CITS3001 Project 2022

Name	StudentID	Degree
Kristijan Korunoski	22966841	Physics & Computer Science
Name	**StudentID**	**Degree**
Amandeep Singh	22904833	Computer Science & Cyber Security

Table of Contents

Introduction

Explain your understanding of the game in one paragraph.

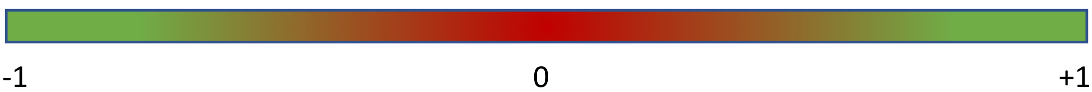
The game consists of 4 agents, red, green, blue, and grey. The green agents represent people in a democratic society, who have the choice to vote or not vote in an election. The red agent represents a political party can influence the opinion and uncertainty of the green agents. Each action by the red agent may cause them to lose the ability to interact with some green agents for the next few rounds, diminishing their influence. Higher power actions cause more follower loss for the red agent. The blue agent represents another political party that can influence the opinion and uncertainty of the green agents. Each action by the blue agent has an energy cost associated with it, the blue agent's actions are restricted by this energy cost. The grey agents can represent politicians that the blue agent can ask to make a speech instead of making an interaction with the green agents. The grey agent will interact with the green agents and influence their opinions and uncertainties without lowering the blue agent energy level. There is a chance that the grey agent is a spy for the red agent and will influence green agents towards the red agent's desired outcome. The blue and red agents take turns interacting with the green agents, at the end of each round the green agent's communicate and influence each other's uncertainties. This goes on for 20 rounds and then either red or blue wins based on the percentage of green agents who choose to vote.

Assumptions

State all your assumptions, including but not limited to:

1. What is the interval of uncertainty in your project? What do -1, 0, +1 represent?

   The interval of uncertainty in the project lies on a distribution between -1 and +1. Where if a green agent has -1 uncertainty the green agent is 100% sure they do not want to vote, at +1 uncertainty the agent is 100% sure they do want to vote, and at 0 the agent is completely uncertain about their decision and can be swayed either way. The figure below shows the uncertainty interval where green represents an agent that is very sure about their decision and red represents an agent that is very unsure about their decision.

2. How are you perceiving green nodes' opinion? Do you perceive it as vote/not vote in election, or are you perceiving it as vote for blue/vote for red?

   The opinion of green agents is perceived as either willing to vote in the election or not willing to vote in the election.

Selection and design of appropriate AI technology

Methodology

1. Describe and justify your methodology for this project, including

    a. Which parameters are hard coded?

    The cost of energy for each attack by blue, the rate of follower loss, and the distribution of green agents willing to vote will be split equally at the start of the game, i.e. 50% of the green agents will be willing to vote at the start of a game, and 50% will not be willing to vote.

    b. Which parameters are to be input at the start of the game?

    There are two options at the start of the game, the player can choose to either create a custom game, with customized parameters, or play a game with default parameters where only the green agent network will be randomized. In a custom game the player can set; the number of green agents, the average number of connections between green agents, a starting uncertainty interval (where the input has influence on how sure a green agent is about their decision but not the decision itself), the number of grey agents, and the number of those grey agents which are spies. The player can then choose to play as the blue agent, red agent, or run automatic simulations of the game.

    c. what type of methods you used to make your agents intelligent.

    We identified that the agents, both red and blue, are facing sequential decision problems (SDP) and we chose to use a value iteration approach to solve the SDP's. For the blue agent rewards are defined by the energy cost, benefit of action, and the energy level of the blue agent. One of the 5 energy levels are only chosen by the blue agent if there is sufficient stored energy to outweigh the loss of energy. For the red agent rewards are defined by the potential for follower loss, benefit of action, and the current percentage of the green network that the red agent can interact with. One of the 5 energy levels are only chosen by the red agent if there is a sufficient population of the green network that will be affected by the message, i.e. red does not want to use a strong action over a small portion of the green network as this will cause further follower loss.

