Kunstmatige Intelligentie 2008 — opdracht 2
RoboCom

Rick van der Zwet
<hvdzwet@liacs.nl>

LIACS
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

18 maart 2008

1 Inleiding

Dit verslag gaat over de tweede programmeer-opgave van het vak kunstmatige intelligentie[1]. Welke als opdracht heeft om softbots te bouwen, die tijdens een simulatie met elkaar samenwerken om een vlag te construeren of tegen andere softbots ten strijden trekken, met als doel als enigste over te blijven. Dit allen als doel om de *PEAS* beschrijving uit het college boek [2] beter te kunnen begrijpen.

2 Uitleg probleem

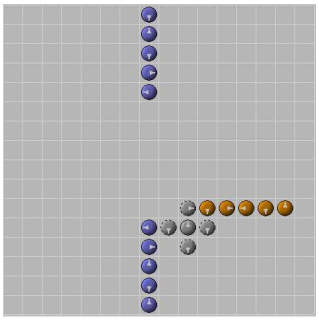
RoboCom is computerprogramma welke een simulatie omgeving bied waarmee met assembly achtige programma code een programma wordt gemaakt welke een software robot kan aansturen een zogenaamde *softbot*. Deze softbots kunnen simpele commando's zoals vermenigvuldigen, omgeving afzoeken, code kopiren en andere acties [1]. Al deze acties kosten *tikken*, waarbij er verschillende gradaties gemaakt worden aan de voorwaarden te scheppen aan de aan de omgeving waar de softbots in gaan 'leven'. Buiten de verschillende

[1] http://www.cyty.com/robocom/help/instr1.html

gradaties in tikken kan er ook nog onderscheidt gemaakt worden via limitatie in de instructieset. Tijdens deze opdracht is deze gelimiteerd tot *Classic* [2] welke de set is waarmee het spel origineel mee ontworpen is.

Het speelbord is vierkant in representatie, maar is werkelijkheid een bol. De onderste rij grenst aan de bovenste rij en hetzelfde geldt voor de meest linker en meest rechtse column. Het bord heeft in totaal 256 vakjes welke allen een softbot kunnen bevallen of leeg zijn. Een softbot op zijn beurt bestaat uit maximaal 50 *banken* welke op hun beurt weer 1000 regels code kunnen bevatten.

Een softbot kan actief zijn of niet. In de actieve stand is hij bezig met het uitvoeren van code. Waarbij begonnen wordt bij regel 1 in bank 1 en zo verder. Als hij bij het einde van een bank komt dan springt hij terug naar het begin (bank 1,regel 1). Als hij geen code kan vinden in bank 1 dan gaat hij dood, wat ook wel *data-hunger* genoemd wordt. Tijdens de niet actieve stand zal hij de lopende instructie stoppen en wachten tot hij weer geactiveerd wordt. Dit kan hij zelf niet doen.



Figuur 1: Impressie van de grafische representatie van het speelbord, elk team krijgt een eigen kleur. Een kruis door de softbot als deze niet actief is (niet weergegeven). Een driehoekje geeft de richting aan in welke hij 'kijkt'. De kleur van het driehoekje geeft aan van wie de instructie bank welke hij op het moment aan het uitvoeren is.

Hieronder volgt een paar regels voorbeeld programma assembly code:

```
Bank Main          ; Daddy Tutorminator's program
  @Loop
  Turn 0             ; turn around, that's cool
  Create 1, 2, 1     ; build a mobile child with 2 banks
  Trans 2, 1         ; transfer RunAround into the first bank
```

[2] http://www.cyty.com/robocom/help/instrall.html

```
                    ; the second bank is left empty
  Set %active, 1    ; activate our baby
  Jump @Loop        ; ...and start over
```

3 Theorie

Over de softbots kan een PEAS beschrijving gegeven worden. **P**restatie maat is het winnen van een simulatie. Als extra punt zo daar eventueel aan toegevoegd kunnen worden dat dit in een zo kort mogelijke tijd gedaan moet worden. **E**nvironment is het speelbord en de andere softbots in het spel. **A**ctuatoren van de softbot zijn de verschillende vragen die de softbot aan de omgeving kan stellen. Dit zijn alle verschillende types van scannen. **S**ensoren zijn de acties die de softbot op de omgeving kan doen. Hieronder vallen acties als kopiren, creren, activeren.

