The First International Maritime Capture the Flag Competition: Lessons Learned and Future Directions

Prithviraj Dasgupta[1*], **John Kliem**[1*], **Zachary Serlin**[2], **Michael Novitzky**[3], **Jordan Beason**[3], **Tyler Errico**[3], **Michael Benjamin**[4], **Tyler Paine**[4], **Peter Crowley**[5,2], **Simon Lucas**[6], **James Chao**[7], **Wiktor Piotrowski**[7], **Gautham Dixit**[7], **Matthew Leprell**[3], **Jeffery Richley**[8], **Christopher Tucker**[8], **Abdullah Alamar**[9], **Katsuki Ohto**[10], **John Meo**[10]

[1]U.S. Naval Research Laboratory, Washington, DC
[2]Massachusetts Institute of Technology - Lincoln Laboratory
[3]United States Military Academy, West Point, NY
[4]Massachusetts Institute of Technology
[5]Boston University
[6]Queen Mary University of London
[7]Naval Information Warfare Center Pacific, San Diego, CA
[8]Naval Information Warfare Center Atlantic, Norfolk, VA
[9] Egypt-Japan University of Science and Technology, Alexandria, Egypt
[10]Independent
{prithviraj.dasgupta.civ, john.kliem3.civ}@us.navy.mil[1]

Abstract

Maritime Capture the Flag (MCTF) is a 3-vs-3 multi-agent real-time strategy game that utilizes a marine robotics simulator with support for hardware deployment. The game presents several research challenges in the areas of coordination and communication of multi-agent teams in adversarial environments with sparse rewards, and safe autonomy. In this paper, we report our experiences and challenges in deploying the MCTF game as an open, public challenge as part of the competition track at the 2024 Autonomous Agents and Multi-agent Systems (AAMAS) conference. The top performing teams were also evaluated on unmanned surface vehicles playing a 3-vs-3 MCTF game in a physical marine environment. We summarize the techniques used by the top eight competition entries that featured control algorithms ranging from multi-agent deep reinforcement learning to heuristic approaches for path planning and search algorithms. Our analysis of the competition results reveals a trade off between winning versus safety, as a key factor differentiating teams' performance was their agents' handling of safety behaviors like collisions with other players. We conclude by highlighting the key research gaps in deploying multi-player game-like encounters in the real world scenarios.

Introduction

Multi-agent systems research has advanced significantly over the last three decades. A recent success story of multi-agent systems and machine learning has been the ability to develop game-playing agents that can play at levels comparable to human champion players in computer-based games such as StarCraft II, Chess, Texas hold-em, Quake-III Capture-the-Flag and Google Research Football 2 (Shao et al. 2019; Jaderberg et al. 2019; Vinyals et al. 2019; Liu et al. 2022). Most of these techniques have been developed within computer-based, gaming and simulation environments. In comparison, there has been limited research on multi-player games where physically embodied agents strategically make decisions and realize their actions in a real-life environment. Recently, DARPA's Alpha Dogfight trials (JHU-APL 2020) challenge involved designing an agent to autonomously control an aircraft in a 1-v-1 engagement against another human-controlled aircraft. Results were reported initially within a simulation environment (Pope et al. 2023), followed by deploying the techniques on physical aircrafts (Harper 2024). The Alpha Dogfight challenge has shown a promising direction towards fielding adversarial games in the real world. However, there are several open questions relevant to fielding adversarial games between physical agents that are worth investigating. Some of these include scaling up the game-playing techniques with number of players in each team, coordinating between teammates and competing with opponent players in real-time and ensuring safety constraints such as collision avoidance between physical agents while playing the game.

We propose to address some of these issues through a multi-player adversarial game called maritime capture-the-flag (MCTF). In this paper, we report our experiences with the MCTF game as part of the First Maritime Capture the Flag competition organized in 2024 (Kliem 2024b). The competition challenge problem was to develop game-playing algorithms for a 3-agent team that would play a 3-v-3 MCTF game against a previously unseen opponent team in a simulated game environment. Post-competition analysis of the different submission entries showed that agents that used AI planning and/or heuristics-based approaches leaning towards defensive tactics were most successful. Reinforce-

[*]These authors contributed equally.

ment learning-based approaches were successful in learning the high-reward task of the game but failed to learn low-penalty tasks like collision avoidance. The top two teams from the simulated game were deployed on unmanned surface vehicles (USVs) in a marine environment to play a 3-v-3 MCTF game. The hardware results highlight the challenges in the algorithms' performance under dynamic environment conditions such as wind and water currents and the need for research to bridge the sim-to-real gap.

