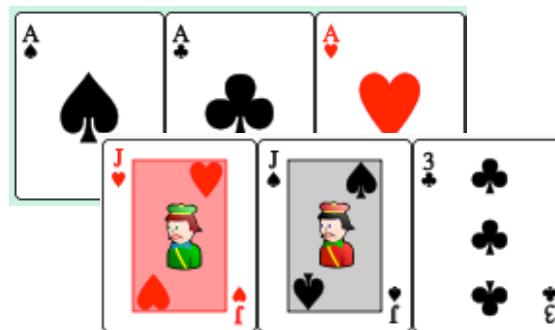
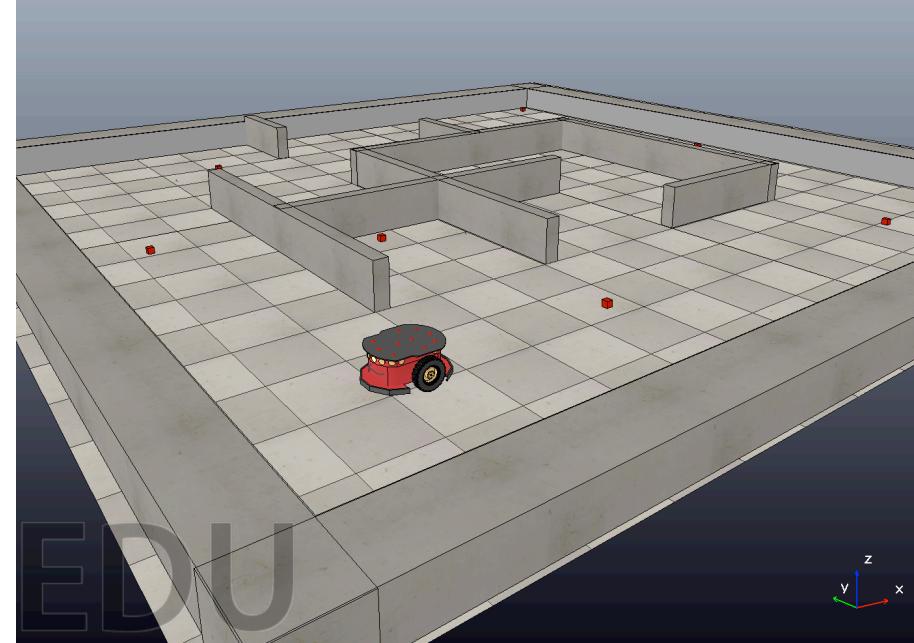


# Artificial Intelligence Lab I: Agents

DT8012 Halmstad University  
November 2018

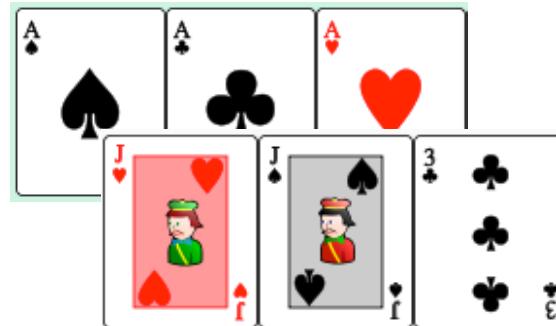
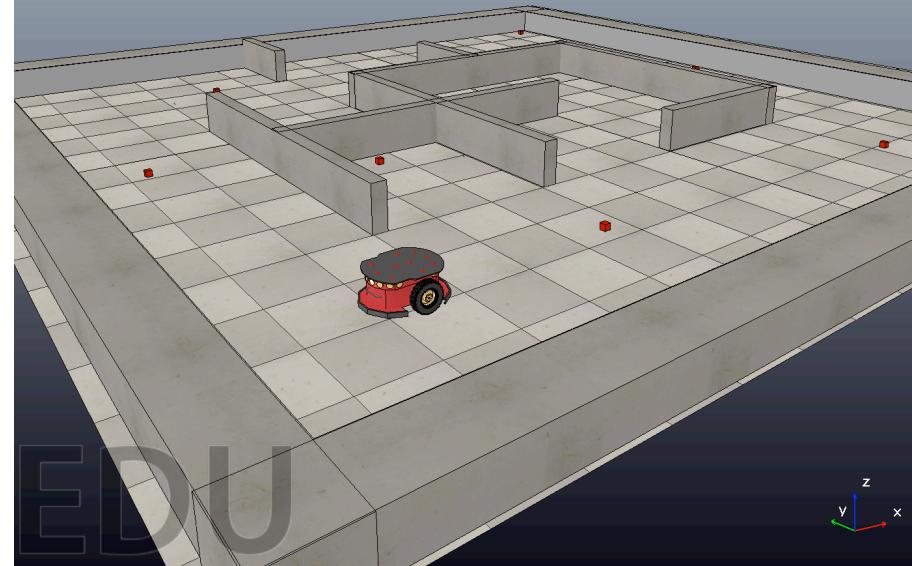
# Lab I

- Mobile robot
  - Collect Energy
- Poker game player
  - Win more games and money



