

Boter, kaas en eieren anders.

De werkgroep "kunstmatige Intelligentie" van de afdeling Lelystad heeft als projekt genomen het spelletje boter, kaas en eieren waarbij de computer het spel al doende leert beheersen totdat hij onverslaanbaar is.

Deze aanpak verschilt wezenlijk van de meeste programma's waarin de ontwikkeling in het snelverloop al vastligt. Uitgaande van een gegeven situatie zal de computer dan altijd op de zelfde manier handelen. Dit handelen is bepaald door de kennis die de programmeur (systeem-analist) er als model inge-stopt heeft.

Bij kunstmatige intelligente programma's is het handelen echter afhankelijk van het geleerde en het is dus de voor-geschiedenis die bepaald welke zet de computer zal gaan doen.

Het is dit idee wat onze werkgroep verder probeert te ont-wikkelen.

TOEGEPASTE TRUKEN.

De computer moet bij kunstmatige intelligentie onthouden dat er voorgevallen is om dan daarna hieraan een conclusie te verbinden.

Bij "boter, kaas & eieren" worden dus de zetten van de computer, die een verloren spel hebben opgeleverd, negatief gewaardeerd.

Dit gebeurt d.m.v. een bij elke zet van de computer behorende preferentie.

Omdat het aantal mogelijke spelmomenten bij boter, kaas en eieren 3⁸ (= 19.682) bedraagt, is het zinvol dit te reduceren tot een werkbaar aantal patronen. Dit gebeurt d.m.v. spiegelen en draaien, immers veel spelmomenten zijn identiek m.b.t. strategie. De computer moet dus

gedraaide en gespiegelde patronen herkennen. Hiermee wordt het aantal teruggebracht tot max. 337.

Ieder spelmoment kan herkend worden aan een tritaire waarde (getal in het 3-tallige getalstelsel). En wel tritair omdat op iedere plaats niets, een nul of een kruisje kan staan.

Dit tritaire getal kan dan op zijn beurt weer omgezet worden naar een decimaal getal waarmee gewerkt kan worden.

PROGRAMMA-OPBOUW

Het computerprogramma is ingedeeld in een hoofd-programma en afgeleide sub-routines.

Het hoofd-programma is zo opgebouwd dat het zo dicht mogelijk bij het natuurlijke algoritme ligt. Iedere stap hierin wordt dan in een subroutine verder afgehandeld.

Tevens wordt bij aanvang van iedere routine wermeld welke variabelen en array's er aangepast dan wel gebruikt worden en welke tijdelijk gebruikt worden. Deze laatste zijn dus weer vrij beschikbaar na het doorlopen van deze routine. In het hoofd-programma zijn 3 fases te onderscheiden, te weten:

1. INITIALISERING (regel 0-99) van de toe te passen array's en variabelen.

2. HET SPELEN (regel 100-299), waarin de computer de menselijke tegenspeler (of andere computer) aan zet laat, om dan daarna zelf de beste tegenzet te doen (d.w.z. de zet die de hoogste preferentiewaarde heeft van al de mogelijke zetten).

Trouwens in dit programma speelt de computer altijd als tweede met een "0".

3. DE EVALUATIE (regel 300-999) van alle door de computer gespeelde zetten na afloop van het spel.

Zoals U zult zien is het programma geschreven in een soort universele BASIC, deze moet dus aan Uw machine worden aangepast. De door ons gebruikte BASIC is een mengeling van Acorn Atom en Sinclairs Spectrum.

Array's worden aangeduid met dubbele gelijke hoofdletters, variabelen met een enkele.

De tekening is grafisch opgebouwd met een resolutie van

64 x 48 blokjes. MOVE en DRAW zijn absoluut; tenzij anders aangegeven.

GEBRUIK VAN HET PROGRAMMA.

Omdat de opzet van het projekt niet zozeer het maken van een leuk spelletje betrof, maar een studie naar de toepassing vna kunstmatige intelligentie, is het programma bepaalt niet snel.

