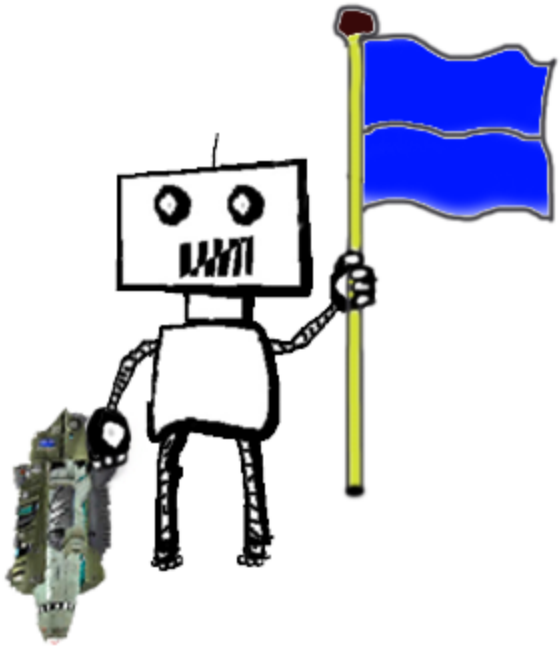
De ontwikkeling van een agent-team in de UT-GOAL omgeving



"21 jaar en ouder"

Groep 21

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

12 juni 2013

Faculteit Elektrotechniek, Wiskunde en Informatica
Technische Universiteit Delft

Voorwoord

Samenvatting

Inhoudsopgave

Inleiding

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 geplaatst in een van de vier categorieën:

1. Must-haves: functionaliteiten die absoluut vereist zijn in een juiste implementatie
2. Should-haves: eigenschappen die van belang zijn, en indien mogelijk geïmplementeerd moeten worden
3. Could-haves: mogelijkheden die nuttig zouden kunnen zijn, maar niet per se van belang
4. Won't-haves: zaken die in een vervolgproject aan bod zouden kunnen komen, maar in dit project niet geïmplementeerd zullen worden.

In dit hoofdstuk zijn een aantal van deze MoSCoW-eisen op een rijtje gezet, te beginnen met de Must-haves.

2.1 Must haves

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

- Communicatie tussen bots - dit is absoluut van belang, omdat het anders vrijwel onmogelijk is een effectief team te maken dat beter is dan teams die wel communiceren. Onder andere om elkaar te verdedigen, samen aan te vallen, dubbel werk te voorkomen etc.
- 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 gebracht.
    - 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.
- 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.
- Bots voeren combo's uit als dit mogelijk is - als bots 100 adrenaline hebben kunnen ze een combo uitvoeren, er zit geen nadeel aan om dit niet te doen, en een klein voordeel om dit wel te doen. De uitdaging zit erin om dit te timen zodat de combo het meest van pas zal komen.

2.2 Should haves

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 duidelijkere implementatie.
- 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 haves

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. Het is echter onduidelijk of dit voordeel biedt ten opzichte van het assisteren van bots die toevallig in de buurt zijn. Een nadeel hieraan is dat sommige items verdwijnen bij het oppakken. Zo krijgt de bot die achter loopt minder items.

2.4 Won’t haves

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. Ook 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.
- 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.

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 tien 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.

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 kosten echter wat tijd. Hierdoor zijn de acties vaak wat vertraagd maar blijft het een realtime omgeving.

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 elke keer 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 die 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 overlapten werd er aan één van hen een andere taak toegewezen.

In de UT-omgeving hebben wij voor een andere aanpak gekozen. De taken van een bepaalde agent worden toegewezen door een manager agent. Deze zal op basis van score, tijd en de positie van iedere bot een geschikte taak vinden en deze toekennen aan de bot. Voorbeelden van taken die een agent kan hebben 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, armor, adrenaline. Aan de hand van de verschillende taken zijn hieronder tien 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 twee 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 door eerst met de look actie de flag-carrier te vinden en vervolgens na het ontvangen van see other bot percept de shoot actie uit te voeren. Dan kan de vlag worden teruggebracht.
Ten tweede kunnen we de vlag van de tegenstanders pakken door middel van de navigate actie. Door dit te doen kan onze vlag niet naar onze basis gebracht worden, om zo punten te scoren.

- 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.

