Computer and Information Sciences
University of Strathclyde

Assignment 1 – PDDL Craftbots – Worth 25% of Overall Mark

There are 2 parts to this assignment, a Simple Config, and a Resource config.

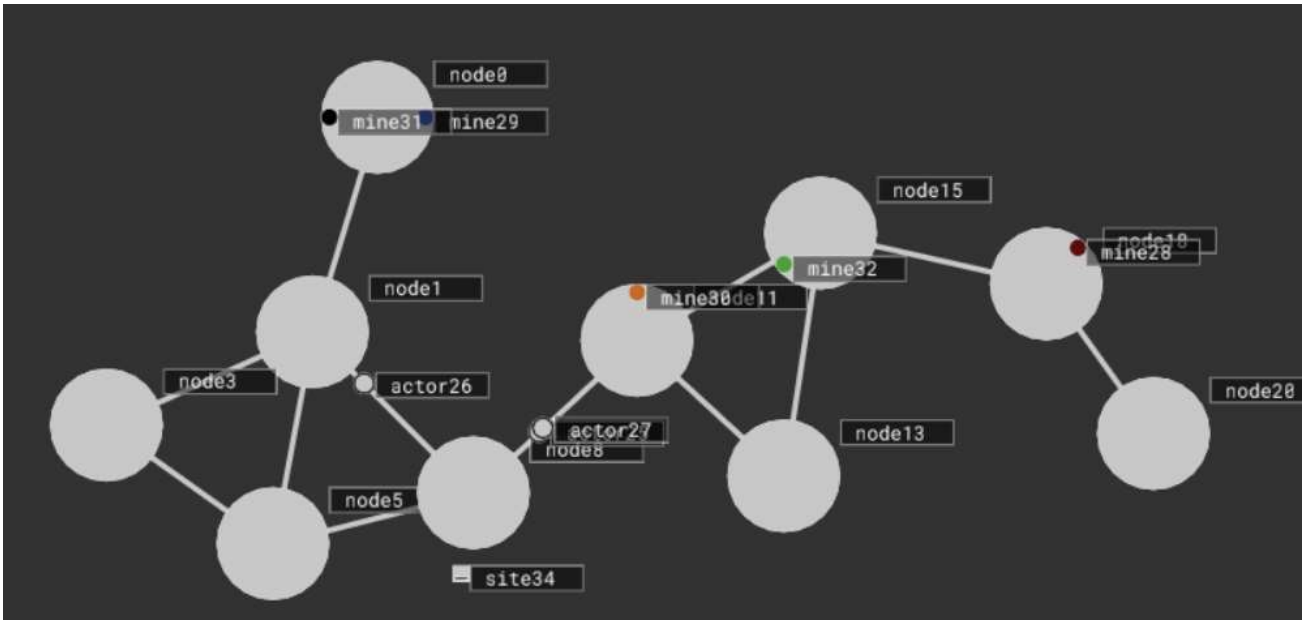
Submission deadline: 14/11/22

Online submission through MyPlace

Background

CraftBots is a light-weight simulation environment with properties that can be configured to increase difficulty or introduce particular problem features. The simulation is written in Python and should run on low-end machines. This section will describe where to find the simulation, how to set it up, and the configurations that we will use in this coursework. The simulation is documented on the GitHub Wiki:

https://github.com/strathclyde-artificial-intelligence/craft-bots/wiki.



The simulation is shown above, which models the same scenario setup that we will use in this coursework. The scenario includes 3 actors (solid grey circles) moving between 10 locations (large white circles) connected by edges (blue lines). The aim of the scenario is for the actors to use the resources to construct as many buildings as possible.

Resources can be gathered at mines (coloured dots) by mining and carried by actors, using pick up, deposit, and drop. Resources come in five types: red, green, blue, black, and orange. Actors can start construction at any location. Actors should deposit resources into partially constructed buildings (purple polygons) and then spend time constructing, until the building is complete.

Buildings require different resources, and completing a building will improve the score, with the aim being to complete as many buildings as possible. Your task will be to create an agent that will control these actors, using PDDL and Python programming.

