Metro Simulatie

Documentsoort:	Behoeftespecificatie
Versie:	2.0
Datum:	19 maart 2020
Auteurs:	Brent van Bladel
Status:	In development

1 Samenvatting

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

2 Context

Sinds 1 februari 2017 is de stad Antwerpen in de kernstad en op linkeroever een lage-emissiezone. In deze lage-emissiezone mogen enkel voertuigen rijden die aan bepaalde milieucriteria voldoen. Op 1 januari 2020 zijn deze criteria aangescherpt. Aangezien een heleboel voertuigen daarom niet langer de stad in mogen, verwacht de gewestelijke vervoersmaatschappij De Lijn een toename in het aantal passagiers. Het is voor De Lijn van groot belang op voorhand een duidelijk beeld te hebben van de situatie (qua bezetting en piekmomenten) van het metronet. Daarom heeft De Lijn geopteerd een simulatie model te laten ontwikkelen dat het tramverkeer 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**
1: Invoer	
1.1. Trams en stations inlezen	VERPLICHT
1.2. Trams en stations met type inlezen	VERPLICHT
1.3. Trams met voertuignummer inlezen	BELANGRIJK
1.4. Signalisatie inlezen	BELANGRIJK
1.5. Stations met meerdere sporen inlezen	NUTTIG
2: Uitvoer	
2.1. Simpele uitvoer	VERPLICHT
2.2. Grafische impressie	BELANGRIJK
2.3. Integratie met graphics	BELANGRIJK
3: Simulatie	
~~3.1. Rijden van trams~~	~~VERPLICHT~~
3.1. Rijden van trams (herzien)	VERPLICHT
3.2. Automatische simulatie	VERPLICHT
3.3. Rijden van trams met type	VERPLICHT
3.4. Botspreventie	BELANGRIJK
3.5. Realistische tijd	BELANGRIJK
3.6. Signalisatie	BELANGRIJK
3.7. Passagiers	NUTTIG
3.8. Omzet per tram	NUTTIG
3.9. Statistische verwerking simulatie	NUTTIG
4: Gebruikersinterface	
4.1. GUI voor de simulatie	NUTTIG
4.2. GUI voor het rijden van trams	NUTTIG
4.3. GUI voor statistische gegevens	NUTTIG

1.1. Trams en stations inlezen

Prioriteit:
VERPLICHT

Doel:
Inlezen van het schema van het metronet. De verschillende stations, hoe die met elkaar verbonden zijn en de verschillende trams.

Preconditie:
Een ASCII bestand met daarop een beschrijving van de stations en trams. (Zie Appendix A voor meer informatie over het XML formaat)

Succesvol einde:
Het systeem bevat een spoorschema met de verschillende stations, en informatie over alle trams.

Stappen:
1. Open invoerbestand
2. WHILE Bestand niet ingelezen
2.1. Herken het soort element (STATION, TRAM)
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 metronet
4. Sluit invoerbestand

Uitzonderingen:
2.1. [Onherkenbaar element] Foutboodschap + positioneer op volgende element in het bestand ⇒ verdergaan vanaf stap 2
2.2. [Ongeldige informatie] Foutboodschap + positioneer op volgende element in het bestand ⇒ verdergaan vanaf stap 2
3. [Inconsistente metronet] Foutboodschap ⇒ verdergaan vanaf stap 4

Voorbeeld:
Een metronet met drie stations (A,B,C), een spoor (12) en een tram (12).

```
<STATION>
    <naam>A</naam>
    <volgende>B</volgende>
    <vorige>C</vorige>
    <spoor>12</spoor>
</STATION>
<STATION>
    <naam>B</naam>
    <volgende>C</volgende>
    <vorige>A</vorige>
    <spoor>12</spoor>
</STATION>
<STATION>
    <naam>C</naam>
    <volgende>A</volgende>
    <vorige>B</vorige>
    <spoor>12</spoor>
</STATION>
<TRAM>
    <lijn>12</lijn>
    <zitplaatsen>32</zitplaatsen>
    <snelheid>60</snelheid>
    <beginStation>A</beginStation>
</TRAM>
```

