BULLETIN

SINCLAIR GEBRUIKERSGROEP GRONINGEN/ASSEN

6e jaargang nummer 3 november '88.

COLOFON

VOORZITTER:
Jan Dirk Burggraaf
Kluivingskampenweg 30
9761 BP Eelde
☎ 05907-1697

SEKRETARIS:
Martin den Hollander
Numero Dertien 8
9644 TV Veendam
☎ 05978-45474

PENNINGMEESTER:
S.E. Kroon
Oosterhoutstraat 96
9401 NK Assen
☎ 05920-15912
Giro 5212298 t.n.v.
rekening SGG

VICE VOORZITTER/
PENNINGMEESTER:
J. van Alteren
De Grouw 6
9351 LP Leek
☎ 05945-15678

VERHUUR:
C. van Krimpen
Koldakker 34
9407 BM Assen
☎ 05920-70093

ALGEMEEN
Roelof Koning
Selwerderstraat 26
9717 GK Groningen
☎ 050-124298

REDAKTIE:
Mevr. F. Elstrodt
Kam. Onnesstraat 172
9727 HS Groningen
☎ 050-263930

Rudy Biesma
Betuwe 18
9405 JJ Assen
☎ 05920-50643

Het SGG-bulletin is een uitgave van de Sinclair Gebruikersgroep Groningen. Het bulletin verschijnt 10 keer per jaar.

Artikelen, listings of andere inzendingen zijn voor verantwoording van de inzender.

De sluitingsdatum voor kopij wordt in elk bulletin vermeld.

Het lidmaatschap van onze gebruikersgroep bedraagt f 17,50 per kalenderjaar voor personen tot en met 17 jaar voor oudere personen is dit f 25,00 per kalenderjaar. Bij deze prijs is het abonnement op het bulletin inbegrepen.

U kunt lid worden van de SGG door U op te geven bij de penningmeester.

SLUITINGSDATUM KOPIJ 26 NOVEMBER

VAN DE REDAKTIE

HALLO ALLEMAAL

Een van onze leden heeft gespeeld met de digitiser en het resultaat ziet U op onze voorplaat. Mooi he.
Voor meer informatie verwijs ik U naar Gebr. Gron. op pag 4.
Onze sluitingsdata voor copy wordt vervroegd, dit om rustiger te kunnen werken. Let dus even op.
Bij het sorteren van onze ruilabonnementen merkten wij dat we enkele bladen misten, dit is niet leuk en ook niet de bedoeling. Wij zien ze graag zo spoedig mogelijk terug, als U een blad wilt lenen om het thuis rustig door te lezen is een vraagje genoeg. Als dit teveel gevraagd is blijven ze in het vervolg thuis bij de redaktie.
Voor U in ram gelezen; "Masterfile voor pc", dus bent U het al op de spectrum gewend, met overstappen geen problemen.
Voor het december bul. kunnen we nog wel wat copy gebruiken wat op die maand slaat, ik denk aan muziek, beep of play. Een mooie tekening gemaakt met the artist of art studio .
Leuk we hebben er weer een medewerker bij, Henk Etten .
Deze keer geen adventures, Ramon heeft het te druk met school.

We hebben dit maal weer erg leuke copy, kijkt U maar.

_ Arie maakte voor ons de mooie voorplaat.
_ Onze voorzitter verteld ???
_ Martin legt ons een twee-dimensionale array uit.
_ Henk laat ons sectoren op disk bekijken en wijzigen.
_ De Red. maakte een kalender met de belangrijkste data.
_ Roelof stoeide voor ons met pixelscroll.
_ Han van Abbe kan nu nog sneller rekenen.
_ Frans speelde voor U het spel Beyond the ice palace.

Volgende keer een artikel over een nieuwe spectrum, we hadden geen tijd dit deze maand nog te vertalen.

Shri Devi

GEBRUIKERSAVOND GRONINGEN

We houden onze gebruikersavond 15 november voor deze maand, en 15 december is de volgende.
Kees wil deze maal met zijn camera en digitiser komen en het leek ons leuk om dan een demonstratie te geven van Art Studio.
U kunt zich dan ook laten screnen (noem je dat zo).
Om half acht gaat de zaal open en om tien uur beginnen we met het opruimen van onze spulletjes.
Afgelopen keer hielden we onze hardwaremarkt, en dat was een leuke avond. Vele spectrumspullen wisselden van eigenaar.
Aan het einde van de avond trad Kees weer op als veilingmeester dit tot groot plezier van toeschouwer en koper.
Het enthousiasme was groot, vanwege een goede ruil of aankoop.
Het adres van onze gebruikersavond is:

School "de Wijert"
van Schendelstraat 1
Groningen.

GEBRUIKERSAVOND ASSEN

De eerste maal in het Markehuus is goed verlopen, er werd enorm veel gedemonstreerd. Gebruikers van andere merken die kwamen kijken, en gebruikers die hun spectrum naar zolder hadden gebracht, omdat ze op een gegeven moment vast zaten keken hun ogen uit naar wat we allemaal aan onze spectrum hadden gekoppeld. Kees was hier met camera en digitiser aanwezig, dat was leuk want zo kon je b.v. een screen$ laten maken van je eigen profiel. Dan nog even wat MEDEDELINGEN:
We hebben een ander zaaltje toegewezen gekregen, eenmaal binnengekomen gaat U voor de bar links, en dan naar het einde van de gang. Het entree bedraagt f2,50 inclusief één comsumptie.
Omdat de redaktie telkens in tijdnood zit, en wij niet kunnen beloven dat ons bulletin altijd voor de gebruikersavond bij U in de bus ligt geven wij U hier alvast wat DATA. NOTEER DIT!!

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

DAN VOLGEN HIER WAT DATA VOOR ASSEN.
10 NOVEMBER, 8 DECEMBER, 12 JANUARI 1989, 9 FEBRUARI 1989 ?
een vraagteken i.v.m. de vakantie., 9 MAART, 13 APRIL.
Het zijn allemaal donderdagen.

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

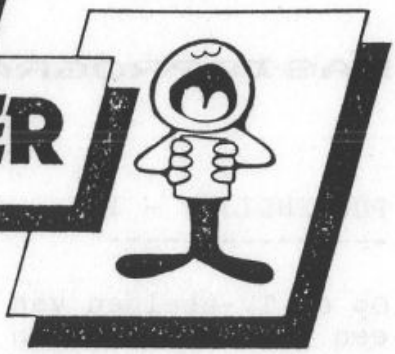
We houden onze gebruikersavond dus in :

Het Markehuus
Scharmbarg
Assen (Peelo = Wijk 7)

KOPIJ

Heeft U iets leuks voor ons bulletin, wilt U het ons dan op tijd d.w.z. zo vroeg mogelijk geven, zodat wij er iets van kunnen maken. Vaak hebben we een aardig idee maar dan ontbreekt ons de tijd er iets mee te doen. Natuurlijk mag U ook uw ideeën uiten en inspraak hebben. Denk nooit als U ons bulletin ontvangt het gaat goed want we hebben alweer een dik nummer, het ziet er leuk uit dus ze hebben mijn bijdrage niet nodig. En als U ontevreden bent over ons bulletin dan weet U wie daar schuldig aan is, juist Uzelf.

