|  |  |
| --- | --- |
| Student name: | Austin Harper (17009191) |
| Supervisor name: | Phil Legg |
| Title of Project: | Multi-Agent Reinforcement Learning System to Promote Teamwork and Emergent Behaviour in Drones |

Initial Meetings: We discussed and refined the ideas for the project, and any technologies that could be used.

Log 0: Planning Pre-19/10/20

I spent time coming up with possible project ideas, and technologies I would like to work with more.

I spoke to Phil Legg, my supervisor, who helped guide me to my current project proposal using academic papers.

I completed the draft proposal for my project, which included considering the tools to be used, the goals for the system, and the possible ways to implement them.

I started getting familiar with AirSim and OpenAI Gym, two of the key technologies I have planned to use. I have also found projects that are similar to my own, that I may be able to use as a good starting point to help me get to grips with the technology.

I also started looking at academic papers that are relevant to my project.

Meeting (20/10): discussed the above over email as neither of us had questions that require long discussion, and no big jumps in progress were made.

Log 1: Prototyping WC-19/10/20

This week I have spent most of my project time prototyping, with the first goal being to build the interface between AirSim and Gym that will be the backbone of the project.

Previously I thought that this would be a fairly straightforward goal, as I had found GitHub repositories that I could use for reference I order to build this link. However this was naïve, as if I do not understand this part of the project completely then the later parts of the project that build on this will take me much longer to understand.

Considering this, I started from scratch, first following Gym documentation to create a basic custom environment, and understanding the way that Gym environments work. I then started slowly adding elements more relevant to AirSim, taking inspiration from the GitHub repo’s I found. The repo’s were very helpful, as even though the two main ones I used (<https://github.com/hoangtranngoc/AirSim-RL>, <https://github.com/Kjell-K/AirGym>) were for a car agent and out of date respectively, the key basic interface between AirSim and Gym was still relevant.

The use of these repo’s as reference has been instrumental. Without them it would have taken me much longer to get this interface started, as the Gym and AirSim documentation only very lightly covered the areas I need most.

I have started on my literature review, with around 10 relevant papers to my project. I have skim read all of these so far, and plan to read them all fully next week.

I have also done some more research on features that could be implemented, and ML algorithms that could be used.

For next week: Read papers, finish implementing AirSim/Gym THEN start working on getting one drone moving to target.

Meeting (27/10): discussed the above, how to share results (via videos), tools that could be used (Google scholar), the end users (researchers), and that the project can work as an extension of the OpenAI hide and seek paper.

Log 2: Prototyping WC-26/10/20

I have continued to progress my understanding of the AirSim and Gym technologies, the Gym environment has now been constructed, although it still requires testing and possible tweaking after the first agent itself has been written. I am still working toward getting one drone flying to the target. When more drones are added, the environment will need adjusting and updating as it currently only supports one drone with simple abilities.

I have been taking notes on academic papers that I have been reading to help with my literature review.

I started filling out my formal project proposal, which I plan to have completed by 4/11.

Next week: Formal proposal, Take notes on all papers, finish code for agent structure.

Meeting (03/11): discussed the above, requirements (must, should, could, wont), proposal (aims etc.), literature review (for now, 1 line summary of each paper, why it’s good, what are the limitations), and what should be shown at the project in progress day (e.g. 3 drones before training).

Log 3: Prototyping WC-02/11/20

I have finished implementing the AirSim/Gym interface, therefore the Gym environment should now be able to interact with the drone agent structure I have set up. The next step is to implement the learning agent that uses these structures, which will then allow me to test the interface.

I have continued reading papers and taking notes for the literature review.

I submitted my formal project proposal once I confirmed my supervisor was happy with it.

I have also started to get coursework assignments for other modules, meaning I will have less time to spend on my project. The assignments will take priority due to having a sooner due date.

Next week I plan to write out my requirements using the categories of Must, Should, Could, Won’t. I will continue taking notes on papers, and start work on the agent itself.