MCTF Game Description and Competition

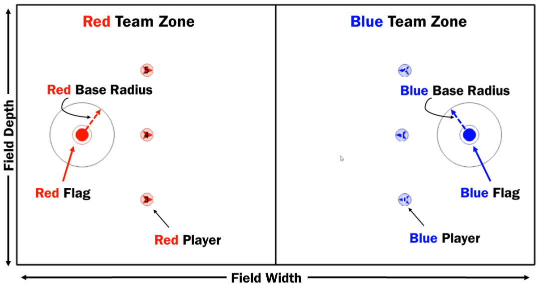

Figure 1: 2024 MCTF Pyquaticus Game Field

MCTF is based on the classic capture-the-flag game that is played between two opposing teams. The simulation and physical versions of the MCTF game environment was developed over a period of three years by research teams spanning three countries. Several incrementally difficult private competitions were held between the developer teams to analyze and refine the game rules and simulator features (Spencer et al. 2021; Beason et al. 2024). For the 2024 MCTF competition, we consider a 3-v-3 game, where a 3-agent team plays against a previously unknown 3-agent team. The game is played in a 80×40 m^2 rectangular, obstacle-free playing field, as shown in Figure 1. Players are initially not aware of the boundaries of the playing field. The playing field is divided into two halves, each half is referred to as a team's zone. Each team has a base within its zone that contains the team's flag. The objective of each team is two-fold: 1) Pick the flag from the opposing team's base (called flag grab) and return with it back to its own base (called flag capture), without getting tagged by a player of the opponent team during this process. 2) Tag a player of the opponent team if it enters into the player's zone. Each game lasts for a fixed duration of approximately 10 minutes. To prevent players from predominantly tagging each other instead of going for the main game objective of grabbing and capturing the flag, a post-tag *cool-down* period was implemented. The cool-down period was a time delay of 10 seconds which prevented the tagging player from immediately tagging again.

Table 1 gives the points for the self and opponent teams for MCTF game events. In addition to the points for in-game events, we also added a collision penalty in the competition scoring to enforce safe USV maneuvers when the game-playing algorithms are deployed on USVs in a physical marine environment. A collision occurs if any agent is within a range $r_{coll} = 2.5$ meters of another agent. The objective of each player is to learn a game-playing strategy that maximizes its team's primary score in the game given by: $Score_{pri} = n_{TFC} - n_{OFC} - n_{TCol}$, where, n_{TFC} and n_{OFC} are the number of flag captures by the self-team and the opposing team respectively, and n_{TCol} is the number of collisions by the team. Ties in the primary score were broken by using a secondary score given by: $Score_{sec} = n_{TFG} + n_{TT} - n_{OFG} - n_{OT}$, where n_{TFG} and n_{OFG} are the number of flag grabs by the self-team and the opposing team respectively, and, n_{TT} and n_{OT} are the number of times a player of the team tags and/or gets tagged by an opponent player.

Table 1: MCTF Game Events and Scores. *: For competition scoring, flag grab points were only used for tie-breaking.

Event	*Self Pts*	*Opp Pts*
Flag Capture	$+1$	-1
Flag Grab*	$+1$	-1
Collision	-0.025	-0.025

Game Simulation Environment. The MCTF game is modeled as a perfect information, deterministic, zero-sum game in the simulation environment. For training and evaluating agents we used the Pyquaticus marine robotics CTF simulator (Serlin et al. 2024). Pyquaticus is built on top of the PettingZoo environment (Terry et al. 2021) that supports training and evaluating agents via multi-agent reinforcement learning. Pyquaticus supports implementations of different deep RL algorithms including Proximal Policy Optimization (PPO) through the Ray RLlib decentralized training library (Schulman et al. 2017; Liang et al. 2018). Pyquaticus also provides a bridge to the MOOS-IVP Aquaticus testbed that enables directly deploying Pyquatics agents onto USVs playing the MCTF game in a physical, marine environment (Novitzky et al. 2019).

