EMC 2019
Tooling and Infrastructure

Bob Hendrikx

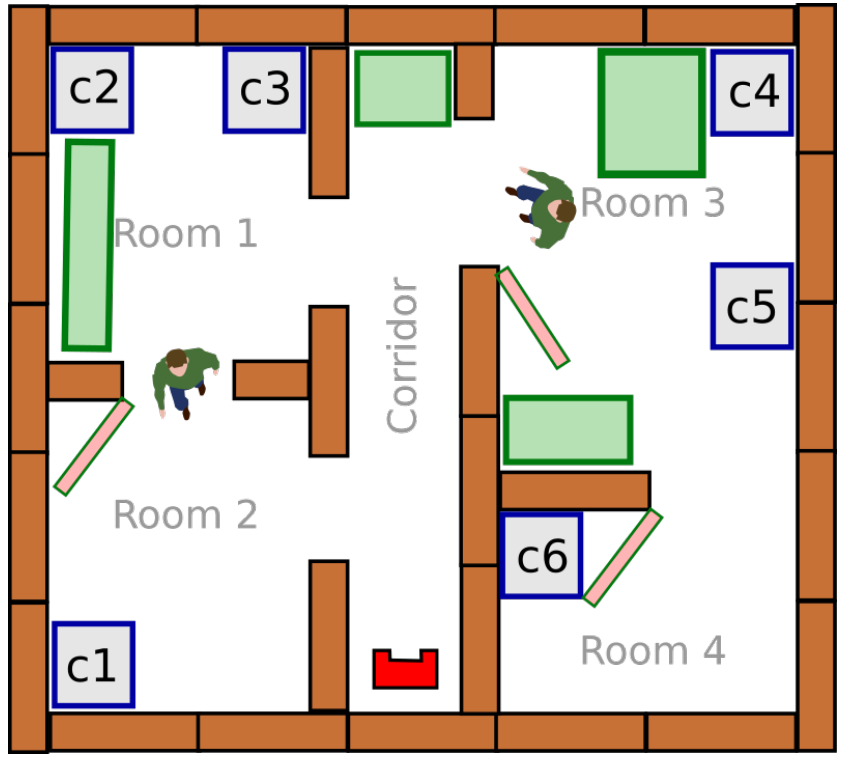
Eindhoven University of Technology
Department of Mechanical Engineering

April 24, 2019

The Assignment

Final Competition: Bring items to cabinets in a dynamic hospital environment, of which a map is provided

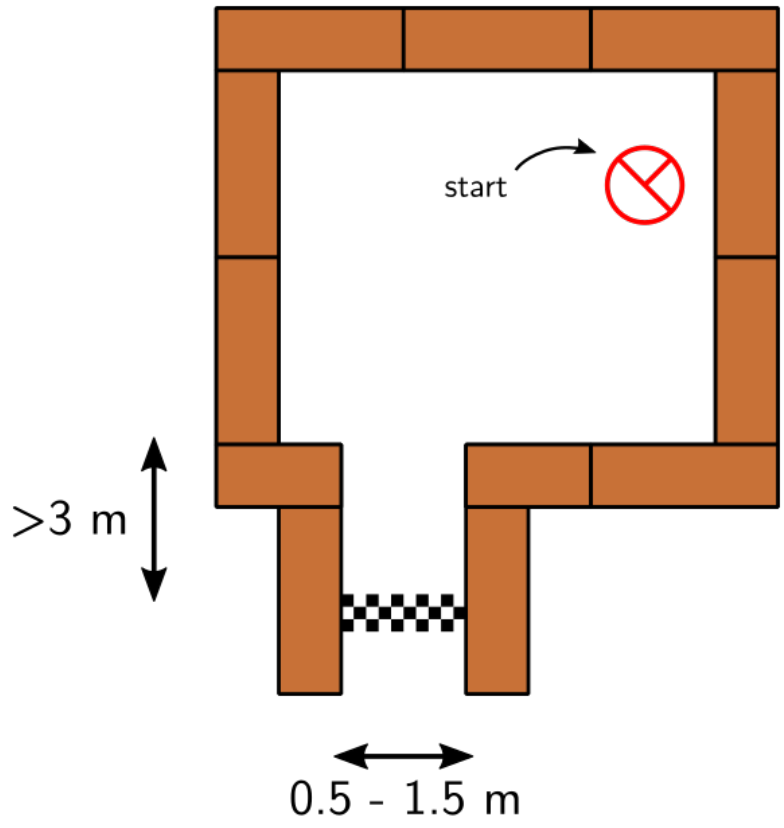
- You have to:
  - visit an unknown number of cabinets as fast as possible
- You can use:
  - The Laser Range Finder to detect walls and doors
  - The encoder data from the wheels
  - The control effort signal to notice touches
- Important Dates:
  - Final Presentations: June 5
  - Competition Day: June 12



Intermediate Assignment

Escape Room Competition: let a robot escape the room through the door.