# Lab I

- Random agent
- Fixed agent
- Reflex agent
- Agent with memory

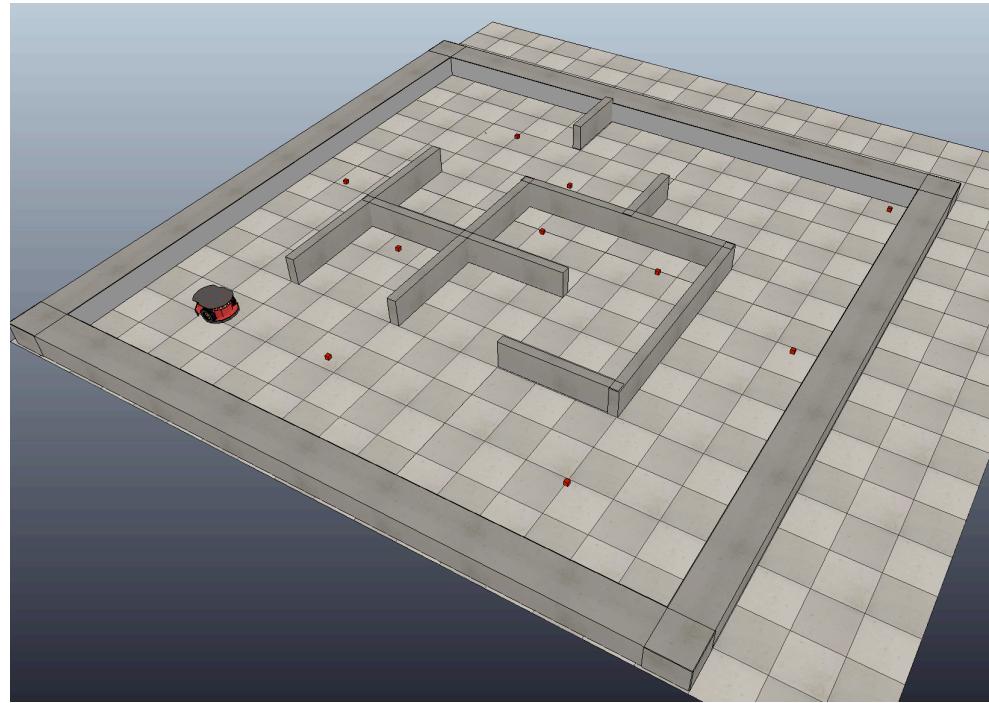


# Interesting videos

- Atlas:  
<https://www.youtube.com/watch?v=rVlhMGQgDkY>
- Machine Learning for video games:  
<https://www.youtube.com/watch?v=qv6UVQOQ0F44>
- Google self-driving cars:  
<https://www.youtube.com/watch?v=TsaES--OTzM>
- Mars rover - curiosity