1.2. Trams en stations met type inlezen

Prioriteit:
VERPLICHT

Doel:
Sinds mei 2015 heeft DeLijn een nieuw model van trams aangekocht: de Albatros. Het doel is om tegen 2018 alle oude PCC-trams te vervangen. Momenteel zitten we echter in een overgangsfase waar beide types tram tegelijk in gebruik zijn. Om tot een meer realistische simulatie te komen moet het mogelijk zijn onderscheid te maken tussen de twee types tram. De albatros tram heeft een lengte van 42 meter, wat aanzienlijk langer is dan de oude PCC-trams. Hierdoor kunnen ze niet stoppen aan bovengrondse haltes, aangezien het autoverkeer dan gehinderd wordt. Daarom moet het ook mogelijk zijn onderscheid te maken tussen twee types stations: het ondergrondse 'metrostation' en een bovengrondse 'halte'. Merk op dat de de `snelheid` en `zitplaatsen` attributen niet langer ondersteund moeten worden, aangezien deze kunnen worden afgeleid uit het type van de tram (Zie Appendix B).

Referenties:
`https://nl.wikipedia.org/wiki/Albatros_(tram)`
`https://nl.wikipedia.org/wiki/Presidents'_Conference_Committee-tram`
Uitbreiding:
Use Case 1.1

Stappen:
[2.2, tijdens] Hou rekening met extra attribuut "type"

Uitzonderingen:
Geen

Voorbeeld:
Voorbeeld van verschillende types. Opgepast: om het kort te houden is dit voorbeeld niet volledig: de stations van lijn 9 ontbreken.

```
<STATION>
    <naam>A</naam>
    <type>Metrostation</type>
    <volgende>B</volgende>
    <vorige>C</vorige>
    <spoor>15</spoor>
</STATION>
<STATION>
    <naam>B</naam>
    <type>Halte</type>
    <volgende>C</volgende>
    <vorige>A</vorige>
    <spoor>15</spoor>
</STATION>
<STATION>
    <naam>C</naam>
    <type>Metrostation</type>
    <volgende>A</volgende>
    <vorige>B</vorige>
    <spoor>15</spoor>
</STATION>
<TRAM>
    <lijnNr>9</lijnNr>
    <type>PCC</type>
    <beginStation>X</beginStation>
</TRAM>
<TRAM>
    <lijnNr>15</lijnNr>
    <type>Albatros</type>
    <beginStation>A</beginStation>
</TRAM>
```

1.3. Trams met voertuignummer inlezen

Prioriteit:
BELANGRIJK

Doel:
Om tot een meer realistische simulatie te komen moet het mogelijk zijn dat meerdere trams op een zelfde spoor voorkomen. Daarvoor moet elke tram voorzien zijn van een voertuignummer.

Uitbreiding:
Use Case 1.1

Stappen:
[2, tijdens] Hou rekening met extra attribuut voertuignummer

Uitzonderingen:
Geen

Voorbeeld:
Twee trams op lijn 12, gegeven de input van 1.1

```
<TRAM>
    <lijnNr>12</lijnNr>
    <type>PCC</type>
    <voertuigNr>1</voertuigNr>
    <beginStation>A</beginStation>
</TRAM>
<TRAM>
    <lijnNr>12</lijnNr>
    <type>Albatros</type>
    <voertuigNr>2</voertuigNr>
    <beginStation>B</beginStation>
</TRAM>
```

1.4. Signalisatie inlezen

Prioriteit:
BELANGRIJK

Doel:
Een spoor kan verschillende soorten signalisatie bevatten. Om dit in de simulatie te kunnen opnemen, moet deze data ook ingelezen worden.

Uitbreiding:
Use Case 1.1

Stappen:
[2, tijdens] Hou rekening met extra element "signaal" en de bijhorende attributen

Uitzonderingen:
Geen

Voorbeeld:
Voorbeeld van signalisatie, gegeven de input van 1.1.

```
<SIGNAAL>
    <type>STOP</type>
    <vorige>A</vorige>
    <volgende>B</volgende>
    <spoor>12</spoor>
</SIGNAAL>

<SIGNAAL>
    <type>SNELHEID</type>
    <limiet>30</limiet>
    <vorige>B</vorige>
    <volgende>C</volgende>
    <spoor>12</spoor>
</SIGNAAL>
```

1.5 Stations met meerdere sporen inlezen

Prioriteit:
NUTTIG

Doel:
Om tot een meer realistische simulatie te komen moet het mogelijk zijn dat een station meerdere sporen kan bevatten. Opgepast: enkel een metrostation kan meerdere sporen bevatten; een halte heeft maximaal één spoor (zie Use Case 1.2).

