

## Intelligente Systemen practicum 4

Thijs Klaver	3711633
Bas Meesters	3700569

### Verslag

Wij hebben zoals ook als tip werd gegeven in de opdracht het programma in twee delen gesplitst. Het eerste deel bestaat uit het zoeken naar het goud en het tweede gedeelte uit het terug vinden van de startpositie nadat het goud is gevonden. Het vinden van het goud wordt gedaan met behulp van het predicaat `explore(Explored, Queue, Breezes, Stenches, Path)`. Dit predicaat maakt gebruik van `depth first search` om zo het goud te vinden. In de `Queue` zal in het begin de startpositie `(1,1)` staan en de andere argumenten zijn lege lijsten. Vervolgens worden alle onontdekte aanliggende vakjes aan de `Queue` toegevoegd aan de voorkant. Er wordt per vakje gekeken of deze veilig is aan de hand van de lijsten van geëxploreerde vakjes en de vakjes met `Breezes` en `Stenches`. Wanneer een vakje veilig is kan deze aan `Path` worden toegevoegd. De informatie die gewonnen wordt door het bekijken van het vakje wordt meegenomen in `Explored`, `Stenches` en `Breezes`. Dit wordt herhaald tot het goud is gevonden of totdat er niet meer op een veilige manier geëxploreerd kan worden.

Voor de lijst van acties om terug te komen van de positie `(X,Y)` waar het goud ligt naar `(1,1)` gebruiken we een aangepaste versie van de `pathfinding` van het vorige practicum. Het verschil zit erin dat we hier de lijst van uiteindelijke acties niet omdraaien en de manier waarop we controleren of een vakje veilig is. Nu doen we dat namelijk door te kijken of we dat vakje al eerder verkend hebben, waardoor we dus ook zeker weten dat het een veilig vakje is.

Voordelen van springen zijn dat je vakjes kunt overslaan in het kiezen van een pad en daardoor een korter pad hebt. Het zal in veel van de verschillende `Wumpus` werelden vaak voorkomen dat je vanuit een vakje twee of meer kanten veilig op kan gaan. Wanneer je dan één van die aanliggende vakjes kiest maar daaruit niet verder kan is het dus sneller om te springen naar één van de andere vakjes aangelegen aan het vorige vakje die ook veilig waren.

Er zijn best een aantal situaties in ons programma waarin het goud niet gevonden kan worden. Dit is bijvoorbeeld wanneer het goud onbereikbaar is doordat deze wordt ingesloten door de pits en eventueel de `Wumpus`. Of wanneer het goud wel bereikbaar is maar dit niet is vast te stellen zonder risico te nemen. Zo ook de situatie in figuur 1. Hier is het namelijk niet mogelijk om zonder risico verder te gaan vanuit vakje `(3,2)`. Er wordt hier namelijk een bries waargenomen en het is niet te achterhalen dat er op positie `(4,2)` geen pit is. Ons programma zal in dit geval `false` terug geven.

4				<b>P</b>
3	<b>W</b>		<b>P</b>	
2				<b>G</b>
1			<b>P</b>	
	1	2	3	4

Legenda:  
**P** = pit  
**W** = wumpus  
**G** = gold

Figure 1: Testwereld voor de wumpus-agent

Een situatie waarin het wel goed gaat is die in figuur 2. Onze veiligheidscheck weet dat er maar op één vakje een Wumpus kan zijn. Doordat er een stench wordt waargenomen op (2,1) en op (1,2) weet de agent dat er geen Wumpus kan zijn op (3,1) of (1,3). Uiteindelijk zal hij dus bij het goud aankomen door er om heen te lopen of rechtdoor te gaan.

4				
3	<b>G</b>			
2		<b>W</b>		
1				
	1	2	3	4

Figure 2: Testwereld voor de wumpus-agent

Grotere werelden dan 4 x 4 zouden geen probleem moeten zijn omdat dit wordt beschreven in world.pl. Wanneer dit op een juiste manier gebeurt zal het programma niet veranderd hoeven te worden. Het zoeken naar het goud kan op een zelfde manier gebeuren alleen is de zoekruimte groter.

Ook drie dimensionale werelden zullen niet echt een groot probleem zijn. De regels zijn het zelfde alleen moet er overal nog een extra coördinaat meegenomen worden. De wereld en het zoeken worden iets complexer maar de manier van beredeneren kan hetzelfde blijven. Met vrij kleine aanpassingen aan het huidige programma kan dit dus gerealiseerd worden.