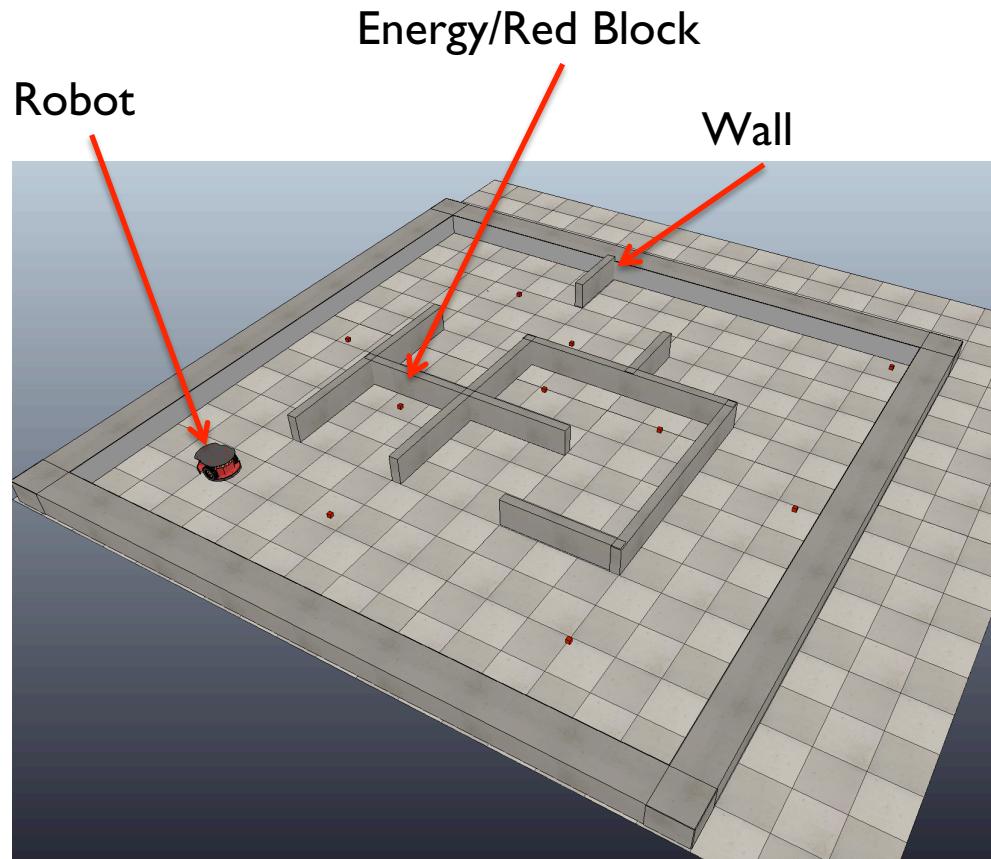
# Lab I

- Mobile robot
  - Pioneer P3-DX
  - Collect energy



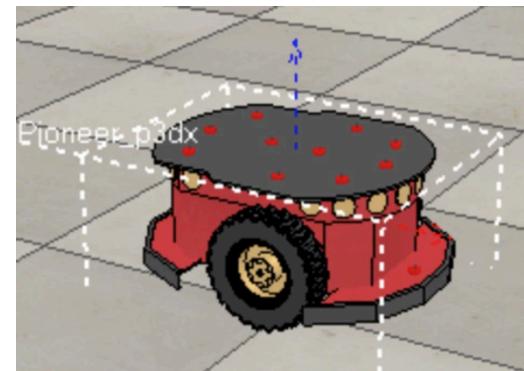
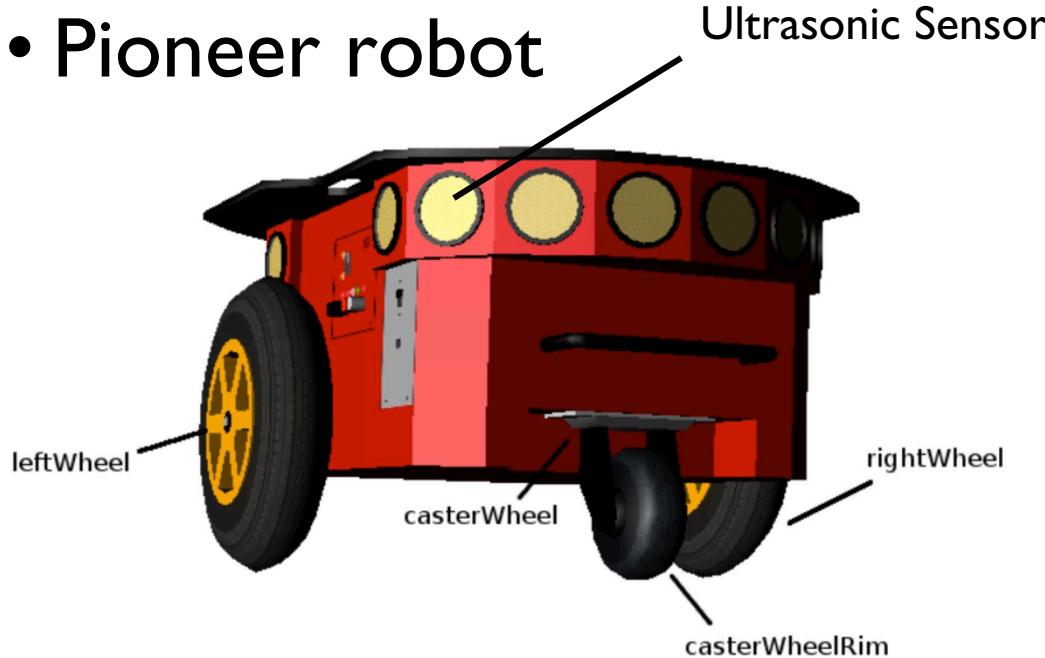
# Lab I

- Mobile robot
  - Pioneer P3-DX
  - Collect energy



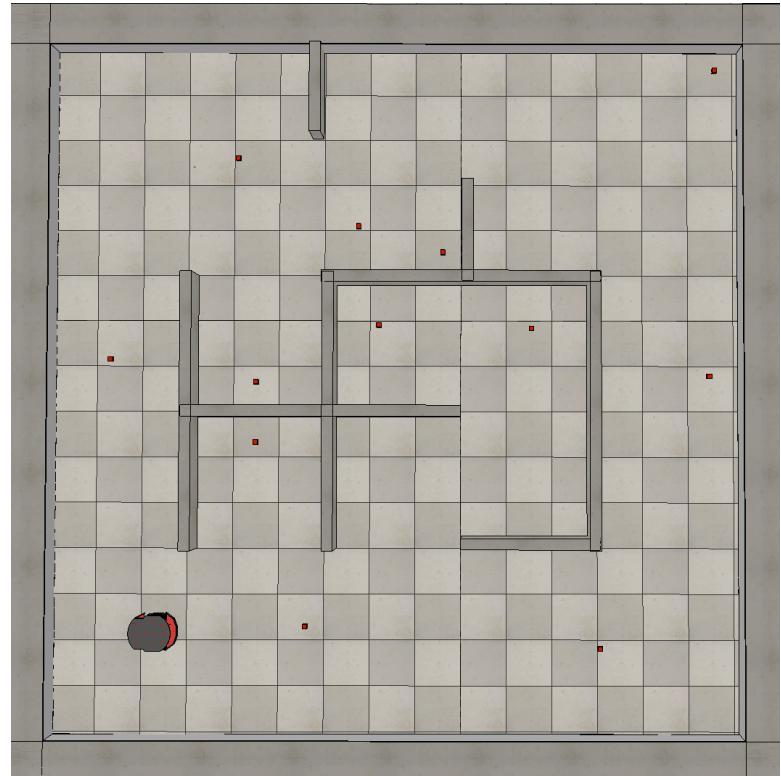
# Pioneer P3-DX

- Pioneer robot



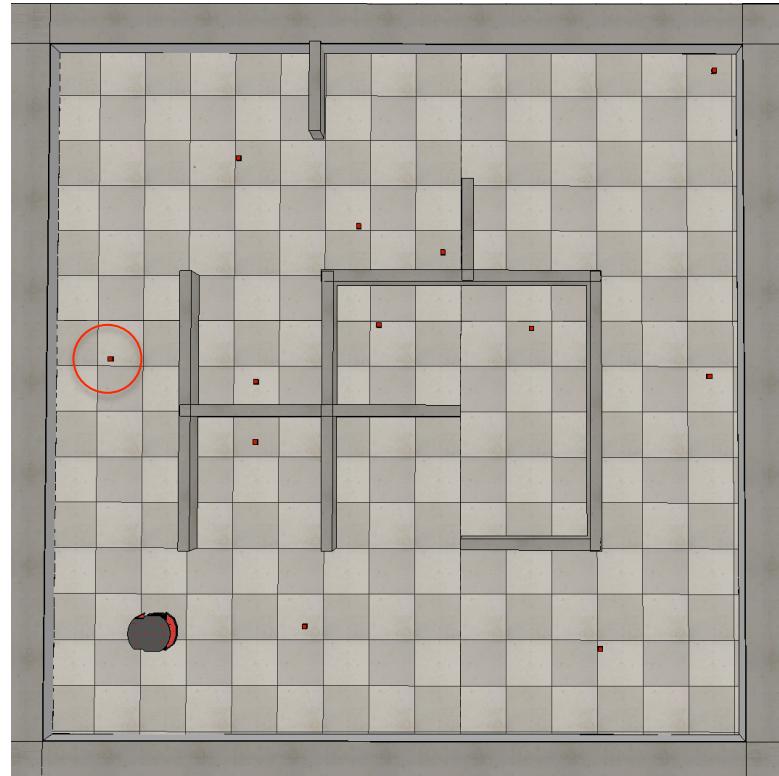
# What can robot do?