Game Play

1. Explain in detail how the game is played?

    If a human wants to play, they have the option to select to either play as the blue agent or the red agent. The game then follows the structure as follows, the player is asked about what action they would like to take. The computer then decides the turn for the opposing agent and the interactions complete, altering the green network. The game continues for 20 rounds, or until the blue agent depletes their energy. If the blue agent depletes all their energy before 10 rounds have been reached, they lose the game regardless of the state of the green network. Otherwise at the end of the game the green network is analysed and the number of green agents who want to vote or do not want to vote is returned. If more green agents want to vote blue wins, if more green agents do not want to vote red wins, otherwise the game is a draw. Before each turn by the player the distribution of uncertainties and a visualisation of the green network is displayed.

2. How are turns organised?

   BLUE takes their turn first, then RED takes their turn, then GREEN agents interact, this cycle repeats for 20 rounds.

3. How opinions and uncertainties are updated?

   Both the opinion and uncertainty of a node are contained in the "uncertainty interval" we have used (see Assumption 1).

   Blue Interactions:
   An interaction by a blue agent effects every green agent's uncertainty. The uncertainty is changed by the magnitude of power chosen modified by a constant "strongBlue" towards the +1 direction. "strongBlue" is randomised each game so the user will have a unique experience. E.g. a power level 5 attack will push every green agent towards the +1 directions of the "uncertainty interval" by 0.27 (this is an example and this number will depend on "strongBlue") so the following network evolves as follows.

Green Agent	Uncertainty Before	Voting Before	Uncertainty After	Voting After
1	-0.14	No	0.13	Yes
2	-0.77	No	-0.50	No
3	0.91	Yes	1.00	Yes

   Red Interactions:
   An interaction by a red agent effects the uncertainty of every green agent who is following the red team (follower loss is discussed later). The uncertainty is changed by the magnitude of power chosen modified by a constant "strongRed" towards the -1 direction, unless a follower is already below -0.7 on the "uncertainty interval" and the power of the attack is over 3 in magnitude, in that case the follower perceives the attack as "over the top" and "fake" and instead their uncertainty is flipped, i.e. their opinion is changed. "strongRed" is randomised each game so the user will have a unique experience. There is one extra case involved with red interactions, if a red move has been played 3 times in the past 5 moves then all green agents that have uncertainties below 0 that red can interact with will have their uncertainties flipped

to the positive, i.e. changing their opinion. E.g. a power level 5 attack will push most green agents that can be communicated with towards the -1 directions of the "uncertainty interval" by 0.25 (this is an example and this number will depend on "strongRed", unless the green agent's uncertainty is below -0.7 where the change would be flipped and halved.

Green Agent	Can Interact	Uncertainty Before	Voting Before	Uncertainty After	Voting After
1	Yes	0.14	Yes	-0.11	No
2	Yes	-0.51	No	-0.76	No
3	No	0.91	Yes	0.91	Yes
4	Yes	-0.82	No	0.82	Yes

Green Interactions:
When green agents interact with another green agent to which they have a connection their "uncertainty interval" is influenced. The absolute value of each interacting agent's uncertainty is taken, this then represents how sure the agent is about their choice, regardless of the choice itself. If both agents have the same value here nothing occurs. If one agent's value is higher, we then find the difference between each agent's position on the uncertainty interval (not the absolute values), this is then divided by 10 and the agent who is less sure about their decision has their uncertainty (on the uncertainty interval) changed by this amount. This therefore affects both uncertainty and opinion of agents.

Grey Interactions:
Grey agents who are not spies will interact in the same way the blue agent does, however it will not decrease the blue agent energy level. A grey agent which is a spy interacts in the same way a red agent does, but it does not cause any follower loss.

Implementation of the Agents

1. State main points about the implementation of agents. This heading is more focused on the code and how you made it efficient.

   Blue:
   - Selection of next move runs in O(1) as it is only dependant on the energy level.
   - Interaction with green network runs in O(n) as blue completes an action for "n" green agents.

   Red:
   - Selection of next move runs in O(n) as to determine the percentage of followers lost in the green network each agent must be checked to see if they still follow red.
   - Interaction with green network runs in O(n) worst case, however it is almost always more efficient as actions only complete for the number of agents red can interact with, which is almost always less than "n" agents.

