

### onderzoeksvragen:

Wat zal beter zijn voor de doorstroom. Stoplichten of een rotonde, en is bij een andere verkeersdichtheid de een misschien weer beter dan de ander.

### hypothese:

Waarschijnlijk zal een stoplicht een betere doorstroom hebben bij een hogere verkeersdichtheid en een rotonde bij een lagere verkeersdichtheid

### Plan van aanpak:

Voor onze simulatie hebben we een mindmap gemaakt:



Als we verschillende dingen van deze mindmap combineren zouden wij de hoofdvraag en onze eigen onderzoeksvraag kunnen beantwoorden.

### Basis simulatie:

De basis simulatie zal bestaan uit een 2D weg met auto's. De auto's hebben een getal boven hun wat staat voor de snelheid van de auto. De lengte van de cel is de som van de gemiddelde lengte van een voertuig en de afstand tussen 2 voertuigen, wat 4.5 meter is. De typische reactietijd van een weggebruiker is 1 seconde en dat is dan ook de duur van een stap. De snelheid per stap is 7.5 meter per seconde wat uitkomt op 27 km/u. De maximumsnelheid is 5 cellen per stap wat neerkomt op 135 km/u.

Wij verplaatsen alle agents op een asynchrone manier. Dat wilt zeggen dat je elke agents apart beweegt. Hier hebben wij onder andere voor gekozen omdat netlogo het niet support om alle agents tegelijkertijd te bewegen. De volgorde die wij hanteren om de agents te verplaatsen is van links naar rechts. Dit hanteren wij omdat de auto's van rechts naar links rijden. Als je in het verkeer kijkt dan zie je ook dat een file ontstaat van voor naar achter, hierdoor leek het onlogisch om de volgorde van achter naar voren te zetten.

## Eind simulatie:

- **GUI:**

We zullen meerdere variabelen aanpasbaar maken door middel van bijvoorbeeld sliders (zoals de max snelheid, aantal auto's en vrachtwagens en bepaalde specs van de agents) . Dit maakt het makkelijk om verschillende simulatie te runnen. Of we dit echt gaan doen door middel van een heftig GUI of door gewoon simpele inputs en sliders is nog de vraag.

- **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 klijksysteem 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 tijd, enz.) en met behulp van dit kunnen we de resultaten over meerdere simulaties analyseren om onze onderzoeksvragen te beantwoorden.

- **Modules:**

Een auto en vrachtwagen hebben wij nodig om te kijken of vrachtwagens de doorstroom hinderen.

Een snelweg hebben we nodig om te kijken of in/uit -voegstroken invloed hebben op de doorstroming van het verkeer op een snelweg. Ook hebben we de snelweg nodig om te testen wat de doorstroming doet op een snelweg 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.

## **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, 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
<b>Suitability</b>	<b>32</b>	<b>32</b>	<b>33</b>
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
<b>Feasibility</b>	<b>17</b>	<b>15</b>	<b>10</b>
Omgaan met tool?	9	7	5
Technisch haalbaar	8	8	5
<b>Totaal</b>	<b>49</b>	<b>47</b>	<b>43</b>

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.