

Verkeerssimulatie (v1.0)

1 Context

Voor de regeerperiode 2024-2029 heeft de Vlaamse regering de ambitie om het aantal verkeersongevallen op het Vlaamse wegennet tot nul te herleiden. Om meer inzicht te krijgen in de werking van het verkeer, heeft het Departement Mobiliteit en Openbare Werken geopteerd om een simulatiemodel te laten ontwikkelen dat het verkeer kan simuleren.

De Universiteit Antwerpen is gevraagd dit systeem te ontwikkelen. In de eerste bachelor informatica zal onder de vakken “Computer Graphics” en “Project Software Engineering” gewerkt worden aan dit project. Tijdens de practica Computer Graphics zal de visualisatie van de simulatie ontwikkeld worden, tijdens de practica Project Software Engineering zal gewerkt worden aan de simulatie applicatie zelf.

2 Legende

De behoeftespecificatie is opgesteld aan de hand van zogenaamde use-cases. Elke use-case beschrijft een klein gedeelte van de gewenste functionaliteit. Het is de bedoeling dat tijdens elke fase van het project verschillende van die use cases geïmplementeerd worden. Een typische use-case bevat de volgende onderdelen:

- **Refertenummer & titel:**

Wordt gebruikt om naar een bepaalde use-case te verwijzen.

- **Prioriteit:**

De specificatie van een systeem vraagt meer dan wat binnen de voorziene tijd op te leveren is. Vandaar dat we per use-case aangeven in hoeverre die functionaliteit belangrijk is. In volgorde van belangrijkheid kan hier staan: VERPLICHT (deze use-case moet opgeleverd worden), BELANGRIJK (niet essentieel maar bij voorkeur toch opleveren), NUTTIG (interessant maar kan weggelaten worden).

- **Doel:**

Summiere beschrijving van het waarom van de use-case, t.t.z. wat de use-case bijdraagt tot de gehele functionaliteit.

- **Preconditie:**

Summiere beschrijving van de uitgangspunten bij aanvang van de use-case.

- **Succesvol einde:**

Summiere beschrijving van wat opgeleverd zal worden als er niks fout is gegaan.

- **Stappen:**

Een sequentiële beschrijving van hoe de use-case precies zal verlopen als alles goed gaat (het zogenaamde "happy day scenario"). De stappen zijn genummerd en kunnen controle instructies (WHILE, IF, ...) bevatten.

- **Uitzonderingen:**

Een lijst van mogelijke probleemgevallen en hoe die behandeld zullen worden. Een probleem geval (a) verwijst naar het nummer van de stap waar het probleem kan optreden, (b) bevat een conditie die aangeeft wanneer het probleemgeval optreedt, (c) omschrijft heel kort (een lijn) hoe het probleem behandeld zal worden.

- **Voorbeeld:**

Een voorbeeld van wat in- of uitgevoerd kan worden.

Soms is een use-case een uitbreiding van een andere use-case, en dan zijn volgende onderdelen relevant:

- **Uitbreiding:**

Een referte naar de use-case waarvan deze een uitbreiding is.

- **Stappen:**

Een lijst van extra en/of aangepaste stappen t.o.v de use-case waarvan deze een uitbreiding is.

Een uitbreiding (a) verwijst naar het nummer van de stap die uitgebreid wordt, (b) zegt of de uitbreiding voor, na of tijdens de normale stap zal gebeuren, (c) omschrijft wat precies in de uitbreiding zal gebeuren.

3 Overzicht

Use-Case	Prioriteit
<i>1: Invoer</i>	
1.1. Verkeerssituatie inlezen	VERPLICHT
1.2. Voertuiggenerator inlezen	BELANGRIJK
<i>2: Uitvoer</i>	
2.1. Simpele uitvoer	VERPLICHT
<i>3: Simulatie</i>	
3.1. Rijden van voertuig	VERPLICHT
3.2. Simulatie van verkeerslicht	VERPLICHT
3.3. Automatische simulatie	VERPLICHT
3.4. Simulatie met voertuiggenerator	BELANGRIJK

1.1. Verkeerssituatie inlezen

Prioriteit:

VERPLICHT

Doel:

Inlezen van het schema van de verkeerssituatie: de verschillende wegen en de verschillende voertuigen.

Preconditie:

Een ASCII bestand met daarop een beschrijving van de wegen en voertuigen. (Zie Appendix A voor meer informatie over het XML formaat)

Succesvol einde:

Het systeem bevat een schema met de verschillende wegen, en informatie over alle voertuigen.