Meeting (10/11): discussed the above over email, no big changes were made in the code, and no questions needed to be raised. Agreed to send the requirements document over once this had been drafted.

Log 4: Planning WC-09/11/20

Started to look into implementing the learning algorithm into an agent. I plan on first using a learning library to have a prototype, as this would mean I can test the underlying Gym/AirSim environment. I will then work on the learning algorithm(s) that I want to implement myself.

I have drafted the requirements, which was difficult as there was no ‘customer’ to consult bar myself. I thought about what I want to see in the project, and the most important things to get out of it.

I have also made notes on another paper, 4 total sources in UWE Harvard referencing with short notes for later expansion in a research section.

Again, there is more focus on other module assignments this week, especially due to the compressed schedule that is running this year. My capacity for work on my project will likely be reduced until term 2. I still plan on dedicating time to it each week, but how much time is dictated by the due dates and priority of other assessments, which are now being set with due dates mostly in mid-December.

See the current state of requirements and literature to this point below:

**Requirements:**

**Must:**

* Provide a structure between AirSim and OpenAI: Gym that can be used to experiment with agents as drones.
* Have 3 drones with reinforcement learning techniques. 2 drones with a goal to reach a target location, 1 drone with a goal to stop them.
* Drones must be able to improve on their performance, and this must be shown to happen/to have happened (via video, graphs etc.).

**Should:**

1. Show the travelling drones using teamwork and emergent behaviour in order to achieve their goal, perhaps one to distract the counter drone.
   1. This may be difficult to achieve depending on how much training time is needed.
2. Experiment with variables such as speed to get the most out of all agents, in regards to teamwork and emergent behaviour.

**Could:**

1. Provide more opportunities for emergent behaviour, such as adding more observations for each drone, or other actions/abilities. Or more complexity such as moveToPosition being replaced with moveByRollPitchYawThrottle.
2. Add more drones to see if different behaviour emerges.

**Won’t:**

* Test in real life, or in a realistic Unreal environment. This is for agent training and proof on concept only, it is expected that the system could be extended to more realistic environments, but not in this project.
* Emulate all the technology that would be needed in real life, such as for communication between drones for teamwork.
* Create any other supplementary tests in order to assess agent intelligence.

**Literature:**

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B. and Mordatch, I. (2019) Emergent Tool Use From Multi-agent Autocurricula. In: Rush, A., ed. International Conference on Learning Representations. Addis Ababa, Ethiopia, 26 April 2020. (no place): OpenReview.net, pp. 1-28.

* <https://arxiv.org/pdf/1909.07528.pdf>
* Agents create a self-supervised autocurriculum with multiple distinct rounds of emergent strategy, much requires tool use and coordination. Each phase creates new pressure for the opposing team to adapt.
* Good basis for multi-agent reinforcement learning emergent behaviour, with multiple agents and emergent behaviour.
* Limited by its relatively simple environment and rules.

Tran, H. / Medium (2020) *Deep Reinforcement Learning for autonomous vehicles with OpenAI Gym, Keras-RL in AirSim simulator*. Available from: <https://medium.com/analytics-vidhya/deep-reinforcement-learning-for-autonomous-vehicles-with-openai-gym-keras-rl-in-airsim-simulator-196b51f148e4> [Accessed 05 November 2020].

* An autonomous Agent learns to drive on waypoints, using AirSim and OpenAI Gym.
* Shows how AirSim and OpenAI Gym can be used in conjunction to perform reinforcement learning.
* Limited in that the environment is simple, it is a single agent, and it is using a car agent.

Barton, S.L., Waytowich, N.R., Zaroukian, E. and Asher, D.E. (2018) Measuring collaborative emergent behavior in multi-agent reinforcement learning. In *International Conference on Human Systems Engineering and Design: Future Trends and Applications.* Reims, France, 25 October 2018. New York City, USA: Springer, Cham, (pp. 422-427).

* <https://arxiv.org/ftp/arxiv/papers/1807/1807.08663.pdf>
* Multi-agents in a predator-prey task, pack hunting is better than hunting alone. Agents use teamwork to catch the prey.
* Similar in scope to my project, agents oppose each other, prey’s only task is to stay away from predator.

