR(DB-64)GOTO280,300:PUTSPRITE0,
,,,W:GOSUB140:GOTO420
440 SPRITEOFF:FORX=0TO3:PUTSPRITE0,(0
,208):NEXT:IFXP<27THENXP=211:P=P+24:V
X=VX-1ELSEIFXP>211THENXP=27:P=P-24:VX
=VX+1ELSEIFYP<39THENYP=159:P=P+512:VY
=VY-1ELSEYP=39:P=P-544:VY=VY+1
450 VX=VX+(VX>5)*5-5*(VX=0):VY=VY+(VY
>4)*4-4*(VY=0):VD=VX+VY*5-5:PS=P:XS=X
P:YS=YP
460 V=V(VD):ONVGOSUB1670,1750,1830,19
10,1990,2070,2150,2230,2310,2390,2470
,2550,2630,2710,2790,2870,2950,3030,3
110,3190
470 P=PS:XP=XS:YP=YS:PUTSPRITE0,(XP,Y
P),7,W:ONSPRITEGOSUB800:SPRITEON:RETU
RN420
480 FORX=0TO3:PUTSPRITE0,(0,209):NEXT
:X=6:Y=8:SOUND1,3:SOUND3,5:SOUND6,25:
SOUND7,246:SOUND12,6:SOUND9,16:SOUND8
,16
490 FORZ=1TO205:A=VAL(MID$("444422223
3311477762111133322225544777762111133
2275132222731111333668436643368888888
8888888888882222233311114477732272727
1711173222233388513851337333322444162
257372688888888888888888888888888888
3333333333",Z,1))
500 IFSTRIG(J)THEN550
510 IFA=4THENS=1:A=A-4ELSESE=0
520 IFA=1THENY=Y-1ELSEIFA=2THENY=Y+1E
LSEIFA=3THENX=X+1ELSEIFA=4THENX=X-1
530 IFSTHENNEXTSELOCATEX,Y:SOUND13,
3:FORT=1TO20:NEXT:PRINTCHR$(126):NEXT
540 FORX=1TO3000:IFNOTSTRIG(J)THENNEX
T
550 FORX=0TO3:PUTSPRITE0,(0,209):NEXT
:SOUND8,0:SOUND9,0:SOUND7,248
560 CLS:IFSC>HITHENHI=SC
570 PRINT"zzzzzzzzzzzzzzzzzzzzzzzzzzz
mvzvmvzvnmmmvzvmvzvvzvvzvqvzvqvstu
rvzvrvzzzzzzvrvzvqvzvqvzvqvzvqvzzzzz
vqnmopvzvrmmmqvzvqvzzzzzzvqsturvzvrst
srvzvqvzzzzzzvqvzvqvzvqvzvqvzvqvzzzzz
zvqvzvrvzvqvzvqvzvrmmomvzzsvzvtzvsv
zsvzvsvtsusvz";
580 PRINT"zzzzzzzzzzzzzzzzzzzzzzzzzz";
590 S0$=STRING$(28,108):S1$=STRING$(3

```

47  
95  
10  
1  
156  
161  
138  
91  
224  
48  
196  
127  
169  
89  
11  
67  
128  
248  
61

```

1,108):S2$=STRING$(6,108):SP$=STRING$
(19,32)
600 PRINTS0$SP$S2$ HI-Score: "USING"
#####;HI:;PRINT " S2$SP$S1$SP$S2$"
[F1] "B$(J)S2$SP$S2$" [F2] INSTRUCTI
ES "S2$SP$S2$" [F3] START SPEL "S2
$SP$S0$;
610 ONKEYGOSUB640,650,730:KEY(1)ON:KE
Y(2)ON:KEY(3)ON
620 IFSTRIG(J)THEN620
630 FORX=1TO10000:KEYSTRIG(J)THEN730EL
SENEXT:KEY(1)OFF:KEY(2)OFF:KEY(3)OFF:
DM=1:LV=0:DT=1:DS=DS+1AND3:GOTO750
640 J=-J+1:LOCATE9,16:PRINTB$(J);:RET
URN
650 CLS:LOCATE5,0:PRINT"## INSTRUCTIE
S ##"
660 PRINT:PRINT"DOEL: VERZAMEL ZO
SNEL MOGELIJK DE 20 GOUD S
TAVEN, IN ELK VELD LIGT ER EEN.
670 PRINT:PRINT"GOUD: z{z":PRINT:PRIN
T"DOOD: zaz zbz zcz
680 PRINT:PRINT"BEIDIENING: JOYSTICK(P
RTI) TOETSEN (CRSR)
690 PRINT"[F1] U VERLIEST EEN LEVEN
EN START OPNIEUW IN HET HUIDI
GE VELD.
700 PRINT:PRINT"[F2] U VERLIEST EEN L
EVEN EN START OPNIEUW IN HET
STARTVELD.
710 LOCATE5,23:PRINT"Druk SPATIEBALK"
;
720 IFSTRIG(0)THENRETURN560ELSE720
730 KEY(1)OFF:KEY(2)OFF:KEY(3)OFF
740 DM=0:LV=5
750 ONKEYGOSUB790,780:KEY(1)ON:KEY(2)
ON:OB=0:SC=0:VY=2:VX=3:XP=139:YP=95:P
=6578:GOSUB110:GOSUB130:W=5:S=0:PLAY"
L64", "L64", "L64":FORX=1TO20:O(X)=1:NE
XT
760 CLS:PRINT"SCORE: 0 LEVENS:
5":PRINT:PRINT"GOUDzzzzzzzzzzzzzzzzzz
zz":IFDMTHENLOCATE6,4:PRINT"DEMONSTRA
TIE"
770 TIME=0:GOSUB450
780 PS=6578:XS=139:YS=95:VX=3:VY=2:VD
=8
790 D=1:RETURN
800 SPRITEOFF:PUTSPRITE0,,9:LV=LV-1
810 IFPLAY(1)ORPLAY(2)THEN810
820 PLAY"03L2EDCL64", "02L2EDCL64"
830 IFPLAY(1)ORPLAY(2)THEN830
840 FORX=1TO900:NEXT:D=0:IFLV=>0THENL
OCATE22,0:PRINTLV:GOTO460ELSEIFDMTHEN
SC=0:GOTO550ELSE480
850 CLEAR400:DEFINTA-Z:COLOR15,4,4:SC
REEN1,2:SCREEN2,2
860 DIMR(8),V(20),O(20),B$(1),D$(3):R
ESTORE860:FORX=0TO8:READR(X):NEXT:DAT
A 0,0,2,2,2,0,1,1,1
870 D$(0)="GGGGGGGGGGGGGGGGGGPGGICCCG
GGPGGIIGGIIIPGILGIIIPGGGGIIIIIPLGP
G@@@@GGPGGGGGGGIICCCGGC@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@CCGGG@
@@@@@ILIPILIPICLGGGGGGGGCCCL"

```

```

880 D$(1)="GGGGGGGGGGGGGGGGGGPGIIIIII
CCCCLIIIIIIIGPPGGGGGGGGCCCLCCGGGPGG
GGPCCLLLCIIICC"
890 D$(2)="GGGGGGGGGGIIIIIGGCCCCCLIP
GGGG@@@@G@LCCCCCCCCCIIIIIIIIICCCCC
CCCCPGGGGPGGGGGGGGG"
900 FORX=1TO20:READV(X):NEXT
910 D$(3)="CIIIIIIIIICCCCLIIIIIGGGGGGG
GGGGGGGGGGGGIICCCCCIIIIIIIGGGIIIIII
GGGGGGGGGGGPGPGGGGGGGGGCCCGGGGGGG"
920 DATA 15, 8, 7,16,17
930 DATA 14, 2, 1,13,12
940 DATA 19, 5, 3, 6,11
950 DATA 20,18, 4, 9,10
960 J=1:B$(0)="TOETSENBORD " :B$(1)="
JOYSTICK "
970 VDP(1)=VDP(1)AND191
980 GOSUB1630
990 GOSUB1280
1000 GOSUB1050
1010 POKE &HFCAF,1
1020 VPOKE8192+2047,255:VPOKE8192+204
8+2047,255:VPOKE8192+4096+2047,255:CL
S
1030 WIDTH25:VDP(1)=VDP(1)OR64:GOTO56
0
1040 RETURN
1050 RESTORE1060:FORX=0TO21:A$="":REA
DB$:FORY=1TO63STEP2:A$=A$+CHR$(VAL("&
h"+MID$(B$,Y,2))):NEXT:SPRITE$(X)=A$:
NEXT:RETURN
1060 DATA 070003070300020007070002020
00003C060A0A0A00060E0C0A060D0D0D0D0C0
1070 DATA 07000307030002191F060101060
6061CC060A0A0A000C0D0A868E0C0EC7E3200
1080 DATA 070003070300030007070003030
00003C060A0A0A00060E0C0A000F0F010D0C0
1090 DATA 070003070300031A1D050203070
6061CC060A0A0A0006070B8F860806C7E3200
1100 DATA 070003070300031B1B034F7F300
00000C060A0A0A000707E0EE0877F39000000
1110 DATA 070003070300020201030300000
00003C060A0A0A0006060E0C020C0C0C0C0C0
1120 DATA 03060505050006070305060B0B0
80B03E000C0E0C0004000E0E00040400000C0
1130 DATA 030605050500030B15160703377
E4C00E000C0E0C0004098F860808060606038
1140 DATA 03060505050006070305000F0F0
80B03E000C0E0C000C000E0E000C0C00000C0
1150 DATA 030605050500060E1D1F0601367
E4C00E000C0E0C000C058B8A040C0E0606038
1160 DATA 0306050505000E7E7007E1FE9C0
0000E000C0E0C000C0D8D8C0F2FE0C000000
1170 DATA 0306050505000607030403030
30303E000C0E0C000404080C0C0000000C0
1180 DATA 00010000000000041FF410101010
1030180C08080808082FF820000000008000
1190 DATA 000028102804020101020428102
8000000001408142040808040201408140000
1200 DATA 0000000000000000304780C67D
7B460000000000000000C0201E30E6ED2B06
1210 DATA 00000000000000007780C0767B
4D06000000000000000E01E30E0E62B0D06
1220 DATA 20100C3FFF3F0C102000000000
00000040830FCFFC3008040000000000000

```

```

1230 DATA 01020C3FFF3F0C02010000000000 18
00000804030FCFFFC30408000000000000000
1240 DATA 3C7EFFFFFFF7E3C0000000000000 41
000000000000000000000000000000000000
1250 DATA 3C7EE7C3C3E77E3C000000000000 144
000000000000000000000000000000000000
1260 DATA 78CCB4FCB4B430303030B4B4FCB 75
4CC780000000000000000000000000000000
1270 DATA 000078CCB4FCB4B4B4B4FCB4CC7 22
800000000000000000000000000000000000
1280 RESTORE 1330:FORX=0T029:READA$:FO
RY=0T07:A=VAL("&h"+MID$(A$,Y*4+1,2)):
B=VAL("&H"+MID$(A$,Y*4+3,2)):P=X*8+Y+
776 243
1290 VPOKEP,A:VPOKEP+8192,B 238
1300 VPOKEP+2048,A:VPOKEP+10240,B 242
1310 VPOKEP+4096,A:VPOKEP+12288,B 226
1320 NEXTY,X:RETURN 98
1330 DATA 0011447144714471EE71EE71FFc
cFFcc 0
1340 DATA FFccFFccEE71EE7144714471447
10011 251
1350 DATA 003100310031CC31CC310031003
10031 24
1360 DATA FF51FF510051005100510051005
10051 221
1370 DATA FFa1C3a1C3a1C3a1FFa1C3a1C3a
1C3a1 14
1380 DATA 7F223F31263126313F317F22001
10011 133
1390 DATA FF22FF3166316631FF31FF22001
1FFff 189
1400 DATA FF22FF3166316631FF31FF22001
10011 171
1410 DATA FE22FC3164316431FC31FE22001
10011 166
1420 DATA 00113Cd13Cd13Cd13Cd13Cd13Cd
13Cd1 3
1430 DATA 3Cd13Cd13Cd13Cd13Cd13Cd13Cd
10011 10
1440 DATA 008e778e778e778e008eEE8eEE8
eEE8e 233
1450 DATA FFfffffFB6eFB6eF06e166eC76
eF76e 224
1460 DATA FFfffff9F6eDE6eC66e106e7B6
e3B6e 168
1470 DATA FFfffff7D6e396e9B6eC36eD86
eDB6e 124
1480 DATA E06e0F6e6F6e626e786e1C6eD96
eC36e 61
1490 DATA 016e746e706e3A6e8B6eE36eEF6
e8E6e 229
1500 DATA 836e306eE76e6F6e686e036e776
e736e 61
1510 DATA E06eCF6e0F6eE86eE36eF36eFFf
fffff 174
1520 DATA 0C6e606e726e636e0F6eBE6eFFf
fffff 80
1530 DATA 436e166e706e676e6F6e0F6eFFf
fffff 160
1540 DATA FFffffffffffFFFFFFfFFFFf
fffff 255
1550 DATA 00113Cd13Cd13Cd13Cd13Cd13Cd
13Cd1 14
1560 DATA 3Cd13Cd13Cd13Cd13Cd13Cd13Cd

```

```

10011 21
1570 DATA 00a17Ea100a17Ea100a17Ea100a
1FFff 116
1580 DATA 001100110011001100110011001
10011 227
1590 DATA 00110011001100117Eb17Eb17Eb
10011 68
1600 DATA 0011001100110371037104e118e
17E41 124
1610 DATA 00110011001103d103d104e118e
17E41 253
1620 DATA 00413c417e417e417e417e413c4
10041 145
1630 OPEN"GRP:"AS1:FORX=0T01:FORY=0T0
2 53
1640 Z=Y*64+8:DRAW"BM=X;=Z;":PRINT#1
,"!"CHR$(34)"#%&'()*+,-./0123456789
:;<=>?" 70
1650 Z=Z+8:DRAW"BM=X;=Z;":PRINT#1,"@
ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_" 124
1660 NEXTY,X:RETURN 116
1670 LOCATE0,5 149
1680 PRINT"le111111111111111111111111zzz1 l
ezzzzzzzzzzzzzzzzzzzzz1 lezzzzzzzzzz
zzzzzzzzzz1 l1111111111111111111111le1
wzzzzzzzzzzzzzzzzzzzzzz1 xzzzzzzzzzz
zzzzzzzzzz1 lfhhhhhhhhghghiddfgize1
lzzzzzzzzzyzzddzyzz1 lzjzzzzzz1l
111zdd111ze1 118
1690 PRINT"lfhizzzz11111zdd111ze1 l
zzzzzzzzzzzzddzzzzzz1 lzfzgzizzzzzz
zzddzzzzzz1 lzzzezzzzzzzzzzddzzzzzz1
lzzzezzzzzzzzzzddzzzzzz1 lzzzeffhhhhh
hgg11111zzz1 zzzzezzzzzzzz1111111zzz1
zzzezzzzzzzz1111111aaa1 1111111111
z1111111111 35
1700 IFO(1)THENVPOKE6382,123 135
1710 SX=3:X1=48:Y1=159:A1=32:B1=120:S
1=2:D1=12:PUTSPRITE1,(X1,Y1),15,D1 27
1720 X2=44:Y2=139:A2=71:B2=139:S2=3:D
2=12:PUTSPRITE2,(X2,Y2),15,D2 55
1730 X3=88:Y3=71:A3=71:B3=139:S3=2:D3
=12:PUTSPRITE3,(X3,Y3),15,D3 184
1740 RETURN 149
1750 LOCATE0,5 145
1760 PRINT"sttuttssuutstusvzzzzle1 z
zzzzzzzzzzzzzzzzzzle1 zzzzzzzzzzzz
zzkzzzzle1 vdvnmnoomnmnmnovdzzle1
zvvpqrrrqprrrqqvzzzzzew zzvprqrrppq
rpqvzzzzzex zzvprqrrtsutuustvfghhgil
zvppqvzzzzzzzzzzzyzzz1 zzvqrrvzzzz
zzzzzzzyzzz1 99
1770 PRINT"zvvpqrvzzzzzzzz|zyzzz1 z
zvpprvdfgghhh1111111111111111111111
zzz11111111111111111111111111111111
zvvpqvz1111111111111111111111111111
zvvpqvz1111111111111111111111111111
izfefghhgiz1 vdvqrvz1111111111111111
zvvrpvz1111111111111111111111111111
zzz1111111111 133
1780 IFO(2)THENVPOKE6618,123 218
1790 SX=2:X1=152:Y1=159:A1=152:B1=208
:S1=2:D1=14:PUTSPRITE1,(X1,Y1),15,D1 133
1800 X2=88:Y2=103:A2=88:B2=200:S2=3:D
2=14:PUTSPRITE2,(X2,Y2),15,D2 169
1810 X3=40:Y3=47:A3=47:B3=167:S3=6:D3

```







```

2730 PRINT"pvyzdvzfizvzzzvrzvvzzzzzzz r
vzdvzzfizzv}rvvzz}zzzz qvzdvvvvmmnn
oopvzvzvzzv uvzdzvzvqprstatvzvszvv
zzzdzvzqvstvzzzzzzzzzz zzzzdvzqvz
wzzzzzzzzzz mvzzfggipv|xzfgghhgzv
rvaazyzvpmnvzlylzylaav pmmnoonnoq
qmmonnommonm
2740 IFO(14)THENVPOKE6837,123
2750 SX=3:Y1=206
2760 X2=200:Y2=99:A2=71:B2=99:S2=1:D2
=16:PUTSPRITE2,(X2,Y2),15,D2
2770 X3=200:Y3=111:A3=111:B3=159:S3=4
:D3=16:PUTSPRITE3,(X3,Y3),15,D3
2780 RETURN
2790 LOCATE0,5
2800 PRINT"vzvzvzvzvzvzvzvzvzvzv v
zvlzzzzzzzzzzzzzzzzzzzzz vzzlzzzzzzzz
zzzzzzzzzzz vzzlzfhghgizfhghgizlzzv
vzzlzzzyzzzzzzzyzzzzzzz vzzlzzd11111
1111111zzzv vzzzzzdbbzbzbzbbbbbzbzv
vzzzzdzvzzzzzzzzzzzzzzzzz vzzzzdzaz
azazzzzzzzzz
2810 PRINT"zfhghhghghghghghghzizzzz v
z}zzzz11lyzzzzzzzzzzzzz v11zzzw11z
zzzz1zzvmn vzzzzzxzzz1zzzzzzlzzvpp
vzzzzfilz|zezezezzzzvrr vzelcc11z1
e1e1l1zzvpq vzezzzzzzzezezezzzzvqr
vzezzzzzzzezezezzlaavpp mnomnmnovm
evnmoomnmmpq
2820 IFO(15)THENVPOKE6455,123
2830 SX=1:X1=128:Y1=135:A1=128:B1=160
:S1=2:D1=14:PUTSPRITE1,(X1,Y1),15,D1
2840 X2=128:Y2=127:A2=128:B2=168:S2=3
:D2=18:PUTSPRITE2,(X2,Y2),15,D2
2850 X3=64:Y3=67:A3=64:B3=176:S3=6:D3
=16:PUTSPRITE3,(X3,Y3),15,D3
2860 RETURN
2870 LOCATE0,5
2880 PRINT"11111111111111111111 1
zbbzbzbbzzzzzzzzzzzzz lzzzzzzzzjzzz
zzzzzzzzzzz lzazzzazzkzzfhghgghhhil
lz11111111fizzzzyyzzzzz lzzzzzzzzzzz
zzzyyyzzz11 lzzzzzzzzzzzdl11111111
11111111zzzdzlbbzbbz1 lzzzzzzzzz
dzllezzzzzzzl
2890 PRINT"lzzzzzzzzlcclezvzzzzzzz l
zfhhghghilaaalezzaazzz lzzzyyzzz11
1le1111zzz lfhghghgizlzzjzezzzzld1
lzzzyyzzz1z|kzezz}z|zal lzfhhghghilz
fgiefgizlzl lzzzyyzzzzzyzeyzwwz1
lzzzl1zzzzazldedlxazzz1 1111111111
111ze111111
2900 IFO(16)THENVPOKE6618,123
2910 SX=3:X1=120:Y1=143:A1=120:B1=18
4:S1=2:D1=18:PUTSPRITE1,(X1,Y1),15,D1
2920 X2=88:Y2=103:A2=103:B2=155:S2=3:
D2=16:PUTSPRITE2,(X2,Y2),15,D2
2930 X3=56:Y3=155:A3=103:B3=155:S3=4:
D3=16:PUTSPRITE3,(X3,Y3),15,D3
2940 RETURN
2950 LOCATE0,5
2960 PRINT"111111111111vpprqvzvpqr 1
zzzzzzzzzzzstutvzevst lzzzzzzzzzzz
jwzzzezzzzz le111111zzzzzk}xzzzezzz
lebzbzblzzezvnmoommnnvzv lezzzzlzzez

```

```

vsssttuusvzv lzzazazlzzzzzzzzzzzzzzz
11111lelzezzzzzzzzzzz vzzzzzelzfh
hhhhhhhhhieyv
2970 PRINT"lzzzzelzzzzzzzzzzzzzezv l
zzzzelzzzzzzzzzzzzzezv lefhhil11111
zzzzddzvm lezzzzzzzzzzzzzzddzvvrr
lezzzzzzzzzzzzzzddzvvrrq lfizzzfhhhi
dzzzzddzvp lzzzzzzzzzzzzzzddzvvrrp
lzzzzzzzzzzzzzzzzddzvvrr lez11ddddd
ddzzz1lzevrrq
2980 IFO(17)THENVPOKE6501,123
2990 SX=3:X1=72:Y1=171:A1=72:B1=120:
S1=4:D1=16:PUTSPRITE1,(X1,Y1),15,D1
3000 X2=136:Y2=87:A2=87:B2=127:S2=6:D
2=20:PUTSPRITE2,(X2,Y2),15,D2
3010 X3=162:Y3=127:A3=87:B3=127:S3=5:
D3=20:PUTSPRITE3,(X3,Y3),15,D3
3020 RETURN
3030 LOCATE0,5
3040 PRINT"qqrstuussttuuttsssvzlll r
rvvzzzzzzzzzzzzzzzzl111l prvzzzzzzzzz
zzzkz1111 rqqvdfhghgizle1z1z1111l
pqrvdzlyzyzzle1z1ld1111l qrvdz1lzfgh
ile1z1lalle1 stuvdz1lzyzzle1z111111l
zzzzdz1l1fghgizle1z1lzzzzel zzzzdz1lzy
zzle1z1zzzzel
3050 PRINT"11111111zfghizle1z1z1}z1 1
11111111zyzzzele11111z1 zzzzzzfhhghg
hhizlezzzzz zzzzzzzzzzyzyzzzzlezzzzx
1111111111111111111111111111111111
zbbzbzbbzzzzz1 zzzzzzzzzzzzzzzzzzzz1
zzazazazazazazaz|aaa|1 11111111111
1111111111111111
3060 IFO(18)THENVPOKE6423,123
3070 SX=3:X1=200:Y1=159:A1=40:B1=200
:S1=4:D1=14:PUTSPRITE1,(X1,Y1),15,D1
3080 X2=105:Y2=127:A2=47:B2=127:S2=3:
D2=20:PUTSPRITE2,(X2,Y2),15,D2
3090 X3=121:Y3=47:A3=47:B3=127:S3=5:D
3=20:PUTSPRITE3,(X3,Y3),15,D3
3100 RETURN
3110 LOCATE0,5
3120 PRINT"sssttuusstuusstuustussu v
zzzzzzzzzzzzzzzzzzzzzzz vzzzzzzzzzzz
zzzzzzzzzzz vddl1fhghghilzzzzfhhizzzv
vddlzzzyzzz1lzzzzzffizzv vddlz11111z1
l1zzzzzzzfv vddlz11111z1l1zzzzfizzzzv
vddlzzzzzzzzzzzzzzzzfizzv vddlzzzzzzz
zzzzzzzzzzzfv
3130 PRINT"vddlz1e1l1z111zzzzfizzzzz v
ddlzlezzzzzzzzzzzzzzzzzfv vddlzezzzzz
zzzzzzzzzfv vddlz1zfghghilaaaaaaaa1evz
vddlzzzyzzzyl1bbb111111111111111111
lzzzzzzzevz vddlzzzzzzz11lzzzzzzzzezj
vz|laazzzz111a}azazazek mnomnvzvmm
nommonnommonm
3140 IFO(19)THENVPOKE6821,123
3150 SX=2:X1=168:Y1=55:A1=168:B1=200
:S1=2:D1=12:PUTSPRITE1,(X1,Y1),15,D1
3160 X2=200:Y2=99:A2=168:B2=200:S2=1:
D2=12:PUTSPRITE2,(X2,Y2),15,D2
3170 X3=96:Y3=119:A3=119:B3=159:S3=2:
D3=20:PUTSPRITE3,(X3,Y3),15,D3
3180 RETURN
3190 LOCATE0,5

```

```

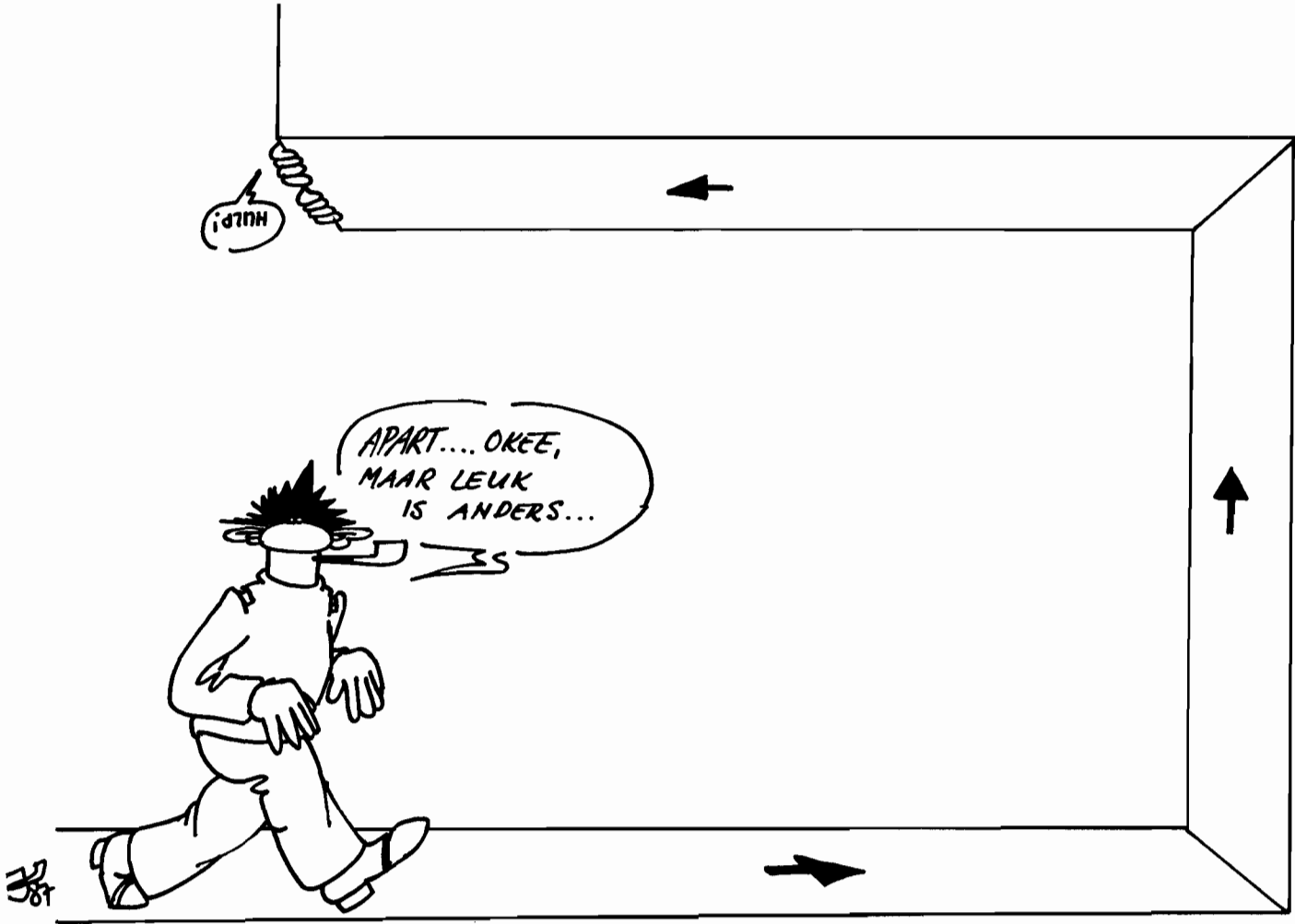
3200 PRINT"vvvvvzzzvvvvvvvvvvvvvvvv v
zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz
zkzzzzzzzzv zzzzzfhfizzzee1111111zv
vzzzz|zzzzzee|zzzzzzzzv zzzzzdzzzzz
ee|zzzzzzzzv zzzzzd}zzzee|zvvvvvvvv
vzzzzzd|zzzee|zzzzzzzz vzzzzzzdzz
zee|zzzzzzzz 55
3210 PRINT"vzzzz}dzzzee111111111 v
zzzz|dzzzzzee111111111 vzzzzdzzzzze
e|zzzzzzzz vzzzzd}zzzee|zzzzzzzz
vzzzzzd|zzzee|ze|11dd| vzzzzzzdzz
ee|ze|11zz| vzzzz}dzzzee|zezzzzz
vzzzz|dzzzzzee|zezzzzzz vzzvvvvvvvv
vvvvvvvvvvzz| 94
3220 IF0(20)THENVPOKE6442,123 200
3230 SX=1:X1=40:Y1=83:A1=40:B1=112:S
1=6:D1=12:PUTSPRITE1,(X1,Y1),15,D1 91
3240 X2=112:Y2=111:A2=40:B2=112:S2=5:
D2=12:PUTSPRITE2,(X2,Y2),15,D2 108
3250 X3=40:Y3=139:A3=40:B3=112:S3=4:D
3=12:PUTSPRITE3,(X3,Y3),15,D3 64
3260 RETURN 147
3270 PLAY"04CDEFGAB05CDEFGAB","05CDEF
GAB06CDEFGAB","06CDEFGAB07CDEFGAB" 185
3280 IFPLAY(1)ORPLAY(2)ORPLAY(3)THEN3
280 48

```

```

3290 SC=SC+2000*(LV+1):FORX=0TO3:PUTS
PRITEX,(0,208):NEXT:LOCATE0,5:FORX=1T
018:PRINT" ":N 244
EXT:LOCATE3,12:PRINT"U HEBT HET GEHAA
LD":LOCATE6,14:PRINT"SCORE:"SC 4
3300 FORX=1TO5000:IFSTRIG(J)THEN480EL
SENEXT:GOTO560

```



## Kategorie: utility's

## MSXBUG

Steven van Loef, de maker van onder andere Jake in the Caves en het prachtige spel Space Walk, heeft het hem weer eens gelapt. Zijn nieuwste produkt heet MSXBUG, een volwaardige debugger voor MSX, helemaal geschreven in machinetaal. Dit prachtige hulpprogramma neemt maar 5 (ja, u leest het goed, vijf) bytes van het Basic-geheugen in beslag! MSXBUG kan daarom steeds 'in het geheugen' blijven en is met een druk op funktietoets 1 meteen op te starten - op MSX computers met een geheugen van 48K of meer, tenminste.

MSXBUG kan op verschillende manieren gebruikt worden. Maar listing 1 moet hoe dan ook ingetypt worden - tenzij u natuurlijk de cassette/diskette, behorende bij dit nummer van MSX Computer Magazine, bestelt. Listing 1, het Basic-programma MSXBUG, bevat de Data voor de ML, zo'n honderd regels in getal. De ML kunt u eventueel apart opslaan op disk of cassette, om later MSXBUG sneller te kunnen laden, maar daar komen we zo meteen op terug.

## Veelzijdig en vriendelijk

MSXBUG is een hulpprogramma, waarmee het volledige MSX-geheugen kan worden bewerkt. Dat wil zeggen: het is mogelijk afzonderlijke bytes in het geheugen te veranderen, naar bytes te zoeken, een stuk geheugen met een bepaalde waarde te vullen of een stuk geheugen te verplaatsen. Verder kan een ML-programma gestart worden door naar het beginadres daarvan toe te springen.

Maar het mooiste is misschien nog wel, dat MSXBUG met de hele MSX-slotstructuur gebruikt kan werken: elke pagina kan in elk slot en subslot geschakeld worden. Dit betekent dat ieder gedeelte van het geheugen, dus ook stukken die

normaal vanuit Basic niet te bereiken zijn, bewerkt kan worden.

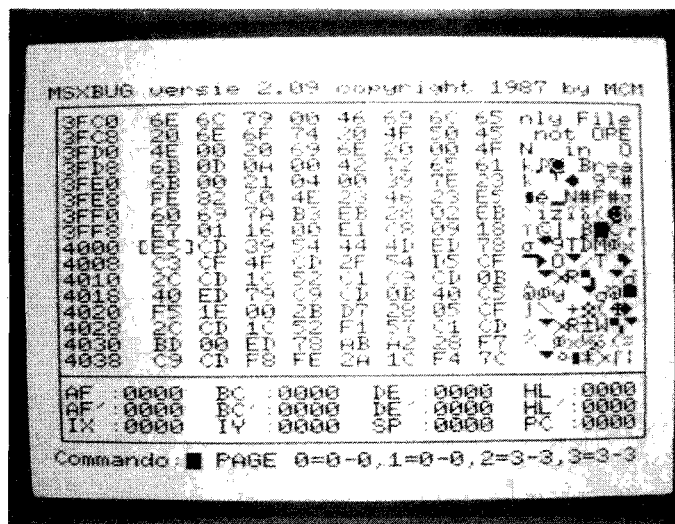
MSXBUG is niet alleen veelzijdig, maar ook nog erg makkelijk te gebruiken. Het ideale MSX hulpprogramma dus!

## Het scherm

Als MSXBUG aangeroepen wordt, verschijnt er een 40 kolommen breed scherm, waarop in het midden een stuk geheugen wordt getoond. De inhoud van dit geheugen wordt weergegeven als hexadecimale getallen, bovendien komt de geheugeninhoud aan de rechterkant van het scherm ook nog als letters en tekens. Hierbij worden alle MSX-tekens gebruikt. Om de inhoud van de middelste geheugenplaats op het scherm staan twee vierkante haken, om aan te geven dat dit het 'huidige adres' is. Het onderste gedeelte van het beeld wordt gevuld met een hexadecimale weergave van de inhoud van alle Z80-registers.

Ook de alternatieve set ontbreekt niet, zie de schermfoto. Op deze manier wordt zoveel mogelijk informatie op een duidelijke, overzichtelijke manier getoond.

Met de cursor-toetsen kan het 'huidige adres' veranderd wor-



den. De weergave van het geheugen op het beeldscherm wordt aangepast, zodat de nieuwe plaats weer in het midden van het scherm staat. Het is ook mogelijk 128 bytes voor- of achteruit te springen, met de - en = - toetsen.

Op de onderste regel staat de instelling van de slots: er staan vier pagina-nummers, elk met een primair en sekundair slotnummer.

Deze instelling is met een MSXBUG-kommando eenvoudig te wijzigen. Ideaal voor MSX2 programmeurs!

## Kommando's

Op de onderste regel staat verder 'Commando:'. Hierachter staat de cursor, klaar voor een kommando. Elk kommando bestaat uit één enkele letter.

Na deze letter verschijnt er vaak een vraag om nog meer informatie, zoals een beginadres, of een byte om naar te zoeken. Het is altijd mogelijk om met de Escape-toets terug te keren naar het hoofdmenu - vooral handig na een per ongeluk ingetypte letter.

Een apart kommando, H van Help, laat een tweede scherm zien met daarop de mogelijkheden. Elke willekeurige toets laat het hoofdscherm weer zien. Deze opzet maakt het programma eenvoudig te bedienen: ingewikkelde kommando's onthouden is er niet bij. De kommando-letters zijn bovendien logisch gekozen: de Z is bijvoorbeeld voor zoeken. Dit zijn de mogelijkheden:

A - adres wijzigen. Hiermee wordt bepaald, welk stuk geheugen er op het hoofdscherm zichtbaar gemaakt wordt. Na het indrukken van de A verschijnt er 'Start (ESC is terug):'.

MSXBUG accepteert alleen adressen van 4 cijfers of letters, hexadecimaal. Ongeldige letters worden gewoon genegeerd en verschijnen niet eens op het scherm. Verkeerde invoer is daardoor niet mogelijk.

C - cursor adres. Dit kommando heeft ongeveer hetzelfde effect als A, maar het beginadres wordt afgelezen op de plaats van de cursor. Staat deze op adres 2300 en is de inhoud van de geheugenplaatsen 2300 en 2301 respectievelijk 20 en 70 hex, dan wordt het geheugen vanaf adres 7020 op het scherm gezet.

Dit komt goed van pas bij sprongadressen en dergelijke, die bij de Z80 immers in 'verkeerde' volgorde staan. MSXBUG houdt daar dus rekening mee.

H - is voor Help.

J - Jump relatief. Dit kommando lijkt weer op C, maar werkt 'relatief', dat wil zeggen ten opzichte van het huidige cursoradres. Dit komt weer overeen met de relatieve adresseerwijze van de Z80. ML-programmeurs zullen deze toevoeging zeker waarderen!

M - keert terug naar MSX-Basic.

O - staat voor Opvullen. Hiermee kan een stuk geheugen ge-

vuld worden met een bepaalde waarde. Wordt als eindadres een lagere waarde opgegeven dan het beginadres, dan merkt MSXBUG dat keurig en voert de opdracht niet uit.

R - registers. Dit kommando maakt het mogelijk de registers van waarde te veranderen. De cursor springt naar het volgende register, wanneer er op de Return-toets gedrukt wordt. Elk register kan eenvoudig veranderd worden door de nieuwe waarde in te typen. ESC keert terug naar het hoofdmenu.

S - stelt de slots in. Voor elke pagina, vier in getal, wordt een twee-cijferig getal gevraagd. Het eerste geeft aan in welk primair slot de pagina gezet moet worden, het tweede stelt het sekundaire slot in. Ook deze opdracht kan halverwege worden afgebroken met Esc.

U - is voor Uitvoeren. MSXBUG springt naar een op te geven adres toe en voert de ML die op dat adres staat uit. Dit kommando leidt maar al te makkelijk tot een vastloper, dus het moet met enig beleid gebruikt worden. Als het beginadres niet wordt opgegeven - de vraag 'Start?' met een Return beantwoorden - dan wordt het beginadres gelijk aan het huidige cursor-adres.

Op het eind-adres wordt door MSXBUG een 'breakpoint' geplaatst: als dat adres bereikt wordt, breekt de ML af en MSXBUG is weer actief. Dit breakpoint mag ook worden weggelaten; in dat geval voert MSXBUG de ML uit, totaan de eerstvolgende RET-instructie. De registers hebben bij terugkeer in MSXBUG nog

de waarde die ze bij het eindigen van de ML hadden, zodat het U-kommando uitstekend geschikt is als test-opdracht.

V - verplaatst een stuk geheugen. MSXBUG vraagt om een beginadres, een eindadres en een adres waarheen het geheugen verplaatst dient worden.

W - wijzigen. Na het W-kommando verschijnt er op de onderste regel 'ESC is terug' en de waarde van de huidige geheugenplaats kan veranderd worden.

Met de Return-toets kan naar het volgende adres gesprongen worden, zodat er een hele serie bytes in een keer kan worden veranderd of ingevoerd. Het is ook mogelijk een byte over te slaan: de waarde wordt dan niet veranderd. Ook hier worden ongeldige letters uiteraard niet geaccepteerd.

Z - Zoeken is de laatste opdracht. Eerst vraagt MSXBUG om een beginadres en een eindadres, vervolgens om een serie bytes. Het einde van de zoek-reeks wordt aangegeven met een lege invoer: meteen op Return drukken dus. MSXBUG zoekt dan flitsend snel het aangegeven stuk geheugen door naar de zoek-reeks.

### MSXBUG gebruiken

Er zijn verschillende manieren om MSXBUG op te starten. De meest eenvoudige maakt alleen gebruik van listing 1. Dit programma, MSXBUG, zet de ML in het geheugen, verplaatst die vervolgens naar een plaats waar MSX-Basic er geen last van heeft en definieert funktietoets 1 zo, dat

een druk op die knop MSXBUG opstart.

Op zich is dit programma dus voldoende om MSXBUG te kunnen gebruiken. Alleen is het jammer om steeds een halve minuut te moeten wachten, totdat de Data-regels zijn gelezen.

Om de inlaad-tijd te verkorten, kan programma 2, RUNBUG, gebruikt worden. Dit programma laadt de ML van cassette of disk in, verplaatst deze en MSXBUG is gebruiksklaar. Dit programma wordt dan voortaan gebruikt om MSXBUG te laden. Een veel snellere methode, vooral voor disk-bezitters.

### De snelle methode

Om dit te bereiken, moet in ieder geval listing 1 worden inge-

instellen. Daarna probeert MSXBUG de ML te verplaatsen. Als dat gelukt is, verschijnt de melding 'Machinecode is ingeladen'. Save nu nog eens voor de veiligheid, alvorens op F1 te drukken om MSXBUG uit te proberen.

### ML wegschrijven

Als alles gelukt is en MSXBUG veilig op cassette of disk staat, typ dan

```
NEW
en daarna
CLEAR 200,&HBF00
```

Het Basic-programma is nu verwijderd, maar de ML staat nog in het geheugen. Geef vervolgens BSAVE"MSX-BIN", &HBF00,&HCA9B

10	REM RUNBUG - listing 2	0
20	REM	0
30	REM MSX Computer Magazine	0
40	REM	0
50	CLEAR 200,(PEEK(&HFC4A)+256*PEEK(&HFC4B))-5:' maak 5 bytes ruimte	80
60	B=PEEK(&HFC4A)+256*PEEK(&HFC4B):' adres van vrij geheugen	11
70	BLOAD "msxbin",R:' Wordt ingelezen op adres BF00H	160
80	DEFUSR=B:' Startadres voor aanroep	79
90	POKE B+0,&HF7:' RST #38	227
100	POKE B+1,PEEK(&HBF9D):' SLOTNUMMER	94
110	POKE B+2,&H0:' BEGIN ADRE	250
S		175
120	POKE B+3,&H40:' VAN MSXBUG	222
130	POKE B+4,&HC9:' RET	0
140	KEY 1,"A=USR(0)" +CHR\$(13)	7
150	CLS	217
160	PRINT "Startadres voor aanroep is :&H"HEX\$(B)	142
170	PRINT	87
180	PRINT "Druk op [F1] om MSXBUG te starten."	188
190	END	

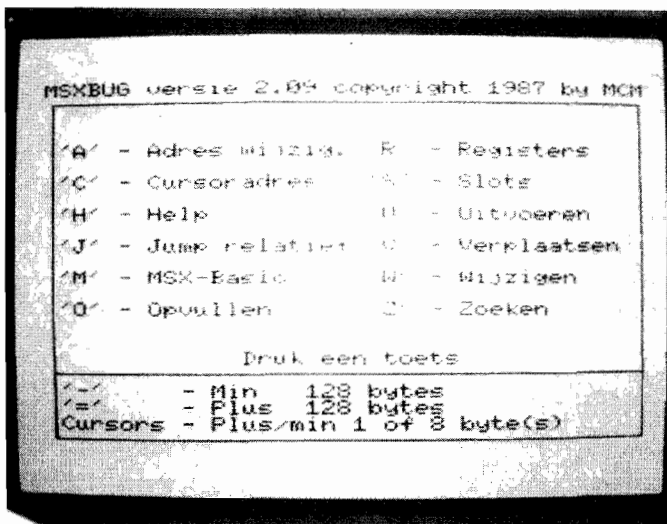
typt. Vergeet niet regelmatig te save, zeker vlak voor de eerste test, want het is tenslotte ML. Vanwege de lengte van de Data-regels adviseren we dringend ICP4 te gebruiken. MSXBUG heeft weliswaar een eigen controle ingebouwd en weigert de ML op te starten als die controle niet klopt, maar die waarschuwing geeft niet aan in welke regel de fout zit.

Als het programma de ML goed gelezen heeft en in het geheugen geplaatst heeft, dan kunt u uw voorkeurskleuren

De ML wordt nu opgeslagen op cassette of disk onder de naam MSXBIN. Het tweede programma, RUNBUG, laadt deze ML later weer in. Dit alles moet zo direct mogelijk achter het runnen van MSXBUG gebeuren. Daarmee bedoelen we, dat er geen ander programma tussendoor mag worden ingeladen en uitgevoerd.

### RUNBUG

Om te begrijpen wat RUNBUG doet, moeten we even onderzoeken hoe MSXBUG



zijn ML opslaat. Steven van Loef heeft het programma zo geschreven, dat het zichzelf installeert in het geheugen van 4000 tot 4A9B hex, zodat Basic er geen last van heeft. Dit is de reden dat MSXBUG alleen op computers met meer dan 32K RAM werkt.

Als MSXBUG het benodigde geheugen niet kan vinden, springt het programma er bij regel 580 uit. Verschijnt er bij het uitvoeren dus 'Break in 580', dan heeft uw computer niet genoeg geheugen.

De ML wordt opgeslagen vanaf adres C000, tot aan adres CA9B hex. Op adres BF00 komt bovendien een klein programmaatje, dat het geheugen doorzoekt naar een vrij stuk geheugen in pagina 1, van 4000 tot 7FFF hex dus. Als dat geheugen gevonden is, wordt de ML 8000 (hex) bytes naar beneden verplaatst. De oorspronkelijke ML op adres C000 is nu niet meer nodig.

Tenslotte installeert MSXBUG nog een ML-routine van vijf bytes, vlak onder de CLEAR-grens, die de verplaatste ML aanroept - zie regel 590 tot 630. Deze vijf bytes zijn de enige, die MSXBUG van het vrije Basic-geheugen afsnoept!

Door nu het geheugen van BF00 tot CA9B weg te schrijven, wordt de nog niet verplaatste ML op disk of cassette gezet. Maar omdat de verplaats-routine op adres BF00 staat, kan de ML worden ingeladen en verplaatst met een enkele opdracht:

**BLOAD"MSXBIN",R**

Dit is precies wat het tweede programma, RUNBUG, doet. Duidelijk een veel snellere methode.

Voortaan is het dus voldoende, om RUNBUG te laden en uit te voeren. Cassette-eigenaars moeten er alleen nog even om denken, dat RUNBUG op zijn beurt weer MSXBIN inlaadt. Het is daarom handig om dit bestand vlak achter RUNBUG op cassette te zetten.

Disk-bezitters hebben daar natuurlijk geen last van.

**Tenslotte**

We zijn razend enthousiast over deze debugger. Steven heeft er een keurig stukje werk mee afgeleverd, we zullen MSXBUG als standaard gaan hanteren in - bijvoorbeeld - de Z80 cursus.

Voor alle duidelijkheid, we hebben geen letter dan het programma veranderd. Zulks in tegenstelling van wat onze gewoonte is, we zijn echt niet te beroerd om een programma naar onze eigen smaak te wijzigen.

Een voorbeeld van de nette manier waarop MSXBUG werkt is het volgende detail.

Als MSX gestart wordt, dan kiest het programma zijn eigen schermkleuren, maar als we het weer verlaten, dan worden de oorspronkelijke instellingen weer hersteld. Werkelijk keurig! Petje af!

```

10 REM MSXBUG
20 REM
30 REM MSX Computer Magazine
40 REM
50 REM Written by:
60 REM
70 REM S.V.L. Software 1987
80 REM
90 ' PAS TOP OF BASIC MEMORY AAN ***
100 CLEAR 200,(PEEK(&HFC4A)+256*PEEK(
&HFC4B))-5
110 B=PEEK(&HFC4A)+256*PEEK(&HFC4B)
120 DEFUSR=B:KEY 1,"A=USR(0)"+CHR$(13)
)
130 ' PRINT INTRODUKTIESCHERM *****
140 W=PEEK(&HF3AE)
150 SCREEN 0:WIDTH 37:CLS
160 LOCATE 9,8:PRINT "MSXBUG versie

```

```

2.09"
170 LOCATE 8,10:PRINT "S.V.L. Softwar
e 1987"
180 LOCATE 7,12:PRINT "Machinecode in
stalleren"
190 LOCATE 9,14:PRINT "Even geduld a
.u.b."
200 ' INSTALLEER MACHINEKODE *****
210 RESTORE 760
220 FOR I=&HC000 TO &HCA9B
230 READ A$
240 D=VAL("&H"+A$)
250 CS=CS+D
260 POKE I,D
270 NEXT I
280 IF CS<>275259! THEN RG=760:GOTO 7
00
290 RESTORE 1620:CS=0
300 FOR I=&HBF00 TO &HBF9D
310 READ A$
320 D=VAL("&H"+A$)
330 CS=CS+D
340 POKE I,D
350 NEXT I
360 IF CS<>18962 THEN RG=1630:GOTO 70
0
370 ' KLEUR VERANDEREN *****
380 CLS
390 PRINT "Voorgrondkleur in MSXBUG i
s nu 1"
400 PRINT "Achtergrondkleur in MSXBUG
is nu 3"
410 PRINT:PRINT "Wilt u dit wijzigen
(j/n)?";
420 A$=INPUT$(1)
430 IF A$="N" OR A$="n" THEN 560
440 IF A$<>"J" AND A$<>"j" THEN BEEP:
GOTO 410
450 PRINT A$
460 LOCATE 0,4:PRINT CHR$(27);"J";
470 LINE INPUT "Nieuwe voorgrondkleur
:";VK$
480 VK=VAL(VK$):IF VK<1 OR VK>15 THEN
BEEP:GOTO 460
490 LOCATE 0,5:PRINT CHR$(27);"J";
500 LINE INPUT "Nieuwe achtergronkle
ur:";AK$
510 AK=VAL(AK$):IF AK<1 OR AK>15 OR A
K=VK THEN BEEP:GOTO 490
520 POKE &HC07B,VK
530 POKE &HC081,AK
540 ' ZOEK RAMSLOT IN PAGE 1 EN
550 ' VERPLAATS MSXBUG NAAR &H4000 **
560 DEFUSRI=&HBF00
570 A=USR1(0):SL=PEEK(&HBF9D)
580 IF SL=&HFF THEN STOP
590 POKE B+0,&HF7:' RST #30
600 POKE B+1,SL:' RAMSLOT
610 POKE B+2,&H0:' ADRES WAAR
620 POKE B+3,&H40:' MSXBUG BEGINT
630 POKE B+4,&HC9:' RET
640 WIDTH W:CLS
650 PRINT "Machinecode is ingeladen."
660 PRINT
670 PRINT "U kunt MSXBUG starten met
[F1] of met"

```

```

680 PRINT "DEFUSR=&H";HEX$(B);":A=USR
(0). " 204
690 NEW 61
700 WIDTH W:CLS 53
710 PRINT USING "U heeft een fout gem
aakt in de data vanaf regel:###";RG 220
720 PRINT 138
730 PRINT "Eerst verbeteren!" 187
740 STOP 239
750 ' MSXBUG versie 2.09 ***** 0
760 DATA ED,73,4F,40,31,FF,BF,CD,53,4
0,CD,6D,44,CD,C9,40,CD,98,41,3A,34,48
,FE,01,20,F0,11,44,48,1A,32,AE 0
770 DATA F3,13,1A,D5,CD,5F,00,D1,13,1
A,FE,FF,D5,CC,CF,00,D1,13,1A,21,E9,F3
,77,13,23,1A,77,D5,CD,62,00,D1 161
780 DATA 13,1A,32,AB,FC,EE,FF,CD,32,0
1,AF,32,34,48,31,00,00,FB,C9,11,44,48
,3A,AE,F3,12,13,3F,28,32,AE,F3 221
790 DATA 3A,AF,FC,12,13,D5,CD,6C,00,D
1,3A,DE,F3,12,13,D5,CD,CC,00,D1,21,E9
,F3,7E,12,13,36,01,23,7E,12,13 131
800 DATA 36,03,D5,CD,62,00,D1,AF,CD,3
2,01,3A,AB,FC,12,3E,FF,32,AB,FC,CD,38
,01,E6,C0,07,07,32,42,48,4F,06 216
810 DATA 00,21,C1,FC,09,7E,FE,00,28,1
5,21,C5,FC,09,7E,E6,C0,CB,3F,CB,3F,CB
,3F,CB,3F,C6,80,81,32,42,48,3A 82
820 DATA 42,48,32,3F,48,32,40,48,C9,2
A,DC,F3,E5,CD,97,44,21,01,01,CD,C6,00
,11,4A,48,CD,73,44,21,51,00,22 45
830 DATA 2A,48,2A,06,48,01,40,00,A7,E
D,42,06,10,C5,E5,CD,CD,46,CD,8D,44,CD
,8D,44,06,08,CD,2F,47,CD,D2,46 88
840 DATA CD,8D,44,23,10,F4,E1,06,08,C
D,7D,44,CD,8D,44,CD,8D,44,C1,10,D8,21
,F9,02,22,2A,48,21,08,48,11,2D 193
850 DATA 49,06,0C,C5,06,04,1A,13,CD,E
E,44,10,F9,C1,E5,7E,4F,23,66,69,CD,CD
,46,78,FE,01,C4,8D,44,C4,8D,44 161
860 DATA E1,23,23,10,DE,21,0B,07,CD,C
6,00,3E,5B,CD,A2,00,26,0A,CD,C6,00,3E
,5D,CD,A2,00,21,18,0C,CD,C6,00 158
870 DATA 11,10,49,CD,73,44,21,18,13,0
6,04,11,3D,48,C5,CD,C6,00,1A,E6,03,C6
,30,CD,A2,00,3E,2D,CD,A2,00,1A 171
880 DATA CB,3F,CB,3F,E6,03,C6,30,CD,A
2,00,13,01,00,06,09,C1,10,DB,E1,22,DC
,F3,C9,21,18,01,CD,C6,00,06,0A 11
890 DATA CD,03,45,11,80,48,CD,73,44,3
E,01,CD,0E,45,3A,38,48,47,21,C7,41,7E
,FE,FF,C8,B8,28,05,23,23,23,18 37
900 DATA F4,23,7E,23,66,6F,E9,43,F8,4
1,56,09,42,57,33,42,4F,60,42,55,8C,42
,48,FF,42,41,35,43,4A,3E,43,5A 170
910 DATA 5D,43,4D,67,44,52,CC,43,53,0
D,44,00,00,00,00,00,00,00,00,00,00,00
,00,FF,2A,06,48,CD,2F,47,4F,23 192
920 DATA CD,2F,47,69,67,22,06,48,C9,C
D,ED,46,D8,CD,FC,46,D8,A7,ED,52,D8,C8
,44,4D,03,CD,0B,47,D8,EB,CD,2F 253
930 DATA 47,D5,E5,62,6B,5F,CD,47,47,E
1,D1,0B,13,23,78,B1,20,EC,C9,CD,FB,44
,11,73,48,CD,73,44,21,0B,08,CD 171
940 DATA C6,00,CD,8A,46,D8,2A,06,48,3
A,38,48,FE,8D,28,07,3A,35,48,5F,CD,47

```

```

,47,23,22,06,48,CD,C9,40,18,D3 104
950 DATA CD,ED,46,D8,CD,FC,46,D8,E5,D
5,A7,ED,52,44,4D,E1,D1,D8,C8,03,CD,1C
,47,D8,3A,38,48,FE,8D,C8,3A,35 144
960 DATA 48,5F,CD,47,47,23,0B,78,B1,2
0,F7,C9,CD,ED,46,D8,ED,53,26,48,CD,FC
,46,D8,3A,38,48,FE,8D,28,0E,CD 47
970 DATA 2F,47,32,37,48,22,22,48,1E,C
9,CD,47,47,22,1E,48,ED,73,24,48,F3,31
,08,48,F1,C1,D1,E1,D9,08,F1,C1 80
980 DATA D1,E1,D9,08,DD,E1,FD,E1,ED,7
B,24,48,CD,5F,47,ED,73,1C,48,ED,73,24
,48,31,1C,48,FD,E5,DD,E5,D9,08 163
990 DATA E5,D5,C5,F5,D9,08,E5,D5,C5,F
5,ED,7B,24,48,3A,38,48,FE,8D,C8,3A,37
,48,5F,2A,22,48,CD,47,47,C9,CD 244
1000 DATA 6D,44,CD,97,44,21,01,01,CD,
C6,00,11,4A,48,CD,73,44,21,03,02,11,5
D,49,06,14,CD,C6,00,CD,73,44,2C 102
1010 DATA 10,F7,21,12,0D,CD,C6,00,11,
DF,48,CD,73,44,CD,C6,00,CD,9F,00,C9,C
D,ED,46,D8,ED,53,06,48,C9,2A,06 189
1020 DATA 48,CD,2F,47,FE,7F,30,09,06,
00,4F,0C,09,22,06,48,C9,06,00,EE,FF,4
F,A7,ED,42,22,06,48,C9,CD,ED,46 124
1030 DATA D8,CD,FC,46,D8,E7,D8,C8,22,
28,48,21,2C,48,06,08,C5,D5,E5,CD,FB,4
4,11,B6,48,CD,73,44,E1,D1,C1,CD 226
1040 DATA 8A,46,D8,3A,38,48,FE,8D,28,
07,3A,35,48,77,23,10,DF,3E,08,90,F5,B
7,28,2E,F1,47,F5,21,2C,48,7E,4F 169
1050 DATA EB,CD,2F,47,EB,13,23,B9,28,
0E,D5,E5,EB,ED,5B,28,48,E7,E1,D1,28,1
0,18,E0,10,E4,EB,F1,F5,16,00,5F 182
1060 DATA A7,ED,52,28,01,EB,ED,53,06,
48,F1,C9,CD,FB,44,11,73,48,CD,73,44,2
1,14,06,CD,C6,00,21,08,48,06,0A 186
1070 DATA E5,C5,CD,45,46,C1,3A,38,48,
FE,8D,28,0E,FE,FF,20,02,E1,C9,EB,E1,7
3,23,72,23,18,09,E1,23,23,11,96 99
1080 DATA 4A,CD,73,44,11,94,4A,CD,73,
44,10,D4,C9,CD,FB,44,11,EF,48,CD,73,4
4,06,00,21,3D,48,E5,C5,21,18,06 86
1090 DATA CD,C6,00,78,C6,30,CD,A2,00,
21,18,10,CD,C6,00,06,02,CD,03,45,CD,8
A,46,C1,E1,D8,FE,8D,28,21,C5,F5 80
1100 DATA E6,30,CB,3F,CB,3F,CB,3F,CB,
3F,4F,06,00,F1,E6,03,CB,27,CB,27,81,E
5,21,C1,FC,09,4E,E1,81,77,C1,23 38
1110 DATA 04,78,FE,04,20,B5,C9,3E,01,
32,34,48,C9,3E,0C,CD,A2,00,C9,1A,13,F
E,24,C8,CD,A2,00,18,F6,CD,2F,47 13
1120 DATA 23,FE,00,20,02,3E,20,CD,EE,
44,10,F1,C9,E5,2A,2A,48,23,22,2A,48,E
1,C9,21,28,00,3E,18,CD,4D,00,23 139
1130 DATA CD,E3,44,3E,19,CD,4D,00,23,
06,10,CD,D1,44,3E,14,CD,4D,00,23,CD,E
3,44,3E,13,CD,4D,00,23,06,03,CD 248
1140 DATA D1,44,3E,1A,CD,4D,00,23,CD,
E3,44,3E,1B,CD,4D,00,C9,3E,16,CD,4D,0
0,11,27,00,19,3E,16,CD,4D,00,23 44
1150 DATA 10,EF,C9,06,26,3E,17,CD,4D,
00,23,10,F8,C9,E5,2A,2A,48,CD,4D,00,2
3,22,2A,48,E1,C9,21,18,01,CD,C6 56
1160 DATA 00,06,27,3E,20,CD,A2,00,10,
F9,CD,C6,00,C9,32,36,48,0E,00,21,38,4

```



```

130 0
2450 DATA130,194,162,146,138,134,130,
0,124,130,130,130,130,130,124,0,252,1
30,130,252
2460 DATA128,128,128,0,124,130,130,13
0,138,132,122,0,252,130,130,252,144,1
36,132,0
2470 DATA124,130,128,124,2,130,124,0,
254,146,16,16,16,16,16,0,130,130,130,
130
2480 DATA130,130,124,0,130,130,130,13
0,68,40,16,0,130,130,130,146,170,198,
130,0
2490 DATA198,68,40,16,40,68,198,0,198
68,40,16,16,16,16,0,254,132,8,16
2500 DATA32,66,254,0,112,64,64,64,64,
64,112,0,0,0,128,64,32,16,8,0
2510 DATA32,66,254,0,112,64,64,64,64,
64,112,0,0
2520 PUTSPRITE0,(0,0),0,0:PUTSPRITE1,
(0,0),0,0:PUTSPRITE2,(0,0),0,0:CLS
2530 A$=" PUCKY
(C) PIRASOFT 1985 EEN SPEL VAN R
ENAN JEGOUZO... SPELEN MET BEHULP VA
N: CURSOR-TOETSEN, JOYSTICK 8 VERS
CHILLENDE NIVEAU'S... "
2540 PRINT"aaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaob bb
PIRASOFT PRESENTEERT... bb
bmaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaal":PRINT:PRINT:PRINT
2550 PRINT" naao c c naad c c c
c b b b b b b n l b b
b b b b b b n l b b haal
b b b hi makal b b b
b bmo b b b b b b
mo b b b b b b
b e maal maade e e
2560 PRINT:PRINT:PRINT:PRINT:PRINT"
naaaaaaaaaaaaaaaaaaaao b
RENAN JEGOUZO b maaaaaaaa
aaaaaaaaal":
2570 VPOKE8204,128:VPOKE8205,128
2580 FORI=0TO13:SOUND1,0:NEXT:SOUND7,
&B00101110:SOUND8,13:SOUND6,20:SOUND1
2,100:SOUND9,16:SOUND13,0
2590 ZC=2:WC=ZC:FORI=0TO252STEP6:PUTS
PRITE0,(I,50),10,WC:PUTSPRITE1,(I-30,
50),12+WC,6:WC=WC-(WC=0)*ZC+(ZC=WC)*Z
C:IFSTRIG(0)ORSTRIG(1)THEN2640ELSEFOR
T=1TO100:NEXT:GOSUB2620:NEXT:SOUND13,
0
2600 ZC=4:WC=ZC:FORI=220TO5STEP-6:PUT
SPRITE0,(I,145),10,WC:PUTSPRITE1,(I+3
0,145),9+WC,6:WC=WC-(WC=0)*ZC+(ZC=WC)
*ZC:IFSTRIG(0)ORSTRIG(1)THEN2640ELSEF
ORT=1TO100:NEXT:GOSUB2620:NEXT:SOUND1
3,0
2610 GOSUB2620:GOTO2590
2620 SOUND0,I:ZR=INT(RND(1)*14+2)*16:
VPOKE8200,ZR:VPOKE8201,ZR:VPOKE8202,Z
R:VPOKE8203,ZR
2630 A$=RIGHT$(A$,LEN(A$)-1)+LEFT$(A$
,1):LOCATE0,20:PRINTLEFT$(A$,31):RET
URN
2640 IFSTRIG(1)THENJK=IELSEJK=0
2650 SOUND8,0:ZR=5*16:VPOKE8200,ZR:VP
OKE8201,ZR:VPOKE8202,ZR:VPOKE8203,ZR:
RETURN
2660 X=128:Y=12
2670 CLOSE:OPEN"GRP:"AS1:SCREEN2:COLO
R1,0,0:RESTORE2750:E=1
2680 LINE(0,0)-(79,192),4,BF:LINE(80,
0)-(175,192),15,BF:LINE(176,0)-(256,1

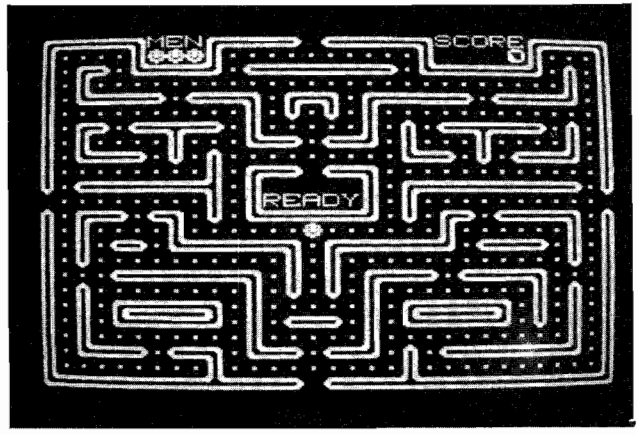
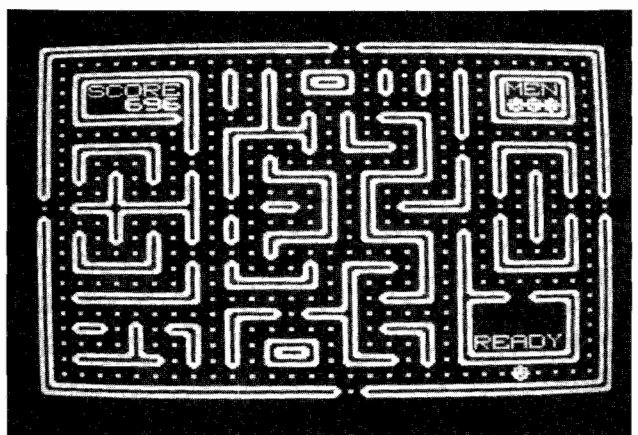
```

168  
31  
111  
58  
115  
186  
111  
84  
166  
2  
46  
126  
217  
85  
229  
117  
145  
182  
174  
19  
244  
171  
52  
207

```

92),8,BF
2690 READA$:IFLEFT$(A$,1)="8"THENE=VA
L(RIGHT$(A$,1)):GOTO2690ELSEREADB$:ON
EGOTO2700,2710,2740
2700 AF=VAL(A$)*2:BF=VAL(B$)*5:GOTO26
90
2710 LINE(AF+X,Y+BF)-(X+VAL(A$)*2,Y+V
AL(B$)*5)
2720 LINE(X-AF,Y+BF)-(X-VAL(A$)*2,Y+V
AL(B$)*5)
2730 AF=VAL(A$)*2:BF=VAL(B$)*5:GOTO26
90
2740 GOTO2760
2750 DATA&1,0,0,82,3,3,7,1,11,2,15,3
,17,4,19,5,23,8,23,11,21,14,21,16,20,
5,17,15,19,14,22,13,24,9,26,7,27,5,27
,5,0,28,8,1,7,10,5,82,14,5,13,15,14,5
,15,15,5,13,16,5,9,16,3,13,5,3,12,5,11
,7,10,5
2760 DATA&1,0,16,82,1,16,3,17,3,19,1
,20,0,20,8,1,13,20,5,82,13,21,5,10,23,1
0,21,13,20,5,8,1,9,21,82,9,23,7,24,7,2
1,9,21,8,1,6,21,82,6,24,4,24,4,21,6,21
2770 DATA&1,3,21,82,3,24,5,24,.5,21,
3,21,83,2,2 :PAINT(X,Y+10)
2780 PRESET(97,1):COLOR8:PRINT#1 "PIR
ASOFT":LINE(95,0)-(160,9),5,B:DRAW"bm
50,160":COLOR12:PRINT#1,"ontwerp en p
roductie":COLOR13:DRAW"bm55,170":PRIN
T#1," van PROGRAMMA'S"
2790 FORI=1TO1100:IFINKEY$<>" "THENRET
URNELSENEXT:RETURN
2800 SCREEN0:PRINT"FOOT IN DATAREGELS
VANAF REGEL";FL:STOP

```





