AFRL-RI-RS-TR-2021-190

AGILE-TEAMS TESTBED (AT2B)

CUBIC DEFENSE APPLICATIONS, INC

NOVEMBER 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2021-190 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
STEVEN L. DRAGER
Work Unit Manager

/ S /
GREGORY J. HADYNSKI
Assistant Technical Advisor
Computing & Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
NOVEMBER 2021	FINAL TECHNICAL REPORT	AUGUST 2017	JUNE 2020

4. TITLE AND SUBTITLE

AGILE-TEAMS TESTBED (AT2B)

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8750-17-C-0204	N/A	61101E
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
TEAM	4C	UB

6. AUTHOR(S)

Daniel V. Magaha and Nicholas Macron

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Cubic Defense Applications, Inc.
101814 Jollyville Road, Bldg. 4, Suite 250
Austin TX 78759

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DARPA
675 North Randolph St.
Arlington VA 22203-2114

Air Force Research Laboratory/RITA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

RI

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-RI-RS-TR-2021-190

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This final technical report documents the Cubic/Intific Technical Area 1 team's efforts, progress and findings under the Defense Advanced Projects Agency Agile Teams program. The objective of the A-Teams Test Bed was to create a dynamic intelligent testbed for team operations and this effort began with a squad-based first-person shooter (Squad Virtual Test Bed), which was previously developed for the Squad X program, and ended with a new testbed called Hide and Strike, which focused on a turn-based strategy mechanic. This report overviews the approach iterations of the research, design and applied software development the team undertook in deriving a human-machine teaming testbed.

15. SUBJECT TERMS

Agile, human-machine teaming testbed, dynamic, intelligent, predictive models, human-intelligent machine systems

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
U	U	U	SAR	87

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
STEVEN L. DRAGER	N/A

PREVIOUS EDITION IS OBSOLETE.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

Over the course of nearly three years of continuous research, design and applied software development, the Cubic Intific team iterated over several approaches to a human-machine teaming (HMT) testbed, starting with the Squad Virtual Test Bed (VTB) that was previously developed by Cubic for the Squad X program.

Figure 1. Squad Virtual Test Bed (VTB)

This new testbed, dubbed Agile Teams (A-Teams) Test Bed (AT2B), was a squad-based first-person shooter (FPS) game built on a commercial video gaming engine, Unreal Engine. AT2B was instrumented for artificial intelligence (AI) and successfully utilized in a number of test events between 2017 and 2018, but extracting interesting results proved elusive due to the sheer amount of variability in the testbed and the real-time operations tempo (OPTEMPO) of the game.

Throughout 2019, substantial refinements and redesigns of the experimental approach and testbed were completed, but these efforts each fell short of producing the desired results; namely, statistically significant evidence of the effects of AI on human-AI teaming.

Finally, in early 2020, the decision was made to pivot away from the AT2B testbed entirely and design an entirely new testbed, called “Hide And Strike Testbed” (HST), specifically to focus on a turn-based strategy mechanic which greatly simplified the experimental environment and, it was hypothesized, would be significantly easier for AI’s to reason over and provide meaningful assistance to human players.

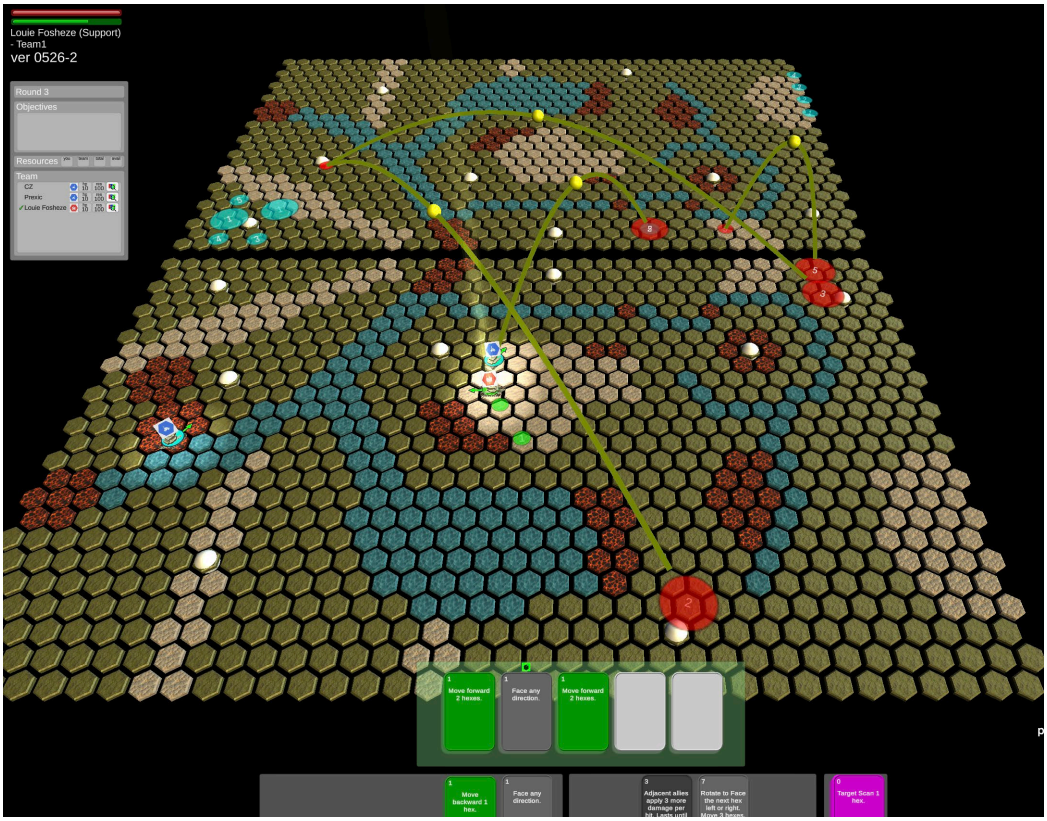


Figure 2. Hide and Strike Testbed

Development on the HST testbed started in earnest in February 2020. By the end of May, the testbed was complete enough to support several hours of internal, 6v6 multiplayer playtest sessions, validating the approach. This testbed showed significantly more promise than the AT2B in the areas of relevant data collection, ability to scale and support larger tests, and suitability for AI integration.

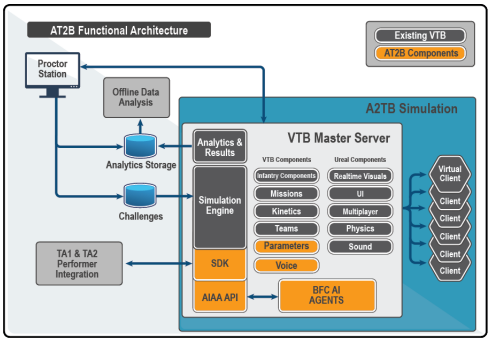
However, by June, it was apparent that the remaining technical lift, combined with the overall objectives and goals of the program, represented too much scope to be completed with the remaining program budget; continued development was too risky. After consulting with the Defense Advanced Research Projects Agency (DARPA) program manager (PM) (Dr. John Paschkewitz), the effort was halted.

2.0 INTRODUCTION

Work on the DARPA A-Teams program began in August 2017. Cubic's sole subcontractor was Boston Fusion Corp. (BFC). The scope of the work was largely development of the AT2B testbed, while BFC focused on the development of machine learning-capable AI.

The initial program quad chart is presented in Table 1.

Table 1. Initial Program Quad Chart

<table>
<tr>
<td>Graphic:
 
</td>
<td>New Ideas:
1. Developing a general-purpose AI Augmented Avatar capability that can be placed anywhere within a team's leadership/team structure;
2. Integrating the AI into AT2B and providing a Software Development Kit (SDK) that transparently supports A-Teams experiments for a broad array of Technical Area (TA) 1/TA2 approaches
3. Expanding the existing military Squad paradigm to other professional or technical fields as the solicitation requires.</td>
</tr>
<tr>
<td>Impact: Highly flexible hybrid testbed
• Extend Squad VTB to provide the A-Teams program the flexibility its TA1/TA2 performers need to test their theories and formalisms
• Provide the current software, SDK, and large amounts of test data for TA1/TA2 performers
• Immersive and Multiplayer with scalability for 50 users</td>
<td>Schedule:
<table>
<tr><td>Q3 2017
Aug-Sep
Kickoff</td><td>Q4 2017
Oct-Dec</td><td>Q1 2018
Jan-Mar
Pilot Test</td><td>Q2 2018
Apr-Jun</td></tr>
<tr><td></td><td>Q4 2018
Oct-Dec</td><td>Q1 2019
Jan-Mar
Scenario Dev</td><td>Q2 2019
Apr-Jun</td></tr>
<tr><td></td><td>Q4 2019
Oct-Dec
Demonstration</td><td>Q1 2020
Jan-Mar</td><td>Q2 2020
Apr-Jun</td></tr>
<tr><td></td><td>Q4 2020
Oct-Dec</td><td>Q1 2021
Jan-Mar
Test Event</td><td>Q2 2021
Apr-Jun</td></tr>
</table></td>
</tr>
</table>

3.0 METHODS, ASSUMPTIONS AND PROCEDURES

The proposal and base assumptions during the initial phase of development were that Cubic would focus on the development of the AT2B testbed, starting with the Squad VTB as a basis and augmenting with application programming interface (API) and support technology as needed, while Boston Fusion would simultaneously focus on generalized AI research and leverage these AI learnings by implementing game-theoretic AI (GTAI) into the actual AT2B testbed.

Initial distribution of labor and milestones can be seen in Figure 3.

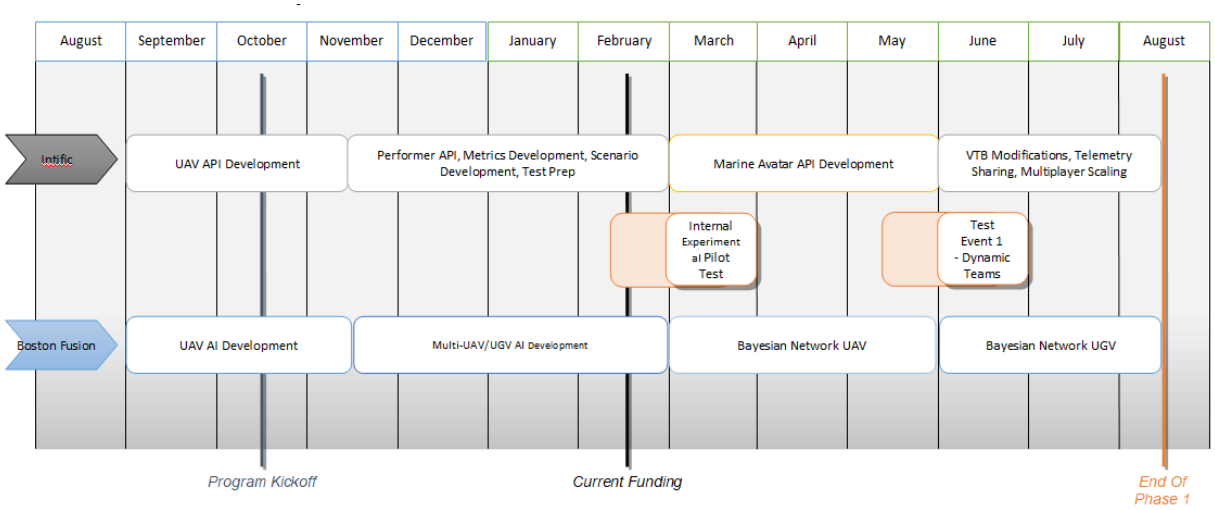


Figure 3. Initial schedule and division of labor

The research team utilized the Scrum agile software development methodology[1], which advocates rapid iteration on regular, short development cycles called “sprints” and seeks to accelerate the delivery of value to the end user as much as possible.

Scrum accommodates requirements changes by maintaining an active “backlog” of customer requirements (referred to as “user stories”), as illustrated in Figure 4, and teams re-plan their work at the conclusion of each two-week sprint. This method effectively provides an active scope management system while preserving maximum flexibility to pivot to new approaches in service of the research and development (R&D) goals of the program.

