



UNIVERSITY OF  
ABERDEEN

University of Aberdeen  
School of Natural and Computing Sciences  
Department of Computing Science  
MSc in Artificial Intelligence  
2020 – 2021

**Assessment Item 1 of 3 Briefing Document – Group Work**

**Title:** CS5079 – Applied AI

**Note:** This assessment accounts for 33% of your total mark for the course.

**Learning Outcomes**

On successful completion of this component, a student will have demonstrated competence in using the OpenAI Gym environment, planning and creating a learning agent and conducting significant experimental investigations using machine learning strategies on different input formats.

**Information for Plagiarism**

The instructor will submit the source code and the report for plagiarism check (e.g., Turnitin). Please refer to the slides available on MyAberdeen for more information about avoiding plagiarism before starting the assessment. Please also read the following information provided by the university: <https://www.abdn.ac.uk/sls/online-resources/avoiding-plagiarism/>.

**Group Support & Contact**

This examination is group work. As a result, the students will have to form groups of **three** individuals, and report their names to Dr Bruno YUN ([bruno.yun@abdn.ac.uk](mailto:bruno.yun@abdn.ac.uk)) by 23:59 on the 2<sup>nd</sup> of December 2020 (UK TIME). After this date, the instructor will assign all students without a group to one, and group compositions will not be changed. Note that each student can only be part of a single group.

Students should send any queries concerning the assessment to the course coordinator.

**Report Guidance**

Each student group will have to produce a written report. The report must conform to the below structure and include the required content as outlined in each task. Each subtask has its mark specified. The necessary code (in Python notebooks) must be supplied, along with a written report containing the following elements:

- An introduction to the problem written using the students own words
- The motivation behind each considered approach.

- An evaluation of the approaches with a critical and reflective account of the processes undertaken. *The student should highlight any further measures taken as a result of the evaluation.*

The report should describe and justify each step that is needed to reproduce the results by using code-snippets, screenshots, and plots. When using screenshots or plots generated in Python, make sure they are readable. If the students used any open-source code, they must point out where it was obtained from (even if the sources are online tutorials or blogs) and detail any modifications they have made to it. The students should mention this both in the code and the report. Failure to do so will result in zero marks being awarded on related (sub)tasks.

### **Marking Scheme**

The instructor will take the following marking criteria into account:

- **Quality of the report**, including structure, clarity, good English, and brevity.
- **Reproducibility**. How easy is it for another MSc AI student to repeat the experiments based on the report and code provided?
- **Understanding**. Do the student show a deep understanding of the approaches used?
- **Quality of the experiments**, including design and the presentation of the results.
- **In-depth analysis of the results**, including critical evaluation and conclusions.
- **Quality of the source code**, including the documentation of the code and comments.

This examination will be marked on 100 marks by the instructor. When submitting the group work, the members of the group can agree to share the marks following a given distribution to reflect the outstanding performance of some of their partners. All students in the group are required to sign the distribution of marks which should not be defined arbitrarily. If this distribution is not specified, all the members of the group will be awarded the same mark.

### **Example**

The instructor awarded the group work of three students A, B, and C a mark of 70/100. The students agreed that A contributed substantially more to the project than B and C and submitted the following distribution: A (36%), B (32%), and C (32%).

As a result, student A will receive  $70 \times 3 \times 0.36 = 75.6$  points, whereas students B and C will receive  $70 \times 3 \times 0.32 = 67.2$  points. A student cannot have more than 100 points or less than 0 points.

### **Submission Instructions**

The students should submit a PDF version of the report along with the corresponding code via MyAberdeen by 23:59 on 7<sup>th</sup> of December 2020 (UK TIME). Any additional files can also be included in the submission (if they are referenced in the report). If you have more than two files to submit, please compress all your files into one “zip” file (other formats of compression files will not be accepted). The ZIP file should have the form “CS5079\_Assessment1”. Please try to make the submission files less than 10MB as issues may occur when uploading large files to MyAberdeen.

Questions about any aspects of this assessment should be addressed to the course coordinator Bruno Yun, [bruno.yun@abdn.ac.uk](mailto:bruno.yun@abdn.ac.uk).

### **Assessment Description**

This assessment focuses on two crucial problems that ML experts might face in real-life situations. The first one refers to the varying forms of available input information that an agent can observe; the second one refers to choosing (and improving) models for learning and selecting optimal actions.

This assessment will focus on the following Atari game:

- **(Seaquest)** The player plays a submarine, which can shoot enemies and rescue divers by bringing them above the water-level. The goal is to get all the divers up before the air level of the submarine vanishes.



