

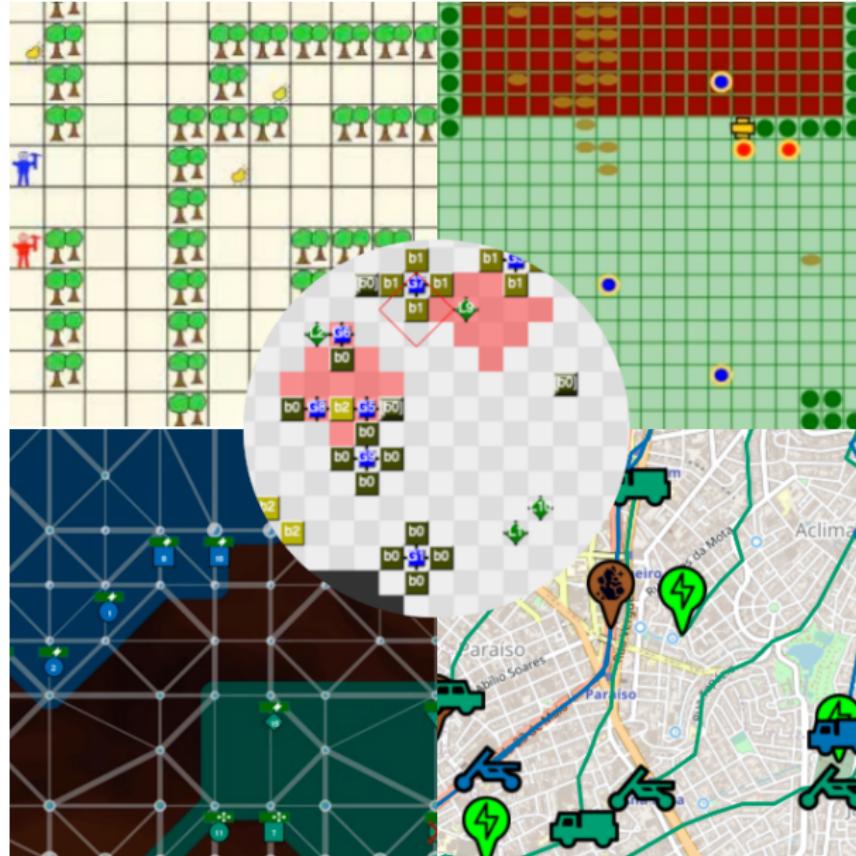


The Multi-Agent Programming Contest

A challenge for everyone

T. Ahlbrecht J. Dix Niklas Fiekas Tabajara Krausburg
Department of Informatics, Clausthal University of Technology
WESAAC Short Course 2, 12 August 2021

TU Clausthal The MAPC [1]





TU Clausthal

General Instructions

- Keep your smartphone around, you might need it!



TU Clausthal

General Instructions

- Keep your smartphone around, you might need it!
- For **questions** you can either:



- Keep your smartphone around, you might need it!
- For **questions** you can either:
 - type it directly in the chat; or



TU Clausthal

General Instructions

- Keep your smartphone around, you might need it!
- For **questions** you can either:
 - type it directly in the chat; or
 - raise your hand (hand emoji in the chat) and speak it out loud





TU Clausthal

General Instructions

- Keep your smartphone around, you might need it!
- For **questions** you can either:
 - type it directly in the chat; or
 - raise your hand (hand emoji in the chat) and speak it out loud
- For the practical part you'll need to collaborate with other participants

- Keep your smartphone around, you might need it!
- For **questions** you can either:
 - type it directly in the chat; or
 - raise your hand (hand emoji in the chat) and speak it out loud
- For the practical part you'll need to collaborate with other participants
 - Having a **microphone** will make it easier

- Keep your smartphone around, you might need it!
- For **questions** you can either:
 - type it directly in the chat; or
 - raise your hand (hand emoji in the chat) and speak it out loud
- For the practical part you'll need to collaborate with other participants
 - Having a **microphone** will make it easier
- **Important:** links in the shared notes!



TU Clausthal

Warming Up!

<https://itempool.com/TabajaraKrausburg/live/>



Table: Schedule for this Short Course

When (UTC-3)	What
9:00	Welcome & Introduction & Agenda
9:15	MASSim Platform & Protocol
9:55	Break
10:00	Agents Assemble II
10:55	Break
11:00	Time for Practicing
11:40	Wrap-up and Q&A

1. What is the MAPC?

Toni/Torroni (2004): Need for a **competition event** for

- 1 modelling (problems using logic-based agents)
- 2 specifying (logic-based multi-agent systems, given a problem)
- 3 programming/implementing (logic-based multi-agent solutions)

Toni/Torroni (2004): Need for a **competition event** for

- 1 modelling (problems using logic-based agents)
- 2 specifying (logic-based multi-agent systems, given a problem)
- 3 programming/implementing (logic-based multi-agent solutions)

Dastani/Dix/Novak were lured into it: **15 Agent Contests by now.**
Jomi Hübner joined the SC in 2017.

Toni/Torroni (2004): Need for a **competition event** for

- 1 **modelling** (problems using logic-based agents)
- 2 **specifying** (logic-based multi-agent systems, given a problem)
- 3 **programming/implementing** (logic-based multi-agent solutions)

Dastani/Dix/Novak were lured into it: **15 Agent Contests by now.**

Jomi Hübner joined the SC in 2017.

Simulation Server: Suitable for problems like simulated mobile/cognitive robotics. **Provide a dynamic environment for those simulated players.**

- Stimulate research in the area of agent-oriented programming

- Stimulate research in the area of **agent-oriented programming**
- Identify **key problems** for MAS

- Stimulate research in the area of **agent-oriented programming**
- Identify **key problems** for MAS
- Collect **suitable benchmarks** that can serve as milestones for evaluating new tools, models, and techniques

- Stimulate research in the area of agent-oriented programming
- Identify key problems for MAS
- Collect suitable benchmarks that can serve as milestones for evaluating new tools, models, and techniques
- Gathering test cases which require and enforce coordinated actions

The MASSim Platform

Server: Unified scenario implementation

Agents: Connect remotely (sockets)

- Receive percepts
(state of the visible environment)
- Deliberate and send actions back

Simulation: Discrete steps

- Actions of all agents processed after each step

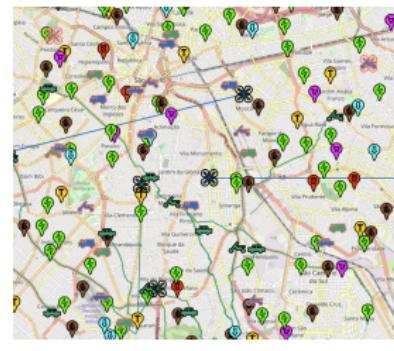
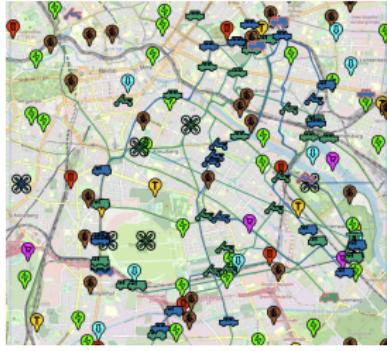
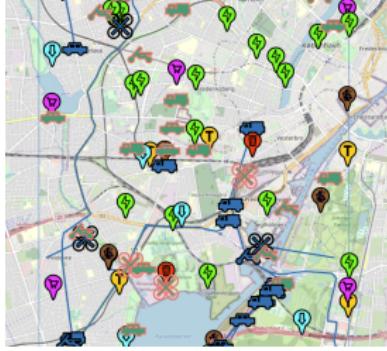


