

Functioneel Ontwerp

Project Team Building Challenge



Klas: EHI1V.Sb

Team: 2

Inhoudsopgave

Inhoudsopgave	2
Project scope	3
Belanghebbenden	3
Doelen	3
Product(en).....	3
Koppelvlakken	3
Risico's.....	4
Product overzicht.....	5
Korte omschrijving.....	5
Diagrammen	5
Wireframe.....	5
Autopilot.....	8
Requirements.....	9
Gebruikersplatform	10
Hardware.....	10
Software.....	10
Uitrol- of installatiehandleiding	11

Documenthistorie:

Versie	Datum	Gewijzigd door	Wijziging
0.1	26-04-2021	Emanuel, Kalli en Jochem	Project scope, Risico's, Requirements en Product overzicht
1.0	21-05-2021	Emanuel, Kalli en Jochem	Gebruikersplatform
1.1	04-06-2021	Emanuel, Kalli en Jochem	Aanpassingen voor sprint 2
1.2	29-06-2021	Emanuel, Kalli en Jochem	Aanpassingen voor sprint 3

Project scope

Belanghebbenden

Wie zijn er allemaal betrokken bij dit project?

Rol	Persoon
Ontwikkelaar, Scrum Master	Emanuel de Jong
Ontwikkelaar	Jochem Bleeker
Ontwikkelaar	Kalli van den Heuvel
Klant, Product owner	Jelle de Jong

Doelen

Een software ontwikkelen dat de FROG zelfstandig kan opereren.

Product(en)

Welke applicatie(s) en andere zaken worden er opgeleverd?

Een Java Swing desktopapplicatie om de THE FROG te monitoren en te besturen.

Koppelvlakken

Welke raakvlakken heeft het project/ de te ontwikkelen applicatie(s) met andere processen en systemen?

Het project heeft te maken met de manier waarop het communiceert richting THE FROG en hoe het informatie terug krijgt.

Risico's

Welke risico's kun je onderkennen bij dit project, hoe groot (Groot, Middel of Klein) schat je de kans in dat deze optreedt, hoe groot (idem G/M/K) schat je de impact als dit risico daadwerkelijk optreedt en wat wordt hier als tegenmaatregel aan gedaan (mitigatie/ actie, kan bijv. ook zijn dat we deze gewoon accepteren indien kans en impact klein zijn)?

Nr.	Risico	Kans (GML)	Impact (GML)	Mitigatie/ actie
1	Tekort aan werkkraft	L	G	Taken onderverdelen
2	Scope onderschatting	M	M	Achterstand inhalen met overuren
3	Onduidelijkheden	M	G	Uitleg vragen en fouten corrigeren

