

18
Winter 2000

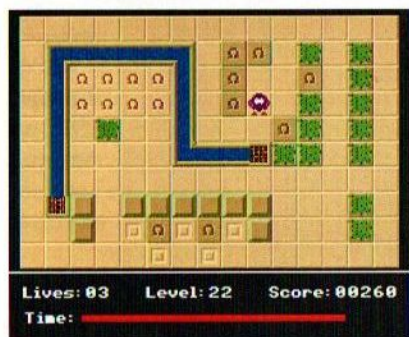
MSX Info Blad

Inhoud:

Voorwoord	2
Bussum beursverslag	3
Final Bout	5
Inbouwen 1 MB SIMM	9
Universele PC geluidskaart interface	12
Moondop #2	14
Cursus PowerBasic	16
Gamesabbonement?!	20
Nieuwe 8-bits slotexpander	22
Ontwikkelingen in Japan	23
3D Games op MSX	24
Push 'em	32



Final Bout: Wie houdt het vol tot het laatste rondje?



Push'em: Schuiven, schuiven en nog eens schuiven...

Zou het er dan nu toch van komen?
ASCII heeft plannen voor een MSX3!

Colofon:

Het MSX-Info Blad is een uitgave van het Totally Chaos Team. Het verschijnt 4 keer per jaar. Een jaarabonnement kost Fl.15,—. Losse nummers kosten Fl.4,50. Op beurzen geldt de speciale prijs van Fl.3,50 per nummer.

Uitgever:

M.T. Stoker
Rotterdamstraat 73
6415 AV Heerlen
Tel: (045) 572 95 09
Mobiel: (06) 23 30 09 60
Giro: 24.95.572
SNS: 92.88.51.222

Redactie:

Gerrit van den Berg
Fokke Post
Jan-Marten van der Reest

Medewerkers aan dit nummer:

Raymond de Heer, David Heremans, Rob Hiep, Sjoerd Mastijn, Hans Oranje, Alex Peetoom, Tristan Zondag

Lay Out:

Fokke Post
Waaierhoek 36
8321 BH Urk

Drukwerk:

Copy Service Hattem
Hoopjesweg 1
8051 DB Hattem
Tel: (038) 444 85 77

Copyright © 2000

Totally Chaos Team

Niets uit deze uitgave mag worden vermenigvuldigd zonder schriftelijke toestemming van de uitgever.

Beste lezer

Allereerst wil ik u mede namens de redactie een gelukkig nieuwjaar toewensen. We hopen dat u dit jaar in een goede gezondheid en voorspoed mag leven. Verder hopen we dat u nog lang doorgaat met MSX. Wij zullen u daarin zo lang mogelijk blijven ondersteunen.

De beurs in Bussum ligt al weer een tijdje achter ons. Aan onze stands zijn diverse leden geweest, die het verhoogde abonnementsgeld alsnog hebben betaald. Wij zijn daar blij mee, maar we hopen dat dit in het vervolg tijdig via de bank voldaan wordt. Ook zijn er nog steeds mensen, die hun abonnementsgeld nog niet hebben betaald. Indien er geen respons volgt op de door ons verzonden brief, zullen de betreffende mensen uit het abonneebestand worden verwijderd.

Wij maken ons op voor de MSX beurs in Oss op 20 januari 2001. Wij hopen hier ook aanwezig zijn met een stand. Naast het verkopen van bladen en software, zoals *Kyokugen*, hopen we u ook de nieuwe *LPE Z380* cartridge en de *Evolucion4* slotexpander te demonstreren. Op dit moment is de eerste bestelling in Spanje gedaan en zal binnenkort worden uitgeleverd. U kunt nog steeds hardware bij ons bestellen. De prijs voor een *LPE Z380* is Fl. 415,— en voor een *Evolucion4* slotexpander Fl. 390,— inclusief kabel. Deze prijzen zijn inclusief verzendkosten.

We zijn bezig om te komen tot de oprichting van een MSX stichting. Met deze stap willen we proberen om een nog betere MSX promotie in gang te zetten en te houden. Hopelijk kunnen we u daar in het volgende nummer meer informatie over verstrekken.

Heeft u ideeën voor ons blad of zou u bepaalde onderwerpen eens behandeld willen zien of zelf willen behandelen, laat het dan even aan ons weten. Het is al vaker gezegd, maar zonder uw hulp komen we nergens. Hopelijk tot ziens op de beurs en veel leesplezier toegewenst van uw uitgever.

Rinus Stoker



Bussum beursverslag

Heden ten dage worden er naast wat kleinere beurzen twee 'grote' MSX beurzen in Nederland gehouden. De Bussumbeurs als vervanger van de Zandvoortbeurs is er daar één van. Wij hebben voor u kort samengevat wat er zoal te doen en te zien was.

Omdat deze beurs een voortzetting is van de Zandvoortbeurs konden we spreken van de 12^e Computerdag. Het was echter de tweede keer dat deze in Bussum gehouden werd. Evenals de eerste keer was het geheel ondergebracht in het Sociaal-cultureel centrum UIT-WIJK aan de Dr. Schaepmanlaan 10 in Bussum. Dit is een oud schoolgebouw en de meeste stands waren dan ook verdeeld over een viertal lokalen.

Veruit de meeste MSX clubs c.q. verenigingen waren aanwezig. Ondanks de lage verwachtingen bleek er toch nog nieuwe software te koop. Zo bracht *TeddyWareZ* een leuke *Sokoban* kloon op de markt, genaamd *Push 'em*. Dit is een spel waarbij het de bedoeling is om de aanwezige blokjes in het speelveld in de daarvoor bestemde vakjes te schuiven. Zoals we van *TeddyWareZ* konden verwachten ondersteunt dit spel de SCC geluidschip.

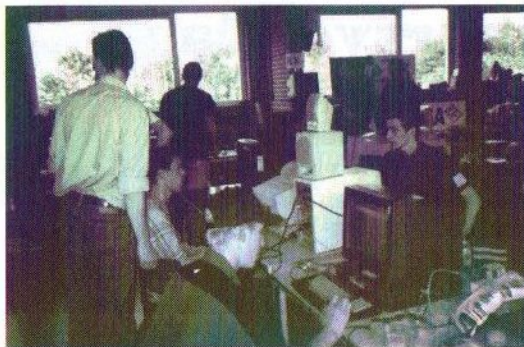
Astennu verkocht een MoonSound muziekdisk genaamd *Unholy symphonies*. Gelukkig dekt de vlag de lading in dit geval niet en betreft het covermuziek uit diverse MSX spellen. Ook werd het project *Vampiric Evil* aan het publiek getoond. Het is echter nog maar de vraag of dit project ooit afgemaakt wordt. Er zijn

namelijk plannen om over te stappen op het PlayStation 2 systeem. *MSX NBNO* verkocht het vechtspel *Final Bout*, waarover u in dit blad meer kunt lezen. *Deltasoft* verkocht het spel *Thunderbirds are go* en tevens werd een aankondiging gemaakt van het spel *Konapuzz*, dat naar verwachting binnenkort gereed zal zijn.

The New Image heeft een zogenaamde crossassembler gemaakt, waarmee op de PC voor MSX geprogrammeerd kan worden. In de toekomst zal bekeken worden of er ook Z380 support ingebouwd kan worden. *Patriek Lesparre* heeft ondertussen een *LPE Z380* besteld en hopelijk kunnen we in de toekomst nog wat van hem verwachten op dat gebied.

Natuurlijk is een beurs zonder *Sunrise* geen echte MSX beurs. Zij waren met een flinke stand aanwezig en lieten hun hardwareassortiment zien. Er was een 8 slotexpander aanwezig, die ze in een PC kast hadden





gebouwd. Tevens was er een nieuwe lading gecombineerde IDE/RS232C interfaces binnen. Voor de mensen die nog een nieuwe MoonSound wilden aanschaffen was er nog de mogelijkheid om zich voor een exemplaar in te schrijven. Er wordt namelijk nog een beperkte oplage gemaakt. Helaas bleek dat het niet mogelijk was om de beloofde MoonSound muziek voor het spel *The Lost World* op de diskettes te laten zetten.

Gelukkig blijven er nog steeds nieuwe magazines uitkomen, al wordt de stof waarover te schrijven valt wel minder. *MSX NBNO* had weer een nieuwe *XSW Magazine* beschikbaar en ook wij verkochten ons laatst uitgekomen *MSX-Info Blad*. *Koen Dols* schijnt ook aanwezig te zijn geweest met de *Futuredisk*, maar deze heb ik dan kennelijk over het hoofd gezien.

Verder waren onder andere *MSX Club West-Friesland* aanwezig met veel kabels en andere hardwaresnuisterijtjes. *Compjoetania TNG* was ook weer present en toonde het programma *Compass 2.1* en verder kon er nog wat oudere software worden

aangeschaft. *Remco van der Zon* trachtte zijn voorraad MP3 Cd's te slijten. Er was zelfs een stand waar je nog grote hoeveelheden DD diskettes kon kopen. Daar is tegenwoordig bijna niet meer aan te komen.

Natuurlijk waren wij zelf ook aanwezig en we probeerden weer wat nieuwe leden voor ons blad erbij te krijgen. Daarnaast verkochten we het spel *Kyokugen* en de muziekcd *AlgoRythm*. Ook hadden we een Sony900 MSX computer staan waaraan een laserdisk gekoppeld was. Dit zeldzame stukje MSX hardware werd door velen bewonderd.

Ook onze vrienden uit Tilburg waren aanwezig om de beurs van 2001 te promoten. Hopelijk wordt ook die beurs weer een succes, want zonder dit soort beurzen zou MSX langzaam doodbloeden. Er is echter eerst nog de beurs in Oss op 20 januari en daarom roepen wij u allen op om ook deze beurs eens te bezoeken.

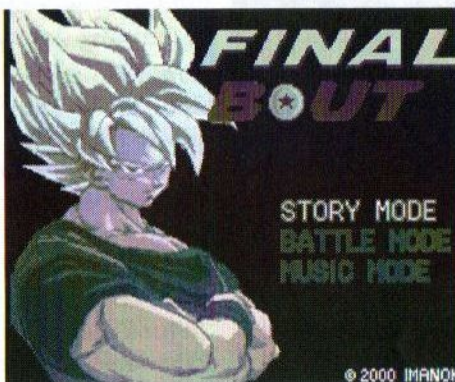
Gerrit van den Berg



Final Bout

Onder bovenstaande titel gaat volgens de maker een zogenaamde 'energy-attack fighting game' schuil. Velen zullen waarschijnlijk niet weten, wat dat precies inhoudt. Hopelijk kunnen we u uitleggen wat ermee bedoeld wordt.

Enkele maanden geleden werden wij benaderd door iemand uit Spanje met het verzoek om eens te kijken of wij zijn nieuwste spel zouden willen distribueren in Nederland. Wij moesten even nadenken met wie wij precies te maken hadden, maar na enig speurwerk kwamen wij er achter wie *Final Bout* heeft ontwikkeld.



Enkele jaren geleden werd namelijk op de beurs in Tilburg door een stel Spanjaarden een spel aangeboden met de naam *Rolling Thunder*. Dit is een vrij leuk kaatsspel, waarbij de mogelijkheid geboden werd om tegen elkaar te spelen. De maker van dat spel is *Fernandez D. Sopra-Bull* alias *Imanok* (Konami omgedraaid). Nu heeft hij dus opnieuw een spel gemaakt, dat ook een tweespeler optie heeft. Dit keer is de bedoeling van het spel niet om een balletje in het doel van de ander te gooien, maar meer om een 'balletje' dwars door de ander heen te gooien.