- Move from Point A to B
- Collect energy
- Avoid obstacles (walls)

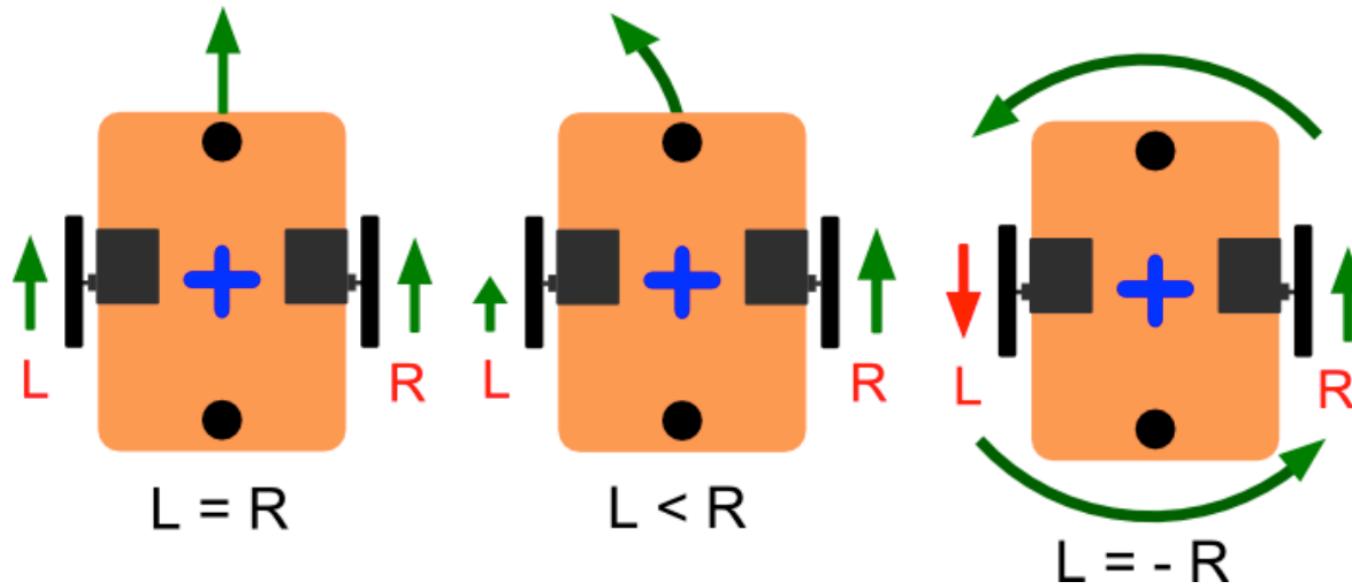


# What can robot do?

- Move from Point A to B
  - Turn & march forward
  - Spline
- Collect red box
  - Energy sensor
- Avoid obstacles (walls)
  - Ultrasonic sensor

