Group Project – Log Book

This logbook will show on a day-to-day basis how I have contributed to the development of all aspects of my groups' COMP5530M Group Project, Exploring Curriculum Learning. Here including the continuous research undertaken by me since the beginning of the project, along with all commissions I have made to the code base that we developed throughout the project. Furthermore, it will include any notes made on meetings conducted between the project members alone, or together with our supervisor. This will be recorded below in the format of sub-headings for each day that I worked on the project, with a succinct overview of the focus of that day, followed by bullet points listing in the steps taken in the aforementioned area, and then finally a short explanation of the total time expended. It should be noted that the format of the listing of steps taken varies, between chat logs and notes that I made personally. Ultimately these will be followed by a short section accounting for the cumulative time spent over the entire period of project development.

Individual Day Logs

30th September 2020

Today I proposed some general areas of study for which I wanted to complete a project in.

- Looked at past projects for inspiration knowing I wanted to do a project in the field of AI, found that a project on reinforcement learning in the Starcraft 2 environment seemed very interesting.

Spent around 3 hours on this day researching potential ideas for which to do a project in. (Total for week 28/09-04/10).

7th October 2020

Today I proposed some initial ideas I had for the project in the Discord server another student made to help us find a team.

- Suggested doing either this environment which would only be possible because of 'StarCraft2LearningEnvironment' which was no longer maintained, found that we would have to find another environment or create our own for a project of this type.
- Also suggested a project in machine learning for procedural content generation, such as the creation of Mario levels, my idea was to use the dataset of ranked 'osu!' maps, a rhythm game, in order to produce software that could generate such maps from only a song input.
- Made a final suggestion of using Kaggle datasets on League of Legends game data to try and create a program that predicted the outcome of games in progress from past data.
- Fu suggest the use of board games, and Rayhan suggested snake, for our game environment for a reinforcement learning project.
- Had a meeting with the module leader, Rayhan ultimately decided that Matteo would be a good fit for our supervisor for this project and contacted him.
- I suggested more reinforcement learning environments such as the PyGame Learning Environment and Unity ML agent, Fu mentioned the hide and seek AI environment and the OpenAI Gym library which we ultimately ended up settling on as our main basis inspiration.

Spent around 6 hours on this day research specific potential environments and conferring with the other team potential team members to try and put together a team for the project.

11th October 2020

On this day I confirmed our project leader with Rayhan, who said that we had yet to receive a response in this regard. Spent 6 hours total working this week, (week 5/10-11/10).

19th October 2020

Received additional team member of interest, and attempted to confer with Rayhan about the confirmation of our supervisor.

20th October 2020

Created a prospective title and summary for our project to post in the teams channel, prior to a proper confirmation from Matteo.

- The initial project title that I came up with was 'Reinforcement Learning in games to explore fringe concepts' and I wrote the summary:
    - "The use of Reinforcement Learning algorithms to teach an agent to play simple games has been well explored, but there are some concepts that provide a challenge for current algorithms, and these concepts therefore invite further exploration. Games with 'simple' concepts in this definition are those that are fully observable, with singular units and small action spaces – therefore partially observable, multi-unit games with large action spaces are desirable targets for the expansion of reinforcement learning algorithms. An algorithm which is effective in such environments could be useful in simulating real life scenarios such as rescue operations carried out with multiple rescuers, in hazardous or initially unknown environments.
      Some existing potential toolkits to consider are 'OpenAI Gym' which is a collection of environments which make no assumptions about the structure of the agent, and 'Unity ML-Agents' which allows games and simulations created through unity to act as environments."

Spent around 3 hours refining our initial ideas and suggestions into a brief summary.

22th October 2020

Matteo replied and agreed to be our supervisor for the project, suggesting the implementation of a method that he had worked on previously, curriculum learning.

- Briefly went over the paper that Matteo sent Rayhan as a basis, 'Curriculum Learning with a Progression Function' or arXiv 2008.00511.
- Organised a meeting time for which to meet Matteo, the meeting was organised for the 29th of October.

Spent around 2 hour communicating and skimming through the paper provided, getting a more specific idea of what direction we wanted the project to go in.

Spent 5 hours total working in this week, (week 19/10-25/10).

29th October 2020

Today day we had our first meeting with our supervisor, Matteo, and discussed with him the general direction for which we were looking to take our project, before settling on the idea of applying the technique of curriculum learning to a more complex task.

- 13:00 – Started our meeting with the supervisor, Matteo, explaining our project ideas and what gave us inspiration to go in this direction.
    - He gave us the direction of implementing an existing reinforcement learning algorithm in an existing environment, then building upon this with a curriculum learning form of implementation.
    - Explaining how the difficulty index in this curriculum learning function would be between 0 and 1, and determine how much difficult the curriculum was to increase by and when, when the agent has learned enough to progress.

    - Suggested the implementation of a dynamic version of such a function and to properly read through the paper he sent.
    - Suggested use of q-learning, Matteo explained problem is exploration and the need to look into transfer learning.
- 14:00 – Sent over the multiple articles that we drew our final inspiration from, including the Vox article about multi-agent emergent tool use within an environment depicting a game of hide-and-seek with two hider and two seeker agents.
    - Matteo explained that he had past experience with OpenAI Gym but that it was not the best software to use for this purpose.
    - Suggested researching into Unity ML agent, where we could potentially have two agents, where one has a pre-determined set of actions at run-time, potentially in a first-person shooter style game.
    - PyGame was decided to be unfit for purpose, along with Starcraft (since no longer being maintained).
    - Looked into 'Half Field Offense' environment, along with a Dota 2 environment, and finally the OpenAI multi-agent hide-and-seek environment, which uses MuJoCo as a basis physics engine.
- Looked further into OpenAI hide and seek environment, found that a code-base was available to the public, and that it would be suitable for our project idea.
- Idea was to face in a curriculum sense, where each environmental object would be added in term to increase the difficulty of the environment, along with agent speed and environment size.
    - Researched into knowledge transfer and transfer learning for this purpose.
- Decided upon limiting OpenAI environment from four agents down to a single-agent environment, with a single opposing agent with fixed, but determined by difficulty index.
- First task set was to look over their code base and try to see how much could be re-used and implement a single agent into an environment inspired by this, with a simple single task for the initial implementation.
    - If they provided code that made it easy to use their environment with a single agent, then great, if not we would have to implement this.

Spent around 4 hours today (week 26/10-1/11) including hour long meeting doing above research.

10th November 2020

Looked into code-base, found that a MuJoCo license was required in order to run it, and then applied for this. Spent around one hour looking into the code. (Total for week 9/11-15/11).

25th November 2020

Received the MuJoCo license, which I applied for using the student procedure. Looked into code further now that it could be run.

- Was unable to run the code fully due many errors within my linux install, even after receiving the MuJoCo license, mainly that GTK windows would not render.
- Also many errors had to be fixed within the python install, specific to my machine such as 'GLEW initialisation error: missing opengl', as well as other drivers installed at a kernel level that needed to be found.
- Solved issue where glfw windows could not render by using 'LD_PRELOAD' command to locate manually for the python program where the OpenGL drivers were on my system.

This fixing of the environment on my machine took a significant amount of time, around 9 hours of research, since there was no specific basis from which to solve each individual issue.

27th November 2020

Was able to run the code properly, as is, now that most of my environmental issues had been fixed, added an additional member to the project.

