Mobile Robot Programming for Education

Jean-François Lalonde, Christopher Bartley, and Illah Nourbakhsh

The Robotics Institute
Carnegie Mellon University

Plan

- Mobile Robot Programming Laboratory class
  - Course overview
- Robotics concepts
  - Fundamental
  - Advanced
- Educational concepts
  - Evolution over 11 years

Plan

- **Mobile Robot Programming Laboratory class**
  - **Course overview**
- Robotics concepts
  - Fundamental
  - Advanced
- Educational concepts
  - Evolution over 11 years

Course description

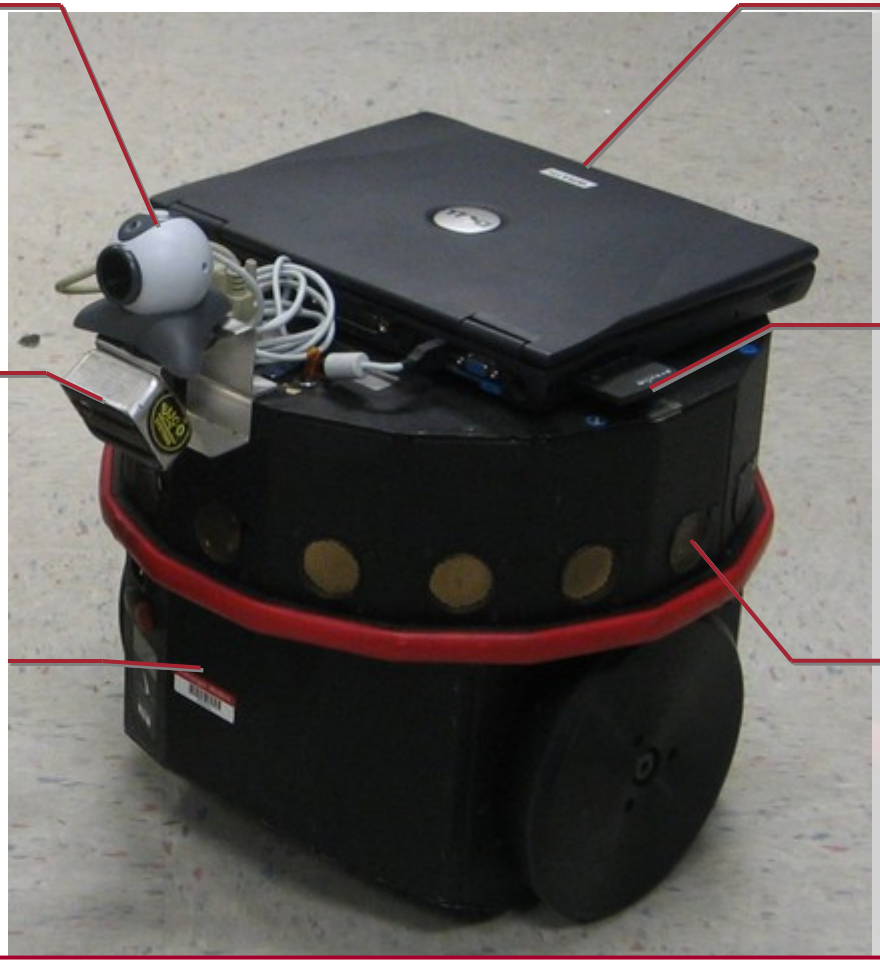
- Mobile Robot Programming Laboratory
  - 11 years at Carnegie Mellon
  - Undergraduates & Graduates
  - ~30 students, 2 TAs
- Goal
  - Students learn how to **program** robots!
  - Weekly assignments
    - Increasingly difficult
    - Build on previous

Platform

USB camera

Magnet

Nomad Scout
Differential drive
robot



Dell Laptop
Java 1.4.2
Windows XP

Wireless
network adapter

16 sonars for
localization

Students can focus on *programming*

Maze navigation

Curriculum

1. Introduction
2. Feedback and reactive control
3. Sensor interpretation
4. Abstract action
5. Executing plans
6. Planning
7. Programming architectures
8. Single-robot game playing
9. Cooperation
10. Final competition

- PID controller
- Basic actions
  WhatdoISee, GotoNextNode
- Execute a given sequence of actions
- Generate a sequence of actions
- Introduce the game
- Multiplayer game