- 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, zullen health pillen steeds belangrijker worden. Hierdoor

zal hij “liever” health oppakken dan bijvoorbeeld wapens en ammo. Om health op te pakken zal de bot de navigate actie gebruiken.

- 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.

- 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 het 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 - Stuck
Percepts: Navigation
Actions: Respawn

Het kan wel eens gebeuren dat een agent vast komt te zitten ergens in de map. Als dit gebeurt dan moet de agent zo snel mogelijk weggaan. Dit kan makkelijk gedaan worden als de agent respawnt. Dan pleegt de agent zelfmoord en komt dan ergens anders in de map terecht waardoor de agent weer verder kan met zijn taak.

- De agent komt vast te zitten en kan niet meer zijn taak volbrengen.
- De agent respawnt en verschijnt nu op een plek in de map waar hij niet vast zit.
- De agent gaat verder met zijn taak.

Scenario 7 - (Re)Spawn
Percepts: Item, Status
Actions: Respawn, Navigate

In het begin is het de bedoeling dat agents zich gaan voorbereiden op hun taak. Om zichzelf goed te beschermen moeten deze agents sterke wapens, armor e.d. verzamelen . Zo kunnen zij lang op het slagveld zitten en aanvallen van de vijand beter overleven. Dit is ook het geval als je zelf respawnt, want bij respawnen verlies je alles wat je tot nu toe had verzameld.

- De agent (re)spawnt en geeft zijn status door.
- De agent gaat op zoek naar wapens, ammo.
- De agent krijgt een taak die hij moet gaan doen.

Scenario 8 - Enemy Spotted
Percepts: See Other Bot
Actions: Navigate, Shoot

Wanneer de agent een vijand tegenkomt, dan gaat de agent naar hem toe om hem uit te schakelen.
Als dit is gebeurd dan gaat de agent verder waarmee hij bezig was. De agent zal ook de locatie van de tegenstander meegeven aan andere agents.

- De agent ziet de vijand.
- De agent gaat naar hem toe totdat de afstand kort genoeg is om op hem te schieten.
- De agent schiet op de tegenstander.

Scenario 9 - Armor Low
Percepts: Status, See
Actions: Navigate

Wanneer de armor van een bot onder de 50% is zal de agent wanneer hij armor of super armor ziet, deze oppakken. Als zijn armor boven de 50% is, dan negeert hij armor behalve super armor. Deze zal hij dan nog steeds oppakken.

- Bot gaat op zoek naar armor totdat hij boven de 50% armor heeft.
- Bot pakt armor op en heeft nu genoeg armor.
- Bot negeert alle armor behalve super armor.

Scenario 10 - Adrenaline Full
Percepts: Status
Actions: Combo action

Via de Status percept kan een agent zien hoeveel adrenaline hij heeft. Wanneer deze vol is kan de agent een combo uitvoeren met de Combo action. Aan de hand van welke taak de agent heeft kan hij kiezen om een van de volgende combo's uit te voeren: speed, booster, invisibility en berserk. Booster is handig om je te beschermen bijvoorbeeld voor FlagDefender. Speed kan je gebruiken als je heel snel ergens moet zijn, voornamelijk handig voor degene die de vlag heeft. Invisibility maakt een agent onzichtbaar. Met Berserk kan je 2x zo snel kogels schieten.

- Bot heeft adrenaline full.
- Bot kiest aan de hand van zijn situatie welke combo hij gaat uitvoeren.
- Bot voert combo uit.

4 Ontwerp

In dit hoofdstuk wordt beschreven welke rollen er zijn binnen het team, hoe deze worden verdeeld en wat de rollen precies inhouden.
Er zullen een aantal rollen zijn, geïmplementeerd in losse modules:

1. Manager
2. FlagCapturer
3. FlagDefender
4. Attacker
5. Defender
6. Assassin

Er is altijd één agent die de rol Manager heeft. Hij zal tijdens het spel bepalen welke rollen er nodig zijn en deze rollen aanwijzen aan de andere agents. Dit doen we zodat er geen onduidelijkheid ontstaat over de rolverdeling tussen de agents en om de communicatie te versimpelen - de meeste berichten hoeven alleen naar de Manager te worden gestuurd en door de Manager te worden opgevangen.

4.1 Manager

Er zal een agent zijn die niet in het spel rondloopt, maar er als het ware boven staat. Deze Manager zal messages krijgen van de andere agents over de status of het verloop van het spel en hierop reageren, door bijvoorbeeld de agents andere rollen te geven om op een juiste manier te reageren op een situatie die zich voordoet.

