Opdracht 3 simulation Bram van Leusden V2B

Opgave 1

Mesa is een goede balans tussen Netlogo en Unity, Netlogo is heel erg simpel en is meer gemaakt om snel een simpele simulatie te maken met een duidelijke visualisatie. Het is ook moeilijk om erg specifieke gedrag eigenschappen toe te passen voor agents als deze complexer worden. Unity heeft veel mogelijkheden voor simpele en complexe simulaties maar is erg complex. Unity is een game engine dus er kunnen prachtige simulaties mee worden gemaakt maar hiervoor is veel kennis vereist. Mesa lijkt visueel redelijk op Netlogo maar sinds Mesa Python gebruikt als codeer taal is er veel meer bewerkbaarheid voor Mesa toegevoegd dan Netlogo (bijvoorbeeld: visualisatie en agent fictionaliteit). Mesa is minder complex dan de C# taal die Unity gebruikt. Ook is de leer curve van Mesa een stuk lager dan Unity helaas zijn de visualisaties van Mesa een stuk minder frappant.

Opgave 2

Mijn model heeft een X aantal agents die starten met een wealth van 1, als een agent op het zelfde vakje staat als een andere agent geeft de agent 1 wealth aan een willekeurig agent op dat vakje. Elke iteratie van het model beweegt elke agent naar een willekeurig aanliggende positie. Als de agent geen geld heeft kan deze geen geld meer weggeven en blijft hij alleen doorlopen zonder geld weg te geven.

Concept 1

De initiële staat van iedere agent is het hebben van 1 wealth en hun positie. Na elke stap kan de positie en de wealth veranderen.

Concept 2

In de functie “ Give_money(self)” in de onderstaande regel code kijkt de agent naar welke agents op hetzelfde vakje als hem staan.

Concept 3

Na concept twee checkt de agent of hij 1 of meer wealth heeft en zo ja dan geeft de agent deze wealth weg aan een willekeurige agent

Concept 4

De wealth die de agent zojuist heeft wegegeven en het vakje waar de agent naartoe is gelopen worden bijgewerkt en het model gaat door naar de volgende iteratie/agent.

Opgave 3

1. Accessible vs inaccessible

Mijn model is inaccessible: de agents lopen willekeurig door de omgeving heen zonder enige voorkeuren. En de agents houden geen rekening met de positie van andere agents.

2. Deterministic vs non-Deterministic (Stochastic)

Mijn model is non- Deterministic: De start posities van de agents zijn willekeurig en de plek waar elke agent heen gaat is ook willekeurig. Je kunt alleen aangeven hoeveel agents er zijn, dus dezelfde input kan compleet verschillende uitkomsten geven.

3. Episodic vs non-episodic (Sequential)

Mijn model is non- Deterministic: De agents hun actie word alleen bepaald op hun huidige status ze houden geen rekening met acties die ze voorheen hebben uitgevoerd. Het enige wat de actie van een agent beïnvloed is de huidige positie van de agent zelf en de andere agents en de huidige wealth van de agent.

4. Static vs Dynamic

De omgeving in mijn model is static: De omgeving waar de agents overheen bewegen veranderd nooit. De agents lopen er over heen geven wealth weg en herhaal dat is het.

5. Discrete vs continuous

De omgeving in mijn model is continuous: de omgeving waar de agents zich op bevinden is “toroidal” wat betekent dat de randen van de omgeving aan elkaar verbonden zijn dus als een agent over de rand heen gaat komt hij aan de andere kant van de omgeving weer opdagen, dus agents kunnen altijd een kant op ongeacht wat de agent wilt doen. Het aantal iteraties word door de persoon aangegeven en is nergens anders van afhankelijk en dus theoretisch oneindig.

Opgave 4

Ik zou de omgeving accessible maken voor de agents, zodat als ze een bepaalde hoeveelheid wealth hebben verkregen. De agents van andere agents proberen weg

te lopen om hun wealth te behouden en als ze geen wealth hebben juist naar andere agents toe lopen om wealth te krijgen, omdat ze nu weten waar andere agents zijn. Ook zou ik de omgeving “Dynamic” maken om bijvoorbeeld wealth dat in de omgeving valt toe te voegen, wat de agents dan kunnen oppakken om zo weer wealth te krijgen zonder wealth weg te geven. Als laatste zou ik de simulatie “episodic” maken zodat er vendetta’s kunnen ontstaan. Sommige agents die vaak wealth aan een specifieke agent hebben moeten geven krijgen een vendetta tegen die agent. De agent met de vendetta zoekt zijn doelwit gericht op als deze hij geen wealth meer heeft.