TÉCNICO LISBOA

AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS

2020

PROJECT v1

Address a real-world problem recurring to an intelligent system in one of two lines: a) autonomous agents, or b) multi-agent systems. Students are free to choose the programming language to engineer such system.

1. AGENTS

1.1 AUTONOMOUS AGENT SYSTEM

Projects focused on designing and implementing a single agent, optionally considering forms of interaction with a user. The project needs to go beyond what is done in the AI course, including several of the following:

- complex environment;
- deliberative, adaptive, curious or/and emotional behavior;
- diversity of goals, sensors and actuators;
- uncertainty and noise in the system;
- rich interactions and feedback.

1.2 MULTI-AGENT SYSTEM (MAS)

Projects focused on designing and implementing a multiagent system, optionally considering different forms of cooperation. RTS games (and peer problems) are not acceptable if each agent centrally controls all the units since this setting is rather an AI project. The project needs to include several of the following:

- agents should have either *conflicting goals* or face complex *coordination problems*;
- the variety of sensors and actuators *should not be too limited*;
- communication and coordination mechanisms (cooperation, negotiation, team formation, arguing);
- pursue of goals at the expense of another (*non-linear decisions*) without a fixed order of priority;

2. OBJECTIVES

The standard objectives for all projects imply the implementation of the following components:

1. **Modeling the (multi)-agent system:**
    - definition of the **problem**: requirements and objectives;
    - specification of **agent properties**: sensors, actuators, environment, decision making behavior;
    - design choices regarding the **architecture** for the implementation of the agent system;
2. **Advanced decision making**: depending on the system, agents may consider deliberative behavior or advanced learning principles (e.g. reinforcement learning or incremental learning);
3. **Engineering the agent system** considering different behaviors (baseline versus advanced behaviors);
4. **Comparative analysis** (*very important*): thorough analysis of the behaviors of the agents for each implemented approach (detailed experiments, tables, graphs, etc.). Discussion of the suitability of the different approaches to address the target problem in accordance with the gathered results.

3. Evaluation Criteria

The project will be evaluated according to the following criteria:

- Clear and complete description of the problem that is being addressed;
- Conceptual correction of the simulation environment and agent behavior;
- Suitability and correctness of any embedded mechanisms of emotions, team-work, learning, etc.;
- Selected approach and architectural principles to address the problem;
- Intelligent, emergent agent behavior;
- Adequacy and completion of the undertaken experimental analyzes (gathered results);
- Relevance of conclusions inferred from empirical results;
- Proper positioning and communication of the findings (report and accompanying video).

4. Final Delivery

4.1 Project Proposal and Presentation

Project proposal by Friday, **March 30th** with a two-page report according to the following template:
https://fenix.tecnico.ulisboa.pt/downloadFile/1970943312341693/ACMWordTemplate.doc with 3 succinct sections:

1. problem definition, relevance and requirements
2. proposal of an (multi)agent system and its adequacy to answer/study the problem
3. properties of the agent(s) and environment in the proposed intelligent system

6-minute presentation in the labs after the Eastern break

4.2 Final Delivery

Final delivery of the project by Monday, **May 25th** with the following elements:

4. **Full source code** and **executable** with a README on how to run the system
5. **Final report** (5-to-8 pages) according to the following template:
   http://aamas2019.encs.concordia.ca/AAMAS19Template.zip (*latex*)
   https://fenix.tecnico.ulisboa.pt/downloadFile/1970943312341693/ACMWordTemplate.doc (*word*)
6. A 2-to-5 minutes **video** demonstrating the agents or algorithms *in-action*, e.g.:
   a. The emergence of some social phenomenon in a population;
   b. The effect of learning new behavior (before and after learning);
   c. The contagion of emotions in a population;
   d. The "team-behavior" exhibited by some group of agents;
   e. A comparison of the agents' behavior between the several approaches.

***Important*: there will be no changes to the given deadlines.**

5. Assessment

Mark is individual. Students should provide the contribution of each member if there are unequal efforts.
The evaluation of the project will consider:

15% Project proposal and presentation
15% Adequacy and correctness of the proposed intelligent system to address the target problem
20% Sound and complete implementation using suitable architectural principles
25% Experimental analyzes, discussion and findings
15% Project description, positioning and communication (report)
10% Video presentation

6. Few project examples

6.1 Autonomous Agents

6.1.1 Collaborative game with a person

Check Sueca http://gaips.inesc-id.pt/parceiro/, Fireboy and Watergirl http://www.fireboynwatergirl.com/. In these settings, agents need to be proactive and adapt to the user. Visuals and game engines will not be evaluated.

6.1.2 Logic-geometric programming

Optimize joint motions of multiple agents to solve cooperative sequential manipulation tasks, which require planning at symbolic and motion level. https://ipvs.informatik.uni-stuttgart.de/mlr/papers/15-toussaint-IJCAI.pdf
https://ipvs.informatik.uni-stuttgart.de/mlr/marc/source-code/16-LGP.tgz

6.2 Multi-Agent Systems

6.2.1 Geometry Friends

Implement a two player cooperative puzzle game developed at GAIPS: http://gaips.inesc-id.pt/geometryfriends/

6.2.2 Sustainable mobility

Model a transportation network using a multi-agent system to understand the impact of routes and span of buses (or the positioning and pricing of bicycles and scooters) for improving urban mobility.

6.2.3 Emergency responses

Use (multi)agents to effectively distribute ambulances and resources to efficiently respond to emergency responses in accordance with the type, gravity, and spatiotemporal distribution of emergency requests.

6.2.4 Traffic with autonomous vehicles *(not suggested for this semester)*

Study consequences of vehicles becoming autonomous: how to improve traffic when cars are autonomous. https://www.youtube.com/watch?v=iHzzSao6ypE&t=148s , http://www.am-lisboa.pt/101000/1/004072,012016/index.htm

Model a circulation network; research and implement human factors affecting driving; study the impact of different control architectures (local vs. global), automatic coordination between vehicles, percentage of autonomous cars in the circulation, traffic lights, size and interconnection of the network, vehicle priority, and cost/payment systems. You can extend (or get inspiration) from the Traffic Grid model at Netlogo.

6.2.5 Logistics *(not suggested for this semester)*

Consider *N* companies receiving requests to transport different products from a location to another using trucks with a limited capacity. We want to understand how to guide a company, and the aspects that lead to the arise of coordination, formation of coalitions, monopolies, and side-payment schemes. Model a network with enough complexity (different suppliers, clients, orders and delivery mechanisms); study different forms of control architectures for each supplier and its impacts (e.g. reactive vs non-reactive); study the impact of service modalities, pricing (cost/payment systems), number of companies and clients.

6.2.6 Collaborative AI in Minecraft

https://www.microsoft.com/en-us/research/academic-program/collaborative-ai-challenge/

6.2.7 UAV-based surveillance system for fire prevention

Design a system to collaboratively plan routes for forest surveillance in accordance with fire occurrence risks. The environment can be model as a heat map with probabilities of fire risk changing along time. A set of drones (UAVs) should be able to autonomously establish efforts to cover areas with higher risk of fire in accordance with a specific utility function. The project must guarantee coordinated behavior in the presence of specific constraints, including: 1) kinematic constraints on speed and rotation of UAVs; 2) number and time availability of UAVs; 3) UAV autonomy; 4) irregularities on terrain surface; and 5) visibility constraints. A project template to develop such multi-agent system in Java can be found in the course's webpage.

6.2.8 Language games and the evolution of language

http://www.nature.com/nature/journal/v417/n6889/abs/nature00771.html
http://ieeexplore.ieee.org/abstract/document/956077/