# Groeidocument

## Robotica


**Schooljaar:** 2019-2020

**Versie:** 4 (Week 5)

**Opgesteld door:** Groep 10

**Datum:** 29 mei 2020

# Inhoudsopgave

# 1 Protocol communicatie controller

## 1.1 Gebruikte protocollen volgens het OSI-model (Nog niet afgerond)

### Laag 1: Fysiek

De fysieke laag is het transportmedium waarover de data wordt verzonden. Dit kan worden gedaan met elektrische-, optische- of radiosignalen.

### Laag 2: Datalink

De datalink laag zorgt voor het transport van data tussen twee fysiek verbonden apparaten.

### Laag 3: Netwerk

De netwerk laag is verantwoordelijk voor de route van de data door een netwerk van fysiek verbonden apparaten.

### Laag 4: Transport

De transport laag zorgt ervoor dat de verstuurde data correct bij een ander apparaat aankomt. Hier wordt de Transmission Control Protocol voor gebruikt.

### Laag 5: Sessie

De sessie laag onderhoudt/beëindigt de verbinding tussen twee apparaten. De controller heeft momenteel nog geen protocol voor de sessie laag. De controller kan wel verbinden met de server maar kan nog niet de verbinding correct afsluiten.

### Laag 6: Presentatie

De presentatie laag vertaalt/formatteert de data zodat de applicatie laag deze kan weergeven aan de gebruiker. Er wordt in de controller (nog) niet een protocol voor gebruikt. We zijn nog aan het definiëren hoe de commandos met een STRING type verstuurd kan worden.

| Key | Value | Type |
|---|---|---|
| MF (Move Forward) | Distance [CM] | Int |
| MB (Move Backwards) | Distance [CM] | Int |
| RO (Rotate) | Angle / -Angle [DEGREES] | Int |
| SD (start Single Dance) | - | Bool |
| LD (start Line Dance) | - | Bool |
| MM (start Moon Maze) | - | Bool |
| | | |

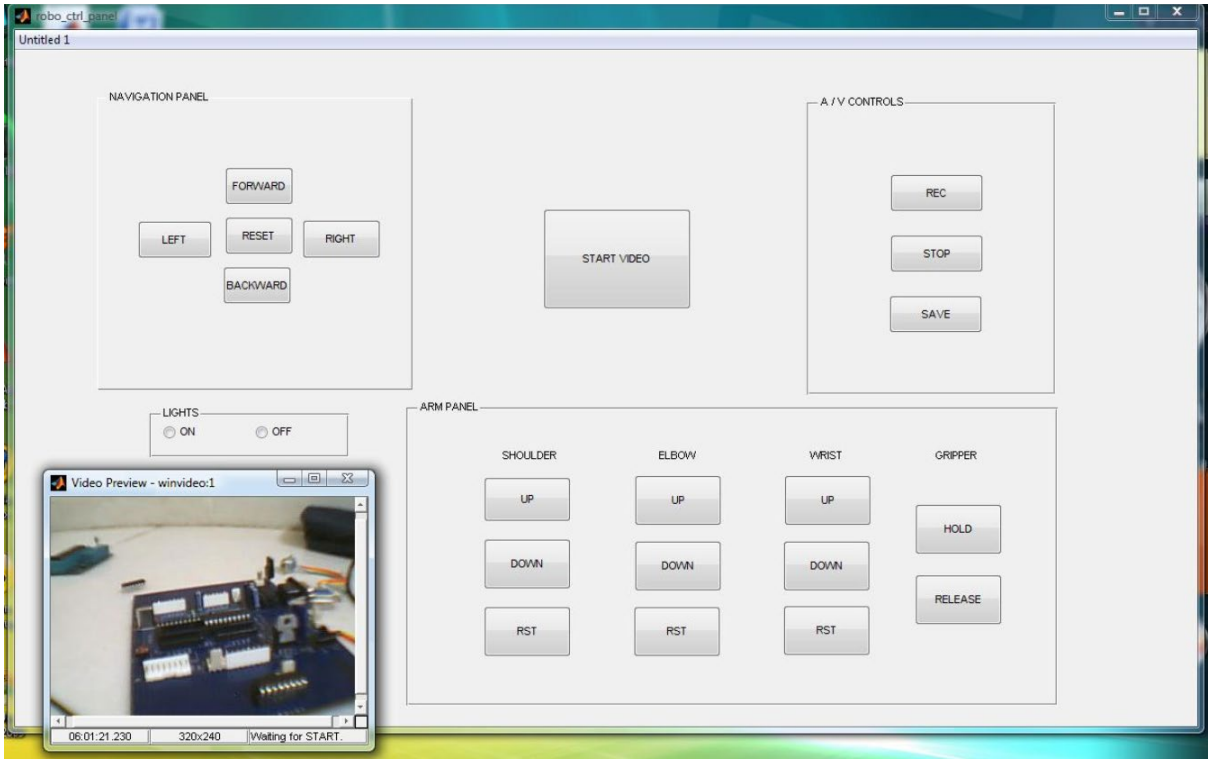
### Laag 7: Applicatie

De applicatie laag staat het dichtste bij de gebruiker en wordt gebruikt om de mens met de machine te laten communiceren. Om de robot aan te sturen maken we gebruik van Windows Forms App. Deze maakt gebruik van de .net framework.

