2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #39981

A SwarmAI Testbed for Workforce Development and Collaborative, Interdisciplinary Research

Martha Cervantes, Johns Hopkins University

Martha Cervantes is a Mechanical Engineer at the Johns Hopkins University Applied Physics Laboratory where she works in mechanical design and integration of robotic systems. Additionally, Martha is the project manger of the CIRCUIT Program at JHU/APL, which connects and mentors students from trailblazing backgrounds to STEM careers through science and engineering projects. Martha received her B.S. in Mechanical Engineering from Johns Hopkins University, and she is currently pursuing a M.S. in Mechanical Engineering at the Johns Hopkins University.

Raphael Norman-Tenazas
Adam Goertz
Mr. Erik C. Johnson, University of Illinois, Urbana-Champaign
William Roberts Gray-Roncal

BOLT: A SwarmAI Testbed for workforce development and collaborative, interdisciplinary research

Abstract

Educating the next generation of AI researchers requires methods which teach the software tools, theoretical concepts, and domain knowledge specific to the field. To help develop these key skills, we focus particularly on the area of Swarm AI, which, in general, covers the autonomous operation of a large number of agents in a single environment. Applications of this field are numerous, including autonomous navigation, defense, robotics, logistics, and search/rescue. Despite the potential for impact in key problem domains and interdisciplinary nature, Swarm AI platforms are not generally used to develop competency in AI education. While some courses and tracks target AI in general, there are few tools to help engage learners in the specific area of Swarm AI. This is partly due to needing strong skills in the intersection of reinforcement learning, software development, and robotics, as well as a framework computing capability on which to test and evaluate. We propose Battle Optimized Laser Tag (BOLT), a Unity-built simulator that enables learners to develop Swarm AI algorithms in the context of a laser tag game between small mobile robots. The simulator is built to be familiar to a video game environment to maximize engagement, while the included documentation for the environment is designed as a gentle introduction into the topic of controlling swarms. Easily modified configurations enable educators to specify tasks and develop curricula to further challenge learners. Furthermore, the simulator is OS-agnostic, simple to install and uses Python to interface with the agents. This allows learners to implement their solutions on their own computers in a programming language common to other AI courses. Finally, extensions of the platform, leveraging the Python interface, can be deployed to physical robots or be used as a testbed for other AI domains beyond robotics including Human-Machine Teaming and Cybersecurity.

Introduction

Developing a diverse Artificial Intelligence workforce is a critical national need[1]. This is recognized by government funding agencies[2], and there is a focus on increasing participation of under-represented groups[3] and addressing the gender gap[4]. A particular interdisciplinary space involving multiple engineering disciplines, mathematics, and computer science is Swarm AI-machine learning techniques to control groups of robots (called swarms) to accomplish a task. This involves skills such as mechatronics, mechanical engineering, sensors and signal processing, wireless communications, computer networking, machine learning, control theory, path planning and optimization, and more. Moreover, Swarm AI is an active area of research, including development of reinforcement learning techniques for control[5] and overcoming performance loss during the transfer from simulated to real environments[6]. Swarm AI has varied applications, from autonomous vehicles, drones[7], logistics[8], entertainment, national security, disaster response, to many more critical areas. Swarm AI is an ideal area for learners to develop critical skills in emerging AI technologies and real-world application spaces, but has often remained inaccessible to new learners.

Several projects have developed swarm testbeds, including for educational contexts. Examples include the Robotarium[9] for remote swarm algorithm development, as well as platforms for industrial swarm robotics[10] and human robot teaming[11]. Due to their distributed, wireless nature, swarms have also been used as an internet of things testbed[12]. Several low-cost ground robotics swarms have also been proposed, which allow for scalable testing[13-14]. Of these platforms, several examples, such as the Pheeno, Spiderino, and Pi-swarm, have been used in educational contexts to teach swarm robotics, often in a K-12 context[15-18].