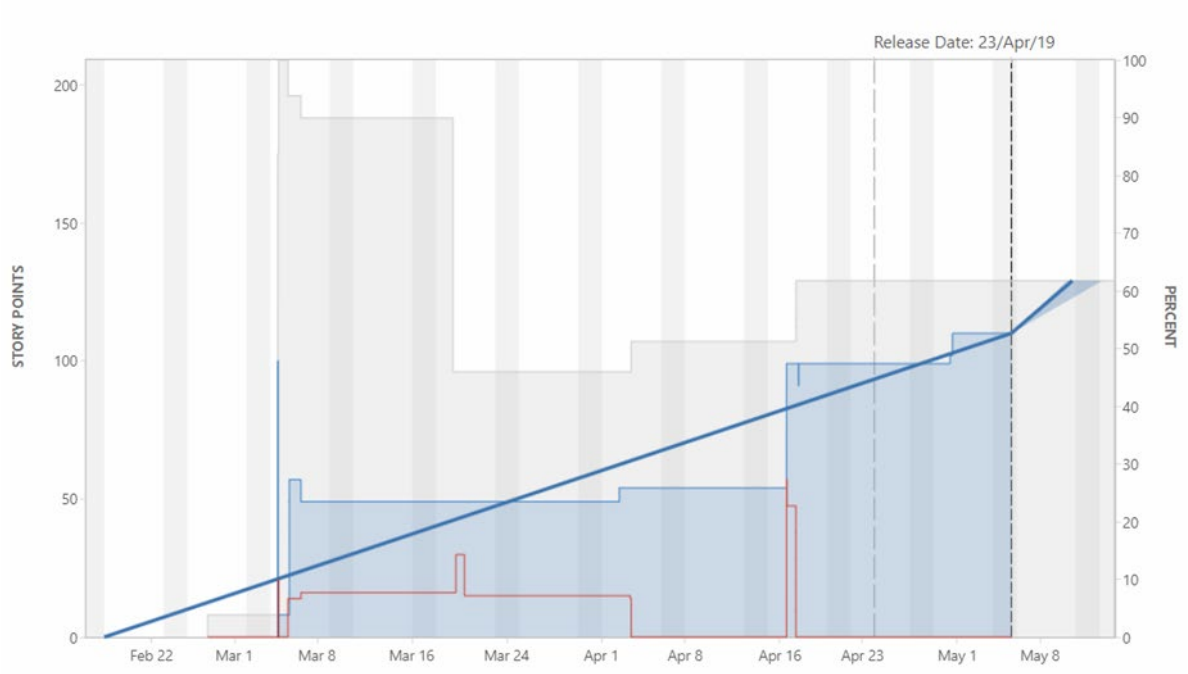


Figure 4. Scrum story point tracking, May 2019

4.0 RESULTS AND DISCUSSIONS

4.1 Phase I: August 15, 2017 – August 14, 2018

After all contract activities were completed, an initial meeting was held to coordinate plans, goals, schedules and expectations. Similar telecoms were conducted with the other A-Teams performers and The MITRE Corporation (MITRE) to collaborate on a test and evaluation technical approach. This provided the initial requirements needed to begin modifying the existing Squad VTB codebase to support A-Teams and create the AT2B testbed.

Specific early modifications to the VTB included:

- Integrating ZeroMQ[2] for messaging that would interface with Python scripts developed by Boston Fusion and potentially other performers in the future.
- Developing APIs for simulation information such as entity states and unmanned aerial vehicle (UAV) control, which allowed Boston Fusion to develop an initial AI engine that would function within the simulation.

BFC developed an UAV AI prototype within the testbed that launched a quadcopter drone from the command post to a specified altitude and began sweeping the camera frustum in a search pattern to identify opposing force (OPFOR) targets. This AI algorithm utilized a Bayesian Network approach and was capable of making rudimentary decisions in the testbed environment and could even control the simulated UAV on-board camera.

While fairly simple in its initial iteration, this established functioning software between the two organizations and a first step towards future development. These efforts were demonstrated at the A-Teams Program kickoff in October 2017.

Figure 5. Twentynine Palms environment in AT2B with UAV's

Q4 2017

In the 4th quarter of 2017, development progressed towards a sample raid mission in which a single Squad navigates through the diplomatic district of Range 220 of Twentynine Palms, Figure 5, towards a suspected target area populated by hostile forces.

Previously, a human participant was required to provide command and control to the AI controlled Blue Force (BLUFOR) and UAV assets. The testbed was modified to instruct the squad to navigate through a series of zones enroute to the target area without human intervention.

The research team augmented the UAV AI engine to allow the direction of a UAV asset with high-level commands and set behaviors. For example, moving the UAV could be given an altitude parameter and the AT2B platform would accommodate for changes in terrain elevation automatically, Figure 6. Other changes to the API included providing detailed target identification information (including the target id, type, and status) when an entity was detected in the UAV's camera frustum.

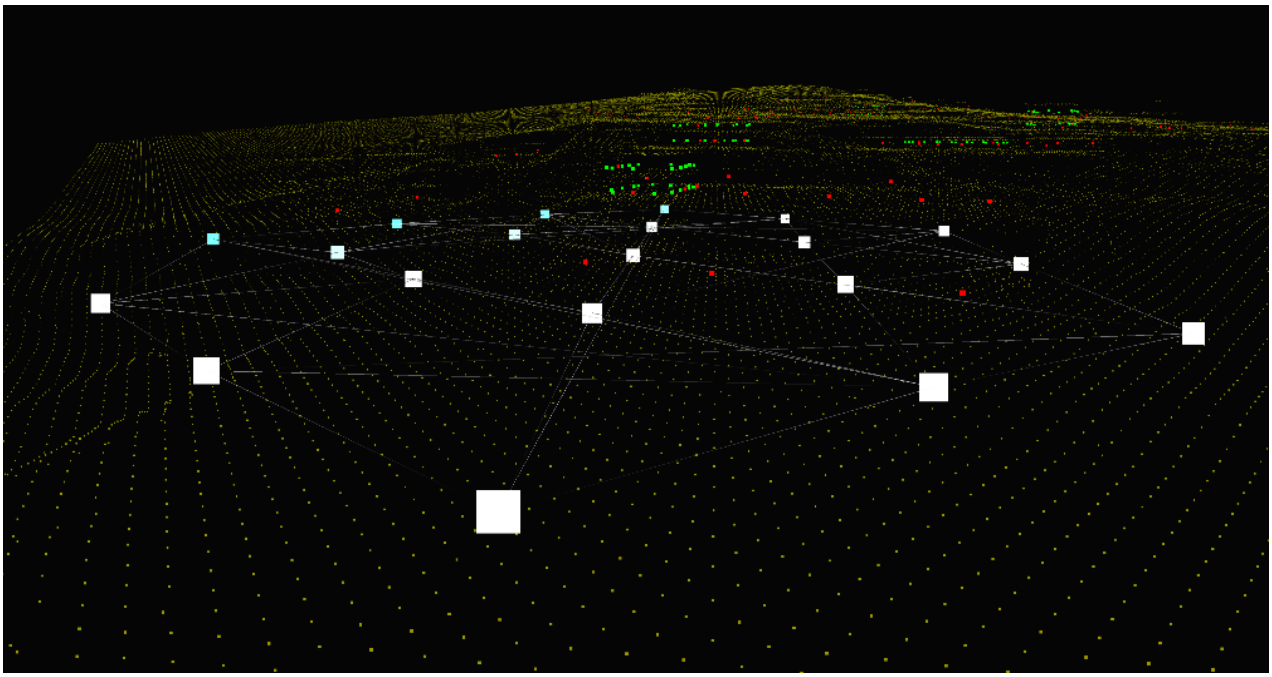


Figure 6. Voxelized terrain representation of space / state data

The first of three UAV's would autonomously deploy to zones enroute to the target area and scan the area for hostiles for a specified time. The second would deploy to and scan the subsequent zone as would the third. When a UAV determined that the area was clear, a signal was provided to the Squad Leader that he was clear to proceed. The UAV would then leapfrog the other deployed UAV's to the next zone. This cycle continued until the UAV found contact at the target area.

New simple behaviors were also created that would trigger Marine avatars to conduct simple building clearing tactics as they entered each zone. The fidelity of these behaviors was meant to provide only a gross representation of the appropriate tactics, techniques, and procedures (TTP).

Q1 2018

In Q1 2018, to address server and cloud infrastructure requirements, as shown in Figure 7, the team developed a new Broker Server that included migrating to a Rust-based service, capable of running hundreds of UAV's with swarm behavior and obstacle avoidance.

Additional Rust[3]-based experiments were built and deployed using DevOps platforms-Docker[4] containers, Jenkins[5], and Kubernetes[6]. The team also experimented with low bandwidth / high latency testing in order to align with DARPA's cloud services provider.

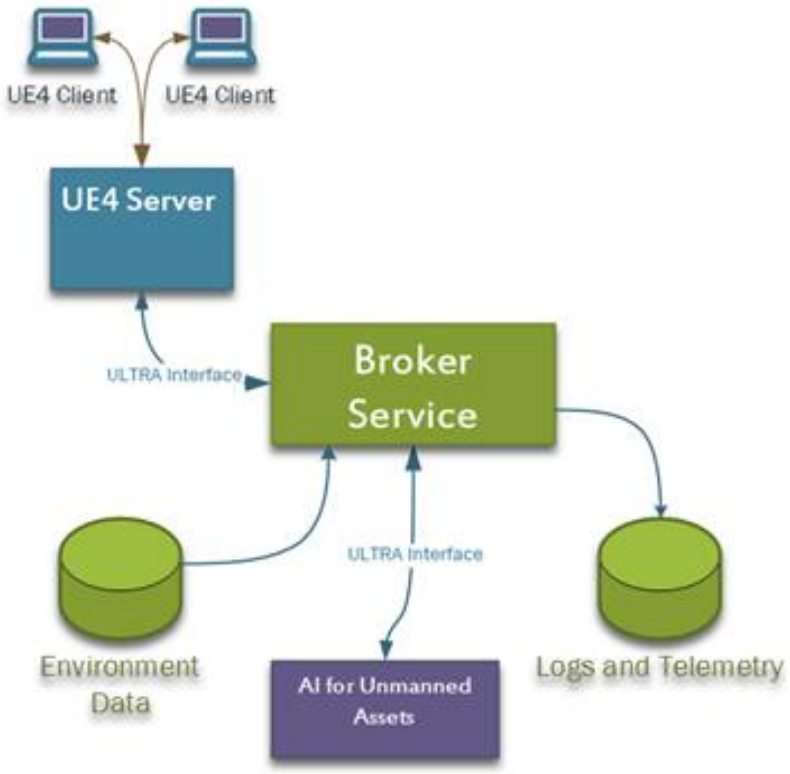


Figure 7. Broker Service

Boston Fusion continued to research “resilient AI’s” across a number of varied subject areas (swarming, centralized AI algorithms, resiliency through recalculation) and focused their applied engineering efforts on simple UAV AI at this point, as represented in Figure 8.

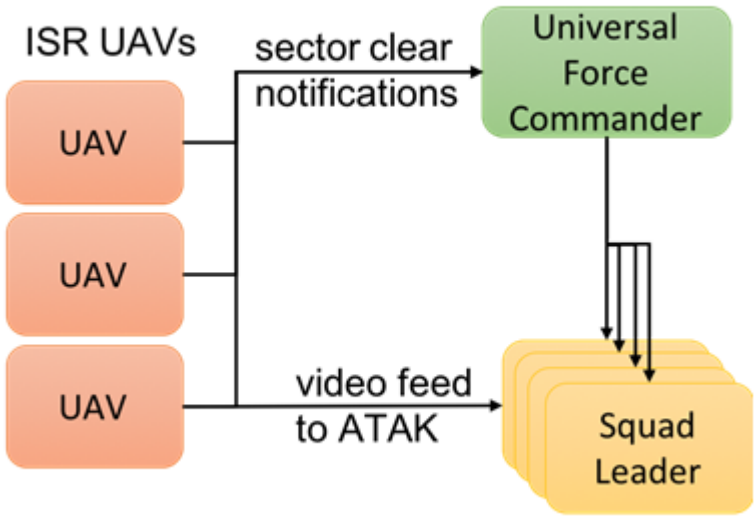


Figure 8. Boston Fusion lightweight UAV AI

During this period, the art team focused on the task of Procedural Urban Environment creation utilizing Houdini[7] software, examples shown in Figure 9. The team developed two paths of procedural creation– city generation and structure generation. City generation focused on creation of terrain, roads, footprints and characteristics for structure. Structure generation focused on the creation of marked-up spaces with needed features (doors, windows) and sizes / height. This process then integrated both city and structures and created AT2B ready environments, with markup needed for interaction, AI, and physics.

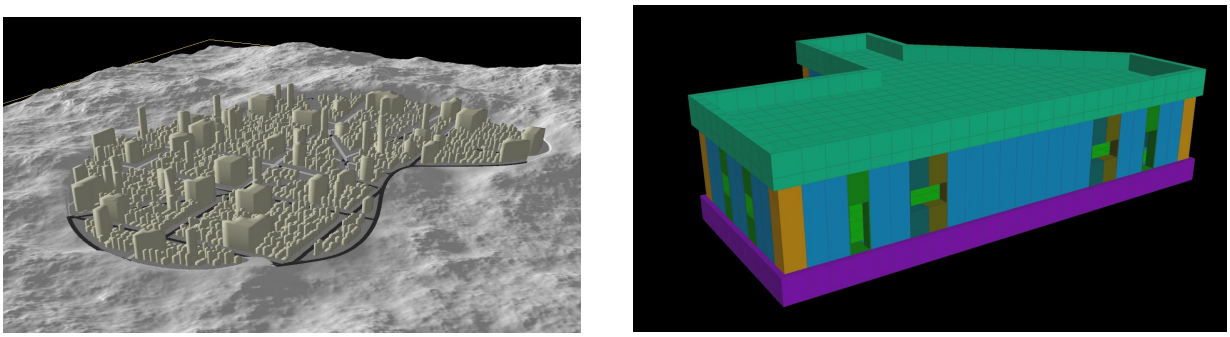


Figure 9. Procedural urban environment experiments

Finally, to stay aligned with the program's experimental plan schedule, meetings were conducted with both MITRE and Aptima to discuss testing approaches, requirements and considerations. Through discussions with these teams, it is was decided to start with two experiment tracks, separating TA-1 and TA-2 concerns and focus on fixed team structure testing. The teams developed a first hypothesis and test objectives, as well as decided on an iterative plan starting with a Pilot Test in April 2018.

