Nieuwsbrief

16de jaargang december 2009 Nummer 4

HCC BASIC interessegroep

Inhoud

Deze uitgave kwam tot stand met bijdragen van:

Naam	Blz
Gordon Rahman	19
Het Bestuur	Penningmeester gezocht: 17

Contacten

Functie	Naam	Telefoonnr.	E-mail
Voorzitter	Willem Gobel	0118-850837	voorz@basic-gg.hcc.nl
Secretaris	Gordon Rahman Tobias Asserstraat 6 2037 JA Haarlem	023-5334881	secr@basic-gg.hcc.nl
Penningmeester	Piet Boere	0348-473115	penm@basic-gg.hcc.nl
Bestuurslid	Titus Krijgsman	075-6145458	t.krijgsman8@upcmail.nl
Redacteur	M.A. Kurvers Schaapsveld 46 3773 ZJ Barneveld	0342-424452	m.a.kurvers@hccnet.nl
Ledenadministratie	Fred Luchsinger	0318-571187	f.luchsinger@kader.hcc.nl
Webmaster	Jan van der Linden	071-3413679	j.vd.linden@kader.hcc.nl

http://www.basic.hcc.nl

Redactioneel

Als u programma's wilt programmeren, zoals spellen of wiskundige grafieken, dan kunt u structuurvormen tegenkomen die u eerder nooit gehad hebt. Elk project heeft wel weer zijn eigen structuurvormen. De structuurvormen, die te maken kunnen hebben met spellen, kwamen in oudere BASIC versies het meest voor. De grafische, makkelijk te gebruiken commando's zien we tegenwoordig nergens meer. Hoe dan ook u kunt die structuurvormen nog steeds maken en er zit zelfs een BASIC programmeertaal bij. De programma's heten DarkBASIC en Game Maker, met een programmeertaal GML (Game Maker Language).

Marco Kurvers

Programmeerstructuurvormen.

We weten dat programma's en projecten opgebouwd zijn uit delen. Elk deel heeft zijn eigen doel. Er kunnen ook onderdelen zijn die juist niet een doel hebben voor een bepaald deel, maar nodig zijn voor het hele programma en project: structuurvormen.

Een structuurvorm kan ook een codeblok zijn, een lusstructuur, een keuzestructuur of een fout-exception blok. Dat laatste bestond zelfs al in Commodore 128 BASIC 7.0 (`TRAP - IF ER = ... EL = ... THEN ... - RESUME [NEXT]`). Maar dit zijn niet de structuurvormen die zonder een doel werken. Ze moeten gemaakt worden, daarna moeten ze gebruikt kunnen worden in het hele programma en project. Hieronder staan de structuurvormen die zonder een doel eerst gemaakt worden:

- enumeraties
- records
- structures (werkt alleen in Visual Basic .NET, sleutelwoord `Type` heet nu `Structure`)
- klassen
- sprites
- zelf getypeerde structuurvormen (met uitzondering punt 3: structures)

Er kunnen natuurlijk nog meer bestaan, nog niet eens genoemd: array elementlijsten, en zoals er staat: zelf getypeerde structuurvormen, kunt u zelf ook maken. Ze kunnen heel nuttig zijn en in veel situaties kunnen ze veel code en tijd besparen. Houd er rekening mee dat, behalve de enumeraties, de klassen en de sprites, alles in `Type ... End Type` blokken gemaakt moet worden wanneer u in een Basic versie bezig bent die niets te maken heeft met het .NET Framework. Deze Basic versie van tegenwoordig ondersteunt niet meer het aangegeven `Type` structuurvorm, zoals u bij bovengenoemde opsommingen ziet staan. Dat betekent niet dat u uw structuurvorm niet meer zou kunnen maken. Het enige waar u aan moet denken is dat u tegenwoordig `Structure ... End Structure` nodig hebt.

Hoe gebruiken we de vormen?

Basic heeft vooraf gedeclareerde structuurvormen. We kunnen die meteen gebruiken. Onderstaand voorbeeld geeft bijvoorbeeld twee tekstregels weer:

```
Print "Hallo allemaal!" & vbCrLf & "Hoe gaat het met jullie?"
```

De variabele `vbCrLf` is eigenlijk een constante. Er is gewoon een structuurvorm gemaakt waar die constante in zit. De waarde van die constante is echter verborgen, maar doordat de naam wel het juiste doelstelling omschrijft, weet de programmeur wat de functie van die constante is. De omschrijving van die constante betekent Carriage Return Line Feed en zorgt ervoor dat de tekst die erop volgt aan het begin op de volgende regel begint.
Dit is het bewijs waarom het zo belangrijk is variabelen, constanten, enumeratie-, record- en klassenvelden duidelijke namen te geven.

Hieronder zijn wat structuurvormen.

`Type TGegevensrij` **(1)**
`  Naam As String`
`  Adres As String`
`  Woonplaats As String`
`End Type`
	`Structure structGegevensrij` **(2)**
	`  Public Naam As String`
	`  Public Adres As String`
	`  Public Woonplaats As String`
	`End Structure`

`Enum enumVoorbeeld` **(3)**
`  VeldNaam0 = 0`
`  VeldNaam1 = 1`
`  VeldNaam2 = 2`
`  ...`
`End Enum`
	`Dim strGroenten() As String = _`
	`  { _`
	`  "Bloemkolen", _`
	`  "Worteltjes", _`
	`  "Spruitjes" _`
	`  }` **(4)**

1. De structuurvorm `TGegevensrij` is een type dat in Visual Basic verouderd is. Het werkt alleen nog in vorige Basic versies zoals Visual Basic 6 en VBA in Office toepassingen. Een type kan niet direct worden gebruikt. U moet eerst een variabele van dat type declareren. De variabele en de velden (Naam, Adres en Woonplaats) houd u gescheiden met een punt. Voorbeeld:
   ```
   Dim Gegevensrij As TGegevensrij
   Gegevensrij.Naam = "Jan"
   ```
2. De structuurvorm `structGegevensrij` lijkt op die van punt 1. Toch zit er wat verschil aan. Deze structuurvorm kan alleen worden gebruikt in Visual Basic .NET en de velden moeten nu een karakter krijgen. U zou misschien denken aan een klasse structuurvorm, maar een `Structure` mist echter onderdelen die we wel in een klasse kunnen vinden, bijvoorbeeld een constructor.
3. Enumeraties geven de code een leesbaar karakter. De waarden van de velden blijven verborgen en u hoeft alleen maar de namen te gebruiken, net als het voorbeeld om de tweede tekstregel op de volgende regel weer te geven. Denk eraan: enumeraties zijn geen typen. De velden zijn globaal te gebruiken en u mag ook de naam van de enumeratie opgeven.
4. Deze structuurvorm is een voorbeeld van een vooraf geïnitialiseerde array met drie elementen. Zoals men eerder gebruik moest maken van het sleutelwoord `ReDim` om met een dynamische array te kunnen werken, zo is er nu een mogelijkheid om dynamisch gelijk in de declaratie de aantal elementen te vullen. Om de structuurvorm duidelijk te houden, worden de elementen per regel gemaakt. Dit kan alleen als achter elke komma een underscore wordt geplaatst, en dat geldt ook voor de accolades die de structuurvorm groepeert.