Uitbreiding:
Use Case 1.1

Stappen:
[2, tijdens] Hou rekening met andere structuur

Uitzonderingen:
Geen

Voorbeeld:
Drie stations met elk twee sporen, waarbij spoor 21 dezelfde route doet als spoor 12 in de omgekeerde richting. Merk op dat met deze uitbreiding van use case 1.1 het invoerformaat voor stations wijzigt.

```
<STATION>
    <naam>A</naam>
    <type>Metrostation</type>
    <SPOOR>
        <spoor>12</spoor>
        <volgende>B</volgende>
        <vorige>C</vorige>
    </SPOOR>
    <SPOOR>
        <spoor>21</spoor>
        <volgende>C</volgende>
        <vorige>B</vorige>
    </SPOOR>
</STATION>
<STATION>
    <naam>B</naam>
    <type>Metrostation</type>
    <SPOOR>
        <spoor>12</spoor>
        <volgende>C</volgende>
        <vorige>A</vorige>
    </SPOOR>
    <SPOOR>
        <spoor>21</spoor>
        <volgende>A</volgende>
        <vorige>C</vorige>
    </SPOOR>
</STATION>
<STATION>
    <naam>C</naam>
    <type>Metrostation</type>
    <SPOOR>
        <spoor>12</spoor>
        <volgende>A</volgende>
        <vorige>B</vorige>
    </SPOOR>
    <SPOOR>
        <spoor>21</spoor>
        <volgende>B</volgende>
        <vorige>A</vorige>
    </SPOOR>
</STATION>
```

2.1. Simpele uitvoer

Prioriteit:
VERPLICHT

Doel:
Uitvoer van alle informatie in de simulatie.

Preconditie:
Het systeem bevat een schema van de virtuele metronet.

Succesvol einde:
Het systeem heeft een tekstbestand (ASCII) uitgevoerd, waarin de informatie over het virtuele metronet netjes is uitgeschreven.

Stappen:
1. Open uitvoerbestand
2. WHILE Nog stations beschikbaar
2.1. Schrijf station-gegevens uit
3. WHILE Nog trams beschikbaar
3.1. Schrijf tram-gegevens uit
4. Sluit uitvoerbestand

Uitzonderingen:
Geen

Voorbeeld:
Gegeven de input van 1.1

```
Station A
<- Station C
-> Station B
Spoor 12
Station B
<- Station A
-> Station C
Spoor 12
Station C
<- Station B
-> Station A
Spoor 12

Tram 12 in Station A, 32 zitplaatsen
```

2.2. Grafische impressie

Prioriteit:
BELANGRIJK

Doel:
De toestand van het metronet wordt grafisch weergegeven.

Preconditie:
Het systeem is correct geïnitialiseerd.

Succesvol einde:
Het systeem heeft een tekstbestand (ASCII) uitgevoerd, waarin de toestand van het metronet staat beschreven.

Stappen:
1. Open uitvoerbestand
2. Teken gegevens uit voor de toestant van het metronet
3. Sluit uitvoerbestand

Uitzonderingen:
Geen
Voorbeeld:
Indien drie STATIONS (A, B, C) en twee trams (T):

```
=A===B===C=
 T   T
```

2.3. Integratie met Graphics

Prioriteit:
BELANGRIJK

Doel:
Om het stadsbestuur van de stad Antwerpen te overtuigen had onze klant graag een 3D vizualisatie gehad van het rondrijden van de trams. Hiervoor kan je een standaard interface voor je graphics engine gebruiken.

Preconditie:
Het systeem is correct geïnitialiseerd.

Succesvol einde:
Elke beweging van trams wordt weergegeven in een 3D omgeving.

3.1. Rijden van trams (herzien)

Prioriteit:
VERPLICHT

Doel:
Simuleren van een rondrijdende tram in het metronet.

Preconditie:
Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:
Een tram bevindt zich op een nieuwe locatie in het metronet. Het systeem heeft een boodschap afgedrukt met de details van de verplaatsing.

Stappen:
1. IF tram in station
1.1. Verwijder tram uit het huidige station.
1.2. De tram is nu "onderweg".
2. ELSE IF tram is "onderweg"
2.1 Plaats de tram in het volgende station.
3. Schrijf overzicht uit

Uitzonderingen:
Geen

Voorbeeld:
Gegeven de input van 1.1

```
Tram 12 vertrekt uit station A richting station B.
```

3.2. Automatische simulatie

Prioriteit:
VERPLICHT

Doel:
Simulatie automatisch laten lopen voor een gegeven tijd.