- Games between **2 teams** of agents (as of now)

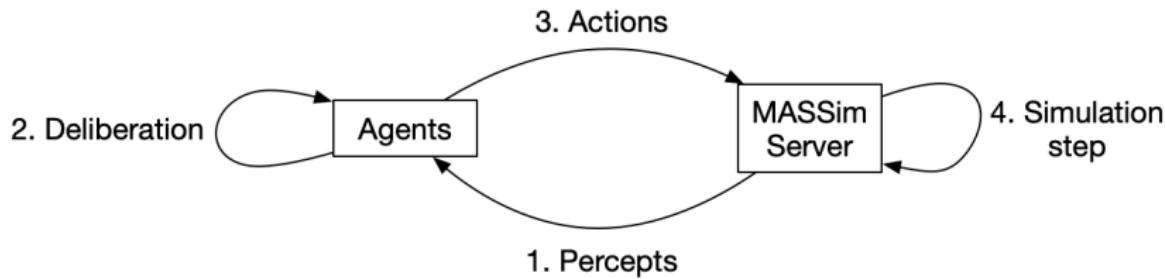
- Games between **2 teams** of agents (as of now)
- Each team plays against each other

- Games between **2 teams** of agents (as of now)
- Each team plays against each other
- Deliberation: **4 seconds** per step for network roundtrip and action computation

- Games between **2 teams** of agents (as of now)
- Each team plays against each other
- Deliberation: **4 seconds** per step for network roundtrip and action computation
- **3 simulations** with different parameters per match
 - environment characteristics
 - agent characteristics (features, number, ...)



2. The MASSim Platform



Features in the MASSim

- The MASSim server is implemented in Java

Features in the MASSim

- The MASSim server is implemented in Java
- Highly configurable
 - Simulations
 - Steps
 - Teams
 - ...

Features in the MASSim

- The MASSim server is implemented in Java
- Highly configurable
 - Simulations
 - Steps
 - Teams
 - ...
- Also, it provides an implementation of EIS [2, 3] (EISMASSim) to Agent Platforms in Java
 - Abstraction to model the **agent-environment** interaction¹
 - Handles the **communication** with the MASSim server

¹<https://github.com/eishub/>

The Environment Interface Standard (EIS)

■ Main terms in EIS:

Agent: an external piece of software

Entity: can **sense** and **act** in the environment

- usually they do not perform reasoning
- controlled by **agents**

Environment: where the entities are

The Environment Interface Standard (EIS)

■ Main terms in EIS:

Agent: an external piece of software

Entity: can **sense** and **act** in the environment

- usually they do not perform reasoning
- controlled by **agents**

Environment: where the entities are

■ Provides a **standard** for actions, percepts, events, etc

- Interface Intermediate Language (IILang)
- Each item becomes an abstract syntax tree

The Environment Interface Standard (EIS)

■ Main terms in EIS:

Agent: an external piece of software

Entity: can **sense** and **act** in the environment

- usually they do not perform reasoning
- controlled by **agents**

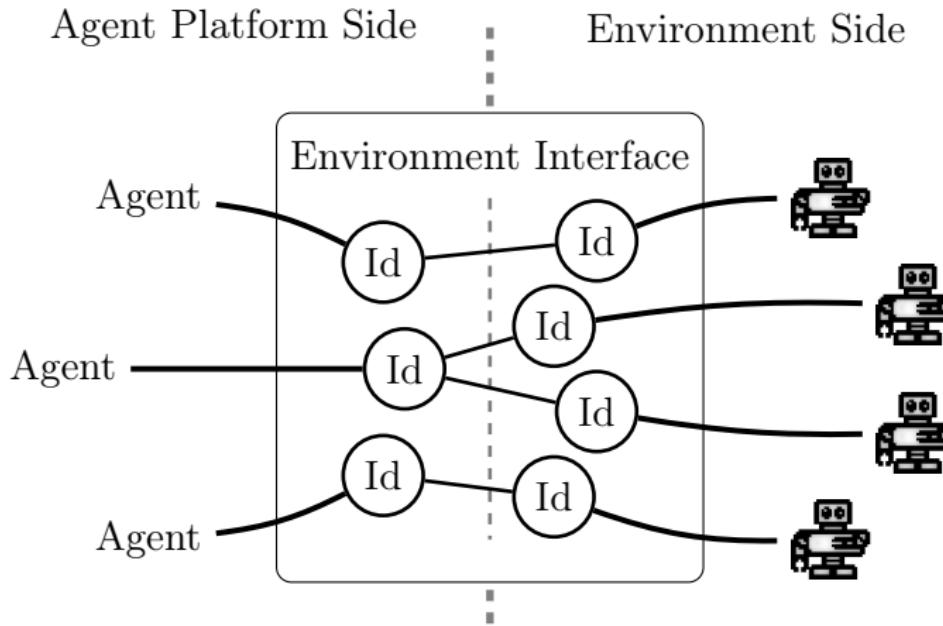
Environment: where the entities are

■ Provides a **standard** for actions, percepts, events, etc

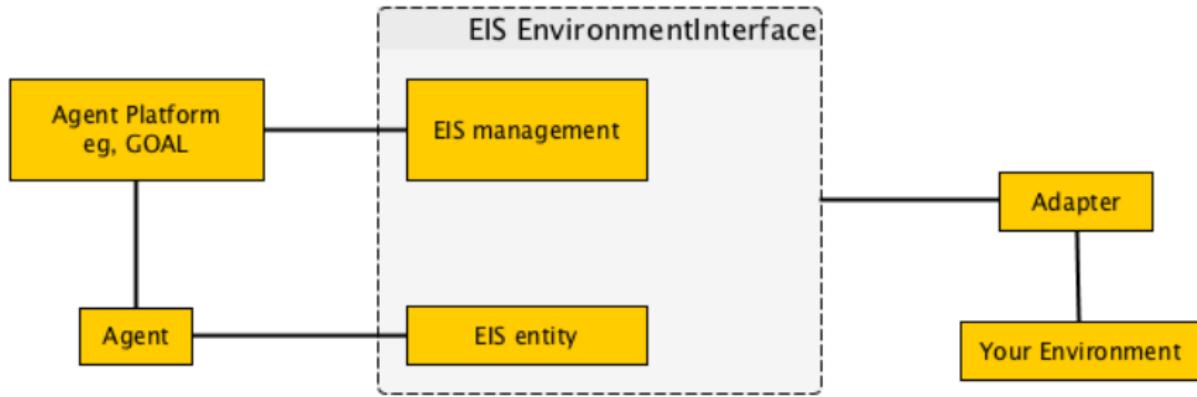
- Interface Intermediate Language (IILang)
- Each item becomes an abstract syntax tree