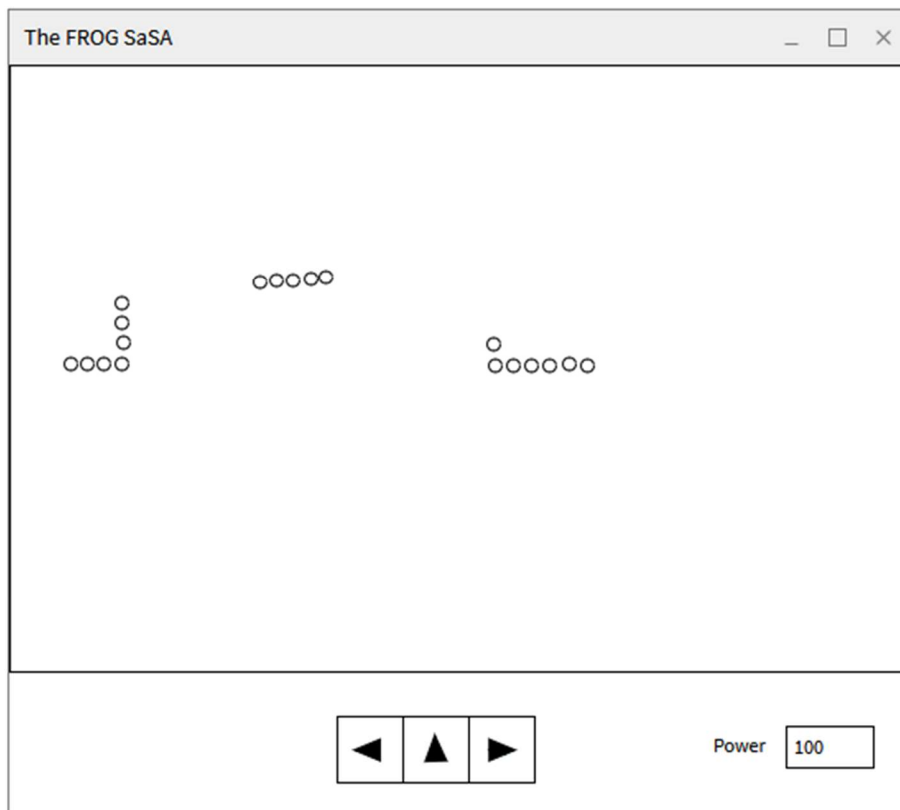
Product overzicht

Korte omschrijving

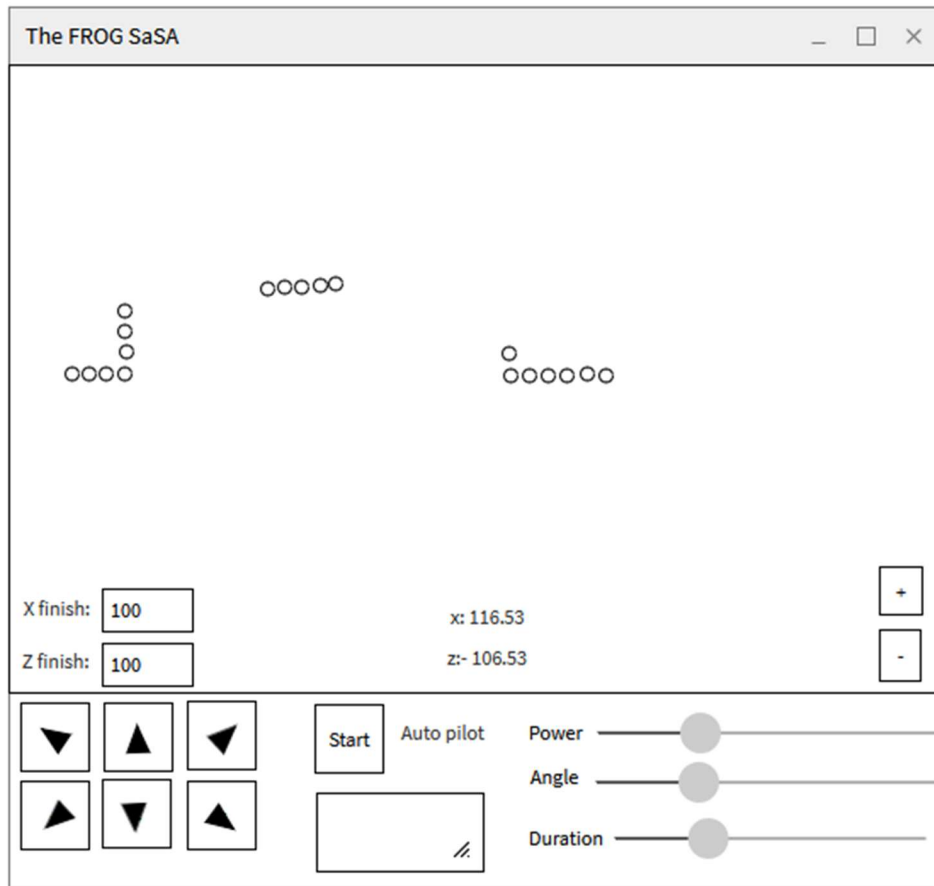
Om de maanrover volledig zelfstandig te laten opereren moet het kunnen navigeren naar een bestemming zonder obstakels te raken. Ook moeten het kunnen inparkeren en terugrijden naar zijn thuisbasis.

