IMPLEMENTING A MULTIAGENT SYSTEM

Karel Horák

BE4M36MAS - Multiagent systems
ASSIGNMENT
Find, collect and carry all gold stones from their location to a depot!

- Miners do not know positions of gold stones and depots — they must find them
- They may carry at most one gold stone at a time
- They have limited range of sight (8-neighbourhood)
Visible: (gold,2,2), (depot,1,0)

No objects visible!
• left(), right(), up(), down() — movement in the grid
• pick(), drop() — manipulating gold stones
• sense() — use it to update your percepts (nearly no delay)
Gold stones are **heavy**.

→ there must be another miner in 4-neighbourhood for `pick()`
Gold stones are **added in runtime**
→ Your miners must be able to find them at any time

2 points
You are racing the **time** now

Your miners should not be much slower than (inefficient) reference solution

(if your agents actively pursue their goal, you will have no problems)

**1 point** / scenario

Your solution may be evaluated on slightly modified versions of mines!
You are racing the **time** now

Your miners should not be much slower than (inefficient) reference solution

(if your agents actively pursue their goal, you will have no problems)

**1 point** / scenario

Your solution may be evaluated on slightly modified versions of mines!
- You are racing the **time** now
- Your miners should not be much slower than (inefficient) reference solution
  (if your agents actively pursue their goal, you will have no problems)

1 point / scenario

Your solution may be evaluated on slightly modified versions of mines!
• You are racing the **time** now
• Your miners should not be much slower than (inefficient) reference solution
  (if your agents actively pursue their goal, you will have no problems)

1 **point** / scenario

Your solution may be evaluated on slightly modified versions of mines!
Mining world — Scenarios 2-8

- You are racing the **time** now
- Your miners should not be much slower than (inefficient) reference solution
  (if your agents actively pursue their goal, you will have no problems)

**1 point / scenario**

Your solution may be evaluated on slightly modified versions of mines!
• You are racing the **time** now
• Your miners should not be much slower than (inefficient) reference solution
  (if your agents actively pursue their goal, you will have no problems)

**1 point / scenario**

Your solution may be evaluated on slightly modified versions of mines!
You are racing the **time** now

Your miners should not be much slower than (inefficient) reference solution

(if your agents actively pursue their goal, you will have no problems)

**1 point / scenario**

Your solution may be evaluated on slightly modified versions of mines!
You can get 1 more point for implementing a fast mining team.

A competition between your submissions will be held

→ Performance of your miners will be evaluated based on Scenarios 2-8

(Multiple runs will be performed, you get $i$ points for being $i$th within the run, top 25% submissions with lowest number of total points get 1 point)
You are asked to submit a short report:

- What approach have you used for discovering gold stones and depots?
- How have you solved synchronization problems?
- What issues have you encountered and how have you overcome them?
- ...

Reward: 1 point
Deadline: 6.11.2017 04:00
• Think before implementation
• Think before implementation
• Be prepared for possible issues!
- Think before implementation
- Be prepared for possible issues!
- Keep it simple!
• Java – recommended
IMPLEMENTATION

• Java – recommended

• Support for other languages is experimental and mostly untested
  • You may encounter some issues and bugs (we will try to solve them quickly)
  • You lose some features (e.g. debugging) and you do not have Java codebase available
  • You are expected to be competent in the language of your choice

• C, C++, Python (and maybe others will get supported)
**Java**

- Familiarization with the framework
- Basic tasks
- Basic communication

**Other languages**

- Follow with us
- Familiarize yourself with the framework
- Understand the protocol (see website)
- Try to implement a very simple agent