

Luchthaven Simulatie

Documentsoort:	Behoeftespecificatie
Versie:	1.0
Datum:	15 februari 2018
Auteurs:	Brent van Bladel
Status:	Opgeleverd

1 Samenvatting

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

2 Context

In 2017 is de uitbreiding van de luchthaven van Deurne tot een succesvol einde gebracht. Nu zijn er plannen om in 2018 te starten met een uitbreiding van *Brussels Airport* (beter bekend als Zaventem). Deze uitbreiding van infrastructuur heeft tot gevolg een verhoging van het vliegtuig-verkeer op de taxi- en landingsbanen van de luchthaven. Om luchtverkeersleiders te helpen met deze verhoging heeft het Departement Mobiliteit en Openbare Werken geopteerd een simulatie model te laten ontwikkelen dat dit vliegtuig-verkeer kan simuleren in een luchthaven.

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. Luchthaven en vliegtuigen inlezen	VERPLICHT
<i>2: Uitvoer</i>	
2.1. Simpele uitvoer	VERPLICHT
<i>3: Simulatie</i>	
3.1. Landen van vliegtuigen	VERPLICHT
3.2. Opstijgen van vliegtuigen	VERPLICHT
3.3. Vliegtuig aan gate	BELANGRIJK

1.1. Luchthaven en vliegtuigen inlezen

Prioriteit:

VERPLICHT

Doel:

Inlezen van het schema van de luchthaven. De verschillende landingsbanen, hoe die met elkaar verbonden zijn en de verschillende vliegtuigen.

Preconditie:

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

Succesvol einde:

Het systeem bevat een schema met de verschillende landingsbanen, en informatie over alle vliegtuigen.

Stappen:

1. Open invoerbestand
2. WHILE Bestand niet ingelezen
 - 2.1. Herken het soort element (RUNWAY, AIRPLANE, AIRPORT)
 - 2.2. Lees verdere informatie voor het element
 - 2.3. IF Verifieer geldige informatie
 - 2.3.1. THEN Voeg element toe aan de virtuele luchthaven
 - 2.3.1. ELSE Foutboodschap + positioneer op volgende element in het bestand
3. Verifieer consistentie van de luchthaven
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 luchthaven] Foutboodschap \Rightarrow verdergaan vanaf stap 4

Voorbeeld:

Een luchthaven met een landingsbaan en een vliegtuig.

```
<AIRPORT>
  <name>Antwerp International Airport</name>
  <iata>ANR</iata>
  <callsign>Antwerp Tower</callsign>
  <gates>10</gates>
</AIRPORT>
```

```
<RUNWAY>
  <name>11R</naam>
  <airport>ANR</airport>
</RUNWAY>
```

```
<AIRPLANE>
  <number>N11842</number>
  <callsign>Cessna 842</callsign>
  <model>Cessna 340</model>
  <status>Approaching</status>
</AIRPLANE>
```

2.1. Simpele uitvoer

Prioriteit:

VERPLICHT

Doel:

Uitvoer van alle informatie in de virtuele luchthaven.

Preconditie:

Het systeem bevat een grondplan voor de virtuele luchthaven.

Succesvol einde:

Het systeem heeft een tekstbestand (ASCII) uitgevoerd, waarin de informatie over de virtuele luchthaven netjes is uitgeschreven.

Stappen:

1. Open uitvoerbestand
2. WHILE Nog luchthavens beschikbaar
- 2.1. Schrijf luchthaven-gegevens uit
3. WHILE Nog vliegtuigen beschikbaar
- 3.1. Schrijf vliegtuig-gegevens uit
4. Sluit uitvoerbestand

Uitzonderingen:

Geen

Voorbeeld:

Gegeven de input van 1.1

Airport: Antwerp International Airport (ANR)

-> gates: 10

-> runways: 1

Airplane: Cessna 842 (N11842)

-> model: Cessna 340

3.1. Landen van vliegtuigen

Prioriteit:

VERPLICHT

Doel:

Simuleren van het landen van een vliegtuig.

Preconditie:

Het systeem bevat een grondplan voor een virtuele luchthaven. Er is een vliegtuig met als status **Approaching**.

Succesvol einde:

Het vliegtuig bevindt zich aan een gate in de luchthaven. Het systeem heeft een boodschap afgedrukt met de details van de landing.

