
Intro to agents and agent-tools

Wat heb ik gemaakt?

Ik heb met Unity een simulatie gemaakt van een demonstratie. Er is een veld en daarin lopen demonstranten random rond. Er loopt ook een politieagent random rond. Als de politieagent te dicht bij een demonstrant in de buurt komt wordt de demonstrant boos. Dit is op te merken omdat de boze demonstrant een rode kleur krijgt.

Mijn project is afgeleid van de gegeven tutorial. Ik heb niet precies de tutorial gevolgd omdat je dan zelf minder nadenkt en er dus minder van leert. Ik heb er dus mijn eigen versie van gemaakt.

Theorie vragen

1. Wat unity anders maakt dan andere tools is dat Unity extreem uitgebreid is en meer gefocust is op graphics en game development. De voordelen hiervan zijn is dat je bijvoorbeeld natuurkundige simulaties heel mooi kan weergeven. Het nadeel is dat je een simpele simulatie niet even 100 keer snel kan laten runnen, daar is het namelijk echt niet voor gemaakt. Mijn programma is wel agent-based omdat er verschillende agents rondlopen die allemaal hun eigen gedrag hebben en daarmee het geheel van de demonstratie kan simuleren.
2.
 1. De initiele staat is dat alle activisten nog vrolijk zijn en rondlopen. Ook de politieagent loopt gewoon rond.
 2. De 'See' functie die kijkt dus of er een politieagent te dichtbij is. Als die te dichtbij is wordt de activist boos. Deze functionaliteit kan alleen in een klein rondje/radius om zich heen kijken of er een politieagent in de buurt is.
 3. De 'act' functie is dat een activist van happy naar boos gaat. Dit wordt getriggerd als de politieagent in zijn radius komt.
 4. In de updatefunctie maakt die veranderingen in het bewegen.
3.
 1. De simulatie is inaccessible. De agents kunnen alleen zien of iemand zich in hun radius bevindt. Verder ziet die niks en beweegt die maar een random kant op.
 2. De simulatie is deterministic, omdat elke actie maar één effect heeft. Komt er een politieagent bij een activist in zijn radius dan verandert de happy activist naar een boze activist. Als die al boos wordt die opnieuw boos dus verandert er niks.
 3. De simulatie is een episodic environment, omdat de agents in de simulatie al hun keuzes baseren op de current episode en niet op voorgaande episodes.
 4. De simulatie heeft een static environment, omdat de agents niks aan de omgeving kunnen veranderen waar ze zich in bevinden, ze kunnen alleen van state happy naar boos gaan. Stel dat boze activisten de omgeving gingen slopen dan was het dynamic geweest, omdat ze dan hun omgeving kunnen aanpassen.
 5. De simulatie heeft een discrete environment omdat de agents de agents vast zitten op een klein grasveld met muren er om heen. De agents blijven dus de hele tijd in het zelfde hokje en de omgeving veranderd dus niet.

4.
 1. Als we van de simulatie in plaats van discrete een continuous environment maken, dan kunnen de agents dus het grasveld verlaten en wordt de kans een heel stuk kleiner wat de politieagent bij hun in de buurt kan komen.
 2. Als we van de simulatie in plaats van een static environment een dynamic environment maken, dan zouden de agents bijvoorbeeld de activisten dingen kunnen gaan slopen. Dan geef je de simulatie dus eigenlijk een heel ander doel door te kijken wat ze dan allemaal kapot maken en hoelang dat duurt.
 3. Als je de simulatie accessible maakt, dan weten de activisten dus waar de politieagent zich de hele tijd bevindt en zou je kunnen programmeren dat ze de politieagent de hele tijd opzoeken of juist er voor weg lopen. Als de activisten de politieagent zouden opzoeken en ze hebben altijd dezelfde start locatie dan zou het altijd even lang duren voordat alle activisten boos zijn.