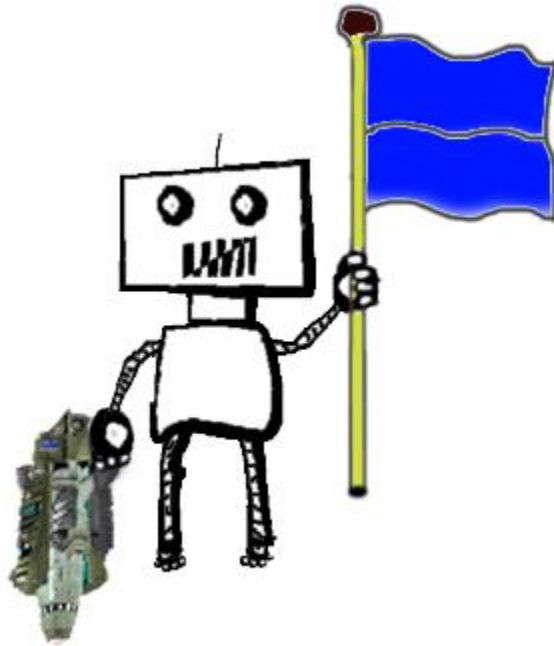


De ontwikkeling van een agent-team in de UT-GOAL omgeving



Groep 21

Özer Ulusoy	4203755
Jeroen Offerijns	4221524
Alexander Felix	1326236
Robbert van Staveren	1527118
Robert Carosi	4242130
Yorick van Pelt	4230264

2. Programma van eisen en analyse

Bij het oriënteren voor de eerste fase, waarbij een ontwerp uitgedacht wordt, is eerst een lijst met eisen en prioriteiten opgesteld, op basis van de MoSCoW-methode, waarbij iedere eis wordt onderverdeeld in vier categorieën: de Must-haves, functionaliteiten die absoluut vereist zijn in een juiste implementatie, de Should-haves, eigenschappen die van belang zijn, en indien mogelijk geïmplementeerd moeten worden, de Could-haves, mogelijkheden die nuttig zouden kunnen zijn, maar niet per se van belang, en de Won't-haves, zaken die in een vervolgproject aan bod zouden kunnen komen, maar in dit project niet geïmplementeerd zullen worden.¹ In dit hoofdstuk zijn een aantal van deze MoSCoW-eisen op een rijtje gezet, te beginnen met de Must-haves.

2.1 Must have's

Dit zijn enkele zaken die in een juiste implementatie vereist zijn.

- Communicatie tussen bots - dit is absoluut van belang², omdat het anders vrijwel onmogelijk is een effectief team te maken dat beter is dan teams die wel communiceren
- Modules voor:
 - Defend the flag - als het andere team de vlag probeert te verkrijgen, dan moet deze verdedigd worden.
 - Capture the flag - de vlag van het andere team moet verkregen worden en naar de eigen basis worden gebracht³.
 - Return the flag - als de eigen vlag in bezit is van het andere team, dan moet degene die de vlag draagt gedood worden, en de vlag teruggebracht worden naar de basis.
 - Attack (help flag capturer)- als het eigen team de vlag heeft, dan moet degene die hem draagt verdedigd worden tot hij bij de basis is.
- Bots verzamelen wapens en health - als bots niets verzamelen, houden zij alleen de twee beginwapens, en gaan ze dood als ze geen health meer hebben, waardoor opgepakte items en positie steeds weer verloren gaan.

2.2 Should have's

Deze functionaliteiten zijn niet vereist, maar indien mogelijk moeten ze wel geïmplementeerd worden.

- Manager - centralisatie van besluiten van bots is vrij nuttig, zodat er geen onenigheid kan zijn, en niet elke bot zijn eigen besluiten hoeft te nemen, of hoeft te communiceren met de rest van het team. Dit scheelt code in de bots, en hoewel het uiteindelijk meer werk is, zorgt het wel voor een simpelere implementatie.
- Bots verzamelen items (ammo, adrenaline, armor) - als bots items verzamelen kunnen ze hiervan de voordelen gebruiken om zo te winnen van tegenstanders. Zo zullen ze minder snel dood gaan als ze armor hebben, door kunnen schieten als ze ammo hebben en combo's uit kunnen voeren als ze adrenaline hebben.

¹ http://en.wikipedia.org/wiki/MoSCoW_Method

² Projecthandleiding Multi-Agent Systemen - eisen voor de uiteindelijke inzending

³ Projecthandleiding Multi-Agent Systemen - eisen voor de uiteindelijke inzending

- Chase opponent - als er eenmaal op een tegenstander geschoten is, heeft deze minder health, en waarschijnlijk ook minder ammo. Als een bot de overhand heeft, is het dus nuttig om een tegenstander achterna te gaan, om hem helemaal te doden.