Stappen:

1. Open invoerbestand
2. WHILE Bestand niet ingelezen
 - 2.1. Herken het soort element (VOERTUIG, BAAN, VERKEERSLICHT)
 - 2.2. Lees verdere informatie voor het element
 - 2.3. IF Verifieer geldige informatie
 - 2.3.1. THEN Voeg element toe aan de simulatie
 - 2.3.1. ELSE Foutboodschap + positioneer op volgende element in het bestand
3. Verifieer consistentie van de verkeerssituatie
4. Sluit invoerbestand

Uitzonderingen:

- 2.1. [Onherkenbaar element] Foutboodschap + positioneer op volgende element in het bestand \Rightarrow verdergaan vanaf stap 2
- 2.2. [Ongeldige informatie] Foutboodschap + positioneer op volgende element in het bestand \Rightarrow verdergaan vanaf stap 2
3. [Inconsistente verkeerssituatie] Foutboodschap \Rightarrow verdergaan vanaf stap 4

Voorbeeld:

Een baan met twee auto's en één verkeerslicht:

```
<BAAN>
  <naam>Middelheimlaan</naam>
  <lengte>500</lengte>
</BAAN>
```

```
<VERKEERSLICHT>
  <baan>Middelheimlaan</baan>
  <positie>400</positie>
  <cyclus>20</cyclus>
</VERKEERSLICHT>
```

```
<VOERTUIG>
  <baan>Middelheimlaan</baan>
  <positie>20</positie>
</VOERTUIG>
```

```
<VOERTUIG>
  <baan>Middelheimlaan</baan>
  <positie>0</positie>
</VOERTUIG>
```

1.2. Voertuiggenerator inlezen

Prioriteit:

BELANGRIJK

Doel:

Inlezen van hoe voertuigen gegenereerd moeten worden.

Preconditie:

Een ASCII bestand met daarop een beschrijving van een voertuiggenerator. (Zie Appendix A voor meer informatie over het XML formaat)

Succesvol einde:

Het systeem bevat een schema met een voertuiggenerator.

Uitbreiding:

Use Case 1.1

Stappen:

[2.1, tijdens] Hou rekening met extra element VOERTUIGGENERATOR

Uitzonderingen:

Geen

Voorbeeld:

```
<VOERTUIGGENERATOR>
  <baan>Middelheimlaan</baan>
  <frequentie>5</frequentie>
</VOERTUIGGENERATOR>
```

2.1. Simpele uitvoer

Prioriteit:

VERPLICHT

Doel:

Uitvoer van alle informatie in de simulatie.

Preconditie:

Het systeem bevat een schema van de virtuele verkeerssituatie.

Succesvol einde:

Het systeem heeft de informatie van de virtuele verkeerssituatie netjes uitgeschreven.

Stappen:

1. Schrijf huidige simulatietijd uit
2. WHILE Nog voertuigen in simulatie
 - 2.1. Schrijf voertuig-gegevens uit

Uitzonderingen:

Geen

Voorbeeld:

Tijd: 0

Voertuig 1

-> baan: Middelheimlaan

-> positie: 20

-> snelheid: 16.6

Voertuig 2

-> baan: Middelheimlaan

-> positie: 0

-> snelheid: 16.6

3.1. Rijden van voertuig

Prioriteit:

VERPLICHT

Doel:

Simuleren van het rijden van een voertuig. Zie Appendix B voor meer informatie over de formules.

Preconditie:

Het systeem bevat een schema van de virtuele verkeerssituatie. Er is een voertuig op een baan.

Succesvol einde:

De positie, snelheid, en versnelling van het voertuig zijn herberekend.

Stappen:

1. Bereken nieuwe snelheid en positie van voertuig
2. Bereken nieuwe versnelling van voertuig
4. IF nieuwe positie valt buiten huidige baan
- 4.1. Verwijder voertuig uit simulatie

Uitzonderingen:

Geen

3.2. Simulatie van verkeerslicht

Prioriteit:

VERPLICHT

Doel:

Simuleren van verkeerslichten.

Preconditie:

Het systeem bevat een schema van de virtuele verkeerssituatie. Er is een verkeerslicht op een baan.

Succesvol einde:

Voertuigen passen zich aan afhankelijk van de staat van het verkeerslicht.