Competition Evaluation. The MCTF competition was hosted on Codalab (Kliem 2024a). The simulation competition was organized in two rounds:

- *Qualifying Round:* Each team played against three hardness levels - easy, medium and hidden, of movement strategies of a 3-player opponent team. Each player in a team using the easy strategy would follow a fixed, pre-determined path - a defending player would circle around in front of the team's base. Meanwhile, an attacking player would follow a circular trajectory between its base, the opponent's flag, and back. In a team using the medium strategy, each player used a potential fields-based reactive strategy on top of the easy strategy. In an attacker role, a player would get attracted to its opponent's flag and repelled by an opponent player; in a defender role a player would get attracted to any opponent player that entered its zone. The code for both easy and medium opponent team strategies was made available to participants to play against, and, refine and fine-tune their game-playing strategies. The hidden strategy was not made available to participants. It consisted of three games in succession- first against a team of 3 medium defenders, then against a team of 3 medium attackers, and finally, against a team of two attackers and one defender all set at the medium difficulty.

- *Final Round:* The top-performing teams from the qualifying round advanced to the final round where each team played against every other team in a round-robin tournament. Scores from successive games of a team were added cumulatively to determine the rank-ordered performance of the team on the competition leaderboard.

Sample code for training a team of 2 agents via PPO (Schulman et al. 2017) inside Pyquaticus to play a 2-v-2 MCTF game was provided on the competition Website (Kliem 2024a) to reduce the entry barrier for potential participants. The challenge was run over a period of 12 weeks and received 15 valid competition entries from teams across four continents for the qualifying round.

MCTF Competition Algorithms

There were two main categories of algorithms submitted to the MCTF competition: heuristics-based and machine learning (ML)-based algorithms. Although some algorithms used rather simple heuristics, we include these to highlight limitations of some of the ML-based approaches. For legibility, we denote each team by the last name of the team lead, followed by the team's abbreviated affiliation. Table 2 summarizes each team's approach and reported development time, including training time for their MARL algorithm, wherever applicable.

Table 2: MCTF Competition Teams, Algos and Dev Time

Team Name	*Algo Type*	*Dev Hrs*
Lucas-QMU	Stochastic Search + heur. plan opt.	~ 80
Ohto-Indep[1]	Naive heuristic	-
Alamer-EJUST[1]	Naive heuristic	-
Meo-Indep[1]	Multi-robot task alloc + path de-conflict	-
Chao-NIWC-PAC	PDDL planner	~ 640
Richley-NIWC-ATL	Deep MARL	~ 160
Crowley-BU-LL	Hier. learning + imitation learning	~ 160
Jin-Indep[1]	RL (code not shared)	-
Leprell-USMA	Deep MARL	~ 85

[1] Participants who did not respond to request for approach description.

Lucas-QMU. The Lucas-QMU team's main approach was a stochastic search technique that implemented a multi-agent version of the Rolling Horizon Evolution (RHE) algorithm (Gaina et al. 2021) called Multi-Unit RHE (MURHE). RHE is a stochastic search algorithm in the same family as Monte Carlo Tree Search (MCTS) algorithms. Whereas MCTS uses selective sampling of next states to expand the search tree from a state, RHE starts with an initial population of roll-outs or trajectories that comprises state-action pairs from the start up to the end of the game or the limits of the planning horizon. It then uses evolution to improve the trajectories selectively. The fitness function inside the evolution ranks trajectories based on a multi-objective cost function that balances expected rewards and possibility of collisions in the trajectory. The multi-agent version of RHE, MURHE, evolves trajectories one to a time for each player in the self- and opponent teams. The trajectories of all players in the team are then updated to maximize the expected reward for the team. Safety was considered by including the penalty for collisions with teammates into the multi-objective cost function (fitness function) of the trajectories in the MURHE algorithm. The final algorithm of this team refined the plan or sequence of moves generated by MURHE using heuristics based on the game scoring rules.

Ohto-Indep., Alamer-EJUST. One of the findings reported by several participating teams was that it was easier to win by defending to ward off attacks rather than to attack to try to grab and capture the flag. This was due to the game rules that penalized agent collisions and getting tagged by an opponent. In addition, if a player did a tag closer to its home base it would need a shorter time after its post-tag cool-down period to get back to defending its flag. These effects culminated in a naive strategy, where all the players of a team just remained stationary near their base to defend against attacks. The Ohto-Indep and Alamer-EJUST submission entries exploited this aspect of the game rules in their agent design. The algorithms performing well against the easy and medium opponent teams in the qualifying round. Although this naive strategy exposed a limitation of the easy and medium opponent team strategies used in our evaluations, it also highlighted the need of more complex opponent strategies to evaluate against and more diversity in the game such as spawning self and opponent team players at different locations at the start of the game, and having higher and/ or gradually increasing post-tag cool-down periods.