Pham, H.X., La, H.M., Feil-seifer, D. and Nefian, A. (2018) Cooperative and Distributed Reinforcement Learning of Drones For Field Coverage. Computing Research Repository (Corr) [online]. abs/1803.07250 [Accessed 11 November 2020].

* <https://arxiv.org/pdf/1803.07250.pdf>
* Distributed multi-agent reinforcement learning for UAVs, to provide coverage of an unknown areas while minimising overlap.
* Uses game theory to solve joint actions of the team, so they use teamwork to minimise the amount of overlap.
* No counter agents, the only goal here is to optimise.

Next week, plan is to implement a learning library to be able to test the AirSim/Gym interface, add to and refine the requirements, and reference another paper with notes.

Meeting (17/11): discussed the above, with focus on requirements. Requirements should be testable and provable, meaning they must also be specific. I should also consider how I will plan these into sprints. What other constraints could be added. How would a standard model compare to these agents once the adversaries have been removed.

Log 5: Planning WC-16/11/20

I have not made much progress this week, as I have needed to prioritise other modules which all have overlapping due dates.

I have however refined the requirements based on last week’s conversation. See below:

1. **Must:**

*The system must:*

* 1. Include an API to allow AirSim to be used as part of an OpenAI: Gym environment.
  2. Be programmed in Python and run in Unreal Engine.
  3. Include multiple Agents.
  4. Report the Reward function for the Agents, which is based on minimising the distance between the travelling Agent(s) and the target flag, or the opposite for the counter Agent(s).
  5. Show the performance of Agents, through graphs and videos.
  6. Form two ‘teams’ of Agents, in order to have adversaries to challenge each other.

*Each Agent must:*

* 1. Have reinforcement learning techniques implemented.
  2. Observe/Input:
     1. It’s current kinematics, including position and velocity.
     2. The location of the other drones in play.
     3. The location of the target flag.
  3. Output:
     1. A command to move the drone to a new position. Simplified to directions. (x+, x-, y+, y-, z+, z-)

1. **Should:**

*The system should:*

* 1. Include a moving target flag that the travelling Agents are trying to reach.
  2. Allow for the environment to be changed, including the addition of obstacles.
  3. Allow for variables to be easily tweaked. These variables are:
     1. Agent speed.
     2. Number of Agents on each ‘team’.
     3. Location of target flag.
     4. Observation modes, allowing for some observations to be turned off or on, such as the camera.

*The Agents should:*

* 1. Have a number of different reinforcement learning techniques able to be applied to them, in order to assess what techniques are best.
  2. Include the data from the front camera of the drone it controls as an observation.

1. **Could:**

*The system could:*

* 1. Include Car agents for a similar task.
  2. Include a realistic Unreal environment, to show how the agents adapt to a realistic environment.
  3. Compare how an Agent trained against adversaries will perform against an Agent trained without adversaries, when the adversaries are removed.

*The Agents could:*

* 1. Include the ability for drones to ‘stun’ each other when close enough.
  2. Include more complexity for drone movement, allowing the drones to move more freely instead of just in directions.

1. **Won’t:**

*The system won’t:*

* 1. Allow for testing in real life.
  2. Simulate the technology that would be needed in real life, such as for communication between drones for teamwork.
  3. Include any other supplementary tests in order to assess agent intelligence.

These requirements simplify the function of the system to start with, and adds more complexity in the later stages of Should and Could categories. This is to help get a working prototype of all agents faster, then more complex behaviour will be added once a solid base is working and tested.

Next week, I expect to spend most of my time on other assignments again, however I will endeavour to dedicate an amount of time to this project to keep ticking over. This will be in the form of research on implementation of reinforcement learning techniques, and paper reading.

Meeting (24/11): Discussed the above over email, as no major changes were made to the code and no other discussion was needed.

Log 6: Planning WC-23/11/20

Due to the changes I made last week in relation to the requirements I had in mind, I needed to edit the code framework for observations and actions.