De reeds beschikbare documentatie [4] beschrijft uitvoering enkele theorien die gebruikt kunnen worden de softbots te schrijven. In het kort komt het neer op de verschillende punten waar rekening gehouden mee moet worden. Probeer het programma zo simpel mogelijk te houden, een heleboel regels en/of banken kosten veel tijd om te dupliceren en/of uit te voeren waardoor je softbot kwetsbaarder is tegen simpele programma's. Gebruik simpele programma's in het begin, deze worden ook wel virussen genoemd, om de vijand zo veel mogelijk te 'pesten' in de ontwikkeling van zijn softbots. Maak gebruik van meerdere type softbots en beperk je niet tot een tactiek. Ga bijvoorbeeld eerst met een simpel virus aan de slag en gebruik daarna ingewikkelde en sterkere softbots om het karwei af te maken.

Als laatste in dit korte rijtje, zorg dat de softbots geen hinder ondervinden van hun eigen tactieken en let op dat de vijand dezelfde tactiek zal gebruiken, dus wapen je ertegen.

4 Aanpak

Bij het maken van de vlag is gebruik gemaakt van het coperatie element in het spel. Tijdens het maken van de twee vecht softbots is zowel het competitie element gebruikt als het coperatie element.

4.1 Vlag softbot

Bij een vlag cruciaal is dat de softbots netjes naast elkaar komen te lopen is dat de voornaamste uitdaging. Het uit programmeren hierna is triviaal. Om te zorgen dat de verschillende softbots elkaar kunnen ontmoeten wordt er

gebruik maakt van het feit dat als je twee haak op elkaar staande lijnen trekt deze altijd ergens in een enkele lijn snijdt. Het algoritme in pseudo-code

```
while begin
    scan vakje voor de softbot
    switch begin
        case leeg vakje
            maak kloon aan
            blijf naar kloon kijken tot deze signaal doodgaan gegeven
            heeft ga dood
        case bot andere kleur
            geef signaal doodgaan aan 'staart'
            ga rechtsom
            continue
        case bot eigen kleur
            als links of recht van je een 'vijand' bot staat dan
                klaar doe niets meer
            anders
            ga rechtsom
            continue
    end
end
```

4.2 Defensieve softbot

Bij het schrijven van de eerste vechtende softbot is gekozen voor een defensieve tactiek, met een soort van drie meertrapsraket methode. Als eerste komt er een zeer simpel schild om moederbot heen welke een verlammend verspreidend virus infecteert aan alle softbots die in de buurt komen. Na een zekere tijd wordt er een nog simpeler virus genjecteerd dat alle softbots vermoord. Hierna worden ingewikkelder softbots gemaakt genaamd slopers welke het bord opruimen. Voordeel hierbij is dat simpele virussen geen vat op deze softbots hebben, want de moederbot is geschreven om hier immuun voor te zijn. Nadeel is echter dat er geen tot weinig ontwikkeling plaatsvind in de softbots, waardoor in de laatste fase van het spel misschien te weinig softbots dood gaan om te zorgen dat rustig de slopers gemaakt kunnen worden.

4.3 Offensieve softbot

De aanvallende softbot heeft een heel andere tactiek. Deze maakt gelijk ingewikkende softbots. Deze ingewikkende softbots infecteren de vijand met een

simpel virus en laten dit even verspreiden. Nadien vernietigd hij deze softbot en maakt hij een kloon van zichzelf aan. Deze tactiek zorgt ervoor dat de softbots niet vatbaar zijn voor de simpele virussen, maar heeft als nadeel dat de softbots veel meer tijd kosten om te bouwen en dan kwetsbaarder kunnen zijn.

5 Implementatie

Er is gekozen om geen gebruik te maken voorbeeld implementaties van andere softbots in de eigen code. De voorbeelden in de handleidingen zijn wel overgenomen. De softbot code is in de RoboCom taal geschreven.

6 Experimenten

Alle testen zijn uitgevoerd met het programma RoboCom Workshop v3.10[5] gedraaid op een Microsoft Windows XP computer. Gebruikt instructie set is Classic.

