

Simulation opdracht 3 – Individueel

Student: Justin Klein

Nummer: 1707815

Bron Tutorial: <https://www.youtube.com/watch?v=VeQkhfDYyMc>

1.)

Mesa is een tool die heel modulair is, maar alleen 2D simulaties kan draaien. Dit zijn dan meestal simulaties met een 'birds-eye view'. Het maken van classes en de server waarop de simulatie draait is vrij gemakkelijk te begrijpen en te maken. Unity heeft het voordeel dat de simulaties in een 3D omgeving worden uitgevoerd wat meer mogelijkheid biedt tot een realistischere simulatie. Unity vereist echter wel meer kennis van programmeren (de taal C#) en is dus lastiger op te pakken. Als laatste is NetLogo een heel handige tool wat ongeveer hetzelfde kan doen als Mesa alleen simpeler. NetLogo gebruikt zijn eigen programmeertaal. In mijn ervaring is Mesa op dit moment beter, omdat je niet in de specifieke NetLogo taal hoeft te schrijven. Mesa gebruikt o.a. Python wat een goede basis is voor programmeren en dus makkelijk op te pakken is. Python is echter geen goede taal voor multi-threaded processes wat grootschalige simulaties mogelijk lastiger zal maken.

2.)

Concept	Beschrijving	Mijn simulatie
1	Dit concept geeft aan dat iedere agent een interne staat heeft aan het begin van de simulatie. (= een bepaalde situatie zoals "Ik zit nu op een bankje" of "Ik sta nu niks te doen")	In mijn simulatie is dit de staat van infectie, omdat de actor geïnfecteerd kan zijn of niet. Verder is de agent namelijk alleen aan het bewegen.
2	Dit geeft aan dat een agent moet kunnen begrijpen in welke omgeving hij zich nu bevindt (See) van zijn huidige staat. Hij moet daar een input uit halen (Perception) om een actie uit te kunnen voeren.	In mijn simulatie kijkt de agent naar zijn huidige omgeving en of daar andere agents zijn die een infectie hebben of niet. Als de agent niet geïnfecteerd is en er is een geïnfecteerde agent in de buurt dan wordt een functie aangeroepen die bepaalt of de non-geïnfecteerde agent wordt geïnfecteerd.

3	Aan de hand van de perceptie (van de huidige staat) moet een handeling worden uitgevoerd.	Zoals bij nummer 2 vermeldt staat voert de simulatie een functie uit die bepaalt of een non-geïnfecteerde agent wordt geïnfecteerd als een geïnfecteerde agent dichtbij is.
4	De agent en de staat moet worden ge-update met nieuwe informatie na het uitvoeren van de actie of maken/krijgen van een perceptie.	In mijn simulatie wordt de staat ge-update waarmee een non-geïnfecteerde agent nu dus geïnfecteerd is.

3.)

- **1. Inaccessible**

De simulatie is inaccessible, omdat de agents wel accurate informatie krijgen over de agents in hun omgeving (en de status van deze agents), maar niet de informatie krijgen van de gehele omgeving.

- **2. Non-deterministic**

De simulatie werkt zo dat een non-geïnfecteerde persoon die naast een geïnfecteerde persoon staat twee "wegen" kan bewandelen. De non-geïnfecteerde persoon wordt geïnfecteerd of hij blijft non-geïnfecteerd. Dit geeft dus non-deterministisch gedrag aan omdat er geen 100% duidelijke weg is bij dezelfde perceptie en er dus iedere keer twee mogelijkheden zijn.

- **3. Episodic**

In de simulatie kijkt de agent alleen naar zijn huidige staat en de staat van de agents die dichtbij zijn. De infectiestaat wordt dus alleen bepaald per stap en er wordt niet gekeken naar de toekomst.

- **4. Static**

In de simulatie wordt alleen de verspreiding van infecties weergegeven, geen effecten zoals koud/warm weer of andere ziektes die ontstaan binnen de omgeving worden weergegeven wat de simulatie dus Static maakt. Mocht het weer of de temperatuur veranderen wat een impact heeft op de infecties dan is de simulatie dynamisch.

- **5. Discrete**

In de simulatie worden de infectiestaten **per stap** gecheckt. Dit betekent dat de personen niet continu rondlopen en het eens in de zoveel tijd wordt gecheckt zoals in de echte wereld.

4.)

- **2. Non-deterministic -> Deterministic**

De simulatie zou in dit geval ALTIJD een ander persoon infecteren als een non-geïnfecteerde agent een geïnfecteerde agent zou tegenkomen. Dit betekent dat er dus maar 1 mogelijkheid is om de staat te veranderen.

- **4. Static -> Dynamic**

Als de simulatie het weer in acht zou nemen, zoals warm weer wat de infectieverspreiding verergerd of koud weer wat de spreiding verminderd, en dit zou willekeurig voorkomen dan wordt de simulatie dynamisch.

- **5. Discrete -> Continuous**

Als de agents in de continu zouden rondlopen en de infectiestaat eens in de zoveel tijd zou wordt gecheckt in plaats van per stap dan zou het dichterbij de echte wereld liggen wat de simulatie continuus maakt.