VAN DE VOORZITTER



Mijn vrouw moet altijd lachen als ik weer eens een telefoontje moet plegen. Vaak begin ik altijd met alle papieren na te kijken of ik het juiste nummer wel heb. Pak toch een telefoonboek wordt er dan geroepen.Mijn eer te na denkt dit persoontje dan. De stekker wordt in het stopkontakt gedaan. De computer begint te leven. Het bestand wordt ingeladen en dan begint de zoekactie. Feilloos weet het apparaat toch weer de betreffende nummers op het scherm te toveren.
Ik besef heel goed dat een computer niet voor telefoonnummers aangeschaft moet worden. Maar ja wat wil je als zo'n apparaat al tijden bij je thuis staat. Je gaat er allerlei toepassingen voor bedenken. Zo zijn er natuurlijk heel wat.
Ook het knutselen in en aan de computer, tenminste als je durft, geeft de nodige voldoening.
Maar als je niet oppast komt er een tijd dat er te veel tijd en geld in gaat zitten. Dat is dan het moment van bezinning. Altijd moet er voor gewaakt worden dat je hobby niet het hoofddoel gaat worden van je leven. Tenzij je tijd genoeg hebt of dat je van je hobby je beroep hebt gemaakt. Op de laatste clubavonden in Assen en Groningen heb ik weer de mooiste opstellingen gezien. In Assen, waar een open avond werd gehouden, bleek eens te meer dat er een heleboel mensen zijn die ooit een Spectrum hebben aangeschaft maar na enige tijd de computer naar zolder lieten verhuizen waar nu de muizen er piano op kunnen spelen. Toen ze echter het Markehuus binnen waren en langs de opgestelde apparatuur liepen, konden we toch vernemen dat men niet wist dat dit allemaal met de Spectrum uitgehaald kon worden. Een paar van deze personen denken er over om de zolder maar te gaan bezoeken, daar de Spectrum op te zoeken en weer aan de praat proberen te krijgen. Zo zien we het natuurlijk graag en wanneer deze mensen zich bij ons aanmelden zullen we er van alles aan doen om al de kennis die momenteel binnen de club is ook aan hun over te dragen.
Hopelijk zijn ze wel zo verstandig om gewoon een klein telefoonklappertje bij de hand te hebben als ze iemand willen bellen.

J.D. BURGGRAAF

BASICPROGRAMMA'S VOOR BEGINNERS 6.

PUNTENLIJST - 1

Op de TV-beelden van de Olympische Spelen zagen we regelmatig een puntentelling in beeld. Daarbij ging het erom dat een aantal personen (beoordelaars) punten gaf aan een aantal andere personen (deelnemers).
Op de fotoclub waar ik lid van ben, gebeurt iets dergelijks (het prestatieniveau ligt daar overigens iets lager dan in Seoul).
Op een clubavond kan iedereen foto's meebrengen. Per deelnemer worden maximaal drie foto's beoordeeld, waarvoor iedereen punten geeft van 1 tot 10 per foto. Een deelnemer kan dus maximaal 30 punten halen per avond. Vrijwel iedereen brengt foto's mee en iedereen doet mee aan de beoordeling, alleen schrijf je voor je eigen foto,s 0 punten.

Aan het eind van de avond heben we dus een aantal (x) lijstjes van beoordelaars waarop een even groot of iets kleiner aantal punten voor deelnemers (y) voorkomt.
Deze lijstjes zien er als volgt uit:

beoordelaar 1		
dln 1	klaas	26
dln 2	jaap	24
dln 3	ben	28
dln 4	arend	21
dln 5	piet	26
dln 6	gerrit	19

beoordelaar 2		
dln 1	klaas	24
dln 2	jaap	28
dln 3	ben	19
dln 4	arend	22
dln 5	piet	21
dln 6	gerrit	18

Van elke beoordelaar dus zo'n lijstje. De namen van de beoordelaars doen niet terzake zoals we later in het programma zullen zien. Alleen een volgnummer is voldoende.

TOEKENNEN VAN VARIABELEN

We hebben dus een (x) aantal lijstjes met elk een (y) aantal cijfers. Het gaat er nu om welke variabelen we aan deze cijfers zullen toekennen. We kunnen natuurlijk de variabele a toekennen aan het cijfer van deelnemer 1, gegeven door beoordelaar 1; de variabele b aan het cijfer van deelnemer 2, gegeven door beoordelaar 1; variabele c voor dlnr 3, enz.
Je voelt wel aan dat hier zeer veel variabelen nodig zijn, die aan het eind van de rit een onoverzichtelijke brei vormen, waaruit moeilijk te distilleren valt wat het totaal is van bijvoorbeeld variabele a + variabele g + variabele m, enz. zoals dat hier met 6 deelnemers het geval zou zijn.
Hiervoor bestaat een heel mooie oplossing, nl. de ARRAY. In dit geval een TWEE-DIMENSIONALE ARRAY.

NU NIET AFHAKEN

Zo, het hoge woord is er uit. Een twee-dimensionale array. Nu zijn er een aantal computeraars die op dit moment willen gaan afhaken, onder het motto: dit is voor mij niet weggelegd. LET, INPUT en desnoods INKEY$ is nog tot daar aan toe, maar ARRAY's is andere koek en dan nog twee- of driedimensionaal (of vier-, of vijf-, of zes-, enz); nou laat maar!
Ik ken die reaktie, want ik heb er zelf lang genoeg last van gehad. Toch nog even doorlezen graag, want als je er induikt blijkt het enorm mee te vallen, zoals met veel computerbegrippen trouwens.
Voor we verder gaan met het programma lijkt het me zinnig eerst uit te leggen wat een array is en wat je er mee doen kunt.
Je kunt een array op verschillende manieren bekijken en nu is merkwaardig dat wij (de mens) en dat ding (de computer) een array op heel verschillende manier zien.
Je hebt waarschijnlijk wel eens de definitie gehoord van een computer als een ding dat niet meer kan dan twee getallen bij elkaar optellen, maar dan wel ontzettend snel. Bijvoorbeeld 300.000 keer per sekonde of zo. Daarvoor is nodig dat de computer grote aantallen cijfers kan onthouden of anders gezegd: kan opbergen in het geheugen. Je weet dat het geheugen van een computer bestaat uit een lange rij vakjes, waarin getallen zijn opgeborgen. EEN rij getallen op volgorde van de eerste naar de laatste. Verder niks. Je kunt dit EEN-dimensionaal noemen, nl. van de eerste naar de laatste in één richting. Niks twee-dimensionaal of wat ook. Gewoon EEN rijtje cijfers.

ARRAY

Nu is dat precies datgene wat wij een array noemen. Een rijtje cijfers in één richting. Dat kan ook niet anders want anders zou een array niet in een computer kunnen. Array betekent trouwens rangorde of rij of reeks.
Een heel andere zaak is wat wij doen met dat rijtje cijfers. Dat heeft te maken met hoe wij deze cijfers willen rangschikken of hoe we ze wilen gebruiken. In het geval van bovenstaande lijstjes is dat een twee-dimensionale opstelling, als volgt:

	beoord.1	beoord.2
klaas	26	24
jaap	24	28
ben	28	19
arend	21	22
piet	26	21
gerrit	19	18

Je ziet dat we deze opstelling, deze ordening, deze array in twee richtingen kunnen lezen, nl:

per deelnemer van links naar rechts en
per beoordelaar van boven naar beneden

Het lijstje kan eindeloos naar rechts (meer beoordelaars) en naar beneden (meer deelnemers) worden uitgebreid. Twee richtingen dus en daarom TWEE-dimensionaal.
Een kenmerk is dat deze richtingen haaks op elkaar staan. Lengte en breedte is ook zo'n geval. Doe er nog hoogte bij onder 90 graden met de beide andere en je hebt een drie-dimensionale array. Doe er de tijd bij en je hebt 4 dimensies. Maar 5 of 6? Wat zou er dan nog bij moeten en hoe zou dat kunnen?

Echter wat we in de computer stoppen is een rijtje cijfers, in dit geval dus: 26 24 28 21 26 19 24 28 19 22 21 18.
Het geheugen van de computer kent maar één dimensie en dat is van de eerste beschikbare geheugenplaats in één richting langs het hele rijtje tot aan de laatste geheugenplaats.
Het twee-dimensionale (of meer-dimensionale) karakter zit in de manier waarop wij met de cijfers willen werken. Dan moeten we dus op een of andere manier kunnen vastleggen, waar de cijfers in de twee-dimensionele array thuishoren. Dat doen we door de variabelen niet alleen een naam, maar ook een soort volgnummer te geven d m v de uitdrukking DIM.