Omdat wij niet geïnteresseerd waren in distributie van *Final Bout* is *Ima-*

nok verder gaan zoeken naar geschikte kandidaten. Op de beurs in Bussum is gebleken, dat uiteindelijk *MSX NBNO* bereid is gevonden om de distributie op zich te nemen. Het spel wordt geleverd op een diskette met kleurenlabe, dat samen met een Engelstalige handleiding in een plastic zakje is verpakt. Gelukkig is het spel ook op harddisk installeerbaar, waardoor de laadtijd aanzienlijk verkort wordt. Het programma is gemaakt in *NestorBasic* en is daardoor niet zo traag als een gewoon BASIC programma.

Na het opstarten verschijnt al snel het *Imanok* logo op het scherm, waarna het titelscherm op het beeld geprojecteerd wordt. Dit alles wordt vergezeld door een vrij vlot MSX-stereo muziekje. Het scherm wordt grotendeels ingenomen door een tekening van een vrij fors uitziend mannetje, dat kennelijk een goede haarversteviger gebruikt heeft, getuige de weelderige en vrij puntig uitziende haardos. Het plaatje ziet er goed uit, maar het heeft er wel alle



schijn van dat het een gedigitaliseerd plaatje betreft. Naast de titel van het spel zijn er drie modes waaruit een keuze kan worden gemaakt.

Story mode

De eerste mode is de zogenaamde 'story mode' die de speler het één en ander uitlegt over de achtergrond van het spel. In de door *Imanok* aan ons verstrekte versie is de tekst in het Spaans geschreven. Deze uitleg is voor degenen die deze taal niet machtig zijn niet verhelderend, maar volgens mij gaat het om *Gohan*, die meedoet aan een kampioenschap voor vechtsporten. Dit introverhaal is voorzien van diverse zwart-wit plaatjes, die er manga-achtig uitzien.

Na deze intro verschijnt het strijdtoneel, waarbij zowel aan de linker- als aan de rechterkant een strijdlustig type te bewonderen is. Deze karakters zien er vrij gedetailleerd uit en zijn best goed getekend. De personage links kan door de speler bestuurd worden. Het is jammer, dat alleen boven in het scherm een indruk wordt gegeven van de omgeving waarin de twee vechtsportgiganten

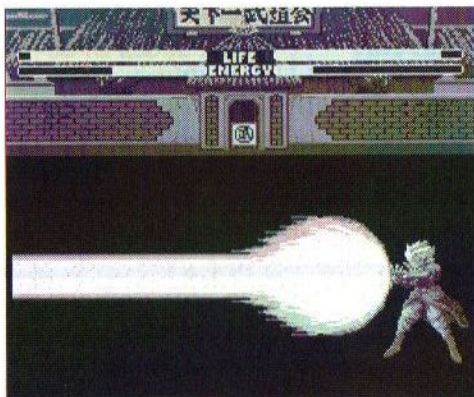
hun strijd uitvechten. Hier is ook een levens- en energiebalk te vinden. Het gedeelte waar alles zich afspeelt is echter geheel zwart en wordt gescheiden door een simpele witte lijn. Niet echt indrukwekkend dus.

Zoals gezegd is dit spel gebaseerd op energy-attacks, wat betekent dat de tegenstander moet worden uitgeschakeld met behulp van energiestoten. Er is dus geen sprake van 'full contact', zoals in bijvoorbeeld een spel als *Fighters Ragnarök* wel het geval is. Door op de cursortoets [omhoog] te drukken of het equivalent van de joystick te gebruiken verschijnt er een gloed om de personage en wordt het energiebalkje langzaam gevuld. Tussentijds kan men de tegenstander echter al wel 'te lijf' gaan door een salvo af te vuren. Er gaat dan wel wat van de opgebouwde energie af. Als een bepaalde energietoestand is bereikt, dan is het mogelijk om een extra zwaar schot af te vuren. Tijdens het afvuren van zo'n schot verdwijnt de tegenstander even uit beeld om vervolgens weer te verschijnen. Na een heftige explosie, waarbij het net lijkt of het personage



in rook opgaat, vervolgt de strijd zich alsof er niet gebeurd is. Of een normaal mens net zo makkelijk weer zal opstaan na een dergelijke explosie durf ik overigens te betwijfelen.

Naast het zelf uitdelen van energiestoten hoeft natuurlijk niet verwacht te worden, dat de tegenstander, die in deze mode door de computer bestuurd wordt, dit alles over zijn kant laat gaan. Voortdurend worden er over en weer vuursalvo's geplaatst. Een niet overbodige optie in het spel is dan ook die van het pareren van de aanval van de tegenstander. Dit kan op verschillende manieren. De eenvoudigste manier is om een tegenaanval te plaatsen, waardoor de energiestoten elkaar als het ware oplossen. Deze mogelijkheid gaat echter niet op voor het afvuren van het extra zware schot. Een andere mogelijkheid is om de 'kogels' met de handen of voeten te blokken.



Als de energiebalk op is, dan is dat te zien aan de houding van het personage. Hij of zij ziet er dan namelijk tamelijk uitgeput uit. Er moet nu eerst gewacht worden tot de tegenstander een schot heeft gelost, voor-

dat er weer opgeladen kan worden. Dit is niet echt consequent. Verder is het opvallend, dat het opladen van de spelers onderbroken wordt als er door één van beiden een schot wordt afgevuurd.

Nadat de tegenstander is verslagen verschijnt er een kleine tussendemo met de bekende zwart-wit plaatjes, waarna vervolgens de volgende tegenstander zich aandient. Voor zover ik heb kunnen zien zijn er in totaal zes verschillende deelnemers.

Battle mode

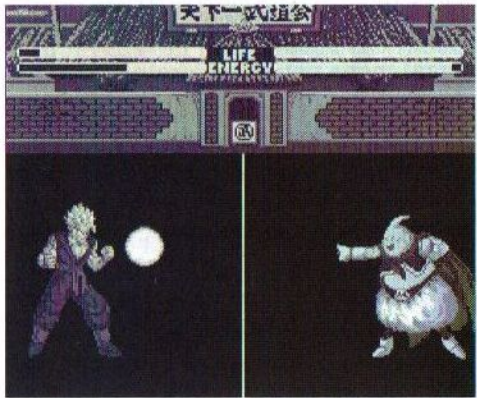
Voor de liefhebbers is er ook de mogelijkheid om tegen iemand anders dan de computer te spelen. Deze optie is bereikbaar door de tweede mode in het titelscherm te kiezen. In het dan volgende selectieschermpje kan iedere speler uit één van de zes beschikbare giganten een keuze maken. Naast veel flitsend uitziende types staan er ook zo op het eerste gezicht wat minder sterk ogende deelnemers op de lijst. Het karakter met de naam *Boo* heeft bijvoorbeeld wel het postuur van een sumoworstelaar, alhoewel hij (of is het een zij?) beduidend meer kleding dan zijn traditionele collegae draagt. Evenals de echte sumoworstelaars dienen zij echter niet onderschat te worden.

Music mode

Als het spel gaat vervelen — en dat doet het naar mijn mening vrij snel — dan kan altijd nog de toevlucht worden genomen tot de muziek, die onder de laatste mode te vinden is. In totaal kan er een keuze worden gemaakt uit tien muziek-



stukken. Deze zijn allemaal gemaakt door *Manuel Dopicor*, die wellicht bekend is van de *Moondop* muziek-diskettes. Zoals al eerder is opgemerkt bestaat de muzikale ondersteuning uit stereo muziek, beter bekend als de combinatie van MSX-Audio en MSX-Music. De muziek is zeker niet slecht, al zijn de meeste liedjes vrij kort en worden er vaak dezelfde muziekinstrumenten gebruikt.



Conclusie

Qua muzikale en grafische omlijsting kan het spel er mee door. Wat betreft de grafische omlijsting dient wel gezegd te worden dat deze met name in het eigenlijke speelveld vrij karig is uitgevoerd. Gelukkig zeggen muzikale en grafische omlijsting ook niet alles en is er ook nog zoiets als het speelplezier dat een spel te bieden heeft. Eerlijkheidshalve dient echter gezegd te worden, dat het spel op dit punt niet al te best scoort. Het gaat namelijk vrij snel vervelen. Dit en de wijze waarop het spel geleverd wordt maakt dat de prijs, die er voor moet worden betaald, aan de hoge kant is. Hopelijk gaat *Imanok* door met het maken van spellen, maar dan wel in de trant van het spel *Rol-*

ling Thunder. Voor meer informatie of voor bestellingen kan op de volgende manieren contact worden gezocht:

MSX NBNO

I. van Portugalstraat 9

5346 PJ OSS

Telefoon: (0412) 63 06 53 (Mari)

(0412) 69 07 57 (Maico)

Gerrit van den Berg

Auteur:

Fernandez D. Sopra-Bul

Soort:

spel

Genre:

vechten

Systeemeisen: MSX2 met 128 kB

Optioneel: MSX-Music/Audio

Prijs: Fl. 20,-

Te koop

- Graphics9000 Fl. 250,-
- Tuner voor gebruik met uw MSX Fl. 75,-
- Enkele PC kasten met voeding om uw harddisk en cd-romspeler in te bouwen, per stuk F25,-
- SONY en NMS 8250 MSX computers
- Diverse MSX boeken (bel voor meer informatie)

Telefoon: (045) 572 95 09 of (06) 23 30 09 60

Rinus Stoker

Inbouwen van 1 MB SIMM in Philips NMS 8245

Nadat we in het verleden hebben uitgelegd hoe een 1 MB SIMM ingebouwd kan worden in een Philips NMS 8250/55/80 steeg de vraag naar een beschrijving van de inbouw van een 1MB SIMM in een NMS 8245. Hans Oranje beschrijft hoe ook deze telg uit de Philips NMS serie van een SIMM module kan worden voorzien.

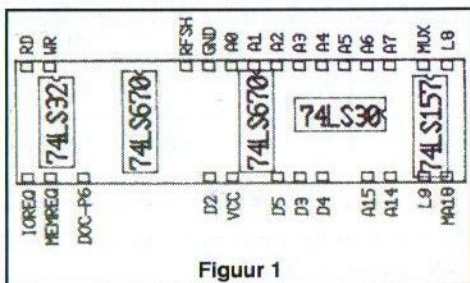
De hieronder staande configuratie is ook uit te voeren voor een Philips VG 8235/20. Voor deze ombouw zijn de volgende onderdelen benodigd:

- 1 x mini-mapper print
- 1 x #CAS selectieprint
- 1 x 1MB 3 chip PC SIMM 30 pins

Een bestuikte mini-mapper print met onderdelen kost Fl. 25,— en een #CAS selectieprint kost met onderdelen Fl. 10,—. Het is ook mogelijk om deze uitbreiding door HPN te laten verzorgen. Dit kost inclusief onderdelen slechts F 50,—. Als u niet zeker bent van uzelf, laat deze uitbreiding dan door iemand met ervaring verzorgen. De auteur van dit artikel noch *MSX-Info Blad* kan aansprakelijk gesteld worden voor schade, die het gevolg is van het (onkundig) uitvoeren van deze uitbreiding aan de hand van deze beschrijving.

Verwijder eerst van de print de geheugen IC's U3, U4, U5 en U6 (41464) door de IC pennen langs zij af te knippen en te desolderen uit de print.