Aanvankelijk zal dit de verdeling zijn:

- 1 FlagCapturer
- 1 Defender (target de FlagCapturer)
- 1 FlagDefender
- 1 Assassin

Dit zorgt voor een goede balans tussen aanvallen en verdedigen.

Als de vlag van de tegenstander wordt gepakt zal de verdeling als volgt zijn:

- 1 FlagCapturer
- 2 Defenders (target de FlagCapturer)
- 1 FlagDefender

Op deze manier wordt de opgepakte vlag goed verdedigd en tegelijk is de eigen vlag ook veilig.

Als de eigen vlag gepakt wordt, zal de verdeling als volgt zijn:

- 1 FlagCapturer
- 3 Attackers (target de vlagdrager van de tegenstander)

Op deze manier zal de vlagdrager van de tegenstander onder druk komen te staan en waarschijnlijk worden gestopt en zal het spel direct doorgaan doordat de vlag van de tegenstander ook gepakt wordt.

4.2 FlagCapturer

Als een agent de rol FlagCapturer heeft zal hij als voornaamste doel hebben om de vlag van het andere team te veroveren, onafhankelijk of hij op de basis van de tegenstander is of ergens op de grond ligt, en deze terug te brengen naar de eigen basis. Tijdens het bewegen richting de vlag zal hij health en armor oppakken, als verdediging zodra hij de vlag eenmaal heeft, maar geen nieuwe wapens of adrenaline. Zodra hij de vlag heeft zal hij direct richting de eigen basis gaan, weer door alleen health en armor op te pakken (als het dichtbij genoeg is).

4.3 FlagDefender

Als een agent de rol FlagDefender heeft zal hij als voornaamste doel hebben om de eigen vlag te beschermen. Indien de eigen vlag nog op de basis is, zal hij in de buurt van die vlag items gaan verzamelen en rond blijven lopen zodat hij enige tegenstanders makkelijk kan doden.
Is de vlag echter al opgepakt door een tegenstander, dan zal hij gaan navigeren naar de tegenstander en proberen deze te doden en achtereenvolgens de vlag op te pakken.

4.4 Attacker

Als een agent de rol Attacker heeft zal hij als doel hebben om een door de Manager aangewezen tegenstander - vaak degene met de eigen vlag - te doden. Hij zal alleen items verzamelen die heel dichtbij de meest directe route naar de tegenstander liggen, om tijd te besparen en er zo snel mogelijk te zijn. De vlag zal hij volledig negeren, behalve als hij er toevallig langskomt (en dus zo dichtbij staat dat het geen extra tijd kost).

4.5 Defender

Als een agent de rol Defender heeft, zal hij als doel hebben om een door de Manager aangewezen teamlid - vaak degene met de vlag van de tegenstanders - te beschermen. Hij zal alle items verzamelen die binnen een redelijke afstand liggen van zijn route en zal de vlag niet oppakken van de basis. In het geval dat het teamlid dat hij moet beschermen toch gedood wordt zal hij eerst zorgen dat enige tegenstanders binnen zicht gedood worden en daarna de vlag oppakken als deze op de grond is gelegd door het teamlid.

4.6 Assassin

Als een agent de rol Assassin heeft zal hij als doel hebben om zoveel mogelijk tegenstanders te doden. Dit zal hij doen door eerst zoveel mogelijk items te verzamelen, de hele tijd rond te lopen en zich naar een tegenstander te bewegen als er een gespot wordt door een teamlid dat in de buurt is.

6 Testen van MAS

In dit hoofdstuk zal worden ingegaan op verschillende tests. Eerst zullen de algemene afspraken over het testen worden toegelicht en daarna worden de scenario-tests besproken behorende bij de scenario's beschreven in hoofdstuk 3. Tot slot geven we de resultaten van deze tests.

6.1 Algemene afspraken

Nadat een functionaliteit is geschreven, zal deze direct handmatig gecontroleerd worden door degene die de code heeft geïmplementeerd. Dit zal indien mogelijk gebeuren door static testing (systematisch doorlopen van de code) en daarnaast door functionele tests (programma uitvoeren en gedrag bekijken). Vervolgens laat hij de functionaliteit door tenminste één ander persoon testen.