■ Controls the **running states** of the environment: initialize, pause, unpause, etc

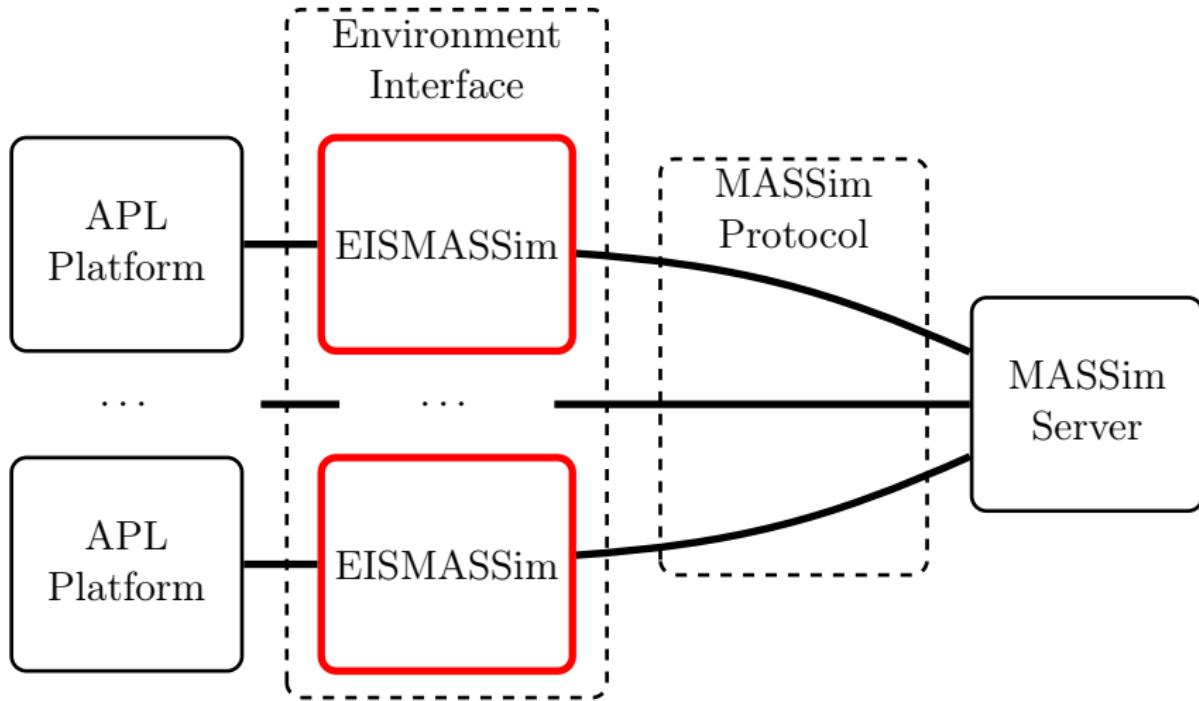
Mapping Entities to Agents



An Overview of the EIS interface



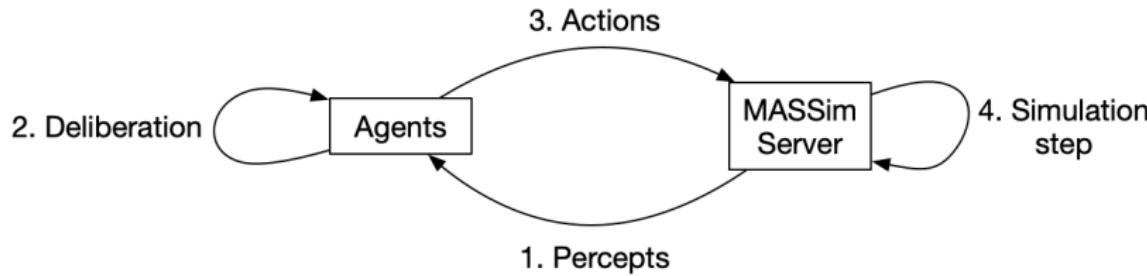
<https://github.com/eishub/eis/wiki/IntroductionToEIS>



3. MASSim Protocol

What: a sequence of **JSON messages** exchanged between the agents and the MAPC server

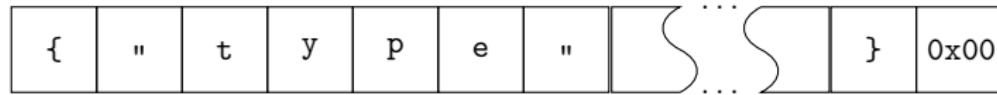
Transport Layer: TCP sockets



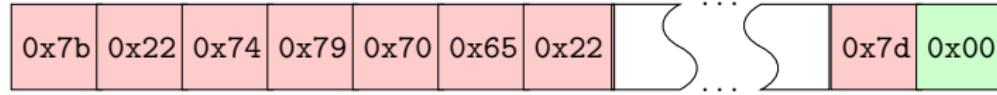
MASSim Server Buffer

```
1 {
2   "type": "auth-response",
3   "content": {
4     ...
5   }
6 }
```

```
1 {
2   "type": "auth-response",
3   "content": {
4     ...
5   }
6 }
```



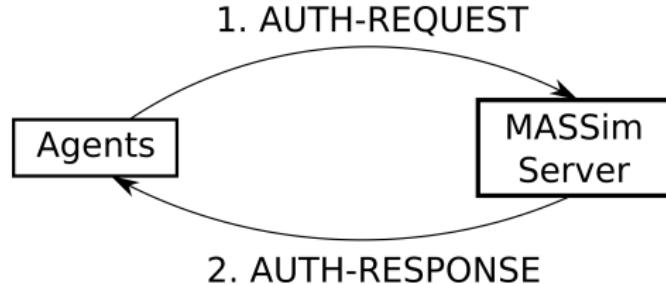
UTF-8 Encoding



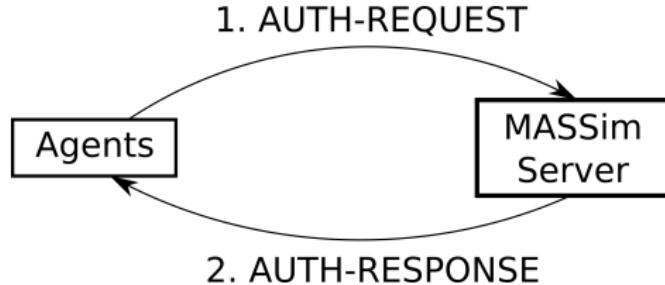
JSON Message

End of Message

Connecting to the Server



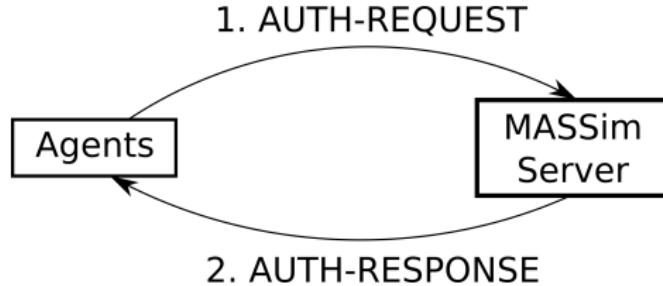
Connecting to the Server



AUTH-REQUEST

```
1 {
2   "type": "auth-request",
3   "content": {
4     "user": "agentA1",
5     "pw": "1"
6   }
7 }
```