Meo-Indep. The Meo team developed a defensive strategy called *Man-to-Man* defense that allocated each agent in the team to intercept an attacker agent of the opponent team using an approach inspired by multi-robot task allocation. Man-to-Man proceeds in two phases: in the first phase, it calculates the intercept points between every player (defender) and every other opponent player along with the distances between each defender and its corresponding set of interception points. It then selects the pairing between a defender and an interception point that minimizes the sum of distances between defenders and interception points. The action for each defender is set to the highest speed and bearing to reach its paired interception point. When a defender reaches its interception point, it stops (speed set to zero). While defenders are traveling to their paired interception points, potential intersections between their trajectories are calculated to identify possible collisions with other teammates. If a potential collision is detected between defenders, the headings of one of them is set to the opposite of its selected direction to avoid collision. This approach too minimizes collisions between teammates but does not avoid collisions with opponent team players.

Richley-NIWC-ATL. This team incrementally developed different MARL algorithms for playing MCTF. Their first attempt was an n-player version of the COCO-Q algorithm (Sodomka et al. 2013). COCO-Q enables agents to negotiate with each other via side payments of rewards. This attempt did show a bit of cooperation between agents in the same team, but did not converge to a strong policy. The second attempt used multi-agent PPO with league training (Vinyals et al. 2019). This resulted in a good strategy that was heavily focused on grabbing and returning with the enemy flag, some instances resulted in highly defensive policies. The third attempt used QMIX that resulted in a nicely balanced strategy between various roles such as attacker, defender, and mid-fielder (Rashid et al. 2018). Agents would dynamically switch between these roles depending on their field positioning. A weakness of this technique was that even though it did well in grabbing the opponent's flag, it had not fully learned how to return with the flag to its base for a flag capture. The final submission used a mixture of QMIX and the league-based PPO. An agent used QMIX as its main game-playing strategy, but as soon as it grabbed the opponent's flag, it would switch to the league-based PPO model to secure a flag capture.

Crowley-BU-LL. This team divided the overall task of playing the game into a set of behavioral primitives. The behavioral primitives that were considered were: go to opponent flag region ("attack"), flank opponent flag region ("flank"), avoid opponents ("avoid"), retreat to own zone ("retreat"), guard against opponents grabbing and capturing flag ("guard"), tag opponent closest to flag ("tag"), and do nothing ("no-op"). Each behavioral primitive was represented as a time extended policy or option using a hierarchical learning-based mixture-of-options framework (Henderson et al. 2018). Human players were then assigned to play the game to determine when to switch between the options. Each human player was randomly assigned to play against the easy and medium defender and attacker agent teams. A total of 47 demonstrations of human game-play was collected. This dataset was then used for training a shared policy-over-options via imitation learning. Two key takeaways from this approach were that learning an options-level demonstrator policy allowed for good generalization against different opponent team strategies from few human demonstrations of game-play, and, using pre-defined intra-option policies produced more predictable and interpretable behaviors.

Chao-NIWC-PAC. This team explored multiple methods including mixed discrete-continuous planning (PDDL+), discrete planning (PDDL), RL, and centralized and decentralized planning methods for multiple agents. The RL approach was dropped in favor of a PDDL+ planner called Nyx (Piotrowski and Perez 2024) which ultimately was more reliable and converged to a policy with both attacking and defending behaviors. To solve the MCTF game as a planning problem, the following modifications were made to the game environment: the RL-based environment was converted to a planning environment by translating relative bearing and distance measurements between players to absolute values. The action space was abstracted for higher-level planning and environment action outcomes were mapped to corresponding planning actions. The environment was modeled as a non-temporal, 16×8 grid where agents could move in the four cardinal directions between grid cells. Exogenous events included undesired collisions and desired flag grabs and captures. Adversaries were assumed to be static, with periodic re-planning. PDDL+ problem instances were auto-generated from observations, and plans were sequentially executed in the environment. The goal given to the planning-based agents was to capture the flag or make it back just inside the team's zone after grabbing the flag.

