

Intelligente Systemen

Documentatie opgave 4

Tijn Kersjes (3855474)

Jordi Vermeulen (3835634)

Globale werking

Er zijn 2 fases, namelijk *exploration* en *path planning*:

- Er wordt een lijst bijgehouden van kamers die bezocht zijn (en waarvan dus vast is gesteld dat ze veilig zijn), een lijst van kamers waarvan de agent zeker weet dat er geen *pit* is, en een lijst van kamers waar geen *wumpus* is. Er wordt een kamer gekozen die grenst aan een kamer waar je al geweest bent, die veilig is (dus zowel in de geen-*pit*-lijst en de geen-*wumpus*-lijst zijn), maar nog niet eerder bezocht is. Vanuit deze kamer worden waarnemingen gedaan, bij gebrek aan *breeze* worden de aangrenzende kamers aan de geen-*pit*-lijst gezet, en bij gebrek aan *stench* in de geen-*wumpus*-lijst. Dit herhaalt zich totdat er geen kamers over zijn die bezocht kunnen worden. De lijst van bezochte kamers is dan de lijst van alle veilig bereikbare kamers.
- Het *path planning*-gedeelte werkt hetzelfde als in de vorige opdracht, met als kleine aanpassing dat er alleen naar kamers in de lijst van veilig bereikbare kamers wordt gekeken.

1. Wanneer je zoekt naar kamers die veilig zijn is het toegestaan om naar alle kamers toe te springen die grenzen aan kamers waar de agent al eerder is geweest. De agent kan namelijk gegarandeerd naar die kamer lopen via al eerder bezochte kamers, maar het bijhouden van een huidige positie en het uitrekenen van de route naar de volgende kamer die je wilt bezoeken is niet nodig en zou dus overbodig veel werk zijn.

2. De agent kan het *gold* niet vinden wanneer hij niet via enkel 100% zeker veilige kamers bij het *gold* kan komen. Wanneer je bijvoorbeeld op positie (1, 1) start en op (2, 2) bevindt zich een *pit*, wordt er op zowel (1, 2) als (2, 1) een *breeze* waargenomen. Het is op dat punt onmogelijk om te bepalen waar de *pit* zich precies bevindt, want het is zelfs mogelijk dat er 3 *pits* aanwezig zijn op respectievelijk (1, 3); (2, 2) en (3, 1).

3. Het programma reageert correct en is niet in staat om bij het *gold* te komen. Vanaf de posities (2, 3) en (3, 2) kun je namelijk niet verder lopen met de garantie om geen *pit* of *wumpus* tegen te komen. Positie (4, 2) - waar het *gold* ligt - is namelijk potentieel een *pit* vanwege de waargenomen *breeze* op (3, 2).

4. Nee. Hierbij geldt hetzelfde als beschreven bij punt 2 met de *pit*. Wanneer er wordt aangegeven dat er maximaal 1 *wumpus* in de spelwereld aanwezig is, kan het *gold* wel worden gevonden. De agent moet dan aan de hand van die informatie kunnen afleiden dat de *stenches* op (1, 3) en (3, 1) afkomstig zijn van dezelfde *wumpus*, die zich dan alleen op positie (2, 2) kan bevinden, en dus niet ook op positie (1, 3) en (3, 1). Vervolgens weet de agent dat de kamers op (1, 2) en (2, 1) veilig zijn.

5. Het '*world.pl*' bestand moet een grotere wereld beschrijven. Het programma blijft werken (mits het gegeven probleem berekenbaar is). Voorbeelduitvoer voor een wereld van 5x5 kamers, met *pits* op (3, 1); (3, 3) en (4, 3), een *wumpus* op (1, 5) en het *gold* op (5, 5):

List = [(1, 1), (2, 1), (1, 2), (2, 2), (3, 2), (2, 3), (1, 3), (1, 4), (2, 4), (3, 4), (2, 5), (3, 5), (4, 5), (5, 5), (5, 4), (4, 4), (5, 3), goNorth, goEast, goNorth, goWest, goNorth, goEast, goNorth, goEast, goEast, goEast, grab]

6. Het programma zou moeten blijven werken bij een andere hoeveelheid dimensies. Wel zal er een extra coördinaat (Z) moeten worden toegevoegd aan de meeste predicaten die nu alleen met (X, Y) posities werken, en de interface moet een *voor/4* en *achter/4* predikaat krijgen. Het is waarschijnlijk mogelijk om de coördinaten waarmee de kamers aangeduid worden volledig af te schermen binnen de *interface*, waardoor het programma niet meer aangepast hoeft te worden, maar alleen nog het '*world.pl*' bestand wanneer er een andere hoeveelheid dimensies gebruikt wordt.