

Touring Machines

Traffic Simulation

Gijsbert Nutma



Inhoudsopgave

Inhoudsopgave	2
1. Inleiding	3
1.2 Onderzoeksvraag	3
1.3 Plan van aanpak	3
1.3.1 Agent	3
1.3.2 Environment	3
1.3.3 Toolkeuze	3
1.3.3.1 Mesa	4
1.3.3.2 NetLogo	4
1.3.3.3 Unity	5
1.3.4 Score toolkeuze	6

1. Inleiding

Voor opdracht 5 van Simulation Tooling zullen we een ABM (Agent-Based Model) gaan bouwen voor het verkeer zoals we dit ook in de werkelijkheid kennen. Wij zullen ons vooral gaan focussen op verkeerscongestie. Dit is een tijdelijke verstopping in het verkeersnetwerk wat in ons geval veroorzaakt wordt door het plotseling remmen van een auto. Hierdoor ontstaat er een golf van vertraging en zullen de auto's op een gegeven moment tot stilstand komen. Het model wat wij gaan maken bestaat is een variatie op: '*The Nagel-Schreckenberg model*'.¹ Dit model is rond 1990 ontworpen door twee Duitse natuurkundigen K. Nagel en M. Schreckenberg.

1.2 Onderzoeksvraag

Met de kennis die we hebben opgedaan met het *Nagel-Schreckenberg model* en voorbeeldsimulaties van *NetLogo* hebben we een onderzoeksvraag geformuleerd die ons helpt bij het onderzoeken van filevorming:

"Hoe hebben drukte en maximumsnelheid effect op filevorming in het verkeer?"

Onder 'drukke' verstaan we het aantal auto's op één rijbaan. Een 'file' wordt gedefinieerd als auto's die niet meer hun originele snelheid hebben en deze niet meer kunnen bereiken.

1.3 Plan van aanpak²

1.3.1 Agent

De Agents in ons model bestaan alleen uit auto's. Deze hebben weer elk hun attribuut 'speed' (snelheid). De Agents beginnen met een start-speed, deze is hetzelfde als de ingestelde maximumsnelheid. De regels die de agenten verder meekrijgen zijn:

- Remmen/stoppen als de Agents te dicht bij elkaar in de buurt komen. Hiervoor hebben we een 'target-brake' die dit beïnvloed. Zodra de gebruiker van de simulatie klikt op deze button zal de target-auto (rood van kleur) op de rem gaan.
- Op één rijbaan blijven.
- Auto's kunnen niet door elkaar heen. Het is de bedoeling dat er op 1 patch van de rijbaan maar 1 auto op kan rijden omdat de auto's elkaar anders gaan overlappen.

1.3.2 Environment

Zoals hierboven al toegelicht zullen de auto's alleen op één rijbaan gaan rijden. De rijbaan bestaat uit een grid met cellen/patches. De environment heeft een oneindige rijbaan (net zoals in het *Nagel-Schreckenberg model*). In ons model zullen de agents alleen op elkaar reageren.

1.3.3 Toolkeuze

Aan de hand van het SFA model zullen we de verschillende tools: Mesa, Unity en NetLogo tegen elkaar afwegen. We zullen alleen op *Suitability* en *Feasibility* beoordelen en punten uitdelen van 1 t/m 5. De tool met de meeste punten zal onze voorkeur hebben.

¹Bron: https://en.wikipedia.org/wiki/Nagel-Schreckenberg_model

²Zie *model.png* voor volledige visualisatie van de planontwikkeling

1.3.3.1 Mesa

Suitability

Tool support: De gewenste user story zou zeer waarschijnlijk gemaakt kunnen worden in Mesa voornamelijk omdat je met goede kwaliteit van alles kan visualiseren. en natuurlijk dat je het programmeert in python waarin je praktisch alles kan maken als je het maar kunt bedenken. Er zit echter geen module ingebouwd die ervoor zorgt dat de Agents elkaar kunnen zien en met elkaar kunnen communiceren, bijvoorbeeld welke snelheid de andere Agent rijdt. Dat zou dan zelf gecodeerd moeten worden wat tijd zou kunnen kosten. **Punten** : 3