Plan

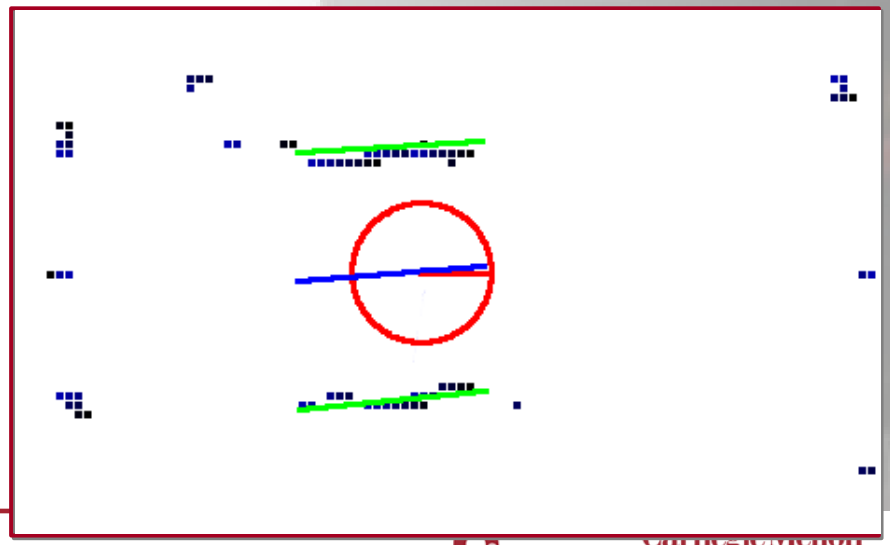
- Mobile Robot Programming Laboratory class
  - Course overview
- **Robotics concepts**
  - **Fundamental**
  - **Advanced**
- Educational concepts
  - Evolution over 11 years

Robotics: Fundamental concepts

- Perception
  - Sonars: localization in maze
  - Camera: lighting-insensitive color detection
- Action
  - PI/PID controllers
  - Movement in the maze
- Cognition
  - Planning
    - DFS, BFS, AND-OR, etc.

Robotics: Advanced concepts

- Robot observability
  - Degree to which outside observer can identify the evolution of the internal state of a robot
    - Audio: speech synthesizer
    - Visual: graphical display, logging mechanism
  - All teams use at least 1 form of interface
  - 86% of students → very useful
- Others
  - Control architectures
  - Reinforcement learning
  - Multi-robot coordination



Plan

- Mobile Robot Programming Laboratory class
  - Course overview
- Robotics concepts
  - Fundamental
  - Advanced
- **Educational concepts**
  - **Evolution over 11 years**

Evolution – Hardware

Year	Hardware
1-2	Nomadic Tech. Serial 1 & 2
3-6	Nomad 150
7-11	Nomad Scout

3-wheels synch, infrared
+ Wheels turn independently of body
- Infrared sensors

3-wheels synch, sonars
+ Independent sensor turret
+ Higher DOF
- Large size

Diff-drive, sonars
+ Smaller size
- Lower DOF

Higher DOF → Higher number of possible solutions

Evolution – Programming environment

Year	Programming environment
1-2	LISP on Macintosh
3-4	LISP on Windows
5-7	C/C++ on Windows
8-11	Java on Windows

LISP
+ Command-line debugger
+ Diagnostic tool for code fragments
- No IDE under Windows

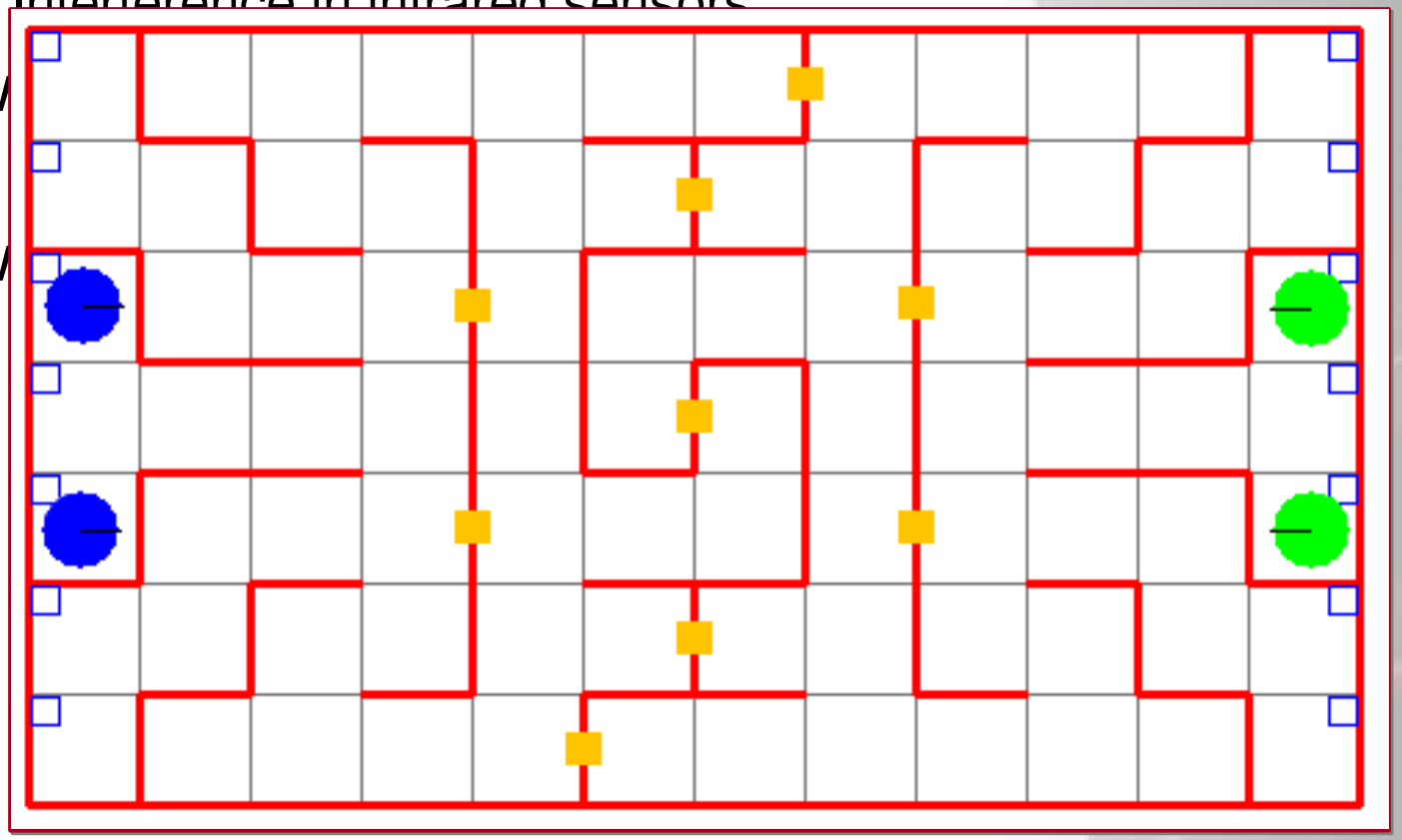
C/C++
+ Popular
- Memory/pointers problems
- Steep learning curve

