

Architectuur, logica, onzekerheid

Aangezien de simulatie nog niet zo groot is is er weinig te zeggen over de logica onzekerheid en architectuur.

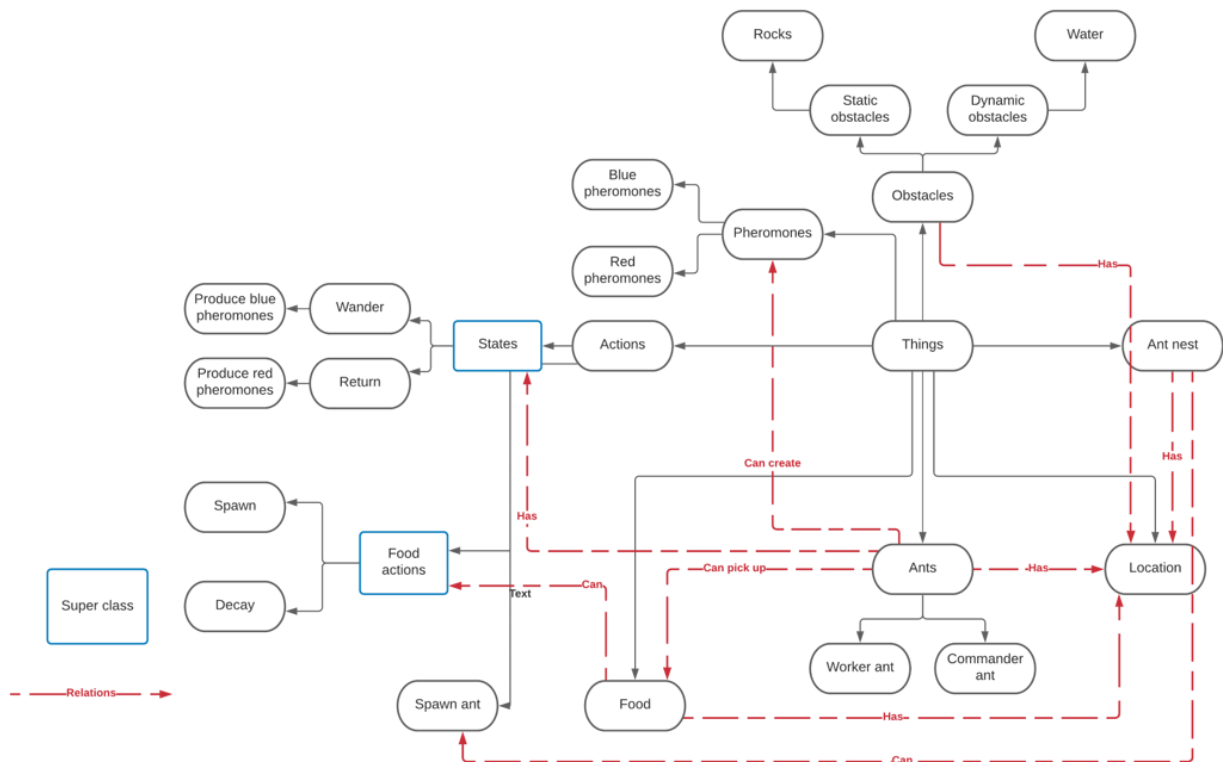
Wander state:

- A: Vind voedsel
 - B: Terug naar nest
 - C: Terug naar voedselbron
- A -> B
B -> C

Onzekerheid:

Als er random clusters van met een diverse grote, bijvoorbeeld een grote van 2 t/m 6, in het veld valen. Dan is er een onzekerheid over hoeveel meer voedsel er te vinden zal zijn op een locatie zodra een mier wat voedsel vindt. Het is dan onduidelijk hoeveel mieren je het best op het voedsel kan afsturen. Als de cluster maar 2 groot is zou je niet meer mieren hoeven sturen, maar als de cluster 5 groot is kan je er misschien beter 4 op af sturen om het voedsel zo snel mogelijk binnen te krijgen. Dit zal al helemaal een groot probleem zijn als het voedsel binnen een bepaalde tijd zou verdwijnen. Waardoor je moet gaan kijken naar het risico dat als je maar 1 mier er op af stuurd hij niet alles op tijd binnen kan halen en als je er 4 op af stuurt zullen sommige misschien voor niks naar die locatie gaan.