6.1 Vlag

Tijdens het testen van het succesvol vormen van de vlag is gebruik gemaakt van 2 type simulaties. Als eerste is 20 keer de softbots op willekeurige plekken neergezet en gekeken of er een vlag uitkomt. Als tweede zijn de softbots 20 keer op elke keer dezelfde plek gezet en gekeken of deze dezelfde vlag vormen.De resultaten zijn te vinden in tabel 1

Tabel 1: Resultaten plaatsingen van vlag, bij goed staat de vlag of horizontaal, bij matig verticaal, bij fout is erna 80000 tikken nog geen uitslag. De vaste plaatsing is op posities (X,Y,richting) $8, 10, 2,\ 14, 14, 0,\ 3, 9, 1$

Type	Goed	Matig	Fout
Willekeurige plaatsing	9	9	2
Vaste plaatsing	4	12	4

6.2 Vechters

Door het programma RoboCom wordt een set softbots meegeleverd die gelabeld zijn als ‘classic legends’. De twee softbots hebben in een tegen een

gevechten tegen al deze classic legends gespeeld. Elke een tegen een gevecht is 20 keer uitgevoerd. De resultaten staan hieronder in de tabel 2 voor de defensieve softbot. Tabel 3 levert de resultaten voor de offensieve softbot.

Tabel 2: Resulaten defensieve softbot na 80000 tikken, bij score telt een gelijkspel voor 1 punt en winst voor 3 punten en verlies niets.

Tegenstander	Winst	Verlies	Gelijkspel	Score
(Cy)Borg 004c83a8n	3	0	7	16
Advanced Speedix 2004	2	1	7	13
Africa	2	0	8	14
Alien v5.13d	1	0	9	12
Bright Star Three	0	5	5	5
Comes the Wuss 2	10	0	0	30
CopyBot	1	1	8	11
Delusion 5	10	0	0	30
DJ Combat - Deep Strike	10	0	0	30
DJ CoNTiNUuM	0	2	8	8
Fungus 0.2a	0	0	10	10
HotBot V2	10	0	0	30
Hurricane6	10	0	0	30
Immer mit der Ruhe! XXXVI	9	0	1	28
Jrg's Bisexa	0	0	10	10
Kreuziger V2.03q anti Killer	10	0	0	30
Malignant Tumor	0	1	9	9
Outer Limits	10	0	0	30
Speed Slug	7	0	3	24
SpiceGirls	10	0	0	30
Spy vs. Spy	0	7	3	3
fighter-defensive-1	17	105	88	139

7 Conclusie

7.1 Vlag

Bij het willikeurig neerzetten van de softbots gaan er slechts 2 gevallen fout, nader onderzoek blijkt uit dat er een fout in de code is geslopen waardoor

Tabel 3: Resulaten offensieve softbot na 80000 tikken, bij score telt een gelijkspel voor 1 punt en winst voor 3 punten en verlies niets.

Tegenstander	Winst	Verlies	Gelijkspel	Score
(Cy)Borg 004c83a8n	10	0	0	30
Advanced Speedix 2004	10	0	0	30
Africa	4	0	6	18
Alien v5.13d	8	1	1	25
Bright Star Three	4	3	3	15
Comes the Wuss 2	4	5	1	13
CopyBot	9	1	0	27
Delusion 5	7	1	2	23
DJ Combat - Deep Strike	10	0	0	30
DJ CoNTiNUuM	3	4	3	12
Fungus 0.2a	0	1	9	9
HotBot V2	0	0	10	10
Hurricane6	10	0	0	30
Immer mit der Ruhe! XXXVI	10	0	0	30
Jorg's Bisexa x	0	10	0	0
Kreuziger V2.03q anti Killer	10	0	0	30
Malignant Tumor	3	7	0	9
Outer Limits	10	0	0	30
Speed Slug	9	0	1	28
SpiceGirls	10	0	0	30
Spy vs. Spy	1	0	9	12
fighter-offensive-2	33	132	45	144

situaties waarbij een volledig 'kruis' gemaakt word door een enkele softbot niet afgevangen word. Waardoor deze in een deadloop beland. Een vaste uitgangspositie levert echter heel vreemde resultaten op. De verwachting was dat er altijd dezelfde situatie uit zal komen, dit is echter totaal niet het geval, 4 gevallen worden zelfs niet opgelost. De oorzaak hiervan ligt niet in de softbots, maar in de implementatie van het programma. Het aantal softbots lijkt niet te kloppen en nader analyse blijkt dat er 'spook softbots' op het bord staan. Verder zijn de grensgevallen niet bekend. Zoals wat er bijvoorbeeld gebeurt op het moment dat 2 softbots op precies hetzelfde moment een instructie op de softbot plaatsen. Er lijkt willekeurig een keuze gedaan te worden, waardoor het spelverloop ernstig verstoord kan worden.