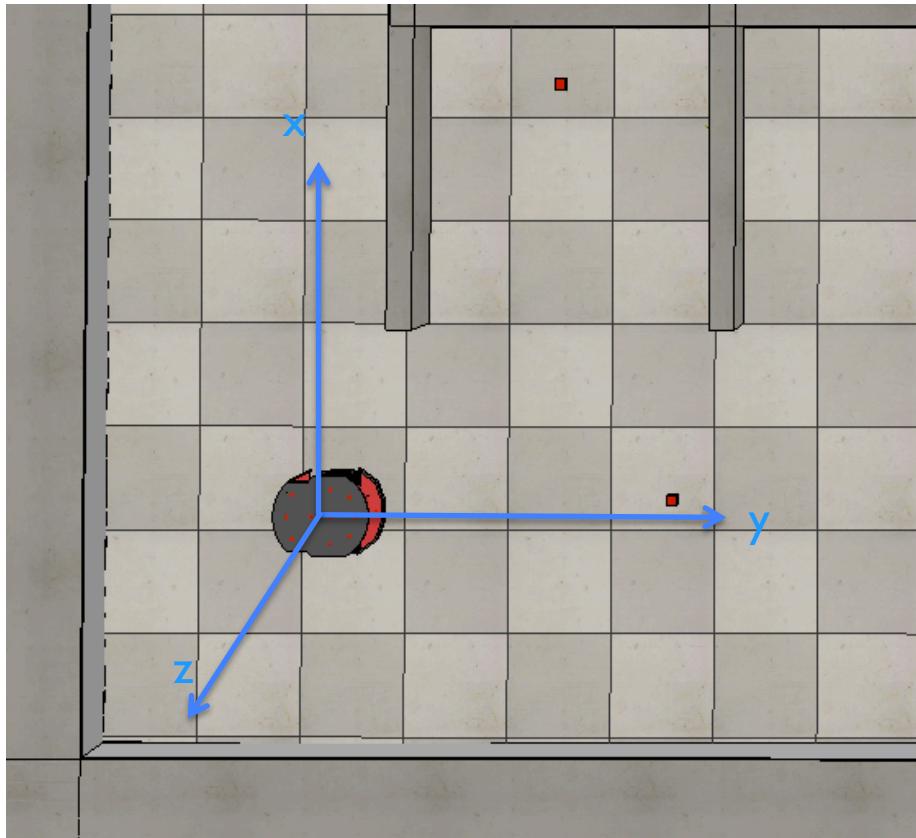


# Pioneer P3-DX: Differential Drive



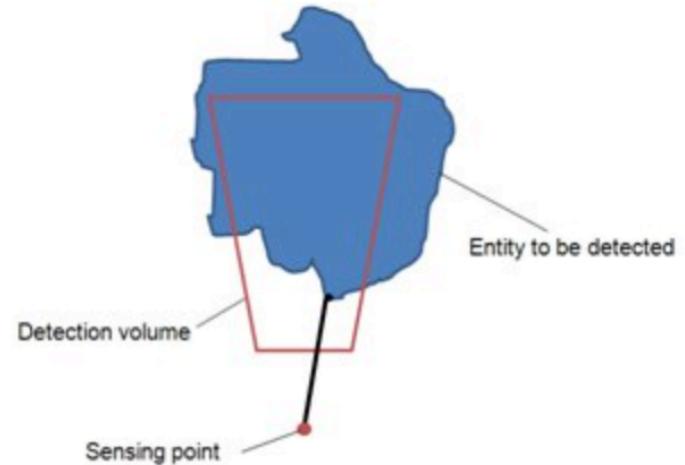
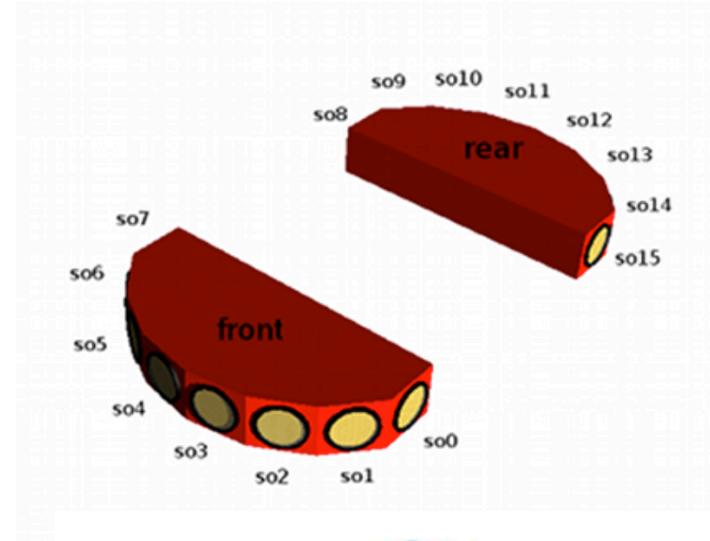
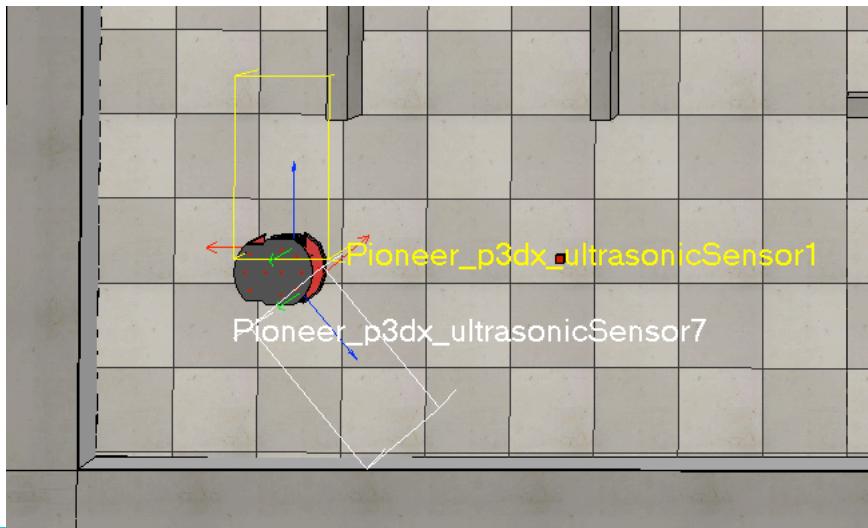
# Red box detector

- Relative position to the box
- E.g.
  - $(0.02, 3, 0)$
  - $(4.5, 2, 0)$



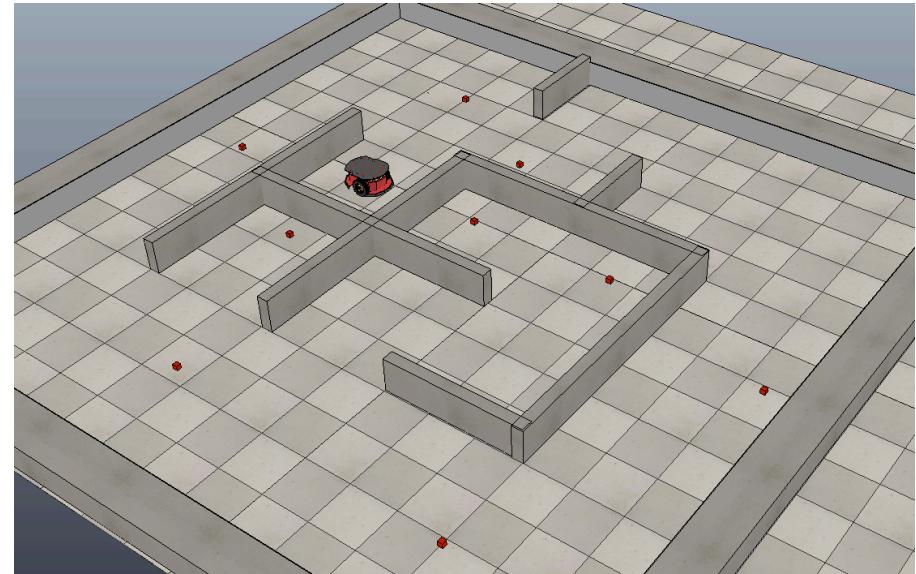
# Ultrasonic Sensor

- Distance to the object in the front



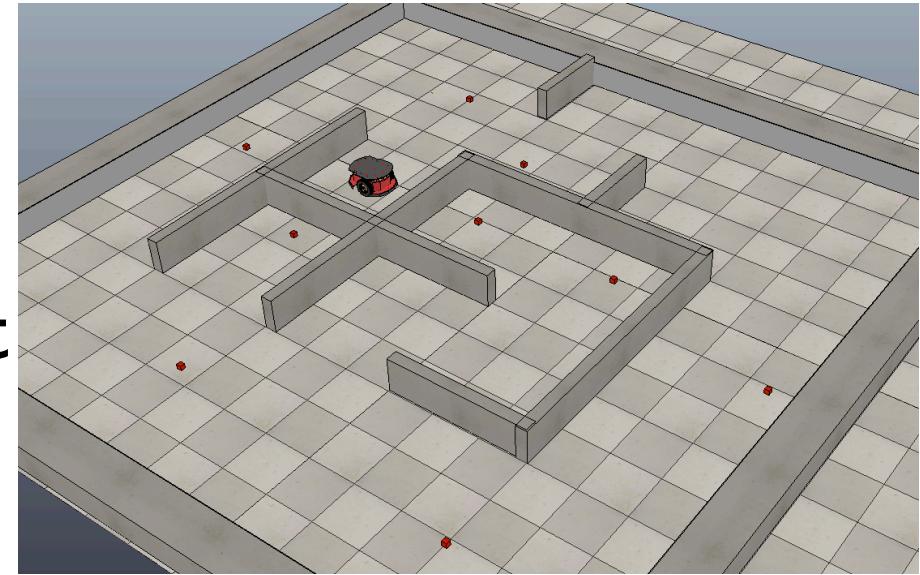
# Tasks

- Random agent
- Fixed agent
  - collect 1-2 blocks
- Reflex agent
  - read sensor input
  - decide based on that
  - e.g. always go towards nearest block
- Agent with memory
  - switch strategy (when stuck)