2.3 Could have's

Deze opties zouden nuttig kunnen zijn, en als er tijd voor is zouden ze geïmplementeerd kunnen worden.

- Change strategy according to score - als de score veel hoger is dan die van het andere team, zouden er meer risico's genomen kunnen worden, of zou een bot 'respawn' kunnen gebruiken om snel health terug te krijgen en op een veilige plek te komen, maar het is goed mogelijk dat dit niet veel voordeel biedt, en daarom is dit een Could-have.
- Follow teammate - het kan handig zijn een teamgenoot te volgen, om elkaar te verdedigen in het geval van een aanval, of samen andere bots aan te vallen, maar het is ons onduidelijk of dit voordeel biedt, ten opzichte van het assisteren door bots toevallig in de buurt zijn, in plaats van elkaar volgen om met meerdere bots dicht bij elkaar te zijn, en een nadeel hieraan is dat sommige items verdwijnen bij het oppakken, en dat de achterste bot zo minder items zal verkrijgen.
- Hide with flag - als een bot de vlag heeft verkregen, dan kan hij zich proberen te verstoppen totdat hij gemakkelijk terug naar de basis kan, maar het is mogelijk dat deze strategie niet goed werkt, omdat bots makkelijker andere bots kunnen vinden dan spelers dat kunnen.

2.4 Won't have's

De volgende zaken zullen in dit project niet aan bod komen, maar zouden in een vervolgproject onderzocht kunnen worden:

- Drop items - het is mogelijk om items op te pakken, en deze aan teamgenoten te geven, maar in de praktijk gebeurt dit nooit, en zijn de mogelijke voordelen gering ten opzichte van de hoeveelheid werk dat dit kost om te implementeren.
- Cheats - er zal geen gebruik worden gemaakt van cheats tijdens het runnen van het programma.

3. Analyse UT-omgeving en beschrijving scenario's

In dit hoofdstuk zal een analyse worden gegeven van de UT-omgeving en aan de hand daarvan zullen er zes verschillende scenario's worden opgesteld die kunnen voorkomen tijdens het spelen van capture the flag in Unreal Tournament.

3.1 Analyse UT omgeving

In deze paragraaf zal eerst een omschrijving worden gegeven van de UT-omgeving waar de bots in rondlopen. Vervolgens hoe de informatie en mogelijke acties uit deze omgeving in een tussenlaag worden omgezet naar percepts en acties in de UT-GOAL omgeving. Deze UT-GOAL omgeving is waar onze eigen agents in uitgevoerd worden. De UT-GOAL omgeving zal tot slot worden vergeleken met de BW4T omgeving.

De bots en het spel capture the flag worden uitgevoerd in het spel Unreal Tournament. De bedoeling is dat een team van 4 bots de vijandige vlag veroverd en terugbrengt naar de thuisbasis, terwijl ze proberen te voorkomen dat de tegenstanders hun eigen vlag veroveren en naar de andere basis brengen. Tijdens dit proces moeten de bots bepaalde obstakels overwinnen. Voorbeelden van obstakels zijn vijandige bots, of onregelmatigheden in het landschap.

Het programma wordt uitgevoerd in de UT-GOAL omgeving. Het programma zorgt ervoor dat het bot-team wordt aangestuurd door agents. Beslissingen van de bots gebeuren in realtime. ~~De beslissingen van de agents echter, worden niet in realtime gemaakt,~~ dit heeft tot gevolg dat bepaalde taken van te voren worden vastgesteld en deze tijdens een verandering in de omgeving worden aangepast.

De BW4T omgeving was ook in realtime, alleen verschilt deze in een paar aspecten van de UT-GOAL omgeving. Zo werd er in de BW4T omgeving telkens een nieuw navigatiepunt als doel opgegeven aan de bot. In de UT-GOAL omgeving is dit niet het geval, omdat de bot anders telkens een korte tijd stil zal moeten staan om een nieuw doel door te krijgen, wat niet erg praktisch is.

Een ander verschil tussen beide omgevingen is dat in de BW4T omgeving het bot-team één hoofddoel had en deze werd opgebroken in een aantal subdoelen. Aan elke bot werd een subdoel als taak toegewezen en vervolgens werd er samen naar het hoofddoel gewerkt. Als de taken van twee bots overlapt werden er aan één van hen een andere tak toegewezen.

In de UT-omgeving zal dit niet het geval zijn. Hier zal vanaf het begin elke agent een vaste taak hebben. Voorbeelden hiervan zijn, aanvaller, verdediger en vlagdrager. Om deze taken uit te voeren heeft de UT-omgeving verschillende opties gegeven in de vorm van vooraf geprogrammeerde acties en percepts. Zo zorgt de actie "Navigate" er bijvoorbeeld voor dat de agent zich kan verplaatsen in de omgeving en houdt een percept als "status" de statistieken bij zoals health, armour, adrenaline. Aan de hand van de verschillende taken zijn er hier onder zes scenario's opgesteld, die daarna kort worden beschreven en vervolgens kort worden samengevat.

