**Containing**

****

**Gemaakt door:**

Joshua Bergsma

Remco de Bruin

Melinda de Roo

Arjen Pander

Jeffrey Harders

Yme van der Graaf

Inhoudsopgave

[Opdrachtbeschrijving 3](#_Toc372544838)

[Planning 4](#_Toc372544839)

[Taakverdeling 5](#_Toc372544840)

[Testplan 6](#_Toc372544841)

[Eisen / wensen overzicht 7](#_Toc372544842)

# Opdrachtbeschrijving

# Planning

# Taakverdeling

# Testplan

# Eisen / wensen overzicht

Hieronder vindt u een overzicht van de eisen en wensen gevolgd door een tabel met daarin de gekozen snelheden.

**Eisen:**

A) De simulatie moet 3D zijn, waarbij je gebruik mag maken van OpenGL.

B) Er moet een wiskundige verantwoording komen van de kortste route.

C) Er moet een complete UML analyse komen

D) Het verslag moet samenhangend, doorlopend en goed geschreven zijn.

E) De controller moet met de gegeven input werken, te weten datafiles in XML format, oplopend van klein naar (heel) groot. NB: er zitten fouten in de XML-files, en het programma moet dus zo robuust zijn dat het daar geen last van heeft.

F) De programma's moeten in Java geschreven worden. Op Apple platform kan het nodig zijn om met de bijbehorende tools te werken.

G) De werking van de software moet getest worden, zó dat de opdrachtgever overtuigd raakt van de kwaliteit. Eis: gebruik Junit.

**Wensen:**

1. Maak de weg waarover de AGV's rijden vierbaans zodat onbeladen AGV's beladen AGV's in kunnen halen.

2. Maak de grafieken op het android-platform real-time.

3. Maak het mogelijk dat de simulatie tot 100 keer versneld kan worden afgedraaid.

4. Maak het mogelijk dat via de simulator (het beeldscherm) een operator in kan grijpen wanner hij of zij dat noodzakelijk vindt. Bijvoorbeeld om de minimale afstand die de AGV's onderling moeten houden te veranderen als het erg druk is. Of om een bepaalde lading containers een hogere prioriteit te geven omdat er verwacht wordt dat er een lucratieve serie orders achterweg kan komen.

5. Gegevens van de containers staan in een database.

6. In de simulator het mogelijk maken dat een operator calamiteiten kan laten ontstaan, bv. De Operator geeft aan dat de een AGV of een Kraan etc..kapot gaat. Visueel moet direct duidelijk zijn dat er zich een probleem voordoet. Een calamiteit moet vanaf het toetsenbord (met -of zonder muis) invoerbaar zijn.

7. Een operator interface voor de simulator waarmee calamiteiten realtime (binnen 10 seconden) kunnen worden in gebracht en zichtbaar gemaakt kunnen worden. De reactie van de Controller moet realtime (binnen 30 seconden) en zichtbaar in de simulator zijn.

8. Het 'in-zoomen' op 1 container, die bekeken en gedraaid moet kunnen worden en waarvan de gegevens in een extra venster moeten komen.

9. Het 'in-zoomen' op één container, zie eis 7) maar nu zonder gebruik te maken van OpenGL.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Zekeren in min.** | **Hijsen in min**. | **Zakken in min.** | **Container verplaatsen** | **Onbeladen snelheid** | **Beladen snelheid** |
| AGV |  |  |  |  | 40 km/u | 20 km/u |
| Opslagrailkraan | 0.5 | 0-3.5 | 0-3.5 | 5 m/s | 3 m/s | 2 m/s |
| Zeeschipkraan | 0.5 | - | - | 5 minuten | 4 km/u | - |
| Treinkraan | 0.5 | 1-2 | 0.5-1 | 5 m/s | 3 m/s | 2 m/s |
| Binnenvaartkraan | 0.5 | 0-3.5 | 0-3.5 | 5 m/s | 8 km/u | 3 km/u |
| Vrachtautokraan | 0.5 | 1 | 1 | - | 20 km/u | 15 km/u |