Diagrammen

Wireframe

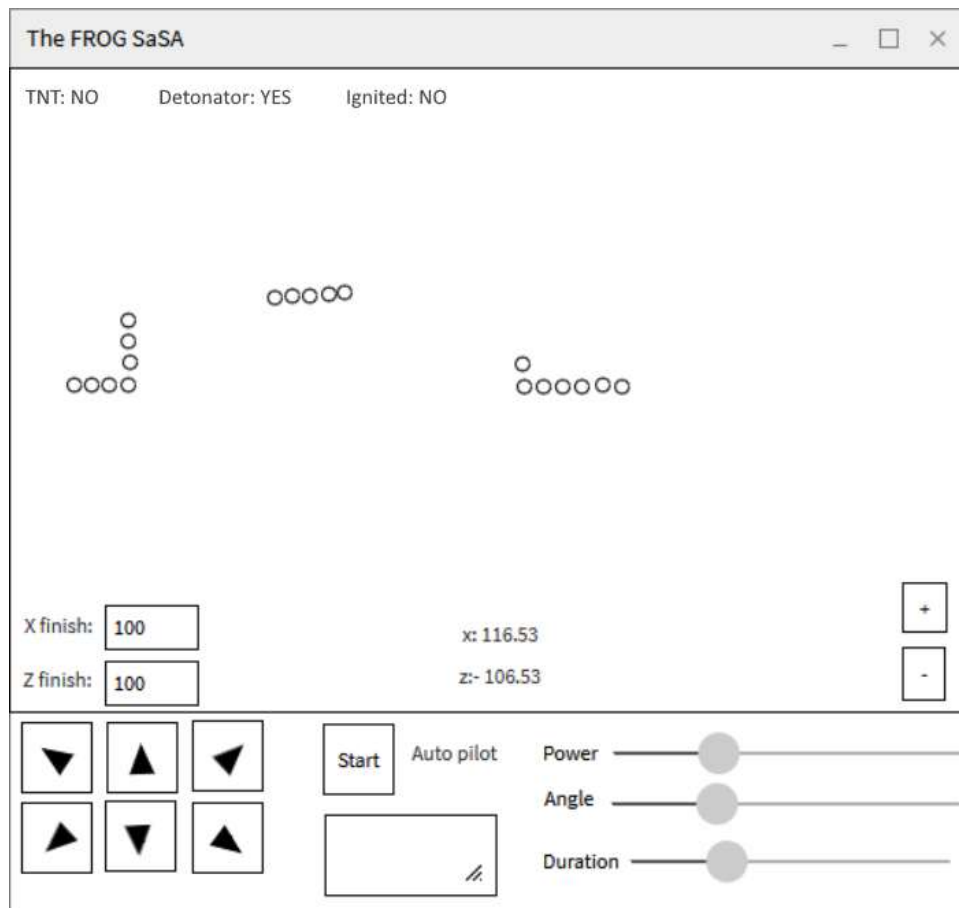


Dit is de wireframe van de applicatie zoals de applicatie er aan het eind van de eerste sprint uitzag. We wilde voor het maken van een Swing UI deze wireframe maken om te zien of de lay-out mooi/kloppend zou zijn.



Dit is de wireframe van de applicatie zoals hij er na de tweede sprint uit ziet. Met dus meer mogelijkheden om te sturen maar ook het starten en stoppen van de auto pilot die je hier ziet.

Ook is de radar data in verschillende kleuren te zien. Deze kleuren staan voor zones waarmee autopilot beslissingen maakt.

