Table of Contents

Welcome

Project Willy

- History of Willy
- Project Willy
- Publicity
- Sponsors

Getting started

- Development Guide
- Driving Willy
- Documentation

Build of Willy

- Design history
- Requirements
- Design reference
- Physical build

- Hardware

Robotic Operating System

- Introduction to ROS
- ROS Tutorials
- Multi master

Architecture

- Software Architecture
- Hardware Architecture
- ROS topic design

Hardware nodes

- sensor node
- si node
- power node
- WillyWRT

Components

- ROS master
- New ROS master on Lubuntu
- Brain
- Sonar
- Lidar
- Localization and navigation
- Motor controller
- Joystick
- Social interaction
- Speech
- Speech recognition

Radeffect App

- Radeffect App

Lessons learned

- Todo & Advice
- Lessons Learned

Archive

- Previous Groups
- Research Archive

- Skylab Architecture
- Skylab

1. Social interaction

The social interaction creates the interaction of Willy. He listens and speaks with people.

The social interaction is build with NodeJS. NodeJS has a webserver and connects with web sockets to the front-end of Willy. The NodeJS server processes all information of the interactions and passes this to the front-end. The front-end doesn't contains any logic of the social interaction, only the logic of displaying information to the website.

1.1. Repository

Windesheim-Willy/social_interaction

1.2. How to run?

```
./START start
```

1.2.1. Nodejs packages (npm)

Never run `npm install` outside the Docker container to prevent OS conflicts between the Nodejs packages.

Run the npm install command:

```
npm install
```

1.2.2. Stylesheets

The source files for the stylesheet are written in SASS. The should be compiled to CSS files what a browser can process. Gulp compiles the SASS files to CSS files.

Compile only once the SASS file:

```
npm run sass:build
```

Compile the SASS file on every file change:

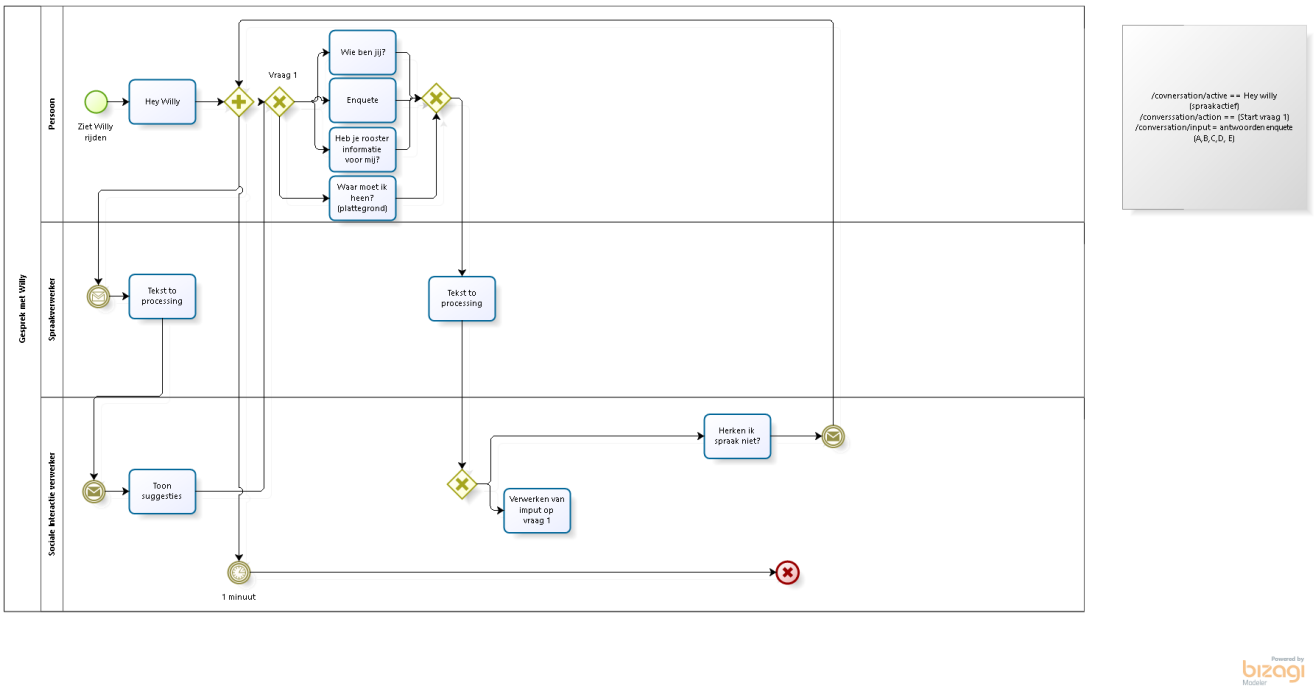
```
npm run sass:watch
```

1.3. Websocket API

API documentation for the connection between de NodeJS server and front-end.

