General Marking Instructions:

- Please provide reasoning for your marking decisions in the form of full sentences.

- Be polite in giving feedback.

- If you do not understand something or are unsure what something does, please state it.

- You should be able to run the code they provided yourself, so you should be able to check if the result is correct.

- Continuation errors only count once. (Sometimes this is tricky, yes.)

- If in doubt state your uncertainty.

- If you really can not figure out what is happening and you can not grade the work send me an email f.riegler@herts.ac.uk and we see how to proceed. (I expect you put in some work before doing this.)

In this worksheet we will lay the foundation for Reinforcement Learning. We will building on the last worksheet, conduct multiple episodes and perform some basic evaluation.

Building our system parts

1. (2 marks) Build an 6x4 GridWorld containing an agent in a random field as well as 10 randomly distributed food pieces.
   Another random field should be a goal state. Just as last worksheet.
   The agent should have an energy meter, a simple int value will suffice, it should start at 4.

   > 1M for a world of the correct size with the correct number of food pieces in it and a goal.
   > 1M for the agent having a variable to keep track of energy and this being set to 4.
   > -2M if the agent and the world are not separate objects.

2. (3 marks) Just as last worksheet, the agent can choose between five action, but each now has an energy cost associated with it:

   - The agent can decide to attempt to move in any of the usual four directions, moving into a wall resets the agent to the field they were in before (bouncy as before), this incurs a cost of -1 energy.

   - The agent can decide to attempt to consume food, if successful removing it from the world and giving itself 5 energy.

   The agent can sense two things:

- When they stand on top of the goal
- When there is food adjacent or below them

Keep track of the agents energy after each step. Additionally track each taken action (a simple counter for all 5 actions is sufficient [a list or dictionary with each action might be an efficient solution]).

> 1M for an agent that can make these five decisions (but does not realise (affect change) itself) and for the sensing abilities
> 1M for each action incurring an reward (-1 or 5)
> 1M for keeping track of all actions activations

3. (3 marks) Let the agent walk through the world the same as last time but without the probability part. The agent moves until the goal is reached. If the agent senses food nearby, it will move to the food and if the agent is on top of food, it will consume it. Otherwise the agent will move randomly. The episode ends once the agent either reaches the goal, or when the agent runs out of energy (energy is 0).
Save the energy value at the end of the episode, as well as the utilised actions. Perform 10 runs on the same world.

> 1M for performing 10 episodes
> 1M for the episodes being in line with the logic above and saving the performance of the 10 runs
> 1M for the world not changing between runs

Evaluation

4. (3 marks) Evaluate the agents performance. What action was the best for the agent in your 10 runs? Take the total energy at the end of a run and divide this value by the total amount of taken actions. This is the average reward per action. To now evaluate each action class/type, multiply how often a specific action was taken with this value. Do this for all 10 runs, and print the average for each action.

> 1M for computing something out of the stored information
> 1M for computing one episode correctly
> 1M for averaging over all runs

11 total marks are available.

Have a wonderful day :)