Stappen:

1. IF tijd sinds laatste verandering > cyclus
 - 1.1 THEN verander de kleur van het licht (groen \leftrightarrow rood)
2. IF verkeerslicht is groen
 - 2.1 THEN voertuigen voor het verkeerslicht mag terug versnellen
- 3.1 IF verkeerslicht is rood
 - 3.1.1 THEN IF het eerste voertuig voor het licht bevindt zich in de vertraagafstand
 - 3.1.1.1 THEN pas de vertraagfactor toe op het voertuig
 - 3.1.1.2 ELSE IF het eerste voertuig voor het licht bevindt zich in de eerste helft van de stopafstand
 - 3.1.1.1.1 THEN laat het voertuig stoppen

3.3. Automatische simulatie

Prioriteit:

VERPLICHT

Doel:

Simulatie automatisch laten lopen.

Preconditie:

Het systeem bevat een schema van de virtuele verkeerssituatie.

Succesvol einde:

Het verkeer in het wegennetwerk wordt gesimuleerd.

Stappen:

1. FOR elk voertuig in het wegennetwerk
 - 1.1 voer use case 3.1 uit op het voertuig
2. FOR elk verkeerslicht in het wegennetwerk
 - 2.1 voer use case 3.2 uit op het verkeerslicht

3.4. Simulatie met voertuiggenerator

Prioriteit:

BELANGRIJK

Doel:

Genereren van voertuigen.

Preconditie:

Het systeem bevat een schema van de virtuele verkeerssituatie.

Er is een voertuiggenerator op een baan.

Succesvol einde:

Er worden automatisch voertuigen toegevoegd tijdens de simulatie.

Uitbreiding:

Use Case 3.3

Stappen:

3. FOR elke voertuiggenerator

3.1 IF tijd sinds laatste voertuig > frequentie

3.1.1 IF geen voertuig op baan tussen posities 0 en $2l$

3.1.1.1 THEN voeg voertuig toe aan baan op positie 0

A Invoer formaat

Het invoerformaat voor de virtuele verkeerssituatie is zodanig gekozen dat nieuwe attributen en elementen makkelijk kunnen worden toegevoegd.

```
Verkeerssimulatie = { Element }
Element = "<" ElementType ">" AttributeList "</" ElementType ">"
ElementType = "VOERTUIG" | "BAAN" | "VERKEERSLICHT" | "VOERTUIGGENERATOR"
AttributeList = Attribute { Attribute }
Attribute = "<" AttributeType ">" AttributeValue "</" AttributeType ">"
AttributeType = "naam" | "lengte" | "baan" | "positie"
               | "cyclus" | "frequentie"
AttributeValue = Integer | String
Integer = Digit { Digit }
Digit = "0" ... "9"
String = Character { Character }
Character = "a" ... "z" | "A" ... "Z" | Digit
```

Merk op dat de attribootlijst een relatief vrij formaat heeft wat sterk zal afhangen van het soort element dat gedefinieerd wordt. De volgende tabel toont de verschillende attributen voor elk element:

Element	Attribuut
BAAN	naam, lengte
VERKEERSLICHT	baan, positie, cyclus
VOERTUIG	baan, positie
VOERTUIGGENERATOR	baan, frequentie

Bovendien zal afhankelijk van het attribuuttype slechts een bepaalde attribuutwaarde toegelaten zijn:

Attribuut	Waarde
naam, baan	String
lengte, positie, cyclus, frequentie	Integer

Bovendien moet de openings tag steeds overeenkomen met de sluitingsdag. Vandaar dat tijdens de invoer moet gecontroleerd worden of de invoer al dan niet geldig is.

Het bestand met de in te lezen verkeerssituatie wordt met de hand geschreven. Om de ingelezen verkeerssituatie te kunnen simuleren moet de informatie consistent zijn.

Een verkeerssituatie is consistent als:

- Elk voertuig staat op een bestaande baan.
- Elk verkeerslicht staat op een bestaande baan.
- Elke voertuiggenerator staat op een bestaande baan.
- De positie van elk voertuig is kleiner dan de lengte van de baan.
- De positie van elk verkeerslicht is kleiner dan de lengte van de baan.
- Er is maximaal één voertuiggenerator op elke baan.
- Een verkeerslicht mag zich niet in de vertraagafstand van een ander verkeerslicht bevinden (zie Appendix B).