Q2 2018

In Q3 2018, the first test of the AT2B testbed was conducted, an internal Pilot Test, with the scenario depicted in Figure 10. The overall goal of the Pilot Test was to run missions that would test five different team configurations developed with input from both A-Teams performers and MITRE. Through these test configurations the team wanted to examine the different interactions between human and AI players within a series of different roles within a mission.

The team also wanted to conduct an internal test in order to set a baseline and start to gather findings prior to the initial external test, which ultimately occurred later in the summer of 2018. The team intended to utilize the data gathered from the Pilot Test to better understand the software and the human-AI relationship, and to influence future development for the benefit of the external test.

The Pilot Test was conducted onsite at the Intific office in Austin Texas, using internal employees that were familiar with the Squad VTB project that AT2B is built on. This previous experience meant that they were familiar with the controls and interface and required only a short briefing of the Pilot Test objectives prior to their play sessions.

The team learned a great deal about the AT2B software, the testing process, and the testing cohort in this Test. The test exposed several testbed and experimental design flaws:

- The prototype Heads Up Display (HUD) system for displaying command and control (C2) interactions was inadequate for the users. The graphics were confusing, the "debug" drawing was hard to see, and the information did not persist.
- The virtual players had very little interaction with the C2-driven UAV assets. There was no easy way to see the intelligence products from the UAV's, unless the player was constantly interacting with the virtual Android Tactical Assault Kit (ATAK)
- Because of the lack of interaction with the UAV / overwatch system, players fell back on their gaming instincts, and utilized aggressive, direct fire tactics, killing enemies rather than engaging the UAV and taking advantage of the AI.

- The mission was too short and too easy. The team did not have any mission failures, indicating that none of the players were truly cognitively overloaded and needed to rely on AI.

Figure 10. Aerial view of AT2B scenario used for internal pilot test

Based on the results of this initial Internal Pilot Test, the team implemented a number of changes to AT2B, including:

- Increased scenario difficulty (random OPFOR spawns, random improvised explosive device (IED) / Jammer spawns, measures of performance (MoP) / measures of effectiveness (MoE) telemetry)
- Enhancing usability (refactored HUD / user interface (UI), expanded tutorials, implemented additional UAV behaviors)
- Developed Monte Carlo (MC) functionality (ability to execute multiple iterations of a scenario heedlessly and generate telemetry)
- Variable team configurations that led to different MoP / MoE results
    - UAV equipment configurations
    - Fireteam Configurations (Size, Formation, Loadouts)

    - Individual Member Variables (Reaction speed, Accuracy, “volume” clearing”)
    - Human variable equivalency calibration
- Environmental IED's that require Team Configuration Changes (spawnable IED’s, combat and movement AI behaviors)

Implementing the ability to randomly spawn OPFOR, IED’s, and jammers provided a number of different parameters that could be adjusted to affect the scenario and provide different paths to success. In addition, the team configurations provided another set of variables that could be adjusted and offer additional ways to play out the scenario.

Refining the environmental IED’s also required the introduction of new behaviors and AI to properly detect and react to those IED’s in the environment.

The “Monte Carlo” build was an effort to allow a way to rapidly simulate multiple runs of the scenario utilizing all of the parameters and provide data outside of a fully manned Pilot Test. This feature could be configured to run with minimal external input and could optionally run “headless” without the need for a client.

Q3 2018

The primary focus of this period was the hardening of the AT2B platform to prepare for the Reserve Officers’ Training Corps (ROTC) External Pilot Test.

Testbed improvements completed during this period included:

- Implemented IED’s into the scenario
    - Includes discovery by UAV and disabling via unmanned ground vehicle (UGV) (the UGV’s were later removed due to defects)
- Added randomized scenario elements
    - Random OPFOR spawn locations throughout the active sectors of the Twentynine Palms map
    - Added hidden IED placements
- Scenario Team Configuration
    - Added unique skills and skill assignments to Marine avatars
- Added a Human Skill Calibration tutorial
    - Shooting Accuracy (Shooting range)
    - Reaction Time
    - Traversal (simple maze)
- Implemented autonomous UAV capabilities for control at the individual level

    - Angel Overwatch (provide adaptive, local Intelligence, Surveillance and Reconnaissance (ISR) and Red Force detection)
    - Responsive to attacks
    - ISR of assigned sector
- AI support for IED tasks
    - IED detection by UAV's
    - Prototyping UGV AIs (IED disposal)
- Finalized IED and Jammer functionality
- Tested MoP / MoE telemetry
- Tuned the randomized OPFOR, jammer and IED functionality
- Updated Team configs to support Large size squads
- Created standalone MoP / MoE Data Visualization Tool
- Tested multiple squad configuration and setups.

4.2 Phase II: August 15, 2018 – August 14, 2019

The first significant event of this period was the first ROTC External Pilot Test. The team had originally planned to conduct this test in June but scheduling this with the local (University of Texas) ROTC proved impossible as those students were on assignment during the summer school break. The team ultimately executed the test in late August 2018, onsite at the University of Texas in the ROTC training classrooms, as seen in Figure 11.

Figure 11. First External Test - University of Texas ROTC, August 2019

The overall goal of the ROTC External Pilot Test was to capture data within the new scenario, with a focus on different team configurations and an overall larger squad deployment. The team broke down the tests with a mix of large and small squads, as well as local and global UAV communication. The loadouts for the Marine Avatars were consistent across those scenarios to minimize the testable variables in the scenario.

Other variables included OPFOR, IED and jammer locations, as well as the sectors of the map that were to be cleared by the fireteams. It was up to the fireteam leads how to approach the clearing of each sector, and they were given time before running through the map to plan out the mission.

Each scenario test cohort consisted of three College Senior level ROTC cadets, and one Master Sergeant to provide commander (CO) duties with each team. Each of the 4 players in the scenario were fireteam leaders, with one designated as the squad lead.

A scenario was considered a “pass” if all sectors were cleared, and all OPFOR eliminated. “Failures” were a result of taking too many casualties. It was important that each cadet approached the scenario in a serious manner, and not simply as a video game. If a player was killed in action (KIA), they were allowed to see a bird’s eye tactical view of the map but were not allowed to communicate with the other players.

A summary of the ROTC External Pilot Test protocol follows:

- 4 groups were tested, consisting of 3 senior cadets and 1 CO each.
- Prior to beginning the test, the team gave each group an overview of the program, as well as a demo of the basic testbed functions.
- Each group completed a tutorial and aptitude test before they started official test runs
- Each group completed 4 runs based on Team Configurations vetted by the team at MITRE:
    - Large and Small Team configs with Global and Local unmanned vehicles (UxVs)
    - Simple and Complex scenarios (defined by greater OPFOR presence and IEDs)
- Telemetry was captured and reviewed with the cadets as part of the after action review (AAR). Figure 12 shows telemetry playback.

After each session, the team was able to conduct an AAR with the cadets and use it as a learning tool, which ROTC leadership found a useful training tool. While this wasn't one of the stated research goals, this was an important generalized discovery from the test.

Figure 12. Telemetry playback in AT2B after ROTC External Pilot Test

The team was also able to develop a standalone visualization tool, shown in Figure 13, that parses MoP / MoE data from completed AT2B scenarios, which was also well-received by ROTC leaders.

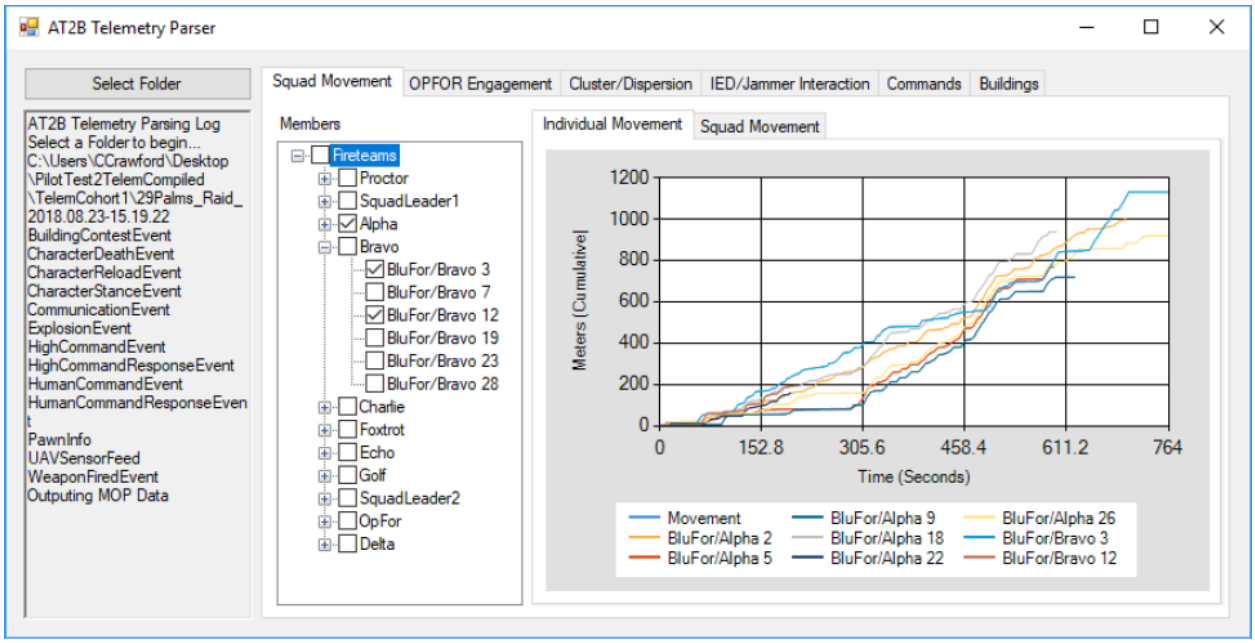


Figure 13. Standalone MoP / MoE data visualization tool

A summary of ROTC External Pilot Test results is as follows:

- Raw data capture was better than expected, primarily due to the stability of the application during the test period.
- In addition to the Telemetry data, the team captured baseline data from the aptitude tests to help normalize human vs AI skill.
- Raw data was delivered immediately following the test to MITRE, Hughes Research Laboratories (HRL), and Aptima for analysis.

While the interaction with the environment and the after-action analysis with the participants was interesting, no tangible insights into the human / AI dynamic were discovered.

During this period, Boston Fusion continued their research into C2 AI techniques, see Figure 14, exploring concepts such as local utility functions like “the Wonderful Life Utility (WLU)” and negotiation mechanisms – but little of this translated to meaningful improvements to the in-game AI.

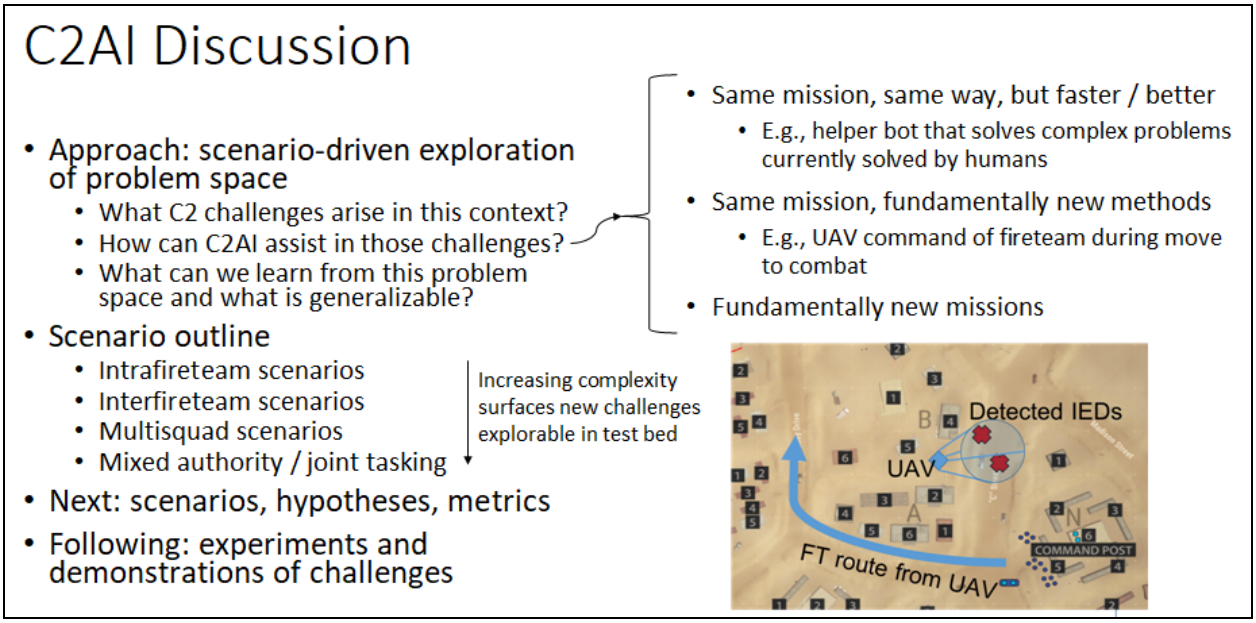


Figure 14. Boston Fusion C2 AI research from September 2018

At the request of Boston Fusion and to support their research efforts, the team added supporting gameplay features such as random enemy placement for adaptive purposes and various UxV classes (Jamming, Scanning, Mortar).

