Surround and Capturing Adversarial Agents through Decentralized Multi-Agent Intelligence

By Anthony Lowhur

Abstract

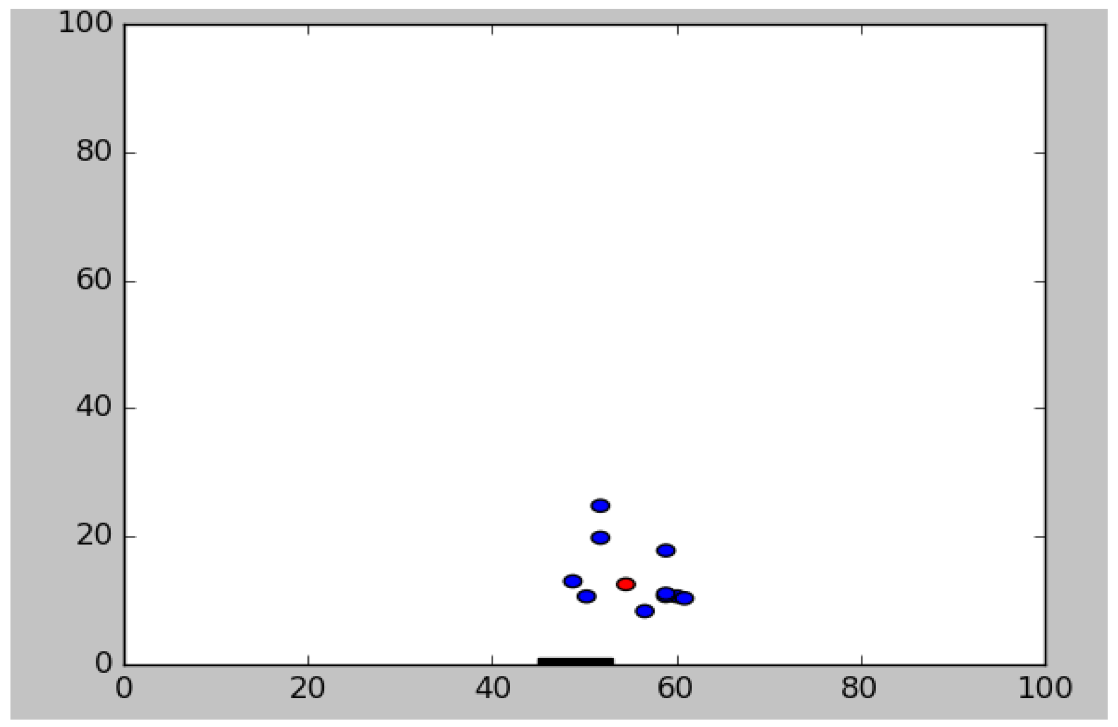
Security and surveillance of buildings and military property has essential for protection of valuable items and even human lives. However, the lives of guards are often at risk when dealing with a quick and cunning intruder. Chasing down an intruder often involves working as a team with other guards and coordinate interception strategies together to capture the intruder. A robot [1] or a team of robot security guards has been looked at favorably for surveillance and security in order to reduce the loss of human lives as well as having optimal performance in guarding an area. With the rapid growth of autonomous systems making up security robots and drones (unmanned aerial or ground vehicles), the need for a robust algorithm to coordinate the group becomes essential. The algorithm we made was based on decentralized approach to multi agent systems. The agents, knowing the location of enemy agent and the exit, behaves independent of each other and works together to surround / block the enemy agent and capture it. This would allow more effective coordination between robot agents that are chasing down an enemy whether it is UAVs and UGVs.

