



datum: juni 1985

auteur: E. Nijeboer en R. Schimmelpenninck

titel: Kinderen en de computer - over verantwoorde invoering van computers in het primair onderwijs

De auteurs in dit artikel pleiten voor het geven van een onderwijskundig verantwoorde manier van informatica in het primair onderwijs.

Kinderen en leerkrachten moeten de mogelijkheden van de computer als werktuig zodanig in de vingers krijgen, dat ze er op eigen wijze creatief mee om kunnen gaan en daardoor mondig blijven of worden ten opzichte van de informatietechnologische ontwikkelingen in de samenleving. Kritiek wordt voorts geleverd op het aan het onderwijs voorgehouden nut van zogenaamde C.A.I.-programma's. Uiteengezet wordt de opzet van een programma voor kinderen om op een natuurlijke wijze te leren omgaan met een computer als werktuig, dat wil zeggen: dat zij de computer leren bestuderen en programmeren. Vervolgens wordt verslag gedaan van een onderzoek naar de vraag of kinderen - van bijvoorbeeld een vierde klas - dat wel kunnen.

Hoewel het onderzoek niet afgerond is, is het toch de bedoeling van dit artikel een bijdrage te leveren tot de discussie in welke richting de informatica-ontwikkeling zou moeten gaan.

Temeer omdat er veel gevaren signaleerd worden in de huidige officiële ontwikkelingen, zoals die zijn verwoord in allerlei artikelen en rapporten.

I Uitgangspunten en overwegingen

Bij het overdenken van de vraag: "Waarom computers invoeren in het (basis)onderwijs en hoe ze te gebruiken?" doen we de volgende constatering:

- Informatietechnologische toepassingen spelen een steeds grotere rol in de samenleving. Steeds meer instanties en bedrijven maken gebruik van computers. Ook worden veel "huiscomputers" verkocht.
- In de verwachtingen van mensen ten aanzien van de toekomst speelt de computer een belangrijke rol. Het idee dat men met de computer moet kunnen werken, om bij te blijven, leeft. Terecht of onterecht, dat doet niet ter zake. De invloed van die verwachtingen is ook op de kinderen groeiende.

- Bovengenoemde factoren spelen een steeds belangrijker rol in het onderwijs en dat wordt nog gestimuleerd door de overheid. Zij spreekt al van "versnelde invoering" van informatica in het onderwijs, omdat Nederland niet achter dient te blijven
.....

Uitgangspunt 1: Over de houding van het onderwijs met betrekking tot informatica

Het onderwijs moet deze ontwikkelingen niet negeren - als dat al zou kunnen - maar ook niet zomaar over zich laten komen.

Het onderwijs heeft wel als taak op onderwijskundig verantwoorde manier in te spelen op ontwikkelingen in de samenleving.

Uitgangspunt 2: Mondigheid

De informatietechnologische ontwikkeling dient zodanig binnen het onderwijs aan de orde te komen dat de jeugd mondig wordt en blijft ten aanzien van deze en toekomstige ontwikkelingen. De (toekomstige) samenleving beheerst uiteindelijk de ontwikkelingen. Dat betekent dat in het onderwijs op een creatieve wijze functioneel met de computer omgegaan moet worden, door leerkrachten én door de kinderen. Om dit te bereiken dient een didactisch verantwoord leerproces gestart te worden.

Overweging ten aanzien van invloeden op scholen

Scholen krijgen allerlei folders toegestuurd en er verschijnen veel artikelen over de computer in het onderwijs, waarin o.a. de volgende suggesties worden gewekt:

- Uiteindelijk ontkom je er niet aan één of meerdere computers aan te schaffen, wil je bijblijven.
- Schaf dan zulke aan waar "veel mee te doen valt". (Die zijn uiteraard vrij kostbaar.)
- Er komen steeds meer programma's (software) beschikbaar, waar kinderen mee kunnen werken. Vooral op de gebieden taal en rekenen, al zijn er ook spelletjes en simulaties. De programma's zijn voor het overgrote deel instructieprogramma's; de zogenaamde C.A.I. (Computer Assisted Instruction). Men spreekt van oefenprogramma's, waarbij termen als "drill and practice" worden gehanteerd en soms wordt gezegd dat deze programma's (didactisch) nog niet zo goed zijn, maar ze zullen worden verbeterd en verfijnd.