De mini-mapper print heeft, zoals figuur 1 aangeeft, meerdere mogelijkheden voor geheugen configuraties.



Figuur 1

De aansluitingen zijn ten opzichte van de Z80 processor gemaakt, omdat deze print aan de onderzijde van het moederbord tussen de pennen van de Z80 wordt gemonteerd. Echter worden niet alle aansluitingen van de print verbonden met de Z80, deze worden onder andere gemonteerd naar de #CAS-selectie print en de 1 MB SIMM.

De aansluitingen die door middel van een draadje aan de Z80 processor verbonden worden zijn weergegeven in tabel 1.

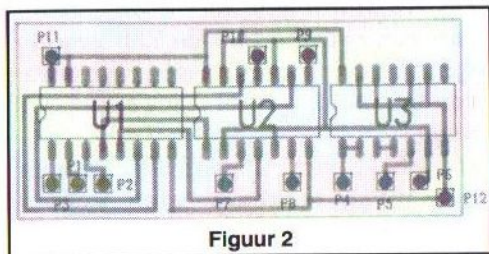
In tabel 2 kunt u zien welke pinnen met welke punten moeten worden verbonden.

Omschr.	Z80 pin	Omschr.	Z80 pin
IOREQ	20	RD	21
MEMREQ	19	WR	22
D2	12	RFSH	28
VCC	11	GND	29
D5	9	A0	30
D3	8	A1	31
D4	7	A2	32
A15	5	A3	33
A14	4	A4	34
		A5	35
		A6	36
		A7	37

Tabel 1

Het volgende onderdeel is de #CAS selectieprint. Zie figuur 2.

Voor deze configuratie worden alleen U2 (74LS08) en U3 (74LS00) op de print en een draadje tussen pen 1 van U1 en pen 1 van U2 gemonteerd. U3 wordt toegepast om tijdens het refreshen te voorkomen dat er test-bits geschreven worden in het geheugen. In combinatie met U2 worden de signalen uit U3 gekoppeld met het #CAS2/E signaal. Zie tabel 3.



Figuur 2

Omschr.	naar
DOC-P6	#CAS selectieprint punt
L8	pen 17 van SIMM
L9	pen 18 van SIMM
MUX	M60003 pen 34 (printspot)

Tabel 2

Omschr.	Naar
P11	+ 5 Volt
P12	Gnd
P7	#CAS pen 2 van SIMM
P5	#WE van pen 4 U3 (voormalig 41464)
P4	afkomstig van mini-mapper print (DOC-P6)
P3	CAS2/E van M60003A pen 80 (TP7) printspot

Tabel 3

Monteer de #CAS selectie print onder de ROM aan de onderzijde van het moederbord.

De 1 MB SIMM wordt gemonteerd aan de onderzijde van het moederbord onder de M60003A en de S-3527.

De aansluitingen voor de 1 MB SIMM vindt u in tabel 4.

Mochten er nog vragen zijn over geheugenuitbreidingen in andere MSX2 computers neem dan contact op via e-mail: msx@hansoranje.demon.nl of op spraak tussen 19.00 uur en 21.00 uur van maandag tot en met vrijdag op telefoonnummer (0182) 61 98 32

Hans Oranje

1 MB SIMM	pin	Naar	1 MB SIMM	pin	Naar
VCC	30	pen 9 van U6 (+ 5 V)	L7	15	pen 10 van U5
D _{in} (9)	29		L6	14	pen 6 van U5
#CAS(9)	28		D3	13	pen 17 van U3 (D7)
#RAS	27	pen 5 van U6 (#RAS)	L5	12	pen 7 van U5
D _{out} (9)	26		L4	11	pen 8 van U5
D7	25	pen 17 van U4 (D3)	D2	10	pen 15 van U3 (D6)
NC	24		Gnd	9	pen 18 van U3
D6	23	pen 15 van U4 (D2)	L3	8	pen 11 van U3
Gnd	22	pen 18 van U4	L2	7	pen 12 van U3
#WE	21	pen 4 van U4 (#WE)	D1	6	pen 3 van U3 (D5)
D5	20	pen 3 van U4 (D1)	L1	5	pen 13 van U3
NC	19		L0	4	pen 14 van U3
L9	18	Afkomstig van mini-mapper print punt L9	D0	3	pen 2 van U3 (D4)
L8	17	Afkomstig van mini-mapper print punt L8	#CAS	2	Afkomstig van #CAS selectieprint punt P7
D4	16	pen 2 van U4 (D0)	Vcc	1	pen 9 van U3

Tabel 4

(Advertentie)

Sunrise for MSX presenteert

Schitterend spel volgens *Nightmare* principe. Baan u een weg door de monsters uit te schakelen. Het spel heeft een perfecte ondersteuning van de MoonSound en FM-PAC (K). Het is ten opzichte van de previewversie grondig aangepast. Kortom een spel dat garant staat voor urenlang spelplezier.

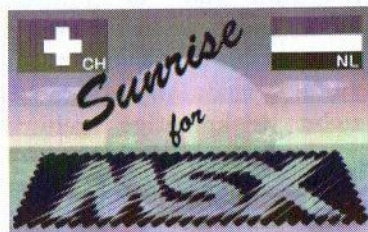
Gamesabbonement prijs: Fl. 27,50

Losse verkoop: Fl. 35,-

Verzendkosten: Fl. 7,50

Wilt u geen verzendkosten betalen en toch het spel snel hebben, bezoek dan onze stand op de beurs in OSS op 20 januari 2001. U kunt het verschuldigde bedrag overmaken op gironummer 48994 t.n.v.:

Stichting Sunrise
Postbus 61054
2506 AB Den Haag



UNIVERSELE PC GELUIDSKAART INTERFACE?

Wij krijgen af en toe brieven van lezers binnen, die een bepaald probleem aan de kaak willen stellen. Eén van deze brieven gaat over de productiestop van de MoonSound en het leek ons voor de hardwaretechnenuten interessant om eens over het volgende na te denken.

Sunrise verkoopt al sinds 1995 een spracht van een geluidskaat voor de MSX computer, de welbekende MoonSound met de OPL4 chip. Door een productiestop van de OPL4 chip bij Yamaha kan Sunrise straks geen nieuwe MoonSound kaarten meer maken. De kans dat er voor de MSX computer nog een nieuwe geluidskaat wordt gemaakt is bijzonder klein. De afzetmarkt wordt elk jaar kleiner en kleiner. Waarom zouden we weer het wiel opnieuw gaan uitvinden door een nieuwe geluidskaat te ontwerpen? Laten wij onze energie in een universele interface stoppen waarmee het mogelijk is om ISA en PCI PC geluidskarten aan te sturen.

Om de MSX computer niet of zo min mogelijk te belasten moet de interface een eigen krachtige 16 of 32 bits processor krijgen. Er kan dan bijvoorbeeld gedacht worden aan een RISC processor met een standaard 72 pins SIMM geheugen en een flashrom voor de interface bios. Eerst moet er over een programmeerstandaard worden nagedacht met een modulair en flexibel concept, zodat toekomstige geluidskarten ook kunnen worden gebruikt.

Uiteraard moet er een veelzijdig en

modulair muziek programma worden geschreven, dat zich aanpast aan de mogelijkheden van de aangesloten geluidskaat. De driver bepaalt wat de mogelijkheden van het muziekprogramma zijn. Standaard moet er vanuit worden gegaan van minimale basismogelijkheden waaraan een geluidskaat moet voldoen. Er moet dus een opwaarts compatible standaard komen. Om te beginnen kan de meest gangbare en populaire geluidskaat worden genomen die op dit moment te koop is. Muziekstukken die voor een 128 kanaals wavetablekaart zijn geschreven zouden ook met een 64 kanaals kaart overweg moeten kunnen. Dit kan verwezenlijkt worden door het aanhouden van een aantal basis kanalen. Daarbuiten kunnen er zoveel mogelijk steunkanalen worden toegevoegd als wenselijk is.

Ook moet er een replayer komen waarmee het mogelijk is om gemaakte muziek te kunnen gebruiken binnen eigen programma's. Verder kan er gedacht worden aan een 4,16,32 kanaals MOD player, een MIDI player, een MP3 player en natuurlijk een MoonSound replayer. Zelfs de MIDI poorten van de soundkaart kunnen worden aangestuurd.



Dit alles behoort tot de mogelijkheden mits de gekozen processor krachtig genoeg is. Het enige wat de MSX doet is data door sturen naar de interface. Er zullen ongetwijfeld nog wel meer mogelijkheden zijn maar dit is wel een heel leuk begin.

Ik hoop dat dit idee een goed alternatief is voor de toekomst en dat ie-

mand de uitdaging aan durft te gaan. Hou het binnen het bereik van de MSX gebruiker want buiten de aanschaf van de interface moet er ook nog een geluidskaart worden aangeschaft. Mensen die hierover met mij van gedachten willen wisselen kunnen mij bellen via telefoon: (06) 55 55 02 03

Alex Peetoom

(Advertentie)

Indien u geïnteresseerd bent in één of meer van genoemde hard- en/of software, dan kan op de volgende manieren contact met ons opgenomen worden:

MSX-Info Blad
P/a M.T. Stoker
Rotterdamstraat 73
6415 AV Heerlen
Telefoon: (045) 572 95 09
BBS: (045) 570 87 63
E-mail: m.t.stoker@hccnet.nl

Product	Omschrijving	Prijs
LPE-EVOLUCION4-V1	slotexpander 4+2 (4Xe&4Pi&6Di 8/16/32 bits) incl. kabel	Fl. 390,—
LPE-CPN-1	1 meter flat cable voor EVOLUCION4	Fl. 75,—
LPE-Z380-V2	Z380 32 bits cartridge (exclusief SIMM RAM)	Fl. 415,—
REF-51	LP SCSI interface met MSX-DOS2 (maximaal 8 aansluitingen)	Fl. 225,—
LPE-CPT-0.5	0,5 meter twisted cable voor EVOLUCION4	Fl. 125,—
LPE-GSA-V1	Advanced Slot Generator kaart (voor oudere expander versies)	Fl. 215,—
LPE-1M-V3	1MB RAM uitbreiding met Mapper	informeer
LPE-4M-V1	4MB RAM uitbreiding met Mapper	Fl. 160,—
LPE-PCK-V2	PC Keyboard Interface (turbo R compatible)	Fl. 150,—
LPE-PRG-V2	EPROM brander + software	Fl. 175,—
LPE-PRGSW-V2	EPROM memory grabber	Fl. 215,—
LPE-FORSW-V4	Programma om formulieren mee te maken	Fl. 25,—
LPE-PSGM-V1	PS2 muiscartridge	Fl. 125,—



Moondop #2

In een eerder *MSX-Info Blad* is deel 1 van de *Moondop* serie al besproken. Toen ging het om een muziekdisk voor de MoonSound, die door enkele Spaanse heren is gemaakt. Het nieuwe exemplaar is gelijksoortig en we zijn dan ook benieuwd naar de muziek op dit tweede deel.

De opzet van de disk en de intro van editie #2 zijn vrijwel identiek aan zijn voorganger. Eerst wordt een logo getoond, dat vergezeld wordt door een introductiemuziekje. Na de gebruikelijke druk op de spatiebalk wordt het menu geladen. Dit menu is ook bijna hetzelfde opgebouwd als zijn voorganger. Er worden, naast een Japans anime plaatje, negen titels van muziekstukken weergegeven. De songs kunnen met gebruik van de cursortoetsen en de spatiebalk geselecteerd worden. Gelukkig is het probleem van de naar boven springende cursor van deel 1 opgelost. Op de diskette zelf zijn overigens twaalf muziekstukken aanwezig. Na aftrek van het intro en het "uitro" muziekje is er dus nog één nummer 'zoek'.