   Grey:

- Grey has no selection of a move and so only its interaction has a complexity in worst case of O(n). This is because grey interactions are always identical to either red or blue agent interactions, both which have worst case complexity O(n)

Green:
- Green has no selection actions and so only its interaction has a complexity in worst case of O(n*a/2). This is because "n" green agents will each interact with "a" amount of other green agents, however as we pass hallway through all agents in the network there is a higher chance that these agents will already have had their interaction links completed, hence the factor of a half.

2. How long the program takes to run a single turn with variable number of green agents. Report for both small and large number of green agents.

   Times to run a single round (Blue, Red, and Green interactions)

Number of Green Agents	Time
10	0.11 s
50	0.54 s
100	0.71 s
200	0.98 s
500	1.55 s
2000	4.95 s

   (All with 2 average connections)
   (This does not take into account time for igraph and matplotlib to import, instead these are taken after these libraries have imported)

3. Which programming language you used? Whether you followed an Object Oriented approach or not.

   We have used Python as the programming language and have followed a functional programming approach. The green network follows an Object Oriented approach to some extent.

4. Which libraries you have used?

   We have used random, matplotlib, and igraph.

Running the game

How can a layman run your game? Provide the commands, and associated parameters needed with an example workflow of the game.

We assume the layman has a version of python with the correct libraries installed and they have enough understanding to be able to locate and run the script. If they do not have this knowledge there is many tutorials online regarding how to set up a python environment.

Once the game is running there are prompts given to the user at each point in the game regarding what inputs should be given. There are also checks to ensure that the user enters the correct input, if the input is incorrect the user will be prompted again.

Below is an example of how a user can run the game.

```
How would you like to play? (Enter 'c' for a custom game or 'p' for a game with set parameters)
p
Who would you like to play as? (Enter 'b' to play as blue team or 'r' to play as red team or 'a' to run an automatic simulation of game)
b
Player Energy Level:  100
GREY agents remaining:  3
Enter a power level or 'G' to use a GREY Agent:
 1 - 10 Energy
 2 - 15 Energy
 3 - 25 Energy
 4 - 30 Energy
 5 - 40 Energy
 G to use a GREY Agent
5
Number of People willing to Vote :  5
Number of People not willing to Vote :  5
Player Energy Level:  75
GREY agents remaining:  3
Enter a power level or 'G' to use a GREY Agent:
 1 - 10 Energy
 2 - 15 Energy
 3 - 25 Energy
 4 - 30 Energy
 5 - 40 Energy
 G to use a GREY Agent
five
Incorrect input, try again
Enter a power level or 'G' to use a GREY Agent:
 1 - 10 Energy
 2 - 15 Energy
 3 - 25 Energy
 4 - 30 Energy
 5 - 40 Energy
 G to use a GREY Agent
```

Agent Design

Green Agents

1. Are you using a static network or a dynamic network?

   We are using a static network. The green network is randomised at the start of the game and remains this way for it's entirety.

2. Can we generate a network when the game start?

   The user can choose to either generate a custom network, as described earlier, or use the default parameters to quickly start play. In both scenarios the network will be randomized, that is the links will not always be between the same green agents, and the same green agents are likely have a different opinion and uncertainty to the last time the game was played, even if the same parameters are given. This allows for a unique game every time.

3. What type of underlying network model you are using?

   We have used a random network model.

4. Other properties of the network, e.g., is it weighed, can links be added or removed during the play?

The network is neither weighted nor can links be changed during the game. Some green agents have more influence over the entire network however as if the first agent who begins communicating to another agent is certain about its decision, whether that be to vote or not to vote it will then influence the next green agent, who will communicate with its own connections but now with an already altered uncertainty/opinion. This means the initial green node that starts communications is a valuable target for both the blue and red agent. As all green nodes are identical and both red and blue agents influence the entire network (except when red experiences follower loss) the high value agent cannot be targeted.

Red and blue agents

1. Describe your design of the message potency (a.k.a. uncertainty of red and blue nodes.)

   The message potency of both red and blue is split into 5 levels of attack potency. For both red and blue 1 has the lowest power, the power increases by one unit to the highest power level, 5.