Wij vinden, dat wie hierop zonder meer ingaat, de ontwikkelingen "over zich laat komen".

Het is de vraag of men zo een goed idee krijgt van wat een computer inhoudt en wat je er eigenlijk allemaal mee zou kunnen doen. Door alleen met CAI-programma's te werken, krijgt men geen greep op de computer. Men kan dan alleen maar reageren op wat de "computer zogenaamd doet en vraagt". Het is en blijft een wonderlijk apparaat, zeker voor kinderen. Het apparaat "bestuurt" de kinderen en "waardeert" hen. In plaats van andersom.

Gezien de genoemde uitgangspunten is het op deze wijze kennis te maken met informatica een zeer twijfelachtige zaak. Nog afgezien van het feit, dat veel CAI-programma's didactisch gezien slecht zijn. Dit komt doordat juist "trucjes" bij rekenen - relatief bekeken - het makkelijkst te programmeren zijn en dat staat de ontwikkeling van een goede reken-wiskundedidactiek vaak in de weg. Het is nog maar de vraag of de ontwikkelaars van programma's - de softwarebureaux - onderwijskundig-didactische kwaliteiten hebben.

Er zijn nog meer fundamentele bezwaren verbonden aan het alleen werken met CAI. Verderop komen ze aan de orde, nadat we eerst nog weer enkele uitgangspunten formuleren.

Uitgangspunt 3: Wat is een computer in wezen?

Computers zijn in wezen werktuigen, hulpmiddelen. En wel op het gebied van informatieverwerving. Men krijgt vat op de functie en het wezen van de computer, indien men de gelegenheid krijgt die op creatieve wijze als werktuig te leren hanteren. Men leert de computer opdrachten te geven in plaats van andersom (zoals bij de CAI). Daarbij leert men uiteraard wat de (voorlopige) mogelijkheden en onmogelijkheden zijn van zowel het werktuig als van de gebruiker.

Dit leidt tot de doelstelling dat men de computer moet leren programmeren, dat geldt zowel voor leerkrachten als voor kinderen. Daardoor verkrijgt men praktisch inzicht in wat een computer wel en wat zij niet kan. Dat opent weer de mogelijkheid creatief met de computer om te gaan en haar op eigen wijze en op eigen niveau toe te passen; uiteraard in een leerproces.

Overwegingen:

Mocht men, tijdens het leren programmeren, na enige tijd ook met CAI-programma's gaan werken, dan passen die al enigszins in een begrijpelijk kader. Men heeft dan het inzicht dat iemand anders (met een bepaald doel) de computer geprogrammeerd heeft.

Het "magische karakter" is dan al niet meer zo groot. Als kinderen programmaatjes maken, dan komt het ook voor dat ze elkaars programma's bekijken. Ook kunnen

kinderen voor elkaar eenvoudige "hulpprogramma's" maken, bijvoorbeeld om sommen te controleren of op te geven. Ze begrijpen dan hoe één en ander in elkaar zit en ze kunnen er veranderingen in aanbrengen. Naast hulpprogramma's zijn ze in staat andere programma's te maken, zoals verjaardagslijsten, lijstjes met betrekking tot sport, documentatie, boeken, e.d.

In welke computertaal gaan we leren programmeren?

De meeste computers die eenvoudig (en relatief goedkoop) zijn en die het meest voorkomen in de omgeving van het kind, werken met "Basic". Deze taal is vrij eenvoudig te leren.

Mede gezien de voorafgaande en volgende uitgangspunten kiezen wij voor Basic.

Een andere computertaal, Logo, is speciaal voor kinderen ontworpen. Als men een computer heeft, waarop deze taal zit, kan er ook mee gewerkt worden. Maar we adviseren er niet mee te beginnen want die taal komt nauwelijks in de maatschappij voor en geeft geen direct inzicht in hoe een computer voor allerlei eigen toepassingen te programmeren is.

Gezien de uitgangspunten is het een nadeel, dat deze taal speciaal voor kinderen bedoeld is.