## 1.2 Controller GUI-aansturing functies

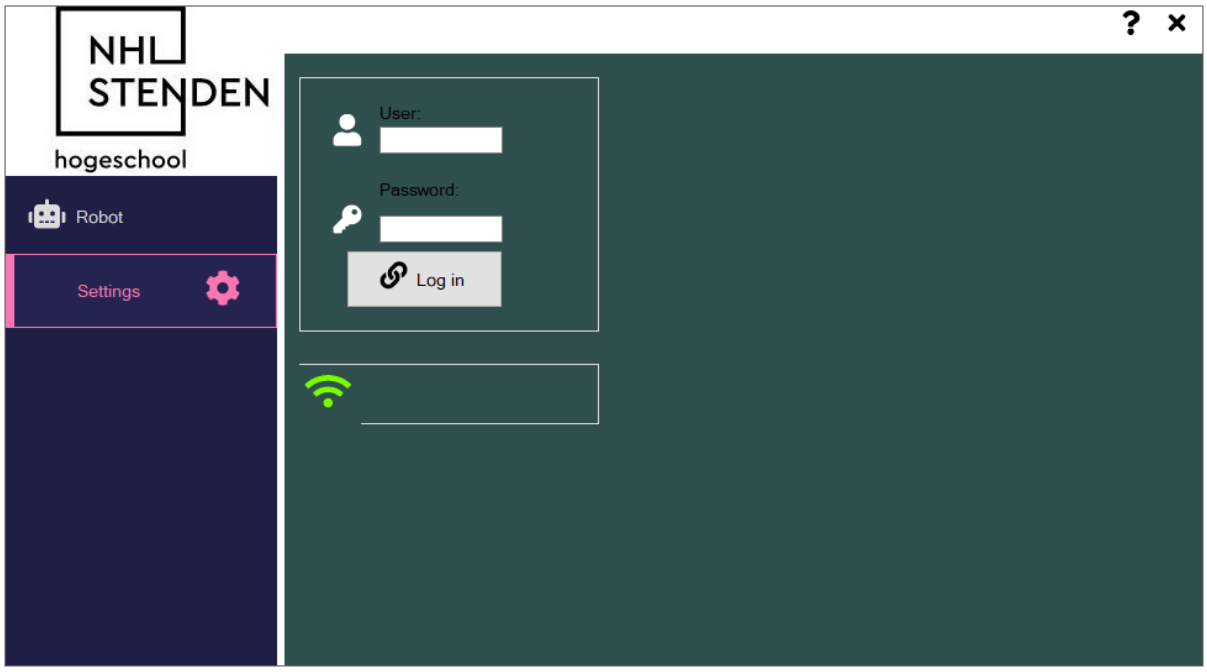
- Aansturing
    - Handmatig
        - Vooruit {w}
        - Achteruit {s}
        - Tegen de klok in draaien {a}
        - Met de klok mee draaien {d}
    - Autonoom
        - “Single” dance knop
        - “Line” dance knop
        - Start “moonmaze” knop
    - Spraak
        - Knop voor spreken
- Uitlezen
    - Camera feed