DIM (Dimension)
Met DIM kunnen we een willekeurig groot aantal variabelen allemaal dezelfde naam geven, maar dan voorzien van één of meer volgnummers. Een bijzonderheid is dat de naam van dit soort variabelen uitsluitend uit één letter mag bestaan. We doen dit bijv. als volgt: (1010 is een willekeurig regelnummer, C=cijfer)

```
1010 DIM C (12)
```

Regel 1010 van de listing introduceert hier 12 variabelen: C1, C2, C3, enz t/m C12 en we kunnen deze variabelen dus gebruiken om het bovenstaande rijtje van 12 cijfers vast te leggen.
In dit geval zijn er 6 deelnemers, dus we moeten bij elkaar tellen: C1 + C7, C2 + C8, C3 + C9, enz. Het is al beter dan het bij elkaar tellen van a + g, b + h, c + i, enz. maar echt handig is het nog niet. Dat klopt ook wel want met een volgnummer 1 t/m 12 kun je maar één rijtje aangeven, één-dimensionaal dus, terwijl we in de verwerking van de cijfers met twee dimensies te maken hebben. Dit verhelpen we nu heel eenvoudig door niet één, maar twee volgnummers te geven, als volgt:

```
1010 DIM C(2,6)
```

Binnen de computer is dit weer gewoon een rijtje van 12 cijfers, één-dimensionaal dus, maar door de volgnummers die we gebruiken, kunnen we zorgen dat de cijfers twéé-dimensionaal gepresenteerd en verwerkt worden.
We krijgen nu de volgende variabelen:

```
C1,1 C1,2 C1,3 C1,4 C1,5 C1,6 C2,1 C2,2 C2,3 C2,4 C2,5 C2,6
```

en je ziet dat er groepen cijfers ontstaan, gerelateerd aan beoordelaar 1 (C1,1 t/m C1,6), aan beoordelaar 2 (C2,1 t/m C2,6) enz.

LISTING
In de volgende listing wordt een twee-dimensionale array opgezet voor een willekeurig aantal beoordelaars en een willekeurig aantal deelnemers:

```
  3 REM Titel  "Telling/1"
500 REM AANVANGSGEGEVENS
510 INPUT "Hoeveel beoordelaars? ";x
520 INPUT "Hoeveel deelnemers? ";y
1000 REM PUNTEN INVOEREN
1010 DIM C(x,y)
1020 FOR b=1 TO x
1040 PRINT " ";x;" beoordelaars": PRINT " ";y;" deelnemers":
 PRINT '': PRINT "beoordelaar ";b;" :": PRINT
```

```
1050 FOR d=1 TO y
1060 INPUT ("Hoeveel punten voor dlnr ";d;" ?";),C(b,d)
1070 PRINT "deelnemer ";d;" : ";C(b,d);" punten"
1080 NEXT d
1090 PAUSE 200
1100 CLS
1110 NEXT b
1120 STOP
```

In REGELS 510 en 520 worden met de variabelen x en y het aantal beoordelaars en deelnemers aangegeven.
In REGEL 1010 wordt met DIM C(x,y) een array gedimensioneerd met precies voldoende ruimte voor x beoordelaars en y deelnemers.
REGEL 1020 en 1110 geven een FOR NEXT lus waar alle beoordelaars van 1 tot x worden doorlopen.
REGEL 1050 en 1080 geven binnen de eerste FOR NEXT lus een tweede FOR NEXT lus waar de deelnemers van 1 tot y worden doorlopen.
Het resultaat van deze twee lussen is dus dat je in regel 1060 achtereenvolgens alle cijfers van alle deelnemers gegeven door beoordelaar 1 kunt intoetsen, daarna alle cijfers van alle deelnemers gegeven door beoordelaar 2, enz t/m beoordelaar x.
REGEL 1040 zorgt ervoor dat je steeds kunt zien hoeveel beoordelaars en deelnemers er zijn en van welke beoordelaar je de cijfers aan het intoetsen bent.
REGELS 1060 en 1070 zorgen er voor dat je ziet welke deelnemer moet worden ingetoetst, terwijl het ingetoetste cijfer ter controle op het scherm gezet wordt.
REGEL 1090 maakt dat je 4 seconden de tijd hebt om te zien of het complete lijstje van een beoordelaar in orde is, voordat het scherm leeggemaakt wordt en je het lijstje van de volgende beoordelaar kunt intikken.
Na afloop van het programma kun je controleren of alle cijfers juist zijn ingetoetst d m v: PRINT C (3,5) of PRINT C (2,7) enz.

De volgende keer zullen we het programma zo uitbreiden dat de opgeslagen gegevens ook op het scherm gepresenteerd en met de printer worden afgedrukt.