The code framework has been changed in order to exclude the camera view observation for now, and simply the possible actions into movement on one of the axes at a time.

I have also made progress on referencing and reading relevant papers.

As expected, most time is prioritising other modules with more pressing deadlines.

Next week, I plan on making more progress on literature, and continue working on implementing a learning technique. This will likely take the form of Q-Learning at first.

**BREAK – For Assignments and Christmas**

Log 7: Implementation WC-04/01/21

Implementation of Q-Learning. I have split the map into a 3d grid, in order to be assigned a Q-Learning state for each grid cube. For now, I have stated there will be 4 specific destination points. The action space is now 6 discrete action, +x, -x, +y, -y, +z, -z.

This means that the actions will better correspond to movement from state to state in Q-Learning. The reward table will be filled by running the drone through the play space.

Most of the code for this is now implemented, debugging and testing is required which will be undertaken next week.

Meeting 12/01: Group meeting, discussion of project in progress day, and plans of amount of completion for then.

Log 8: Notes WC-11/01/21

More notes on implementation, and the report itself. For now I am planning my workload for after the exam period where I will have most of my time available to dedicate to DSP. I feel like this is a better use of my time for now, then I can focus on implementation when I can spend longer amounts of time on it in blocks, rather than bits here and there.

**BREAK – For Exams and Assignments**

Log 9: Implementation WC-08/02/21

Implementation of Q-Learning, with the aim to have a working demo of 1 drone moving to a target for the Project in Progress day. This has been a good week for my understanding of Q-Learning, and I have found that much of my exam prep for Biocomputation has been very beneficial here.

I’m hoping that the demo version will be implemented and working by 20/02/21.

No meeting after this week, however discussion have taken place about different questions over email. Such as how the Agents will be able to ‘think ahead’ to avoid objects etc.

Log 10: Implementation + Poster WC-15/02/21

More implementation and debugging issues when running. I seem to have underestimated the extent of the AirSim simulation which has caused some problems. The axes of the drone are not tied in to unreal at all, (0,0,0) is where the drone starts, and -z is up instead of the expected +z. Due to realism if the velocity is set too fast the drone will flip almost horizontally, causing a big loss in altitude, and the velocity also means that the drone will overshoot most points. In addition the drone movement takes up most of the training time, and I do not have the computational resources to run at faster than real time.

Some of these issues have been solved, but some like the training time have persisted. I have created an alternate implementation so that Q-Tables can be trained and saved without the need for a drone to be moving. This does not mean that I will not be using AirSim, it simply means I have an alternate but identical environment that can speed up the training and testing process where possible.

I have also been working on the poster in this week.

Meeting (23/02): Spoke about Phil’s notes on the draft of my poster, keys points are to include activity diagrams, screenshots of the code, and a Gantt chart.

Log 11: Poster and Demo WC-22/02/21

Mainly working on debugging and ensuring the work above continued as expected.

All other work was dedicated to the Poster and Demo, which were completed on 25/02.

PiP: Discussion with supervisor and second marker on the current state of the project. Key points discussed were

* Saving of a path taken by an agent in a non-simulated environment, and running this path in the simulation to save training time.
* MiniMax or similar to allow for agents to maximise their reward and minimise the other teams.
* First implementation of multi-drone could be moving on a simple path, to test the learning capability of the agents.
* ‘Scaffolding’, training of agents for simple task and building from there. Train Q-tables for no obstacles, introduce obstacles after.
* Use one Q-Table for each team, feedback individual learning at end of episode.
* Microsoft Azure.

Log 12: Formalising Requirements and Methodology WC-01/03/21

Requirements defined in log 5 have been formalised and testing has been considered for each requirement. The requirement definition process has also been written into that section of the report.

Methodology has been written into the report, mainly discussing the semi agile approach to the development, where they have been most useful.

Both of these will need to be revisited nearer the end of the project as it progresses, but the main points are now formalised.

Next week is dedicated to implementing the running of 2 agents on the system.

