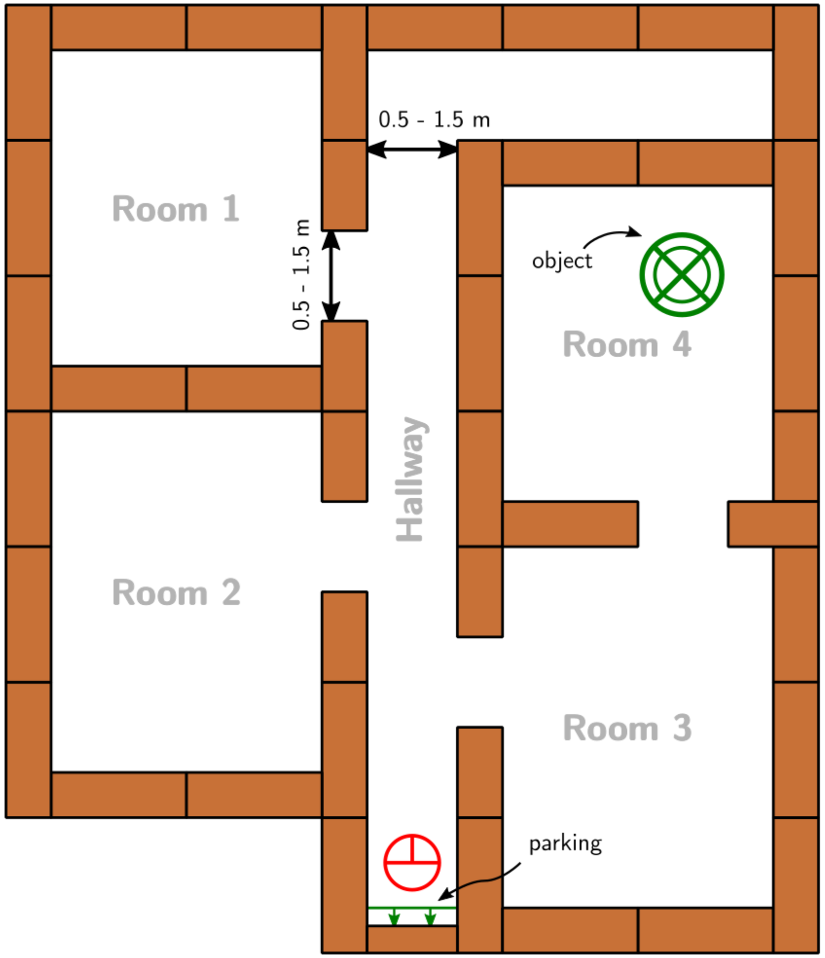
Technische Universiteit
Eindhoven
University of Technology

Final Presentation: Group 6

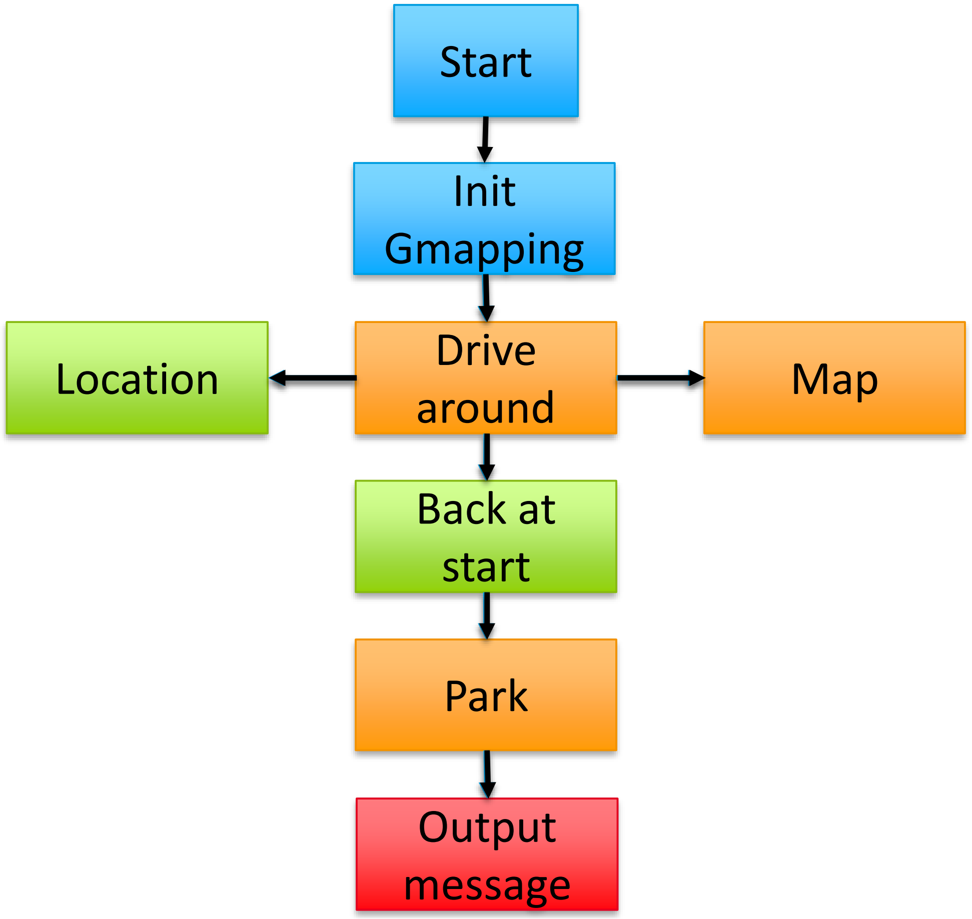
Thomas Bosman
Raaf Bartelds
Bas Scheepens
Josja Geijsberts
Rokesh Gajapathy
Tim Albu