- Was able to run the code at different levels of learned policy using the provided '.npz' saved policy files and limiting the environment to 2 agents, one hider and one seeker.
- Was unable to make the environment learn anything, but unsure why as for now, had to focus time elsewhere.
- Started to organize a meeting with all the members, including the new members for the first time, had to wait for their response in this regard.

Spent around 3 hours doing the above research into the code base and communication, and reading through the article properly on the OpenAI environment.

Spent 12 hours doing work this week, (week 23/11-29/11).

3rd December 2020

Finalised our meeting based on member's response.

- We decided upon a meeting for the first time people were free (due to deadlines), the 7th of December.
- Planned out a few things to discuss during the meeting: delegation of roles, creation of git repository, and creation of shared resources.

Spent around 1 hour planning for the upcoming meeting. (Total for week 30/11-6/12).

7th December 2020

On this day we had our first meeting without Matteo, and got properly started on the project code-base in a collaborative fashion, had to wait for other members to respond in this regard in order to have this meeting.

- Created a git repository, setting up the environment with all the required dependencies and created a readme file that explained how to set the environment up so that the others could follow these instructions instead of having to set it up from scratch themselves.
- Created multiple google documents, one for the collaborative creation of our scoping document, one for planning, and one for the final group project.
- Helped other team members in setting up the code from our code base which I initialised.
- Assigned some roles to team members, some members were to work solely on research, others (including myself) mainly on the code-base until closer to the deadline.
- Added all team members to gitlab repository, and to google document shared files.

Spent around 3 hours in meeting and setting up all the shared work. (Total for week 7/12-13/12).

10th January 2020

Started adding to the Scoping Document, and properly finalising our project plan.

- Created a slightly more specific version of the project summary from before, with citations.
- Drafted up a short paragraph on the aim of the project, 'exploring the emerging technique of curriculum learning', with specifics about the environment.
- Had to research to complete the above and find correct sources and definitions.

Spent around 2 hours doing the above. (Total for week 4/1-10/1).

22nd January 2020

Discussed with other members about meeting with Matteo (27th January) and what work they had complete over the Christmas break. Very little work had been completed.

- Discussed with other members about completing the scoping document, delegating some tasks to other members Luke, Rayhan, Collin and Fu.
- Added a lot of information to the Aims and Summary to specify more precisely and succinctly the goal of the project, since the module leader had trouble understanding this.
    - Re-wrote the summary, and added to aims.

Spent around 2 hours doing the above.

24th January

Re-wrote the majority of the scoping document that needed to be changed, adding a lot to all areas.

- Finalised summary section.
- Added lots of content and finalised aims section, three paragraphs of content in the end.
- Changed and made additions to objectives and deliverables section of document.
- Suggested to members a plan for the project, but did not finalise.

The above took around 3 hours.

I spent a total of 5 hours working in this week, (week 18/1-24/1).

25th January

Rayhan had looked into the code-base and found that it was possible to make our own environments within it, and created a sample video of the environment using the existing code.

- Looked more deeply into .npz files, was unable to find anywhere in the code where they were generated, since they are the saved models, was also unable to find where in the code-base the learning was carried out and therefore policies saved.
- Found .npz structure to be simply a zip file, but how to read them, rather than just running them, is still a mystery.
- Found other important files 'Jsonnet' files to be modified JSON containing the metadata for the model at that number of iterations, specific to npz file.
- Suggested use of .npz file at specific levels of knowledge as basis for our curriculum.
- Generated an example video for the upcoming presentation, using one of their saved policies on a slightly modified environment of our creation, where the number of agents was limited to one hider and one seeker.

Spent around 5 hours looking into the code-base more specifically.

26th January

Had a meeting and planned for the presentation on the following day, including delegation of sections and relevant research, before scripting out the meeting and producing a PowerPoint.

- Delegated tasks for upcoming meeting, I ended up choosing to complete 'plan to completion' section, which meant explaining what our current plans going forward were.
- Uploaded a new sample video to YouTube with many iterations which better displayed the current status of the project.
- Organised a meeting with Matteo on the upcoming Thursday.
- Created script for my section on the plan to completion, and PowerPoint slides from which to present.

Spent around 4 hours, doing the various tasks above, most of which was spent in writing my script.

27th January

Had our first presentation which went reasonably well, also received communication from Matteo.

- Did our 15 minute presentation.
- Matteo said to properly go through his paper prior to the upcoming meeting on the 28th.

- Went through the entire paper, making detailed notes on every section, and submitted them to our Discord channel for the other members to more quickly be able to cover before the meeting.
- Extracted what the most important information was and formulated a few questions to ask Matteo in the upcoming meeting.

Spent about 5 hours going through each section of the paper, making notes etc.

28th January

Had our meeting with Matteo, found out a team member (Luke) had contracted covid, delegated tasks. Questions asked:

- Would you recommend implementing the Framework as suggested in your paper, or another Curriculum Learning approach? Such as Prioritized Experience Replay or Heuristic Task Sequencing for Cumulative Return?
    - Answer: Implement the framework as it would help to get the experience of it in another other domain
- The 'Point Maze' experimental setting from the paper used a MuJoCo environment, like what our project is using, was the learning done through TensorFlow? And if so is there any way we could have access to this code to use as a reference for how to implement a Curriculum Learning approach for our environment?
    - Answer: Have to contact student from York but would give us as basis
- Do you recommend that we use a Fixed Progression function, or Adaptive? Or start with Fixed to ease implementation, then consider an Adaptive function if we manage to implement a fixed (since the adaptive would also require the implementation of a performance function).
    - Answer: Fixed to begin with.

Other notes from meeting:

- We need to figure out the action space by looking at the policies, we can write a program to write a policy.
    - Fix the behaviour of the hider.
- Try to get the agent to learn at least something.
    - What is the input of the learning agent, it needs to learn the features, either from visual or from variables.
- Decide what makes the task harder, increase environmental complexity, opposing agent difficulty.
- Increase action space over time, moving to begin with, grabbing object, locking objects.
- At the very least we want a reinforcement learning agent that is able to learn a task.
    - Hopefully we can be able to create a curriculum ourselves.
- Could create an image, which is the environment segmented, a circle where the agent is of a certain color, solve the computer vision part ourselves instead and use that as the input for the agent.
- Focus more on behaviour of agent
    - Next: states and actions of the user agents
- We will be using PPR first probably, or DDPG or something more simple for initial implementation
- Disentangle the environment from the learning code.
    - Once we can learn in the simplest environment then we can start adding things to the environment
- What should the reward look like:

    - Positive reward for going towards the hider initially, negative for moving away, doing nothing negative too?
    - Reward shaping, using external information such as Euclidean distance
- We need to avoid a closed loop generating positive feedback, as it won't learn from there.
    - A cycle should have a sum of zero - hint (doesn't always work)
- Adding a wall and moving directly towards wouldn't be the right thing to do but it needs a starting point.
    - All actions should have feedback.
    - Small world with no objects to begin with.
- Received reply from Andrea (Matteo's collaborator from University of York), providing us with the code-base they used for their paper.

The above took around 3 hours to complete in its entirety.

I spent around 16 hours working this week, (week 25/1-31/1).

6th February

Rayhan created the simple environment as described by Matteo, with a single agent and a goal.