```

00030200030004000
1740 DATA000000000000000000,
000000000000000000000000,
1750 DATA030000080808000,
089008080000000000400,
1760 DATA000000000000000000,
001100000000000000000000,
1770 DATA000000000000000000,
001100000000000000000000,
1780 DATA010002000001000,
0500020000000010200,
1790 DATA000000000000000000,
000000000000000000000000,
1800 DATA060005020104001,
0802030201000040000,
1810 DATA000000000000000000,
000000000000000000000000,
1820 DATA0300070030905,
0005090400701030400,
1830 DATA000000000000000000,
000000000000000000000000,
1840 DATA01020706010703,
0904060007000102000,
1850 DATA0000060400000000,
0000000003070000000000,
1860 DATA0000000010500,
0800050200000000000000,
1870 DATA0030890040000,
00000030980400000000,
1880 DATA117000000000001,
08020000000000000611,
1890 DATA0010070100703,
0004060020600200000,
1900 DATA000000000000000000,
000000000000000000000000,
1910 DATA0000060700609,
0009070060700000000,
1920 DATA0305070300400,
000003004060050400,
1930 DATA0000000000000000,
001000000000000000000000,
1940 DATA01050890098,
08089000980050200,
1950 DATA0000000000000000,
000000000000000000000000,
1960 DATA0000108020180,
09008201080200000000,
1970 DATA060400000030700,
000000000000000000000000,
1980 DATA0000000000000001,
0502000000000000000000,
1990 DATA03080808080804,
00030808008080808040,
2000 DATA0000000000000000,
001100000000000000000000,
2010 DATA0000000000000000,
001100000000000000000000,
2020 DATA0100090900902,
0001090090900000200,
2030 DATA0000000000000000,
000000000000000000000000,
2040 DATA0000000001400,
0000032000000000000000,
2050 DATA06000400000000,
00000000030000700,
2060 DATA000000003203,
0804014000000000000000,
2070 DATA030007000000,
00000000010000400,
2080 DATA00000600500,
01200509400000000000,
2090 DATA010007000000,
00000000030000200,
2100 DATA000000000001,

```

```

46
163
84
22
17
243
178
253
156
31
162
53
204
255
58
109
220
155
146
145
224
171
170
141
212
247
126
239
242
8
147
40
195
138
195
26
157

```

```

0802000000000000000
2110 DATA00000001809403,
000400000000000000,
2120 DATA11700000000000,
00000000000000611,
2130 DATA000000003000,
090008070000000000,
2140 DATA060008200000,
0000060900900400,
2150 DATA00000609020,
1820109400000000000,
2160 DATA0001002000000,
00000000060000200,
2170 DATA060700607060,
40307000140000000,
2180 DATA0003004000000,
0000000006020000,
2190 DATA000000608089,
090980807000000000,
2200 DATA0600900040000,
000000006007000000,
2210 DATA00000060401004,
000300200006040000,
2220 DATA0000010200000,
000000000000000000,
2230 DATA0300804030800,
050000808008000400,
2240 DATA0000000000000,
011000000000000000,
2250 RESTORE2270:CO=0:FORI=264TO735:R
EADA:CO=(CO+A)MOD255:VPOKEI,A:NEXT:IF
CO<>17THENFL=2280:GOTO2800
2260 RETURN
2270 DATA6,14,28,56,48,192,192,0,80,8
0,80,0,0,0,80,80,248,80
2280 DATA248,80,80,0,32,120,160,112,4
0,240,32,0,192,200,16,32,64,152,24,0
2290 DATA64,160,64,168,144,152,96,0,1
6,32,64,0,0,0,16,32,64,64
2300 DATA64,32,16,0,64,32,16,16,32
64,0,32,168,112,32,112,168,32,0
2310 DATA0,32,32,248,32,32,0,0,0,0,
0,0,32,32,64,0,0,120
2320 DATA0,0,0,0,0,96,96,0,0,
0,8,16,32,64,128,0
2330 DATA124,130,130,194,194,226,124,
0,16,16,16,24,24,24,124,130,2,12
2340 DATA48,96,254,0,124,130,2,28,2,1
94,124,0,128,128,200,200,254,8,8,0
2350 DATA254,128,192,252,2,2,252,0,12
4,128,192,252,130,194,124,0,254,132,4
8
2360 DATA8,16,16,0,124,130,194,124,13
0,194,124,0,124,130,194,126,2,2,124,0
2370 DATA0,0,32,0,0,32,0,0,0,32,0,0
32,32,64,24,48,96,192
2380 DATA96,48,24,0,0,248,0,248,0,0
0,192,96,48,24,48,96,192,0
2390 DATA120,204,12,24,48,0,48,48,112
136,8,104,168,168,112,0,124,130,130,
254
2400 DATA130,130,130,0,252,130,130,25
2,130,130,252,0,124,130,128,128,128,1
30,124,0
2410 DATA252,130,130,130,130,130,252,
0,254,130,128,248,128,130,254,0,254,1
30,128,248
2420 DATA128,128,128,0,124,130,128,15
8,130,130,124,0,130,130,130,254,130,1
30,130,0
2430 DATA124,84,16,16,16,84,124,0,126
74,8,8,136,136,112,0,68,72,80,96
2440 DATA80,72,68,0,128,128,128,128,1
28,130,254,0,130,198,170,146,130,130,

```

```

72
33
135
67
48
3
202
205
20
245
200
143
222
217
255
146
253
60
199
129
189
9
73
219
175
165
242
122
242
8
20
41
41

```

1290 RESTORE1770:CO=0:FORI=0T023:FORJ  
=0T030:READT(J,I):CO=CO+T(J,I):NEXT:N  
EXT:IFCO<>687THENFL=1760:GOTO2800

1300 RETURN

1310 VPOKE8204,64:VPOKE8205,64:GOSUB1  
060

1320 PRINT"aaaaaaaaaaaaopnaaaaaaaaa  
aaaaobppppppppppppppppppppppppppppbb  
pnaaopcpndpcppppcpfopcpnaaoppbmaalp  
pppppppppppppppppppppppppppppppppp  
pepephdpppppppppppppppppppppppppppp  
ppppnaaadpp";

1330 PRINT"ppppppppppmdpmaaaa lpf lpbpp  
ppppbhaaaaaadpppppppppppppppppppppp  
ppppppppppnopnadqfaopcpnppppppppnaaak  
dpm lpb bpbphjaaaaopbeppMENbppppppb  
WAIT bpbpbSCOREbpeppb bpnopfjaaaa  
pepp?????bpp";

1340 PRINT"cpmaaa lpmippppppppppppppmaaa  
aalpcbppppppppppmaaaakdpfkaadppppppppbb  
pnaaaaopppppppppppppppppppppppppppp  
epcpcepcepcepnp lpbpppppppppppppppppp  
pppppppppppppppppppppppppppppppppp  
eepnpppppp";

1350 PRINT"bmaaaa lpppppppppppppppppppp  
cpbpbppppppppppnaadpcppfaaopppeppppb  
ppppnaaaipppppppppppppppppppppppppp  
pppppppppppppppppppppppppppppppppp  
pppppppppppppppppppppppppppppppppp  
aaaaaaaaaaaa";

1360 VPOKE6911 108:VX=3:VY=11:SX=23:S  
Y=11:PX=13:PY=10:FI=348:X9=15:Y9=12

1370 RESTORE2010:CO=0:FORI=0T023:FORJ  
=0T030:READT(J,I):CO=CO+T(J,I):NEXT:N  
EXT:IFCO<>657THENFL=2010:GOTO2800

1380 RETURN

1390 DATA0,255,255,0,0,255,255,0,102,  
102,102,102,102,102,102,24,60,126,  
102,102,102,102,102,0,252,254,7,7,25

4,252,0,102,102,102,102,102,126,60,24  
1400 DATA0,63,127,224,224,127,63,0,23  
1,231,195,0,0,195,231,231,103,103,99,  
96,96,99,103,103,230,230,198,6,6,198,  
230,230,231,231,195,0,0,255,255,0

1410 DATA0,255,255,0,0,195,231,231,23  
0,230,198,6,14,254,252,0,103,103,99,9  
6,112,127,63,0,0,63,127,112,96,99,103  
103,0,252,254,14,6,198,230,230,0,0,0  
24,24,0,0,0,0,0,0,0,0,0,24,126,9  
0,255,219,102,126,24

1420 DATA24,126,126,255,255,126,126,2  
4,0,66,66,231,231,126,126,24,24,126,1  
20,224,224,120,126,24,24,126,126,231,  
231,66,66,0,24,126,30,7,7,30,126,24,2  
4,126,90,255,219,102,126,24

1430 DATA126,189,219,255,231,219,255,  
165

1440 DATA240,240,80,80,80,80,128,128,  
160,128

1450 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,  
11,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
1,0,0,2,0,0,0,0,0,1,0,0,0,8,0,0,0,2  
0,0,0,0,0,0,0,0,0,1,0,2,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0

1460 DATA0,0,0,10,8,0,0,9,0,0,8,0,9,0  
0,0,9,0,0,8,0,0,0,9,0,0,4,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,9,0,0,  
8,0,0,2,0,0,0,10,0,0,0,1,0,9,0,0,8,0,  
0,0,9,0,7,0

1470 DATA0,0,0,0,0,0,0,0,0,0,0,0,3,0,  
9,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,10,8,0,2,0,1,0,7,0,0,0,0,0,0,0,0

78  
129  
97  
161  
221  
216  
227  
102  
153  
144  
224  
84  
66  
137  
141  
218  
33

3,0,2,0,1,0,0,4,0,0,0,0,0,0,0,0,0,  
0,0,0,6,0,0,0,8,0,0,0,7,0,0,0,0,0,0,  
0,0,0,0,0

1480 DATA0,6,0,0,0,0,8,0,4,0,0,0,0,  
0,0,0,6,0,0,0,8,0,8,0,0,0,7,0,0,0,  
0,0,0,0,0,0,0,0,0,0,1,0,8,0,2,0,0,0,  
0,0,0,0,0,0,0,0,0,0,11,5,0,0,0,9,0,  
0,2,0,0,0,3,0,0,0,4,0,0,0,1,0,0,9,0,0,  
0,0,0,5,11

1490 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,1,0,8,2,0,3,0,8,0,0,0,9,0,0,0,8,0,  
4,0,1,8,0,0,0,2,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0

1500 DATA0,0,0,6,0,0,8,0,0,0,0,2,0,  
0,0,1,0,0,0,0,9,3,0,0,0,0,7,0,0,0,6,  
0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,2,0,0,0,0,6,0,7,0,0,0,6,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,1,0,0,8,0,0,0,2,  
0,0,0,0,0

1510 DATA0,0,0,0,0,0,0,0,0,0,0,6,0,  
8,0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,9,0,  
0,0,7,0,3,0,9,0,4,0,6,0,9,0,0,0,8,  
0,4,0,0,0

1520 DATA0,0,0,0,0,0,0,0,0,0,0,0,3,  
0,0,0,0,8,0,10,0,3,0,0,0,9,0,0,4,0,  
3,0,0,0,0,0,0,0,4,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,11,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0

1530 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,11,0,0,0,0,0,0,0,0,0,0,0,0,0,  
1540 DATA0,1,0,0,0,0,0,0,9,0,9,0,9,  
0,0,8,9,0,9,0,9,0,9,0,0,0,0,2,0,  
1550 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
1560 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
1570 DATA0,0,0,0,0,0,0,0,0,0,0,6,  
0,9,0,5,0,8,0,7,0,0,0,0,0,0,0,0,0,  
1580 DATA0,0,0,0,0,0,0,0,6,0,4,0,10,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
1590 DATA0,6,0,0,0,0,0,2,0,0,0,0,0,  
0,0,0,3,0,2,0,0,0,6,0,0,0,0,7,0,  
1600 DATA0,0,0,0,0,0,0,0,0,0,0,1,0,2,  
0,0,0,0,0,0,6,0,7,0,0,0,0,0,0,0,  
1610 DATA0,0,0,1,0,2,0,6,0,7,0,0,0,3,  
0,8,9,0,0,4,0,0,0,0,0,1,0,2,0,0,0,  
1620 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
1630 DATA0,6,0,4,0,3,0,4,0,0,0,6,0,0,  
2,0,0,0,1,0,0,4,0,0,0,0,0,0,0,0,  
1640 DATA11,7,0,0,0,0,0,0,6,0,7,0,0,  
0,0,0,0,0,0,0,0,6,0,7,0,6,0,5,11,  
1650 DATA0,6,0,2,0,1,0,2,0,0,0,6,0,0,  
4,0,0,0,3,0,0,9,0,7,0,0,0,0,0,0,  
1660 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
1670 DATA0,0,0,3,0,4,0,6,0,5,0,7,0,1,  
0,9,8,0,2,0,0,0,0,0,3,0,4,0,0,0,  
1680 DATA0,0,0,0,0,0,0,0,0,0,0,3,0,4,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
1690 DATA0,6,0,0,0,0,0,4,0,0,0,0,0,  
0,0,0,1,0,4,0,0,0,3,0,12,0,0,0,7,0,  
1700 DATA0,0,0,0,0,0,0,0,0,6,0,0,9,  
0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,9,  
1710 DATA0,6,0,0,9,0,9,0,0,7,0,0,0,  
0,0,0,6,0,0,0,7,0,1,0,8,0,2,0,0,0,  
1720 DATA0,0,0,0,0,0,0,0,0,0,0,1,0,8,  
0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,  
1730 DATA0,6,0,0,4,0,3,2,0,0,0,0,0,

239  
51  
158  
139  
98  
69  
9  
235  
162  
165  
176  
3  
54  
189  
198  
155  
204  
109  
142  
167  
22  
157  
193  
35  
26  
45

```

T0870ELSEBB=-1:GOTO870ELSE870 242
760 IFBB=0THENBB=SGN(Y-BY):BA=0ELSEBB 72
=0:BA=-1
770 IFBA=BBTHENIFRND(1)<.5THENBB=1:GO 246
T0870ELSEBB=-1:GOTO870ELSE870
780 IFBA=0THENBA=SGN(X-BX):BB=0ELSEBB 102
=-1:BA=0
790 IFBA=BBTHENIFRND(1)<.5THENBA=1:GO 170
T0870ELSEBA=-1:GOTO870ELSE870
800 IFBA=0THENBA=SGN(X-BX):BB=0ELSEBB 33
=1:BA=0
810 IFBA=BBTHENIFRND(1)<.5THENBA=1:GO 155
T0870ELSEBA=-1:GOTO870ELSE870
820 BA=-BA:BB=-BB:GOTO870 233
830 IFBX=0THENBX=30ELSEIFBX=30THENBX=
0ELSEIFBY=0THENBY=23ELSEIFBY=23THENBY
=0 59
840 GOTO870 223
850 IFP2THENBA=SGN(X-BX):BB=0ELSE870 109
860 IFBA=0THENBA=-1:GOTO870ELSE870 20
870 GOTO260 165
880 BEEP 3
890 PLAY"V13S1M6000L6105DFGECDFGDGFC
C" 114
900 PUTSPRITE0,(X*8+8,Y*8-1),10,5 150
910 IFPLAY(0)THEN910 207
920 NI=NI+1:GOTO180 160
930 END 184
940 FORI=0TO13:SOUND1,0:NEXT:SOUND7,&
B00111110:SOUND8,13 146
950 PUTSPRITE1,(0,0),0,0:PUTSPRITE2,(
0,0),0,0 122
960 VI=VI-1:ZC=0:FORI=30TO230STEP50:Z
C=ZC+1:PUTSPRITE0,(X*8+8,Y*8-1),10,ZC
:SOUND0,I:FORT=0TO300:NEXT:NEXT:SOUND
8,0 120
970 IFVI>0THEN190 216
980 LOCATE9,10:PRINT" " :LOC
ATE9,11:PRINT" GAME OVER " :LOCATE9,12
:PRINT" " 92
990 IFINKEYS<>" " THEN990ELSEOP=0:LOCAT
EVX,VY:PRINT" " : 172
1000 OP=OP+1:IFOP=800THEN160 157
1010 IFSTRIG(0)=0ANDSTRIG(1)=0THEN100
0 114
1020 IFSTRIG(0)THENJK=0:GOTO170ELSEJK
=1:GOTO170 83
1030 END 106
1040 W=W+1:IFW>3THENW=0 255
1050 RETURN 138
1060 IFNI>4THENVPOKE8204,0:VPOKE8205,
0:RETURNELSERETURN 81
1070 PUTSPRITE0,(0,0),0,0:PUTSPRITE1,
(0,0),0,0:PUTSPRITE2,(0,0),0,0:CLS:ON
NI-((NI-1)\4)*4GOTO1080,1160,1310,123
0 133
1080 VPOKE8204,128:VPOKE8205,128:GOSU
B1060 64
1090 PRINT"naaaaoMENnaaad faaaaoSCORE
naaaobppppb bpppppppppb?????bppbb
pndpmaaa lpfaaaaaadpmaaaaa lpcbbppppp
ppppppppppppppppppppbbpmaaadpfaop
naopnaaadpfaaa lpb"; 213
1100 PRINT"bpppppppppbpepepbppppppp
ppppbbndpfaadpbbppppbpcpfaakaadpcbb
ppppppppppmadpfa lpbppppppppbbpmaadp
epcbbppppppmadpfaa lpbppppppppbna
dqfaoppppppppppp"; 154
1110 PRINT"epfaaaaaa lpbqqqqqbnnaaaaa
aadpe pppppppppbWAIqbpppppppppp c
pnadp opepmaaaaa lpepndpfaaopcbppppp
bppppppppppppppppppbbpbf dpmaaaaa
opnaaaaa lpfadpbb"; 241
1120 PRINT"bpepppppppppppppppppppp

```

```

pppebbppppfaaaaaopbbpnaadpfaaoppbb
pcppppppppbpepepbppppppppbpcppbbpnaa
aaopbbppppbnaaaaopbbppbbpmaaaa lpbp
fadpbbmaaaa lpepbp"; 227
1130 PRINT"bpbpppppppppppppppppppp
ppbpbpmaadpfaopmadpfa lpcpfaaaaaa lpbp
ppppppppppppppppppppppppppppmaaaaa
aa jaaad faaa jaaaaaaaaa";VPOKE6911,1
08:VX=6:VY=1: SX=21:SY=1:PX=12:PY=11:F
I=348:X9=14:Y9=13 15
1140 RESTORE1450:CO=0:FORI=0TO23:FORJ
=0TO30:READT(J,I):CO=CO+T(J,I):NEXT:N
EXT:IFCO<>642THENFL=1420:GOTO2800 70
1150 RETURN 140
1160 VPOKE8204,112:VPOKE8205,112:GOSU
B1060 224
1170 PRINT"naaaaaaadaadpfaaaaaaa
aaaaobppppppppppppppppppppppppppbb
pnaaaaopcpnaopcpnaaaaopbbpbbSCOR
Epbpbpma lpepepbpMENbpbpbp?????bpepb
pppppppppb bpbpmaaaadbppbpcpfp
aopbbmaaa lpbppppppppbna ja lpbpppbpep
pppppb"; 89
1180 PRINT"bnaaaopepbpppepmadpbbppna
aaopbbpbbppppppppppppppppppppbbpbb
pbcpepcpbpmaoopnaa lpbpbpcpbpeppppbb
pbpepppppppppppppbpepbpepfaaaa ippf
dpbbpfaa lpppppppppppppppppppppppp
pppppcpbpcpbpcpepcpepnaa lmaoopcb
pepbp"; 92
1190 PRINT"bpbpppppppppeppppppppbbpbb
ppbpbpmaaa lpcpcpppcpnadpbbpmaaa lpb
ppppppppmaaa lpbppppppppppppppbfaaaa
a lpppppppppfa lphadqfaopbbppppppppnad
pfa ippppb bpbpfdpcpfopbbppppppf
aopb bpb"; 29
1200 PRINT"bpbppppppbpbnaopbbppbbWAI
T bpbpfaa jdppepma lmadpemaaaaa lpb
ppppppppppppppppppppppppppppmaaaaa
aaaaaaaaadpfaaaaaa";VPOKE6911,1
08:LOCATE0,0:VX=25:VY=4:SY=4: SX=3:PX=
23:PY=20:X9=25:Y9=22:FI=348 48
1210 RESTORE1530:CO=0:FORI=0TO23:FORJ
=0TO30:READT(J,I):CO=CO+T(J,I):NEXT:N
EXT:IFCO<>644THENFL=1510:GOTO2800 220
1220 RETURN 133
1230 VPOKE8204,32:VPOKE8205,32:GOSUB1
060 128
1240 PRINT"naaaaaoMENnaaad faaaaoSCORE
naaaobppppb bppppppppb?????bppbb
pfadpmaaa lpnadpfaopmaaaa lpcbbppppp
pbppppppppppppppppppppbbpfa dpcpbf a l
pfadpmaadpnaaaa lpbpppppppppppppppp
pppppppppp"; 149
1250 PRINT"haaaopbmadpfa dpcpfa lpb
naaa lpbpppbpepppppppppppppppppepbppbb
pcbpbppnadpmaadpfa lpf aoppbpcbbpbbpfa
a lppppppppppppmadpbbpbbppppppppcna
adqfaopcppppppppe phaadpnaa lpb b
pmaoopfaa"; 134
1260 PRINT"cpbppppppppbWAI bppppbb
ppbpcpbpnoepnopmaaaaa lnoepnopbb
pemm lpppm lppppppppm lpppm lpebbppppp
cpppcpnaopcpccppppppbhadp dmaaaa l
pepema lpf dfa lpbpppppppppppppppp
pppppppppp"; 249
1270 PRINT"bpcpnadpfa adpfa adpfa
opcbpbbpbbppppppppppppppppppbbpbb
ppbbpnaopbnadpfa opbnopbbpppfa lpm
a lpepepppppepema lmadpbbpppppppppppp
pc cppppppppppppmaaaaaa lmaa
aaaaaaaa"; 102
1280 VPOKE6911,108:VX=7:VY=1: SX=21:SY
=1:PX=13:PY=12:FI=357:X9=15:Y9=14 44

```

```

10 REM PUCKY
20 REM
30 REM MSX Computer Magazine
40 REM
50 REM door RENAN JEGOUZO
60 REM
70 REM Copyright MicroMSX (Frankrijk)
80 REM
90 GOSUB2660
100 CLEAR500:DEFINTA-Z:SC=0:NI=1:KEYO
FF:VI=3:JK=0:DIMT(31,24),X(8),Y(8):QQ
=RND(-TIME)
110 RESTORE110:FORI=0TO7:READX(I),Y(I)
:NEXT:DATA0,-1,1,0,0,1,-1,0
120 SCREEN1,0,0:COLOR2,0,0:WIDTH31:CLS:LOCATE12,11:PRINT"WACHTEN!!!":GOSUB
2250
130 RESTORE1390:CO=0:FORI=776TO919:RE
ADA:VPOKEI,A:CO=CO+A:NEXT:IFCO<>15696
THENFL=1330:GOTO2800
140 FORI=0TO6:A$="" :FORJ=1TO8:READA:A
$=A$+CHR$(A):NEXT:SPRITE$(I)=A$:NEXT
150 FORI=8198TO8207:READA:VPOKEI,A:NE
XT
160 GOSUB2520
170 SC=0:NI=1:VI=3
180 GOSUB1070
190 X=X9:Y=Y9:PUTSPRITE0,(X*8+8,Y*8-1)
,10,5:ZC=0
200 LOCATESX,SY:PRINTUSING"#####";SC:
RD=1:IFVI=3THENLOCATEVX,VY:PRINT"rrr"
;ELSELOCATEVX,VY:FORI=1TOVI:PRINT"r";
:NEXT:PRINT" ";
210 LOCATEPX,PY:PRINT"WAIT"
220 AX=PX:AY=PY:AA=1:AB=0:BX=PX:BY=PY
:BA=0:BB=-1
230 AD=6145+X+Y*32:FORI=0TO13:SOUNDI,
0:NEXT:SOUND7,8B00111110:SOUND6,5:SOU
ND12,2:SOUND1,2:SOUND11,0:SOUND8,16:P
1=1:P2=1
240 IFINKEY$<>" "THEN240
250 LOCATEPX,PY:PRINT"READY":IFSTRIG(
JK)=0ANDSTICK(JK)=0THEN250ELSELOCATEP
X,PY:PRINT"PUCKY"
260 ST=STICK(JK):ONSTGOTO270,280,290,
300,310,320,330,340:ZC=5:GOTO410
270 Y=Y-1:CC=1:IFY<0THENY=23:GOTO350E
LSE350
280 AD=6145+X+Y*32:IFVPEEK(AD+1)=32OR
VPEEK(AD+1)=112THEN290ELSEIFVPEEK(AD-
32)=32ORVPEEK(AD-32)=112THEN270ELSEZC
=5:GOTO410
290 X=X+1:CC=2:IFX>30THENX=0:GOTO350E
LSE350
300 AD=6145+X+Y*32:IFVPEEK(AD+1)=32OR
VPEEK(AD+1)=112THEN290ELSEIFVPEEK(AD+
32)=32ORVPEEK(AD+32)=112THEN310ELSEZC
=5:GOTO410
310 Y=Y+1:CC=3:IFY>23THENY=0:GOTO350E
LSE350
320 AD=6145+X+Y*32:IFVPEEK(AD+32)=32O
RVPEEK(AD+32)=112THEN310ELSEIFVPEEK(A
D-1)=32ORVPEEK(AD-1)=112THEN330ELSEZC
=5:GOTO410
330 X=X-1:CC=4:IFX<0THENX=30:GOTO350E
LSE350
340 AD=6145+X+Y*32:IFVPEEK(AD-32)=32O
RVPEEK(AD-32)=112THEN270ELSEIFVPEEK(A
D-1)=32ORVPEEK(AD-1)=112THEN330ELSEZC
=5:GOTO410
350 ZC=CC-(ZC=CC)*-CC:AD=6145+X+Y*32:
VP=VPEEK(AD):IFVP=112THENVPOKEAD,32:S
OUND13,5:SC=SC+2:LOCATESX,SY:PRINTUSI

```

```

NG"#####";SC:FI=FI-1:IFFI=0THEN880ELS
E410
360 IFVP<>32THENONCCGOTO370,380,390,4
00ELSE410
370 Y=Y+1:GOTO410
380 X=X-1:GOTO410
390 Y=Y-1:GOTO410
400 X=X+1
410 SN=0:PUTSPRITE0,(X*8+8,Y*8-1),10,
ZC:IF(X=AXANDY=AY)OR(X=BXANDY=BY)THEN
940
420 AX=AX+AA:AY=AY+AB:T=T(AX,AY):PUTS
PRITE1,(AX*8+8,AY*8-1),13,6:IFX=AXAND
Y=AYTHEN940
430 ONTGOTO440,450,460,470,480,510,53
0,550,570,590,600,620:GOTO640
440 IFAA=0THENAA=1:AB=0:GOTO640ELSEAA
=0:AB=1:GOTO640
450 IFAA=0THENAA=-1:AB=0:GOTO640ELSEA
A=0:AB=1:GOTO640
460 IFAA=0THENAA=1:AB=0:GOTO640ELSEAA
=0:AB=-1:GOTO640
470 IFAA=0THENAA=-1:AB=0:GOTO640ELSEA
A=0:AB=-1:GOTO640
480 YY=Y-AY:XX=X-AX:Q=-AA:W=-AB:IFABS
(YY)>=ABS(XX)THENAB=SGN(YY):AA=0ELSEA
B=0:AA=SGN(XX)
490 IFAA=0ANDAB=0THENELSE640
500 AB=X(WW):AA=Y(WW):GOSUB1040:GOTO6
40
510 IFAB=0THENAB=SGN(Y-AY):AA=0
520 IFAA=ABTHENIFRND(1)<.5THENAB=1:GO
TO640ELSEAB=-1:GOTO640ELSE640
530 IFAB=0THENAB=SGN(Y-AY):AA=0
540 IFAA=ABTHENIFRND(1)<.5THENAB=1:GO
TO640ELSEAB=-1:GOTO640ELSE640
550 IFAA=0THENAA=SGN(X-AX):AB=0ELSEAB
=-1:AA=0
560 IFAA=ABTHENIFRND(1)<.5THENAA=1:GO
TO640ELSEAA=-1:GOTO640ELSE640
570 IFAA=0THENAA=SGN(X-AX):AB=0
580 IFAA=ABTHENIFRND(1)<.5THENAA=1:GO
TO640ELSEAA=-1:GOTO640ELSE640
590 AA=-AA:AB=-AB:GOTO640
600 IFAX=0THENAX=30ELSEIFAX=30THENAX=
0ELSEIFAY=0THENAY=23ELSEIFAY=23THENAY
=0
610 GOTO640
620 IFP1THENAA=SGN(X-AX):AB=0:P1=0ELS
E640
630 IFAA=0THENAA=1:GOTO640ELSE640
640 REM
650 BX=BX+BA:BY=BY+BB:T=T(BX,BY):PUTS
PRITE2,(BX*8+8,BY*8-1),9,6:IFX=BXANDY
=BYTHEN940
660 ONTGOTO670,680,690,700,710,740,76
0,780,800,820,830,850:GOTO870
670 IFBA=0THENBA=1:BB=0:GOTO870ELSEBA
=0:BB=1:GOTO870
680 IFBA=0THENBA=-1:BB=0:GOTO870ELSEB
A=0:BB=1:GOTO870
690 IFBA=0THENBA=1:BB=0:GOTO870ELSEBA
=0:BB=-1:GOTO870
700 IFBA=0THENBA=-1:BB=0:GOTO870ELSEB
A=0:BB=-1:GOTO870
710 YY=Y-BY:XX=X-BX:Q=-BA:W=-BB:IFABS
(YY)>=ABS(XX)THENBB=SGN(YY):BA=0ELSEB
B=0:BA=SGN(XX)
720 IFBA=0ANDBB=0THENELSE870
730 BB=X(WW):BA=Y(WW):GOSUB1040:GOTO6
40
740 IFBB=0THENBB=SGN(Y-BY):BA=0ELSEBA
=-1:BB=0
750 IFBA=BBTHENIFRND(1)<.5THENBB=1:GO

```

## Kategorie: spellen

# Pucky

Mogen we u eventjes sterkte wensen? Dat zal u namelijk best kunnen gebruiken, als u de listing van Pucky gaat intikken. Vele regels uiterst kompakte Basic, zonder kommentaar-regels of extra leesbaarheids-spaties. En als extra hinderpaal hebben we een paar 'gemene' controles ingebouwd, waarmee het programma zelf eventjes checkt of u de vele DATA-regels wel goed hebt ingetikt. Is dat niet het geval, dan breekt het af met een verwijzing naar de regel waar u een foutje heeft laten zitten.

Maar, voordat u de moed in de schoenen zakt, al dat noeste tik- en controle-werk is zeer zeker de moeite waard. Pucky is een 'dijk van een spel', een onvervalste Pacman-kloon. De spookjes jagen u onvermoeid na over maar liefst acht speelschermen. Vooral de schermen 5 tot en met 8 zijn daarbij werkelijk vreselijk moeilijk!

### Snel

Wat ons vooral trof aan dit spelprogramma - dat we van onze Franse kollega's van MicroMSX hebben overgenomen - is de prima speelsnelheid. Hoewel het puur Basic is, waarbij de vele mogelijkheden van schermtype 1 uitstekend gebruikt worden, speelt het als een echt machinetaal-spel. De bewegingen zijn snel en vloeiend, vooral die van de achtervolgers.

De programmeur van dit spel, Renan Jegouzo, is een ware tovenaars als het om Basic-spellen gaat. Ook een ander programma van zijn hand, KUUB'ERT, is van grote kwaliteit. Wie dat spel wil kunnen proberen zal echter de MCM-cassette nummer 7 moeten bestellen, het programma is nooit in MSX Computer Magazine afgedrukt. Een tweede mogelijkheid is om KUUB'ERT uit Viditel te halen, alwaar het in het MSX-bestand van Micro-Technology staat.

Overigens, de naam die Renan Jegouzo aan zijn software

geeft is wel opvallend. 'Pira-soft' is toch geen naam voor een programmeur met een onbesproken levenswandel. En die doodskop op de Franse vlag - het intro-beeld van Pucky - geeft ook al te denken.

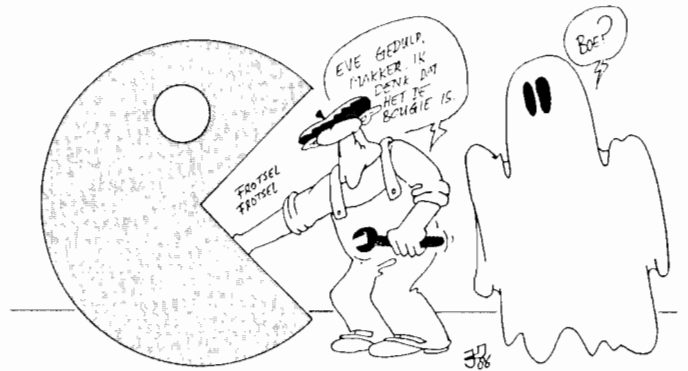
### Het spel

Op zich is Pucky heel simpel. Met het gele happertje moet men alle punten op het speelveld opeten. Jammer genoeg dwalen er echter ook nog een tweetal gemeen uitziende spoken rond in het doolhof en iedere ontmoeting met die tegenstanders is dodelijk. Dat maakt het nog tot een hele opgave om al die pillen te eten, temeer daar Pucky geen 'krachtpillen' kent. In de meeste Pacman-spellen komen die wel voor, na het eten van zo'n krachtpil kan de speler dan gedurende korte tijd de spookjes de baas.

Maar Pucky moet het alleen maar van snelheid en inzicht hebben. Je niet laten insluiten is de boodschap. Daarbij zijn de vier openingen die elk van de doelhoven kent een groot voordeel. Door bijvoorbeeld door het gat aan de bovenzijde weg te lopen verschijnt men automatisch weer aan de onderkant van het veld. Handig om op het laatste moment aan de spoken te kunnen ontsnappen!

### Veel velden

Pucky kent in totaal 8 speelvelden, die een oplopende moeilijkheidsgraad vertonen. De



eerste vier velden zijn nog wel te spelen, althans voor ervaren joystickers, maar de laatste vier velden uitspelen is naar ons idee slechts aan de echte fanaten voorbehouden.

Het programma heeft daarbij namelijk een hele aardige truk in petto; men kan op die tweede serie velden alleen de pillen zien liggen. De muren zijn voor het gemak onzichtbaar!

In feite zijn die velden 5 tot en met 8 precies dezelfde als de velden 1 tot en met 4, maar doordat men het eigenlijke doolhof niet meer kan zien zijn is een en ander veel lastiger. Temeer daar de spookjes natuurlijk geen last van 'nachtblindheid' hebben!

Een tip voor diegenen die alle velden willen uitspelen maar het niet redden met de drie Pucky's, in regel 960 wordt in de variabele VI dat aantal levens bijgehouden. Als u in die regel het kommando VI=VI-1 verwijderd, dan heeft u meteen onbeperkte levens.

### Intikken

Het intikken van Pucky is een heel karwei, het is het grootste programma dat tot nog toe in MSX Computer Magazine gepubliceerd is. Vandaar ook dat we dit keer hebben afgezien van allerlei extra's, zoals commentaar en leesbaarheids-spaties. Aan de ene kant maakt dat het programma weliswaar heel erg compact en lastig leesbaar, maar aan de andere kant zal juist daardoor het intikken weer iets makkelijker zijn. Immers, men kan nu geen spaties vergeten ook, waardoor bijvoorbeeld de checksums niet meer kloppen.

Over checksums gesproken, tik Pucky niet in zonder ICP/4 te gebruiken! Dat is vragen om problemen met een dergelijk lang en gekompliceerd programma.

Maar ongetwijfeld zullen er toch lezers zijn die zo dapper- of moeten we zeggen 'dom' - zijn, dat ze dit programma zullen proberen in te kloppen zonder daarbij gebruik te maken van onze checksums. Speciaal voor diegenen die goede raad graag in de wind slaan hebben we nog een tweede checksum-systeem in het programma ingebouwd. In de praktijk blijkt namelijk keer op keer dat de meeste fouten worden gemaakt in de DATA-regels, en daar is Pucky ruim van voorzien. Ieder telefonisch vragenuurtje opnieuw worden we geconfronteerd met klachten over programma's die met allerlei foutmeldingen afbreken, juist in die regels waar de DATA worden ingelezen en verwerkt. Dat zo'n fout wel eens niet in de door de MSX gemelde regel zou hoeven te steken, maar juist in de DATA-regels aan de andere kant van het programma kan zitten willen sommige opbellers niet eens van ons aannemen.

Vandaar dat we 'booby-traps' in Pucky hebben aangebracht. Iedere keer als het programma grotere reeksen DATA-regels verwerkt, controleert het intern ook nog eens of deze wel goed zijn ingetikt. Als dat niet het geval is stopt het, met een keurige melding welke reeks DATA men nog maar eens goed moet nakijken. Hopelijk scheelt dat in de telefoontjes, zodat we de mensen met echte vragen ook weer eens te woord kunnen staan.

```

260 IF RV THEN 410
270 ' Met regelinspringen *****
280 LN=VAL(A$): ' isoleer regelnummer
290 H2=LEN(STR$(LN))-1: ' bepaal breedte
   te regelnummer
300 AK=INT((LEN(A$)-1)/TK)+1
310 ES=(AK-1)*H2
320 FOR N=1 TO INT((LEN(A$)+ES-1)/TK)+1
330   T$(R,C)="": H1=0
340   IF N>1 THEN T$(R,C)=SPACE$(H2):
   H1=H2
350   T$(R,C)=T$(R,C)+MID$(A$,1+((N-1)*TK)-((N-2)*H1),TK-H1)
360   IF N=INT((LEN(A$)+ES-1)/TK)+1 THEN T$(R,C)=T$(R,C)+SPACE$(TK-LEN(T$(R,C)))
370   GOSUB 510
380 NEXT N
390 GOTO 230
400 ' Zonder regelinspringen *****
410 FOR N=0 TO (LEN(A$)-1)\TK
420   T$(R,C)=MID$(A$,N*TK+1,TK)
430   IF N=(LEN(A$)-1)\TK THEN T$(R,C)=T$(R,C)+SPACE$(TK-LEN(T$(R,C)))
440   GOSUB 510
450 NEXT N
460 GOTO 230
470 ' Einde bestand bereikt *****
480 GOSUB 630: ' LAATSTE PAGINA
490 PRINT: PRINT "KLAAR"
500 END
510 ' Check printarray *****
520 R=R+1
530 IF R<RP THEN RETURN: ' nog regels over
540 IF C<KP-1 THEN C=C+1: R=0: RETURN: ' nog kolommen over
550 GOSUB 630: ' print pagina
560 R=0: C=0
570 FOR N2=0 TO RP-1
580   FOR N3=0 TO KP-1
590     T$(N2,N3)=" "
600   NEXT N3
610 NEXT N2
620 RETURN
630 ' Print een blad *****
640 IF FP>PN THEN 870
650 IF VK THEN 700
660 LOCATE 0,15: PRINT "Zet pagina gerreed, spatie voor vervolg"
670 IS=INKEY$: IF IS="" THEN 670
680 LOCATE 0,15: PRINTSPACE$(40)
690 IF IS<>" " THEN 660
700 LPRINT USING "Bestand: \
  \ Datum: \ \ Tijd: \
  \ Bladzijde: ###";FS,DA$,TI$,PN
710 LPRINT
720 LPRINT KRS
730 LPRINT
740 LPRINT
750 FOR N4=1 TO RP
760   LPRINT SPACE$(LM);
770   FOR N5=0 TO KP-1
780     T1$=T$(N4-1,N5)
790     LPRINT T1$;
800     IF N5<KP-1 THEN LPRINT SPACE$(TK+SK+EP-LEN(T1$));
810   NEXT N5
820 LPRINT
830 NEXT N4
840 FOR N4=1 TO PL-RP-5
850 LPRINT
860 NEXT N4
870 PN=PN+1

```

```

247 0
86 7
93 6
115 7
9 9
251 7
218 6
60 11
59 5
0 6
60 28
218 55
55 6
54 5
0 157
113 113
174 174
0 250
251 8
118 208
103 152
79 38
196 181
213 185
0 110
145 153
149 151
195 193
127 5
143 7
109 166
66 72
172 72
204 204

```

```

880 RETURN
890 ' Start scherm *****
900 CLS
910 PRINT FN PC$("MSX Computer Magazine's afdruk-programma")
920 PRINT
930 PRINT "Ketting-formulieren (j/n):";
940 INPUT I$: IF FN JN(I$) THEN IF FN JA(I$) THEN VK=1 ELSE ELSE GOSUB 1350: GOTO 930
950 PRINT "Aantal kolommen:";
960 INPUT KP
970 PRINT "Kolom-breedte:";
980 INPUT TK
990 PRINT "Linker-kantlijn:";
1000 INPUT LM
1010 PRINT "Aantal spaties tussen kolommen:";
1020 INPUT SK
1030 PRINT LM+KP*TK+(KP-1)*SK;"tekens per regel. Akkoord (j/n):";
1040 INPUT I$: IF FN JN(I$) THEN IF FN NE(I$) THEN FOR N=1 TO 5: GOSUB 1350: NEXT N: GOTO 950 ELSE ELSE LOCATE 0,CSRLIN-1: GOTO 1030
1050 PRINT "Pagina-lengte:";
1060 INPUT PL
1070 PRINT "Regels per pagina:";
1080 INPUT RP
1090 IF PL-RP<11 THEN FOR N=1 TO 3: GOSUB 1350: NEXT N: GOTO 1050
1100 PRINT "Geef pagina-nummer eerste blad:";
1110 INPUT PN
1120 PRINT "Met welke pagina beginnen:";
1130 INPUT FP
1140 IF V2 THEN GET DATE DA$: GET TIME TI$: GOTO 1190
1150 PRINT "Geef datum";
1160 INPUT DA$
1170 PRINT "Geef tijd";
1180 INPUT TI$
1190 PRINT "Geef kopregel";
1200 LINE INPUT KRS
1210 PRINT "Welk bestand:";
1220 INPUT F$
1230 PRINT: PRINT "Alleen voor Basic-programma's:"
1240 PRINT "Regelnummers inspringen (j/n):";
1250 INPUT I$: IF FN JN(I$) THEN IF FN NE(I$) THEN RV=1 ELSE ELSE GOSUB 1350: GOTO 1240
1260 PRINT: PRINT "Alles goed ingevuld:";
1270 INPUT I$: IF FN JN(I$) THEN IF FN NE(I$) THEN GOTO 890 ELSE ELSE GOSUB 1350: GOTO 1270
1280 RETURN
1290 ' Functie-definities *****
1300 DEF FN PC$(X$)=SPACE$( (WI-LEN(X$))/2)+X$
1310 DEF FN JN(X$)=INSTR("JjNn",X$)
1320 DEF FN JA(X$)=INSTR("Jj",X$)
1330 DEF FN NE(X$)=INSTR("Nn",X$)
1340 RETURN
1350 ' Ga een regel terug *****
1360 LOCATE 0,CSRLIN-1
1370 PRINT SPACE$(WI-1)
1380 LOCATE 0,CSRLIN-1
1390 RETURN

```

```

210 0
5 5
203 140
205 205
179 33
251 3
38 164
247 247
217 55
102 102
118 123
44 43
126 178
41 57
43 225
62 90
7 48
53 210
65 211
164 152
64 64
55 55
87 87
174 151
0 247
227 144
219 141
0 211
26 217
156 156

```

Als er eenmaal een ASCII-bestand klaar staat om met behulp van Print af te drukken hoeven we Print alleen nog maar te laden en te runnen.

Print zal ons stap voor stap vragen hoe we de uitvoer precies gedacht hadden.

## Keuzes

Daarbij zijn er heel wat mogelijkheden om uit te kiezen. Sommige van de vragen spreken voor zich, maar andere keuzes behoeven mogelijk enige toelichting. We zullen ze eens op een rijtje zetten.

De vraag 'Ketting-formulieren (j/n)' zal geen verdere verduidelijking nodig hebben, hier wordt bepaald of Print al dan niet zal wachten tussen de pagina's.

'Aantal kolommen', 'Kolombreedte', 'Linker-kantlijn' en 'Aantal spaties tussen kolommen' bepalen tezamen hoe de pagina's eruit zullen gaan zien.

Daarbij bent u eigenlijk helemaal vrij in het invullen van waarden. Wanneer u voor 8 kolommen van ieder 10 tekens breed kiest, dan heeft Print daar geen enkele moeite mee.

Wel berekent Print eventjes snel hoe breed uw definitie is, in tekens gerekend. Op de vraag 'xx tekens per regel' dient u met j of n te antwoorden. Bij printers die op simpele wijze naar smal schrift - 136 tekens per regel - om te schakelen zijn kunt u maximaal profijt trekken van de mogelijkheid om meer dan 80 tekens per regel af te drukken.

De volgende vragen, 'Paginalengte' en 'Regels per pagina', zijn voor zowel ketting-formulieren als losse vellen belangrijk. Hier wordt namelijk bepaald om de hoeveel regels er een nieuwe bladzijde moet komen. Het verschil tussen de lengte en het aantal regels wordt gebruikt om bij ketting-formulieren te bepalen hoeveel regels er moeten worden opgevoerd om over de scheurrand heen te springen.

Onmogelijke waarden worden overigens niet geaccepteerd door Printer, de paginalengte dient minimaal 11 regels langer

te zijn dan het gekozen aantal regels per pagina. Deze extra regels dienen om bijvoorbeeld de pagina-kop af te drukken en een boven- en onder-marge te kunnen handhaven. 'Regels per pagina' slaat namelijk alleen op door Print af te drukken regels uit het bestand.

## Koppen

Op iedere pagina wordt automatisch een keurige kop afgedrukt. Daarin staat ondermeer het pagina-nummer, waarbij we zelf, na de vraag 'Geef pagina-nummer eerste pagina' kunnen kiezen met welk pagina-nummer we willen beginnen. Handig als een groot bestand in meerdere keren wordt afgedrukt!

De vraag 'Met welke pagina beginnen' is op het eerste gezicht misschien wat onduidelijk. Print biedt echter de mogelijkheid om slechts een gedeelte van een bestand af te drukken, bijvoorbeeld nadat de printer eens een keertje is vastgelopen. Stel dat er een bestand van 10 pagina's moest worden afgedrukt maar dat het ketting-papier na de achtste bladzijde op was, dan komt deze optie handig van pas. We kunnen dan namelijk opgeven dat we vanaf pagina 9 pas willen gaan printen, die eerste 8 waren immers al goed afgedrukt. Het programma moet echter wel die eerste 8 bladzijden doorwerken, voordat vel 9 afgedrukt wordt, omdat het anders niet mogelijk is om vast te stellen waar er nu precies begonnen moet worden.

In die pagina-kop staan behalve het pagina-nummer ook de datum en de tijd. MSX1-bezitters moeten die zelf invullen, op een MSX2 computer zoekt Print dat zelf wel uit. Ook verschijnt er een commentaar-regel op iedere pagina, die men naar behoeven mag invullen.

De op twee na laatste vraag luidt 'Welk bestand'. Hier dient men niet alleen de bestandsnaam op te geven, maar ook het randapparaat waarop dat af te drukken bestand te vinden is. Als men alleen een cassette-recorder of alleen een diskdrive heeft hoeft men zich daar niets van aan te trekken.

Die 'apparaatnaam' is eigenlijk alleen van belang als het bestand op tape staat maar er ook een diskdrive aangesloten is; in dat geval kan men door "cas:" voor de bestandsnaam te zetten de recorder kiezen. Ook deze naam verschijnt overigens op de blad-kop.

Nu mag men nog opgeven of men al dan niet van de inspring-mogelijkheid gebruik wil maken die Print heeft voor Basic-programma's. Oftewel, wil men de programma's zonder meer afgedrukt hebben of mag Print proberen te ietsje leesbaarder te maken, door de regelnummers eruit te laten springen.

De laatste vraag is 'Alles goed ingevuld'. Nu kan de gebruiker nogmaals het hele scherm overzien en besluiten of alles wel echt naar wens is. Zo nee, dan kan men het vragenlijstje nogmaals doorlopen; zo ja, dan zal Print beginnen met afdrukken.

## Afdrukken

Tijdens dat afdrukken laat Print steeds zien met welke regel het bezig is, zodat men weet dat het programma in de weer is. Anders zou het soms wel eens kunnen lijken of er iets niet in orde is, want Print zal

steeds alle tekst voor een hele pagina inlezen en formatteren voor er afgedrukt kan worden.

Gebruikers van losse vellen moeten steeds een vel klaarzetten - indraaien in de printer tot de printkop op de juiste positie staat - alvorens er afgedrukt kan worden. De spatiebalk start het afdrukken.

## Tenslotte

We hopen veel lezers een plezier te doen met Print. Het is een prima programma om eens wat nettere uitvoer te produceren, terwijl de kolommen-uitvoer eigenlijk zo gebruikt kan worden voor bijvoorbeeld het verenigings-krantje.

We hebben er vanaf gezien om de werking van Print uit te leggen, daar deze tamelijk ingewikkeld is. Maar voor gevorderde programmeurs is het waarschijnlijk toch wel aardig om eens uit te pluizen hoe het nu allemaal in zijn werk gaat. De tekstbehandeling in Print kan namelijk op allerlei andere plekken ook heel goed gebruikt worden. Maar al te vaak zien we programmatuur die weliswaar uitstekend werkt maar waarbij de uitvoer toch wel wat fraaier gemogen had. In kolommen kunnen afdrukken zou daar zijn steentje aan kunnen bijdragen.

```
10 ' PRINT
20 '
30 ' MSX Computer Magazine
40 '
50 ' Initialisatie *****
60 CLEAR 9500: DEFINT A-Z 145
70 V2=PEEK(&H2D): ' msx1 of 2? 170
80 IF V2 THEN WI=80 ELSE WI=40 67
90 WIDTH WI 143
100 GOSUB 1290: ' functie-definities 108
110 GOSUB 890: ' parameters vragen 204
120 MA=600: ' max aantal regels 28
130 DIM T$(RP-1,KP-1): 'hulparray prin 49
strings
140 ' Hoofdflus *****
150 CLS 7
160 PRINT FN PC$( "MSX Computer Magazi 205
ne's afdruk-programma")
170 PRINT 142
180 PRINT "Bestand wordt geopend" 133
190 OPEN F$ FOR INPUT AS #1 12
200 GOSUB 1350 46
210 ' Lees en verwerk *****
220 R=0: C=0 107
230 IF EOF(1) THEN GOTO 470: ' einde b 221
estand
240 LINE INPUT #1,A$ 47
250 LOCATE 20,10: PRINT SPACE$(255):
LOCATE 20,10: PRINT A$ 251
```

8,CD,9F,00,FE,1B,CA,97,45,77,FE 21  
 1170 DATA 08,28,65,FE,10,CA,F5,45,3A,  
 36,48,FE,01,20,1F,7E,FE,1C,28,71,FE,1  
 D,CA,B8,45,FE,1E,CA,C1,45,FE,1F 240  
 1180 DATA CA,CD,45,FE,2D,CA,DB,45,FE,  
 3D,CA,E9,45,C9,7E,FE,0D,28,2B,FE,20,3  
 8,BF,FE,7F,28,BB,3A,36,48,B9,28 13  
 1190 DATA B5,FE,01,28,11,7E,FE,30,38,  
 AC,FE,3A,38,08,FE,41,38,A4,FE,47,30,A  
 0,7E,0C,23,CD,A2,00,18,98,3A,36 242  
 1200 DATA 48,B9,C8,AF,B9,20,8F,C9,AF,  
 B9,28,8A,0D,2B,11,90,4A,CD,73,44,C3,1  
 6,45,3A,36,48,FE,01,CA,16,45,3E 155  
 1210 DATA FF,32,38,48,C9,C5,D5,E5,2A,  
 06,48,2B,22,06,48,CD,C9,40,E1,D1,C1,C  
 3,16,45,C5,D5,E5,2A,06,48,23,18 139  
 1220 DATA EB,C5,D5,E5,2A,06,48,11,08,  
 00,19,18,DF,C5,D5,E5,2A,06,48,11,08,0  
 0,A7,ED,52,18,D1,C5,D5,E5,2A,06 53  
 1230 DATA 48,11,80,00,A7,ED,52,18,C3,  
 C5,D5,E5,2A,06,48,11,80,00,19,18,B7,C  
 5,D5,E5,CD,A8,00,28,B5,2A,22,F9 215  
 1240 DATA 0E,18,06,28,CD,4A,00,FE,FF,  
 20,03,3A,CC,FB,5F,3A,17,F4,B7,7B,20,0  
 F,FE,20,30,15,F5,3E,01,CD,A5,00 171  
 1250 DATA F1,C6,40,18,0A,FE,80,30,04,  
 FE,20,30,02,3E,2E,CD,A5,00,23,10,CF,3  
 E,0D,CD,A5,00,3E,0A,CD,A5,00,0D 187  
 1260 DATA 20,C0,C3,B2,45,3E,04,CD,0E,  
 45,3A,38,48,FE,FF,28,0E,FE,0D,20,0C,F  
 6,80,32,38,48,2A,06,48,A7,C9,37 165  
 1270 DATA C9,CD,BA,46,21,38,48,7E,CB,  
 27,CB,27,CB,27,CB,27,57,23,7E,82,5F,2  
 3,7E,CB,27,CB,27,CB,27,CB,27,57 187  
 1280 DATA 23,7E,82,6F,63,22,20,48,A7,  
 C9,C5,D5,E5,3E,02,CD,0E,45,3A,38,48,F  
 E,FF,28,1C,FE,0D,28,11,CD,BA,46 64  
 1290 DATA 21,38,48,CD,76,46,32,35,48,  
 E1,D1,C1,A7,C9,F6,80,32,38,48,18,F4,E  
 1,D1,C1,37,C9,21,38,48,06,04,7E 27  
 1300 DATA D6,30,FE,0A,38,02,D6,07,77,  
 23,10,F3,C9,7C,CD,D2,46,7D,F5,CB,3F,C  
 B,3F,CB,3F,CB,3F,CD,DF,46,F1,E6 49  
 1310 DATA 0F,C6,30,FE,3A,DA,EE,44,C6,  
 07,C3,EE,44,CD,FB,44,11,8A,48,CD,73,4  
 4,CD,45,46,54,5D,C9,D5,CD,FB,44 204  
 1320 DATA 11,A0,48,CD,73,44,CD,45,46,  
 D1,C9,C5,D5,CD,FB,44,11,B6,48,CD,73,4  
 4,CD,45,46,D1,C1,C9,C5,D5,E5,CD 186  
 1330 DATA FB,44,11,CB,48,CD,73,44,CD,  
 8A,46,E1,D1,C1,C9,E5,D5,C5,CD,EC,47,3  
 A,43,48,FE,03,28,46,3A,41,48,CD 15  
 1340 DATA 0C,00,C1,D1,E1,FB,C9,E5,D5,  
 C5,CD,EC,47,3A,43,48,FE,03,28,51,3A,4  
 1,48,CD,14,00,C1,D1,E1,FB,C9,D9 124  
 1350 DATA 08,F5,E5,2A,26,48,CD,EC,47,  
 3A,43,48,FE,03,28,4B,E5,DD,E1,3A,41,4  
 8,F5,FD,E1,E1,F1,D9,08,CD,1C,00 44  
 1360 DATA FB,C9,F3,CD,38,01,F5,E5,3A,  
 41,48,CD,24,00,E1,F1,5E,CD,3B,01,7B,F  
 5,21,00,D0,3A,42,48,CD,24,00,F1 179  
 1370 DATA C1,D1,E1,FB,C9,F3,CD,38,01,  
 F5,E5,3A,41,48,D5,CD,24,00,D1,E1,F1,7  
 3,CD,3B,01,18,DA,F3,E1,F1,D9,08 41  
 1380 DATA F5,C5,D5,E5,2A,26,48,E5,DD,  
 E1,3A,41,48,CD,24,00,E1,D1,C1,F1,CD,E

A,47,F5,C5,D5,E5,21,00,D0,3A,42 89  
 1390 DATA 48,CD,24,00,E1,D1,C1,F1,FB,  
 C9,DD,E9,C5,E5,CB,04,CB,04,7C,E6,03,3  
 2,43,48,4F,06,00,21,3D,48,09,7E 192  
 1400 DATA 32,41,48,E1,C1,C9,00,00,00,  
 00,00,00,00,00,00,00,00,00,00,00,0  
 0,00,00,00,00,00,00,00,00,00,00,00 212  
 1410 DATA 00,00,00,00,00,00,00,00,00,  
 00,00,00,00,00,00,00,00,00,00,00,0  
 0,00,00,00,00,00,00,00,00,00,00,00 28  
 1420 DATA 00,00,00,00,00,00,00,00,00,  
 00,4D,53,58,42,55,47,20,76,65,72,73,6  
 9,65,20,32,2E,30,39,20,63,6F,70 130  
 1430 DATA 79,72,69,67,68,74,20,31,39,  
 38,37,20,62,79,20,4D,43,4D,24,45,53,4  
 3,20,69,73,20,74,65,72,75,67,24 163  
 1440 DATA 43,6F,6D,6D,61,6E,64,6F,3A,  
 24,53,74,61,72,74,20,28,45,53,43,20,6  
 9,73,20,74,65,72,75,67,29,3A,24 42  
 1450 DATA 45,69,6E,64,65,20,28,45,53,  
 43,20,69,73,20,74,65,72,75,67,29,3A,2  
 4,4E,61,61,72,20,28,45,53,43,20 192  
 1460 DATA 69,73,20,74,65,72,75,67,29,  
 3A,24,4D,65,74,20,28,45,53,43,20,69,7  
 3,20,74,65,72,75,67,29,3A,24,FF 44  
 1470 DATA 44,72,75,6B,20,65,65,6E,20,  
 74,6F,65,74,73,24,50,41,47,45,20,20,2  
 0,69,6E,20,53,4C,4F,54,3A,20,20 1  
 1480 DATA 20,28,45,53,43,20,69,73,20,  
 74,65,72,75,67,29,24,50,41,47,45,20,3  
 0,3D,20,20,20,2C,31,3D,20,20,20 101  
 1490 DATA 2C,32,3D,20,20,20,2C,33,3D,  
 20,20,20,24,41,46,20,3A,42,43,20,3A,4  
 4,45,20,3A,48,4C,20,3A,41,46,27 216  
 1500 DATA 3A,42,43,27,3A,44,45,27,3A,  
 48,4C,27,3A,49,58,20,3A,49,59,20,3A,5  
 3,50,20,3A,50,43,20,3A,24,24,27 47  
 1510 DATA 41,27,20,2D,20,41,64,72,65,  
 73,20,77,69,6A,7A,69,67,2E,20,27,52,2  
 7,20,2D,20,52,65,67,69,73,74,65 205  
 1520 DATA 72,73,24,24,27,43,27,20,2D,  
 20,43,75,72,73,6F,72,61,64,72,65,73,2  
 0,20,20,27,53,27,20,2D,20,53,6C 185  
 1530 DATA 6F,74,73,24,24,27,48,27,20,  
 2D,20,48,65,6C,70,20,20,20,20,20,20,2  
 0,20,20,20,27,55,27,20,2D,20,55 19  
 1540 DATA 69,74,76,6F,65,72,65,6E,24,  
 24,27,4A,27,20,2D,20,4A,75,6D,70,20,7  
 2,65,6C,61,74,69,65,66,20,27,56 143  
 1550 DATA 27,20,2D,20,56,65,72,70,6C,  
 61,61,74,73,65,6E,24,24,27,4D,27,20,2  
 D,20,4D,53,58,2D,42,61,73,69,63 87  
 1560 DATA 20,20,20,20,20,27,57,27,20,  
 2D,20,57,69,6A,7A,69,67,65,6E,24,24,2  
 7,4F,27,20,2D,20,4F,70,76,75,6C 46  
 1570 DATA 6C,65,6E,20,20,20,20,20,20,  
 27,5A,27,20,2D,20,5A,6F,65,6B,65,6E,2  
 4,24,24,24,24,27,2D,27,20,20,20 233  
 1580 DATA 20,20,2D,20,4D,69,6E,20,20,  
 20,31,32,38,20,62,79,74,65,73,24,27,3  
 D,27,20,20,20,20,20,2D,20,50,6C 242  
 1590 DATA 75,73,20,20,31,32,38,20,62,  
 79,74,65,73,24,43,75,72,73,6F,72,73,2  
 0,2D,20,50,6C,75,73,2F,6D,69,6E 13  
 1600 DATA 20,31,20,6F,66,20,38,20,62,  
 79,74,65,28,73,29,24,08,20,08,24,1C,1



```

550 PRESET (213,149):PRINT #1,"11":PR
ESET (232,156):PRINT #1,"12"
560 LINE (16,153)-(19,153):LINE (16,1
24)-(19,124):LINE (16,94)-(19,94)
570 LINE (16,65)-(19,65):LINE (16,36)
-(19,36)
580 LINE (16,7)-(231,12),15,B
590 LINE (69,13)-(69,16),6
600 LINE (123,13)-(123,18),6
610 LINE (159,13)-(159,16),6
620 PRESET (117,16):PRINT #1,"30"
630 PRESET (63,16):PRINT #1,"15"
640 PRESET (172,16):PRINT #1,"45"
650 PRESET (224,16):PRINT #1,"60"
660 X3=15
670 FOR I=0 TO 12
680 LINE (X3,5)-(X3,12),15
690 X3=X3+18
700 NEXT I
710 LINE (14,178)-(30,182),7,BF
720 ' Hoofdflus *****
730 T2=TIME
740 IF T2<T1 THEN T1=T2
750 T=T2-T1
760 IF T<50 GOTO 730
770 T1=T1+50
780 S=S+1
790 IF S>59 THEN S=0:M=M+1:LINE (11,1
82)-(13,15),1,BF:PLAY B$,C$
800 IF M>59 THEN M=0:U=U+1:LINE (16,8
)-(230,10),1,BF:LINE (249,9)-(250,182
),7,BF:GOSUB 1370
810 IF U>11 THEN GOSUB 1420
820 PUT SPRITE 0,(95,147),1,1
830 PUT SPRITE 2,(31,141),15,5
840 PUT SPRITE 3,(31,150),7,6
850 FOR X4=225 TO 32 STEP -5
860 PUT SPRITE 4,(X4,150),15,2
870 NEXT X4
880 S1=INT(S*2.83)
890 LINE (11,182)-(13,182-S1),7,BF
900 M1=INT(M/1.4):X1=23:Y1=62
910 IF U=0 THEN GOTO 970
920 FOR I=0 TO U-1
930 LINE (X1,Y1)-(X1+9,Y1-42),7,BF
940 X1=X1+19
950 Y1=Y1+8
960 NEXT I
970 LINE (X1,Y1)-(X1+9,Y1-M1),7,BF
980 M3=INT(M*3.6)
990 LINE (16,9)-(16+M3,10),7,BF
1000 M4=INT(M*2.8)
1010 LINE (249,10)-(250,11+M4),1,BF
1020 PUT SPRITE 0,(95,147),1,0
1030 PUT SPRITE 2,(31,150),15,4
1040 PUT SPRITE 3,(31,143),0,8
1050 PUT SPRITE 6,(194,25),10,D
1060 GOTO 720
1070 ' Inlezen sprites *****
1080 FOR J=0 TO 7
1090 SP$="":SR$=""
1100 FOR I=1 TO 16
1110 READ A
1120 SP$=SP$+CHR$(A)
1130 NEXT I
1140 FOR I=1 TO 16
1150 READ A
1160 SR$=SR$+CHR$(A)
1170 NEXT I
1180 SPRITE$(J)=SP$+SR$
1190 NEXT J
1200 RETURN
1210 DATA 1,1,1,1,1,1,1,255,255,1,1,1
,1,1,1,1
1220 DATA 128,128,128,128,128,128,128
,255,255,128,128,128,128,128,128

```

121  
108  
99  
17  
231  
56  
61  
221  
152  
169  
46  
172  
133  
195  
65  
205  
31  
0  
82  
55  
151  
82  
184  
26  
220  
137  
225  
47  
160  
67  
96  
102  
174  
103  
65  
199  
61  
219  
109  
40  
216  
219  
252  
112  
134  
89  
95  
153  
52  
118  
207  
50  
0  
89  
202  
171  
204  
186  
90  
183  
216  
234  
102  
90  
119  
127  
110  
247

```

1230 DATA 0,0,16,56,28,14,7,3,3,7,14,
28,56,16,0,0
1240 DATA 0,0,8,28,56,112,224,192,192
,224,112,56,28,8,0,0
1250 DATA 0,0,0,0,0,0,0,0,0,0,0,0,6
,9,16
1260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,2
4,32,192
1270 DATA 0,0,0,0,0,0,0,0,24,36,2,62,
66,66,66,63
1280 DATA 0,0,0,0,0,0,0,0,68,170,145,
145,145,145,145
1290 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1
27,127,127
1300 DATA 128,128,128,128,128,128,128
,128,128,128,128,128,128,254,254,254
1310 DATA 1,1,127,127,127,0,0,0,0,0,0
,0,0,0,0
1320 DATA 128,128,254,254,254,0,0,0,0
,0,0,0,0,0,0
1330 DATA 127,127,127,127,127,127,127
,127,127,127,127,127,127,127,127
1340 DATA 254,254,254,254,254,254,254
,254,254,254,254,254,254,254,254
1350 DATA 0,0,0,0,0,0,0,0,124,66,66,6
6,124,64,64,64
1360 DATA 0,0,0,0,0,0,0,0,68,170,145,
145,145,145,145
1370 ' Ur-geluid *****
1380 PLAY "S3M10006C32D32","05E32G32"
,"04A32B32"
1390 PLAY "S3M10005C32D32","04E32G32"
,"03A32B32"
1400 PLAY "S3M5004C32D32","03E32G32"
,"03A32B32"
1410 RETURN
1420 ' Wissen na 12 uur *****
1430 X1=23:Y1=62
1440 FOR I=0 TO U-1
1450 LINE (X1,Y1)-(X1+9,Y1-42),1,8F
1460 X1=X1+19
1470 Y1=Y1+8
1480 NEXT I
1490 U=0
1500 IF D=3 THEN D=7:GOTO 1520
1510 IF D=7 THEN D=3
1520 RETURN

```

145  
228  
102  
120  
116  
86  
81  
152  
140  
165  
0  
99  
15  
82  
0  
69  
238  
51  
134  
0  
104  
22  
57  
246  
119  
111  
205  
137  
91  
139

## Categorie: utility's

# Tstbld

Zo langzaam maar zeker beginnen er wat specifieke MSX2 programma's op de redactie binnen te druppelen. Tot nog toe geen flitsend snelle spellen, maar toch wel programma's die duidelijk gebruik maken van de vele extra grafische mogelijkheden van MSX2.

Een voorbeeld daarvan is Tstbld - testbeeld - dat ons door de heer D. A. Dauwerse uit Huizen werd toegezonden. Een op zich heel simpel programma, dat echter fraai gebruik weet te maken van de mogelijkheden van schermtipe 8.

Het zet namelijk een goed gelijkend testbeeld op uw scherm, dat wel heel sterk lijkt op de testbeelden zoals de Nederlandse televisie-zenders ze uitzenden. Handig om de monitor eens eventjes goed af te regelen!

### Adjust

Dat testbeeld kan dan ook nog eens gebruikt worden om het scherm zo goed mogelijk te centreren. MSX2 kent namelijk het kommando SET ADJUST, waarmee de plaats van het beeld op het scherm kan worden bepaald. Na dat SET ADJUST moeten twee waarden opgegeven worden, een voor de horizontale instelling en eentje voor de verticale instelling. Deze beide waarden dienen tussen de 8 en de -7 te liggen.

In feite staan die waarden voor een *pixel-verschuiving*, met andere woorden, we kunnen de exakte positionering van het beeld op het scherm over een bereik van zestien beeldpuntjes zowel horizontaal als vertikaal verschuiven. Op die

manier kan het MSX2 beeld op vrijwel iedere televisie of monitor keurig in het midden gezet worden.

### Fijnregeling

In het programma Tstbld wordt die fijnregeling gedaan met behulp van de cursor-toetsen, waarmee het beeld puntje voor puntje over het scherm verschoven kan worden. Om die nieuwe instelling echter vast te kunnen houden moet er een kleine wijziging in het programma aangebracht worden. Momenteel zal - als we Tstbld met de spatiebalk afbreken - in regel 590 weer een SET ADJUST (0,0) worden uitgevoerd, om de uitgang-instelling weer terug te halen. Zoals ieder goed opgevoed programma laat Tstbld geen rotzooi slingeren, het laat de computer na afloop in dezelfde toestand als toen het begon. Tenminste, als die aanvangs-toestand inderdaad een in het midden geplaatst schermbeeld was.

Maar om nu die verschuivingen wel te bewaren nadat Tstbld gestopt is zijn er twee



mogelijkheden. De simpelste is natuurlijk om Tstbld ruwweg met Control-Stop af te breken. Dan wordt die regel 590 *niet* uitgevoerd, en blijft de instelling van SET ADJUST gehandhaafd. Wat ook kan - en eigenlijk netter is - is om de SET ADJUST (0,0) in regel 590 te verwijderen.

### Vasthouden

In beide gevallen echter geldt de zo ingestelde SET ADJUST slechts tot men de machine uitzet of een reset geeft. Om de nieuwe instelling permanent te maken - althans tot er weer een keertje SET ADJUST wordt gegeven - moet men nog eventjes het kommando:

SET SCREEN

geven. Dit kommando slaat namelijk de huidige scherminstellingen - zoals kleuren, aantal tekens per regel en schermtipe - op in het speciale stukje 'vast' geheugen van de MSX2. De volgende keer dat de computer dan opstart - na een reset of na uitgezet te zijn - zullen uit dat stukje geheugen de instellingen voor het scherm worden gelezen.

Voor de technisch geïnteresseerden, dat 'vaste' geheugen

bevindt zich in de MSX2 klok- en datum-chip, die voortdurend van stroom voorzien wordt door een piepklein Ni-Cad-celletje. Dat IC is namelijk ook verantwoordelijk voor het bijhouden van de klok- en kalender-functies, zelfs als de computer verder uitstaat. Die accu is voor zo'n drie maanden berekend, zodat u - als er tenminste nog een beetje interesse is voor de computer-hobby - niet bang hoeft te zijn dat die informatie opeens verdwenen blijkt.

### Besluit

Na deze uitweiding nog even terug naar Tstbld. We hopen dat u het programmaatje net zo aardig zal vinden als wij dat deden. Hoewel het geen vreselijk ingewikkelde dingen doet is het toch leuk genoeg. Vooral zo'n klein detail als het kunnen aan- en uitzetten van een pieptoon - zoals die meestal ook bij het 'echte' testbeeld te horen is sprak ons aan.

Voor de heer Dauwerse was het een van zijn eerste probeerseltjes, zoals hij ons via het scherm meedeelde. Inderdaad, via het scherm, want de enveloppe bevatte slechts een diskette met de tekst 'Reset to start'. De begeleidende brief stond ook op schijf.