Uitgangspunt 4: Hoe een computertaal te leren?

Het leren omgaan met nieuwe hulpmiddelen, zoals het leren programmeren van de computer kan het best op een natuurlijke manier gebeuren. De computertaal wordt dan, net als een gewone taal, gefaseerd aangeleerd, te beginnen bij de eenvoudigste, maar veel voorkomende belangrijke woorden. De woordenschat breidt zich langzamerhand uit en de structuur wordt ingewikkelder. In elk stadium is de taal functioneel voor de lerende zelf en op zijn eigen niveau. Vanaf het begin kan men met het geleerde iets doen. Als is het maar een t.v.-scherm volschrijven met woorden, zinnen of sommen. Naarmate de (computer)taalbeheersing groter wordt en uitgebreider, ervaart men dat men er steeds meer mee kan doen.

Al gauw vallen in dit leerproces programma's gemaakt door anderen in een begrijpelijk kader, al hoeft men deze niet helemaal te kunnen volgen. Net zoals "gewone taal" van iemand die een uitgebreidere woordenschat heeft, niet direct te begrijpen hoeft te zijn, maar toch in zekere mate verstaanbaar is en dus geen "magisch" karakter heeft.

Verderop beschrijven we hoe we op de basisschool proberen de kinderen te leren programmeren op een zo natuurlijk mogelijke wijze.

Overwegingen

Dat we daarbij de kinderen "helpen" en het leerproces richting geven, is niet in strijd met het natuurlijke leren. In het dagelijkse leven ondervinden kinderen invloed van de handelwijze van volwassenen. Ze zien hoe werktuigen gebruikt worden en willen imiteren. Al gauw worden die werktuigen op eigen wijze en voor eigen doeleinden gebruikt. Zij leren bijvoorbeeld ook schrijven van ouderen, maar dat wil niet zeggen dat ze ook alleen maar zinnen schrijven die volwassenen precies voorzeggen.

Er wordt ons weleens verweten dat we van kinderen en leerkrachten beroepsprogrammeurs zouden willen maken. Zover willen we zeker niet gaan. Het verwijt is net zo min juist, als wanneer men beweert dat men kinderen bij het aanleren van een taal tot taalkundigen opleidt.

II Gevaren van de huidige ontwikkelingen rond informatica in het primair onderwijs

In de ontwikkeling, zoals die zich tot nu laat aanzien, wordt nergens voldoende aandacht besteed aan of gedacht over het programmeren door kinderen. Steeds wordt gesproken over oefen- en hulpprogramma's, door ons gemakshalve onder de verzamelnaam "CAI" genoemd. Alleen Logo komt ter sprake.

Naast het reeds eerder genoemde bezwaar tegen CAI dat men geen "greep" krijgt op de computer en op informatica, treedt het effect van mystificatie op.

Het is net alsof men met pen en papier werkt, maar daarbij steeds een "inzichtbaar sturende hand" ervaart, zonder te weten dat hij stuurt.

Laten we de computer vergelijken met een hulpmiddel, zoals een stuk papier dat is, waarop gedachten, feiten, processen gevisualiseerd en vastgelegd kunnen worden.

Met de boekdrukkunst, de radio en de t.v. is er een scheiding tussen zender en ontvanger tot stand gekomen.

Het programma op de t.v. is duidelijk zichtbaar gemaakt. De makers worden naderhand bij name genoemd. Het boek draagt voorin de naam van de schrijver en omdat de ontvanger ook kan schrijven (zenden) en hij de principes van de boekdrukkunst kent, zal hij de persoon achter het boek nog kunnen herkennen. (Hoewel mystificatie ook hier optreedt, aangezien vaak wordt gedacht, dat wat zwart op wit staat, waar is.)

Revolutionair op het communicatieve vlak is nu, dat de zender door de ontwikkeling van de computer direct kan inspelen op de signalen van de ontvanger. De zender blijft buiten beeld, maar kan via de computer in het programma interacteren.

Je ziet nu al dat typistes achter de terminal door de computer gewezen worden op fouten en dat de computer vragen stelt en klachten afhandelt.