- Allowed the merge request for the code Rayhan produced, introducing a new requirements file and readme for our new code-base.
- He had a brief look at the multi-agent-cooperation-learning repository, and at the code provided by Andrea.
- We decided to delegate the cooperation learning observation to me, and Andrea's code to himself, since I was unable to get Andrea's code to run on my machine.

Spent around 4 hours on this day trying to get Andrea's code to run, to no success due to driver errors, then the multi-agent-cooperation-learning code to run also which took an hour.

7th February

More deeply looked into code structure of our base environment, and also slightly into an alternative we found that uses a similar basis. Rayhan looked into the code Andrea supplied us. The following are chat logs and notes, unedited.

- [14:41] okay so i've figured out that npz files are stored numpy data which is stored through
    - numpy.savez
- and you can load it with
    - npzfile = np.load('./directory_of_npz/file.npz')
- look at which variables it contains with
    - npzfile.files
- and look at the value of that variable (usually arrays) with
    - npzfile['name_of_variable']
- [14:43] so I did
    - npzfile = np.load('./multi-agent-emergence-environments-master/examples/blueprint.npz')
    - npzfile.files
    - npzfile['policy/policy_out/action_pull/dense/bias:0']
- and got as output
    - array([0.00671348, -0.00671162], dtype=float32)
- [14:45] trying to figure out where they've implemented it to save the policy variables to an npz file rn, if i can do that we can just reverse engineer from that
- [14:46] if I can't i'll have a look at that code Andrea sent us

- [16:15] it doesn't look like they've used np.savez anywhere though, so it seems they've left that part out of the repository, unless a wrapper for it of some sort is used somewhere, there isnt a single reference to it 'savez' in any of the python files in the 'multi-agent-emerge...' repo
- [16:20] Try to use modified jsonnet with modified 'env_simple.py' that contains at least one object?
- Found that this code would be unusable, due to missing files as explained in above.

Spent around 5 hours doing the above code analysis.

I spent around 9 hours doing work this week, (week 1/2-7/2).

10th February

Did a large amount of Code Observation into the alternative basis that we found that uses the hide-and-seek environment as its basis.

- [12:33] For the cooperation learning environment, they did not include one of the files they wrote i model-contents-tensorflow-version I found that the spinningup implementation of PPO had three members of its dict: Key Value x Tensorflow placeholder for state input. pi Samples an action from the agent, conditioned on states in x. v Gives value estimate for states in x. So at first I figured i would morph the 'action samples' from the recorded data (i ran it for 10 epochs) into an object similar to 'pi' after running with that it told me that it could not find the value of 'x', figures... They must have rewritten the 'load_tf_policy' function such that it accepts the data in the format they recorded it in
- [12:35] which is: dict_keys(['action_movement', 'action_pull', 'agent_qpos_qvel', 'box_obs', 'mask_aa_obs', 'mask_ab_obs', 'mask_ab_obs_spoof', 'observation_self'])
- [12:36] So for using the cooperation environment, we either need to get their 'test_policy_ppo' file, or try to rewrite the 'load_tf_policy' in spinningups 'test_policy' to accept the format of data they used or it might be possible to morph their data into x, pi and v(edited)
- [12:40] Taken straight from their pdf: The observations in the MACL environment are a dictionary constituted of six key-value pairs:
  - ’observation_self’: a set o matrices describing (global) linear and angular position and velocity of each agent;
  - 'agent_qpos_qvel’: a set of matrices describing (global) linear and angular position and velocity of all other agents in the perspective of each agent;
  - ’box_obs’: a set of matrices describing all (global) generalized coordinates of the boxes present in the environment in the perspective of each agent;
  - ’mask_aa_obs’: binary mask describing for each agent which other agents it sees in its vision cone;
  - ’mask_ab_obs’: binary mask describing for each agent which boxes it sees in its vision cone;
  - ’mask_ab_obs_spoof’: an all ones vector that can be used instead of ’mask_ab_obs’ when the agents receive full information from the environment (they can "see" the boxes even though it may not be in their vision cones).
- The actions taken by the agents are also defined by a dictionary. In this case, there are two key-value pairs:
  - ’action_movement’: a set of vectors describing the speed level for each agent in every movement component (X motion, Y motion and rotation around the Z axis);

    - ’action_pull’: a binary mask describing for each agent whether it is pulling or not pulling.
- All low level construction of these dictionaries is done automatically by the mae_envs wrappers.
- [12:40] which to me indicates that pi = {'action_movement', 'action_pull}
- [12:41] but to me it seems that all of the other values would make up either 'x' or 'v' for a single state
- [12:42] but im not sure which, and in either case, we end up without the third key
- [12:44] Im gonna try morphing all of the remaining ones into x, and see if it runs
- [12:44] i've gotten it to open the visualisation window so far, but it just crashes instantly
- 13:00 it ran for longer but that doesnt seem to be the ticket, looking at the ppo file they wrote, they have used 'pi' and 'v' internally
- [13:01] so it might be possible to re-engineer their ‘test_policy_ppo’, or the 'load_tf_policy' if i dig a bit deeper
- [13:01] Did you make any progress with Andrea's code? @rayhan
- [13:14] huh, figured pi would be ‘out_act_dict’, and v would be ‘out_state_dict’ i got:
- [13:15] so pi is definitely ‘out_act_dict’
- [13:15] but ‘out_state_dict’ seems to be empty for whatever reason
- [13:27] ah, the other stuff comes under 'logger_inputs' and the ‘out_state_dict’ doesn't ever seem to be updated
- [13:47] Realised they have the policy set to save every 10 epochs, but have been running most test's on a file run to 10 epochs, changing to 1 epoch and running for multiple to check whether some data is missing due to their only having been one save of policy and value function.
- [13:58] On https://www.mdeditor.tw/pl/2c2Z The person uses
- For 'Sample actions for given observations'

```
return session.run([self.action, self.value, self.neg_log_pi],
        feed_dict={self.obs: obs})
```

- 'Get value function for given observation':

```
return session.run(self.value,
        feed_dict={self.obs: obs})
```

- The cooperation environment has its 'session.run()' missing, the one they based it off is

```
action_op = model['pi']
...
get_action = lambda x : sess.run(action_op, feed_dict={model['x']: x[None,:]})[0]
```

- But the cooperation environment does not have 'pi', 'x' or 'v' as part of its model (explicitly at logger output)
- Trying to reverse engineer... Pi and V are being 'trained' in the written 'PPO.py' at lines, 409 and 426
- Furthermore in the file MA_policy, they added a function 'sess_run' which if i can run instead of sess.run, may be key in rewriting the function that they left out of the repo within the file 'test_policy_ppo' that was referenced for imports.
- [18:13] NVM that’s an internal function, I've been experimenting with different inputs for 'action_op' and ‘feed_dict’, currently have

```
'action_op' = model_pi
```

- Where model_pi is a list of what could be defined as pi

```
Model_pi = [model['action_movement'], model['action_pull']]
```

