

Verkeerssimulatie

Documentsoort:	Behoeftespecificatie
Versie:	1.0
Datum:	21 februari 2019
Auteurs:	Brent van Bladel
Status:	In development

1 Samenvatting

Dit document bevat de specificaties voor een informaticasysteem ter ondersteuning van een verkeerssimulatie. Het is geschreven in het kader van het vak “Project Software Engineering” (1ste bachelor informatica - Universiteit Antwerpen).

2 Context

Op 25 juni 2018 heeft het politiek stuurcomité van Antwerpen 18 projecten geselecteerd die voor een overkapping van de Antwerpse ring zullen zorgen. De eerste werken beginnen in de zomer van 2019 en zullen een hoop verkeershinder met zich mee brengen. Om deze hinder zo goed mogelijk in te perken, 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.

3 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.

4 Overzicht

Use-Case	Prioriteit
<i>1: Invoer</i>	
1.1. Wegen en voertuigen inlezen	VERPLICHT
1.2. Wegennetwerk inlezen	BELANGRIJK
<i>2: Uitvoer</i>	
2.1. Simpele uitvoer	VERPLICHT
<i>3: Simulatie</i>	
3.1. Rijden van voertuigen	VERPLICHT
3.2. Automatische simulatie	VERPLICHT

1.1. Wegen en voertuigen 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)
 - 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 die stilstaand vertrekken 10 meter van elkaar:

```
<BAAN>
  <naam>E19</naam>
  <sneldheidslimiet>100</sneldheidslimiet>
  <lengte>2000</lengte>
</BAAN>
```

```
<VOERTUIG>
  <type>AUTO</type>
  <nummerplaat>1THK180</nummerplaat>
  <baan>E19</baan>
  <positie>10</positie>
  <sneldheid>0<sneldheid>
</VOERTUIG>
```

```
<VOERTUIG>
  <type>AUTO</type>
  <nummerplaat>651BUF</nummerplaat>
  <baan>E19</baan>
  <positie>0</positie>
  <sneldheid>0<sneldheid>
</VOERTUIG>
```

1.2. Wegennetwerk inlezen

Prioriteit:

BELANGRIJK

Doel:

Inlezen van hoe de verschillende wegen met elkaar verbonden zijn.

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, hoe deze verbonden zijn, en informatie over alle voertuigen.

Uitbreiding:

Use Case 1.1

Stappen:

[2.2, tijdens] Hou rekening met extra attributen tijdens het parsen van 'BAAN' elementen

Uitzonderingen:

Geen

Voorbeeld:

```
<BAAN>
  <naam>E19</naam>
  <snellheidslimiet>100</snellheidslimiet>
  <lengte>2000</lengte>
  <verbinding>E313</verbinding>
</BAAN>
```

```
<BAAN>
  <naam>E313</naam>
  <snellheidslimiet>120</snellheidslimiet>
  <lengte>5000</lengte>
</BAAN>
```

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 een tekstbestand (ASCII) uitgevoerd, waarin de informatie over de virtuele verkeerssituatie netjes is uitgeschreven.

Stappen:

1. Open uitvoerbestand
2. WHILE Nog banen beschikbaar
- 2.1. Schrijf baan-gegevens uit
3. WHILE Nog voertuigen beschikbaar
- 3.1. Schrijf voertuig-gegevens uit
4. Sluit uitvoerbestand

Uitzonderingen:

Geen

Voorbeeld:

Gegeven de input van 1.1

Baan: E19

-> snelheidslimiet: 100

-> lengte: 2000

Voertuig: auto (1THK180)

-> baan: E19

-> positie: 10

-> snelheid: 0

Voertuig: auto (651BUF)

-> baan: E19

-> positie: 0

-> snelheid: 0

3.1. Rijden van voertuigen

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:

Het voertuig heeft een nieuwe positie.

Stappen:

1. Bereken nieuwe positie van voertuig
2. Bereken nieuwe snelheid van voertuig
3. Bereken nieuwe versnelling van voertuig
4. IF nieuwe positie valt buiten huidige baan
 - 4.1. IF huidige baan heeft verbinding
 - 4.1.1 Zet voertuig op verbindingssbaan
 - 4.2. ELSE
 - 4.2.1. Verwijder voertuig uit simulatie

Uitzonderingen:

Geen

3.2. Automatische simulatie

Prioriteit:

VERPLICHT

Doel:

Simulatie automatisch laten lopen.

Preconditie:

Het systeem bevat een schema van de virtuele verkeerssituatie.

Succesvol einde:

Alle voertuigen hebben het wegennetwerk verlaten en de simulatie stopt.