Structuurvormen in spellen.

In spellen komen we ook structuurvormen tegen, zoals sprites.
Het is echter een stuk moeilijker dan wanneer we een normaal type maken of een klasse. BASIC had vroeger sprite structuurvormen. In BASIC 7.0 zit zelfs een complete sprite editor met allerlei sprite functies en commando's. Helaas moeten we die tegenwoordig allemaal missen, waardoor we een sprite klasse moeten maken die de sprite moet inladen.

In de klasse worden dan de methoden gemaakt die voor de juiste besturing van de sprite moet zorgen.

Gelukkig is er wel wat hulp in de code. Er zijn mogelijkheden om de objecten van de sprites te roteren, te vergroten en te verkleinen. We moeten zelf de code, die de mogelijkheden heeft, opzoeken. Het zit niet standaard in Basic, maar wel in DarkBASIC.

DarkBASIC en Game Maker.

DarkBASIC is zeer geschikt om games te programmeren. Het is zelfs mogelijk om 3D vormen te gebruiken en we kunnen ze allerlei texturen geven, watertexturen, metaaltexturen, vuurtexturen, noem maar op. DarkBASIC kunt u downloaden om te proberen, maar om er echt mee te werken kunt u het kopen. Het pakket DarkBASIC Professional kost €44,99.

Een ander programma om games te ontwerpen is Game Maker. Het is zeer makkelijk en de games kunnen zonder programmacode gemaakt worden. Wilt u wat professionelere games maken, dan is wat functionele code niet verkeerd voor bijvoorbeeld extra textuureffecten, zoals explosies en muisbesturingen.

Het verschil tussen DarkBASIC en Game Maker is het gebruik van de structuurvormen. In DarkBASIC moeten we de structuurvormen inladen en plaatsen op het scherm. Het scherm moeten we echter ook instellen. In Game Maker kunnen we direct zien wat er gebeurt. We hoeven dan ook geen aparte sprite programma te openen om dan daarna een object van te maken. Game Maker heeft alles zelf. Het programma Paint is zelfs niet nodig als we sprites willen maken. Het enige wat we wel nodig hebben is een sound editor. In Game Maker kunnen we de eigenschappen van het geluidsbestand wijzigen, maar we kunnen geen nieuwe ontwerpen. Alleen inladen.

Als u Game Maker wilt proberen, kunt u die downloaden van Internet. In de zoekmachine typt u 'Game Maker' en u kunt dan de juiste website kiezen. Het programma is gratis, maar is niet volledig. Als u het programma volledig wilt, kunt u het kopen. Game Maker is veel goedkoper dan DarkBASIC en u kunt echt met plezier ermee werken. De maker van Game Maker heet: Mark Overmars.

Basic en GML

Als u Basic voor het eerst in DarkBASIC ziet, is de syntaxis even wennen. In C++ zou u veel structuur resources moeten maken om de structuurvormen in objecten te kunnen gebruiken. In DarkBASIC zijn die niet nodig waardoor een project meteen stukken kleiner wordt. Het enige wat u moet doen is een structuurvorm initialiseren, bijvoorbeeld door het instellen van de sprite coördinaten.

Hieronder ziet u een programmavoorbeeld in DarkBASIC. Het voorbeeld is van de code 'PelicanDemo.dba'. Alles wat vet is zijn sleutelwoorden.

```
Rem Project: PelicanDemo
Rem Created: 14/08/2003 04:34:19

rem Best display
if check display mode(1024,768,32)=1
   set display mode 1024,768,32
endif

rem Init app
sync on : sync rate 60 : backdrop off : hide mouse

rem Load bird and apply effect
load object "media\pelican.x",1
load image "media\dbprocubemap.dds",1
set cube mapping on 1,1,1,1,1,1,1

rem Move camera closer
move camera 25

rem Create another camera for backdrop effect
make camera 1
color backdrop 1,0
set camera to image 1,1,256,256
set camera fov 1,2
position camera 1,0,50,0
point camera 1,0,0,0
set current camera 0

rem Use sprite to fill backdrop
sprite 1,-10000,-10000,1
size sprite 1,1024,768
set sprite diffuse 1,100,100,255
set sprite alpha 1,128
draw to back

rem Setup nice font
ink 0x88FFFFFF,0
set text font "Tahoma"
set text size 32

rem Main loop
do

rem Paste backdrop using sprite
paste sprite 1,0,0 : mirror sprite 1
paste sprite 1,0,0 : flip sprite 1
paste sprite 1,0,0 : mirror sprite 1
paste sprite 1,0,0 : flip sprite 1
```

```
rem Rotate object
rotate object 1,object angle x(1)+0.1,object angle y(1)+0.2,object
angle z(1)+0.3

rem User prompt
center text screen width()/2,screen height()-50,"DBPro Pelican Demo
( Cube Mapping + Camera Backdrop Techniques )"

rem Update screen
sync

rem End loop
loop
```

Helaas heb ik geen uitvoer van het project. DarkBASIC is bij mij expired, dus moet ik de professionele versie kopen. Toch denk ik dat ik Game Maker blijf gebruiken. De GML code ziet er minder technisch uit dan DarkBASIC, ook al is GML een programmeertaal die tussen Basic en C in zit. Het is ook geen applicatie programmeertaal, maar een script taal. Te vergelijken met VBA in Excel.

Na een object te hebben gemaakt in Game Maker, kunt u direct de events en acties plaatsen. Nog wel zonder de GML code, want u kunt dit doen alsof u werkt met Visual Basic formulieren. Nadat een object de juiste besturing heeft, kunt u de instanties van het object op de room plaatsen. Ook de room moet eerst gemaakt worden. Ook dat wordt zonder code gedaan.

Hieronder ziet u een voorbeeld van een GML script methode. Deze heb ik zelf geschreven die voor muis besturing met een object moet zorgen. Zonder die methode zou het object niet meebewegen met de muis. De randen van het scherm zijn ook erg belangrijk. Het object moet stoppen met bewegen zodra de muis buiten het scherm komt.

