COM00071M

UNIVERSITY of York

MEng/MMath/MSc Degree Examinations 2017/18

DEPARTMENT OF COMPUTER SCIENCE

Evolutionary Computation (EVCO)

Individual Assessment

Issued: 25th October 2017

Submission due: 12 noon, 24th January 2018

Feedback and marks due: 21st February 2018

All students should submit their answers through the electronic submission system: http://www.cs.york.ac.uk/student/assessment/submit/
by **12 noon, 24th January, 2018**. An assessment (or part of an assessment) submitted after this deadline will be marked initially as if it had been handed in on time, but the Board of Examiners will normally apply a lateness penalty to the whole assessment.

The feedback and marks date is guided by departmental policy but, in exceptional cases, there may be a delay. In these cases, all students expecting feedback will be emailed by the module owner with a revised feedback date. The date that students can expect to see their feedback is published on the module descriptor: http://www.cs.york.ac.uk/modules/

Your attention is drawn to the section about Academic Misconduct in your Departmental Handbook: https://www.cs.york.ac.uk/student/handbook/

Any queries on this assessment should be addressed to Dr. Dan Franks: daniel.franks@york.ac.uk
Answers that apply to all students will be posted on the EVCO VLE.

Rubric:
Note there is a page limit of 10 pages for the report, using size 12 Arial font and a minimum of 2cm margins on all four sides for the report. Parts of answers that go beyond the page limit may not be marked. References must be listed at the end of the document and do not count towards page limits.

Your exam number should be on the front cover of your assessment. You should not be otherwise identified anywhere on your submission.

Evolve a Player for the Video Game 'Snake'

The Scenario

Snake is a classic video game where the player controls a snake moving around a grid. In each time-step the snake moves forward in the direction it is facing. The aim of the game is for the snake to collect food that appears in random positions around the grid. Each food item increases the score by one. The game is over if the head of the snake collides with its own tail or with the wall around the grid (or if it has been too long since it last ate). Each food item eaten makes the snake longer, and thus the game becomes increasingly difficult. You are provided with a playable version of the game on the VLE, to become familiar with the gameplay.

Your Task

Design an evolutionary algorithm in Python (of your own design and choosing) using the DEAP package to create an agent to play this game. Your agent should attempt to gain as high score as possible.

What you are expected to produce and submit:

a) A 10-page report in the style of an academic journal paper
b) A file containing the Python code for your evolutionary algorithm

What to include in the 10-page report

You are expected to show a systematic approach to the above investigation and communicate your findings effectively and methodically. You should write an introduction, detail your solution space and representation, provide details of your evolutionary algorithm including the effect of different decisions and parameter settings, show results in terms of performance along with your analysis and then present conclusions, interpretation, critical analysis, and discussion. You should provide all the information you believe necessary and use appropriate references throughout to demonstrate evidence of engagement with (and critical understanding of) the literature.

The main part of your report not exceed 10 pages (see Ruberic). This includes any tables and graphs, but excludes any front page, table of contents, references and any appendices. Marks are given out of 100.

The Rules

- You *cannot* manually add any further movement options to the snake. It can only change direction using the provided commands.
- You *cannot* hard-code solutions for the snake. They must be found by evolutionary computation.
- You *can* add any environmental sensing functions that you like to the snake (from the full state of the grid, down to local sensing).

The Files Provided

Go to the exam page on the EVCO VLE and download the two files provided. These are:

a) `snakePlay.py`
b) `snakeProblem.py`

File (a) will allow you to familiarize yourself with the game by playing it. Do this by executing the game from the terminal with $ `python snakePlay.py`

File (b) code on which you should build your algorithm. Snake movement commands are already included, along with a simple example of how to sense the environment.

General marking outline <u>for students</u>:

1) Suitable code and solution for the evolutionary algorithm using Python and DEAP and a good final solution and runs on a test set when compiled [5 marks]

2) Introduction [15 marks]
- Well written and clear
- Connects well with the literature and provides relevant references
- Demonstrates synthesis of the literature
- Demonstrates critical ability
- Makes the problem and challenges clear
- Highlights the approach that will be used and justifies it

3) Methods [35 marks]
Quality of the design of the algorithm (e.g. choice of representation, fitness assessment, other details) to produce a snake AI. Give full details of the algorithm choices. You should provide details and good rationale for your choices.

4) Results [35 marks]
Use your evolutionary algorithm to find as good a solution as possible. Evaluate your snake(s) and your algorithms and use appropriate statistical methods in your reporting

5) Conclusions [10 marks]
- Well written and clear
- Summarizes findings well and highlights key points
- Connects well with the literature and provides relevant references
- Demonstrates critical ability
- Provides good suggestions for future work