

ROBO Opdrachten Set 1

Arvid Halma

18 februari 2007

In deze verzameling van opdrachten vind je puzzels van verschillende moeilijkheidsgraden door elkaar. Het is handig als je een kijkje neemt in het introductieboek dat je op de website kunt vinden voordat je begint.

1 Verplaats baken

1.1 Met de afstandsbediening

Gebruik de afstandsbediening (Uitvoeren ▷ Afstandsbediening) om het baken ten zuid-oosten van de robot naar de zuid-west hoek te verplaatsen.



1.2 Schrijf het script zelf

Probeer hetzelfde voor elkaar te krijgen door zelf de commando's in te typen. In tegenstelling tot de code die de afstandsbediening levert,

```
1 ...  
2 vooruit(1)  
3 vooruit(1)  
4 vooruit(1)  
5 ...
```

kan je bijvoorbeeld direct

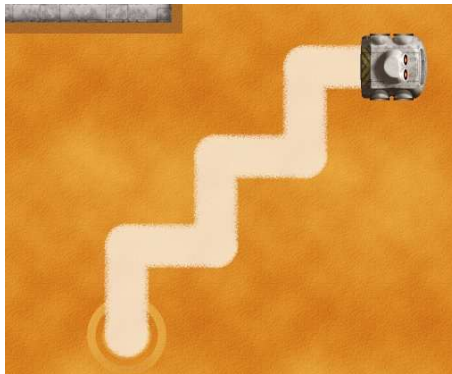
```
1 ...  
2 vooruit(3)  
3 ...
```

gebruiken.

2 Trappen tekenen

2.1 Met de afstandsbediening

Gebruik de afstandsbediening om een trappetje te tekenen zoals hieronder staat afgebeeld.



Figuur 1: Een trappetje met 3 treden.

2.2 Met een herhaal lus

Gebruik een herhaal-lus om de drie treden te tekenen. Hint:

```
1 # Begin met verven  
2 herhaal(3)  
3 {  
4     # Teken een trede  
5 }  
6 # Stop met verven
```

2.3 Met een procedure

Maar een procedure `tekenTrap()` die de trap met drie treden tekent zoals in de vorige opdracht. Roep deze procedure aan zodat de robot ook daadwerkelijk gaat tekenen.

2.4 Generaliseer deze procedure

Maak deze procedure breder toepasbaar zodat het trappen kan tekenen met een willekeurig aantal treden. Deze procedure neemt een argument, dat je bijvoorbeeld **aantal** kunt noemen, waarmee je in een procedure-aanroep aan kunt geven hoeveel treden er getekend moeten worden.

Hint:

```
1  tekenTrap(2)
2
3  procedure tekenTrap(aantal)
4  {
5      # teken een trap met 'aantal' treden
6  }
```

tekent een trap met twee treden. Als je regel 1 vervangt door `tekenTrap(5)` tekent de robot vijf treden.

3 Spoorzoeken

In de kaart *goRightAtWhite1.map* vind je een aantal witte stippen. Deze kan je volgen door vooruit te rijden tot je er een tegenkomt, en dan op de stip rechtsaf te slaan totdat je de volgende tegenkomt. Op een gegeven moment kom je een bakken tegen. Het is de bedoeling dat je dit bakken vind.

Schrijf een script dat werkt in de volgende kaarten:

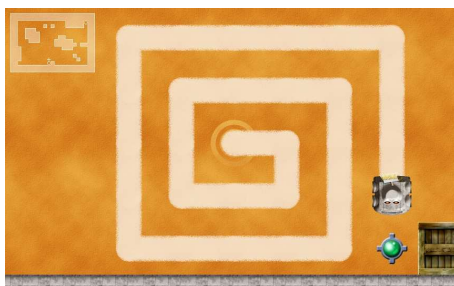
goRightAtWhite1.map, *goRightAtWhite2.map*, *goRightAtWhite3.map*.



4 Spiraal

4.1 Eerste versie

Schrijf een script dat een spiraal tekent totdat de robot niet verder kan omdat er een obstakel in de weg staat.



4.2 Tweede versie

Maak een script dat de robot exact hetzelfde gedrag laat vertonen, maar gebruik recursie om de robot langs de zijden van de spiraal te laten rijden. Hint: maak eerst een procedure die de robot langs één zijde laat rijden.

5 Herschrijf herhaal-lus

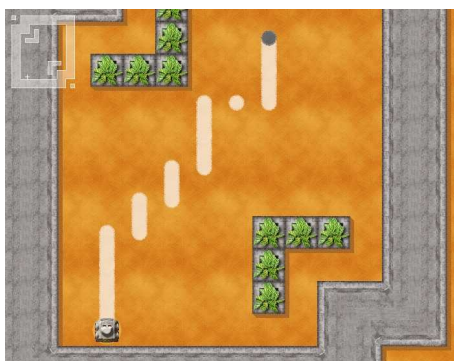
De volgende code laat de robot net zolang vooruit rijden totdat het een onstakel tegenkomt.

```
1  herhaalZolang(voorIsVrij())
2  {
3      vooruit(1)
4  }
```

Herschrijf deze code, zodat het geen gebruik meer maakt van `herhaalZolang(...)`, maar van ondermeer `herhaal()` en `als(...)`.

6 Spring van lijn naar lijn

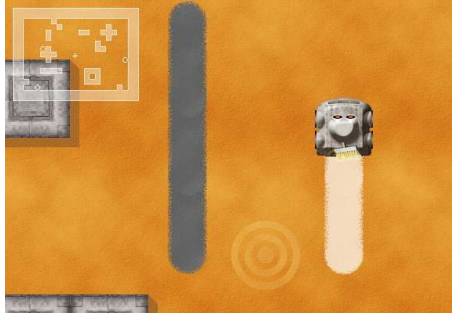
Maak een programma waarmee je de robot naar de grijze stip laat navigeren in de kaart *verticalLines.map*.



7 Kopieer lijn

7.1 Maak een even lange lijn

In the kaart *copyLine1.map* vind je links van de robot een zwarte lijn. Het doel is om een even lange parallelle lijn te tekenen rechts van het startpunt. Doe dit op een manier zodat je niet van te voren hoeft uit te meten hoe lang de zwarte lijn is. Zorg ervoor dat je stapje voor stapje aanpakt.



7.2 Maak twee keer zo lange lijn

Gebruik dezelfde kaart om een twee keer zo lange lijn te maken. Dit is een stuk lastiger, omdat je tijdens het kopieëren niet meer weet welk stuk er nu al gedaan was. Hier zul je iets op moeten bedenken. Hint: teken niet alleen de kopie, maar verf ook tussen de twee lijnen om bij te houden waar je bent geweest.