De naam van het script heet: scr_bat_move

```
{
    var i, s;     in Basic werkt dit als: Dim i, s
    s = argument0;     het eerste argument van de methode wordt toegekend aan variable s
    switch (global.bat_length)   in Basic: Select Case global.bat_length
    {
        case 0: i = 12; break;   in Basic hetzelfde, maar geen commando break
        case 1: i = 19; break;   laat ook de dubbelepunt : weg en plaats de toekenning
        case 2: i = 38; break;   op de volgende regel (eventueel een inspringende tab)
    }    gebruik in Basic hier: End Select
    s.y = 464;   de y coördinaat wordt bepaald; argument0 is dus sprite object bat
    if (mouse_x <= 32 + i)     GML kent ook statement then, het is echter niet nodig
        s.x = 32 + i      indien mouse_x buiten de rand komt behoud x zijn eigen
                          coördinaat
```

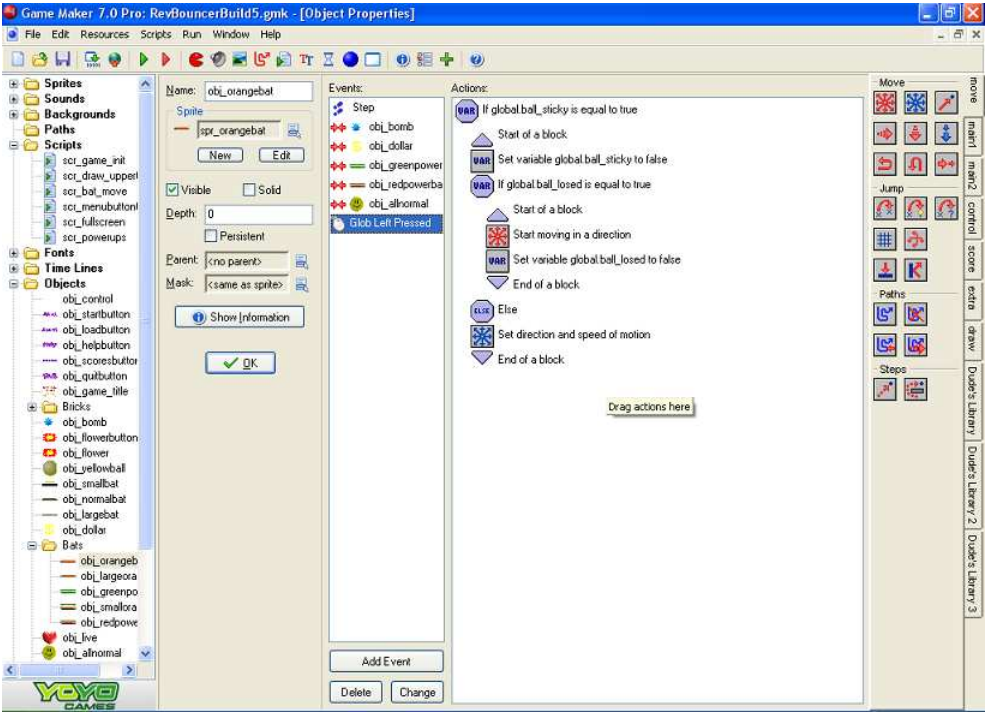
```
    else
        if (mouse_x >= room_width - (32 + i))
            s.x = room_width - (32 + i)
        else
            s.x = mouse_x;      is mouse_x binnen de randen, laat sprite object bat
                                meebewegen
    if (global.ball_sticky)     controleert of de bal vast moet blijven aan de bat
    {
        hier is het 'ja' dus moeten de bal coördinaten met het bat object meegaan
        obj_yellowball.x = s.x + 4;
        obj_yellowball.y = s.y - 11;
    }   gebruik in Basic hier: End If
    if (global.weapon_sticky)     controleert of het wapen object vast moet zitten
    {
        obj_weapon.x = s.x;
        obj_weapon.y = s.y;
    }   ook hier: End If
}
```

Wanneer u de accolades lastig vindt kunt u ook gebruik maken van de statements `begin` en `end`. Overal waar de accolades staan plaatst u de statements. Zoals u in schuinschrift ziet staan, kent GML ook het statement `then`.

Figuur 1 geeft de uitvoer van de room, de onderkant van het scherm.

Figuur 1: Telkens wanneer de muis bewogen wordt bewegen de bat en de bal mee. Er is een control object zonder sprite die op de room niet zichtbaar is. Dit object heeft een step event die elke keer uitgevoerd wordt en bovenstaande script scr_bat_move zal aanroepen.

Er is ook een globale muisklik event. Het bat object heeft de muisklik event. Waarom de muisklik globaal moet zijn, komt doordat de muis over het hele scherm moet kunnen werken en zich niets van de andere objecten aan mag trekken. Zouden we een lokale muisklik event kiezen, dan zouden de acties alleen uitgevoerd worden zodra de muis op het bat object komt. Zie verder Figuur 2.



Figuur 2: Dit voorbeeld laat zien dat Game Maker een zeer uitgebreid programma is. Hier kunt u zien dat eigenlijk alles met de muis gewerkt kan worden. Aan de linkerkant ziet u een boomlijst zoals een mappenstructuur in Windows Verkenner. De scripts en de objecten staan open. In de map Scripts kunt u de script naam zien die ik als voorbeeld heb laten zien: scr_bat_move. In de map Objects is het object obj_orangebat geselecteerd waarvan de eigenschappen, events en acties aan de rechterkant te zien zijn.
In de eerste regel van de Actions wordt de bekende globale variabele `ball_sticky` gecontroleerd. Als deze `true` is, wordt het codeblok gestart en krijgt de variabele de waarde `false`. Is daarvoor echter de bal onderaan de bat verdwenen, dan moet de bal opnieuw weggeschoten worden en moet de globale variabele `ball_losed` de waarde `false` krijgen. Is dat niet het geval `else` dan zal de bal ook weggeschoten worden, maar dan met behulp van de formule die de hoek in graden berekent (Set direction and speed of motion). Daarna zal het codeblok van de globale `ball_sticky` controle worden beëindigd.

Bepaal zelf hoe u expressies, condities en de resultaten ervan gebruikt. Controleer ze wanneer u denkt dat bepaalde code wel of niet uitgevoerd mag worden. Maak gebruik van operatoren als meerdere condities gecontroleerd moeten worden.

Situaties met objecten en instanties.

De gebeurtenissen met objecten hoeven niet speciaal in games en in game editors actief te zijn, want hoe zouden we het programma Game Maker hebben als we niet eens object situaties in programmeertalen kunnen gebruiken? Stel dat we zo'n editor willen maken: om een goede game functionaliteit te krijgen zoals u in Figuur 2 ziet, zouden we in het Basic project, of in een andere programmeertaal, dezelfde functies moeten programmeren om de gebruiker van een goede editor te voorzien. Dat betekent niet, wat de gebruiker als sprite kiest, dat die ook in uw project moet staan. Sterker nog, dat is onmogelijk. Het gaat erom dat u methoden in uw project maakt die ervoor moeten zorgen dat de gebruiker allerlei soorten afbeeldingen kan openen en kan bewaren, bijvoorbeeld sprites, achtergronden, cursors, tiles, enzovoort. Tiles zijn trouwens tegels. Ze dienen als voorgrond objecten die niet op dezelfde manier werken als normale objecten. Tiles zijn ook geen sprites. Ze worden als achtergrond afbeeldingen gebruikt.

Instanties

Wanneer op een Basic formulier de besturingselementen zijn geplaatst en in Game Maker alle objecten op een room zijn geplaatst, kunnen we alle objecten laten werken. Maar... dat we het alle objecten noemen is echter onjuist. Plaatsen we bijvoorbeeld 5 tekstvakken op het formulier dan hebben we 1 tekstvak object met 5 tekstvak instanties. Elke instantie heeft zijn eigen eigenschappen. Daarom kunnen we ook elke tekstvak van andere effecten voorzien. We kunnen het allemaal veranderen, maar we kunnen niet de instanties uitbreiden. Alles wat erin zit komt uit het object dat gecreëerd is van de tekstvak klasse. Als we toch wat erbij willen hebben dan kunnen we wel de klasse gebruiken door een nieuwe control aan te maken en met de tekstvak klasse nieuwe functionaliteit toe te voegen (eigenschappen, methoden, gebeurtenissen). Voordat we dan de nieuwe control kunnen gebruiken moet die eerst geïnstalleerd worden. Zie op internet en in Basic boeken meer daarover: Ontwerpen ActiveX Controls.