```
10 REM      TSTBLD
20 REM      Een testbeeld voor MSX2
30 REM      door D.A.Dauwerse
40 REM      MSX Computer Magazine
50 REM
60 CLS: SCREEN 0: WIDTH 37: COLOR 15,4,4: KEY OFF
```

```

70 IF PEEK(&H2D)=0 THEN PRINT "Dit programma werkt niet op MSX1!": STOP
80 LOCATE 7,2: PRINT "MSX2 TESTBEELD"
90 LOCATE 0,5: PRINT "Met de cursortoetsen kunt u het beelddover het scherm versc
huizen."
100 PRINT: PRINT "[TAB] = Pieptoon uit"
110 PRINT "[ESC] = Pieptoon aan"
120 PRINT:PRINT"Met de spatie stopt u het testbeeld"
130 PRINT:PRINT"Druk op RETURN voor het testbeeld: ";
140 IF INPUT$(1)<>CHR$(13) THEN 140
150 SCREEN 8: KEY OFF: CLS: COLOR 73,1,73: OPEN "GRP:" AS #1: ON STOP GOSUB 590:
STOP ON
160 LINE (0,0)-(255,212),73,BF: CIRCLE (124,105),96,255,,1.09: PAINT (124,17),2
55
170 FOR LH=0 TO 255 STEP 14.6: LINE (LH,0)-(LH,210),255: NEXT LH
180 FOR LV=0 TO 212 STEP 16.3: LINE (0,LV)-(248,LV),255: NEXT LV
190 LINE (94,16)-(154,33),0,BF: LINE (169,33)-(169,48),0
200 LINE (169,33)-(181,33),0: LINE (169,48)-(193,48),0: CIRCLE (124,105),96,0,.6
7,.84,1.09
210 PAINT (171,39),0: LINE (79,33)-(79,48),0: LINE (79,48)-(53,48),0: LINE (79,3
3)-(65,33),0
220 CIRCLE (124,105),96,0,2.31,2.47,1.09: PAINT (65,37),0: LINE (86.5,33)-(87,48
),0,BF: LINE (79,48)-(182,48),0: LINE (44,64)-(196,64),0
230 LINE (196,50)-(196,64),0: CIRCLE (124,105),96,0,.5,.7,1.09: CIRCLE (124,105)
,96,0,2.47,2.65,1.09: PAINT (55,50),0
240 LINE (51,48)-(62,64),146,BF: CIRCLE (124,105),97,73,2.47,2.65,1.09: LINE (47
,49)-(52,49),73
250 FOR AA=24 TO 140 STEP 24.4: LINE (51+AA,48)-(62+AA,64),146,BF: NEXT AA
260 CIRCLE (124,105),96,146,.5,.63,1.09: LINE (197,53)-(197,64),146,BF: LINE (19
7,64)-(203,64),146,BF
270 PAINT (200,62),146: CIRCLE (124,105),97,221,2.7,3.07,1.09: LINE (36,97)-(65,
97),221: LINE (64,65)-(64,97),221
280 LINE (65,65)-(43,65),221: PAINT (50,72),221: LINE (65,65)-(93,97),167,BF: LI
NE (94,65)-(122,97),200,BF: LINE (123,65)-(151,97),31,BF
290 LINE (152,65)-(181,97),24,BF: LINE (182,97)-(212,97),3: LINE (182,97)-(182,6
5),3: LINE (182,65)-(204,65),3
300 CIRCLE (124,105),96,3,.1,48,1.09: PAINT (185,70),3: LINE (36,98)-(212,113),
0,BF: CIRCLE (124,105),96,0,3.2,3.63,1.09: LINE (45,145)-(50,145),0
310 LINE (50,145)-(50,114),0: PAINT (47,115),0: LINE (55,114)-(59,145),0,BF: LIN
E (64,114)-(68,145),0,BF: LINE (73,114)-(77,145),0,BF
320 FOR L=81 TO 110 STEP 1.5: LINE (L,114)-(L,145),0,BF: NEXT L
330 FOR L=111 TO 138 STEP 2: LINE (L,114)-(L,145),0,BF: NEXT L
340 FOR L=140 TO 170 STEP 3: LINE (L,114)-(L,145),0,BF: NEXT L
350 FOR A=1 TO 30 STEP 2: LINE (171+A,114)-(171+A,145),0,BF: NEXT A
360 LINE (201,145)-(204,145),0: CIRCLE (124,105),96,0,5.82,6.21,1.09: PAINT (205
,118),0: LINE (116,82)-(130,129),0,BF
370 LINE (36,105)-(212,105),255: LINE (123.5,82)-(123.5,129),255
380 FOR LH=43 TO 212 STEP 14.6: LINE (LH,98)-(LH,113),255: NEXT LH
390 CIRCLE (124,105),96,0,3.63,3.82,1.09: LINE (55,163)-(65,163),0
400 LINE (65,145)-(65,163),0: LINE (50,146)-(65,146),0: PAINT (60,150),0: LINE (
64,146)-(93,163),73,BF: LINE (93,146)-(123,163),146,BF
410 LINE (123,146)-(153,163),110,BF: LINE (153,146)-(183,163),219,BF: LINE (80,1
64)-(168,179),0,BF: LINE (67,180)-(180,180),221
420 CIRCLE (124,105),96,221,4,5.41,1.09: PAINT (124,190),221: LINE (117,180)-(13
0,201),16,BF: LINE (29,17)-(42,47),38,BF
430 LINE (15,17)-(28,105),97,BF: LINE (15,106)-(28,194),30,BF: LINE (29,164)-(42
,194),157,BF: LINE (205,164)-(219,194),157,BF: LINE (220,106)-(232,194),47,BF
440 LINE (220,17)-(232,105),108,BF: LINE (205,17)-(219,47),43,BF
450 Y=169: ST=6: B$="MSX2 TESTBEELD"
460 FOR I=1 TO LEN(B$): PRESET ((-250-ST*LEN(B$))/2)+X1,Y): COLOR 255,0: PRINT #
1,MID$(B$,I,1): X1=X1+ST: NEXT I
470 SOUND 0,90: SOUND 1,0: SOUND 11,255: SOUND 12,100: SOUND 8,8
480 X=0: Y=0
490 A$=INPUT$(1)
500 IF A$=CHR$(9) THEN SOUND 8,0
510 IF A$=CHR$(27) THEN SOUND 0,90: SOUND 1,0: SOUND 11,255: SOUND 12,100: SOUND
8,8
520 IF A$=CHR$(28) THEN X=X+1: IF X>8 THEN X=8
530 IF A$=CHR$(29) THEN X=X-1: IF X<-7 THEN X=-7
540 IF A$=CHR$(30) THEN Y=Y-1: IF Y<-7 THEN Y=-7
550 IF A$=CHR$(31) THEN Y=Y+1: IF Y>8 THEN Y=8
560 IF A$=CHR$(32) THEN 590
570 SET ADJUST (X,Y)
580 GOTO 490
590 SOUND 8,0: SET ADJUST(0,0): SCREEN 0: WIDTH 80: COLOR 15,4,4

```

214  
169  
162  
255  
173  
208  
124  
235  
170  
249  
237  
132  
66  
63  
8  
152  
70  
125  
141  
56  
196  
172  
83  
219  
171  
46  
23  
186  
251  
24  
1  
24  
190  
183  
71  
241  
192  
179  
150  
144  
216  
115  
174  
220  
28  
176  
205  
202  
210  
62  
69  
145  
66

## Kategorie: klokken

# Repwek

Toen we, in MSX Computer Magazine nummer 8, een oproep plaatsten aan de programmeurs onder onze lezers om eens een klok-programma in te stellen hebben we duidelijk gesteld dat die programma's niet al te lang moesten zijn. Hooguit een halve pagina, dat stond er.

Maar daar hebben sommige inzenders zich toch niet helemaal aan gehouden, merkten we.

Zo stuurde John Werk uit Leiden ons een prachtige repeteerwekker, die echter wel een stuk groter is. Desondanks hebben we besloten dit programma toch op te nemen, omdat we er behoorlijk van onder de indruk zijn.

### Wijzerklok

Het is een keurige wijzerklok, die desgewenst de hele en halve uren kan slaan. Het scherm is niet echt indrukwekkend, maar meer dan afdoende.

Zo kunnen we bij het opstarten van de klok de huidige tijd en de wektijd opgeven. Ook kan desgewenst gekozen worden voor niet wekken, terwijl ook het al dan niet slaan op de hele en halve uren ingesteld kan worden.

Als de klok eenmaal ingesteld is verschijnt er een nogal eenvoudige wijzerplaat, waarbij telkens ook aanwijzingen in het beeld staan die aangeven welke toetsen welke functies vervullen.

### Repeteerwekker

Het bijzondere aan Repwek is dat het een echte repeteerwekker is, die u desgewenst met muziek kan wekken. En daarbij bent u dan niet overgeleverd aan wat de radio op dat moment te bieden heeft!

In het programma zit namelijk een stukje muziek ingebouwd, dat u met de 'Nocturne' van Chopin uit dromenland teruglokt naar de harde werkelijkheid van het moeten opstaan. Mocht u dat moment nog eventjes willen uitstellen, dan kan dat ook. De spatiebalk

fungeert namelijk als sluimerknop, die u nog een korte poos uitslapen biedt.

Met de F5 kan de wekker helemaal uitgezet worden als u eenmaal opgestaan bent.

### Andere muziek

Voor mensen die liever met een ander stukje muziek gewekt willen worden heeft de inzender nog twee melodietjes mee gestuurd. Voor de muzikliefhebbers: deze programmaatjes werken ook zonder dat ze in Repwek zijn opgenomen.

Om echter de melodie in Repwek zelf te wijzigen moeten er een paar stappen zorgvuldig worden gevolgd. Het is namelijk een MERGE-operatie, en daar is niet iedere MSX-gebruiker even bekend mee.

Ten eerste is het absoluut noodzakelijk om de nieuwe melodie als afzonderlijk programma in te tikken, precies zoals deze in de listing staat. Dus regelnummering etcetera mogen niet gewijzigd worden.

Ook de REM-regels waarmee beide sub-programmaatjes eindigen zijn allemaal nodig, deze kunnen niet weggelaten worden.

Dan moet het op te nemen melodietje als ASCII-bestand worden weggeschreven. Voor disk-gebruikers gaat dat met:

SAVE"naam".A

Cassettegebruikers maken zo'n ASCII-file aan met:

SAVE"cas:naam"

Nu moet eerst het eigenlijke programma, Repwek, geladen worden. Het als ASCII-file weggeschreven melodietje kan vervolgens door diskgebruikers met:

MERGE"naam"

worden. Cassettegebruikers moeten:

MERGE"cas:naam"

gebruiken.

De truuk van dit mergen zit hem daarin dat tijdens het samenvoegen - want dat betekent merge - de nieuw ingelezen programmaregels in het programma in het computergeheugen worden tussengevoegd, net zoals dat vanaf het toetsenbord kan worden gedaan. Zo'n nieuwe regel zal dan ook een al bestaande regel met hetzelfde nummer overschrijven.

Dat alles leidt uiteindelijk tot een keurige nieuwe versie van Repwek, met het nieuwe wekmuziekje op de juiste plek in het programma.

### Mogelijkheden

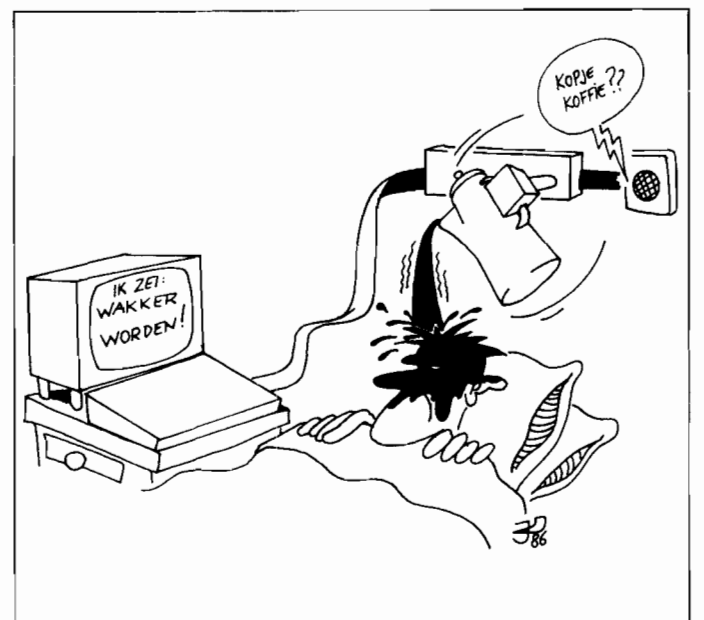
De schrijver van Repwek, John Werk, heeft nog een extra foefje in gebruik in zijn eigen programma. Dat foefje kan trouwens in alle klok-programma's met een wek-functie worden ingebouwd.

Hij maakt namelijk gebruik van de cassette-aansluiting om een paar minuten voor de wekker afloopt de koffiezet-machine aan te zetten. Dat is heel simpel te realiseren door gebruik te maken van de schakelfunctie voor de cassette-motor die MSX-Basic bezit in de vorm van het MOTOR kommando. MOTOR ON zet die motor aan, MOTOR OFF doet het tegenovergestelde.

Om die motor te kunnen besturen zit er in een MSX computer een klein relais ingebouwd, wat in feite in de stroomvoorziening van de motor opgenomen wordt. De zwarte - kleinste - plug van de cassettekabel, die normaal gesproken in de remote-aansluiting van de recorder gestoken wordt, is niets anders dan een rechtstreekse verbinding met dat relais.

Nu is dat ingebouwde relais niet zwaar genoeg om er een apparaat op aan te sluiten. Maar het kan wel weer een ander relais laten schakelen, een-tje die wel tegen 220 volt kan en zwaar genoeg is uitgevoerd om er desgewenst de wasmachine mee te schakelen.

Met andere woorden, wie daar behoefte aan heeft kan een MSX computer niet alleen als repeteerwekker met muziek gebruiken, maar er ook de koffie mee maken. Mogelijk een goede manier om uw MSX1 nog nuttig te kunnen blijven gebruiken nu MSX2 zo goedkoop begint te worden.....



```

10 REM REPWEK
20 REM
30 REM *****
40 REM *
50 REM * Door John Werk *
60 REM * LEIDEN *
70 REM * MSX Computer Magazine *
80 REM *
90 REM *****
100 REM
110 REM
120 REM
130 REM Wekmuziek:NOCTURNE (Chopin).
140 REM Eventueel ander wekmuziek in
150 REM te mergen, zie aparte lis-
160 REM tings, regelnummering is
170 REM al aangepast.
180 REM
190 REM
200 CLEAR:CLS:COLOR15,1,1:KEYOFF:WIDT
H40
210 OPEN"grp:"AS1
220 GOSUB1190
230 INTERVALON:ONINTERVAL=75GOSUB119
0
240 PRINTSTRING$(17,174);" KLOK ";STR
INGS$(17,175)
250 LOCATE13,2:PRINT"MET INGEBOUWDE"
260 LOCATE13,4:PRINT"REPETEERWEKKER"
270 LOCATE 6:PRINTSTRING$(20,174);STR
INGS$(20,175)
280 LOCATE 8:PRINT" Deze klok heeft "
;CHR$(34);"ouderwetse";CHR$(34);" wij
zers, " " een wekker die telkens na 9
minuten " " repeteert, een gong die ie
der heel uur " " het aantal uren slaat
en een 24-uurs " " cyclus."
290 PRINT:PRINTSTRING$(40,195)
300 LOCATE1,15:PRINT"Met F1+F2 kunt U
resp. de wekker en " " klok gelijk ze
tten, een muziekje zal U wekken.":PR
INTSTRING$(40,192)
310 LOCATE,21:PRINT"DE COMPUTER BEREK
ENT NU DE WIJZERSTANDEN";
320 LOCATE14,23:PRINT"< MOMENT >";
330 DEFFNX(X)=INT(37+X/1.4+.5)
340 DEFFNX(XX)=INT(37+XX/1.4+.5)
350 DIMX(60),Y(60),XX(60),YY(60),SS(3
):PI=4*ATN(1):W=6*PI/180
360 FORI=0TO60
370 W1=I*W
380 J=ABS(I-60)
390 X(J)=INT(128+70*COS(W1+.5*PI)+.5)
400 Y(J)=INT(96-70*SIN(W1+.5*PI)+.5)
410 XX(J)=INT(128+50*COS(W1+.5*PI)+.5)
)
420 YY(J)=INT(96-50*SIN(W1+.5*PI)+.5)
430 NEXT
440 RESTORE1220
450 FORJ=0TO3
460 SS(J)="":FORI=1TO32
470 READA:SS(J)=SS(J)+CHR$(A)
480 NEXTI,J
490 INTERVALOFF:GOSUB840
500 KEY(1)ON:KEY(2)ON:KEY(5)ON
510 ONKEYGOSUB840,910,1810
520 ' Hoofdprogramma *****
530 FORI=0TO3:SPRITES(I)=SS(I):NEXT
540 W1=TIME/50
550 IFW1<WVTHENT=T+1:CAL=CAL+2:K=1
560 WV=W1:W1=W1+T*1311+ST+CAL
570 U=INT(W1/720)MOD60:IFINT(W1/3600)
->24THENT=0:GOSUB1170:GOTO540
580 UU=(W1/3600)MOD24

```

```

590 IFUU<12THENPUTSPRITE2,(123,17),1,
2ELSEPUTSPRITE2,(123,17),1,3
600 M=INT(W1/60)MOD60
610 S=W1-60*INT(W1/60)
620 TT=INT((UU*3600)+(M*60)+INT(S))
630 PUTSPRITE0,(FNX(X(S)-1),Y(S)-1),8
0
640 IFWEK=1THENIFWT=TTTHENGOSUB680:GO
SUB1450
650 IFS<.65ORK=1THENGOSUB680
660 GOTO540
670 ' Wijzers en gong *****
680 K=0:IFM<>0THENLINE(128,96)-(FNX(X
(M-1)),Y(M-1)),1ELSELINE(128,96)-(FN
X(X(59)),Y(59)),10
690 IFU<>0THENLINE(128,96)-(FNX(XX(U-
1)),YY(U-1)),1ELSELINE(128,96)-(FNX(
XX(59)),YY(59)),10
700 LINE(128,96)-(FNX(X(M)),Y(M)),1
710 LINE(128,96)-(FNX(XX(U)),YY(U)),1
720 FORI=0TO55STEP5:PSET(FNX(X(I)),Y(
I)),1:NEXT
730 IFM=30ANDGNG=1THENSOUND7,56:PLAY"
t42S0M1100002L4G" "T42S0M1100004L4E"
,"T42S0M1100004L4C"
740 IFM=0THENGOSUB770
750 RETURN
760 ' Gong hele uren *****
770 IF WT=TT OR GNG=0 THEN RETURN
780 SOUND7,56:U=U/5
790 FORI=1TOU
800 PLAY"T42S0M900003L4G" ,"T42S0M9000
04L4E" ,"T42S0M900004L4C"
810 NEXT
820 RETURN
830 ' Instelling wektijd en klok ****
840 COLOR15,1:KEYOFF:SCREEN0:WIDTH39:
DEFUSR3=&H156:US=USR3(0)
850 SL=0:INTERVALOFF:LOCATE0,2:PRINT"
Wilt u gewekt worden (J/N)?:":W$=INPU
T$(1):FINSTR("JjNn",W$)THENIFINSTR("
Nn",W$)THENWEK=0:GOTO890ELSEWEK=1ELSE
850
860 LOCATE0,4:PRINT"Hoe laat?"
870 PRINT:INPUT"Uur";UW:IFUW>24THENPR
INT"Foute invoer, opnieuw":GOTO870
880 PRINT:INPUT"Minuten";MW:IFMW>60TH
ENPRINT"Foute invoer, opnieuw":GOTO88
0
890 WT=INT((UW*3600)+(MW*60))
900 ' Gelijkzetten klok *****
910 COLOR15,1:SCREEN0:DEFUSR3=&H156:U
S=USR3(0)
920 IFWEK=1THENPRINT"Wektijd is:":ELS
EPRINT"U wordt niet gewekt.":GOTO950
930 IFUW<10THENPRINTUSING"0#:";UW:EL
SEPRINTUSING"##:";UW;
940 IFMW<10THENPRINTUSING"0#";MWELSEP
RINTUSING"##";MW
950 LOCATE0,4:PRINT"Moet gong ieder u
ur slaan (J/N)?:":W$=INPUT$(1):FINSTR
("JjNn",W$)THENIFINSTR("Nn",W$)THENG
NG=0:ELSEGNG=1ELSE950
960 LOCATE0,2:IFGNG=1THENPRINT"Gong s
laat alle hele en halve uren."ELSEPRI
NT"Gong slaat niet."
970 LOCATE0,4:PRINT"Moet klok gelijk
gezet worden (J/N)?:":W$=INPUT$(1):IF
INSTR("JjNn",W$)THENIFINSTR("Nn",W$)T
HENGOSUB1070:RETURNS0ELSE980ELSE970
980 LOCATE0,4:PRINT"Hoe laat is het n
u?":PRINT
990 PRINT:INPUT"UUR";UU
1000 IFUU>24THENPRINT"Foute invoer, o

```

```

pnieuw":GOTO990 150
1010 PRINT:INPUT"Minuten";MM 73
1020 IFMM>60THENPRINT"Foute invoer, o
pnieuw":GOTO1010 217
1030 PRINT:INPUT"Secondes";SS:SS=SS+6 121
1040 IFSS>60THENPRINT"Foute invoer, o
pnieuw":GOTO1030 173
1050 ST=(UU*3600)+(MM*60)+SS:GOSUB117
0:GOSUB1070:RETURN500 188
1060 ' Teken en wijzerplaat ***** 0
1070 COLOR15:1:SCREEN2 139
1080 CIRCLE(128,96),80,10,,,1.33:PAIN
T(128,20),10 194
1090 IFWEK=1THENGOSUB1130 234
1100 IFSL=1THENGOSUB1820 163
1110 K=1:RETURN 255
1120 ' Afdrukken wektijd ***** 0
1130 PUTSPRITE1,(10,10),7,1:PRESET(28
,13):COLOR7:IFUW<10THENPRINT#1,USING"
0#:";UW;ELSEPRINT#1,USING"##:";UW;
1140 IFMW<10THENPRINT#1,USING"0#";MW
LSEPRINT#1,USING"##";MW 139
1150 COLOR15:RETURN 43
1160 ' reset klok ***** 0
1170 TIME=0:T=0:CAL=0:W1=0:WV=0:K=1:U
U=0:RETURN 219
1180 ' Intro muziek ***** 0
1190 SOUND7,56:PLAY"T100S0M4500003L4F
#BBL804C#D#EC#L1D#." "T100S0L404F#BBL
805C#D#EC#L1D#." "T100S0L402F#BBL803C
#D#EC#L1D#." :RETURN 134
1200 ' Sprites ***** 0
1210 ' Seconden-punt ***** 0
1220 DATA224,224,224,0,0,0,0,0 149
1230 DATA0,0,0,0,0,0,0,0 202
1240 DATA0,0,0,0,0,0,0,0 205
1250 DATA0,0,0,0,0,0,0,0 208
1260 ' Weksymbool ***** 0
1270 DATA1,1,3,7,15,15,31,31 16
1280 DATA31,31,63,63,127,127,24,24 198
1290 DATA128,128,192,224,240,240,248,
248 183
1300 DATA248,248,252,252,254,254,0,0 255
1310 ' AM ***** 0
1320 DATA34,83,138,138,250,138,138,0 18
1330 DATA0,0,0,0,0,0,0,0 204
1340 DATA32,96,160,160,32,32,32,0 203
1350 DATA0,0,0,0,0,0,0,0 210
1360 ' PM ***** 0
1370 DATA242,139,138,242,130,130,130,
0 223
1380 DATA0,0,0,0,0,0,0,0 219
1390 DATA32,96,160,160,32,32,32,0 218
1400 DATA0,0,0,0,0,0,0,0 197
1420 REM 0
1430 ' WEKMUZIEK***** 0
1440 REM 0
1450 SOUND7,56 211
1460 PUTSPRITE0,(-32,192),8,0 202
1470 STRIG(0)ON:ONSTR1GGOSUB1800 18
1480 COLOR15:LINE(75,179)-(190,187),1
,BF:DRAW"BM15,179":PRINT#1,"Spatie=S
i
uimer ---- F5=Stop" 219
1490 INTERVALON:ONINTERVAL=27000GOSUB
1450 162
1500 PLAY"T15502L4V11E-R4L404E-O2L4E-
R4L404S0M19000B-" "T155V1203L2B-R4B-"
"T156V12R4R2L404E-" 82
1510 PLAY"05L1GL4FGL2F." "V1102L4E-R4
04E-O2E-R404E-O2E-R404E-" "T155V1103L
2B-R4C-R4B-" 159
1520 PLAY"L2E-L404B-L205GL4C06L2C05L4
G" "L402D03L2B-O2L4C03B-A-O2L4C03B-O4
C" "R403L4G04C03L2E.G." 101

```

```

1530 PLAY"L2B-.A-L4GL2F." "02L4F03B-R
402F03L2A-O2L4B-O3L2B-" "04L4D-FD-R4L
204CR4D" 121
1540 PLAY"L2GL4DL2E-.C." "02L4B03L2BC
04CL803BB-O2L4A04L2C" "R404L2DE-R403L
2A." 106
1550 PLAY"L404B-O6DCL805B-A-GA-CDL2E-
." "L202B-.V12B-.V11L4E-L8B-R8B-" "03
L2B-.V12L804DC03B-O4C03A-B-V11L2G." 127
1560 PLAY"R2L404B-L205G.L8FGFEFG" "L4
02E-O3L2G02L4E-O3L2G02L4E-O3L2A-" "R4
03L4B-O4E-R403B-O4E-R4C-E-" 112
1570 PLAY"05L4FL2E-.L8FE-DE-FG04B05CD
-CF" "02L4E-O3L2G02L4D03L2G02L4C03L8G
R8G" "R403L4B-O4E-R403B-O4D03L2E." 120
1580 PLAY"EA-G06D-C05G" "02L4C03L8B-R
8B-" "03L2GR804L8C" 139
1590 PLAY"L2B-.A-L4G" "L402F03B-R402F
03L2A-" "L4D-FD-R4L2C" 223
1600 PLAY"05L2F.L32EFL4GGD" "02L4B-O3
L2B-R1602L4B03L2B" "R404DR4R16D" 26
1610 PLAY"L2E-.C.O4L4B-O6DC" "L404DL4
CL803BB-L2A.A-." "04L2E-.L402A04L2C02
B-." 60
1620 PLAY"L805B-A-GA-CD" "V1304L8DC03
B-O4C03A-B-" "02L2B-." 39
1630 PLAY"L1E-L4DE-L2F." "V1102L4E-O3
B-O4E-O2E-O3B-O4E-O2B-O4DF" "V1103L2G
L4B-L2G.R8L8B-O4L4FD 201
1640 PLAY"L2GL4FL2F." "02L4B-O4DF02A0
4CF" "R8L803B-L404FDR803L8AL404FC" 4
1650 PLAY"05L2C.L4E-E-E-" "02L4A04CF0
2A-O4CE-" "R8L803AL404FCR8L803A-L404E
-C" 103
1660 PLAY"E-L8DE-F.L16E-" "L402A-O3B0
4E-" "R8L8B-L404E-O3B" 215
1670 PLAY"L2E-." "02L4E-O3B-O4E-" "R8
L803GL404E-O3B-" 101
1680 PLAY"L204B-." "02L4E-O3B-O4E-" "
R8L803GL404E-O3B-" 134
1690 PLAY"05L2B-.AL4GL2F.D." "L402E03
L2G02L4EL203GL402F03FA02G03GB-" "R404
L4D-E-R403AFR4B-G" 110
1700 PLAY"05L2E-.L4DCD" "03L4CG04E-O2
F04L2D" "R404L4E-O3GV12L2A." 103
1710 PLAY"04B-BB-O5CCD" "V1302L4B-AG+
S0M1870004B-AA-V11" "L404DD+EO3GAB-V1
1" 77
1720 PLAY"05L2GL804AB-O4BB-O5C+DG.L16
F" "L402E-L804G.R16GR802L4E-O3L8A-.R1
6A-R8" "V1003R4L8B-.R16B-R8R4B-.R16B-
R8" 153
1730 PLAY"L2FL4E-.L8FE-DE-F" "02L4E-L
803G.R16GR802L4D03L8G.R16GR8" "V10R40
3L8B-.R16B-R8R4B-.R16B-R8" 19
1740 PLAY"G04B05CD-CFEA-G06D-C05G" "V
1102L4C03L8G.R16GR802L4C03L8G.R16GR8"
"03R4L8B.R16BR8R4B.R16BR8" 118
1750 PLAY"L2B-.A-L4G" "02L4F03L8B-.R1
6B-.R1602L4F03L8A-.R16A-." "R404L8D-.
R16D-.R16R4C.R16C." 29
1760 PLAY"L2F.L64EFL4GGD" "R3202L4B-O
3L8B-.R16B-.R1602L4B03L8B.R16B." "R32
R404L8D.R16D.R4D.R16D." 72
1770 PLAY"L2E-.C." "03L4CL8G.R16G.R16
02L4A03L8A.R16A." "R4C.R16C.R16R4C.R1
6C." 27
1780 PLAY"04L4B-O6DC05L8B-A-GA-CDS0M3
1000L1E-" "02L4B-O3L8A-.R16A-.R1602L4
B-O4L8C.R1603B-.O2L4E-GB-O3L8E-" "R40
4L8F.R16D.R16R403G.R16G.R16L2G." 70
1790 SL=1:GOSUB1820:RETURN540 70
1800 SOUND7,63:SL=1:BEEP:GOSUB1820:SL
=1:RETURN540 168

```

```
1810 INTERVALOFF:BEEP:SOUND7,63:SL=0:
K=1:STRIG(0)OFF:LINE(15,179)-(260,190
),1,BF:RETURN540
1820 K=1:STRIG(0)OFF:LINE(15,179)-(26
0,187),1,BF:DRAW"BM100,179":PRINT#1,"
Sluimer":RETURN
```

```
1500 PLAY"T80V15L1604R16G05CD","V15T8
004L4C","T80V15" 72
1505 PLAY"T80V15L1604R16G05CD","V15T8
004L4C" "" 33
1510 PLAY"T80V15L1604R16G05CD","V15T8
004L4C" "" 16
1515 PLAY"T80V15L1604R16G05CD","V15T8
004L4C" "" 36
1520 PLAY"R1604G05E04G","C","S0M14000
02L4C" 115
1525 PLAY"R1604G05E04G","C","02L4C" 180
1530 PLAY"R1604G05E04G","C","02L4C" 163
1535 PLAY"R1604G05E04G","C","02L4C" 183
1540 PLAY"T80S0M5000005L4EL2EL8E.L16F
","T80V1203L8CG04C03G04C03G04C03G","T
80V13L1C" 96
1545 PLAY"L2FR16L16S0M9944FFFFF","0
3L8DA04D03A04D03A04D03A","D" 39
1550 PLAY"S0M15000L4GL2GL8G.L16A","GB
04D03B04D03B04D03B" "G" 140
1555 PLAY"S0M30000L2E","03CG04C03G04
C03G04C03G" "C" 249
1560 PLAY"T80S0M9640L405EEL16EEEEEEEF
","T80S0M964005L4CCL16CCCCC04GA","T8
0L803CG04E03G04E03G04E03G" 43
1565 PLAY"S0M29000L2FR16S0M9040L16FFF
FFF" "S0M2900004L2AR16S0M904005L16DD
DD04AB","DF04D03F04D03F04D03F" 196
1570 PLAY"S0M30000L205G,S0M9000L8G.L1
6A","S0L204B.L8B.L1605C","L803GB04D03
```

```
B04D03B04D03B" 28
1575 PLAY"S0M3600005L1E","S004L1G","C
G04E03G04E03G04E03G" 160
1580 PLAY"T80R806L8EDCR805BAG" "T80R8
06L8C05BAR8GF#E" "T80V1302L2AE" 37
1585 PLAY"L805AR8GFE","L805FCEDC0C04G
" "V14L4FGL2C" 27
1590 PLAY"R8L805EDCR804BAG" "R805L8C0
4BAR8GF#E" "V13L2AE" 43
1595 PLAY"AGFS0M10000GS0M30000L2G","F
EDDL2D" "L4FL8FGL2G" 169
1600 PLAY"V15L1603B04DFGDFG05D04G05DF
GDFG06D" "V14L403FGB04V15D","V14L402G
03FGV15B" 138
1605 PLAY"S0M200004T124V15L1604G05DFG0
4G05DFG04G05DFG04G05DFG" "T62V14L803F
GFG" "" 125
1610 PLAY"T80S0M30000L4EL2ES0M9500L8E
.L16F" "T80S0M30000L404GL2GL8G.L16A"
" "T80V13L803CG04C03G04C03G04C03G" 217
1615 PLAY"S0M30000L4FFR16S0M9000L16FF
FFFF" "S004L4AAR16L16AAAAAAB" "03L8V
13DA04D03A04D03A04D03A" 229
1620 PLAY"S0M30000L4GL2GS0M9000L8G.L1
6A" "S0L404BL2BS0L8B.L1605C" "03GB04D
03B04D03B04D03B" 191
1625 PLAY"S0M30000L4EV14L16GE-04B-G05
FC04AF05D04BGD" "L404G03V12GAG" "V140
2L4CE-FG" 42
1630 PLAY"S0M30000L406EES0M8400L16EEE
EEEE" "05L4S0GGS0L16GGGGGGGG" "L803C
G04C03G04C03G04C03G" 231
1635 PLAY"S0M30000L4FFS0M8450L16FFFFF
FFG" "S0L4AAS0L16AAAAAAB" "DF04D03F0
4D03F04D03F" 105
1640 PLAY"S0M30000L4GL2GS0M9000L8G.L1
6A" "S0L4BL2BS0L8B.06L16C" "03GB04D03
B04D03B04D03B" 54
1645 PLAY"S0M3000005L4ES0M15000L8D04G
05E04G05EG" "S004L4GV1404L4DEC" "V140
2L4CR204E" 72
1650 PLAY"S0M30000L205ES0M12000L1604A
FA05C04BGB05D" "S0L204G" "" 137
1655 PLAY"S0M30000L205EL4CS0M10000L8G
.L16A" "S004L2GL4AS0L8B." "V14L803EG0
4CECC04BB" 141
1660 PLAY"T70S0M30000L205EL4CS0M10000
T65L8G.L16A" "T70S004L2GL4AS0T65L8B."
" "T70V14L803EG04CECCT6504BB" 144
1665 PLAY"T60S0M55000L104E" "T60S004L
1G" "T60S005L1C" 103
1670 ' 0
1675 ' 0
1680 ' 0
1685 ' 0
1690 ' 0
1695 ' 0
1700 ' 0
1705 ' 0
1710 ' 0
1715 ' 0
1720 ' 0
1725 ' 0
1730 ' 0
1735 ' 0
1740 ' 0
1745 ' 0
1750 ' 0
1755 ' 0
1760 ' 0
1765 ' 0
1770 ' 0
1775 ' 0
1780 ' 0
```

1500 PLAY"T105V1404L8B.05L16CL8D.L16C  
+L8D.L16C+L8D.L16C+L2DL8E.04L16B05L2C  
L8D.L1604AL2B", "T105V1404L8G.L16AL8B.  
L16A#L8B.L16A#L8B.L16A#L2B05L8C.04L16  
G#L2AL8B.L16F#L2G", "T105V15R402L4GB03  
DG02GR404D03DR4G02G" 55  
1505 PLAY"04L8G.L16AL8B.L16A#L8B.L16A  
#L8B.L16A#L2BL8AG", "03L8B.04L16DL8G.L  
16F#L8G.L16F#L8G.L16F#L2GL8F#E", "L4R4  
GB03D#G02E" 63  
1510 PLAY"EF#DAGEL4DR8", "GDF#F#EC#03L  
4F#", "02AR403AL8D" 57  
1515 PLAY"05L8DGL4DF#GL206CL1605GF#ED  
", "R4L405GC04BL205D", "R4L402BAGF#03D0  
2F#" 192  
1520 PLAY"T105V14L405CL804BR805E.L16C  
04L4BL8AR8G.L16A", "T105V14L4AL8GR805C  
.L16AL4GL803DR803B.L1604D", "T105V15L2  
G03L4CD04F#02" 205  
1525 PLAY"04L8B.L16A#L8B.L16A#L8B.L16  
A#L2B05L8C.L1604G#", "04L8D.L16C#L8D.L  
16C#L8D.L16C#L2DL8E.L1603B", "L402GB03  
DGFE" 16  
1530 PLAY"V14L2AL8B.L16F#L4G", "L204CL  
8D.L16CL403B02V15L8GR8", "V1403EE-D02G" 201  
1535 PLAY"V14L805DC#D04B05DS0M960004G  
BDB", "V13R8R203L4GB", "V14R4R802L4GV13  
03G04D" 232  
1540 PLAY"V14A05CS0M960004F#ADF#V14GF  
#S0M9600GAB05C", "V1403DV13F#F#V1402BV  
1303GG", "R403A04DR403B04D" 135  
1545 PLAY"V1405L8DC#DEDC#04BA#B05C04B  
A", "V1402F#V1303F#04CV1502GV1303G04D#  
", "R404CDR403BA" 13  
1550 PLAY"S0M9600GBAGF#AEF#GEC#03AV15  
04DL8V1403D", "V1402EV1303GF#V14GV1303  
GV1402AD", "R4BAR4BAF#" 154  
1555 PLAY"V14L805DC#D04B05DS0M960004G  
BDB", "V13R8R203L4GB", "V14R4R802L4GV13  
03G04D" 238  
1560 PLAY"V14A05CS0M960004F#ADF#V14GF  
#S0M9600GAB05C", "V1403DV13F#F#V1402BV  
1303GG", "R403A04DR403B04D" 141  
1565 PLAY"V1405L8DC#DEDC#04BA#B05C04B  
A", "V1402F#V1303F#04CV1502GV1303G04D#  
", "R404CDR403BA" 19  
1570 PLAY"S0M9600GBAGF#AEF#GEC#03AV15  
04DL8V1403D", "V1402EV1303GF#V14GV1303  
GV1402AD", "R4BAR4BAF#" 160  
1575 PLAY"V15L805DC#D", "V1504L8DA#D",  
" 58  
1580 PLAY"05EC04ABA#B", "05C04AF#GF#G"  
"V15L403D" 153  
1585 PLAY"05C04AF#05DC#DECO4ABA#B", "A  
F#DBA#B05C04AF#GF#G", "DR2D" 179  
1590 PLAY"05C04AF#05DC#D", "04AF#DDEF#  
", "D" 207  
1595 PLAY"04B05DS0M1000004GBD05GV15EG  
S0M100000CE04A05C", "L402G03BR4CE", "R4D  
R2G" 105  
1600 PLAY"V15F#AS0M100000DEFF#V15L4AL8  
G", "03DF#F#04CL8D", "R404CR4D." 62  
1605 PLAY"V15L805DC#D", "V1504L8DA#D",  
" 39  
1610 PLAY"05EC04ABA#B", "05C04AF#GF#G"  
"V15L403D" 134  
1615 PLAY"05C04AF#05DC#DECO4ABA#B", "A  
F#DBA#B05C04AF#GF#G", "R2D" 198  
1620 PLAY"05C04AF#05DC#D", "04AF#DDEF#  
", "D" 188  
1625 PLAY"04B05DS0M1000004GBD05GV15EG  
S0M100000CE04A05C", "L402G03BR4CE", "R4D  
R2G" 86

1630 PLAY"V15F#AS0M100000DEFF#V15L4AL8  
G", "03DF#F#04CL8D", "R404CR4D." 71  
1635 PLAY"V105V1404L8B.05L16CL8D.L16C  
+L8D.L16C+L8D.L16C+L2DL8E.04L16B05L2C  
L8D.L1604AL2B", "T105V1404L8G.L16AL8B.  
L16A#L8B.L16A#L8B.L16A#L2B05L8C.04L16  
G#L2AL8B.L16F#L2G", "T105V15R402L4GB03  
DG02GR404D03DR4G02G" 86  
1640 PLAY"04L8G.L16AL8B.L16A#L8B.L16A  
#L8B.L16A#L2BL8AG", "03L8B.04L16DL8G.L  
16F#L8G.L16F#L8G.L16F#L2GL8F#E", "L4R4  
GB03D#G02E" 57  
1645 PLAY"EF#DAGEL4DR8", "GDF#F#EC#03L  
4F#", "02AR403AL8D" 88  
1650 PLAY"05L8DGL4DF#GL206CL1605GF#ED  
", "R4L405GC04BL205D", "R4L402BAGF#03D0  
2F#" 186  
1655 PLAY"T105V14L405CL804BR805E.L16C  
04L4BL8AR8G.L16A", "T105V14L4AL8GR805C  
.L16AL4GL803DR803B.L1604D", "T105V15L2  
G03L4CD04F#02" 236  
1660 PLAY"04L8B.L16A#L8B.L16A#L8B.L16  
A#L2B05L8C.L1604G#", "04L8D.L16C#L8D.L  
16C#L8D.L16C#L2DL8E.L1603B", "L402GB03  
DGFE" 10  
1665 PLAY"L2AL8B.L16F#L4G", "L204CL8D.  
L16CL403B02V15L8G", "V1403EE-D02G" 106  
1670 ' 0  
1675 ' 0  
1680 ' 0  
1685 ' 0  
1690 ' 0  
1695 ' 0  
1700 ' 0  
1705 ' 0  
1710 ' 0  
1715 ' 0  
1720 ' 0  
1725 ' 0  
1730 ' 0  
1735 ' 0  
1740 ' 0  
1745 ' 0  
1750 ' 0  
1755 ' 0  
1760 ' 0  
1765 ' 0  
1770 ' 0  
1775 ' 0  
1780 ' 0



**Categorie: dichters**

# Strkls

Heeft u ook zo'n moeite met het schrijven van uw Sinterklaas-gedichtjes? Dan hebben we dat probleem bij deze voor u opgelost!

Strkls - oftewel Sinterklaas - schrijft voortaan uw kreupelrijmen voor het heerlijk avondje.

Nadat u Strkls hebt verteld wat u aan wie wilt geven verschijnt er een keurig zesregelig gedichtje op het scherm, waar die naam en het cadeau in genoemd worden. Mocht u niet helemaal tevreden zijn met het rijmpje dan kunt het nog eens proberen, want STRKLS kent een aantal verschillende Sinterklaas-gedichtjes. Overigens zijn het er ook weer niet al teveel, in totaal zijn er 18 varianten beschikbaar in Strkls. In de data zijn namelijk twee beginregels, drie midden-regels en twee eind-regels opgenomen.

Er is echter ook nog ruimte voor eigen creativiteit, het is een fluitje van een cent om extra dicht-regels op te nemen in de data. Per soort - begin, midden of einde - mogen er maximaal 10 regelsparen in staan. Of meer, als u de variabele MR in

regel 110 even aanpast. Wie wilt mag ons wel eens een aangepaste versie sturen, want er is natuurlijk veel meer te doen

met dit programma dan alleen maar Sinterklaas-rijmpjes maken.

```

10 REM STRKLS
20 REM
30 REM MSX Computer Magazine
40 REM
50 REM Een Sinterklaas-surprise
60 REM
70 CLEAR 5000
80 X=RND(-TIME)
90 'Instelbare waarden *****
100 MR=6: 'aantal regels v.h. gedicht
110 MV=10: 'maximaal aantal varianten
120 DIM R$(MR,MV): 'beschikbare regels
130 DIM G$(MR): 'het gedicht
140 DIM LN(MR): 'aantal varianten per regel
150 'Handige functie *****
160 DEF FNVOEG$(A$,X,B$)=LEFT$(A$,X-1)+B$+MID$(A$,X+2)
170 'Scherm initialiseren *****
180 SCREEN 0: WIDTH 40
190 'Lees alle varianten in en bepaal hoeveel het er zijn *****
200 FOR REGEL=1 TO MR STEP 2
210 V=1
220 READ A$
230 IF A$<>"*" THEN R$(REGEL,V)=A$: READ R$(REGEL+1,V): V=V+1: GOTO 220
240 LN(REGEL)=V-1
250 NEXT REGEL
260 'Maak scherm *****
270 CLS: PRINT "MSX Computer Magazine presenteert:": PRINT
280 PRINT "STRKLS - een programma om": PRINT "Sinterklaasgedichten te maken.": PRINT
290 'Vraag om gegevens *****
300 PRINT: INPUT "De gelukkige is"; N$
310 INPUT "Wat is het geschenk"; K$
320 PRINT: PRINT "Ik stel het volgende gedicht voor:": PRINT
330 'Maak het gedicht *****
340 FOR REGEL=1 TO MR
350 IF (REGEL AND 1) THEN V=INT(RN

```

```

D(1)*LN(REGEL))+1: 'Kies een variant
360 P=INSTR(R$(REGEL,V),"%N"):
'Staat er %N in?
370 IF P THEN G$(REGEL)=FNVOEG$(R$(REGEL,V),P,N$): GOTO 410
380 P=INSTR(R$(REGEL,V),"%K"):
'Staat er %K in?
390 IF P THEN G$(REGEL)=FNVOEG$(R$(REGEL,V),P,K$):GOTO 410
400 G$(REGEL)=R$(REGEL,V)
410 NEXT REGEL
420 'Druk de regels af:
430 FOR REGEL=1 TO MR
440 PRINT G$(REGEL):'verander even tueel in LPRINT
450 NEXT REGEL
460 'Nog een gedicht? *****
470 PRINT: PRINT "Druk op de spatiebalk voor het volgende gedicht"
480 WS=INKEY$: IF WS="*" THEN 270 ELSE 480
490 'De dichtregels: *****
500 DATA Urenlang liep Sint te denken ,wat hij toch aan %N zou schenken.
510 DATA De Sint heeft me toch wat af gelopen,om iets voor %N te kopen.
520 DATA *
530 DATA Toen heeft hij de krant gelezen,en dacht 'Het moet een %K wezen!'
540 DATA Nadat hij zolang had gezocht ,heeft hij maar een %K gekocht.
550 DATA En toen midden in de nacht,h eeft hij een %K bedacht.
560 DATA *
570 DATA Helaas voor %N is er niet me er,want Sint's voeten doen zo zeer!
580 DATA Sint hoopt dat je er blij me e bent,en het niet terug naar Spanje zendt!
590 DATA "Beste %N, veel plezier erme e",Sint had het in ieder geval voor t weel!
600 DATA *

```



## Een programmeer-voorbeeld

# SCLPTR

Sculptor - met als werknaam scptr, je moet heel wat verzinnen om binnen die 6 tekens voor een programmaam te blijven - is een tekenprogramma voor MSX2 dat op een wat merkwaardige wijze ontstaan is. Een aantal weken terug moest er namelijk op de redactie voor een bepaald programma nog even pijlsnel een plaatje bijgewerkt worden. Een tekenprogramma dat een plaatje kon opslaan, om later in een ander MSX2 programma te gebruiken, was echter niet voor handen.

Nu is op de redactie niemand te beroerd in zo'n situatie even een 'ad-hoc' programmaatje te schrijven en al doende werd duidelijk hoe goed het riante MSX2 video-geheugen zich leent voor het schrijven van tekenprogramma's met handige extra's. Sculptor is daarom uitgegroeid tot een MSX2 tekenprogramma waarin het video-geheugen ten volle wordt benut. Bovendien zijn er een paar aardigheidjes met systeem-variabelen in verwerkt.

### Listing-formaat

De listing bij dit artikel is misschien wat ongebruikelijk van formaat, op het eerste gezicht tenminste. Normaal gesproken drukken we onze listings met een regel-breedte van 37 kolommen, zodat het programma in de listing er net zo uitziet als op het scherm.

Dat is ook precies de reden waarom we MSX2 programma's op 80 kolommen per regel afdrukken. We gaan er van uit dat de meeste MSX2 bezitters hun machine standaard op 80 kolommen zullen hebben staan, hetgeen nu eenmaal veel overzichtelijker is. Bovendien kan men zo in een oogopslag zien of een bepaald programma nu MSX1 of MSX2 is.

### Geheugen-pagina's

Uitgaande van een machine met 4 geheugen-pagina's (128K VRAM) voor scherm-type 5, ligt een eerste toepassing voor de hand. Het zal een

ieder wel eens zijn overkomen bij het maken van een tekening; een PAINT-kommando op de verkeerde plek gegeven of even niet goed op de randkleur gelet en weg is die mooie tekening waarin toch al een klein uurtje werk zat. Je kan dan hoog springen of laag springen, de tekening komt er niet door terug.

Een MSX2 machine echter kent behalve de weergegeven pagina die op dat moment verloren gaat nog drie min of meer verborgen, althans niet weergegeven, scherm-pagina's. Door handig gebruik van deze andere pagina's te maken kan dit soort rampen worden voorkomen.

In Sculptor wordt bij het geven van een PAINT-opdracht eerst snel een kopie van het tekenschermbereik gemaakt naar een van de andere pagina's. Omdat er geheugen-ruimte genoeg is, gebruikt het programma ook nog een van de 'pages' voor het maken van een extra veiligheidskopie.

### Back-up

Sculptor werkt tijdens het tekenen op pagina 1 van screen 5. De pagina's 2 en 3 worden als 'back-up' scherm gebruikt: pagina 2 voor de kopie in de paint-routine en pagina 3 voor de veiligheidskopie, die op elk willekeurig moment gemaakt kan worden.

Bij het geven van een paint-opdracht - met de p-toets - maakt Sculptor automatisch een kopie van pagina 1 naar pagina 2.



Door op de t-toets te drukken wordt het tekenschermbereik zoals dat er voor het paint-kommando uitzag weer teruggeladen. Mocht door een mislukte paint-opdracht een tekening 'beschadigd' raken, dan is er niets verloren.

Door de d-toets - staat voor dump - in te drukken wordt het huidige tekenschermbereik naar pagina 3 gekopieerd. Met de v-toets kan deze veiligheidskopie tot de werktekening gemaakt worden.

Page 0 kan niet als tekenschermbereik worden gebruikt, omdat dit geheugen-gebied beschikbaar moet blijven voor de tekstschermbereiken in screen 0.

### Wel of geen drive

Aan het begin van het programma worden eerst enige zaken geïnitialiseerd. Zo worden hier de funktietoetsen 1 tot en met 5 geactiveerd, de achtergrondkleur bepaald en enkele variabelen gezet die later als vlag gebruikt worden. Regel 100 bevat een hiertoe een handige PEEK naar een systeem-variabele.

In dat systeem-gebied houdt de computer allerlei lopende zaken bij zoals het scherm-type, de scherm-breedte, of er een diskdrive is aangesloten en zo ja, hoeveel. Op adres FFA7H staat te lezen of er een drive is aangesloten. Als deze variabele de waarde 201 - hexadecimal C9H - heeft dan is er geen diskdrive aangesloten. Wordt die waarde 201 gevonden,

dan wordt in het programma de vlag V1 op 0 gezet.

Later in het programma komt deze vlag handig van pas in de laad- en save-routines. Al naar gelang de waarde van deze vlag wordt het 'device' bepaald voor de BLOAD- en BSAVE-opdrachten. Deze vlag zorgt er ook voor dat er zonder diskdrive geen FILES-kommando kan worden gegeven.

In feite is de variabele op adres FFA7 een 'hook'-adres, dat bij het opstarten van de computer gezet wordt. Wordt er daarbij geen drive gevonden, dan krijgt deze systeem-variabele de waarde 201, hetgeen staat voor een return - in machinetaal tenminste - naar de ROM-BIOS.

Overigens bleek dat ook Sculptor - waar een van de redactieleden toch de nodige tijd op had zitten zweten - in eerste instantie niet geheel aan de hoge eisen van MSX Computer Magazine voldeed. In regel 75 kunt u wat kleine maar o zo belangrijke aanvullingen zien die het programma de 'finishing touch' gaven die het toch nog net even ontbeerde.

Het verschijnsel dat de programmeur op een gegeven moment door de bomen het bos niet meer ziet is maar al te bekend, vooral in de initialisatie wilt men wel eens een steekje laten vallen. Het controleren of het programma inderdaad op een MSX2 werkt is daar een uitstekend voorbeeld van. Zonder zo'n controle kunnen we er namelijk zeker van zijn

dat er de nodige klachten van lezers zullen binnenkomen, omdat het 'bij hen niet werkt...'

## De hoofdlus

De hoofdlus op zich is vrij kort en loopt van regel 180 tot regel 300. In deze lus 'kijkt' de functie STRIG voortdurend of de spatiebalk is ingedrukt. Is dit het geval dan wordt er met het PSET-kommando een puntje in de huidige tekenkleur gezet. Op mogelijke andere kommando's hoeft dan even niet gelet te worden. Deze kunnen worden 'geskippt'.

Hierdoor is de snelheid van de grafische cursor groter wanneer er getekend wordt dan wanneer er enkel over het scherm wordt gelopen. Door in regel 190 het GOTO-kommando weg te laten kan de teken-snelheid enigszins terug gebracht worden.

## Wrap-around

De tekenpijl en de grafische cursor zijn 'wrap-around'. Dit houdt in dat, wanneer we met de tekenpijl aan de ene kant van het scherm aflopen, we meteen daarop aan de tegenovergestelde zijde weer in beeld verschijnen. Voor alle acht richtingen is er hiervoor een aparte subroutine.

Het ON STICK-kommando 'scant' in elke lus de cursor-toetsen en laat het programma op basis van de gevonden waarde naar een van de subroutines springen die voor de besturing van de tekensprite en de grafische cursor zorgen.

Zo geeft cursor rechts de waarde 3 waarna het programma naar de bijbehorende subroutine springt die begint op regel 360. De waarde 6-cursor links onder - laat het programma naar de subroutine springen die begint op regel 440.

## Meer systeem-variabelen

De lus bevat nog een PEEK in het systeem-gebied. We willen de GRAPH-toets gebruiken om lijnen te wissen of, anders gezegd, in de achtergrondkleur te zetten. Een simpele manier om het ingedrukt zijn

van een speciale toets te detecteren is de ASCII waarde ervan te bekijken. Dit blijkt echter niet altijd mogelijk. De SHIFT-, de GRAPH- en de CODE-toetsen bijvoorbeeld hebben geen ASCII waarde. Het is daarom lastig te ontdekken of deze zijn ingedrukt.

Toch bestaat ook hiervoor een oplossing. In het systeemgebied staat namelijk een zogenaamde keyboard-matrix, waarin voor elke toets een bitje bestaat. Deze matrix heeft 11 rijen en 8 kolommen. Elke rij beslaat precies een byte. De kolommen worden gevormd door de afzonderlijke bits in zo'n byte. De wijze waarop de rijen en kolommen zijn georganiseerd hebben echter weinig van doen met de layout van het toetsenbord.

Het beginadres van deze systeembuffer is FBE5H. Op de zesde rij (adres FBEBH) treffen we de GRAPH-toets aan. Wordt er geen toets uit deze rij ingedrukt dan staan alle bitjes op 1. Deze byte heeft dan de waarde 255. Wordt een van de toetsen die in deze rij worden bijgehouden ingedrukt dan krijgt het korresponderende bitje de waarde nul. Door nu de waarde van deze byte te OR-en met 251 kan ontdekt worden of de GRAPH-toets is ingedrukt. Toegegeven, eenvoudig is het niet, er is kennis van logische bewerkingen voor nodig, maar het is een methode om er achter te komen of een GRAPH- of CODE-toets is ingedrukt.

We hopen met deze misschien wat specialistische uitweiding enkele vragen die we de afgelopen tijd hieromtrent hebben gekregen beantwoord te hebben.

We moeten hier echter wel een kanttekening bij plaatsen. De gebruikte adressen liggen weliswaar vast, maar er is een kleine kans dat de exakte inhoud ervan op sommige MSX-en anders is. In dat geval is het beter voor het wissen de ESC-toets te gebruiken, die aan zijn ASCII-waarde te herkennen is. Regel 260 moet er dan als volgt uitzien:

```
260 IF C$=27 THEN PRESET (X,Y),BC
```

Regel 270 moet in dat geval worden weggelaten.

## Schermtypes

Het wisselen van pagina gaat razend snel en levert geen enkel probleem op. De automatische back-up bij een Paint merkt u niet eens op.

Bij het wisselen van scherm-type zitten we echter opgezadeld met een lastig probleem. In een tekstscherm kunnen we geen grafische kommando's geven, en in een grafisch scherm kunnen we moeilijk tekst kwijt. Voor het bekijken van de directory met de FILES-opdracht moeten we zelfs in tekstmode zitten willen we deze lijst te zien krijgen, anders werkt het kommando gewoon niet.

De extra subroutines onder de funktietoetsen kunnen zowel vanuit het menu als vanuit het grafische scherm worden aangeroepen. Hierbij stuiten we op ons eerder genoemde probleem. Wordt er namelijk een bepaalde subroutine met een tekstscherm vanuit het grafische scherm opgeroepen, dan moet eerst SCREEN 0 geselecteerd worden.

Wordt zo'n subroutine echter vanuit het menu-scherm gebruikt, dan mag juist geen SCREEN 0-kommando gegeven worden. Deze zou het scherm immers wissen en we willen de keuze-mogelijkheden graag voor ogen houden.

Gelukkig komt ons hier al weer een systeem-variabele te hulp. De variabele op adres FCAFH bevat namelijk steeds het nummer van het scherm-type. Na SCREEN 0 is deze variabele 0, na SCREEN 1 is deze 1, enz. Door nu aan het begin van een subroutine te bepalen wat het huidige scherm-type is kan er al of niet een SCREEN-opdracht gegeven worden.

Wordt er vanuit het grafische scherm gesprongen (FCAFH bevat dan de waarde 5) dan wordt een SCREEN 0-opdracht gegeven. Was het scherm-type al 0 dan wordt het SCREEN-kommando niet verwerkt.

## Vlaggen

Bij het terugspringen uit een subroutine moet steeds bepaald worden of er terug naar het menu of terug naar het grafische scherm gekeerd moet worden. Dit geschiedt aan de hand van de vlag V2. Aan het begin van de menu-subroutine deze vlag op 1 gezet.

Staat deze vlag bij terugkeer op 1 dan vallen we terug in het menu-scherm. Wordt de routine vanuit het grafische scherm opgeroepen dan staat V2 op 0 en wordt eerst SCREEN 5 gekozen. Ook de vlaggen V3 en V4 zorgen ervoor dat er op de juiste wijze wordt teruggesprongen en het programma niet ergens blijft hangen.

## Wissen

Onder de F5-toets is de subroutine geplaatst die alle pagina's wist. Het CLS-kommando wist weliswaar het scherm, maar niet de sprites. Daarom is er ook een zogenaamde BIOS-call toegevoegd naar de routine in de ROM die alle sprites 'reset'. Deze actie moet voor elke pagina herhaald worden. De 4 pagina's die op scherm 5 mogelijk zijn worden namelijk als afzonderlijk schermen gezien door de MSX. De verschillende opdrachten - en zo ook het CLS-kommando - hebben alleen betrekking op de pagina die op dat moment actief is.

## Tot slot

Het staat een ieder vrij om nog enkele subroutines aan Sculptor toe te voegen. Een routine die het kleurenpalet gebruikt zou het programma niet mistaan. Wij zelf laten het hierbij.

Sculptor werkt, zoals het programma nu gepubliceerd is, met de spatiebalk en de cursor-toetsen. Wie het liever met de joystick wil besturen dient de STRIG(0)- en de STICK(0)-kommando's te vervangen door STRIG(1) en STICK(1).

Voor alle duidelijkheid moet in de tekst 'spatiebalk' dan nog gewijzigd worden in 'vuurknop'.