2. Describe your method for changing the followers' number in case of red agent

   Each attack that a red agent can make has a different message potency and an associated follower loss. Whenever an attack is made by the red agent, each green agent that the red agent is interacting with has a chance to close communication between themselves and the red agent for the next few rounds. We refer to this as follower loss. The probability that a follower is lost is related to how potent the attack by red was. Follower loss is not permanent, and followers will return to the red agent after 2 rounds of play.

3. Describe your method for changing the energy level of blue nodes (a.k.a. lifeline)

   Each attack that a blue agent can make has a different message potency and an associated cost of that attack. Whenever an attack is made by the blue agent, the energy level decreases by the corresponding cost of attack. After each attack by the red agent, the blue energy level is increased by 15.

Grey Agents

Describe the working of the grey agents…

Grey agents work the same way as blue agents, without the risk of the blue agent’s energy level decreasing. If a grey agent is a spy, then it works the same way as a red agent would, without the risk of losing followers. When the blue agent chooses to call for help from a grey agent that agent is randomly selected from an array of grey agents, where some may be spies. The selected grey agent will then interact with the appropriate green agents, depending on whether it is a spy or not. Following this round the grey agent will disappear and will not be able to be used again in the game.

Validation of Agents

1. Report any tests you conducted to ensure the agents are doing the task they have been asked to do.

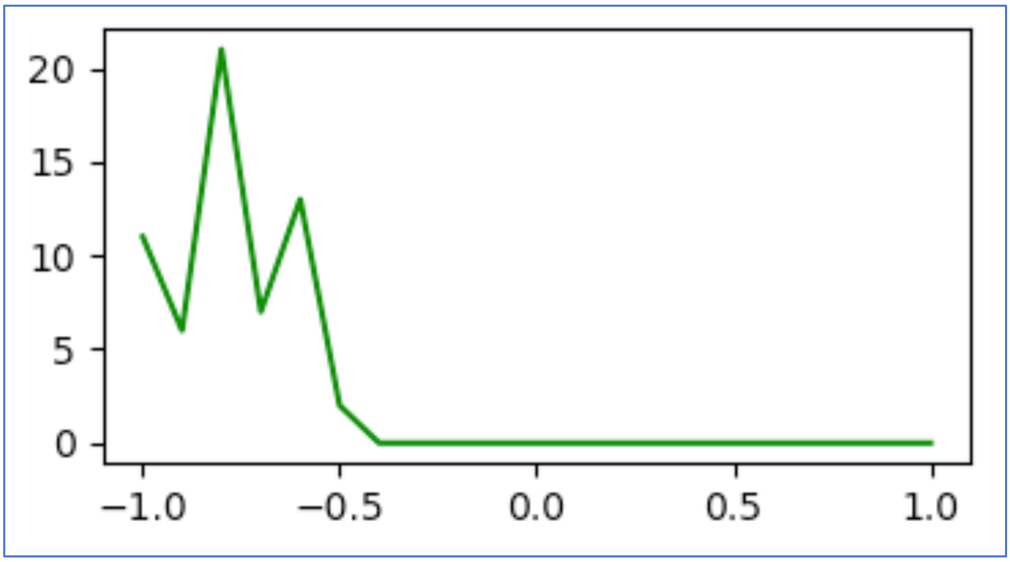
   We called functions for agents to act on various test green agent network while graphing the uncertainties of all green agents to check that they were working as intended. We then implemented systems for agents to be able to be played by a computer, i.e. a computer selects an attack, for both red and blue whilst playing as the opposing agent. We also played the agents against each other and simulated many rounds, making alterations if we found that an agent was not performing optimally.

Perform Various Simulations for the following set of questions.