Connecting to the Server



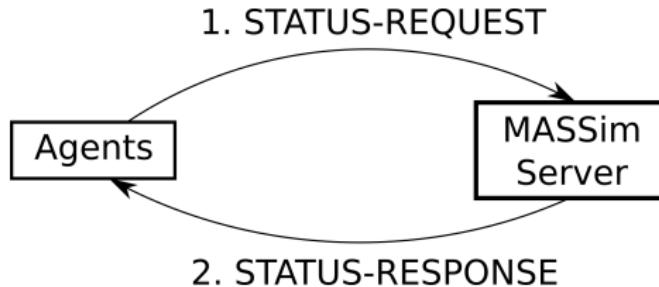
AUTH-REQUEST

```
1 {
2   "type": "auth-request",
3   "content": {
4     "user": "agentA1",
5     "pw": "1"
6   }
7 }
```

AUTH-RESPONSE

```
1 {
2   "type": "auth-response",
3   "content": {
4     "result": "ok"
5   }
6 }
```

Checking the Status



STATUS-REQUEST

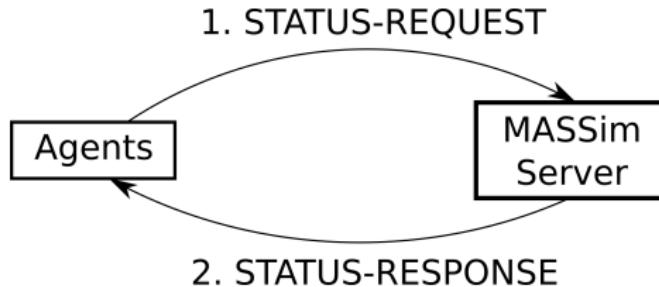
```
1 {
2   "type": "status-request",
3   "content": {}
4 }
```

Can you connect to the server?

Your task is:

- 1 go to the google drive folder;
- 2 open up the file **Activity_1.ipnyb** ;
- 3 read the **status** sending to the MASSim server the previous JSON; then
- 4 go to the **Quiz!!!**





STATUS-REQUEST

```
1 {
2   "type": "status-request",
3   "content": {}
4 }
```

STATUS-RESPONSE

```
1 {
2   "type": "status-response",
3   "content": {
4     "teams": ["A", "B"],
5     "time": 1588865434131,
6     "teamSizes": [15, 30, 50],
7     "currentSimulation": 0
8   }
9 }
```

Starting the Simulation

The simulation begins
either when:

- time is up
- key pressed
- all agents connected



Starting the Simulation

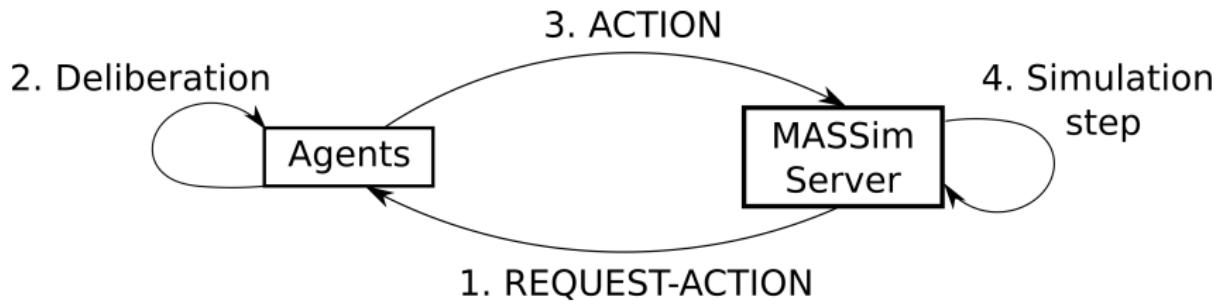
The simulation begins
either when:

- time is up
- key pressed
- all agents connected



SIM-START

```
1 {  
2   "type": "sim-start",  
3   "content": {  
4     "time": 1489514146201,  
5     "percept": {  
6       ...  
7     }  
8   }  
9 }
```



Cycle of Actions

REQUEST-ACTION

```
1 {
2   "type": "request-action",
3   "content": {
4     "id": 2,
5     "time": 1556636930397,
6     "deadline": 15566369344,
7     "step": 27,
8     "percept": {
9       ...
10    }
11  }
12 }
```

ACTION

```
1 {
2   "type": "action",
3   "content": {
4     "id": 2,
5     "type": "move",
6     "p": ["n"]
7   }
8 }
```

REQUEST-ACTION

```
1 {
2   "type": "request-action",
3   "content": {
4     "id": 2,
5     "time": 1556636930397,
6     "deadline": 15566369344,
7     "step": 27,
8     "percept": {
9       ...
10    }
11  }
12 }
```

ACTION

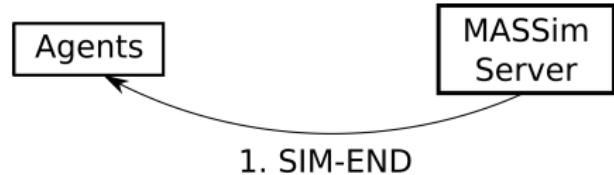
```
1 {
2   "type": "action",
3   "content": {
4     "id": 2,
5     "type": "move",
6     "p": ["n"]
7   }
8 }
```

Where:

- time is when the message was created
- deadline is up to when the server expects an ACTION

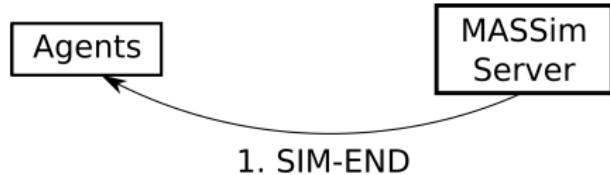
Ending a Simulation

All steps are processed!



Ending a Simulation

All steps are processed!

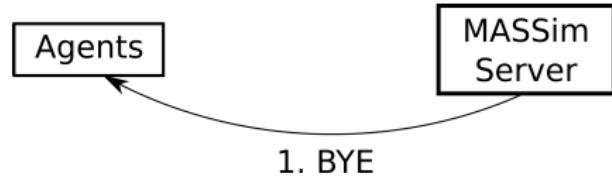


SIM-END

```
1 {  
2   "type": "sim-end",  
3   "content": {  
4     "score": 9001,  
5     "ranking": 1,  
6     "time": 1556638423323  
7   }  
8 }
```

If there are more simulations to run, then we go back to Starting a Simulation

No more simulations to run?
then...