Een ondefinieerbaar gevoel van machteloosheid ontstaat juist door de combinatie "interacterend en voorwerp", zonder kennis hoe de zender dit stuk "papier" heeft kunnen laten interacteren. Agressie wordt gekanaliseerd op de computer.

De mystificatie en daarmee machteloosheid, wordt nu in het basisonderwijs geïntroduceerd door middel van de computer, die interacterende taal- en rekenprogramma's met de kinderen gaat doen.

Een niet-levend voorwerp wordt zo een almachtige meester, die reageert en een pluim geeft als ze het wel goed gedaan hebben. Juist in het laatste geval, de computer als externe motivator, is de mystificatie het grootst.

Juist het niet weten en begrijpen waar het vandaan komt en hoe dit kan, maakt dat kinderen hier heel gevoelig voor zijn. Kinderen, die al eenvoudige programma's hebben gemaakt, merken op dat je alles wat de computer moet doen, tegen hem moet zeggen. En dan nog wel heel precies ook, anders kan hij het niet. Eén van de eerste ontdekkingen is, dat bij het schrijven van een zin op het scherm middels de Print-opdracht, waarbij spelfouten worden gemaakt, de computer deze niet verbetert, maar gewoon overneemt. In het begin zijn kinderen daar vaak verbaasd over. Door zelf programmeren - hoe eenvoudig ook - valt voor velen de computer van zijn wonderlijk voetstuk, en wordt tot werktuig. Als je er goed mee kunt omgaan een veelzijdig werktuig.

Een ander risico dat men loopt wanneer in de huidige ontwikkeling van informatica de nadruk op "CAI" ligt en niet op zelf programmeren, is dat men afhankelijk wordt van softwarebureaux.

Bovendien kunnen veel CAI-programma's alleen maar op één merk computer gedraaid worden. Dan ben je dus ook nog grotendeels afhankelijk van een bepaald merk hardware.

De huidige ontwikkeling loopt kans op dood spoor te geraken, indien men zich toelegt op toepassing van CAI en men een (dure) computer aanschaft die "alles" kan. In het primair onderwijs kan men zich niet veroorloven vrij veel van die apparaten aan te schaffen. Men is afhankelijk van de software-mogelijkheden voor die computers en men kan ze slechts toepassen op die punten, waar software voor is. Als je niet kunt programmeren kun je ze nergens anders voor gebruiken. Veel leerkrachten hebben er geen verstand van, ze krijgen het idee dat zo'n prachtig in elkaar gezet

CAI-programma toch niet te maken valt. Andere leerkrachten, die wel de gevaren van CAI en het gebrek aan didactische kwaliteit ervan zien, vormen een terechte stroming binnen het onderwijs, waarvoor het "allemaal niet hoeft".

Door de "versnelde invoering" van informatica in het basisonderwijs is er te weinig tijd voor goede door-denkning van één en ander en voor een gerichte bij-scholing.

Illustratief zijn de volgende reacties van schoolhoofden bij computertentoonstellingen: "Ik ben door mijn bestuur gestuurd, want zij vinden dat de school iets aan computers moet doen". "Ik kijk rond, want de ouders hebben een bazar georganiseerd voor een computer en we hebben nu f 2.000,--. Kan ik er dan wel een kopen?" "Ik weet het nog niet wat we ermee gaan doen. Kan de administratie er ook op?"

Wij zijn bang dat op veel scholen de computer na enige tijd in de kast wordt gezet en dat alleen "hobbyisten" er soms iets mee doen.

Een dergelijke ontwikkeling zouden wij zeer betreuren, want we vinden dat het onderwijs op verantwoorde manier moet inspelen op belangrijke ontwikkelingen in de samenleving, zoals we dat verwoord hebben in onze uitgangspunten.

Wanneer naar ons idee kinderen en leerkrachten zelf leren programmeren, is men minder afhankelijk van software-bureaux, omdat dan het zwaartepunt niet bij CAI-toepassing ligt. Men is beter in staat de computer toe te passen op eigen specifieke problemen. Bovendien is het ook niet nodig een kostbare computer aan te schaffen. Een relatief goedkope voldoet uitstekend. We denken bijvoorbeeld aan de ZX 81 en de Spectrum van Sinclair. Men kan er daarvan meer dan één aanschaffen en is zo breder inzetbaar. Deze computers hebben als voordeel dat opdrachten, zoals PRINT, GOTO, INPUT al op de toetsen vermeld staan. Zelfs zijn voor de Spectrum vrij veel onderwijsprogramma's beschikbaar, waardoor dus ook CAI toepasbaar is als men dat wil.