Voor verder onderzoek is het aan te raden contact op te nemen met de ontwikkelaars om de fouten uit de simulatie te halen en dan de kijken hoe het geval van het 'kruis' opgelost kan worden.

7.2 Vechters

Beide vechtende softbots hebben ongeveer dezelfde score, maar halen wel op verschillende gebieden hun punten binnen. De defensieve weet er in 41% van de gevallen gelijkspel uit te slepen en is in 9% zelfs winnend, waarbij dus voor 50% van de gevallen de softbot goed presteert. Hij presteer slecht tegen softbots die gebruik maken van verschillende tactieken en meer effectieve opruim methodes hebben.

De offensieve is effectief in het opruimen tegen verschillende type softbots waarbij de defensieve nog met een gelijkspel genoegen moet nemen, maar houdt het een stuk minder uit tegen de 'grote jongens'.

Voor verder onderzoek zou gekeken moeten worden of een combinatie van de vechters een goede opstelling is. Een analyse van alle verschillende tactieken van de tegenstanders zou ook aan te raden zijn. Door een softbot zo veel mogelijk te storen in zijn ontwikkeling is de beste methode omdat niet veel softbot bouwers rekening hebben gehouden met bijvoorbeeld een situatie waar alle softbots niet meer in controle zijn van de bouwer.

Referenties

[1] W.A. Kosters, Kunstmatige intelligentie Programmeer-opgave 2 van 2008 – Bridge, `http://www.liacs.nl/~kosters/AI/robot.html`

[2] S.J. Russell en P. Norvig, Artificial Intelligence, A Modern Approach, Second ediion, Prentice Hall, 2003.

[3] Robocom, official website, latest version `http://cyty.com/robocom/`

[4] Robocom help, official website, latest version `http://www.cyty.com/robocom/?area=help`

[5] Robocom Workshop, official website, latest version `http://www.cyty.com/robocom/?area=d\_rcws`