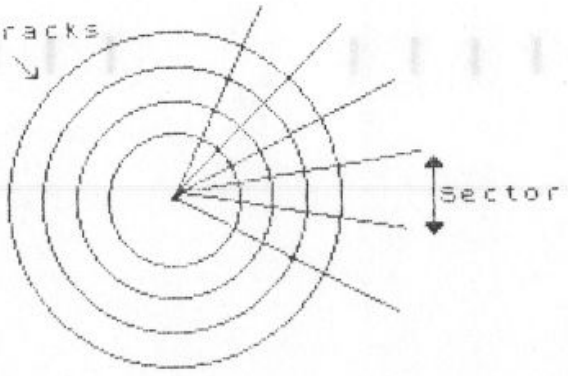
Martin den Hollander

Geachte SECTORLIST gebruiker

Het programma is ontworpen om elke sektor van de DISK te kunnen bekijken en eventueel te wijzigen.
De oplichtende cursor is te besturen met de cursortoetsen, als je een andere toets indrukt wordt deze toets op de plaats van de cursor gepoked.
Als je edit drukt kun je zelf een getal poken.
Het programma vraagt altijd of de gewijzigde sector gesaved moet worden.

```
  1 FOR k=60300 TO 60375
  2 READ a: POKE k,a
  3 NEXT k
  5 DATA 205,8,23,33,0,40,17,96,234,6,2,14,0,62,1,205,41,15
,24,24,0,201
  7 DATA 205,8,23,33,0,40,17,96,234,6,0,14,0,62,1,205,41,15
,205,72,23,201
  8 DATA 33,0,0,17,84,236,6,7,197,213,229,6,2,14,0,62,1,205
,41,15,225,35,209,20,193,16,237,205,72,23,201,0
 10 CLS : INPUT AT 20,0;"ENTER SECTOR No.";sector
 15 IF sector<0 OR sector>718 THEN GO TO 10
 17 LET c=sector
 18 PRINT AT 10,5;"LOADING SECTOR NO:";sector
 20 LET d=INT (c/256): LET c=c-(d*256)
 22 POKE 60304,c: POKE 60326,c: POKE 60305,d: POKE 60327,d
 25 RANDOMIZE USR 60300
 26 CLS : PRINT BRIGHT 1; INVERSE 1;" copyricht by h.etten
"
 27 LET a=60500: PRINT BRIGHT 1;"Disknaam:";: FOR k=a+6 TO
a+15: PRINT CHR$ PEEK k;: NEXT k
 28 PRINT : IF sector<7 THEN PRINT BRIGHT 1;"Behoort tot de
 CATALOGFILE": GO TO 34
 29 PRINT BRIGHT 1;"Behoort tot FILE:";
 30 LET z=PEEK (a+18)+256*PEEK (a+19): LET y=PEEK (a+20)+25
6*PEEK (a+21)
 31 IF PEEK (a+21)=255 OR PEEK (a+22)=229 THEN PRINT FLASH
1;"LEGE SECTOR": GO TO 34
 32 IF z<=sector AND y>=sector THEN FOR k=a+22 TO a+31: PRI
NT CHR$ PEEK k;: NEXT k: GO TO 34
 33 LET a=a+16: GO TO 30
 34 PRINT : PRINT AT 5,0; BRIGHT 1;"PRINT # IF CHR$ < 32";A
T 6,0;"PRINT "; FLASH 1;"#"; FLASH 0;" IF CHR$ > 165 "; INVE
RSE 1;"=KEYWORD": PRINT BRIGHT 1;AT 8,0;"LIST SECTOR NR:";"
"; INVERSE 1;sector: PRINT AT 10,20;" "
 35 FOR k=60000 TO 60256
 36 LET a$=CHR$ PEEK k
```

```
  37 IF a$<" " THEN PRINT "#";: NEXT k
  38 IF a$>="RND" THEN PRINT FLASH 1;"#";: NEXT k
  39 PRINT a$;: NEXT k
  44 LET x=11: LET xx=0: LET a=60000
  45 PRINT OVER 1; BRIGHT 1;AT x,xx;" "
  50 IF INKEY$=CHR$ 9 AND (a+1)<=60256 THEN LET xx=xx+1: PRI
NT OVER 1;AT x,xx-1;" ": LET a=a+1: IF xx>31 THEN LET xx=0:
LET x=x+1
  60 IF INKEY$=CHR$ 8 AND (a-1)>60000 THEN LET xx=xx-1: PRIN
T OVER 1;AT x,xx+1;" ": LET a=a-1: IF xx<0 THEN LET xx=31: L
ET x=x-1
  70 IF INKEY$=CHR$ 10 AND (a+32)<=60256 THEN LET x=x+1: LET
 a=a+32: PRINT OVER 1;AT x-1,xx;" "
  80 IF INKEY$=CHR$ 11 AND (a-32)>=60000 THEN LET x=x-1: LET
 a=a-32: PRINT OVER 1;AT x+1,xx;" "
  90 IF INKEY$=CHR$ 13 THEN GO TO 140
 100 IF INKEY$=CHR$ 12 THEN GO TO 10
 110 IF INKEY$>CHR$ 31 AND a<60257 THEN PRINT AT x,xx;INKEY$
: POKE a,CODE INKEY$: FOR h=1 TO 25: NEXT h: IF (a+1)<=60256
 THEN PRINT OVER 1;AT x,xx;" ": LET xx=xx+1: LET a=a+1: IF x
x>31 THEN LET xx=0: LET x=x+1
 115 IF INKEY$=CHR$ 7 THEN INPUT BRIGHT 1;" GEEF GETAL !!";m
: POKE a,m: IF PEEK a>32 AND PEEK a<165 THEN PRINT BRIGHT 1;
AT x,xx;CHR$ PEEK a: GO TO 45
 120 IF PEEK a<32 THEN PRINT #1; BRIGHT 1;AT 1,0;" POKE adre
s= ";a;",";PEEK a;" ";AT 2,0;" CHR$=";" besturing ": GO TO 4
5
 125 PRINT #1; BRIGHT 1;AT 1,0;" POKE adres=";a;",";PEEK a;"
 ";AT 2,0;" CHR$= ";CHR$ PEEK a;"            "
 130 GO TO 45
 140 PRINT #1;AT 1,0; BRIGHT 1;"  Save  sector j/n ?  "
 145 PRINT #1;AT 2,0;"                              "
 150 IF INKEY$="j" OR INKEY$="J" THEN GO TO 200
 160 IF INKEY$="n" OR INKEY$="N" THEN FOR h=1 TO 50: NEXT h:
 GO TO 180
 170 GO TO 150
 180 PRINT #1; BRIGHT 1;AT 1,0;" Gereed met deze sector j/n
? "
 185 IF INKEY$="j" OR INKEY$="J" THEN GO TO 10
 190 IF INKEY$="n" OR INKEY$="N" THEN GO TO 26
 195 GO TO 185
 200 RANDOMIZE USR 60322
 210 GO TO 10
9999 CLEAR : SAVE *1;"sectorlist" LINE 1
```



NOVEMBER 1988

ZO	-	6	13	20	27	-
MA	-	7	14	21	28	-
DI	1	8	15*	22	29	-
WO	2	9	16	23	30	-
DO	3	10	17	24	-	-
VR	4	11	18	25	-	-
ZA	5	12	19	26	-	-

DECEMBER 1988

ZO	-	4	11	18	25	-
MA	-	5	12	19	26	-
DI	-	6	13	20	27	-
WO	-	7	14	21	28	-
DO	1	8	15*	22	29	-
VR	2	9	16	23	30	-
ZA	3	10	17	24	31	-

Gebruikersavonden Groningen:
dinsdag 15 nov, donderdag 15 dec.
Gebruikersavonden Assen:
donderdag 10 nov, 8 dec, 12 jan, 9 febr.
Gebruikersbijeenkomsten Eemsmond:
zat 5 nov, 19 nov, 10 dec, 17 dec.

WAT EEN POPPEKAST

Rommelend tussen mijn tape-cassettes vond ik een oud (1985(!)) programmeer-experiment waar ik weliswaar nooit meer iets mee gedaan heb, maar dat toch te aardig is om bij andere oude koeien in de sloot te dumpen.

Het ging over 'pixelscroll' vanuit BASIC, en mijn idee was het volgende: Als karakters alleen op 'karakterposities' geprint kunnen worden en niet daar tussen in, waarom dan niet een nieuw karakter gemaakt waarvan de pixels binnen het karakter verschoven zijn, en deze op dezelfde positie geprint. We hebben immers U.D.G.s tot onze beschikking!

Nog beter zou het zijn om van te voren acht 'verschoven' karakters aan te maken in U.D.G.s, en deze te printen?

Snel een programmaatje in elkaar gedraaid:

```
  9 REM  maak UDGs
 10 LET q=256
 20 FOR f=USR "a" TO USR "a"+(8*8)-1 STEP 8
 30 LET q=INT (q/2)
 40 FOR g=0 TO 7: POKE f+g,q: NEXT g
 50 NEXT f
 99 REM  test
100 CLS : FOR f=144 TO 151: PRINT PAPER 4;CHR$f: NEXT f
200 FOR f=0 TO 7
210 PRINT AT 7,4;CHR$ (144+f)
215 PAUSE 2
220 NEXT f
230 GOTO 200
```

Dit werkt dus, en wel zo snel dat er een pauze nodig is om de zaken niet dubbel te gaan zien. Hoera!

Nu een poging om de volle schermbreedte te gebruiken, toevoegen:

```
300 LET q=0
310 FOR f=0 TO 7
320 PRINT AT 10,q;CHR$ (144+f)
330 NEXT f
340 LET q=q+1
350 IF q=32 THEN GOTO 300
360 GOTO 310
```

Dan GOTO 300, en ook dit werkt. Alleen laat het karakter een spoor na, in de vorige printpositie, wat niet de bedoeling is.

Een extra regel om dit te verhelpen:

```
335 PRINT AT 10,q;" ": REM 1 spatie
```

Het principe werkt dus, zoals je kunt zien wanneer je tot zover hebt meegetypt. Ook hebben we nu een indicatie van de snelheid. Maar helemaal zonder knipperen gaat het nog niet, en dit komt omdat de oude printpositie schoongemaakt wordt voordat er op de nieuwe positie is geprint. De tijdsduur tussen deze twee computerakties is blijkbaar zo lang dat dit met het oog waarneembaar is. De oplossing hiervoor zou zijn om het 'schone' karakter, (een spatie), meteen méé te printen met de U.D.G., en het liefst in één commando.

Tijd vond ik sowieso een belangrijke faktor, omdat een eventueel programma waarin dit principe toegepast zou worden, ook nog andere taken moest kunnen verrichten behalve het verzorgen van het scherm. Na enig geklooi met o.a. een stopwatch, (ik vond in een schrift nog zeven bladzijden met tijdmetingen!), kwam ik tot de volgende overwegingen: 1 FOR-NEXT lussen vermijden,
2 Rekenwerk zoals 'CHR$ (144 + f)' vermijden, en
3 De 'schone spatie' samen met de U.D.G. printen in één string.

Dit laatste bracht mij ook op het idee om de beweegrichting om te keren, zodat de printrichting van de computer (v.l.n.r.!) opties gezien (woordgrapje) in mijn voordeel zou werken.