Leprell-USMA. The primary approach in this team's competition entry was attempting to reduce the uncertainty experienced by the agents during training. Dense reward functions resulted in poor convergence of deep RL policies. To overcome this, sparser rewards were used while only assigning rewards for the time-step in which captures or grabs occurred, resulting in revised scoring scheme of $+1/-1$ (captures), and $+0.5/-0.5$ (grabs). The agents used the Ray RLlib Proximal Policy Optimization (PPO) algorithm with the default hyper-parameters (Liang et al. 2018) during training. Each agent was trained in a decentralized manner for 17000 episodes, on a system with 18 CPU cores, and an RTX4090 GPU. The safety aspect (collision avoidance) of the MCTF game was not considered. Despite this, the DRL approach remained consistent against the previously unseen opponents during the final, round robin phase.

Results and Observations

Table 3: MCTF final round team rankings including collision penalties.

Rank	*Team Name*	*Score*	*No. Collisions*
1	Lucas-QMU	-1.0	4
2	Ohto-Indep	-3.0	12
2	Alamer-EJUST	-3.0	12
3	Meo-Indep	-4.25	17
4	Chao-NIWC-PAC	-6.75	11
5	Richley-NIWC-ATL	-10.0	109
6	Jin-Indep	-10.25	226
7	Crowley-BU-LL	-15.5	183
8	Leprell-USMA	-49.5	153

We report the team scores from the final round of the MCTF competition (in simulation) in Table 3. The final round had 72 round robin games featuring the top 9 teams from the qualifying round[2]. Each of the 8 teams played every other team twice, once on each side of the playing field. The opponent team's strategies were not revealed to the team before the game. Team scores were determined using the $Score_{pri}$ metric (discussed in Section 2) that rewarded flag captures while penalizing collisions. The top scoring algorithm prioritized defensive behaviors via a stochastic search

[2] The qualifying round results are available at (Kliem 2024a). They are not analyzed further here as their main purpose was for participants to refine and fine-tune their algorithms.

algorithm and integrated collision avoidance into its learning objective function. Submissions ranked $2-3$ implemented defensive-only behaviors via heuristics without using machine learning. These algorithms did not explicitly avoid collisions with teammates. Rather they leveraged the game scoring rule structure and kept their agents stationary just outside their flag base or at the closest point for intercepting an attacker. The 4-th ranked team used planning, again with collision avoidance built into the objective function. Notably, prioritizing defense[3] and avoiding collisions turned out be the key factors in the success of the top 5 teams. Also, none of these teams had used an ML- or RL-only solution. Teams ranked $6-8$ used MARL-based solutions. They mainly lost points due to large number of collisions indicating that the RL algorithm was not able to learn to avoid collisions when faced with a previously unseen opponent team (that it was not trained against).

Table 4: MCTF final round team rankings without collision penalties.

Rank	*Team Name*	*Number of Captures*
1	Jin-Indep	31
2	Crowley-BU-LL	19
3	Richley-NIWC-ATL	15
4	Chao-NIWC-PAC	4
5	Leprell-USMA	0 Captures 5 Grabs
6	All Other Teams	0 Captures 0 Grabs

As handling collision avoidance was a problem for most of the submitted algorithms, we wanted to evaluate how well the algorithms had learned the main objective of the MCTF game of capturing the flag. For this, we re-ran all the games of the round-robin phase for a second time while turning off the collision penalty in the $Score_{pri}$ metric. The scores for these evaluations are reported in Table 4. Interestingly, we observe a complete reversal in the team's rankings when collision penalties were ignored. MARL-based algorithms performed much superior, making multiple flag captures while the non-MARL algorithms, owing to their defense-focused behaviors ended up with few or zero captures most of the times. This finding points in the direction that safety constraints like collision avoidance might be better off being implemented as a separate component or control mechanism instead of integrating it inside the same RL objective function as the game's main task.

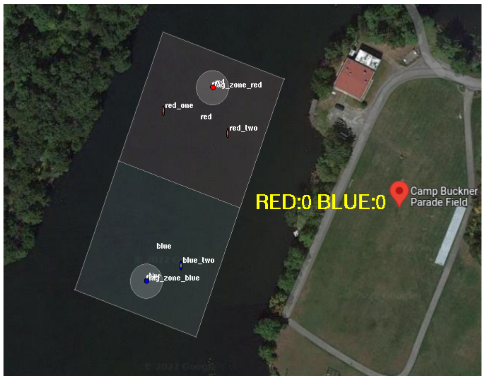

Figure 2: Marine environment used for evaluation of MCTF competition algorithms deployed on USVs, playing 3-v-3 MCTF game.