* All simulations were performed on a network with 20 green agents with 3 average connections *

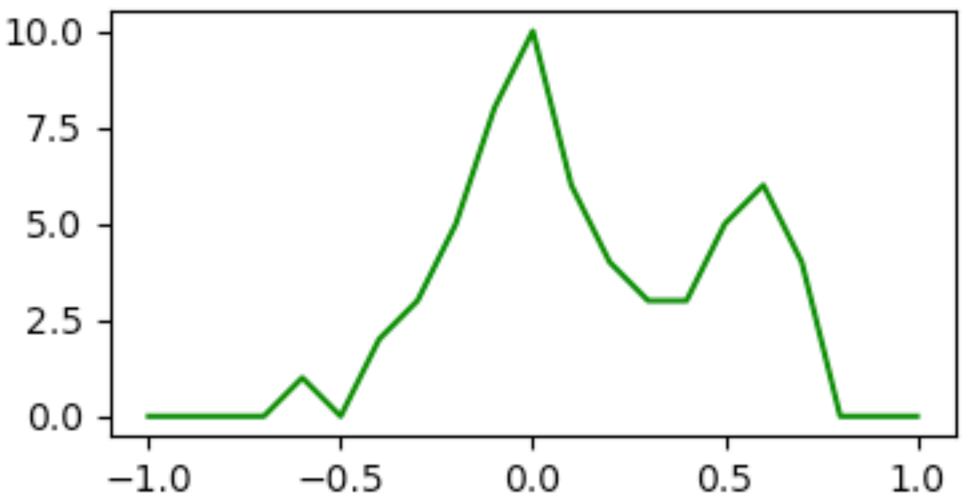
2. How does the game change if you have a tight uncertainty interval at the beginning of the game?

   If we start with a tight uncertainty [-0.1,0.1] (as given in the project specifications) we observe that the agent who by randomness makes a slightly stronger play earlier in the game will take control of the game earlier. This will then make it much more difficult for the opposing agent to make a comeback as the interaction of the green network will also work in favour of the agent who played a stronger move. By stronger move we mean for example the red agent's followers, if the red agent losses a follower that has many links they are now disadvantaged. Below we see plot of the distribution of uncertainties at the end of a game. Almost all nodes are at the -1 pole indicating most nodes are not willing to vote and very certain about their decision.



3. How does the game change if you have a broad uncertainty interval at the beginning of the game?

   If we start with a broad uncertainty interval [-0.5,0.5] (as given in the project specifications) at the start of a game the game is more forgiving. Even if an agent is lucky and the opposing agent begins to lose many nodes, the broad uncertainties in the green network will allow the opposing agent to possibly make a comeback. As green network interactions have some nodes with high uncertainties towards the opposing agents opinion the green interaction has a chance to support the opposing agent. Below we see plot of the distribution of uncertainties at the end of a game. Although there is still a high number of nodes towards the +1 pole indicating most nodes are willing to vote, there is still a large number of nodes that are closer to the middle of the interval. This indicates that most nodes will vote but given another round they may have changed their mind as they are not very certain about their decision.



4. Plot distribution of uncertainties for each of the above questions.

5. In order for the Red agent to win (i.e., a higher number of green agents with the opinion “not vote”, and an uncertainty less than 0 (which means they are pretty certain about their choice)), what is the best strategy?

   The best strategy for the red agent is to play a high power move, followed by a medium, power move, followed by a high power move, followed by another medium power move, followed by a low power move. By repeating this cycle of moves the red agent is able to influence many green agents without risking losing high value targets for too many rounds. This also allows the red agent to avoid being perceived by some green agents as “over the top” and “fake”.

   a. Discuss and show with simulation results how many rounds the red agent needs in order to win.

      As our implementation requires 20 rounds to end a game there is no correct answer for how many rounds it takes for the red agent to win. If there was infinite rounds, due to blue’s energy loss, and red’s followers returning the game can be swayed back and forth forever if both agents are equally matched.

6. In order for the Blue agent to win (i.e., a higher number of green agents with an opinion “vote”, and an uncertainty less than 0 (which means they are pretty certain about their choice)), what is the best strategy?