- And feed_dict is
  Feed_dict={model_x: x[None, :]})
- The current format of 'model_x' is:
  model_x = {'agent_qpos_qvel': model['agent_qpos_qvel'], 'box_obs': model['box_obs'], 'mask_aa_obs': model['mask_aa_obs'], 'mask_ab_obs': model['mask_ab_obs'], 'mask_ab_obs_spoof': model['mask_ab_obs_spoof'], 'observation_self': model['observation_self']}
- ON https://www.mdeditor.tw/pl/2c2Z again their example for feed_dict was:

```
feed_dict = {self.sampled_obs: samples['obs'],
    self.sampled_action: samples['actions'],
    self.sampled_return: samples['values'] + samples['advantages'],
    self.sampled_normalized_advantage: Trainer._normalize(samples['advantages']),
    self.sampled_value: samples['values'],
    self.sampled_neg_log_pi: samples['neg_log_pis'],
    self.learning_rate: learning_rate,
    self.clip_range: clip_range}

  evals = [self.policy_reward,
           self.vf_loss,
          self.entropy_bonus,
         self.approx_kl_divergence,
         self.clip_fraction,
         self.train_op]
```

And evals = action_op

- So as it turns out, what I was putting as x, is actually v, and placeholders need to be introduced as x... in
  Feed_dict={model['x']: x[None, :]})
- Model['x'] is a tensor placeholder and x[None, :] is the data being fed into it which can't be in tensor format:
  'feed with key ' + str(feed) + '.')
- TypeError: The value of a feed cannot be a tf.Tensor object. Acceptable feed values include Python scalars, strings, lists, numpy ndarrays, or TensorHandles. For reference, the tensor object was Tensor("policy_0/observation_self:0", shape=(?, ?, 8), dtype=float32) which was passed to the feed with key Tensor("Placeholder_3:0", dtype=float32).

Around 8 hours spent investigating this code-base as a potential basis, eventual deciding that it wasn't possible due to missing parts of the code and moving on to use Andrea's code. Which the observation for was delegated to Rayhan.

14th February

Got our starting point for the base of the reinforcement learning aspect of the code.

- Rayhan managed to get some of Andrea's code working also in this time, which came to the base of inspiration of our project from here.
- Most members not contributing currently, Luke still recovering from Covid.
- Ideas of Andrea's code implemented using PPO from OpenAI Static Baselines.

Spent around 1 hour on this day communicating and organising what to do next.

I spent around 9 hours doing work this week, (week 8/2-14/2).

19th February

Suggested to Rayhan to commit his incomplete code so that I could look at it, and suggested meetings for delegation of tasks.
Discord chat log (timestamps mostly unavailable):

- Have you made any more progress since then?
- it would probably be best for you to commit the code and assuming everything's somewhat organised someone else may be able to pick up from where you left off if you're busy doing other things
- think its best if we start to have a more collaborative workflow lmao
- We need to plan stuff out so, maybe we should have a meeting with everyone here soon? before the next time we meet Matteo at the very least
- then once everything is planned out somewhat we can have everyone involved and start making a lot more progress

Spent around 1 hour communicating and planning somewhat. (Week 15/2-21/2).

24th February

Rayhan replied, having trained the agent with our first base, the cylinder environment. Was unable to respond here due to other deadlines. He asked for a merge request. (Week 22/2-28/2).

1st March

Handled the merge request, ensuring his code ran as expected, got it all working on my machine. Spent around 2 hours doing this, have deadlines on 1st, 2nd, 5th, 10th and 11th so unable to work on this for a while. (Total for week 1/3-7/3).

11th March

Attempted to organise meeting with Matteo to follow up from where we are.

- Meeting not yet organised.

I spent no hours doing work this week, (week 8/3-14/3).

20th March

Received communication from Matteo, Rayhan found some promising results after comparing base reinforcement learning, along with a rudimentary implementation of fixed progression curriculum learning. Matteo said:

- "I think this is a great time to investigate what effect different progressions have. You already have an hypothesis as to why the more fine-grained progression isn't working as well, we should test that hypothesis. Maybe try with a logarithmic or exponential growth? Or it could be a good time to try to implement that adaptive progression function from the paper. I'm sure Andrea would be happy to answer your questions if you need help with that."
- I'll be busy for the next week or two, we'll have to work on the showcase stuff too in the upcoming week. In the meantime if anyone wants to have a go at implementing any of those progressions, please feel free to.

Then added some small headings to report. Around 1 hours work today.

I spent around 1 hour doing work this week, (week 15/3-21/3).

22nd March

Upcoming showcase:

- Matteo said regarding the showcase:

"The demo will be nice if you can show different "before and after" or accelerated videos of learning with and without the curriculum."

- We should get started on the presentation. I'll probably record those videos on Wednesday and we can embed them into the presentation.
- Started trying to implement to implement fixed exponential progression but ran into many errors getting stable-baselines code to run so could not make any.

Spent around 2 hours total in implementing and communication.

23rd March

Attempted to get Rayhan's code running more, communicated with the team here is that:
Some notes:

- Managed to get Rayhan's code to run, by changing the floor_size parameter from a tuple to an index because

  cell_size = floor_size / grid_size
- in the mae_envs/modules/util.py failed to run otherwise

Need to ask Rayhan about this

- Could implement comand line feature to run code in specific format rather than commenting sections in and out, this could implement choice of progression function

Discord chat log (timestamps mostly unavailable):

- Hey, apologies about that, not made progress on implementing the progression function yet, but i understand where we are at now
- I've been working on understanding the code you've implemented up until this point and handled the merge request accordingly
- I was unable to run the static baselines code until earlier due to many issues which I have now solved, and am now to grips with what you have implemented
- so I should be able to implement the exponential progression function tomorow if all goes well
- one small issue i've been having though when i run the code, as is, i get this error

```
til.py", line 41, in rejection_placement
    cell_size = floor_size / grid_size
TypeError: unsupported operand type(s) for /: 'tuple' and 'int'
```

  - which in order to solve, i had to change the 'floor' size from a tuple to an integer
    - this makes the code run, i think as expected?

```
116     # load_and_render((8,16), 120,
117     #                 SAVE_DIR+'example/cl/3_prog/step_2.zip',
118     #                 20, 100)
119
120     load_and_render(8, 120,
121                     SAVE_DIR+'example/cl/180_prog/step_179.zip',
122                     20, 100)
```

- but im not sure why it will run with a tuple on your machine, but only with a single value on mine
- i have been able to run all the saved policies you committed with this change, but they might not be working as expected due to this disparity
  - the training works as expected also, is the code for multiple environments (one on each core) completely not working?
- also if you could point me in the direction of any documentation you found useful for implementing the progression function as you did that would be great

    - would definitely speed up my implementation of the exponential progression function tomorrow
    - just to ensure its all working correctly, is this what is output to console when you run a saved policy and it terminates?

```
Creating window glfw
num_done: 20 / 20 with mean 20.65
```

- either way, sorry for not having made as much progress as you might have hoped, I was very burnt out but i'm genuinely back to working on this properly now
- hopefully in time to make some progress on this for the meeting
- we should have a meeting to discuss who's gonna talk about what and plan those sections out a little like we did for the previous presentation

Spent around 5 hours on the above, including communication and fixing code (no references available).

24th March

Managed to get Rayhan's code running in full and started preparing for upcoming showcase.
Rayhan replied and I managed to solve the issue:

- I needed to re-setup my python environment, removing all redundant code, as some of the functions were being run from 'multi-agent-cooperation-learning' rather than 'multi-agent-emergence-environments'.

Spent around 1 hour accomplishing this.