- You have to:
  - try to be as fast as possible
- You can use:
  - The Laser Range Finder to detect walls
  - The encoder data from the wheels
  - The control effort signal to notice touches
- Competition day: May 15



Simple, right?

Don't worry, we supply you with some tools to get you started!

Introducing the Robot: PICO

- PICO is the robot you have to use!
- Telepresence Robot from Aldebaran
  - Robot type: Jazz
- Sensors:
  - Laser Range Finder (LRF)
  - Wheel encoders (odometry)
  - 170° wide-angle camera
- Actuators:
  - Holonomic base (omni-wheels)
  - Pan-tilt unit for head
- Computer:
  - Intel I7
  - Running Ubuntu 16.04

ROS

- Robot Operating System
  - Open-source meta-operating system for robots
- Won't be using it!
- Instead, we will provide our own 'software layer'
  - It is simpler to understand, and 'cleaner' to use
- However, you are still allowed to use ROS!

Ubuntu

Development of PICO's software will be done in Ubuntu.

- Linux-based operating system
- Use version **16.04** (not 14.10, 15 or 17!)
- 32- and 64-bit (64-bit recommended)
- Easy dual boot installation with *e.g.*, Windows
- Download: see tutorial!
  - Any problems? → Check the wiki.
  - No info? → Send us an email.

C++

- We will use C++ as programming language
- C++ is object-oriented C
  - "C with Classes"
  - Encapsulate data and functionality within objects
- It is a powerful but complex programming language.
- However, we provide you the EMC framework to get you started

Creating code: Qt Creator

- Integrated Development Environment
  - Advanced code editor
- Many advantages over 'simple editors':
  - Syntax highlighting
  - Code completion
  - Visual compiler feedback
  - Static code checking
  - Refactoring tools
  - Parenthesis matching
  - ...

Git Version Control

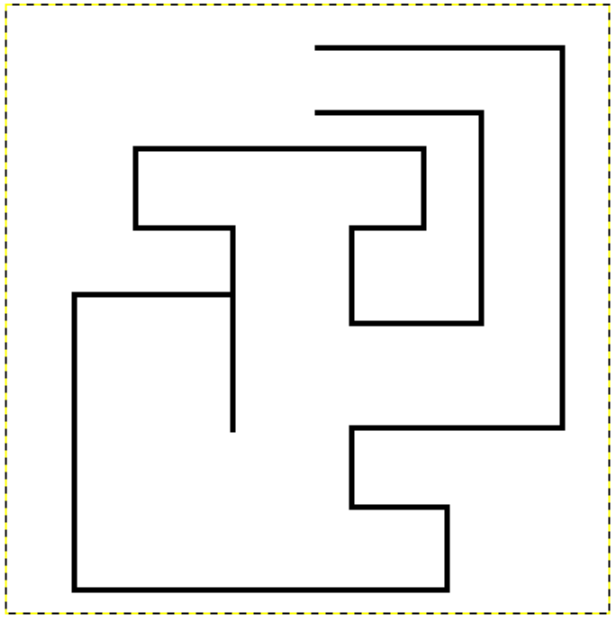
- Version Control System:
  - *'Manages files and directories, and the changes made to them, over time'*
- Used to store and maintain your code on the server
  - (Like Dropbox)
- Maintains version history
- Is distributed
  - You always have the full history on your pc
  - You can always go back to a version, show differences, even when off-line
- More info on the Wiki