However, education tools and programs around AI and Swarm AI do not generally have a standard curriculum, as many different traditional fields are needed to come together to learn about and develop AI at the level of professional practitioners. In Swarm AI in particular, aspects of robotics, engineering, and computer science are often seen as the barriers of entry for the field. In this paper, BOLT (Battle Optimized Laser Tag) is proposed as a platform to accelerate Swarm AI education by removing some of the barriers in entering the field. Compared to previous platforms, this platform adds several unique features: a Swarm AI curriculum, simulation and real-world environment, and tasks relevant to reinforcement learning algorithm development. Users of the platform are able to interact with swarm agents in a Unity simulation in the context of a laser tag game through an intuitive Python API. The users, often students but even researchers, do not need to know the underlying lower level controls and communications systems in order to command and control the agents in the swarm. This platform offers a pathway for learners to rapidly build the interdisciplinary skills which will be required for an AI-enabled workforce.

Methods

BOLT is a system for swarm robotics controller development. This allows the development of skills such as AI development, networking and robot operating system development, and real-time systems development. In the environment, robots (currently Turtlebots https://www.turtlebot.com/) are divided into two teams and assigned a task in the form of a laser tag scenario. Custom laser hardware and receivers, compatible with the Turtlebot development boards, have been developed to implement this physically. Learners develop scripts to implement different kinds of swarm controllers, which could range from simple path planning scripts and logical rules to complex reinforcement learning algorithms. The BOLT platform consists of two major components: a Python-based API and a Unity-based standalone application.

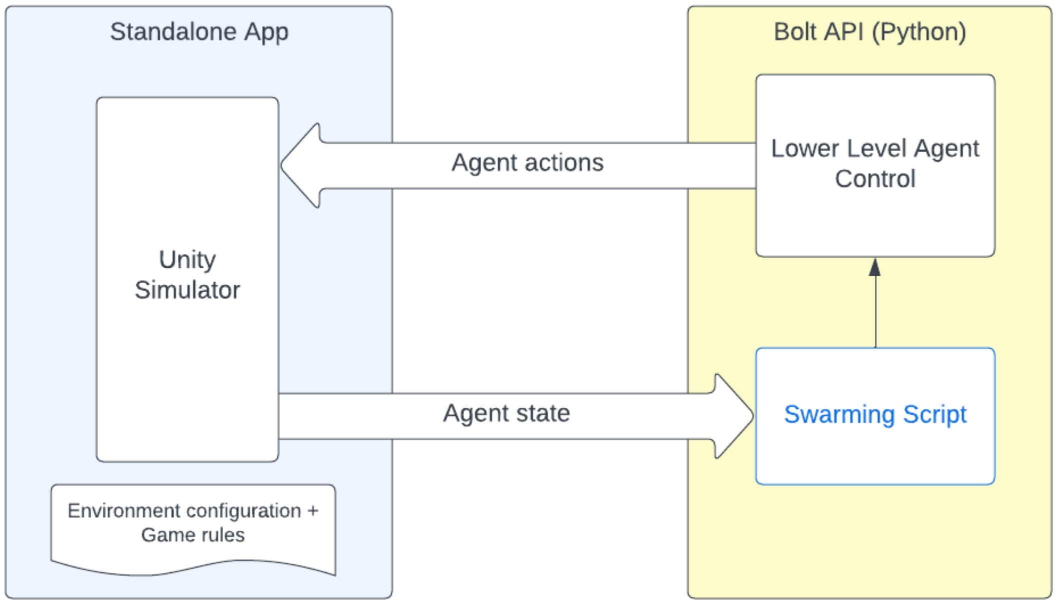


Figure 1. BOLT System Architecture. Users develop python code in the Swarming Script, which is abstracted from the rest of the testbed.

Software Architecture:

A focus of this effort was to minimize setup time for the user (student or expert) in order to allow maximum time to test and evaluate swarming algorithms. Students are often time limited when engaging with similar platforms (for example in a summer program or in a course), so it is necessary to create a system where the majority of student time is focused on development of swarm techniques, robotic controllers, and AI algorithms. This involved abstracting both the simulation environment and lower lever control of the agents through a standalone application and intuitive Python-based API.

One common barrier to entry for AI students is the installation of the various libraries, SDKs and development platforms, especially if learners are not experienced in software development. While the mastery of software environments is certainly a necessary skill for every AI developer, the frustration when facing issues getting a package working due to environment setup may cause disengagement early on, affecting the educational trajectory of the learner. In addition, tutorials are usually simple scripts using text-based output, which, while showing basic functionality, rarely demonstrate the benefits of the package, and may make the student question the necessity of coding. Therefore, we designed our simulator to be as simple to install as possible so that students may see their first algorithms take shape immediately and with impact on the robotic system.