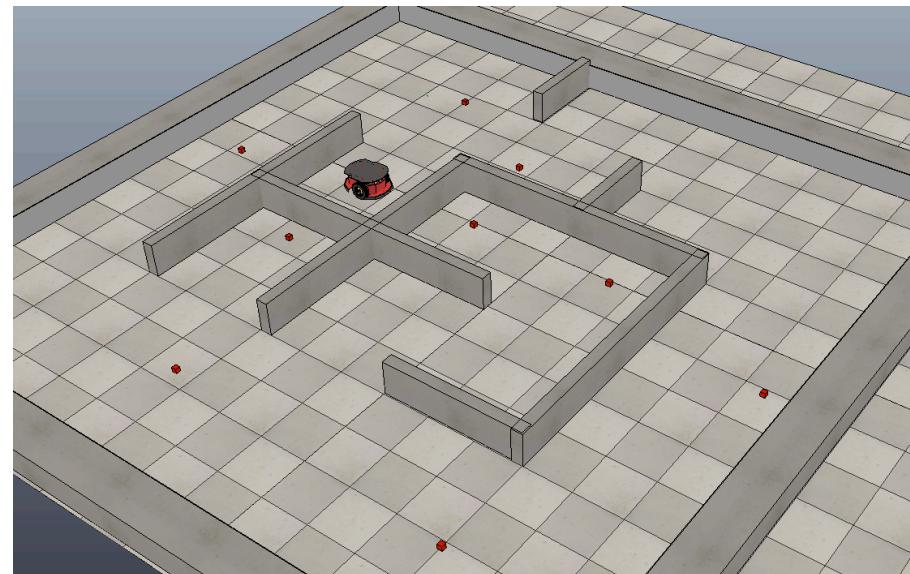
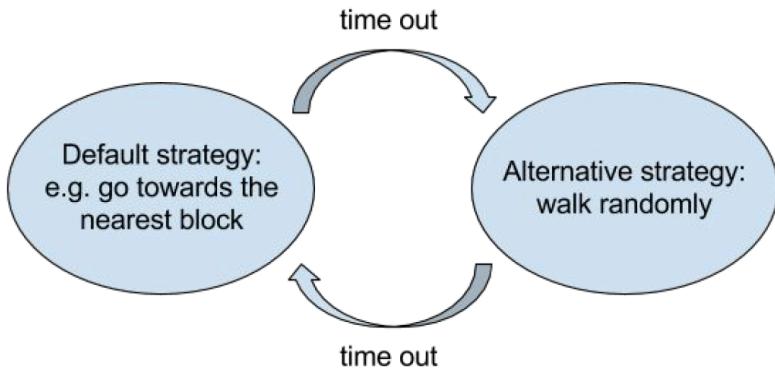
# PEAS description

- Think about the task
- In what way(s) could it be formalized
  - performance measure
  - environment
  - sensors
  - actions



# Tasks

- Random agent
- Fixed agent
- Reflex agent
- Agent with memory

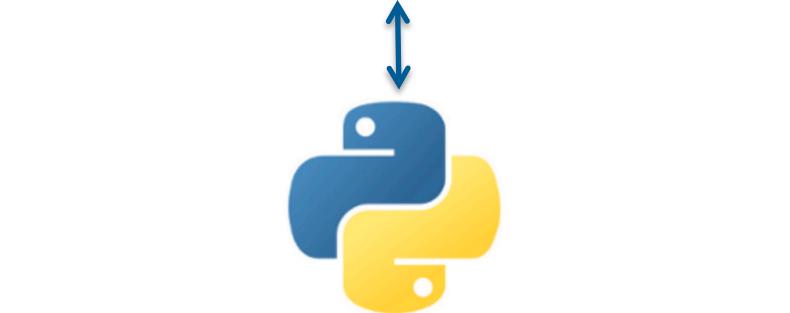
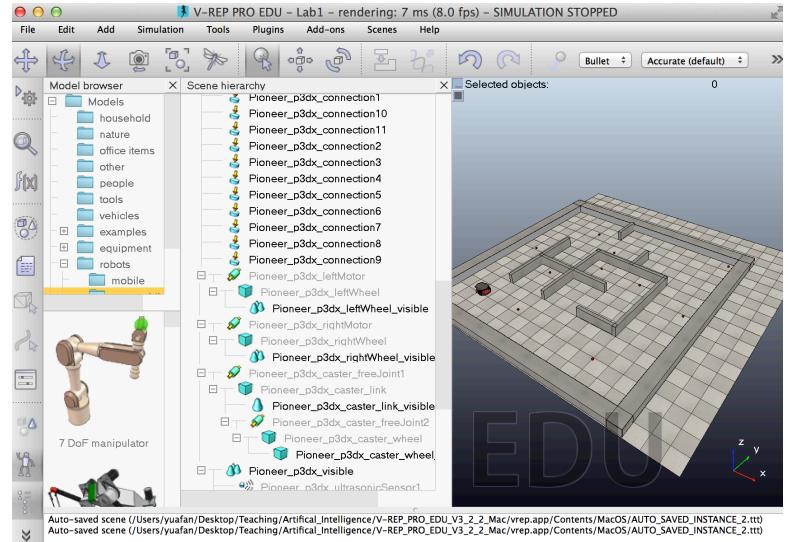


# Functions

- World.getSensorReading(...)
- World.setMotorSpeeds(...)
- World.collectNearestBlock(...)
- World.execute(...)