```

10 REM SCULPTOR
20 REM
30 REM MSX Computer Magazine
40 REM
50 REM Een MSX2 tekenprogramma
60 REM
70 ' INITIALISATIES *****
80 IF PEEK(&H2D)=0 THEN PRINT "programma werkt niet op MSX1!": STOP
90 SCREEN 0: WIDTH 80: KEY OFF: SCREEN 5: FOR I=1 TO 3: SET PAGE 1,I: CLS: NEXT
I
100 V1=1: V2=1: C=1: BC=4
110 DEFUSR1=&H69
120 IF PEEK(&HFFA7)=201 THEN DV$="cas:": V1=0 ELSE DV$="a:"
130 ON KEY GOSUB 540,780,1140,880,1330
140 FOR I=1 TO 5
150 KEY(I) ON
160 NEXT I
170 GOSUB 1230: ' tekenpijl
180 GOSUB 540: ' menu
190 ' DE HOOFDLUS *****
200 IF STRIG(0) THEN PSET (X,Y),C: GOTO 290
210 C$=INKEY$: IF C$<>" " THEN C$=CHR$(ASC(C$) OR 32): 'MAAK KLEINE LETTERS
220 IF C$="t" THEN GOSUB 850: 'HAAL VOORLAATSTE SCHERM TERUG
230 IF C$="v" THEN GOSUB 860: 'HAAL VEILIGHEIDSKOPIE OP
240 IF C$="d" THEN GOSUB 840: 'DUMP (VEILIGHEIDSKOPIE)
250 IF C$="p" THEN GOSUB 830: PAINT (X-1,Y),C,C: 'VUL VLAK
260 IF C$="s" THEN GOSUB 1020: 'SAVE SCHERM
270 GR=PEEK(&HFBEB) OR 251
280 IF GR=251 THEN PRESET(X,Y),BC
290 PUT SPRITE 0,(X,Y-1),1,0
300 ON STICK (0) GOSUB 330,350,380,400,430,450,480,500
310 GOTO 200
320 ' BESTURING EN WRAP AROUND *****
330 Y=Y-1: IF Y<0 THEN Y=211
340 RETURN
350 X=X+1: Y=Y-1: IF X>255 THEN X=0
360 IF Y<0 THEN Y=211
370 RETURN
380 X=X+1: IF X>255 THEN X=0
390 RETURN
400 X=X+1: Y=Y+1: IF X>255 THEN X=0
410 IF Y>211 THEN Y=0
420 RETURN
430 Y=Y+1: IF Y>211 THEN Y=0
440 RETURN
450 X=X-1: Y=Y+1: IF X<0 THEN X=255
460 IF Y>211 THEN Y=0
470 RETURN
480 X=X-1: IF X<0 THEN X=255
490 RETURN
500 X=X-1: Y=Y-1: IF X<0 THEN X=255
510 IF Y<0 THEN Y=211
520 RETURN
530 ' MENU *****
540 SCREEN 0
550 KEY(1) OFF: V2=1
560 PRINTTAB(24)"***** SCULPTOR *****"
570 PRINT
580 PRINT TAB(4) " tekenen met spatiebalk": PRINT
590 PRINT TAB(4) " wissen met graph-toets": PRINT
600 PRINT TAB(27) "FUNKTIE-TOETSEN": PRINT
610 PRINT TAB(4) "F1: menu" TAB(40) "F2: nieuwe kleur"
620 PRINT TAB(4) "F3: files" TAB(40) "F4: laad file"
630 PRINT TAB(4) "F5: alle schermen wissen"
640 PRINT
650 PRINT TAB(22) "MOGELIJKHEDEN OP TEKENSCHERM": PRINT
660 PRINT TAB(4) "p: vul vlak" TAB(40) "t: voorlaatste scherm terug"
670 PRINT TAB(4) "d: maak veiligheidskopie " TAB(40) "v: haal veiligheidskopie"
680 PRINT TAB(4) "s: save tekening"
690 PRINT
700 PRINT "spatiebalk voor grafisch scherm": PRINT
710 IF V3=1 THEN 730
720 IF NOT STRIG(0) THEN 710
730 V2=0: V3=0: V4=0

```

740 KEY(1) ON	208
750 SCREEN 5: SET PAGE 1,1	211
760 RETURN	205
770 ' TEKENKLEUR *****	0
780 IF PEEK(&HFCAF)=5 THEN SCREEN 0	34
790 INPUT "kleurkode (1-15)":C	36
800 IF V2=0 THEN SCREEN 5: SET PAGE 1,1 ELSE PRINT: PRINT "spatiebalk voor grafi sch scherm": PRINT	33
810 RETURN	196
820 ' KOPIEER-ROUTINES *****	0
830 COPY (0,0)-(255,211),1 TO (0,0),2: RETURN	18
840 COPY (0,0)-(255,211),1 TO (0,0),3: RETURN	57
850 COPY (0,0)-(255,211),2 TO (0,0),1: RETURN	11
860 COPY (0,0)-(255,211),3 TO (0,0),1: RETURN	39
870 ' LAAD SCHERM *****	0
880 IF PEEK(&HFCAF)=5 THEN SCREEN 0	35
890 V3=1: V4=1	0
900 PRINT	136
910 INPUT "naam te laden file";F\$	23
920 IF V1=1 THEN 970	26
930 PRINT	142
940 PRINT "recorder op afspelen?": PRINT	93
950 PRINT "spatiebalk om te starten": PRINT	146
960 IF NOT STRIG(0) THEN 960	230
970 SCREEN 5	163
980 SET PAGE 1,1	215
990 BLOAD DV\$+F\$,S	224
1000 RETURN	123
1010 ' SAVE SCHERM *****	0
1020 SCREEN 0	71
1030 PRINT	246
1040 INPUT "naam te save file";F\$	97
1050 IF V1=1 THEN 1100	192
1060 PRINT	255
1070 PRINT "recorder op opnemen?":PRINT	2
1080 PRINT "spatiebalk om te starten": PRINT	134
1090 IF NOT STRIG(0) THEN 1090	202
1100 SCREEN 5: SETPAGE 1,1	192
1110 BSAVE DV\$+F\$,0,&H76A0,S	158
1120 RETURN	131
1130 ' FILE-DIRECTORY *****	0
1140 IF PEEK(&HFCAF)=5 THEN SCREEN 0	142
1150 V3=1: PRINT	250
1160 IF V1=1 THEN FILES ELSE PRINT "geen diskdrive aangesloten"	138
1170 PRINT	4
1180 PRINT: PRINT "spatiebalk voor grafisch scherm, F4 voor laden"	121
1190 IF V4=1 THEN 1220	42
1200 IF NOT STRIG(0) THEN 1190	207
1210 IF V2=0 THEN SCREEN 5: SET PAGE 1,1	37
1220 RETURN	133
1230 ' TEKENPIJLTJE *****	0
1240 RESTORE 1300	63
1250 FOR K=1 TO 8	123
1260 READ S	97
1270 S\$=S\$+CHR\$(S)	123
1280 NEXT K	129
1290 SPRITE\$(0)=S\$	188
1300 DATA 224,192,160,16,8,4,2,1	24
1310 RETURN	132
1320 ' MAAK SCHERMEN SCHOON *****	0
1330 SCREEN 5	145
1340 FOR I=1 TO 3	17
1350 SET PAGE 1,I: CLS: X=USR1(0)	145
1360 NEXT I	103
1370 SET PAGE 1,1	145
1380 GOSUB 1240	121
1390 GOSUB 530	177
1400 RETURN	131

## Categorie: simulaties

# Ijsfabriek

Een van de aardigste programma-ideeen is toch altijd weer de simulatie, de nabootsing. Veel van de beste spelletjes zijn in feite computersimulaties, zoals vluchtnabootsers. Maar er zijn ook allerlei andere zaken die met behulp van de computer gesimuleerd kunnen worden.



Dit programma, ijsfabriek, is daar een prima voorbeeld van. Het is een economische simulatie, waarin men moet proberen om als directeur van een fabriek die ijsjes maakt het hoofd boven water te houden. Of liever nog, winst te maken, want zo werkt het in de werkelijke wereld nu eenmaal ook. Ijsfabriek is ons toegezonden door de heer H. Kloosterman uit Putten, die we bij deze daar nog hartelijk voor willen danken.

### Edukatief

Dergelijke simulatie-programma's zijn aan de ene kant als edukatief in te schatten. De speler leert immers omgaan met de wetten van vraag en aanbod, waarbij het juist inschatten van de markt heel belangrijk is. Zo zal men bedrogen uitkomen wanneer men

denkt in de winter evenveel ijsjes af te kunnen zetten als in de zomer.

Om echter alleen ijsjes te maken wanneer er vraag naar is blijkt ook al niet de juiste oplossing. Want dan zouden er veel machines gedurende de winter ongebruikt blijven staan, hetgeen een onzorgvuldig gebruik van produktiemiddelen is.

Kortom, ijsfabriek 'leert' een zeker inzicht aan, een juist inschatten van situaties.

Aan de andere kant is ijsfabriek ook zonder meer een leuk spelletje, waarbij men moet proberen de computer te slim af te zijn. Want ergens voelt het toch wel zo aan, in een bepaald opzicht is de computer de tegenstander in dit spel. Het feit dat die 'tegenstand' in feite niets anders is dan het toepassen van een

vastgelegde serie van regels - het 'model' van de ijsjes-konsumenten bijvoorbeeld - doet daar niets aan af.

### Spelregels

Op zich is ijsfabriek, als spel gezien, heel simpel. Men wordt verwelkomd met een soort telegram, waarin staat dat men zojuist directeur geworden is van een slecht lopende ijsfabriek. Blijkbaar heeft de een of andere oudoom u bedacht in zijn testament.

Als premie heeft de overheid daar een bedrag van f. 500,- bijgedaan, een soort starterspremie als het ware. Bovendien liggen er nog 1000 ijsjes in de koeling, en staat er een machine in de fabriek die maximaal 1200 ijsjes per maand kan maken.

Ijsjes maken kost, afhankelijk van het aantal dat er in een maand gemaakt moet worden, tussen de 10 en de 20 cent. Opslag van de voorraad kost 3 cent per ijsje per maand, de energiekosten zijn niet mals.

Op ieder gewenst moment kan men ijsmachines kopen, voor f. 750,- per stuk. Verkopen kan natuurlijk ook, maar dan leveren deze machines een stuk minder op.

### Extra informatie

Overigens kan men steeds door de CONTROL-STOP combinatie wat extra informatie opvragen. Zo kan men dan de spelregels nogmaals bekijken, of een overzicht van de voortgang tot dat moment. Een derde optie in dit menu is het op het scherm roepen van een overzicht van de

vorige ronden, waarmee bijvoorbeeld een leerkracht de resultaten van de leerlingen kan vergelijken.

Deze laatste keuze is door de schrijver speciaal afgeschermd, zodat de leerlingen niet zelf kunnen spieken. De gevraagde overzichtscode is de CONTROL-o, die ingetikt kan worden door de letter o tezamen met de CONTROL toets in te drukken. Een aardige suggestie lijkt ons om ijsfabriek te combineren met het screendump-programma nummer 8 van MSX Computer Magazine, zodat deze schermen ook afgedrukt kunnen worden.

De informatie op de voortgangs- en overzichts-schermen is overigens wat uitgebreider dan de standaardinformatie; zo kan men hier ook zaken vinden als de trendfactor, de reclamefactor en de gerealiseerde verhouding tussen vraag en aanbod. Al met al genoeg informatie om de verschillen tussen de spelers te kunnen analyseren.

### 24 maanden

Met deze fabriek is nu de opdracht om in de loop van vierentwintig maanden - twee jaar - zo veel mogelijk te verdienen. Daartoe kan men eens per maand opgeven hoeveel ijsjes er aangemaakt moeten worden, of men machines wil bijkopen of juist afstoten en of men al dan niet reclame wil maken in die maand. Die reclame mag niet minder dan f. 25,- kosten, maar ook niet meer dan f. 100,-. Onder of boven die bedragen gaan heeft geen enkele zin, de potentiële konsument heeft blijkbaar zo-

MAAND: JANUARI	max 1200 topscore	8933
(nog 23 maanden)	Er zijn verkocht	310 ijsjes
	Je voorraad is nu	1190 ijsjes
Wat wil je aan reclame besteden: 100		
-----		
Je bezit was	+	450.00
Vorige maand verkocht	+	124.00
		-----
magazijnkosten	+	35.70
Je bezit is	+	588.70

```

***** EINDSTAND IJSFABRIEK *****
Laatste maal verkocht 990 ijsjes

opbrengst daarvan + 395.00
verdiend in 2 jaar + 7923.00
verkoop van machines (1) + 500.00
verkoop voorraad + 108.00
NIEUWE TOPSCORE + 8933.00

**** Druk op RETURN ****
    
```

wel een minimum drempel voor reclame als een maximum, waarboven meer advertenties niets meer uit maken.

Iedere maand krijgt men een keurig overzichtje op het scherm over de bedrijfsresultaten van die maand, met zowel voorraden als de kas-situatie.

Op grond van die gegevens moet de directeur dan maar proberen te beslissen wat er te gebeuren staat, een echte managers-positie dus.

**Het spel**

Om het lastig te maken begint het spel in januari, een beroerde maand voor de ijs-industrie. Want - zo kunnen we u verzekeren na een aantal rondjes spelen - de ijs-verkoop is behoorlijk seizoensgebonden. Al besteed men de maximale reclame-bedragen in de maanden rond de jaarwisseling, de mensen kopen er geen ijsje meer om. De omzet is gewoon heel laag.

Nu lijkt het misschien verleidelijk om dan maar snel en veel te produceren voor de voorraad, want het mooie weer komt er toch weer aan, maar de opslagkosten verbieden dat. Drie cent per ijsje per maand - terwijl een ijsje maken maar 10 cent kost als er genoeg tegelijkertijd gemaakt worden - hakt er lelijk in.

Het is dan ook zaak om op precies het juiste moment de productie aan te zwengelen, eventueel zelfs door zich voor nieuwe machines in de schulden te steken. En daarbij moet de directeur vooral niet vergeten dat ergens na augustus de ver-

koop weer in elkaar zakt. Wat dat betreft hebben we de indruk dat het programma een soort weersverwachting gebruikt, sommige septembermaanden zijn ware topmaanden, terwijl een andere keer er men in september de ijsjes niet aan de straatstenen kwijt kan. We mogen wel zeggen dat vooral die maand een kwestie van gokken is. Misgokken houdt echter wel in dat de fabriek met een grote en kostbare voorraad de winter in gaat.

**Naar!**

Ronduit vervelend is ook het zinnetje 'U had er nog meer kunnen verkopen', dat in de zomermaanden regelmatig op het scherm verschijnt. Want juist in de zomer is de markt zo willig dat het ijs niet aan te slepen valt. Al zet u tien machines neer - gelukkig is de fabriek daar blijkbaar groot genoeg voor - er valt niet tegenop te produceren.

Dat houdt in dat men als wakere ondernemer iedere zomer weer de winstkansen ziet passeren, tenzij men in het vroege voorjaar vast een voorraadje gemaakt had. Maar ook dat is riskant natuurlijk, want als de lente op zich laat wachten liggen al die ijsjes maar in het magazijn...

**Simulatie**

Zoals al gesteld, ijsfabriek is slechts ten dele een spel. Als spel is het onderhoudend genoeg trouwens, hoewel er geen geluid of graphics bij komen kijken.

Maar ijsfabriek heeft ook een zekere edukatieve waarde. Dit programma leert namelijk een

soort management aan, het maken van de juiste keuzes. Want hoewel de simulatie beperkt is - zo zult u geen loonissen van de vakbond op het directiebureau krijgen - is het zeer zeker een verantwoorde simulatie.

Een aardig voorbeeld daarvan is het feit dat ook rente over het kapitaal en verlies door backorders bij deelbenutting in het programma meegenomen worden. Op het eerste gezicht lijkt dat misschien wat overdreven, maar voor het uiteindelijk levenschte karakter van de simulatie zijn juist dergelijke details heel belangrijk. Anders zou de voortgang maar al te snel heel erg voorstelbaar worden.

Juist het extra - afgeschermd - overzicht, waarmee men de

resultaten van diverse spelers eens kan bekijken, kenschetst ijsfabriek als een echt edukatief programma. Men zo bij wijze van spreken het programma gewoon kunnen laden en er daarna een hele klas leerlingen op los laten. Nadat iedereen 'gespeeld' heeft, kan de leraar of lerares simpel opvragen hoe de diverse leerlingen het er vanaf gebracht hebben, om daarna die resultaten eens door te spreken.

Al met al, naar onze mening, een prima voorbeeld van een speels programma met heel duidelijke edukatieve waarde. Leuk voor zowel de speler als de leerkracht.

Petje af voor de programmeur, de heer H. Kloosterman!

```

10 REM IJSFAB 0
20 REM 0
30 REM MSX Computer Magazine 0
40 REM 0
50 REM ingezonden door H. Kloosterman
   Putten 0
60 REM 0
70 ' INITIALISATIE ***** 0
80 CLS:COLOR 1,15,15:KEY OFF:SCREEN 0 96
   :WIDTH 40:NAS="MSX":TP=2500:L=0
90 DIM NAS(30),K(30),M(30),VO(30),TV( 26
   30),TA(30),TK(30),TG(30),VK(30),TR(30)
100 POKE &HFB00,1:DEFUSR=&H156 210
110 ON STOP GOSUB 2770:STOP ON 20
120 ' STARTSCHERM ***** 0
130 M=1:VO=1000:K=500:N=0:L=L+1:TA=0:
   TK=0:TG=0:TF=0:RF=0:RB=0:TR=0 28
140 LOCATE 9,5:PRINT"MSX Computer Mag
   azine" 12
150 LOCATE 9,7:PRINT"*****
   *****" 226
160 LOCATE 9,8:PRINT"*
   *" 50
170 LOCATE 9,9:PRINT"* IJSFABRIEK
   *" 240
180 LOCATE 9,10:PRINT"*
   *" 99
190 LOCATE 9,11:PRINT"*****
   *****" 242
200 LOCATE 10,15:PRINT"(c) H. Klooste
   rman" 14
210 LOCATE 13,17:PRINT"oktober 1986" 55
220 FOR WA=0 TO 1700:NEXT:R=USR(R) 124
230 LOCATE 5,23:PRINT"Wil je eerst in
   formatie (j/n)?":Z$=INPUT$(1):IF Z$=
   "j" OR Z$="J" THEN GOSUB 2220:GOTO 260 175
240 IF Z$="n" OR Z$="N" THEN 260 ELSE
   230 12
250 ' SPELBEGIN ***** 0
260 CLS:R=USR(R) 97
270 LOCATE 3,3:PRINT"Wil je tijdens h
   et spel":LOCATE 3,5:PRINT"reclame mak
   en (j/n)?" 252
280 Z$=INKEY$:IF Z$="" THEN 270 191
    
```

```

290 IF Z$="j" OR Z$="J" THEN RE=1:GOT
0 310 ELSE IF Z$="n" OR Z$="N" THEN R
E=0:GOTO 310
300 GOTO 270
310 CLS:R=USR(R)
320 LOCATE 3,5:PRINT"wat is de voorna
am van de nieuwe":PRINT:PRINT" dire
kteur:";
330 INPUT NA$(L)
340 IF NA$(L)=" " THEN 320
350 H=ASC(MID$(NA$(L),1,1)):IF H>96 A
ND H<123 THEN H=H-32
360 MID$(NA$(L),1,1)=CHR$(H):'le lett
er wordt hoofdletter
370 CLS
380 RESTORE
390 PRINT"Je voorraad aan ijsjes is:
1000"
400 PRINT:PRINT"Je bezit is nu: f 50
0"
410 GOSUB 1410
420 K=K+A*.1*(A>=500 AND A<=1000)+A*.
2*(A<500 OR A>1000)
430 ' MAANDLUS *****
440 CLS
450 READ MA$,T:IF N=11 THEN RESTORE
460 TF=TF+.06:IF RE=1 THEN RF=RF/2:IF
RF<.1 THEN RF=0:'invloed reclame+tre
nd
470 N=N+1:IF N=23 THEN GOSUB 1940:GOT
0 490
480 IF N=24 THEN 560
490 PRINT"MAAND:";MA$;:LOCATE16:PRINT
"max";M*1200;:LOCATE 25:PRINT"topscor
e:";TP
500 NA=LEN(NA$):LOCATE 38-NA,1:PRINT"
(";NA$;)"
510 LOCATE 6,1:PRINTSTRING$(LEN(MA$),
195)
520 IF N<23 THEN LOCATE 4,2:PRINT"(no
g";24-N;"maanden)"
530 FOR WA=0 TO 800:NEXT
540 ON T GOSUB 1790,1820,1850,1880,19
10
550 VR=VR+100*INT((TF*VR+RF*VR)/100):
IF VR>15000! THEN VR=15000!:'vraag ko
rrigeren met trend en reclame
560 VV=VO+A:' nieuwe voorraad
570 IF VR>VV THEN VS=VR-VV:VO=0:VK=VV
:TEL=1 ELSE VK=VR:TEL=0:VO=VV-VR:'vra
ag groter dan aanbod
580 TG=TG+VR:TK=TK+VK
590 IF N=24 THEN 790
600 GOSUB 1530:'aantal verkocht
610 IF TEL=1 THEN GOSUB 1490:GOTO 630
:'te weinig verkocht
620 GOSUB 1090:'wachtlus
630 GOSUB 1570:'nieuwe voorraad+koste
noverzicht
640 IF RE=1 THEN GOSUB 1270:'reclame
maken?
650 GOSUB 1340:' machines kopen/verko
pen?
660 GOSUB 1410:'input aanmaak
670 ' KOSTENBEREKENINGEN *****
680 IF M=1 THEN KA=ABS(A*.1*(A>=500 A
ND A<=1000)+A*.2*(A<500 OR A>1000)):G
OTO 710:'gedifferentieerde aanmaakkos
ten
690 IF M=2 THEN KA=ABS(A*.1*(A>=400 A
ND A<=2000)+A*.25*(A<400 OR A>2000)):
GOTO 710:'gedifferentieerde aanmaakko
sten
700 KA=ABS(A*.1*(A>=400 AND A<=3200)+

```

102  
85  
88  
88  
100  
109  
100  
202  
13  
194  
27  
196  
10  
255  
0  
8  
214  
23  
48  
165  
7  
14  
112  
17  
190  
232  
167  
33  
123  
174  
5  
1  
137  
101  
12  
146  
221  
238  
0  
63  
95

```

A*.28*(A<400 OR A>3200):'gedifferent
ieerde aanmaakkosten m>2
710 KO=VO*.03:K=K+VK*.4-KA-KO:'nieuw
kapitaal
720 GOSUB 1690:'laatste stuk van deze
serie
730 GOTO 440
750 ' MAANDEN EN FAKTOREN *****
760 DATA JANUARI,1,FEBRUARI,1,MAART,2
,APRIL,2,MEI,3,JUNI,4,JULI,4,AUGUSTUS
,4,SEPTEMBER,3,OKTOBER,2,NOVEMBER,1,D
ECEMBER,5
780 ' EINDSTAND *****
790 FOR X=0 TO 37
800 LOCATE 0,0
810 PRINTRIGHT$(" ***** EINDSTA
ND IJSFABRIEK *****",X)
820 BEEP
830 FOR WA=0 TO 25:NEXT
840 NEXT X
850 PRINT:PRINT" Laatste maal ver
kocht:";VK;"ijsjes"
860 PRINT:PRINT:PRINT:PRINT"
opbrengst daarvan: f";:PRINTUSING"##
###.##";VK*.4
870 PRINT" verdiend in 2 jaar:
f";:PRINTUSING"#####.##";K
880 PRINT"verkoop van machines (";M;"
): f";:PRINTUSING"#####.##";M*500
890 PRINT" verkoop voorraad:
f";:PRINTUSING"#####.##";VO*.15
900 PRINT"
+
910 TV=VK*.4+K+M*500+VO*.15
920 K(L)=K:M(L)=M:VO(L)=VO:TV(L)=TV:T
A(L)=TA:TK(L)=TK:TG(L)=TG:VK(L)=VK:TR
(L)=TR
930 IF TV>TP THEN GOSUB 2070:GOTO 130
:'nieuwe topscore
940 PRINT" TOTAAL VERDIEND:
f";:PRINTUSING"#####.##";TV
950 GOSUB1110
960 GOTO 130
970 ' VERLIES TE HOOG *****
980 CLS:LOCATE 13,4:PRINT"JAMMER!!!!"
990 LOCATE 0,9:PRINT"Je verlies is op
gelopen tot";ABS(K);"gulden."
1000 PRINT:PRINT"Dat wordt toch te ge
k!"
1010 PRINT:PRINT:PRINT"Hopelijk gaat
ijs eten je beter af....."
1020 FOR WA=0 TO 3500:NEXT
1030 CLS:GOTO 790
1060 ' KORT WACHTEN *****
1070 FOR WA=0 TO 400:NEXT:RETURN
1080 ' LANG WACHTEN *****
1090 FOR WA=0 TO 1500:NEXT:RETURN
1100 ' RETURN VRAGEN *****
1110 R=USR(R)
1120 LOCATE 8,22:PRINT"#### DRUK OP R
ETURN ####"
1130 IF INKEY$<>CHR$(13) THEN 1120
1140 RETURN
1150 ' TEVEEL AANMAKEN *****
1160 BEEP:LOCATE 0,8:PRINT"Dat kan ni
et; je kunt maximaal maar ijsje
s per maand maken!"
1170 FOR X=1 TO 8
1180 LOCATE 35,8:PRINT M*1200;
1190 FOR WA=0 TO 250:NEXT
1200 LOCATE 36,8:PRINTSTRING$(4,32);
1210 FOR WA=0 TO 200:NEXT
1220 NEXT

```

0  
214  
153  
82  
0  
111  
0  
134  
205  
39  
167  
40  
108  
248  
184  
136  
192  
17  
167  
212  
36  
165  
116  
240  
49  
0  
111  
96  
98  
82  
8  
169  
0  
159  
0  
81  
0  
137  
15  
182  
137  
0  
222  
1  
235  
99  
252  
233  
246



```

1230 LOCATE 0,8:PRINTSTRING$(75,32)
1240 LOCATE 28,7:PRINTSTRING$(5,32)
1250 RETURN
1260 ' REKLAME-BUDGET *****
1270 RB$="":R=USR(R)
1280 LOCATE 0,6:LINEINPUT"Wat wil je
aan reclame besteden:";RB$
1290 RB=VAL(RB$):IF RB=0 THEN 1310 EL
SE IF RB>100 THEN LOCATE 32,6:PRINTST
RING$(10,32):GOTO 1280
1300 RF=RF+RB*.04:K=K-RB:TR=TR+RB
1310 LOCATE0,6:PRINTSTRING$(40,32)
1320 RETURN
1330 ' MACHINES VERHANDELEN *****
1340 Z$="":R=USR(R):MMS$=""
1350 LOCATE 0,6:PRINT"Machines kopen
(=k), verkopen (=v) of niets doen (
=n):";Z$=INPUT$(1)
1360 IFZ$="k"ORZ$="K"THENLOCATE0,6:PR
INTSTRING$(80,32):LOCATE0,6:PRINT"Hoe
veel machines wil je kopen (kost
f 750,-):";:R=USR(R):MMS=INPUT$(1):MM
=VAL(MMS):IFMM>3THEN1360ELSEK=K-MM*75
0:M=M+MM:LOCATE19,0:PRINTM*1200:GOTO
1380
1370 IF(Z$="v"ORZ$="V")ANDM>1THENLOCA
TE0,6:PRINTSTRING$(80,32):LOCATE0,6:P
RINT"Hoeveel machines wil je verkopen
(opbrengst f 375,-):";:R=USR(
R):MMS=INPUT$(1):MM=VAL(MMS):IF MM>=M
THEN1370 ELSE K=K+MM*375:M=M-MM:LOCA
TE19,0:PRINTM*1200
1380 LOCATE 0,6:PRINTSTRING$(80,32)
1390 RETURN
1400 ' AANTAL IJSJES AANMAKEN *****
1410 AS$="":R=USR(R)
1420 LOCATE 0,7:LINEINPUT"Hoeveel ijs
jes wil je maken:";AS$
1430 IF AS$="" THEN 1420 ELSE A=VAL(AS$
)
1440 IF A>1200*M THEN GOSUB 1160:GOTO
1410
1450 IF N>0 THEN LOCATE 0,4:PRINT"Je
voorraad wordt nu:";:PRINTUSING"####"
;VO+A;:PRINT" ijsjes"
1460 TA=TA+A
1470 RETURN
1480 ' TE WEINIG IJSJES *****
1490 BEEP:LOCATE 0,5:PRINT"Je had er
nog";VS;"meer kunnen verkopen als
je voorraad groter was geweest..."
1500 FOR WA=0 TO 2600:NEXT
1510 LOCATE0,5:PRINTSTRING$(80,32)
1520 RETURN
1530 ' VERKOCHTE IJSJES *****
1540 LOCATE 4,3:PRINT"Er zijn verkoch
t:";:PRINTUSING"####";VK;:PRINT" ijsj
es"
1550 RETURN
1560 ' BEZIT & KAPITAAL *****
1570 LOCATE 3,4:PRINT"Je voorraad is
nu:";:PRINTUSING"####";VO;:PRINT" ijs
jes"
1580 GOSUB 1070
1590 LOCATE 0,10:PRINTSTRING$(40,45)
1600 LOCATE 9,11:PRINT"Je bezit was:
f";:PRINTUSING"####.##";K
1610 LOCATE 0,12:PRINT"Vorige maand v
erkocht: f";:PRINTUSING"####.##";VK*
.4
1620 LOCATE 25,13:PRINT" + "
1630 LOCATE 23,14:PRINT" f";:PRINTUSIN
G"####.##";K+VK*.4

```

```

105
168
142
0
82
29
236
247
56
135
0
211
63
14
185
223
156
0
5
188
167
110
245
165
152
0
234
13
162
139
0
3
148
0
187
141
182
195
154
191
101

```

```

1640 LOCATE 7,15:PRINT"magazijnkosten
: f";:PRINTUSING"####.##";VO*.03
1650 LOCATE 25,16:PRINT" - "
1660 LOCATE 10,17:PRINT"Je bezit is:
f";:PRINTUSING"####.##";K+VK*.4-VO*.
03
1670 RETURN
1680 ' KOSTEN & RENDE *****
1690 LOCATE 0,18:PRINT"reclame/aanmaa
k/mach.: f";:PRINTUSING"####.##";KA+
RB
1700 LOCATE 25,19:PRINT" - "
1710 LOCATE 4,20:PRINT"Je bezit wordt
nu: f";:PRINTUSING"####.##";K
1720 GOSUB 1110
1730 K=INT(1.05*K):'5% rente over kap
itaal
1740 IF VR>VV THEN TF=TF-ABS(TF*VS/VK
*9000):IF TF<-.6 THEN TF=-.6:'verlies
door backorders bij deelbenutting+gr
ote vraag
1750 IF K<-600 AND M>1 THEN GOSUB 198
0:GOTO 1770
1760 IF K<-1200 THEN RETURN 980
1770 RETURN
1780 ' VRAAG BEREKENEN *****
1785 '
1790 VR=10*INT(RND(-TIME)*35+15)
1800 RETURN
1810 '
1820 VR=100*INT(RND(-TIME)*8+4)
1830 RETURN
1840 '
1850 VR=100*INT(RND(-TIME)*11+9)
1860 RETURN
1870 '
1880 VR=100*INT(RND(-TIME)*75+15)
1890 RETURN
1900 '
1910 VR=12*INT(RND(-TIME)*35+15)
1920 RETURN
1930 ' LAATSTE MAAND *****
1940 LOCATE 0,5:PRINT"PAS OP!! Je he
bt nog een maand, dus probeer ALLE
S kwijt te raken..."
1950 FOR WA=0 TO 2000: NEXT
1960 CLS:RETURN
1970 ' VERKOPEN? *****
1980 CLS:LOCATE 0,5:PRINT"Het gaat ni
et goed!!"
1990 PRINT:PRINT:PRINT"Je verlies is
nu f";ABS(K);"gulden..."
2000 PRINT:PRINT"Je kunt een van je m
achines verkopen voor f 500 gulden
"
2010 PRINT:PRINT"Antwoord met j of n"
:R=USR(R)
2020 Z$=INKEY$:IF Z$=""THEN 2020
2030 IF Z$="j" OR Z$="J" THEN M=M-1:K
=K+500:RETURN
2040 IF Z$="n" OR Z$="N" THEN RETURN
2050 GOTO 2020
2060 ' NIEUWE TOPSCORE *****
2070 FOR X=1 TO 9
2080 LOCATE 11,11:PRINTSTRING$(30,32)
2090 FOR WA=0 TO 300:NEXT
2100 LOCATE 11,11:PRINT"NIEUWE TOPSCO
RE: f";:PRINTUSING"####.##";TV
2110 FOR WA=0 TO 300:NEXT
2120 NEXT X
2130 GOSUB 1110:'return drukken
2140 CLS:LOCATE 7,2:PRINT"Oude topsco
re:";TP;" (";NA$;)"

```

```

159
61
176
156
0
66
99
144
56
69
121
63
59
158
0
0
70
139
0
244
148
0
233
157
0
223
166
0
70
147
0
55
209
103
0
212
235
77
94
104
149
196
155
0
17
96
86
19
64
253
235
141

```

```

2150 LOCATE5,3:PRINT"Nieuwe topscore:
":INT(TV) 39
2160 NA$=NA$(L):LOCATE 7,7:PRINT"Gesc
oord door: ";NA$;" !!" 248
2170 GOSUB 1110 60
2180 TP=INT(TV):'nieuwe topscore afro
nden 78
2190 RETURN 153
2210 ' INFO ***** 0
2220 WIDTH 37:CLS 100
2230 FOR X=0 TO 36 99
2240 LOCATE 0,0 23
2250 PRINTRIGHT$( "***** GEFEL
ICITEERD!! *****" X):BEEP 126
2260 FOR WA=0 TO 30:NEXT 25
2270 NEXT X 14
2280 WIDTH 37:PRINT:PRINT"DOOR EEN ER
FENIS BEN JE DIREKTEUR GEWORDEN VA
N EEN SLECHT DRAAIENDE I
JSFABRIEK." 57
2290 PRINT:PRINT"Je krijgt 24 maanden
de tijd om iets goeds van deze fabri
ek te maken en veel geld te verdien
en.....OF TE VERLIEZEN natu
urlijk." 178
2300 PRINT:PRINT"De fabriek had nog 1
000 ijsjes in voorraad in de diepv
rieziers liggen." 101
2310 PRINT:PRINT"En als premie krijg
je van de overheid f 500 gulden o
m mee te starten." 162
2320 GOSUB 1110 49
2330 CLS 207
2340 PRINT"WAT JE EERST MOET WETEN":P
RINT:PRINT 61
2350 PRINT"Je moet dus ijsjes maken." 98
2360 PRINT"Die kosten 10 tot 20 cent
per stuk, afhankelijk van het aantal
." 215
2370 PRINT:PRINT"Je begint met een ma
chine." 240
2380 PRINT"Die kan maximaal 1200 ijsj
es per maand maken." 113
2390 GOSUB 1110 70
2400 CLS:LOCATE 0,4 103
2410 PRINT"In de winter verkoop je mi
nder dan inde zomer." 45
2420 PRINT"Daar moet je dus rekening
mee houden als je ijsjes maakt..." 101
2430 PRINT:PRINT"De voorraad ijsjes k
ost ook geld!" 247
2440 PRINT"Aan diepvrieskosten betaal
je daarvoor 3 cent per ijsje per
maand." 174
2450 GOSUB 1110 60
2460 CLS:LOCATE 0,5 140
2470 PRINT"Als je wilt kun je machine
s kopen (kost f 750,-) of verkopen
(opbrengst375 gulden)." 109
2480 PRINT:PRINT"Elke machine kan dus
maximaal 1200 ijsjes maken." 159
2490 PRINT:PRINT"Aan het eind van de
24 maanden (2 jaar) krijg je ee
n overzicht hoeveel je totaal verdi
end hebt." 202
2500 PRINT:PRINT:PRINT"Je begint in d
e maand JANUARI." 187
2510 GOSUB 1110 50
2520 CLS:LOCATE 0,5 130
2530 PRINT"Tijdens het spel kun je ee
n deel van je geld aan reclame bested
en." 208
2540 PRINT:PRINT"Hierdoor zul je meer
ijsjes kunnen verkopen." 243

```

```

2550 PRINT"Minder dan 25 gulden per k
eer heeft geen zin en meer dan 100 g
ulden ook niet!" 110
2560 PRINT:PRINT"Je hoeft zeker niet
elke maand rekla-me te maken; het zal
je steeds opnieuw gevraagd wor
den." 22
2570 GOSUB 1110 68
2580 CLS:LOCATE3,5 51
2590 PRINT"WORD JIJ DE NIEUWE TOPSCOR
ER???:PRINT 26
2600 LOCATE 3,6:PRINTSTRING$(31,"-") 114
2610 PRINT:PRINT NA$;" heeft een scor
e behaald van: ";PRINT TP;"gulden na 2
4 maanden....." 174
2620 GOSUB 1110 55
2630 CLS 213
2640 PRINT"WAT JE TOT SLOT MOET WETEN
" 142
2650 PRINT:PRINT:PRINT"Boven in beeld
zie je steeds:" 21
2660 PRINT:PRINT" - de maand" 224
2670 PRINT:PRINT" - max. aantal ijs
jes dat je kan maken" 240
2680 PRINT:PRINT" - topscore en de
naam" 158
2690 PRINT:PRINT:PRINT"Je kunt deze i
nformatie altijd weer oproepen met d
e toetsen ' CTRL+STOP'" 3
2700 PRINT:PRINT"Je hebt dus 24 maand
en de tijd..." 89
2710 LOCATE 6,18:PRINT"VEEL SUCCES, D
IREKTEUR!!" 106
2720 LOCATE6,19:PRINTSTRING$(22,195) 187
2730 GOSUB 1110 60
2740 WIDTH 40 201
2750 RETURN 153
2760 ' ON STOP ROUTINE ***** 0
2770 CLS:PRINT" KIES VOOR:" 128
2780 LOCATE5,3:PRINT"Overzicht (o):" 217
2790 LOCATE4,4:PRINT"Instruktie (i):" 53
2800 LOCATE0,5:PRINT"Info voortgang (
v):" 162
2810 Z$=INKEY$:IF Z$=""THEN 2810 70
2820 IF Z$="i" OR Z$="I" THEN 2280 38
2830 IF Z$="o" OR Z$="O" THEN 2870 160
2840 IF Z$="v" OR Z$="V" THEN 3150 156
2850 GOTO 2770 64
2860 ' OVERZICHT ***** 0
2870 R=USR(R):CLS:PRINT"OVERZICHTKODE
:" 144
2880 KO$=INPUT$(1) 118
2890 IF KO$<>CHR$(15) THEN CLS:RETURN 97
2900 I=1:COLOR 1,7,7 227
2910 CLS 213
2920 LOCATE 8,0:PRINT"OVERZICHT RESUL
TATEN":LOCATE 0,3 75
2930 PRINT"Naam: ";NA$(I):PRINT:PRINT 84
2940 PRINT" verkoop laatste maand
: f";:PRINTUSING"#####.##";VK(I)*.4 4
2950 PRINT" verdiend in 2 jaar
: f";:PRINTUSING"#####.##";K(I) 34
2960 PRINT"verkoop van machines (";M(
I);"): f";:PRINTUSING"#####.##";M(I)*
500 5
2970 PRINT" verkoop voorraad
: f";:PRINTUSING"#####.##";VO(I)*.15 5
2980 PRINT" + 178
2990 PRINT" TOTAAL VERDIEND
: f";:PRINTUSING"#####.##";TV(I) 86
3000 PRINT:PRINT:PRINT" Totale vra
ag in 24 maanden: ";:PRINTUSING"#####
";TG(I) 172

```

3010 PRINT " Totaal aangem	60
aakt: ";PRINTUSING "#####";TA(I)	
3020 PRINT " Totaal verk	218
ocht: ";PRINTUSING "#####";TK(I)	
3030 PRINT " Besteed a	63
an reclame: ";PRINTUSING "#####";TR(I	
)	
3040 IF TA(I)>0 THEN PRINT:PRINT "	
TOTALE VRAAG:AANGEMAAKT="";PRINTUS	
ING "##.##";TG(I)/TA(I)	107
3050 I=I+1	168
3060 IF I<=L-1 THEN GOSUB 1110:GOTO 2	241
910	
3070 LOCATE 4,22:PRINT "nog een keer b	236
ekijken (j/n)?"	
3080 Z\$=INPUT\$(1)	1
3090 IF Z\$="j" OR Z\$="J" THEN 2900	40
3100 IF Z\$="n" OR Z\$="N" THEN 3120	0
3110 GOTO 3070	222
3120 COLOR 1,15,15:CLS	245
3130 RETURN	136
3140 ' VOORTGANG *****	0
3150 CLS	210
3160 LOCATE 8,0:PRINT "VOORTGANGS-INFO	94
RMATIE "	
3170 LOCATE 0,3:PRINT "MAAND: ";MA\$;"	51
(NOG";24-N;"MAANDEN)":PRINT:PRINT	
3180 PRINT " Aantal machines: ";M	135
3190 PRINT " Maximaal";M*1200;"ijsj	99
es per maand":PRINT	
3200 PRINT " Kapitaal: ";INT(	192
K);"gulden":PRINT	
3210 PRINT " Totale vraag: ";TG;"	164
ijsjes "	
3220 PRINT " Totaal verkocht: ";TK;"	61
ijsjes "	
3230 PRINT "Besteed aan reclame: ";TR;"	165
gulden":PRINT	
3240 TF!=TF:PRINT " Trendfaktor	122
:";TF!	
3250 RF!=RF:PRINT " Reklamefaktor	65
:";RF!	
3260 GOSUB 1110	60
3270 CLS:RETURN	94

# TELEFONISCHE HULPDIENST

Een unieke service van MSX Computer Magazine.

Als enige onder de Nederlandse MSX-bladen bieden wij een telefonische hulpdienst.

Iedere dinsdag kunt u rechtstreeks naar de redactie bellen met uw technische vragen.

**Elke dinsdag  
vanaf 4 uur 's middags  
tot 7 uur 's avonds  
op telefoonnummer  
020-931263**

Let wel, dit geldt alleen voor vragen die betrekking hebben op de inhoud van MSX Computer Magazine!

Voor alle andere zaken, zoals de (abonnemen-ten)administratie of de cassette-service moet u **020-657884** bellen, dagelijks tijdens kantooruren bereikbaar.

## SPELREGELS

Natuurlijk zijn er wel een paar spelregels aan ons telefonische vragenuurtje verbonden.

zet blijven en kunnen andere lezers ons niet meer bereiken.

Zo is het niet mogelijk om op andere momenten naar de redactie te bellen voor technische problemen. Alleen op dinsdag, tussen 1600 en 1900 uur, kunnen we uw vragen beantwoorden. Maar dan bent u er ook van verzekerd dat er een of meer redactieleden aanwezig zijn.

Stel uw vragen zo kort en bondig mogelijk, probeer de lijn zo kort mogelijk bezet te houden. Het is altijd razend druk gedurende het vragenuurtje en we willen zoveel mogelijk lezers kunnen helpen. Zorg ervoor dat u eventuele listings etcetera bij de hand hebt en leg pen en papier gereed.

Bovendien zal het niet mogelijk zijn om willekeurig iedere vraag telefonisch af te handelen. Het kan voorkomen dat vragen zo ingewikkeld en/of specialistisch zijn dat we u alsnog moeten verzoeken om ze schriftelijk in te dienen. Anders zou het telefoonnummer te lang be-

Vragen over programma's die in andere bladen verschenen zijn kunnen we tot onze spijt niet beantwoorden.

En, tenslotte, wordt niet boos als het even wat moeite kost om ons te bereiken. Als we in gesprek zijn, dan is het om iemand anders ook te helpen.

CATEGORIE: UTILITIES

# Koppie

Een MSX2 met ingebouwde diskdrive is een mooi bezit. Alleen het kopiëren van disks, dat blijft een crime. Ondanks de macht aan geheugen die zo'n apparaat bezit, gaat dat kopiëren verdraaid traag. De ingebouwde COPY-opdracht vertikt het nu eenmaal om al dat geheugen te gebruiken.

Onder MSX-DOS leest de COPY iedere keer slechts zo'n 55K aan gegevens in, voordat men de disk mag wisselen. Als we ons even bedenken dat een dubbelzijdige disk 720K kan bevatten, dan is het plaatje wel duidelijk. Wie geen MSX-DOS gebruikt kan al helemaal aan een carrière als disk-jockey beginnen. Het Basic COPY-

kommando werkt op zich prima, maar dan wel met hele kleine stukjes tegelijk. Tientallen keren diskettes verwisselen is eerder regel dan uitzondering. Met Koppie zijn al die problemen gelukkig de wereld uit, althans voor MSX2 bezitters. Het programma leest per keer 180K in, zodat een 720K disk in slechts vier keer kan wor-

den gekopieerd. Bovendien biedt Koppie nog een groot voordeel; het is namelijk een sector-kopieerder. Dat betekent in de praktijk dat Koppie een volledige disk overzet, niet alleen maar de bestanden. Waar DOS en Basic slechts de in de directory genoemde bestanden kopiëren, daar maakt Koppie een letterlijke kopie van uw disk. Inclusief eventuele extra informatie, zoals gewiste bestanden. Als u op een disk met problemen zit, kunt u Koppie gebruiken om eerst een kopietje te maken, alvorens u met een disk-monitor of iets dergelijks gaat proberen om de zaak te herstellen. Overigens, Koppie is géén breekijzer voor kraakpraktijken, de meeste beschermde programma's worden niet door Koppie gekopieerd. Een aardige extra: Koppie laat tijdens het ko-

piëren zien of er 'rotte' sectoren op een disk aanwezig zijn. Mocht dat het geval wezen, stel die diskette dan zo snel mogelijk buiten gebruik. Na opnieuw formatteren kan zo'n disk weer veilig zijn. Koppie gebruiken is de eenvoud zelf. De Basic-listing maakt namelijk het eigenlijke - machinetaal - Koppie-programma, wat u daarna naar keuze op een diskette of een cassette kan wegschrijven. Als u dat ML-programma opstartt rolt de hele gebruiksaanwijzing voorbij in een klein venstertje. Een kind kan de was doen! Overigens, als Koppie eenmaal begint te kopiëren, dan ziet u allerlei patronen op het scherm verschijnen. Dat lijkt vreemd, maar in feite ziet u daar een deel van uw diskette-inhoud. Koppie gebruikt namelijk het Video-RAM ook als opslag.

```

10 REM KOPPIE
20 REM versie 1.1
30 REM
40 REM MSX Computer Magazine
50 REM
60 REM door R.W.L.
70 REM
80 ' !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
90 ' De regelnummers van de DATA-regels mogen NIET veranderd worden, in verband
met de interne checksum van KOPPIE!
100 '
110 ' KOPPIE is alleen voor MSX2 geschikt, met 128K VRAM en een 128K Memory-
Mapper. Het programma moet opstarten in hetzelfde slot als de mapper.
120 '
130 CLEAR 600,&HC000: DEFINT A-Z: CLS: WIDTH 80
140 PRINT SPACE$(28); "MSX Computer Magazine's"
150 PRINT SPACE$(30); "diskette quick-copy"
160 PRINT
170 PRINT "Dit Basic-programma installeert KOPPIE, een snelle disk-kopieerder di
e per"
180 PRINT "keer wisselen 180K kopieert."
190 PRINT
200 PRINT "Koppie werkt alleen op een MSX2 met 128K Video-RAM en een 128K Memory
-Mapper!"
210 PRINT "Bovendien moet het programma in dezelfde slot opstarten als die Memor
y-Mapper."
220 PRINT
230 PRINT "Dat zal echter (bijna?) altijd wel kloppen. In de praktijk werkt KOPP
IE op ie- dere MSX2 die in Nederland verkocht is."
240 PRINT
250 PRINT "Het programma moet eerst uit de DATA-regels worden gelezen en in het
geheugen geplaatst worden. Even geduld aub."
260 PRINT
270 PRINT "Er moeten nog DATA-regels worden verwerkt."
280 ' Inlezen en wegpoken KOPPIE-machinetaal *****
290 AD=&HC100: RESTORE
300 FOR F=0 TO 140

```

```

310 LOCATE 15,15: PRINT 141-F 23
320 READ A$: X1=0: X2=0 5
330 FOR G=0 TO 23 210
340 PL=G*2+G\2+G\8+1 195
350 X=VAL("&h"+MID$(A$,PL,2)) 33
360 POKE AD,X: AD=AD+1 212
370 IF GMOD2=0 THEN X1=X1 XOR X 182
380 IF GMOD2=1 THEN X2=X2 XOR X 7
390 NEXT G 249
400 Y1=VAL("&h"+MID$(A$,64,2)) 81
410 Y2=VAL("&h"+MID$(A$,66,2)) 150
420 IF Y1<>X1 OR Y2<>X2 THEN GOTO 760 ' fout gevonden 206
430 NEXT F 178
440 ' Machinetaal-programma succesvol ingelezen en in geheugen geplaatst ***** 0
450 CLS 10
460 PRINT SPACE$(28); "MSX Computer Magazine's" 64
470 PRINT SPACE$(30); "diskette quick-copy" 26
480 PRINT 147
490 PRINT "KOPPIE staat nu in het geheugen vanaf &HC100 tot en met &HCE2D." 72
500 PRINT "Het start-adres is &HC100." 209
510 PRINT 134
520 PRINT "U kunt het nu als kant en klaar machinetaal-programma op diskette of
cassette laten zetten, zodat u het de volgende keer met BLOAD kan starten." 129
530 PRINT "Daarna kunt u KOPPIE meteen laten opstarten." 210
540 PRINT 140
550 PRINT "Om een disk-versie aan te maken moet u nu op de 'D' drukken." 94
560 PRINT "Voor een cassette-versie moet u de 'C' kiezen." 181
570 PRINT 146
580 PRINT "Om KOPPIE meteen te starten drukt u nu op de spatiebalk." 203
590 PRINT "Zorg er echter wel voor dat u eerst KOPPIE een keer weggeschreven hee
ft, als machinetaal-programma of als Basic-loader!" 115
600 I$=INPUT$(1): PRINT 117
610 IF I$=" " THEN DEFUSR0=&HC100: PRINT USR0(0) 23
620 IF I$="d" OR I$="D" THEN NM$="koppie.bin" 98
630 IF I$="c" OR I$="C" THEN NM$="cas:koppie" 205
640 PRINT "De volgende keer kunt u KOPPIE starten met:" 168
650 PRINT "BLOAD";CHR$(34);"KOPPIE";CHR$(34);",R" 110
660 PRINT "of, voor cassette:" 129
670 PRINT "BLOAD";CHR$(34);"CAS:KOPPIE";CHR$(34);",R" 96
680 BSAVE NM$,&HC100,&HCE2D,&HC100 47
690 CLS 20
700 PRINT "KOPPIE is weggeschreven!" 229
710 PRINT "Desgewenst kunt u nog een kopie wegschrijven, of het programma nu sta
rten." 19
720 PRINT 138
730 GOTO 540 92
740 END 184
750 ' Fout gevonden ***** 0
760 BEEP 254
770 PRINT 148
780 PRINT "Er is een fout in de DATA gevonden!" 145
790 PRINT "Kijk regel";F*10+820;" na!" 111
800 STOP 232
810 ' Koppie-DATA ***** 0
820 DATA "21F0 CD5B 5B23 F3ED 53FE CD22 23F3 ED73 EEC0 CD56 0131 F0C0 C6E5" 39
830 DATA "2100 003A C1FC CD24 0021 0040 3AC1 FCCD 2400 3E08 CD5F 003E 3CE6" 33
840 DATA "0032 F5FA 32F6 FACD 3D01 F3AF D3FE 1100 8021 00C0 0100 40ED F0AF" 234
850 DATA "B0D3 FF3E 02D3 FE11 0080 2100 CE01 3500 EDB0 3DD3 FEFB 2100 6636" 91
860 DATA "00CD 7101 3A07 004F 3AA2 FC57 062C C506 00ED 5114 10FB 1414 1B40" 165
870 DATA "C110 F306 0821 2608 C5E5 CD4E CCE1 C124 10F6 0608 2126 1CC5 324C" 103
880 DATA "E5CD 4ECC E1C1 2410 F61E 000E 2ECD 7DF3 CDF0 C4CD 9F00 FE1B A3D8" 61
890 DATA "CAAC CD26 CD2E 72E5 DDE1 DD7E 002B AEDD AEFE FE49 C2AC CD11 4922" 250
900 DATA "261C ED53 E1CA 2100 0022 E3CA E5CD 80C2 E122 E3CA CD33 C33A 8183" 29
910 DATA "55CC 3D28 1811 261E ED53 E1CA 2AE3 CAE5 CDE0 CBCD 96C2 E122 CBB9" 45
920 DATA "E3CA CD33 C311 2620 ED53 E1CA 2AE3 CAE5 CDE0 CBCD 96C2 E122 569A" 81

```





## Categorie: toepassingen

# Teller

Het programma Teller is op de MSX Computer Magazine redactie uit nood geboren. We moesten nl. eventjes 'snel' bepalen welk programma de MSX Computer Magazine TROFEE gewonnen zou hebben, welk programma het meest genoemd was op de lezers-enquête formulieren.

En dat waren er heel wat, die formulieren. Daarbij kwam nog dat we van tevoren niet konden inschatten hoeveel verschillende programma's er genoemd zouden worden.

Al met al reden genoeg voor de redactie-assistente om eens diep te zuchten en zich af te vragen of daar nu geen handige oplossing voor te vinden zou zijn. Om nu alles maar op een bloknoet-velletje te gaan turven leek haar niet zo'n geweldig goed idee.

### Teller

Om kort te gaan, we hebben er inderdaad een handiger oplossing voor bedacht, in de vorm van een klein programmaatje. Dat programma voldeed prima, maar zat natuurlijk wel wat 'slordig' in elkaar. Ook bij ons wil het wel eens gebeuren dat zo'n snel geschreven programma nu niet echt een wondertje van gestructureerd programmeren is.

Het basis-principe was echter wel goed, en we beseften ons dat een dergelijk tel-programma - een soort geautomatiseerd schrap-blaadje - voor veel meer zaken te gebruiken is. Vandaar dat we dat oorspronkelijke programma maar eens netjes hebben opgepoetst teneinde het in MCM te kunnen opnemen.

### Gebruik

Teller is voor heel veel zaken te gebruiken. In feite kan men alles wat men op moet tellen ook in Teller stoppen. Maar vooral als het niet van tevoren bekend is wat voor zaken ergeteld moeten worden is Teller ideaal.

Stel dat men een grote berg postzegels wil uitzoeken, om te weten hoeveel van iedere zegel men heeft, dan zijn daar natuurlijk meerdere methoden voor. Zo zal men gewoonlijk eerst de zegels allemaal netjes op soort sorteren, om daarna al die stapeltjes te tellen.

Met Teller kunnen die twee zaken keurig gekombineerd worden, onder het sorteren kan men steeds als er een nieuwe soort wordt gevonden - en er een nieuw stapeltje moet worden gemaakt - het programma ook een nieuw 'vakje' laten aanmaken.

Een ander voorbeeld, stel dat men het resultaat van de bestuursverkiezing van de lokale klaverjas-klub moet bepalen. Iedereen heeft op een briefje de naam geschreven van de persoon die hij of zij het meest geschikt acht als penningmeester voor het komende seizoen. Als die klaverjas-klub maar groot genoeg is, kan dat een hele klus zijn. Met Teller wordt ook dat een stuk simpeler, temeer omdat men niet van tevoren hoeft te weten welke namen men tegen kan komen. Zelfs hoeveel namen het zijn is van tevoren niet echt van belang.

### Werking

In het gebruik is Teller heel makkelijk. De capaciteit van het programma is sterk afhankelijk van het feit of het op een MSX1 of een MSX2 draait, aangezien het gebruik maakt van het scherm. Dat zoekt het dan ook netjes

even zelf uit, in regel 120, waarna allerlei andere zaken daar op worden aangepast.

Allereerst wil Teller weten of u met een diskdrive of met een cassette-recorder werkt. Waarom, dat wordt al gauw duidelijk.

Want het volgende dat Teller wil weten is of men een al eerder begonnen telling van disk of tape wil terugladen. Dat kan namelijk ook nog, als men halverwege een telling besluit dat het voor vandaag wel genoeg geweest is kan men de tussenresultaten netjes opbergen tot de volgende dag.

Stel dat u met een oud bestandje door wil gaan, dan zal dat netjes geladen worden, waarna u rechtstreeks in het 'telscherm' terecht komt. Maar als u een nieuwe telling wilt beginnen verschijnt er eerst nog een vraag.

Teller wil dan van u horen hoe lang de namen zijn van de zaken die wil gaan optellen. Aangezien Teller zometeen die namen op het scherm zal laten zien is dat die lengte van belang. Hoe langer de namen, hoe minder er tegelijkertijd op het scherm passen.

Vandaar ook dat Teller op grond van de door u opgegeven lengte eventjes berekend hoeveel namen er dan wel op een scherm passen, waarna u die waarde alsnog kan aanpassen.

### Telscherm

Nu Teller eenmaal alle gegevens heeft komt u op het eigenlijke werkscherm. Op dit

scherm laat Teller steeds alle namen die het programma kent zien, met voor die namen een volgnummer en achter die namen een telling. U kunt steeds kiezen uit drie mogelijkheden.

Ten eerste kunt u de telling van een van de bekende rubrieken met een verhogen, door het volgnummer van die rubrieknaam in te tikken. Maar u kunt ook nieuwe rubrieken invoeren, door als getal een nul in te tikken. Daarna vraagt Teller u om de nieuwe naam, die meteen keurig op zijn plaats wordt gesorteerd.

De derde mogelijkheid is om de speciale kode '999' in te tikken, waarna u op een hulpschermje komt. Op dit scherm kunt u kiezen uit het saven van de telling tot dan toe of het afdrukken van het resultaat van de telling. In beide gevallen komt u daarna weer terug op het eigenlijke telscherm.

Om Teller te stoppen moet u de Control-Stop toets-kombinatie gebruiken, er is geen 'nette' manier om het programma af te breken.

### Tenslotte

Hopelijk spreekt Teller u aan. Het is een simpel maar voor sommige dingen verdraaid handig programmaatje. Bovendien is het een leuk voorbeeld van hoe een programma zichzelf kan aanpassen aan een MSX1- of een MSX2-omgeving.

```
10 REM TELLER
20 REM
30 REM MSX Computer Magazine
40 REM
50 ' initialisatie vars. *****
60 CLEAR 5000: DEFINT A-Z:
70 MX=250
80 DIM NAS$(MX),NT(MX)
90 ' initialisatie screen *****
```

```
0
0
0
0
0
22
7
26
0
```



```

100 KEY OFF
110 SCREEN 0
120 TY=PEEK(&H2D)
130 IF TY=0 THEN RL=40 ELSE RL=80
140 WIDTH RL
150 TT$="MSX Computer Magazine's Tell
ertje"
160 TT$=SPACE$((RL-LEN(TT$))/2)+TT$
170 ' start *****
180 GOSUB 980
190 PRINT "Diskette of cassette (c/d)
";
200 P$=INPUT$(1)
210 IF INSTR("Cc",P$) THEN DV$="cas:"
ELSE IF INSTR("Dd",P$) THEN DV$="a:"
ELSE PRINT:GOTO 190
220 PRINT:PRINT
230 PRINT "bestand laden (j/n)";
240 P$=INPUT$(1)
250 IF INSTR("Jj",P$) THEN GOSUB 830:
GOTO 360 ELSE IF INSTR("Nn",P$) THEN
GOTO 260 ELSE PRINT: GOTO 240
260 PRINT:PRINT
270 INPUT "Aantal posities per naam";
BM
280 AR=RL\ (9+BM)
290 PRINT
300 PRINT "Er passen dan";AR;" namen
per regel"
310 PRINT "In totaal ";20*AR;" namen
per scherm"
320 PRINT
330 PRINT "Akkoord (j/n)
340 P$=INPUT$(1)
350 IF INSTR("Jj",P$) THEN GOTO 360 E
LSE IF INSTR("Nn",P$) THEN GOTO 270 E
LSE 330
360 ' hoofdlus *****
370 GOSUB 480
380 PRINT
390 PRINT "Nummer? (0 => nwe naam, 99
9 => menu)";
400 INPUT A$
410 IF LEN(A$)=0 THEN 390
420 H=VAL(A$)
430 IF H=0 THEN GOTO 610: ' nieuwe naa
m
440 IF H=999 THEN GOSUB 1020: ' menu
450 IF H<1 OR H>LA THEN 390
460 NT(H)=NT(H)+1
470 GOTO 370
480 ' bouw scherm *****
490 GOSUB 980
500 FOR N=1 TO LA STEP AR
510 FOR M=0 TO AR-1
520 IF N+M=LA+1 THEN M=AR-1: N=LA
: GOTO 570
530 PRINT USING "### ";N+M;
540 PRINT LEFT$(NA$(N+M),BM);
550 IF LEN(NA$(N+M))<BM THEN PRIN
T SPACE$(BM-LEN(NA$(N+M)));
560 PRINT USING " ###";NT(N+M);
570 NEXT M
580 PRINT
590 NEXT N
600 RETURN
610 ' nieuwe naam toevoegen *****
620 PRINT: INPUT "Geef naam";NN$
630 LA=LA+1
640 NA$(LA)=NN$: NT(LA)=1
650 GOSUB 1190
660 GOTO 360
670 ' save bestand *****
680 GOSUB 980

```

```

170
83
135
216
108
59
119
0
240
154
229
39
227
118
237
74
235
37
164
147
182
80
134
148
238
193
0
185
146
26
108
96
172
252
120
54
243
110
245
217
19
246
155
51
39
22
185
14
15
192
0
99
203
187
88
99
0
245

```

```

690 IF DV$="cas:" THEN 710
700 PRINT "Bestanden op deze schijf:"
:FILES:PRINT:PRINT
710 INPUT"naam weg te schrijven besta
nd?";SC$
720 OPEN DV$+SC$ FOR OUTPUT AS 1
730 PRINT: PRINT "Bestand wordt gesav
ed":PRINT:PRINT
740 PRINT #1, LA
750 PRINT #1, BM
760 PRINT #1, AR
770 FOR N=1 TO LA
780 PRINT #1,NA$(N)
790 PRINT #1,NT(N)
800 NEXT N
810 CLOSE 1
820 RETURN
830 ' laad bestand *****
840 GOSUB 980
850 IF DV$="cas:" THEN 870
860 PRINT "Bestanden op deze schijf:"
:FILES:PRINT:PRINT
870 INPUT"naam te laden bestand?";LE$
880 OPEN DV$+LE$ FOR INPUT AS 1
890 PRINT: PRINT "Bestand wordt gelad
en":PRINT:PRINT
900 INPUT #1, LA
910 INPUT #1, BM
920 INPUT #1, AR
930 FOR N=1 TO LA
940 INPUT #1,NA$(N),NT(N)
950 NEXT N
960 CLOSE 1
970 RETURN
980 ' nieuw scherm *****
990 CLS
1000 PRINT TT$: PRINT
1010 RETURN
1020 ' menu *****
1030 GOSUB 980
1040 PRINT "Bestand saven (j/n)";
1050 P$=INPUT$(1)
1060 IF INSTR("Jj",P$) THEN GOSUB 670
ELSE IF INSTR("Nn",P$) THEN GOTO 107
0 ELSE PRINT: GOTO 1040
1070 PRINT: PRINT
1080 PRINT "Telling afdrukken (j/n)";
1090 P$=INPUT$(1)
1100 IF INSTR("Jj",P$) THEN GOSUB 112
0: GOTO 360 ELSE IF INSTR("Nn",P$) TH
EN GOTO 1110 ELSE PRINT: GOTO 1080
1110 GOTO 360
1120 ' afdrukken *****
1130 LPRINT "NAAM
Aantal"
1140 LPRINT
1150 FOR N=1 TO LA
1160 LPRINT USING "\
###";NA$(N),NT(N)
)
1170 NEXT N
1180 RETURN
1190 ' bubble sort *****
1200 IF LA=1 THEN 1280
1210 FOR T1=1 TO LA-1
1220 V1=0
1230 FOR T2=1 TO LA-T1
1240 IF NA$(T2+1)<NA$(T2) THEN
SWAP NA$(T2+1),NA$(T2): SWAP NT(T2+1)
,NT(T2): V1=1
1250 NEXT T2
1260 IF V1=0 THEN T1=LA-1
1270 NEXT T1
1280 RETURN

```

```

140
111
228
156
109
27
71
138
109
191
221
0
74
198
0
239
52
124
183
98
113
67
111
178
103
190
11
85
209
0
23
126
0
10
149
186
139
71
157
198
108
41
0
245
80
125
31
157
149
0
183
49
250
78
92
130
130
45
151

```

## Kategorie: utility's

# SPRITE

Een van de eigenaardigheden van computers is dat een en hetzelfde probleem op ontzettend veel manieren kan worden aangepakt. Er zijn bijvoorbeeld vele manieren om een tekstverwerker te programmeren, allemaal met hun eigen voor- en nadelen. Maar al die benaderingswijzen hebben toch een ding gemeen, namelijk dat het uiteindelijke programma teksten moet kunnen verwerken.

Een soort programma waarbij dat ook maar al te duidelijk is vormt de sprite-editor. Kort gezegd zijn sprite-editors hulp-programma's waarmee het ontwerpen van de sprites - de handige grafische 'tekens' waar ieder spel gebruik van maakt - op het beeldscherm kan gebeuren. Een hele verbetering vergeleken met het ontwerpen op papier!

### Overvloed

Sprite-editors komen we dan ook met regelmaat tegen. Vele software-huizen hebben hun eigen editors ontworpen en op de markt gegooid, maar ook vele hobby-programmeurs kunnen de verleiding om eens een sprite-editor te ontwerpen niet weerstaan.

We worden weliswaar niet overstroomd met sprite-editors, maar we krijgen ze toch wel met enige regelmaat toegezonden. De meeste van die toegezonden sprite-editors zijn echter niet om over naar huis te schrijven, laat staan dat ze geschikt zijn voor publikatie in MSX Computer Magazine.

### Toeters en bellen

Een goed stuk gereedschap - of het nu een computerprogramma is of een draaibank - moet namelijk in eerste instantie goed ontworpen zijn. Het moet op zijn minst redelijk efficiënt kunnen doen wat het verondersteld wordt te moeten doen.

De bediening moet eenduidig zijn en bij voorkeur zo simpel mogelijk. Als men iedere keer

dat er een wat minder gebruikte functie aan bod komt eerst eens moet gaan nazoeken hoe dat ook alweer diende te gebeuren is er per definitie iets niet goed aan dat stuk gereedschap.

Pas als er aan die basis-eisen is voldaan kan men gaan overwegen om verdere mogelijkheden aan het programma toe te voegen, dat is althans onze bescheiden mening.

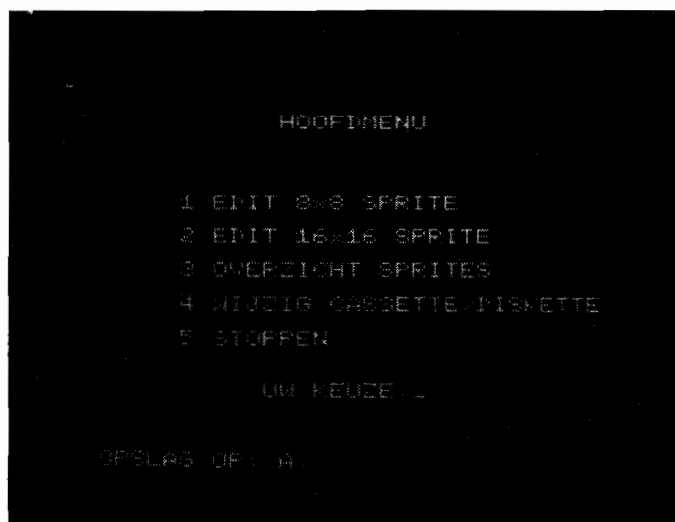
Veel van de programma's die wij onder ogen krijgen munten echter uit door allerlei overbodige toeters en bellen. De eigenlijke basis-functies - waar het toch uiteindelijk om draait - zijn dan wel aanwezig, maar dreigen te verdrinken in alle extra's.

Soms ook is dat verdrinken bijna letterlijk op te vatten, een en ander wordt dan - mede door de veelheid aan mogelijkheden - onaanvaardbaar traag. En dat is toch ook wel een eis die aan een goed stuk gereedschap gesteld mag worden, dat het zijn werk redelijk rap doet.

### Blij verrast

Vandaar dat we met de programma's die ons onlangs werden toegezonden door Roelof Ridderman uit Ruinerwold blij verrast werden. Roelof stelde zelf - heel voorzichtig - dat zijn programma's niet uitblonken door originaliteit. Mogelijk is dat waar, maar ze blonken in ieder geval wel uit door degelijkheid!

Zo bleek zijn sprite-editor, die we hier afdrukken, tot de beste te behoren die we tot nog toe



gezien hebben. Behoorlijk compleet qua mogelijkheden, helder van opzet zowel in het gebruik als wat programmering betreft.

De inzender zelf schreef ons in zijn begeleidende brief: 'Toen ik de sprite-editor in het MCM-listingboek zag staan werd ik wat wit om de neus. Mijn meesterwerk was geëvenaard, dacht ik.'

Daar kunnen we het hier op de redactie echter niet mee eens zijn. Onze eerste sprite-editor - een winnaar in de Sony-MCM programmeerwedstrijd die oorspronkelijk in MCM nummer 3 verschenen is en ook in het grote listingboek is opgenomen - was heel aardig van opzet en zeker uitstekend bruikbaar. Het programma van de heer Ridderman is echter toch wel van een andere klasse. Handige menu's, goed doordachte interactie met de gebruiker, slimme extra mogelijkheden, het geheel geeft de indruk een goed uitgerijpt programma te zijn.

### Programmering

Maar niet alleen naar buiten toe straalt deze sprite-editor een professionalisme uit, ook van binnen is het een prima programma. We zullen in dit kader de werking van het Basic-programma niet gaan uitleggen, maar voor de wat gevorderde programmeur is het zeer zeker de moeite waard om eens wat in deze listing rond te neuzen.

Een goede gestructureerde opzet, heldere programmering, en vooral een prima pro-

gramma-lussen opzet maken dat deze editor snel en foutloos zijn wer doet. Het vele commentaar maakt een en ander bovendien bijzonder leesbaar.

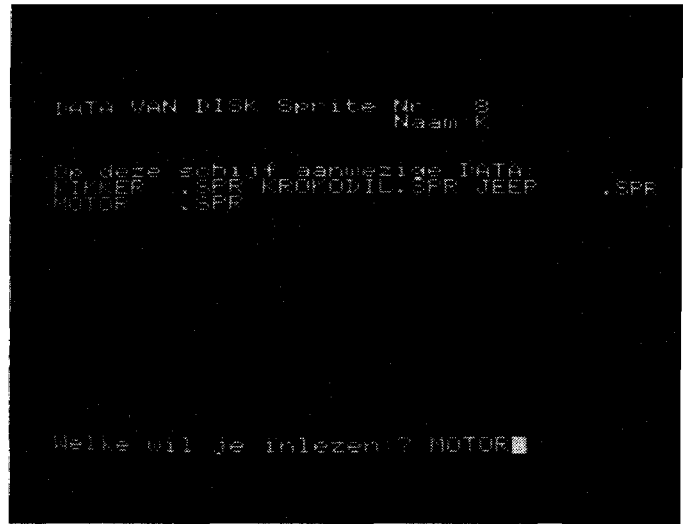
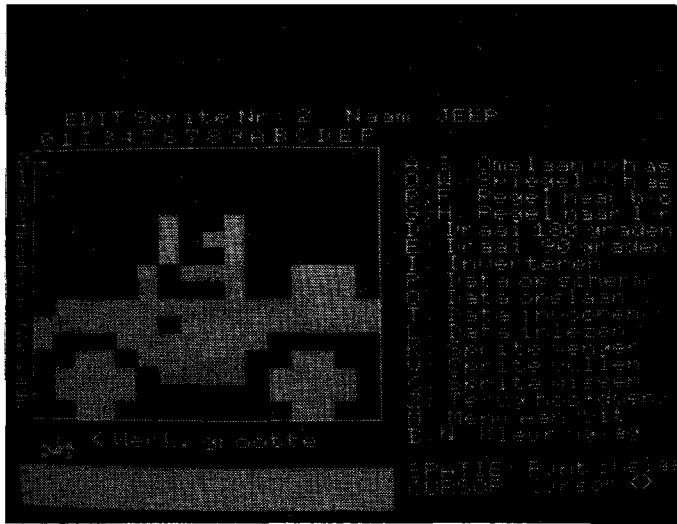
Het enige wat we zelf aan het programma hebben moeten toevoegen was de mogelijkheid om de sprites ook op cassette op te slaan. De ingestuurde versie werkte alleen op disk, wat we toch iets te beperkt vonden. We hebben echter naar ons idee een prima mogelijkheid gevonden hiervoor; door de keuze diskette of cassette in het hoofdmenu op te nemen konden we deze verandering doorvoeren zonder de eigenlijke kommando-structuur te hoeven omschrijven. Dat had alleen maar kans op fouten gegeven!

Heel fraai vonden we de routine om teksten op het edit-scherm te zetten. Normaal gesproken verschijnen de letters op dat scherm met een te grote tussenruimte, maar in Sprite is dit ondervangen. Het printen van tekst op het scherm wordt er weliswaar wat trager van, maar er past wel meer tekst op het scherm. En dat is, zoals we zullen zien, heel handig!

### Gebruik

In het gebruik blijkt Sprite erg prettig te hanteren te zijn. Op een dingetje na, het programma verwacht alle invoer in hoofdletters, maar schakelt de CAPS LOCK-toets niet zelf in. Eerst even de kapitalenmode instellen dus.

Na het opstarten komen we in een hoofdmenu terecht, waar



we kunnen kiezen tussen de mogelijkheden:

- Edit 8 bij 8 sprite;
- Edit 16 bij 16 sprite;
- Overzicht sprites;
- Wijzig cassette/diskette;
- Stoppen.

Verder krijgen we een melding te zien welk rand-apparaat - cassette of diskette - momenteel gekozen is, als het programma begint zal dit 'A:' ofwel de diskdrive zijn. Een tip voor cassette-gebruikers:

In regel 170 staat onder meer EO\$="A:" waarmee die diskdrive bij het opstarten wordt vastgelegd. Door dit stukje van regel 170 te veranderen in: EO\$="CAS:" wordt het programma standaard gestart met de cassette-recorder.

De keuze 'stoppen' is echt noodzakelijk, Sprite is beschermd tegen de soms gebruikelijke maar wat botte CTRL-STOP toetskombinatie. Die twee toetsen brengen u - waar u ook bent - terug naar het hoofdmenu.

Als we kiezen voor het editten van een sprite krijgen we nog wat vragen te beantwoorden. Eerst wil het programma het nummer van de editten sprite weten, waarbij we kunnen kiezen tussen 0 en 19. In totaal kunnen er dan ook 20 sprites tegelijkertijd in het geheugen aanwezig zijn.

Dan moeten we de sprite benoemen. In deze sprite-editor heeft iedere sprite namelijk behalve een nummer ook nog een eigen naam. Heel handig,

in de overzichten, want zo'n naam is natuurlijk veel makkelijker te onthouden dan een nummer. Overigens moet die naam - net als alle andere invoer - in hoofdletters opgegeven worden, kleine letters worden niet geaccepteerd.

Tenslotte mogen we nog kiezen of we het edit-menu aan of uit willen zetten. De eerste keer zal een A het juiste antwoord zijn.

### Edit-scherm

Dan verschijnt het eigenlijke werkscherm. Het is fraai en overzichtelijk van indeling, met bovenaan de naam en het nummer van de sprite waar we aan bezig zijn.

Dan zien we, links op het scherm, de eigenlijke edit-ruimte, die keurig van coördinaten voorzien is. Daaronder wordt de sprite op werkelijke grootte getoond en helemaal onderaan vinden we het berichten-vakje. Alle meldingen die Sprite aan de gebruiker wilt geven op dit scherm zullen daar verschijnen, met een korte attentie-piep.

De meest gezien melding is overigens 'U kunt verder!', waarmee aangegeven wordt dat er een functie beëindigd is.

Maar het fraaiste op dit scherm is nog niet aan bod geweest! Rechts vinden we namelijk een groot vlak met daarin alle mogelijke kommando's in de edit-mode. Dat soort dingen maakt nu het verschil uit tussen een aardig programma en een goed programma, door dat hulpmenu hoeven we namelijk niet iedere functie op te gaan

zoeken in de gebruiksaanwijzing. Alles is in beeld.

Wie er - na een tijdje intensief gebruik - geen behoefte meer aan heeft kan dat menu desgewenst ook uitzetten. Daar sloeg die keuze op het hoofdmenu - edit-menu aan of uit - op. Als men de kommando's eenmaal uit het hoofd kent kan dat uitzetten van het edit-menu weer wat snelheidswinst opleveren. Want door de speciale print-methode waarmee de programmeur kans gezien heeft om zoveel mogelijk informatie in dat vakje te propen is het wel wat traag van opbouw geworden.

Het eigenlijke ontwerpen doen we met de cursor-toetsen en de spatiebalk. Met de cursor-toetsen bewegen we de cursor over het ontwerpveld, compleet met wrap-around, als we er links vanaf lopen komen we rechts weer terug. De spatiebalk zet de punt waar de cursor op staat aan of uit, afhankelijk van het feit of dat punt al dan niet aanstond.

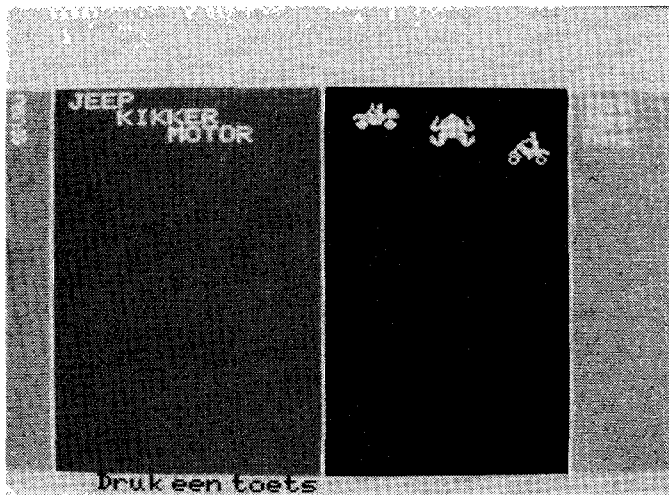
De menu-keuzes zijn:

- A: omslaan om de verticale as, wat links is komt dan rechts en omgekeerd;
- S: omslaan om de horizontale as, wat boven is komt dan onder en omgekeerd;
- Q: spiegelen om de verticale as, de linkerzijde wordt omgekeerd naar rechts gekopieerd;
- W: spiegelen om de horizontale as, de bovenste helft wordt omgekeerd naar onderen gekopieerd;
- R: verplaats ontwerp een regel naar boven;

- F: verplaats ontwerp een regel naar onder;
- G: verplaats ontwerp een regel naar links;
- H: verplaats ontwerp een regel naar rechts;
- D: roteer ontwerp 180 graden;
- E: roteer ontwerp 90 graden;
- I: inverteer, verwissel voor- en achtergrondkleuren;
- P: print sprite-data op scherm;
- O: schrijf sprite-data naar cassette of diskette;
- T: data invoeren via toetsenbord, naar keuze in decimaal of in hexadecimaal;
- L: lees sprite-data van cassette of diskette;
- K: sprites mengen of kopiëren, met deze functie kunnen twee spritesa desgewenst tot één samengevoegd worden, terwijl mengen naar een leeg ontwerpveld in feite een kopiëren-functie is;
- V: sprite vullen, alle puntjes worden gezet nadat er om een bevestiging gevraagd is;
- C: sprite wissen, na bevestiging;
- Z: keer terug naar hoofdmenu;
- M: edit-menu aan of uit, om sneller te kunnen werken;
- B: voorgrond-kleur kiezen, kleurnummer wordt gevraagd;
- N: idem voor de achtergrond-kleur.

### Indrukwekkend

Een werkelijk indrukwekkende lijst van mogelijkheden. Bij sommige keuzes verschijnt weer een hulpscherm, zoals bij het laden en saven van sprites. Vooral voor diskgebruikers zijn deze routines heel vriendelijk van opzet, ze geven namelijk een keurig lijstje van de sprite-bestanden op het



scherm. Dat kan omdat Sprite er van uitgaat dat files die een sprite bevatten van het achtervoegsel .SPR voorzien zijn.

Als het programma zelf bestandjes aanmaakt - op disk tenminste - zal het deze toevoeging altijd aan de opgegeven naam plakken.

Die bestanden bevatten per stuk slechts één enkele sprite, maar wel in een handig formaat. Het zijn namelijk regelrechte Basic-programmaatjes, in ASCII-kode weggeschreven zodat ze heel eenvoudig met andere programma's gemerged kunnen worden. De regelnummers houden daar zelfs al rekening mee, ieder spritenummer krijgt als bestandje zijn eigen, niet overlappende Basic-regelnummers. Kompleet zelfs met een keurige REM-regel voorop, die de naam van de sprite bevat.

Petje af, wat ons betreft. Dit soort extraatjes maakt een programma nu werkelijk gebruikersvriendelijk!

### Bestaande definities

Het is ook zonder meer mogelijk om al bestaande Basic-programma's met sprite-definities in data-regels in de editor in te lezen, maar er worden dan wel een paar eisen aan die programma's gesteld.

De regelnummers van zo'n in te lezen programma dienen exact vijf cijfers lang te zijn en ze moeten in ASCII-vorm gesaved zijn, net zoals de programmaatjes die Sprite zelf wegschrijft. Erg ingewikkeld is dat gelukkig niet.

De eerste regel mag geen sprite-data bevatten, het mag bijvoorbeeld een REM-regel zijn. Daarna moeten de eigenlijke sprite-dataregels volgen, waarbij de data in hex-notatie moeten staan. Op iedere regel moeten precies 8 hex-getallen staan, niet meer en niet minder.

Hoeveel regels Sprite inleest hangt er van af of er een 8 bij 8 of een 16 bij 16 sprite ge-edit wordt. In het eerste geval slechts eentje, in het tweede geval vier. Per programma-bestand kan er maar een enkele sprite worden ingelezen.

Ook heel prettig is het feit dat Sprite er niet zonder meer vanuit gaat dat de gebruiker wel geen foutjes zal maken. Zo wordt er bijvoorbeeld - in het meldingen-venster - om bevestiging gevraagd als er een al bestaande sprite zou worden overschreven door een editkommando. Veiligheid is nu eenmaal de moeder van de porceleinkast.

### Sprite-overzicht

Nog niet genoemd is de keuze 3 van het hoofdmenu, dat een overzicht van alle sprites op het scherm roept. Dit overzicht is alweer fraai en overzichtelijk uitgevoerd. Niet alleen verschijnen de naam, nummer en vorm van de sprites in keurige tabellen, maar er is ook een kolom gereserveerd om aan te geven of de sprite in kwestie al is weggeschreven naar tape of disk.

Dergelijke zaken worden door dit prima stuk gereedschap nu eenmaal allemaal bijgehouden, om de gebruiker werk uit handen te nemen.

Toch hebben we bij deze functie van Sprite een bepaald gemis ervaren. Er is nog een extra mogelijkheid denkbaar, die erg prettig zou kunnen zijn.

### Gemis

Vaak worden er namelijk meerdere sprites over elkaar gebruikt om bepaalde figuurtjes te vormen, waarbij ieder van de 'gestapelde' sprites een bepaalde kleur van het uiteindelijke figuurtje voor zijn rekening neemt. Dat kan nu eenmaal niet anders bij MSX1, aangezien elke sprite slechts een kleur kan hebben.

Daar een soort van 'proefveld' voor te hebben in de overzichtsmode lijkt ons een erg fraaie extra. Gewoon de gewenste sprite kunnen oppikken, van een kleur voorzien en op de juiste plek manoeuvreren, waarna de volgende sprite kan worden gepositioneerd. Dat is dan ook een idee tot uitbreiding van Sprite dat we bij deze aan het lezers-forum voor willen leggen. Wie schrijft er een nette uitbreiding op Sprite waarmee dit mogelijk wordt? We zullen zo'n extra functie - mits goed geprogrammeerd,

wat wil zeggen zonder dat daarvoor het hele huidige edit-programma voor overhoop gehaald moet worden - graag publiceren. Bijvoorbeeld in de vorm van een 'patch', een stukje Basic-programma dat met de huidige Sprite gemerged kan worden.

### Konklusie

We dachten dus dat er weinig er meer te behalen viel aan sprite-editors. Dat blijkt hard mee te vallen, we zijn behoorlijk lyrisch geworden over dit programma van de hand van R. Ridderman. Fraai, doorzicht en vooral anticiperend op hoe de gebruiker zo'n gereedschap zou willen kunnen hanteren, dat is onze indruk van Sprite. Dezelfde overzichtelijke stijl van programmeren vonden we overigens ook in zijn andere programma's, waarvan er nog eentje - Keuken - in dit nummer is opgenomen.

We willen de inzender dan ook van harte bedanken voor deze bijdrage en hopen stilletjes nog eens wat vaker van hem te zien te krijgen.