Other robot simulators such as Gazebo[19] typically rely on Linux or require a complicated install process using an emulator or virtual machine, or reach limitations when running more advanced simulations including multiple agents. We designed the BOLT simulator to be a Unity app (https://unity.com/) that is OS agnostic; that is, it is built to run natively on Linux distributions, Mac OSX, and Windows. This enables students to run the simulator on any computer, regardless of OS, and reduces overall time spent debugging the environmental setup.

While it can be exposed to the learner, the configuration of the simulation is meant to be set by the curriculum developers. Our configuration system allows the developer to customize the teams (such as their names, colors, and number of agents), and other parameters about the environment such as adding obstacles. These configurations are meant to be easily written in the human-readable markup language YAML, but could also be auto-generated.

Use Cases

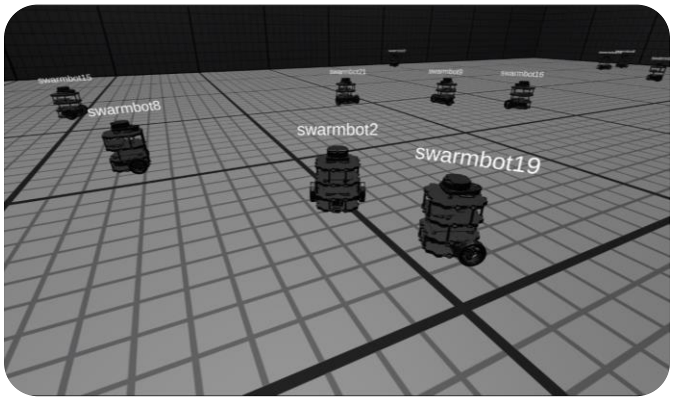
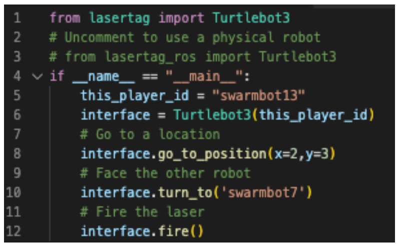


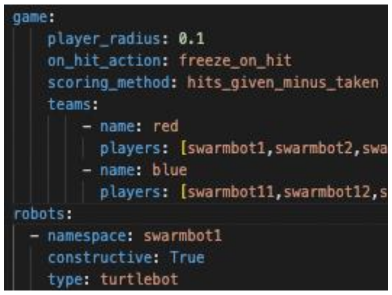
Fig 2. The BOLT Unity simulation environment. From the perspective of the student developer, this simulation framework is identical to the physical environment.

Once students have installed the simulator, they can immediately begin the process of developing algorithms to control their agents and teams. To do so, we created an abstracted Python API that allows different kinds of high level agent actions, such as "go_to_position" and "set_body_velocity", while managing the lower level, robot-specific commands separately, such as sending individual wheel or joint commands. This enables the development of curricula that begin with simple tasks such as "command an agent to go to a location and shoot at a target" and progress to more complex tasks such as "design an algorithm deployed on the entire team to maximize points" within limited timeframes. Having different control options also allows for testing and evaluation of a wider variety of swarming algorithms and strategies on the same platform.



```python
from lasertag import Turtlebot3
# Uncomment to use a physical robot
# from lasertag_ros import Turtlebot3
if __name__ == "__main__":
    this_player_id = "swarmbot13"
    interface = Turtlebot3(this_player_id)
    # Go to a location
    interface.go_to_position(x=2,y=3)
    # Face the other robot
    interface.turn_to('swarmbot7')
    # Fire the laser
    interface.fire()
```

Fig 3. Example code block for control of swarm agent. Changing the import library enables rapid transition between simulated and real systems.



```yaml
game:
  player_radius: 0.1
  on_hit_action: freeze_on_hit
  scoring_method: hits_given_minus_taken
  teams:
    - name: red
      players: [swarmbot1,swarmbot2,swa
    - name: blue
      players: [swarmbot11,swarmbot12,s
robots:
  - namespace: swarmbot1
    constructive: True
    type: turtlebot
```

Fig 4. Example task design, defining two teams and parameters for a laser tag game.