Niet alleen heeft de computer bedektijd nodig, maar voordat hij in staat is een beetje redelijk te spelen moeten er heel

HOOFD-PROGRAMMA.

```

1 REM *****
2 REM * BOTER, KAAS & EIEREN *
3 REM *****
4 REM WERKGRPEP A.I LELYSTAD
5 REM DATUM : 8-9-1983

10 REM INITIALISERING
20 DIM AA(9),BB(9),CC(8),DD(9),FF(337),GG(337),HH(9),XX(9)
40 FF(1)=99999
100 REM START SPEL
102 REM INIT
104 R=0:Z=0
106 FOR I=1 TO 9 : AA(I)=0 : NEXT I
108 GOSUB 1000 ;REM TEKEN SPEELVELD
110 REM VOLGENDE BEURD
115 GOSUB 1500 ;REM SPELER DOET ZET
117 GOSUB 1100 ;REM SPELMOMENT IN TEKENING
120 R=R+1 ;REM VERHOOG ZETTEN-TELLER
130 GOSUB 4000
132 IF Z<>0 OR R=9 THEN GOTO 300 ;REM SPEL AFGELOPEN ?
140 GOSUB 2000 ;REM COMP. BEKIJKT ALLE SPEELMOG.
150 GOSUB 2800 ;REM COMPUTER DOET ZET
172 P=FF(0):GOSUB 5000 ;REM OMZETT. NAAR AA
174 GOSUB 1000:GOSUB 1100 ;REM SPELMOMENT IN TEKENING
180 R=R+1 ;REM VERHOOG ZETTEN-TELLER
200 GOSUB 4000
205 IF Z<>0 OR R=9 THEN GOTO 300 ;REM SPEL AFGELOPEN ?
299 GOTO 110 ;REM VOLGENDE BEURD
300 REM EVALUATIE
305 PRINT "30" ;REM AANPASSING PREF.WAARDES;
310 IF Z=1 THEN PRINT "'0'"
312 IF Z=-1 THEN PRINT "'X'"
314 IF R=9 AND Z=0 THEN PRINT "NIEMAND"
320 PRINT "HEEFT GEWONNEN."
340 FOR I=2 TO R STEP 2 ;REM AANPASSING PREF.WAARDES;
    GG(XX(I))=GG(XX(I))+2;
NEXT I
390 GOTO 100 ;REM START NIEUW SPEL

```


wat spelletjes gespeeld zijn. Pas nadat 2 computers 100 hr. tegen elkaar gespeeld hadden was er sprake van intelligent spel. Er waren toen 850 spelletjes gespeeld.

Het is dan ook van belang dat de array's FF en GG met de spelmomenten en pref. waarden niet verloren gaan.

Door de honorering in de preferentie-waarde) bij de

spel-evaluatie aan te passen kan mogelijk een versnelling bereikt worden.

In regel 340 wordt gebruik gemaakt van een indirecte verwijzing naar de elementen in de Preferentie Waarde lijst (array GG).

In array XX staan nl. de laatste van de spelmomenten die de computer gespeeld heeft.

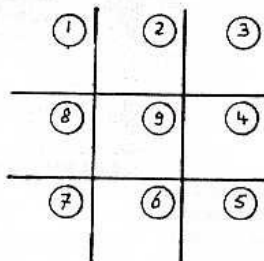
```
1000 REM SUBR. MAAK TEKENING
1010 CLEAR 0
1020 MOVE 24,16;DRAW 24,40
1030 MOVE 32,16;DRAW 32,40
1040 MOVE 16,24;DRAW 40,24
1050 MOVE 16,24;DRAW 40,32
1099 RETURN
```

```
1100 REM SUBR. TEKEN SPELMOMENT IN TEKENING
in: AA
variabelen: AA, I
1110 FOR I=1 TO 9
1120 IF AA(I)=0;NEXT
1121 IF I=1 MOVE 18,34
1122 IF I=2 MOVE 26,34
1123 IF I=3 MOVE 34,34
1124 IF I=4 MOVE 34,26
1125 IF I=5 MOVE 34,18
1126 IF I=6 MOVE 26,18
1127 IF I=7 MOVE 18,18
1128 IF I=8 MOVE 18,26
1129 IF I=9 MOVE 26,26
1130 IF AA(I)=1;GOSUB 1300
1140 IF AA(I)=2;GOSUB 1400
1150 NEXT I
1190 RETURN
```

```
1300 REM SUBR. TEKENEN X
1310 DRAW(relatief) 4,4;MOVE (rel.) -4,0;MOVE (rel.) 4,-4
1399 RETURN
```

```
1400 REM SUBR. TEKENEN "O"
1410 DRAW(rel.) 4,0;DRAW(rel.) 0,4;DRAW(rel.) -4,0;DRAW(rel.) 0,-4
1499 RETURN
```