```

10 REM SPRITE 0
20 REM 0
30 REM MSX Computer Magazine 0
40 REM 0
50 REM Inzender: R. Ridderman, Ruiner 0
wold 0
60 REM 0
70 '***** 0
80 ' * 0
90 ' * SPRITE EDITOR DE LUXE * 0
100 '* ----- * 0
110 '* Rori-soft sept. 1986 * 0
120 '* * 0
130 '***** 0
140 ' 0
150 'Initialisatie ***** 0
160 ' 0
170 CLEAR 500:DEFINT A-Z:EO$="A:" 108
180 DEFUSR=342 183
190 ON ERROR GOTO 3660 222
200 ON STOP GOSUB 4710:STOP ON 187
210 SCREEN 0:WIDTH 40:COLOR 15,4,4:CL 103
S 63
220 OPEN "GRP:" AS #1 74
230 DIM SP(19,4,8),SB(19),HS(4,8)
240 FOR I=0 TO 7 :M(I)=2^(I-7): NEXT 180
I 230
250 DIM SN$(19):CP=-1:VC=2:AC=1
260 'Hoofdmenu ***** 0
270 ' 0
280 GOSUB 350 139
290 I$=INKEY$:IF I$="" THEN 290 118
300 IF I$>"5" OR I$<"1" THEN 290 27
310 PRINT I$:ON VAL(I$) GOSUB 680,680 85
,3720,5600,1620 92
320 CLS:GOTO 280 0
330 'Scherm hoofdmenu ***** 0
340 ' 0
350 TH=0:SCREEN 0: KEY OFF: CLS 57
360 LOCATE 16,0:PRINT"HOOFDMENU" 252
370 LOCATE 10,5:PRINT"1 EDIT 8x8 SPRI 77
TE" 99
380 LOCATE 10,7:PRINT"2 EDIT 16x16 SP 100
RITE" 249
390 LOCATE 10,9:PRINT"3 OVERZICHT SPR 51
ITES" 247,
400 LOCATE 10,11:PRINT"4 WIJZIG CASSE 46
TTE/DISKETTE" 136
410 LOCATE 5,20:PRINT"OPSLAG OP: ";EO 200
$ 0
420 LOCATE 10,13:PRINT"5 STOPPEN" 0
430 LOCATE 15,16:PRINT"UW KEUZE: _" 0
440 LOCATE 24,16 186
450 RETURN 243
460 'Sprite edit scherm ***** 228
470 ' 228
480 COLOR 15,4,4:CLS 89
490 LINE (0,0)-(255,10),13,BF 161
500 X=20:Y=2:C=13:KK$="EDIT Sprite Nr 2
:"+STR$(SN)+" Naam: "+SH$ 193
510 LINE (0,0)-(255,10),13,BF 155
520 GOSUB 1650 84
530 G$="":FOR G=0 TO GR-1 106
540 G$=G$+HEX$(G) 251
550 NEXT G 240
560 PRESET (10,12),4:PRINT#1,G$ 203
570 Y=22:FOR I=1 TO GR 81
580 PRESET (0,Y),4:PRINT#1,MID$(G$,I, 1) 106
590 Y=Y+8:NEXT I 251
600 IF HH THEN GOSUB 4480 240
610 LINE (7,21)-(8+8*GR,22+8*GR),15,B 203
620 GOSUB 1280 81
630 C=4:X=30:Y=156:KK$="< Werk. groot

```

```

te":GOSUB 1650 60
640 LINE (0,172)-(140,192),2,BF 72
650 RETURN 202
660 'Edit sprite 8*8 en 16*16 ***** 0
670 ' 0
680 IF I$="1" THEN GR=8 ELSE GR=16 249
690 GOSUB 990 3
700 SCREEN 2:GOSUB 460 91
710 IF SB(SN)=0 THEN 780 69
720 BEEP:C=2:X=4:Y=174:KK$="Sprite is 17
gemaakt. naam:":GOSUB 1650
730 Y=182:KK$=SN$(SN)+" Editten (J/N) 45
? ":GOSUB 1650
740 I$=INKEY$:IF I$<>"J" AND I$<>"N 104
THEN 740 247
750 LINE(0,172)-(140,192) 2,BF 206
760 IF I$="J" THEN GOSUB 1200 ELSE RE 61
TURN 320 91
770 SN$(SN)=SH$ 179
780 BEEP:C=2:X=4:Y=174:KK$="U kunt be 139
ginnen!":GOSUB 1650:B=1 50
790 FOR I=1 TO 100:NEXT: LINE(0,172)- 6
(140,192) 2,BF:BEEP 130
800 SB(SN)=1:SN$(SN)=SH$ 138
810 IF CP THEN X=9:Y=24:C=POINT(X+4,Y 89
+4):KK$="+":GOSUB 1650 255
820 CP=0 234
830 AA=7:X=8:Y=22:XP=8:YP=154 231
840 I$=INKEY$:IF I$="" THEN 840
850 IF I$="" THEN GOSUB 1330
860 S=STICK(0):IF S<>0 THEN GOSUB 141 108
0 191
870 T=INSTR("SAWQDMFRHGIPOLKZVCTBNE", 238
I$) 0
880 IF T=0 THEN GOTO 840 75
890 ON T GOSUB 1710,1770,1830,1870,19 17
10,1970,2010,2130,2250,2360,2470,2530 210
,2730,2950,3240,3440,3470,3570,4960,5 81
210,5210,5340 211
900 IF TH=1 THEN SN=(SN+1)MOD19:RETUR 0
N 0
910 GOSUB 920:GOTO 810 191
920 'Boodschap verder kunnen ***** 0
930 ' 0
940 LINE (0,172)-(140,192),2,BF 75
950 BEEP:C=2:X=4:Y=175:KK$="U kunt ve 17
rder!":GOSUB 1650 210
960 FOR I=1 TO 300:NEXT I:BEEP 81
970 LINE (0,172)-(140,192),2,BF 211
980 RETURN 0
990 'Naam/nummer sprite inv. ***** 0
1000 ' 0
1010 LOCATE 0,18:PRINT"Geef nummer sp 191
rite (0-19):?":SN:LOCATE 26,18
1020 INPUT S$:IF VAL(S$)<0 OR VAL(S$) 100
>19 THEN BEEP:GOTO 1010 208
1030 SN=VAL(S$)
1040 LOCATE 0,20:PRINT"Geef naam spr 93
ite:?" :LOCATE 17,20 17
1050 INPUT SH$ 14
1060 SH$=LEFT$(SH$,8) 169
1070 FOR I=8 TO 1 STEP -1 28
1080 IF MID$(SH$,I,1)<>"_" THEN 1110 106
1090 NEXT I 221
1100 BEEP:GOTO 1040 85
1110 SH$=LEFT$(SH$,I) 135
1120 FOR I=1 TO LEN(SH$) 95
1130 A=ASC(MID$(SH$,I,1))
1140 IF A<48 OR (A>57 AND A<65) OR A> 150
90 THEN BEEP: GOTO 1040 96
1150 NEXT I 81
1160 LOCATE 0,22:PRINT"Edit Menu (A)a 81
n/(U)it? "
1170 I$=INKEY$:IF I$<>"A" AND I$<>"U"

```

```

THEN 1170 58
1180 IF I$="A" THEN HH=-1 ELSE HH=0 90
1190 RETURN 152
1200 'Gemaakte sprite invullen ***** 0
1210 ' 0
1220 CP=-1:COLOR VC:FOR I=1 TO (GR-8)
/2:ON I GOSUB 4030,4040,4050,4060:XM=
X:XN=XP 95
1230 FOR J=1 TO 8:FOR K=7 TO 0 STEP -
1:H=2^K 93
1240 IF (SP(SN,I,J) AND H) THEN PRESE
T(X,Y):PRINT#1,CHR$(219):PRESET(XP,YP
),VC 144
1250 X=X+8:XP=XP+1:NEXT K 185
1260 X=XM:XP=XN:Y=Y+8:YP=YP+1:NEXT J,
I 243
1270 COLOR 15:X=8:Y=22:XP=8:YP=154:RE
TURN 87
1280 'Sprite vlakken schoon ***** 0
1290 ' 0
1300 LINE (8,22)-(7+GR*8,21+GR*8),AC,
BF 154
1310 LINE (8,154)-(7+GR,153+GR),AC,BF 96
1320 RETURN 135
1330 'Punt veranderen ***** 0
1340 ' 0
1350 P=POINT(X+1,Y+1) 137
1360 IF P=VC THEN COLOR AC:C=AC ELSE
COLOR VC:C=VC 121
1370 PRESET(X,Y):PRINT#1,CHR$(219) 168
1380 PRESET(XP,YP),C 172
1390 GOSUB 1560:PRESET(X+1,Y):COLOR 1
5:PRINT#1,"+" 249
1400 COLOR 15:RETURN 840 158
1410 'Cursor verplaatsen ***** 0
1420 ' 0
1430 COLOR 15:IF S<>1 AND S<>3 AND S<
>5 AND S<>7 THEN 1470 82
1440 P=POINT(X+1,Y+1):LINE(X,Y)-(X+7,
Y+7),P,BF 204
1450 ON S GOSUB 1480,,1500,,1520,,154
0 243
1460 PRESET(X+1,Y),P:PRINT#1,"+" 50
1470 RETURN 840 152
1480 IF Y<=22 THEN Y=14+8*GR:YP=153+G
R ELSE Y=Y-8:YP=YP-1 81
1490 RETURN 158
1500 IF X>=8*GR THEN X=8:XP=8 ELSE X=
X+8:XP=XP+1 141
1510 RETURN 136
1520 IF Y>=14+8*GR THEN Y=22:YP=154 E
LSE Y=Y+8:YP=YP+1 20
1530 RETURN 142
1540 IF X<=8 THEN X=8*GR:XP=7+GR ELSE
X=X-8:XP=XP-1 190
1550 RETURN 148
1560 J=((Y-22)\8)MOD8+1 26
1570 AA=2^(7-(((X-8)\8)MOD8)) 41
1580 I=1:IF Y>=86 THEN I=I+1 250
1590 IF X>=72 THEN I=I+2 173
1600 H=SP(SN,I,J):IF (H AND AA) THEN
SP(SN,I,J)=(H XOR AA) ELSE SP(SN,I,J)
=(H OR AA) 133
1610 RETURN 138
1620 'Einde ***** 0
1630 ' 0
1640 SCREEN 0:KEY ON: END 1
1650 'Tekst afdrukken op scherm ***** 0
1660 ' 0
1670 XM=X:FOR I=1 TO LEN(KK$) 96
1680 PRESET(X,Y),C:PRINT#1,MID$(KK$,
I,1) 187
1690 IF MID$(KK$,I,1)=" " THEN X=X+3
ELSE X=X+6 221

```

```

1700 NEXT I:X=XM:RETURN 47
1710 'Omslaan om hor. as ***** 0
1720 ' 0
1730 GOSUB 1280 170
1740 IF GR=8 THEN GOSUB 4310 ELSE GOS
UB 4380 201
1750 GOSUB 1200 64
1760 RETURN 155
1770 'Omslaan om vert. as ***** 0
1780 ' 0
1790 GOSUB 1280 188
1800 IF GR=8 THEN GOSUB 4130 ELSE GOS
UB 4200 63
1810 GOSUB 1200 54
1820 RETURN 145
1830 'Spiegelen om hor. as ***** 0
1840 ' 0
1850 GOSUB 1280:IF GR=8 THEN GOSUB 47
40 ELSE GOSUB 4790 52
1860 GOSUB 1200:RETURN 127
1870 'Spiegelen om vert. as ***** 0
1880 ' 0
1890 GOSUB 1280:IF GR=8 THEN GOSUB 48
40 ELSE GOSUB 4900 233
1900 GOSUB 1200:RETURN 111
1910 '180 graden draaien ***** 0
1920 ' 0
1930 GOSUB 1280 174
1940 IF GR=8 THEN GOSUB 4130:GOSUB 43
10 ELSE GOSUB 4200:GOSUB 4380 117
1950 GOSUB 1200 68
1960 RETURN 159
1970 'Menu Edit aan/uit ***** 0
1980 ' 0
1990 IF HH THEN HH=0:LINE (144,21)-(2
55,192) 4,BF ELSE HH=-1:GOSUB 4480 170
2000 RETURN 124
2010 'Regel omlaag ***** 0
2020 ' 0
2030 GOSUB 1280:ERASE HS:DIM HS(4,8) 84
2040 FOR I=1 TO (GR-8)\2 251
2050 IF I=1 OR I=3 THEN HS((I+1)/2,1)
=SP(SN,I,8) 165
2060 FOR J=8 TO 2 STEP -1 194
2070 SP(SN,I,J)=SP(SN,I,J-1) 166
2080 NEXT J,I 56
2090 IF GR=8 THEN SP(SN,1,1)=0:GOTO 2
120 235
2100 FOR I=1 TO 4:IF (I=2 OR I=4) THE
N SP(SN,I,1)=HS(I/2,1) ELSE SP(SN,I,1)
=0 25
2110 NEXT I 85
2120 GOSUB 1200:RETURN 102
2130 'Regel omhoog ***** 0
2140 ' 0
2150 GOSUB 1280:ERASE HS:DIM HS(4,8) 92
2160 FOR I=1 TO (GR-8)\2 3
2170 IF I=2 OR I=4 THEN HS((I+1)/2,1)
=SP(SN,I,1) 129
2180 FOR J=1 TO 7 104
2190 SP(SN,I,J)=SP(SN,I,J+1) 122
2200 NEXT J,I 36
2210 IF GR=8 THEN SP(SN,1,8)=0:GOTO 2
240 8
2220 FOR I=1 TO 4:IF (I=1 OR I=3) THE
N SP(SN,I,8)=HS((I+1)/2,1) ELSE SP(SN
,I,8)=0 62
2230 NEXT I 93
2240 GOSUB 1200:RETURN 110
2250 'Regel rechts ***** 0
2260 ' 0
2270 GOSUB 1280:ERASE HS:DIM HS(4,8) 100
2280 FOR I=1 TO (GR-8)/2 0
2290 FOR J=1 TO 8:IF SP(SN,I,J) AND 1

```

```

THEN HS(I,J)=128 ELSE HS(I,J)=0
2300 SP(SN,I,J)=SP(SN,I,J) AND 254
2310 SP(SN,I,J)=SP(SN,I,J)/2
2320 NEXT J,I
2330 FOR I=3 TO 4:FOR J=1 TO 8:SP(SN,
I,J)=SP(SN,I,J) OR HS(I-2,J)
2340 NEXT J,I
2350 GOSUB 1200:RETURN
2360 'Regel links *****
2370 '
2380 GOSUB 1280
2390 ERASE HS:DIM HS(4,8):FOR I=1 TO
(GR-8)/2
2400 FOR J=1 TO 8:IF SP(SN,I,J) AND 1
28 THEN HS(I,J)=1 ELSE HS(I,J)=0
2410 SP(SN,I,J)=SP(SN,I,J) AND 127
2420 SP(SN,I,J)=SP(SN,I,J)*2
2430 NEXT J,I
2440 FOR I=1 TO 2:FOR J=1 TO 8:SP(SN,
I,J)=SP(SN,I,J) OR HS(I+2,J)
2450 NEXT J,I
2460 GOSUB 1200:RETURN
2470 'Inverteren *****
2480 '
2490 GOSUB 1280:FOR I=1 TO (GR-8)/2
2500 FOR J=1 TO 8
2510 SP(SN,I,J)=SP(SN,I,J) XOR 255
2520 NEXT J,I:GOSUB 1200:RETURN
2530 'Data op scherm *****
2540 '
2550 CLS:SCREEN 0
2560 PRINT "DATA Sprite Nr: ";SN;" Na
am: ";SN$(SN)
2570 RN=0:GOSUB 2650:LOCATE 0,5:PRINT
SD$
2580 FOR I=1 TO (GR-8)/2
2590 RN=RN+10:GOSUB 2650
2600 LOCATE 0,I+5:PRINT SD$
2610 NEXT I
2620 LOCATE 0,21:PRINT"Toets <E> voor
terug edit", "Toets <H> voor terug ho
ofdmenu:";
2630 B$=INPUT$(1):PRINT B$
2640 IF B$="E" THEN TH=0:SCREEN2:GOSUB
B 460:GOSUB 1200:RETURN ELSE IF B$="H
" THEN TH=1:RETURN ELSE BEEP:LOCATE 3
1,22:GOTO 2630
2650 'Data string maken *****
2660 '
2670 IF RN=0 THEN SD$=RIGHT$(STR$(300
00+SN*50),5)+" 'Data Sprite "+SN$(SN)
: RETURN
2680 SD$=RIGHT$(STR$(30000+SN*50+RN),
5)+" DATA "
2690 FOR J=1 TO 8
2700 SD$=SD$+RIGHT$("0"+HEX$(SP(SN,I,
J)),2)+" ":NEXT J
2710 SD$=LEFT$(SD$,LEN(SD$)-1)
2720 RETURN
2730 'Sprite opslaan *****
2740 '
2750 CLS:SCREEN 0:CLOSE
2760 PRINT "OPSLAAN Sprite Nr: ";SN,S
PC(17);"Naam ";SN$(SN);
2770 IF EO$="CAS:" THEN FS$=LEFT$(FSS
,6) ELSE FS$=SN$(SN)+" .SPR"
2780 LOCATE 0,6:PRINT SPC(78):LOCATE
0,6:PRINT"Naam ASCII bestand: ";FSS
2790 BEEP:PRINT"O.K. (J/N)?";
2800 I$=INKEY$:IF I$="J" THEN 2870 EL
SE IF I$="N" THEN 2810 ELSE 2800
2810 LOCATE 0,7:INPUT "Geef naam:";FS
$
2820 IF EO$="CAS:" THEN SN$(SN)=FSS:F

```

225  
175  
248  
44  
23  
50  
115  
0  
0  
178  
228  
164  
151  
118  
49  
104  
55  
120  
0  
52  
105  
29  
157  
0  
0  
227  
16  
126  
6  
73  
107  
95  
171  
48  
184  
0  
164  
111  
134  
168  
132  
144  
0  
0  
180  
42  
160  
248  
57  
81  
108

```

SS$=LEFT$(FSS,6):GOTO 2850
2830 I=INSTR(FSS," ")
2840 IF I=0 THEN SN$(SN)=FSS:FSS=LEFT
$(FSS,8)+" .SPR" ELSE SN$(SN)=LEFT$(FS
$,I-1):IF RIGHT$(FSS,4)<>".SPR" THEN
BEEP GOTO 2810
2850 SH$=SN$(SN)
2860 LOCATE 0,0:GOTO 2760
2870 OPEN EO$+FSS FOR OUTPUT AS#1:SB(
SN)=2
2880 RN=0:GOSUB 2650:LOCATE 0,12:PRIN
T SD$:PRINT#1,SD$:FOR I=1 TO (GR-8)/2
2890 RN=RN+10:GOSUB 2650
2900 LOCATE 0,I+12:PRINT SD$
2910 PRINT#1,SD$
2920 NEXT I
2930 SB(SN)=2:CLOSE#1
2940 OPEN"GRP:" FOR OUTPUT AS #1:GOTO
2620
2950 'Sprite lezen *****
2960 '
2970 SCREEN 0:WIDTH 38:CLOSE
2975 IF EO$="CAS:" THEN PRINT "DATA V
AN CASSETTE Sprite Nr: ";SN,SPC(21);"N
aam: ";SN$(SN): GOTO 2990
2980 PRINT"DATA VAN DISK Sprite Nr: "
;SN,SPC(21);"Naam: ";SN$(SN)
2990 IF EO$="CAS:" THEN 3030
3000 LOCATE 0,4:PRINT"Op deze schijf
aanwezige DATA:"
3010 FILES "*.SPR"
3020 LOCATE 0,20:PRINT SPC(76);
3030 LOCATE 0,20:INPUT "Welke wil je
inlezen:";FSS
3040 IF EO$="CAS:" THEN FSS=LEFT$(FSS
,6):GOTO 3070
3050 I=INSTR(FSS," ")
3060 IF I=0 THEN SN$(SN)=LEFT$(FSS,8)
:FSS=LEFT$(FSS,8)+" .SPR" ELSE SN$(SN)
=LEFT$(FSS,I-1)
3070 SH$=SN$(SN)
3080 OPEN EO$+FSS FOR INPUT AS#1
3090 FOR I=4 TO 22:LOCATE 0,I:PRINT S
PC(38);:NEXT I
3100 LINE INPUT#1,SD$
3110 LOCATE 0,5:PRINT SD$
3120 FOR I=1 TO (GR-8)/2
3130 IF EOF(1) THEN 3180
3140 LINE INPUT#1,SD$
3150 LOCATE 0,I+5:PRINT SD$
3160 GOSUB 3200
3170 NEXT I
3180 CLOSE#1
3190 OPEN"GRP:" AS#1:GOTO 2620
3200 'Data uit string halen *****
3210 '
3220 FOR J=1 TO 8:SP(SN,I,J)=VAL("&H"
+MID$(SD$,9+3*J,2)):NEXT J
3230 RETURN
3240 'Sprite mengen *****
3250 '
3260 SCREEN0:CLS:
3270 PRINT"SAMENVOEGEN"
3280 LOCATE 0,3:PRINT"Huidige Sprite
Nr: ";SN,SPC(15);"Naam: ";SN$(SN)
3290 LOCATE 0,6:PRINT"Verder aanwezig
e Sprites:"
3300 X=0:Y=8:FOR I=0 TO 19
3310 IF SB(I)=0 OR I=SN THEN 3340
3320 LOCATE X,Y:PRINT I;" ";SN$(I)
3330 IF X=20 THEN X=0:Y=Y+1 ELSE X=0
3340 NEXT I
3350 LOCATE 0,21:PRINT"R: Return edit
mode

```

184  
108  
42  
204  
71  
176  
64  
79  
83  
210  
104  
139  
23  
0  
0  
190  
137  
240  
31  
180  
133  
213  
49  
7  
99  
235  
195  
139  
122  
204  
192  
237  
71  
216  
113  
81  
104  
37  
219  
0  
0  
52  
138  
0  
0  
195  
42  
63  
104  
27  
16  
7  
110  
99  
42

```

3360 LOCATE 0,20:INPUT"Welk nummer wi
lt u samenvoegen:":SS$ 12
3370 IF SS$="R" THEN 3430 4
3380 SS=VAL(SS$):IF SS<0 OR SS>20 THE
N BEEP:GOTO 3360 51
3390 IF SB(SS)=0 THEN 3430 161
3400 FOR I=1 TO (GR-8)/2:FOR J=1 TO 8
124
3410 SP(SN,I,J)=SP(SN,I,J) OR SP(SS,I
J) 35
3420 NEXT J,I 47
3430 SCREEN2:GOSUB 460:GOSUB 1200: RE
TURN 196
3440 'Terug hoofdmenu ***** 0
3450 ' 0
3460 TH=1:RETURN 204
3470 'Sprite vullen ***** 0
3480 ' 0
3490 C=2:X=4:Y=175:KK$="Vullen (J/N)?
":GOSUB 1650 60
3500 I$=INKEY$:IF I$<>"J" AND I$<>"N"
THEN 3500 213
3510 IF I$="N" THEN 3560 85
3520 LINE (8,22)-(7+8*GR,21+8*GR),VC,
BF 69
3530 LINE (8,154)-(7+GR,153+GR),VC,BF
33
3540 FOR I=1 TO 4:FOR J=1 TO 8 43
3550 SP(SN,I,J)=255:NEXT J,I 167
3560 RETURN 153
3570 'Sprite wissen ***** 0
3580 ' 0
3590 C=2:X=4:Y=175:KK$="Wissen (J/N)?
":GOSUB 1650 160
3600 I$=INKEY$:IF I$<>"J" AND I$<>"N"
THEN 3600 4
3610 IF I$="N" THEN 3650 86
3620 GOSUB 1280 167
3630 FOR I=1 TO 4:FOR J=1 TO 8 42
3640 SP(SN,I,J)=0:NEXT J,I 43
3650 RETURN 152
3660 'Error routine ***** 0
3670 ' 0
3680 IF ERL=2870 AND ERR=56 THEN BEEP
:RESUME 2810 160
3690 IF ERL=3080 THEN BEEP:RESUME 302
0 123
3700 SCREEN 0:COLOR 15,4,4:ON ERROR G
OTO 95
3710 'Overzicht gemaakte sprites **** 0
3720 ' 0
3730 SCREEN 2,2: 106
3740 LINE (0,0)-(255,8),13,BF 199
3750 LINE (0,9)-(22,183),2,BF 203
3760 LINE (23,9)-(118,183),6,BF 171
3770 LINE (119,9)-(206,183),1,BF 108
3780 LINE (207,9)-(255,183),2,BF 194
3790 LINE (0,184)-(255,192),3,BF 165
3800 C=13:X=3:Y=0:KK$="Nr. Naam"+S
PACE$(31)+"Vorm"+SPACE$(13)+"Gesaved"
:GOSUB 1650 171
3810 LINE (23,0)-(23,183),15 122
3820 LINE (119,0)-(119,183),15 98
3830 LINE (207,0)-(207,183),15 27
3840 LINE (0,8)-(255,8),15 163
3850 YH=12:T=-1:FOR ST=0 TO 19 69
3860 IF SB(ST)=0 THEN 3960 114
3870 T=T+1:C=2:X=5:Y=YH+8*T 125
3880 IF (T+1)MOD3=0 THEN YH=YH+2 8
3890 GOSUB 4000 90
3900 KK$=RIGHT$("0"+STR$(ST),2):GOSUB
1650 182
3910 C=6:DX=(TMOD3):X=30+18*DX 215
3920 KK$=SN$(ST):GOSUB 1650 11
3930 PUT SPRITE ST,(X+100+9*DX,Y),15,
ST 242

```

```

3940 IF SB(ST)=2 THEN KK$="Ja" ELSE K
K$="Nee" 102
3950 C=2:X=214:GOSUB 1650 241
3960 NEXT ST 211
3970 C=3:COLOR 1:X=40:Y=184:BEEP:KK$=
"Druk een toets":GOSUB 1650 124
3980 B$=INPUT$(1) 131
3990 COLOR 15:SCREEN0:RETURN 165
4000 RESTORE 4090:SS$="":FOR I=1 TO 4:
FOR J=1 TO 8 20
4010 SS=SS+CHR$(SP(ST,I,J)) 22
4020 NEXT J,I:SPRITE$(ST)=SS:RETURN 142
4030 X=8:Y=22:XP=8:YP=154:RETURN 31
4040 X=8:Y=86:XP=8:YP=162:RETURN 132
4050 X=72:Y=22:XP=16:YP=154:RETURN 197
4060 X=72:Y=86:XP=16:YP=162:RETURN 50
4070 'Data hulpsprite ***** 0
4080 ' 0
4090 DATA FF,80,80,80,80,80,80,80 161
4100 DATA 80,80,80,80,80,80,80,FF 124
4110 DATA FF,01,01,01,01,01,01,01 43
4120 DATA 01,01,01,01,01,01,01,FF 181
4130 'Omslaan vert. as gr=8 ***** 0
4140 ' 0
4150 FOR J=1 TO 8:HS(1,J)=SP(SN,1,J):
SP(SN,1,J)=0 233
4160 NEXT J 113
4170 FOR J=1 TO 8: FOR K=0 TO 7 148
4180 IF HS(1,J) AND 2^K THEN SP(SN,1,
J)=SP(SN,1,J) OR 2^(7-K) 104
4190 NEXT K,J:RETURN 93
4200 'Omslaan vert. as gr=16 ***** 0
4210 ' 0
4220 FOR I=1 TO 4:FOR J=1 TO 8 32
4230 HS(I,J)=SP(SN,I,J):SP(SN,I,J)=0 249
4240 NEXT J,I 50
4250 FOR I=1 TO 4 38
4260 IF I<3 THEN I1=I+2 ELSE I1=I-2 129
4270 FOR J=1 TO 8:FOR K=0 TO 7 93
4280 IF HS(I,J) AND 2^K THEN SP(SN,I1
,J)=SP(SN,I,J) OR 2^(7-K) 165
4290 NEXT K,J,I 8
4300 RETURN 132
4310 'Omslaan hor. as gr=8 ***** 0
4320 ' 0
4330 FOR J=1 TO 8:HS(1,J)=SP(SN,1,J) 173
4340 SP(SN,1,J)=0:NEXT J 80
4350 FOR J=1 TO 8 118
4360 SP(SN,1,9-J)=HS(1,J) 252
4370 NEXT J:RETURN 74
4380 'Omslaan hor. as gr=16 ***** 0
4390 ' 0
4400 FOR I=1 TO 4 27
4410 FOR J=1 TO 8:HS(I,J)=SP(SN,I,J) 209
4420 SP(SN,I,J)=0:NEXT J,I 34
4430 FOR I=1 TO 4 36
4440 IF INT(I/2)=I/2 THEN I1=I-1 ELSE
I1=I+1 167
4450 FOR J=1 TO 8 120
4460 SP(SN,I1,9-J)=HS(I,J) 152
4470 NEXT J,I:RETURN 69
4480 'Edit menu opbouwen ***** 0
4490 ' 0
4500 LINE(144,21)-(255,192),6,BF 89
4510 C=6:X=146:Y=26:KK$="A,S: Omslaan
v/h as":GOSUB 1650 15
4520 Y=34:KK$="Q,W: Spiegel v/h as":G
OSUB 1650 154
4530 Y=42:KK$="R,F: Regel naar b/o":G
OSUB 1650 16
4540 Y=50:KK$="G,H: Regel naar l/r":
GOSUB 1650 47
4550 Y=58:KK$="D: Draai 180 graden":G
OSUB 1650 29

```



```

4560 Y=66:KK$="E: Draai 90 graden":G
OSUB 1650 151
4570 Y=74:KK$="I: Inverteren":GOSUB 1
650 174
4580 Y=82:KK$="P: Data op scherm":GOS
UB 1650 141
4590 Y=90:KK$="O: Data opslaan":GOSUB
1650 52
4600 Y=98:KK$="T: Data invoeren":GOSU
B 1650 249
4610 Y=106:KK$="L: Data inlezen":GOSU
B 1650 21
4620 Y=114:KK$="K: Sprite mengen":GOS
UB 1650 235
4630 Y=122:KK$="V: Sprite vullen":GOS
UB 1650 173
4640 Y=130:KK$="C: Sprite wissen":GOS
UB 1650 202
4650 Y=138:KK$="Z: Terug hoofdmenu":G
OSUB 1650 245
4660 Y=146:KK$="M: Menu Aan/Uit":GOSU
B 1650 190
4670 Y=154:KK$="B,N: Kleur vg/ag":GOS
UB 1650 220
4680 Y=170:KK$="SPATIE: Punt omslaan"
:GOSUB 1650 43
4690 Y=178:KK$="CURSOR: Cursor <":GO
SUB 1650 213
4700 RETURN 140
4710 'ON STOP Routine ***** 0
4720 ' 0
4730 CLOSE:OPEN"GRP:"AS#1:COLOR 15,4,
4:RETURN 320 78
4740 'Spiegelen hor. as gr=8 ***** 0
4750 ' 0
4760 FOR J=1 TO 4 61
4770 SP(SN,1,9-J)=SP(SN,1,J) 80
4780 NEXT J:RETURN 85
4790 'Spiegelen hor. as gr=16 ***** 0
4800 ' 0
4810 FOR I=1 TO 3 STEP 2:FOR J=1 TO 8 182
4820 SP(SN,I+1,9-J)=SP(SN,I,J) 28
4830 NEXT J,I:RETURN 65
4840 'Spiegelen vert. as gr=8 ***** 0
4850 ' 0
4860 FOR J=1 TO 8: SP(SN,1,J)=SP(SN,1
J) AND 240:FOR K=7 TO 4 STEP -1 253
4870 IF SP(SN,1,J) AND 2^K THEN SP(SN
1,J)=SP(SN,1,J) OR 2^(7-K) 251
4880 NEXT K,J 98
4890 RETURN 169
4900 'Spiegelen vert. as gr=16 ***** 0
4910 ' 0
4920 FOR I=1 TO 2:FOR J=1 TO 8:SP(SN,
I+2,J)=0 241
4930 FOR K=7 TO 0 STEP -1 169
4940 IF SP(SN,I,J) AND 2^K THEN SP(SN
I+2,J)=SP(SN,I+2,J) OR 2^(7-K) 155
4950 NEXT K,J,I:RETURN 68
4960 'Data invoeren toetsenbord ***** 0
4970 ' 0
4980 SCREEN 0:CLS 202
4990 PRINT "INVOEREN DATA Sprite Nr:
":SN 92
5000 PRINT TAB(21)"Naam:":SN$(SN) 92
5010 LOCATE 0,4:PRINT"In welk taistel
sel wilt u invoeren?":PRINT 100
5020 PRINT TAB(10)"1 Decimaal" 57
5030 PRINT TAB(10)"2 Hexa-decimaal" 56
5040 PRINT:PRINT TAB(15)"Uw keuze:_" 48
5050 LOCATE 24,9:G$="DUMMY" 21
5060 I$=INKEY$:IF I$="" THEN 5060 187
5070 PRINT I$ 166
5080 IF I$="1" THEN G$="" 186

```

```

5090 IF I$="2" THEN G$="&H" 20
5100 IF G$="DUMMY" THEN 5050 171
5110 X=0:Y=15:FOR I=1 TO (GR-8)/2 76
5120 LOCATE X,Y:PRINT"DATA ":X=X+5 176
5130 FOR J=1 TO 8 109
5140 LOCATE 0,22:PRINT SPC(11):LOCATE
0,22:INPUT"DATA ":SD$ 48
5150 SD=VAL(G$+SD$) 95
5160 LOCATE X,Y:IF I$="2" THEN PRINT
RIGHT$("00"+SD$,2): ELSE PRINT SD$: 135
5170 IF J<8 THEN PRINT",":X=POS(0) 246
5180 SP(SN,I,J)=SD 18
5190 NEXT J:Y=Y+1:X=0:NEXT I 120
5200 GOTO 2620 219
5210 'Voor/achter kleur instellen**** 0
5220 ' 0
5230 GOSUB 1280 164
5240 LINE (0,172)-(140,192),2,BF 171
5250 C=2:X=4:Y=180:KK$="Geef kleur nu
mmer:":GOSUB 1650 250
5260 I$=INKEY$:IF I$="" THEN 5260 253
5270 G$=I$ 37
5280 I$=INKEY$:IF I$="" THEN 5280 67
5290 IF I$=CHR$(13) THEN 5300 ELSE G$
=G$+I$ 188
5300 K=VAL(G$):IF K=0 OR K>15 THEN 52
50 191
5310 IF T=20 THEN VC=K ELSE AC=K 28
5320 LINE (0,172)-(140,192),2,BF 167
5330 GOSUB 1280:GOSUB 1200:RETURN 129
5340 '90 graden draaien ***** 0
5350 ' 0
5360 GOSUB 1280 175
5370 IF GR=8 THEN GOSUB 5400 ELSE GOS
UB 5470 14
5380 GOSUB 1200 69
5390 RETURN 160
5400 '90 graden draaien gr=8 ***** 0
5410 ' 0
5420 ERASE HS:DIM HS(1,8):FOR I=1 TO
8: HS(1,I)=SP(SN,1,I):SP(SN,1,I)=0:NE
XT I 233
5430 FOR I=1 TO 8:FOR K=0 TO 7 76
5440 IF HS(1,I) AND 2^K THEN SP(SN,1,
K+1)=SP(SN,1,K+1) OR 2^(8-I) 52
5450 NEXT K,I 69
5460 RETURN 153
5470 '90 graden draaien gr=16 ***** 0
5480 ' 0
5490 ERASE HS:DIM HS(4,8) 254
5500 FOR I=1 TO 4:FOR J=1 TO 8 33
5510 HS(I,J)=SP(SN,I,J):SP(SN,I,J)=0 250
5520 NEXT J,I 51
5530 FOR J=1 TO 8:FOR K=0 TO 7 88
5540 IF HS(1,J) AND 2^K THEN SP(SN,2,
K+1)=SP(SN,2,K+1) OR 2^(8-J) 217
5550 IF HS(2,J) AND 2^K THEN SP(SN,4,
K+1)=SP(SN,4,K+1) OR 2^(8-J) 146
5560 IF HS(3,J) AND 2^K THEN SP(SN,1,
K+1)=SP(SN,1,K+1) OR 2^(8-J) 162
5570 IF HS(4,J) AND 2^K THEN SP(SN,3,
K+1)=SP(SN,3,K+1) OR 2^(8-J) 91
5580 NEXT K,J 93
5590 RETURN 164
5600 'Externe opslag kiezen ***** 0
5610 ' 0
5620 IF EO$="CAS:" THEN EO$="A:" ELSE
EO$="CAS:" 207
5630 RETURN 148

```

## Kategorie: utility's

# Joysor

Hoewel op MSX-computers de cursortoetsen al op een hele handige manier zijn opgesteld, is een joystick voor sommige programma's toch niet te versmaden. Helaas, wat blijkt? Er zijn nog ontzettend veel programma's die alleen naar cursortoetsen luisteren en de joystick volkomen negeren.

Uw lijden is verleden tijd: hier is JOYSOR, een ML-programma om de cursor met de joystick te besturen.

### Truuk

JOYSOR bevat de inmiddels bekende Basic-lader, die de ML op zijn plaats zet, naar keuze onder de Clear-grens of in de Play-wachtrij. Ook het eigenlijke assembler-programma staat in de Data-regels.

Het programma doet iets, dat alleen vanuit ML mogelijk is (en dat geeft het programmeren in ML zijn eigen charme): het houdt de computer voor de gek. Elke keer als de joystick een richting aangeeft of de vuurknop gebruikt wordt, stuurt de ML de computer een bericht om te vertellen dat er een cursor-toets werd ingedrukt! Deze weet niet beter en reageert precies zoals hij op een cursor-toets zou reageren.

Dit betekent, dat er voor de computer geen verschil meer is tussen de cursortoetsen en de joystick, dus alle Basic-programma's - en sommige ML-programma's ook! - die normaal gesproken alleen met de cursortoetsen werken, zullen nu ook de joystick gehoorzamen. Dit geldt ook voor Basic zelf, bijvoorbeeld tijdens het schrijven en bewerken van programma's: de cursor is met de joystick over het scherm te bewegen. Natuurlijk blijven de cursortoetsen zelf ook gewoon actief.

### Snelheid

Bij het schrijven van dit programma ontdekten we een verschil tussen MSX-1 en MSX-2 computers. Het probleem van het langzame toetsenbord,

waar een aantal MSX-1's aan lijden, is bij de Sony HB-F500P op de redactie definitief verholpen. Het bleek om een kleine ingreep te gaan, die we maar meteen in JOYSOR hebben ingebouwd. Op MSX-1 computers is de snelheid van het toetsenbord met behulp van JOYSOR instelbaar tussen 1 (zeer snel) en 3 (normaal).

Er is nog een ander voordeeltje. Als een toets wat langer ingedrukt wordt, begint hij normaal gesproken pas na een tijdje te herhalen. Bij de joystick is dat niet zo, zodat de cursor sneller en direkter beweegt. Het nadeel hiervan is, dat de cursor bij een hoge ingestelde snelheid wat moeilijk voor precisiewerk te gebruiken is, tenminste met de joystick. Maar dat zal afhangen van de handigheid van degene die hem gebruikt; op de redactie vonden we snelheid 2 het meest comfortabel.

### Werking

Zoals gezegd is JOYSOR een typisch ML-programma: het werkt op de achtergrond, dus de gebruiker merkt alleen dat het aanwezig is als hij iets bepaalds wil of doet, de joystick bewegen bijvoorbeeld. Het maakt gebruik van de TIMER-hook. Deze hook wordt voortdurend aangeroepen, zo'n 50 keer per seconde, door de *interrupt-routine*. Dit is een stuk programma in het ROM dat onder andere het toetsenbord leest, de muziek op peil houdt, enzovoorts. Het verricht in feite huishoudelijk werk, allerlei

taken die doorlopend verricht moeten worden. Als JOYSOR geïnstalleerd is, wordt het adres START (zie regel 1070) dus 50 keer per seconde aangeroepen.

Hier wordt gekeken of de waarde op adres hex F3F6 gelijk is aan 3 - voor MSX-1 - of 1, voor MSX-2 computers. Volgens de MSX-specificaties is op adres &h2D te vinden om welke MSX-versie het gaat: oorspronkelijk stond daar een 0, maar op MSX-2 computers is dit een 1. Latere versies zullen daar steeds een hoger nummer hebben staan. JOYSOR test zelf of het met een MSX-2 te maken heeft en verandert de waarde 3 uit de Data-regels dan in een 1.

Op geheugenplaats F3F6 telt de computer via de interrupt-routine steeds van 3 (of 1) tot 0. Als de teller 0 is, wordt er onderzocht of er soms een toets is ingedrukt; in alle andere gevallen doet JOYSOR helemaal niets.

Als de teller 3 (1) is gaat het programma verder bij JOYST, regel 1160. Hier wordt eerst de teller op de waarde gezet die in WACHT staat; WACHT is adres FDA3, zie regel 1030. Voor MSX-1 computers is dit precies de door ons gekozen snelheid: 3 was normaal, 1 zeer snel; voor MSX-2 is deze waarde altijd 1. Is de teller eenmaal aangepast, dan worden de beide vuurknoppen gecontroleerd. Blijkt een van beide ingedrukt, dan springt het programma naar TRIG - zie 1600 -, anders gaat het gewoon verder bij JOY2.

Bij JOY2 wordt de stand van de joystick ingelezen via de BIOS-routine GTSTCK, wat voor 'Get Stick' staat. Als dat niets oplevert - oftewel twee nullen - dan springt JOYSOR terug naar END in regel 1130 en keert terug naar de interrupt routine, die gewoon verder gaat waar hij gebleven was.

### Bit-patroon

BEWEEG vertaalt de joystick-stand - een getal tussen 1 en 8 - naar een patroon van 4 bits, respectievelijk van links naar rechts: links, omlaag, rechts, omhoog.

Dit gebeurt met behulp van TABEL, regel 1790. Stand 1 blijft 1, 2 wordt 2, 3 wordt 2 en zo verder tot 8 wordt 9. Een bit in het patroon betekent een richting, dus stand 6 - links-onder - wordt vertaald naar binair 1100, oftewel 12. Tussen BIT en BIT2 wordt dit patroon doorlopen en voor elk bit wordt er een letter naar de computer doorgestuurd.

Deze letters staan in de Data-regels 2010 tot 2040: voor elke richting staat hier de Ascii-kode voor een (cursor-)knop. Deze waarden zijn natuurlijk makkelijk te veranderen, zodat de werking van JOYSOR in een handomdraai aan te passen is, bijvoorbeeld voor programma's waarin de cursor met de Z, X, K en M-toetsen bestuurd wordt.

### Toets-wachtrij

De juiste Ascii-kodes worden door JOYSOR naar de computer gestuurd door ze in de toets-wachtrij te zetten. Iedereen heeft wel eens gemerkt, dat een MSX-computer een aantal toetsen vooruit kan onthouden, bijvoorbeeld als er een heleboel tegelijk worden ingedrukt. Al deze toetsen worden opgeslagen in een wachtrij en de computer haalt ze er een voor een weer uit. JOYSOR lokt de computer in de val door zelf toetsen aan de wachtrij toe te voegen en nog wel aan het einde, zodat ze ook meteen als eerste weer aan de beurt komen. Dit gebeurt met de routine INRIJ, zie regel 1640 en verder. Deze routine staat precies zo in het MSX-ROM.

### Interrupt-hook

Zoals gezegd buigt JOYSOR de TIMER-hook op adres FD9F, zie regel 1020, om. Dit is een nogal riskante hook, omdat diskdrives en andere randapparatuur er nog al eens gebruik van maken. Problemen hiermee worden door JOYSOR omzeild op de volgende manier: achter het programma worden 8 bytes gereserveerd.

De eerste drie bevatten een Call-instructie naar START, de op-kode van Call is CD, zie regel 200: de variabele AD bevat de eerste vrije plaats na het

programma. De volgende 5 bytes zijn een kopie van de oorspronkelijke hook, zodat die nog steeds actief is. De hook zelf, tenslotte, wordt omgeleid naar de CALL START-instructie.

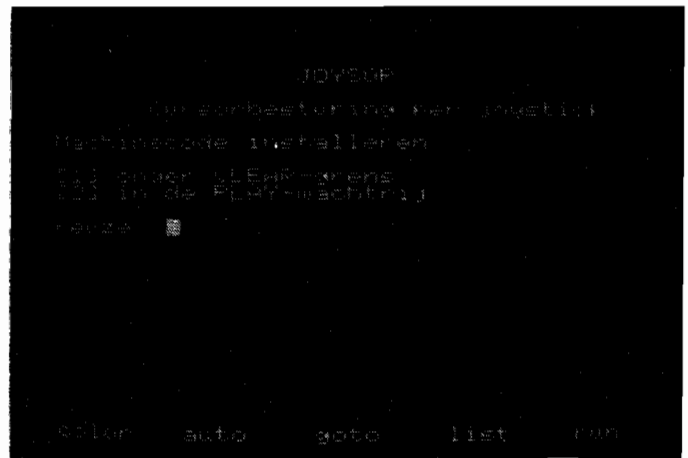
**Let op!**

Dit betekent, dat de computer zal vastlopen, wanneer JOYSOR twee keer achter elkaar wordt uitgevoerd! ML-routines gaan zichzelf dan herhaaldelijk aanroepen en juist bij interrupt-routines is dat linke

soep, waarbij niet zelden de computer het bijtje erbij neergooit. Oppassen geblazen dus!

Een tweede gevolg is, dat JOYSOR zich niet simpelweg laat uitschakelen door de hook te elimineren met POKE &hFD9F,201. Dit gaat soms goed, maar soms ook behoorlijk verkeerd, afhankelijk van aangesloten randapparatuur en dergelijke.

Al met al is JOYSOR een goed voorbeeld van wat er met machinetaal-programma's mogelijk is.



```

10  '          JOYSOR
20  '
30  ' MSX Computer Magazine
40  '
50  SCREEN 0: WIDTH 39
60  LOCATE 15,2:PRINT"JOYSOR":LOCATE 6,4:PRINT"Cursorbesturing per joystick"
70  PRINT:PRINT"Machinecode installeren":PRINT:PRINT"[1] onder CLEAR-grens":PRINT
   "[2] in de PLAY-wachtrij":PRINT:PRINT"Keuze: ";
80  A=VAL(INPUT$(1)):IF A<1 OR A>2 THEN 80
90  PRINT A:IF A=2 THEN ST=&HF9F5+65536!:GOTO 120
100 CLEAR 200,PEEK(&HFC4A)+256*PEEK(&HFC4B)-&H71-9
110 ST=PEEK(&HFC4A)+256*PEEK(&HFC4B)
120 L=&H71:CS=0
130 FOR I=0 TO L: READ A$: CS=CS+ASC(LEFT$(A$,1))+ASC(RIGHT$(A$,1))
140 IF A$="*0" THEN READ A$:A=ST+VAL("&H0"+A$):POKEST+I,A-INT(A/256)*256:I=I+1:P
OKEST+I,INT(A/256):GOTO 160
150 POKE ST+I,VAL("&h"+A$)
160 NEXT I
170 IF CS<>11183 THEN PRINT:PRINT"Fout in de DATA-regels!":PRINT"Eerst verbeter
   n!":STOP
180 CLS:PRINT"Machinecode geïnstalleerd op "HEX$(ST)
190 FOR I=1 TO 5:READ A: POKE ST+L+I,A:NEXT
200 AD=ST+L+6:HOOK=&HFD9F
210 FOR I=0 TO 4:POKE AD+3+I,PEEK(HOOK+I):NEXT
220 POKE AD,&HCD
230 POKE AD+1,ST-INT(ST/256)*256
240 POKE AD+2,INT(ST/256)
250 POKE HOOK,&HC9
260 POKE HOOK+1,AD-INT(AD/256)*256
270 POKE HOOK+2,INT(AD/256)
280 POKE HOOK+4,1: 'startsnelheid
290 POKE HOOK,&HC3
300 PRINT:PRINT"De cursor is nu bestuurbaar met een joystick in een van de in
   gangen."
310 PRINT"Daardoor zullen alle programma's die via de cursortoetsen werken, nu
   ook met een joystick bestuurbaar zijn."
320 IF PEEK(&H2D)=1 THEN POKE ST+5,1: GOTO 370: 'kennelijk MSX 2: geen snelheid
330 PRINT:PRINT"De cursor en het toetsenbord kunnen sneller gezet worden."
340 PRINT:PRINT"Geef de snelheid van de cursor:":PRINT:PRINT"[1] zeer snel":PRIN
   T"[2] snel":PRINT"[3] normaal":PRINT:PRINT"Keuze: ";
350 A=VAL(INPUT$(1)):IF A<1 OR A>3 THEN 350
360 PRINT A:POKE HOOK+4,A
370 END
1000 ' het programma in assembler
1010 '
1020 '          : ' HOOK          EQU    #FD9F
1030 '          : ' WACHT          EQU    HOOK+4
1040 '          : ' GTSTCK        EQU    #D5
1050 '          : ' STRIG         EQU    #D8
1060 '
1070 DATA F5 : ' START          PUSH   AF          beveilig Accu
1080 '
1090 DATA 3A,F6,F3: '          LD     A,(#F3F6)  teller
1100 DATA FE,03 : '          CP     3          3 (MSX2: 1)?

```

0  
0  
0  
0  
227  
16  
23  
150  
194  
59  
186  
177  
19  
85  
240  
211  
184  
218  
57  
248  
111  
246  
245  
20  
186  
79  
242  
49  
86  
101  
24  
173  
190  
65  
106  
159  
186  
0  
0  
208  
225  
184  
242  
0  
116  
0  
136  
49

1110	DATA 28,02	:	JR	Z,JOYST	ja	249
1120						0
1130	DATA F1	:	POP	AF	nee, terug	108
1140	DATA C9	:	RET			150
1150						0
1160	DATA 3A,A3,FD	:	LD	A,(WACHT)	stel snelheid in	110
1170	DATA 32,F6,F3	:	LD	(#F3F6),A		71
1180	DATA 06,02	:	LD	B,2	2 joysticks testen	244
1190	DATA C5	:	PUSH	BC		12
1200	DATA 78	:	LD	A,B		49
1210	DATA CD,D8,00	:	CALL	STRIG	trigger?	10
1220	DATA C1	:	POP	BC		30
1230	DATA B7	:	OR	A	ingedrukt?	111
1240	DATA 20,2E	:	JR	NZ,TRIG	ja!	228
1250	DATA 10,F5	:	DJNZ	JOY1	nee,opnieuw	29
1260						0
1270	DATA 06,02	:	LD	B,2	twee joysticks	55
1280	DATA C5	:	PUSH	BC		36
1290	DATA 78	:	LD	A,B		76
1300	DATA CD,D5,00	:	CALL	GTSTCK		123
1310	DATA C1	:	POP	BC		29
1320	DATA B7	:	OR	A	richting?	9
1330	DATA 20,04	:	JR	NZ,BEWEEG	ja: doe het	56
1340	DATA 10,F5	:	DJNZ	JOY2		120
1350						0
1360	DATA 18,DC	:	JR	END	niets gevonden	96
1370						0
1380	DATA 21,*0,69	:	LD	HL,TABEL-1	vertaal richting	73
1390	DATA 16,00	:	LD	D,0	naar bitpatroon	54
1400	DATA 5F	:	LD	E,A	uit tabel	29
1410	DATA 19	:	ADD	HL,DE		205
1420	DATA 7E	:	LD	A,(HL)		77
1430						0
1440	DATA 06,04	:	LD	B,4	4 bits testen	92
1450	DATA 21,*0,72	:	LD	HL,TABEL+8	schuif rechter bit	243
1460	DATA CB,3F	:	SRL	A	naar carry	41
1470	DATA 30,07	:	JR	NC,BIT2	nul: volgende bit	178
1480						0
1490	DATA F5	:	PUSH	AF		174
1500	DATA 7E	:	LD	A,(HL)	haal letter uit tabel	164
1510	DATA B7	:	OR	A	=0?	70
1520	DATA C4,*0,51	:	CALL	NZ,INRIJ	nee: zet in wachtrij	26
1530	DATA F1	:	POP	AF	herstel A	220
1540						0
1550	DATA 23	:	INC	HL	verhoog tabelwijzer	30
1560	DATA 10,F2	:	DJNZ	BIT	herhaal totdat b=0	163
1570						0
1580	DATA 18,BF	:	JR	END	klaar.	177
1590						0
1600	DATA 3A,*0,76	:	LD	A,(TABEL+12)	neem laatste getal uit	213
1610	DATA CD,*0,51	:	CALL	INRIJ	tabel en zet in rij	196
1620	DATA 18,B7	:	JR	END	spring terug	51
1630						0
1640	DATA E5	:	PUSH	HL		54
1650	DATA 2A,F8,F3	:	LD	HL,(#F3F8)	wijzer naar plaats in rij	92
1660	DATA 77	:	LD	(HL),A	sla A op in rij	58
1670	DATA 23	:	INC	HL	verhoog rijwijzer	251
1680	DATA 7D	:	LD	A,L		123
1690	DATA FE,18	:	CP	#18	L-register = &h18?	177
1700	DATA 20,03	:	JR	NZ,IN1	nee, alles OK	73
1710	DATA 21,F0,FB	:	LD	HL,#BF0	anders terug naar begin rij	167
1720	DATA 3A,FA,F3	:	LD	A,(#F3FA)	L-byte van einde-wijzer	91
1730	DATA BD	:	CP	L	gelijk aan wijzer?	44
1740	DATA 28,03	:	JR	Z,INEND	ja, doe niets	139
1750	DATA 22,F8,F3	:	LD	(#F3F8),HL	nee, sla nieuwe wijzer op	113
1760	DATA E1	:	POP	HL	herstel HL	146
1770	DATA C9	:	RET		einde routine	130
1780						0
1790	DATA 01,03,02	:	DEFB	1,3,2,6,4,12,8,9		30
1800	DATA 06,04,0C,08,09	:				75
1810						0
2000						0
2010	DATA 30	:			omhoog	108
2020	DATA 28	:			rechts	237
2030	DATA 31	:			omlaag	164
2040	DATA 29	:			links	95
2050	DATA 32	:			vuurknop	136

**CATEGORIE: TOEPASSINGEN**

**Digiklok**



We zullen de laatsten zijn om te beweren dat dit een erg nuttig programma is. Integendeel, een polshorloge - of een wekker - lijkt ons veel gemakkelijker. Bovendien is dan de computer niet bezet en dus beschikbaar voor echt nuttige zaken, zoals Zaxxon of Mandragore. Desondanks wilden we u dit stukje moderne techniek niet onthouden. We stellen u dan ook voor aan de MSX als digitale klok.

Toch is dit stukje huisvlijt, dat we van een van onze lezers kregen toen we hem alweer een hele tijd geleden tegenkwamen op de HCC dagen, zonder meer leuk genoeg om op te nemen. Het zit uitstekend in elkaar. De nauwkeurigheid zal op de lange duur echter wel wat te wensen overlaten, een MSX is per slot van rekening geen kwarts-uurwerk.

**Uitdaging**

Aan dit programma willen we meteen een kleine uitdaging voor de programmeurs onder onze lezers vastknopen. Namelijk, wie schrijft het beste, mooiste, kortste of wat-dan-oookste klokprogramma voor MSX. Laat uw programmeer-

talenten eens los op het idee 'klok'. Of 'wekker', voor ons part.

Er zijn heel wat mogelijkheden. Wat te denken van een grootvadersklok, of een repeeterwekker? Wijzers in plaats van cijfers mag natuurlijk ook.

Of een tikkende pendule met BigBen geluid. Speel er eens mee, bedenk eens wat.

**Spelregels**

De spelregels zijn simpel. Er zijn namelijk geen prijzen te winnen. De programma's mogen natuurlijk niet al te lang worden, hooguit een halve pagina MCM. Insturen natuurlijk alleen op cassette of diskette, want intikken vin-

den wij ook niet echt leuk. Als u uw disk of cassette terug wilt hebben dan kan dat, als u tenminste een voldoende gefrankeerde en aan uzelf geadresseerde envelop mees- tuurt.

Wij op onze beurt beloven al die klokken, wekkers en zandlopers te bekijken. De beste - of de vreemdste - eks- emplaren zullen in MCM ge- plaatst worden. Wie stuurt ons een klok?

Insturen naar:

MSX Computer Magazine  
Postbus 1392  
1000 BJ Amsterdam

Als u in de linkerbovenhoek van de envelop even het motto 'TikTak' vermeldt, dan weten wij wat erin zit en wordt de rest van de uitgeverij weer eens nieuwsgierig naar wat die computergekken nu weer bedacht hebben. Kortom, houdt de redactie bij de tijd!