Wat betreft de objecten die op de room zijn geplaatst in Game Maker; die instanties hebben geen naam, maar unieke ID nummers. Net zoals we de eigenschappen van een instantie op een Basic formulier kunnen wijzigen, heeft elke instantie op de room zijn eigen code om bijvoorbeeld de kleur te wijzigen. Wat we in de objecten doen, zie Figuur 2, staat los van de code die we in elke instantie kunnen programmeren op de room.

Marco Kurvers

Eerst toekennen dan pas verwijzen.

We kunnen geen gesloten deur openen als we geen sleutel hebben. We kunnen ook niet aan iemand de weg wijzen als we zelf niet eens weten waar de straat is. Zo kan de computer geen controle uitvoeren als er nog geen waarde aan een variabele is toegekend. Hoewel BASIC daar niet zo moeilijk over doet, zijn er toch BASIC versies en programmeertalen die de variabelen helemaal niet kennen als ze nog niet geïnitialiseerd zijn. In veel versies en dialecten is dus onderstaande stukje code niet goed en kan het problemen veroorzaken:

```
Dim A As Integer

If A = 0 Then A = 4
```

Gelukkig zullen de meeste BASIC versies variabele A meteen initialiseren, na het declareren van de variabele. Variabele A zal het getal 0 als standaard waarde krijgen. We weten echter dat BASIC een taal is die niet moeilijk doet als we de variabele niet zouden declareren. Mogen we dan aan de variabele een waarde verwijzen in een `If ... Then`? Zouden we in zo'n BASIC versie bovenstaande `Dim` regel verwijderen, dan openen we dus de deur (controle) en pakken we daarna pas de sleutel (de verwijzing). Om eerst de variabele te declareren en eventueel voor de zekerheid een waarde toe te kennen, hebben we de sleutel en kunnen we de variabele op een waarde controleren.

Hoewel dit niet echt foutmeldingen zal veroorzaken en dit onderwerp ook niet echt het belangrijkste van het programmeren is, zal het toch beter zijn om goed uit te kijken wat u met variabelen doet.

Visual Basic .NET kent nu een manier van initialiseren in een declaratieregel. En weer kunnen we zeggen: het is optioneel, het hoeft niet. Prima, uw keuze, maar het is niet verkeerd om het te doen. Onderstaande coderegel werkt in de .NET versies:

```
Dim A As Integer = 0
```

en het hoeft niet eens de standaard waarde te zijn. Een ander getal mag ook. Het is zelfs mogelijk om in een declaratie een complete instantie van een object te initialiseren:

```
Dim objMijnKlasse As clsMijnKlasse = New objMijnKlasse
```

Marco Kurvers

Grafisch programmeren in GW-BASIC.

Vroeger, in de IBM-BASIC en GW-BASIC tijd, hadden we best wel goede grafische mogelijkheden. Vooral met de IBM-BASICA (Advanced BASIC), want die ondersteunde een groot aantal grafische mogelijkheden. Deze grafische opdrachten konden echter alleen functioneren wanneer de IBM PC voorzien was van een grafische kaart.

Bij de IBM PC-compatible microcomputers is zo'n grafische kaart vrijwel standaard aanwezig (hardware) en GW-BASIC (software). We moeten het doen met een meegeleverd monochrome (éénkleurig) beeldscherm, maar men kan ook een color/graphics adaptor installeren en het aanschaffen van een kleurenbeeldscherm.

In de vorige nieuwsbrief heb ik laten zien wat voor tekenprobleem er is en hoe de correctiefactor in elkaar zit. Hieronder geef ik nogmaals een deel ervan om u op de hoogte te houden waarom we geen vierkant kunnen tekenen en er wel een rechthoek van gemaakt wordt.

BASIC Tip! Als u merkt dat u in een BASIC versie ook bovenstaand probleem hebt en de schaal niet klopt, probeer dan ook om gebruik te maken van de formule, die zometeen komt, om de lijnstukken te corrigeren. Probeer ook eens te kijken wat de formule doet met de grafische statements die nu nog bestaan.

De juiste schaal, correctiefactor en de formule.

De linkerbovenhoek van ons scherm is (0,0), de rechterbenedenhoek is het punt (320,320). De computer gaat er ook van uit dat schermbeeldpunten gespecificeerd worden met een horizontale en een verticale coördinaat, waarbij de linkerbovenhoek de 'oorsprong' is en de x-as (horizontale coördinaat) naar rechts wijst en de y-as (verticale coördinaat) naar beneden. Het midden van ons beeldscherm heeft dan de mooie coördinaten (160,160).

De computer gaat echter uit (in superresolutiestand) van een beeldscherm met 640 punten horizontaal en 325 punten verticaal of van 320 bij 200 punten in de middenresolutiestand. We moeten dus 'ons' scherm op 'zijn' scherm afbeelden. Daarbij moeten we erom denken dat, als we de computer twee 'even lange' lijnstukken horizontaal en verticaal laten tekenen, hij het horizontale lijnstuk korter tekent dan het verticale. Willen we dat ons beeldscherm van 320 bij 320 ook 'vierkant' op het computerscherm wordt afgebeeld, dan moeten we de horizontale coördinaten (x-coördinaten) met 1,55 vermenigvuldigen. Dit geldt voor zowel de middenresolutie als de hogere resolutiestand. Ga eventueel na of uw computer een andere factor nodig heeft.

In de vorige nieuwsbrief heb ik u laten zien hoe we de juiste correctiefactor moeten gebruiken. Nogmaals zal ik hieronder de twee regels laten zien die in alle listings nodig zijn.

```
100 CLEAR ,19202 : SCREEN 105,,3,3
110 DEF FNX(X)=INT(1.55*(50+X)+.5)
```

Hebt u een 256K of groter systeem, laat dan in regel 100 de opdracht `CLEAR ,19202` weg, anders gaat uw systeem 'down'. Hebt u geen superresolutie, neem dan in regel 100 `SCREEN 1,,3,3` voor middenresolutie of `SCREEN 2,,3,3` voor hoge resolutie. Neem dan ook in regel 110 niet 50 maar 5.
De functie zal dan worden: `DEF FNX(X)=INT(1.55*(5+X)+.5)`
Pas ook zonodig de factor 1.55 aan uw systeem aan en denk erom dat u dan met een vierkant beeld van 200 bij 200 werkt, hetgeen tot gevolg heeft dat U=160 en V=160 veranderd moeten worden in respectievelijk U=100 en V=100. Zie straks meer over U en V voor de omschrijvingen.