25th March

Carried on from previous day, lots of planning for presentation for everyone else to complete their own sections, and set up document for everyone to work from.
Plan for right now:

- Go through presentation script that we wrote for the previous presentation and cross out sections that are no longer relevant
    - suggest things to replace those sections
- Stuff I added:
    - Part of this (old intro and auto curricula section) can be used for 'Summary of Technology Used' along with an explanation of our deviation from it and what we've moved towards instead:
        - Since we've scaled back the environment, the hide and seek concept is somewhat less relevant but we can still give a brief explanation. Along with why we've moved away from this as a concept, and towards our current concept. (I need clarity on what exactly this is)
    - 'Demo' can be covered by Rayhan as he is recording the videos for it...
    - Part of this (env intro) can be used in 'Something on impact made in terms of addressing the problem' but since we've deviated from this original plan it needs to be changed accordingly:
        - Talk about the use of PPO baselines and Andreas code
        - The progress we've made since then was largely exploratory, having explored two avenues of progression due to the limitations of the original OpenAI environment we explored:
        - 'multi-agent-cooperation-learning' for the potential of its application (by me)

        - Did not go down this avenue due to missing function that was beyond the scope of the project to re-implement
        - and what we ultimately chose as our base which is the 'multi-agent-emergence-environments' repository with Andreas code? PPO Baselines (Rayhan)
        - Rayhan should cover this with the things he implemented
    - Other things to cover (implementation):
        - Changes in what we currently have to do for plan to completion:
        - Scrap sections about the measurement of time to convergence from the original source code
            - Replace with talking about implementation of different progression functions in the direction of, Matteo's email:
                - "I think this is a great time to investigate what effect different progressions have. You already have an hypothesis as to why the more fine-grained progression isn't working as well, we should test that hypothesis. Maybe try with a logarithmic or exponential growth? Or it could be a good time to try to implement that adaptive progression function from the paper. I'm sure Andrea would be happy to answer your questions if you need help with that."

- Think of things for the other parts of the above brief to cover and in which ways
- Draft up prospective bullet points for each section for each person to discuss
    - Could go through the Report for help in finding such sections
        - Communication from Matteo could also be useful in this
    - Draft up prospective delegation of section
- Finally, drop a message in discord suggesting an appropriate meeting time for tomorrow.

Discord chat log (timestamps mostly unavailable):

- Started planning for presentation properly, here is chat log
- Okay guys, i've created a separate document for us to plan this presentation on, and given you all access
- here is the link: <<link>>
- i've lightly gone through the all stuff we wrote for the previous section, tried to figure out which parts we should cut, and what we might be able to keep
- All of the stuff from last time is in green,
- The brief for the new presentation is at the top, and i've separated it into sections based on what he has asked us to cover, with those sections in bold
- with my explanation of my understanding of what needs to changed and omitted (from our last presentation) and what else needs to be covered above each section, each on a new page
- obviously we need to meet later today to discuss the delegation of each section, I (and some others I assume) need clarification on a few things
- then hopefully we should be able to come up with a few more specific bullet points for each of us to write a script regarding (like last time) together during this meeting
- and write those scripts in time for the presentation at 3pm friday
- what time is good for a meeting later today? im thinking 4-6pm but it depends on when everyone is available

- [5:32] Im gonna head off and try to sleep now to wake for around these times to attend the meeting that I hope we can arrange, but hopefully the document i've created and things I've laid out should give us a good starting point if nothing else

Spent around 4 hours doing the above. Then later had a meeting at 6pm where we discussed the presentation delegation, slides and content. Spent around 2 hours doing so.

26th March

Wrote my script for my part of the presentation, uploaded YouTube videos for demo.

- Started and finished my script.
- Uploaded videos of environment as recorded by Rayhan.
- Finished adding slides for my section of presentation.
- Did presentation, discussed how it went afterwards (not great) and emailed Matteo for help.

Spent around 3 hours doing preparation, and 1 hour doing presentation and discussing afterwards.

I spent around 18 hours doing work this week, (week 22/3-28/3).

1st April

Received feedback from Matteo about presentation, and advice regarding our plan to completion from here.

- We did not do a good job explaining the core concept of our project.
- We confused the audience with conflicting explanations due to being flustered by what felt like rude questions.
- Matteo wants our first draft of the report due for a week before the deadline, so in about 17 days from now.
- Rayhan suggests we should try to get one a week before then.
- Rayhan implemented a moving agent, rather than just the simple cylinder as a target.

Spent around 1 hour communicating today, rather burnt out due to how presentation went.

I spent around an hour doing work this week, (week 29/3-4/4).

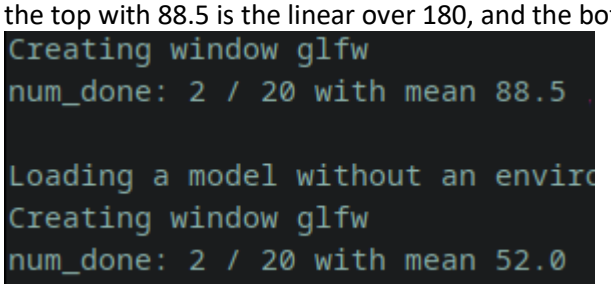
9th April

Communicated with team that I was away from a working environment due to family conditions, also Minerva was having issues in this time. Around 1hr of communication.

10th April

Merged Rayhan's changes from the 1st into the main branch and implemented logarithmic progression, using his linear progression as a basis for implementation.

- [09:00] – Merged Rayhan's changes to main
- [10:54] – I have implemented the exponential/logarithmic progression function now, it seems like it may be promising especially over 180 cl steps, although it doesn't seem to improve the success rate compared with linear over 180
- the top with 88.5 is the linear over 180, and the bottom with 52 is the exponential over 180



- [11:26] oh, i had implemented it to have the target move at max speed initially rather than minimum, I have switched it around now just waiting on some results
- [11:38] in my head i was like ye ye inverse and it should scale
    - then i get numbers like this

```
step: 159 speed: 4.3579013294484664e-48
step: 160 speed: 7.84422239300724e-46
step: 161 speed: 1.4119600307413031e-43
step: 162 speed: 2.5415280553343453e-41
step: 163 speed: 4.5747504996018223e-39
step: 164 speed: 8.23455089928328e-37
step: 165 speed: 1.4822191618709903e-34
step: 166 speed: 2.667994491367783e-32
step: 167 speed: 4.802390084462009e-30
step: 168 speed: 8.644302152031616e-28
step: 169 speed: 1.5559743873656909e-25
step: 170 speed: 2.8007538972582435e-23
```

- speeds very unreasonable
- so i thought
- maybe instead of (1/no_steps^(no_steps-current_step))
- which gave that
- i could do
- (1/no_steps^(1/current_step))
- and that gave me genuinely reasonable values

```
step: 1 speed: 0.005555555555555556
step: 2 speed: 0.07453559924999299
step: 3 speed: 0.1771097615304352
step: 4 speed: 0.2730120862709067
step: 5 speed: 0.3539529195983115
step: 6 speed: 0.42084410597088706
step: 7 speed: 0.476231604647823 27
step: 8 speed: 0.5225055849183879
step: 9 speed: 0.561583276146998
step: 10 speed: 0.5949394251504194
step: 11 speed: 0.6236992571771699
step: 12 speed: 0.6487249848517376
step: 13 speed: 0.6706833108959689
step: 14 speed: 0.690095359097439
step: 15 speed: 0.7073730335899061
step: 16 speed: 0.7228454778985532
step: 17 speed: 0.7367784444861712
step: 18 speed: 0.7493886015592965
step: 19 speed: 0.7608542083856764
step: 20 speed: 0.7713231651846192
step: 21 speed: 0.7809191469245005
step: 22 speed: 0.7897463245733848
step: 23 speed: 0.7978930343797629
step: 24 speed: 0.8054346558546742
step: 25 speed: 0.8124358887579103
step: 26 speed: 0.8189525693811388
step: 27 speed: 0.8250331305443869
step: 28 speed: 0.8307197837402447
```