III Kunnen kinderen programmeren?

Computertaal is een specifieke taal. Er zit een bepaald soort logica in. Kunnen kinderen deze logica aan? En kunnen ze die op eigen wijze creatief gebruiken? Is er formeel denken voor nodig of niet? Volgens Piaget zijn kinderen van ± 9 à 10 jaar in deze fase. Op een Amsterdamse lagere school, de Rijk Kramerschool, zijn we in 1983 gestart met een project om kinderen van klas 3 te leren programmeren volgens de principes, verwoord in ons vierde uitgangspunt. Het leerprogramma is zo natuurlijk mogelijk opgezet, waarbij men eerst de betekenis van enkele belangrijke

toetsen leert waardoor men er al gauw zelf iets mee kan doen. In het begin zijn de opdrachten vrij gesloten, maar die worden steeds opener van karakter. Geleerd wordt: hoe kun je met de computer schrijven; hoe onthoudt de computer; hoe kan hij tekeningen maken; hoe verander je een programma?

Het programma was als volgt gefaseerd:

1. Inleiding

Alle twintig kinderen leren in groepjes van twee de allereerste beginselen van het werken op de computer (ZX 81). Dat gaat zo:

Een gedeelte van het toetsenbord wordt afgedekt.

- Op het t.v.-scherm de uitkomsten opschrijven van getalbewerkingen (+, -, x, /). Instructies: "print en new line" (de zogenaamde "doe-knop").

Bij invoer van foute gegevens: "Uitgummen/Rub-out". Sommige tekens staan vermeld in rode kleur. Daarbij shiftknop ingedrukt houden.

Via het afgedekte toetsenbord.

Schrijven van de tekst op het scherm. Tekst tussen "aanhalingstekens". Space-toets.

- Eenvoudige programma's opstellen.

Regelnummers. 1, 2, enz. Wat inhoudt: hij doet het straks in volgorde. Leest 1, dan 2, etc.

Bij uitvoering "run" en dan "doe-knop".

Instructie "goto (regel nr?)".

Hoe loopt het achtereenvolgens? Goto 1: alles wordt gerepeteerd, goto 2 bijvoorbeeld, niet alles, etc.

Teksten komen onder elkaar op het scherm.

Hoe komen teksten (opgenomen per regelnummer) achter elkaar?

Door "achter de print".... "te zetten, dus print" ..."

- Ook instructies als "new" (alles uitwissen: scherm + programmeergeheugen) komen soms aan de orde. En de pijlen voor de cursor.

2. Klassikale consolidatie

Bovenstaande wordt klassikaal herhaald, gevarieerd en geconsolideerd. Eventueel volgt enige uitbreiding, bijvoorbeeld een werkkaart maken.

3. Vervolg basisvaardigheden

Het functioneren van de geheugens moet in eerste instantie duidelijk worden:

- Het opslaan van getal gegevens. Eerst A=.... tot en met Z=.... Dit kan door middel van "raad- en verstopspelletjes".
- Het opslaan van tekst in geheugens ... Stringen.

A\$=".."; ..Z\$=";;" (NB: teksten weer tussen "tekens".) Raadspelletjes. Iets onthouden op volgorde. Namen van de kinderen?

- Maken van werkkaarten voor dit soort spelletjes.

Verdere belangrijke instructies zijn:

- De for..to..statement. Dit is een vervanging van de "goto-programma's" uit deel I. Uiteraard ook "next"..".

Men heeft dan al een idee van de lus.

Combinatie van lineaire stukken en lussen. Blokdigrammen.

- De "if..then...statement". Via de computer beslissen.

De voorwaarde na de "if" kan op getalgegevens slaan of op tekst. De regel waar naar toe, na "then", moet ingevuld worden. Indirecte adressering lijkt nu nog niet aan de orde.