Created October 2022
By Dr Andrew Abel, modified from a version from Dr Michael Cashmore

Computer and Information Sciences
University of Strathclyde

Scenario Configuration: The coursework will use two configurations. The Simple Config will generate 6 random tasks to be solved by the agent. The Resource Config will enable additional tasks, deadlines, and some additional resource properties. Further set up advice will be given later in this assignment sheet.

Part 1: Planning for the Simple Config (30 marks)

The simple config will create 10 randomly generated nodes, and 6 fixed tasks (again, randomly generated). Each task will require a different number of resources to solve, and you must solve the problem using a combination of PDDL programming and Python programming. There are a number of steps in this process:

1.1 Write a PDDL Domain for the Simple Config (10 marks)

This part of the assignment is to write a PDDL domain for the CraftBots simulation Simple Config.

• Marks are gained for correctly modelling the domain such that the actors can successfully move, collect resources, start construction, and complete tasks.

• The actions in the domain model should match the available actions in the API. You do not have to model all of the actions if the goals can be achieved without them.

• The domain must be compatible with the world info provided in the program

1.2 Automatically Generate the PDDL problem file using the simulation API (6 marks)

This part of the assignment is to automatically generate the PDDL problem using the simulation API. Marks will be gained for correctly including the static elements of the scenario, the initial state, and the goal. A template class and function header has been provided to assist with writing (see guidance).

1.3 Create a valid PDDL problem (6 marks)

The aim here is to create a PDDL problem that is compatible with your PDDL domain, and the program. This can either be created by hand, or generated by the program (See above). The problem file must be able to generate a plan using your domain file and a planner.

1.4 Execute the Plan (8 marks)

Modify the template agent to also implement your PDDL plan, as well as generate one. It should be able to solve the generated problem, and execute all the actions using the API. A good agent is one that will be able to solve all the tasks by the end of the simulation (the tasks have no time limit).

Part 2: Planning for the Resource Config (20 marks)

The simple config will create 10 randomly generated nodes, and 6 (randomly generated) tasks. Each task will require a different number of resources to solve, and additional tasks will generate throughout the simulation. There are also a number of resource properties which need to be considered:

```
Resource.RED    = 0 # can only be collected within time windows
Resource.BLUE   = 1 # takes longer to collect
Resource.ORANGE = 2 # requires multiple actors to mine
Resource.BLACK  = 3 # cannot be carried with any other resource
```

(there is also a green resource property, but we will not use this)

Created October 2022
By Dr Andrew Abel, modified from a version from Dr Michael Cashmore

Similar to the problem above, you must solve the problem using a combination of PDDL programming and Python programming.

2.1 Write a PDDL Domain for the Advanced Config (10 marks)

This part of the assignment is to write a PDDL domain for the CraftBots simulation Resource Config.

• Marks are gained for correctly modelling the domain such that the actors can successfully move, collect resources, start construction, and complete tasks.

• You must use temporal planning

• The domain must be compatible with the world info provided in the program

2.2 Automatically Generate the PDDL problem file using the simulation API (6 marks)

This part of the assignment is to automatically generate the PDDL problem using the simulation API. Marks will be gained for correctly including the static elements of the scenario, the initial state, and the goal. A template class and function header has been provided to assist with writing (see guidance).

2.3 Execute the Plan (4 marks)

Modify the template agent to also implement your PDDL plan, as well as generate one. It should be able to solve the generated problem, and execute all the actions using the API.

Submission Instructions

You must submit 2 zip files:

1. one zip file containing a domain, a plan, a problem, your completed agent, and your completed interface for part one.

2. one zip file containing a domain, a plan, a problem, your completed agent, and your completed interface for part one.

Craftbots Guidance

Getting Started

More information and the initial files are provided at:

https://github.com/strathclyde-artificial-intelligence/craft-bots/wiki

You need to clone the Github repository (download it), install the required dependencies, and then run the python files. Initially the file to run is main.py.