However, by this point, a shared understanding began to emerge that a continued focus solely on UxV AI techniques alone would not be productive.

Q4 2018

By this time, the inability to extract consensus from other program performers on the appropriate direction for the AT2B testbed had become a recurring and growing concern. It was critical that the team make changes to maximize chances of interesting experimental AI discoveries, but there was not broad agreement on how best to accomplish this.

The regularly scheduled Principal Investigator (PI) Meeting on October 25th and 26th, 2018 would prove to be an opportune time to discuss this concern. During the PI meeting, it was decided that MITRE would take a larger role in gathering information from the performers and disseminating it to all other groups in an effort to harmonize feedback and provide the team with a "single voice" to guide experimental design, AI research and engineering focus.

Testbed improvements completed during this period included:

- Built API for fireteam controls (Suppress / Overwatch, Move To, and Hold)
- Monte Carlo setup for Wright State
- Updates for jammers and UGVs to support Boston Fusion

R&D was conducted in the following areas, based on feedback / direction from MITRE:

- Smoke and cover (Human & API)
- OPFOR Python API to allow scripted commands for OPFOR.
- “Suppress a particular location” command (Human & API)

Boston Fusion AI developments completed during this period included:

- Prototype C2 AI for building clearing with Fire Teams
    - Overwatch / Suppression coordinates with assault
    - Updated framework supports:
- Coordination among entities
    - Defining sequence of commands as one action
    - Simultaneous actions (command combos)
    - Pause actions of programmable length
    - Facilitates incorporation of commercial game AI architectures (behavior trees, blackboards, etc.) for faster AI prototyping

Q1 2019

During this period, consensus among the performers built on the need to redesign the testbed around a player-vs-player scenario with C2 AI as the initial focus. This was conceptualized as a “Domination” play mode, shown in Figure 15, in which players compete to hold a given position for a length of time to score points and win the game.

Additionally, the DARPA PM expressed his desire to modify the testbed such that it would qualify for Distribution A, ensuring a much larger potential candidate testing pool that could be released publicly. This would require reworking game modes, environment and weapons assets to genericize anything that might constitute actual Marine TTP’s.

Figure 15. "Domination" game mode environment

In early February, the team began work on the Domination game mode, with a focus on getting a Monte Carlo build delivered to Boston Fusion so that AI training could start as early as possible.

At the end of February, a group of Marines were visiting for a meeting on the DARPA Persistent, Resilient Operations Testbed for Expeditionary Urban Scenarios (PROTEUS) program. The team was able to set up AT2B for an impromptu playtest with them. The telemetry viewer / AAR tool was very well received.

During this period, to mitigate the limited pace of applied AI functionality Boston Fusion was able to deliver, the team added an internal engineer with AI / machine learning (ML) experience to the team. His initial tasks were research related, with a plan to integrate the Agent AI into the testbed later in 2019.

In mid-March, the team presented a new AT2B “Vision Doc” to the entire group of DARPA A-Teams performers for review and feedback. This document described the basic parameters of the new Domination mode, see Figure 16, the rationale behind the changes, and how the new mode would work.

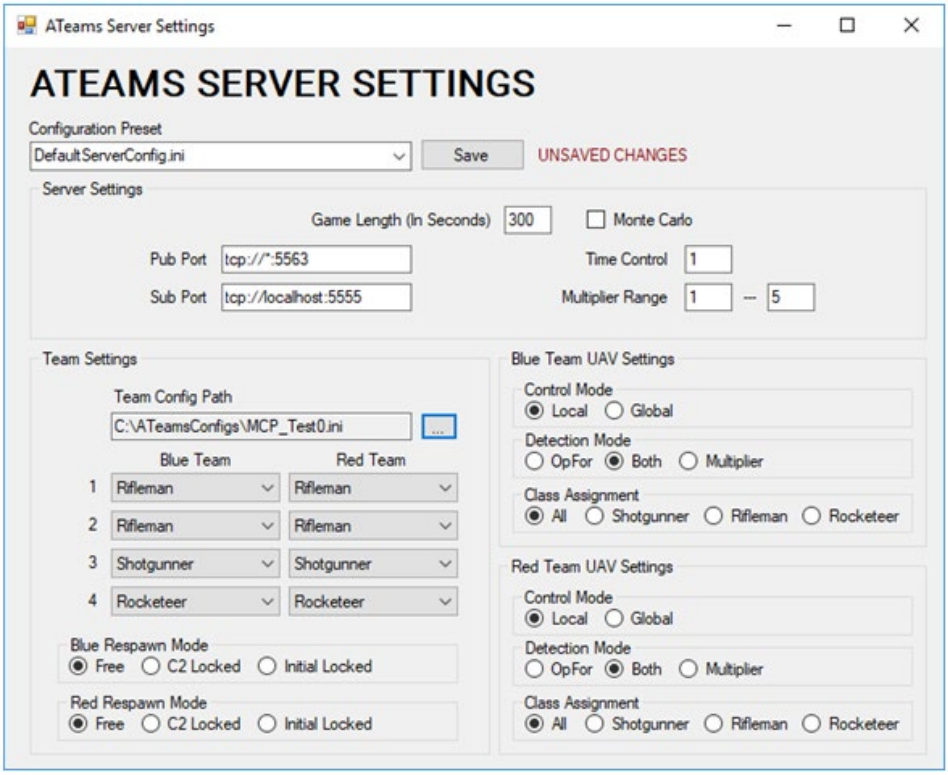


Figure 16. New AT2B Server options supporting "Domination" mode

- 4 v 4 Human players w/updated spawning system
- Class based avatar package
    - Shotgunner – short range, smoke grenade gadget
    - Rifleman – medium range semi-auto rifle
    - Rocketeer – long range explosive payload, UAV gadget
- New Distro-A-friendly map to support 8 players and game mode changes
    - Clutter nodes to provide additional AI cover
- Telemetry updates to support new mode / classes
- AI Design
    - Map to be discretized into ~200 tactically important nodes, see Figure 17
    - C2 AI observes state of testbed (e.g., score, positions of units, control point ownership)
    - C2 AI sends units to different nodes with some basic instructions (e.g., overwatch control point (CP))

- Game AI (or human players) execute the commands as best they are able
- Reward is calculated from change in score. Policy function is updated.

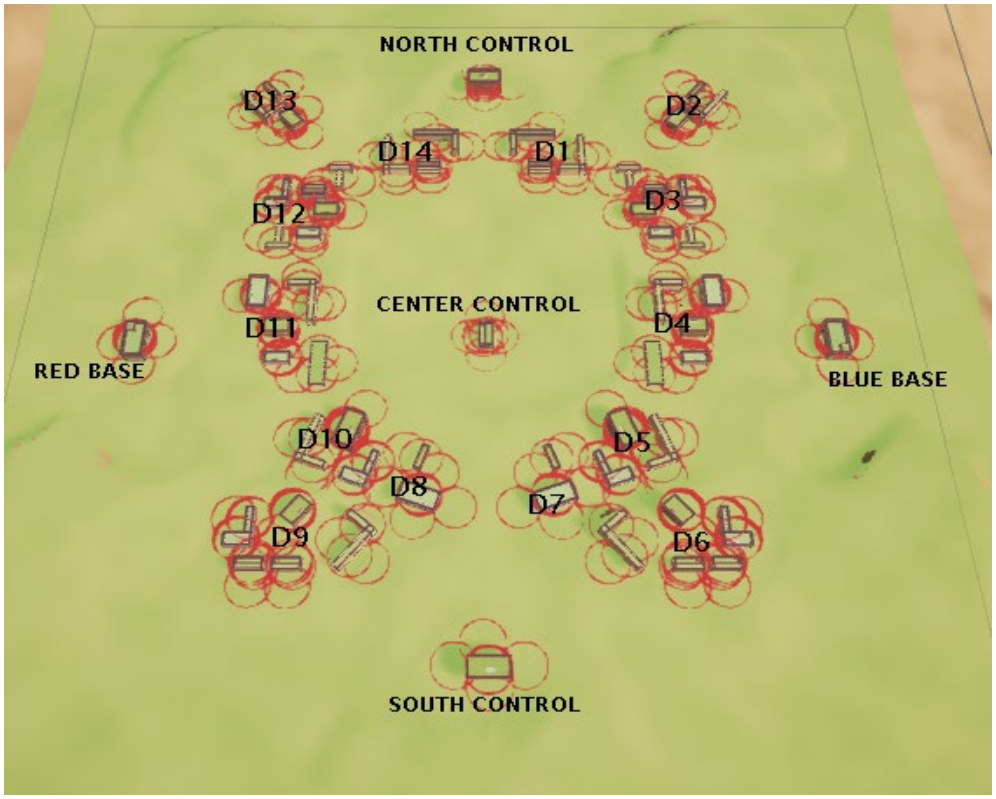


Figure 17. "Domination" map visualization of discretized AI nodes

Testbed improvements completed during this period included:

- Added a server setting to adjust the colors of the different team "vests". This allows for much easier visual identification, and this server setting gives the team a simple way to adjust colors as needed without needing an art pass.
- Updated the existing AIs for "commandable" behaviors to allow them to be tested and trained properly.
  - Updated the Overwatch / Suppress API 's
  - Completed Python unit tests to help with AI training
  - Developed and refined API's for the new Domination game mode
- HUD and user interface (UI) updates

- Implemented Lobby UI functionality to allow players to choose their classes at respawn
- Foundational work for the C2 to human interface display
- New HUD elements to display key information for the Domination game mode, shown in Figure 18

Figure 18. Prototype HUD elements to support "Domination" mode

R&D conducted in this period included a new reinforcement learning model, shown in Figure 19, which was intended to learn from gameplay sessions and offer players helpful C2 strategies and actions based on “learning”.

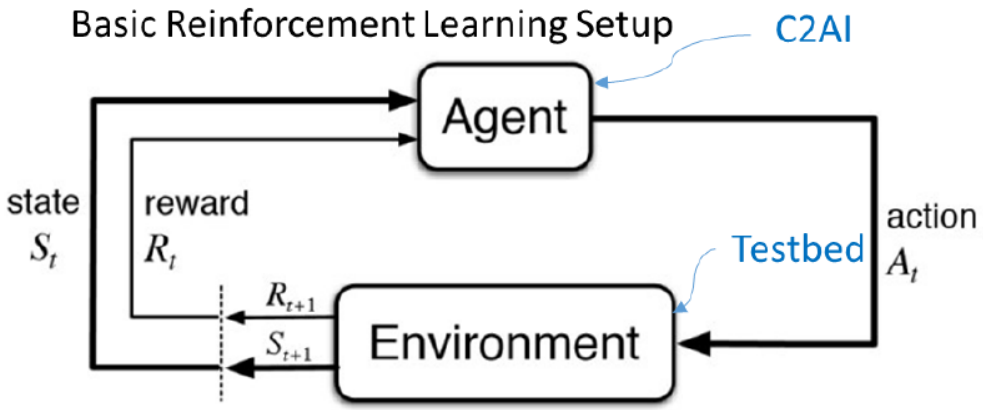


Figure 19. Reinforcement Learning (RL) model for AT2B C2 AI

Goal: Train a policy function that selects the best action set, given game state

- Train in AT2B using modified self-play in MC build
- Policy gradient optimization with Long Short-Term Memory (LSTM) policy functions
- Evaluate and expand to include additional units, information, capabilities (e.g., UGVs, Electronic Warfare (EW)) once the team have initial agent working

RL for C2 AI

- C2 AI provides instructions to marines and UAVs
- Individual entity AIs execute those actions
- C2 AI learns from experience where and when to send different units
- Control size of state-action space to limit data requirements

Q2 2019

The main focus of efforts in this period was the May Playtest at the MITRE Lab in Virginia. The team implemented, tested, refined and hardened the Domination game mode that had previously been designed through a series of internal playtest events to validate the game's stability and playability as well as exercise the APIs and Telemetry system, see Figures 20 and 21. Sample telemetry outputs were distributed to partners to verify analysis pipelines prior to testing.

Figure 20. Class selection UI developed for MITRE playtest

The team worked with both Boston Fusion and MITRE concurrently to plan risk mitigations addressing BFC's challenges in yielding a convergent C2 AI. Several potential solutions were proposed, including utilizing a human subject matter expert (SME) in-the-loop, implementing "bots" using native game controller software and an internal parallel development of a C2 AI.

Ultimately, the team started a short burn effort to train a comparable C2 AI as a backup plan in the event BFC was not able to converge. The C2 AI eventually yielded some successful game winning strategies and was chosen for the May playtest.