```

10 REM DIGKLOK
20 REM
30 REM MSX Computer Magazine
40 REM
50 '-----
60 '---DIGITALE KLOK---
70 '---VOOR MSX---
80 '---COMPUTERS---
90 '-----
100 '---S.HIJMANS---
110 '---UTRECHT---
120 '-----
130 CLS: COLOR 15,0,0: WIDTH 40: KEY
OFF: A=2
140 LOCATE A,9: IN$="INVOER VAN DE TI
JD"
150 LL$="-----"
160 PRINT TAB(20-LEN(IN$)/2);IN$
170 PRINT SPC(11);LL$
180 LOCATE A,12
190 LINE INPUT "Seconden: ";S$
200 S=VAL(S$)
210 IF S<0 OR S>59 THEN 180
220 LOCATE A,13
230 LINE INPUT "Minuten : ";M$
240 M=VAL(M$)
250 IF M<0 OR M>59 THEN 220
260 LOCATE A,14
270 LINE INPUT "Uren : ";H$
280 H=VAL(H$)
290 IF H<0 OR H>23 THEN 260
300 TIME=0: X=50: SCREEN 3
310 OPEN "grp:" FOR OUTPUT AS #1
320 IF X=6500 THEN X=50: TIME=0
330 IF TIME=>X THEN S=S+1: X=X+50
340 IF S=60 THEN S=0: M=M+1
350 IF M=60 THEN M=0: H=H+1
360 IF H=24 THEN H=0
370 IF MMOD60=0 THEN BEEP
380 IF OS<>S THEN LINE(196,56)-(255,8
0),1,BF
390 IF OM<>M THEN LINE(100,56)-(164,8
0),1,BF
400 IF OH<>H THEN LINE(4,56)-(68,80),
1,BF
410 PRESET (4,56): COLOR 15
420 PRINT #1, USING "###:###:###";H;M;S
430 OS=S: OM=M: OH=H
440 GOTO 320
    
```

# MCM2B

Ooit, in MSX Computer Magazine nummer 2, publiceerden we ons MCMbase programma. Een prima, geheugen-georiënteerde database, waar – gezien de vele brieven met complimentjes, die we nog steeds over dit programma krijgen – door vele mensen dankbaar gebruik van gemaakt wordt. Natuurlijk hebben we sindsdien niet stilgezeten. We hebben diverse malen andere database-programma's bekeken met het idee ze mogelijk in MCM op te nemen.

Geen van die programma's had echter die kwaliteit, die we de lezer nu eenmaal wensen te bieden. De reden daarvoor is eigenlijk heel simpel; een database is nu eenmaal of in machinetaal of in Basic geprogrammeerd. Een volledig in ML geschreven database is echter een heel groot project, hetgeen zal verklaren waarom we tot nog toe dergelijke programma's niet gezien hebben buiten het commerciële circuit.

## Basic

Een in Basic geschreven programma loopt echter meteen tegen de grootste beperking van de MSX-standaard op: te weinig geheugenruimte. Van de op zich riant 64K, waar bijna alle MSX computers mee uitgerust zijn, is onder Basic niet meer dan 32K bereikbaar. Van die 32K snoept het MSX-systeem ook nog het een en ander af – zoals diskgebruikers wel weten – zodat er uiteindelijk niet veel ruimte overblijft. In dat geheugen moeten dan en het programma, en de eigenlijke gegevens opgeslagen worden.

Kortom, een database in Basic, die de gegevens in RAM opslaat, zal wat snel aan bloedarmoede lijden. Te weinig opslag-kapaciteit.

Wat dergelijke programma's betreft is het aloude MCMbase nog steeds onovertroffen.

De andere mogelijkheid, een Basic-database die van de disk gebruik maakt om de gegevens in random-bestanden op te slaan, is blijkbaar ook wat ingewikkeld. We hebben een aantal van dergelijke programma's onder ogen gehad – nog afgezien van eigen experimenten in die richting – maar echt handig vonden we ze geen van allen. Om een voorbeeld te geven van de problemen die een dergelijke benadering met zich meebrengt, het *sorteren* wordt een moeizaam en vooral langdurig proces. Toch sluiten we zeker niet uit dat we in de toekomst nog eens met een dergelijk programma op de proppen komen.

## Memory-mapper

Met de komst van de MSX2 is er echter ook een andere geheugen-standaard beschikbaar gekomen, de memory-mapper. Zo'n memory-mapper is in feite een uitkomst om – ook vanuit Basic – grote hoeveelheden informatie in op te slaan. Hoe dat in zijn werk gaat, dat kunt u elders in dit artikel lezen. Bovendien hebben we

een tijdje terug – in MSX Computer Magazine nummer 12 – er al uitgebreid aandacht aan besteed. Maar het echte praktische werk met de mapper is tot nog toe wat ondergewaardeerd gebleven, naar onze mening. Terwijl er toch heel wat mee te presteren zou moeten zijn.

Al met al aanleiding genoeg voor MCM om eens te onderzoeken of er met gebruik van die memory-mapper geen goede database te ontwerpen was. Na de nodige weken stug doorwerken kwam daar MCM2B uit rollen. Hoe het technisch allemaal in zijn werk gaat, dat kunt u in een apart kadertje bij dit artikel terugvinden. Maar wees gerust, om MCM2B te gebruiken hoeft u niets van memory-mappers te begrijpen. Het is een heel gebruikersvriendelijk programma geworden.

## Geslaagd

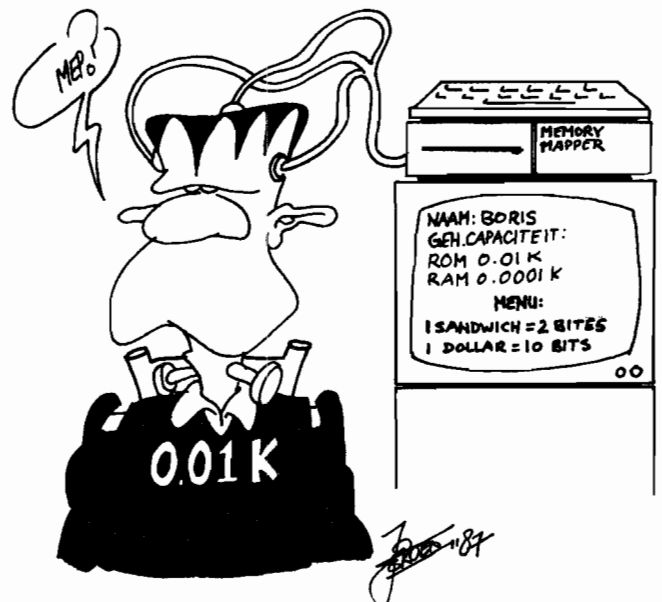
Misschien klinkt het wat overdreven, maar we zijn zelf best tevreden met het programma. Qua mogelijkheden is het – voor het moment – nog wat beperkt, maar door de modulaire structuur kunnen we er in de toekomst nog van alles aan vast knopen. Zo zal er op

korte termijn een print-mogelijkheid bijkomen.

MCM2B werkt alleen op een MSX2 met memory-mapper – van 64K of meer – en een diskdrive. Er is bewust afgezien van het inbouwen van cassette-mogelijkheden. Immers een bestand van tegen de 100K is op cassette eigenlijk niet meer hanteerbaar. Bovendien bestaat MCM2B uit een aantal afzonderlijke programma's, die ieder apart bepaalde functies vervullen. Er moet dus regelmatig een nieuw stuk programma geladen worden; hetgeen op een cassette-systeem de bruikbaarheid feitelijk tot nul reduceert.

## Eisen

Om MCM2B te kunnen gebruiken moet u dus de beschikking hebben over een MSX2 met diskdrive en een mapper van minimaal 64K RAM. Eventuele meerdere mappers worden niet ondersteund. Het programma zal de mapper in het meest lage slot gebruiken, wat echter wel de mogelijk geeft om bij – bijvoorbeeld – een Philips VG 8235 een losse 256 K mapper te gebruiken, als deze maar in een van de cartridge-slots geplaatst is. De



ingebouwde 128K mapper is dan echter buiten werking.

Plaats echter nooit 'gewone' RAM-cartridges in een lager genummerd slot dan dat waarin de mapper staat!

Gezien het feit dat MCM2B zelf in erg weinig geheugen moet draaien, is er voor gekozen om het programma slechts met een enkele drive te laten werken. U moet de tweede drive – of deze nu echte aanwezig is of alleen maar de 'logische' B-drive is – dan ook echt uitschakelen. Dat gebeurt vanzelf als u de computer opstart met de Control-toets ingedrukt.

### Kapaciteit

MCM2B gebruikt altijd een memory-mapper om zijn gegevens in op te slaan. Zo'n mapper kan bijvoorbeeld 128K zijn, maar ook 256K of zelfs 512K. Ongeacht de grootte van de mapper zal MCM2B altijd 16K zelf gebruiken, voor interne opslag. In een kleine mapper houdt u dan ook 96K ruimte over voor uw gegevens.

Bij het opzetten van een bestand dient u voor ieder veld een maximale veldlengte op te geven. Om tijd te winnen bij het opzoeken gebruikt MCM2B altijd de hele ruimte van zo'n veld, het is dus zaak deze veldlengte's niet te ruim te kiezen.

Stel nu dat u een bestand opzet, mat daarin boektitels en auteursnamen. Een goede lengte voor het veld titel zou dan 40 posities zijn, terwijl de meeste auteursnamen zich wel in 25 posities laten opslaan. Ongetwijfeld zullen deze veldlengtes in sommige gevallen te krap zijn, maar met een enkele afkorting past het allemaal best.

In dat geval zou een 128K mapper – waarvan dus 16K voor interne opslag gebruikt wordt terwijl het programma en de systeemvariabelen ook nog eens 16K gebruiken – plaats bieden aan 96K/65 re-

cords, oftewel 1512 kaarten. Mocht u echter besluiten om de velden zo lang te kiezen dat elke naam er in past, dat komt u al snel op de dubbele veldlengtes uit. En dus op de halve capaciteit!

Bij een recordlengte van 10 bytes bedraagt de theoretische capaciteit van een 96K mapper ongeveer 9000 records. Dit wordt in de praktijk echter tot 5250 gelimiteerd, doordat er anders teveel overhead door interne administratie zou ontstaan.

### Opbouw

MCM2B bestaat momenteel uit een drietal programma's; in de toekomst zullen daar nog meer modules bijkomen. Deze programma's zijn:

- MCM2BS, het startprogramma dat de computer voorbereidt op de eigenlijke database;
- MCM2BM, het centrale menu-programma, van waaruit de andere modules worden opgeroepen en
- MCM2B1, het eerste echte werkprogramma van de database, waarin men kan muteren, sorteren etcetera.

Om MCM2B te kunnen gebruiken moeten deze programma's alledrie op de werkdisk te staan, waarop men ook de gegevens zal opslaan. Hoewel zowel MCM2BM als MCM2B1 op zich losse Basic-programma's zijn, kunnen ze toch niet los gebruikt worden. Pas nadat het voorloop-programma, MCM2BS, zijn werk gedaan heeft kan via het menu, MCM2BM, de eerste echte toepassing, MCM2B1, gerund worden.

Wat al deze onderdelen doen zullen we nu eens wat gedetailleerder beschrijven. De voorloper geeft de naam van het programma-systeem aan, controleert of het wel op een MSX2 of hoger draait, een MM bezit, installeert de machinetaal routines en slaat in het video geheugen alvast het hoofdscherm op.

## Geheugenbeheer

Hoe MCM2B nu precies zijn geheugen beheerd is een aardig voorbeeld van wat een gehaaid programma kan met de mapper.

In een 16 Kb blok wordt een tabel van 5250 bij 3 bytes bijgehouden, waarin een tabel van pointers naar de records staat. Zo'n pointer bestaat uit 3 bytes.

Het eerste byte geeft het bloknummer aan waar het record te vinden is; de volgende twee zijn voor het adres. Voor elk aanwezig record wordt dus een 3-byte pointer aangemaakt en elke pointer krijgt een plaatsje in de tabel.

Het maximaal mogelijke aantal records is zo 'beperkt' tot 5250 stuks, zelfs als er een record van één veld van maar 10 bytes opgezet zou worden en een computer met een 4 Mb Memory Mapper gebruikt zou worden.

Dit is echter een extreem geval, meestal zal de 5250 records grens nooit bereikt worden. En wel om de eenvoudige reden dat een normale mapper 128 of 256K omvat. Van het totale geheugen – inclusief de mapper – zal 32K niet voor werkelijke opslag gebruikt kunnen worden. De pointer-tabel neemt ongeveer 15K in beslag, daarnaast wordt nog 1K voor het bewaren van veldlengtes, veldnamen, aantal records etcetera benodigd.

Dit is al een blok van 16 Kb. Verder moet het programma zelf ook nog wat ruimte hebben. Dit staat in pagina 3 en begint op C000h; het neemt nogmaals 16K van de RAM in de mapper in beslag.

Een consequentie daarvan is dat het vrije programmeerbare geheugen beperkt is tot zo'n 6 tot 7 Kbytes. Daar moest dus wel even het hele programma in 'gepropt' worden.

De rest van het werkgeheugen is dus vrij voor werkelijke gegevensopslag; 96K bij een 128K machine en 224K bij een 256K MSX2-computer.

Als laatste hoort het de Basic-start op naar C000h, laadt het echte menu-programma MCM2BM in en start deze op. Onder het na weer op het scherm teruggekeerde intro-blok verschijnen dan de volgende keuzes:

- Opzetten nieuw recordbestand;
- Laden reoord-bestand.

### Opzetten bestand

Bij het opzetten van een nieuw bestand moet natuurlijk eerst de veldindeling bepaald worden. Het aantal velden is beperkt tot 15, daar er niet meer op het beeldscherm passen. Daar MCM2B op een 80-kolomsscherm werkt en elk veld een hele regel beslaat,

blijft er na aftrek van de veldnaam – de veldnaamlengte mag maximaal 14 tekens lang zijn – en scheidingstekens, een ruimte van 60 tekens over voor het veld zelf.

De veldlengte is natuurlijk vrij instelbaar, zoals bij elke database, alleen kan de lengte vrij lang genomen worden, iets wat bij de meeste bekende MSX databases niet kan.

### Laden bestand

Als er binnen het menu-programma voor het laden van een bestand gekozen wordt, verschijnen de namen van alle databestanden op de huidige disk op het scherm. Is er geen diskette aanwezig in de drive, of als de diskette

cord gebruikt wordt. Doet men dit niet dan zou het geheugen 'vol gaan lopen'.

## Wegschrijven naar diskette

Het wegschrijven van een bestand naar een diskette gaat van start door het intoetsen van de F2. In het status-blok licht 'SAVEN' op. Eerst kijkt MCM2B of er voldoende vrije ruimte op de diskette is voor het bestand. Zo niet, dan verschijnt achter 'SAVEN' de tekst 'DISK TE VOL'. Met een druk op een toets keert men dan weer terug, zodat het mogelijk is om een andere diskette te gebruiken.

Als er genoeg ruimte is, dan vraagt MCMB2 om de naam van het te saven bestand. Wil men bij nader inzien toch niet wegschrijven, dan kan er een liggend streepje — het minteken — worden ingetoetsd en daarna RETURN, waarna de SAVE-routine wordt verlaten.

Blijkt bij het wegschrijven de diskette beveiligd te zijn, dan verschijnt de tekst 'DISK BEVEILIGD'. Na een toetsdruk zal de SAVE-routine verlaten worden, waarna men alsnog een andere diskette kan kiezen, dan wel de beveiliging verwijderen.

## Sorteren

Met behulp van de F3 kan men gaan sorteren. Dit sorteren kan op elk gewenst veld plaatsvinden. Door middel van de rode aanwijsbalk wordt gekozen welk veld de volgorde na sortering zal bepalen.

Sorteren kan op twee manieren, oplopend of aflopend. Men kan A-Z of Z-A sorteren. Dit geeft men aan door een 0 of een 1 achter de desbetreffende vraag te geven. Wil men bij nader inzien toch niet sorteren, dan zal iedere andere waarde dan 0 of 1 een ontsnapping mogelijk maken. Ook na het typen van een letter in plaats van een cijfer verlaat men het sorteren.

Tijdens het sorteren zullen de rode aanwijsblokken knippen. Dit knippen is bedoeld om aan te geven dat de machine bezig is, hoewel het toetsenbord 'dood' is.

Enkele opmerkingen:

- Het sorteren gaat geheel in machine-taal. In Basic zou dit anders letterlijk uren of zelfs dagen gaan duren, daar de opslag van de records niet door middel van variabelen gaat.
- Het sorteren gaat op ASCII-waarden. Een hoofdletter en kleine letter zijn dus verschillend.
- MCM2B sorteert twee keer sneller dan bijvoorbeeld MT-BASE. Het programma sorteert dan ook niet de records zelf, maar de pointertabel, wat natuurlijk een heleboel tijd scheelt. De routine is een gewone bubblesort-routine, hoewel deze wel speciaal geschreven is voor een memory-mapper en een pointertabel.

Voor de kenners: het knippen van de rode blokken op het scherm gaat niet ten koste van de sorteertijd, want dit knippen gebeurt op de interrupt van de VDP, niet die van de Z80.

## Zoeken

Een zeer eenvoudige zoekroutine zit onder de F8 funktietoets. Na op de F8 te hebben gedrukt kan men een zoektekst intikken.

In het veld wat door de rode

balk gemarkeerd wordt zal dan naar die string gezocht worden

De zoektekst kan zowel in het begin als ergens in het midden van het veld voorkomen. Wordt de string gevonden dan, komt het betreffende record in beeld. Na een toetsdruk verschijnt ook het nummer en kan men verder. Wordt de tekst echter niet gevonden, dan ziet men een korte mededeling op het scherm en kan men na een toetsdruk verder. Ook bij het zoeken maakt MCMB2 verschil tussen kleine en hoofdletters.

## F4 definiëren

Het komt vaak voor dat bij het intikken van records een bepaald veld — een plaatsnaam bijvoorbeeld — veel voorkomt. Speciaal voor dergelijke gevallen kan men in MCMB2 de funktietoets F4 zelf een inhoud geven, daarna gebruikt kan worden om velden in de database mee in te vullen.

Om zo'n vaste tekst aan de F4 toe te kennen moet men eerst die toets met behulp van de F9 definiëren.

## Tenslotte

Dat waren de mogelijkheden van MCMB2. Althans, de mogelijkheden zoals die tot nog toe zijn gerealiseerd. Want door de speciale opbouw van het programma — of beter: de programma's — kan MCMB2 nog een heel stuk verder uitgroeien.

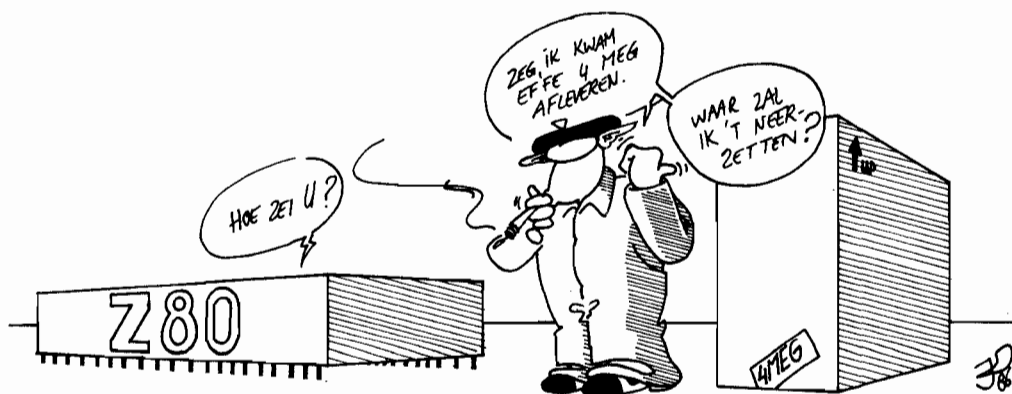
Zo kunt u heel snel een prin-

ter-programma vinden, en misschien een gespecialiseerde etiketten-generator. Met selectie-mogelijkheden, natuurlijk. Schrijf maar eens een briefje naar de redactie met een lijstje van wensen, dan weten we waar we het eerst aan moeten gaan werken.

Hopelijk zal MCMB2 u net zo bevallen als dat bij ons het geval geweest is. Dit programma is namelijk een ontwikkeling in fases geweest, nadat iemand op de redactie zich opeens realiseerde dat dergelijke toepassingen met de memory-mapper in principe mogelijk moesten zijn is onze geheugen-specialist nog een hele tijd aan het zwoegen geweest voordat de eerste versie draaide.

En aangezien die geheugenman slechts heel zelden op de redactie aanwezig is was die eerste versie ook voor ons een soort van donderslag bij heldere hemel. Dat zo iets fraais uit een losse gedachte zou komen rollen hadden we niet durven denken.

Bovendien is MCMB2 een schoolvoorbeeld van getruikt programmeren geworden. Allerlei nauwelijks bekende eigenschappen van de hardware zijn door de schrijver tot het uiterste uitgebuit. Zo kan een MSX officieel maar een kleur gebruiken op het schermtype 0, maar MCMB2 laat er toch echt meer zien. Wie echter wilt weten hoe dat zit, die moet de listings maar eens gaan uitpluizen.





```

10 REM MCM2BS      deel 1 - initialisatie en laden - van MCM'S MSX2BASE      0
20 REM                                                    0
30 REM MSX Computer Magazine                                                    0
40 REM                                                    0
50 WIDTH80:COLOR1,7,7:SCREEN0:KEYOFF:LOCATE0,0:PRINTSTRING$(80,223):LOCATE18,2:P
RINT"-- MSX COMPUTER MAGAZINE'S MSX2BASE --":LOCATE23,4:PRINT"by ARMAND SIMONIS
, MONSTER NL"
60 LOCATE1,6:PRINT"WARNING: THIS PROGRAM NEEDS MSX-BASIC VERSION 2.0 OR HIGHER" 129
70 LOCATE1,7:PRINT"          AND REQUIRES 64 KB MEMORY MAPPER AT LEAST.          " 11
80 IFPEEK(&HFB21)=1ANDPEEK(&HFB23)=0THEN90ELSEPRINT"SORRY, START YOUR COMPUTER W 51
ITH ONLY ONE DISK-DRIVE. <PRESS CONTROL BY START>":IFPEEK(&HFB23)<=0THENPRINT"AN
D REMOVE THE SECOND INTERFACE":END ELSE END
90 LOCATE1,8:PRINT"- CHECKING ..":IFPEEK(&H2D)=0THENLOCATE30,9:PRINT"SORRY, THIS 88
IS AN MSX1-COMPUTER!":END
100 IF INP(&HFD)=INP(&HFC)THENLOCATE30,9:PRINT"SORRY, NO MEMORY MAPPER IN THIS M 124
ACHINE !":END
110 LOCATE30,9:PRINT"..OK !..":LOCATE25,10:PRINT"--- DUTCH VERSION ---":LOCATE0, 170
12:PRINTSTRING$(80,223)
120 DEFINTA-Z:FORT=&H800TO&H90D:VPOKET,0:NEXT:VDP(14)=&HF0 67
130 GOSUB240:GOSUB180:CLEAR:DEFINTA-Z:VDP(13)=&H12 245
140 FORT=&H1800TO&H1830:VPOKET,0:NEXT 60
150 X=&HC000:POKEX,0:X=X+1 242
160 POKE &HF676,PEEK(VARPTR(X)):POKE &HF677,PEEK(VARPTR(X)+1) 76
170 RUN"MCM2BM" 255
180 FORT=2068TO2077:VPOKET,255:NEXTT:VDP(13)=&H17:COLOR7,7:C1$=CHR$(1)+CHR$(86) 1
190 LOCATE0,16:PRINTSTRING$(79,192):FORT=17TO22:LOCATE39,T:PRINTC1$:SPC(8):C1$:N 179
EXT:LOCATE41,17:PRINT"status:":LOCATE40,18:PRINT"OPBERGEN":LOCATE40,19:PRINT"EDI
T":LOCATE0,22:PRINT"F1:<=<= 10":LOCATE70,22:PRINT"F5:10 =>>"
200 LOCATE40,20:PRINT"ORTEREN":LOCATE40,21:PRINT"SAVE":LOCATE1,17:PRINTUSING"MA 255
X AANTAL RECORDS:#### ";RM;:PRINTC1$(1):LOCATE36,20:PRINT"F3:":LOCATE35,19:PRINT
"RET:":LOCATE36,21:PRINT"F2:":LOCATE26,18:PRINTSTRING$(13,195)
210 LOCATE26,17:PRINT"NAAM: ";NN$:LOCATE3,20:PRINT"RECORD NO 0":LOCATE5,18:PRINTU 53
SING"AANTAL RECORDS:#### ";AR;:PRINTC1$:LOCATE0,19:PRINTSTRING$(25,195):LOCATE36
,22:PRINT"F7:":LOCATE40,22:PRINT"WISSEN"
220 LOCATE52,18:PRINT"F9: ";C1$:"F4 DEFIN":C1$:LOCATE52,19:PRINT"F8: ";C1$:"ZOEKEN 48
";C1$:LOCATE49,17:PRINT"F4=":LOCATE21,20:PRINT"ESC: ";C1$:"MENU":LOCATE22,21:PR
INT"F6: ";C1$:"BEGIN":LOCATE21,22:PRINT"F10: ";C1$:"EINDE"
230 FORT=1280TO1920:VPOKE&H2000+T,VPEEK(T):VPOKET,32:NEXTT:COLOR1,7,7:FORT=2068T 7
02077:VPOKET,0:NEXTT:RETURN
240 FORT=&HDD00TO&HDDC7:READA$:POKET,VAL("&H"+A$):NEXTT 101
250 RESTORE400:FORT=&HDDCDO&HDCFF:READA$:POKET,VAL("&H"+A$):NEXTT 36
260 POKE&HFDA5,&HCD:POKE&HFDA6,&HDC:RETURN 191
270 : 197
280 DATA E1,E1,E5,CD,18,DD,23,CD,18,DD,23,CD,18,DD,3E,01,32,C3 210
290 DATA DD,00,C3,4D,DD,00,E5,DD,E1,7E,DD,4E,03,DD,77,03,71,C9 102
300 DATA 2A,C4,DD,ED,5B,C6,DD,06,00,1A,BE,13,23,28,04,30,06,18 98
310 DATA 0D,10,F4,18,12,3A,CA,DD,A7,20,0C,CD,00,DD,3A,CA,DD,A7 243
320 DATA 28,03,CD,00,DD,E1,C9,E5,DD,2A,C4,DD,CD,61,DD,DD,2A,C6 83
330 DATA DD,CD,61,DD,C3,24,DD,7E,23,5E,23,56,23,D3,FE,E5,62,6B 114
340 DATA ED,4B,C8,DD,09,01,00,00,DD,E5,D1,ED,B0,E1,3E,01,D3,FE 78
350 DATA C9,ED,4B,CB,DD,C5,AF,32,C3,DD,2A,CF,DD,C5,CD,4F,DD,23 76
360 DATA 23,23,C1,C5,E5,2A,D1,DD,37,3F,ED,42,44,4D,E1,7D,B9,20 66
370 DATA E9,7C,B8,20,E5,C1,3A,C3,DD,A7,28,12,C1,03,03,03,ED,5B 195
380 DATA CD,DD,7B,B9,20,C9,78,BA,20,C5,FB,C9,C1,FB,C9,00,D3,DD 26
390 DATA 10,DE 47
400 DATA E1,FE,0B,28,0A,FE,0C,28,06,E6,FE,EE,1E,20,03,F1,AF,F5 71
410 DATA E9,21,00,00,11,00,00,01,00,00,ED,B0,C9,F3,CD,7F,DD,FB 122
420 DATA C9,21,00,80,11,01,80,01,FF,3F,36,20,ED,B0,C9 60

```

```

10 REM MCM2BM      deel2 - Menu gedeelte - van MCM's MSX2BASE
20 REM
30 REM MSX Computer Magazine
40 REM
50 MAXFILES=1: CLEAR1250, &HCCC: POKE&HFDA4, &HC3: DEFINTA-Z: OUT&HFE, 1: FORT=1TO10: KE
YT, "": NEXT: T=0: COLOR1, 7, 7: ONERRORGOTO670: ONSTOPGOSUB660: STOPON: IFVPEEK(&H1800)TH
ENBF=1: GOTO100ELSEBF=0
60 FORT=1TO255: OUT&HFE, T: POKE&HBFFE, 0: NEXT: T=1
70 OUT&HFE, T: A=PEEK(&HBFFE): IF (A=T)ORA=0THENPOKE&HBFFE, TELSEGOTO90
80 T=T+1: GOTO70
90 AB=T-3: OUT&HFE, 1: DEFUSR1=&HDCF2: FORT=1TOAB+1: OUT&HFE, T: A2=USR1(0): NEXTT: T=0: O
UT&HFE, 1: POKE&H8000, 2: POKE&H8001, 0: POKE&H8002, &H80: VPOKE&H1800, AB: DEFUSR=&HDCE0:
DEFUSR1=&HDCEC: DEFUSR2=&H156
100 READA$(T), VW(T): IFA$(T)="*"THENT=T-1ELSE T=T+1: GOTO100
110 C1$=CHR$(1)+CHR$(86): C2$=CHR$(1)+CHR$(87): C3$=CHR$(1)+CHR$(88): C4$=CHR$(1)+C
HR$(89): C5$=CHR$(1)+CHR$(90): C6$=CHR$(1)+CHR$(91)
120 LOCATE21, 11: PRINTC3$: :FORP=1TO20: PRINTC2$: :NEXT: PRINTC4$: LOCATE21, 12: PRINTC
1$: " ***** MENU ***** "; C1$: LOCATE21, 13: PRINTC1$: SPC(20): C1$
130 A=14: FORP=0TOT: IFBF=VW(P) THEN LOCATE21, A: PRINTC1$: :PRINT USING " \
\ ": A$(P): :PRINTC1$: A=A+1 ELSE T=T-1
140 NEXT: FORY=0TO1: LOCATE21, A+Y: PRINTC1$: SPC(20): C1$: NEXTY: LOCATE21, A+Y: PRINTC5$
: :FORP=1TO20: PRINTC2$: :NEXT: PRINTC6$
150 IFSTICK(1)THEN150ELSEP=0: A2=USR2(0): GOTO190
160 D=STICK(0)ORSTICK(1): ONINSTR("A ", INKEY$)GOTO170, 210
170 IFSTRIG(1)THEN210
180 D=DAND(STICK(0)ORSTICK(1)): IFD=0THEN160ELSEA=P
190 P=P+(D=1)-(D=5): IFP<0THENP=0: GOTO160: ELSEIFP>TTHENP=T: GOTO160
200 VPOKE2191+A*10, 0: VPOKE2192+A*10, 0: VPOKE2191+P*10, 255: VPOKE2192+P*10, 255: GOTO
160
210 VPOKE2191+P*10, 0: VPOKE2192+P*10, 0: ONBF*2+P+1GOTO230, 240, 250, 270
220 GOTO210
230 VPOKE&H1801, 1: GOSUB430: FORT=&H800TO&H90E: VPOKET, 0: NEXT: RUN"MCM2B1"
240 VPOKE&H1801, 3: GOSUB290: RUN
250 VPOKE&H1801, 3: RUN"MCM2B1"
260 'VPOKE&H1801, 4: RUN"MCM2B2"
270 IFSTRIG(1)THEN270ELSELOCATE23, P+16: PRINT"ZEKER WETEN (J, N)": CHR$(255): A2=USR
2(0)
280 B$=INKEY$+CHR$(-STRIG(1)*74)+CHR$(-STRIG(3)*78): IFB$=CHR$(0)+CHR$(0)ORB$=" "T
HEN280ELSELOCATE40, P+16: PRINTB$: IFINSTR(CHR$(ASC(B$)AND&HDF), "J")THENRUN"MCM2BS"
ELSERUN
290 LOCATE0, 16: PRINTCHR$(27)"J": CHR$(1)+CHR$(88): :FORT=1TO77: PRINTC2$: :NEXT: PRIN
TC4$: POKE&HF3B0, 78: FILES".1": POKE&HF3B0, 80: P=PEEK(&HF3DC)
300 FORT=17TOP-1+(VPEEK(80*(P-2)+66)>32AND(VPEEK(80*(P-1)+2)=32)): LOCATE0, T: PRIN
TC1$: LOCATE78, T: PRINTC1$: NEXT: G=T-18: PRINTC5$: :FORT=1TO77: PRINTC2$: :NEXT: PRINTC6
$
310 DEFFNS(X, Y)=VAL("&H"+MID$("7FF80003FFC01FFE00FFF00007FF803FFC00", ((X-1)/13)*
6+1+Y*2, 2)): R=0: T=1: GOTO380
320 DY=STICK(0)ORSTICK(1)
330 A=T: T=T+(DY=7)*13-(DY=3)*13: IFT<1THENT=1: GOTO320ELSEIFT>66THENT=66: GOTO320
340 B=R: R=R+(DY=1)-(DY=5): IFR<0THENR=0: GOTO320ELSEIFR>GTHENR=G: GOTO320
350 IFVPEEK(T+(R+17)*80)=32THENT=A: R=B: GOTO320
360 IFSTRIG(0)ORSTRIG(1)THENNN$="": FORE=0TO7: NN$=NN$+CHR$(VPEEK(T+(R+17)*80+E)):
NEXTE: GOTO400ELSEIFDY=0THEN320
370 FORY=0TO2: VPOKE&H800+Y+(B+17)*10+A\8, 0: NEXTY
380 FORY=0TO2: VPOKE&H800+Y+(R+17)*10+T\8, FNS(T, Y): NEXTY
390 GOTO320
400 BLOADNN$+".1": GOSUB420: OB=PEEK(&HBFFC): VPOKE&H1820, OB: FORY=0TO5: VPOKE&H800+Y
+(R+17)*10+T\8, 0: NEXTY
410 D1=PEEK(&HBFFC): IFD1>AB+1THENERROR(60)ELSEFORD=2TOD1: OUT&HFE, D: BLOADNN$+". "+
RIGHT$(STR$(D), LEN(STR$(D))-1): NEXTD: OUT&HFE, 1: FORT=1TOLEN(NN$): VPOKE&H1805+T, AS
C(MID$(NN$, T, 1)): NEXTT: RETURN
420 FORT=1TOLEN(NN$): DU=ASC(MID$(NN$, T, 1)): IF(DU>64ANDDU<91)OR(DU>96ANDDU<123)TH
ENMID$(NN$, T, 1)=CHR$(ASC(MID$(NN$, T, 1)AND&HDF): NEXTT: RETURNELSENEXTT: RETURN
430 FORT=&H800TO&H90E: VPOKET, 0: NEXT: Y=0: AV=0: A=0: LOCATE0, 0: PRINTCHR$(27)"J": WIDT
H80: LOCATE23, 0: PRINT"*** OPZETTEN VAN HET RECORD ***": LOCATE14, 2: PRINT"VELDNAMEN

```

```

";SPC(31);"VELDLENGTE max=60"
440 PRINTSPC(11);C3$;:FORT=1TO16:PRINTC2$;:NEXTT:PRINTCHR$(1)+CHR$(82);:FORT=1TO
22:PRINTC2$;:NEXTT:PRINTCHR$(1)+CHR$(82);:FORT=1TO16:PRINTC2$;:NEXTT:PRINTC4$:FO
RT=1TO17:PRINTSPC(11);C1$;SPC(16);C1$;SPC(22);C1$;SPC(16);C1$:NEXT:PRINTSPC(11);
C5$;
450 FORT=1TO16:PRINTC2$;:NEXTT:PRINTCHR$(1)+CHR$(81);:FORT=1TO22:PRINTC2$;:NEXTT
:PRINTCHR$(1)+CHR$(81);:FORT=1TO16:PRINTC2$;:NEXTT:PRINTC6$:LOCATE12,22:PRINT"IN
S:ERTUSSEN";SPC(4);"DEL:ERUIT";SPC(7);"RET:EDIT";SPC(8);"ESC:KLAAR"
460 POKE&HF3B0,16:POKE&HF3B1,42:LOCATE0,10:PRINTCHR$(27)"J":LOCATE0,9:PRINT"TOT
AL:";DIMVN$(14),VL$(14):VPOKE&H82C+10,&H3F:VPOKE&H82D+10,&HFC:GOTO510
470 DY=STICK(0)ORSTICK(1):A=Y:Y=Y-(DY=5)+(DY=1):ONINSTR("a"+CHR$(13)+CHR$(&H7F)+
CHR$(&H12)+CHR$(27),INKEY$)GOTO480,510,540,570,500
480 IFDY=0THEN470ELSEIFY<0THENY=0:GOTO470ELSEIF(Y>AV)OR(Y>=14)THENY=AV-1
490 VPOKE&H82C+A*10+10,0:VPOKE&H82D+A*10+10,0:VPOKE&H82C+Y*10+10,&H3F:VPOKE&H82D
+Y*10+10,&HFC:GOTO470
500 VPOKE2276,255:GOSUB590:POKE&HF3B1,24:POKE&HF3B0,80:POKE&HBFFF,AV-1:FORT=0TOA
V-1:POKE&HBF01+T,VAL(VL$(T)):FORY=1TO14:POKE&HBDBF+T*14+Y,ASC(MID$(VN$(T),Y,1)):
NEXTY,T:LOCATE0,0:PRINTCHR$(27)"J":RETURN
510 VPOKE2274,255:AV=AV-(Y=AV):A2=USR2(0):POKE&HF3B0,80:LOCATE35,5+Y:PRINT"VELD"
;STR$(Y+1):POKE&HF3B0,16:LOCATE0,Y*2+10:LINEINPUTVN$(Y):HL=LEN(VN$(Y)):VN$(Y)=LE
FT$(VN$(Y),14):VN$(Y)=VN$(Y)+SPACE$(14-LEN(VN$(Y)))
520 RL=RL-VAL(VL$(Y)):IFHL>14THENLOCATE0,Y*2+10:PRINTVN$(Y);" ":IFHL>16THENLOCA
TE0,Y*2+11:PRINTSPC(15):LOCATE0,Y*2+11:PRINTVL$(Y)
530 A2=USR2(0):LOCATE0,Y*2+11:LINEINPUTVL$(Y):IFVAL(VL$(Y))<1ORVAL(VL$(Y))>60THE
N530ELSERL=RL+VAL(VL$(Y)):Y=Y+(Y<AV)*(Y<14):LOCATE7,9:PRINTRL;" ":VPOKE2274,0:G
OTO490
540 IFY>AV-1THEN470ELSEVPOKE2272,255:GOSUB590:RL=RL-VAL(VL$(Y)):POKE&HF3B0,16:LO
CATE7,9:PRINTRL;" ":IFY=AV-1ORY=14THENR=Y:GOTO560ELSEDE=Y:FORR=DETOAV-2:VN$(R)=
VN$(R+1):VL$(R)=VL$(R+1):POKE&HF3B0,80:LOCATE35,5+R:PRINT"VELD";STR$(R+1)
550 POKE&HF3B0,16:LOCATE0,R*2+10:PRINTSPC(15):PRINTSPC(15):LOCATE0,R*2+10:PRINTV
N$(R):PRINTVL$(R):NEXTR
560 VN$(R)="" :VL$(R)="" :LOCATE0,R*2+10:PRINTSPC(15):PRINTSPC(15):P
OKE&HF3B0,80:LOCATE35,R+5:PRINTSPC(7):POKE&HF3B0,16:AV=AV-1:GOTO470
570 IFAV>=15THEN470ELSEVPOKE2270,255:GOSUB590:DE=Y:FORR=AVTODE+1STEP-1:VN$(R)=VN
$(R-1):VL$(R)=VL$(R-1):POKE&HF3B0,80:LOCATE35,5+R:PRINT"VELD";STR$(R+1):POKE&HF3
B0,16:LOCATE0,R*2+10:PRINTSPC(15):PRINTSPC(15):LOCATE0,R*2+10:PRINTVN$(R):PRINTV
L$(R):NEXTR
580 VN$(R)="" :VL$(R)="" :LOCATE0,R*2+10:PRINTSPC(15):PRINTSPC(15):POKE&HF3B0,80:L
OCATE35,R+5:PRINTSPC(7):POKE&HF3B0,16:AV=AV+1:GOTO510
590 POKE&HF3B0,80:LOCATE32,20:PRINT"ZEKER WETEN (J,N)";:A2=USR2(0):A$=INPUT$(1):
LOCATE32,20:PRINTSPC(19):VPOKE2270,0:VPOKE2272,0:VPOKE2276,0:IFINSTR("Jj",A$)THE
NRETURNELSERETURN470
600 DATA "OPZETTEN BESTAND",0
610 DATA "LADEN BESTAND",0
620 DATA "NAAR BESTAND",1
630 DATA "PRINTEN BESTAND",1
640 DATA "NIEUW BESTAND",1
650 DATA "***",-1
660 DEFUSR=&H3E:A2=USR(0):POKE&HFDA4,&HC9:POKE&HF3B1,24:POKE&HF3B0,80:ONERRORGOT
00:END
670 IFERR=53ANDERL=400THENBEEP:RESUME400
680 IFERR=60THENLOCATE1,22:PRINT"TE WEINIG GEHEUGEN VOOR DIT BESTAND AANWEZIG !
<TOETS>";:A2=USR2(0):A$=INPUT$(1):CLEAR:ONERRORGOTO670:RUN"MCM2BS"
690 IFERR=53ANDERL=290THENBEEP:PRINT"GEEN DATA BESTANDEN AANWEZIG ! <TOETS>":POK
E&HF3B0,80:P=PEEK(&HF3DC)-1:NF=1:GOSUB300:LOCATE40,17:A$=INPUT$(1):VPOKE&H1800,0
:RUN
700 IFERR=56ORERR=69ORERR=70THENCOLOR15,4,4:BEEP:COLOR1,7,7:BEEP:RESUME
710 ERROR(ERR):PRINT"IN:";ERL

```

```

10 REM MCM2B1      deel 3 - muteren,sorteren etc. - van MCM'S MSX2BASE      0
20 REM                                                    0
30 REM MSX Computer Magazine      0
40 REM                                                    0
50 MAXFILES=0:DEFINT A-Z:AB=VPEEK(&H1800):OUT&HFE,1      64
60 ONKEYGOSUB500,350,600,,510,520,540,430,680,530:STOPON:ONSTOPGOSUB690:ONERRORG
OTO700      115
70 K=VPEEK(&H1801):ONKOSUB80,120,120:RUN"MCM2BM"      48
80 AV=PEEK(&HBFFF):FORT=&HBFF0TO&HBFFE:POKET,0:NEXT:NN$=STRING$(8,32):GOTO130      227
90 POKE&HBFFF,AV:POKE&HBFFC,D1:POKE&HBFFE,AR\256:POKE&HBFFD,ARMOD256:VPOKE&H1802
,TBMOD256:VPOKE&H1803,TB\256:VPOKE&H1804,Y:FORT=1TOLN(NN$):VPOKE&H1805+T,ASC(MI
D$(NN$,T,1)):NEXTT:POKE&HBFF2,PL:POKE&HBFF3,PH      248
100 POKE&HBFF0,ADMOD256:POKE&HBFF1,AD\256:RETURN      192
110 FORT=1TOLN(NN$):DU=ASC(MID$(NN$,T,1)):IF(DU>64ANDDU<91)OR(DU>96ANDDU<123)TH
ENMID$(NN$,T,1)=CHR$(ASC(MID$(NN$,T,1))AND&HDF):NEXTT:RETURNELSENEXTT:RETURN      146
120 AV=PEEK(&HBFFF):D1=PEEK(&HBFFC):AR=PEEK(&HBFFE)*256+PEEK(&HBFFD):AF=AR:PL=PE
EK(&HBFF2):PH=PEEK(&HBFF3):FORT=1TO8:NN$=NN$+CHR$(VPEEK(&H1805+T)):NEXTT      151
130 DIMVN$(AV),A$(AV):AD=PEEK(&HBFF0)+256*PEEK(&HBFF1):RL=0:FORT=0TOAV:VN$(T)="
:RL=RL+PEEK(&HBFF0+T):FORY=1TO14:VN$(T)=VN$(T)+CHR$(PEEK(&HBDBF+T*14+Y)):NEXTY,T
:GOSUB300:TB=VPEEK(&H1802)+256*VPEEK(&H1803):OB=VPEEK(&H1820):GOSUB390:GOSUB310      161
140 RB=16384\RL:BL=2+(AR\RB):IFAB*RB>=5250THENRM=5250ELSERM=AB*RB      220
150 PRINTCHR$(27)"H";:POKE&HF3B1,16:PRINTCHR$(27)"J":POKE&HF3B1,24      81
160 FORT=0TOAV:LOCATE0,T:PRINTUSING"\
\":VN$(T):PRINTA$(T):NEXT:FORT=1280TO1920:VPOKET,VPEEK(T+&H2000):NEXT:LOCATE20,17:PRINTUSING"####
":RM+1:LOC
ATE31,17:PRINTNN$:LOCATE13,20:PRINTTB/3+1:LOCATE20,18:PRINTUSING"####
":AR+1      14
170 GOSUB420:Y=VPEEK(&H1804):GOTO220      203
180 DY=STICK(0)ORSTICK(1):ONINSTR(" "+CHR$(13)+CHR$(27),INKEY$)GOTO190,250,260      195
190 DY=DYORSTICK(0)ORSTICK(1):ON(DY+1)\2+1GOTO180,200,320,200,320      13
200 BY=Y:Y=Y+(DY=1)-(DY=5):IFY>AVTHENY=AV:GOTO180ELSEIFY<0THENY=0:GOTO180      197
210 SWAPBY,Y:FL=0:GOSUB230:SWAPBY,Y      220
220 FL=&HFF:GOSUB230:GOTO180      220
230 VL=PEEK(&HBFF0+Y):LB=1:IFVL>=8THENFORLB=1TOVL\8:VPOKE2049+Y*10+LB,FL:NEXT      35
240 VPOKE2049+Y*10+LB,FLAND(256-2^(8-VLMOD8)):RETURN      206
250 VPOKE2243,255:LOCATE16,Y:MF=0:GOSUB400:LINEINPUTA$(Y):LOCATE16+PEEK(&HBFF0+Y
),Y:PRINTCHR$(27)"K":FL=0:GOSUB230:MF=1:Y=Y-(Y<AV):GOSUB420:VPOKE2243,0:IFTB/3=A
RTHENEX=0:GOTO200ELSE200      196
260 FL=0:GOSUB230:GOSUB270:GOSUB90:CLEAR:ONERRORGOTO700:RUN"MCM2BM"      123
270 Z=((TB/3)=AR)*(AR>AF):AF=AR*Z-AF*NOT(Z):VPOKE2233,255:GOSUB300:GOSUB390:OUT&
HFE,MB      227
280 FORQ=0TOAV:VL=LEN(A$(Q)):A1=VARPTR(A$(Q)):POKE&HDCE1,PEEK(A1+1):POKE&HDCE2,P
EEK(A1+2):POKE&HDCE4,GL:POKE&HDCE5,GH:POKE&HDCE7,VLMOD256:GH=GH-((GL+VL)MOD256<>
GL+VL)      232
290 GL=(GL+VL)MOD256:A2=USR(0):NEXTQ:GL=0:GH=0:OUT&HFE,1:VPOKE2233,0:RETURN      203
300 FORT=0TOAV:A$(T)=LEFT$(A$(T),PEEK(&HBFF0+T)):A$(T)=A$(T)+STRING$(PEEK(&HBFF01
+T)-LEN(A$(T)),32):NEXT:RETURN      58
310 OUT&HFE,MB:FORQ=0TOAV:VL=LEN(A$(Q)):A1=VARPTR(A$(Q)):POKE&HDCE4,PEEK(A1+1):P
OKE&HDCE5,PEEK(A1+2):POKE&HDCE1,GL:POKE&HDCE2,GH:POKE&HDCE7,VLMOD256:GH=GH-((GL+
VL)MOD256<>GL+VL):GL=(GL+VL)MOD256:A2=USR(0):LOCATE16,Q:PRINTA$(Q):NEXT:OUT&HFE,
1:RETURN      147
320 IFMF=1THENGOSUB270:MF=0      96
330 TB=TB+(DY=7)*3-(DY=3)*3:DY=0:IFTB<=0THENTB=0ELSEIFTB>3*ARTHENIFA$(0)<>STRING
$(LEN(A$(0)),32)ANDTF=0ANDAR<RMTHENGOSUB480:EX=1ELSETB=3*AR      88
340 TF=0:LOCATE13,20:PRINTTB/3+1,"":GOSUB390:GOSUB310:FORT=10TO1STEP-1:KEY(T)
ON:NEXTT:GOTO180      243
350 GOSUB400:VPOKE2263,255:LOCATE49,21:PRINT"NAAM:";:BU$=NN$:LINEINPUTNN$:NN$=LE
FT$(NN$,8):IFNN$="-"THENN$=BU$:GOTO380ELSEIFNN$=""THENN$=BU$:GOSUB110ELSEGOSUB
110      40
360 DS=DSKF(0):IF((DS<=2+(BL-OB)*16)AND(NN$=BU$))OR((NN$<>BU$)AND(DS<=2+(BL+1)*1
6))THENLOCATE49,21:PRINT"DISK TE VOL <TOETS>";:A2=USR(0):A$=INPUT$(1):NN$=BU$:GO
TO380      242
370 LOCATE31,17:PRINTSPC(8):LOCATE31,17:PRINTNN$:GOSUB270:OUT&HFE,1:D1=BL:GOSUB9
0:BSAVENN$+".1",&H8000,&HBFFF:FORD=2TOBL:OUT&HFE,D:BSAVENN$+". "+RIGHT$(STR$(D),L
EN(STR$(D))-1),&H8000,&HBFFF:NEXTD:OUT&HFE,1      88
380 BU$="":LOCATE49,21:PRINTSTRING$(25,32):VPOKE2263,0:GOSUB420:RETURN      209
390 MB=PEEK(&H8000+TB):GL=PEEK(&H8001+TB):GH=PEEK(&H8002+TB):RETURN      118

```

```

400 FORT=1TO10:KEY(T)OFF:NEXTT:IFMF=1THENGOSUB270:MF=0 42
410 RETURN 192
420 FORT=10TO1STEP-1:KEY(T)ON:NEXTT:RETURN 19
430 GOSUB400:VPOKE2245,255:LOCATE49,20:LINEINPUTS$:U$=STRING$(LEN(A$(Y)),32):VL=
LEN(U$):DP=0:FORT=0TOY:DP=DP+PEEK(&HBF01+T):NEXTT:DP=DP-PEEK(&HBF00+T):FORT=TB/3
TOAR:OUT&HFE,1:MB=PEEK(&H8000+T*3):GL=PEEK(&H8001+T*3):GH=PEEK(&H8002+T*3):OUT&H
FE,MB 57
440 A1=VARPTR(U$):POKE&HDCE4,PEEK(A1+1):POKE&HDCE5,PEEK(A1+2):POKE&HDCE7,VL:GH=G
H-((GL+DPMOD256)MOD256<->GL+DPMOD256)+(DP\256):GL=(GL+DP)MOD256:POKE&HDCE1,GL:POK
E&HDCE2,GH:A2=USR(0):IFINSTR(U$,S$)THEN460ELSENEXTT 66
450 OUT&HFE,1:LOCATE49,20:PRINT"NIET GEVONDEN";:GOTO470 40
460 OUT&HFE,1:TB=T*3:GOSUB390:GOSUB310 56
470 DP=0:T=0:S$="":U$="":A2=USR2(0):LOCATE78,20:A$=INPUT$(1):LOCATE49,20:PRINTST
RING$(30,32):VPOKE2245,0:GOSUB420:RETURN320 92
480 AR=AR+1:IFAD<>0THENAD=AD-1ELSEBO=BL:BL=BL-(AR>=RB*(BL-1)):PH=(PH-((PL+RLMOD2
56)MOD256<->PL+RLMOD256)+(RL\256))*(-(BO=BL)):PL=((PL+RL)MOD256)*(-(BO=BL)):POKE&
H8000+TB,BL:POKE&H8001+TB,PL:POKE&H8002+TB,PH+&H80 40
490 FORT=0TOAV:LOCATE16,T:PRINTSTRING$(LEN(A$(T)),32):NEXTT:LOCATE20,18:PRINTUSI
NG"#### ";AR+1:LOCATE13,20:PRINTTB/3:FL=0:GOSUB230:FL=&HFF:Y=0:VL=PEEK(&HBF01):G
OSUB230:RETURN 10
500 GOSUB400:TB=TB-30:RETURN320 117
510 GOSUB400:TB=TB+30:TF=1:RETURN320 195
520 GOSUB400:TB=0:RETURN320 114
530 GOSUB400:TB=AR*3:TF=1:RETURN320 176
540 GOSUB400:IFMF=1THENGOSUB270:MF=0 206
550 IFAR=<0ANDA$(0)=STRING$(LEN(A$(0)),32)THEN590ELSEVPOKE2273,255:LOCATE49,22:P
RINT"ZEKER WETEN (J,N)";:A2=USR2(0):A$=INPUT$(1):IFINSTR("Jj",A$)THEN560ELSEVPOK
E2273,0:LOCATE49,22:PRINTSTRING$(18,32):GOTO590 164
560 LOCATE49,22:PRINTSTRING$(18,32):PP=TB+&H8000:M1=PEEK(PP):M2=PEEK(PP+1):M3=PE
EK(PP+2):IFTB=AR*3THEN580 97
570 POKE&HDCE1,(TB+3)MOD256:POKE&HDCE2,(TB+3)\256+&H80:POKE&HDCE4,TBMOD256:POKE&
HDCE5,TB\256+&H80:POKE&HDCE7,(AR*3-(TB))MOD256:POKE&HDCE8,(AR*3-(TB))\256:A2=USR
(0):POKE&HDCE8,0 249
580 PP=AR*3+&H8000:POKEPP,M1:POKEPP+1,M2:POKEPP+2,M3:OUT&HFE,M1:FORT=VAL("&H"+HE
X$(M2+M3*256))TOVAL("&h"+HEX$(M2+M3*256+RL-1)):POKET,32:NEXTT:OUT&HFE,1:GOSUB390
:GOSUB310:AD=AD-(AR>0):AR=AR+(AR>0):LOCATE20,18:PRINTUSING"#### ";AR+1 251
590 VPOKE2273,0:RETURN320 223
600 GOSUB400:IFMF=1THENGOSUB270:MF=0:GOTO620 5
610 AF=AR-EX 255
620 IFAF<1THEN670 67
630 LOCATE49,20:INPUT"A->Z=2,Z->A=1 NIET=0":S$:IFVAL(S$)>2ORVAL(S$)<1THEN670ELSE
POKE&HDDCA,VAL(S$)-1:POKE&HDDCB,0:POKE&HDDCC,0:POKE&HDDCD,(AF*3)MOD256:POKE&HDDC
E,(AF*3)\256:POKE&HDDCF,0:POKE&HDDDD0,&H80:POKE&HDDDD1,(AF*3)MOD256 41
640 POKE&HDDDD2,(AF*3)\256+&H80:POKE&HDD72,PEEK(&HBF01+Y):POKE&HDD73,0:POKE&HDD2C
,PEEK(&HBF01+Y) 156
650 L=0:IFY=0THEN660ELSEFORE=0TOY-1:L=L+PEEK(&HBF01+E):NEXTE 7
660 POKE&HDDC8,LMOD256:POKE&HDDC9,L\256:VDP(14)=&H42:VPOKE2253,255:A2=USR1(0) 24
670 VDP(14)=&HFF:VPOKE2253,0:LOCATE49,20:PRINTSTRING$(30,32):RETURN320 20
680 GOSUB400:VPOKE2235,255:LOCATE52,17:PRINTSTRING$(27,32):LOCATE52,17:LINEINPUT
F4$:KEY4,F4$:F4$="":GOSUB420:VPOKE2235,0:RETURN 22
690 DEFUSR=&H3E:A2=USR(0):POKE&HFDA4,&HC9:GOSUB90:ONERRORGOTO0:END 156
700 IFERR=68ANDERL=370THENLOCATE49,21:PRINT"DISK BESCHERMD <TOETS>";:A$=INPUT$(1
):NN$=BU$:LOCATE31,17:PRINTSPC(8):LOCATE31,17:PRINTNN$:RESUME380 219
710 IFERR=56ORERR=69ORERR=70THENCOLOR15,4,4:BEEP:COLOR1,7,7:BEEP:RESUME 154
720 A=ERL:ERROR(ERR):PRINT"IN:";A 14

```

## CATEGORIE: UTILITIES

# Disass

Het ene programma is het andere niet, om maar eens een vreselijke dooddoener te gebruiken. Maar waarom twee programma's — die in principe precies hetzelfde doen — zo verschillend kunnen zijn wat gebruiksgemak betreft, dat is het geheim van de smid. Of, in dit geval, van de goede programmeur.

En die goede programmeur is in dit geval Steven van Loef. Zo langzaam maar zeker beginnen we op hem te rekenen, als het om echt goede programma's gaat. Zijn spellen zijn prima, maar zijn utility's zijn nog beter. Vooral het gemak, waarmee die programma's zich laten gebruiken, is formidabel!

### MSXbug

Onlangs, in MSX Computer Magazine nummer 16, publiceerden we MSXbug, een prachtige monitor voor zo-

scherm-pagina georiënteerd, met andere woorden, er wordt steeds een heel scherm tegelijkertijd gebruikt om de disassembly te tonen. Dat zijn overigens 40-koloms schermen, zodat Disass zowel op MSX1 als op MSX2 kan werken. Een gebruiksaanwijzing is bij Disass eigenlijk overbodig. Het programma wijst zichzelf; alle kommando's zijn voortdurend in beeld.

Die kommando's luiden:

– A, voor Adres, waarmee men de disassembler

280 IF CS<>275407! THEN RG=760:GOTO 7	
00	212
560 DEFUSR9=&HBF00	79
570 A=USR9(0):SL=PEEK(&HBF9D)	218
910 DATA 5D,43,4D,67,44,52,CC,43,53,0	
D,44,44,00,50,00,00,00,00,00,00,00,00,00,00	
,00,FF,2A,06,48,CD,2F,47,4F,23	17

### MSXbug aanpassingen voor Disass

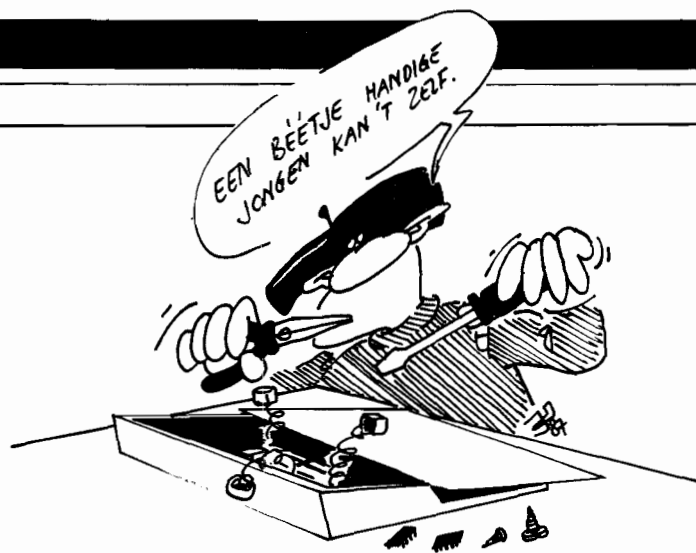
wel MSX1 als MSX2. Een van de eerste monitoren die we ooit gezien hebben waarbij men alle vrijheid heeft, wat betreft slot- en pagina-selectie. Disass, het nieuwste programma van Steven, sluit daar werkelijk naadloos op aan. Een disassembler met (bijna) alles wat de serieuze MSX programmeur zich maar kan wensen. Razendsnel — want het is geheel in machinetaal geschreven — en alweer geheel aangepast aan de MSX2-mogelijkheden. Het programma is beeld-

rechtstreeks vanaf een bepaald adres kan laten disassembleren;

– B, wat voor Blad staat. Met de B kan men steeds de volgende pagina van de disassembly-list op het scherm roepen;

– M, waarmee men terugkeert naar Basic, of eventueel naar MSXbug;

– P, voor Print. Na het P kommando zal Disass vragen om een begin- en eind-adres van het te disassembleren geheugendeel, waarna dat gedeelte op de printer verschijnt.



De Escape-toets biedt de mogelijkheid om weer te 'ontsnappen', als men hier per ongeluk verzeild is, maar ook als er geen printer is aangesloten loopt het programma niet vast; – S, hetgeen de afkorting is voor Slots. Met de S brengt men de cursor naar de rechterbovenkant van het beeld, waar daarna de vier pagina's van het MSX-geheugen naar believen over de slots verdeeld kunnen worden.

De 'cursor-omhoog' toets, tenslotte, laat de disassembly per regel verlopen. Oftewel, alle regels scrollen er één omhoog, waarbij een nieuwe instructie verschijnt op de onderste regel.

U ziet, Disass kent niet zo vreselijk veel kommando's. Terecht, want een disassembler heeft er ook niet zoveel nodig. Het programma is echter méér dan geschikt voor zijn doel, want het is vreselijk snel. Wachten is er echt niet bij.

### Koppelen

Omdat Steven MSXbug al geschreven had met een eventuele uitbreiding in het achterhoofd, is Disass eenvoudig te koppelen aan MSXbug. Daarvoor moet MSXbug echter wel een beetje veranderd worden. De eerste verandering is in regel 280: het getal 275259 moet daar vervangen worden door 275407. Verder moeten de eerste drie 00-bytes in regel 910

vervangen worden door het getal 44,00,50.

Tenslotte moet in de regels 560 en 570 twee maal USR1 vervangen worden door USR9.

Deze vier veranderde regels staan bij dit artikel nog eens in hun geheel afgedrukt.

Vanuit aldus gewijzigde versie van MSXbug kan de disassembler nu simpelweg worden aangeroepen door het kommando D te geven.

Het Disass-scherm verschijnt en de disassembler is actief. Als vanuit Disass dan het kommando M gegeven wordt, om terug te keren naar Basic, dan stopt Disass en neemt MSXbug de draad weer op. Netjes, nietwaar?

Alleen kan MSXbug niet 'zien' of Disass wel aanwezig is. Als dat niet zo is, zal de computer na het D-kommando onherroepelijk vastlopen.

Disass is iets slimmer: na aanroepen vanuit Basic — direkt met de funktietoets dus — keert hij terug naar Basic; maar als Disass vanuit MSXbug gestart is, dan zal het programma na beëindigen ook naar MSXbug terugkeren.

### IJzersterke combinatie

Steven heeft in beide programma's de mogelijkheid ingebouwd om de schermkleuren te kiezen. Alletwee herstellen de oude schermkleuren ook weer, als het programma verlaten wordt.

```
DISASSEMBLER v 1.0 Copyright 1987 by MCH
0000 F3          DI          #0416          PAG SLT
0001 C331684    JP          #0416          0000-0000
0004 BF          CP          #0416          0000-0000
0005 1B          DEC         DE          0000-0000
0006 36          SBC         D, B        0000-0000
0007 98          SBC         D, B        0000-0000
0008 C38326     JP          #2683          0000-0000
000B 00          NOP          #2683          0000-0000
000C C3F501     JP          #01F5          0000-0000
000F 00          NOP          #01F5          0000-0000
0010 C38626     JP          #2686          0000-0000
0013 00          NOP          #2686          0000-0000
0014 C32502     JP          #0225          0000-0000
0017 00          NOP          #0225          0000-0000
0018 C3451B     JP          #1B45          0000-0000
0019 00          NOP          #1B45          0000-0000
001C C31702     JP          #0217          0000-0000
001F 00          NOP          #0217          0000-0000
0020 C3040C     JP          #0C04          0000-0000
0023 00          NOP          #0C04          0000-0000

A=Adres
B=Blad
M=Basic
P=Print
S=Slots
↑=Regel
```

Het kan dus handig zijn om de schermkleuren van MSXbug en Disass verschil- lende in te stellen, zodat altijd duidelijk is welke van de twee er actief is.

Bovendien wordt Disass aangeroepen met funktie- toets 2 – en MSXbug met funktietoets 1. Dus als de beide programma's na el- kaar worden geïnstalleerd zitten ze elkaar daar ook niet in de weg.

De MSXbug-Disass kombi- natie is een sterk staaltje van goed programmeren!

**Bezoek**

Dit keer kwam Steven zijn nieuwste produkt zelf langs brengen op de redactie.

Dat is een nog heel gezellige middag geworden, nadat we hem hadden laten beloven niet uit de school te klappen over de zwijnenstal, die het soms bij ons kan zijn.

De machinetaal-redakteur stelde onder het experimen- teren met Disass min of meer

verbaasd vast, dat de kombi- natie van MSXbug en Disass zeker even goed was als de professionele programma- tuur die daarvoor op de markt is.

Gelukkig sprak hij, zoals bij- na altijd, in hexadecimale co- des, zodat onze adverteer- ders die dergelijke produk- ten aanbieden zich niet op de tenen getrapt hoeven te voe- len.

Eigenlijk hadden we maar een kleine aanmerking op Disass, namelijk dat het niet naar een bestand kon disas- sembleren.

Dat zou een aantal aardige extra's mogelijk maken, zo- als disassembleren, wijzigen en weer opnieuw assemble- ren.

Maar wat niet is kan nog kom- men; Steven zou eens kijken of die mogelijkheid mis- schien nog te realiseren valt. In ieder geval is Disass, zoals bij dit verhaal staat afge- drukt, een dijk van een disas- sembler.

Veel plezier ermee!