Ontologie



Dit is de ontologie die past bij de tweede versie van onze simulatie. Zoals je ziet zijn er een aantal dingen in deze simulatie veranderd of toegevoegd, dit zijn:

De locatie, de plek waar de mieren kunnen lopen, deze plek heeft:

- Obstakels, obstakels kunnen meerdere dingen zijn:
 - Dynamische obstakels
 - § Een dynamisch obstakel is water.
 - Statische obstakels.
 - § Een statisch obstakel is een steen.
- Ook heeft de locatie een mieren nest, plek waar mieren wonen en hun voedsel naar toe brengen. Dit mierennest kan mieren 'in spawnen'.
- Op de plek zijn dan natuurlijk ook Ants, de mieren hebben meerdere states:
 - Wander, het rond open om te zoeken naar voedsel en teruggaan naar een plek waar voedsel is gevonden om te kijken of er meer is.
 - § Als de mier in de wander state is produceert hij blue pheromones.

- Return, teruggaan naar het nest met het voedsel.

§ Als de mier in de return state is produceert hij red pheromones.

de mier kan ook eten oppakken.

- Een plek heeft ook Food, het eten dat de mieren zoeken om terug naar het nest te brengen. Food heeft meerdere eigenschappen:
 - Het spawnen, er komt eten op de map.
 - Het verrotten van het eten, na een bepaalde tijd verrot het eten.

Coding Adventure: Ant and Slime Simulations. (2021, 25 maart). [Video]. YouTube.
<https://www.youtube.com/watch?v=X-iSQQgOd1A>

Beschouwen

1. Hoe verhouden jouw experimenten zich toch experimenten in je bronnen? Is het een uitbreiding, verdieping, herhaling, etc?

Wij willen de experimenten uit onze bronnen herhalen, en eventueel uitbreiden als dat mogelijk is, en als er tijd voor is. De herhaling is het vinden van zo veel mogelijk voedsel. De uitbreidingen zijn het effect van een commander ant, het effect van het strooien van voedsel in de simulatie en het effect van aanpassingen aan de variabelen, bijvoorbeeld de decay snelheid van feromonen.

2. Welke relatie zie je tussen de resultaten van jouw experimenten en je bronnen? Verklaar die relatie.

Er is nu nog niet veel overlap met de bronnen die we hebben, omdat onze feromonen niet helemaal geïmplementeerd zijn. Wij willen het effect van een commander mier achterhalen, en kijken hoeveel positieve/negatieve effecten dit heeft. Zodra de feromonen ook goed werken, willen we ook observeren of het aanmaken van feromonen ook werkt.

3. Hoe zou jouw simulatie gebruikt worden in nader onderzoek naar artificiële en/of biologische intelligentie?

In onze simulatie worden mieren gesimuleerd die zo veel mogelijk eten naar hun nest moeten brengen. Er zou een vervolgonderzoek gedaan kunnen worden naar de evolutie van mieren, en of het mogelijk zou zijn om bijvoorbeeld een commander mier te hebben in een mierenkolonie die mieren leiden. Evolutie zal namelijk altijd een lokaal optimum bereiken, en nooit een organisme 'verslechteren' om het uiteindelijk beter te maken. Evolutie is de verandering van een organisme om zich aan de omgeving aan te passen en een betere fitness te krijgen, de omgeving van een organisme is hierbij een belangrijke invloedsfactor. Dus mieren zullen niet zomaar gaan evolueren naar een hiërarchie met command, maar ze zullen dat misschien pas gaan doen zodra de omgeving zo ongunstig is dat een hiërarchie nodig zal zijn.

De simulatie met de commander zal inzicht bieden en een evolutie vorm voor mieren waarbij een hiërarchie is gecreëerd. De mieren zullen dan gaan werken via commands gegeven door de mier die hoger staat in het command en dan kan er gekeken worden of dit efficiënter is. Als dit zo zou zijn zouden de mieren kunnen gaan evolueren tot deze vorm. Wel is het zo dat voor het evolueren van een organisme je een ongunstige omgeving nodig hebt waar een organisme zich kan aanpassen voor een betere fitness.

Bronnen

- Coding Adventure: Ant and Slime Simulations. (2021, 25 maart). [Video]. YouTube.
<https://www.youtube.com/watch?v=X-iSQQgOd1A>