

Project Multi-Agent Systemen – Groep 22

Jan-Willem Gmelig Meyling	4305167
Huub Goltstein	4246683
Tom Harting	4288319
Lars den Herder	4216873
Christian Rosheuvel	4283244
Paul van Wijk	4285034

8 mei 2014

Inhoudsopgave

1	Inleiding	3
1.1	Aanleiding	3
1.2	Doelstelling	3
1.3	Structuurbeschrijving	3
2	Programma van Eisen	4
2.1	Minimale Eisen (Must Have)	4
2.2	Geavanceerde functionaliteit (Should Have)	4
2.3	Uitgebreide functionaliteit (Could Have and Won't Have)	4
3	Analyse van de UT-Omgeving	5
3.1	Scenario 1: Niemand heeft de vlag	5
3.2	Scenario 2: Eigen partij heeft de vlag	5
3.3	Scenario 3: Andere partij heeft de vlag	5
3.4	Scenario 4: Wij hebben de vlag gedropt	5
3.5	Scenario 5	6
3.6	Scenario 6	6
3.7	Scenario 7	6
3.8	Scenario 8	6
3.9	Scenario 9	6
3.10	Scenario 10	6
4	Ontwerp	7
4.1	Organisatiestructuur van het Agent-Team	7
4.2	Strategie	7
4.3	Validatie	7
5	Tests van de Implementatie	8
6	Conclusie en Aanbevelingen	9
7	Literatuurlijst	10

1 Inleiding

1.1 Aanleiding

1.2 Doelstelling

1.3 Structuurbeschrijving

2 Programma van Eisen

Om dit project in goede banen te leiden, hebben we een programma van eisen opgesteld die de verschillende aspecten van het project belichten. We hebben het programma van eisen opgesteld volgens het MoSCoW model. [?]

2.1 Minimale Eisen (Must Have)

De volgende eisen zijn de basis van het product en moeten in het programma zitten:

- Het verzamelen van items. De Bots moeten effectief voorwerpen zoals wapens en bepantsering verzamelen om een tactisch voordeel te hebben. Het is essentieel dat de bots dit kunnen, daarom staat het bij de minimale eisen.
- Vlaggen veroveren. De spelmodus die wordt gespeeld is Capture the Flag, oftewel Vlagverovering. Om punten te scoren moet de vlag worden gestolen en teruggebracht naar de eigen basis. De bots moeten de vlag kunnen halen en vervolgens terugbrengen naar de eigen basis.
- Offensief en defensief spelen. In Capture the Flag zijn er veel verschillende scenario's, afhankelijk van welke teams vlaggen beethebben. De bots moeten andere tactieken toepassen als de situatie verandert. Dit is een basisbenodigdheid en zit dus in de minimale eisen.

2.2 Geavanceerde functionaliteit (Should Have)

- Powerup gebruik. De omgeving waarin de bots opereren bevat voorwerpen die het verloop van het spel drastisch kunnen veranderen. De bots moeten kunnen bepalen wanneer zo een voorwerp ingezet moet worden.
- Navigatie. De bots moeten kunnen bepalen naar welke plaats ze moeten navigeren afhankelijk van de situatie. Zonder dit zouden ze doelloos in de omgeving rondlopen.
- Communicatie, coördinatie en teamstrategie. In het product hebben we meerdere bots tot onze beschikking. Om zo effectief en strategisch mogelijk de game te spelen, moeten de bots informatie kunnen uitwisselen, en aan de hand van die informatie een plan opstellen voor elke individuele bot.

2.3 Uitgebreide functionaliteit (Could Have and Won't Have)

- Tactisch herinzetten. Bots zijn in staat om zichzelf te vernietigen om opnieuw in eigen basis te verschijnen. Dit kan gebruikt worden om de vlag te verdedigen als alle bots zich in vijandig gebied bevinden.
- Teamformaties. Door de bots op elkaar af te stemmen met communicatie en eigen waarnemingen, kunnen de bots formaties aanhouden om zo offensief sterker te staan of de vlagdrager te verdedigen.
- Doelwit prioriteren. Door bepaalde waarnemingen te behandelen kunnen bots een voorranglijst opstellen van welke vijandelijke spelers moeten worden uitgeschakeld en in welke volgorde.
- Verbeterde wapenverzameling. Door de bot te laten analyseren hoeveel munitie hij nog met zich meedraagt, kan hij beter bepalen of hij van zijn route moet uitwijken om een wapen of voorwerp op te pakken. Zo verliest de bot minder tijd en word er meer tijd besteed aan het veroveren van de vlag

3 Analyse van de UT-Omgeving

In dit hoofdstuk worden 10 scenario's uit de Unreal Tournament omgeving geschetst. Deze worden later gebruikt om bijvoorbeeld de bots te testen.