Java
+ No memory problems
+ Easy graphical interface
+ Well documented

Need readily available, fast debugging tools

Evolution – Final challenge

- One-on-one in shared maze
  - Interference in infrared sensors
- Tw
  -
- Tw
  -



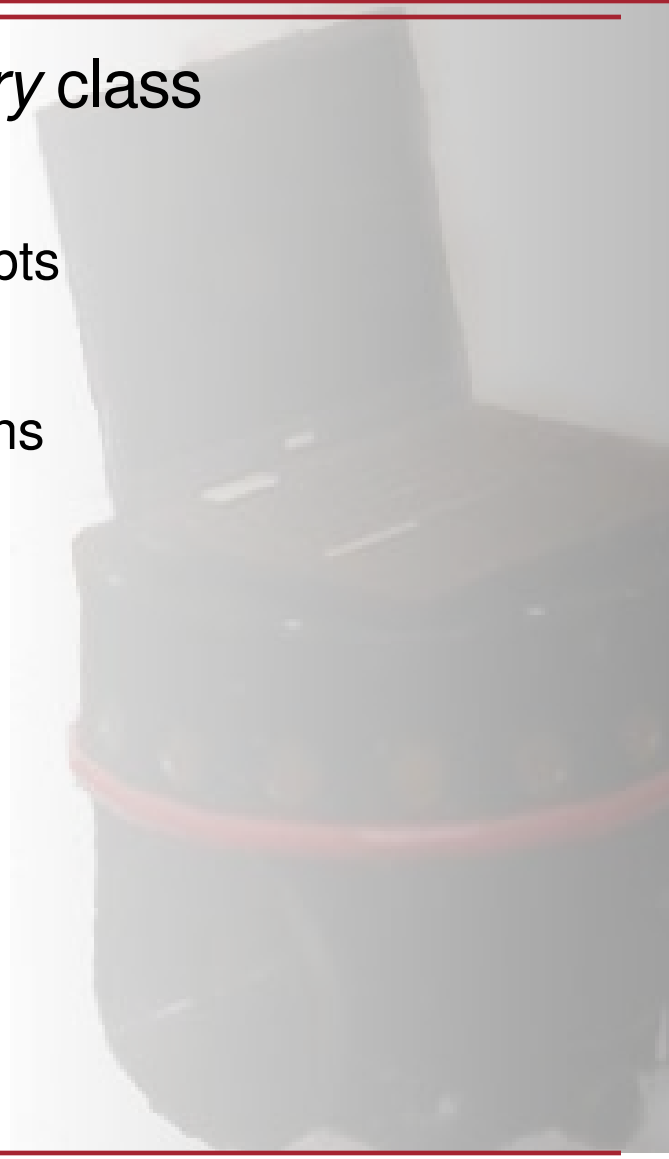
Must be challenging, but feasible

Evolution – Teamwork

- Great teamwork opportunity
- 3 members is the best
  - > 3: splits within teams, members left out
  - < 3: not enough to complete tasks
- Same-gender teams are more efficient
  - Students also feel that way

Conclusion

- *Mobile Robot Programming Laboratory* class
- Students learn
  - Fundamental & advanced robotics concepts
- Important points
  - Enough flexibility to allow creative solutions
  - Readily available debugging tools
  - Challenging but reachable problems
  - Small, well-balanced teams work best



Thank you!

- Any questions?