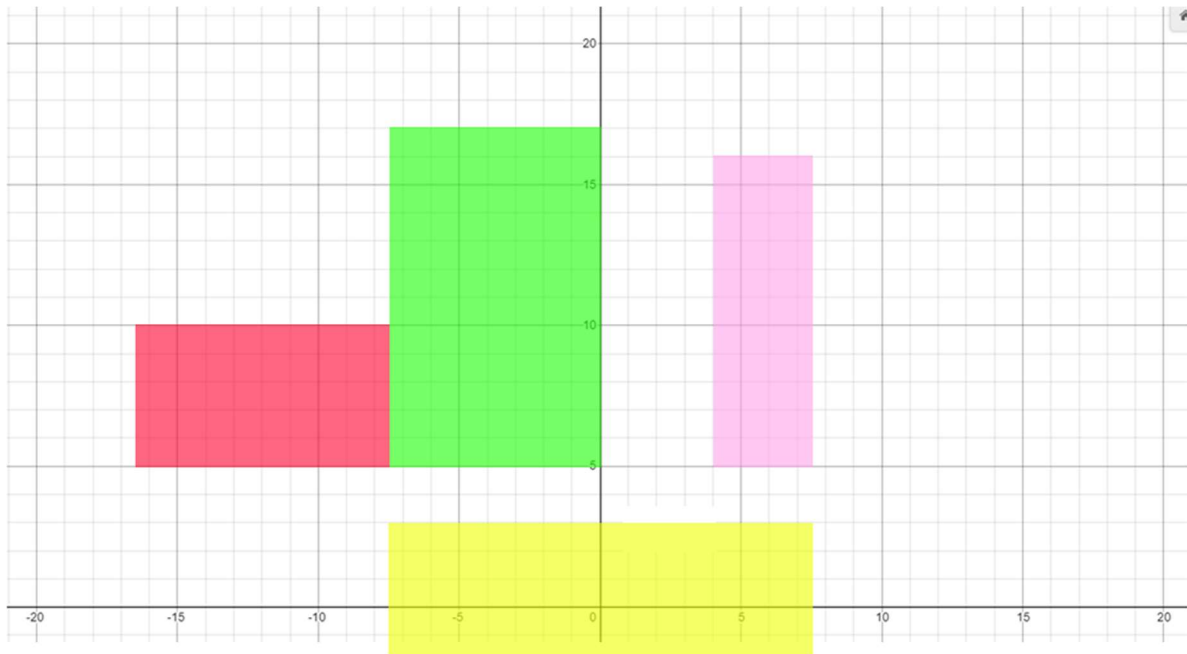


Er is nu een nieuw data te zien linksboven in het scherm. Deze data laat zien hoever de rover in zijn missie is. Als ignited op YES staat gaat de rover gecontroleerd naar achter rijden en wacht geduldig op de ontploffing. Daarna rijdt de rover weer voor uit en vervolgt zijn autopilot gedrag.

Autopilot

De autopilot neemt het sturen van de rover over. Het zoekt een linker muur op en laat de rover langs deze muur rijden.

Het kijkt naar verschillende zones binnen de radar data en beslist hiermee wat de volgende beweging word. Als er datapunten aanwezig zijn in een zone word de zone als actief gezien.



Hiërarchie	Zones	Beslissing
1	Geel	Achteruit
2	Groen	Rechts
3	Roze	Links
4	Rood	Rechtdoor
5	Geen	Links

Requirements

De lijst van zaken waaraan het op te leveren product moet/ zou moeten/ kan/ zou kunnen (de MoSCoW categorieën) voldoen. Deze lijst maakt daarbij een onderscheid tussen F (Functionele – voor de eindgebruiker zichtbare) en NF (Niet Functionele – interne, voor de eindgebruiker niet zichtbare) requirements. Ook wordt hierbij een bron (van wie komt deze requirement) aangegeven, meestal een (eindgebruiker) rol.

Nr.	Requirement	F/NF	MoS CoW	Bron
1	MVC-template maken	NF	M	Eindgebruiker
2	Rover data inlezen	NF	M	Eindgebruiker
3	Rover data visualiseren op het dashboard	F	M	Eindgebruiker
4	Het besturen van de rover met knoppen in het dashboard	F	M	Eindgebruiker
5	Rover moet zelf naar een bestemming kunnen rijden.	F	M	Eindgebruiker
6	Rover moet zelf kunnen inparkeren	F	M	Eindgebruiker
7	Rover moet weten waar zijn thuisbasis is.	NF	M	Eindgebruiker
8	Rover moet de efficiëntste route nemen	NF	C	Ontwikkelaar

Gebruikersplatform

Hier beschrijf je de benodigde hard- en software waar de eindgebruiker de eindapplicatie op moet kunnen draaien dan wel welke servers er nodig zijn en hoe deze moeten worden ingericht.

Hardware

Een volledig inzicht in de te gebruiken hardware om de applicatie te kunnen draaien. Maak eventueel een onderscheid in minimale eisen en geadviseerde eisen.

Categorie	Product
CPU	Pentium 2 266 MHz
RAM	4GB 1066 MHz DDR3

Software

Inventarisatie van alle software die nodig is om de applicatie te draaien (buiten de ontwikkelde software zelf dus).

Categorie	Product
Besturingssysteem	Windows 7+, Linux, MacOS X 10.8.3+
Platform	Java JRE 11

Uitrol- of installatiehandleiding

1. Download en installeer Java: <https://www.java.com/en/download/>
2. Download het JAR bestand van MEGA: https://mega.nz/file/xupj1IhJ#fASMkFUYGxTgi6DnI8fkfV-xquaJGJSU0T_49wWRanY
3. Zet de WinFrog folder in dezelfde folder als het JAR bestand.
4. Dubbelklik eerst het JAR bestand en dat de "The Frog.exe" in de WinFrog folder.
5. De dashboard zou nu moeten werken.