Performance efficiency: Een nadeel van Mesa is als je heel veel code hebt en/of als je code niet heel efficiënt opgebouwd hebt, de animatie in Mesa minder goed zal verlopen. Python is te lang bezig met de calculaties waardoor je animatie langzaam en stotterend kan verlopen. **Punten** : 2

Compatibility: Voor dit project zal er waarschijnlijk geen externe data nodig zijn maar mochten we het alsnog nodig hebben is het geen probleem omdat Mesa in python is geschreven. Hierdoor kan Mesa met praktisch alle externe data werken. **Punten** : 4

Feasibility

User-friendly: Bij Mesa moet je de hele simulatie maken en is er geen basis om vanaf te werken. Dit kan voor beginners best lastig zijn. **Punten** : 1

Technically feasible: Wij denken dat het voor ons niet mogelijk is om een goede simulatie te maken in Mesa binnen 2 weken omdat Mesa behoorlijk groot is en omdat er best wel veel coding nodig is om de simulatie werkend te krijgen. **Punten** : 1

1.3.3.2 NetLogo

Suitability

Tool support: De user story kan gemaakt worden in NetLogo, de tool is geschikt voor simulaties. We hebben een simulatie in NetLogo gezien die lijkt op de simulatie die wij willen maken. **Punten:** 5

Performance efficiency: NetLogo heeft een ‘simulation-speed’ slider waarmee de snelheid van de simulatie aangepast kan worden. Hiermee is het mogelijk om de simulatie heel snel te laten runnen. **Punten:** 4

Compatibility: Voor onze simulatie maken we geen gebruik van externe data maar gebruiken we slechts alle tools in NetLogo om zelf data te creëren. NetLogo heeft wel een GIS extensie die bepaalde data kan inladen, mocht je het nodig hebben.

Punten: 3

Feasibility

User-friendly: Het gebruiksvriendelijke aan NetLogo is dat het al een ‘Models Library’ bevat waardoor gebruikers al gemaakte simulaties kunnen inladen en uitvoeren. Verder zijn er drie verschillende kolommen; interface waarin de simulatie zichtbaar is, waarin verdere informatie staat met uitleg en code waarin de gebruiker kan zien hoe de simulatie is opgezet en zelf nog aanpassingen kan doorvoeren.

Punten: 4

Technically feasible: Door alle elementen die onder het kopje *User-friendly* zijn genoemd is het mogelijk om binnen de deadline een eigen simulatie te maken. Dit omdat het gemakkelijk is om de tool binnen de deadline te begrijpen. **Punten** : 4

1.3.3.3 Unity

Suitability

Tool support: Unity is per definitie geschikt voor de user story, een baan met een object (auto) erop is te modelleren in het programma. Hier moet wel rekening gehouden worden met de soms lastige omgeving van Unity, aangezien deze meer geschikt is voor games dan simulaties. **Punten:** 4

Performance efficiency: Unity is in principe snel genoeg om simulaties uit te voeren. Misschien kan het performance-wise lastig zijn voor Unity3D om met veel Agents te werken als daar niet specifiek op geprogrammeerd wordt. **Punten:** 3

Compatibility: Unity heeft C# als achterliggende programmeertaal. Met deze programmeertaal zijn veel libraries en mogelijkheden om externe data te gebruiken.

Punten: 4

Feasibility

User-friendly: Bij het opstarten van Unity is de interface best overweldigend, er zijn veel mogelijkheden en dingen die aangepast kunnen worden. Dit kan het moeilijk maken om snel een werkend model te maken. **Punten:** 2

Technically feasible: Doordat Unity erg uitgebreid is, kan het lastig worden om deze tool binnen twee weken te leren en te gebruiken om een goede simulatie mee te maken. **Punten:** 2

1.3.4 Score toolkeuze

	Mesa	Unity	NetLogo
Suitability	9	11	12
Feasibility	2	4	8
Totaal punten	11/25	15/25	20/25

Door voor elk onderdeel punten toe te kennen en deze bij elkaar op te tellen zijn we tot de conclusie gekomen dat NetLogo met 20 van de 25 punten de voorkeur heeft.