Name	Purpose	Type	Example data
changeMood	Change the body class of the website	String	blue
changeFormat	Change the format of the content and Willy image wrapper	Object	{willy_height: '80%', content_height: '20%'}
changeContent	Change the content below Willy	String	Hi!
textInput	All the text input will be pushed	String	hello willy

1.4. Social interaction Flow



1.5. Interactions of Willy

The interaction has multiple type of interactions.

1.5.1. About Willy

Tell information about Willy and show the image of Willy.

1.5.2. Information with a map

Show a map at the front-end of Willy.

The map requires a image file which will be shown to the user. This file should be located at the path `public/assets/map.png`.

1.5.3. Information with the schedule

Show a image with the schedule at the frontend of Willy.

The schedule requires a image file which will be shown to the user. This file should be located at the path `public/assets/schedule.png`.

1.5.4. Information with the directions

Show a image with the directions at the frontend of Willy.

The map requires a image file which will be shown to the user. This file should be located at the path `public/assets/directions.png`.

1.5.5. Survey

Take a survey with Willy. The user can ask for the survey and Willy will ask the questions.

Survey questions

Survey information file `src/interactions/assets/survey.json`. The information file contains the name, description and author of the survey.

Survey questions file `src/interactions/assets/survey.csv`. The questions file contains all the questions for the survey.

Survey answers

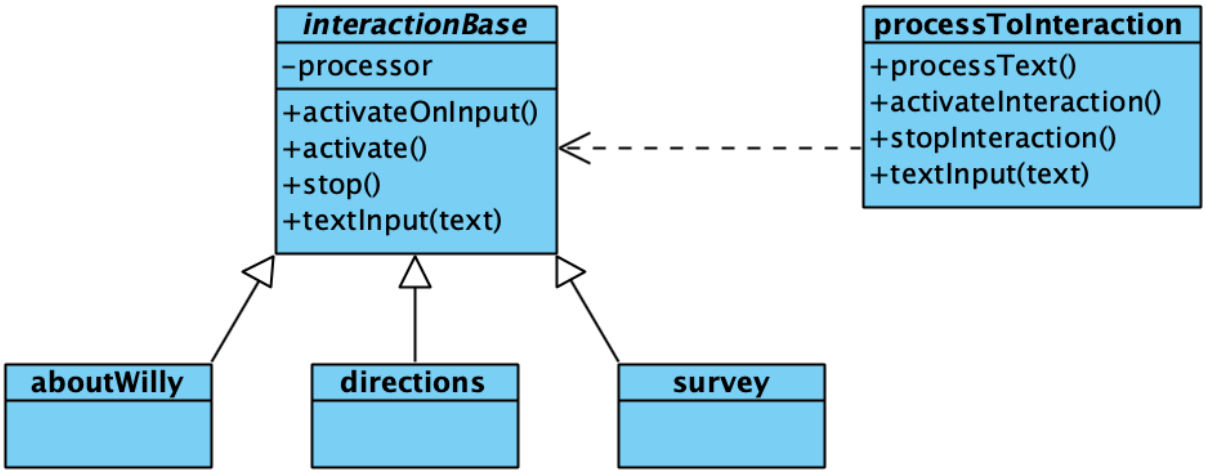
Survey answers `src/interactions/assets/survey_answers.csv`.