Puntgraphics

Bij puntgraphics worden de coördinaten x en y van een bepaald punt berekend en wordt dit punt door een bepaalde graphic opdracht, voorbeeld: `PSET (x,y)`, op het scherm zichtbaar. Willen we een dergelijk punt weer onzichtbaar maken, dan is daar een andere opdracht, voorbeeld: `PRESET (x,y)`, voor nodig. In de komende listings zal echter voor het tekenen van figuren niet deze puntgraphics-techniek worden gebruikt.

Lijngraphics

Bij lijngraphics worden door een bepaalde opdracht, die als machineroutine in de microcomputer aanwezig is, razendsnel twee berekende punten $P_1(X_1,Y_1)$ en $P_2(X_2,Y_2)$ door een rechte lijn met elkaar verbonden. Zo'n rechte verbindingslijn wordt opgebouwd uit een groot aantal kleine horizontale en verticale lijnstukjes. Op een afstand van zo'n 50 cm van het scherm lijken deze verbindingslijnen inderdaad bijna recht.

In de programma's zal gebruik worden gemaakt van deze lijngraphics. Alles wat getekend wordt ontstaat door het steeds verbinden van twee punten door een rechte lijn, of door een recht lijntje. De programma's zijn kort, zodat duidelijk wordt op welke wijze de tekeningen ontstaan. U kunt de programma's zelf uitbreiden en verfraaien. U kunt kleuren gebruiken of regels toevoegen waarmee de door u in te toetsen waarden gecontroleerd worden, zodat dit niet leidt tot het afbreken van het programma. In de appendix vindt u een aantal uitbreidingen van enkele programma's. Hierin ziet u hoe u met kleuren kunt werken, hoe u vlakken vult en hoe u tekst op het grafische scherm kunt afdrukken. De appendix zal later van toepassing komen. Eerst is het belangrijk om te weten hoe het allemaal werkt in GW-BASIC en in Visual Basic.

BASIC Tip! Het is altijd te proberen om in de tegenwoordige BASIC versies de programma's over te nemen van GW-BASIC. Bent u een Visual Basic gebrui-

ker, plaats dan de code in de Paint event en initialiseer de programmastructuur, zoals de nodige variabelen, in de Load event van het formulier. De meeste teken opdrachten bestaan nog steeds, maar Visual Basic herkent echter niet de opdracht `PRESET` om een punt onzichtbaar te maken.

BASIC Tip! Onthoud dat de formule voor de correctiefactor in Visual Basic niet meer nodig is. Dat komt, doordat het tekenen automatisch op schaal werkt. U kunt dus de listings overnemen zonder gebruik te maken van de correctiefactor.

Programmastructuur

In alle programma's gebruiken we steeds dezelfde variabelen en dezelfde programmastructuur. De verschillende variabelen betekenen:

X1, Y1	coördinaten van het beginpunt
X2, Y2	coördinaten van het eindpunt
U, V	oorsprong van het wiskundige coördinatensysteem; dikwijls is dit het midden van het beeldscherm (160, 160)
H	de waarde 0.5 voor het afronden op helen
K	de vermenigvuldigingsfactor voor functiewaarden
W,W1	hoeken bij trigonometrische functies
RD	het getal $\pi/180$ voor het omrekenen van graden in radialen

Een grof structuurdiagram voor de programma's ziet er zo uit:

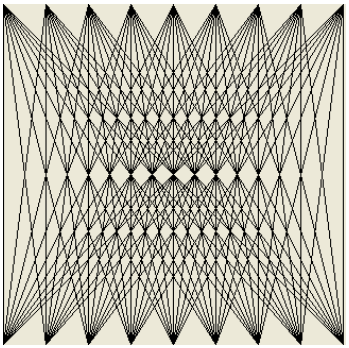
begin programma	
graphic-stand kiezen	
variabelen U, V, H, K, RD, enz. vastleggen	
punt P1(X1, Y1) vastleggen	
	doe zolang nodig nieuw punt P2(X2, Y2) berekenen verbind P1 met P2 punt P2 wordt punt P1 (X1=X2 Y1=Y2)
einde programma	

De voorbeelden

Genoeg theorie, nu de programma's. Elke twee programma's, GW-BASIC en Visual Basic, zal bij een figuur horen die aan de rechterkant komt te staan. Als u toch de formule ook in Visual Basic wilt gebruiken, schrijf dan een Private functie in de code van het formulier met de formule als return-waarde. U kunt dan met de factor experimenteren om de tekening op verschillende schalen af te beelden.

Programma 1 tekent een 'diagonaalweb'. Elk van de acht bovenste punten wordt door een rechte lijn verbonden met elk van de onderste acht punten. Gebruik alleen argument `e` in Visual Basic .NET. In Visual Basic 6 of e.v.t. in andere versies kunt u gebruik maken van de `LINE` opdracht.

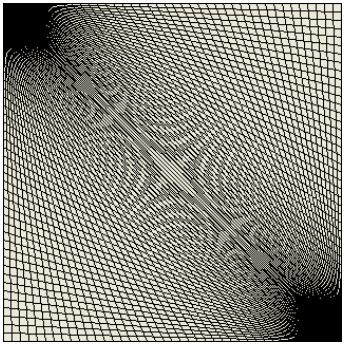
```
100 '          programma 1     DIAGONAALWEB
110 CLEAR ,19202 : SCREEN 105,,3,3
120 DEF FNX(X)=INT(1.55*(50+X)+.5)
130 CLS: KEY OFF
140 FOR X1=0 TO 320 STEP 40
150    FOR X2=0 TO 320 STEP 40
160       LINE (FNX(X1),0) - (FNX(X2),320),1
170    NEXT X2
180 NEXT X1
190 A$=INKEY$: IF A$="" THEN 190
200 CLS: KEY ON: END
```



```
Dim X1, X2 As Integer
For X1 = 0 To 320 Step 40
   For X2 = 0 To 320 Step 40
      e.Graphics.DrawLine(Pens.Black, X1, 0, X2, 320)
   Next
Next
```

Programma 2 is een fraai voorbeeld van het Moiree-effect. Geniet ervan.
Let op, gebruik voor niet .NET Basic versies normale lussen en declareer eerst de variabelen.

```
100 '       programma 2       MOIREE-EFFECT
110 CLEAR ,19202 : SCREEN 105,,3,3
120 DEF FNX(X)=INT(1.55*(50+X)+.5)
130 CLS: KEY OFF
140 FOR J=0 TO 320 STEP 8
150    LINE (FNX(0),0)-(FNX(J),320),1
160    LINE (FNX(0),0)-(FNX(320),320-J),1
170 NEXT J
180 FOR J=0 TO 320 STEP 8
190    LINE (FNX(320),320)-(FNX(0),320-J),1
200    LINE (FNX(320),320)-(FNX(J),0),1
210 NEXT J
220 A$=INKEY$: IF A$="" THEN 220
230 CLS: KEY ON: END
```



```
With e.Graphics
   For J As Integer = 0 To 320 Step 8
      .DrawLine (Pens.Black, 0, 0, J, 320)
      .DrawLine (Pens.Black, 0, 0, 320, 320 - J)
   Next
   For J As Integer = 0 To 320 Step 8
      .DrawLine (Pens.Black, 320, 320, 0, 320 - J)
      .DrawLine (Pens.Black, 320, 320, J, 0)
   Next
End With
```

Bron: IBM- en GW-BASIC graphics van Academic Service
Tekst overname, tips en veranderingen: Marco Kurvers

BASIC nieuws en tips

Penningmeester gezocht.