3.2 Scenario's

Scenario 1 - Flag Taken

Percepts: Flag Status, See Flag, See Other Bot

Actions: Look-Action, Shoot, Navigate

Op het moment dat een tegenstander onze vlag heeft gepakt, wordt door de Flag Status percept aangegeven dat onze vlag gepakt is. De agents kunnen 2 dingen doen om te verhinderen dat de tegenstander onze vlag naar zijn basis terug brengt en zo punten scoort.

Ten eerste kunnen de agents de tegenstander uitschakelen. Dit begint met de Look-Action om de agent de prioriteit te geven om naar de tegenstander te gaan kijken. De See Other Bot percept geeft aan wanneer de tegenstander zichtbaar is. Dan is het tijd om de Shoot actie te ondernemen en op de tegenstander te schieten. Als de vlag valt en zichtbaar is, wordt dit opgenomen door de See Flag percept en kan de Navigate actie worden gebruikt om naar onze vlag toe te lopen. oppakken van de vlag gebeurt wanneer een agent door de vlag heen loopt.

Ten tweede kunnen we de vlag van de tegenstanders pakken. Door dit te doen kunnen ze onze vlag niet brengen naar hun basis om zo punten te scoren. De agents gaan dan de Navigatie actie ondernemen om naar de vlag van de tegenstander te gaan. Als de vlag zichtbaar is, wordt dit aangegeven door de See Flag percept. De vlag wordt dan opgepakt en de agent die de vlag heeft gaat terug naar de basis met de Navigate actie.

- Als onze vlag gepakt is, moet de tegenstander die de vlag heeft, worden opgespoord.
- De agents die onze vlag beschermden, gaan op zoek naar de vijandige vlagdrager.
- Als de agents de tegenstander hebben opgespoord, wordt er op hem geschoten totdat de vlag valt.

Scenario 2 - Enemy Flag Taken

Percepts: Flag Status, See Flag

Actions: Navigate

Zodra de vlag van de tegenstander gepakt is door één van onze agents, is het de bedoeling dat deze wordt gebracht naar de eigen vlagbasis. Als eerste wordt de Navigate actie gebruikt om naar de vlag te gaan. Door de See Flag percept wordt de vlag gezien. Er wordt gelopen naar de vlag en de vlag wordt opgepakt. en als laatste wordt de vlag naar de basis gebracht. Door dit te doen kunnen er punten worden gescoord. Aangezien deze punten slechts toegekend kunnen worden als onze vlag in de vlag basis aanwezig is. Dit is te zien door de Flag Status percept, moet deze blijvend beschermd worden door andere agents.

- Als de vlag van de tegenstander is gepakt, moet deze naar de eigen vlagbasis worden gebracht.
- De overige agents blijven in de buurt van onze vlag en verdedigen deze.
- De teamgenoot met de vlag moet ook beschermd worden als dit mogelijk is.

Scenario 3 - Health Low

Percepts: Status, See

Actions: Navigate

Wanneer de health van een bot onder de 50% is zal hij op zoek gaan naar health totdat de bot 100% health heeft. De bot zal zijn oude post verlaten dus zal er een andere bot nodig zijn die zijn taak zal opvangen. De bot zal met de Status percept weten hoe het met hem gaat. als hij health nodig heeft dan gaat hij met Navigate actie lopen naar health totdat de bot met de See percept health ziet. om het op te pakken loopt hij door het health heen.

- Bot gaat op zoek naar health totdat hij 100% health heeft
- De dichtstbijzijnde bot neemt de taak over van de bot die op zoek is naar health
- Als de eerste bot genoeg health heeft, keert hij terug en zal de tweede bot zijn oude taak hervatten

Scenario 4 - Flag Dropped

Percepts: Flag Status, See Flag

Actions: Navigate

De vlag valt in een paar gevallen: de tegenstander wordt uitgeschakeld door één of meer agents of de tegenstander schakelt zichzelf uit. Op het moment dat de tegenstander wordt uitgeschakeld, moet de dichtstbijzijnde agent de vlag terugpakken. In het geval dat de tegenstander zichzelf uitschakelt, moet onze vlag zo spoedig mogelijk worden opgespoord en worden teruggebracht naar onze vlagbasis. in ieder geval laat de Flag Status percept zien wanneer onze vlag door de tegenstander is gedropt. Met de Navigate actie gaat een agent naar de vlag. Hij ziet de vlag met de See Flag percept en hij loopt door de vlag heen om de vlag terug te laten verschijnen op de basis.

