02285 AI and MAS, F25

Programming Project

Thomas Bolander, Andreas Garnæs, Martin Holm Jensen,
Mikkel Birkegaard Andersen, Mathias Kaas-Olsen

Contents

1 General information about the programming project **1**
1.1 Expectations and conditions 1
1.2 Groups and group sizes 2
1.3 Competition 2
1.4 Assessment of the programming project 3

2 Schedule and detailed instructions **3**
2.1 Submission of your competition level 3
2.2 Release of competition levels 4
2.3 Submission of competition results 4
2.4 The competition 4
2.5 Submission of video 5

1 General information about the programming project

The goal of the project is to construct an advanced client for the *MAvis* server using the hospital domain. So essentially the problem is the same as in the warmup assignment, except here we expect you to experiment with advanced techniques and architectures to be able to solve the widest possible range of multiagent levels within the hospital domain efficiently. The client of the warmup assignment is not particularly fine-tuned or efficient, so you are strongly encouraged to rebuild your own client from scratch. You can obviously take inspiration in the warmup assignment, but building your own client from scratch and thinking about how to e.g. represent states efficiently is a very valuable exercise, and will also make it easier for you to afterwards extend the client with more features.

1.1 Expectations and conditions

The project is purposely designed to be very open-ended and flexible, allowing many different group sizes, types of solutions and levels of ambition. Successful implementations of the project range all the way from basic solutions using standard techniques to highly research relevant multi-agent systems. Some overall expectations and conditions:

- **Significantly more advanced than warmup assignment**. A basic extension of the warmup assignment will **not** count as a sufficient solution. You are not going to be able

to create an efficient client by simply further tweaking the heuristics and/or use a slightly modified variant of graph search. What you deliver in the programming project has to be **significantly** more advanced and efficient than what you did in the warmup assignment.

- **Only multiagent**. In the programming project, you will only work with multi-agent levels. You might of course use single-agent levels to try out your algorithms for controlling the individual agents of a multi-agent system, but in the final competition all levels will be multi-agent levels.

- **Workload**. The programming project constitutes approximately 3.75 of the 7.5 ECTS in the course. This means that the expected total workload of the programming project is approximately 100 hours per student (including participating in the competition, preparing the video, etc).

1.2 Groups and group sizes

Ideally you keep your group from the warmup assignment. However, this is not a strict requirement, you are allowed to restructure if you find it necessary. The same conditions regarding group sizes apply as for the warmup assignment: Group sizes can range from 2 to 5 students with 3–4 as the recommended numbers. Groups of 2 will be expected to deliver the same as groups of 3. Groups of 2 students are by default **not encouraged** as it gives you a bit too few resources for a project of this size and level of ambition. Groups of 5 students are also by default not recommended as it can be hard to distribute tasks between you in such a large group. But if you think you can manage and are able to handle your project management efficiently, it will of course give you more ressources to create a nice solution. Groups of 4–5 are expected to deliver a somewhat more substantial and thorough solution to the assignment than a group of 3, however, of course realising that a group of 4 can't necessarily produce 30% more than a group of 3, and similarly when going from 4 to 5.

1.3 Competition

Towards the end of the course there will be a competition where your different clients will compete against each other. You will be given a number of multi-agent levels, and your implemented client should then try to complete each level as fast as possible and using as few joint actions as possible. Some details and conditions:

- **3 minute time-out**. In the competition, the environment server will time out clients after 3 minutes, so your AI only has 3 minutes to complete each level.

- **20.000 joint actions limit**. Your client can use at most 20.000 joint actions to solve a level, otherwise it counts as unsolved.

- **Two scores per level: action score and time score**. Your client will get two scores for each level, an *action score* and a *time score*. The action score is for number of actions in the solution found (fewer actions = better score). The time score is for computation time (less time = better score). The time score is logarithmic to avoid groups being too harshly penalised for running their client on less powerful hardware, or in a computationally less efficient programming language.

- **Prizes and winners**. Time and action scores will be summed up and there will be prizes for the groups with the best overall action score and the best overall time score. There will also be a prize for the most interesting level submitted to the competition. As an unsolved level gives a score of 0, your first priority in designing the solution should always be to be able to solve as many levels as possible within 3 minutes and 20.000 joint actions.

1.4 Assessment of the programming project

The assessment of the programming project will be based on

- **Algorithmic ideas**. The novelty of the algorithmic ideas that you have come up with. How interesting are they, how general are they, how well do they perform in your implementation?
- **Video presentation**. How clear and technically precise have you presented your algorithmic ideas? How well have you followed the instructions in `guide_for_video.pdf`.