Preconditie:
Het systeem bevat een schema van de virtuele metronet.

Succesvol einde:
De simulatie stopt na het gegeven aantal stappen.

Stappen:
1. WHILE huidige tijd < eind tijd
1.1 FOR elke tram in het metronet
1.1.1 voer use case 3.1 uit op de tram

3.3. Rijden van trams met type

Prioriteit:
VERPLICHT

Doel:
Een specifieke tram zal zich anders gedragen afhangend van zijn type. Zie Appendix B voor meer informatie over de verschillende soorten trams. Hiervoor moet je ook use Use Case 1.2 implementeren.

Uitbreiding:
Use Case 3.1

Uitzonderingen:
Geen

3.4. Botspreventie

Prioriteit:
BELANGRIJK

Doel:
Om tot een meer realistische simulatie te komen moeten meerdere trams tegelijkertijd kunnen rondrijden op hetzelfde spoor. Hiervoor moet je ook use case 1.3 implementeren. Trams op eenzelfde spoor kunnen elkaar niet inhalen, maar moeten wachten tot de tram voor hen in een bepaald station vertrekt.

Preconditie:
Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:
Een tram die naar station X moet, moet wachten in station X-1 als er nog een tram in station X staat.

Uitbreiding:
Use case 3.2

Stappen:
[1.1.1, voor]
1.1.1 IF volgend station bezet
1.1.1.1 wacht in huidig station

3.5. Realistische tijd

Prioriteit:
BELANGRIJK

Doel:
Om een meer realistische simulatie te bekomen moet de tijd correct gesimuleerd worden. Start de simulatie om 12:00:00, en voer de simulatie uit volgens de tijden in Appendix C.

Preconditie:
Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:
De simulatie stopt na een gegeven tijd.

Uitbreiding:
Use case 3.2

3.6. Signalisatie

Prioriteit:
BELANGRIJK

Doel:
Een spoor kan twee soorten signalisatie bevatten: een snelheidslimiet en een stopplaats. Hiervoor moet je ook use Use Case 1.4 implementeren.
Indien een spoor een snelheidslimiet heeft, zal elke tram maximaal aan deze snelheid rijden (hiervoor moet je use case 3.5 ondersteunen).
Indien een spoor een stopplaats bevat, zullen trams hier stoppen als het volgende station reeds een tram bevat in plaats van in het vorig station te blijven staan (hiervoor moet je use case 3.4 ondersteunen).

Preconditie:
Het systeem bevat een schema van de virtuele metronet. Dit schema bevat minstens 1 signaal.

Succesvol einde:
Trams houden rekening met de signalisatie.

Uitbreiding:
use case 3.1

3.7. Passagiers

Prioriteit:
NUTTIG

Doel:
Om een meer realistische simulatie te bekomen moet elke tram bijhouden hoeveel passagiers in de tram zitten.

Preconditie:
Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:
De tram kan op elk moment zijn huidige bezetting opvragen.

Uitbreiding:
Use case 3.1

Stappen:
[1.1., voor]
1.1.1 genereer een willekeurig geheel getal tussen 0 en het aantal vrije zitplaatsen op de tram
1.1.2 verhoog de tram zijn huidige bezetting met dit getal

[2.1., na]
2.1.1 genereer een willekeurig geheel getal tussen 0 en de huidige bezetting van de tram
2.1.2 verlaag de tram zijn huidige bezetting met dit getal

3.8. Omzet per tram

Prioriteit:
NUTTIG

Doel:
Om de mogelijke opbrengst te simuleren houdt elke tram zijn omzet bij (alle passagiers betalen op de tram). Hiervoor moet je ook use Use case 3.7 implementeren.

Preconditie:
Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:
De tram kan op elk moment zijn omzet van de volledige simulatie tot nog toe opvragen.

Uitbreiding:
Use case 3.7

Stappen:
[1.1.1., na]
1.1.1.1. Omzet tram gaat omhoog met 2 euro per passagier

3.9 Statistische verwerking simulatie

Prioriteit:
NUTTIG

Doel:
Gedurende de uitvoering worden er relatieve gegevens verzamelt ivm bezettingsgraad en totale omzet. Alsook absolute gegevens over het gebruik van bepaalde tramlijnen.

Preconditie:
Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:
Rapport afgedrukt met statistische gegevens.

