

## Opdracht 3 'King of the Mesa'

1) Ik heb de tutorial gevolgd die werd aangeraden, het model waar waar agents elkaar 'wealth' geven. Ik heb wel één kleine aanpassing gemaakt omdat dat van mij verwacht werd. De aanpassing is dat, als de agent die je aanspreekt geen wealth heeft, geef je die agent 2 wealth inplaats van één (één van jezelf en één van de 'game'). Ik ben op het idee gekomen omdat ik de Gini-coëfficiënt<sup>1</sup> wel een beetje té hoog vond. Na de aanpassing is de Gini score gedaald van tussen de 0,7-0,8 naar tussen 0,4-0,5. Dus mijn 'doel' is behaald (De code is beter en uitgebreider uitgelegd in de notebook). Dit is een simulatie van de economie, deze aanpassing zou natuurlijk niet werken in de echte wereld, omdat er uit het niets geld gegenereerd wordt, dat zou natuurlijk leiden naar inflatie. Wat ik één van de voordelen vindt aan Mesa is dat het in Python is, daarom is het voor iedereen die python beheerst simpel om de code te snappen, aan te passen en te maken. In het algemeen is het leren van Python ook makkelijker, dat is ook de reden dat meeste mensen Python als eerste taal hebben geleerd. Dat is nog een voordeel voor de mensen die alleen agent-based willen leren programmeren, als ze het dan in Mesa doen pikken ze het sneller op omdat het Python is. Een groot voordeel is ook dat het een alles-in-een is, je kan het gebruiken voor het bouwen, analyseren en visualiseren van agent-based modellen. Het enige nadeel wat ik zou kunnen bedenken is dat de visualisatie niet echt top tier is, het is niet te vergelijken met bijvoorbeeld Unity maar dat is ook wel te begrijpen aangezien Unity een game engine is die onder andere wordt gebruikt voor grote games zoals Hearthstone, Rust en Cities: Skylines

2) De agents hebben een interne staat en die bestaat uit hoeveel wealth hij momenteel heeft en de locatie waar hij zich bevindt (See), de agent kan alle agents zien die in hetzelfde veld staan als hij. Als de agent een andere agent tegenkomt wordt er wealth gegeven aan die agent, als die agent 0 wealth heeft krijgt hij 2 wealth. Als die agent meer dan 0 wealth heeft krijgt hij 1 wealth (Act).

3) Accessible vs inaccessible: mijn omgeving is inaccessible, de agents kunnen niet alles zien. De agent kan allen zijn wealth geven aan een agent op hetzelfde veld.

Deterministic vs non-Deterministic (Stochastic): mijn omgeving is deterministic, het is gegarandeerd dat iemand wealth krijgt. Er is geen mogelijke state waar niemand wealth krijgt.

---

<sup>1</sup> Mr. Chadd. (2020, 20 november). *De Gini-coëfficiënt* - Mr. Chadd Academy. Direct uitleg voor ieder vak.

Episodic vs non-episodic (Sequential): dit is non-episodic omgeving, de actie van de agent wordt beïnvloed door de andere agents. Bijvoorbeeld: Als er geen andere agent op een veld staat (wat komt omdat de andere agent is weggegaan) of als hij een agent tegenkomt zonder wealth (wat altijd het gevolg is van een agent omdat iedere agent start met één wealth) dan moet hij namelijk meer geven.

Static vs Dynamic: Mijn omgeving is static, niets heeft invloed op de omgeving behalve de acties van andere agents. De omgeving wordt bijvoorbeeld niet opeens kleiner of zo iets dergelijks.

Discrete vs continuous: Mijn omgeving is discrete, er is maar één keuze wat een agent kan maken, de enige keuze is aan welke agent de agent zijn wealth geeft en hoeveel wealth. Dat zijn dus maar een bepaald aantal opties. Dus het is een discrete omgeving

4) Een paar aanpassingen die gemaakt zouden kunnen worden om 3 dichotomie tegenovergesteld te krijgen zijn:

1. Het zou een Dynamic map kunnen worden. Door bijvoorbeeld de map kleiner te maken als de Gini-coëfficiënt bijvoorbeeld hoger is dan gewenst, om zo het gevoel te simuleren wat mensen kunnen krijgen als het geld niet goed verdeeld is, en dan niet kunnen doen wat ze willen omdat ze geen geld hebben. Nog een idee om een map dynamisch te maken is, het weer. Het weer kan veel invloed hebben op het humeur van mensen. Als het slecht weer is zullen mensen sneller geïrriteerd worden en zullen ze misschien geen geld geven. (het tweede voorbeeld is een combinatie van punt 1 en 2.)
2. De omgeving zou non-Deterministic gemaakt worden door bijvoorbeeld iets toe te voegen als humeur. Zodat als een agent een andere agent tegenkomt er niet altijd geld gegeven wordt, maar misschien juist geld gestolen omdat hij in een slecht humeur is. Als je dit implementeert is het simuleren van de economie toch nog één tikkie dichtbij reëel, omdat er in het ook zulke dingen gebeuren natuurlijk. De uitkomst van de data lijkt mij ook wel interessant om dan te vergelijken met de data voor dat het humeur werd toegevoegd.
3. Een goed idee om de omgeving non-episodic te maken is dat als iemand bijvoorbeeld net is beroofd om iemand heeft beroofd. Dat ze dan anders behandeld worden dan de rest, als iemand met beroofd is geven andere agents misschien wel meer wealth bijvoorbeeld. En als de agent net iemand beroofd heeft is er een kans dat hij genegeerd wordt bijvoorbeeld.

De voorbeelden lopen een beetje in elkaar over, punt 3 is niet mogelijk zonder combinatie met punt 2 (het humeur van de mens). Net zoals het tweede voorbeeld van punt 1, die is ook niet mogelijk zonder punt 2. Het humeur van een mens heeft namelijk zoveel inpakt op de mensen om hem/haar heen.