```
10 REM DISASS 0
20 REM version 1.0 0
30 REM 0
40 REM MSX Computer Magazine 0
50 REM 0
60 REM S.V.L. Software Vleuten 0
70 REM 0
80 '***** 0
90 ' PAS TOP OF BASIC MEMORY AAN **** 0
100 CLEAR 200, (PEEK(&HFC4A)+256*PEEK( 238
&HFC4B))-5
110 B=PEEK(&HFC4A)+256*PEEK(&HFC4B) 121
120 DEFUSR1=B:KEY 2,"A=USR1(0)"+CHR$(
```

```
13) 6
130 ' PRINT INTRODUCTIESCHERM ***** 0
140 W=PEEK(&HF3AE) 253
150 SCREEN 0:WIDTH 37:CLS 85
160 LOCATE 9,8:PRINT "DISAS version 1 100
.0"
170 LOCATE 7,10:PRINT "S.V.L. Softwar 6
e 1987" 199
180 LOCATE 15,12:PRINT "voor:" 71
190 LOCATE 7,14:PRINT "MSX Computer M 80
agazine"
200 LOCATE 6,16:PRINT "Machinecode in 19
stalleren"
210 LOCATE 8,18:PRINT "Even geduld a 0
.u.b."
220 ' INSTALLEER MACHINECODE ***** 127
230 RESTORE 800 176
240 FOR I=&HC000 TO &HCE8F 163
250 READ A$ 249
260 D=VAL("&H"+A$) 73
270 CS=CS+D 207
280 POKE I,D 218
290 NEXT I 56
300 IF CS<>366958! THEN RG=800:GOTO 7 170
20 186
310 RESTORE 2010:CS=0 160
320 FOR I=&HBF00 TO &HBF9D 246
330 READ A$ 70
340 D=VAL("&H"+A$) 204
350 CS=CS+D 215
360 POKE I,D 137
370 NEXT I 0
380 IF CS<>19033 THEN RG=2010:GOTO 72 0
0 170
390 ' KLEUR VERANDEREN ***** 0
400 CLS 0
410 PRINT "Voorgroendkleur in DISAS is 170
nu 1"
420 PRINT "Achtergroendkleur in DISAS 126
is nu 3"
430 PRINT:PRINT "Wilt u dit wijzigen 21
(j/n)?"
440 A$=INPUT$(1) 164
450 IF A$="N" OR A$="n" THEN 580 5
460 IF A$<>"J" AND A$<>"j" THEN BEEP: 222
GOTO 440
470 PRINT A$ 76
480 LOCATE 0,4:PRINT CHR$(27);"J"; 34
490 LINE INPUT "Nieuwe voorgroendkleur 234
:";VK$
500 VK=VAL(VK$):IF VK<1 OR VK>15 THEN 182
BEEP:GOTO 480
510 LOCATE 0,5:PRINT CHR$(27);"J"; 35
520 LINE INPUT "Nieuwe achtergroendkle 218
ur:";AK$
530 AK=VAL(AK$):IF AK<1 OR AK>15 OR A 39
K=VK THEN BEEP:GOTO 510
540 POKE &HC086,VK 138
550 POKE &HC08C,AK 234
560 ' ZOEK RAMSLOT IN PAGE 1 EN 0
570 ' VERPLAATS DISASS NAAR &H5000 0
580 DEFUSR9=&HBF00 83
590 A=USR9(0):SL=PEEK(&HBF9D) 222
600 IF SL=&HFF THEN STOP 77
610 POKE B+0,&HF7:' RST #30 45
```

```

620 POKE B+1,SL: ' RAMSLOT 18
630 POKE B+2,&H0: ' ADRES WAAR 215
640 POKE B+3,&H50: ' DISAS BEGINT 142
650 POKE B+4,&HC9: ' RET 122
660 WIDTH W:CLS 64
670 PRINT "Machinecode is ingeladen." 240
680 PRINT 149
690 PRINT "U kunt DISAS starten met [
F2] of met" 133
700 PRINT "DEFUSR1=";HEX$(B);":A=USR1
(0)" 135
710 NEW 46
720 WIDTH W:CLS 57
730 PRINT USING "U heeft een fout gem
aaktin de data vanaf regel:####";RG 166
740 PRINT 142
750 PRINT "Eerst verbeteren!" 191
760 STOP 243
770 REM 0
780 REM DISASS v 1.0 0
790 REM 0
800 DATA ED,73,5A,50,2A,5A,50,11,00,C
0,E7,38,03,31,FF,BF,CD,5E,50,CD,EC,50
,CD,46,56,DD,CB,00,6E,28,F7,2A 241
810 DATA 3D,5A,22,3B,5A,11,4F,5A,1A,3
2,AE,F3,13,1A,D5,CD,5F,00,D1,13,1A,FE
,FF,D5,CC,CF,00,D1,13,1A,21,E9 139
820 DATA F3,77,13,23,1A,77,D5,CD,62,0
0,D1,13,1A,32,AB,FC,EE,FF,CD,32,01,DD
,36,00,00,31,00,00,FB,C9,11,4F 209
830 DATA 5A,3A,AE,F3,12,13,3E,28,32,A
E,F3,3A,AF,FC,12,13,D5,CD,6C,00,D1,3A
,DE,F3,12,13,D5,CD,CC,00,D1,21 234
840 DATA E9,F3,7E,12,13,36,01,23,7E,1
2,13,36,03,D5,CD,62,00,D1,AF,CD,32,01
,3A,AB,FC,12,3E,FF,32,AB,FC,01 87
850 DATA 08,00,11,F0,0A,21,09,5B,CD,5
C,00,CD,38,01,E6,C0,07,07,32,55,5A,4F
,06,00,21,C1,FC,09,7E,FE,00,28 32
860 DATA 15,21,C5,FC,09,7E,E6,C0,CB,3
F,CB,3F,CB,3F,CB,3F,C6,80,81,32,55,5A
,3A,55,5A,32,58,5A,32,59,5A,DD 170
870 DATA 21,4E,5A,DD,36,00,00,FD,21,3
F,5A,C9,3E,0C,DF,2A,DC,F3,E5,CD,48,51
,21,01,01,CD,C6,00,11,5C,5A,CD 20
880 DATA 6C,57,CD,C3,51,CD,01,52,21,0
3,01,CD,C6,00,2A,3B,5A,22,3D,5A,06,14
,C5,DD,36,00,00,2A,3B,5A,22,39 74
890 DATA 5A,E5,CD,EF,57,E1,CD,5E,57,C
D,13,52,C5,3E,0F,32,DD,F3,CD,B4,52,C1
,2A,3B,5A,09,22,3B,5A,CD,1C,58 46
900 DATA C1,10,D3,E1,22,DC,F3,C9,21,2
8,00,CD,B4,51,21,48,00,3E,12,CD,4D,00
,21,4C,00,3E,12,CD,4D,00,21,70 80
910 DATA 00,06,14,11,28,00,3E,16,CD,4
D,00,19,10,F8,21,74,00,06,06,3E,16,CD
,4D,00,19,10,F8,21,99,00,CD,BF 56
920 DATA 51,21,9C,00,3E,15,CD,4D,00,2
1,60,01,3E,14,CD,4D,00,23,CD,BF,51,21
,98,00,3E,14,CD,4D,00,21,64,01 78
930 DATA 3E,11,CD,4D,00,21,70,03,CD,B
4,51,21,90,03,3E,11,CD,4D,00,C9,06,28
,3E,17,CD,4D,00,23,10,F8,C9,06 8
940 DATA 07,18,F3,21,03,22,CD,C6,00,1
1,CE,5A,CD,6C,57,21,05,23,06,04,CD,C6

```

```

,00,3E,34,90,DF,2C,10,F6,21,05 163
950 DATA 26,06,04,11,56,5A,CD,C6,00,1
A,E6,03,C6,30,DF,3E,2D,DF,1A,CB,3F,CB
,3F,E6,03,C6,30,DF,13,2C,10,E6 193
960 DATA C9,06,06,11,D6,5A,21,0A,22,C
D,C6,00,CD,6C,57,2C,10,F7,C9,FD,21,3F
,5A,FD,7E,00,21,11,5B,0E,18,ED 65
970 DATA B1,28,1A,0E,1A,ED,B1,28,18,F
E,DD,28,1C,FE,FD,28,18,FE,CB,28,43,FE
,ED,28,4C,06,01,18,64,06,02,18 168
980 DATA 60,06,03,18,5C,06,04,18,58,D
D,CB,00,DE,FD,7E,01,21,43,5B,0E,0B,ED
,B1,28,E4,0E,18,ED,B1,28,E2,0E 0
990 DATA 04,ED,B1,28,E0,FE,CB,20,2E,F
D,7E,03,FE,36,28,27,E6,06,FE,06,20,21
,18,CD,FD,7E,01,FE,30,38,BE,FE 47
1000 DATA 38,30,BA,18,12,FD,7E,01,21,
6A,5B,0E,32,ED,B1,28,AC,0E,06,ED,B1,2
8,AE,DD,CB,00,9E,DD,CB,00,D6,06 73
1010 DATA 01,48,78,32,47,5A,3E,20,DF,
21,3F,5A,7E,CD,F4,57,23,10,F9,C9,DD,C
B,00,56,20,53,DD,CB,00,5E,28,02 129
1020 DATA FD,23,FD,7E,00,FE,CB,28,55,
FE,ED,CA,56,53,21,A2,5B,01,17,00,ED,B
1,CA,0D,54,FE,40,DA,19,54,FE,80 27
1030 DATA DA,95,54,FE,C0,DA,B1,54,E6,
07,FE,00,CA,30,55,FE,01,CA,4F,55,FE,0
2,CA,45,55,FE,04,CA,4A,55,FE,05 185
1040 DATA CA,58,55,FE,06,CA,5D,55,FE,
07,CA,6E,55,11,CD,5B,CD,6C,57,3A,3F,5
A,CD,F4,57,DD,CB,00,96,C9,FD,7E 1
1050 DATA 01,DD,CB,00,5E,28,03,FD,7E,
02,F5,0F,0F,0F,E6,07,4F,F1,F5,FE,40,3
8,07,07,07,E6,03,C6,07,4F,79,21 154
1060 DATA 1E,5D,CD,15,56,CD,6C,57,F1,
F5,FE,40,D4,3A,56,F1,E6,07,CD,BE,55,C
9,FD,7E,01,F5,FE,A0,30,6B,21,BD 77
1070 DATA 5B,01,0C,00,ED,B1,20,0E,3E,
0B,91,21,A3,5C,CD,15,56,CD,6C,57,F1,C
9,E6,07,FE,00,28,0E,FE,01,28,1A 236
1080 DATA FE,02,28,1B,FE,03,28,2C,F1,
C9,11,0B,5E,CD,6C,57,F1,0F,0F,0F,E6,0
7,CD,BE,55,C9,11,1B,5E,18,EE,F1 85
1090 DATA F5,0F,0F,0F,E6,01,21,25,5E,
CD,15,56,CD,6C,57,F1,CD,D9,55,C9,F1,F
5,0F,0F,0F,E6,01,21,81,5D,CD,15 157
1100 DATA 56,CD,6C,57,F1,CD,D9,55,C9,
F1,F5,E6,03,FE,03,28,26,F1,F5,E6,08,F
E,08,28,15,F1,F5,E6,03,21,F2,5C 163
1110 DATA CD,15,56,CD,6C,57,F1,CB,67,
C8,3E,52,DF,C9,F1,F5,E6,03,21,FE,5C,1
8,E9,F1,21,C9,5B,01,04,00,ED,B1 26
1120 DATA 3E,03,91,21,0A,5D,CD,15,56,
CD,6C,57,C9,21,D4,5B,3E,16,91,CD,15,5
6,C3,6C,57,FE,10,CA,7C,55,FE,18 0
1130 DATA CA,9D,55,21,B9,5B,01,04,00,
ED,B1,CA,A8,55,FE,32,CA,C9,54,FE,3A,C
A,CF,54,FE,22,CA,D4,54,FE,2A,CA 5
1140 DATA D9,54,E6,0F,FE,01,CA,DE,54,
FE,02,CA,EC,54,FE,03,CA,19,55,FE,09,C
A,0A,55,FE,0A,CA,FD,54,FE,0B,CA 153
1150 DATA 19,55,E6,07,D6,04,21,66,5D,
CD,15,56,CD,6C,57,FD,7E,00,E6,38,CB,3
F,CB,3F,CB,3F,CD,BE,55,3A,47,5A 167
1160 DATA FE,01,C8,DD,CB,00,5E,28,03,

```



FE, 04, C0, 3E, 2C, DF, 11, 63, 5D, C3, 6C, 57, 1  
1, 72, 5D, CD, 6C, 57, FD, 7E, 00, F5, 0F 94

1170 DATA 0F, 0F, E6, 07, CD, BE, 55, 3E, 2C,  
DF, F1, E6, 07, CD, BE, 55, C9, FD, 7E, 00, F5, 0  
F, 0F, 0F, E6, 07, CD, 05, 56, F1, DD, CB 191

1180 DATA 00, 76, C0, E6, 07, CD, BE, 55, C9,  
11, A3, 5D, C3, 6C, 57, 11, AF, 5D, 18, F8, 11, B  
B, 5D, 18, F3, 11, C7, 5D, 18, EE, 11, 72 99

1190 DATA 5D, CD, 6C, 57, CD, D6, 55, 11, 5B,  
5D, 18, E0, 11, 72, 5D, CD, 6C, 57, 3E, 28, DF, C  
D, D6, 55, 11, 5F, 5D, 18, CF, 11, 78, 5D 201

1200 DATA CD, 6C, 57, CD, D6, 55, 3E, 29, DF,  
C9, 11, D3, 5D, CD, 6C, 57, CD, F9, 55, 3E, 2C, D  
F, C3, D6, 55, FD, 7E, 00, E6, 08, CB, 3F 63

1210 DATA CB, 3F, CB, 3F, 21, 66, 5D, CD, 15,  
56, CD, 6C, 57, C3, D6, 55, 11, D9, 5D, CD, 6C, 5  
7, CD, 29, 56, 3A, 47, 5A, FE, 01, C8, 11 203

1220 DATA 5B, 5D, C3, 6C, 57, 11, E5, 5D, 18,  
E9, 11, EB, 5D, 18, E4, 11, DF, 5D, CD, 6C, 57, C  
3, D6, 55, 11, F1, 5D, 18, F5, DD, CB, 00 90

1230 DATA F6, CD, B1, 54, DD, CB, 00, B6, 11,  
63, 5D, C3, 6C, 57, 11, F7, 5D, CD, 6C, 57, FD, 7  
E, 00, E6, 38, C3, F4, 57, 11, FE, 5D, CD 76

1240 DATA 6C, 57, 2A, 39, 5A, 23, FD, 7E, 01,  
FE, 7F, 06, 00, 30, 06, 4F, 0C, 09, C3, EF, 57, E  
E, FF, 4F, A7, ED, 42, 18, F5, 11, 05, 5E 155

1250 DATA CD, 6C, 57, 3E, 23, DF, 18, DA, 11,  
05, 5E, CD, 6C, 57, FD, 7E, 00, D6, 20, CD, 2C, 5  
6, 3E, 2C, DF, 3E, 23, DF, 18, C4, 21, 4B 57

1260 DATA 5E, CD, 15, 56, 1A, FE, 5F, 20, 0A,  
DD, CB, 00, 5E, C2, D5, 57, 11, 71, 5E, C3, 6C, 5  
7, FD, 7E, 00, 21, 5B, 5E, FE, ED, 28, 07 195

1270 DATA FE, C0, 38, 03, 21, 66, 5E, E6, 30,  
CB, 3F, CB, 3F, CB, 3F, CB, 3F, CD, 15, 56, 1A, F  
E, 48, 20, DA, DD, CB, 00, 5E, C2, D0, 57 192

1280 DATA 11, 76, 5E, 18, CE, 21, 6D, 5C, CD,  
15, 56, CD, 6C, 57, 1B, 1B, 1A, FE, 2C, C8, C9, F  
E, 00, 28, 0E, 0E, 00, 57, 7E, FE, 24, 23 96

1290 DATA 20, FA, 0C, 79, BA, 20, F5, EB, C9,  
FD, 7E, 00, 21, 37, 5E, 0F, 0F, 0F, E6, 07, CD, 1  
5, 56, C3, 6C, 57, 0F, 0F, 0F, E6, 07, C6 235

1300 DATA 30, DF, 3E, 2C, DF, C9, CD, 0B, 58,  
11, 89, 5A, CD, 6C, 57, 3E, 01, CD, 23, 58, 06, 0  
5, 3A, 49, 5A, 21, 6C, 56, BE, 28, 07, 23 91

1310 DATA 23, 23, 10, F8, 18, E0, 23, 7E, 23,  
66, 6F, E9, 41, 7B, 56, 42, EC, 50, 4D, 87, 56, 5  
0, 8C, 56, 53, F2, 56, CD, AC, 59, C0, ED 124

1320 DATA 53, 3B, 5A, CD, EC, 50, C9, DD, CB,  
00, EE, C9, CD, 0B, 58, CD, AC, 59, C0, CD, 0B, 5  
8, CD, C1, 59, C0, ED, 53, 43, 5A, 22, 45 64

1330 DATA 5A, E7, C8, D8, 3E, 01, 32, 16, F4,  
11, 5C, 5A, CD, 6C, 57, DD, CB, 00, 9E, 2A, 43, 5  
A, 22, 39, 5A, CD, EF, 57, CD, 5E, 57, CD 137

1340 DATA 13, 52, C5, 3E, 09, DF, 3A, 47, 5A,  
FE, 01, 20, 03, 3E, 09, DF, CD, B4, 52, C1, 2A, 4  
3, 5A, 09, 22, 43, 5A, CD, 1C, 58, 2A, 43 147

1350 DATA 5A, ED, 5B, 45, 5A, E7, 38, C7, 28,  
C5, 3E, 0C, DF, AF, 32, 16, F4, C9, 21, 05, 26, 1  
1, 56, 5A, 06, 04, C5, E5, CD, C6, 00, CD 197

1360 DATA 49, 57, DD, CB, 00, 46, 20, 35, DD,  
CB, 00, 4E, 20, 27, 4F, E1, E5, 24, 24, CD, C6, 0  
0, C5, CD, 49, 57, CB, 27, CB, 27, C1, DD 67

1370 DATA CB, 00, 46, 20, 18, DD, CB, 00, 4E,  
20, 0A, 81, 06, 00, 21, C1, FC, 09, 4E, 81, 12, E  
1, C1, 13, 2C, 10, BF, 18, 02, E1, C1, 2A 144

1380 DATA 3D, 5A, 22, 3B, 5A, CD, EC, 50, C9,  
3E, 01, CD, 23, 58, 3A, 49, 5A, FE, 0D, 28, 04, D  
F, E6, 03, C9, DD, CB, 00, CE, C9, 11, 3F 14

1390 DATA 5A, 06, 04, CD, D8, 59, 12, 13, 23,  
10, F8, C9, 1A, 13, FE, 24, C8, FE, 21, 28, 1D, F  
E, 25, 28, 2B, FE, 26, 28, 3E, FE, 1D, 20 243

1400 DATA 0E, F5, 3A, 16, F4, FE, 01, 20, 05,  
F1, 3E, 08, 18, 01, F1, DF, 18, DA, 3A, 47, 5A, 2  
A, 41, 5A, FE, 03, 20, 03, 2A, 40, 5A, CD 162

1410 DATA EF, 57, 18, C8, FD, 7E, 01, DD, CB,  
00, 5E, 28, 09, DD, CB, 00, 7E, 20, 03, FD, 7E, 0  
2, CD, F4, 57, 18, B1, D5, 11, 76, 5E, DD 80

1420 DATA CB, 00, 5E, 28, 05, CD, D0, 57, 18,  
03, CD, 6C, 57, D1, 18, 9C, 21, 79, 5E, 18, 03, 2  
1, 7F, 5E, FD, 7E, FF, E6, 20, 07, 07, 07 100

1430 DATA CD, 15, 56, DD, CB, 00, FE, CD, 6C,  
57, DD, CB, 00, BE, C9, 7C, CD, F4, 57, 7D, F5, 0  
F, 0F, 0F, 0F, E6, 0F, CD, FF, 57, F1, E6 145

1440 DATA 0F, C6, 30, FE, 3A, 38, 02, C6, 07,  
DF, C9, 06, 27, 21, 18, 01, CD, C6, 00, 3E, 20, D  
F, 10, FB, CD, C6, 00, C9, 3E, 0D, DF, 3E 147

1450 DATA 0A, DF, C9, 32, 48, 5A, DD, CB, 00,  
86, DD, CB, 00, 8E, 0E, 00, 21, 49, 5A, CD, 9F, 0  
0, FE, 1B, 28, 55, FE, 08, 28, 56, FE, 10 59

1460 DATA CA, 17, 59, 77, 3A, 48, 5A, FE, 01,  
7E, 20, 05, FE, 1E, 28, 54, C9, FE, 0D, 28, 2B, F  
E, 20, 38, DA, FE, 7F, 28, D6, 3A, 48, 5A 14

1470 DATA B9, 28, D0, FE, 01, 28, 11, 7E, FE,  
30, 38, C7, FE, 3A, 38, 08, FE, 41, 38, BF, FE, 4  
7, 30, BB, 7E, 0C, 23, CD, A2, 00, 18, B3 60

1480 DATA 3A, 48, 5A, B9, 28, 04, AF, B9, 20,  
A9, DD, CB, 00, CE, C9, DD, CB, 00, C6, C9, AF, B  
9, 28, 9B, 0D, 2B, D5, 11, 05, 5B, CD, 6C 87

1490 DATA 57, D1, 18, 8F, C5, D5, E5, 2A, DC,  
F3, E5, 06, 20, 21, 03, 01, CD, C6, 00, CD, 13, 5  
8, CD, F5, 58, 06, 20, 21, 16, 01, CD, C6 192

1500 DATA 00, CD, 13, 58, DD, 36, 00, 00, 2A,  
3B, 5A, 22, 39, 5A, CD, EF, 57, CD, 5E, 57, CD, 1  
3, 52, C5, 3E, 0F, 32, DD, F3, CD, B4, 52 237

1510 DATA C1, 2A, 3B, 5A, 09, 22, 3B, 5A, CD,  
1C, 58, E1, 22, DC, F3, E1, D1, C1, C3, 33, 58, 1  
1, 50, 00, 21, 78, 00, 06, 13, C5, 06, 20 102

1520 DATA CD, 4A, 00, EB, CD, 4D, 00, 13, 23,  
EB, 10, F4, 01, 08, 00, 09, EB, 09, EB, C1, 10, E  
7, C9, C5, D5, E5, CD, A8, 00, 28, 45, 2A 126

1530 DATA 22, F9, 0E, 18, 06, 28, CD, 4A, 00,  
FE, FF, 20, 03, 3A, CC, FB, 5F, 3A, 17, F4, B7, 7  
B, 20, 0F, FE, 20, 30, 15, F5, 3E, 01, CD 220

1540 DATA A5, 00, F1, C6, 40, 18, 0A, FE, 80,  
30, 04, FE, 20, 30, 02, 3E, 2E, CD, A5, 00, 23, 1  
0, CF, 3E, 0D, CD, A5, 00, 3E, 0A, CD, A5 89

1550 DATA 00, 0D, 20, C0, E1, D1, C1, C3, 33,  
58, 3E, 04, CD, 23, 58, DD, CB, 00, 46, C0, CD, 9  
9, 59, 21, 49, 5A, 7E, CB, 27, CB, 27, CB 0

1560 DATA 27, CB, 27, 57, 23, 7E, 82, 5F, 23,  
7E, CB, 27, CB, 27, CB, 27, 57, 23, 7E, 8  
2, 6F, 63, C9, 21, 49, 5A, 06, 04, 7E, D6 154

1570 DATA 30, FE, 0A, 38, 02, D6, 07, 77, 23,  
10, F3, C9, CD, 0B, 58, 11, 93, 5A, CD, 6C, 57, C  
D, D1, 59, CD, 6A, 59, EB, DD, CB, 00, 46 8

1580 DATA C9, D5, 11, 9F, 5A, CD, 6C, 57, CD,  
D1, 59, CD, 6A, 59, D1, 18, EB, 11, BD, 5A, CD, 6  
C, 57, C9, E5, D5, C5, DD, E5, FD, E5, CD 126

1590 DATA 1F, 5A, 3A, 5A, 5A, FE, 03, 28, 0F,  
3A, 58, 5A, CD, 0C, 00, FD, E1, DD, E1, C1, D1, E

1,FB,C9,F3,CD,38,01,F5,E5,3A,5B  
 1600 DATA 5A,CD,24,00,E1,F1,5E,CD,3B,  
 01,7B,F5,21,00,D0,3A,55,5A,CD,24,00,F  
 1,FD,E1,DD,E1,C1,D1,E1,FB,C9,C5  
 1610 DATA E5,CB,04,CB,04,7C,E6,03,32,  
 5A,5A,4F,06,00,21,56,5A,09,7E,32,5B,5  
 A,E1,C1,C9,00,00,00,00,00,00,00  
 1620 DATA 00,00,00,00,00,00,00,00,00,00,  
 00,00,00,00,00,00,00,00,00,00,00,00,  
 00,00,00,00,00,00,44,49,53,41  
 1630 DATA 53,53,45,4D,42,4C,45,52,20,  
 76,20,31,2E,30,20,43,6F,70,79,72,69,6  
 7,68,74,20,31,39,38,37,20,62,79  
 1640 DATA 20,4D,43,4D,0D,0A,0D,0A,24,  
 43,6F,6D,6D,61,6E,64,6F,3A,24,53,74,6  
 1,72,74,20,61,64,72,65,73,24,45  
 1650 DATA 69,6E,64,20,61,64,72,65,73,  
 24,44,61,74,61,20,62,6C,6F,6B,6B,65,6  
 E,20,28,4A,2F,4E,29,24,20,28,45  
 1660 DATA 53,43,20,69,73,20,74,65,72,  
 75,67,29,3A,24,50,41,47,1C,53,4C,54,2  
 4,41,30,41,64,72,65,73,24,42,3D  
 1670 DATA 42,6C,61,64,24,4D,3D,42,61,  
 73,69,63,24,50,3D,50,72,69,6E,74,24,5  
 3,3D,53,6C,6F,74,73,24,5E,3D,52  
 1680 DATA 65,67,65,6C,24,08,20,08,24,  
 20,70,A8,20,20,20,20,06,0E,10,16,1  
 8,1E,20,26,28,2E,30,36,38,3E,C6  
 1690 DATA CE,D3,D6,DB,DE,E6,EE,F6,FE,  
 01,11,21,22,2A,31,32,3A,C2,C3,C4,CA,C  
 C,CD,D2,D4,DA,DC,E2,E4,EA,EC,F2  
 1700 DATA F4,FA,FC,09,19,23,29,2B,39,  
 E1,E3,E5,E9,F9,34,35,46,4E,56,66,6E,7  
 0,71,72,73,74,75,77,7E,86,8E,96  
 1710 DATA 9E,A6,AE,B6,BE,5E,21,22,2A,  
 36,40,41,42,44,45,46,47,48,49,4A,4D,5  
 0,51,56,57,58,59,5A,5E,60,61,62  
 1720 DATA 67,68,69,6A,6F,72,78,79,7A,  
 A0,A1,A2,A3,AB,A9,AA,AB,B0,B1,B2,B3,B  
 8,B9,BA,BB,52,4F,5F,43,4B,53,5B  
 1730 DATA 73,7B,00,08,76,C3,C9,CD,D3,  
 D9,DB,EB,F3,FB,E3,E9,F9,07,0F,17,1F,2  
 7,2F,37,3F,20,28,30,38,44,45,4D  
 1740 DATA 46,56,5E,47,57,67,6F,4F,5F,  
 A3,AB,B3,BB,44,45,46,42,20,23,24,4E,4  
 F,50,24,45,58,20,20,20,41,46,2C  
 1750 DATA 41,46,27,24,48,41,4C,54,24,  
 4A,50,20,20,20,23,21,24,52,45,54,24,4  
 3,41,4C,4C,20,23,21,24,4F,55,54  
 1760 DATA 20,20,23,25,2C,41,24,45,58,  
 58,24,49,4E,20,20,20,41,2C,23,25,24,4  
 5,58,20,20,20,44,45,2C,48,4C,24  
 1770 DATA 44,49,24,45,49,24,45,58,20,  
 20,20,28,53,50,29,2C,26,24,4A,50,20,2  
 0,20,28,26,29,24,4C,44,20,20,20  
 1780 DATA 53,50,2C,26,24,52,4C,43,41,  
 24,52,52,43,41,24,52,4C,41,20,24,52,5  
 2,41,20,24,44,41,41,20,24,43,50  
 1790 DATA 4C,20,24,53,43,46,20,24,43,  
 43,46,20,24,41,44,44,20,20,41,2C,24,4  
 1,44,43,20,20,41,2C,24,53,55,42  
 1800 DATA 20,20,24,53,42,43,20,20,41,  
 2C,24,41,4E,44,20,20,24,58,4F,52,20,2  
 0,24,4F,52,20,20,20,24,43,50,20  
 1810 DATA 20,20,24,4E,45,47,24,52,45,

154  
 235  
 63  
 86  
 73  
 127  
 44  
 220  
 178  
 26  
 101  
 122  
 238  
 38  
 24  
 40  
 102  
 89  
 36  
 178  
 173  
 82

54,4E,24,52,45,54,49,24,49,4D,20,20,2  
 0,30,24,49,4D,20,20,20,31,24,49  
 1820 DATA 4D,20,20,20,32,24,4C,44,20,  
 20,20,49,2C,41,24,4C,44,20,20,20,41,2  
 C,49,24,52,52,44,24,52,4C,44,24  
 1830 DATA 4C,44,20,20,20,52,2C,41,24,  
 4C,44,20,20,20,41,2C,52,24,4C,44,49,2  
 4,43,50,49,24,49,4E,49,24,4C,44  
 1840 DATA 44,24,43,50,44,24,49,4E,44,  
 24,4F,55,54,49,24,4F,55,54,44,24,4F,5  
 4,49,52,24,4F,54,44,52,24,52,4C  
 1850 DATA 43,20,20,24,52,52,43,20,20,  
 24,52,4C,20,20,20,24,52,52,20,20,20,2  
 4,53,4C,41,20,20,24,53,52,41,20  
 1860 DATA 20,24,24,53,52,4C,20,20,24,  
 42,49,54,20,20,24,52,45,53,20,20,24,5  
 3,45,54,20,20,24,2C,23,21,24,29  
 1870 DATA 2C,41,24,23,25,24,49,4E,43,  
 20,20,24,44,45,43,20,20,24,4C,44,20,2  
 0,20,24,4C,44,20,20,20,41,2C,28  
 1880 DATA 24,4C,44,20,20,20,28,23,21,  
 29,2C,24,4C,44,20,20,20,20,2C,28,2  
 3,21,29,1D,1D,1D,1D,1D,1D,1D,1D  
 1890 DATA 1D,1D,24,4C,44,20,20,20,28,  
 23,21,29,2C,41,24,4C,44,20,20,20,41,2  
 C,28,23,21,29,24,4C,44,20,20,20  
 1900 DATA 28,23,21,29,2C,26,24,4C,44,  
 20,20,20,26,2C,28,23,21,29,24,41,44,4  
 4,20,20,24,52,45,54,20,20,24,50  
 1910 DATA 4F,50,20,20,24,4A,50,20,20,  
 20,24,43,41,4C,4C,20,24,50,55,53,48,2  
 0,24,52,53,54,20,20,23,24,44,4A  
 1920 DATA 4E,5A,20,23,24,4A,52,20,20,  
 20,24,49,4E,20,20,20,20,2C,28,43,29,1  
 D,1D,1D,1D,1D,24,4F,55,54,20,20  
 1930 DATA 28,43,29,2C,24,53,42,43,20,  
 20,48,4C,2C,24,41,44,43,20,20,48,4C,2  
 C,24,4E,5A,24,5A,24,4E,43,24,43  
 1940 DATA 24,50,4F,24,50,45,24,50,24,  
 4D,24,42,24,43,24,44,24,45,24,48,24,4  
 C,24,5F,24,41,24,42,43,24,44,45  
 1950 DATA 24,26,24,53,50,24,42,43,24,  
 44,45,24,26,24,41,46,24,28,48,4C,29,2  
 4,48,4C,24,49,58,24,49,59,24,28  
 1960 DATA 49,58,2B,23,25,29,24,28,49,  
 59,2B,23,25,29,24  
 1970 '  
 1980 '  
 1990 ' VERPLAATS ROUTINE  
 2000 '  
 2010 DATA 26,7F,CD,2D,BF,FE,FF,20,71,  
 11,16,BF,1A,FE,24,C8,CD,A2,00,13,18,F  
 6,47,65,65,6E,20,36,34,4B,20,52  
 2020 DATA 41,4D,20,67,65,76,6F,6E,64,  
 65,6E,21,24,0E,00,79,E6,03,16,00,5F,E  
 5,21,C1,FC,19,7E,E1,81,32,5B,BF  
 2030 DATA 3E,FF,32,9D,BF,C5,CD,5A,BF,  
 C1,3A,9D,BF,FE,FF,C0,0C,79,FE,10,20,D  
 9,3A,9D,BF,C9,3E,00,F5,CD,0C,00  
 2040 DATA 2F,5F,F1,F5,CD,14,00,F1,F5,  
 D5,CD,0C,00,D1,AB,28,03,F1,FB,C9,F1,3  
 2,9D,BF,FB,C9,01,8F,0E,11,00,C0  
 2050 DATA 21,00,50,1A,CD,8F,BF,0B,13,  
 23,78,B1,20,F5,C9,C5,D5,E5,5F,3A,9D,B  
 F,CD,14,00,E1,D1,C1,C9,00

228  
 87  
 21  
 64  
 113  
 195  
 111  
 189  
 2  
 16  
 70  
 85  
 125  
 134  
 198  
 236  
 0  
 0  
 0  
 0  
 54  
 8  
 84  
 115  
 63

# VarIn 2

Een tijdje geleden alweer, in MCM nummer 7, publiceerden we een tweetal handige hulpjes voor programmeurs, Varlst en Linlst. Die beide programma's maakten het lezen van de Basic-programmeur wat eenvoudiger door keurige lijsten te maken van welke variabelen waar gebruikt werden en vanuit welke regels naar andere regels gesprongen werd.

Maar niets is ideaal, want hoewel Varlst en Linlst hun werk uitstekend deden waren ze - zacht gezegd - niet van de snelste. Beide hulpjes waren in feite wel erg traag, hetgeen voor ons een reden was om nog eens wat verder te zoeken naar snellere oplossingen. Gelukkig hebben we die nu gevonden en we stellen u dan ook met een zekere trots voor aan ons nieuwe programma: VarIn2.

Een korte test leerde ons dat het samenstellen van een variabele-lijst voor een bepaald programma - met Varlst kostte zo'n zestien minuten - nu in zestien seconden gepiept was!

Als u meer wilt weten over het hoe en waarom van dergelijke *cross-reference* programma's moet u het artikel in MCM nummer 7 er nog maar eens op nalezen.

## Machinetaal

Het zal niemand met enige ervaring op MSX-gebied verbazen dat we voor die snellere oplossing een geheel andere route hebben moeten kiezen. In Basic is een dergelijke klus nu eenmaal niet snel te doen. Vandaar ook dat deze versie in machinetaal geschreven is, een hele verbetering. Niet alleen in snelheid, maar ook in gebruiksgemak. Want nadat de Basic-loader eenmaal gerund is blijft de eigenlijke werkende kode rustig in het geheugen zitten, op een 'onzichtbare' plaats. Men kan rustig in Basic programmeren of zelfs een nieuw programma laden. Als men gebruik wilt maken van de diensten van het hulpprogramma kan dat

meteen gebeuren, simpelweg door een funktietoets in te drukken.

Inderdaad, alletwee de funkties, zowel variabelen overzichten aanmaken als regelnummer-tabellen samenstellen zijn namelijk in een enkel stuk ML verenigd. Sterker nog, men mag zelfs met een funktietoets kiezen tussen de twee verschillende uitvoermogelijkheden, scherm of printer. Een kind kan de was doen!

## Werkwijze

Zoals reeds gezegd, het eigenlijke *cross-reference* programma is geheel in machinetaal geschreven. Toch zult u bij dit verhaal een Basic-listing aantreffen.

Die listing omvat namelijk naast onze handige *reloceer-routine* de eigenlijke ML zelf. Dat Basic-programma is een verdraaid handig ladertje, dat in staat is de machinetaal zo gunstig mogelijk in het MSX-geheugen onder te brengen.

Normaal gesproken moet een machinetaal-programma namelijk zo geschreven worden dat het maar op een bepaalde plek in het geheugen van een computer kan worden geladen. Dat komt doordat zo'n programma, net als Basic-programma's dat doen, allerlei sprongen maakt. Maar daar waar zo'n Basic-programma naar een bepaald *regelnummer* springt, daar springt een ML-programma naar een *adres*. Die Basic-regelnummers nu worden tijdens het uitvoeren van het Basic-programma iedere

keer opgezocht, ze hoeven niet steeds op precies dezelfde plaats te staan. Als we veranderingen in een Basic-programma aanbrengen zullen we ook bijna altijd de lokatie van een of meer regelnummers verschuiven, doordat regels ervoor korter of langer worden.

Bij machinetaal echter gaan sommige sprongen keihard naar een bepaald adres, en moeten wij er maar voor zorgen dat op dat adres de juiste routine begint.

## Geheugen-verlies

Dat levert vooral bij MSX-computers nogal wat problemen op. De simpelste wijze namelijk om ervoor te zorgen dat die interne sprongen binnen een ML-programma kloppen is dat programma altijd vanaf hetzelfde adres te laden. Daarvoor wordt dan een stukje geheugen afgeknipt van de eigenlijk voor Basic gereserveerde ruimte, met het CLEAR-kommando. Het is daarbij echter wel zaak om er voor te zorgen dat dat stukje geheugen zo hoog mogelijk ligt, zodat er niet teveel Basic-ruimte verloren gaat.

En daar zit hem nou net de kneep. Want de ene MSX is de andere niet. Zo heeft een MSX zonder diskdrive meer vrij Basic-geheugen dan een tje die wel een drive gebruikt, terwijl er ook tussen MSX1 en MSX2 verschillen bestaan wat dit betreft. Om nu zonder meer uit te gaan van een startadres dat voor *alle* MSXen veilig is lijkt ons een wat al te simpele oplossing; immers, dan zou er op de meeste machines geheugen verkast worden.

Vandaar dat deze Basic-lader heel wat slimmer in zijn werk is gemaakt. Eerst wordt het hoogste voor Basic beschikbare adres bepaald waarna dit adres zoveel lager wordt gesteld als nodig is voor onze ML. De CLEAR-opdracht in regel 190 handelt dit allemaal in een keer af. Daarna wordt de ML uit de DATA-regels ingelezen en op zijn plaats gepoke'd, waarbij die *absolute* sprongen

door sterretjes worden aangegeven. In regel 240 worden die sterren - ASCII-waarde 42 - ondervangen en berekend ons ladertje het eigenlijke adres waarheen er gesprongen zal moeten gaan worden.

Op die manier geeft deze Basic-lader - dezelfde die we bijvoorbeeld ook voor het Invoer Controle Programma gebruiken - u het beste van twee werelden, namelijk een snel ML-programma dat op de meest handige plek in het geheugen geplaatst wordt.

## Extra controle

Bovendien controleert dit laad-programma voor alle veiligheid ook nog eens of u de machinetaal die in de DATA-regels staat wel goed heeft ingetikt. In regel 220 wordt alle data-stukjes bij elkaar opgeteld in een variabele, die in regel 270 vergeleken wordt met de waarde' deze zou moeten hebben. Klopt dat niet, dan weigert het programma meteen dienst. Dat voorkomt dat u met een verkeerd ML-programma probeert te werken, wat de meest onvoorspelbare resultaten op kan leveren.

Bovendien voorkomt hopelijk ook de wat vermoeiende gesprekken die onze redactie iedere week moet voeren tijdens het telefonisch vragenuurtje. Ongelofelijk hoe hardnekkig sommige mensen volhouden toch echt geen tikfouten gemaakt te hebben!

## Gebruik

In het gebruik is deze bundeling van twee gereedschappen werkelijk heel simpel. Na het RUNnen van de lader kan deze geNEWd worden, waarna u of een oud programma kan inladen of meteen een heel nieuw programma kunt gaan ontwikkelen.

Uw computer reageert net zoals altijd, er is alleen iets minder dan 1 Kilobyte Basic-geheugen minder beschikbaar, terwijl de funktietoetsen F1, F2, F6 en F7 nieuwe teksten gekregen hebben.

Zodra u echter een van die nieuwe kommando's intikt,

```

10 REM VarLin - CMD versie
20 REM
30 REM MSX Computer Magazine
40 REM
50 REM
60 SCREEN 0:WIDTH 40
70 CLS:LOCATE 15,2:PRINT"VarLin"
80 PRINT:PRINT:PRINT"VarList en Lin
Lister ineen."
90 PRINT:PRINT"Dit zijn de kommando's
:
100 PRINT:PRINT"F1: CMD LLIST LineLi
st (scherm)"
110 PRINT"F2: CMD VLIST Varlist (sch
erm)"
120 PRINT:PRINT"F6: CMD LPRINT LineLi
st (printer)"
130 PRINT"F7: CMD VPRINT VarList (pri
nter)"
140 KEY 1,"cmd llist"+CHR$(13)
150 KEY 2,"cmd vlist"+CHR$(13)
160 KEY 6,"cmd lprint"+CHR$(13)
170 KEY 7,"cmd vprint"+CHR$(13)
180 PRINT:PRINT"Even geduld a.u.b."
190 CLEAR 200,PEEK(&HFC4A)+256*PEEK(&
HFC4B)-&H339-9
200 ST=PEEK(&HFC4A)+256*PEEK(&HFC4B)
210 FOR I=0 TO &H339:READ AS
220 CS=CS+ASC(LEFT$(AS,1))+ASC(RIGHT$
(AS,1))
230 IF ASC(AS)=42 THEN BS=RIGHT$(AS,1
):READ AS:A=ST+VAL("&H"+BS+AS):POKEST
+I,A-INT(A/256)*256:I=I+1:POKEST+I,IN
T(A/256):GOTO 250
240 POKE ST+I,VAL("&h"+AS)
250 NEXT I
260 ' KONTROLEER DATA-WAARDES *****
270 IF CS<>928381 THEN CLS: PRINT "U
heeft een fout gemaakt in de data-
regels!": PRINT: PRINT "Eerst verbet
eren!": STOP
280 PRINT:PRINT"Machinecode geinstall
eerd op "HEX$(ST)
290 DEF USR=ST:S%=ST-65536!:A=USR(S%)
300 PRINT:END
310 DATA 2A,F8,F7,11,39,03,19,E5,E5,2
A,F8,F7,11,32,00,19
320 DATA EB,E1,23,73,23,72,23,EB,21,0
D,FE,01,05,00,ED,B0
330 DATA 3E,C9,32,0D,FE,D1,21,0E,FE,7
3,23,72,3E,C3,32,0D
340 DATA FE,C9,F5,C5,D5,E5,0E,00,16,0
1,FE,9E,28,1E,0C,FE
350 DATA 9D,28,19,FE,56,20,10,E1,D7,E
5,16,00,0E,00,FE,93
360 DATA 28,0A,0C,FE,91,28,05,E1,D1,C
1,F1,C9,2A,C2,F6,22
370 DATA C4,F6,22,C6,F6,E5,23,23,22,E
E,F6,E1,7A,32,F0,F6
380 DATA 79,32,16,F4,ED,73,F2,F6,3E,0
0,77,23,77,CD,F7,54
390 DATA 21,*2,CE,3A,F0,F6,B7,28,03,2
1,*2,F2,7E,B7,28,04
400 DATA DF,23,18,F8,2A,76,F6,E5,7E,2
3,86,28,1B,23,4E,23
410 DATA 46,ED,43,E8,F6,3A,F0,F6,B7,F
5,CC,*1,4E,F1,C4,*1
420 DATA F1,E1,5E,23,56,EB,18,DF,E1,2
A,C2,F6,5E,23,56,23
430 DATA 7B,B2,28,51,3E,0D,DF,3E,0A,D
F,3A,F0,F6,B7,20,16
440 DATA 7A,E6,80,28,04,3E,2A,18,02,3
E,20,DF,7A,E6,7F,DF
450 DATA 7B,DF,7E,DF,18,06,EB,D5,CD,1

```

```

0
0
0
0
151
20
28
247
135
140
233
124
148
63
252
167
39
64
185
214
248
242
239
210
0
116
90
238
217
119
11
197
72
116
43
96
86
239
213
141
2
60
182

```

```

2,34,E1,3E,3A,DF,23
460 DATA 5E,23,56,23,7A,A3,FE,FF,28,C
2,3E,09,DF,E5,EB,CD
470 DATA 12,34,CD,9C,00,28,0A,CD,9F,0
0,CD,9F,00,FE,1B,28
480 DATA 03,E1,18,DC,E1,3E,0D,DF,3E,0
A,DF,3E,00,32,16,F4
490 DATA E1,D1,C1,F1,F1,F1,D7,C9,E5,D
5,EB,11,32,00,19,37
500 DATA 3F,ED,72,D1,E1,D8,21,*3,17,7
E,B7,28,04,DF,23,18
510 DATA F8,ED,7B,F2,F6,18,CE,FE,41,3
F,00,FE,5B,C9,D7,CA
520 DATA *1,0B,FE,20,38,6F,FE,84,CA,*
1,E0,FE,8F,28,77,FE
530 DATA E6,28,73,FE,22,28,65,CD,*1,4
7,30,E2,57,D7,CD,*1
540 DATA 47,38,10,FE,30,38,04,FE,3A,3
8,08,1E,00,ED,53,EC
550 DATA F6,18,0B,5F,ED,53,EC,F6,D7,C
D,*1,47,38,FA,47,FE
560 DATA 24,28,13,FE,23,28,0F,FE,21,2
8,0B,FE,25,28,07,3E
570 DATA 00,32,EA,F6,18,07,32,EA,F6,D
7,47,18,01,78,FE,28
580 DATA 20,08,3A,ED,F6,F6,80,32,ED,F
6,C5,E5,CD,*2,1C,E1
590 DATA C1,78,B7,18,8A,2A,66,F6,7E,B
7,18,83,D7,B7,C8,FE
600 DATA 22,20,F9,C3,*1,4E,D7,B7,20,F
C,C9,B7,C2,*1,4E,C9
610 DATA D7,28,F8,FE,22,20,F9,D7,B7,2
8,F0,FE,22,20,F8,18
620 DATA EF,D7,28,24,FE,20,30,F9,FE,0
E,20,15,2A,66,F6,2B
630 DATA 56,2B,5E,ED,53,EC,F6,3E,00,3
2,EA,F6,E5,CD,*3,33
640 DATA E1,C3,*3,28,7E,B7,18,DA,B7,2
0,D6,C9,2A,C2,F6,E5
650 DATA 4E,23,46,23,7E,32,EB,F6,E1,7
8,B1,28,36,E5,ED,5B
660 DATA EC,F6,60,69,E7,E1,20,09,E5,2
A,EA,F6,7C,BD,E1,28
670 DATA 57,78,92,38,11,20,1C,79,93,3
8,0B,20,16,E5,2A,EA
680 DATA F6,7C,BD,E1,30,0D,23,23,23,7
E,23,A6,23,FE,FF,20
690 DATA F8,18,BC,EB,D5,2A,EE,F6,37,3
F,ED,52,44,4D,2A,EE
700 DATA F6,54,5D,13,13,13,13,ED,5
3,EE,F6,2B,1B,CD,*1
710 DATA 28,ED,B8,E1,E5,ED,5B,EC,F6,7
3,23,72,23,3A,EA,F6
720 DATA 77,23,3E,FF,77,23,77,E1,23,2
3,23,E5,5E,23,56,2A
730 DATA E8,F6,E7,E1,28,27,30,F1,EB,D
5,2A,EE,F6,37,3F,ED
740 DATA 52,44,4D,2A,EE,F6,54,5D,13,1
3,ED,53,EE,F6,2B,1B
750 DATA CD,*1,28,ED,B8,E1,ED,5B,E8,F
6,73,23,72,C9,0D,0A
760 DATA 4D,53,58,20,43,6F,6D,70,75,7
4,65,72,20,4D,61,67
770 DATA 61,7A,69,6E,65,20,56,61,72,4
C,69,73,74,65,72,0D
780 DATA 0A,00,0D,0A,4D,53,58,20,43,6
F,6D,70,75,74,65,72
790 DATA 20,4D,61,67,61,7A,69,6E,65,2
0,4C,69,6E,65,4C,69
800 DATA 73,74,65,72,0D,0A,00,0D,0A,4
F,75,74,20,6F,66,20
810 DATA 6D,65,6D,6F,72,79,21,CD,FE,0
1,CA,*1,F1,2A,66,F6
860 DATA C3,*2,14,7B,B2,C8,C3,*2,1C,C
D

```

```

169
174
17
57
99
189
157
38
145
46
195
155
171
170
30
253
224
140
147
117
6
97
103
82
52
164
230
128
25
166
76
119
243
255
3
11
60
216

```

# Blast

Natuurlijk zijn we op de MSX Computer Magazine redactie niet voor een kleintje vervaard, als het op typen aankomt. Ondanks het feit dat we geen van allen met tien vingers tikken, laat staan blind kunnen typen, weten we toch een redelijk tempo te halen. Dat mag ook wel, gezien de hoeveelheid kopij die er voor ieder nummer van het blad nodig is.

Vandaar misschien dat we een spel als Blast zelf heel leuk vinden. Blast is namelijk een type-spelletje, waarbij de speler de woorden op het scherm moet intikken voor die de laserstraal bereiken. Een soortement edukatief spel dus, hoe sneller men tikken kan, hoe meer punten er te behalen zijn. En hoe gaat dat met mensen, dan wil men wel snel leren tikken.

## Leuk

Nu hebben we al heel wat type-spelletjes onder ogen gehad. Wat heet, we hebben er zelfs al eens eentje gepubliceerd. Maar Blast spant wat ons betreft toch wel de kroon, als het om een verzorgd beeldscherm gaat. De graphics zijn prima!

Behalve de woorden zelf — die in een eigen lettertje zijn uitgevoerd — komt er nog wel meer over het scherm voorbij zeilen. Sterren bijvoorbeeld, die het geheel een wat spacy uiterlijk verlenen. Heel knap: die sterren komen met verschillende snelheden voorbij, hetgeen een soort dieptewerking geeft.

Blast is qua geluid ook niet op zijn mondje gevallen, er komen hele aardige geluidseffecten in voor. Jammer genoeg geen echte muziek, maar wel een duidelijk nerveus-makend ritme. Een geluidje dat bovendien tot on-

vrede op de redactie geleid heeft, aangezien sommigen er niet tegenkonden. Maar dat waren dan ook geen spel-liefhebbers.

Over de kleurkeuze voor de letters zou men kunnen twisten. Het donkerblauw is wat lastig leesbaar en maakt daarom Blast eigenlijk iets moeilijker dan nodig. Maar aan de andere kant; de kleuren zijn wel zodanig gekozen dat ook op een monochroom beeldscherm gespeeld kan worden.

Grappig detail: het caps-lock lampje, dat aangeeft of de hoofdletters-toets vast staat, knippert driftig mee onder het spelen.

## Score

Het spel kent een high-score, die in eerste instantie op 5000 is gezet. Om dat te overtreffen is nog geen heksen-tor. Maar om boven de tienduizend punten te komen, dat is al heel wat anders.

Op de redactie zijn we tot ongeveer 16000 gekomen, maar dat was nadat we een truukje bedacht hadden om een beetje vals te spelen.

Gemeen is dat Blast de speler in de luren legt. Als men eenmaal een tijdje gespeeld heeft, denkt men al snel dat men alle woorden nu wel eens gezien heeft. Kortom, zodra de letters 'ee' verschij-

nen, begint de dappere speler of speelster meteen het woord 'eekhoorn' in te tikken. Mis! Het moest dit keer 'eenhoorn' zijn, een goede reden om vooral ook op het scherm te blijven kijken.

Het spel kent een aantal niveaus, die van elkaar verschillen doordat de laserstraal meer naar rechts verschoven is. Met andere woorden: dichterbij de linkerzijde van het scherm, waar de weg te tikken woorden verschijnen. Men heeft dus korter de tijd om een woord weg te vagen.

De puntentelling is aan die niveaus gekoppeld. Per letter krijgt men namelijk altijd 5 punten, plus nog eens het niveau maal twee. Met andere woorden, het woord 'electromagneet', dat 14 letters telt, levert op het eerste niveau 14 maal (5 plus 2 maal 1) punten op. Voor wie niet zo snel in hoofdrekken is, dat zijn 98 punten.

Hetzelfde woord op level 10 geeft echter 350 punten op. Juist op de lastiger niveaus liggen de punten voor het oprapen, als men tenminste snel genoeg tikken kan.

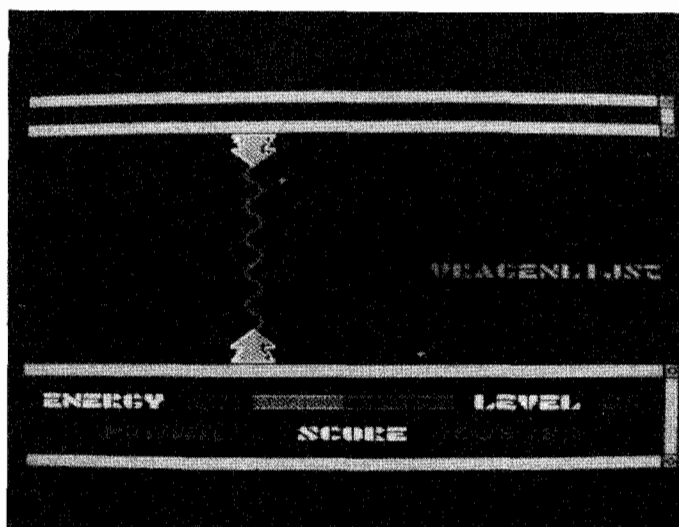
heel aardige oplossing. Alles draait daarbij om de 'energie', die de speler heeft. Deze energie wordt in procenten uitgedrukt en valt af te lezen op een balk, onderaan het scherm. In eerste instantie staat die indicator op honderd procent; voor iedere letter die in de laserstraal verzeild raakt verliest de speler 2% energie. Kortom, na vijftig gemiste letters is het spel gewoon over?

Nee, want voor ieder uitgespeeld niveau krijgt men er weer 4% energie bij! Deze wat flexibeler manier van omgaan met fouten — want dat is het in feite — maakt Blast des te aantrekkelijker. Hoe hoger men komt, hoe meer foutjes zijn toegestaan.

## Extra's

Wie wil kan heel simpel andere woorden invoegen. Echter, alleen kleine letters worden juist weergegeven, hoofdletters in de in te tikken woorden zijn verboden.

Bovendien moeten er dan wel een paar andere zaken ook worden aangepast. Zo wordt in regel 330 het array A\$ gedimensioneerd, waarin



## Energie

Natuurlijk zal Blast niet eindeloos doorgaan, er is ook een manier om de beurt te verliezen. De programmeur, Rob van Hout, heeft daarbij gekozen voor een wat gecompliceerde maar eigenlijk

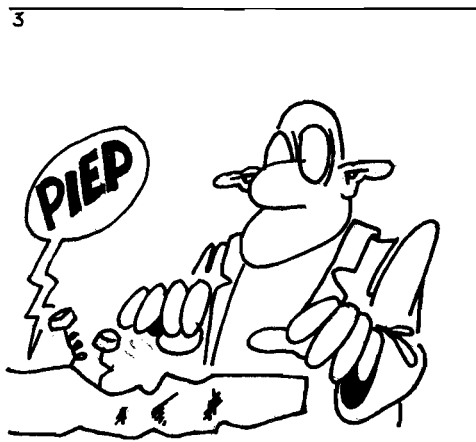
de woorden worden opgeslagen. Momenteel wordt er ruimte voor precies honderd woorden gereserveerd; dat aantal zal men dienen te verhogen.

Hetzelfde geldt voor regel 360, waar de woorden uit de data-regels in het array wor-

den ingelezen. Ook daar moet een teller worden aangepast.

Overigens, Blast is beveiligd tegen de Control-Stop toets-kombinatie. De enige manier om het programma te onderbreken is middels de toets-kombinatie Control + Shift + Graph + Code. Een concertpianist slaat zoiets met één hand aan, de rest van de mensheid mag twee handen gebruiken.

Lastig is alleen dat ook na het onderbreken de Stop-toets uitgeschakeld blijft. Een reset is de eenvoudigste manier om dit euvel weer uit de wereld te helpen, maar tijdens de laatste fase van het 'debuggen' van het ingetikte programma zult u er wel eens tureluurs van worden, zo'n listing die onstopbaar op het scherm voorbij scrollt. Ook iets om op te letten tijdens het intikken zijn de op het eerste gezicht speciale tekens in bijvoorbeeld regel 580. Al die vreemd ogende 'majorette-stokjes' zijn echter gewone procent-tekentjes. De korte verticale streepjes in de regel daaronder zijn gewone enkele aanhalingstekens, zoals die ook in plaats van een REM-kommando gebruikt mogen worden. Blast bevat – zoals gebruikelijk – een extra interne check op de data-regels. Als u een foutje maakt bij het intikken van de vele getallen die het eigen lettertype bevatten, dan zal het programma afbreken met



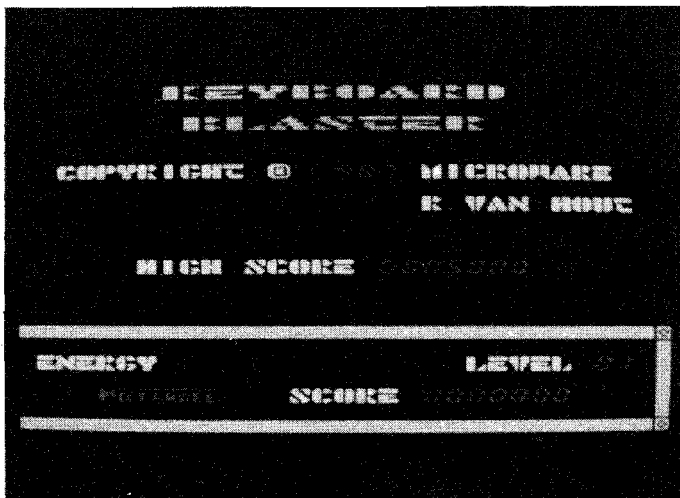
een foutmelding. Gewoon gaan zoeken tot u de fout gevonden hebt, is onze raad. Het programma zal zich echt niet vergissen, als het aangeeft dat u een fout gemaakt heeft, dan is dat ook zo. Wat u in géén geval moet doen is de MSX Computer Magazine redactie opbellen, want dat levert alleen maar frustratie op, zowel voor u als voor ons. Ook wij kunnen in zo'n geval namelijk niets an-

ders zeggen dan dat u toch echt een foutje gemaakt moet hebben.

### Verzorgd

We zeiden het reeds, Blast is tot in de puntjes verzorgd. Goede graphics, die echter zo simpel gehouden zijn dat dit geheel in Basic geschreven spel toch snel genoeg blijft. Een aardig intro-scherm, met daarop een (Engelstalige) gebruiksaan-

wijzing, die keurig voorbij rolt, geluidseffekten op de juiste momenten. Naar onze mening doet Blast niet of nauwelijks onder voor het door Philips uitgebrachte Tempo Typen. Toegegeven, het is weer eens een hele lap om in te tikken, maar zeker de moeite waard. We willen bij deze de gelegenheid aangrijpen om de inzender, Rob van Hout, nogmaals hartelijk te bedanken voor zijn prima programma.



```

10 REM BLAST
20 REM
30 REM MSX Computer Magazine
40 REM
50 REM ingezonden door: Rob van Hout,
   Helmond
60 REM
70 '
80 ' initialisatie *****
90 '
100 ' CTRL+SHIFT+GRPH+CODE => STOP
110 '
120 SCREEN 1,2,0,2: COLOR 15,1,1: WID
  
```

```

TH 32: KEY OFF: POKE &HFBB1,1: POKE &
HFBB0,1 114
130 FOR A=384 TO 727: READ B: CH=CH+B
: VPOKE A,B: NEXT A 131
140 FOR A=280 TO 303: READ B: CH=CH+B
: VPOKE A,B: NEXT A: VPOKE 8196,225 167
150 LOCATE 6,8: PRINT "#"+STRING$(18,
37)+"#": LOCATE 6,9: PRINT "$";SPC(18
);"$": LOCATE 6,10: PRINT "$ KEYBOARD
BLASTER $": LOCATE 6,11: PRINT "$";S
PC(18);"$": LOCATE 6,12: PRINT "#"+ST
RING$(18,37)+"# 67
160 LOCATE 11,17: PRINT "F1 PAUSE":
LOCATE 11,19: PRINT "F2 STOP" 234
170 SOUND 13,5: SOUND 9,16: SOUND 10,
16: SOUND 1,1: SOUND 2,250: SOUND 3,3
: SOUND 4,230: SOUND 5,3: SOUND 12,70
: SOUND 11,0: SOUND 13,5 162
180 FOR A=1024 TO 1087: READ B: CH=CH
+B: VPOKE A,B: NEXT A 193
190 RESTORE 2020: FOR A=1088 TO 1151:
READ B: CH=CH+B: VPOKE A,B: NEXT A 213
200 RESTORE 2020: FOR A=1152 TO 1215:
READ B: CH=CH+B: VPOKE A,B: NEXT A 11
210 FOR A=1272 TO 1407: READ B: CH=CH
+B: VPOKE A,B: NEXT A 153
220 FOR A=1 TO 2: S$="": FOR B=1 TO 3
2: READ C: CH=CH+C: S$=S$+CHR$(C): NE
XT B: SPRITE$(A)=S$: NEXT A 214
230 IF CH<>94042! THEN CLS: SCREEN 0:
PRINT "Fout in dataregels.": END 87
240 FOR A=8197 TO 8199: VPOKE A,97: N
EXT A 180
250 FOR A=8204 TO 8207: VPOKE A,81: N
EXT A 97
260 VPOKE 8208,97: VPOKE 8209,161: VP
OKE 8210,33 53
270 FOR A=520 TO 727: VPOKE A+256,VPE
EK(A): NEXT A 84
280 FOR A=384 TO 391: VPOKE A-48,VPEE
K(A): NEXT A 50
290 FOR A=281 TO 287: VPOKE A-17,VPEE
K(A): NEXT A 216
300 FOR A=289 TO 295: VPOKE A+15,VPEE
K(A): NEXT A 158
310 FOR A=296 TO 302: VPOKE A+15,VPEE
K(A): NEXT A 176
320 FOR A=8211 TO 8213: VPOKE A,65: N
EXT A 66
330 CLEAR 11000: DEFINT A-Z: DIM E$(9
6),A$(100): RESTORE 2350 246
340 FOR A=1 TO 2: FOR B=1 TO 32: READ
C: CH=CH+C: S$(A)=S$(A)+CHR$(C): NEX
T B: NEXT A 219
350 IF CH<>2352 THEN CLS: SCREEN 0: P
RINT "Fout in dataregels.": END 102
360 FOR A=1 TO 100: READ A$(A): A$(A)
=A$(A)+" ": NEXT A 211
370 N$="microware": HS=5000: E$(0)="L

```

```

OW " 222
380 FOR A=128 TO 135: E1$=E1$+CHR$(A)
: NEXT A 48
390 FOR A=136 TO 143: E2$=E2$+CHR$(A)
: NEXT A 67
400 FOR A=144 TO 151: E3$=E3$+CHR$(A)
: NEXT A 67
410 E4$=STRING$(3,128)+STRING$(4,136)
: E5$=STRING$(3,128) 190
420 Q=5: FOR A=96 TO 57 STEP -8: Q=Q-
1: FOR B=1 TO 8: E$(A-(B-1))=E4$+STRI
NG$(Q,144)+MID$(E3$,B,1): NEXT B: NEX
T A 84
430 Q=4: FOR A=56 TO 25 STEP -8: Q=Q-
1: FOR B=1 TO 8: E$(A-(B-1))=E5$+STRI
NG$(Q,136)+MID$(E2$,B,1): NEXT B: NEX
T A 219
440 Q=3: FOR A=24 TO 1 STEP -8: Q=Q-1
: FOR B=1 TO 8: E$(A-(B-1))=STRING$(Q
,128)+MID$(E1$,B,1): NEXT B: NEXT A 158
450 ' 0
460 ' titelbeeld ***** 0
470 ' 0
480 CLS: E=96: L=0: SOUND 12,18: DEFU
SR=&H156: ON KEY GOSUB 1460,1520 62
490 A$=CHR$(159)+CHR$(161)+CHR$(160)+
CHR$(162)+CHR$(163)+CHR$(164)+CHR$(15
9)+CHR$(165)+CHR$(166)+CHR$(168)+CHR$(
167)+CHR$(169)+CHR$(159)+CHR$(170)+C
HR$(159)+CHR$(168) 94
500 B$=CHR$(159)+CHR$(165)+CHR$(159)+
CHR$(172)+CHR$(167)+CHR$(169)+CHR$(17
1)+CHR$(173)+CHR$(174)+CHR$(175)+CHR$(
160)+CHR$(162)+CHR$(159)+CHR$(170) 26
510 FOR A=8211 TO 8213: VPOKE A,225:
NEXT A 145
520 SOUND 8,10: FOR A=1 TO 16 STEP 2:
FOR B= 30 TO 7+A STEP -1: SOUND 0,25
5-(B*7): LOCATE B,2: PRINT MID$(A$,A,
2)+" ": NEXT B: NEXT A 141
530 FOR A=13 TO 1 STEP -2: FOR B=0 TO
7+A STEP 1: SOUND 0,(B*7)+30: LOCATE
B,4: PRINT " "+MID$(B$,A,2): NEXT B:
NEXT A: SOUND 8,0 158
540 FOR A=8211 TO 8213: VPOKE A,65: N
EXT A 72
550 LOCATE 3,7: PRINT "COPYRIGHT © 19
87 MICROWARE" 169
560 LOCATE 20,9: PRINT "R.VAN HOUT" 22
570 LOCATE 7,13: PRINT USING "HIGH SC
ORE **####";HS 48
580 LOCATE 0,17: PRINT "#?????????????
????????????????????#?
$$$ ENERGY L
EVEL 01 $$
$$$ :;<=>? SCORE $$$
8"; 166
590 PRINT "!.....
!.....": VPOKE 6911,33: LOCATE 9,19:

```

```

PRINT USING "\          \";E$(96): LO
CATE 20,21: PRINT USING "***#####" ;SC      15
600 FOR X=95 TO 1 STEP -1: LOCATE 9,1
9: PRINT USING"\          \";E$(X): I
F STRIG(0) THEN 680      169
610 NEXT X      100
620 FOR X=2 TO 96 STEP 1: LOCATE 9,19
: PRINT USING"\          \";E$(X): IF
STRIG(0) THEN 680      70
630 NEXT X      104
640 GOTO 600      59
650 '      0
660 ' spel hoofd lus *****      0
670 '      0
680 FOR A=0 TO 15: LOCATE 0,A: PRINT
SPC(32);: NEXT A: NT=0: SC=0: A=USR(0)
): LOCATE 20,21: PRINT "0000000"      127
690 LOCATE 0,0: PRINT "#";STRING$(30,
"%");"#&";SPC(30);"!";STRING$(30, "!"
);"! "      62
700 SPRITE$(10)=CHR$(&B1)      142
710 SPRITE$(11)=CHR$(&B10)+CHR$(&B111
)+CHR$(&B10)      47
720 FOR A=10 TO 14 STEP 2: PUT SPRITE
A,(RND(1)*255,(A-10)*22+24),10,10: N
EXT A      226
730 FOR A=11 TO 15 STEP 2: PUT SPRITE
A,(RND(1)*255,(A-10)*21+24),11,11: N
EXT A      80
740 FOR A= 16 TO 25: PUT SPRITE A,(RN
D(1)*255,(A-16)*11+29),8,10: NEXT A      27
750 FOR A=26 TO 30: PUT SPRITE A,(RND
(1)*255,(A-26)*22+23),8,10: NEXT A      97
760 C$=SPACE$(7): D$=SPACE$(30)+"welc
ome to KEYBOARD BLASTER"+C$+"written
by r. van hout"+C$+"type away the wor
ds before they reach the laser beam"+
C$+"good luck..." +C$+"high score"+STR
$(HS)+" by "+N$+" "      170
770 A$="KEYBOARD BLASTER": B$="keyboa
rd blaster"      239
780 SOUND 13,14: SOUND 0,200: FOR A=1
TO LEN(D$): LOCATE 1,1: PRINT MID$(D
$,A,30): SOUND 8,14: FOR W=0 TO 70: N
EXT W: SOUND 8,11: FOR W=0 TO 30: NEX
T W: SWAP A$,B$: MID$(D$,42,16)=A$      213
790 IF X<96 THEN X=X+1: LOCATE 9,19:
PRINT USING"\          \";E$(X) ELSE
LOCATE 9,19: PRINT E$(96)      230
800 NEXT A      126
810 SOUND 8,0      114
820 KEY(1) ON: KEY(2) ON      198
830 LOCATE 0,0: PRINT "#";STRING$(30,
"%");"#&";SPC(30);"$#";STRING$(30, "%
");"# "      230
840 V=0: W=34: S1=1: S2=2: PUT SPRITE
1,(-16,23),15,2: FOR A=2 TO 6: PUT S
PRITE A,(-16,(A-1)*16+23),8,3: NEXT A
: PUT SPRITE 7,(-16,119),15,1: GOTO 8

```

```

80      227
850 IF NT=10 THEN NT=0 ELSE GOTO 930      42
860 IF E<=92 THEN E=E+4 ELSE E=96      31
870 GOSUB 1180      81
880 IF L<14 THEN L=L+1: SOUND 12,16-L
: V=V+2: W=W-2: LOCATE 28,19: PRINT U
SING "##";L ELSE GOTO 930      66
890 IF L<10 THEN LOCATE 28,19: PRINT
"0"      206
900 SOUND 0,200: SOUND 7,56: FOR A=(V
+1)*8 TO (V+3)*8: SOUND 8,11+S1*2: VP
OKE 6917,A: VPOKE 6941,A: SPRITE$(3)=
S$(S1): SWAP S1,S2: FOR B=2 TO 6: VPO
KE 6913+(B*4),A: NEXT B      184
910 OUT &HAA,INP(&HAA) XOR &H40      167
920 FOR C=10 TO 20: D=6913+C*4: VPOKE
D,255-(CMOD3+257-VPEEK(D))MOD255: NE
XT C: NEXT A: SOUND 8,0      61
930 I=INT(100*RND(-TIME)+1): H=INT(9*
RND(1)+5): X=USR(0)      77
940 S=33: N=LEN(A$(I)): T=34      4
950 A$=SPACE$(33)+A$(I)      90
960 FOR J=1 TO 33+N      155
970 OUT &HAA,INP(&HAA) XOR &H40      179
980 LOCATE 33,H: PRINT MID$(A$,J,33)      250
990 FOR C=10 TO 20: D=6913+C*4: VPOKE
D,255-(CMOD3+257-VPEEK(D))MOD255: NE
XT C      141
1000 SPRITE$(3)=S$(S1): SWAP S1,S2      168
1010 IF T=LEN(A$) THEN NT=NT+1: GOTO
850      196
1020 POKE &HFCAB,0: I$=INKEY$:IF I$=M
ID$(A$,T,1) THEN MID$(A$,T)=" ": T=T+
1: S=S+1: SC=SC+2*L+5: GOSUB 1180 ELS
E IF I$>" " THEN E=E-1: GOSUB 1180      69
1030 IF J-(S-34)=W THEN J=33+N      40
1040 NEXT J      102
1050 VPOKE (H+1)*32+6144+V,VPEEK((H+1
)*32+6144+V)-32      168
1060 SOUND 8,14: SOUND 7,48: E=E-2: I
F E>-1 THEN SOUND 6,E/3.1: SOUND 0,E*
2.65      163
1070 OUT &HAA,INP(&HAA) XOR &H40      249
1080 GOSUB 1180      158
1090 FOR C=10 TO 20: D=6913+C*4: VPOK
E D,255-(CMOD3+257-VPEEK(D))MOD255: N
EXT C      114
1100 SPRITE$(3)=S$(S1): SWAP S1,S2      170
1110 MID$(A$,T)=" ": T=T+1: S=S+1      166
1120 LOCATE V,H+1: PRINT MID$(A$,T,32
-V): VPOKE (H+1)*32+6144+V,VPEEK((H+1
)*32+6144+V)-32      232
1130 IF T=LEN(A$) THEN SOUND 8,0: NT=
NT+1: GOTO 850      196
1140 GOTO 1060      194
1150 '      0
1160 ' plaatsen score & energie *****      0
1170 '      0
1180 IF E<=0 THEN E=0      224

```



```

1190 LOCATE 9,19: PRINT USING "\
    \";E$(E): LOCATE 20,21: PRINT US
ING "***####";SC
1200 IF E=0 THEN FOR A=2 TO 6: PUT SP
RITE A,(200,200): NEXT A: SOUND 8,0:
LOCATE 0,H+1: PRINT SPC(32): FOR W=0
TO 50:NEXT W: RETURN 1250
1210 RETURN
1220 '
1230 ' einde spel *****
1240 '
1250 KEY(1) OFF: KEY(2) OFF: VPOKE 82
19,0: FOR A=3 TO 16: LOCATE 0,A: PRIN
T STRING$(32,219): NEXT A
1260 SOUND 7,21: SOUND 6,31: SOUND 8,
16: SOUND 9,0: SOUND 12,100: SOUND 13
,1
1270 FOR A=1 TO 40: VPOKE 8219,102: F
OR W=0 TO 90: NEXT W: VPOKE 8219,0: F
OR W=0 TO 90:NEXT W:NEXT A:FOR W=0 TO
100: NEXT W
1280 SOUND 8,0: SOUND 9,16: SOUND 7,5
6: SOUND 12,5
1290 A=USR(0): FOR W=0 TO 1000: NEXT
W
1300 FOR A=0 TO 31: PUT SPRITE A,(0,0
),0,0: NEXT A
1310 IF NOT SC>HS THEN 1390
1320 HS=SC: LOCATE 9,7: PRINT "NEW HI
GH SCORE"
1330 LOCATE 6,12: PRINT "NAME .....
.....": LOCATE 12,12
1340 N$="": FOR A=1 TO 14
1350 POKE &HFCAB,255: A$=INPUT$(1): I
F A$=CHR$(13) THEN 1370 ELSE IF A$=CH
R$(8) AND A>1 THEN A=A-1: N$=LEFT$(N$
,A-1): LOCATE A+11,12: PRINT ".":CHR$(
8);: GOTO 1350 ELSE IF A$<" " AND A
$<"A" OR A$>"Z" THEN 1350
1360 PRINT A$;: SOUND 13,1: FOR W=0 T
O 120: NEXT W: N$=N$+A$: NEXT A
1370 IF N$>" " THEN FOR A=1 TO LEN(N$)
: IF ASC(MID$(N$,A,1))>64 THEN MID$(N
$,A,1)=CHR$(ASC(MID$(N$,A,1))+32): NE
XT A ELSE NEXT A
1380 IF N$="" THEN N$="a lazy person"
1390 FOR A=3 TO 16: LOCATE 0,A: PRINT
SPC(32): NEXT A
1400 LOCATE 9,7: PRINT "#%????????????
#": LOCATE 9,8: PRINT "$ $
": LOCATE 9,9: PRINT "$ GAME OVER $"
: LOCATE 9,10: PRINT "& &"
: LOCATE 9,11: PRINT "!!!!!!!!!!!!!"
1410 FOR W=0 TO 3500: NEXT W: GOTO 48
0
1420 GOTO 1420
1430 '
1440 ' pause *****
1450 '

```

85  
27  
130  
0  
0  
0  
54  
61  
13  
175  
51  
192  
102  
80  
53  
117  
89  
183  
251  
58  
212  
25  
153  
190  
0  
0  
0

```

1460 SOUND 8,0: SOUND 13,0: FOR A=3 T
O 16: LOCATE 0,A: PRINT SPC(32): NEXT
A: LOCATE 13,8: PRINT "PAUSED": LOCA
TE 4,10: PRINT "PRESS RETURN TO CONTI
NUE": KEY(1) OFF
1470 IF INKEY$<>CHR$(13) THEN 1470
1480 LOCATE 13,8: PRINT SPC(6): LOCAT
E 4,10: PRINT SPC(24): X=USR(0): KEY(
1) ON: SOUND 13,14: RETURN
1490 '
1500 ' stop *****
1510 '
1520 KEY(1) OFF: KEY(2) OFF: SOUND 8,
0: SOUND 13,0: SOUND 0,56: FOR A=0 TO
31: PUT SPRITE A,(0,0),0,0: NEXT A:
RETURN 480
1530 '
1540 ' data lettertype *****
1550 '
1560 DATA 60,102,102,0,204,204,120,0
1570 DATA 4,6,6,0,12,12,8,0
1580 DATA 60,6,6,56,192,192,120,0
1590 DATA 60,6,6,56,12,12,120,0
1600 DATA 36,102,102,56,12,12,8,0
1610 DATA 60,96,96,56,12,12,120,0
1620 DATA 60,96,96,56,204,204,120,0
1630 DATA 60,6,6,0,12,12,8,0
1640 DATA 60,102,102,56,204,204,120,0
1650 DATA 60,102,102,56,12,12,120,0
1660 DATA 226,246,254,254,234,226,226
,0
1670 DATA 255,0,204,218,216,218,204,0
1680 DATA 255,0,227,214,230,214,211,0
1690 DATA 255,0,52,181,189,189,53,0
1700 DATA 255,0,206,173,174,237,173,0
1710 DATA 248,0,120,96,112,96,120,0
1720 DATA 126,193,221,217,221,193,126
,0
1730 DATA 16,56,56,92,92,238,238,0
1740 DATA 236,238,236,224,236,238,236
,0
1750 DATA 108,238,224,224,224,238,108
,0
1760 DATA 236,238,238,238,238,238,236
,0
1770 DATA 254,254,0,252,224,254,254,0
1780 DATA 254,254,0,252,224,224,224,0
1790 DATA 108,238,224,224,238,230,108
,0
1800 DATA 238,238,238,254,238,238,238
,0
1810 DATA 56,56,56,56,56,56,56,0
1820 DATA 14,14,14,14,238,238,108,0
1830 DATA 238,236,232,224,232,236,238
,0
1840 DATA 224,224,224,224,226,230,238
,0
1850 DATA 130,198,238,238,238,238,238
,0

```

30  
205  
0  
0  
0  
201  
0  
0  
109  
199  
20  
62  
201  
125  
7  
17  
80  
0  
21  
7  
218  
45  
42  
236  
70  
123  
169  
190  
143  
144  
211  
82  
175  
146  
140  
224  
25  
208



## Kategorie: klokken

# Linklk

Een tijdje geleden hebben we in MSX Computer Magazine eens een oproep geplaatst om ons klok-programma's toe te sturen. Programma's dus die op de een of andere manier een klok op het beeldscherm toveren.

Nu, dat hebben we geweten. Het waren er weliswaar niet zoveel, die klokken die de PTT bij ons in de bus deponeerde, maar er zaten wel hele vreemde exemplaren bij. Sommige daarvan willen we u dan ook niet onthouden, zoals deze Linklk, die door N. Janssen uit Breda ontworpen is.

Linklk is qua programmering tamelijk simpel, het is puur het idee dat het programmaatje zo aantrekkelijk maakt. Er ge-

beuren namelijk wel erg aardige zaken op het scherm, maar met het aangeven van de tijd heeft het allemaal weinig te

maken. Hoewel die er natuurlijk ook op staat.

### Lijnen

Linklk beeldt het voortschrijden van de tijd namelijk uit door een boeiend lijnenspel op het beeldscherm te toveren. Daarbij begeleid het programma zijn grafische grappen met random-geluiden. Niet fraai, maar ze trekken wel de aandacht!

Dat we daarnaast de tijd - en de datum - te zien krijgen lijkt wel bijzaak.

Het geheel maakt in ieder geval wel een heel aparte klok, waar geen enkele huis-tuin-en-keuken kwarts-klok tegenop kan boksen. Want hoewel bij klokken veel aandacht aan de vormgeving geschonken wordt - meestal tenminste, maar over smaak valt niet te twisten - zijn die gewone klokken natuurlijk

zwaar in het nadeel vergeleken met een computer. Op een beeldscherm kan nu eenmaal heel wat makkelijker gegoocheld worden met vormen en kleuren dan met plastic mogelijk is.

### Opvallend

Een van de meest opvallende eigenschappen van dit programma is de hoofdlus, in regel 240. Inderdaad, die hoofdlus doet helemaal niets. Dat hoeft ook niet, want Linklk handelt het hele tijdgebeuren af middels een ON INTERVAL.

Met andere woorden, Linklk staat voor het grootste gedeelte van de tijd alleen maar te wachten tot er weer een hele seconde verstreken is. En zo'n seconde is lang voor een computer!

Wie zich groepen voelt kan Linklk best nog een andere taak laten uitvoeren in die tijd.