Het belangrijkste van een muziekdisk is altijd de muziek zelf. Over *Moondop #1* heb ik geschreven, dat er weinig en ook wat korte muziek op stond. Naar mijn mening is dat met #2 ook het geval. De muziek is niet slecht maar ook niet briljant. Sommige muziekstukken zijn hier en

daar misschien wat kinderlijk, want er zitten erg vreemde (lees: onsmakelijke) samples in verwerkt. Eén van de muziekstukken, *Bach song*, is hooguit twintig seconden lang en wordt tot in het oneindige herhaald. Dat wordt na een minuut al behoorlijk irritant met als gevolg, dat er snel naar het volgende muziekje wordt gesprongen. Dit blijkt dan gelijk het langste en beste nummer van de disk te zijn: een cover van de *Ys II intro*.

De exit functie activeert het einde van de disk: een gedigitaliseerde foto van de componist *Manuel Dopico* waarnaast in het Spaans enkele pagina's tekst worden afgedrukt. Ik ben de Spaanse taal niet machtig maar kan wel uit de tekst opmaken, dat er op de disk nog een aantal BASIC files aanwezig zijn waarmee verdere informatie over de disk opgevraagd kan worden. Deze moeten vanuit BASIC geladen worden.

Conclusie

Ik krijg een beetje het gevoel dat deze disk wat snel in elkaar is gezet. Er zijn vrijwel geen verbeteringen

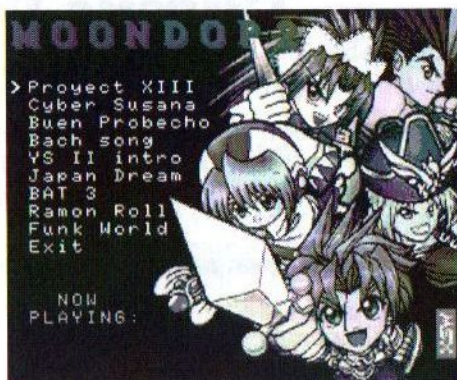


ten opzichte van het eerste deel. Ik denk dat *Manuel* beter werk kan afleveren als hij wat meer tijd besteedt aan de kwaliteit. Deze disk is alleen voor de MoonSound bezitters, die graag iets nieuws voor deze geluidskaart willen hebben. Als je een muzikliefhebber bent, die op zoek is naar goede muziek dan kan beter naar een andere disk worden uitgekoken. Voor meer informatie kan contact worden opgenomen met *MSX Club Hnostar* uit Spanje of met de makers zelf:

Moai-Tech
Santiago Rusinol 51 4-1
CP08304 Mataro (BCN)
Spanje
E-mail: mdopicor@olemail.com

Tristan Zondag

Makers: Moai Tech
Uitgebracht: 1999
Soort: Muziekdisk
Muziekchip: MoonSound
Prijs: 500 Pesetas
Medium: 1 x DD diskette



Nieuwe serie FM-PACs

Regelmatig krijgen wij op beurzen de vraag gesteld of wij nog aan een FM-PAC kunnen komen. Tot op heden konden wij niet meer doen, dan uitzien naar een tweedehandsje. Onlangs hoorden wij echter, dat *Leonardo Padial* in samenwerking met *MSX Club Hnostar* bezig is een nieuwe serie FM-PAC kaarten te maken. Deze cartridges zullen uitgerust worden met een zogenaamde true stereo 0.3W+0.3W versterker. Meer gegevens zijn op dit moment nog niet bekend. Wij hebben een richtprijs gekregen van rond de 120 gulden inclusief verzendkosten. Degenen die in één of meerdere exemplaren geïnteresseerd zijn, worden verzocht contact op te nemen met de uitgever van dit blad. Bij voldoende vraag zullen wij dan enkele kaarten naar Nederland halen.



Cursus Powerbasic

- deel zes -

Deze keer een voorbeeld van het gebruik van de VDP-commando's door het vullen van de juiste registers. Een overzichtje van de V9990 registers is niet geheel overbodig, maar daar is vast wel ergens aan te komen.

In de voorbeelden wordt weer gebruik gemaakt van listing 1 van 'Cursus PowerBasic deel 5'. Er zit echter nog een klein foutje in deze listing. In regel 9080 moet $C<64$ worden vervangen door $C<32$. Dan is het ook mogelijk om 32.768 kleuren te selecteren, iets wat we dus maar gelijk in de praktijk zullen brengen in het eerste voorbeeld.

Blokken

In BASIC kan met `line(x1,y1)-(x2,y2),k,bf` een gevulde rechthoek getekend worden. BASIC tekent zelf niets, maar stuurt het commando door naar de MSX2 videochip. De V9990 heeft zo'n commando ook (*Yamaha* noemt het LMMV) en wij gaan dat hier gebruiken door direct een paar registers te vullen. Om het vullen van de registers te vergemakkelijken heb ik met `DEFFN` een paar functies gedefinieerd. Het eerdergenoemde `line`-commando wordt dan: `A$=USR1(FNBL$(x1,y1,x2-x1+1,y2-y1+1,k,12))`. `x1` en `y1` geven altijd de hoek linksboven aan. Daarna moeten de breedte en de hoogte opgegeven worden, gevolgd door de kleur en de logische operatie. Op de V9990 zijn meer logische operaties mogelijk dan op de MSX2-videochip,

maar de meeste zijn praktisch waardeveloos. In tabel 1 staan de meest bruikbare, niet toevallig dezelfde die op MSX2 al mogelijk waren. Elke waarde van 0 tot en met 15 kan gebruikt worden als logische operatie. De kleurregisters zitten zo in elkaar dat voor 16, 8, 4 en 2-bits kleuren verschillende `FNBL$()`-functies gedefinieerd moeten worden.

In voorbeeld 1 wordt een schermresolutie van 256 x 212 ingesteld met 32.678 mogelijke kleuren (regel 30). De bedoeling is alle 32.768 kleuren in beeld te krijgen. In regel 70 wordt het onderste deel van het scherm gewist. Door voor logische operatie 0 te kiezen, is het opgeven van een kleur eigenlijk overbodig, omdat alle doelbits op 0 worden gezet. Voor logische operatie 15 geldt iets vergelijkbaars, maar dan worden alle doelbits op 1 gezet. In regel 90 worden 256 lijnen naast elkaar gezet. Het trekken van rechte lijnen doen we natuurlijk ook met `FNBL$`. Daarna worden er in regel 100 met `OR` horizontaal een paar (128) lijnen over de getekende lijnen gezet, met zulke kleurnummers dat na de `OR` operatie 256 x 128 = 32.768 kleuren in beeld staan. In de regels 130 en 140 worden hier willekeurig



met XOR wat blokken overheen getekend.

Voorbeeld 1 heeft de routines nodig uit listing 1 van 'Cursus PowerBasic deel 5'. Vergeet niet om regel 9080 te verbeteren! Nu gaat het opbouwen van het scherm niet geweldig snel. Dit ligt gelukkig niet aan de Graphics 9000 en ook niet aan mijn routines. BASIC is gewoon vrij traag met het opbouwen van een string door middel van DEFFN-functies. Zeker niet als die ook nog eens 'genest' zijn (DEFFN functies die andere DEFFN functies gebruiken). Dit is te verhelpen door bijvoorbeeld veel voorkomende strings te onthouden zodat ze niet steeds opnieuw opgebouwd hoeven te worden. De `USR1`- en `USR2`-routines veranderen niets aan de invoer-string, `A$` blijft dus ongewijzigd na `A$=USR1(A$)`. Als de regels uit listing 1 worden toegevoegd aan voorbeeld 1, dan gaat het trekken van de lijnen een stuk sneller, omdat het grootste deel van de string al vastligt. Het is op deze manier echter ook ingewikkelder om de string juist te vullen.

ByteMove

Doordat *PowerBasic* compatibel is met de MSX videochip, ondersteunt deze niet alle mogelijkheden van de V9990. De V9990 kan ook reeksen bytes verplaatsen en niet slechts rechthoekige delen schermdata. Ook kan een reeks bytes omgezet worden naar een rechthoekige vorm en andersom. Voordelen zijn onder andere dat lineaire stukken data compacter kunnen worden opgeslagen, zodat de

beschikbare geheugenruimte beter kan worden benut, en dat schermdata makkelijker aan een andere resolutie kan worden aangepast.

In voorbeeld 2 wordt de functie `FNXL$()` gedefinieerd, zodat met `A$=USR1(FNXL$(ad,dx,dy,nx,ny,arg))` een lineair stuk omgezet kan worden naar een rechthoekig stuk. `ad` is het adres waar het lineaire stuk staat. Deze functie kan alleen voor de eerste 64K gebruikt worden. In `dx` en `dy` komen de doelcoördinaten, in `nx` en `ny` de breedte en de hoogte. In `arg` komt het argument. Het argument geeft de richting aan (zie tabel 2).

Voorbeeld 2 heeft ook de routines nodig uit listing 1 van 'Cursus PowerBasic deel 5'. In regel 100 wordt een `SCREEN 8`-plaatje ingeladen en dat wordt verplaatst naar de Graphics 9000. Voor een `SCREEN 12` plaatje kan er in regel 30 weer `C=0` gebruikt worden in plaats van `C=256`. Het verplaatsen gebeurt met `A=USR(0)`.

Het plaatje wordt altijd vanaf adres 0 in het geheugen geschreven, waarbij er geen rekening wordt gehouden met de scherm breedte (of beter: met de image space). In de regels 110-140 wordt het plaatje vier keer gekopieerd. In regel 150 wordt er een scroll register ingesteld, zodat de schermopbouw bij coördinaat 128 begint.