We designed the API to keep both students and seasoned AI experts in mind, such that the platform can be used for not only student development, but the testing and evaluation of cutting edge swarm AI, for example using multi-agent reinforcement learning techniques.

In order to better engage students, we gamify our simulation in a laser-tag scenario, in which agents compete solo or in teams by firing lasers at each other. Not only does this make our simulator more "fun", but it also presents a relevant challenge to the national security community[20]. Our game has customizable set of rules to allow for varied and increasing challenges:

1. What action to do when getting hit by a laser.

2. How the points are calculated based on who is firing and who is hit

3. Game end conditions

The rule system is tightly integrated with our configuration system, which is a YAML file that is loaded at the beginning of the game. For each rule, we pre-define strings for several variations. For example, for an "on_hit_action", we define the self-explanatory strings "return_home", "freeze_for_time", and "stop_until_game_end". Each of these strings are tied to Python functions that fulfill the action using the API. Therefore, it is also possible to define more rules by extending our Python API.

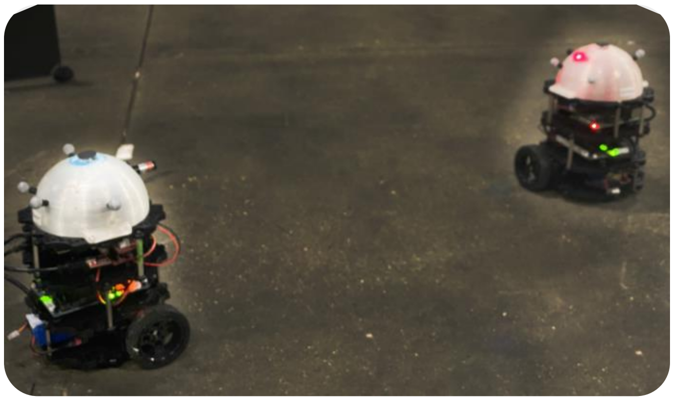


Fig 3. Physical Turtlebots for swarm testing, with mounted laser tag receivers.

To increase accessibility for new learners, the API was developed with good documentation practices. We created our documentation with an educational focus, keeping it simple and providing sample code, while also being detailed enough that experienced researchers can leverage it as a fully functional and customizable robotics testbed.

Discussion

Although it is not the main focus of the presented architecture, we also designed the software to be able to be easily deployed on real robots. To do this, we have integrated the API with ROS (Robot Operating System), a set of tools commonly used to enable robot software applications. By using the same message format as typical ROS messages, users will be able to use the same scripts and commands in either simulation or physical platforms. While many swarm robotics platforms have been developed, this unique and abstracted game environment allows for rapid and intuitive acquisition of skills related to swarm controllers and machine learning.

We hope to expand the simulator and accompanying API in a number of ways, including adding more robot models, developing more complex actions and scenarios, and creating additional environments. This would allow student and other researchers to explore additional aspects of swarming robotics such as swarm-human teaming, resilience of distributed systems, and emergent behaviors.

Acknowledgements

The authors would like to thank the CIRCUIT Program for their support, as well as Adam Polevoy for his technical contributions.

References

1. Zwetsloot, Remco, Roxanne Heston, and Zachary Arnold. "Strengthening the US AI workforce." *Center for Security and Emerging Technology, Georgetown University* (2019).
2. James, Sylvia M., and Susan R. Singer. "From the NSF: The National Science Foundation's investments in broadening participation in science, technology, engineering, and mathematics education through research and capacity building." *CBE—Life Sciences Education* 15, no. 3 (2016): fe7.