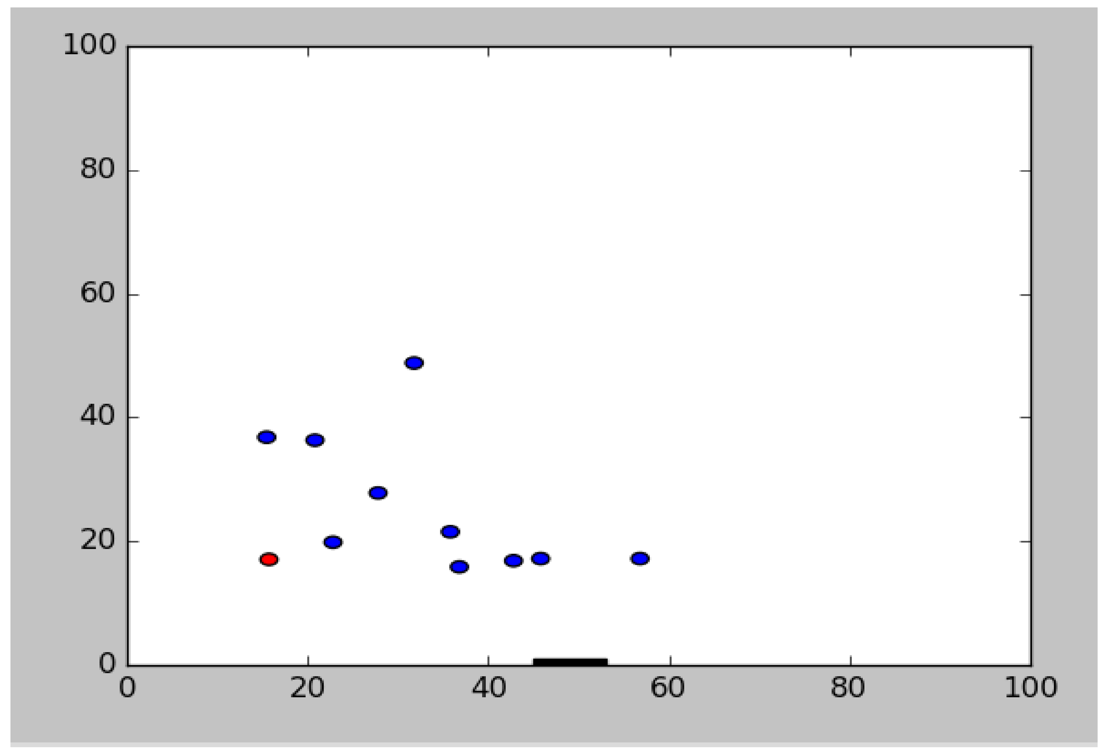
Background

We have been working on a simulation where a team of robots (blue) are trying to capture an enemy agent (red) who is faster and more agile than any of the blue agents. However the team of blue agents must be able to capture the enemy agent before the enemy agent escapes through an exit. We used the matplotlib (Python) and its animation functions in order to construct the agent.

In the simulator, there are the initial position of the red agent and each of the blue agent is randomly initialized. There is an exit, displayed as a black rectangle, that the enemy agent will try to run towards to. For the enemy's navigation algorithm, we created an artificial potential field [2] around both the exit and each of the blue agents. In respect to the enemy agent, the exit has an attraction field that attracts the agent towards the exit, and the blue agents have a repulsive field that pushes the away from the exit. This would allow the enemy agent to maneuver around the individual blue agents while heading towards the exit.

The blue agent team wins if one of the blue agents manage to touch the red agent. The red agent wins if it arrives at the exit without being touched.



