

Verkeerssimulatie



Namen: Storm Joannes, Charlie Choffat, Brandon Betz

Klas: V2B

Einddatum: 17 dec 2020

Inhoudsopgave:

Abstract	3
Onderwerp	3
Onderzoeksvragen	4
Vooronderzoek	4
Plan van aanpak	5
Experiment design	8
Resultaten experiment	8
Conclusie	8
Discussie	8
Bronnen	8

Abstract

Onderwerp

Het onderwerp van dit verslag is het simuleren van een autoweg. Deze simulatie moet een ABMS zijn. Dat betekent agent-based modelling and simulation. Deze simulatie moet zo gemaakt worden dat we met behulp van de simulatie beide onderzoeksvragen kunnen beantwoorden. We waren begonnen met het maken van één rijstrook waar de auto's van rechts naar links bewegen en stopte wanneer het in de buurt kwam van een andere auto. Deze code hebben we met netlogo geschreven. Netlogo is een tool waarmee je makkelijk dingen kan visualiseren. We moeten ervoor zorgen dat auto's bepaalde acties kunnen uitvoeren, zoals versnellen en vertragen. Als basiscode hebben we de Nagel-Schreckenberg model gebruikt. Deze code moet natuurlijk worden uitgebreid zodat we nog bepaalde omgevingsveranderingen kunnen maken om het te laten werken met de onderzoeksvragen.

Onderzoeksvragen

Onderzoeksvraag 1:

Wat is het effect van de maximumsnelheid van de weg op de doorstroom van de weg?

Hypothese 1:

Een hogere maximumsnelheid van de weg zal leiden tot een betere doorstroming over de weg omdat auto's sneller de weg op komen en de weg af gaan.

Onderzoeksvraag 2:

Welke invloed heeft een kruispunt met stoplichten en een kruispunt zonder stoplichten (gelijkwaardig kruispunt) op de doorstroming, en hangt dit ook af van de verkeersdichtheid?

Hypothese 2:

Een stoplicht zal succesvoller zijn voor een goede doorstroom bij een hoge verkeersdruk. Een gelijkwaardig kruispunt zal succesvoller zijn bij een lage verkeersdruk.

Vooronderzoek

Om onze simulatie zo goed mogelijk te maken gaan we eerst kijken naar andere simulaties. Hiermee kunnen we zien hoe zij bepaalde gebeurtenissen simuleren en waarom ze het op die specifieke manier doen.

Traffic grid (netlogo):

Netlogo heeft zelf meerdere voorbeeld simulaties waar gebruikers naar kunnen kijken voor inspiratie of waar ze zelf aanpassingen op kunnen maken. De traffic grid is zo een simulatie. Deze simulatie heeft meerdere kruispunten waar constant stoplichten wisselen van kleur. Vervolgens tonen ze de gemiddelde wachttijd, snelheid en de hoeveelheid auto's die stoppen. Wat bij deze simulatie interessant is is hoe ze werken met de stoplichten. Er wordt door de agents (de auto's) gekeken naar of het vlak voor hun rood is en als dat zo is stoppen ze. Als het vak vervolgens groen is mogen ze door. Het is heel simpel, maar het constant switchen van stoplichten en de auto's hier constant naar laten kijken zodat ze niet per ongeluk door rood rijden zou als je het niet goed aanpakt problemen opleveren.

Deze simulatie geeft een al een goed idee van hoe we aan de slag gaan. We maken een simpele kruispunt waar 4 stoplichten op komen waar auto's op moeten letten om te kunnen rijden. We kunnen vervolgens met hetzelfde kruispunt een gelijkwaardig kruispunt maken waar auto's dus niet op de stoplichten letten. We konden alleen

helaas geen project/simulatie hierover vinden dus de verkeersregels hierbij toepassen wordt nog interessant.

Plan van aanpak

Plan van aanpak:

- **GUI:**
We zullen meerdere variabelen aanpasbaar maken door middel van bijvoorbeeld sliders (zoals de acceleratie, deceleratie, aantal auto's en bepaalde specs van de agents). Dit maakt het makkelijk om verschillende simulatie te runnen. Uiteraard zal de GUI buttons bevatten om de simulatie te runnen en ongeveer de helft van het beeld zal de visualisatie van de simulatie zijn.
- **Tijd:**
De tijd kunnen we updaten in stappen of ticks en dan ervoor zorgen dat bijvoorbeeld bij elke tick bepaalde dingen gecontroleerd worden (zoals snelheid en afstand met andere agents) en op basis daarvan de actie bepalen. Zo blijft de simulatie constant geupdate. Als er niet zoiets is als een tick of stap in de tool kunnen we werken met een timer. Die bijvoorbeeld om elke seconde alles runt. We hopen de tijd symmetrisch te maken, maar dat is lastig met netlogo. Netlogo zegt zelf dat het niet mogelijk is, maar er zijn voorbeeldcodes waar dit wel gebeurt.
- **Staat:**
We kunnen kijken wat de staat van bepaalde waardes zijn. zoals de verkeersdichtheid (hoeveel auto's op de weg), gemiddelde snelheid op de weg, en hoeveel auto's bij een bepaald punt zijn gekomen. Dit kan bij bepaalde tools makkelijk weergegeven worden met bijvoorbeeld monitors. Bij sommige tools is het moeilijk om met behulp van een kliksysteem de huidige staat van een agent te tonen dus of we dat gaan doen is nog niet zeker.
- **Resultaat:**
We kunnen de resultaten van de simulaties overzetten in apart bestand. In dit bestand komen meerdere waardes (zoals de gemiddelde snelheid, enz) en met behulp van dit kunnen we de resultaten over meerdere simulaties analyseren om onze onderzoeksvragen te beantwoorden.
- **Modules:**
Een snelweg hebben we nodig om te testen wat de doorstroming doet op een

weg waar je 100 km/h mag in vergelijking met een normale weg waar je 50 km/h mag.

De gewone weg hebben we ook nodig om te meten wat stoplichten en kruispunten doen bij verschillende verkeers dichtheden, en welke de voorkeur zou hebben wanneer.

Welke tool gaan we gebruiken?

Voor en nadelen elke tool:

Netlogo:

- Voordelen:
 - Agents zijn makkelijk te maken
 - Syntax is makkelijk
 - Omgeving kan gemaakt worden met patches
 - Tijd kan goed bijgehouden worden
 - Snel
- Nadelen:
 - Moeilijk om de speciale situaties (rotondes, verkeersregels, in en uitvoegstrook) in te simuleren
 - Weinig ervaring met externe data
 - Omgeving moet niet te ingewikkeld zijn

Mesa:

- Voordelen:
 - Agents zijn makkelijk te maken
 - Omgeving is makkelijk te maken
 - Goed om data te collecten
 - Maakt gebruik van Python
- Nadelen:
 - Lastig om een weg te visualiseren

Unity:

- Voordelen:
 - duidelijke omgeving te visualiseren.
 - makkelijk om te gaan met speciale situaties, bijvoorbeeld een rotonde visualiseren.
- Nadelen:
 - lastig in gebruik.
 - wil je code werkende krijgen en testen moet je ook de hele visualisatie af hebben.
 - Meeste hebben nog niet met C# en unity gewerkt.

SF(A):

De ratings zijn 1 t/m 10 (1 laagste , 10 hoogste)

Suitability, feasibility en Modules zijn de totale scores van de onderwerpen eronder.

	Netlogo	Mesa	Unity
Suitability	32	32	33
Modules	20	18	21
- Agents	8	7	6
- Omgeving	7	7	8
- Speciale situatie	5	5	7
Efficiëntie	7	6	5
Compatibiliteit	5	9	7
Feasibility	17	15	10
Omgaan met tool?	9	7	5
Technisch haalbaar	8	8	5
Totaal	49	47	43

Kijken naar de voor en nadelen en naar het SFA zijn we op de conclusie gekomen om netlogo te gebruiken. De syntax is makkelijk te leren, snel en niet al te ingewikkeld. Het enige grote nadeel is dat het simuleren van speciale situaties een probleem kan zijn.

Experiment design

Ons design zal voor onderzoeksvraag één anders zijn dan bij de tweede. Bij onderzoeksvraag 1 is het de bedoeling dat we 2 wegen boven elkaar hebben. Bij weg 1 zullen we een maximum snelheid hebben van 50 km/h, en bij weg 2 een maximumsnelheid van 100 km/h. Hier kunnen wij vervolgens een grafiek op toepassen en zien welke weg de beste doorstroom heeft.

Voor onderzoeksvraag 2 maken wij een kruispunt van 4 wegen met stoplichten, en een zonder stoplichten. Deze twee zetten wij naast elkaar en testen we met verschillende verkeers dichtheden. Hier kunnen wij ook weer een grafiek op zetten zodat je kan zien bij welke verkeersdichtheid welke beter is.

Voor beiden onderzoeksvragen geldt, je laat beide scenario's met dezelfde verkeersdichtheid starten. De simulatie zal oneindig doorgaan totdat de gebruiker het zelf stopt.

Om verschillende resultaten te krijgen zullen we voor bepaalde waardes sliders maken zodat deze makkelijk aangepast kunnen worden. De code zal zo reproduceerbaar mogelijk gemaakt worden door middel van functies en parameters zodat andere mensen er ook gewoon mee zouden kunnen werken zonder 100% van alle code te snappen.

Resultaten experiment

Conclusie

Discussie

Bronnen