- Combinaties van lineaire stukken, lussen, sprongen (onvoorwaardelijk goto) en beslissingen.

Verwerken van bovenstaande ook in werkkaarten (van gesloten en open type) ter herhaling en toepassing. Eigen ontdekkingen en uitbreidingen.

De verdere fasering hangt af van de tussenresultaten en bevindingen.

In elk geval willen we bekijken of de kinderen zelf toepassingen vinden, zoals het opstellen van verjaardagslijstjes, het controleren van sommen, op documentatiegebied, e.d.

Uitwerking en organisatie

Op dit moment leren de kinderen van klas 4 van de Dr. Rijk Kramerschool te Amsterdam programmeren op een ZX 81-computer. Dit gebeurt in een speciaal daarvoor in de klas opgezette computerhoek. De kinderen leren het programmeren door in tweetallen met behulp van werkkaarten aan de computer te werken.

De methode wordt ontwikkeld door Richard Schimmelpennink (in het kader van een afstudeerproject onderwijskunde) en Egbert Nijeboer (docent pabo Amsterdam).

Verantwoording

De dr. Rijk Kramerschool is gekozen omdat de auteurs goede contacten onderhouden met het team van deze school, waarvan enkele leden en ook een aantal ouders in het project geïnteresseerd bleken.

De 4e klas bleek de laagste klas te zijn, waar leesvaardigheid voldoende was om via werkkaarten aan de computer te werken. De eerstgekozen klas viel door

leesproblemen af. In een zo laag mogelijke klas beginnen heeft de bedoeling om aan te tonen dat de kinderen al op zeer vroege leeftijd op een voor ons wenselijke wijze met de computer geconfronteerd kunnen worden. Daarnaast geeft het de mogelijkheid om de methode zo "pupil-proof" mogelijk te maken. Als kinderen van de 4e klas ermee overweg kunnen dan moet het zeker in de 5e en 6e klas kunnen.

De ZX 81 werd gekozen omdat het de goedkoopste computer is (f 199,--, nu incl. 16kRAM) en de keywords (bijvoorbeeld PRINT) voluit op het toetsenbord staan. Een bijkomend voordeel is de geringe omvang, gepaard aan overzichtelijkheid. Alles zit in het ene apparaat. Een belangrijk nadeel is dat de ZX 81 geen gewone toetsen heeft, maar tactiel ononderscheidbare tiptoetsen. Je moet dus steeds controleren of het gewenste teken wel op het scherm is verschenen.

De werkkaarten als leermateriaal werden gekozen vanuit organisatorische overwegingen. Er zijn gemakkelijker kaarten toe te voegen of uit te halen. Onze didactische overweging was de relatie boek-schrift-computer. De werkkaarten moeten een onderdeel zijn van de methode, waarin ook plaats is voor klassikale behandeling, spelvormen, e.d. Tweetallen bleken in vergelijkbare projecten het meest succesvolle aantal.

De werkkaarten blijken goed te functioneren, hoewel het beperkte begrip leesvermogen in combinatie met het ongeduld syntaxis en de opbouw zeer goed verzorgd zijn. De observator bleek soms een remmende factor (vragende blik van: "doe ik het wel goed?"), maar ook vaak genoeg een reddende factor als het tweetal vastzat. Bijsturing van de betreffende kaart bleek dan weer noodzakelijk.

Huidige stand van zaken

Ondanks zijn beperkte mogelijkheden blijkt de ZX 81 voor ons doel toch goed te voldoen. Vergeleken bij zijn nieuwere broer, de Spectrum, is zelfs de kleinere rekensnelheid een voordeel. De kinderen zien per regel hoe een som wordt uitgerekend. Een nadeel is, dat de ZX 81 bij elke nieuwe opdracht het scherm schoonveegt. Het indrukken van de "toetsen" blijft onhandig. Kinderen zeggen: "Het is gemakkelijker als de knopjes eruit steken". Maar dit probleem blijkt toch overkomelijk.