En, wanneer ik toch elke keer twee karakters ging printen, dan kon de pixelverschuiving ook over twee karakters plaatsvinden. Hierdoor kunnen ook meteen bredere pixelpatronen 'gescrolld' worden in kleine stappen.

Een nieuw testprogramma, waarin ook vergeleken wordt met de 'standaard-BASIC-scroll':

```
 10 DATA 0,1,3,7,15,31,63,255
 15 LET d=0
 20 FOR f=1 TO 8
 30 READ a
 40 FOR g=0 TO 7
 50 POKE USR "a"+d+g,a
 60 POKE USR "a"+d+g+8,(255-a)
 70 NEXT g
 80 LET d=d+16
 90 NEXT f
 99 REM  test
100 LET q=30
110 PRINT AT 10,q;"AB": REM UDGs intypen!
111 PRINT AT 10,q;"CD"
112 PRINT AT 10,q;"EF"
113 PRINT AT 10,q;"GH"
114 PRINT AT 10,q;"IJ"
115 PRINT AT 10,q;"KL"
116 PRINT AT 10,q;"MN"
117 PRINT AT 10,q;"OP"
120 PRINT AT 8,q;"OP": REM vgl. zonder 'pixelscroll'
130 LET q=q-1
140 IF q=-1 THEN LET q=30
150 GOTO 110
```

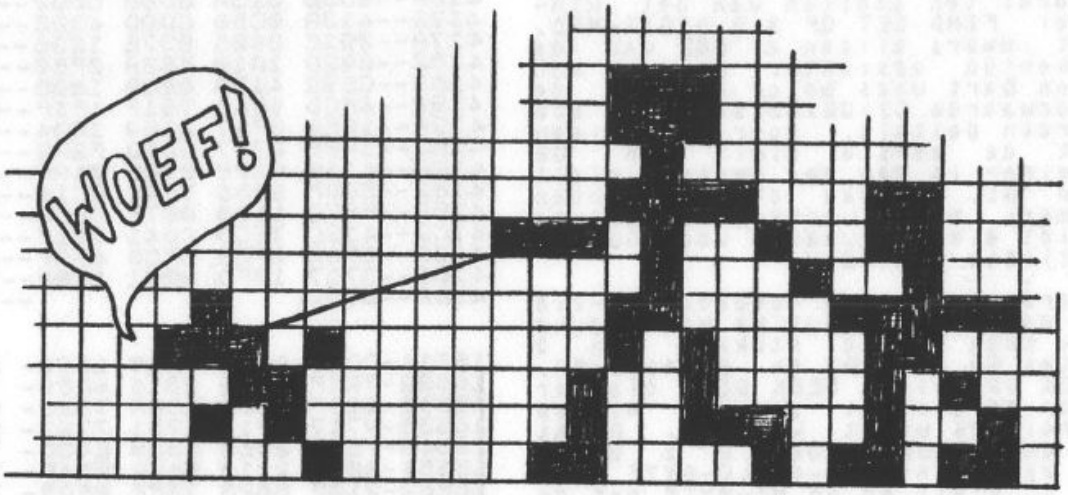
Deze 'scroll' ziet er toch keurig uit, mag ik wel zeggen! Ook wanneer het aantal printstappen verminderd wordt, bijv. tot vier, blijft het beeld acceptabel, en de snelheid wordt meteen groter.

Maar waarom zou ik een 'dood blok' over het scherm laten bewegen? Mijn volgende stap was dan ook (néé Martin den Hollander, geen autótje!) om een lopend mannetje te ontwerpen. Hierbij bleek dat het voor een redelijk geanimeerd beeld voldoende was om drie verschuivingen binnen een karakterpositie te doen, en dat de snelheid voldoende was om een heel regiment mannetjes te laten lopen.

Op dit punt aangekomen vond ik het opeens erg interessant om uit te zoeken wat er wel niet allemaal geanimeerd kon worden binnen één karakter, en hoeveel pixels er nodig zijn om een herkenbaar beeld van iets te krijgen. Je begrijpt het al, deze nieuwe uitdaging was zo groot dat ik het ontwerpidee voor een leuk programma aan de kant legde, en wegdroomde met ruitjespapier en een potlood. Om een lang verhaal kort te maken, er is nooit meer iets van gekomen, (van het programma, bedoel ik), en het enige wat ik hier kan presenteren is mijn laatste experiment:

```
 10 GO SUB 8000: GO TO 900
 99 REM  sub's , typ UDG's en tel de spaties!
100 PRINT AT 3,a;"A ";AT 3,b;"D  ";AT 3,c;"F  ";AT 0,d;
    "I  ": RETURN
110 PRINT AT 3,a;"B ";AT 3,b;"E ";AT 3,c;"G ";AT 0,d;
    "J ": RETURN
120 PRINT AT 3,a;"C ";AT 3,b;"H ";AT 3,c;"H ";AT 0,d;
    "K ": RETURN
130 LET a=a-1: IF a<0 THEN LET a=30
131 LET b=b-1: IF b<0 THEN LET b=30
132 LET c=c-1.85: IF c<0 THEN LET c=30
133 LET d=d-1.15: IF d<0 THEN LET d=30
140 RETURN
899 REM  init
900 BORDER 0: PAPER 0: BRIGHT 1: CLS
910 FOR f=0 TO 3
920 PRINT AT f,2;"                                ": REM 28!
930 NEXT f
940 INK 8: PAPER 8
950 LET a=30: LET b=27: LET c=24: LET d=20
960 LET p=8
999 REM  mainloop
1000 GO SUB 100
1010 PAUSE p
1100 GO SUB 110
1110 PAUSE p
1200 GO SUB 120
1210 PAUSE p
2000 GO SUB 130
2100 GO TO 1000
8000 REM  UDG's
8010 CLS : PRINT "wacht"
8020 RESTORE
8030 FOR f=0 TO 87: READ a: POKE USR "A"+f,a: NEXT f
8040 RETURN
```

Vergeet niet de DATAregels op de volgende bladzijde....




```
8200 DATA 6,6,2,31,6,6,9,27
8201 DATA 24,24,8,62,88,24,20,52
8202 DATA 48,48,80,56,180,240,16,48
8210 DATA 0,0,0,0,9,31,7,9
8211 DATA 0,0,0,0,17,62,14,9
8212 DATA 0,0,8,29,6,10,1,0
8213 DATA 0,0,0,0,20,60,14,16
8214 DATA 0,0,0,0,64,248,48,40
8220 DATA 0,0,4,15,2,0,0,0
8221 DATA 0,0,16,60,0,0,0,0
8222 DATA 0,0,32,240,64,0,0,0
```

Deze listing, hoewel enigszins gestroomlijnd en ontdaan van de ergste rafels, is zeer zeker niet het 'onderste uit de kan' voor wat betreft animatie in Sinclair-BASIC, maar dat was ook niet de bedoeling. Veel plezier ermee!

Tot zover dit kijkje in mijn 'keuken', ik hoop dat je het leuk vond.

Roelof Koning.

NOG EENS: "REKENEN MET DE ZX 81" (17)

De vorige keer eindigde mijn artikel met een oproep mij tijdsbesparende suggesties te sturen, vooral ten aanzien van het RTN-deel FIND SET OF 3 3 DIGIT-NRS, dat immers alleen al 80% van de rekentijd opslokte. Welnu mijn zoon Bart wees me er op, dat de voorwaarde G1>G2>G3 scherper kon worden gesteld, door te eisen dat de eerste digit van G2 kleiner is dan de eerste digit van G1. Gelijke cijfers mogen immers niet voorkomen! Evenzo geldt die voorwaarde voor G3 ten opzichte van G2.