1.6. ROSNodeJs

Read the information about the ROSNodeJs

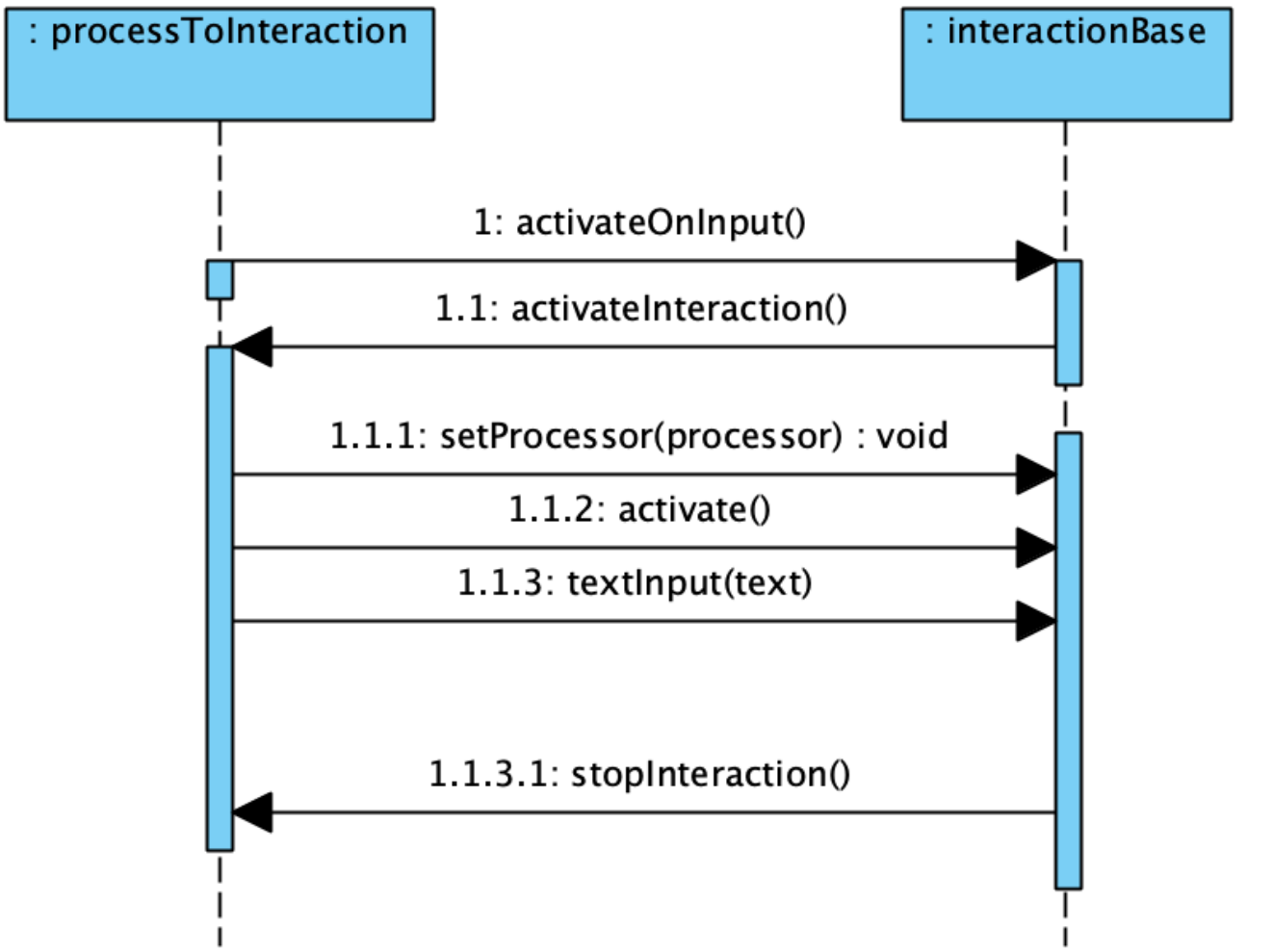
1.7. Diagrams interactions

Class diagram:



NodeJS has no interfaces, so the application connects to a instance of the interaction base.

Sequence diagram:



1.8. Screen Designs

1.8.1. English

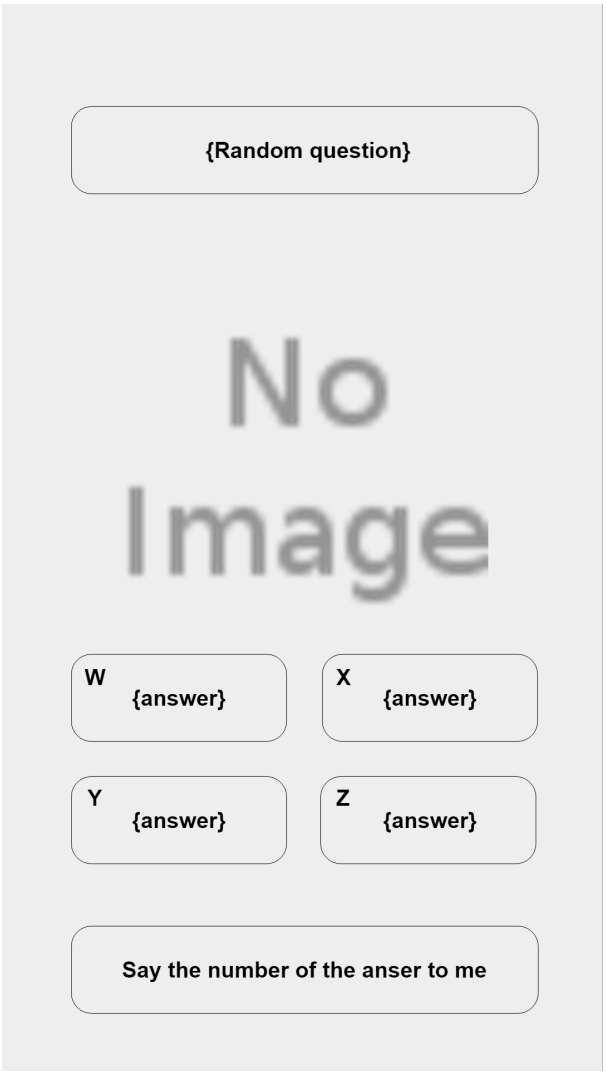
1. This is the screen shown when Willy is driving around, and there is not yet a person to interact with.

2. This screen is shown when a person is starting to interact with Willy. At the bottom there are two arroys pointing to the garbage bins.

3. After that Willy asks permission to ask a question to the person in front of Willy.

4. When the person says 'no', then this screen is shown, after wich the robot continues driving.

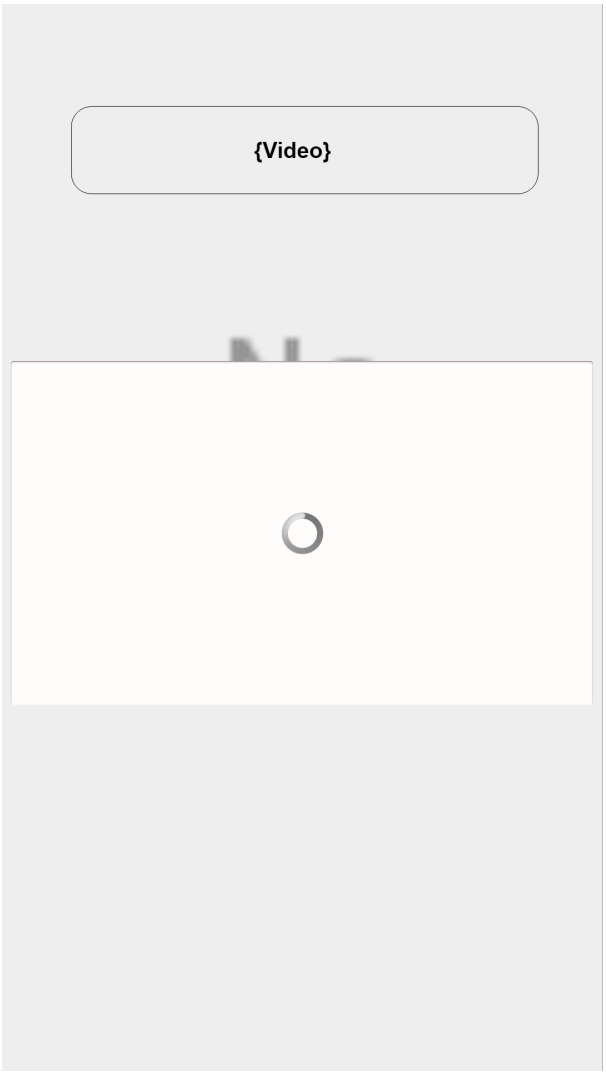
5. When the person says 'yes', a random question is shown with a number of multiple choise answers.



6. When the given answer is wrong, this screen is shown.

7. When the given answer is right, this screen is shown.

8. After the question, Willy starts to play an informative video about garbage.



1.8.2. Dutch

For comments at each image, see the English version above.

