FACULTY OF ENGINEERING AND ARCHITECTURE

DEPARTMENT OF TELECOMMUNICATIONS AND INFORMATION PROCESSING
GROUP FOR ARTIFICIAL INTELLIGENCE AND SPARSE MODELLING (GAIM)

Practical sessions

Shaoguang Huang, Srđan Lazendić, Xian Li, Laurens Meeus, Nina Žižakić
professor: Aleksandra Pižurica

GHENT UNIVERSITY

imec

Friday, 11 Oct 2019

Contact

- ai@lists.ugent.be

- Office hours – by appointment

GHENT UNIVERSITY

Course organisation

- Lectures
- Theoretical problem-solving sessions
- Practical sessions

GHENT UNIVERSITY

Practical sessions

- Goal: apply learned theoretical knowledge in practice
- 3 sessions (not including this one)
- 3 graded assignments (one per session)
- Grading based on submitted code and written report
  - Deadline: 2 weeks from the session
- Every next practical session oral questioning of randomly selected students about previous assignment

  => presence is obligatory
- 1/3 of the final grade
- Groups of two, join on Ufora

GHENT UNIVERSITY

Calendar for practical sessions

- 0th – Introduction to Python (today) – no graded assignment
- 1st – Path finding (18 Oct 2019)
- 2nd – Bayesian networks (8 Nov 2019)
- 3rd – Neural networks (29 Nov 2019)

Python

- High-level programming language
- Free & open-source
- Dynamically typed
- Automatic memory management
- ‘Easy to read’ code (Matlab-like)

- For the assignments:
  - Path finding: list, dictionary, function, class
  - Neural networks: packages Keras and Tensorflow

Python installation

- All-in-one package Anaconda
- Includes the following:
  - Python 3
  - PIP python package management system (how-to)
  - Jupyter Notebook: Matlab-like
  - Integrated development environment (IDE)
    - Alternatives: PyCharm, running from the terminal...

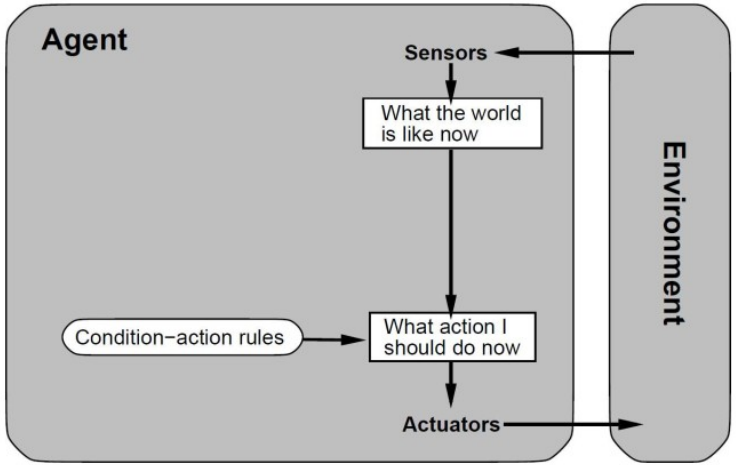
Agents: from theory to practice

```
class GhostAgent( Agent ):
    def __init__( self, index ):
        self.index = index

    def getAction( self, state ):
        dist = self.getDistribution(state)
        if len(dist) == 0:
            return Directions.STOP
        else:
            return util.chooseFromDistribution( dist )

    def getDistribution(self, state):
        "Returns a Counter encoding a distribution over actions from the provided state."
        util.raiseNotDefined()
```

Simple reflex agents



GHENT UNIVERSITY

Agents: from theory to practice

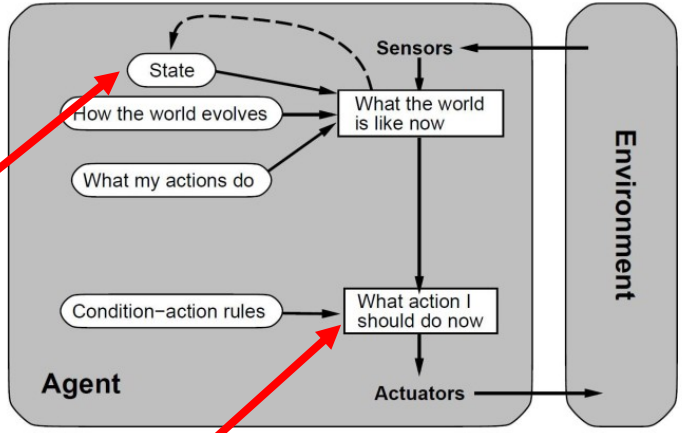
```
class GhostAgent( Agent ):
    def __init__( self, index ):
        self.index = index

    def getAction( self, state ):
        dist = self.getDistribution(state)
        if len(dist) == 0:
            return Directions.STOP
        else:
            return util.chooseFromDistribution( dist )

    def getDistribution(self, state):
        "Returns a Counter encoding a distribution over actions from the provided state."
        util.raiseNotDefined()

class RandomGhost( GhostAgent ):
    "A ghost that chooses a legal action uniformly at random."
    def getDistribution( self, state ):
        dist = util.Counter()
        for a in state.getLegalActions( self.index ): dist[a] = 1.0
        dist.normalize()
        return dist
```

Reflex agents with state



Today's schedule

- Starting to use Python

  File: Python_basics.ipynb
- Agents and environments in Pacman world

  File: search.zip
- State-Space Search with Pacman

  File: search.zip
- Multi-Agent Search with Pacman

  File:multiagents.zip

GHENT UNIVERSITY

Extra material

- Keras + Tensorflow (last practical session on neural networks)
  - https://keras.io/

- Codingame
  - challenge-based/sandbox training platform to improve your coding skills
  - Examples:
    - Easy (simple reflex agent): https://www.codingame.com/training/easy/mars-lander-episode-1
    - Increasing difficulty (from simple reflex agent to goal-based to utility-based): https://www.codingame.com/multiplayer/bot-programming/coders-strike-back