# Practical Issues

- Server-client style communication
- Library
  - Numpy
  - Remote API bindings
- \*Communication delay\*



python™

HALMSTAD  
UNIVERSITY

# Expectations

- Random agent
- Fixed agent
- Reflex agent
- Agent with memory

# Poker game agents

- 2 agents
  - observe that you need to create two agents to play the game
  - they may be the same or different
- 50 hands each game

Card dealing phase

Bidding phase

Showdown phase

# Game flow

- Card dealing phase
  - Assign 3 cards to each agent
- Bidding phase
  - Amount \$0-50
- Showdown phase

Card dealing phase

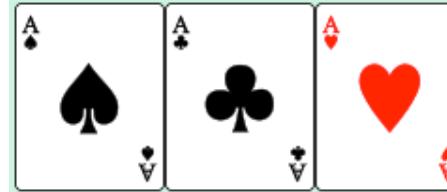
Bidding phase

Showdown phase

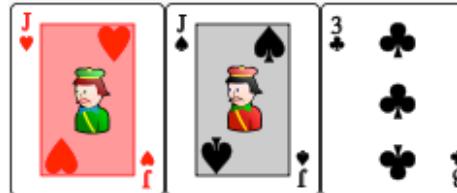
# Game flow

- Card dealing phase
  - Assign 3 cards to agents
- Bidding phase (3 times)
  - Amount \$0-50
- Showdown phase

Player 1



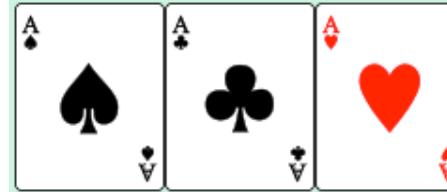
Player 2



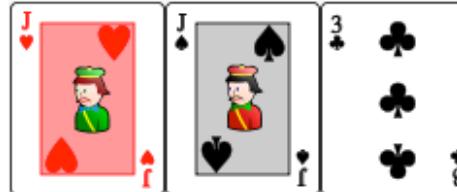
# Game flow

- Card dealing phase
  - Assign 5 cards to agents
- Bidding phase (3 time)
  - Amount \$0-50
- Showdown phase

Player 1: 50



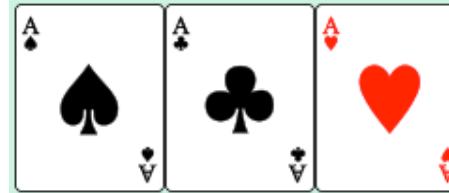
Player 2: 10



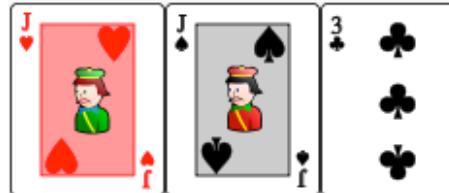
# Game flow

- Card dealing phase
  - Assign 5 cards to agents
- Bidding phase (3 time)
  - Amount \$0-50
- Showdown phase

Player 1: 50 50



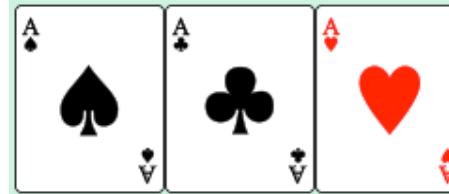
Player 2: 10 5



# Game flow

- Card dealing phase
  - Assign 5 cards to agents
- Bidding phase (3 time)
  - Amount \$0-50
- Showdown phase

Player 1: 50 50 50



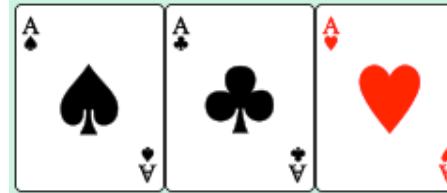
Player 2: 10 5 1



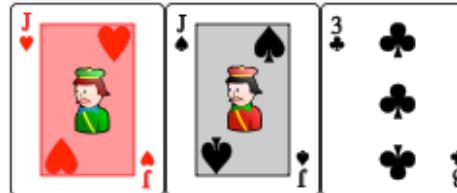
# Game flow

- Card dealing phase
  - Assign 5 cards to agents
- Bidding phase (3 time)
  - Amount \$0-50
- Showdown phase

Player 1: 50 50 50



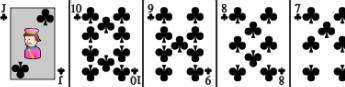
Player 2: 10 5 1



Player 1 wins: Player 1 got \$166

# Poker game agents

- Showdown phase
  - Evaluate hand

Rank	Name	Example
0	Five of a kind*	
1	Straight flush**	
2	Four of a kind	
3	Full house	
4	Flush**	
5	Straight**	
6	Three of a kind	
7	Two pair	
8	One pair	
9	High card	

# Poker game agents

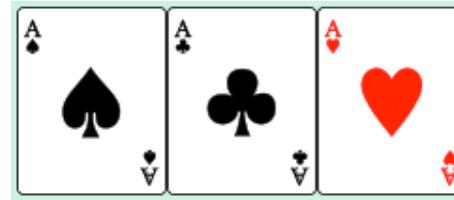
- Showdown phase
  - Evaluate hand
- Three card poker
  - 3 of a kind
  - 1 pair
  - High card

Rank	Name	Example
0	Five of a kind*	
1	Straight flush**	
2	Four of a kind	
3	Full house	
4	Flush**	
5	Straight**	
6	Three of a kind	
7	Two pair	
8	One pair	
9	High card	

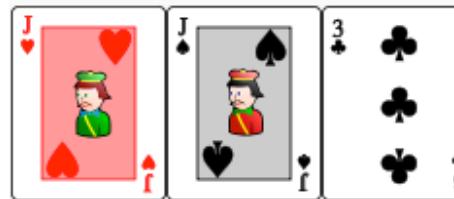
# PEAS description

- Think about the task
- In what way(s) could it be formalized
  - performance measure
  - environment
  - sensors
  - actions

Player 1: ?