BYE

```
1 {  
2   "type": "bye",  
3   "content": {}  
4 }
```

- During a match any agent might get disconnected
 - Bugs
 - Poor connection

- During a match any agent might get disconnected
 - Bugs
 - Poor connection
- When that happens, the agent needs to:
 - reconnect using an **AUTH-REQUEST** message

- During a match any agent might get disconnected
 - Bugs
 - Poor connection
- When that happens, the agent needs to:
 - reconnect using an **AUTH-REQUEST** message
- After a successful authentication, it receives:
 - a **SIM-START** message; and
 - the **REQUEST-ACTION** messages to come

- During a match any agent might get disconnected
 - Bugs
 - Poor connection
- When that happens, the agent needs to:
 - reconnect using an **AUTH-REQUEST** message
- After a successful authentication, it receives:
 - a **SIM-START** message; and
 - the **REQUEST-ACTION** messages to come
- **Important:** an agent might reconnect in the middle of something
 - Collaborating to accomplish a task
 - Carrying a block

4. MAPC Scenario—Agents Assemble II

- Origin
- Obstacles
- Clear
- Event
- Entity
- Task
- Boards
- Dispensers
- Goal Zones



- Origin
- Obstacles
- Clear Event
- Entity
- Task Boards
- Dispensers
- Goal Zones



- Origin
- Obstacles
- Clear Event
- Entity
- Task Boards
- Dispensers
- Goal Zones



- Origin
- Obstacles
- Clear Event
- Entity
- Task Boards
- Dispensers
- Goal Zones



- Origin
- Obstacles
- Clear
- Event
- Entity
- Task
- Boards
- Dispensers
- Goal Zones



- Origin
- Obstacles
- Clear
- Event
- Entity
- Task
- Boards
- Dispensers
- Goal Zones

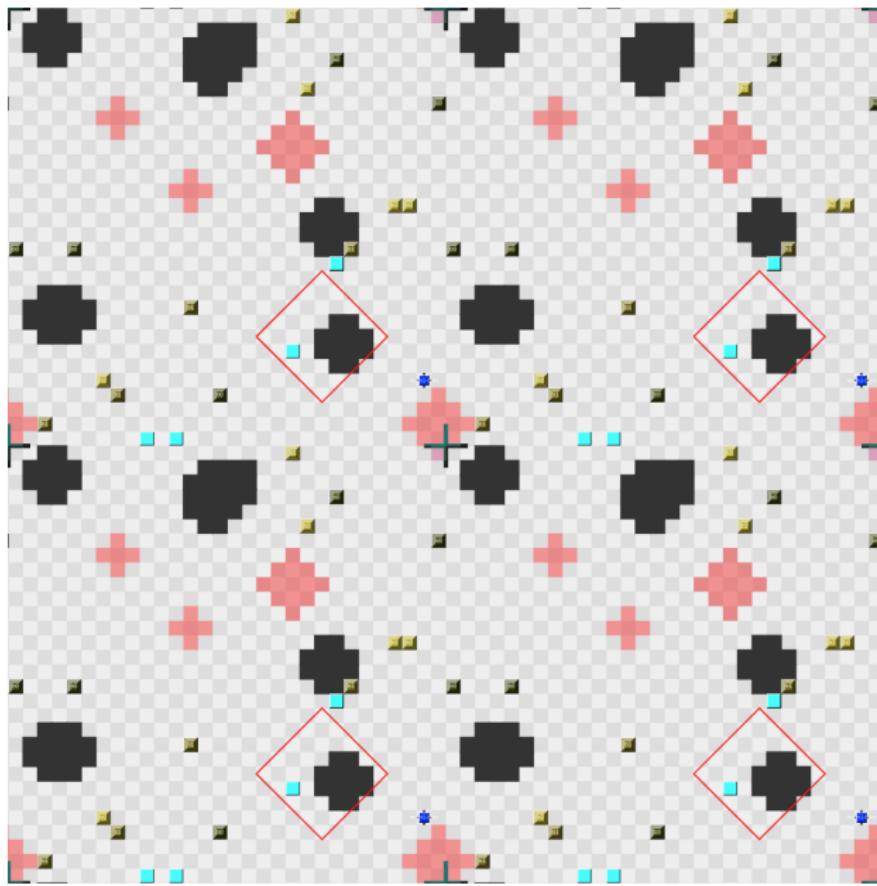


- Origin
- Obstacles
- Clear
- Event
- Entity
- Task
- Boards
- Dispensers
- Goal Zones

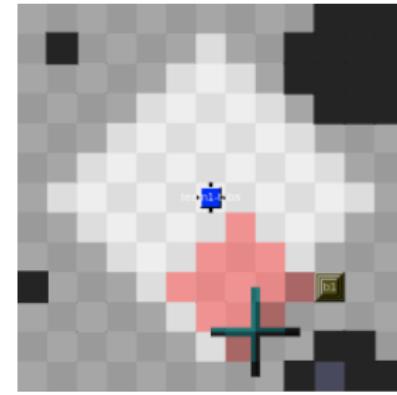
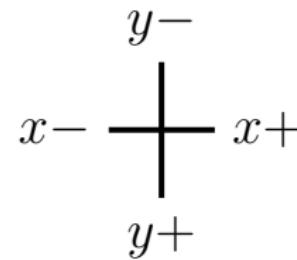


- Origin
- Obstacles
- Clear
- Event
- Entity
- Task
- Boards
- Dispensers
- Goal Zones

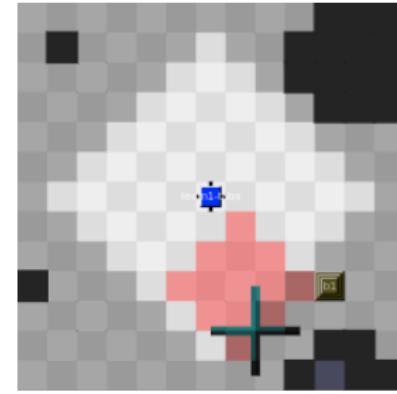
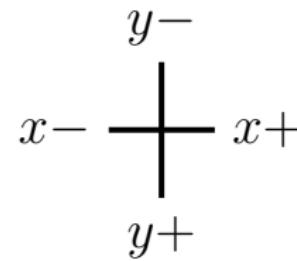




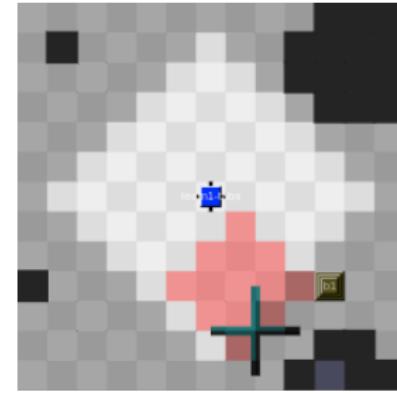
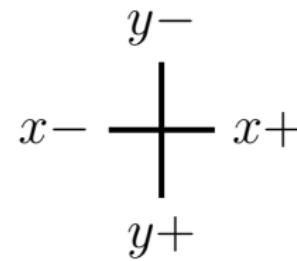
- controls one entity
- does not know its absolute position
- has four “arms”
- has energy (for clearing up)
- automatically recharges (1 unit per step)
- has a vision range