De tekening zal iedereen bekend zijn, maar om te weten waar een X of een O moet komen te staan hebben we de vakken als volgt genummerd:



```
1500 REM SUBR. SPELER DOET ZET
in: AA uit: AA
variabelen: I, S, Z
1510 S=INKEY$
1520 I=VAL$Z
1530 IF I<1 OR I>9 OR AA(I)<>0 THEN GOTO 1510
1550 AA(I)=I
1599 RETURN
```

```
2000 REM SUBR. "BEKIJK ALLE SPEELMOGELIJKHEDEN".
in: AA, FF, GG out: AA, FF, GG, HH, P
variabelen: I, J, S, T, V, BB, CC, DD
2010 P=0
2020 FOR V=1 TO 9
2030 IF AA(V)<>0 THEN NEXT V;RETURN ;REM ZOEK VRIJE PLAATS
2040 FOR I=1 TO 9;BB(I)=AA(I);NEXT I ;REM COMP. PROBEERT ZET
2050 BB(V)=2;P=P+1
2060 GOSUB 2200 ;REM ZOEK LAAGSTE SPELMOMENT
2100 REM ZOEKEN PLAATS PREF. WAARDE
2110 Q=0
2120 DO Q=Q+1;UNTIL FF(Q)=L OR FF(Q)=99999
2130 IF FF(Q)=L THEN HH(P)=Q;NEXT V;RETURN
2140 REM TOEGEVOEGEN NIEUW SPELMOMENT
2142 FF(Q)=L;FF(Q+1)=99999;GG(Q)=0;HH(P)=Q
2150 NEXT V
2199 RETURN
```

DE SUBROUTINE: BEKIJK ALLE SPEELMOGELIJKHEDEN.

Deze routine zoekt naar alle vrije plaatsen in het speelveld, doet daar voorlopig zijn zet en onthoudt, in array HH met P elementen, wat de preferentie waarde daarvan is.

De waarde 99999 is de stop waarde in de array met spelmomenten. Daarna hoeft niet verder gezocht te worden, en het betreft dan een nog niet eerder gespeelde zet. De computer voegt dan dit nieuwe spelmoment toe aan de lijst met spelmoment-waarden (array FF).

Tevens plaatst hij in de preferentie-waarde lijst (array GG), op dezelfde plaats, de waarde 0.

De array's FF en GG worden dus langzamerhand al spelenderwijze opgebouwd. De waarden in array FF (dit zijn de spelmoment-waarden in decimale getallen) liggen niet op waarde-volgorde, maar op chronologische volgorde.

```
2200 REM SUBR. ZOEK IDENTIEK SPELMOMENT MET LAAGSTE WAARDE
in: BB, V out: L, V
variabelen: I, J, BB, CC, DD, S, T, Z
2210 GOSUB 2400;CC(I)=S ;REM MAAK SPELMOM. IN BB DEC.
2220 REM 3x DRAAIEN
2222 FOR I=2 TO 4;
2224 GOSUB 2300;GOSUB 2400;CC(I)=S;
2226 NEXT I
2230 REM NU SPIEGELEN
2232 DD1=BB3;DD8=BB4;DD7=BB5;DD3=BB1;DD4=BB8
2234 DD5=BB7;DD2=BB2;DD9=BB9;DD6=BB6
2236 FOR I=1 TO 9;BB(I)=DD(I);NEXT I
2238 GOSUB 2400;CC(I)=S
2240 REM NU WEER 3x DRAAIEN
2242 FOR I=6 TO 8;
2244 GOSUB 2300;GOSUB 2400;CC(I)=S;
2246 NEXT I
2250 REM VINDT LAAGSTE SPELMOMENT
2252 L=CC(1);
2254 FOR I=2 TO 8;
2256 IF CC(I)<L THEN L=CC(I);
2258 NEXT I
2299 RETURN
2300 REM SUBR. 90 GRADEN DRAAIEN
1310 FOR J=1 TO 8;DD(J+2)=BB(J);NEXT J
2320 DD1=BB7;DD2=BB8;DD9=BB9
2330 FOR J=1 TO 9;BB(J)=DD(J);NEXT J
2399 RETURN
2400 REM SUBR. MAAK DECIMAAL
2410 S=0
2420 FOR J=1 TO 9
2422 S=S+BB(J)*3^(J-1)
2424 NEXT J
2499 RETURN
```