Onze huidige penningmeester, Piet Boere, heeft te kennen gegeven dat hij op de Algemene Ledenvergadering van 2010 zijn functie ter beschikking stelt.
Het bestuur roept daarom de leden op zich voor deze functie beschikbaar te stellen.
Aanmeldingen graag per e-mail naar een van de volgende adressen :

voorz@basic-gg.hcc.nl
secr@basic-gg.hcc.nl
penm@basic-gg.hcc.nl

Mijn variabele werkt niet!

Programmeertaal BASIC heeft nooit moeilijk gedaan met variabelen. Variabelen waren eerder altijd globaal, ze waren overal in het programma te gebruiken zonder ze hoeven door te geven. We hoefden er niet naar om te kijken en wanneer we ze weer wilden gebruiken, moesten we even nadenken: wat was de naam ook alweer?

Er zijn nog steeds BASIC versies en dialecten die niet moeilijk doen en we gewoon een variabele zonder poespas mogen gebruiken. Dat mag bijvoorbeeld in PowerBASIC.

Tegenwoordig speelt het declareren van variabelen ook een rol. Er zijn nu grenzen waarmee we eerst over moeten nadenken hoe we de variabelen gebruiken. Er zijn twee soorten declaraties die op bepaalde plekken wel en niet zijn toegestaan:

- gebruik `DIM` wanneer het u uitkomt om een variabele te gebruiken (normale BASIC versies en dialecten);
- gebruik `Dim` in methoden om variabelen te gebruiken (Visual Basic en andere Windows Basic versies), variabelen declaren met `Public` is in methoden niet toegestaan;
- een `Public` array declareren in een klasse is niet toegestaan, gebruik `Private` om array-variabelen te declareren en maak e.v.t. `Item` functies om de array elementen door te kunnen geven;
- maak gebruik van lokale variabelen, die in Visual Basic .NET zeer nuttig zijn. Een variabele zal in een methode niet meer in de hele methode bereikbaar zijn. U kunt zelfs lokale variabelen declareren binnen in een lus, zie voorbeeld hierboven bij Programma 2: het Moiree-effect. Variabele J zal buiten de lus niet meer gebruikt kunnen worden.

Marco Kurvers

BASIC cursus: Liberty BASIC (2)

Dit is het tweede deel uit de serie. Hierin zullen we een bestand opzoeken en openen en de inhoud op het scherm plaatsen. Ik laat eerst een logo zien. Bijna aan elk plaatje, lettertype, enz. zit een copyright en gebruiksaanwijzingen. Ik heb het volgende plaatje op het Internet laten genereren voor eigen gebruik.

U kunt dit plaatje vanaf het Liberty BASIC forum downloaden. Ga, om het plaatje te downloaden, naar het onderstaand adres en gebruik het gerust in je eigen listings.

Adres: http://www.libertybasic.nl/viewtopic.php?f=15&t=428

Plaats het plaatje in de map van Liberty BASIC of maak eerst een aparte oefenmap onder Liberty BASIC. We besluiten ons programma in een grafische omgeving te schrijven. Microsoft heeft (met Visual Basic) voor een standaard venster gezorgd. In een dergelijk venster kunnen we plaatjes plaatsen als we eerst binnen het algemene venster een graphicbox definiëren waarin we de plaatjes zullen plaatsen.
We besluiten echter de Liberty BASIC grafische omgeving te gebruiken. Dan hoeven we geen graphicbox vooraf te definiëren, omdat het hele venster al een grafisch veld is.

```
Open "Ledenadministratie hcc BASIC IG" for graphics_nsb as #main1
```

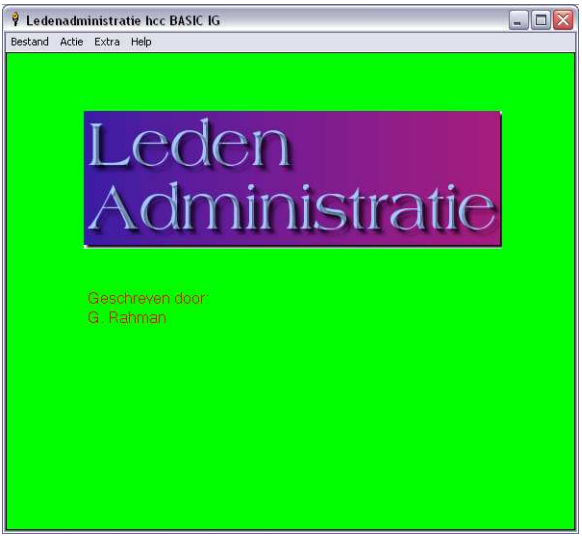
In VB termen zouden we eerst een standaard venster moeten ontwikkelen met:
`Open "Ledenadministratie hcc BASIC IG" for window as #main1`
En daarvoor zouden we een graphicbox moeten definiëren met:
`graphicbox #main1.g, 0, 0, WindowWidth, WindowHeight`
Maar door in een keer een grafisch venster te openen kunnen we het definiëren van een graphicbox overslaan. Dat heeft weer voordelen bij het plaatsen van plaatjes. Het heeft ook enige voordelen bij het plaatsen van vaste teksten. Voor het plaatsen van teksten zijn nu **vooraf** ook geen statictext boxen meer nodig. Wel moeten we nog steeds aangeven waar de tekst in het venster geplaatst moet worden, maar we kunnen nu het lettertype en de tekstkleur direct beïnvloeden.
Kijk eens goed naar de onderstaande listing waarbij een standaard venster met grafische box en een grafisch venster worden vergeleken.

```
WindowWidth = 400
WindowHeight = 350
Open "Ledenadministratie hcc BASIC IG" for graphics_nsb as #main1
```

```
#main1 "place 30 30"
#main1 "\voorbeeldtekst"
#main1 "flush"

WindowWidth = 400
WindowHeight = 350
Graphicbox #main2.gr, 0, 100, WindowWidth, WindowHeight - 100
Statictext #main2.stat, "", 30, 30, 200, 20
Open "Ledenadministratie hcc BASIC IG" for window as #main2
#main2.stat "voorbeeldtekst"
Wait
```

In de listing valt verder op dat je alle ruimten voor plaatjes vooraf moet kennen als je gebruik wilt maken van een standaard venster. Die ruimten moet je vooraf declareren. In sommige programmeertalen moet je daarnaast ook de variabelen vooraf declareren en voorafgaan van de juiste nauwkeurigheidsgraad voorzien. Het commando `flush` is nodig, omdat de tekst op het scherm anders niet zichtbaar blijft als het tweede venster `#main2` boven het eerste venster `#main1` komt te staan.