3.1 Scenario 1: Niemand heeft de vlag

Aan het begin van het spel worden 2 teams van beide 4 bots aan de uiteinden van de map gezet. Deze teams hebben beide een eigen vlag. Het is de bedoeling om de vlag van de tegenstander naar het eigen kamp te brengen.

Als dus niemand de vlag heeft is het bemachtigen van de vlag van de tegenstander dus het belangrijkste doel. Om dit te bereiken moeten de bots dus naar het andere uiteinde van de map lopen om daar de vlag van de tegenstander op te halen. Ze moeten er echter ook voor zorgen dat ze onderweg niet worden neergeschoten door de tegenstander. Het is daarom slim om er eerst voor te zorgen dat de bots voldoende wapens hebben.

Wat ook belangrijk is, is het beschermen van de eigen vlag. Zo kun je bijvoorbeeld 1 bot als een soort keeper bij de eigen vlag laten staan, om deze te verdedigen.

3.2 Scenario 2: Eigen partij heeft de vlag

Wanneer een van de bots er in slaagt om de vlag te bemachtigen, moet deze proberen de vlag naar de eigen basis te brengen. De tegenpartij zal proberen de vlag terug te bemachtigen of ervoor zorgen dat de vlag van de tegenpartij hun basis eerder bereikt. De beste tactiek hangt dus af van deze twee situaties. Wanneer de tegenstander nog niet over de vlag beschikt, kan het best worden gekozen om zowel de eigen vlag als de *flag carrier* te beschermen. Wanneer de tegenstander wel over de eigen vlag beschikt, zal ook offensief gespeeld moeten worden om de vlag weer terug te bemachtigen en een tegenpunt te voorkomen. Wanneer de *flag carrier* wordt doodgeschoten moeten de andere bots zo snel mogelijk proberen de vlag weer te bemachtigen.

3.3 Scenario 3: Andere partij heeft de vlag

Het is mogelijk dat er een scenario ontstaat waar de tegenstander ons vlag heeft. In dit geval komt naast het bemachtigen van de vlag van de tegenstander, een andere doel om de vlag van ons team terug te brengen voordat het genomen wordt naar hun kamp. Dit is alleen mogelijk als de status van de vlag *dropped* is en een van ons bot erover loopt. Er zijn 2 mogelijkheden om status van de vlag naar *dropped* te krijgen: 1. Als de *enemy flag carrier* een *translocator* gebruikt, dan wordt de flag automatisch neergezet. 2. Als de *enemy flag carrier* doodgeschoten wordt.

Zo is bijvoorbeeld een tactiek om in de groep taken te verdelen. twee bots handelen het doodschieten van de *enemy flag carrier*/terugbrengen van de gestolen vlag, twee bots houden de tegenstanders onder druk door hun vlag proberen te bemachtigen.

3.4 Scenario 4: Wij hebben de vlag gedropt

Er kan een situatie ontstaan dat een bot van ons team de vlag vasthad maar dat die door een bepaalde oorzaak *dropped* is. Een van de oorzaken kan zijn dat de bot is doodgeschoten door een tegenstander terwijl hij de vlag vasthad. Een andere oorzaak kan zijn dat de bot door een *translocator* (ref: goal manual) is gegaan terwijl hij de vlag vasthad. Voor dit scenario kan communicatie handig zijn zodat de andere bots weten waar de vlag voor het laatst geweest was. Een voorbeeld van zo een communicatie is dat een bot steeds aan de andere teamgenoten meldt waar hij is als hij de vlag heeft. Hierdoor weten de teamgenoten al waar de vlag ligt wanneer zij geloven dat de vlaghouder de vlag niet meer vastheeft. Dit is als de bots het percept `FlagState(dropped)` ontvangen.

- 3.5 Scenario 5
- 3.6 Scenario 6
- 3.7 Scenario 7
- 3.8 Scenario 8
- 3.9 Scenario 9
- 3.10 Scenario 10

4 Ontwerp

4.1 Organisatiestructuur van het Agent-Team

4.2 Strategie

4.3 Validatie

5 Tests van de Implementatie

6 Conclusie en Aanbevelingen

7 Literatuurlijst

Bijlage B

DATUM	UREN	BEZIGHEDEN
1 mei 2014	4	<ul style="list-style-type: none">• Start van het project• Aanmaken van de eerste versie van het verslag

Bijlage C

Bijlage D

Bijlage E