The goal of the competition is to motivate you to make the most efficient possible solution, and to allow comparison of the strengths and weaknesses of the various submitted solutions. Your result in the competition will not directly affect your grade. A group might for instance challenge itself by trying out advanced novel techniques or focus mostly on optimising a certain aspect of the solution, e.g. multi-agent communication and coordination. That group might not be able to produce the most efficient solution, but the novelty and quality of both the algorithmic ideas and the video could still be very high.

2 Schedule and detailed instructions

2.1 Submission of your competition level

You should design and submit one multi-agent level (i.e., containing at least 2 agents). The deadline is

13 May at 20.00

The submitted level will be included in the set of competition levels, but only if it meets *all* of the following conditions:

1. **Level size limit 50x50**. Your level is required to be of size at most 50x50 grid cells. No tab characters.
2. **Correct format**. The level satisfies all format requirements in `hospital_domain.pdf`.
3. **Solvable by own client**. Your own client can solve the level within the 20.000 joint actions limit and 3 minutes limit.
4. **Correct naming**. Pick a group name of at most 9 letters (restricted to the letters `a-z` and `A-Z`), e.g. `TheMAtrix`, `MASochist` or `NoOp`. This will be your group name in the competition, so choose wisely! Your file name must then be `<group_name>.lvl`, e.g. `TheMAtrix.lvl`. Important: The name of the level after the `#levelname` line inside the file must also be `<group_name>`!

2.2 Release of competition levels

The competition levels will be made available in `complevels.zip` under `Content/Programming Project` on DTU Learn on

15 May at 20.00

Download and unpack the zip file in a separate directory and then run the server with the following command (note that each level takes up to 3 minutes, so this command could run for several hours):

```
java -jar server.jar -c "<client-cmd>" -l "<path-to-complevels-dir>"
                     -t 180 -o "<group-name>"
```

Here `<group-name>` must be the same as you used when submitting your competition level (see Section 2.1. It produces these files:

- `<group-name>.zip`: log files for all the competition levels. Unzip to check how you did. To replay a log file, use

  ```
  java -jar server.jar -r "<path-to-logfile>" -g
  ```

  Check that your client is correctly identifying itself with your group name by inspecting the top bar in the replay window. It should say "Client: ¡group-name¿".

- `summary.txt`: a summary of the actions and time spent on each level.

- `<group-name>.out`: an encrypted file with the competition results. This is the one to be handed in, see Section 2.3.

2.3 Submission of competition results

The deadline for submitting the results of running the competition levels on your client as well as your source code is

19 May at 20.00

To be handed in:

1. `<group-name>.out`: The file produced by running the server on the competition levels, see Section 2.2. Make sure your client identifies itself with your group name, see Section 2.2.

2. `code.zip`: A zip-file containing all the source code and executable code of your implemented software. No `.rar`, `.tar`, `.gz` or other formats, please!

2.4 The competition

The presentation of the competition results will take place on

22 May 14.00-18.00

It doesn't take place in the standard auditorium, so check the schedule for which room it is. Requirements:

- **1 minute presentations**. Each group should be prepared to give a short 1 minute explanation of the behaviour of their client on any of the competition levels.
- **Required presence**. It's the official day of examination in the course, so by default everybody should be able to participate. In rare cases, some students have other exams on that day, and can of course be excused. Just make sure that there is at least one representative from your group present.

During the event, we will present the detailed scores of all groups, present head-to-head battles between pairs of clients running on the same level, and announce winners. There will be snacks, drinks and prizes.

2.5 Submission of video

The deadline for submitting your video is

3 June at 20.00

Requirements:

1. **Follow the official video instructions**. Make sure to read and follow all the instructions and advice given in the separate document `guide_for_video.pdf`.
2. **Include a group declaration**. Your first slide should include a *group declaration* specifying who did what in the project, in terms of ideas, literature search, implementation, preparation of the video, etc (with the same overall conditions as for the warmup assignment).
3. **Include relevant demos**. Include relevant demos in your video, as a minimum a video demo of your client running on your own level, including an analysis of its behaviour (why is it doing what it's doing? How could it potentially be improved?).
4. **Maximal length is 12 minutes**. The length of the video should be **at most 12 minutes** for groups of *any size*. Anything after the 12 minute mark will not be watched and assessed.

Good luck with your project. Be creative! Have fun!