Uitbreiding:
Use case 3.2

Stappen:
1. WHILE simulatie actief
1.1 Verzamel gegevens
2. Bereken waarden
3. Druk rapport af

4.1. GUI voor de simulatie

Prioriteit:
NUTTIG

Doel:
Een gebruiksvriendelijke userinterface hebben voor het controleren van de simulatie. Deze bevat knoppen voor onder andere start en stop, alsook volgende en vorige stap.

Preconditie:
Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:
De simulatie kan bestuurd worden aan de hand van een grafische userinterface.

4.2. GUI voor het rijden van trams

Prioriteit:
NUTTIG

Doel:
Een gebruiksvriendelijke userinterface hebben voor het individueel besturen van trams.

Preconditie:
Het systeem bevat een grondplan voor een virtueel metronet.

Succesvol einde:
Trams kunnen manueel bestuurd worden aan de hand van een grafische userinterface.

Stappen:
1. Bepaal een tram in the GUI
2. Heb de mogelijkheid om de tram te wachten in, of onmiddelijk te vertrekken van, zijn huidige station

4.3. GUI voor statistische gegevens

Prioriteit:
NUTTIG

Doel:
Visualizatie van de statistische gegevens verzamelt in use case 3.9.

Preconditie:
Het systeem heeft statistische gegevens verzameld.

Succesvol einde:
De statistische gegevens worden weergegeven in een grafiek door middel van een grafische userinterface.

Uitbreiding:
Use case 3.9

A Invoer formaat

Het invoerformaat voor het virtueel metronet is zodanig gekozen dat nieuwe attributen en elementen makkelijk kunnen worden toegevoegd.

```
MetroNet = { Element }
Element = "<" ElementType ">" AttribuutLijst "</" ElementType ">"
ElementType = "STATION" | "TRAM" | "SIGNAAL" | "SPOOR"
AttribuutLijst = Attribuut { Attribuut }
Attribuut = "<" AttribuutType ">" AttribuutWaarde "</" AttribuutType ">"
AttribuutType = "naam" | "vorige" | "volgende" | "spoor" | "lijn"
                       | "type" | "beginStation" | "voertuigNr"
                       | "limiet"
AttribuutWaarde = Primitief
Primitief = Integer | String Integer = Digit { Digit }
Digit = "0" ... "9"
String = Letter { Letter }
Letter = "a" ... "z" | "A" ... "Z"
```

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

Element	Attribuut (verplicht)	Attribuut (optioneel)
Station	naam, volgende, vorige, spoor, type	/
Tram	lijn, beginStation, type	voertuigNr
Signaal	/	type, vorige, volgende, spoor, limiet
Spoor	/	spoor, volgende, vorige

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

Attribuut	Waarde
naam, vorige, volgende, beginStation, type	String
spoor, lijn, voertuigNr, limiet	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 het in te lezen metronet wordt met de hand geschreven. Om het ingelezen metronet te kunnen simuleren moet de informatie consistent zijn.

Een metronet is consistent als:

- elk station is verbonden met een voorgaand en een volgend station voor elk spoor.
- elke tram heeft een lijn die overeenkomt met een spoor in zijn beginstation.
- er geen sporen zijn waarvoor geen tram bestaat.
- het startstation van een tram is een geldig station in het metronet.
- elk spoor komt maximaal 1 keer voor in een station.

B Tram data

De volgende tabel toont de data voor elk type tram:

	PCC	Albatross
Zitplaatsen	16	72
Snelheid	40	70
Bedient	metrostation, halte	metrostation

C Realistische tijd simulatie

De tijd (in seconden) die nodig is voor een tram om van een station naar het volgende te rijden kan berekend worden met de volgende formule:

$$t = 3600 * \frac{afstand}{snelheid}$$

Echter, omdat de tram niet altijd aan zijn maximale snelheid zal rijden, zou deze formule onrealistisch snelle trams veroorzaken. Daarom zullen we een meer realistische benadering gebruiken door het resultaat van de formule te verdubbelen. Bovendien kunnen we de afstand tussen twee stations in Antwerpen benaderen met 1 kilometer. We krijgen dan de volgende formule:

$$t = 2 * \left(3600 * \frac{1}{snelheid}\right)$$

of

$$t = \frac{7200}{snelheid}$$

Voor de tijd dat een tram in een station staat om passagiers te laten op- en afstappen, gebruiken we een constante 60 seconden. Merk op dat de Albatross trams niet stoppen bij een bovengrondse halte, en dus terug vertrekt na 1 seconde.