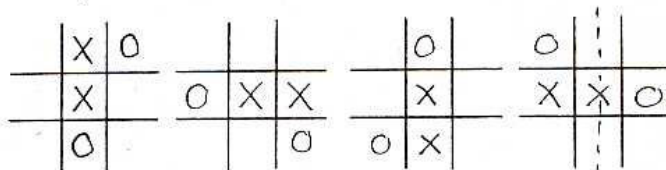
SOBROUTINE: ZOEK LAAGSTE SPELMOMENT (regel 2200-2299)

In het bekijken van een mogelijk speelbare zet, moet de computer alle strategisch identieke spelmomenten, op een na, uitsluiten. Daartoe

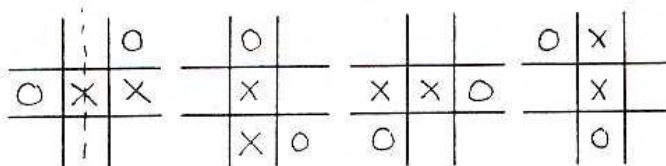
worden in array CC alle strategisch identieke spelmomenten, die verkregen wordt na draaiing en spiegeling, (in decimale waarden) verzameld. Met het spelmoment dat de laagste waarde heeft wordt

nu verder gewerkt.
De volgende tekeningetjes
verduidelijken dit:

1. OORSPRONKELIJKE SPEL-
MOMENT + 3 x 90 GRADEN
DRAAIEN:



2. SPIEGELEN + 3 x 90 GRA-
DEN DRAAIEN:



Laagste waarde nl. 100120020 tritair

```
2800 REM SUBR. BEPALING BESTE SPEELMOGELIJKHEID
      in: R,P,GG,HH      out: Q,R,XX,SG
      variabelen: I,H
2810 H=GG(HH(1));Q=HH(1)
2820 FOR I=2 TO P:
      IF H<GG(HH(I)) THEN H=GG(HH(I));Q=HH(I):
NEXT I
2830 XX(R)=Q
2899 RETURN
```

```
4000 REM SUBR. SPEL AFGELOPEN ?
      in: AA      out: Z,AA
4005 Z=0
4010 IF AA1=AA2 AND AA2=AA3 AND AA1<>0 THEN Z=-1:IF AA1=2 THEN Z=1:RETURN
4020 IF AA8=AA9 AND AA9=AA4 AND AA8<>0 THEN Z=-1:IF AA8=2 THEN Z=1:RETURN
4030 IF AA7=AA6 AND AA6=AA5 AND AA7<>0 THEN Z=-1:IF AA7=2 THEN Z=1:RETURN
4040 IF AA1=AA8 AND AA8=AA7 AND AA1<>0 THEN Z=-1:IF AA1=2 THEN Z=1:RETURN
4050 IF AA2=AA9 AND AA9=AA6 AND AA2<>0 THEN Z=-1:IF AA2=2 THEN Z=1:RETURN
4060 IF AA3=AA4 AND AA4=AA5 AND AA3<>0 THEN Z=-1:IF AA3=2 THEN Z=1:RETURN
4070 IF AA1=AA9 AND AA9=AA5 AND AA1<>0 THEN Z=-1:IF AA1=2 THEN Z=1:RETURN
4080 IF AA3=AA9 AND AA9=AA7 AND AA3<>0 THEN Z=-1:IF AA3=2 THEN Z=1:RETURN
4099 RETURN
```

```
5000 REM SUBR. OMZETTING DEC. NAAR TRITAIR
      in: P      out: AA,BB
      variabelen: I,Y
5010 FOR I=9 TO 0 STEP -1
5020 Y=INT(P/3^(T-1))
5030 AA(I)=Y:BB(I)=Y
5040 P=P-Y*3^(T-1)
5050 NEXT I
5099 RETURN
```

BELANGRIJKSTE VARIA- BELEN.

AA = werkelijk spelmoment.
BB = een tussentijds spelmoment tijdens zoeken.
CC = decimale spelmomenten van gedraaide en gespiegelde situaties.
DD = een tussentijds spelmoment tijdens spiegelen en draaien.
FF = de lijst met alle door de computr gespeelde spelmomenten (dec.).
GG = de lijst met preferentie-

waardes behorende bij de spelmomenten in FF.
HH = bevat de plaatsen waar de spelmomenten vande afgezochte speelmogelijkheden te vinden zijn.
XX = bevat de plaatsen van de spelmomenten die ontstaan na een zet van de computer.
R = zetten-teller.

Steven Verhoef