[11:39]

- only problem is it never actually reaches 1
- which makes sense but
- i need it too
- so im at a crossroads lmao

```
step: 159 speed: 0.9678674760543318
step: 160 speed: 0.9680650628095526
step: 161 speed: 0.9682602346590283
step: 162 speed: 0.9684530356024336
step: 163 speed: 0.9686435085770893
step: 164 speed: 0.9688316954898296
step: 165 speed: 0.9690176372477297
step: 166 speed: 0.9692013737877393
step: 167 speed: 0.9693829441052704
step: 168 speed: 0.969562386281778
step: 169 speed: 0.9697397375113781
step: 170 speed: 0.969915034126542
step: 171 speed: 0.9700883116229003
step: 172 speed: 0.9702596046831975
step: 173 speed: 0.970428947200427
step: 174 speed: 0.9705963723001821
step: 175 speed: 0.9707619123622501
step: 176 speed: 0.970925599041483
step: 177 speed: 0.971087463287971
step: 178 speed: 0.9712475353665458
step: 179 speed: 0.9714058448756397
step: 180 speed: 0.9715624207655261
```

- [11:39] it just gets closer and closer to 1, but i need it to reach 1 at step 180 somehow
- [11:40] trying to wrap my head around it
- [11:53] okay so, using a purely exponential method
    - set_speed(1 / steps**(steps-prog_step)
- gave me speeds for the target (agent/cylinder) that were useless (in scale) until the final cl step, so i tried implementing it using
    - set_speed(1 / steps**(1/prog_step)
- which actually gave reasonable values throughout except that it never reached full speed (1)
- so i eventually implemented it using both, ensuring the final cl step was of the maximum difficulty
- [11:53] an example of the result of this with 20 cl steps:
- [11:53] this isn't exactly exponential i know, it achieves what we're aiming for i believe

- [11:54] so im gonna train this over the full 120,000 time steps and 180 cl steps and see if the results are promising

```
step: 1 speed: 0.05807931748207713
step: 2 speed: 0.06746414238367816
step: 3 speed: 0.07836542688315398
step: 4 speed: 0.091028210151304
step: 5 speed: 0.10573712634405641
step: 6 speed: 0.12282280261157905
step: 7 speed: 0.14266929093832778
step: 8 speed: 0.16572270086699933
step: 9 speed: 0.19250122715283496
step: 10 speed: 0.22360679774997896
step: 11 speed: 0.2597386039534327
step: 12 speed: 0.30170881682725814
step: 13 speed: 0.35046084319304355
step: 14 speed: 0.40709053153690444
step: 15 speed: 0.4728708045015879
step: 16 speed: 0.5492802716530589
step: 17 speed: 0.6380364656795914
step: 18 speed: 0.7411344491069477
step: 19 speed: 0.8608916593317347
step: 20 speed: 1.0
```

- [12:05] so, it never reached one
- cause it did from
    - 1/no_steps
- all the way to no_steps/no_steps
- then i realise
    - if i did 0/no_steps
- it would = 1
- so using 20 as an example
- [12:06] i finally got it how i want it →

- 12:11 here are some rudimentary results now after training for 1200 timesteps
- [12:12] top is linear, bottom is exponential

```
Creating window glfw
num_done: 3 / 20 with mean 32.333333333333336

hider/cl/exponential/test_180/step_last_step.z
Loading a model without an environment, this m
Creating window glfw
num_done: 4 / 20 with mean 42.5
```

- [12:12] slight increase in success rate
- [12:12] i will test over 120,000 timesteps now

Also implemented:

- Save_model function to avoid code duplication
- Trained model for 180cl steps and 120,000 timesteps and uploaded save model for testing by other team members.

Spent around 4 hours doing the above work and 1 hour running the code.

11th April

After communication with Rayhan, found the reason for not as large an increase in performance as expected, he implemented code to fix this issue, I thought of looking into console output as a metric for our report, also thought of another reason why the tests may not be accurate.

- I figured that we could use the console output to maybe plot some graphs and wrote some code over about 3 hours to create csv files from this console output.
- Rayhan wrote code to limit the speed of the hider agent as he figured the reason the learning agent did not improve as expected was due to the overall difficulty of the environment not being complex enough to make the extra steps worth it.

I spent 1 hour communicating until midnight.

I spent around 6 hours doing work this week, (week 5/4-11/4).

12th April

Continued from previous day, worked on ideas that I had communicated about then.

- He implemented a change in speed cap for the hider, testing for the correct value, I merged these changes into the main branch.

- [00:19] okay, i've essentially merged the changes into the master branch and pushed it
- I did it commit by commit as you did, testing it, and included in the commits that it was your work
- if you could pull the master branch when you start work on it again, rather than using the branch your on, that would be great
- a small test after implementing your changes does show some improvement

```
num_done: 12 / 20 with mean 56.333333333333336

hider/cl/exponential/test_180_120000/step_last_step.zip
Loading a model without an environment, this model cannot be trained until it has a valid environment.
Creating window glfw
num_done: 13 / 20 with mean 40.15384615384615
```

- I figured that maybe the tests were not as expected due to the random floor size used in testing, if the exponential algorithm were tested – by chance – on larger floor sizes, its performance would decrease.
  - [00:44] Okay so i've moved over the function from the base class and added the imports as necessary, i also added a print statement fro the floor size which has confirmed my suspicions see:

```
Use tf.cast instead.
22.61138290134548
Creating window glfw
17.911437453202538
20.884351902008046
17.5551713416255
16.74140184779071
num_done: 5 / 5 with mean 32.6

hider/cl/exponential/test_180_120
20.556607486368215
Loading a model without an enviro
23.03294943530224
Creating window glfw
22.248490156261887
20.503118020587888
22.94145986666982
21.565780157809197
num_done: 2 / 5 with mean 49.5
```

  - the second call of load_and_train on the exponential function underperformed but, the floor sizes on this call were on average much larger than in the first call
- So I implemented the ability to seed the floor size generator, rather than using np.random.uniform so that then we could eliminate that as a potential issue and cause of variance
  - [01:00] I overrode the _get_sim(self, seed): function from the 'Base' class which 'SimpleEnv' inherits from, since this is where np.random is called. The idea was to generate the seed for np.random in the main function once, and use this multiple times
    - This worked but I soon realised that every episode had the same floor size, due to having the same seed. The solution was to generate a list of seeds, equal in size to the number of episodes, with each seed being a random

integer between 0 and (2^32) - 1 which is the range of possible seeds that np.random.seed() can take as a seed. This list was then fed into MakeSimpleEnv in order to generate as many environments of different floor sizes as there where episodes. These environments were generated in the load_and_render function, and then a list of them was fed into the visualize agent function, which was modified to, in its for loop for episodes, use the appropriate environment per episode. This worked as intended and two separate models were ran sucessfully over identical environments as shown below. The idea behind this was to eliminate any variance that the randomness could introduce that would impact testing results over especially over smaller sample sizes.

