

Programmeren opdracht 3: Intro to agents and agent-tools – individueel

Naam: Stan Haakman

Datum: 3-12-2021

Klas: V2A

De tutorial

Het grootste voordeel van Unity is dat het erg toegankelijk is voor nieuwe gebruikers door de duidelijke interface. Het is makkelijk te leren maar er zijn ook vele mogelijkheden waardoor het voor gevorderde gebruikers nog steeds geschikt is. Voor mij het nadeel was dat bij het coderen juist wel heel veel voorkennis van Unity nodig is om goed gebruik te maken van alle voor gedefinieerde instanties en objecten. Het is super uitgebreid doordat al erg veel gemaakt is maar daardoor is het voor een nieuwe gebruiker moeilijker om te begrijpen ten opzichte van een andere taal.

Ik heb super veel tijd gestopt in het proberen uit te breiden maar dit is mij naast dat wat ik heb op dit moment niet gelukt. De interface alleen was al best veel om uit te vogelen en de code ernaast viel mij heel erg zwaar. Ik heb alle 2 de simulaties gemaakt en ook nog extra tutorials geprobeerd. Ik lever beide de tutorials vanuit canvas in en daar zal ik verder in dit document naar refereren.

Stateful agent

De simulatie die ik heb proberen te maken is tikkertje. Er is dus een agent die probeert tegen de enemies aan te lopen.

De agent kan in meerdere states zijn. Een agent kan op zoek zijn naar enemies of achter een enemy aan rennen of ze kunnen, terwijl ze achter een enemy aan rennen, andere agents roepen om te komen helpen. De initiële staat van de agent bestaat dus uit hun positie en waar ze naartoe lopen. De initiële staat is dus de state waar een agent in zit als hij/zij spawned.

De functie See neemt de state van de omgeving dus waar obstakels zich bevinden, of er andere agents in de buurt zijn en luistert naar geluid van een andere agent. Ook kan die enemies zien. Vervolgens gooi de See functie het gepaste precept eruit. Dus bijvoorbeeld bij het horen van andere agent zal deze worden waargenomen en gaat de agent naar het geluid totdat hij op locatie zijn of hij een ander enemy ziet.

Vervolgens bepaalt de functie Act wat er daadwerkelijk gebeuren moet. Bijvoorbeeld als er een geluid gehoord wordt, dan wordt de A of actie het volgen van het geluid. Of als de agent een enemy ziet, dan gaat de agent daar achteraan.

Dichotomiën

1. Mijn simulatie is Accessible omdat de agent toegang heeft tot de info van de omgeving en in welke state die zicht bevat.
2. Mijn simulatie is Non-Deterministic omdat bijvoorbeeld het horen van de 'call' van een andere agent niet altijd leidt tot het achtervolgend van een enemy. Dit kan ook eindigen tot het verder zoeken.
3. Mijn simulatie is Episodic omdat hij niet voorspelt waar de enemies zouden kunnen zijn of heen zouden kunnen gaan. Hij maakt dus geen gebruik van mogelijk toekomstige states.
4. Mijn simulatie is Static omdat er bijvoorbeeld geen veranderingen in de omgeving kunnen plaats vinden waardoor de agent zich anders zou gaan gedragen.
5. Mijn simulatie is discrete omdat er een vast bepaald aantal acties en percepties kunnen plaatsvinden en er dus geen verdere acties/percepties kunnen plaatsvinden wat nieuwe acties/percepties naar boven brengt.

Alternatieve omgeving

1. De dichotomie met de meeste invloed in mijn simulatie zou zijn Static → Dynamic. Het kan namelijk veranderingen bij de agent geven zoals:
 - a. Bij mist minder zicht hebben en dus langer op zoek moeten zijn voor een enemy.
 - b. Tijdens een storm kan het roepen van een andere agent niet meer gehoord worden en dus niet altijd kunnen reageren.
 - c. Of als er bijvoorbeeld plekken in de omgeving zijn waar de agent zou kunnen schuilen. Dat zorgt voor een goede strategie als het donker is maar weer minder efficiënt als het licht is.
2. Daarnaast zou het wijzigingen van Episodic → Non-episodic ook een grootte invloed hebben op de kwaliteit van de agents. Maar ook een grote invloed kunnen hebben op de kwaliteit van de enemies. Als die vooruit kunnen kijken en kunnen voorspellen wat er gaat gebeuren dan kan er een betere strategie gebruikt worden om langer weg te blijven van de agent of om eerder bij de enemy te komen.
3. Om de bovenstaande wijzigingen extra veel effect te laten hebben is het wijzigen van Deterministic → Non-deterministic nodig. Hierdoor kan de agent of enemy niet op ieder moment de state van de environment/omgeving zien waardoor hij moet reageren op wat er om zich heen gebeurd. Waardoor de agents/enemies veel meer adaptable moeten worden naar iedere situatie omdat ze niet alles aan zien komen.
4. Ook heeft het door dezelfde reden als de vorige veel invloed als Accessible gewijzigd word naar Inaccessible. Hierdoor moet de agent veel preciezer op onderzoek gaan om een enemy te vinden.