Sjoerd Mastijn



Tabel 1*Logische operaties*

XOR 6
 AND 8
 PSET 12
 OR 14
 TXOR 22
 TAND 24
 TPSET 28

Tabel 2*Richtingsargumenten*

Linksboven → rechtsonder 0
 Rechtsboven → linksonder 4
 Linksonder → rechtsboven 8
 Rechtsonder → linksboven 12

Voorbeeld 1

```
10 ' Deel zes Voorbeeld een (geen PowerBasic)
20 DEFINT A-Z: GOSUB 9160: ' Lees ML in
30 X=256: XI=256: IL=0: C=32: GOSUB 9070: ' Kies scherm
40 DEFFN P$(X)=CHR$(XMOD256)+CHR$(X\256): DEFFN C$(X,Y)=FNP$(X)+FNP$(Y)
50 DEFFN A$(L)=" "+CHR$(L)+CHR$(255)+CHR$(255): N$=CHR$(0)
60 DEFFN BL$(DX,DY,NX,NY,K,L)="$"+FNC$(DX,DY)+FNC$(NX,NY)+FNA$(L)+FNP$(K)
+"-- "
70 A$=USR1(FNBL$(0,128,256,128,0,0)): ' Wis onderste deel van het scherm
80 ' Vertikale lijnen (PSET=12)
90 FOR I=0 TO 255: A$=USR1(FNBL$(I,0,1,128,I,12)): NEXT
100 ' Horizontale lijnen met OR (14)
110 FORI=0 TO 127: A$=USR1(FNBL$(0,I,256,1,I*256,14)): NEXT
120 ' 'willekeurig' wat blokjes tekenen met XOR (6)
130 X=RND(1)*255: Y=RND(1)*211: NX=RND(1)*(256-X): NY=RND(1)*(212-Y)
140 A$=USR1(FNBL$(X,Y,NX,NY,I,6)): I=(I+1)MOD32767!: GOTO130
150 END
```

Listing 1

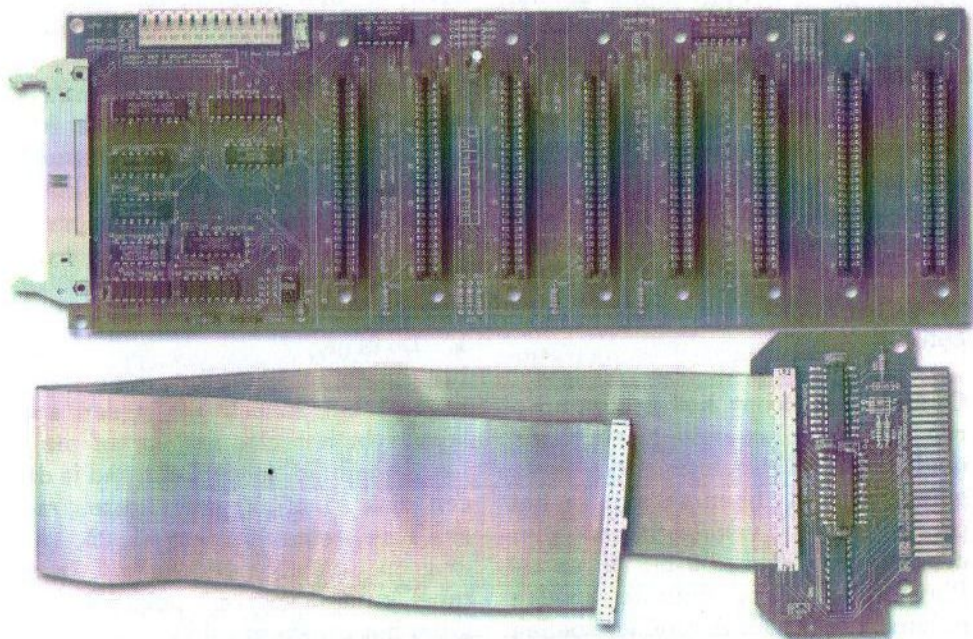
```
85 I$=N$+N$+N$+CHR$(1)+N$+CHR$(128)+N$+FNA$(12)
90 FOR I=0 TO 255: A$=USR1("$"+CHR$(I)+I$+CHR$(I)+N$+"-- "): NEXT
105 I$=N$+N$+CHR$(1)+CHR$(1)+N$+FNA$(14)+CHR$(0): N$=N$+N$
110 FOR I=0 TO 127: A$=USR1("$"+N$+CHR$(I)+I$+CHR$(I)+"-- "): NEXT
```

Voorbeeld 2

```
10 ' Deel zes Voorbeeld twee (geen PowerBasic)
20 DEFINT A-Z: GOSUB 9160: ' Lees ML in
30 X=512: XI=512: IL=3: C=256: GOSUB 9070: ' Kies scherm
40 DEFFN P$(X)=CHR$(XMOD256)+CHR$(X\256): DEFFN C$(X,Y)=FNP$(X)+FNP$(Y)
50 DEFFN A$(A,L)=CHR$(A)+CHR$(L)+CHR$(255)+CHR$(255)
60 DEFFN D$(A,P)=CHR$(AMOD256)+" "+CHR$(A\256)+CHR$(P)
70 DEFFN XL$(SA,DX,DY,NX,NY,A)=" "+FND$(SA,0)+FNC$(DX,DY)+FNC$(NX,NY)+FNA$(A,12)+"--- "+CHR$(128)
80 FILES: PRINT: PRINT "Geef de naam van het plaatje: ";
90 LINE INPUT N$
100 SCREEN 8: BLOAD N$,S: A=USR(0)
110 A$=USR1(FNXL$(0,0,128,256,212,0))
120 A$=USR1(FNXL$(0,511,128,256,212,4))
130 A$=USR1(FNXL$(0,0,551,256,212,8))
140 A$=USR1(FNXL$(0,511,551,256,212,12))
150 A$=USR1(CHR$(17)+CHR$(128))
160 END
```



De Sunrise 8 slotexpander nu weer verkrijgbaar!



Met 4 sub-slots voor algemeen gebruik, waaronder IDE, SCSI, SCC, FM-Pac en gamecartridges, en 4 I/O slots voor I/O-only kaarten, zoals Graphics9000, Video9000, MoonSound, en Sunrise RS232C

Nu voor slechts Fl. 299,-

(Fl. 314,- inclusief verzendkosten)
(Verzendkosten bij rembours: Fl. 20,-)

Sunrise for MSX
Postbus 61054
2506 AB Den Haag
Telefoon: (070) 360 97 07
(tussen 22:00 en 24:00 uur)
E-mail: rhiep@msx.ch
Internet: <http://www.msx.ch>



Gamesabbonnement?!

Enkele jaren geleden werd door *Sunrise* het Gamesabbonnement opgezet om de ontwikkeling van MSX spelsoftware in gang te houden. De laatste jaren loopt onder MSX-ers de belangstelling voor het abbonnement echter terug. *Rob Hiep* doet een hernieuwde poging om de voordelen van het abbonnement duidelijk te maken.

Het meest belangrijke motief voor het opzetten van het Gamesabbonnement destijds was dat MSX alleen kan overleven als er goede en betaalbare spellen voor worden uitgebracht. Wij zijn er trots op om het voortouw hiertoe genomen te hebben.

Ondertussen loopt het programmeren in de MSX-wereld sterk terug, omdat programmeergroepen meer geïnteresseerd lijken te zijn in het programmeren van software voor computersystemen als de PC en de Playstation. Onze duurste spellen halen de prijs van dergelijke spellen niet. Wij maken spellen voor het plezier. Als we echter niet slagen in onze missie om programmeurs en potentiële kopers te interesseren dan vrezen wij dat MSX ten dode is opgeschreven.

Abbonnement versus normale spellen

De systeemeisen voor spellen die via het abbonnement worden verkocht zijn minimaal, namelijk een MSX2 met minstens 128 kB RAM geheugen. Dit betekent echter niet, dat we geen andere spellen verkopen. Sommige spellen vereisen echter extra hardware, zoals bijvoorbeeld een MSX2+ of turbo R. Hierdoor ver-

koopt het spel minder goed en kan het niet via het Gamesabbonnement worden verkocht. Het topspel *Sonyc* is hier een goed voorbeeld van.

Wat voor voordelen heeft het Gamesabbonnement?

1. De prijs.

In de tabel kunt u voor elk spel zien hoeveel het scheelt als u een spel koopt via het abbonnement of als u het gewoon los koopt. Leidraad is dat een disk Fl. 27,50 gaat kosten met een luxe verpakking en Fl. 25,- als er geen luxe doos omheen zit. In de tabel kunt u voor elk spel zien hoe groot het prijsverschil is.

De prijzen zijn exclusief verzendkosten. In geval van aankoop van het spel in een luxe doos bedraagt deze Fl. 7,50 en zonder doos Fl. 5,-. Uitzonderingen gelden voor speciale aanbiedingen en prijswijzigingen.

2. Informatieverstrekking

Middels onze website <http://www.msx.ch> en via brieven met aangehechte acceptgiro zullen we proberen om u op de hoogte te brengen van de nieuwste spellen. Ook is het raadzaam om de (disk) magazines te blijven lezen, omdat hierin vaak aankondigingen worden gedaan.



3. Minimale systeemeisen

Door minimale systeemeisen te stellen waaraan een spel moet voldoen kunnen we een groter publiek bereiken. Bijkomend voordeel is, dat we hierdoor kostenbesparend te werk kunnen gaan.

Zijn er geen nadelen?

Het abonnement wordt aangegaan voor minimaal twee spellen. Daarna betaalt u per spel (in Nederland) of meer tegelijk in het buitenland, want dat drukt de verzendkosten. U

bent daarna vrij om op te zeggen per spel. Overslaan van een spel kunnen we alleen toestaan als er keuze is uit meer dan één spel tegelijk. Draag uw steentje bij aan de promotie en instandhouding van het MSX systeem en wordt (weer) lid! Dit kan door een briefkaartje te sturen naar:

Sunrise for MSX
Postbus 61054
2506 AB Den Haag

Rob Hiep

Spel	Normale prijs	Abonneeprijs
The Shrines of Enigma	Fl. 17,50	Fl. 15,—
Akin	Fl. 25,—	Fl. 22,50
Pumpkin Adventure III	Fl. 30,—	Fl. 25,—
Match Maniac	Fl. 12,50	Fl. 10,—
Pentaro Odyssey	Fl. 25,—	Fl. 20,—
The Lost World	Fl. 40,—	Fl. 36,—
Bebopbout	Fl. 32,50	Fl. 25,—
Nuts	Fl. 32,50	Fl. 25,—
Realms of Adventure	Fl. 25,—	Fl. 17,50
Pentaro Odyssey 2	Fl. 32,50	Fl. 25,—
The Blue Warrior	Fl. 35,—	Fl. 27,50

Distributie LPE hardware

Leonardo Padial Electronics heeft bekend gemaakt, dat MSX Club Hnostar toestemming heeft gekregen om als zijn vertegenwoordiger in Spanje te gaan optreden. Dit betekent dat geïnteresseerden in Spanje niet langer contact op kunnen nemen met Rafael Corrales om hardware van LPE te bestellen. MSX-Info Blad en Hnostar zullen op dit gebied dus gaan samenwerken. Naast MSX-Info Blad en Hnostar is het natuurlijk ook mogelijk om hardware direct bij Padial te bestellen. Het is echter aan te raden dit via MSX-Info Blad of Hnostar te doen, aangezien er dan korting kan worden verleend.



Nieuwe 8 bits slotexpander

Op de HCC beurs in Utrecht was traditiegetrouw ook de MSX gebruikersgroep aanwezig. Op deze stand presenteerde *Hans Oranje* zijn nieuwste slotexpander en veel (ex) MSX-ers hebben dan ook een werkend exemplaar kunnen bewonderen.

Naast de 8 slotexpander van *Sunrise* en de *Evolucion4* expander van *Leonardo Padial* kan een potentiële koper van een slotexpander nu dus ook terecht bij *Hans Oranje*. Net als de *Evolucion4* bevat deze expander zes slots, waarvan er vier standaard MSX slots zijn en twee IO slots.

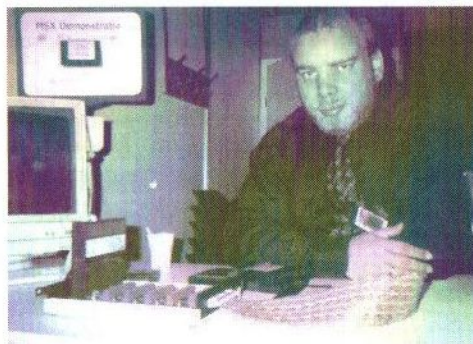
Deze expander is nog helemaal gericht op de huidige MSX generatie en kan alleen 8 bits cartridges aan. Dit in tegenstelling tot de eerder genoemde expander van *Padial*, die ook 16 bits en zelfs 32 bits kaarten aankan. Deze nieuwste telg uit de hardwarekeuken van *Oranje* is volledig MSX compatible, werkt vanaf een MSX2 en hoger en kan vrijwel alle modules en cartridges probleemloos aan. Zelfs met geheugenuitbreidingen schijnt deze expander geen moeite te hebben. Dat laatste kan

helaas niet van alle slotexpanders gezegd worden.