Opmerkingen:

- Lengte, positie, cyclus, en frequentie moet altijd positief zijn.
- De naam wordt gebruikt als unieke identificatie van een baan.

B Simulatiemodel

B.1 Variabelen

Onderstaande tabel bevat een overzicht van de variabelen die gebruikt worden in het simulatiemodel.

Afkorting	Naam	Betekenis
l	lengte	Lengte van een voertuig.
x	positie	Huidige positie van een voertuig, gemeten vanaf de voorste bumper.
v	snelheid	Huidige snelheid van een voertuig.
v_{max}	maximale snelheid	De gewenste maximale snelheid van een voertuig. Kan veranderen door externe factoren (zoals een verkeerslicht).
V_{max}	maximale snelheid	De absolute maximale snelheid dat een voertuig kan rijden.
a	versnelling	Huidige versnelling van een voertuig.
a_{max}	maximale versnelling	De maximale versnelling van een voertuig.
b_{max}	maximale remfactor	De maximale remfactor van een voertuig.
f_{min}	minimale volgafstand	De minimale gewenste volgafstand van een voertuig.
Δt	simulatietijd	De tijd tussen twee stappen van de simulatie.
Δx_s	vertraagafstand	De afstand tot een verkeerslicht waarin een voertuig moet vertragen, inclusief de stopafstand.
Δx_{s0}	stopafstand	De afstand tot een verkeerslicht waarin een voertuig moet stoppen.
s	vertraagfactor	Factor waarmee een voertuig moet vertragen voor een verkeerslicht.

B.2 Formules positie en snelheid

Voor het aanpassen van de positie en snelheid van een voertuig maken we onderscheid tussen twee situaties:

- Indien $v + a\Delta t$ kleiner is dan nul, dan zou de snelheid negatief worden. Dit is niet toegestaan in ons model. In dit geval passen we de positie aan als volgt:

$$x = x - \frac{v^2}{2a}$$

Vervolgens zetten we de snelheid gelijk aan nul.

- Indien dit niet het geval is, dan passen we eerst de snelheid aan:

$$v = v + a\Delta t$$

Vervolgens passen we de positie aan:

$$x = x + v\Delta t + a\frac{\Delta t^2}{2}$$

B.3 Formules versnelling

Voor het aanpassen van de versnelling van een voertuig i berekenen we eerst δ , de interactieterm met voertuig $i - 1$ (het voertuig dat voor voertuig i rijdt).

We berekenen eerst de volgafstand Δx :

$$\Delta x = x_{i-1} - x_i - l_{i-1}$$

We berekenen dan het snelheidsverschil Δv :

$$\Delta v = v_i - v_{i-1}$$

Hiermee berekenen we δ :

$$\delta = \frac{f_{min} + \max(0, v + \frac{v\Delta v}{2\sqrt{a_{max}b_{max}}})}{\Delta x}$$

Indien er geen voertuig voor rijdt, en dus voertuig i het eerste voertuig op de baan is, dan zetten we δ gelijk aan nul.

We kunnen dan de versnelling berekenen als volgt:

$$a = a_{max}(1 - \left(\frac{v}{v_{max}}\right)^4 - \delta^2)$$

B.4 Vertragen en versnellen

Om het eerste voertuig op de baan te laten vertragen, dan zetten we v_{max} gelijk aan sV_{max} , met s de vertragsingsfactor. Voertuigen achter dit voertuig zullen automatisch ook vertragen.

Indien het eerste voertuig op de baan terug mag versnellen, dan zetten we v_{max} gelijk op V_{max} . Voertuigen achter dit voertuig zullen automatisch ook versnellen.

B.5 Stoppen

Indien het eerste voertuig op de baan tot stilstand moet komen, dan passen we elke simulatiestap de versnelling aan als volgt.

$$a = -\frac{b_{max}v}{v_{max}}$$

Voertuigen achter dit voertuig zullen automatisch ook tot stilstand komen.

B.6 Standaard waarden

Onderstaande tabel bevat een overzicht van de standaard waarden die gebruikt worden in het simulatiemodel.

Afkorting	Naam	Waarde
l	lengte	4
V_{max}	maximale snelheid	16.6
a_{max}	maximale versnelling	1.44
b_{max}	maximale remfactor	4.61
f_{min}	minimale volgafstand	4
Δt	simulatietijd	0.0166
Δx_s	vertraagafstand	50
Δx_{s0}	stopafstand	15
s	vertraagfactor	0.4