Organisatie

Doordat enerzijds de keren, dat de kinderen achter de computer kunnen zitten, zeer beperkt zijn (twee keer

per week tweetallen tijdens het keuzeur en af en toe tussendoor) en anderzijds de ontwikkeling van de methode noodzaakt dat de kinderen per keer er langer achter zitten dan normaal, is de frequentie waarin de kinderen kunnen programmeren verre van ideaal te noemen. Het komt voor dat een kind meer dan twee maanden niet met de computer heeft gewerkt. Verbazingwekkend is dan overigens hoeveel het kind dan nog onthouden heeft. Dat betreft vooral de betekenis en de plaats van de eerder gebruikte toetsen. De kinderen zijn zeer gemotiveerd om met de computer te mogen werken.

Wat is bereikt

Alle kinderen kunnen nu de computer als rekenmachine gebruiken en een programma maken dat het hele scherm vol schrijft met hun naam en/of grafische symbolen (graphics). Ook kunnen ze al een simpel CAI-programma schrijven dat willekeurige tafels op het scherm schrijft.

Een aantal kinderen heeft een programma gemaakt, waarbij de computer om twee namen vraagt en die kwalificeert in "een lelijke naam" of "een mooie naam".

We hebben nog onvoldoende kunnen onderzoeken in hoeverre de kinderen een volledig begrip hebben van de gemaakte programma's. Opmerkingen tijdens het programmeren, zoals "De computer print nu maar één keer Jennefer en de rest Rebecca, omdat hij Rebecca een mooiere naam vindt" wijzen nog op het tegendeel. Duidelijk blijkt hier hoe ook tijdens het programmeren de mystificatie nog niet helemaal opgeheven is.

Hoe verder?

Onze opzet moet nog verder uitgebreid en bijgestuurd worden in eerste instantie met de opdrachten: FOR-NEXT, IF-THEN en RANDOM. De kinderen krijgen dan meer mogelijkheden zelf creatief met het programmeren om te gaan en zelf verdere toepassingen te verzinnen. Zoals: CAI-programma, spelletjes, een databank (bijvoorbeeld verjaardagslijstjes, e.d.).

Onderzoek moet gedaan worden naar het effect van het vermogen om programmeren op de autoriteitsgevoeligheid met betrekking tot de computer. Ook is het interessant na te gaan of zelf programmeren leidt tot een groter probleemoplossend vermogen.

IV Iets over het project "PIPO" van de V.U.

In Nederland bestaat het project PIPO, dat de kinderen ook laat programmeren. Het is een onderzoek van de

V.U. naar vergroting van de oplosvaardigheid door middel van programmeren. Daarin wordt in een tweetal scholen kinderen van een 6e klas in een speciaal hiervoor ontwikkelde programmeertaal "PIPO" (Programmeren in Pedagogische Omgeving) geleerd te programmeren. Evenals uit onderzoeken met betrekking tot LOGO is gebleken, zijn de effecten niet significant.

Het verschil tussen de programmeertaal PIPO en het door ons gebruikte BASIC is dat PIPO gebaseerd is op de betere gestructureerde, maar ook moeilijker taal PASCAL en dat bij PIPO de sleutelwoorden in het Nederlands zijn (bijvoorbeeld PRINT is bij PIPO "Geef!").

Hoewel de makers van PIPO onderkennen dat Engelse sleutelwoorden weinig problemen geven (zoals ook onze ervaring is) en zelfs minder mogelijkheden tot verwarring geeft, kozen ze toch voor het Nederlands in verband met de mindere graad van abstractie en verondersteld gemak bij het memoriseren. Uit de resultaten van het PIPO-onderzoek zijn weinig generaliseerbare conclusies te trekken, stellen de onderzoekers zelf. Tijdgebrek en moeilijke onderzoeksomstandigheden zijn daarvan de oorzaken. Voor de PIPO-onderzoekers is het nog steeds de vraag of het de moeite loont om in het basisonderwijs tijd vrij te maken voor het leren programmeren. Maar zij kiezen dan ook primair voor CAI.

Tenslotte:

Ons onderzoek is nog lang niet afgerond, maar toch hopen wij door middel van een onderwijskundige doordenking van informatica op de basisschool en aan de hand van onze ervaringen met kinderen en computers een bijdrage te leveren aan een goede onderwijskundige en maatschappelijke vormgeving voor de ontwikkeling van informatica.