De expander wordt zonder behuizing geleverd. De aansluitkabel is echter wel voorzien van een connector, die keurig via een *Konami* kastje aan de MSX kan worden gekoppeld. Er is geen externe voeding nodig, maar degenen die de expander in een PC kast willen bouwen kunnen de voeding van deze kast op de expander aansluiten. Er moeten dan wel enkele jumpers verzet worden.

Voor Fl. 250,- exclusief verzendkosten kunt u in bezit komen van deze slotexpander. Neem hiervoor even contact op met *Hans Oranje* via e-mail: msx@hansoranje.demon.nl of op sprak tussen 19.00 uur en 21.00 uur van maandag tot en met vrijdag op telefoonnummer (0182) 61 98 32.

Gerrit van den Berg



Ontwikkelingen in Japan

Op verzoek van *Kuniji Ikeda* zet *Rob Hiep* van *Sunrise* uiteen wat er recentelijk in Japan aan MSX nieuws naar buiten is gekomen.

Op de spectaculaire *Denyuland* beurs in Japan was naast prominente MSX groepen tevens *Kay Nishi* als vertegenwoordiger van *ASCII*, de ontwikkelaar van de MSX-BIOS, aanwezig. Hij heeft op de beurs twee plannen naar voren gebracht:

1. Men wil een emulator uitbrengen om MSX op een andere computer te kunnen draaien. Uiteraard zal die emulator iets speciaals moeten zijn om zich te kunnen meten met de bestaande emulatoren.
2. Daarna wil men een nieuwe chip gaan ontwikkelen, wat men een 'all-in one' chip wil gaan noemen. Veel componenten die in de huidige MSX zitten zullen in één chip worden samengevoegd. Daarbij zal ook nieuwe hardware worden toegevoegd zoals een snellere processor, een MPEG-chip en meer geheugen. Men wil dit MSX3 gaan noemen.

ASCII gaat deze computer echter niet zelf ontwerpen. Daartoe is voornog geen interesse. MSX-ers over de hele wereld zullen *ASCII* of een ander organisatie daartoe moeten overhalen. In Japan zijn de voorvechters *ESE Artists factory*, *Kay Nishi*, en *Yokoi*. Ook is medewerking gevraagd van *Sunrise for MSX* en aan *Leonardo Padial* en *Agony* uit

Spanje. Na het bezoek van onze Japanse vrienden in Tilburg is via de MSX-mailinglist een aantal tegenstrijdige berichten verspreid voor wat betreft de zogenaamde desinteresse van *Stichting Sunrise* en *Leonardo Padial*. *Koen van Hartingsveldt* en ikzelf weten alleen wat *Kuniji Ikeda* die dag tegen ons heeft gezegd. *Ikeda* vroeg ons wat wij dachten over een nieuwe MSX computer zonder in detail te vertellen wat de plannen zijn. *Koen* vroeg *Ikeda* toen om één en ander te bespreken onder het avondeten. Geen van de Japanners kwam echter tijdens het avondeten op dit onderwerp terug. Dit heeft bij hen de indruk gewekt, dat we niet geïnteresseerd zouden zijn. Ik heb ze na veel heen en weer gemail kunnen overtuigen, dat we wel degelijk geïnteresseerd zijn.

Wij zijn bezig met nieuwe ontwikkelingen zoals een netwerkkaart en een VGA-aansluiting voor de MSX. Dit zijn natuurlijk maar kleine ontwikkelingen maar kunnen uiterst interessant zijn om toegepast te worden in een eventuele nieuwe computer. Natuurlijk is het nog maar de vraag of er een nieuwe computer komt. Veel zal afhangen van het vinden van een geldschieter. *Sunrise for MSX* kan daarin niet voorzien maar is wel bereid om op andere manieren haar steentje bij te dragen.

Rob Hiep



Het programmeren van 3D-games

- Slot -

In het vorige nummer is het één en ander verteld over de (on)mogelijkheden van het programmeren van een "First Person Perspective" 3D game op de MSX. In dit tweede en tevens laatste deel zal wat uitvoeriger worden stilgestaan bij de datastructuur van een dergelijk spel en hoe deze zo goed mogelijk geoptimaliseerd kan worden.

Natuurlijk is het mogelijk om alle benodigde data in allerlei arrays te stoppen, maar het kan zeker geen kwaad een goede structuur in de data aan te brengen. In het vorige deel is verteld dat voor een ruimte met n muren, $n + 1$ coördinaten nodig zijn om te vertellen waar de muren beginnen en eindigen. Wanneer alle punten in willekeurige volgorde in een array worden opgeslagen zal een lijst moeten worden gebruikt om te vertellen welke punten bij welke muur horen. Er kan echter meer structuur in de data worden aangebracht door het datamodel dusdanig op te zetten, dat elke punt en het direct erop volgende punt respectievelijk het begin- en eindpunt van de muur definiëren. Hierdoor wordt het tekenen van de ruimte op het scherm veel eenvoudiger en sneller, omdat we nu kunnen volstaan met een eenvoudige FOR...NEXT- lus.

Grootte van de kaart

Wanneer een kaart van het speelveld op 8-bits niveau wordt beschreven betekent dat, dat de kaart uit maximaal 256×256 punten kan bestaan. Als er geen schalingsfactor wordt gebruikt om de kaartcoördina-

ten naar schermcoördinaten om te rekenen, zal de uiteindelijke kaart niet groot genoeg zijn om ook echt interessant te kunnen zijn voor de speler. Wanneer op 16-bits niveau wordt gewerkt zal de kaart uit maximaal 65.536×65.536 punten kunnen bestaan en dat is in principe groot genoeg om ook interessant te kunnen zijn voor de speler. Echter moet nu wel met 16-bits wiskunde worden gewerkt en wordt de rotatiefunctie behoorlijk ingewikkeld. Bovendien wordt de kans dat de MSX een "memory overflow" geeft erg reëel.

Daarom is het een beter idee om niet met slechts één kaart te werken, maar iedere ruimte zijn eigen kaart en coördinatenstelsel te geven. Hierdoor kan gewoon op 8-bits niveau worden gerekend en kan voor iedere ruimte dezelfde omrekenfactor worden gebruikt om de kaartcoördinaten om te rekenen naar schermcoördinaten. Het betekent wel dat voor elke portal niet alleen gedefinieerd moet worden waar deze naar toe leidt, maar ook hoe het coördinatenstelsel van de andere ruimte vertaald kan worden naar die waarin de speler zich op dat moment bevindt. Op deze



manier ontstaat er dus een relatief coördinatenstelsel, waarin de coördinaten van een ruimte afhangen van die waarin de speler zich bevindt. Wel moet ervoor gezorgd worden dat bij de rotaties en/of translaties geen punten ontstaan die boven de maximale (8-bits) waarde uitkomen. Hoewel elke ruimte wordt beschreven door een 8-bits coördinatenstelsel kan de uiteindelijk kaart vele malen groter zijn en makkelijk uitgroeien tot 32 x 64-bits groot (dat is 4,3 miljard bij 18,4 triljoen pixels).

Hoogte in de kaart

Om alles zo eenvoudig mogelijk te houden en zo veel mogelijk geheugenruimte te besparen, is in deel 1 gesteld dat de hoogte van de vloer en het plafond in de ruimte constant is. Deze hoogte wordt gebruikt bij het tekenen van de ruimte op het scherm. Hoewel het renderen van een vloer en/of plafond met hoogteverschillen erin niet erg ingewikkeld is, zouden de routines voor de bewegingen van de speler wel veel ingewikkelder worden. Ook zal clipping lastiger worden: het is immers veel gemakkelijker en sneller uit te voeren als er geen rekening hoeft te worden gehouden met de hoogte van iedere afzonderlijke muur. Er kan worden volstaan met het uitlezen van de globale hoogte.

Er zijn twee opties om de hoogte van elke ruimte op te slaan:

1. Als een absolute waarde, waardoor alle ruimtes een hoogte ten opzichte van een vast referentiepunt, bijvoorbeeld zeeniveau, krijgen

2. Als een relatieve waarde ten opzichte van de ruimte waar de portal naartoe leidt.

De eerste methode is simpel en ook de routine om de hoogte naar het scherm te vertalen is betrekkelijk eenvoudig. Bovendien is de kans op vergissingen veel kleiner, doordat de programmeur vrij snel kan controleren of de hoogte goed is uitgelezen en goed is vertaald naar het scherm.

Als door portals meerdere ruimtes tegelijkertijd zichtbaar zijn, moet volgens de tweede methode bij het berekenen van de uiteindelijke hoogte van een ruimte, alle relatieve hoogtes van de ruimtes ertussen bij elkaar worden opgeteld.

Welke methode er ook wordt gekozen, het blijft altijd mogelijk om ruimtes met verschillende niveaus te creëren. Bij de eerste methode is het mogelijk om twee verschillende ruimtes elkaar te laten overlappen. Aangezien je alleen die ruimte op het scherm tekent waarin de speler zich bevindt en die kamers die zichtbaar zijn via portals, merk je daar tijdens het spelen niets van. Met de tweede methode is het zelfs mogelijk om de constant klimmende trap van *Escher* te creëren, doordat de trap beschouwt kan worden als een verzameling (kleine) ruimtes, waarbij elke ruimte relatief hoger ligt dan de ruimte ervoor. Het creëren van een lift is dan ook vrij eenvoudig: middels een subroutine wordt de hoogte van de vloer van de ruimte waarin de speler zich bevindt, veranderd.



Laten we als voorbeeld een gang nemen die is opgebouwd uit drie kleinere ruimtes. Ruimte A heeft een lage vloer, ruimte C een hoge vloer en ruimte B zal dienst doen als lift. De subroutine verandert de hoogte van de vloer van ruimte B en zodra de vloer ongeveer de hoogte van die van ruimte A heeft, kan van ruimte A naar B worden gegaan. Zo lang de vloer van ruimte B hoger is dan in ruimte A, is dat niet mogelijk. Voor de portal van ruimte B naar C geldt hetzelfde. Overigens is het altijd mogelijk om van C naar B en van B naar A te gaan (je kunt je altijd laten vallen naar een lager gelegen ruimte).

Een iets andere methode om een lift te realiseren is door de hoogte waarop het plafond zich bevindt te veranderen. De speler stapt in een kleine ruimte, waarna een subroutine de plafondhoogte verandert. Zodra de plafondhoogte van de andere ruimte is bereikt kan de speler door de portal naar de andere ruimte. Natuurlijk kun je dit zo verfraaien als je zelf wilt en kun je bijvoorbeeld schuifdeuren creëren, zoals in de vele science fiction series te zien is.