   Blue can never lose all of its energy, if it does it loses the game, therefore a basic strategy for blue is to keep it’s energy above 0. The results from simulation show that a good strategy for blue is to use 3 high power attacks (4 or 5) for the first move then play a grey agent (and hope it isn’t a spy). Following this blue will play low power attacks to regain energy and release a grey spy following this, in the case the grey spy is a spy if the red agent has caused the uncertainty to be low the grey spy may be perceived as “fake” and assist the blue agent despite being a spy. The blue agent will then conservatively for a few rounds, not losing or gaining much energy (level 3 attacks). Following this the final grey agent is released and a strong attack is made by blue. For the rest of the game blue uses low to medium power moves (1, 2 or 3).

   a. Discuss and show with simulation results how many rounds the blue agent needs in order to win.

      As our implementation requires 20 rounds to end a game there is no correct answer for how many rounds it takes for the blue agent to win. If there was infinite rounds, due to blue’s energy loss, and red’s followers returning the game can be swayed back and forth forever if both agents are equally matched.

b. What impact did grey agents have on the simulation? Explain how did you test the impact of grey agents?

Grey agents impact the simulation heavily, the proportion of grey agents which are red spies will also impact the simulation as the blue agent is unaware that they are spies. To test the impact of grey agents we first simulated many games with 0 grey agents, in these tests' red seemed to have an advantage, winning 72% of the time. We then simulated many games with all grey agents not being spies, this helped blue win more games, giving them an advantage, winning 77% of games. We then ran more simulations with grey agents all being spies, this turned out to overwhelmingly assist the red team to victory very often, winning 83% of games.

Performance of Agent when playing with a human

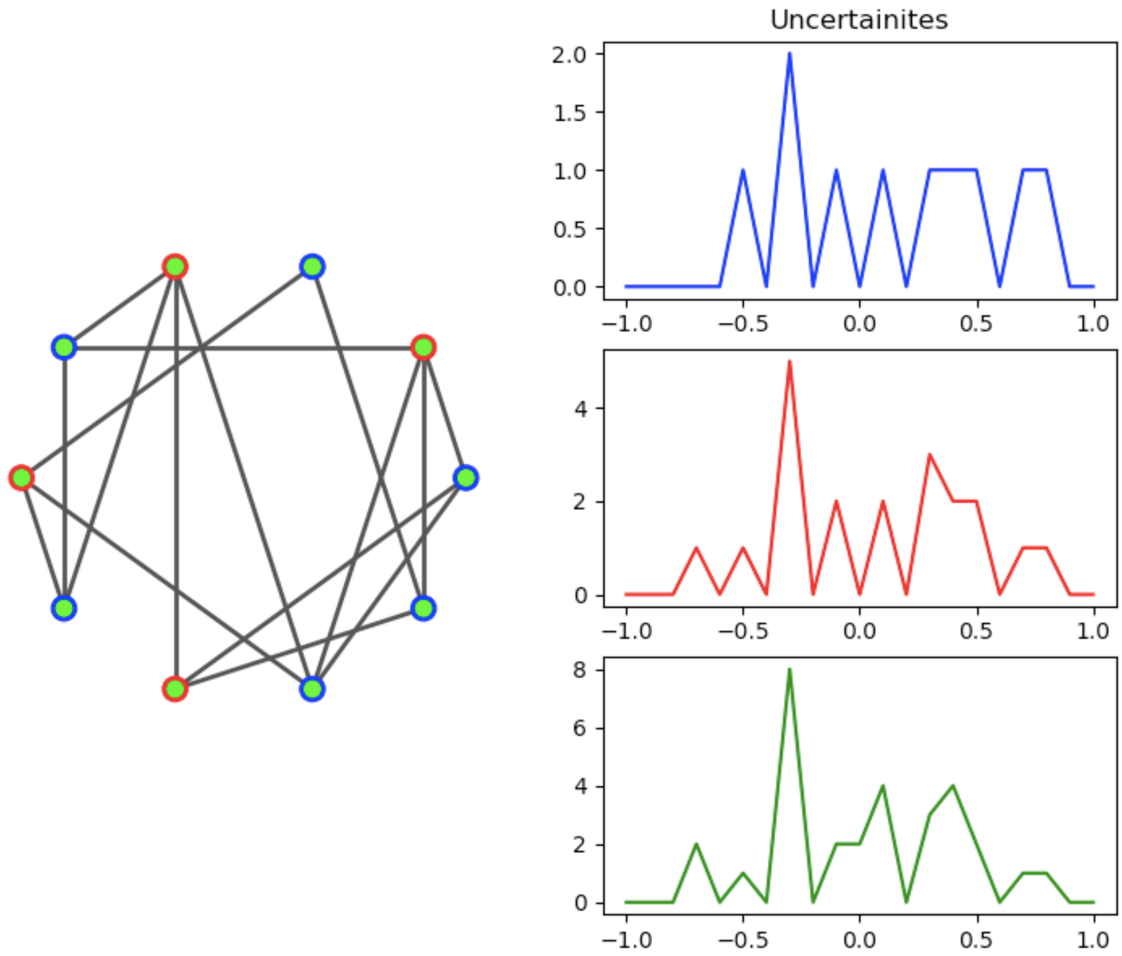
Does the agent run and performs at an excellent level with challenging play when the opponent is a human? Discuss your findings