Figuur 1

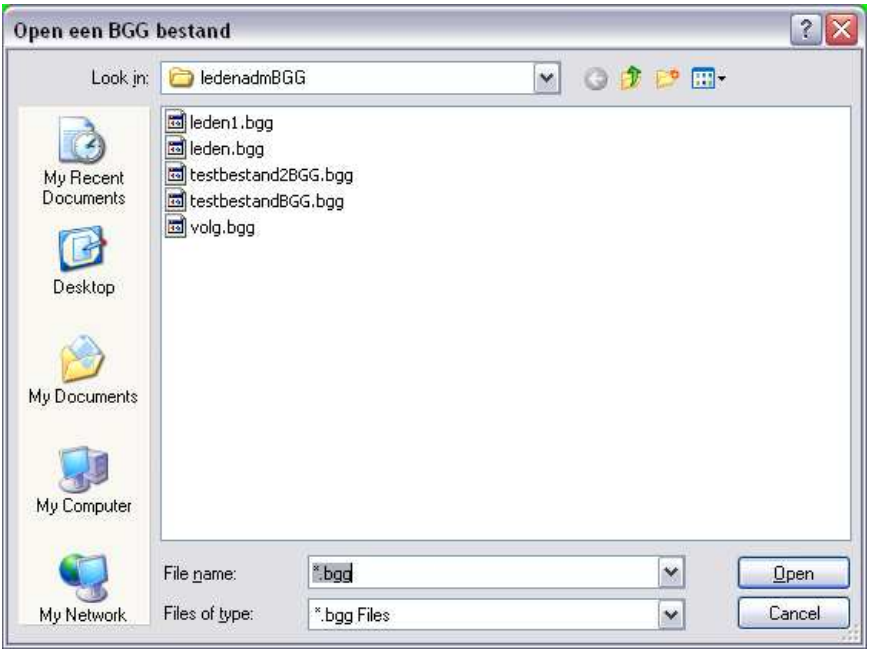
Zo zou het eerste scherm van ons leden programma eruit kunnen zien. Ik heb gekozen voor een groen scherm met een plaatje erop. Verder heb ik een menubalk aangemaakt met 4 pulldown menu's.

Bestand – menu met:	Open BGG bestand	-> [openen]
	Toon BGG lijst	-> [toonlijst]
	Exit	-> [quit1]
Actie – menu met:	Records aanvullen	-> [wijzigen1]
	Records sorteren	-> [sorteren1]
Extra – menu met:	Adressen uitprinten	-> [adressen1]
Help – menu met:	Help onderwerpen	-> [help1]
	Versie	-> [versie]

Het plaatsen van menu's doen we in Liberty BASIC als volgt:
Schrijf voor elk menu een nieuwe opdracht waarin je aangeeft in welk venster het menu geplaatst moet worden. Hier volgt een voorbeeld.

```
Menu #main1, "Bestand", "Open BGG bestand", [openen1], "Toon BGG
lijst", [toonlijst1], "Exit", [quit1]
Menu #main1, "Actie", "Records aanvullen", [wijzigen1], "Records
sorteren", [sorteren1]
Menu #main1, "Extra", "Adressen uitprinten", [adressen1], "Mailing",
[mailing1]
Menu #main1, "Help", "Help onderwerpen", [help1], "Versie", [versie]
```

Ik behandel in deze aflevering nog het opzoeken van een bestand met behulp van het file dialoog commando. Als de gebruiker straks via de menubalk Bestand -> Open BGG bestand kiest, dan vervolgt het programma met de commando's die volgen na de label `[openen1]` totdat een `Wait` statement bereikt wordt. In mijn listing springt de uitvoering naar de label `[openen1]`, waar het eerste commando dat de computer tegenkomt het commando `filedialog` is.



Figuur 2.

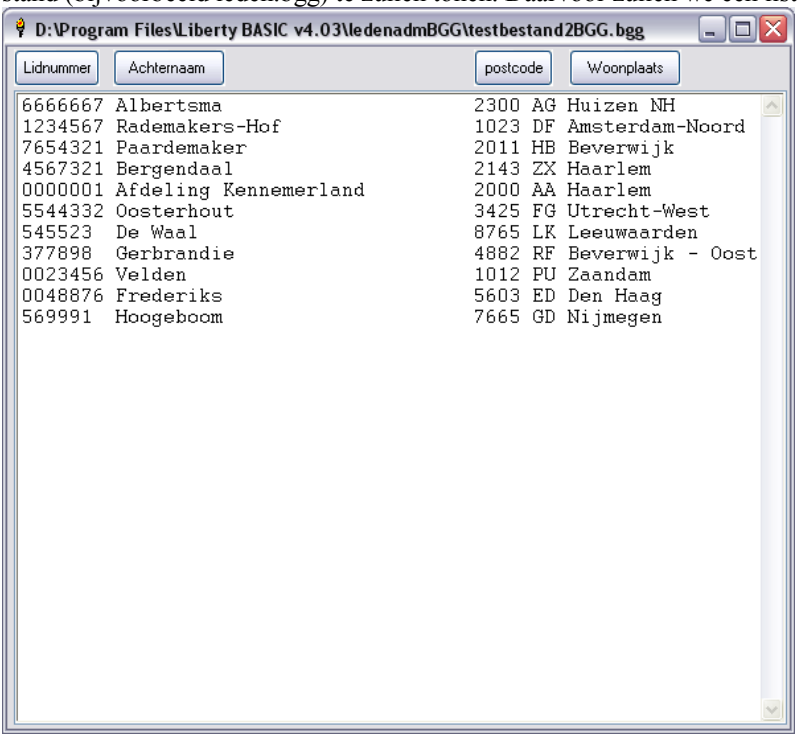
Om bovenstaande Figuur 2 te produceren, volstaan de volgende regels:

```
[openen1]
   filedialog "Open een BGG bestand", "*.bgg", bestand$
   if bestand$ <> "" then goto [verzamelItems]
   Wait
```

De vertaalde help tekst voor het `filedialog` commando vind je op:
http://www.libertybasic.nl/filedialog1.htm
Natuurlijk kun je van alle Liberty BASIC commando's een goede omschrijving in het Helpmenu vinden. Voor een Nederlandse vertaling adviseer ik je om een kijkje te nemen op: www.libertybasic.nl/index1.html
Filedialog kent een vorm met een OPEN en CANCEL knop of een vorm met een SAVE en CANCEL knop. Deze beide vormen worden getoond afhankelijk van de aanwezigheid van het woordje "open" of "save" in de titelomschrijving van het filedialog venster. Hier volgt een oplossing waarbij de OPEN variant wordt getoond terwijl in de titelomschrijving het woordje open ontbreekt.

```
filedialog "Kies een BGG bestand"+chr$(0)+"open", "*.bgg", bestand$
```

Op enig moment heb ik beloofd nu ook een scherm met de inhoud van het geopende bestand (bijvoorbeeld leden.bgg) te zullen tonen. Daarvoor zullen we een listbox gebruiken.



Figuur 3.

