Intelligent Agents

Assessed CourseWork: Planetary Rovers

Julian Padget – j.a.padget@bath.ac.uk

Covid-secure in-person teaching

Please:

- if you have symptoms, go home: isolate, test, tell
- wear a face covering (unless you are exempt)
- respect each other's personal space
- do not move furniture

Be safe. Be kind.
Belong. together

UNIVERSITY OF
BATH

#BelongatBath

Assessed Coursework: Planetary Rovers

- Design and implement agents that collaborate to explore a simulated environment
- Learning objectives:
  - Develop agents using the Jason platform
  - Build on existing code
  - Explain, evaluate and reflect upon your decisions
- The coursework counts for 50% of the unit (UG), 40% for PGT, and is expected to take about 40 hours. However, programming is a skill and agent programs are declarative, not procedural, so require a different set of techniques. Different students will spend very different amounts of time.

Coursework published: **Friday week 3**
Code due: **20:00 Friday week 10**.
Video due: **20:00 Friday week 11**.

Backstory

The situation is a simulated planetary exploration, for which you must develop “rover” agents to explore, locate resources, and return them to base. Agents can run out of energy and stop. A run ends when all agents stop, or all resources have been collected. There are different scenarios of increasing difficulty that place different demands on your team of rovers, where:

- Rover agents start at a base at a randomly selected location
- Each agent is given a limited amount of energy
- Resources and obstacles are put randomly in the environment

In the later scenarios, there is more than one resource type to collect, but a given rover can only collect one type. Rover communication and collaboration are essential for rover team effectiveness. Each run generates a fresh map, so initial position, resource location and obstacle location are different each time.

The Basic System

- You are provided with a basic system from which to start the coursework. See github link on Moodle
- This system consists of:
  - An initial Jason platform configuration, that sets up the environment and simulation
  - A skeleton Jason agent, that can interact with the environment
  - An information display of the state of the environment
  - Test scenarios
- Support will be provided for installation and configuration through videos and lab sessions

Functional Requirements

1. Extend the basic rover agent to create an agent or agents that can successfully gather resources in the different scenarios (see slide 7)
2. Agent to agent communication must use the Jason communication primitives

Non-functional Requirements

1. The program must be written in AgentSpeak, primarily, and Java, for any additional agent actions. The use of Java should be limited to the implementation of agent navigation only.
2. The program should work correctly on the Jason platform provided.
3. Only the provided platform configuration may be used, in conjunction with rover agents built on the skeleton agent.

Scenarios I

S1. (0%) One agent, one resource type, plentiful resources inclusive of high amounts at one location, small map, high energy budget.

Aim: interact with the environment, and track visited locations. This is not marked, but you can get assistance in the lab for this scenario

S2. (15%) Two agents, one resource type, plentiful resources with high amounts at one location, medium or large map, low energy budget.

Aim: interact with the environment, track visited locations, agent communication and collaboration.

Scenarios II

S3. (20%) Three or four agents, two resource types, sparse resources with high amounts at one location sometimes of different type, large map, low or moderate energy budget.

Aim: agent communication, collaboration and planning. At least one agent must be a dedicated scanner with no carrying capacity

S4. (20%) Four agents, two resource types, moderate resources inclusive of high amounts at one location sometimes of different type, moderate obstacles – some resource stacks will be located close to obstacles, medium map, moderate energy budget.

Aim: cope with obstacles, agent communication, collaboration and planning. At least one agent must be a dedicated scanner with no carrying capacity

Scenarios III

S5. (25%) Five agents, two resource types, sparse resources with varying stack configuration, moderate energy budget, obstacles are designed to mimic a maze on a medium or large map, each agent can only enter the maze once and the base will be at the exit of the maze. Agents must have a carrying capacity of at least 5 to be able to collect all the resources on the map.

Aim: agent communication, collaboration, planning, managing limited resources (energy), navigating a maze.

Scenarios IV

S6. (20%) Competitive. A scenario where you submit two agents to compete against agents of other students in the class, a large map, moderate resources with varying stack configuration, high energy budget and several obstacles. One agent to collect each resource type.

Aim: coping in an environment when there is competition for resources and a fixed number of timesteps.

Functional assessment – 50% of component mark

- Agent performance is assessed by the number of resources collected. If your agent collects all the resources in scenario 1, that's 15%.
- The mark for each scenario is:

$$\frac{\sum \text{resource collected}}{\sum \text{resources available}} \times \text{scenario weighting}$$

- The marks for each scenario will be the average score of running the scenario 5 times
- The set of maps will be the same for each agent in a given scenario so each submission's agent(s) face the same situations
- Functional assessment will use Unidesk: these are the only marks that will count. Make sure your submission delivers the same results on Unidesk as in your development environment.

Video presentation – 50% of component mark

The video presentation shall be a screen capture and must address the following 5 issues:

1. A detailed discussion of the different kinds of agents you implemented
2. A justification for the decisions made
3. A critical analysis of your implementation
4. Directions for further improvements
5. Evidence from and discussion of the performance on the published scenarios

You may choose to use slides with voice-over, plus demos to illustrate points, but the choice content is yours. The submission will be assessed on how effectively you have addressed the 5 issues identified above. Maximum video duration is 10 minutes. Overlong videos are capped at 50%. Sped up videos will get a mark of zero.

Conditions

- The coursework is to be carried out individually
- Attention is drawn to the University and Departmental rules on plagiarism (see the Programmes Handbook)
- Support for the coursework will be available during the lab sessions
- Offline support is via the use of Moodle forums
- The coursework also requires work during private study time
- This coursework is not marked anonymously due to the presentation element

Presentation marking guidelines

- 1st. The presentation reaches 2:1 standard. In addition, it shows a deep understanding of the problem and ideas from background reading. Information is presented clearly and concisely, and shows an appreciation of other areas of knowledge and sources of expertise.
- 2:1. The problem is clearly well understood. Evidence of the identification of background reading sources is shown. Rationale is presented concisely and completely.
- 2:2. The presentation shows how the student addressed the problem in line with their skills and understanding, along with some deeper understanding of problem domain. Description and justifications indicate effort to make the different kinds of agent meet the requirements in the time-frame.
- 3rd. The presentation explains the supplied basic rover agent. The description reports only naive strategies, and only simple ways of moving beyond the code provided are discussed.

Coursework Submission

There are two submission points corresponding to the two deliverables for this assignment: one is for code, one for video.

1. Submission of your rover code as a zip file containing:
   - A directory containing the source code for your agents.
   - A directory containing the source code for internal actions developed (where necessary).
   - A mas2j file for each scenario.
   - Upload this to the moodle submission activity
2. Submission of the video of your presentation has two steps:
   - Upload the recording to the Re:View assignment folder.
   - In the Re:View assignment folder, choose 'Share', copy the link, then paste and submit the link to the recording in the Moodle assignment submission point.

It would be helpful for video filenames to identify the submitter, in the form of `firstname_lastname.mp4`, rather than `agents_video.mp4` so marks go to the right person.