CITS3001 The Resistance AI Project

This is my collation of information related to the project. If there is anything wrong feel free to comment or ping me @hilmi on the UWA CS/DS discord. I give no guarantee of completeness or correctness of the info presented

Project Page (from here)

- Due: 19/10/2021 Tuesday 5pm (GMT +8)
- Submit: via cssubmit
- 30% of final grade
- Template code

Submission

- **50%** Project/Research Report (2000-3000 words, i.e. roughly 4-6 pages)
    - **20%** Literature review of suitable techniques to The Resistance
    - **20%** Design description and rationale of selected technique
    - **10%** Validation of agent performance
- **50%** Python or Java Implementation
    - **20%** Correctness and code quality (no runtime errors, readable code)
    - **20%** Design (effective use of data structures, algorithms)
    - **10%** Design (effective use of data structures, algorithms)
    - **BONUS 10%** Competition (How well it plays in a tournament)
        - **NOTE**: the environment/validation of your agent does not need to conform to the tournament rules. You can tailor the game, implementation/validation of your agents to your research question
        - i.e. research report and agents are **not** restricted to tournament rules/performance

Getting started

Agent-Oriented Architecture

Your agent will need to follow an Agent Oriented architecture. It will have a number of methods that the environment will call. Most of these methods will simply pass parameters to the agent containing information about the environment (e.g. who was on the mission, and how many betrayed the mission). The agent should use these parameters to update their internal state. Some of these methods will require the agent to perform an action (e.g. propose a mission or vote on a mission). The agent should use the information in their internal state to strategically select the best action.

Probabilities

This scenario is an imperfect information game. True resistance members do not know who the spies are. Resistance members typically want to limit their uncertainty whilst spies (who know everything) want to maximise the uncertainty in their opponents. In the worst case there are hundreds of different possibilities to consider. Good agents will need to consider which scenarios are most likely given the observations they have seen so far. Good spies will need to consider what the resistance members consider most likely scenarios and exploit this. **Statistical modelling will be important here. Bayes rule, Markov models and Monte Carlo tree search are all potential techniques to try.**

Other techniques

Adventurous people may want to consider **epistemic logic, logic programming and theorem proving for reasoning agents, or perhaps try genetic algorithms or neural nets** to build optimised populations of agents.

Research

You are given a lot of scope to choose your research question. For example, you may choose to focus on opponent modelling (which would require **modifying the test harness**), social evolution where population norms may emerge, or identifying optimal strategies in a **perfect information version of the game**. To write a good research report you should be clear on:

1. What the research question is;
2. Why the question is interesting;
3. How you intend to answer the question;
4. What data you collected; and
5. What the data tells us about the question.

This is not the first time an AI resistance tournament has been held. This paper describes a number of different techniques and how well they performed. Note that the **rules of the tournament are different and opponent modelling will be restricted**. Use this paper as a starting point to look up extra references and do a thorough literature review. Make sure that you cite all resources you use in full.

Rules (from the github)

The Resistance

(edits in bold, or strikethrough)

Resistance is a multiplayer game, requiring at least 5 players. One third of the players are selected to be government spies, and the remaining players are members of the resistance. The spies know who all the other spies are, but the resistance members are unable to distinguish the spies.

The game play consists of 5 rounds. In each round, a leader is ~~randomly~~ selected. **The leader is chosen deterministically; all the agents are shuffled and given an id from 0 to n-1, the leaders are then chosen in order from 0 to n-1 and repeated** (credits to Dhruv). That leader then proposes a group to be sent on a mission. The size of the group depends on the numbers of players and the round. All players vote on the group. If players vote to accept the group, the players are sent on a mission. If players vote to reject a group, a new leader is ~~randomly~~ selected **(ordered by id)** and the process repeats. If five groups are rejected in a row, the mission fails **(NOTE: in the original game, spies win the whole game, but Tim has defined it differently in the code)**. When a group of players is selected for a mission, if one person betrays the group the mission will fail, otherwise it will succeed. Only spies can betray the group, but they may choose not to. The mission itself simply involves the players on the mission choosing whether or not to betray the group. This is done privately, and the only public information released is how many people betrayed the group.

If at least 3 missions succeed, the resistance wins. Otherwise the government wins.

AI Bots

1. AI bots must implement the provided agent interface, and we will add restrictions for the amount of computation, and system resources they can use.
2. Agents should have a parameterless constructor
3. Java classes should have unique names. Please append your student number to the end of each class name to avoid clashes.
4. The tournament will be a selection of random games. Game.java has a sample tournament file.
5. Agents will be randomly selected to play in games between 5 to 10 players. There may be two or more instances of the one agent in a game.
6. There will be a large number of plays to give all agents a chance to play both sides.
7. Agents will be ranked on the percentage of games they win, regardless of whether they are a spy or a resistance member.
8. Any attempt to cheat will result in immediate disqualification.
9. Any Agent who crashes, or gets stuck in an infinite loop will be removed from the tournament.
10. Agents who go over time will be penalised or removed from the competition.