```

10 REM linklk
20 REM
30 REM MSX Computer Magazine
40 REM
50 REM ingezonden door N. Janssen, Breda
60 REM
70 ' Initialisatie *****
80 KEY OFF:COLOR 10,1,1
90 CLS:SCREEN ,0
100 INPUT "dag: ";DAY
110 INPUT "maand: ";MTH
120 INPUT "jaar: ";YEA
130 INPUT "uren: ";HRS
140 INPUT "minuten: ";MIN
150 INPUT "sekonden: ";SEC
160 TIME=50*SEC
170 CLS:COLOR 10,1,1
180 OPEN "grp:" AS #1
190 SCREEN 2,0
200 X=1:C=2
210 ON INTERVAL=50 GOSUB 250
220 INTERVAL ON
230 ' Hoofdlus die niets doet! *****
240 GOTO 240
250 ' Timesubroutine *****
260 INTERVAL OFF
270 LOCATE 0,0,0:' cursor uit
280 ' Tijd- en datum-logika *****
290 IF HRS>=24 THEN HRS=0:LINE (50,180)-(115,188),1,BF:LINE (180,180)-(200,188),1,BF:DAY=DAY+1
300 IF DAY>=32 THEN DAY=1:MTH=MTH+1
310 IF MTH>=13 THEN MTH=1:YEA=YEA+1
320 IF MTH=2 AND DAY>=29 THEN DAY=1:MTH=MTH+1:' februari
330 IF MTH=4 AND DAY>=31 THEN DAY=1:MTH=MTH+1:' april
340 IF MTH=6 AND DAY>=31 THEN DAY=1:MTH=MTH+1:' juni
350 IF MTH=9 AND DAY>=31 THEN DAY=1:MTH=MTH+1:' september
360 IF MTH=11 AND DAY>=31 THEN DAY=1:MTH=MTH+1:' november

```

```

370 IF TIME>=3000 THEN MIN=MIN+TIME\3
000:TIME=TIME MOD 3000:LINE (205,180)-(222,188),1,BF
380 IF MIN>=60 THEN HRS=HRS+1:MIN=MIN MOD 60:LINE(180,180)-(200,188),1,BF
390 ' Lijnenspel *****
400 LINE (130,X)-(X,65),C
410 X=X+4
420 LINE (230,180)-(250,188),1,BF
430 DRAW "bm 5,180"
440 ' Print datum en tijd *****
450 PRINT #1, USING " date:##:##:##";DAY;MTH;YEA
460 DRAW "bm135,180"
470 PRINT #1, USING " time:##:##:##";HRS;MIN;TIME\50:GOSUB 560
480 ' Koordinaten en kleur bijwerken
490 IF X>=239 THEN X=1
500 IF C>=14 THEN C=2
510 IF MIN MOD 60=0 THEN PLAY "T255L64N"+STR$(INT(RND(1)*90))
520 INTERVAL ON
530 RETURN
540 END
550 ' Kleur op 1 stellen op de oneven minuten *****
560 IF MIN/2=INT(MIN/2) THEN C=C+1 ELSE C=1
570 RETURN

```

## CATEGORIE: SPELLEN

# Jake

Het lijkt wel of de spellen in MCM met de keer beter worden. Althans dat is onze onbescheiden mening, maar we zijn natuurlijk wel wat bevooroordeeld.

Jake in the Caves is echter werkelijk een juweeltje van een aktiespel, dat fraaie graphics, vloeiende beweging, snelle actie en een volkomen belachelijk spelplot aan elkaar weet te koppelen.

Want zeg nou zelf, wie komt er in vredesnaam op het idee om de spelfiguur *kuikens* te laten verzamelen, op een hele konstruktie van platforms en ladders nog wel? Bovendien, die kuikens, die komen natuurlijk uit eieren. Die eieren worden door een soort van kamikaze-eendjes gelegd, in volle vlucht.

### Dodelijk

Goed, de kuikens moeten worden verzameld, maar de eieren - zolang ze vallen - zijn puur vergif voor Jake. Net zoals de eendjes, de rondrennende honden - hoewel, eigenlijk alleen maar hondekoppen - en natuurlijk de gifgroene driehoekjes die zo hier en daar in de platforms zitten.

Gelukkig maar dat Jake niet doodvalt wanneer hij van een platform afstapt, anders was de strijd helemaal ongelijk!

Als we ons dan ook nog even voorstellen dat deze waanzin zich op maar liefst zes schermen afspeelt, dan krijgt u misschien een beetje een indruk van wat 'Jake' te bieden heeft.

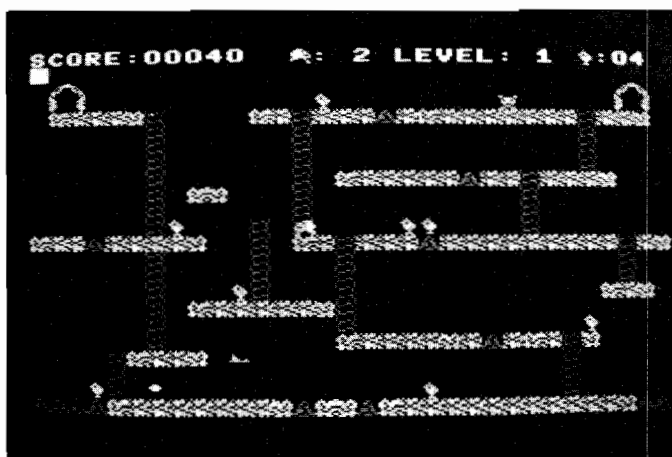
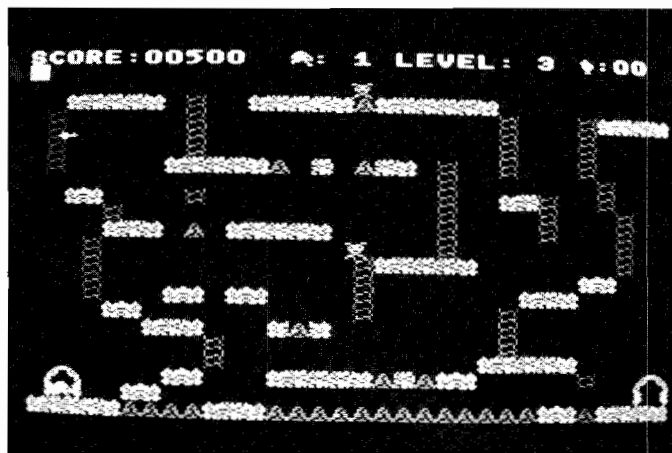
Een goed advies: gebruik een stevige joystick. Want met de cursor-toetsen - die natuurlijk ook gebruikt kunnen worden - is de strijd wel erg zwaar.



### Spelregels

De spelregels zijn simpel: men moet op ieder veld een aantal kuikentjes verzamelen alvorens men naar het volgende

scherm door mag. Op scherm 1 zijn dat er nog maar 5, op scherm 2 worden dat er acht enzovoort. Voor ieder volgens scherm moeten er steeds weer drie extra kuikentjes worden



gered voor men een niveau hoger kan, op het laatste scherm - scherm 6 - zijn het er al 20. Als extra handicap moeten ook alle kuikens verzameld worden, voor men verder kan. Er kunnen dus meer dan 5 kuikens op het eerste scherm verzameld worden, hetgeen dan ook meer punten oplevert.

Per kuiken krijgt de speler namelijk 10 punten toegekend, terwijl er per scherm een bonus van schermnummer maal 100 te verdienen valt. De hoogste score wordt natuurlijk bijgehouden, zodat men kan proberen elkaar of zichzelf te overtreffen.

**Aardig**

De aardigheid bij dit spel zit hem vooral in de logische problemen, die de verschillende schermen opleveren. De eerste paar schermen zijn nog vrij rechttoe-rechtaan, maar op de latiger schermen kan men zichzelf vastzetten. Als men bijvoorbeeld door een wat ongelukkige sprong op een 'eilandje' terechtgekomen is rest er nog maar een uitweg: zelfmoord.

Een heel prettige bijkomstigheid is dat men de uitstekende muziek - Hello Dolly - kan uitzetten door de 2-toets even in te tikken. Hoewel die muziek namelijk prima klinkt is ze wel uiterst irritant na een tijdje. Dat is geen kritiek overigens, naar onze mening moet de muziek bij een spel als Jake inderdaad als extra afleiding dienen. Dat maakt het alleen maar lastiger!

**Werking**

'Jake in the Caves' is weer eens een typisch voorbeeld van wat een slimme programmeur - zijn naam is overigens Steven van der Loef - met een combinatie van Basic en machinetaal kan bereiken. Het spel zelf bestaat uit pure ML, de schermopbouw en allerlei andere zaken zijn in Basic gedaan. Een slim gebruik van schermtype 1 is de sleutel van dergelijke schermen, die feitelijk gewoon middels print-kommando's worden opgebouwd. Kijk maar eens in de regels vanaf 2570, daar staan de zes schermen keurig onder elkaar.

Om te stoppen met spelen kan op ieder moment de Control-Stop toetscombinatie ingedrukt worden. In dat geval zal de computer echter wel een reset uitvoeren, aangezien Jake op MSX2 computers met diskdrives anders allerlei hele vervelende problemen oplevert. Daar kwamen we op de redactie ook door schade en schande achter, het heeft ons een van onze diskettes met allerlei programmatuur die nog in ontwikkeling is gekost. Die disk lag namelijk zo overhoop dat we hem alleen nog opnieuw konden formatteren. Gelukkig hadden we een vrij recente backup...

**Extra controle**

Ook in Jake hebben we - gezien de grote lappen data-regels - onze extra controle ingebouwd. Voor zowel de spelmachinecode, vanaf regel 4410, als voor de muziek-ML, regel 4700 en verder, wordt een extra check uitgevoerd. Daardoor kan het gebeuren dat, als u het programma voor de eerste keer gaat proberen, er opeens een melding verschijnt zoals:  
fout in dataregels 4410-4550.

Daarna zal de machine zich resetten, wegens de al eerder vermelde problemen met MSX2. Het is dus zaak om uw programma eerst weg te schrijven, voordat u het gaat uitproberen.

Zo'n foutmelding moet u natuurlijk wel serieus nemen, u dient dan de genoemde data-regels nog eens heel erg zorgvuldig na te kijken. Wat u zeker niet moet doen is meteen naar de telefoon grijpen om de redactie te gaan vertellen dat u zeker bent dat u geen fouten maakt. Al heeft u het tien keer nagekeken, en uw buurman zelfs al vijftien maal, die foutmelding verschijnt daar echt niet zomaar.

**Tenslotte**

Goed, het is even werk, om Jake in the Caves in te tikken, maar het spel is de moeite waard. Wij waren in ieder geval behoorlijk onder de indruk van dit spel. Hopelijk stuurt Steven van der Loef ons nog eens zoiets moois op. Of iemand anders natuurlijk....

```

10 REM JAKE
20 REM
30 REM MSX Computer Magazine
40 REM
50 REM *****
60 REM *
70 REM * JAKE IN THE CAVES *
80 REM *
90 REM * S.V.L. Software '86 *
100 REM *
110 REM *****
120 REM
130 REM INITIALISATIE
140 REM
150 SCREEN 1,3,0,2:WIDTH 32:KEY OFF:C
OLOR 15,1,1:CLER 200,&HDFFF
160 DEFINT A-Z:DEFSNG S
170 ON STOP GOSUB 880:STOP ON
180 GOSUB 1370:' intro-scherm
190 LOCATE 0,0:PRINT "MACHINECODE VOO
R MUZIEK INLEZEN "
200 GOSUB 4590
210 DEFUSR=&HE500:ML=USR(0)
220 LOCATE 0,0:PRINT " MACHINECODE V
OOR SPEL INLEZEN "
230 GOSUB 4300
240 GOSUB 1620:' SPELREGELS
250 SCREEN 1,0
260 LOCATE 2,11:PRINT "KARAKTERS EN S
PRITES INLEZEN "
270 GOSUB 2070
280 HIS$="00000"
290 GOSUB 1770:' CURSOR OF JOYSTICK
300 T=RND(-TIME):K=1:LV=1:J=3:SC$="00
000"
310 REM
320 REM PRINT HET SCHERM
330 REM
340 ON K GOSUB 2600,2880,3170,3440,37
20,4010
350 POKE &HF000,0:POKE &HF001,0:FOR I
=&HF003 TO &HF00B:POKE I,0:NEXT I
360 POKE &HF00C,4:POKE &HF00D,4:POKE
&HF00E,4
370 POKE &HF007,K+K*3-(K-1)
380 GOSUB 1260
390 LOCATE 29,0:PRINT "00"
400 LOCATE 0,0:PRINT USING "SCORE:\
 \ X:## LEVEL:## p:";SC$;J;LV
410 PUTSPRITE 0,(X,Y),7,13
420 FOR P=1 TO 500:NEXT P
430 A=PEEK(&HF014):POKE &HF014,0
440 FOR I=12 TO 10 STEP-1
450 VPOKE 6914,I
460 FOR P=1 TO 30
470 SOUND 0,P*4+I*10:SOUND 1,0:SOUND
8,14
480 NEXT P,I
490 VPOKE 6914,0
500 FOR P=1 TO 30
510 SOUND 0,P*4+90
520 NEXT P
530 POKE &HF014,A:SOUND 8,0
540 PUTSPRITE 1,(254,72),14,6
550 REM
560 REM BEGIN VAN HET SPEL
570 REM
580 DEFUSR0=&HE019
590 ML=USR0(0)
600 IFVPEEK(6917)=255THENVPOKE 6916,(
8+INT(RND(1)*150))AND248
610 POKE&HF000,RND(1)*20
620 ONPEEK(&HF001)GOTO670,950
630 GOTO590
640 REM
650 REM DOOD

```

```

660 REM 0
670 VPOKE 6915,6:VPOKE 6916,208:VPOKE 161
6920,209 42
680 GOSUB 1150
690 IF PEEK(&HF004)=255 THEN POKE &HF 128
009,PEEK(&HF009)-1 38
700 POKE &HF001,0:POKE &HF004,0 230
710 GOSUB 1130:J=J-1:IF J>=0 THEN 400
720 REM 0
730 REM EINDE VAN HET SPEL 0
740 REM 0
750 IF SC$>HI$ THEN HI$=SC$ 151
760 CLS 15
770 LOCATE 0,4:PRINT "UW SCORE:":SC$ 176
780 LOCATE 0,6:PRINT "DE HOOGSTE SCOR 45
E:":HI$
790 LOCATE 0,10:PRINT "WILT U NOG EEN 204
KEER SPELEN?" 250
800 LOCATE 0,12:PRINT "UW ANTWOORD:": 202
810 I$=INPUT$(1) 208
820 IF I$="j" OR I$="J" THEN 290 171
830 IF I$<>"n" AND I$<>"N" THEN 810 0
840 REM 0
850 REM STOPPEN 0
860 REM 0
870 REM ER MOET EEN RESET WORDEN GEDA 0
AN VOOR MSX2 MET DISKDRIVES!
880 SCREEN0:WIDTH37:CLS 172
890 PRINT"Er moet een reset gedaan wo 135
rden in verband met mogelijke di sk- 149
problemen op MSX2!" 236
900 FOR N=1 TO 2500:NEXT N 0
910 DEFUSR=0:A=USR(0) 0
920 REM 0
930 REM VOLGENDE LEVEL 0
940 REM 0
950 VPOKE 6916,208 178
960 GOSUB 1130 15
970 SC=SC+LV*100 32
980 SC$=MID$(STR$(SC),2):SC$=RIGHT$(" 32
00000",5-LEN(SC$))+SC$
990 K=K+1:LV=LV+1:IF K>6 THEN K=1 157
1000 X=VPEEK(6913):Y=VPEEK(6912) 188
1010 VPOKE 6914,0 252
1020 IF X>XX THEN VX=-1 ELSE VX=1 88
1030 IF Y>YY THEN VY=-1 ELSE VY=1 174
1040 IF X=XX THEN VX=0 32
1050 IF Y=YY THEN VY=0 87
1060 VPOKE 6913,X 164
1070 VPOKE 6912,Y 169
1080 X=X+VX 187
1090 Y=Y+VY 215
1100 IF VX<>0 OR VY<>0 THEN 1040 176
1110 GOSUB 1150 97
1120 GOTO 340 20
1130 SC$="":FOR I=6150 TO 6154:SC$=SC 172
$+CHR$(VPEEK(I)):NEXT I:SC=VAL(SC$)
1140 RETURN 137
1150 A=PEEK(&HF014):POKE &HF014,0 228
1160 FOR I=10 TO 13 62
1170 VPOKE 6914,I 185
1180 FOR P=1 TO 30 191
1190 SOUND 0,P*4+I*10:SOUND 1,0:SOUND 64
8,14
1200 NEXT P,I 101
1210 POKE &HF014,A:SOUND 8,0 12
1220 RETURN 133
1230 REM 0
1240 REM ZET BEESTEN OP HET SCHERM 0
1250 REM 0
1260 READ AB:POKE &HF006,AB:AB=AB-1 28
1270 FOR I=3 TO 3+AB 49
1280 READ X,Y,C,S 13
1290 PUTSPRITEI,(X,Y),C,S 246
1300 NEXT I 85
1310 PUTSPRITEI,(0,208),0,0 102

```

```

1320 READ X,Y,XX,YY 62
1330 RETURN 138
1340 REM 0
1350 REM INTRO 0
1360 REM 0
1370 RESTORE 1530 152
1380 READ SP:IF SP=255 THEN 1420 ELSE 63
SS$="":FOR H=1 TO 32
1390 READ A$:SS$=SS$+CHR$(VAL("&H"+A$)) 220
1400 NEXT H:SPRITES(SP)=SS$ 12
1410 GOTO 1380 253
1420 PUTSPRITE 0,(80,8),7,27:PUTSPRIT 67
E 1,(80,40),7,28
1430 LOCATE 14,6:PRINT "oftware" 37
1440 LOCATE 11,9:PRINT "PRESENTS:" 229
1450 PUTSPRITE 2,(96,96),6,29:PUTSPRI 139
TE 3,(128,96),6,30
1460 LOCATE 10,16:PRINT "IN THE CAVES 55
"
1470 PUTSPRITE 4,(112,144),10,31 23
1480 LOCATE 0,23:PRINT "Copyright 86 162
By:S.V.L. Software";
1490 RETURN 158
1500 REM 0
1510 REM DATA VOOR INTRO SPRITES 0
1520 REM 0
1530 DATA 27,03,07,0C,18,18,19,1A,18, 132
18,0C,07,03,00,00,01,01,F8,FC,06,00,4
0,20,A0,A0,40,00,F0,F8,0C,C6,23,03
1540 DATA 28,01,01,01,01,01,18,0F,07, 134
00,00,00,00,00,00,00,00,03,03,03,C3,2
6,0C,F8,F0,00,00,00,00,00,00,00,00
1550 DATA 29,7F,88,08,08,08,08,08,88, 228
89,70,00,00,00,00,00,00,38,44,44,44,5
C,64,44,44,45,82,00,00,00,00,00,00
1560 DATA 30,21,51,12,14,18,14,14,12, 54
51,21,00,00,00,00,00,00,1E,21,20,20,2
2,3C,20,20,21,1E,00,00,00,00,00,00
1570 DATA 31,03,04,09,0A,0A,09,04,03, 100
00,00,00,6E,2A,2E,22,2E,C0,20,90,10,1
0,90,20,C0,00,00,00,EE,A8,EE,AA,EE
1580 DATA 255 126
1590 REM 0
1600 REM SPELREGELS 0
1610 REM 0
1620 SCREEN 0:WIDTH 40 208
1630 LOCATE 11,0:PRINT "JAKE IN THE C 170
AVES"
1640 LOCATE 0,2:PRINT "Help Jake met 117
het pakken van kuikens, maar pas op
! Er schuilen vele gevaren."
1650 PRINT "Als u een van de volgende 16
voorwerpen aanraakt dan verliest
u 1 van uw 4 levens: driehoek,
vogel, beest en het vallende ei."
1660 PRINT:PRINT "U gaat naar het vol 208
gende level als alle kuikens weg zijn
en als dat er tenminste5 in level 1,
8 in level 2, 11 in level 3, 14 in l
evel 4, 17 in level 5 of 20 inlevel 6
zijn."
1670 PRINT:PRINT "Voor een kuiken kri 214
jgt u 10 punten."
1680 PRINT "Als u een level gehaald h 236
eeft krijgt u een bonus (100 x level
)."
1690 PRINT:PRINT "Toets 1 .. muziek a 40
an:":PRINT "Toets 2 .. muziek uit."
1700 LOCATE 11,22:PRINT"DRUK EEN TOET 55
S IN"
1710 IF INKEY$<>" " THEN 1710 93
1720 IF INKEY$=" " THEN 1720 45
1730 RETURN 146
1740 REM 0
1750 REM CURSORTOETSEN OF JOYSTICK 0
1760 REM 0

```



```

2930 PRINT" i i 138
";
2940 PRINT" i giab h gab 43
";
2950 PRINT" i i 255
";
2960 PRINT" i i 2
";
2970 PRINT" gaaaaiaaab i i 213
igab";
2980 PRINT" i i 117
";
2990 PRINT" i gaaaab h g 139
ab";
3000 PRINT" i 1
";
3010 PRINT" i 4
";
3020 PRINT" i ggab 46
";
3030 PRINT" i gaaaia 142
b i";
3040 PRINT"gaab i a i 1
i gb";
3050 PRINT" gaab i i 214
";
3060 PRINT" i gaaab i 227
";
3070 PRINT" i i 236
";
3080 PRINT" i 171
";
3090 PRINT" ef i gb 65
i";
3100 PRINT" cd i gb gb 48
i";
3110 PRINT"gaaaaabhahaaabhhhhhhgaabhg
aahab";
3120 RESTORE 1940 111
3130 RETURN 211
3140 REM 136
3150 REM SCHERM 3 0
3160 REM 0
3170 CLS 216
3180 LOCATE0,3 116
3190 PRINT" gaaab i gaaabhgaaaab 50
";
3200 PRINT" i i i 100
igaab";
3210 PRINT" i i i 38
";
3220 PRINT" i i i 41
";
3230 PRINT" i gaaabh a hab i i 29
";
3240 PRINT" i i 52
";
3250 PRINT" gb i i gai 17
";
3260 PRINT" i i i 160
";
3270 PRINT" gab h gaaab i i 46
";
3280 PRINT" i i 208
";
3290 PRINT" i igaaab 217
";
3300 PRINT" i i 6
gb";
3310 PRINT" i gb gb i ga 138
b";
3320 PRINT" gb i i 113
";
3330 PRINT" gab ghb i 3
";
3340 PRINT" i i

```

```

";
3350 PRINT" i gaaa 166
b";
3360 PRINT" ef gb gaaabhahgb 84
i ef";
3370 PRINT" cd gb 55
cd";
3380 PRINT"gaaabhhhhgabhhhhhhhhhhhhhg
bhgaab";
3390 RESTORE 1960 247
3400 RETURN 12
3410 REM 133
3420 REM SCHERM 4 0
3430 REM 0
3440 CLS 213
3450 LOCATE0,2 99
3460 PRINT"ef
";
3470 PRINT"cd 128
";
3480 PRINT"gaaab 81
";
3490 PRINT" gaabhaha gaingahabhg 218
ab";
3500 PRINT" i 60
";
3510 PRINT" i i 156
";
3520 PRINT" gib i i 227
";
3530 PRINT" i gb i gb i 93
";
3540 PRINT" i i i i 145
";
3550 PRINT" i gb gb 18
gab";
3560 PRINT" i i gb 150
";
3570 PRINT" gb gb gb i gb 165
";
3580 PRINT" i i i 2
";
3590 PRINT" gaab gb gb i 69
";
3600 PRINT" i ga 27
bhgbi";
3610 PRINT" i i i 146
";
3620 PRINT" i i i gb i 7
";
3630 PRINT" i i i i i 142
iaab";
3640 PRINT" i i i i i 71
i ef";
3650 PRINT"i i i gb i i i 110
i cd";
3660 PRINT"hhgbhhgabhhhhgaaabhhhgabh
hgaaab";
3670 RESTORE 1980 94
3680 RETURN 44
3690 REM 161
3700 REM SCHERM 5 0
3710 REM 0
3720 CLS 213
3730 LOCATE0,1 85
3740 PRINT"
ef";
3750 PRINT" 202
cd";
3760 PRINT" gab i i 35
gaab";
3770 PRINT" ihh i 217
";
3780 PRINT" i i gab ga 80
b";
3790 PRINT" gab gab i 147

```



```

i ";
3800 PRINT" i gab gb i 122
i ";
3810 PRINT" i gb i 144
i ";
3820 PRINT" i hh i a 42
";
3830 PRINT" gab i i 225
a ";
3840 PRINT" i i 218
a ";
3850 PRINT" i giaabhahahgaab h 47
a ";
3860 PRINT" i i 41
i ";
3870 PRINT" hah i i 106
i ";
3880 PRINT" i i i a 10
i ";
3890 PRINT" i i gaab a ih 88
hgab ";
3900 PRINT" i i a i 118
";
3910 PRINT" i i hh a i 11
";
3920 PRINT" gib gai i 46
";
3930 PRINT" i gb i 30
gb ef";
3940 PRINT" gaab 146
i cd";
3950 PRINT" hhahgbhh hngabhhgaabhh 252
hhgaab";
3960 RESTORE 2000 140
3970 RETURN 54
3980 REM 164
3990 REM SCHERM 6 0
4000 REM 0
4010 CLS 197
4020 LOCATE0,1 69
4030 PRINT" 186
ef";
4040 PRINT" i i 159
cd";
4050 PRINT" i gaabhahgaib i gaaab 249
aab";
4060 PRINT" gab i i 170
i ";
4070 PRINT" i i i a 38
gb i ";
4080 PRINT" i i i a a 127
i ";
4090 PRINT" i gib gab i i 138
i ";
4100 PRINT" i gb i gb i 95
i ";
4110 PRINT" i i i i i i 245
i ";
4120 PRINT" i i gib i i i i 144
i ";
4130 PRINT" i i i i i i 142
i ";
4140 PRINT" i gb i gb i h 30
i ";
4150 PRINT" i gaab 199
";
4160 PRINT" gaaab i a 157
gaab ";
4170 PRINT" i i gb i i 126
";
4180 PRINT" i i i i 167
";
4190 PRINT" a i i i i 239
";
4200 PRINT" i a ghab i gab i 224
";

```

```

4210 PRINT" i i 233
";
4220 PRINT" i a gb 188
ef";
4230 PRINT" gb a gab gaab i a 50
a cd";
4240 PRINT" hhhhhhahh hhh hghhhgbhhh 23
hhgaab"; 70
4250 RESTORE 2020 148
4260 RETURN 0
4270 REM 0
4280 REM MACHINECODE SPEL 0
4290 REM 0
4300 RESTORE 4410 129
4310 SH=0 15
4320 FOR I=&HE000 TO &HE485 147
4330 READ AS 137
4340 SH=SH+VAL("&H"+AS) 133
4350 POKE I,VAL("&H"+AS) 217
4360 NEXT I 106
4370 IF SH=132741! THEN RETURN 166
4380 SCREEN0:PRINT"FOUT IN DATAREGELS 120
4410-4550"
4390 FOR N=1 TO 3000:NEXT N 156
4400 GOTO 880 126
4410 DATA 21,80,01,06,2B,C5,06,08,CD, 222
4A,00,4F,CB,2F,B1,CD,4D,00,23,10,F3,C
1,10,ED,C9,CD,35,E0,CD,8E,E0,CD,6C,E1
,CD,74,E0,CD,54,E0,CD,0F,E2,CD,CE,E2,
CD,4B,E4,CD,43,E3,C9,3A,02,F0,CD,D5,0
0,FE,00,C8,32,0B,F0,FE,01,28,55,FE,03
,CA,F0,E0,FE,05,28,76,FE,07
4420 DATA CA,39,E1,C9,CD,AA,E1,FE,69, 94
C8,CD,D1,E1,FE,69,C8,FE,20,C8,3A,03,F
0,FE,00,C0,CD,99,E1,C6,04,E6,F8,CD,4D
,00,C9,CD,AA,E1,FE,69,C8,CD,D1,E1,FE,
20,C0,3A,03,F0,FE,00,C0,CD,99,E1,3C,C
D,4D,00,C9,3A,02,F0,CD,DB,00,FE,FF,CA
,55,E1,C9,3A,03,F0,FE,00,C0
4430 DATA CD,D1,E1,FE,69,28,06,CD,AA,
E1,FE,69,C0,CD,AA,E1,FE,60,C8,FE,61,C
8,FE,62,C8,CD,99,E1,3D,FE,00,C8,CD,4D
,00,C9,3A,03,F0,FE,00,C4,94,E1,CD,AA,
E1,FE,69,28,06,CD,D1,E1,FE,69,C0,CD,D
1,E1,FE,60,C8,FE,61,C8,FE,62,C8,CD,99
,E1,3C,FE,B7,C8,CD,4D,00,C9 182
4440 DATA CD,A0,E1,3C,FE,F7,28,17,CD,
4D,00,CD,A5,E1,FE,03,38,0D,3E,00,21,0
2,1B,CD,4D,00,3E,04,32,0C,F0,3A,0C,F0
,3D,32,0C,F0,FE,00,C0,3E,04,32,0C,F0,
CD,A5,E1,3C,FE,06,28,09,FE,03,20,0B,A
F,CD,4D,00,C9,3E,03,CD,4D,00,C9,CD,4D
,00,C9,CD,A0,E1,3D,FE,00,28 60
4450 DATA CE,CD,4D,00,CD,A5,E1,FE,03,
30,C4,3E,03,21,02,1B,CD,4D,00,18,BA,3
A,03,F0,FE,00,C0,CD,D1,E1,FE,20,C8,3E
,FF,32,03,F0,3E,0A,32,0A,F0,C9,3A,03,
F0,FE,00,C8,3A,0A,F0,3D,0A,F0,FE,0
0,28,17,CD,AA,E1,FE,69,28,04,FE,20,20
,0C,CD,99,E1,3D,FE,08,28,04 240
4460 DATA CD,4D,00,C9,AF,32,03,F0,C9,
21,00,1B,CD,4A,00,C9,21,01,1B,18,F7,2
1,02,1B,18,F2,CD,A0,E1,C6,04,CB,3F,CB
,3F,CB,3F,4F,CD,99,E1,CB,3F,CB,3F,CB,
3F,47,21,00,18,11,20,00,19,10,FD,16,0
0,59,19,CD,4A,00,C9,CD,AA,E1,11,20,00
,19,CD,4A,00,C9,21,09,1B,CD 87
4470 DATA 4A,00,CB,3F,CB,3F,CB,3F,4F,
2B,CD,4A,00,CB,3F,CB,3F,CB,3F,3C,18,C
9,CD,4A,00,C6,04,CB,3F,CB,3F,CB,3F,4F
,2B,CD,4A,00,CB,3F,CB,3F,CB,3F,3C,18,
B0,21,05,1B,CD,4A,00,3C,CD,4D,00,3A,0
D,F0,3D,32,0D,F0,FE,00,C0,3E,04,32,0D
,F0,21,06,1B,CD,4A,00,3C,FE 185
4480 DATA 0A,20,02,3E,06,CD,4D,00,3A,
04,F0,FE,00,20,40,3A,00,F0,FE,00,C0,2

```

```

1,04,1B,CD,4A,00,C6,04,4F,23,CD,4A,00
,FE,18,D8,FE,E8,D0,E6,F8,21,09,1B,CD,
4D,00,2B,79,CD,4D,00,23,23,3E,0E,CD,4
D,00,23,3E,0F,CD,4D,00,3E,FF,32,04,F0
,3A,09,F0,3C,32,09,F0,C9,21
204 4490 DATA 08,1B,CD,4A,00,3C,CD,4D,00,
CD,DC,E1,FE,68,28,0B,FE,60,28,07,FE,6
1,28,03,FE,62,C0,11,20,00,A7,ED,52,CD
,4A,00,FE,69,20,01,23,CD,4A,00,FE,70,
28,15,3E,70,CD,4D,00,CD,2A,E4,21,08,1
B,3E,D1,CD,4D,00,AF,32,04,F0,C9,3A,09
,F0,3D,32,09,F0,18,EA,3A,0E
160 4500 DATA F0,3D,32,0E,F0,FE,00,20,05,
3E,04,32,0E,F0,3A,06,F0,47,21,0C,1B,C
5,E5,CD,F3,E2,E1,11,04,00,19,C1,10,F3
,C9,23,23,23,CD,4A,00,FE,0B,28,1B,3E,
03,CD,4D,00,2B,2B,CD,4A,00,3C,CD,4D,0
0,23,23,E5,2B,2B,CD,F6,E1,E1,FE,20,20
,1B,3E,0B,CD,4D,00,2B,2B,CD
117 4510 DATA 4A,00,3D,CD,40,00,23,23,E5,
2B,2B,CD,F6,E1,E1,FE,20,28,CA,3A,0E,F
0,FE,01,C0,2B,CD,4A,00,EE,01,CD,4D,00
,C9,CD,D1,E1,FE,68,CA,F3,E3,FE,70,CA,
D1,E3,CD,AA,E1,FE,70,28,7A,CD,3E,01,C
B,6F,C8,CD,A0,E1,4F,CD,99,E1,57,21,04
,1B,3A,06,F0,3C,3C,47,C5,D5
143 4520 DATA E5,CD,7E,E3,E1,11,04,00,19,
D1,C1,10,F1,C9,CD,4A,00,BA,30,16,C6,0
7,BA,38,11,23,CD,4A,00,B9,30,19,C6,07
,B9,38,14,E1,D1,C1,18,59,3E,07,82,57,
CD,4A,00,BA,D0,C6,07,BA,30,E1,C9,3E,0
7,81,4F,CD,4A,00,B9,D0,C6,07,B9,30,DE
,C9,21,09,18,CD,4A,00,3C,FE
248 4530 DATA 3A,C2,4D,00,3E,30,CD,4D,00,
2B,18,EF,21,1E,18,18,EA,3E,20,CD,4D,0
0,CD,FE,E3,CD,B8,E3,CD,CC,E3,3A,08,F0
,3C,32,08,F0,47,3A,09,F0,B8,C0,3A,07,
F0,B8,00,18,07,3E,01,32,01,F0,E1,C9,3
E,02,18,F7,3E,08,1E,10,CD,93,00,3E,01
,1E,00,CD,93,00,3E,0C,1E,28
91 4540 DATA CD,93,00,3C,1E,01,CD,93,00,
1E,C8,3E,00,CD,93,00,CD,5E,E4,1D,7B,F
E,96,20,F2,C9,3E,08,1E,10,CD,93,00,3E
,0C,1E,14,CD,93,00,3E,01,1E,04,CD,93,
00,3D,1E,00,CD,93,00,3E,0D,CD,93,00,C
9,E5,21,00,02,7C,B5,2B,20,FB,E1,C9,E5
,21,00,01,18,F3,4A,41,4B,45
71 4550 DATA 20,42,79,20,53,2E,56,2E,4C,
2E,20,53,6F,66,74,77,61,72,65,00,28,6
3,29,20,43,6F,70,79,72,69,67,68,74,20
,31,39,38,36
151 4560 REM 0
4570 REM MACHINECODE VOOR MUZIEK 0
4580 REM 0
189 4590 RESTORE 4700
4600 SH=0 18
170 4610 FORI=&HE500TO&HE87F
140 4620 READ AS
136 4630 SH=SH+VAL("&H"+AS)
220 4640 POKE I,VAL("&H"+AS)
109 4650 NEXT I
218 4660 IF SH=54304! THEN RETURN
153 4670 SCREEN0:PRINT"FOUT IN DATAREGELS
4700-4810"
159 4680 FOR N=1 TO 3000:NEXT N
157 4690 GOTO 880
209 4700 DATA 21,B1,E5,22,10,F0,3E,04,32,
15,F0,32,12,F0,32,13,F0,3E,0A,32,14,F
0,01,03,00,11,9F,FD,21,22,E5,ED,B0,C9
,C3,25,E5,3A,15,F0,3D,32,15,F0,FE,00,
20,53,3E,04,32,15,F0,3A,12,F0,3D,32,1
2,F0,FE,00,20,1C,2A,10,F0,7E,FE,FF,28
,5B,32,12,F0,23,5E,3E,02,CD
4710 DATA 93,00,23,5E,3C,CD,93,00,23,
22,10,F0,3A,13,F0,3D,32,13,F0,FE,00,2
0,1C,2A,10,F0,7E,FE,FF,28,34,32,13,F0

```

```

,23,5E,3E,04,CD,93,00,23,5E,3C,CD,93,
00,23,22,10,F0,3A,14,F0,5F,3E,09,CD,9
3,00,3C,CD,93,00,3A,DA,FB,FE,FD,20,06
,3E,0A,32,14,F0,C9,FE,FB,C0
146 4720 DATA AF,18,F6,21,B1,E5,22,10,F0,
3E,01,32,12,F0,C3,25,E5,04,E2,00,04,1
D,01,10,BE,00,10,E2,00,04,E2,00,0C,1D
,01,04,1D,01,04,E2,00,04,BE,00,04,E2,
00,10,AA,00,10,1D,01,04,E2,00,0C,1D,0
1,04,1D,01,04,E2,00,04,BE,00,04,E2,00
,04,97,00,04,BE,00,04,8F,00
176 4730 DATA 04,E2,00,04,97,00,04,BE,00,
04,8F,00,04,E2,00,04,97,00,04,CA,00,0
4,8F,00,04,CA,00,04,7F,00,04,CA,00,02
,00,00,02,00,00,02,8F,00,02,D6,00,10,
BE,00,04,BE,00,04,AA,00,04,8F,00,04,A
A,00,04,00,00,04,97,00,04,CA,00,04,AA
,00,04,BE,00,04,BE,00,04,B4
125 4740 DATA 00,04,CA,00,10,AA,00,10,D6,
00,04,D6,00,0C,FE,00,04,FE,00,04,D6,0
0,04,AA,00,04,D6,00,10,A0,00,10,FE,00
,04,D6,00,0C,FE,00,04,FE,00,04,D6,00,
04,AA,00,04,D6,00,04,71,00,08,AA,00,0
4,7F,00,04,71,00,08,8F,00,04,7F,00,04
,71,00,08,97,00,04,7F,00,04
230 4750 DATA 97,00,04,BE,00,04,BE,00,04,
D6,00,04,00,00,08,E2,00,08,AA,00,04,B
E,00,04,97,00,04,CA,00,04,BE,00,04,D6
,00,04,E2,00,04,E2,00,04,D6,00,04,FE,
00,04,CA,00,04,F0,00,10,BE,00,10,E2,0
0,04,E2,00,0C,1D,01,04,1D,01,04,E2,00
,04,BE,00,04,E2,00,10,AA,00
19 4760 DATA 10,1D,01,04,E2,00,0C,1D,01,
04,1D,01,04,E2,00,04,BE,00,04,E2,00,0
4,7F,00,10,A0,00,04,8F,00,04,7F,00,04
,8F,00,04,7F,00,08,A0,00,04,8F,00,04,
7F,00,04,BE,00,04,8F,00,04,E2,00,10,A
A,00,10,D6,00,10,97,00,04,00,00,04,D6
,00,04,AA,00,04,B4,00,04,00
125 4770 DATA 00,10,AA,00,08,8F,00,04,AA,
00,08,BE,00,08,97,00,02,E2,00,02,BE,0
0,06,BE,00,06,97,00,01,8F,00,01,AA,00
,01,00,00,01,00,00,01,8F,00,01,AA,00,
01,00,00,01,00,00,03,8F,00,03,AA,00,0
1,00,00,01,00,00,04,8F,00,04,AA,00,04
,AA,00,04,CA,00,08,BE,00,08
116 4780 DATA E2,00,02,E2,00,02,BE,00,06,
BE,00,06,E2,00,01,71,00,01,8F,00,01,0
0,00,01,00,00,01,71,00,01,8F,00,01,00
,00,01,00,00,03,71,00,03,8F,00,01,00,
00,01,00,00,03,71,00,03,8F,00,01,00,0
0,01,00,00,03,71,00,03,8F,00,01,00,00
,01,00,00,04,71,00,08,8F,00
219 4790 DATA 04,7F,00,04,6B,00,04,97,00,
04,00,00,04,BE,00,01,71,00,08,8F,00,0
1,00,00,01,71,00,01,00,00,03,71,00,01
,00,00,03,71,00,08,A0,00,01,00,00,03,
71,00,01,00,00,04,71,00,08,AA,00,04,7
F,00,04,6B,00,04,97,00,04,00,00,04,BE
,00,01,71,00,08,8F,00,01,00
65 4800 DATA 00,01,71,00,01,00,00,03,71,
00,01,00,00,03,71,00,08,A0,00,01,00,0
0,03,71,00,01,00,00,04,71,00,08,AA,00
,04,7F,00,04,6B,00,04,97,00,04,71,00,
04,BE,00,14,8F,00,14,E2,00,08,00,00,0
8,00,00,FF,48,65,6C,6C,6F,2C,20,44,6F
,6C,6C,79,21,20,42,79,20,4A
64 4810 DATA 65,72,72,79,20,48,65,72,6D,
61,6E,20,31,39,36,33
62

```

## Kategorie Spellen

# Keuken

Een spel mag best heel simpel van gegeven zijn. Dat houdt nog lang niet in dat het daardoor minder aantrekkelijk zou wezen.

Zo is de opgave in Keuken op zich doodeenvoudig, je moet allerlei zaken die uit het keukenkastje vallen met de koekepan opvangen en zo richting afwasteil sturen. En dat is heel erg leuk!

Tot de grote verbazing van mensen die *niet* van computerspelletjes houden, overigens. Die zien nu eenmaal werkelijk de lol er niet van in om een dergelijke handeling steeds sneller en sneller te moeten verrichten. Inderdaad is de vraag 'wie programmeert nu wie, de mens of de computer of precies andersom'. Maar daar kunnen we ons op het moment dat we aan het spelen zijn het hoofd niet over breken, die computer maakt het ons al lastig genoeg...

### Vertaling

Keuken werd ons toegezonden door Roelof Ridderman, uit Ruinerwold, samen met nog wat heel erg aardige programma's. Daarvan treft u elders in dit nummer de spritceditor 'Sprite' aan, een geheel door hemzelf ontworpen programma.

Keuken is echter 'slechts' een vertaling van een programma dat oorspronkelijk voor de Spectrum geschreven was. Dat doet echter niets aan de kwaliteit van het spel - en de programmering - af. Iedereen die daar zijn of haar tanden wel eens in heeft gezet zal kunnen beamen dat het vertalen van een programma van de ene computer naar de andere een hele klus kan wezen! Ook daarin gaan heel wat uurtjes ingespannen denkwerk zitten.

### Eenvoudig

Zoals al gezegd. Keuken is qua opzet heel erg eenvoudig. Na-

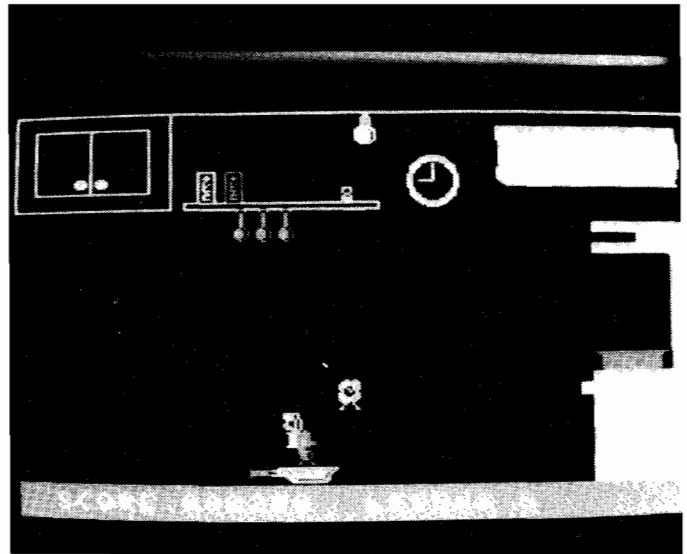
dat er een simpel achtergrondje getekend is kan de pret beginnen, als de introductie-tune afgelopen is.

In dit spel bestuurt u een koekepan, die onder in het beeld van links naar rechts bewogen kan worden met de cursor-toetsen of een joystick in poort 1. Met die koekepan moet u allerlei zaken die uit het keukenkastje - dat links-boven op het scherm zit - komen vallen zien op te vangen.

Aardappels, blikjes en nog andere - door ons niet nader geïdentificeerde - objecten zullen op de keukenvloer kapot vallen, als u ze niet tijdig met de koekepan onderschept. Die koekepan is namelijk heel veerkrachtig, als u eenmaal een bepaald object opgevangen heeft stuiter het weer in een keurige boog omhoog.

Maar ook in deze keuken gaat de wet van de zwaartekracht op, zodat even later - midden op het scherm - de etenswaren alweer kapot dreigen te vallen.

Ook daar is het dan ook zaak om ze netjes met die koekepan op te vangen, waarna ze weer terug stuiteren. Dat herhaalt zich nogmaals helemaal rechts in het beeld, vlak naast het aanrecht. Maar als u ook daar op tijd was met uw reddende koekepan, dan zullen de etenswaren met een fraaie boog in de afwasteil op het aanrecht belanden. En dat is natuurlijk de bedoeling, want wat in het water valt breekt niet en gaat ook niet stuk. Een fraaie plons is uw beloning!



### Steeds meer

Simpel, niet? Zoiets mag natuurlijk niet al te lastig zijn.

Dat is het dan ook niet, in het begin althans. Een zo'n blikje netjes naar de afwasteil laten stuiteren is eigenlijk geen kunst aan, als men het truukje eenmaal door heeft.

Maar daarmee begint het pas! Er vallen namelijk steeds meer dingen uit dat verdraaide keukenkastje! En zodra er ook maar eentje de grond raakt bent u een van uw drie koekepannen kwijt.

Op een gegeven moment zult u merken dat u als een gek van links naar rechts over het scherm moet racen om al die spullen heel te houden, en vroeger of later gaat het gegarandeerd mis. Het valt helemaal niet mee om de high-score - die op vijfduizend staat - te halen. Op de redactie hebben we daarvoor 'vals' moeten spelen, door het aantal levens - in regel 290 - aan te passen.

### Basic

Nee, makkelijk is Keuken zeker niet. Het is echter wel een goed voorbeeld van de uitstekende spellen die er in gewoon simpel Basic geschreven kunnen worden. Het geheim van de kok is daarbij eenvoudig: 'hou het simpel'.

Natuurlijk zou ook een spel als Keuken van allerlei extra's voorzien kunnen worden, ware het niet dat het daardoor zijn snelheid - en dus zijn char-

me - zou verliezen. Geen ingewikkelde geluids-routines, geen extra beeld-effecten, simpel rechttoe rechtaan programmeren, dan is Basic meer dan snel genoeg om heel aardige actiespellen in te ontwikkelen.

### Tenslotte

Keuken was een programma dat direkt aansloeg op de redactie. We hebben er een hele tijd mee gespeeld, waarbij opviel dat het juist ook door zijn eenvoud aansprak. We hebben er ook vrijwel niets aan veranderd; de listing bij dit artikel is vrijwel gelijk aan de versie die R. Ridderman ons opstuurde. Met wat extra commentaar was Keuken wat ons betreft af.

Aan de hand van dat commentaar kunt u desgewenst zelf eens proberen uit te vinden hoe het programma in zijn werk gaat. Het geheel is behoorlijk overzichtelijk van opbouw, met duidelijk onderscheiden subroutines die ieder bepaalde functies verrichten. Daardoor is het niet lastig om de diverse sprongen te volgen.

De uiterst speelbare snelheid van het geheel komt vooral door de goed opgezette hoofd-lus, regels 460 tot en met 590. Door die lus zo 'kaal' mogelijk te houden wordt erg veel tijd gewonnen.

Al met al vinden we Keuken een erg aardig - en nogmaals, bijzonder lastig - spel. Een goede prestatie van lezer Roelof Ridderman, die we bij deze nogmaals willen bedanken!

```

10 REM KEUKEN 0
20 REM 0
30 REM MSX Computer Magazine 0
40 REM 0
50 REM ingezonden door R. Ridderman, Ruinerwold 0
60 REM 0
70 '***** 8
80 '* kitchen chaos * 17
90 '* ----- * 116
100 '***** 0
110 ' 0
120 CLEAR 2000: KEY OFF: CLS: GOSUB 6 96
130 ON STOP GOSUB 2440: STOP ON 160
140 DEF USR0=&HC0: DEF USR1=&HI56 29
150 ON STRIG GOSUB 1990 212
160 DEFINT A-Z 120
170 HI=5000 192
180 DIM V(3,96),Z(3),I(3),S(3) 229
190 FOR F=1 TO 96 56
200 READ V(1,F): V(2,F)=V(1,F) 92
210 V(3,F)=V(1,F): 175
220 NEXT F 174
230 SCREEN 2,0,0 224
240 OPEN "GRP:" AS #1 67
250 GOSUB 710 129
260 GOSUB 600 108
270 GOSUB 840 180
280 U=USR1(0) 141
290 SC=0: LI=3 237
300 GOSUB 1400 250
310 FOR F=1 TO 3 52
320 Z(F)=1 184
330 NEXT F 177
340 C=0:Y=64:X=24 182
350 S(1)=1: I(1)=1 187
360 FOR F=2 TO 3 73
370 S(F)=INT(RND(-TIME)*60)+40 232
380 I(F)=0 65
390 NEXT F 189
400 H=8: GOSUB 2190 193
410 '***** 0
420 '* hoofd loop * 0
430 '***** 0
440 ' 0
450 U=USR0(0) 127
460 C=C+1 83
470 FOR N=1 TO 3 137
480 IF C=S(N) THEN I(N)=1 62
490 IF I(N)=0 THEN 540 15
500 X=V(N,Z(N)): Y=V(N,Z(N)+1): Z(N)=Z(N)+2 57
510 ON N GOSUB 1520,1550,1580 22
520 IF Y=150 AND X<>H+16 THEN GOTO 1610 46
530 IF Z(N)=97 THEN Z(N)=1: GOSUB 2050 254
540 B=STICK(0) OR STICK(1) 200
550 IF (B<>3 AND B<>7) THEN 580 248
560 IF (B=3 AND H<140) THEN H=H+80: GOSUB 2190 102
570 IF B=7 AND H>16 THEN H=H-80: GOSUB 2190 1
580 NEXT N 13
590 GOTO 460 114
600 '***** 0
610 '* tune * 0
620 '***** 0
630 ' 0
640 PLAY "" 127
650 PLAY "04T60S0M10000","04T60S0","04T60S0" 135
660 PLAY "L8BBBB","L8GGGG","L8DDDD" 137

```

```

670 PLAY "L1605DCC04BA4","L16BAAGE4","L16F+EEDC4" 237
680 PLAY "L8AAB05C","L8F+F+GA","L8DDEF+" 148
690 PLAY "05D4GR8","A405DR8","F+4BR8" 81
700 RETURN 193
710 '***** 0
720 '* sprites definieren * 0
730 '***** 0
740 ' 0
750 FOR I=1 TO 19 4
760 AS="" 233
770 FOR J=1 TO 8 96
780 READ A 14
790 AS=AS+CHR$(A) 61
800 NEXT J 188
810 SPRITES(I)=AS 137
820 NEXT I 210
830 RETURN 200
840 '***** 0
850 '* scherm opzetten * 0
860 '***** 0
870 ' 0
880 COLOR 15,1,1: CLS 134
890 PSET (255,0),5 48
900 DRAW "A0S4C5L255D48R60U48" 187
910 PSET (8,8),5 210
920 DRAW "C5D32R44U32L44R22D32" 171
930 CIRCLE (26,35),2.5,,,1.3 111
940 PAINT (26,35),5 173
950 CIRCLE (34,35),2.5,,,1.3 100
960 PAINT (34,35),5 163
970 PUT SPRITE 26,(127,0),4,15 226
980 PUT SPRITE 25,(127,8),11,16 79
990 PSET (180,8),13 2
1000 DRAW "C13R70D8R1D12L1D5L1D2L1D1R1D1R1L68U1L1U1L1U1L1U5R1U15L1U6" 247
1010 PAINT (181,9),13 215
1020 PSET (200,12),0 4
1030 PRINT#1,"HIGH" 240
1040 B$=RIGHT$(STR$(HI+1000000!),6) 51
1050 PSET (192,25),0 213
1060 PRINT#1,B$ 192
1070 CIRCLE (155,35),12,7,,,1.3 32
1080 CIRCLE (155,35),11,7,,,1.3 12
1090 CIRCLE (155,35),10,7,,,1.3 248
1100 PSET (155,35),5 50
1110 DRAW "U8D8L5" 166
1120 PSET (65,48),5 175
1130 DRAW "R70U3L70D3" 130
1140 PUT SPRITE 24,(70,29),14,13 27
1150 PUT SPRITE 23,(70,37),14,14 18
1160 PUT SPRITE 22,(80,29),6,13 42
1170 PUT SPRITE 21,(80,37),6,14 32
1180 PUT SPRITE 18,(90,47),6,17 26
1190 PUT SPRITE 17,(90,55),6,18 16
1200 PUT SPRITE 20,(82,47),6,17 156
1210 PUT SPRITE 15,(98,55),6,18 134
1220 PUT SPRITE 19,(82,55),6,18 56
1230 PUT SPRITE 16,(98,47),6,17 153
1240 PUT SPRITE 14,(120,37),13,19 120
1250 PSET (255,175),7 191
1260 DRAW "C7U120L40D4R18D6L18D4R32D56L32D50R40" 105
1270 PAINT (254,174),7 144
1280 PSET (215,116),6 43
1290 DRAW "C6R2D1R1D6R1D1R22U1R1U6R1U1R2L30" 218
1300 PAINT (220,118),6 225
1310 LINE (0,176)-(255,191),2,BF 93
1320 LINE (215,130)-(210,135),4,BF 47
1330 PSET (50,180),2 186
1340 IF PLAY(0) THEN 1340 215
1350 RETURN 144

```

```

1360 '*****
1370 '* telling printen *
1380 '*****
1390 '
1400 LINE (0,176)-(255,191),2,BF
1410 B$=RIGHT$(STR$(SC+1000000#),6)
1420 PSET (16,180),2
1430 PRINT#1,"SCORE:";B$
1440 PSET (130,180),2
1450 B$=RIGHT$(STR$(LI+10),1)
1460 PRINT#1,"LEVENS:";B$
1470 RETURN
1480 '*****
1490 '* sprites bewegen *
1500 '*****
1510 '
1520 PUT SPRITE 1,(X,Y),6,5
1530 PUT SPRITE 2,(X,Y+8),6,6
1540 RETURN
1550 PUT SPRITE 3,(X,Y),10,7
1560 PUT SPRITE 4,(X,Y+8),10,8
1570 RETURN
1580 PUT SPRITE 5,(X,Y),14,9
1590 PUT SPRITE 6,(X,Y+8),14,10
1600 RETURN
1610 '*****
1620 '* mispikkel bijtellen *
1630 '*****
1640 '
1650 LI=LI-1
1660 ON N GOSUB 2240,2270,2300
1670 FR=5: GOSUB 2490
1680 FOR I=1 TO 100
1690 NEXT I
1700 GOSUB 2330
1710 IF LI=0 THEN 1730
1720 GOTO 300
1730 '*****
1740 '* the end *
1750 '*****
1760 '
1770 CLOSE #1: SCREEN 1: COLOR 15,1,1
1780 IF SC>HI THEN HI=SC
1790 LOCATE 9,9: PRINT "K I T C H E N
1800 LOCATE 11,11: PRINT "C H A O S"
1810 LOCATE 8,15: PRINT "UW SCORE:";
:PRINT USING "#####";SC
1820 LOCATE 8,16: PRINT USING "HIGHSC
ORE:#####";HI
1830 MS$="PRESS SPACE TO PLAY"
1840 STRIG(0) ON
1850 MS$=SPACE$(27)+"* "+MS$+" *"+SPA
CE$(10):ML=LEN(MS$):
1860 DM$=LEFT$(MS$,27)
1870 FOR I=27 TO ML
1880 DM$=RIGHT$(DM$,25)+MID$(MS$,I,
1)
1890 FOR N=1 TO 50
1900 NEXT N
1910 LOCATE 2,21
1920 PRINT DM$
1930 NEXT I
1940 DM$=RIGHT$(MS$,ML-27): GOTO 1870
1950 '*****
1960 '* nieuw spel starten *
1970 '*****
1980 '
1990 RETURN 2000: ' ombuigen on strig
2000 STRIG(0) OFF: BEEP
2010 SCREEN 2
2020 SOUND 7,57
2030 OPEN "GRP:" AS #1
2040 GOTO 260

```

```

2050 '*****
2060 '* score bijtellen *
2070 '*****
2080 '
2090 ON N GOSUB 2240,2270,2300
2100 FR=30: GOSUB 2490
2110 SC=SC+(100*N)
2120 GOSUB 1400
2130 IF N>1 THEN C=0: I(N)=0: S(N)=IN
T(RND(-TIME)*10+1)
2140 RETURN
2150 '*****
2160 '* sprites bewegen *
2170 '*****
2180 '
2190 PUT SPRITE 27,(H,166),6,1
2200 PUT SPRITE 28,(H+8,166),4,2
2210 PUT SPRITE 29,(H+16,166),4,3
2220 PUT SPRITE 30,(H+24,166),4,4
2230 RETURN
2240 PUT SPRITE 1,(X-4,Y+8),6,11
2250 PUT SPRITE 2,(X+4,Y+8),6,12
2260 RETURN
2270 PUT SPRITE 3,(X-4,Y+8),10,11
2280 PUT SPRITE 4,(X+4,Y+8),10,12
2290 RETURN
2300 PUT SPRITE 5,(X-4,Y+8),14,11
2310 PUT SPRITE 6,(X+4,Y+8),14,12
2320 RETURN
2330 PUT SPRITE 1,(-4,-8),6,1
2340 PUT SPRITE 2,(-4,-8),6,12
2350 PUT SPRITE 3,(-4,-8),10,11
2360 PUT SPRITE 4,(-4,-8),10,12
2370 PUT SPRITE 5,(-4,-8),14,11
2380 PUT SPRITE 6,(-4,-8),14,12
2390 RETURN
2400 '*****
2410 '* ctrl-stop routine *
2420 '*****
2430 '
2440 SCREEN0: CLS: END
2450 '*****
2460 '* plons-geluid *
2470 '*****
2480 '
2490 SOUND 0,0: SOUND 1,0: SOUND 6,FR
: SOUND 7,6
2500 SOUND 8,16: SOUND 9,16
2510 SOUND 10,16
2520 SOUND 11,0: SOUND 12,40
2530 SOUND 13,0
2540 RETURN
2550 '*****
2560 '* sprite data *
2570 '*****
2580 '
2590 DATA 24,74,24,78,24,82,24,86,24,
94,24,102,24,110,24,118,24,126,24,134
,24,142,24,150,28,142,32,134,36,126,4
0,118,44,110,48,102,56,98,64,94,72,98
,80,102,84,110,88,118,92,126,96,134
2600 DATA 100,142,104,150,108,142,112
,134,120,126,128,118,136,114,144,110,
152,114,160,118,168,126,176,134,180,1
42,184,150,188,142,192,134,196,126,20
0,118,206,110,214,100,220,92,226,100
2610 DATA 0,128,255,254,255,128,0,0,0
,63,255,45,255,7,3,1,0,255,255,255,25
5,255,128,255,0,255,250,244,232,232,2
4,240
2620 DATA 24,60,124,124,108,111,111,1
24,112,124,122,58,58,28,34,99
2630 DATA 0,54,127,255,231,215,74,195
,231,255,254,116,36,66,195,0

```

2640 DATA 252,244,244,122,74,202,234, 122,29,125,61,61,61,61,61,127	110
2650 DATA 8,0,26,25,8,35,51,1,0,134,1 06,96,74,24,32,0	230
2660 DATA 254,130,146,186,146,162,170 ,178,162,186,146,186,162,186,130,124	95
2670 DATA 24,24,60,36,60,36,60,126,25 1,253,253,253,253,122,60	7
2680 DATA 24,36,24,24,24,24,24,24,24, 52,122,122,122,122,52,24	53
2690 DATA 60,36,36,24,36,60,60,60	105



# TDK. Technologie in optima forma.



TDK diskettes dragen als enige ter wereld, het kwaliteitszegel van het Reliability Center for Electronic Components uit Japan. Een betere referentie bestaat er niet!

- Getest op 20 miljoen passages per spoor.
- Precisie-oppervlak voor optimaal kop/diskette-contact.
- Sterk verhoogde window-exactheid.
- Beveiligd tegen temperaturen tot 60° en tegen extreme vochtigheid.
- Super-White versterkingsring voor extra duurzaamheid.
- Leverbaar in 8, 5¼ en 3½ inch en in 13 uitvoeringen.



 **TDK**®

Audiocassettes | Videocassettes | Floppy Disk