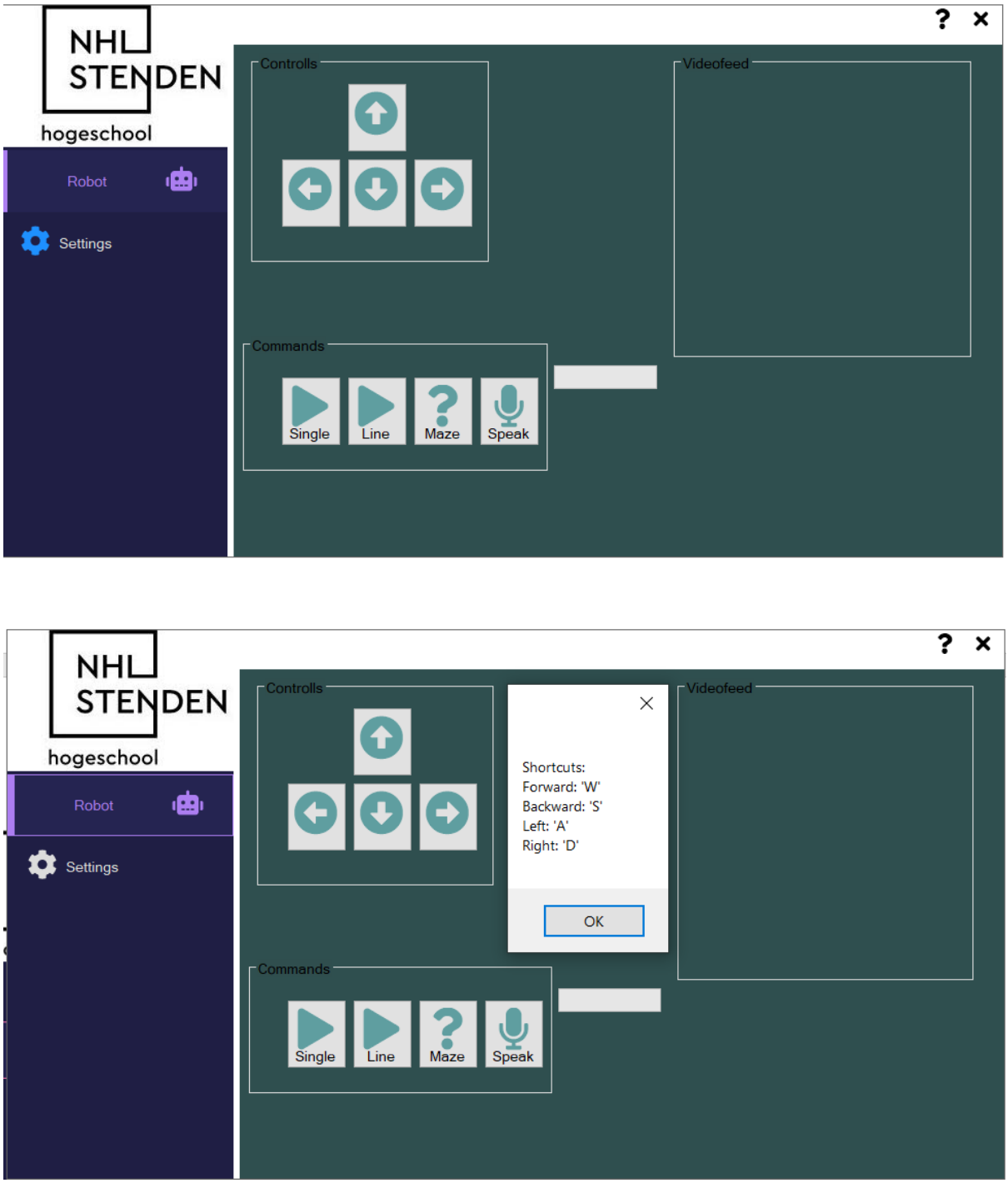
Inspiratie:



## 1.3 Controller

In samenwerking met SE gaan wij onze controller maken in Visual Studio. Deze applicatie kan straks communiceren via TCP met een webserver. De bedoeling is dat dit een fullduplex systeem wordt. De GUI geeft een aantal dingen realtime weer op onze interface (zie 8.2).




*Figuur 1 vraagteken knop functie*


*Figuur 2 Ontvangen berichten van de server*

# 2 Implementatie binnen WeBots

Met de UML-diagrammen uitgewerkt en een eerste prototype van het model van de robot kan de SE kant van het team eindelijk een begin maken aan de implementatie van het besturingssysteem van de robot. Hiertoe is er echter eerst nog een stukje onderzoek gedaan naar de werking hiervan binnen de WeBots omgeving.

## 2.1 Functioneel maken van het prototype

Voordat het door WTB aangeleverde prototype gebruikt kan worden om mee te testen tijdens het programmeren, moet deze allereerst functioneel worden gemaakt binnen de virtuele omgeving. Hiernaar is een kort onderzoek gedaan, door te kijken naar bestaande vergelijkbare robots en door de documentatie van WeBots zelf te bestuderen.

Op basis hiervan is uitgevonden hoe de verschillende onderdelen omgezet kunnen worden naar functionele onderdelen. Zo zijn onder andere de physics van het model aangezet en zijn er werkende rupsbanden geplaats om de wielen van de robot.

### 2.1.1 Aansturing

In deze sprint is het team van aansturing begonnen met het maken van de daadwerkelijke code voor de aansturing. Na grondig onderzoek te hebben gedaan is dit snel verlopen. Het skelet van de code is opgezet en de implementatie hiervan begonnen. Bungie kan op dit moment rijden en remmen.

## 2.2 Spraakaansturing

De spraakaansturing pakt de microfoon goed op en werkt goed in het Engels, de spraak wordt omgezet in tekst. En afhankelijk van wat die tekst precies is zal het een commando uitvoeren van de normale aansturing. Bijvoorbeeld “Drive forward” zal dan aansluiten naar het stuk code dat de goeie componenten van de robot aanstuurt om vooruit te rijden.