    - [02:12] for each episode, the same seed is used in generating the floor size
        - which can be seen in the print statements
        - this seems to have worked as intended, and seems to have made the improvement I thought it would (ensuring parity for comparison between algorithms)
        - the only issue with the implementation as is, is that it generates a glfw per episode
        - which means that you have to speed up the simulation speed manually every episode
        - which obviously isn't great for running many simulations
        - but there is probably a way to set the simulation speed to a higher number (than real time) by default which would mitigate this issue
    - [4:10] Current implementation of seeded run creates a window for each environment, which is not ideal, would be better to create a single window for each load_and_render call, rather than each floor_size
- [6:21] Starting to work on testing the models as Rayhan Suggested:
    - I ran the code for many hours and saved the models.

Spent around 6 hours working, as can be seen from timestamps above, also discussion for around 1 hour prior about seed function and about how Rayhan's additions worked.

14th April

Received communication from Andrea about more proper use of his code, and about 'continous curriculum', set up plan to pair program with Luke to implement an adaptive progression function into our code base. Spent around 1 hour planning this.

16-17th April

Spent from 11pm till 2:30am going through the report as is, sectioning it off and adding sections, and bullet points to write about. Then I communicated with everyone:

- [02:58] you're definitely right on that the report should be our primary focus
- [02:58] especially until the 19th (when we get this draft in)(edited)
- [03:00] I've gone through the report, and changed up the structure, using some of the example projects as a guideline for the top level headings and moving some sections into different headings
- [03:01] I've added lots of sub-headings, numbered all of them, and added bullet points under many with sources of information for which to write about that topic
- [03:02] I've added in pretty much every topic I could think of from our project

- [03:02] Looking at the examples I noticed they explain topics down to the very very basics and so I added a heading for machine learning in general in the background research - just to give us something to buff out the report with
- [03:03] I've also filled out the appendix to a degree
- [03:05] @LukeMcMahon made some progress today, adding in a few paragraphs in the intro and a few suggestions which is good
- [03:06] sucks that noone else has had a chance to add anything yet
- [03:06] we have until the 19th, and its the 17th now @everyone
- [03:10] Me and Luke are going to be doing pair programming later today to try and implement the adaptive progression function

I delegated tasks for every member of the group to complete for the report, since no one else had taken control of the group and the deadline was looming:

- [03:32] To try and give everyone direction, I'm just gonna assign some things to people based on my limited knowledge of your knowledge of the sections:
    - @rayhan you will have to do 3.3.1 for the draft as you implemented the wrappers, along with 3.2.2 as you implemented the Cylinder and Hider environment, and maybe 2.4.1 since i remember you doing some stuff with auto-curricula in the first presentation that you could use
    - @Co1in You should do 2.1 which is a new section I added on Machine Learning, You should probably do 1.1 the hypothesis as you did about that in the recent presentation so you might be able to re-use some of that... once you've done that you could do 2.3 as well but I haven't but any sub-headings in that one but just research transfer learning i guess
    - @LukeMcMahon Good job doing the start of the intro, tho i think the content could probably be moved to a different section but dw about that for now, I would focus on covering 3.1 Dependencies if i were you since you might be able to re-use some of the content from our last presentation, also 2.5.3 if you're familiar with Andreas code tho I didn't name it, along with 2.4.2.2 since i think you have been reading some papers on that?
    - @Fu You should do 2.2 Reinforcement Learning since you covered that in the presentations and could probably reuse parts of that although theres a few things you will need to research for that and also 2.4.3 once you get done with 2.2
    - I will write the section about implementation of progression functions which is 3.3.2, I could do the background research for this too which 2.4.2 except maybe adaptive which Luke could do if he knows more I'll also set myself 2.5.1 (since im familiar) and 2.5.2 (since i was the one looking into this option)
- Don't forget to add your sources to the bibliography as you go, also u can choose a colour to write in to make it obvious who's done what like me n Luke have done
- [03:36]The stuff that hasn't been assigned:
    - The rest of the intro can be mostly taken from our scoping document probably so isn't worth spending time on now, and any left-over sections can just be completed by whoever gets the time or feels that they are suited for it
    - Chapter 4 and 5 are gonna be left mostly blank for now, though I will add in a few visualisations of the output of training in chapter 5 using the code that I wrote the other day
- Hope @everyone can cover those sections by the 19th so we can get this draft in to Matteo, we can work on the rest of the sections and getting the project finished while he's going over that
- If me and Luke manage to implement the adaptive progression tomorrow, I'd be happy with just an alternative environment factor for increasing difficulty on top of that.

Spent around 5 hours working on the report this night.

17-18th April

Began pair programming with Luke to fully understand Andrea's code-base, how it links in with ours, and the implementation of an adaptive progression function.

- First we went through Andrea's code, finding out that he used his 'EpisodicWrapper' to implement adaptive progression which required:
    1. Define performance (performance_functions.py)
    2. Mapping Function (mapping_functions.py)
    3. plug in an instance of progression (example in train_with_progression in train.py)
- First implemented an 'apply_complexity_factor' function which changed the way our code used the 'difficulty index' generated.
- We then implemented a reward performance function, simply one that took the distance of the agent from the target as its reward to figure out performance.
- We used the rudimentary 'mapping function' Rayhan had already implemented.
- Then we wrote a progression function that used this reward function, updating every cl_step throughout the learning steps. This rudimentary function had many bugs but essentially achieved what we wanted it to.
- Then I trained this function using the parameters from before, 180,000 timesteps with 180 curriculum learning steps.

Luke and I were pair programming to implement this for 11 hours, with a 1 hour break.

I spent around 22 hours doing work this week, (week 12/4-18/4).

19th April

Communicated with the guys for about an hour about my inability to work this day due to mental health and family issues. In this time Fu had moved our project report over to LaTeX (Overleaf).

20th April

After massive pair programming session, I ran the code above, it finished at 4pm on the 18th and I slept most of the time until the early morning on the 20th. I had to learn LaTeX syntax, I also wrote a section on how we implemented our progression functions.

- Learnt basic LaTeX syntax.
- Wrote section on implementation of progression functions as me and Luke implemented.
- Wrote Summary for report.
- Wrote Acknowledgements.
- Removed multiple sections of the report that were no longer relevant.
- Added some references.
- Added draft Aims, Objectives and Deliverables section.
- Added in Appendix A and B.

This work was carried out between 2am and 8:30am so around 6 hours 30 of work.

21st April

Slept most of the day, really bad mental health at the moment, worked on re-working report sections after feedback from group members and having slept on it. Luke had begun to overhaul the code-base after a meeting with Andrea, in order to get it working with his EpisodicWrapper.

- 22:00 - re-worked the Aims section, moving parts of it to other parts of the report and restricting its contents to solely explain the main, and side-aim of the project
- 23:11 - re-worked the first paragraph of the 'Reinforcement Learning' section to include the content moved from 'Aims' while still incorporating what Fu wrote

- 23:45 - re-worked the 'Experimental Setting - OpenAI Multi-Agent Emergence Environment' to include the content moved from 'Aims' while still incorporating what Luke wrote previously

Worked for 2 hours this day, and then into the next day.