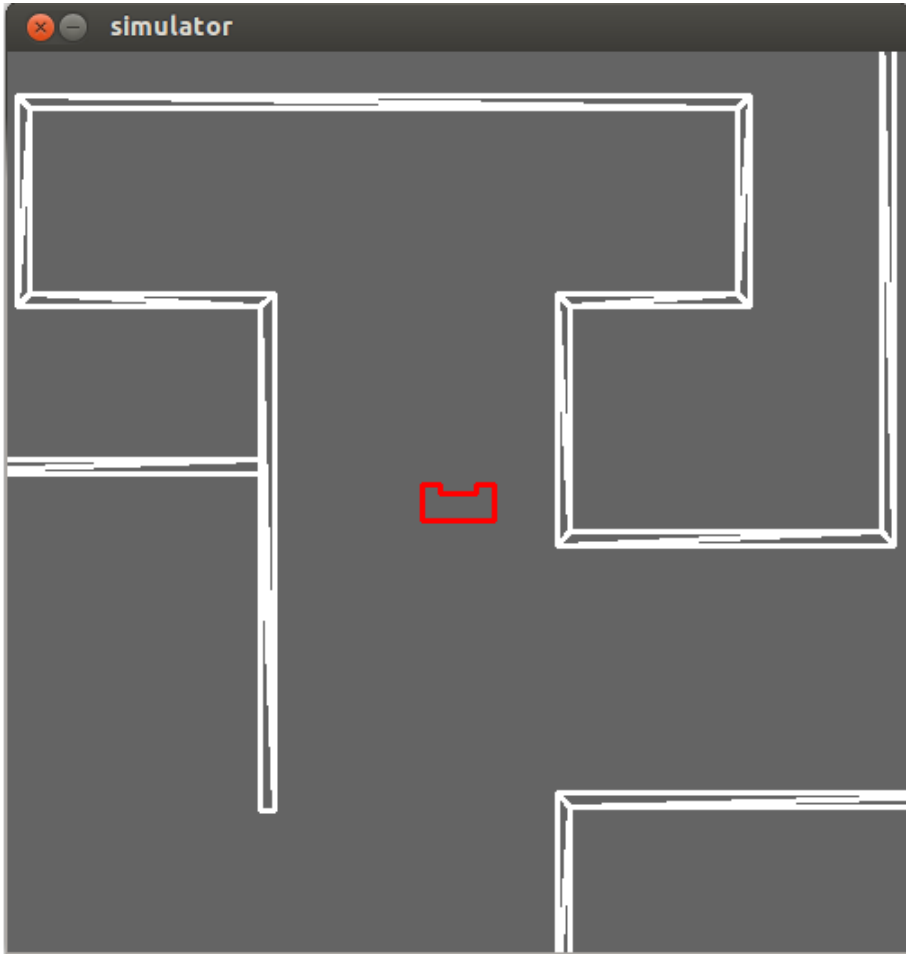
PICO Simulator

- You will have to work with the real robot, but we only have one. Therefore:
- Test time is limited and has to be scheduled, see Wiki!
- PICO Simulator:
  - Simulates:
    - Sensors (Laser, odometry)
    - Actuators (Base)
    - Environment (maze)
- Can easily create test environments using height maps
- Integrates well with our provided software
  - If your software runs in the simulator, it runs on the robot
  - This does not guarantee that it will also work...

You still need to test on the real system!

PICO Simulator





Example

- Full Example: from requirements, through Task-Skill-Motion to Software Executable.
- (far) from perfect!
- Focus on decoupling parts of functionality, explicitly in the code.

Wiki

- EMC Wiki:
  - http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control_2019
  - Info on practical assignment, installation, getting started
  - Frequently Asked Questions
  - Log-in: student account
- Group pages on EMC Wiki:
  - Each group gets its own page
  - Update at least weekly
- Overall use:
  - Everyone can edit
  - If you see a mistake: correct it
  - If you had a problem and know how to fix it: add it

Recap

- Robot: PICO
- OS: Ubuntu 16.04
- Programming language: C++
- Code editor: Qt Creator
- Version control: git
- Simulation: PICO simulator
- Documentation: Wiki

That should get you started!

Groups

Each group will be supervised by a tutor:

1. Wouter Kuipers
2. Wouter Houtman
3. Jordy Senden
4. Bob Hendrikx
5. Hao Liang Chen
6. Marzieh Dolatabadi
7. Yanick Douven
8. René & Herman
9. Bob Hendrikx

It is *your* responsibility to get in touch with your tutor (see Wiki)

What should I do now?

- Make your own groups of max. 5 people
  - If complete, add your group details to the wiki.
- Send an email to your tutor:
  - to schedule the first meeting,
  - with one username for access to your Git, (tutorial)
- With your group:
  - schedule test-time with PICO starting 6th of May, see test scheme on Wiki!
- Check the Wiki & Finish the Tutorials:
  - `http://cstwiki.wtb.tue.nl/index.php?title=Embedded_Motion_Control`