Rekensnelheid

Door op 8-bits niveau te werken zal de snelheid van vermenigvuldigingsroutines veel sneller zijn dan wanneer er op 16-bits niveau wordt gewerkt. Een rotatie kan met behulp van de onderstaande formules worden berekend:

$$\begin{aligned}x' &= x \cos(a) - y \sin(a) \\ y' &= x \sin(a) + y \cos(a)\end{aligned}$$

Aangezien x en y 8-bits waarden zijn kunnen we voor de sinustabel een fixed point lijst gebruiken. Wanneer deze lijst waarden tussen -1 en 1 moet bevatten, moet er gewerkt worden met 2.6 fixed waardes. Dit wil niets anders zeggen dan dat databits 0 tot en met 5 informatie over alles na de komma bevatten, terwijl in bits 6 en 7 alles voor de komma wordt opgeslagen. Zo wordt het getal 0.5 dan 00100000; het getal -0.75 wordt dan geschreven als 1101000. Wanneer we een 8-bit "signed" versus 8-bit "unsigned" tabel gebruiken waarvan het resultaat in HL komt, kunnen we door een vrij eenvoudige schuifinstructie bepalen wat er voor de komma staat: we hoeven slechts twee plaatsen te schuiven (add HL, HL; add HL, HL).

De snelheid van berekeningen kan worden verhoogd door de ruimte eerst om zijn eigen, lokale centrum te roteren en vervolgens een translatie in het nieuwe coördinatenstelsel uit te voeren. Hierdoor kunnen 8-bits "add"-instructies worden gebruikt en hoeft er niet met ingewikkelder 16-bits wiskunde gerekend te worden. Door impliciete data te gebruiken kan de rekensnelheid nog verder verhoogd worden.

Wanneer ervoor wordt gekozen om alle ruimtes in het punt (0,0) te laten beginnen, hoeft dit punt niet in de berekening van rotaties mee te worden genomen (het levert toch altijd hetzelfde punt (0,0) op). Als gesteld wordt dat alle andere punten positief zijn ten opzichte van het beginpunt

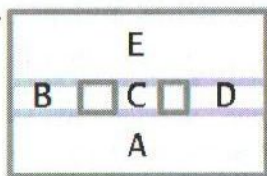


(0,0), kunnen de routines waarin vermenigvuldigingen worden uitgevoerd verder geoptimaliseerd worden. We weten immers dat er altijd een positieve met een "signed" 6 decimalen fractie wordt geroeteerd.

Met het oog op het laatste punt zou het wel handig zijn als alle ruimtes rechthoeken zijn. Wanneer er bijvoorbeeld vijfhoekige kamers worden ontworpen is het niet meer mogelijk om alle coördinaten positief te laten zijn ten opzichte van het startpunt.

Voorkom dubbele berekeningen

De rekensnelheid kan dus verhoogd worden door dingen die niet (her) berekend hoeven worden ook daadwerkelijk niet te berekenen. Het weglaten van het startpunt (0,0) bij rotaties is een stap in de goede richting, maar er kan veel verder worden



Wanneer ruimte E wordt 'gerenderd', worden berekeningen drie maal uitgevoerd: eerst door de portal van B naar E, dan door de portal van C naar E, en tenslotte nog eens via de portal van D naar E. Telkens als de speler zich verplaatst worden berekeningen voor ruimte E dus drie maal uitgevoerd. Aangezien het berekenen van rotaties behoorlijk processorintensief is, is het zaak om een oplossing te bedenken om niet telkens dezelfde berekeningen uit hoeven te voeren.

Een goede oplossing is het aanmaken van een cache, waarin de berekende rotatiecoördinaten worden opgeslagen. Het tekenen van een ruimte op het scherm kan dan behoorlijk worden versneld door de coördinaten uit de cache te lezen. Vooral eenvoudige voorwaartse en achterwaartse bewegingen kunnen snel naar het scherm worden vertaald. De eenvoudigste manier om een dergelijke cache aan te maken is door een dusdanige structuur in de data aan te brengen, dat de berekende rotatiecoördinaten direct naast de niet-rotatiecoördinaten kunnen worden opgeslagen. Echter brengt cachegeheugen wel een klein probleempje met zich mee: is de informatie die in de cache is opgeslagen nog steeds geldig?

Dat probleem kan worden opgelost door ook de kijkrichting van de speler als een variabele op te slaan. Het is dan eenvoudig te bepalen of de cache nog steeds geldige informatie bevat door naar de kijkrichting van de speler te kijken. Als de huidige kijkrichting dezelfde is als die in de cache is opgeslagen, bevat de cache nog steeds geldige informatie.

Recursie

In het gebruikte datamodel overlappen muren elkaar niet en kunnen ruimtes door recursief tekenen op het scherm worden gezet. Als de muur een portal bevat wordt bij het tekenen de portal gevolgd en wordt begonnen met het op het scherm zetten van de ruimte waar de portal naartoe leidt. En daar schuilt wel een klein addertje onder het gras.

Laten we als voorbeeld een speelveld nemen dat uit slechts twee ruimtes A en B bestaat. Ruimte A heeft een portal naar ruimte B en in ruimte B bevindt zich een portal naar ruimte A. Wanneer de voorgestelde methode klakkeloos wordt gevolgd, gebeurt het volgende:

We bevinden ons in ruimte A en dus wordt begonnen met het tekenen van de muren van ruimte A. In één van de muren komen we een portal naar B tegen, dus wordt begonnen met het tekenen van ruimte B. De muren van ruimte B worden getekend totdat een portal wordt tegengekomen. We komen een portal tegen, dus wordt deze gevolgd en wordt begonnen met het tekenen van die ruimte. We zijn echter wel weer in ruimte A! Muren van de ruimte worden getekend totdat we een portal tegen komen. Deze wordt gevonden, gevolgd, waardoor we ons weer in ruimte B bevinden, etc!

Dit is dus een eindeloze lus en het spreekt voor zich dat dat niet de bedoeling is. De gemakkelijkste manier om dit probleem te voorkomen is een limiet te stellen aan het maximale aantal portals dat wordt ge-'traced' en dit bij te houden door een teller. Op zich is dit gemakkelijk te programmeren, maar deze methode zal te langzaam zijn om echt bruikbaar te zijn. Een andere én bruikbaarere methode is om bij het tekenen van een ruimte gebruik te maken van een zogeheten referentiecode in de datastructuur. Wanneer de ruimte is "gerenderd" wordt de referentiecode op 1 gezet en voordat een nieuwe

ruimte wordt gerenderd wordt eerst de referentiecode uitgelezen. Is deze gelijk aan 1 dan is de ruimte al gedaan en wordt deze dus niet nog eens gerenderd. Ook dit is eenvoudig te implementeren in het programma en zal een hoop tijd besparen, maar heeft wel als nadeel dat veel van de verfraaiingen zoals voorgesteld in het vorige nummer van *MSX-Info Blad* niet meer mogelijk zijn. Zo kunnen meerdere portals niet als teleporters in dezelfde ruimte uitkomen. Ook de teleporter met preview is vrijwel niet meer te realiseren.

Dat zou jammer zijn en daarom is de beste oplossing om muren alleen te tekenen als er naar de voorzijde van de muur wordt gekeken. Wordt er tegen de achterzijde aangekeken dan wordt de muur niet getekend. Deze methode maakt in principe gebruik van normaalvectoren en wordt in veel PC spellen toegepast. Een normaalvector geeft de oriëntatie van een vlak weer en er kunnen vectorberekeningen mee worden uitgevoerd. De achterliggende en vrij ingewikkelde wiskunde wil ik hier niet uitleggen. Daarvoor kan beter een wiskundeboek worden geraadpleegd.

De normaalvector van het vlak waarin de muur zich bevindt kan worden berekend uit de informatie over de begin- en eindcoördinaten en vervolgens wordt een vectorvermenigvuldiging uitgevoerd met de vector van de kijkrichting van de speler. Dit levert een waarde van de cosinus van de hoek tussen beide vectoren op. Als de uitkomst negatief is, wordt naar de achterkant van de



muur gekeken; is deze positief dan wordt tegen de voorzijde van de muur aangekeken. De waarde van de cosinus wordt in veel PC spellen gebruikt om de mate van lichtreflectie te bepalen en daarmee ook hoe een muur wordt ingekleurd. Negatieve waarden betekenen dan dat de muur een donkere kleur (schaduw) krijgt. Op de huidige PC's kunnen normaalvectoren tijdens het spelen worden berekend zonder dat dat voor de speler merkbaar is. De MSX is daar veel te langzaam voor. Natuurlijk kunnen we alle normaalvectoren ook in de datastructuur als variabele opslaan, maar eigenlijk is dat niet echt nodig. In tegenstelling tot de PC spellen zijn wij niet echt geïnteresseerd in de mate van lichtreflectie, maar willen we alleen antwoord op de vraag: *Is de hoek tussen de normaalvector en kijkrichting zó dat de muur zichtbaar is?*

Bij negatieve waarden is dat niet het geval en wordt de muur niet getekend, bij positieve waarden wel. Het komt er dus op neer om een formule te vinden waarmee we kunnen bepalen of een muur zichtbaar is of niet. Deze kan worden gevonden in de vorm van:

ABS (hoek kijkrichting speler - hoek van de normaalvector muur) < 90°

Hierdoor kan volstaan worden met het opnemen van de hoek van de normaalvector als variabele in het datamodel. De formule heeft echter wel een storend effect op het tekenen van muren op het scherm. Wanneer de uitkomst van de formule precies

90° is wordt een muur afgebeeld als een lijn op het scherm. Wanneer de hoek tussen beide vectoren groter is dan 90° wordt de muur niet afgebeeld, maar er wordt dan over het hoofd gezien dat punten die verder van de speler af liggen dichterbij het centrum worden afgebeeld. Hoewel de hoek tussen de kijkrichting en de normaalvector dan groter is dan 90° kan die muur wel degelijk zichtbaar zijn. Daarom moet er een kleine veiligheidsmarge worden ingebouwd en wordt de formule:

ABS (hoek kijkrichting speler - hoek van de normaalvector muur) < (90 + veiligheidsmarge)°

Wiskundig gezien is dit niet helemaal correct (een verder weggelegen punt wordt anders op het scherm geprojecteerd dan een dichterbij gelegen punt, terwijl de normaalvector hetzelfde is), maar is misschien de beste methode om op de MSX te gebruiken. Overigens kan er nog steeds met een cache voor de berekeningen van rotaties worden gewerkt, omdat voor een bepaalde kijkrichting telkens dezelfde punten zichtbaar zijn...

Pseudo-code

Dit en het vorige deel worden samengevat door het geven van pseudo-C code. Ik wens iedereen die een 3D-game op de MSX wil programmeren veel plezier met het vertalen van de pseudo-code naar echte code...