We zullen nu wel wat meer moeten doen om Figuur 3 te kunnen produceren. In Figuur 3 is de listbox duidelijk te herkennen. Boven de listbox heb ik 4 knoppen geplaatst om in een later stadium de sorteeropdracht voor de betreffende kolommen te kunnen geven. Liberty BASIC heeft een ingebouwde sorteerroutine om array's (rijtjes) te sorteren. Hier vind je wat hulp http://www.libertybasic.nl/sortArray1.htm (deze tekst is een vertaling van de tekst in de helpfile). We laten het sorteren voor wat het is. Dat komt in de volgende aflevering goed ter sprake. Om Figuur 3 te tonen moeten we eerst het hele bestand bgg inlezen en bepaalde gegevens uit elk record in een array plaatsen.

```
[verzamelItems]
Open bestand$ for random as #r len = 163
Field #r, 3 as LIDCODE$,_
        7 as LIDNUMMER$,_
        30 as NAAM$,_
        10 as VOORVOEG$,_
        10 as VOORLETT$,_
        25 as ADRES$,_
        7 as POSTC$,_
        25 as PLAATS$,_
        3 as LAND$,_
        20 as TEL$,_
        6 as LIDSINDS$,_
        3 as AFD$,_
        2 as MAILING$,_
        6 as MUT$,_
        6 as OPZEG$

aantalRecords = lof(#r)/163
DIM rs$(aantalRecords)

For Recnummer = 1 to aantalRecords
   GET #r, Recnummer
   rs$(Recnummer) = LIDNUMMER$+" "+NAAM$+POSTC$+" "+PLAATS$
Next Recnummer
Wait
```

Oké, we hebben het eerder gevonden bgg bestand geopend en we hebben meteen de aantal records bepaald met behulp van `aantalRecords = lof(#r)/163`. LOF staat voor "length of file". Die functie geeft de lengte (grootte) van een bestand in bytes weer. De functie werkt met de handle van het geopende bestand. In dit geval heb ik de handle genoemd: `#r`, gewoon omdat ik het vele tikken zat was.
Array's moeten ook in Liberty BASIC vooraf gedimensioneerd worden. Dat gebeurd in mijn geval met `DIM rs$(aantalRecords)`.
Tenslotte lees ik alle record items (FIELDs) in met een `FOR ... NEXT` loop. Na het inlezen staan nu alle gegevens in de array `rs$`, die ik straks wil tonen. In de volgende afleve-

ring zal ik de record gegevens "fields" in een twee dimensionale array plaatsen. Daarmee kan ik die velden per items groep sorteren.

Het volgende stukje listing spreekt voor zichzelf. Hiermee plaats ik gewoon de listbox in een venster `#main3` en plaats ik vier knoppen (Lidnummer, Achternaam enz.) boven in het venster. Het is verstandig om als lettertype COURIER te gebruiken als je de gegevens in een listbox netjes in kolommen wilt plaatsen. Dit lettertype en andere niet-proportionele lettertypen hebben als voornaamste eigenschap dat alle letters, zoals de "i" en de "w", even breed getoond worden, waardoor teksten goed onder elkaar geplaatst kunnen worden.

```
[toonlijst1]
button #main3.b1, "Lidnummer", [Lnum],ul,5,2,65,28
button #main3.b2, "Achternaam ", [Lnum],ul,80,2,85,28
button #main3.b3, "postcode", [Lnum],ul,353,2,60,28
button #main3.b4, "Woonplaats", [Lnum],ul,425,2,85,28
listbox #main3.list1, rs$(),[list],5,33,585,480
Open bestand$ for Window as #main3
#main3.list1 "singleclickselect"
#main3.list1 "font courier 12"
#main3.list1 "reload"
Wait
```

Er is veel gratis documentatie inclusief enkele gigabytes aan Liberty BASIC listings op het Internet te downloaden. Raadpleeg de programmeurs encyclopedie wikispace site even op http://lbpe.wikispaces.com/ voor de Engelse Liberty BASIC nieuwsbrieven. Natuurlijk is er ook een gewone wiki site http://basic.wikispaces.com/ waar je zelf iets mag publiceren. Trouwens waarom schrijf je geen artikel voor onze Nieuwsbrief? De redacteur zit met smart te wachten.

Stuur een e-mail naar info@libertybasic.nl en ontvang een link voor een gratis "Liberty BASIC cursusboek" en een gratis "Liberty BASIC naslagwerkboek". Beide documenten zijn ongeveer 100 pagina's groot.

Gordon Rahman

Cursussen

Qbasic: Cursus, lesmateriaal en voorbeelden op CD-ROM € 6,00 voor leden. Niet leden € 10,00.
QuickBasic: Cursusboek en het lesvoorbeeld op diskette,
€ 11,00 voor leden. Niet leden € 13,50
Visual Basic 6.0: Cursus, lesmateriaal en voorbeelden op CD-ROM,
€ 6,00 voor leden. Niet leden € 10,00
Basiscursus voor senioren, Windows 95/98,
Word 97 en internet voor senioren, (geen diskette). € 11,00 voor leden. Niet leden € 13,50

Software

Catalogusdiskette,	€ 1,40 voor leden. Niet leden € 2,50
Overige diskettes,	€ 3,40 voor leden. Niet leden € 4,50
CD-ROM's,	€ 9,50 voor leden. Niet leden € 12,50

Hoe te bestellen

De cursussen, diskettes of CD-ROM kunnen worden besteld door het sturen van een e-mail naar penm@basic-gg.hcc.nl en storting van het verschuldigde bedrag op:

ABN-AMRO nummer 49.57.40.314
HCC BASIC ig
Haarlem

onder vermelding van het gewenste artikel. Vermeld in elk geval in uw e-mail ook uw adres aangezien dit bij elektronisch bankieren niet wordt meegezonden. Houd rekening met een leveringstijd van ca. 2 weken.
Teksten en broncodes van de nieuwsbrieven zijn te downloaden vanaf onze website (http://www.basic.hccnet.nl). De diskettes worden bij tijd en wijlen aangevuld met bruikbare hulp- en voorbeeldprogramma's.
Op de catalogusdiskette staat een korte maar duidelijke beschrijving van elk programma.

Alle prijzen zijn inclusief verzendkosten voor Nederland en België.

Vraagbaken

De volgende personen zijn op de aangegeven tijden beschikbaar voor vragen over programmeerproblemen. Respecteer hun privé-leven en bel alstublieft alleen op de aangegeven tijden.

Waarover	Wie	Wanneer	Tijd	Telefoon	Email
Liberty Basic	Gordon Rahman	ma. t/m zo.	19-23	(023) 5334881	grahman@planet.nl
MSX-Basic	Erwin Nicolai	vr. t/m zo.	18-22	(0516) 541680	basic@lordthanatos.com
PowerBasic CC	Fred Luchsinger	ma. t/m vr.	19-21		f.luchsinger@kader.hcc.nl
QBasic QuickBasic	Jan v.d. Linden				j.vd.linden@kader.hcc.nl
Visual Basic voor Windows Visual Basic .NET	Jeroen v. Hezik Marco Kurvers	ma. t/m zo. do. t/m zo.	19-21 19-22	(0346) 214131 (0342) 424452	j.a.van.hezik@kader.hcc.nl m.a.kurvers@hccnet.nl
Basic algemeen, zoals VBA Office Web Design, met XHTML en CSS	Marco Kurvers	do. t/m zo.	19-22	(0342) 424452	m.a.kurvers@hccnet.nl