Daarom heb ik nu toegevoegd -zie de machinecoderoutine van figuur 2B- twee gelijke stukken van 8 bytes AD 40CC/D3 en AD 4103/0A, SEEK G2 resp. SEEK G3, die er voor zorgen dat aan de nieuwe konditie wordt voldaan. Omdat in de G-TABLE iedere G 2 bytes in beslag neemt -de LO-BYTE met de 1. digit en de HI-BYTE met de 2. en 3. digit- moeten we steeds 2 adressen verder zoeken. En ook 2 adressen teruggaan, als de G2 resp. G3 gevonden is, die aan de voorwaarde voldoet. Gaande van een even adres (LO-BYTE) naar een oneven adres (HI-BYTE) in de G-TABLE kan H nooit van waarde veranderen. Vandaar dat INC L mag op AD 40CE resp. AD 4105. In overeenstemming hiermee zou je denken dat op AD 40D3 en AD 410A DEC HL moet staan, omdat daar immers van een oneven adres naar een even adres wordt gegaan. Aangezien echter bij geen enkele overgang naar een lager honderdtal van de G-waarde de MSB van het adres in de tabel verandert, is DEC L "gevaarloos".

FIG. 1B HEXDUMP REGEL 1 FIG. 1A

```
4082--CDE7 0221 1050 1609--256
408A--7A5F 42B8 2816 4AB9--314
4092--280F 78B9 280A 7323--230
409A--A717 1717 1781 7723--21E
40A2--7B0D 20EB 0520 E43D--2D9
40AA--20DF 2110 5422 FB4F--2F0
40B2--210F 5023 7EFE 06D8--2FD
40BA--CDBC 4128 4E23 46ED--399
40C2--430A 5022 F64F ED53--344
40CA--F14F 237E 2CB9 30FA--3F0
40D2--2D2D 237E FE04 3005--232
40DA--2AF6 4F18 D6CD BC41--427
40E2--3AF2 4FA2 20EC 3AF1--454
40EA--4FA3 20E6 ED4B F14F--470
40F2--EB09 22F3 4FEB 2B4E--3BC
40FA--2346 ED43 0C50 22F8--30F
4102--4F23 7E2C B930 FA2D--32C
410A--2D23 7EA7 2005 2AF8--28C
4112--4F18 BFCD BC41 3AF4--41E
411A--4FA2 20ED 3AF3 4FA3--41D
4122--20E7 2B4E 2346 ED43--319
412A--0E50 E5CD 5B41 7EA7--3D1
4132--2824 CDBC 410E 03CD--2F4
413A--C341 0D20 FA21 01FF--34C
4142--ED52 E120 C4E5 2100--40A
414A--50ED 5BF8 4F01 1000--2F3
4152--EDB0 ED53 FB4F E118--520
415A--B021 FF4F 1100 5001--281
4162--0A00 EDB0 2A0A 5022--24D
416A--0350 623A 0D50 CD92--2AB
4172--413A 0C50 CD9D 412E--2B0
417A--091E 0406 037E 1236--0FA
4182--002D 1D10 F83A 0F50--1E8
418A--CD92 413A 0E50 180B--25B
4192--4FCD 9B41 791F 1F1F--2CE
419A--1FE6 0F47 2E09 1E04--1B4
41A2--1A8E 2777 2D1D F2A2--324
41AA--4110 F12E 0106 04AF--22A
41B2--ED6F 2D86 772C 2C10--2EE
41BA--F6C9 1100 007E CDD0--3E8
41C2--412C 7ECD CD41 7E1F--363
41CA--1F1F 1FE6 0FC8 47AF--310
41D2--3717 10FD 3001 1C82--22A
41DA--57C9                --120

16514-CDE7 0221 1050 1609-  598
16522-7A5F 42B8 2816 4AB9-  788
16530-280F 78B9 280A 7323-  560
16538-A717 1717 1781 7723-  542
16546-7B0D 20EB 0520 E43D-  729
16554-20DF 2110 5422 FB4F-  752
16562-210F 5023 7EFE 06D8-  765
16570-CDBC 4128 4E23 46ED-  921
16578-430A 5022 F64F ED53-  836
16586-F14F 237E 2CB9 30FA-1008
16594-2D2D 237E FE04 3005-  562
16602-2AF6 4F18 D6CD BC41-1063
16610-3AF2 4FA2 20EC 3AF1-1108
16618-4FA3 20E6 ED4B F14F-1135
16626-EB09 22F3 4FEB 2B4E-  956
16634-2346 ED43 0C50 22F8-  783
16642-4F23 7E2C B930 FA2D-  812
16650-2D23 7EA7 2005 2AF8-  700
16658-4F18 BFCD BC41 3AF4-1054
16666-4FA2 20ED 3AF3 4FA3-1053
16674-20E7 2B4E 2346 ED43-  793
16682-0E50 E5CD 5B41 7EA7-  977
16690-2824 CDBC 410E 03CD-  756
16698-C341 0D20 FA21 01FF-  844
16706-ED52 E120 C4E5 2100-1034
16714-50ED 5BF8 4F01 1000-  755
16722-EDB0 ED53 FB4F E118-1312
16730-B021 FF4F 1100 5001-  641
16738-0A00 EDB0 2A0A 5022-  589
16746-0350 623A 0D50 CD92-  683
16754-413A 0C50 CD9D 412E-  688
16762-091E 0406 037E 1236-  250
16770-002D 1D10 F83A 0F50-  491
16778-CD92 413A 0E50 180B-  603
16786-4FCD 9B41 791F 1F1F-  718
16794-1FE6 0F47 2E09 1E04-  436
16802-1A8E 2777 2D1D F2A2-  804
16810-4110 F12E 0106 04AF-  554
16818-ED6F 2D86 772C 2C10-  750
16826-F6C9 1100 007E CDD0-1003
16834-412C 7ECD CD41 7E1F-  867
16842-1F1F 1FE6 0FC8 47AF-  784
16850-3717 10FD 3001 1C82-  554
16858-57C9                -  286
```

