6CCS3ML1 and 6CCS3PRE Coursework 2 FAQ

Adrian Salazar and Yani

1 COURSEWORK OVERVIEW

This coursework exercise requires you to implement Q-learning to control Pacman's movement.

2 FREQUENTLY ASKED QUESTIONS

2.1 What do I need to submit?

Your submission should consist of a single ZIP file. (KEATS will be configured to only accept a single file.) This ZIP file must include a single Python file (your code called `mlLearningAgents.py`). The ZIP file must be named: `cw2-<lastname>-<firstname>.zip`

Submitting in a different format will lead to points being deducted.

DO NOT SUBMIT THE `mlLearningAgents.pyc` FILE! That is the compiled file, not the human readable file.

2.2 Which map should I use to test my agent?

Your code will be evaluated on smallGrid and another secret map. Your agent needs to win 8 out of 10 games on the smallGrid.

2.3 Why are you running it on a secret map?

This is done so we can check if your agent's behaviour is hard coded.

2.4 What does hard coded mean?

Hard coded behaviour means that the agent has a predefined set of actions he will be making, and not deciding based on what the Q-learning algorithm is telling it to do.

2.5 I am worried that the uncertainty of the environment will affect my grade. What will you do about that?

RL approaches are specifically designed to behave well in environments with high uncertainty. A well implemented Q-learning agent will win all 10 games most of the time (the map is really small and the win condition is really easy to achieve). However, just to be sure, any agent that will not reach the requested win ratio will be re-run. If it does not reach the threshold the second time, then you will lose some marks.

2.6 How should my code be styled?

You should comment your code and have a consistent style over all the file and a good separation of tasks in methods and classes.

2.7 Can I use code that I find online?

Depends if you do significant changes to it. Just copy/pasting will lead to marks being deducted or even to a review by the department's plagiarism committee. You will need to put references in your code where this happens and specify where you took it from.

2.8 What is that `final()` method?

When running several games in a run (i.e. you use the `-n` parameter), the `__init__()` will only be called at the beginning, when the object is created. As such, you will need to use the `final()` method to reset your values for the next game to the initial values (of course, that if it is needed).

2.9 What libraries can I use?

You can use the standard python libraries. If you do not know what the standard python libraries are, here is a list with all of them: [LIST]. Additionally, you are allowed to use other basic libraries such as `numpy` and `pandas`.

2.10 What is a good coding style?

Depending on who you ask, this answer is going to be different. Generally speaking, the most important aspect for this matter is readability. We ask that by default you observe the following requirements in order to be able to reach the full set of marks for style (i.e., you will lose marks if you do not): Variable naming style should be set to PascalCase for class names; UPPER_CASE for constants; camelCase for others. Also, ensure you make good use of whitespaces (e.g., you typically add a whitespace before and after an operator such as '<')

2.11 How can I get the maximum grade in the comments part?

Good comments should explain how the different parts of your implementation work. These can be high-level explanations with some references to theory. If you are using functions, make sure to explain their input parameters.

2.12 Can I reuse my code from the 6CCS3AIN Coursework?

You can reuse parts of the code you wrote for that coursework, but it will need significant changes for it to get a good mark. Just resubmitting your MDP agent will not get you many points.