Appendix

De obot programma's zagen er als volgt uit:

```
; RoboCom program

; Instead of having to change all the values below with every new program,
; it's suggested that you adjust the template header to your needs once
; in the Prefs->Editor dialog. You can also remove these comments.
; See documentation for the meaning of these and more header fields.

Published  Name        figure-flag-v2          ; Name of this program
Published  Author      Rick van der Zwet       ; Name of author
Published  EMail       hvdzwet@liacs.nl        ; Author's e-mail address
Published  Country     The Netherlands         ; Author's home country
Published  Comment     Part AI Course @ LIACS  ; A comment on this prog
Secret     Password    YOUR_PASSWORD_HERE      ; Password for competitons

Published  OpenSource  yes                     ; This prog is open source
Published  Language    RC300                   ; Written in RC300 language
Published  OptionSet   Classic                 ; Recommended OptionSet

Bank Flag
  @Start
  ;Indentify field
  Scan #1
  Comp #1,0
  Jump @FieldBusy

  ;New robot
  Create 2,1,0
  Scan #1
  Comp #1,2
  Jump @EnemyBot
  Trans 1,1
  Set %active,#active
  Jump @ScanForDieMode

  ;Indentify enemy or friendly
  @FieldBusy
  Comp #1,2
  Jump @EnemyBot

  ;We completed the tail, let's check whether
  ;we are next to some other friend

  @FriendlyBot
  Turn 0
  Scan #1
  Turn 0
  Turn 0
  Scan #2
  Turn 0

  Comp #1,0
  Jump @ScanForDieMode
  Comp #2,0
  Jump @ScanForDieMode
  Comp #active,1
  Jump @EnemyBot
  Set #active,2
  Turn 0
  Jump @Start

  ;Start looking whether we need to kill ourself
  @ScanForDieMode
  Comp %active,3
  Jump @ScanForDieMode
  Set #active,3
  ;Delay to allow 'message' to pass trough
  Turn 0
  Turn 0
  Turn 0
  Die

  ;If enemy bot, start looking which direction we are in
  @EnemyBot
  Set #active,3 ; enable destruction of tail
  ;Delay to allow 'message' to pass trough
  Turn 0
  Turn 0
  Turn 0
  Set #active,1
```

```
001: ; RoboCom program
002:
003: Published  Name        fighter-defensive-1      ; Name of this program
004: Published  Author      Rick van der Zwet        ; Name of author
005: Published  EMail       hvdzwet@liacs.nl         ; Author's e-mail address
006: Published  Country     The Netherlands          ; Author's home country
007: Published  Comment     Part AI Course @ LIACS   ; A comment on this prog
008: Secret     Password    YOUR_PASSWORD_HERE       ; Password for competitons
009:
010: Published  OpenSource  yes                      ; This prog is open source
011: Published  Language    RC300                    ; Written in RC300 language
012: Published  OptionSet   Classic                  ; Recommended OptionSet
013:
014: ; General overview:
015: ; * Create really simple shield around mother bot with very simple virus,
016: ;   called protectors
017: ; * After some cycles call autodestroy on all affected bots, called
018: ;   kamikazes
019: ; * Start creating 'destroyers' with basicly kill all robots
020: ;
021:
022: ; Jump bank to safer grounds
023: Bank SafeJump
024:         Bjump 25,1
025:
026: ; Bank 2-20, empty buffer banks
027: Bank Bank2
028: Bank Bank3
029: Bank Bank4
030: Bank Bank5
031: Bank Bank6
032: Bank Bank7
033: Bank Bank8
034: Bank Bank9
035: Bank Bank10
036: Bank Bank11
037: Bank Bank12
038: Bank Bank13
039: Bank Bank14
040: Bank Bank15
041: Bank Bank16
042: Bank Bank17
043: Bank Bank18
044: Bank Bank19
045: Bank Bank20
046: Bank Bank21
047: Bank Bank22
048: Bank Bank23
049: Bank Bank24
050:
051: ;Bank 25
052: Bank MainParent
053:         ;Protectors build
054:         @MainParentProtectorStart
055:         Scan #10
056:         Comp #10,0
057:         Jump @MainParentExists
058:         Create 1,1,0
059:         @MainParentExists
060:         Trans 26,1
061:         Set %active,1
062:         Turn 0
063:         Add #1,1
064:         Comp #1,50
065:         Jump @MainParentProtectorStart
066:         Set #1,1
067:         Add #2,1
068:         Comp #2,4
069:         Jump @MainParentProtectorStart
070:
071:         ;Kamikazes build
072:         Set #1,1
073:         @MainParentKamikazeStart
074:         Trans 27,1
075:         Turn 0
076:
077:         Add #1,1
078:         Comp #1,30