Indien we vermoeden dat een nieuwe functionaliteit invloed kan hebben op een andere functionaliteit zullen beide getest worden na het schrijven van de nieuwe functionaliteit, zowel door de auteur als één andere persoon.
Ten slotte zullen alle functionaliteiten iedere fase eenmaal getest worden om te zorgen dat deze ondanks wijzigingen in de code correct blijven werken.

6.2 Scenario-tests

Natuurlijk zullen alle scenario's uitgebreid worden getest, met als doel achter alle fouten in de implementatie en ongeïmplementeerde zaken te komen. De manier waarop dit gebeurt zal hieronder worden beschreven.

Scenario 1 - Flag Taken

Een manier om dit scenario te testen is een agent en een human in de UT-environment te laden. De human (één van de ontwikkelaars), gaat de tegenstander spelen. Hij gaat dan de vlag pakken en ergens verstoppen. Vervolgens wordt de environment 5 x versneld en wachten we totdat de human wordt uitgeschakeld en de vlag valt.

Scenario 2 - Enemy Flag Taken

Bij het testen van dit scenario zal er op een aantal punten worden gelet. Ten eerste of de bot met de vlag direct de eigen basis als nieuwe bestemming heeft, ten tweede of de verdedigers hun taken behouden en onze vlag in de vlagbasis beschermen en tenslotte of de overige agents de agent met de vlag proberen te beschermen, door bijvoorbeeld de tegenstanders die op hem jagen, te blokkeren.

Scenario 3 - Health Low

Dit scenario kan getest worden door ervoor te zorgen dat de health onder de grens van 50% valt. Dit zou bijvoorbeeld gedaan kunnen worden door de health van te voren in te stellen op een getal onder de 50%. Vervolgens wordt gekeken of de bot ook daadwerkelijk op zoek gaat naar health. Tegelijkertijd zal in de

gaten worden gehouden of een andere bot de taak van de vorige overneemt. Ten slotte zal gecontroleerd worden of de bot stopt met zoeken naar health als hij 100% heeft en hij zijn oude taak zal hervatten.

Scenario 4 - Flag Dropped

Het scenario dat een vlag op de grond is gevallen kan op de volgende manier worden getest. Er worden één agent en twee humans in de UT-environment geladen. Één human speelt de tegenstander. De human steelt de vijandelijk vlag en gaat ergens zitten. De andere human gaat naar hem toe en schakelt hem uit. Dan wordt er gekeken of de bot reageert zoals het hoort. En er wordt gekeken of hij de gevallen vlag zoekt en vindt.

Scenario 5 - Enemy Flag Dropped

Als de vlag van de tegenstander op de grond is gevallen, moet een bot de vlag oppakken. Zorg er vervolgens voor dat de vlagdrager wordt gerespawnd in de visualizer. Kijk of de rol van vlagdrager wordt overgenomen door een teamlid en een tweede teamlid deze zal beschermen tot aan de thuisbasis.

Scenario 6 - Stuck

Om dit te testen zal een willekeurige bot gevolgd worden totdat hij ergens vast komt te zitten, zodra dit gebeurd hoort de bot vanzelf te respawnen.

Scenario 7 - (Re)Spawn

Scenario 8 - Enemy Spotted

Scenario 9 - Armor Low

Scenario 10 - Adrenaline Full

6.3 Testresultaten

Scenario 1 - Flag Taken

Bij het testen van Flag Taken voegde één van de leden van ons team zich bij het vijandige team (het

blauwe team in dit geval) en werd de vlag van de bots (het rode team) gepakt, vervolgens werd er geprobeerd om naar de blauwe basis te lopen. Terwijl dit gebeurde werd de gebruiker succesvol gestopt door het rode team van bots en de vlag vervolgens weer terug gebracht naar de rode basis. De test is hiermee succesvol verlopen.

Scenario 2 - Enemy Flag Taken

Bij het testen van Enemy Flag Taken werd het programma gedraaid totdat de vijandige vlag werd gepakt en deze terug werd gebracht naar de thuisbasis. Tijdens de test werd één bot erop uit gestuurd om de vlag te halen en nadat hij de vlag in handen had, bracht hij deze ook terug naar de thuisbasis. Terwijl hij dit deed was het de bedoeling dat de beschermers van de thuisbasis op hun post zouden blijven en de vlaggendrager vergezeld zou worden door een ander teamlid dat als taak had om de vlaggendrager te beschermen. Dit was echter niet het geval, de andere bots namen niet hun beschreven taken aan, in plaats daarvan bleven zij hangen op de basis van de tegenstander en ging de vlaggendrager alleen, zonder ondersteunend teamlid terug naar de thuisbasis.