When playing as either the red agent or the blue agent a human player will experience a challenge. A player who plays the game for the first time will likely lose to the opposing agent, unless the player starts the game with an uncertainty interval that benefits their agent. Even with this added benefit the opposing agent still poses a challenge and if a player naively chooses a random move, they will likely lose. Even after playing a few games the player will still be challenged by the opposing agent. When agents are opposed against one another (computer vs computer) and neither team has an uncertainty interval that benefits them initially the outcome will be around half of the wins to each agent.

Visualisation

Describe your visualisation methods with some screenshots

The program plots a visualisation of the green network along with a plot of the uncertainty distribution after the interactions by each agent.



In the above diagram we have started a game with 10 green nodes, each with an average of two connections. The left side shows the green network, where each green circle is a green agent. The outlines around the green agents show whether the agent is willing to vote (blue outline) or not willing to vote (red outline). The lines between green agents represent connections between agents. On the right we see distributions of uncertainties after each agent interacts with the green network, first blue interacts, then red, then green. On the top we see the blue agent has interacted with the network and influenced all green nodes towards the +1 end of the interval. In the middle we see the red agent has interacted with some of the green network, this is expected as during gameplay the red agent loses followers with attacks. On the bottom we see that all green agents have interacted with one another via their links, this has altered the distribution.

When the player selects to play as the blue team the text-based user interface shows other information and the cost of each possible action.

```
Number of People willing to Vote :  5
Number of People not willing to Vote :  5
Player Energy Level:  50
GREY agents remaining:  3
Enter a power level or '6' to use a GREY Agent:
 1 - 10 Energy
 2 - 15 Energy
 3 - 25 Energy
 4 - 30 Energy
 5 - 40 Energy
 6 to use a GREY Agent
```

When the player selects to play as the red team a similar text-based user interface is shown.

```
Number of People willing to Vote :  1
Number of People not willing to Vote :  9
Red can interact with  70 % of the population
Enter a power level :
 1 - Around 10% Followers Loss
 2 - Around 20% Followers Loss
 3 - Around 30% Followers Loss
 4 - Around 40% Followers Loss
 5 - Around 50% Followers Loss
```

When running an automatic simulation the output states of each simulation are displayed both in the text-based user interface and as a visualisation of the green network, as shown earlier.

```
Enter no. of times you want the simulation to run (e.g. 10) (Maximum : 100): 10
Game No.  1 -- VOTING :  0 -- NOT VOTING :  10
Game No.  2 -- VOTING :  6 -- NOT VOTING :  4
Game No.  3 -- VOTING :  10 -- NOT VOTING :  0
Game No.  4 -- VOTING :  3 -- NOT VOTING :  7
Game No.  5 -- VOTING :  8 -- NOT VOTING :  2
Game No.  6 -- VOTING :  10 -- NOT VOTING :  0
Game No.  7 -- VOTING :  0 -- NOT VOTING :  10
Game No.  8 -- VOTING :  0 -- NOT VOTING :  10
Game No.  9 -- VOTING :  0 -- NOT VOTING :  10
Game No.  10 -- VOTING :  10 -- NOT VOTING :  0

Red Won  5  Games

Blue Won  5  Games

Draw  0  Games
```