

ABD design document

Life Of A Nanite: LOAN

Inleiding

Het idee is om het gedrag van nanite-agents door het lichaam te simuleren. Dit zijn futuristische minuscule autonome robots. Hierbij bestaat het lichaam uit een undirected weighted graph en heeft de agent een bepaalde snelheid en energie. Zodra er iets gebeurt (op een node van de graaf) moet de agent daar zo snel mogelijk komen met zoveel mogelijk energie. Het bewegen tussen twee nodes kost namelijk energie. Hierbij kan de agent kiezen om óf met de bloedsomloop mee te gaan óf juist tegen de bloedsomloop in te gaan. Meegaan met de bloedsomloop kost weinig energie maar kan langer duren, tegen de bloedsomloop ingaan kost veel energie maar duurt minder lang.

Verder is er een bepaalde kans is dat iedere beweging tussen twee nodes een kans heeft om fout te gaan, het kost dan bijvoorbeeld meer energie of de agent beweegt niet. Ook is er een kans is dat de agent bij een kruispunt, wanneer een node meer dan twee edges op zich heeft, een verkeerde richting opgaat.

Samenwerking

Werktijden:

- Maandag
 - ◆ 17:00 - 18:00 (overleg voor de week)
- Dinsdag
 - ◆ 9:30 - 12:30 (ABD-les)
- Woensdag
 - ◆ 9:30 - 12:30 (werksessie)
 - Eventueel uitbreiden naar gehele woensdag in samenspraak met INNO groepjes
- Donderdag
 - ◆ 9:30 - 12:30 (werksessie) (Stan kan niet altijd)
- Vrijdag
 - ◆ 9:30 - 12:30 (ABD-les)

Communicatie:

De communicatie geschiedt volledig via Discord. Eventuele berichten met prioriteit worden via Whatsapp verstuurd.

Code:

De code wordt opgeslagen in een prive-repository op GitHub:
<https://github.com/Casper-Smet/LOAN>

Projectmethodologie

Er wordt gewerkt op een Kanban-manier. Iedere twee weken wordt er een nieuw Kanban-bord opgeslagen onder een verhaal op de gezamenlijke Jira. Alle leden kunnen taken op het bord toevoegen. In overleg worden deze taken toebedeeld en daadwerkelijk uitgevoerd.

<https://assistance-intern.atlassian.net/secure/RapidBoard.jspa?rapidView=2&useStoredSettings=true&atlOrigin=eyJpIjoiZDFiMGFkNTMzYWY5NGUxMmI3NzczOWRhZGNIYTc2MjcjLCJwIjoiIj9>

PEAS

Performance measure

- Aantal problemen verholpen voordat energie op is:
 - Meer problemen verhelpen is beter
- Het totaal aantal “hit points” van het lichaam aan het einde van de simulatie

Environment

- Menselijk lichaam, met het bloedvatstelsel gerepresenteerd door een Undirected weighted graph of twee directed weighted graphs (met weights voor energie)
- Nodes in lichaam worden “ziek”, de agent kan dit genezen
- De environment heeft een terminal state, want:
 - De mens kan dood
 - Of n -tijdstappen worden behaald
- Hoogte van de ‘severity’ van de verwonding per node (hurt points):
 - Iedere tijdstap geeft de verwonding een severity score
- Eigenschappen van de omgeving:
 - Fully observable
 - Voorlopig single-agent, uiteindelijk multi-agent
 - Stochastisch
 - Episodic
 - Dynamic
 - Discrete
 - Known

Actuators

- Tentakels voor beweging
- Dynamo voor energie generatie
- Chemische rotzooi voor opsparen energie
- Magische krachten voor het helen van zieke nodes (op dit moment instant healing)

Sensors

- Kruispunten
- Eigen locatie
- Eigen energie
- Locatie alle nodes in het lichaam
- Energieverbruik tussen nodes (weights)

Toepassen van de challenge kennisrepresentatie & redenatie

- Architectuur
 - Er wordt gebruik gemaakt van Mesa en NetworkX. De agent is omniscience en heeft een model van de gehele wereld. Daarom is de agent tenminste model-based. Het doel van de agent is om de 'happiness' van de omgeving te maximaliseren. Hierbij streeft de agent niet naar het eigen geluk. Daarom is de agent goal-based.
- Ontologie
 - $Nanite(n) \rightarrow HasEnergyCapacity(n) \wedge PartOf(n, Graph) \wedge MovesOn(n, Graph) \wedge CanHealIllVertex(n, Graph)$
 - $HumanModel(h) \rightarrow \forall_v [Vertex(v) \wedge v \in h \rightarrow CanBeIll(v)] \wedge HasHitpoints(h) \wedge Nanite(n) \wedge PartOf(n, h) \wedge CanDie(h) \wedge Graph(h)$
 - $E_1 \in Healings \wedge Nanite(n) \wedge Vertex(v) \wedge IsOn(n, v) \wedge Terminates(E_1, IsIll(v), t)$
- Gebruik van logica

v	Vertex
n	Nanite
t	Tick

PerceivableBy	Vertex is vindbaar door Nanite
PositionedAt	Nanite bevindt zich op Vertex
GetsIllAt	Vertex wordt ziek bij tijdstap

- Alle Vertices zijn bekend bij alle Nanites
 - $\forall_v, \forall_n [PerceivableBy(v, n)]$
- Iedere Nanite bevindt zich op één Vertex, op één Vertex kunnen zich 0 of meerdere Nanites bevinden
 - $\forall_n, \exists !_v [PositionedAt(n, v)]$
- Elke tick kunnen een of meerdere vertices kunnen ziek worden
 - $\forall_t, \exists_v [GetsIllAt(v, t)]$
- Elke tick beweegt een Nanite naar een volgende node, of geneest de huidige node

- Omgang met onzekerheid

Op dit moment komt er nog geen specifieke onzekerheid voor in de agent wanneer hij op weg gaat, echter komt er wel een onzekerheid voor tijdens het bewegen en in de omgeving.

- De agent weet waar een ill vertex is, maar hij weet niet wanneer er een andere komt. Hij kan dus op weg gaan naar een ill vertex, terwijl op dat punt ergens anders een ill vertex kan voorkomen. Op dat punt moet de agent opnieuw berekenen waar hij beter eerder naartoe kan gaan.
- De agent kan meegezogen door het bloed (onverwachte event of een extra sterke pomp van het hart) naar een verkeerde richting wanneer hij langs een kruispunt komt (hoeveelheid edges > 3) en hij heeft een kans van $P(0.2)$ om een verkeerde kant op te worden gestuwd.
- Verder is er nog een kans van $P(0.2)$ dat er ergens in het lichaam een vertex ill wordt. Dit kan iedere stap bij maximaal 1 vertex gebeuren.