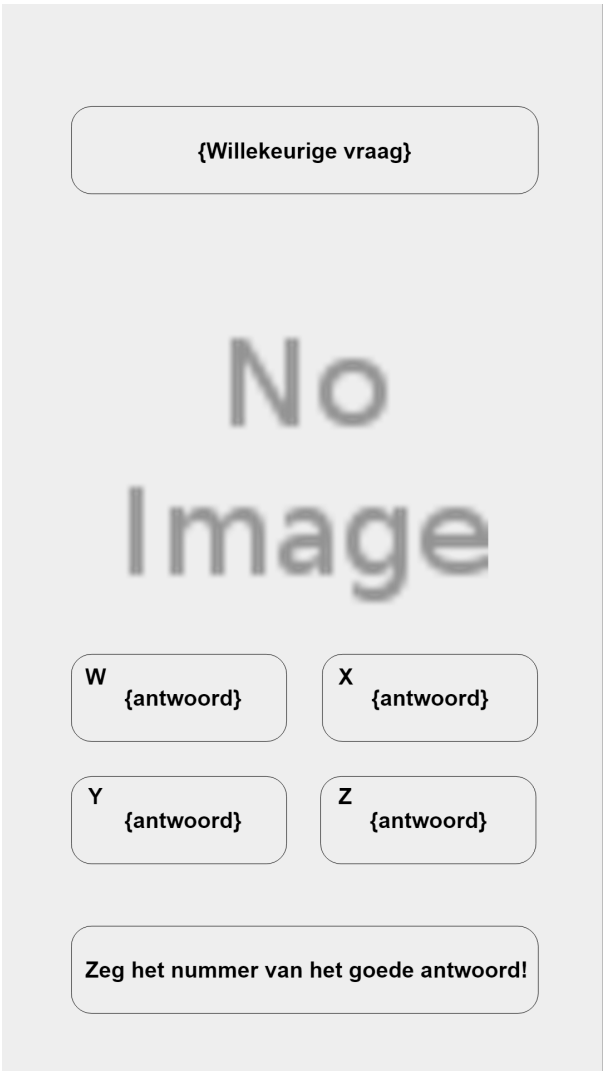
1.

2.

3.

4.

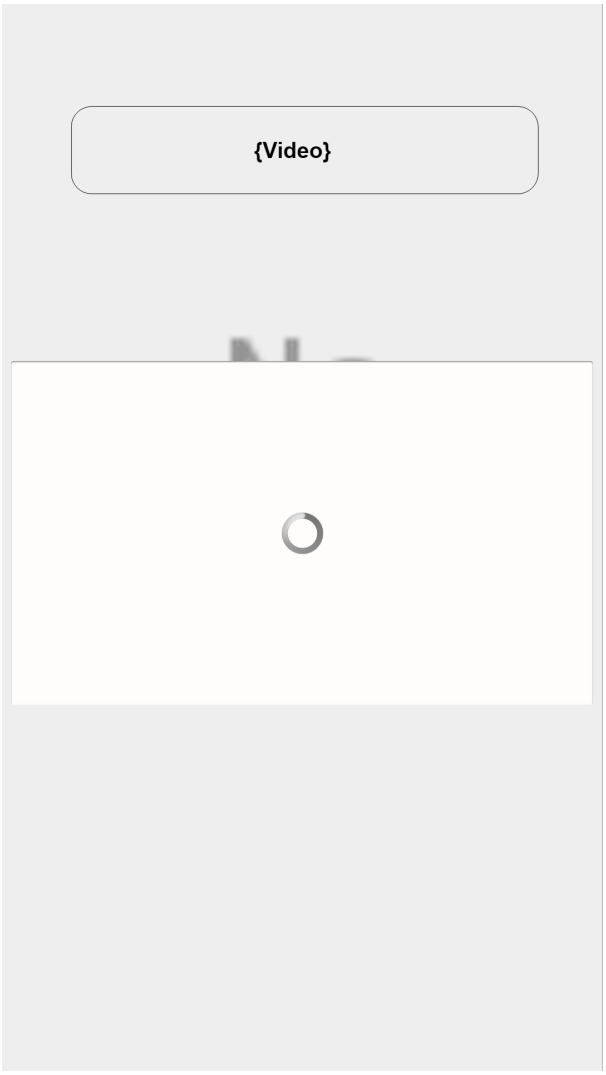
5.


{Willekeurige vraag}
No Image
W {antwoord}
X {antwoord}
Y {antwoord}
Z {antwoord}
Zeg het nummer van het goede antwoord!

6.

7.

8.


{Video}