079:         Jump @MainParentKamikazeStart
080:
081:         ;Destroyer build
082:         @MainParentDestroyerStart
083:         Turn 0
084:         Scan #1
085:         Comp #1,0
086:         Jump @MainParentDestroyerStart
087:         Create 2,6,0
088:         Trans 28,5
089:         Trans 29,4
090:         Trans 28,1
091:         Set %active,1
092:         Jump @MainParentDestroyerStart
093:
094: ;Bank 26
095: Bank Protectors
096:         Trans 1,1
097:         Trans 1,2
098:         Trans 1,4
099:         Turn 1
100:
101: ;Bank27
102: Bank Kamikazes
103:         Trans 1,1
104:         Add #1,1
105:         Turn 0
106:
107: ;Bank28
108: ;SafeJump Destroyers
109: Bank DestroyerJump
110: Bjump 4,1
111:
112: ;Bank29
113: Bank DestroyerMain
114:         @DestroyerStart
115:         Turn 0
116:         Scan #1
117:         Comp #1,1
118:         Jump @DestroyerFriendly
119:         Set #2,%banks
120:         @DestroyerBankwhipe
121:         Trans 7,#2
122:         Sub #2,1
123:         Comp #2,0
124:         Jump @DestroyerBankWhipe
125:         Set %active, 1
126:         Jump @DestroyerStart
127:         @DestroyerFriendly
128:         Comp #1,0
129:         Jump @DestroyerStart
130:         Create 2,6,0
131:         Trans 4,4
132:         Trans 5,5
133:         Trans 5,1
134:         Set %active,1
135:         Jump @DestroyerStart
136:
137:
138: ;Bank 30-50, empty buffer banks
139: Bank Bank30
140: Bank Bank31
141: Bank Bank32
142: Bank Bank33
143: Bank Bank34
144: Bank Bank35
145: Bank Bank36
146: Bank Bank37
147: Bank Bank38
148: Bank Bank39
149: Bank Bank40
150: Bank Bank41
151: Bank Bank42
152: Bank Bank43
153: Bank Bank44
154: Bank Bank45
155: Bank Bank46
156: Bank Bank47
157: Bank Bank48
158: Bank Bank49
159: Bank Bank50
160:
161:
```

```
001: ; RoboCom program
002:
003: Published  Name        fighter-offensive-2      ; Name of this program
004: Published  Author      Rick van der Zwet        ; Name of author
005: Published  EMail       hvdzwet@liacs.nl         ; Author's e-mail address
006: Published  Country     The Netherlands          ; Author's home country
007: Published  Comment     Part AI Course @ LIACS   ; A comment on this prog
008: Secret     Password    YOUR_PASSWORD_HERE       ; Password for competitons
009:
010: Published  OpenSource  yes                      ; This prog is open source
011: Published  Language    RC300                    ; Written in RC300 language
012: Published  OptionSet   Classic                  ; Recommended OptionSet
013:
014: ; Jump bank to safer grounds
015: ;Bank Bank1
016: Bank SafeJump
017:     Bjump 11,1
018:
019: ; Bank 2-9, empty buffer banks
020: Bank Bank2
021: Bank Bank3
022: Bank Bank4
023: Bank Bank5
024: Bank Bank6
025: Bank Bank7
026: Bank Bank8
027: Bank Bank9
028:
029: ;Bank Bank10
030: Bank BigJump
031:     BJump 11,1
032:
033: ;Bank Bank11
034: Bank BigMain
035:     @BigStart
036:     Turn 0
037:     Trans 12,1
038:     Scan #1
039:     Comp #1,1
040:     Jump @Friendly
041:
042:     ;Enemy
043:     Set #2,0
044:     @Delete
045:     Add #2,1
046:     Trans 13,#2
047:     Comp #2,%Banks
048:     Jump @Delete
049:     Set %active,1
050:
051:     @Friendly
052:     Comp #1,2
053:     Create 2,20,0
054:     Comp %active,25
055:     Trans 11,11
056:     Trans 10,1
057:     Trans 13,2
058:     Trans 13,3
059:     Trans 13,4
060:     Set %active,25
061:     Trans 10,10
062:     Trans 12,12
063:     Jump @BigStart
064:
065: ;Bank Bank12
066: Bank FirstVirus
067:     Turn 0
068:     Trans 1,1
069: ;Trans 1,5 Wins against cyborg
070:     Trans 1,3
071:
072: ;Bank 13-50, empty buffer banks
073: Bank Bank13
074: Bank Bank14
075: Bank Bank15
076: Bank Bank16
077: Bank Bank17
078: Bank Bank18
079: Bank Bank19
080: Bank Bank20
081: Bank Bank21
082: Bank Bank22
083: Bank Bank23
084: Bank Bank24
085: Bank Bank25
086: Bank Bank26
087: Bank Bank27
088: Bank Bank28
089: Bank Bank29
090: Bank Bank30
091: Bank Bank31
092: Bank Bank32
093: Bank Bank33
094: Bank Bank34
095: Bank Bank35
096: Bank Bank36
097: Bank Bank37
098: Bank Bank38
099: Bank Bank39
100: Bank Bank40
101: Bank Bank41
102: Bank Bank42
103: Bank Bank43
104: Bank Bank44
105: Bank Bank45
106: Bank Bank46
107: Bank Bank47
108: Bank Bank48
109: Bank Bank49
110: Bank Bank50
111:
```