22nd April

Continued working on report from previous day, feedback had been given by team members about how since the code was overhauled my implementation details about the progression functions was no longer accurate and needed to be re-done.

- 00:01 - slightly re-worked 'Our Environment' to include part of content moved from 'Aims' while still maintaining most of the original authors content (Luke or Rayhan)
- 00:19 - Implemented changes to 'Aims' section as highlighted by Rayhan
- 01:10 - Rewritten 'Progression Functions' explanation to fit background research section, still need to rewrite section on fixed and adaptive progressions to fit this section better (going to work on data science cwk now)
- 05:40 - Rewritten sections on 'Progression Functions' Fixed and Adaptive to fit background research section
- 05:50 - Gone through report and commented out unnecessary sections and notes, cleaned up formatting. Now to move Colin's work over from gdocs to Overleaf and clean that up.
- 07:40 - Completed adding Colin's report contributions and editing to improve the english. Added his references to the bibliography and citated them correctly. Also added his images. Furthermore I added **Incomplete** with an explanation to each section of the document that is incomplete - since this version is what is being sent as a draft.
- 07:45 - Sent Matteo the work as a draft.

This day I worked for 8 hours on the above sections.

23rd April

Considered applying for an extension since we had yet to receive feedback on our report, but only two team members had yet to receive one, and I had already applied for other modules. Only 1 hour of work in communicating with team, since code had been overhauled, but most of which had not been pushed yet, this halted progress on the project at a really inconvenient time. Waiting on multiple team members to finish their delegated sections of the report.

24th April

Waiting for Rayhan to push his final changes to the code overhaul that Luke started so I could begin the final training steps of our models. Helped Luke with LaTeX and re-wrote some sections.

- Went over section 2.2.2 as it was added by Luke to google docs, re-wording some sections and changing sentence structure.
- Helped to explain LaTeX syntax and referencing so that it could be moved over to the Overleaf doc.
- Delegated section on existing reinforcement learning algorithms and PPO to Collin.

Spent around 3 hours working on this day, continuing onto the next day in the morning.

25th April

Rayhan pushed his code, we went over it in a call, and he suggested some changes for me to make, and some training for me to run while I wrote more sections of the report. Could not sleep after working until 11am, worst possible timing with construction and landlord visits.

- Going over 2.2.2 which was a large section on the current state of reinforcement learning, ensuring its structure, content and grammar was up to standard.

    - 01:26 – Re-wording some parts of section 2.2.2 'Current State of Reinforcement Learning'
    - 01:55 – Gone over first two paragraphs of 2.2.2 to makes changes
- Had a meeting with Rayhan and Luke for about 2 hours where Rayhan explained the code as it was and he suggested some changes for me to make, and some training parameters for me to run while I wrote more sections of the report.
    - 04:08 – Code running, plotter working, issues resolved
    - 04:40 – Running Code to train models over larger amount of timesteps (epochs*batchsize) [currently 100*2048] to test before running on Final amount of timesteps which will be [1000*2048], going back to work on report now
        - Need to remember rename 'results/simple_env' after each run to a representative name as is overwritten with each successive run
        - Need to remember to run over at maximum timesteps over both adaptive with 'CumulativeReward' performance function and also 'EpisodeSuccess' performance function and rename the file in 'saved_policies' to reflect this between those runs so that they can be pushed successfully.
    - 05:17 – Added simple parameter to allow turning 'render' of in testing phase
- 05:18 - Note: can change 'final_performance' in FBP_PARAMS to a value lower than 0.5 if on the ct graph it never reaches 1 even with 2048000 epochs.
- 05:20 – Training Algorithms over 2048000 epochs (with Cumulative Reward)
- 05:50 – Going over last paragraph of 2.2.2
- 06:22 – Completed 2.2.2 Going over 2.1 now,
    - FBP algorithm on epoch 705/1000 after 1 hour
- 07:47 – Completed re-wording of 2.1 Machine Learning overview, onto 2.1.1 Supervised Learning now
    - FBP algorithm completed and, standard reinforcement learning algorithm on epoch 655/100 after 1 hour.
- 08:15 – Completed re-wording of first paragraph of 2.1.1,
    - Standard RL aglorithm on epoch 950
- 09:19 – Completed re-wording of entire 2.1.1
- 10:34 – Re-written first paragraph of 2.1.2
    - All algorithms trained finally
    - Now to run adaptive with alternative performance function (episode success)
- 11:32 – 144/1000 epochs of episodesuccess friction based complete
- 11:33 – Rayhan suggests using smaller batch size with more epochs for next run
- 13:20 – Code crashed, large runs require a restart of my PC, lost models.
- 22:30 – Changed some minor code to prevent overwriting across multiple runs.

Worked from 1am till 11am, which is about 10 hours, then worked an additional hour from 22:30pm as I could not sleep that day, really poor mental health here.

I spent around 30 hours and 30 minutes doing work this week, (week 19/4-25/4).

26th April

Woke up at 21:00, we had received our draft back with feedback, ran the code with new parameters as suggested by Rayhan.

- 21:00 – ran the code with new parameters suggested by Rayhan after he made some code changes after a meeting with Andrea.

- I went through the feedback, added the comments that had still not been addressed to our Overleaf document, and assigned those sections to team members.

Worked for 3 hours this day until midnight, continued working after midnight.

27th April

Completed most of changes suggested by supervisor, and had a call with Luke and Rayhan to decide on some structure. Later added in Collin's work and re-wrote a large amount, also implemented all maths equations and sources for his information.

- 00:00 – Summarised into discord and started working on 1 Introduction, 1.2 Objectives and 1.3 Deliverables
- 00:50 – Completed 1.2 Objectives as per Matteo's advice
- 2:23 – Added in hypothesis to Introduction, considering removal of 1.1 Aims and integration of content
- 4:04 – Through discussion with Luke and Rayhan restructured intro, aims and improved objectives
- 4:42 – Added deliverables and cemented project plan and report structure
- 06:46 – Re-worded 2.3 Transfer Learning to a higher level of English
- 14:34 – IDEA: mess around with the matplotlib to make better graphs
    - Or depickle the data and use tableau
- 20:00-23:10 – Added in Collin's RL algos and PPO work, with all equations and citations that he provided
- 23:15-23:50 – Started changing what Rayhan commented on, then wrote description of 5, 5.2 Eval of implementation, Deliverables section and went over conclusion making changes
- 23:50-23:55 filled out signatures and deliverables section in template at top, added final sentence to summary explaining our reached conclusion.
- 23:59 – Barely submitted on time!

Worked 7 hours on this day before going to bed, being unable to sleep, ran the algorithms more times during the time I was unable to sleep, slept for a few hours after 3pm, woke at 8pm and worked until 23:59, which is another 4 hours of work – so 11 hours total for this final day of work.

I spent around 10 hours doing work this week, (week 26/4-1/5).

Conclusion

In total I spent 168.5 hours working on this project, which is short of the prescribed 300 hours. This was largely due to having to wait for responses from team members before progress could be made, along with slightly inaccurate timekeeping, and poor mental health throughout. Had we started making much more significant progress before Christmas, and not ran into as many roadblocks in terms of the code implementation, I am certain we could have achieved a much greater implementation, and thus final report, and that we could have spent more time each on the project.