FIG. 28 MACHINECODE-RTN "ALL 9"

```
   281    MAKE TABLE 504 3-DIGIT-NUMBERS

          4082--CDE702      CALL  02E7       SET-FAST
          4085--211050      LD    HL,5010    G-TABLE
          4088--1609        LD    D,09       begin value LOOPs
          408A--7A          LD    A,D        A = 9....1
   A-LOOP 408B--5F          LD    E,A        store A
          408C--42          LD    B,D        B = 9....1
   B-LOOP 408D--B8          CP    B
          408E--2816        JR    Z,40A6     NEXT B if B = A
          4090--4A          LD    C,D        C = 9....1
   C-LOOP 4091--B9          CP    C
          4092--280F        JR    Z,40A3     NEXT C if C = A
          4094--78          LD    A,B
          4095--B9          CP    C
          4096--280A        JR    Z,40A2     NXT C  if C = B
          4098--73          LD    (HL),E     1. dig of G
          4099--23          INC   HL
          409A--A7          AND   A          → No Carry
          409B--17          RLA
          409C--17          RLA
          409D--17          RLA
          409E--17          RLA              A = 2. dig of G
          409F--81          ADD   A,C        C = 3. dig of G
          40A0--77          LD    (HL),A
          40A1--23          INC   HL
    NXT C 40A2--7B          LD    A,E        retrieve A
   NEXT C 40A3--0D          DEC   C
          40A4--20EB        JR    NZ,4091    C-LOOP
   NEXT B 40A6--05          DEC   B
          40A7--20E4        JR    NZ,408D    B-LOOP
   NEXT A 40A9--3D          DEC   A
          40AA--20DF        JR    NZ,408B    A-LOOP

   282    FIND SET OF 3 3-DIGIT-NUMBERS

SET-R-PTR 40AC--211054      LD    HL,5410    R-TABLE
          40AF--22F84F      LD    (4FF8),HL  PTR R-TABLE
          40B2--210F50      LD    HL,500F    G-TABLE -1
       G1 40B5--23          INC   HL
          40B6--7E          LD    A,(HL)     1. dig G1
          40B7--FE06        CP    06         >= 6 ?
          40B9--D8          RET   C          to BASIC if not
          40BA--CDBC41      CALL  41BC       CONV-3-NIB
          40BD--2B          DEC   HL
          40BE--4E          LD    C,(HL)
          40BF--23          INC   HL
          40C0--46          LD    B,(HL)
          40C1--ED430A50    LD    (500A),BC  G1-STORE
          40C5--22F64F      LD    (4FF6),HL  G1-AD
          40C8--ED53F14F    LD    (4FF1),DE  G1-BITS
   SEEK G2 40CC--23         INC   HL         HL = AD LO-BYTE G
          40CD--7E          LD    A,(HL)     A = 1. dig G
          40CE--2C          INC   L          HL = AD HI-BYTE G
          40CF--B9          CP    C          A < 1. dig G1 ?
          40D0--30FA        JR    NC,40CC    SEEK G2 if not
          40D2--2D          DEC   L          HL = AD LO-BYTE G2
          40D3--2D          DEC   L
       G2 40D4--23          INC   HL
          40D5--7E          LD    A,(HL)     1. dig G2
          40D6--FE04        CP    04         >= 4 ?
          40D8--3005        JR    NC,40DF    CONT if yes
          40DA--2AF64F      LD    HL,(4FF6)  G1-AD
          40DD--18D6        JR    40B5       G1
     CONT 40DF--CDBC41      CALL  41BC       CONV-3-NIB
          40E2--3AF24F      LD    A,(4FF2)   BIT "9"-G1
          40E5--A2          AND   D          D = BIT "9"-G2
          40E6--20EC        JR    NZ,40D4    G2 if coincidence
          40E8--3AF14F      LD    A,(4FF1)   BITs "8....1"-G1
          40EB--A3          AND   E          E= BITs "8...1"-G2
          40EC--20E5        JR    NZ,40D4    G2 if coincidence
NO-COINC. 40EE--ED4BF14F    LD    BC,(4FF1)  G1-BITS
          40F2--EB          EX    DE,HL      HL = BITs G2
          40F3--09          ADD   HL,BC
          40F4--22F34F      LD    (4FF3),HL  G1- and G2-BITS
          40F7--EB          EX    DE,HL      HL = AD HI-BYTE G2
          40F8--2B          DEC   HL
          40F9--4E          LD    C,(HL)
          40FA--23          INC   HL
          40FB--46          LD    B,(HL)
          40FC--ED430C50    LD    (500C),BC  G2-STORE
          4100--22F84F      LD    (4FF8),HL  G2-AD
```

```
SEEK G3 4103--23          INC   HL          HL = AD LO-BYTE G
        4104--7E          LD    A,(HL)      A = 1. dig G
        4105--2C          INC   L           HL = AD HI-BYTE G
        4106--B9          CP    C           A < 1. dig G2 ?
        4107--30FA        JR    NC,4103     SEEK G3 if not
        4109--2D          DEC   L           HL = AD LO-BYTE G3
        410A--2D          DEC   L

     G3 410B--23          INC   HL
        410C--7E          LD    A,(HL)      1. dig G3
        410D--A7          AND   A           zero ?
        410E--2005        JR    NZ,4115     NOT-END if not
        4110--2AF84F      LD    HL,(4FF8)   G2-AD
        4113--18BF        JR    40D4        G2
NOT-END 4115--CDBC41      CALL  41BC        CONV-3-NIB
        4118--3AF44F      LD    A,(4FF4)    BIT "9"-G1&G2
        411B--A2          AND   D           D = BIT "9"-G3
        411C--20ED        JR    NZ,410B     G3 if coincidence
        411E--3AF34F      LD    A,(4FF3)    BITs "8...1"-G1&G2
        4121--A3          AND   E           E= BITs "8...1"-G3
        4122--20E7        JR    NZ,410B     G3 if coincidence
        4124--2B          DEC   HL
        4125--4E          LD    C,(HL)
        4126--23          INC   HL          HL = AD HI-BYTE G3
        4127--46          LD    B,(HL)
        4128--ED430E50    LD    (500E),BC   G3-STORE
```

2B3 MULTIPLY + TEST RESULT: "ALL 9?"

```
        412C--E5          PUSH  HL          stack G3-AD
   MULT 412D--CD5B41      CALL  415B        CLEAR/MULTIPLY
        4130--7E          LD    A,(HL)      HL = 5005
        4131--A7          AND   A           (HL) = 00 ?
        4132--2824        JR    Z,4158      TO-G3 if yes
ALL 9 ? 4134--CDBC41      CALL  41BC        CONV-3-NIB
        4137--0E03        LD    C,03        3 rounds
NXT-2-NIB 4139--CDC341    CALL  41C3        CONV-2-NIB
        413C--0D          DEC   C
        413D--20FA        JR    NZ,4139     NXT-2-NIB
        413F--2101FF      LD    HL,FF01     LH= 0..01 11111111
        4142--ED52        SBC   HL,DE       ED = R-BITs
        4144--E1          POP   HL          retrieve G3-AD
        4145--20C4        JR    NZ,410B     G3 if not ALL 9
```

2B4 STORE RESULT AND FACTORS

```
        4147--E5          PUSH  HL          stack G3-AD
        4148--210050      LD    HL,5000     NUMBERS-STORE
        414B--ED5BFB4F    LD    DE,(4FFB)   PTR R-TABLE
        414F--011000      LD    BC,0010     see text
        4152--EDB0        LDIR
        4154--ED53FB4F    LD    (4FFB),DE   new value PTR
  TO-G3 4158--E1          POP   HL          retrieve G3-AD
        4159--18B0        JR    410B        G3
```

2B5 MULTIPLY DEC 3 3-DIGIT-NUMBERS

```
 CLEAR 415B--21FF4F       LD    HL,4FFF
        415E--110050       LD    DE,5000     (DE) and (HL)
        4161--010A00       LD    BC,000A        are
        4164--EDB0         LDIR                 cleared

MULTIPLY 4166--2A0A50      LD    HL,(500A)   G1 digs
        4169--220350       LD    (5003),HL   to (DE)
        416C--62           LD    H,D         50 hex
        416D--3A0D50       LD    A,(500D)    G2-LO
        4170--CD9241       CALL  4192        MULT-BYTE
        4173--3A0C50       LD    A,(500C)    G2-HI
        4176--CD9D41       CALL  419D        MULT-NIB
        4179--2E09         LD    L,09        HL = end (HL)
        417B--1E04         LD    E,04        DE = end (DE)
        417D--0603         LD    B,03        3 bytes
     NX 417F--7E           LD    A,(HL)        from (HL)
        4180--12           LD    (DE),A          to (DE) and
        4181--3600         LD    (HL),00           clear (HL)
        4183--2D           DEC   L
        4184--1D           DEC   E
        4185--10F8         DJNZ  417F        NX
        4187--3A0F50       LD    A,(500F)    G3-LO
        418A--CD9241       CALL  4192        MULT-BYTE
        418D--3A0E50       LD    A,(500E)    G3-HI
        4190--180B         JR    419D        MULT-NIB
```

```
      MULT-BYTE 4192--4F        LD   C,A        store A
                4193--CD9B41    CALL 419B       RIGHT-NIB
                4196--79        LD   A,C        retrieve A
                4197--1F        RRA             Left nibble
                4198--1F        RRA
                4199--1F        RRA               to
                419A--1F        RRA
      RIGHT-NIB 419B--E60F      AND  0F         Right nibble
       MULT-NIB 419D--47        LD   B,A        A rounds
            NXT 419E--2E09      LD   L,09       HL = end (HL)
                41A0--1E04      LD   E,04       DE = end (DE)
      NEXT-BYTE 41A2--1A        LD   A,(DE)
                41A3--8E        ADC  A,(HL)
                41A4--27        DAA             see text
                41A5--77        LD   (HL),A
                41A6--2D        DEC  L
                41A7--1D        DEC  E          5 rounds
                41A8--F2A241    JP   P,41A2     NEXT-BYTE
                41AB--10F1      DJNZ 419E       NXT
  SHIFT-4BITS-L 41AD--2E01      LD   L,01       1. byte (DE)
                41AF--0604      LD   B,04       4 rounds
        NXT-BYTE 41B1--AF       XOR  A
                41B2--ED6F      RLD
                41B4--2D        DEC  L
                41B5--86        ADD  A,(HL)
                41B6--77        LD   (HL),A
                41B7--2C        INC  L
                41B8--2C        INC  L
                41B9--10F6      DJNZ 41B1       NXT-BYTE
                41BB--C9        RET             HL = 5005

          286   CONVERT DEC DIGITS INTO BITS
                3 OR 2 NIBBLES

     CONV-3-NIB 41BC--110000    LD   DE,0000    clear BIT-register
                41BF--7E        LD   A,(HL)
                41C0--CDD041    CALL 41D0       SET-BIT
     CONV-2-NIB 41C3--2C        INC  L
                41C4--7E        LD   A,(HL)
                41C5--CDCD41    CALL 41CD       RIGHT-DIG
                41C8--7E        LD   A,(HL)
                41C9--1F        RRA             Left
                41CA--1F        RRA               digit
                41CB--1F        RRA
                41CC--1F        RRA               to
      RIGHT-DIG 41CD--E60F      AND  0F           Right
                41CF--C8        RET  Z          if digit is 0
        SET-BIT 41D0--47        LD   B,A        A rounds
                41D1--AF        XOR  A          clear A-register
                41D2--37        SCF             begin with carry
    NXT-SHIFT 41D3--17          RLA
                41D4--10FD      DJNZ 41D3       NXT-SHIFT
                41D6--3001      JR   NC,41D9    NOT-"9"
                41D8--1C        INC  E          SET BIT "9"
       NOT-"9" 41D9--82         ADD  A,D        SET
                41DA--57        LD   D,A          BITs "8....1"
                41DB--C9        RET
```