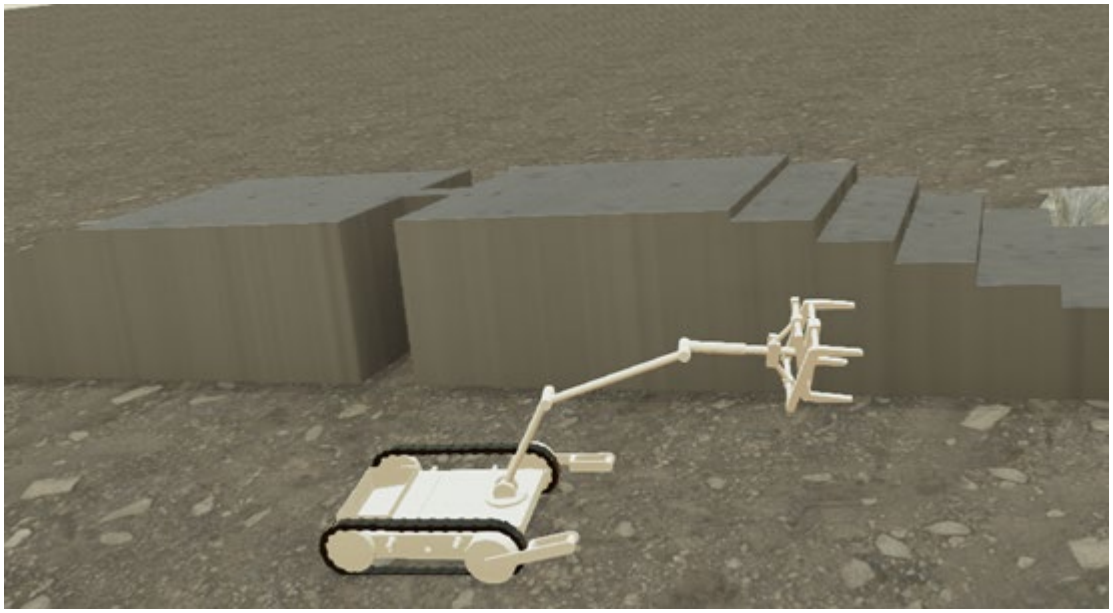
```
speak now
Command was: Drive forward
Well done!
[Finished in 9.781s]
```

Hier even een voorbeeld, het print nu nog “Well done!” uit maar dat zal natuurlijk aangepast worden naar de correcte implementaties. Er moet nog onderzoek gedaan worden naar hoe lang hij blijft luisteren na dat het commando is gegeven, qua versnelling van uitvoering.

## 2.3 Machine intelligence

Machine intelligence heeft de basiscode van de meeste vereiste opdrachten verwerkt. Deze vereiste opdrachten (de must haves) vanuit de MoSCoW zijn omgezet naar een basis skelet voor Machine Intelligence. Het gaat hier vooral om de structuur van de code, de stappen die Machine Intelligence moet uitvoeren en hoe vanuit deze code gecommuniceerd gaat worden met Vision en Aansturing. De structuur van deze code is gebaseerd op de eerder gemaakte UML diagrammen van Machine Intelligence, Vision en Aansturing. Op het moment dat deze code samengevoegd gaat worden met Vision en Aansturing, kan er een start worden gemaakt aan het implementeren van de gemaakte functionaliteit.

Ook is er een start gemaakt aan het realiseren van de autonome “maze exploration” opdracht. Voor deze opdracht hebben we gekozen om onder andere het A* algoritme te implementeren. Om dit algoritme te kunnen implementeren volgen we een aantal YouTube tutorials, deze zijn via deze link te vinden. Het testen van het algoritme in WeBots zal plaatsvinden wanneer de NSA het “maze” worldobject beschikbaar maakt.



*WeBots Robot op het maanlandschap voor de opdracht “Moon Survival”*

Voorafgaand aan deze sprint stond alle code van Machine Intelligence in een aparate branch op github. Deze code is samengevoegd met het WeBots project, zodat Machine Intelligence de geimplementeerde methoden van Vision en Aansturing kan aanroepen.

# 3 Bronvermelding

https://nl.wikipedia.org/wiki/Gelijkstroommotor

https://www.zxy.nl/techniek/robotica_motoren.php
https://nl.wikipedia.org/wiki/Morfologisch_overzicht

https://en.wikipedia.org/wiki/Brushless_DC_electric_motor

http://nl.zonsutech.com/info/servo-motor-working-advantages-disadvanta-38993021.html

https://www.elektrischvaren-accu.nl/Verschillende-type-accus

https://www.zonne-energiegids.nl/soorten-zonnepanelen/

https://www.zonnepaneelprijzen.nl/soorten/

https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller

https://en.wikipedia.org/wiki/OSI_model

https://osi-model.com/