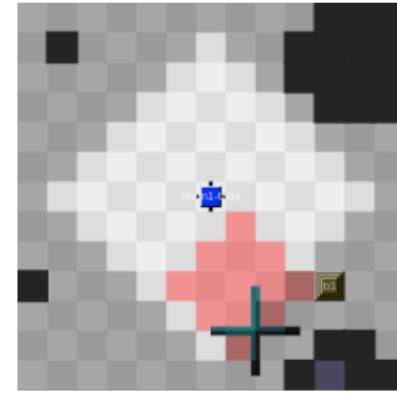
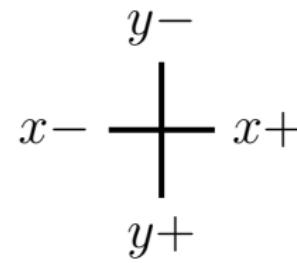
- controls one entity
- does not know its absolute position
- has four “arms”
- has energy (for clearing up)
- automatically recharges (1 unit per step)
- has a vision range



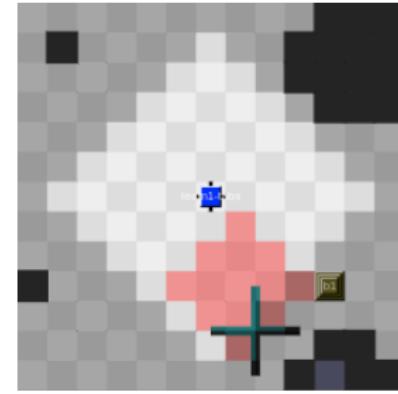
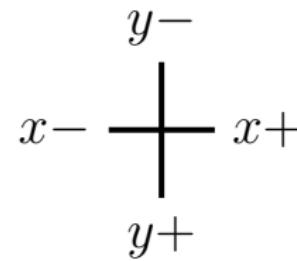
- controls one entity
- does not know its absolute position
- has four “arms”
- has energy (for clearing up)
- automatically recharges (1 unit per step)
- has a vision range



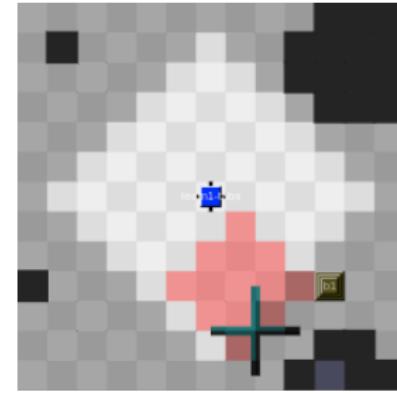
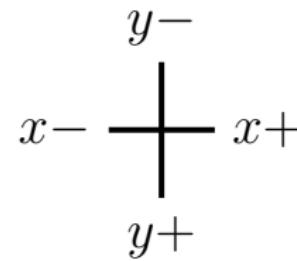
- controls one entity
- does not know its absolute position
- has four “arms”
- has energy (for clearing up)
- automatically recharges (1 unit per step)
- has a vision range



- controls one entity
- does not know its absolute position
- has four “arms”
- has energy (for clearing up)
- automatically recharges (1 unit per step)
- has a vision range



- controls one entity
- does not know its absolute position
- has four “arms”
- has energy (for clearing up)
- automatically recharges (1 unit per step)
- has a vision range



The MAPC server parses its internal state to JSON and sends it to the agents

```
1 "percept": {  
2     "score": 0,  
3     "lastAction": "move",  
4     ...  
5     "task": "task95",  
6     "attached": [[2,-1]],  
7     ...  
8     "things": [ {  
9         "x": 2,  
10        "y": -1,  
11        "details": "b1",  
12        "type": "block"  
13    },  
14    ...  
15  ],  
16  ...
```

The MAPC server parses its internal state to JSON and sends it to the agents

```
1 "percept": {  
2     "score": 0,  
3     "lastAction": "move",  
4     ...  
5     "task": "task95",  
6     "attached": [[2,-1]],  
7     ...  
8     "things": [ {  
9         "x": 2,  
10        "y": -1,  
11        "details": "b1",  
12        "type": "block"  
13    },  
14    ...  
15  ],  
16  ...
```

```
17      ...
18      "terrain": {
19          "goal": [[-4,-1],[ -4,0],[ -5,0]],
20          "obstacle": [[4,-1],[4,0],[4,1]]
21      },
22      "tasks": [ {
23          "name": "task2",
24          ...
25          "requirements": [ {
26              "x": 1,
27              ...
28              "type": "b0"
29          },
30          ...
31      ]
32      },
33      ...
34  ],
35 }
```

```
17     ...
18     "terrain": {
19         "goal": [[-4,-1], [-4,0], [-5,0]],
20         "obstacle": [[4,-1], [4,0], [4,1]]
21     },
22     "tasks": [ {
23         "name": "task2",
24         ...
25         "requirements": [ {
26             "x": 1,
27             ...
28             "type": "b0"
29         },
30         ... ▶ more requirements
31     ]
32     },
33     ... ▶ more tasks
34 ],
35 }
```

Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

Showroom



Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

Showroom



Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

Showroom



Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

Showroom



Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

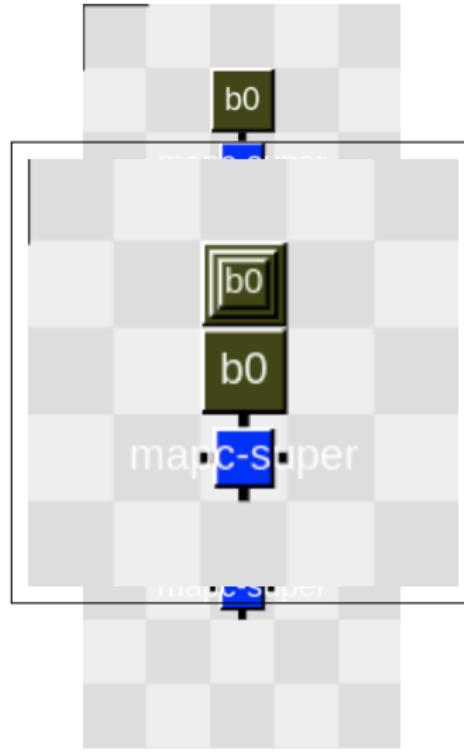
Showroom



Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

Showroom



Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

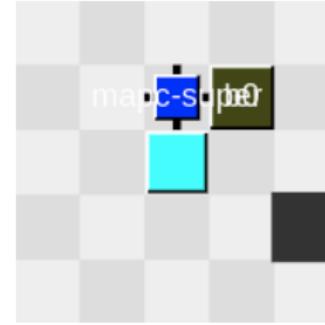
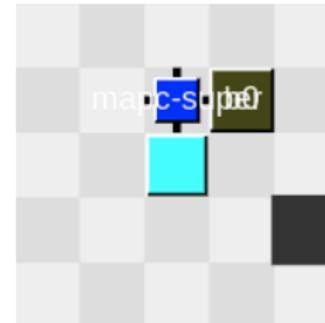
Showroom



Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

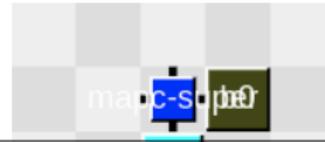
Showroom



Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

Showroom



\$8 for task12 until step 188 (1 accepted) ▾

Accepted by 1 agent:

- mapc-super

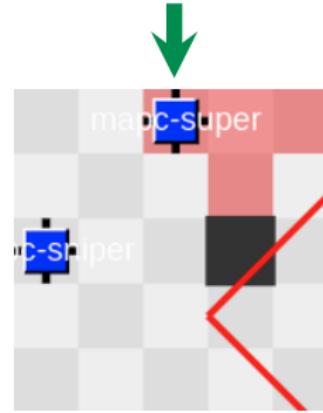
2 blocks



Actions available to the agents

- Skip
- Move
- Request
- Attach
- Rotate
- Accept
- Submit

Showroom



Playing in the Agents Assemble II Scenario

General Instructions:

- All actions you'll need are already implemented
 - You only use them
 - Of course, you may customize them as you like!
- After movement actions you'll see what the agent is seeing in its range view
 - Colors changing? initially agents might be stacked on top of each other
- After opening the colab file:
 - 1 time — run the first cell to get the code
 - 1 time — run the second cell to connect to the MASSim server
 - Should you lose connection, you need to run it again
 - x times — the remaining cells
- The agents' view in the Monitor will help you a lot!
<http://agentcontest1.in.tu-clausthal.de/>

Playing in the Agents Assemble II Scenario

General Instructions:

- All actions you'll need are already implemented
 - You only use them
 - Of course, you may customize them as you like!
- After movement actions you'll see what the agent is seeing in its range view
 - Colors changing? initially agents might be stacked on top of each other
- After opening the colab file:
 - 1 time — run the first cell to get the code
 - 1 time — run the second cell to connect to the MASSim server
 - Should you lose connection, you need to run it again
 - x times — the remaining cells
- The agents' view in the Monitor will help you a lot!
<http://agentcontest1.in.tu-clausthal.de/>

Playing in the Agents Assemble II Scenario

General Instructions:

- All actions you'll need are already implemented
 - You only use them
 - Of course, you may customize them as you like!
- After movement actions you'll see what the agent is seeing in its range view
 - Colors changing? initially agents might be stacked on top of each other
- After opening the colab file:
 - 1 time — run the first cell to get the code
 - 1 time — run the second cell to connect to the MASSim server
 - Should you lose connection, you need to run it again
 - x times — the remaining cells
- The agents' view in the Monitor will help you a lot!

<http://agentcontest1.in.tu-clausthal.de/>

Playing in the Agents Assemble II Scenario

General Instructions:

- All actions you'll need are already implemented
 - You only use them
 - Of course, you may customize them as you like!
- After movement actions you'll see what the agent is seeing in its range view
 - Colors changing? initially agents might be stacked on top of each other
- After opening the colab file:
 - 1 time — run the first cell to get the code
 - 1 time — run the second cell to connect to the MASSim server
 - Should you lose connection, you need to run it again
 - x times — the remaining cells
- The agents' view in the Monitor will help you a lot!
<http://agentcontest1.in.tu-clausthal.de/>

Can you submit a task?

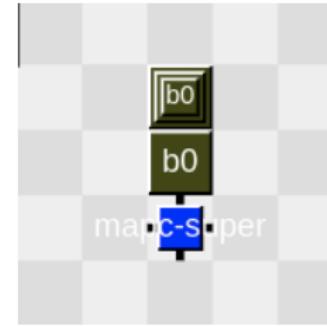
Try the following:

- 1 go to the google drive folder;
- 2 open up the file **Activity_2.ipnyb** ;
- 3 connect to the MAPC server;
- 4 wait until the connection is established (can take a while)
 - We wait for the connection of **all** participants!
- 5 once connected:
 - 1 chose a task (tasks require a single block);
 - 2 collect the block;
 - 3 go to a goal zone; and
 - 4 submit the task
- 6 once **all** participants have submitted their tasks, the activity is over!

Actions available to the agents

- Detach
- Connect
- Disconnect
- Clear

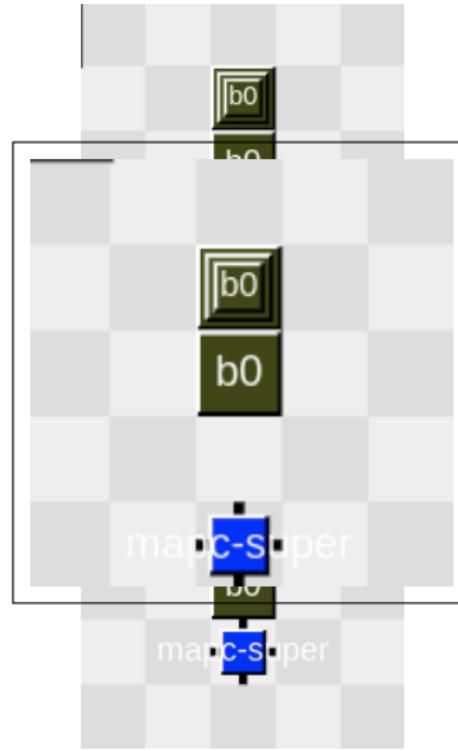
Showroom



Actions available to the agents

- Detach
- Connect
- Disconnect
- Clear

Showroom



Actions available to the agents

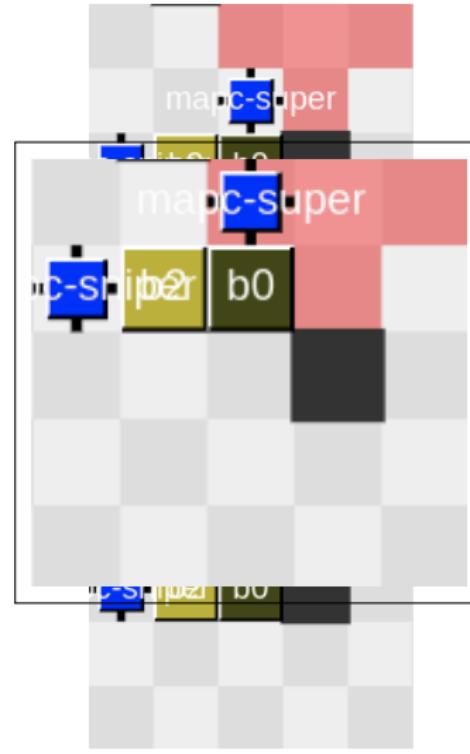
- Detach
- Connect
- Disconnect
- Clear

Showroom



Actions available to the agents

Showroom



- Detach
- Connect
- Disconnect
- Clear

Actions available to the agents

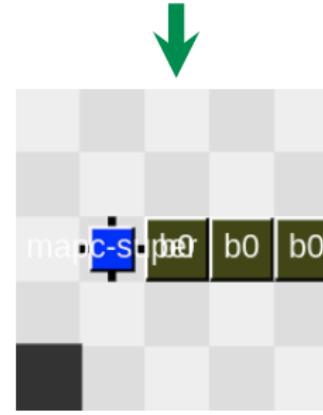
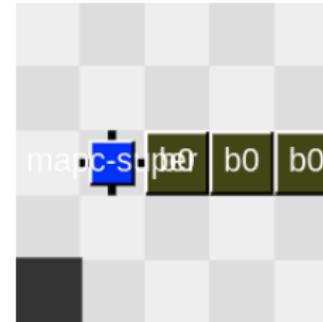
- Detach
- Connect
- Disconnect
- Clear