- Als de vlag is gevallen, moet deze worden opgespoord.
- De agent die de vlag heeft gevonden, pakt deze.
- De vlag wordt teruggebracht naar de vlagbasis.

Scenario 5 - Enemy Flag Dropped

Percepts: Flag Status, See Flag

Actions: Navigate

Wanneer de vlagdrager de vlag laat vallen zal het dichtstbijzijnde teamlid de vlag zo snel mogelijk proberen op te pakken en de rol van vlagdrager overnemen. Vervolgens moet een tweede teamlid de nieuwe vlagdrager beschermen. Met de Flag status weten de bots dat een van de bots de vlag heeft laten vallen, vervolgens moet de dichtstbijzijnde teamlid met de Navigate actie naar de vlag lopen. Met de See Flag percept ziet hij de vlag en loopt er doorheen om het op te pakken. Hij krijgt nu de opdracht om naar de basis te lopen met de Navigate actie.

- De vlagdrager laat de vlag van de tegenstander vallen
- Dichtstbijzijnde teamlid zorgt ervoor dat de vlag wordt opgepakt en neemt de rol van vlagdrager over
- Een tweede teamlid zal worden opgeroepen om de vlagdrager te beschermen
- De vlagdrager brengt de vlag van de tegenstander naar de thuisbasis

Scenario 6 - Both Flags Taken

Percepts: Flag Status, See Flag, See Other Bot

Actions: Look-Action, Shoot, Navigate

Dit is een speciale situatie waarin beide vlagdragers, de vlag niet terug kunnen brengen naar de vlagbasis. Om dit te kunnen doen moet je eigen vlag in de vlagbasis zitten. de beste manier om met deze situatie om te gaan is om de vlagdrager te laten verstoppen met meerdere beschermers. De rest gaat dan op zoek naar de tegenstander die onze vlag in bezit heeft. De tegenstander moet uitgeschakeld worden en de vlag terugpakken. Daarna kan onze vlaghouder de vlag naar de vlagbasis brengen en punten scoren. De Flag status percept laat zien dat beide vlaggen in bezit zijn van bots. een groep agents met de vlag lopen dan met de Navigate actie naar een willekeurige plek dichtbij de basis. de andere groep gaan op zoek naar de andere vlag. Wanneer de bot met onze vlag wordt gezien door de See Other Bot percept, wordt er met de Shoot actie op hem geschoten totdat de vlag valt.

- Beide vlaggen zijn gepakt.
- De agent die vlagdrager verstoopt zich met een paar beschermers.
- De resterende groep gaat op zoek naar de tegenstander met onze vlag en schakelt deze uit
- Onze vlag moet dan worden teruggebracht en de vlagdrager moet de vlag brengen naar de vlagbasis.

Bijlage B - Urenverantwoording

Özer Ulusoy

Bedenken en schrijven/ verbeteren van scenario's - 6.5 uur

Titelpagina Productverslag aanpassen - 0,5 uur

Logo Vernieuwen - 1,5 uur

Roadmap maken - 0,5 uur

Alexander Felix

Bedenken en schrijven/ verbeteren van scenario's - 3 uur

Bedenken hoofdpunten programma van eisen - 0.5 uur

Verbeteren spelling/grammatica en opmaak - 3 uur

Analyse UT omgeving, scenario's schrijven /verbeteren - 8 uur

Robbert van Staveren

Bedenken hoofdpunten programma van eisen - 0.5 uur

Bedenken en schrijven verbeteren van scenario's - 3 uur

Algemene afspraken testen maken en uitwerken - 2 uur

Scenario tests - 3 uur

Analyse UT omgeving schrijven en verbeteren - 4 uur

Yorick van Pelt

Uitwerken en verbeteren van programma van eisen - 2 uur

Verbeteren code - 10 uur

Ontologie definitie verbeteren - 1 uur

Samenwerkingscontract schrijven - 1 uur

Robert Carosi

Verbeteren code - 10 uur

Tickets maken in trac - 0.5 uur

Samenwerkingscontract schrijven - 1 uur

Jeroen Offerijns

Verbeteren code - 4 uur

Ontologie definitie - 2 uur

Literatuurlijst

Bij het schrijven van dit verslag zijn een aantal externe bronnen gebruikt, deze worden hieronder opgesomd.

- 1 Website: http://en.wikipedia.org/wiki/MoSCoW_Method MoSCoW Method, geraadpleegd 12-05-2013.
- 2 Externe Bron: Project Multi-Agent Systemen Projecthandleiding 2012/2013, door Koen Hindriks paragraaf 4.2.2 (Deliverables Fase 2: Implementatie van Basisfunctionaliteit).