Stappen:

0. Aankomende vliegtuigen vliegen op een hoogte van 10000 ft.
1. WHILE vliegtuig hoger dan 1000 ft.
 - 1.1 Vliegtuig daalt 1000 ft.
2. Vliegtuig landt op een vrije landingsbaan
3. Vliegtuig taxiëert naar een vrije gate

Uitzonderingen:

Geen

Voorbeeld:

Gegeven de input van 1.1

```
Cessna 842 is approaching Antwerp International Airport at 10.000 ft.  
Cessna 842 descended to 9.000 ft.  
Cessna 842 descended to 8.000 ft.  
Cessna 842 descended to 7.000 ft.  
Cessna 842 descended to 6.000 ft.  
Cessna 842 descended to 5.000 ft.  
Cessna 842 descended to 4.000 ft.  
Cessna 842 descended to 3.000 ft.  
Cessna 842 descended to 2.000 ft.  
Cessna 842 descended to 1.000 ft.  
Cessna 842 is landing at Antwerp International Airport on runway 11R  
Cessna 842 has landed at Antwerp International Airport on runway 11R  
Cessna 842 is taxiing to Gate 1  
Cessna 842 is standing at Gate 1
```


3.2. Opstijgen van vliegtuigen

Prioriteit:

VERPLICHT

Doel:

Simuleren van het opstijgen van een vliegtuig.

Preconditie:

Het systeem bevat een grondplan voor een virtuele luchthaven. Er is een vliegtuig met als status **Standing at gate**.

Succesvol einde:

Het vliegtuig heeft de luchthaven **verlaten**. Het systeem heeft een boodschap afgedrukt met de details van het vertrek.

Stappen:

1. Vliegtuig taxiëert naar een vrije landingsbaan
2. Vliegtuig stijgt op
3. WHILE vliegtuig lager dan 5000 ft.
 - 3.1 Vliegtuig stijgt 1000 ft.

Uitzonderingen:

Geen

Voorbeeld:

Gegeven de input van 1.1

```
Cessna 842 is standing at Gate 1
Cessna 842 is taxiing to runway 11R
Cessna 842 is taking off at Antwerp International Airport on runway 11R
Cessna 842 ascended to 1.000 ft.
Cessna 842 ascended to 2.000 ft.
Cessna 842 ascended to 3.000 ft.
Cessna 842 ascended to 4.000 ft.
Cessna 842 ascended to 5.000 ft.
Cessna 842 has left Antwerp International Airport
```

3.3. Vliegtuig aan gate

Prioriteit:

BELANGRIJK

Doel:

Simuleren van het staan van een vliegtuig aan een gate.

Preconditie:

Het systeem bevat een grondplan voor een virtuele luchthaven. Er staat een vliegtuig aan een gate.

Succesvol einde:

Het vliegtuig is klaar om te vertrekken.

Stappen:

1. IF vliegtuig is net geland
 - 1.1. laat passagiers afstappen
 - 1.2. technische controle van het vliegtuig
2. IF vliegtuig gaat vertrekken
 - 2.1. vul brandstof bij van het vliegtuig
 - 2.2. laat passagiers opstappen

Uitzonderingen:

Geen

Voorbeeld:

```
4 passengers exited Cessna 842 at gate 1 of Antwerp International Airport
Cessna 842 has been checked for technical malfunctions
Cessna 842 has been refueled
4 passengers boarded Cessna 842 at gate 1 of Antwerp International Airport
```

A Invoer formaat

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

```
AirportSim = { Element }
Element = "<" ElementType ">" AttributeList "</" ElementType ">"
ElementType = "AIRPORT" | "RUNWAY" | "AIRPLANE"
AttributeList = Attribute { Attribute }
Attribute = "<" AttributeType ">" AttributeValue "</" AttributeType ">"
AttributeType = "name" | "iata" | "callsign" | "gates" | "airport"
                | "number" | "model" | "status" | "passengers"
                | "fuel"
AttributeValue = Integer | String
Integer = Digit { Digit }
Digit = "0" ... "9"
String = Character { Character }
Character = "a" ... "z" | "A" ... "Z" | Digit
```

Merk op dat de attributlijst 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:

ElementType	Attribute (mandatory)	Attribute (optional)
AIRPORT	name, iata, callsign, gates	passengers
RUNWAY	name, airport	-/-
AIRPLANE	number, callsign, model, status	passengers, passengerCapacity

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

AttributeType	AttributeValue
name, iata, callsign, airport, number, model, status	String
gates, passengers, fuel	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 luchthaven wordt met de hand geschreven. Om de ingelezen luchthaven te kunnen simuleren moet de informatie consistent zijn.

Een luchthaven is consistent als:

- Elke landingsbaan is verbonden met een bestaande luchthaven.
- Elk vliegtuig initieel de status “approaching” of “standing at gate” heeft.
- Er maximaal 1 vliegtuig aan elke gate staat.
- Initieel er maximaal 1 vliegtuig met de status “approaching” is.

Opmerkingen:

- Gates worden gewoon genummerd van 1 tot n, waarbij n het aantal gates.
- De iata code wordt gebruikt als unieke identificatie van een luchthaven.
- De registratienummer (number) wordt gebruikt als unieke identificatie van een vliegtuig.
- De combinatie van naam en bijhorende luchthaven wordt gebruikt als unieke identificatie van een landingsbaan.
- De attributen “fuel” en “passengers” geven de maximale hoeveelheid weer. We gaan er van uit dat vliegtuigen altijd vliegen met maximale hoeveelheid.