3. Freire, Ana, Lorenzo Porcaro, and Emilia Gómez. "Measuring diversity of artificial intelligence conferences." In *Artificial Intelligence Diversity, Belonging, Equity, and Inclusion*, pp. 39-50. PMLR, 2021.
4. Stathoulopoulos, Konstantinos, and Juan C. Mateos-Garcia. "Gender diversity in AI research." *Available at SSRN 3428240* (2019).
5. Hüttenrauch, Maximilian, Sosic Adrian, and Gerhard Neumann. "Deep reinforcement learning for swarm systems." *Journal of Machine Learning Research* 20, no. 54 (2019): 1-31.
6. Zhao, Wenshuai, Jorge Peña Queralta, and Tomi Westerlund. "Sim-to-real transfer in deep reinforcement learning for robotics: a survey." In *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737-744. IEEE, 2020.
7. Chung, Soon-Jo, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. "A survey on aerial swarm robotics." *IEEE Transactions on Robotics* 34, no. 4 (2018): 837-855.
8. Wen, Jinming, Li He, and Fumin Zhu. "Swarm robotics control and communications: Imminent challenges for next generation smart logistics." *IEEE Communications Magazine* 56, no. 7 (2018): 102-107.
9. Pickem, Daniel, Paul Glotfelter, Li Wang, Mark Mote, Aaron Ames, Eric Feron, and Magnus Egerstedt. "The robotarium: A remotely accessible swarm robotics research testbed." In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1699-1706. IEEE, 2017.
10. Jones, Simon, Emma Milner, Mahesh Sooriyabandara, and Sabine Hauert. "DOTS: An open testbed for industrial swarm robotic solutions." *arXiv preprint arXiv:2203.13809* (2022).
11. Dhanaraj, Neel, Nathan Hewitt, Casey Edmonds-Estes, Rachel Jarman, Jeongwoo Seo, Henry Gunner, Alexandra Hatfield et al. "Adaptable platform for interactive swarm robotics (apis): a human-swarm interaction research testbed." In *2019 19th International Conference on Advanced Robotics (ICAR)*, pp. 720-726. IEEE, 2019.
12. Farnham, T., Jones, S., Aijaz, A., Jin, Y., Mavromatis, I., Raza, U., ... & Sooriyabandara, M. (2021, January). Umbrella collaborative robotics testbed and iot platform. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)* (pp. 1-7). IEEE.
13. Ospina, Nestor I., Eduardo Mojica-Nava, Luis G. Jaimes, and Juan M. Calderón. "Argrohbots: An affordable and replicable ground homogeneous robot swarm testbed." *IFAC-PapersOnLine* 54, no. 13 (2021): 256-261.
14. Rubenstein, Michael, Christian Ahler, and Radhika Nagpal. "Kilobot: A low cost scalable robot system for collective behaviors." In *2012 IEEE international conference on robotics and automation*, pp. 3293-3298. IEEE, 2012.
15. Couceiro, Micael S., Carlos M. Figueiredo, J. Miguel A. Luz, Nuno MF Ferreira, and Rui P. Rocha. "A Low-Cost Educational Platform for Swarm Robotics." *International Journal of Robots, Education & Art* 2, no. 1 (2012).
16. Hilder, James, Rebecca Naylor, Artjoms Rizihs, Daniel Franks, and Jon Timmis. "The pi swarm: A low-cost platform for swarm robotics research and education." In *Advances in Autonomous Robotics Systems: 15th Annual Conference, TAROS 2014, Birmingham, UK, September 1-3, 2014. Proceedings 15*, pp. 151-162. Springer International Publishing, 2014.
17. Wilson, Sean, Ruben Gameros, Michael Sheely, Matthew Lin, Kathryn Dover, Robert Gevorkyan, Matt Haberland, Andrea Bertozzi, and Spring Berman. "Pheeno, a versatile swarm robotic research and education platform." *IEEE Robotics and Automation Letters* 1, no. 2 (2016): 884-891.
18. Jdeed, Midhat, Melanie Schranz, and Wilfried Elmenreich. "A study using the low-cost swarm robotics platform spiderino in education." *Computers and Education Open* 1 (2020): 100017.
19. Koenig, Nathan, and Andrew Howard. "Design and use paradigms for gazebo, an open-source multi-robot simulator." *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. Vol. 3. IEEE, 2004.
20. E. Schmidt, Ed., "Mid-Decade Challenges to National Competitiveness," *Special Competitive Report Project*, Sep-2022. [Online]. Available: https://www.scsp.ai/wp-content/uploads/2022/09/SCSP-Mid-Decade-Challenges-to-National-Competitiveness.pdf. [Accessed: 10-Feb-2023].