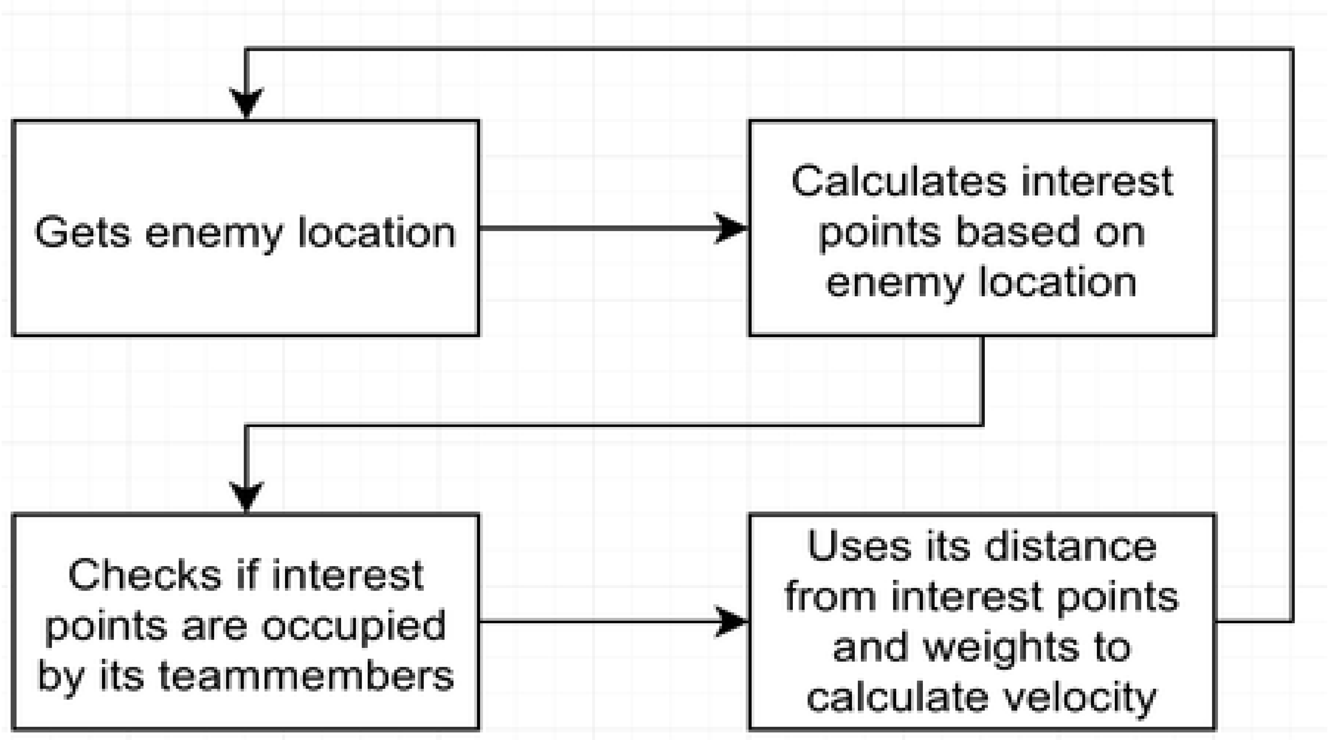


Set of 100 runs	Success ratio (out of 100 runs)
1	100/100
2	98/100
3	99/100
4	98/100
5	97/100
6	100/100
7	99/100
8	100/100
9	97/100
10	96/100
11	99/100
12	97/100
13	97/100
14	99/100
15	96/100
16	100/100
17	100/100
18	99/100
19	98/100
20	100/100
21	99/100
22	100/100
23	98/100
24	99/100
25	100/100
26	97/100
27	98/100
28	98/100
29	99/100
30	97/100

Results

We ran the simulation multiple times and measured their results in terms of the amount of victories there were out of the total amount of runs. In this case, we ran the simulation for a 100 runs per set and ran a total of 30 sets (therefore, the simulation ran 3000 times). The blue agents seem to be rather successful at capturing the red agent, scoring a range of 96-100 victories out of a 100 runs each time. Even though the red agent has strong maneuver abilities to avoid the agent, the blue agents managed to work together to surround and finally capture the agent each run. While there are some instances where the red agent outmaneuver all the blue agents and escape, the blue agents managed to capture the red agent a majority of the time.

Algorithm of Blue Agent



Future Works:

As the blue agents in the simulation tends to compete for certain interest points, we may add an algorithm that would help improve the efficiency of the selection of interest points in order for quicker surround and capture formations.

We will also try to test this on an enemy with a better navigation and obstacle avoidance algorithm to test on how well our multi-agent algorithm stand up to more intelligent adversarial agents.

Potential fields also seem to be promising in controlling multiple agents and creating complex behaviors, so we may investigate in implementing potential fields somewhere in our multi-agent algorithm.

Acknowledgements:

We would like to thank Dr. Zhang and Dr. Gelfond for their support and advice in the project as well as maintaining the Texas Tech University REU program.

References

[1] Rimon, E., & Koditschek, D. E. (1992, October 1). Exact Robot Navigation Using Artificial Potential Functions. Retrieved from http://repository.upenn.edu/cgi/viewcontent.cgi?article=1364&context=ese_papers

[2] Di, D., Milella, A., Cicirelli, G., & Distante, A. (2010). An Autonomous Mobile Robotic System for Surveillance of Indoor Environments. Int J Adv Robotic Sy International Journal of Advanced Robotic Systems, 1. doi:10.5772/7254

[3] Kennedy, J., & Eberhart, R. (n.d.). Particle Swarm Optimization. Retrieved from http://www.cs.tufts.edu/comp/150GA/homeworks/hw3/_reading6 1995 particle swarming.pdf