Several Intific team members traveled to the MITRE lab to support the play test event. The testbed performed well and was received well by the United States Marine Corps (USMC) participants. The event provided Intific's Data Scientist and AI lead for A-Teams with good insights into the game / human dynamics of such events as well as highlighted future balancing and game design opportunities to Intific's A-Teams PI. Overall the event was a success albeit the Intific C2 AI was not as effective as initially hoped.

Figure 21. New HUD developed for MITRE playtest

June was spent conducting post-experiment analysis as well as testbed refinement and planning. The team presented analysis and metrics to DARPA and MITRE during the June monthly call as well as initial plans towards a refined November play test. Certain human exploited aspects of the May testbed were addressed in the updated design with the purpose of cultivating conditions where the AI's assistance to humans was necessary and useful.

Although the Intific C2 AI was able to provide player guidance, it needed substantially more training time to be effective, as can be seen in Figure 22. A complete analysis of this test can be found in *Appendix A* of this report.

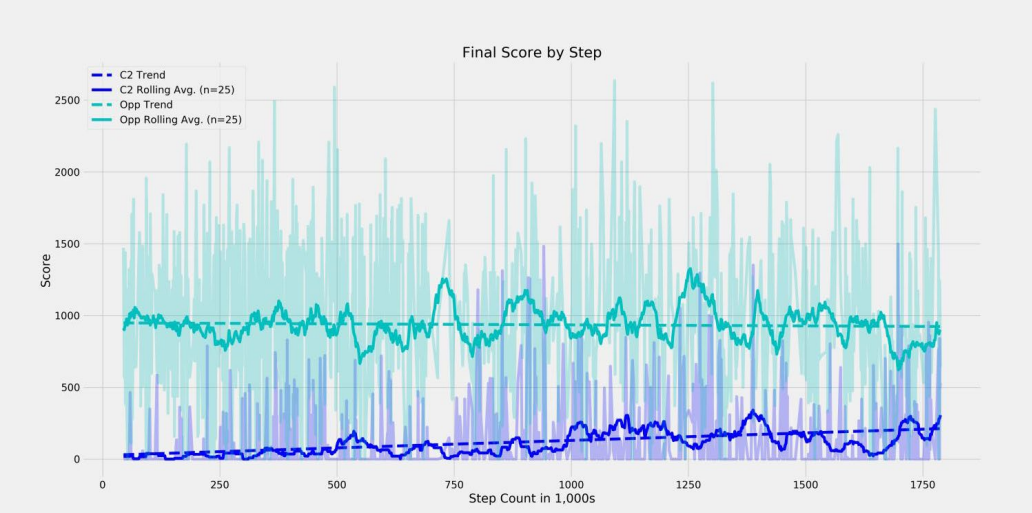


Figure 22. Intific C2 AI Training - Final Score vs Step

Q3 2019

Development during this period was focused on preparing for the September 2019 Internal Playtest, which was conducted at Cubic offices. The purpose of this test was to validate the suitability of the changes that had been made to the testbed, the efficacy of the AI, and ensure the testbed was in good shape for first potential human subjects research (HSR) experiment which was slated for December.

The team made the decision to formally issue a Stop Work order to Boston Fusion during this period, noting the lack of progress in developing an applied AI capable of supporting the experimental efforts initially set forth in the A-Teams program goals. The internal RL / AI developer had made sufficient progress in this area to deem further funding of BFC unnecessary. As a result of this change, the team began to work much more closely with Aptima and MITRE to collaborate on experimental design and testbed changes. From this point forward, Aptima provided any C2 AI the team used, and internal engineering resources provided rudimentary ISR AI.

Testbed improvements completed during this period included:

- **Reducing map visibility** through volumetric fog, which was implemented to limit players' view of the entire battlespace and force the use of UAV's to gain additional situational awareness.
- Adding **dynamic objective points**, which add a sense of urgency and adaptation to player strategies, increasing the need to explore and reducing the chances of players repeating the same tactics each session.
- Adding the **symbol match system**, which serves as a proxy for intelligence gathering. This system rewarded players with extra points for matching their individual symbol to the current global symbol, encouraging players to visit team bases to respec when the symbol changed.

- Adding **resupply points** at team bases, which allowed players to **respec** their player class and “symbol” and heal, see Figure 23.
- Implemented several **player avatar updates**, such as increasing movement speed, modifying weapon damage and adding a respawn penalty. These changes discourage solo tactics that are less likely to rely on the C2 AI.
- Implemented several **UAV modifications**, making the UAV more useful but also more work for human players, thereby increasing the value of the C2 AI performing those functions.

Figure 23. Symbol and player class respec functionality

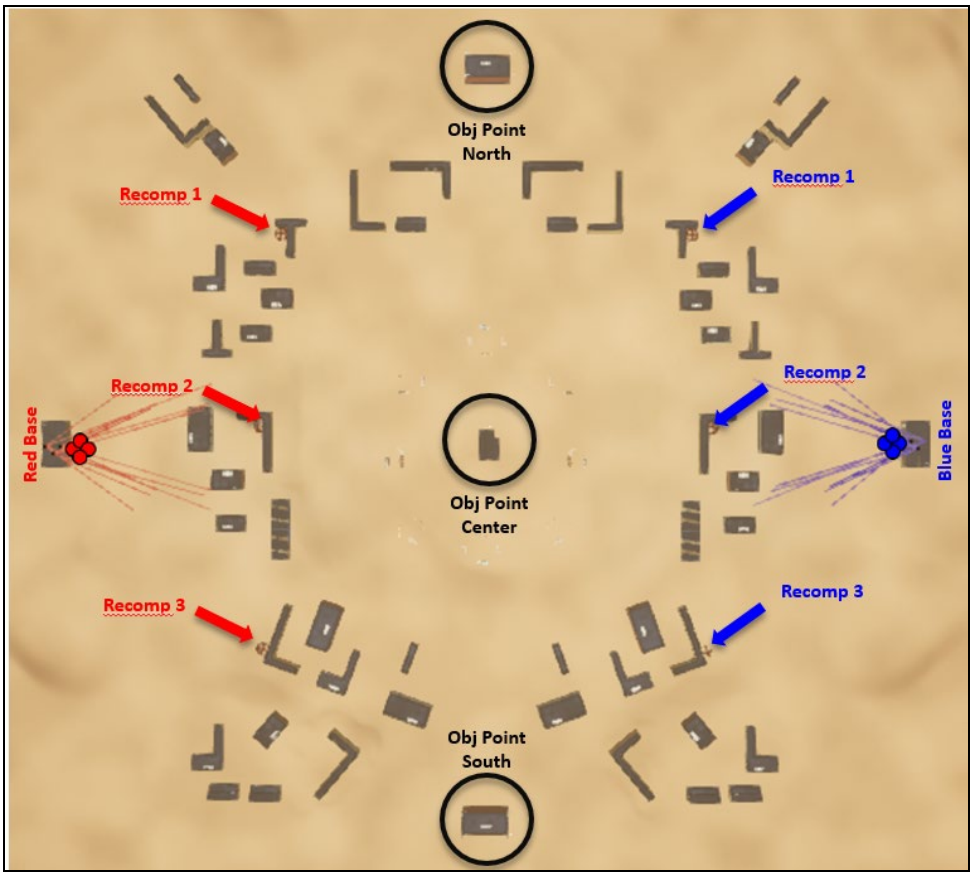


Figure 24. Overhead view of map developed for September / December 2019 tests

4.3 Phase III: August 15, 2019 – August 14, 2020

The first significant event of this period was the internal September 2019 playtest, see Figure 24, which as previously stated, was intended to serve as a dry run for the subsequent December 2019 test. It was also an opportunity to evaluate telemetry and data science capabilities to ensure the necessary data was being captured and could conduct solid analyses of that data afterwards.

In this event, participants received surveys prior to, during, and after the test, and were presented with instructions on how to play the game, Figure 25. Participants played four rounds of four matches each, totaling 16 games. Half of those 16 games isolated the ISR AI (ISR AI with no C2 AI) against a non-AI assisted team and the other half were played with both an ISR and C2 AI against a non-AI assisted team.

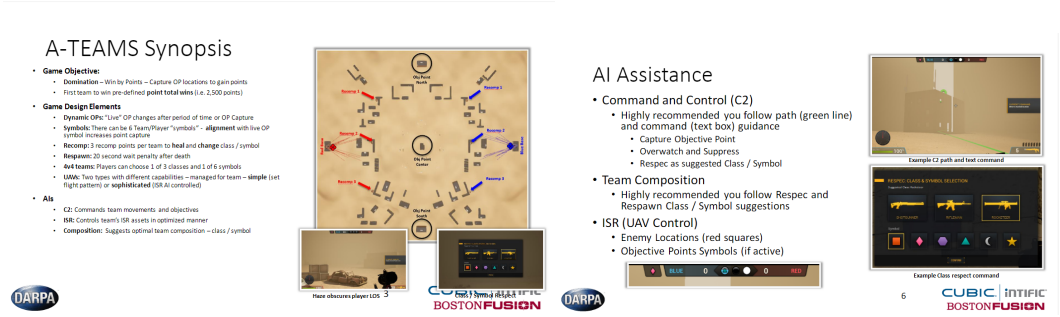


Figure 25. September 2019 test instructions

In this context of evaluating the testbed's ability to support a more formal experiment, the September 2019 test was successful. Despite being a smaller, internal test, the team were able to collect enough data to perform quantitative analysis on the results and draw some conclusions. The overall conclusions were:

1. Novice users benefited from any AI support.
2. Advanced users did not benefit from either C2 AI or ISR AI support, indicating the test design was too easy for advanced players.

Additionally, the team learned certain assumptions made during test design were completely incorrect. For example, the team assumed that the entirety of the map would be "in play" during the test. However, during the playtest, the team observed nearly all combat occurring at or between the three control points. While players transited through a large portion of the map, nearly all combat was contained in three specific areas. A majority of the buildings and terrain acted merely as scenery.

Another example relates to the intended design of the ISR platform and its tactical value to the player. It was originally assumed that players would find the ISR platform useful for scouting for opposition forces. In practice, this produced diminishing returns. Since all players fought at the control points, the real benefit lay in knowing which specific part of the building the opponent was occupying. With this knowledge, for example, a well-placed rocket could dislodge the opponent. This revealed a design flaw in the ISR AI: less movement, not more, allowed the UAVs to provide better information. An ISR AI optimized for sitting on top of control points, not scouting for enemy locations, could be more advantageous for this experiment.

The complete report of the analysis of this experiment can be found in *APPENDIX B: September 2019 Test Analysis*. A detailed description of the telemetry format can be found in *APPENDIX C: September 2019 Test Telemetry Details*.

Q4 2019

October through December 2019 was focused on two fronts. The first was preparation for a test event at MITRE's Tyson's Corner facility in conjunction with the December A-Teams PI meeting. The second was a testbed design effort which was targeted at a longer-term June 2020 test event.

During this period, the proposed December 2019 test protocol was submitted to the Institutional Review Board (IRB), Advarra IRB, and were notified that the protocol was exempt from IRB oversight.

The December test event was a refinement of the September test configuration, with gameplay changes designed to exercise AI options awareness and address unintended player behaviors. A summary of the changes in the testbed are shown in Figure 26.

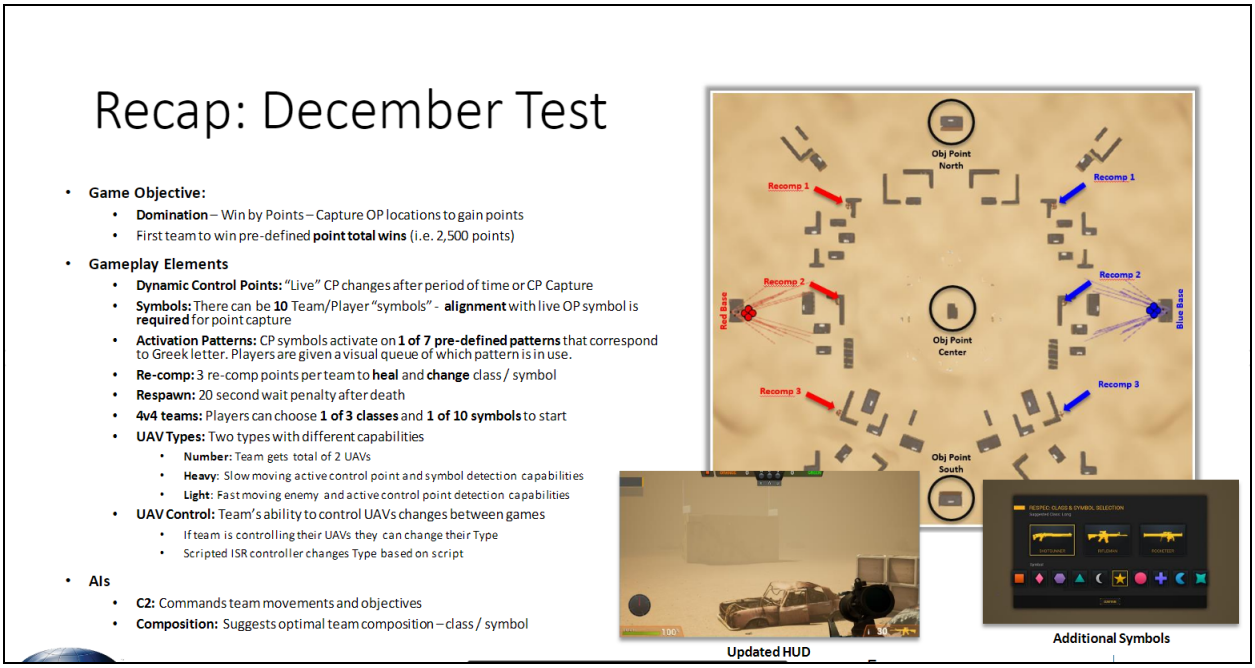


Figure 26. December 2019 Testbed Changes

The team also worked with Aptima to produce UI mockups that could be shown to a group of participants with guidance from the MITRE UI / user experience (UX) team. A few notable mockups are presented in Figures 27 and 28.

