

Huidige situatie BW4T

Grote bak met code zonder tests. BW4T wordt gebruikt voor onderwijs (eerste jaar maar ook master) en door onderzoekers die strategieën willen testen. Voor elk van deze contexten is een andere handleiding en documentatie vereist.

Restructuring

- Documentatie opbouwen uit losse onderdelen die per context bij elkaar gezocht kunnen worden. Dit kan in latex gebeuren. Een andere mogelijkheid is om alles in 1 groot bestand te zetten met een deel per context.
Hierbij moet gelet worden op Goal vs Java en expert vs beginner.
- **Belangrijk:** Her-structureren. Eerst bekijken hoe de structuur in elkaar zit en in hoeverre het spaghetti is.
Belangrijk is om te kijken naar dependencies en method lengths. Daarna tests schrijven.
- **Belangrijk:** Unit tests. Op dit moment zijn er geen tests. Er moeten unit tests komen voor alle onderdelen. Als er iets nieuws toegevoegd wordt moet de originele code ook nog werken naar behoren.
- Plugin architecture: Het gaat niet zo zeer om plugins alhoewel Koen het wel zo genoemd heeft. Het gaat erom dat toegevoegde onderdelen er uit gegooid kunnen en dat het programma daarna nog werkt.
Dit kan door gebruik te maken van interfaces en packages. Zorgen dat de basis functionaliteit van BW4T er in blijft zitten en dat die uitgebreid kan worden.
- EIS: makkelijk om aan goal te koppelen. We moeten de implementatie van EIS bekijken.
Percepts gaan nu naar alle agents, willen we dat wel?
- **Belangrijk:** Repast: We moeten komen met een voorstel. We moeten kijken of we Repast wel echt goed gebruiken waar het voor is. Het is namelijk een tool om simulaties te bouwen. Het heeft geen zin om zelf een stepper te bouwen als er al 1 in Repast zit.
Repast waar de robots lopen en Goals waar de agents zich bevinden draaien los van elkaar en kunnen elkaar niet beïnvloeden. Nu is de vraag wat de link is tussen de agents en de robots. Gaat Repast wachten tot er voor elke agent een instructie is?
Repast heeft als het goed is een discrete event simulator en dat is wat we willen. Dat er per tick acties plaatsvinden. Dat levert elke run eenzelfde eindtoestand als we geen niet-deterministische agents gebruiken. Dus hoe belangrijk is Repast, het moet haalbaar zijn qua codering. Als het te veel werk dan niet doen. Dus uitzoeken en daarna overleg.

Scenario GUI

Nu is er een init functie met parameters die veranderd kunnen worden. Deze lijst van parameters moet groter. Nieuwe opties toevoegen dus. (Later behandeld)

De stakeholders willen graag een Scenario GUI waarin configuratie files gemaakt kunnen worden die naast de MAS file gebruikt kunnen worden. Hierin kan worden aangegeven welke map gebruikt wordt, hoeveel bots, welke human, welke niet, eigenschappen per bot.

Er moet voorkomen worden dat sommige logische combinaties niet kunnen. Er kan bijvoorbeeld geen e-partner zijn zonder human of andere onmogelijke combinaties. Let ook op dat de hoeveelheid bots wel in de initieele ruimte past.

Verder de optie of een aparte GUI om simulaties te runnen. Hierin moet aangegeven kunnen worden hoe vaak en welke configuraties uitgevoerd moeten worden. Deze simulaties moeten doorgaan als er 1 faalt en als output de resultaten hebben. Dus eigenlijk batchruns kunnen doen. GOAL kan dat al, dus even kijken hoe we dat aanpakken.

Human player GUI

- Er is een experiment geweest met BW4T met pijltjes toetsen. Dit was geen succes onder alle testers. Er zitten niet altijd gamers tussen en mensen komen vast te zitten. Daarbij moet het aansturen van de bots en humans hetzelfde werken. Commando's moeten dus vergelijkbaar zijn.
Belangrijk: De commando's die gegeven kunnen worden moeten van toepassing zijn op de desbetreffende bot. Een blinde bot mag geen opdracht kunnen krijgen om een rood blokje te pakken. Dus geen commando's weergeven die er niet zijn.
- Misschien messages van E-partner weergeven in lijst

Communicatie

Lage prioriteit: De vraag is of dit mogelijk is. Agents kunnen namelijk altijd communiceren als ze dat willen. Als een agent geprogrammeerd wordt om iedereen te melden wat hij beleeft dan zullen alle agents dit ook weten. Dit kan niet worden gereguleerd in BW4T.

Er zit wel een msg actie in BW4T om tussen robots te communiceren. De vraag is of we hier iets mee kunnen en de communicatie dus zelf moeten regelen voor een deel.

In principe kunnen limited range en package loss gewoon in Goal geprogrammeerd worden door de gebruiker.

Bot

Belangrijk: Dit zijn belangrijke aspecten. Door heel abstract problemen te definiëren kunnen werkelijke problemen gesimuleerd worden.

- Kamer is dicht
- Waarom weet iedereen wat er in de dropzone gebeurt
- Batterij die leeg gaat, oplader
- Deur opmaken
- Deuren gaan random dicht
- Kleurenblind
- Gripper, kan een bot iets oppakken
- Wat nog meer?

Botstore

Eigenlijk al behandeld

e-Partner

Belangrijk! (Birna is fan): Een bot die assisteerd. Denk aan een tablet. Een e-partner in BW4T is een blokje. Als de human het blokje vast heeft kan het meer dingen. De human stuurt het blokje aan en kan met andere e-partners communiceren.

Misschien sneller bewegen? Omdat alles makkelijker gaat.

Fysische aspecten

Belangrijk: Bots mogen niet over elkaar heen kunnen lopen. Dus in een gang moeten ze op elkaar wachten. Als ze botsen kunnen ze stuk gaan of dood gaan. Of minder goed werken.

Environment store

Belangrijk: Een environment store waar maps gegenereerd kunnen worden.

Twee gewilde opties:

- Random map generator: Waar parameters ingevoerd kunnen worden.
 - Aantal maps
 - Aantal kamers
 - Aantal bots
 - Blokkades
 - Zonder beperkingen van bijvoorbeeld max 26 kolommen.
- Handmatige map maken: Waarbij zelf onderdelen geplaatst kunnen worden om een realistische omgeving te creëren. Dus minder die structuur van kolommen.
- Pathfinder: als 2 bots niet langs elkaar kunnen blijven ze misschien stil staan tegen elkaar aan en loopt het vast.
- Environments kunnen opslaan.