

MPIA–Artificial Intelligence in Games: Lab Project 2

The objective of this lab project is to program both a simple planning and a path finding algorithms and make them work together. Your implementation should also be able to display the programmed behaviours of your agent.



You will create a game environment on Unreal¹ with at least one NPC and one player. The game must simulate a small city (or similar environment) seen from above. The city contains victims and hospitals. The NPC, as well as the player, are the rescuers in the city. They collaborate to get all the victims to hospitals as quickly as possible while following the available paths and avoiding collisions with other victims and rescuers.

There must be at least one hospital in the city and there can be many victims, that are placed randomly on the map at the beginning of the game. Once the game starts, the NPC and the player know where the victims and the hospitals are, as they can see the whole map from above. They can carry one victim at a time to the hospital. They can also see what each other is doing, but they cannot communicate to coordinate their actions.

You will have to implement a simple planner so that the NPC chooses what victim will be rescued first. You will also have to implement a path finder and the steering behaviours to drive the NPC to the chosen victim, and then to the nearest hospital. Since the objective is to rescue the victims as quickly as possible, if that victim is rescued by another NPC or the player, the current NPC must choose another victim to rescue as soon as possible. The behaviour of the NPC must seem intelligent, but it does not need to be the absolute optimal behaviour.

Note that the city in the example above is very nice, but you do not need to do something so detailed. For example, a bunch of cubes simulating the city blocks will do. Also note that your instructor will evaluate the algorithms, not the graphics. So, be sure to make that part of your project work well and do not loose too much time with the looks of your game.

What to submit

You must create a GitHub project for your implementation. The submission will be done by creating a “release” of your implementation and then uploading a text file containing the link to that release to Moodle. Your release should include a “readme” file with instructions on how to install and run your implementation on Unreal.

Note: be careful to not make changes on the release once it has been submitted, for the date of the last change will be considered as the submission date of your project.

¹You must ask the permission of your instructor if you want to use a different platform.

Defense

You will have to defend your project. In the defense, you will have to show the program running, explain its features, explain the code, and answer some questions made by the instructor.