Figure 27. Mockup of top-down "routing" guidance provided by AI

The routing concept displayed in Figures 27 and 28 is inspired by popular driving apps with which users are assumed to be familiar. Presenting options to the user with metadata to show the supporting rationale for the “best” route and other similar routes reinforces the AI’s rationale to the user and in turn builds model trust. This top down view transitioning to a first person “executing the route view” was collaborated upon by both Aptima and Intific.

Figure 28. First-person AI routing guidance mockup

In conjunction with the University of Washington, the team also experimented with the idea of “nudges,” as shown in Figure 29. Nudges are subtle suggestions which the user can quickly accept or reject. UW’s long-term vision for nudges included simple image base signaling to which the user could quickly respond (or ignore).

Figure 29. Concept of an AI "Nudge"

Ultimately, the design discussions pivoted away from a first-person shooter game construct towards a “Plan-Play-Review” game where teams (and potentially teams of teams) would coordinate effects and resources against an opponent of similar size, as shown in Figure 30.

Figure 30. "Plan - Play - Review" concept

Despite iterating through numerous design ideas around this concept, by the end of the PI meeting, it was clear that incremental changes to the AT2B testbed were not viewed as substantial enough to meet program needs, and the team was challenged by the DARPA program manager to go "back to the drawing board" and fully reconsider how to address program goals. This effort would begin in earnest in January of 2020.

Q1 2020

The efforts from January through March of 2020 were focused on design of a re-imagined testbed which put increased focus upon team composition, planning, coordination and potential for cognitive load where human + AI teaming might prove itself to be beneficial.

Conversely, the team sought to minimize if not eliminate the effects of "twitch" skill, derived from mastery of hand-eye coordination, from the improvement equation. This enables the team to more clearly focus on a team's adaptation to new strategies, new challenges and their leverage of AI assistance without conflating the effects of a user's proficiency with this type of gameplay.

The specific program goals the HST testbed proposed to accomplish were as follows:

1. Ability to support teams ranging in size from 6-10 (small) to 50 (large)
2. Distribution A – stylized gamified representation of desired challenges and objectives with ability to make it more military relevant through slight art and parameter changes
3. Co-Evolving Learning and Expertise: Humans and AI learn from each other
    a. Humans learn different strategic and tactical options via nudges and predictive metrics. AI's learn by tailoring presented options to the idiosyncrasies of the team
    b. Waze-like team / human symbiosis – individual players report environment information which informs both team and AI, in turn AI provides better guidance to team and individual players
4. Determine if individual / team performance can be improved via AI augmentation and result in non-obvious outcomes
5. Testbed mechanics and scenario moves team through a cycle of:
    a. Understand Mission / Goal

b. Planning

    i. Individual Role / Behavior / Task assignment (Division of Labor) – who does what

    ii. Planned Grouping (team of teams) – how smaller groups of teams will work together

    iii. Planned Coordination (Integration of Effort) – how team plans to accomplish mission / goal

c. Execute

    i. Begin executing plan to accomplish mission

    ii. Face challenges which cause dynamic re-planning

        1. Role / Task allocation

        2. Grouping

        3. Re-consideration of mission / objective and coordination to accomplish

d. Review

    i. Team and Individuals interpret performance

    ii. Identify potential improvements

    iii. Prepare for Mission / Goal → Plan → Execute cycle again

6. Expose team to AI through UI / UX which enables:

    a. Rapid and individualized feedback - “nudges” for behavior & role changes, and skill boosting for collective performance improvement

    b. Automated adaptation of productive behavior space through roles and strategy discovery

    c. Role / behavior adaptation with changing objectives

    d. Dynamic temporal grouping and role assignment in response to environment changes

    e. Co-evolving learning and expertise: humans and AI learn from each other

7. Capture telemetry data to enable a clear measurement of success

The result of Cubic's design efforts was a digitized, card-based strategy board game. The overall objective for each game session was to complete as many objectives as possible against the opposing team in a competitive, turn-based, environment. Inspirations included Battleship[8], Robo Rally[9], and Slay the Spire[10]. The team called it Hide and Strike Testbed, a play on the phrase "Hide and Seek," as the gameplay mechanics encouraged players to hide their location, strike the enemy, and quickly move to avoid detection (and counterattack).

HST incorporated a "team-of-teams" structure in a competitive environment. A commander would control each team and be presented with limited information, see Figure 31; assigning and outfitting combatants across multiple arenas. Success would require communication and information was limited. Combatants would only be aware of their particular arena and even their information about their arena would be limited. This new testbed was turn-based to allow more deliberate decision-making time, but still designed to induce cognitive overload in players.

The system was also designed around the concepts of hidden and stale information that were hoped would encourage the application of human intuition and bluffing, commonly seen in games such as poker.

A game session was comprised of several rounds, each comprised of a planning phase and an execution phase. At the end of the game session, a total score is tallied, and the higher scoring team achieves victory.

Gameplay simultaneously occurred on 2 different arenas (with a theoretically unlimited number of arenas, thus supporting increasingly large teams). Each arena's environment was different, but each team's view of an arena was symmetrical (either reflected or rotated) to the other team's view. Players could see everything on their side of each arena, but nothing would initially be visible on the opposing side.

Each team in an arena could field up to 5 players, for a total of 10 potential player assignments per team across the arenas. A team consisted of 6 total players, one of which is the commander, leaving 5 combatants to be assigned by the commander across the arenas.

Figure 31. UI Concept of HST Commander View

Each arena had a main objective that “wins” the arena and was worth the most points when tallying final scores. If a main objective was completed, the arena would be closed, and all players assigned to the arena must be reassigned to the other arena on the next planning phase.

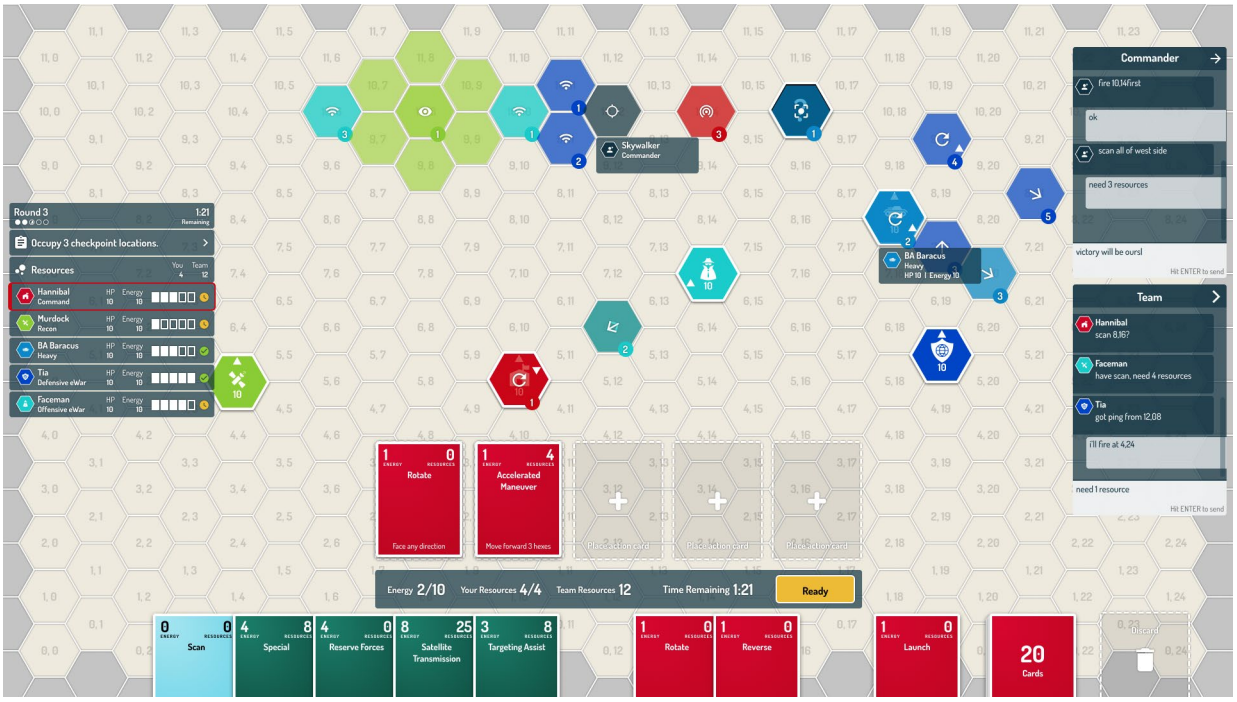


Figure 32. UI Concept of HST Combatant View

Each arena also had a time limit governed by a "chess clock" that would decrement as the teams are planning. When a commander executed his team's actions for that arena, the team's clock would stop, and all planning by that team would be locked. If either team's clock ran out of time during a match, that team would lose that arena by forfeit. This mechanic made efficient planning a key priority and enforced pressure on commanders.

By March 2020, design was complete on the HST, UI concepts were completed, see Figure 32, and tasks were decomposed, establishing an aggressive development timeline that was to culminate in an internal validation event prior to the June PI meeting.

The HST architecture would be built on scalable, commercial cloud backend technology and the client would run as a Web Graphics Library (WebGL) context inside a web browser, enabling nearly anyone with a modern browser and an internet connection to participate (and generate data).

The team believed this "thin client" approach would enable an experiment to achieve statistically significant results, and in the future, the research could potentially build a persistent experimentation platform akin to the crowdsourced protein folding game platform "foldit," shown in Figure 33, that could be used to rapidly design, deploy, test and iterate over new hypotheses.

Figure 33. Foldit crowdsourced protein folding game platform

Q2 2020

During this period, the team began to develop the new HST testbed, Figure 34, in earnest, targeting a late May / early June test validation event prior to the A-Teams June PI meeting. The goal was to demonstrate that the core premise of this new testbed was viable and to attempt to run a small test and collect meaningful data from it, akin to efforts conducted previously in 2018 on the AT2B testbed.

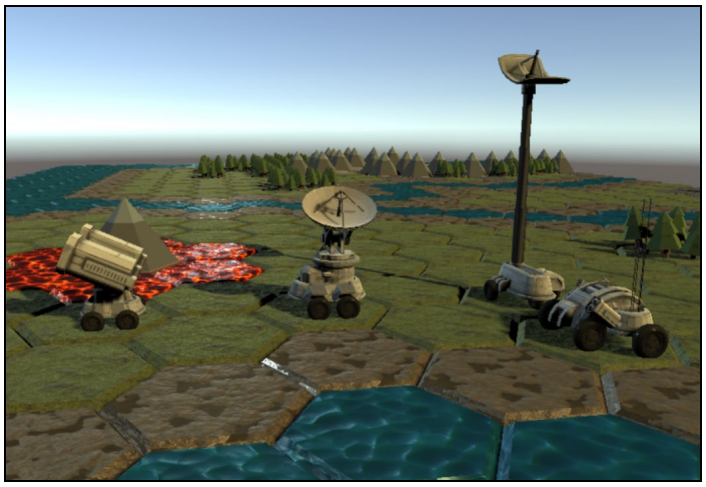


Figure 34. Artist rendering concept of HST testbed environment

Over the eight-week period from April through May, the team iterated continuously on game systems, Figure 35, and technology, Figure 36, and were able to achieve most of the engineering goals. Some systems, such as the “chess clock” timer, had to be scoped out of the effort, and other elements of the game design were implemented in only a partial fashion (the commander role, while functional, was particularly limited). The team worked closely with Aptima and MITRE throughout this process to maximize the potential experimental value of the testbed. The team also began to work on the next set of HSR test protocols the team would need to run a true experiment with this new testbed in the fall.

Figure 35. Early April prototype of HST hex-grid environment, units and card system

Aptima would not be able to implement a functional AI for this new environment in time for the validation test but designed a selection of surveys that queried players about their level of confidence in enemy locations which served as a proxy for functions an AI would eventually perform. This would allow the team to measure player sentiment which could provide interesting insight.

