

# Simulation Tooling Opdracht 3.

## 1.

Deze programmeer taal wordt geïnterpreteerd en niet gecompileerd.

De programmeer taal is ook echt agent based, alles draait om de 'turtles' en de 'patches' waar ze op staan, ook zijn er veel functies die helpen met het ordenen/ organiseren en beheren van de agents.

Er zijn wel wat nadelen aan de taal zoals dat het zoeken tussen de agents niet gemakkelijk aan te passen is waardoor als je zoekopdrachten uit gaat voeren het langer kan duren en je programma zal vertragen.

Ook is het een nadeel dat je maar in een bestand kan werken.

Wel is het erg gemakkelijk om de simulatie weer te geven en statistieken te produceren en input variabelen aan te passen.

Het voorbeeld dat ik heb gevolgd is die van de simpele economy, deze heb ik aangepast door neighbours aan te duiden.

Het originele model liet per ronde elke agent een euro betalen aan een willekeurig andere agent. Dit heb ik aangepast door een subset van agents te maken waar één specifieke agent zijn geld aan zal geven.

Deze subset bestaat uit alle andere agents die binnen  $-x\%$  van zijn eigen wealth en  $+x\%$  van zijn eigen wealth liggen.

$X$  is een getal tussen 0 en 100 dat een percentage aanduidt relatief gezien van de totale spreiding (de armste en de rijkste agent)

Voorbeeld:

Agent 1 heeft 40 euro, de rijkste agent heeft 450 euro, en de neighbourhood size is 20%

Dan zal Agent 1 alleen maar zijn 1 euro willekeurig uitgeven bij iemand die minimaal -5 (of dus 0) euro heeft en maximaal 95 euro heeft.

## 2.

1. De initiele staat is in het begin altijd hetzelfde  $x$  aantal agents, met  $y$  hoeveelheid geld, een voorkeur  $z\%$  voorkeur waar de hun geld uitgeven en daar uit een subset met agents die in hun voorkeur ligt.
2. De staat  $S$  is alle agents met  $x$  aantal geld  
De Perceptie van één specifieke agent hangt  $-z\%$  en  $+z\%$  af van zijn eigen welvaart.
3. De Act is een transactie die plaatsvindt tussen de agent en een van de agents die in de update vallen.
4. De Update is anders omdat dit model alleen purely reactive agents bevat, hun interne staat van wealth bepaald de perceptie die ze hebben en zal de interne staat aanpassen.  
Het enige dat geupdate wordt aan de hand van de interne staat is of ze wel of geen geld uitgeven, een agent met 0 wealth zal niks uitgeven.

3.

1. Accessibility: Dit model is inaccessible omdat de agent alleen maar met een subset van andere agents interacteren.
2. Dit model is non-Deterministic omdat de overdracht van geld random gebeurt.
3. Het model is ook Episodic omdat de agents zelf bepalen met welke andere agents ze interacteren hier wordt niet met een verleden gewerkt en alleen naar de huidige staat gekeken.
4. De omgeving is statisch omdat de agents de enige zijn die invloed hebben op de omgeving.
5. Ik denk dat het een continue omgeving is omdat de roulette van geld nooit een duidelijke eind staat heeft.

4.

Je kan een beter economisch model maken door iedereen met elkaar te laten handel en relatief aan hun inkomen te laten uitgeven zo wordt de accessibility **accessible** dit is ook realistischer.

Door een overheid te introduceren zou je de omgeving kunnen veranderen door agents een percentage belasting te laten betalen maar belangrijker armen mensen te belonen, de toeslag die de overheid aan agents uitdeelt zou ook kunnen afhangen aan hoeveel armen agenten hij moet voorzien zo wordt de omgeving **dynamisch**.

Ook zou je armere agents die een tijd onder x geld zitten extra overheidssteun kunnen toewijzen, zo zorg je voor een **Non-episodic** model.