Stappen:

1. WHILE voertuigen in het wegennetwerk
 - 1.1 FOR elk voertuig in het wegennetwerk
 - 1.1.1 voer use case 3.1 uit op het voertuig

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"
AttributeList = Attribute { Attribute }
Attribute = "<" AttributeType ">" AttributeValue "</" AttributeType ">"
AttributeType = "naam" | "snelheidslimiet" | "lengte" | "type"
                | "nummerplaat" | "baan" | "positie" | "snelheid"
                | "verbinding"
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 verplichte en optionele attributen voor elk element:

Element	Attribuut (verplicht)	Attribuut (optioneel)
BAAN	naam, snelheidslimiet, lengte	verbinding
VOERTUIG	type, nummerplaat, baan, positie, snelheid	-/-

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

Attribuut	Waarde
naam, type, nummerplaat, baan, verbinding	String
snelheidslimiet, lengte, positie, snelheid	Integer

Bovendien moet de openings tag steeds overeenkomen met de sluitingstag. 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.
- De positie van elk voertuig is kleiner dan de lengte van de baan.
- Er zijn geen 2 voertuigen op minder dan 5 meter van elkaar.
- Indien een baan een verbinding heeft, is deze verbinding een bestaande baan.

Opmerkingen:

- Snelheden worden in kilometer per uur genoteerd.
- Lengtes en posities worden in meter genoteerd.
- Lengtes en snelheden zijn altijd positief.
- Het enige type voertuig dat ondersteund wordt is 'AUTO'.
- De nummerplaat wordt gebruikt als unieke identificatie van een voertuig.
- De naam wordt gebruikt als unieke identificatie van een baan.

B Simulatie

B.1 Positie

De positie van een voertuig wordt berekend aan de hand van de snelheid van het voertuig. We gaan uit van een eenparige rechte lijnige beweging gedurende 1 stap van de simulatie. De positie p op stap i van de simulatie wordt dan gegeven door

$$p_i = v \cdot t + p_{i-1}$$

waarbij v de snelheid en t de duur van 1 simulatiestap. We veronderstellen dat de simulatie gebeurt in stappen van 1 seconde. Dit zorgt ervoor dat we in de formule $t = 1$ kunnen stellen. We krijgen

$$p_i = v + p_{i-1}$$

Merk op dat de positie hier uitgedrukt wordt in meter, en de snelheid in meter per seconde.

B.2 Snelheid

De snelheid van een voertuig wordt berekend aan de hand van de versnelling van het voertuig. De snelheid v op stap i van de simulatie wordt gegeven door

$$v_i = a \cdot t + v_{i-1}$$

waarbij a de versnelling en t de duur van 1 simulatiestap. We veronderstellen dat de simulatie gebeurt in stappen van 1 seconde. Dit zorgt ervoor dat we in de formule $t = 1$ kunnen stellen. We krijgen

$$v_i = a + v_{i-1}$$

Merk op dat de snelheid hier uitgedrukt wordt in meter per seconde, en de versnelling in meter per seconde kwadraat.

B.3 Versnelling

De versnelling van een voertuig wordt door zijn chauffeur bepaald op elke simulatiestap. We veronderstellen dat de chauffeur probeert om de ideale volgafstand te bekomen.

De ideale volgafstand in meter wordt gedefinieerd als 75% van de snelheid in kilometer per uur. Omdat we de positie meten als het voorste punt van het voertuig, moet de lengte van het voertuig hierbij opgeteld worden. Verder willen we ook een minimale afstand van 2 meter opleggen. We kunnen dan de ideale volgafstand berekenen als

$$\Delta d_{ideal} = \frac{3}{4}v + l_{prev} + 2$$

waarbij l de lengte van het voorgaande voertuig en v de snelheid van het voertuig in kilometer per uur.

De eigenlijke volgafstand in meter wordt berekend als

$$\Delta d_{actual} = p_{prev} - l_{prev} - p$$

waarbij l_{prev} de lengte van het voorgaande voertuig, p_{prev} de positie van het voorgaande voertuig in meter, en p de positie in meter.

De versnelling in meter per seconde kwadraat wordt elke simulatiestap berekend als

$$a = \frac{1}{2}(\Delta d_{actual} - \Delta d_{ideal})$$

Merk op dat, indien het voertuig de snelheidslimiet overschrijdt, de versnelling nooit groter dan nul mag zijn. Merk ook op dat, indien er geen voertuig voor rijdt, de maximale versnelling genomen wordt.

B.4 Grenswaarden

Om onze simulatie realistisch te houden, gaan we een aantal grenswaarden respecteren. De volgende tabel geeft een overzicht van deze grenswaarden:

Variabele	Minimale Waarde	Maximale Waarde
positie	0	lengte van baan
snelheid AUTO (km/u)	0	150
versnelling AUTO (m/s ²)	-8	2

Merk op dat de snelheid niet negatief mag worden. Een negatieve snelheid zou theoretisch wel mogelijk zijn indien het voertuig achteruit rijdt, maar dit is niet gewenst in onze simulatie. De versnelling kan wel negatief zijn: in dit geval remt het voertuig.

B.5 Lengte

De volgende tabel geeft een overzicht van de lengte van een voertuig:

Voertuig	Lengte (m)
AUTO	3