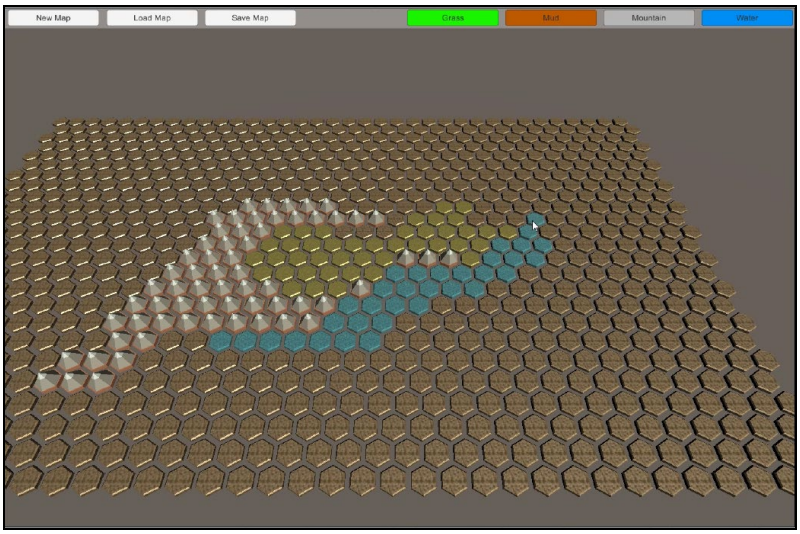


Figure 36. Map editor functionality for creating new arenas

By the end of May, the following systems had been built, deployed and tested, Figures 37, 38 and 39:

- Backend Amazon Web Services (AWS) hosted multiplayer gameplay server with MongoDB database
- Frontend Unity3D-based WebGL client (runs in standard browser)
- Continuous Integration / Continuous Deployment (CI / CD) system implemented and integrated; development and staging environments established and daily builds / smoke tests automatically deploying
- Multiple players, roles and classes supported
- Round-based, data-driven card action system with action points, special actions, deck shuffling
- Full, parseable telemetry system to capture all player actions and world state
- Basic environmental change implemented into the testbed to force new tactics and strategies
- Rudimentary UI / UX including visualizations of player scans, attacks and damage sustained

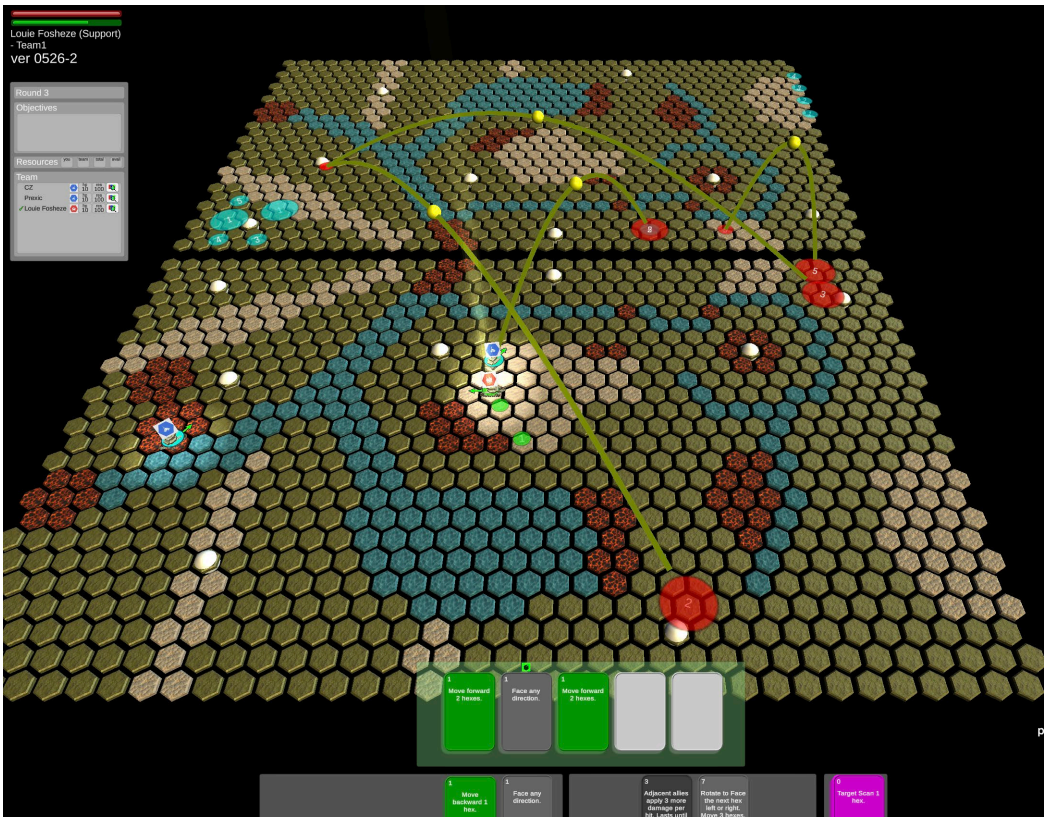


Figure 37. Final HST client as of May Validation Test

Figure 38. Multiple HST clients in browsers showing Combatant and Commander views

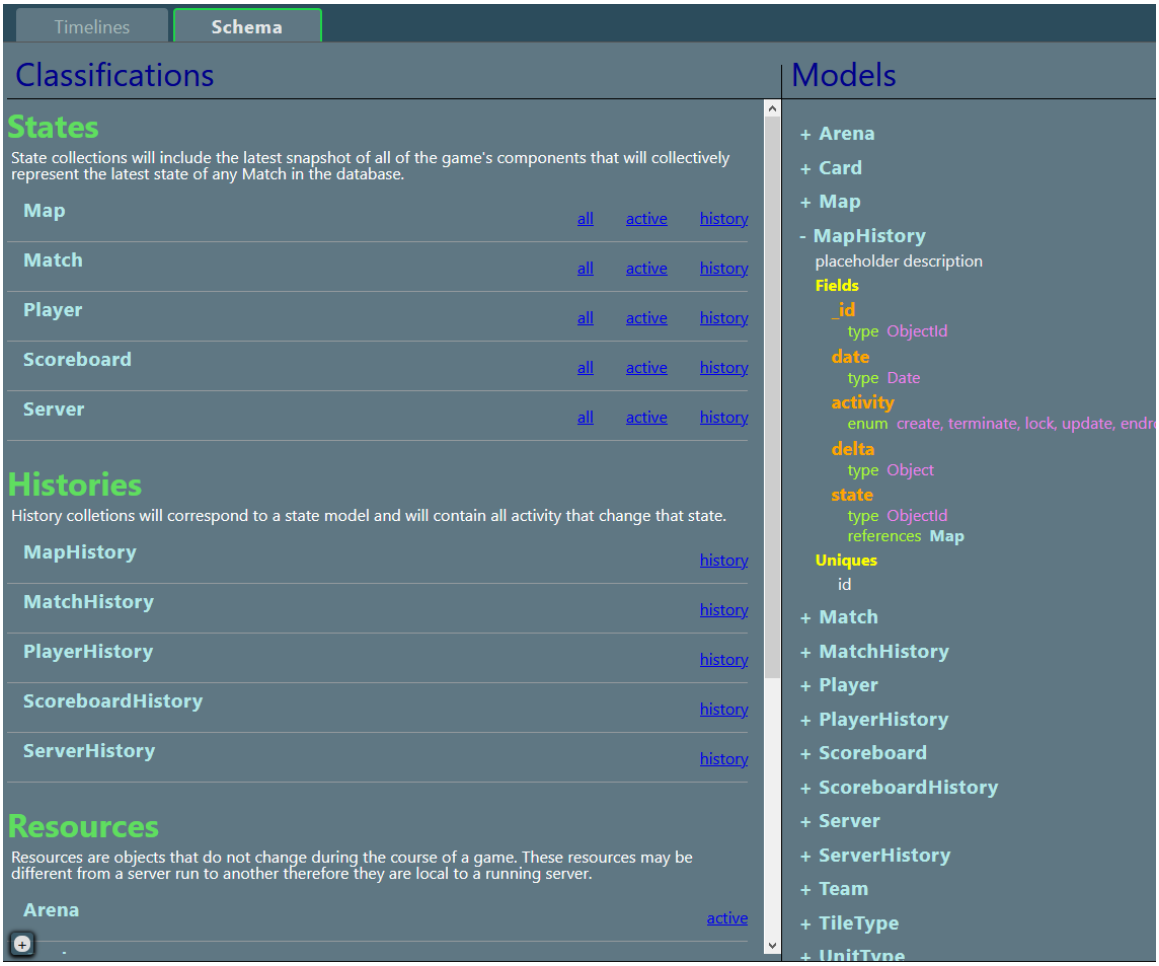


Figure 39. Integrated telemetry browser UI

May Validation Test

The final test the team completed with the new HST testbed was the Validation Test. This test, like many previous tests the team conducted with AT2B, was conducted internally with Cubic employees. One significant difference, however, was that this test was the first the team had conducted entirely virtually, due to the COVID-19 lockdown.

However, the web-based thin-client strategy the team adopted with the HST architecture was well-suited for these constraints and the teams were able to play for 6 hours and 49 minutes over the course of four days:

- 6v6 format: 1 commander + 5 combatants per team (in most games)
- 16 games completed
- 22 unique players – No Personal Identifiable Information (PII) collected
- 108 rounds of gameplay totaling 6 hrs., 49 minutes of play

Because the team had not yet been able to implement a dedicated in-app chat system, all communications during the test were conducted in a dedicated, anonymized Slack instance, and teams were not allowed to eavesdrop on each other's chat rooms. This afforded teams some level of strategizing. Additionally, prior to each session, at the end of each game, and every four gameplay rounds, players were prompted to complete a brief external survey (via a hyperlink) which was designed by Aptima.

At the conclusion of each day, all telemetry and chat logs were provided to Aptima to be analyzed alongside the survey results.

PI Meeting and Conclusion of Effort

On June 1, 2020, the A-Teams PI Meeting took place. Updates were presented, as did each of the other performers, many of whom were significantly further along in their progress toward demonstrating program goals and could demonstrate it through experimental results. This was to be expected heading into the final year of the program.

This was in stark contrast to the platform, which had only been conceptualized eight weeks prior, but which the team had been able to scaffold, construct, and validate in a prototype fashion. The team was and is proud of efforts made on behalf of the program but recognized that the technical lift and costs involved with completing the development and testing of the new platform were simply too high, and could not determine a path forward that was not exceptionally risky.

The team felt duty bound to communicate these concerns with the DARPA PM early to allow him maximum flexibility, and after a brief discussion, he concurred that although our progress on the new testbed had been impressive, the climb remaining would simply be too risky and the team should not proceed with the final phase of the program.

5.0 CONCLUSIONS

5.1 Conclusions for Robust ML Testbed Development

Fast Execution Server. It is ideal while developing a testbed to take into account the faster than real time requirements for ML / RL rapid training. Ideally, a server that can be capable of 100x to 1000x speed for training purposes would greatly reduce training cost. Be aware, however, that many ML / RL AI's are written in the scripting language Python, which by nature is not optimized for performance. In those cases, the AI code itself will be the bottleneck.

Robust / Fast API. Another potential bottleneck for execution is the API between the testbed and AI code. This layer needs to be simple and straightforward to use; however, the more universal this layer is, the larger the potential for parsing slowdown to occur during training.

Robust Telemetry System. All game state information must be saved and stored during execution for post-test analysis. A feature of this system should include the ability to rewind to any state and fork execution to determine different outcomes. This process could greatly simplify the AI training.

Challenging tasks. The testbed should be capable of posing problems with challenging solutions. Examples may include hidden Information that needs to be uncovered, stale information that requires deduction of action state, and deterministic data that requires forecasting of potential future action state.

5.2 Conclusions for Machine Learning AI Development

Trust. Trust in AI recommendations must be established. Although the AI recommendations during testing were poor, that wasn't necessarily apparent to users. However, most users would ignore AI commands if they had any experience and the AI recommendation did not fall within their experiential expectation. If the AI recommendations could lead to novel strategies, a trust relationship must be built over time initially starting with strategies that fit into user expectation. Another option would be exposing users to the variables and rational behind novel recommendations. Blind obedience was rarely observed.

Co-learning. One potential solution to the AI trust problem would be a robust co-learning model. If the system learned player preference, it could recommend action that fits within parameters that match said player preference. Once a trust relationship is established it would be much easier to exert influence over the user via subtle adjustments to recommendations that eventually change behavioral strategy.

Layered AI. A recommended approach to AI in a complex environment such as the final HST testbed would be layered task specific AI that could be used independently or in tandem. Here is the high-level approach the team at Intific would have used to develop the system:

- Card Counting AI: This AI would focus on using known data to determine which cards have been seen / used vs what could remain to narrow possibility action space with statistical analysis. This structure could be used easily for user hands and with lower accuracy on enemy hands.
- Player Categorization AI: This AI would use play style, specifically identifying aggressive vs stealthy on a scale for use in player action prediction.
- Enemy Position AI: This AI would build a probability heat map trying to determine enemy unit locations. It would begin with initial ping info on enemy positions and then use terrain and future pings to narrow the probability of position. In tandem with a card counting AI and player categorization, it could narrow that action space.
- Commander AI: This AI would handle unit assignment to arena and class meta's to cover objectives. It would also deal with high level objectives and reassignment options. To train probably would require numerous games with history and win / loss ratios.
- Combatant Advisor AI: Using the knowledge of all previously mentioned AI, this module would use the predicted options for a player along with commander high level goals and enemy position information to give recommendations on patterns of user action (set of cards in order). This would filter by type and could use player aggressiveness level to offer strategies that initially align, but gradually nudge to align with more radical and interesting behaviors.