Competition in Physical Environment. The top team from each category during the final simulation round, Lucas-QMU (most captures with collision penalty) and Jin-Indep (most captures without collision penalty), were deployed against each other to play a 3-v-3 MCTF game in a physical, marine environment at Lake Popolopen, West Point, NY[4]. The autonomous platform for each player was a SeaRobotics Surveyor USV. The playing field size and flag placements were kept as close possible to the simulation environment. A snapshot of an on-water MCTF game is shown in Figure 2. Before deploying the algorithms from the simulation, a few adjustments were made to the control mechanisms on the USVs to enable turning at higher speeds and reaching the maximum speed quickly. The collision avoidance was implemented manually for safety reasons - when the personnel managing the on-water game saw that two USVs were getting close, within $\sim$ 2.5 m of each other, they manually overrode the control of each USV to slow them down and turn away from each other. When the USVs were farther apart, the manual control was relinquished to the autonomous control algorithm. The game was run for $\sim$ 10 minutes. Differing from the simulation results where Lucas-QMU's solution had been successfully able to defend the flag and achieve highest score, the dynamic conditions in the marine environment like wind and water currents degraded their algorithm's performance in defense. This allowed Jin-Indep's ML-based approach to capture the flag successfully one time and grab Lucas-QMU's team flag a total of three times. Similarly differing from simulation was the number of collisions encountered where Jin's team required human collision avoidance twice, and Lucas-QMU's approach required four human collision avoidance interactions. This difference highlights the significant divide while transitioning algorithms between simulation and real world emphasizing the need for future research in this direction. As the on-water competition was not held simultaneously with the simulation competition, further analysis and explanation of the result was planned to be done *post-hoc*, for the next edition of the competition.

Conclusions and Future Competitions

We reported the different algorithms used in the first MCTF competition and their performances in a round robin matchup between the qualifying submitted entries. While deep RL-based techniques have demonstrated significant successes in computer-based real-time strategy games, we saw that for the MCTF game, adding a secondary objec-

[3] In 44% of all games in the final round, a team never attempted to capture the opponent team's flag.

[4] The on-water competition took place 3 months after the simulation competition.

tive like a safety constraint degraded the score of MARL-only approaches. Collision avoidance in an adversarial game proved to be particularly difficult as players sometimes needed to proactively avoid collision with another oncoming player that does not implement collision avoidance. Intuitively, we felt that most solutions, especially the ones using MARL lost points on collisions because they had trained against players that also used collision avoidance. Transitioning some of the competition algorithms to an on-water 3-v-3 MCTF game with USVs highlighted the limitations of transitioning techniques developed in simulation-only on to physical systems.

The MCTF simulation environment and competition provide an accessible and extensible simulation environment for advancing research on multi-player, real-time, adversarial games. The first competition provided competition organizers and participants with valuable insights for future competitions. We are rolling out the next competition with game rule changes to make the game more interesting and challenging. Some updates to the game include increasing the post-tag cool-down period to incentivize attack strategies and to discourage loitering or standing still near the home flag base, and adding improved visualization capabilities to the simulation environment for analyzing and learning from past game-plays. Future editions of the game will include harder problems like obstacles in the game field, noisy sensors on the players' platforms with limited perception range, heterogeneous autonomous platforms (e.g., aerial and marine robots) and deceptive play by opponent team players. The MCTF organizers envisage that lessons learned through future iterations of simulated and physical MCTF competitions will help us understand the challenges and issues of deploying multi-player adversarial games in the real world and bridge the sim-to-real gap.

References

Beason, J.; Novitzky, M.; Kliem, J.; Errico, T.; Serlin, Z.; Becker, K.; Paine, T.; Benjamin, M.; Dasgupta, P.; Crowley, P.; O'Donnell, C.; and James, J. 2024. Evaluating Collaborative Autonomy in Opposed Environments using Maritime Capture-the-Flag Competitions. arXiv:2404.17038.