Verder heb ik me gerealiseerd dat de tests G1>=612 en G2>=412 niet via kennis van de tabel-adressen behoeven te worden uit-gevoerd. Deze kondities kunnen namelijk worden vertaald in: honderdtal van G1 >= 6 en dat van G2 >= 4. Dit leidt tot de 2 3-bytes-kortere en ook snellere stukjes routine op AD 40B6/B9 resp. AD 40D5/D8.

Al met al is het verbeterde programma 10 bytes langer en veranderen alle CALL-adressen. Omdat (zeker een deel van) de handgeschreven bijschriften bij de MC-routine in het oktobernummer slecht leesbaar zijn uitgevallen (mea culpa), geef ik de gehele figuur 2B nog eens, maar nu met "getypte" labels en uitleg. Dat kost met mijn beperkte tikkunst en de dito ZX 81-wordprocess mogelijkheden wel vele uren extra, maar maakt het volgen van mijn gedachtenkronkels toch gemakkelijker, naar ik meen. Voor de goede orde geef ik ook de Hexdump van de verbeterde RTN. In figuur 1A wordt nu de eerste regel 1 REM-REGEL MET 346 BYTES. De figuren 2A 2C en 3 veranderen niet.

De Redaktie heeft vorige keer de goede gedachte gehad om met alfanumerieke "kodes" aan te geven, waar de verschillende delen van de MC-RTN van figuur 2B in de tekst worden behandeld. Die kodering vergemakkelijkt ook het volgen en begrijpen van het programma en ik heb deze daarom dankbaar overgenomen.

Het resultaat van de verbetering is dat de totale rekentijd nu 627s is, 46s minder dan de vorige versie en 50-maal zo snel als de in de HCC Nieuwsbrief gepubliceerde oplossing. Maar het verbeterde deel van de RTN gebruikt nog altijd 488s, dat is bijna 78% van dat totaal. Daarom blijven brilliante ideeen voor verdere tijdsbesparing welkom.

De toegezegde volledige PRINTOUT van de antwoorden, zoals ze worden gegenereerd en een sorteerprogramma om die resul-taten in de juiste volgorde te zetten worden in het volgende nummer besproken.

H A N V A N A B B E

Beyond the Ice Palace:

Dit spel is vooral leuk omdat, ik het zelf (met poke's) uitgespeeld heb.
Jij bent een held en je moet het land achter het ijspaleis bevrijden van de kwa de monsters en, uiteindelijk, van de heks.

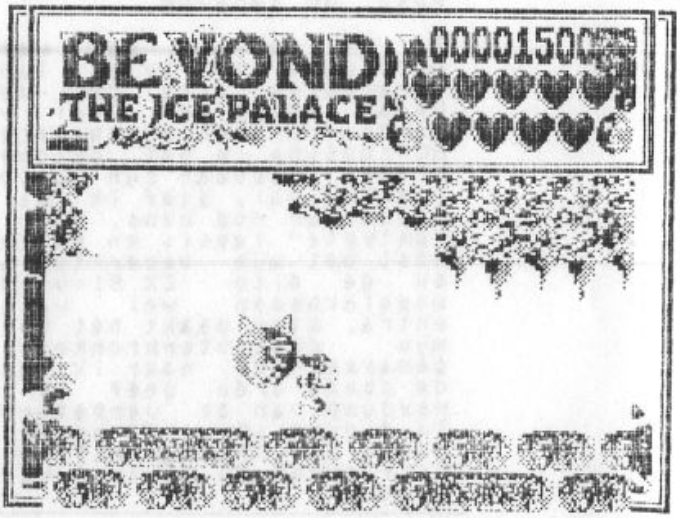
Bovenin het scherm zie je je levens (10) en daarnaast twee 'gezichtjes'. Deze gezichtjes , waar je er maar twee van krijgt en ongeveer per level één erbij (je kunt er maximaal 2 hebben!) , kun je gebruiken als er wat al te veel gemene beesten rondlopen die worden dan gedood door dat gezicht.
Het gezicht blijft ongeveer 5 seconden rond je zweven.

Voordat je bij de heks komt moet je eerst nog langs 2 andere levels. De weg vinden in level 1 is niet moeilijk, zolang je niet probeert over de 1e kloof te springen (dat is ook niet de bedoeling!), maar level 2 is een waar doolhof!
Je moet 'gewoon' omhoog maar , sommige ladders lopen naar platforms die nergens op uitkomen! Dan is het zeer handig om (zoals ik dus had) een kaart te hebben.

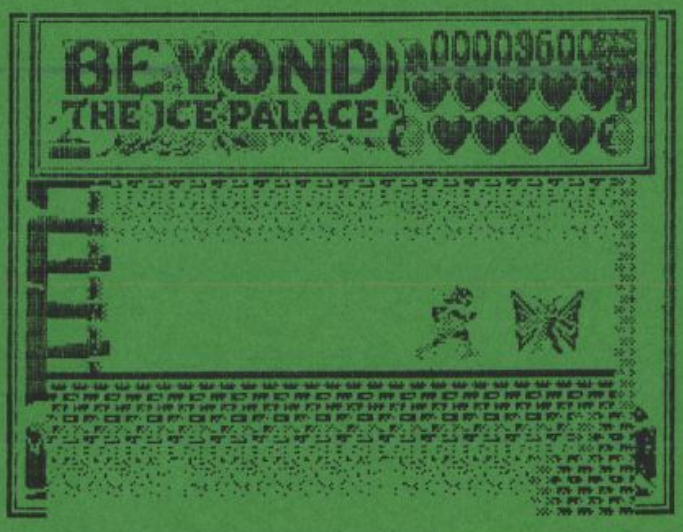
Level 3 (laatste level) is ongeveer 't zelfde als level 2 alleen, hier kan je (bijna) niet op een platform komen wat nergens heen gaat.
Maar er lopen hier heel wat meer enge beesten rond! Op het einde van level 3 kom je bij de heks en als je die gedood hebt krijg je een boodschap op het scherm ,en zo heb je weer een spel uitgespeeld!!

PS:poke voor eeuwig level = 38279,0.

Frans Postma.



HCC DAGEN 25 EN 26 NOVEMBER
JAARBEURS UTRECHT



DRUKWERK

PORT BETAALD
GRONINGEN

AFZ:
SGG
REDAKTIEADRES
Mevr.F.Elstrodt
Kam.Onnesstr 172
9727 HS Groningen

26

AAN:

_____

_____