Quotes from Tim

Teams

S for students, T for Tim.

- S: If we are in a group, do we have to create two research questions (one for each agent) or just one question is fine ?

    - S: My understanding is that it is important that the report is cohesive even with two agents. So I think the two agents should somehow contribute to the aim of the report. So I don't necessarily think two questions are required.
    - T: Yes, one research question is preferable with two agents. The easiest way to do this is to make the question "Is X better than Y" and one person makes an X agent and one person makes a Y agent.
- S: Hi which java version is the tournament server supporting?
    - T: All of them.
    - S: is there a particular java code style we can refer to that you think is appropriate? I am thinking of the marking of code quality and design. I'd like to make use of things like enhanced for loops over for loops as well as type inference.
    - T: I don't mind as long as it's consistent. If in doubt, use the Google Java Codestyle.
- S: Hi Tim, in the spec for the project it says "...Note that the rules of the tournament are different and opponent modelling will be restricted." What exactly does this mean? How is it that OM would be restricted - is it because we won't be playing the same agent enough times to produce a coherent model of another player? I am considering an OM approach, so wanted to know what we can expect. Thanks!
    - T: I'm not sure where it says that, but it should probably be updated. The rules are pretty much what it says on the box now. However, agents don't have a persistent identity between games, so fine grained opponent modelling won't help much in the tournament. You're free to change this in your own experiments though (i.e. generate a group of agents, give them a persistent identity and have an opponent modeller learn how different agents behave under different circumstances.
- T: You can use libraries as long as they are completely referenced, but you will only be marked on code you have written yourself. You should only submit agents to the tournament that have been substantially written by yourself. We're working on a framework for the tournament where actions are passed as JSON packets so you can run whatever type of agent you like as long as it is able to respond in a timely way,
- S: does using another RL project that's not about The Resistance as a reference and citing it, but reimplementing it ourselves counts as writing our own code?
    - T: probably, depending on what you mean by "using" and "reimplementing". If you create the code that's fine (even if it is inspired by someone's earlier code), but if you just transcribe the code, then it's not your creation.
- S: For the report, does the report have to relate to the resistance? Or can you do a general report on something like perfect vs imperfect information games?
    - T: No, it has to be resistance.
- S: Hi Tim French, I am wondering how we can get 40 percent of the code part? Is it accessible if the code is runnable, readable and has a reasonable data structure? Is there a limit to how many times my agent must win against random agents, for example?
    - T: It will depend a lot on what your agent is but, well designed and well implemented is what we are looking for. Performance isn't explicitly assessed, although anything obvious defects will be penalised. As you decide the research question and the validation, you essentially get to propose the performance metrics against which you will be assessed. eg. if you only focus on optimising resistance play, then that's what you are assessed on.

Discord

Go to the CS/DS discord and use the search feature:
in:cits3001 from:Dr Tim French

- T: Also, don't feel you have to write agents just to compete in the tournament. If you want to to investigate untimed games, with opponent modelling and reinforcement learning using cloud resources, go for it. It will make an interesting research report, and you can submit all your files in the zip. We just want you to submit a tournament ready agent as well for fun (ie the tournament at the end).
- T: sure you can write the report for one agent, but submit another. It's often the case students might write an interesting machine learning agent, and a dull bespoke strategy agent to train against, but end up playing the bespoke agent in the tournament, if the learning doesn't go according to plan.
- S: What is the significance of the zip file that we submit? i mean is it just meant to be the supplementary files to make our agent run or if your agent can run with everything inside its own file, do you need to submit a zip?
    - T: It should enable me to rerun your validation experiments. Eg your report could be on agent learning rates under different settings, which would require a very different experimental setup to the tournament.

Email

Feel free to send me screenshots or comments of emails from Tim that contain good info.

Personal Tips:

- Read/skim parts of The Resistance research paper
- Check out Game Theory and how it relates to The Resistance
- Look up the techniques bolded earlier online (wikipedia, youtube, etc.)
    - Find reference implementations (similar target game = better)
    - Decide what interests you and what is (in)feasible for you to implement based on
        - Time, knowledge, complexity, skills etc.
    - Once you decide the technique(s) you want to use
        - Find more resources
        - Start designing/implementing your agent (basic -> complex)
            - Keeping in mind the agents and supporting code should be geared towards your research question
            - You can even modify the game to suit your research question
    - Keep the reference for everything you looked up for...

        - the literature review of suitable techniques (20% of mark, and roughly 40% of your report)
        - The design description and rationale of your selected technique (marks and importance as above)
- Follow Tim's advice on how to write a good research report.
    - Reformulate the technique you're implementing into a research question
    - Keep the list of questions Tim presents in mind
- Don't tunnel vision on tournament performance (10%), focus on writing a good report and implementing your agents well (100%)