Showroom



Actions available to the agents

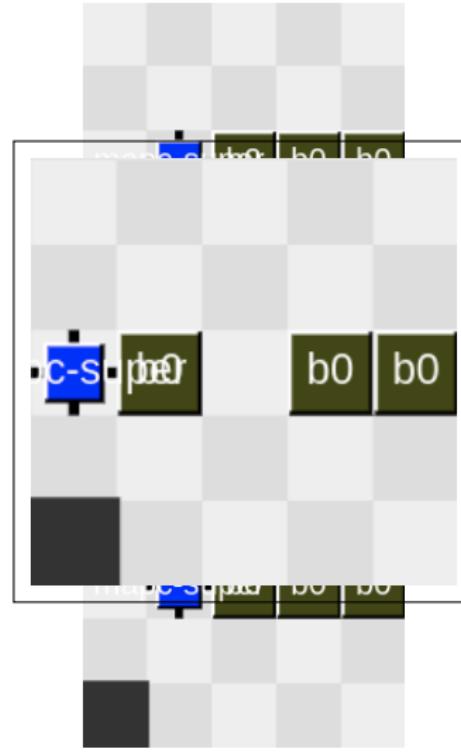
Showroom



- Detach
- Connect
- Disconnect
- Clear

Actions available to the agents

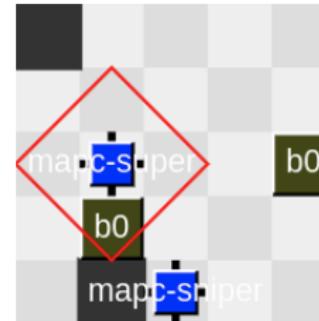
Showroom



- Detach
- Connect
- Disconnect
- Clear

Actions available to the agents

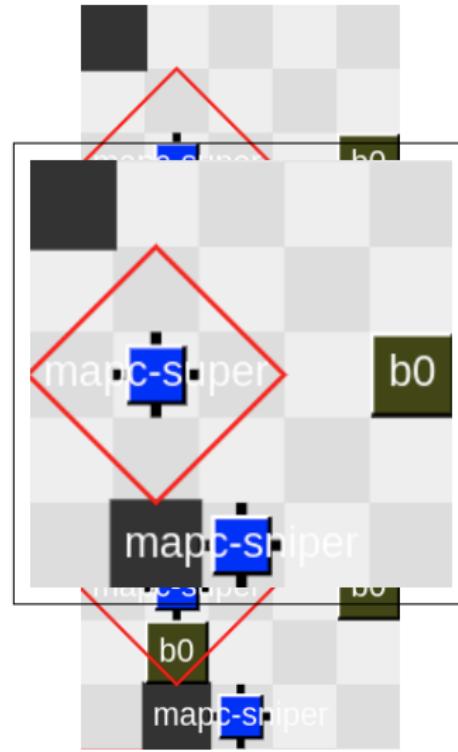
Showroom



- Detach
- Connect
- Disconnect
- Clear

Actions available to the agents

Showroom



- Detach
- Connect
- Disconnect
- Clear

5. Time for Practicing

Challenge I

■ Preliminaries:

- We'll partition all participants into teams (randomly)
- You'll be assigned to a breakout room
 - (up to 4 persons per room)
- Keep in mind the **breakout room id**
 - This will be your **team id!!!**
- Go to the google drive folder
- Open up the file **Challenge.ipnyb**
- Connect to the MASSim server

■ Challenge:

- 1 Cooperate with everyone in your breakout room
- 2 Submit a single task of four blocks

Challenge II

■ Preliminaries:

- We'll partition all participants into 2 teams (randomly)
- Same as before:
 - Breakout room
 - Keep the breakout room id
 - Use the same file in google drive:
`Challenge.ipnyb`
- Connect to the MASSim server

■ Challenge:

- 1 Cooperate with your roommates
- 2 Submit as many tasks as you can!

Play and have fun!!!

6. Wrap-up

In this short course you have learned:

- About the MASSim framework:

In this short course you have learned:

- About the MASSim framework:
 - MASSim server;

In this short course you have learned:

- About the MASSim framework:
 - MASSim server;
 - EISMASSim; and

In this short course you have learned:

- About the MASSim framework:
 - MASSim server;
 - EISMASSim; and
 - MASSim protocol.

In this short course you have learned:

- About the MASSim framework:
 - MASSim server;
 - EISMASSim; and
 - MASSim protocol.
- How to connect properly to the MASSim server;

In this short course you have learned:

- About the MASSim framework:
 - MASSim server;
 - EISMASSim; and
 - MASSim protocol.
- How to connect properly to the MASSim server;
- About the Agents Assemble scenario; and

In this short course you have learned:

- About the MASSim framework:
 - MASSim server;
 - EISMASSim; and
 - MASSim protocol.
- How to connect properly to the MASSim server;
- About the Agents Assemble scenario; and
- Some challenges that come up while **coordinating** agents.

Thank you for your time!

Curious? Excited?

Check out the MAPC! Join and enjoy this contest with us!

multiagentcontest.org

Contact: niklas.fiekas@tu-clausthal.de
tkr19@tu-clausthal.de

- [1] T. Ahlbrecht, J. Dix, N. Fiekas, and T. Krausburg, “Accept a challenge: The multi-agent programming contest,” in *Engineering Multi-Agent Systems*, C. Baroglio, J. F. Hubner, and M. Winikoff, Eds. Cham: Springer International Publishing, 2020, pp. 129–143.
- [2] T. M. Behrens, K. V. Hindriks, and J. Dix, “Towards an environment interface standard for agent platforms,” *Annals of Mathematics and Artificial Intelligence*, vol. 61, pp. 261–295, Apr. 2011. [Online]. Available: <http://dx.doi.org/10.1007/s10472-010-9215-9>
- [3] T. Behrens, K. V. Hindriks, R. H. Bordini, L. Braubach, M. Dastani, J. Dix, J. F. Hübner, and A. Pokahr, *An Interface for Agent-Environment Interaction*. Springer, 2012, ch. 8, pp. 139–158. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28939-2_8