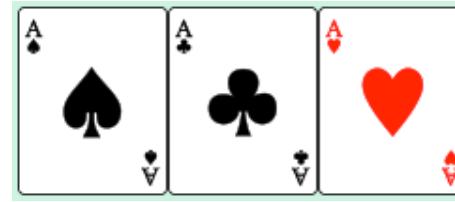
Player 2: ?



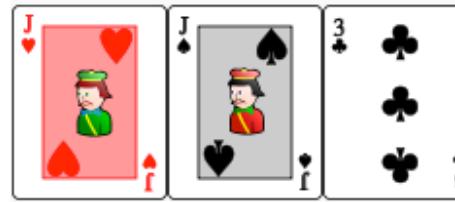
# Tasks

- Random agent
- Fixed agent
- Reflex agent
- Agent with memory

Player 1: ?



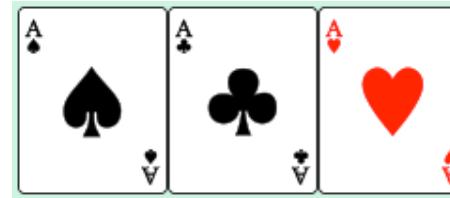
Player 2: ?



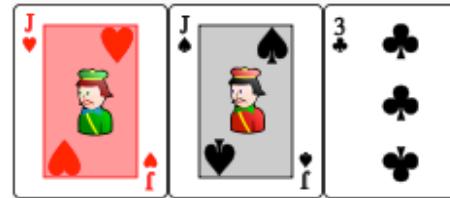
# Tasks

- Random agent
- Fixed agent
- **Reflex agent**
  - strength of own hand

Player 1: ?



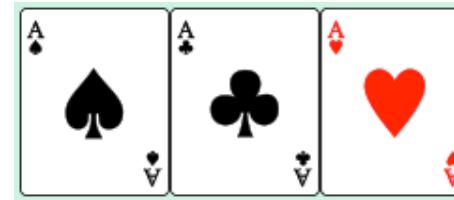
Player 2: ?



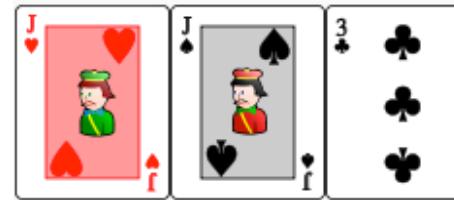
# Tasks

- Random agent
- Fixed agent
- Reflex agent
- Agent with memory
  - strength of own hand
  - amount of money that the opponent has bid
  - opponent's previous plays

Player 1: ?



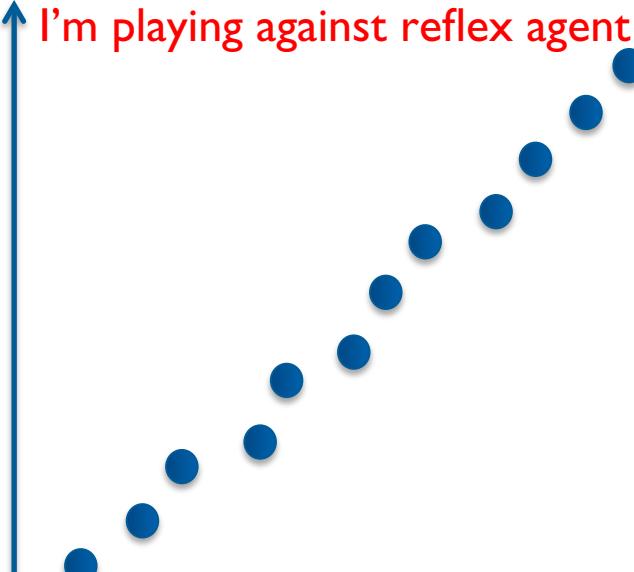
Player 2: ?



# Agent with memory

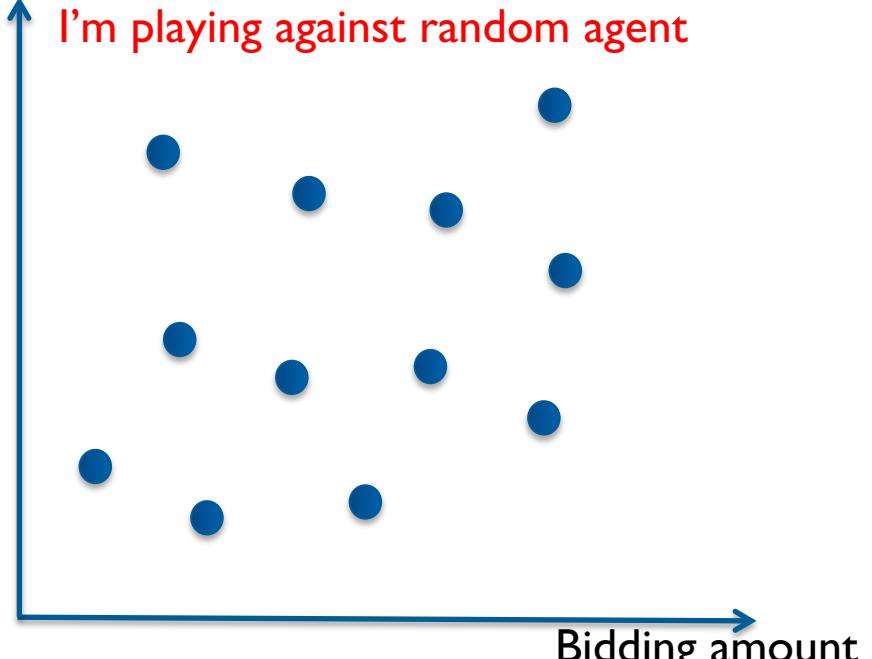
Hand strength

I'm playing against reflex agent



Hand strength

I'm playing against random agent



# Grading

- Pass/fail/extra credits
- Submit your lab on Blackboard
  - a short (1-2 pages) report
  - what you have done
  - code
- Yuantao Fan
  - [yuantao.fan@hh.se](mailto:yuantao.fan@hh.se)
  - E513