Scenario 3 - Health Low

Bij het testen van Health Low speelde één van de leden van ons team weer de rol van tegenstander. Vervolgens werd er op een bot geschoten totdat deze minder dan 50% levenspunten had. Hierna werd gekeken of de beschoten bot ook daadwerkelijk op zoek ging naar meer Health packs totdat deze weer het volle aantal levenspunten had.
Dit was ook het geval, alleen werd er tijdens dit proces ook naar wapens gezocht en had het zoeken naar Health packs niet altijd de prioriteit. In de gevallen waar de bot zowel bij een Health pack als een wapen stond, kreeg de Health pack niet altijd de voorkeur en werd er schijnbaar willekeurig een keuze gemaakt tussen beide items. De bot bleef echter wel zoeken naar Health packs totdat hij weer het volle aantal levenspunten had en hervatte vervolgens weer zijn oude taak.

Scenario 4 - Flag Dropped

Bij het testen van het flag-dropped scenario, is de vlag ergens achtergelaten, en is gekeken of de bot de vlag oppakte, dit was inderdaad het geval. Echter, de bot liep als de vlag ergens gevallen was, gewoon naar de basis van de tegenstander om de vlag te zoeken, en zocht deze niet systematisch zelf naar de vlag. Pas toen de vlag in het zicht was liep de bot ernaartoe om hem terug te brengen.

Scenario 5 - Enemy Flag Dropped

Bij het testen van het enemy-flag-dropped scenario, is gekeken of bots die de vlag in het zicht hadden deze oppakten, en of er systematisch door aangewezen bots naar de vlag werd gezocht. Het eerste was wel het geval: bots die de vlag zagen liggen pakten hem op en brachten hem naar hun eigen basis. Het tweede echter niet, de bots sloegen niet op waar de vlag was achtergelaten en zochten niet meer naar deze plek.

Bijlage B - Urenverantwoording

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
MAS testen schrijven - 1 uur
Module flag defender schrijven - 9 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
Module defender/flag capturer schrijven - 8 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
MAS testen schrijven - 1 uur
GOAL syntax highlighting - 5 uur
Code main schrijven - 4 uur

Robert Carosi
Verbeteren code - 10 uur
Tickets maken in trac - 0.5 uur
Samenwerkingscontract schrijven - 1 uur
Samenwerkingscontract nakijken - 1 uur
Attacker module maken - 7 uur
Helpen met de manager module - 2 uur

Jeroen Offerijns
Verbeteren code - 6 uur
Ontologie definitie - 2 uur
Samenwerkingscontract doorlopen - 1 uur

Ontwerpverslag - 4 uur
Manager module maken - 9 uur
Productverslag verbeteren - 1 uur

Bijlage C - Begrippenlijst

Agent	Een klein computerprogramma dat in opdracht van een gebruiker of van een logische redenering bepaalde taken uitvoert.
BW4T	Blocks World For Teams, een omgeving waar door middel van de GOAL programmeertaal bepaalde taken kunnen worden uitgevoerd.
Functionele tests	Testen door middel van uitvoering van het programma observatie van het gedrag.
GOAL	Een logische programmeertaal voor het maken van programma's die inspele op de omgeving waarin deze worden uitgevoerd en zo beslissingen maken.
Realtime	Realtime betekent zonder wachttijden als gevolg van verwerken van gegevens. In deze context wordt ermee bedoeld dat de agents kunnen inspelen op de omgeving zonder dat hier (significante) tijd tussen zit.
Static testing	Testen door middel van systematisch doorlopen van de code.
UT-GOAL	De omgeving tussen Unreal Tournament en GOAL. Met behulp van deze omgeving kunnen onze GOAL programma's communiceren en inspelen op de UT omgeving.

Literatuurlijst

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

- Website: http://www.projectsmart.co.uk/moscow-method.html MoSCoW Method, geraadpleegd 12-05-2013.
- Externe Bron: Project Multi-Agent Systemen Projecthandleiding 2012/2013, door Koen Hindriks paragraaf 4.2.2 (Deliverables Fase 2: Implementatie van Basisfunctionaliteit).