

IMPLEMENTING A MULTIAGENT SYSTEM

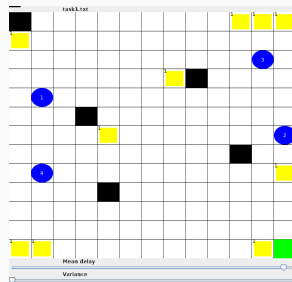
Karel Horák

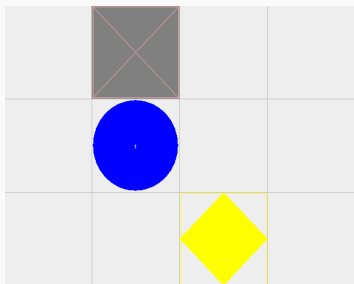
BE4M36MAS - Multiagent systems

MINING WORLD

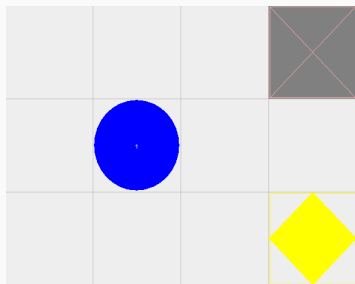
Find, collect and carry all gold stones from their location to a depot!

- 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)



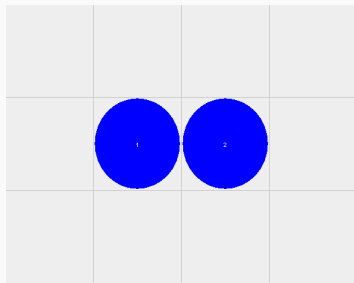
No objects visible!

- `left()`, `right()`, `up()`, `down()` — movement in the grid
- `pick()`, `drop()` — manipulating gold stones
- `sense()` — use it to update your percepts (nearly no delay)

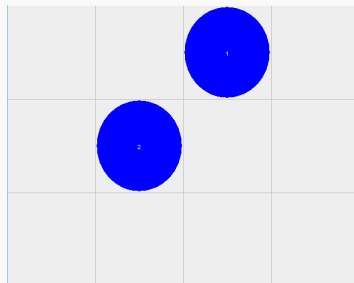
MINING WORLD — PROBLEM 1

Gold stones are **heavy**.

→ there must be another miner in 4-neighbourhood for `pick()`



`pick()` succeeds



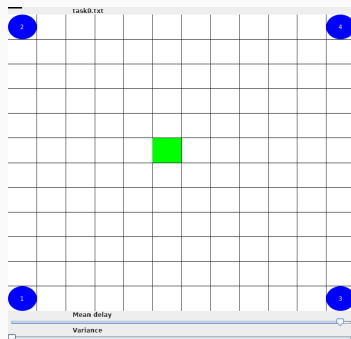
`pick()` fails

SCENARIOS & GRADING

Gold stones are **added in runtime**

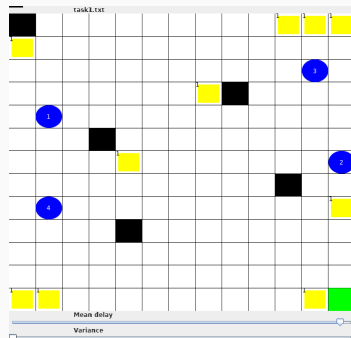
→ Your miners must be able to find them at any time

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)

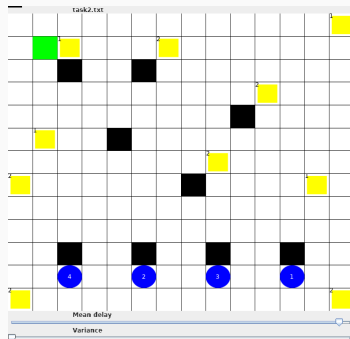
1 point



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)

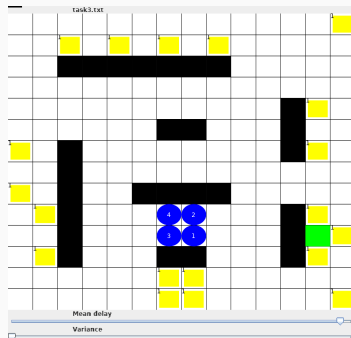
1 point



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)

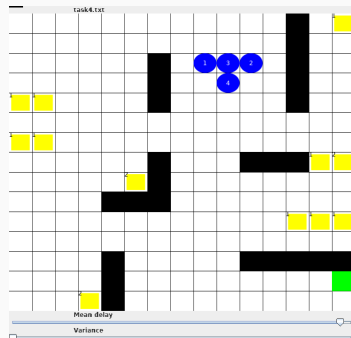
1 point



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)

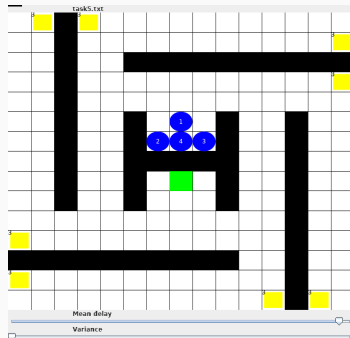
1 point



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)

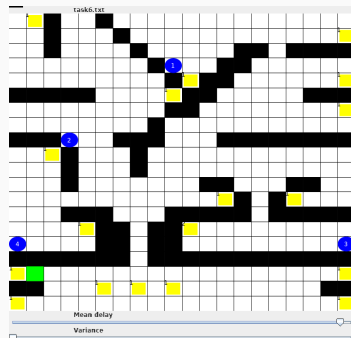
2 points



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)

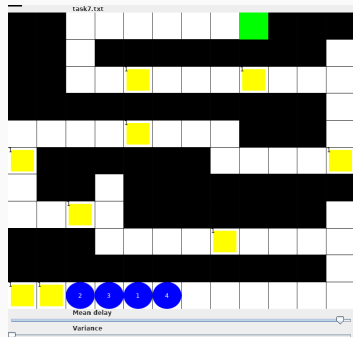
2 points



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)

2 points



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: 2. 11. 2020 04:00

IMPLEMENTATION

- Think before implementation

- Think before implementation
- Be prepared for possible issues!

- Think before implementation
- Be prepared for possible issues!
- Keep it simple!

- Think before implementation
- Be prepared for possible issues!
- Keep it simple!
- Do not try to hack the framework!

To start with the implementation,

Follow the Instructions on Courseware

- In the `AbstractAgent` class
- `right()`, `left()`, `up()`, `down()`, `drop()`, `pick()`
- Sensor input is returned from all actions. Also, there is a special action just for sensing: `sense()`.

- send: `sendMessage(agentID, messageObject)`
- read: `Message m = readMessage();`

To be successful, you will probably need

- to communicate between agents,
- to have a representation of the environment,
- to plan somehow, at least for a short time horizon,
- and to cooperate and distribute the task between agents

NEXT

Normal-Form Games by Dominik Seitz

Link to poll