Hospital Challenge: Three main parts

- Map the environment
- Park backwards
- Find the object



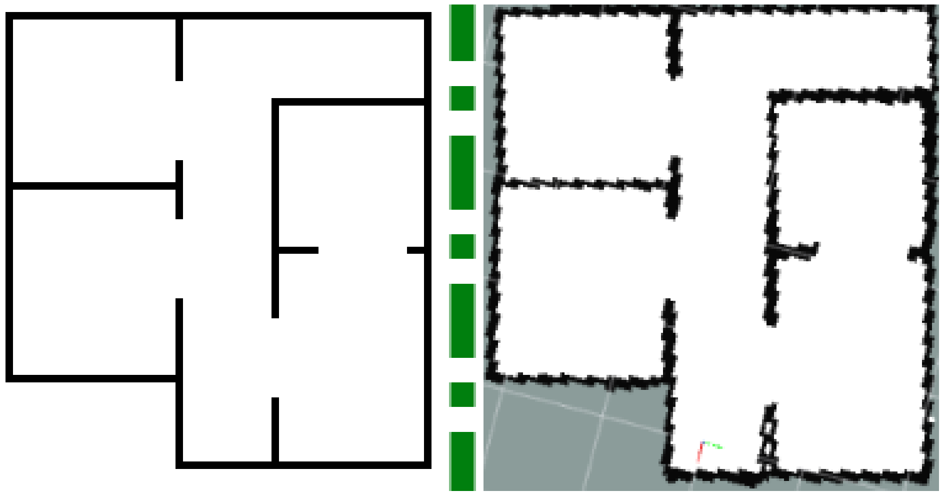
Mapping the environment



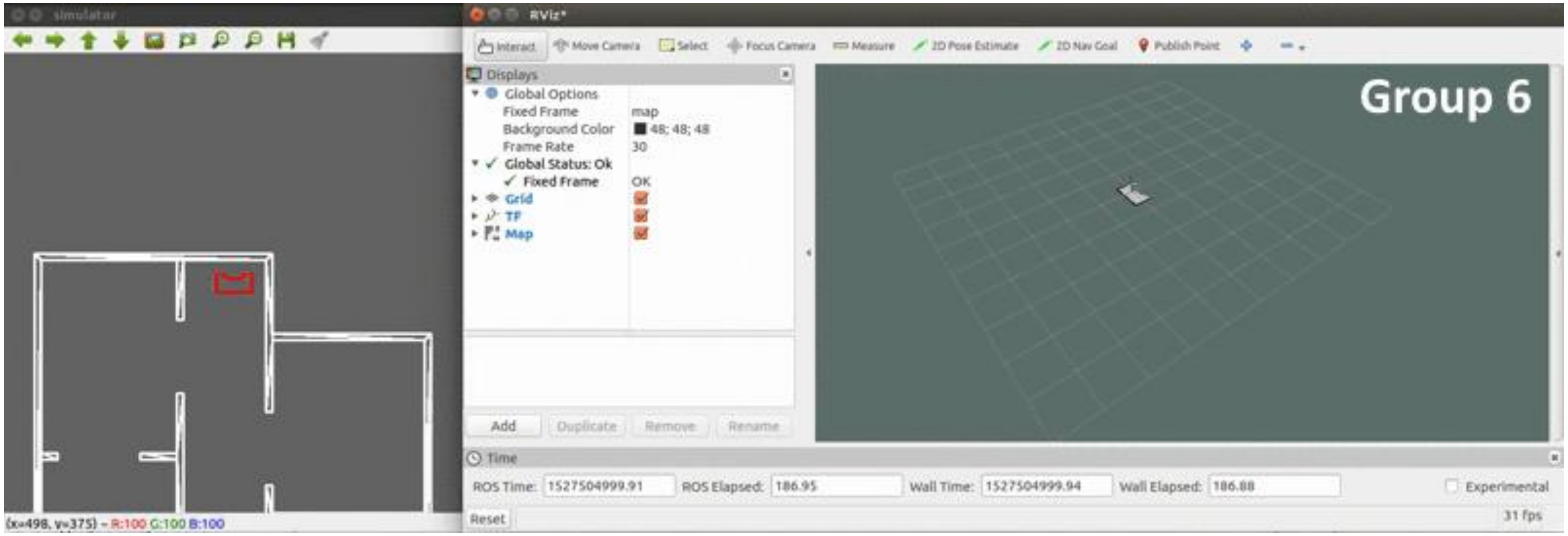
Mapping: gMapping

Why gMapping:
- Available on openSLAM
- Lot of functionalities
- Support available
- Output is a matrix

What is gMapping:
- SLAM (Simultaneous Localization And Mapping) algorithm
- Particle filter



gMapping



Global Options
Fixed Frame: map
Background Color: 48; 48; 48
Frame Rate: 30
Global Status: Ok
Fixed Frame: OK

ROS Time: 1527504999.91 ROS Elapsed: 186.95 Wall Time: 1527504999.94 Wall Elapsed: 186.88

(x=498, y=375) ~ R:100 G:100 B:100

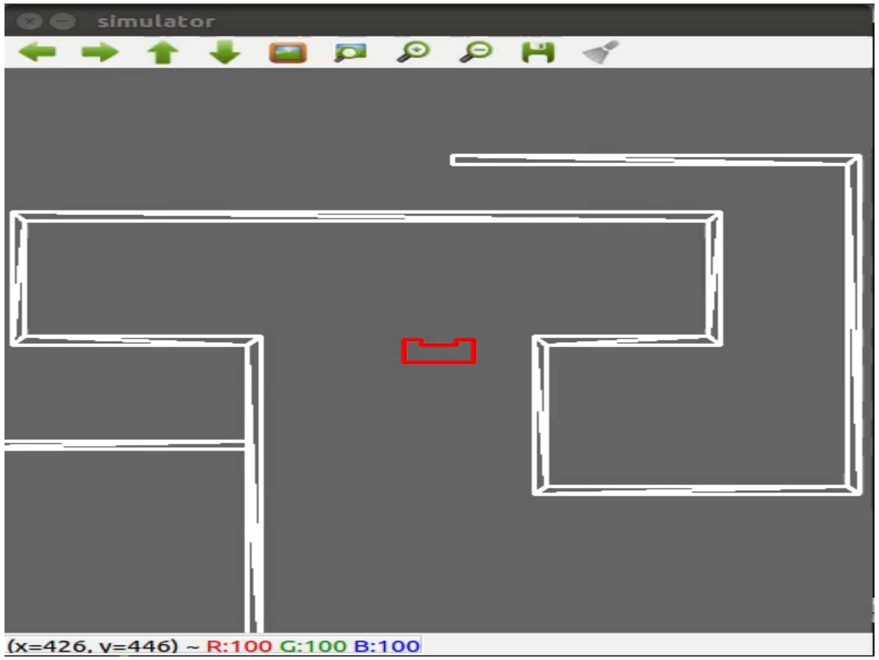
Driving around

Two possibilities:

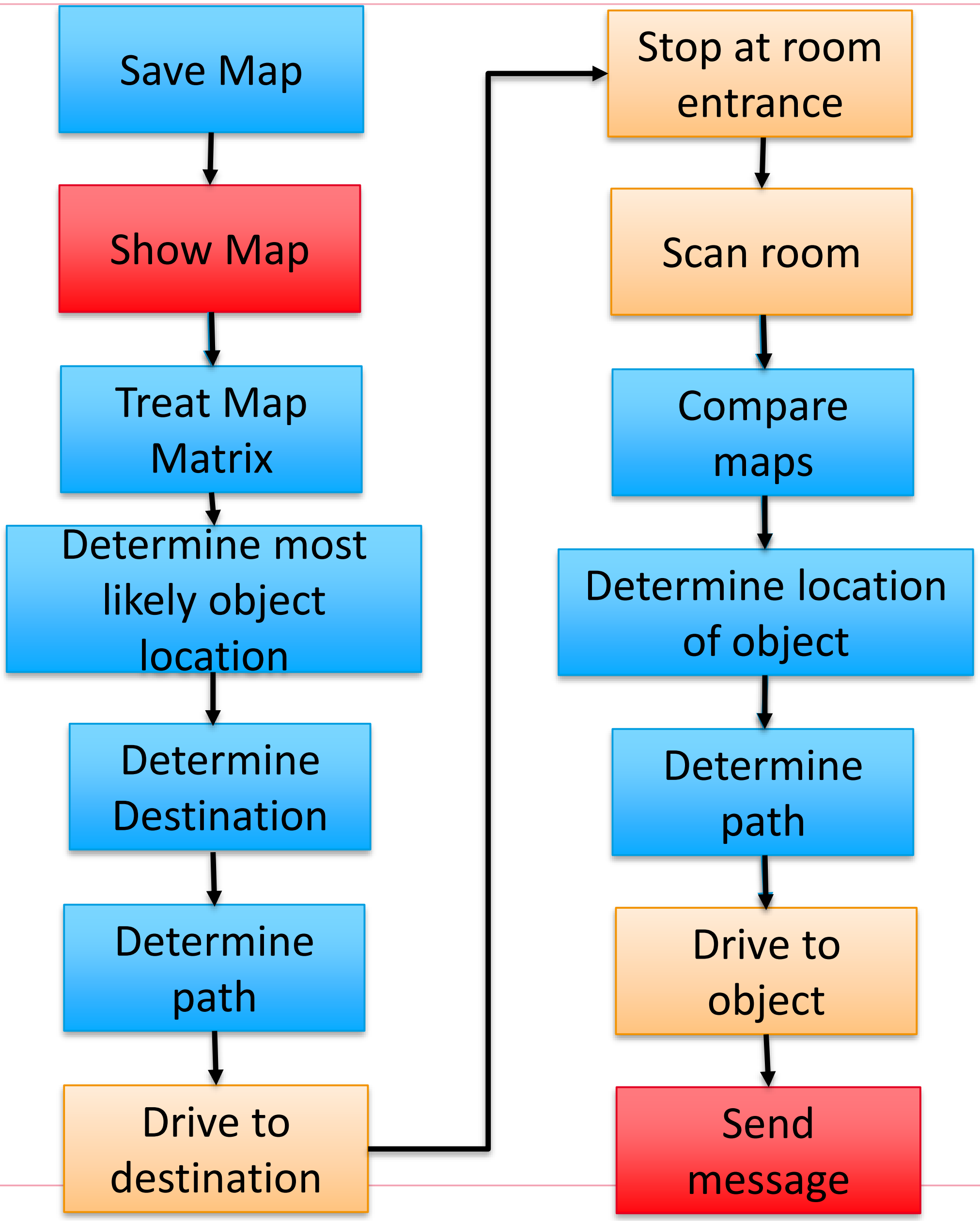
- Advanced solution: Path planning to unknown space
    - Can go wrong is space between walls is detected
    - Faster solution
- Simple solution: Use Wall Follower used in Escape room challenge to go around and map:
    - Robust
    - Slow

Parking Backwards

- Driving until fixed distance from the wall
- Steps:
    - Turn around 180^o
    - Drive backwards fixed distance



Finding the object



Matrix Treatment

Matrix is output of gMapping and represents the map:

- Variable Size: 400x400 with resolution of 5 cm
- Choice of grid size: Resolution and Matrix size trade-off

Semantics:

- Recognize corners
- Recognize rooms
- Recognize doors

World Model

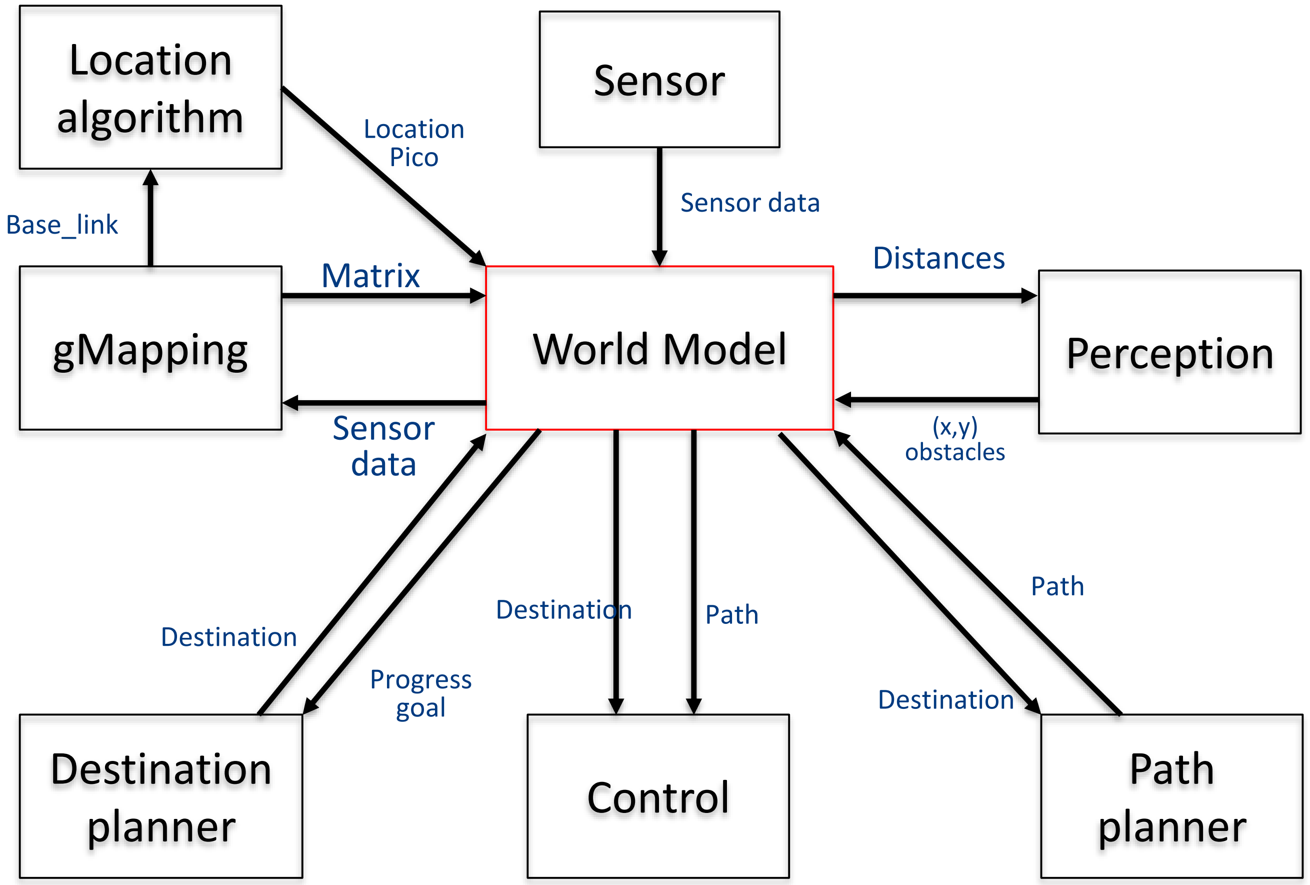
What is inside the world model:

- Sensor data
- Obstacle location
- Map matrix
- Destination
- Location of PICO

Hypothesis:

- Room where the object is
- Location of the object

Functions and interactions



Overview of the algorithms

Location:

- Function: Get current position of PICO with respect to starting position
- Components: TF listener to base_link
- Input: References and transformation tree from GMapping
- Output: pair of double
- Use:
    - Localization: Back at start, at object , etc…
    - Driving

Plan: Destination planner

- Function: Determine where PICO should move to
- Input: Stages, goal
- Output: pair of double -> Destination of PICO
- Use:
    - Direction: Go to room x, Go to object

Overview of the algorithms

Plan: Path planner

- Function: Create a path to required location using Dijkstra
- Input: Location from Destination planner, Current location
- Output: array of doubles
- Use:
    - Localization: Back at start, at object , etc…
    - Driving

Control:

- Function: Follow the path
- Input: Array of locations
- Output: Actuator outputs
- Use:
    - Send out motor commands to follow the path

Testing and implementation

- What has been tested:
  - Mapping
  - Parking
- What is going to be tested and when:
  - Thursday: Path planning and control together
  - Friday: Matrix treatment and object finding
  - Monday: Simulation of the final challenge