De simulation

In onze agent based simulatie heb je in de eerste iteratie 1 agent. Deze agent (medic) is degene die probeert om geïnfecteerde patiënten te genezen met een cure. Deze cure moet eerst door de agent gevonden kunnen worden. De agent kan de cure uit zichzelf vinden maar ook de locatie in zijn database verkrijgen door eerst ook een radio te vinden.

De objecten in de simulatie zijn dus de radio, patiënten, genezen patiënten en de cure. Alleen de radio en patiënten maken geluid in hun directe omgeving via “static” of een “scream”. Dat is informatie dat de agent kan gebruiken om steeds meer informatie van zijn omgeving te verkrijgen. De geluiden worden alleen 1 positie naast de geluidsbron gegeven (niet diagonaal)

De omgeving is een grid van minstens 4 bij 4 groot waar alleen recht overeen kan bewogen worden en niet diagonaal. De omgeving kan aangepast worden om groter, maar niet kleiner, dan 4 bij 4 te zijn.

De simulatie eindigt wanneer er geen patiënten meer zijn (ze zijn allemaal genezen), of de agent bevindt zich op dezelfde locatie van een patiënt. Een patiënt kan genezen worden door naast hen te staan en de cure in zijn richting te geven.

Deze iteratie van de simulatie is bedoeld om een agent een bepaald doel te laten volbrengen door maten van logica, kansberekeningen en deductie. De agent moet doormiddel van logica meer informatie verkrijgen dan dat het krijgt door zijn perceptie. Deze simulatie zal nog uitgebreid worden om een meer ingewikkeld doel te simuleren waar nieuwe informatie uit kan komen.

Uitleg keuzes

De eerste keuze van onze simulatie was het gebruik van de Mesa library voor het programmeren van de agent en omgeving. Mesa is relatief makkelijk te gebruiken maar is ook handig voor meer ingewikkelde simulaties, waar deze simulatie naartoe gewerkt zal worden. Daarnaast is Mesa ook geschreven in python, de makkelijkste taal om te gebruiken en waar ons team het meest competent in is. Hiernaast was ook de keuze van NetLogo, maar dat blijkt niet de beste keuze te zijn in het maken van complexe simulaties.

Dan hebben we de keuze in de bewegingsvormen van de agent(s). We hebben gekozen voor een zo simpel mogelijke omgeving om vooral te kunnen focussen op de logica en deductie aspecten van de simulatie en agent. Hiervoor is dus daarom gekozen om de constraints zo simpel mogelijk te maken en is de bewegingsmogelijkheden van de agent alleen aangrenzend en dus niet diagonaal. Deze keuze zal vooral invloed hebben om de lage aspecten van het implementeren van de simulatie, maar een grotere grid zal ook minder deze contraint een affect laten hebben.

De cure kan niet met deductie zomaar gevonden worden omdat je agent met geluk het alleen kan vinden. Hiervoor is dus bedacht om een radio toe te voegen met een geluidseffect. Dit zorgt ervoor dat de agent niet alleen makkelijker de mogelijkheid heeft om de cure te vinden, maar 1 cure kan meerder radio’s bevatten met grotere geluidsbronnen waardoor de variabelen die invloed hebben op hoe makkelijk de cure gevonden kan worden veranderd kunnen worden.

Iteraties

In onze eerste iteratie hebben we voornamelijk de ontologie en logica van de simulatie concreet uitgebracht, hierin is dus verwoord wat de objecten en aspecten zijn van de simulatie, wat de agent is, wat zijn doel is, hoe hij kan interactie met zijn omgeving en welke keuzes in acties het heeft. Er is gekeken naar de logica van de knowledge base en dit beschreven. De ontologie is gemaakt met alle aspecten en relaties. (Zie file ABDOntologieChal1). De keuze in de environment soort was gemaakt waaronder de thema, idee, concept en doel zoals beschreven in de paragraaf 1: De simulation,

In onze tweede iteratie zijn wij begonnen met het coderen van de simulatie, hierin zijn we voornamelijk gaan kijken naar het werkend maken van een programmable agent die “een interactie” kan hebben met zijn environment. Dit is dan ook het moment wanneer we de GitHub hebben opgezet. In het programma zijn we ook wat basis acties die de agent kan uitvoeren geïmplementeerd, de simulatie is opgesteld, de objecten en de events die kunnen plaatsvinden. Daarnaast zijn wij ook in de theorie begonnen met het maken van de inference voor de agent waar hij met zijn tot zo ver verkregen logica meer informatie kan verzorgen.

In de laatste iteratie van deze demo hebben we de gehele GitHub klaargezet en alle benodigde onderdelen erin gezet plus een ReadMe, In de logica van de Agent is de omgang met onzekerheid vermeld en uitgelegd. Daarnaast is de presentatie compleet gemaakt, in deze presentatie zijn ook schetsen gemaakt om de simulatie beter te kunnen demonstreren. In de simulatie kan nu alles in termen van objecten neergezet worden zonder dat iets overlapt, waaronder Cures, radio’s, patiënten en medics. De medic kan nu sterven als het op dezelfde vak van een patiënt komt en daarmee de simulatie eindigen. De medic krijgt nu alle info over het vak waar hij op staat. Het model kan nu ook met meerdere variabelen gemaakt worden met verschillende grid groottes en aantallen in medics, patiënten en radio’s. De grid werkt functioneel. Naast de Radio en patiënten worden ook de screams en static geluidsbronnen geplaatst. Als de medic op een radio terecht komt zal het de coördinaten van een cure krijgen.