Meeting 09/03/21: Discussed the important points to be made in the methodology section. Also spoke about how to progress with multiple agents and drones, including how team rewards will work and how this can be made to still ensure both agents work together.

Log 13: Implementation – 2 agents WC-08/03/21

Most of the ability to run 2 agents had been setup previously. A lot of time this week was used for cleaning up code and implementing better systems now that I had a more firm grasp on what was needed. For example epsilon greedy policy was implemented, and the saving and reading of q-tables expanded.

Some issues remain, such as AirSim not identifying the second drone in the simulation, and the question of how team rewards work to ensure drones achieve the best outcome. I would also like to add an opposition drone.

Next week I will be working on my literature review, and where time allows I will complete these tasks.

Meeting 16/03/21: Discussed how best to document the progress being made, likely through short videos, and how to go about solving the issues that remain.

Log 14: Literature Review WC-15/03/21

This week was spent collating, referencing, condensing, and combing sources of background research. I focussed on pointing out where my project fills a ‘gap’ in research, and what it builds upon. This allowed completion of a first draft, which will be built upon and refined as the project continues.

No meeting this week, however the draft literature review has been sent out for pointers.

Log 15: Implementation + Testing WC-22/03/21

This week has been focussed on implementation and testing. A few important points to mention are that I have a counter drone running, mainly as a moving obstacle, I have manual camera control in AirSim, and I have created a system to save runs and view them later in a text based interface. This is allowing for the following testing that I will attempt to have as a minimum baseline by the end of next week. This is to meet the draft deadline my supervisor has set, of the 8th of April. I have also edited my planning to ensure that I can show a draft that is representative of the final project.

Testing to be done and compared. All options will be available to be selected in the final system interface:

* Basic Q-Learning, 1 moving obstacle, episodes of 1, 1000, 100000, 1000000.
* Basic Q-Learning, 1 learning obstacle drone, episodes of 1, 1000, 100000, 1000000
* Basic Q-Learning, 2 learning obstacle drone, episodes of 1, 1000, 100000, 1000000
* Basic Q-Learning, 2 learning obstacle drone, obstacles in environment, episodes of 1, 1000, 100000, 1000000
* As above for more advanced methods such as MinMax, more intelligent policies, at least one advanced method if not more

I will then spend the time after that writing a complete draft for the 8th, covering all topics. If unable to formally write it all I will note what will be written in detail.

No meeting – Easter break.

Log 16: Report Writing WC-29/03/21

As expected, this week has been spent on report writing in order to create a draft report for comments that is representative of the final report. Therefore this has covered all content of the report, writing about the current implementation and deciding on sub-heading and focusses for what is yet to be finalised.

This process has also helped in thinning out the final state of the project, looking at what things are really needed in order to fulfil the objectives set for myself and what I want to get out of it. This has led to a clearer objective for the tool itself, to create something that is specifically designed to be expanded upon in future. The more research focussed side of the project will be comparing different techniques and understanding how to achieve teamwork and emergent behaviour, even teamwork and emergent behaviour are not inherently achieved in the implementation. This is really what I should have put at the heart of the project from the start, and built out from there, instead of the other way around. This will be something to talk about in the reflection section, and I will be able to consider this in future when planning a project that may be too ambitious.

With this being said, the tool is mostly complete, however I would like to include some other reinforcement learning techniques from a library like keras-rl/tensorflow. This will allow for comparisons to be made and will show the framework for expanding on the tool. Other than that the software will require some polish, and some documentation.

Next week will be the final implementation based week, aiming to get the above achieved to allow for time completing the report, and only adding to the implementation where needed.

Log 17: Report Writing and Finalising WC-05/04/21 to 28/04/21

This time period was covered in one log due to the many topics covered, as all areas of the project have been finalised.

In regards to software, evidence has been gathered and analysed for the report, code has been tidied and commented correctly, and the GitHub project has been filled with the current files needed.

In the report, all sections have been added to or completed. The main portion of work has been dedicated to implementation and evaluation.