*Figure 1: Screen of the Seaquest game.*

The game can output observations as pixels or RAM (128 bytes represented by an array of size 128 with values between 0 and 255). The students' goal is to study different approaches using only information from the screen, only RAM, or both.

To use the Atari environments, the students will need to install both “gym” and “atari\_py” using pip. The OpenAI Gym official website provides more details about the Atari environments: <https://gym.openai.com/envs/#atari>.

For the assigned game, each student group will have to complete each of the **three tasks** described below. Please use Python for all programming tasks. The students are encouraged to use python-based frameworks, such as **Tensorflow and Keras**.

## **Task 1: Reinforcement Learning from the Screen Frames (35 Marks) [max. three pages]**

### **Subtasks:**

- 1.1) Please describe the game, along with how you imported the environment, providing snippets of code and detailed descriptions. The descriptions should include the following elements (3 marks):
  - Observations
  - Action space
  - Reward
  - The environment's info dictionary
  - Episode
- 1.2) *Using your own words*, first explain the traditional Q-learning algorithm and why it “cannot” be used for the Seaquest game. Second, explain how deep Q-learning algorithms work and how they can be applied in this case for training the agent (3.5 marks).
- 1.3) Working directly with Atari frames can be computationally demanding. Explain the appropriate pre-processing techniques that you used and provide snippets of code (6.5 marks).
- 1.4) Implement an agent based on a deep convolutional network, using a parameter  $\epsilon$ , for making random moves, that decreases from 1 to 0.1 and experience replay. Please describe in detail how you deployed your agent and adjusted its parameters, going into detail on what each parameter does as well. You may use open source code and libraries as long as you acknowledge them (12 marks).
- 1.5) Train an agent on the game and evaluate its performance w.r.t. the total reward collected during an episode. Please note that a high number of episodes may be required for the agent to create “interesting” strategies. Present the training process, the experiments (including the experimental setting), and discuss your results. You should make use of figures, including a line plot where the x-axis is the training epochs and the y-axis is the average reward (per epochs) (10 marks).

\*\*\*\*\*

**Bonus – Optional:** If you decide to develop a “frame-skipping” technique without using any packages or pre-built libraries, there will be a bonus of up to **3 marks**. That is to say; the agent will only be training every  $k^{\text{th}}$  frame instead of every frame. This will allow the agent to play more games without increasing the runtime. Here, we recommend fixing  $k$  to 4.

Please note that the maximum overall mark for this assessment remains at 100/100; however, attempting the bonus exercise will increase your understanding of the algorithms, fasten the training process and enhance your chances of getting a higher mark overall.

\*\*\*\*\*

## **Task 2: Reinforcement Learning with RAM (35 Marks) [max. three pages]**

In this task, we will train and evaluate agents with RAM only as input.

### **Subtasks:**

- 2.1) For the Seaquest game, write a short description of the RAM and apply any appropriate processing techniques. Choose and plot a visualization of some RAM inputs to highlight the RAM cells that are important for the game (**6 marks**).
  - 2.2) Develop an agent equipped with a dense neural network to choose the best action. The number of layers, units, etc. can be chosen by yourself after experimenting with different values. You may develop this using Keras (**13 marks**).
  - 2.3) Train an agent and evaluate its performance w.r.t. the total reward collected during an episode. Please present the training process, the experiments, and discuss your results. You should make use of figures, including a line plot where the  $x$ -axis is the training epochs and the  $y$ -axis is the average reward. How can the model be further refined? (**10 marks**).
  - 2.4) Compare the performance of the RAM only agent with the agent obtained in task 1.5. What can you conjecture? Elaborate on your findings (**6 marks**).
- .....

### **Task 3: Reinforcement Learning by Mixing Screen and RAM (30 Marks) [max. three pages]**

#### **Subtasks:**

In this task, we want to investigate how mixing the RAM and information from the screen can be used to train an agent.

- 3.1) Create an agent based on a simple mixed network architecture where we concatenate the output of the last hidden layer of the convolutional with the RAM input. The number of layers, units, and the other parameters can be chosen by the students after experimenting with different values (**15 marks**)
- 3.2) Train your agent and evaluate its performance w.r.t. the total reward collected during an episode. Please present the training process, the experiments, and discuss your results. You should make use of figures, including a line plot where the  $x$ -axis is the training epochs and the  $y$ -axis is the average reward. (**7 marks**)
- 3.3) In a table, summarise the results obtained from agents developed in tasks 1.5, 2.4, and 3.2. What can you conclude? (**8 marks**)