Originele tekst: *David Heremans*
Bewerking: *Raymond de Heer*



```

type PLAYERDATA {
  byte X; // relatieve coördinaten t.o.v. de as huidige ruimte
  *room currentroom;
  byte viewingdirection;
  byte currentheight;
  byte weaponholding
  byte ammoleft
  ....
}
type WALLSIDE {
  byte X;
  byte Y;
  byte ROTATEDX;
  byte ROTATEDY;
  byte NORMALANGLE;
  byte ISVISIBLE
}
type WALLS{
  byte type; //0=portal en anders codering voor texture-map van de muur
  byte iswalkthrough;
  .. // extra info zoals het veranderen van de kleur van de muur
  .. // animatiestap als portal een schuifdeur is totdat de portal toe
  // gankelijk is
  *ROOM nextroom; // in geval van portal leidt dit naar volgende ruimte
  byte xtrans; // waardes te gebruiken om translate-berekeningen van
  // volgende ruimte in huidige
  byte ytrans; // coördinatenstelsel uit te voeren
}
type ROOM {
  byte nr_wallpoints;
  WALLSIDE POINTS[];
  WALLS ROOMSIDES[];
  byte rotation_angle;
  byte floorheight;
  byte ceilingheight;
  byte nr_enemy;
  ENEMYTYPE ENEMY[];
}
void Main(void){
  while (not game over){
    moveplayer();
    moveenemy();
    drawroom (player.currentroom,
              player.viewingdirection,
              player.X,
              player.Y,
              player.height,
              0,0,255,212
    );
    check collision between bullets and player;
    check collision between bullets and enemy;
    check collision between enemy and player;
  }
}
void drawroom (*ROOM room,
              byte viewdirection,
              byte xtrans,

```

```

byte ytrans,
byte ztrans,
byte clipx1,clipy1,clipx2,clipy2){
if (room.rotation_angle != viewdirection) {
    room.rotation_angle = viewdirection;
    calculate rotation of all points for which NORMALANGLE tells us they
    are visible, and update the ISVISIBLE;
}
// het coördinatenstelsel van het scherm loopt van -64 tot + 64 , nul
// is het centrum van het scherm dit is gedaan om een x-correctie
// afhankelijk van de schermdiepte van een object makkelijk te
// kunnen berekenen door middel van een eenvoudige vermenigvuldiging.
// Deze vermenigvuldigingsfactor is van te voren al berekend en
// opgeslagen in een tabel. Hierbij is de schermdiepte als index
// gebruikt.
screenx1=2*( (room.POINTS[0].ROTATEDX-xtrans)
    *lookuptable(room.POINTS[0].ROTATEDY-ytrans)
)+128;
// aangezien het eerste punt altijd (0,0) is, moet dit nog
// geoptimaliseerd worden
prevvisible=false;
for (byte i = 1 ; i<room.nr_wallpoints ; i++ ){
    next if !ISVISIBLE;
    screenx2=2*( (room.POINTS[i].ROTATEDX-xtrans)
        *lookuptable(room.POINTS[i].ROTATEDY-ytrans)
    )+128;
    if (prevvisible){
        if (
            (screenx1<clipx1) && (screenx2<clipx1) )
            || (screenx1>clipx2) && (screenx2>clipx2) )
        ) {
            // zodra muren volledig buiten de zichtbare array liggen kunnen
            // ze worden overgeslagen
        } else {
            clip if needed to part in visible range
            if (room.ROOMSIDES[i-1].type==0){
                drawroom (room.ROOMSIDES[i-1].nextroom,
                    room.ROOMSIDES[i-1].xtrans + xtrans,
                    room.ROOMSIDES[i-1].ytrans + ytrans,
                    ztrans,
                    screenx1,clipy1,screenx2,clipy2);
                draw effect, like bars or colorfilter, if needed
            } else {
                draw wall of type room.ROOMSIDES[i-1].type
            }
        }
        draw sprites
    }
}
screenx1=screenx2;
prevvisible=true;
}

```

Push em

TeddyWareZ is één van de weinige MSX groepen die nog regelmatig nieuwe software uitbrengt. Hun laatste product is een spel genaamd *Push'em*, dat op de beurs in Bussum werd verkocht. *Push'em* is een puzzelspel gebaseerd op het welbekende *Sokoban*, één van de eerste puzzelspellen die voor de computer verschenen.

Push'em wordt geleverd op een merkloze diskette met een kleurenlabel. Het label ziet er echter niet zo professioneel uit, doordat deze niet zo nauwkeurig uitgeknipt en op de diskette geplakt is. Er wordt geen handleiding op papier meegeleverd, maar er is een uitleg op de disk aanwezig in de vorm van een tekstbestand. De minimale systeemeisen voor dit spel zijn een MSX2 met 128 kB RAM / VRAM. Voor een beetje leuke muziek is het aan te raden om een SCC cartridge in één van de slots van de MSX te stoppen. Dit is echter niet noodzakelijk om het spel te kunnen spelen.

Beginmenu

Wanneer we het schijfje in de disk-drive stoppen, komt na niet al te lang laden het beginmenu tevoorschijn. Hierin bevinden zich vier keuzemogelijkheden, namelijk 'arcade', 'classic', 'options' en 'password'. Eerst maar eens kijken wat zoal de mogelijkheden zijn bij 'options'. Deze mogelijkheden stellen niet al te veel voor, want hier bevindt zich alleen een BGM-optie om alle SCC-muziekjes die het spel bevat te kunnen beluisteren en kan er gekozen worden tussen 50 of 60 Hz.

Het spel

De keuze 'arcade' zorgt ervoor dat met het spel wordt begonnen. Voor dat het eerste level begint, verschijnt er een bemoedigende tekst in beeld, namelijk 'Good Luck'. Dit zullen we ook zeker nodig hebben! Hierna wordt het speelveld zichtbaar. Linksboven in het scherm bevindt zich nu een paars rondvormig beestje met wel zeer grote ogen in verhouding tot de rest van zijn lichaam. Dit beestje komt me trouwens zeer bekend voor van een ander spel, namelijk *Pixess*. Verder zijn er in het veld diverse vierkante blokken verspreid. Het is nu de bedoeling om deze blokken te verplaatsen naar een daarvoor bestemd vakje. Dit kan door met het figuurtje tegen de blokken aan te lopen. Dit lijkt op het eerste gezicht niet zo'n moeilijke taak, maar dat zal naarmate je verder komt nog wel eens tegen kunnen vallen. Soms kan een op het eerste gezicht logische handeling ervoor zorgen dat een level niet meer uit te spelen is. Het is daarom ook van belang goed vooruit te kijken wat de diverse mogelijkheden zijn om het level tot een goed einde te brengen. Maar te lang denken is ook weer uit den boze, omdat er onderaan het



scherm een tijdbalk meeloopt. Wanneer deze balk tot nul is gereduceerd en nog niet alle blokken op de plaats van bestemming zijn gebracht, gaat er een leven verloren. In totaal beschik je over vijf levens. Als het op een gegeven moment, door het niet al te handig verschuiven van de blokken, niet meer mogelijk is om een level uit te spelen, kan met behulp van [F5] opnieuw worden begonnen met het huidige level. Bedenk dat hiermee wel een leven verloren gaat.

Passwords

In totaal bevat *Push'em* 50 verschillende levels. Om niet elke keer opnieuw te hoeven beginnen wanneer alle levens op zijn, is er de mogelijkheid om een password in te voeren. Dit is de laatste optie in het beginmenu. Na elke tien levels die zijn uitgespeeld wordt er een password gegeven. In totaal zijn er dus vier passwords.

Om het spel niet eentonig te maken, verandert tevens om de tien levels de achtergrond. Level één tot en met tien spelen zich af in de woestijn, waarbij je piramidevormige blokken



moet verschuiven. Verder komen in de volgende levels onder andere de thema's elektriciteit en computers aan de orde.

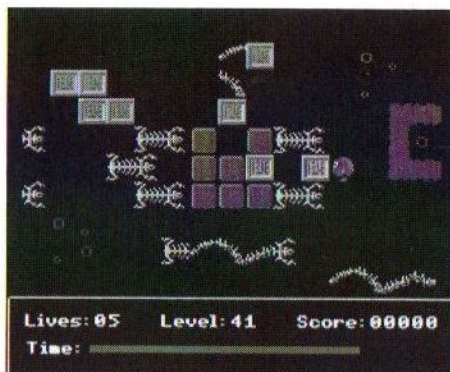
Minpuntjes

Over het algemeen is het grafische deel van het spel goed verzorgd, alleen de levels 31 tot en met 40 vind ik wat minder geslaagd. Het beeld is wat te donker en de verschillende onderdelen zijn lastig te onderscheiden. Verder is volgens mij een aantal levels niet uit te spelen binnen het gegeven tijdsbestek. Ik heb van alles geprobeerd om bijvoorbeeld level 22 uit te spelen, maar wat ik ook probeer, ik krijg het niet voor elkaar. Als iemand dit wel gelukt is, hoor ik het graag. De opbouw van de moeilijkheidsgraad is ook niet altijd even logisch. Een hoger level kan namelijk een stuk makkelijker uit te spelen zijn dan een lager level, al is een makkelijk level tussendoor natuurlijk ook wel eens lekker.

Wanneer het je uiteindelijk toch gelukt is om alle 50 levels uit te spelen en voldaan onderuit in de stoel zakt om de einddemo te bekijken, zal dit toch wel enigszins een teleurstelling zijn. Het enige wat namelijk tevoorschijn komt is een plaatje van een zonsondergang op zee. Hieronder verschijnt een tekstbalk met de credits en de tip om de *Sokoban* versie van dit spel te spelen. Toch wel een domper na al die inspanningen!

Sokoban classic

Een leuk extraatje krijgt men als in het beginmenu wordt gekozen voor 'classic'. Nu kan namelijk de klassie-



ke versie van *Sokoban* worden gespeeld. De beeldkwaliteit is natuurlijk een stuk minder, maar het blijft een leuk spelletje. Hierbij is er wel alle gelegenheid om een level goed uit te zoeken, omdat hier geen tijdslimiet aan verbonden is. Dat is trouwens maar goed ook, want hier zitten echt pittige levels tussen.

Conclusie

Push'em is een heel leuk puzzelspelletje. Het was voor mij weer een tijdje geleden, dat ik zo lang achter elkaar een spel op de MSX heb gespeeld. Grafisch ziet het spel er op een aantal levels na prima uit en ook de SCC-muziek is van goede kwaliteit, al vind ik wel dat de huidige spellen eigenlijk allemaal geschikt

moeten zijn om met de MoonSound te werken. Toch zijn er toch wat slordigheden te noemen, zoals het onzorgvuldig plakken van de etiketten. Tevens is er naar mijn mening een aantal velden te snel in elkaar gezet en niet goed getest op de haalbaarheid in de gestelde tijdslimiet.

Al met al is dit toch een leuk spel, dat het aanschaffen zeker waard is, gezien de redelijke prijs die ervoor gevraagd wordt. Ik ken vrij recente spellen die meer kosten en die van veel mindere kwaliteit zijn dan *Push'em*. Neem voor meer informatie contact op met:

Frederik Boelens
Jan Palachweg 17
9403 JS Assen
E-mail: fboelens@dds.nl

Jan-Marten van der Reest

Auteur: TeddyWareZ
Soort: spel
Genre: puzzel
Systeemeisen: MSX2 met 128 kB
Optioneel: SCC
Medium: 1 x DD-diskette
Prijs: Fl. 10,—

Te koop

- Philips NMS 8280 met 1 MB intern geheugen
- Philips NMS 8220 met toetsenbord
- MSX2 module
- NMS 1210 omgebouwd + 14k4 modem
- MT Telecom modem
- Philips 1431 matrixprinter
- Philips Grafic tablet (tekenbord)
- Circa 25 MSX boeken en handleidingen

- Circa 800 diskettes met spel, cliparts en dergelijke
- MT Base kaartenbak
- Game Master
- Penguin adventure
- Bubble Bobble
- Circa 500 MSX Bladen (MCCM, PTC en wat anderen)

Telefoon (010) 485 03 03 of (06) 24 22 67 52



MSX-NBNO organiseert:

MSX 2001 fair

20 januari 2001



't oude Theater
Bram v.d. Berghstraat
Oss



Direct tegenover het
centraal station



Ruim voldoende gratis
parkeergelegenheid

Voor meer informatie:

(0412) 64 06 79 Raymond

(0412) 63 52 87 Richard

www.msx4ever.demon.nl/msx-nbno

e-mail: msx2001@msx4ever.demon.nl



Sunrise for MSX presentent:

Blue Warrior