Gaina, R. D.; Devlin, S.; Lucas, S. M.; and Perez-Liebana, D. 2021. Rolling horizon evolutionary algorithms for general video game playing. *IEEE Transactions on Games*, 14(2): 232–242.

Harper, J. 2024. Pentagon takes AI dogfighting to next level in real-world flight tests against human F-16 pilot. https://defensescoop.com/2024/04/17/darpa-ace-ai-dogfighting-flight-tests-f16/. [Online; accessed 24-November-2024].

Henderson, P.; Chang, W.; Bacon, P.; Meger, D.; Pineau, J.; and Precup, D. 2018. OptionGAN: Learning Joint Reward-Policy Options Using Generative Adversarial Inverse Reinforcement Learning. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 3199–3206. AAAI Press.

Jaderberg, M.; Czarnecki, W. M.; Dunning, I.; Marris, L.; Lever, G.; Castaneda, A. G.; Beattie, C.; Rabinowitz, N. C.; Morcos, A. S.; Ruderman, A.; et al. 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443): 859–865.

JHU-APL. 2020. Alpha Dogfight Trials. https://www.jhuapl.edu/work/projects-and-missions/alphadogfight-trials. [Online; accessed 24-November-2024].

Kliem, J. 2024a. Codalab Maritime Capture the Flag Competition. https://codalab.lisn.upsaclay.fr/competitions/17254#learn_the_details-overview. [Online; accessed 24-November-2024].

Kliem, J. 2024b. First Maritime Capture the Flag Competition. https://sites.google.com/view/mctf2024/. [Online; accessed 24-November-2024].

Liang, E.; Liaw, R.; Moritz, P.; Nishihara, R.; Fox, R.; Goldberg, K.; Gonzalez, J. E.; Jordan, M. I.; and Stoica, I. 2018. RLlib: Abstractions for Distributed Reinforcement Learning. arXiv:1712.09381.

Liu, B.; Pu, Z.; Zhang, T.; Wang, H.; Yi, J.; and Mi, J. 2022. Learning to play football from sports domain perspective: A knowledge-embedded deep reinforcement learning framework. *IEEE Transactions on Games*, 15(4): 648–657.

Novitzky, M.; Robinette, P.; Benjamin, M. R.; Fitzgerald, C.; and Schmidt, H. 2019. Aquaticus: Publicly Available Datasets from a Marine Human-Robot Teaming Testbed. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 392–400.

Piotrowski, W.; and Perez, A. 2024. Real-World Planning with PDDL+ and Beyond. arXiv:2402.11901.

Pope, A. P.; Ide, J. S.; Mićović, D.; Diaz, H.; Twedt, J. C.; Alcedo, K.; Walker, T. T.; Rosenbluth, D.; Ritholtz, L.; and Javorsek, D. 2023. Hierarchical Reinforcement Learning for Air Combat at DARPA's AlphaDogfight Trials. *IEEE Transactions on Artificial Intelligence*, 4(6): 1371–1385.

Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 4295–4304. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Serlin, Z.; Mann, M.; Crowley, P.; Gonsalves, T.; and Kliem, J. 2024. Pyquaticus Capture the Flag Gymnasium. https://github.com/mit-ll-trusted-autonomy/pyquaticus.

Shao, K.; Tang, Z.; Zhu, Y.; Li, N.; and Zhao, D. 2019. A Survey of Deep Reinforcement Learning in Video Games.

Sodomka, E.; Hilliard, E. M.; Littman, M. L.; and Greenwald, A. 2013. COCO-Q: learning in stochastic games with side payments. In *Proc. 30th Intl. Conf. Machine Learning*, volume 28, III–1471–III–1479. JMLR.org.

Spencer, P.; Dasgupta, P.; McCarrick, M.; Novitzky, M.; Hubczenko, D.; Redfield, S.; James, J.; Jeffery, A.; and Mittu, R. 2021. Opposed Artificial Intelligence: Developing Robustness to Adversarial Attacks in Attacker-Defender Games via AI-based Strategic Game-Playing. In *Proc. of NATO STO MP-IST-190: AI, ML & Big Data for Hybrid Military Operations (AI4HMO)*.

Terry, J.; Black, B.; Grammel, N.; Jayakumar, M.; Hari, A.; Sullivan, R.; Santos, L. S.; Dieffendahl, C.; Horsch, C.; Perez-Vicente, R.; et al. 2021. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 15032–15043.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.