There are two agents initially installed, a TestAgent (random movement), and a rule based agent. You can see a demonstration of the rule based agent by choosing it in the main.py file:

```
agent = RBAgent()
sim.agents.append(agent)
```

If not already selected, you can use the simple config to run the initial program:

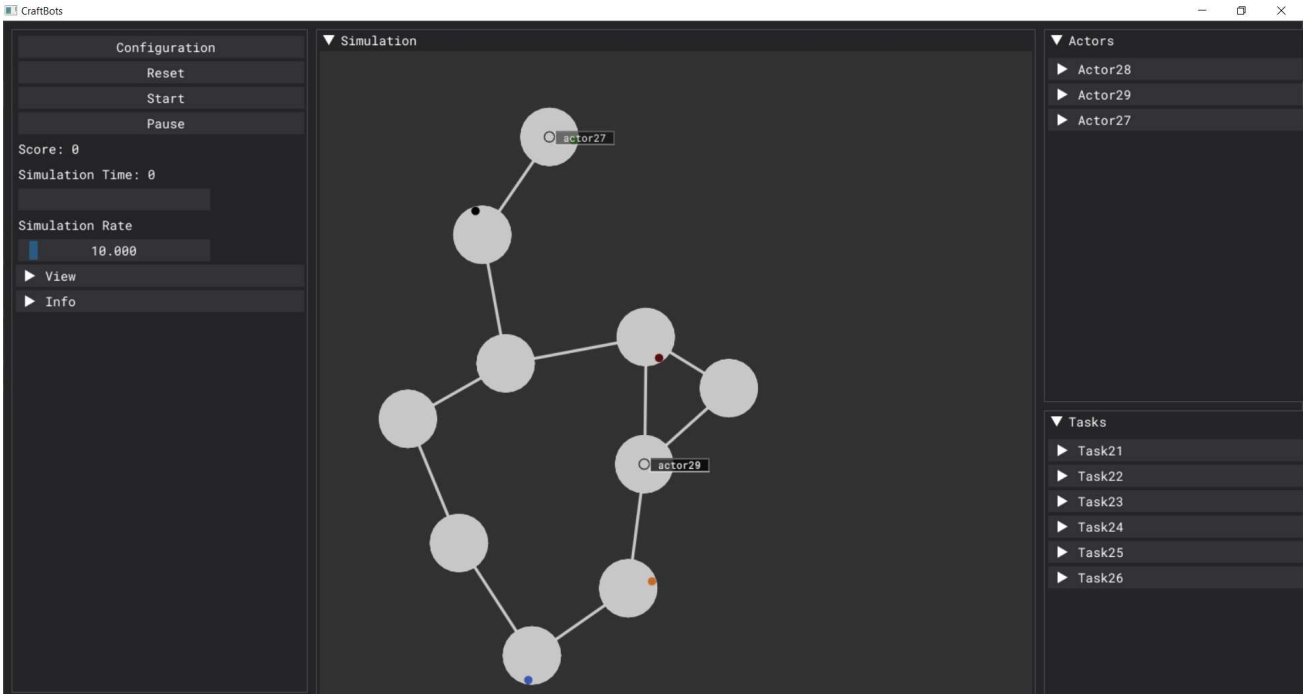
```
arg_parser.add_argument('-f', help="configuration file", type=str,
default='craftbots/config/simple_configuration.yaml')
```

You should then be able to run your program.

Created October 2022
By Dr Andrew Abel, modified from a version from Dr Michael Cashmore

Computer and Information Sciences
University of Strathclyde

The interface



To run the simulation press the "Reset" button, and then "Start". The program should then run. You have a number of options to experiment with.

Useful Files

There are a number of useful files within the initial project.

1. Configuration files give the properties of different configurations. These are .yaml files within the Config folder
2. Agents. There are 2 agents within the current project, a rule based agent, and a random movement agent, both can be selected

Project Files

There is a zip file on MyPlace that contains a number of files to get you started. It contains:

1. the two different configurations (part1 and part2), which you can add to the config folder.
2. A template agent, with 2 functions to test, and one function, send_action, to complete. Add this to the agents folder.
3. A PDDL interface file, with 2 completed functions, and a function to complete, writeProblem. Add this to the agents folder.
4. An incomplete domain file. Add this to the agents folder.

Created October 2022
By Dr Andrew Abel, modified from a version from Dr Michael Cashmore