6.0 REFERENCES

1. The Scrum Alliance, "The Scrum Framework",
   URL: https://www.scrumalliance.org/about-scrum/framework

2. The ZeroMQ Authors, “ZeroMQ | Get Started”
   URL: https://zeromq.org/get-started/

3. The Rust Team, “Learn Rust – Rust Programming”
   URL: https://www.rust-lang.org/learn

4. Docker, Inc., “Docker Documentation”
   URL: https://docs.docker.com/

5. The Jenkins Project, “Jenkins User Documentation”
   URL: https://www.jenkins.io/doc/

6. The Kubernetes Authors, “Kubernetes Documentation”
   URL: https://kubernetes.io/docs/home/

7. SideFX, Inc., “Getting Started | Learning Houdini”
   URL: https://www.sidefx.com/learn/getting_started/

8. Wikipedia.org, “Battleship (game)”
   URL: https://en.wikipedia.org/wiki/Battleship_(game)

9. Wikipedia.org, “RoboRally”
   URL: https://en.wikipedia.org/wiki/RoboRally

10. Wikipedia.org, “Slay the Spire”
    URL: https://en.wikipedia.org/wiki/Slay_the_Spire

APPENDIX A: MAY 2019 TEST ANALYSIS OF C2 AI PERFORMANCE

It was clear during the exercise that the C2 AI was still in its infancy. It had yet to develop a coherent strategy and failed to garner trust from participants. A few brief takeaways are discussed below showing the progress and potential of the C2 AI. These metrics are intended to serve as one way of evaluating the AI's impact in future assessments.

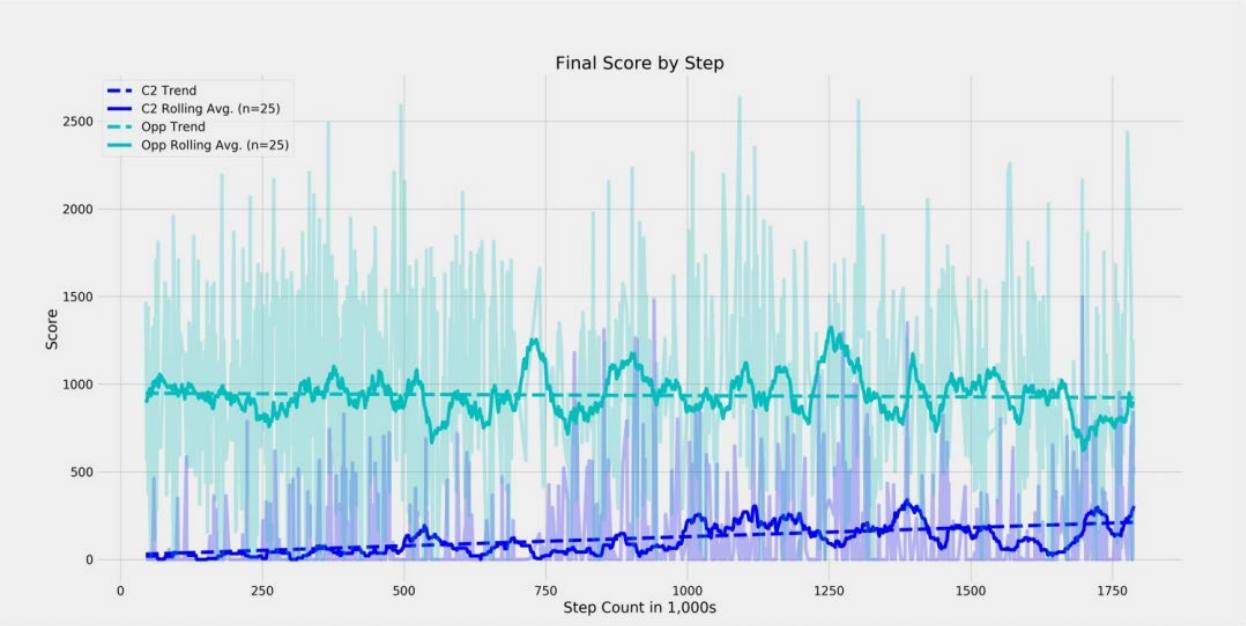


Figure 40. A comparison of the AI's game score against the aggressive random agent it trained against. As expected, the random agent achieves a near constant score. The AI begins to reduce the random agent's score as it increases its own score.

Figure 40 shows the score throughout training of both the AI and the random agent it trained against. While the number of iterations the AI completed was paltry relative to the task's difficulty, it managed an upward trend during training.

Figure 41 shows the training curve for DeepMind's most recent capture the flag agent. While DeepMind's AI was trained to play the game as an individual rather than orchestrate strategic positioning, it can still serve as a useful benchmark. While the C2 AI played less than 2,000 games, it took DeepMind's agent over 100,000 games to reach an average human's skill level. Upgrading the testbed for AI training performance and improving the training regimen will greatly increase the AI's performance ceiling.

Figure 41. The training curve for DeepMind's capture the flag agent (https://deepmind.com/blog/capture-the-flag-science/)

In the C2 AI's naïveté, it consistently sent units to the same locations. Keeping units grouped together allowed it to control areas of the map yielding consistently high kill-to-death (K / D) ratios as shown in Figure 42.

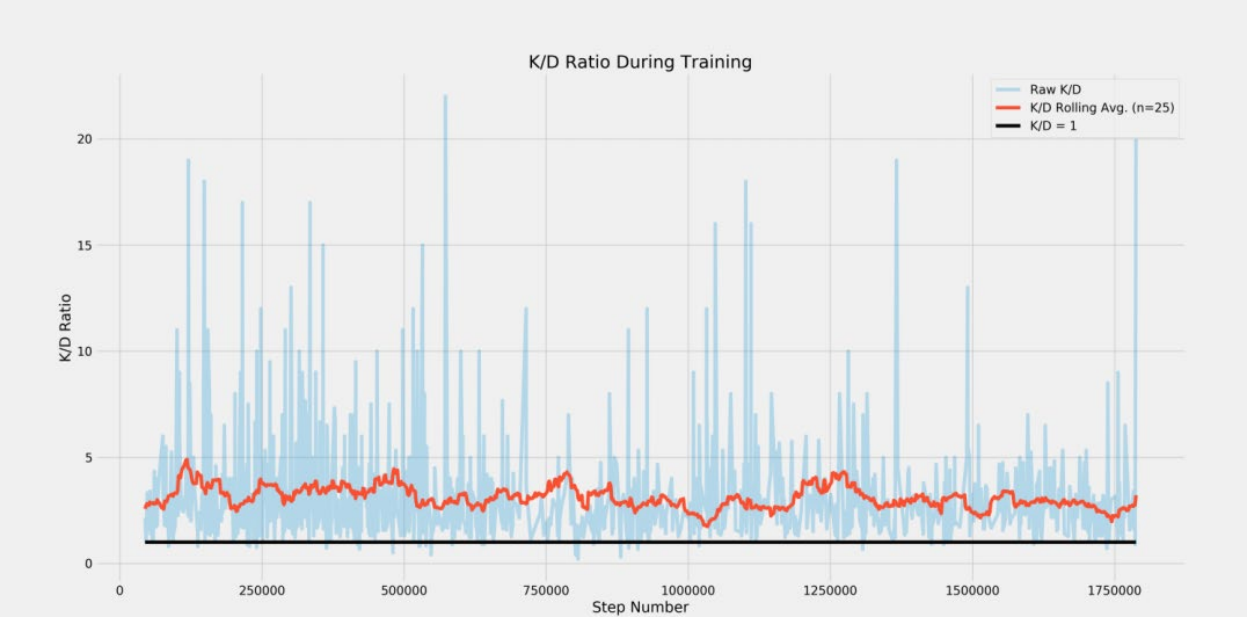


Figure 42. Despite not developing a strategy to consistently capture control points, the AI is able to achieve a high kill-to-death ratio. K / D ratio is a common team death match benchmark.

The AI exhibited a preference for sending units to control nodes 0 and 252, a result of clipping the action space early on in training. Since those nodes were inherently difficult to reach and occupy due to their locations, the lifespan of C2 AI controlled units was noticeably lower than their human C2 controlled counterparts. Figure 43 shows the lifespan of C2 AI commanded players during the demo event grouped by team. The C2 AI, not surprisingly, caused the lifespan of players to decrease with the exception of Team X. Team X was subjectively deemed the most novice team in terms of familiarity with video games. This may indicate the C2 AI helped guide novice players who were otherwise playing aimlessly without a formal commander.

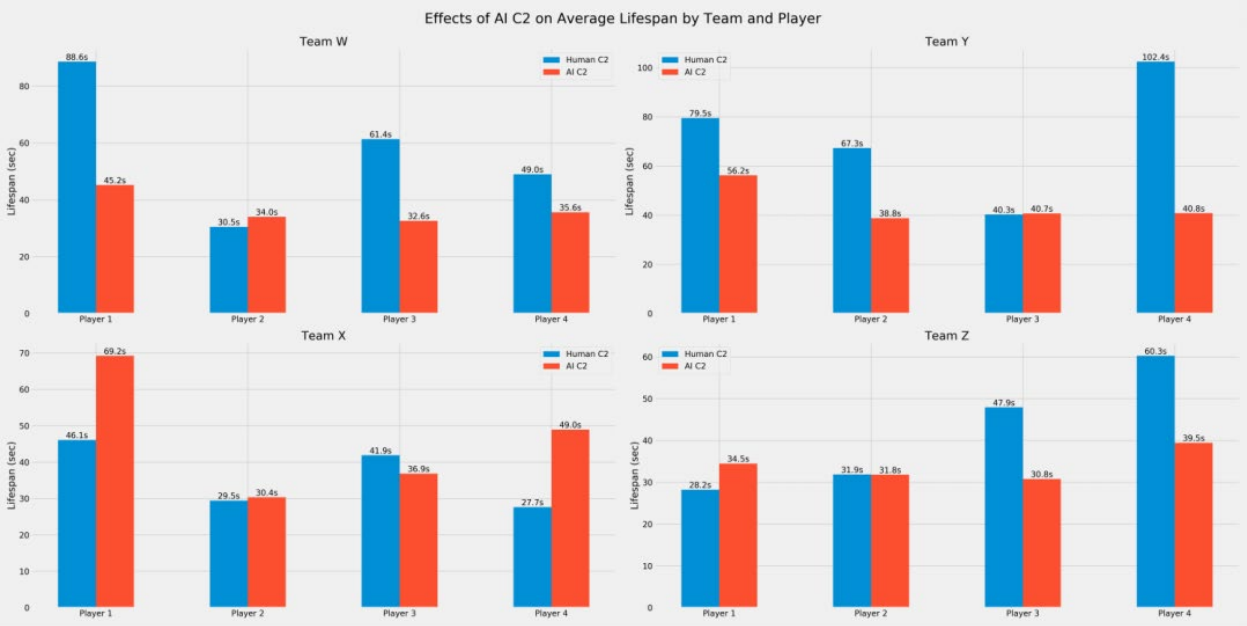


Figure 43. Each team and unit's average lifespan with and without a C2 AI

Figure 44 through Figure 46 show each player's and the team's average lifespan plotted as lines along with the team's score shown as bars for each round played. Results were separated by match to keep the players on each team constant (since some players left the event early). Matches 5 and 6 were grouped together since they were played back to back like matches 2 and 4. There does not appear to be a connection between lifespan and game score.

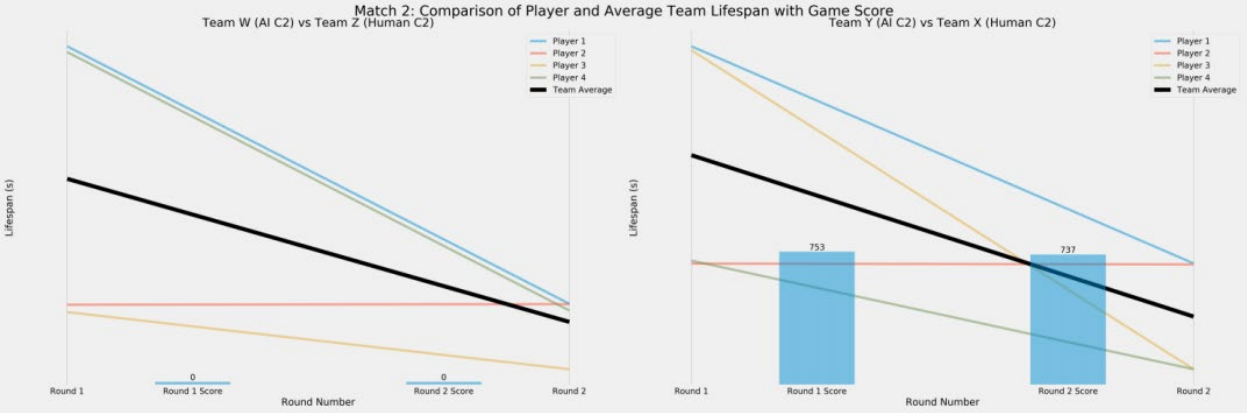


Figure 44. The lifespan (lines) of each player and the team's average along with the team's score (bars) for both rounds of match 2

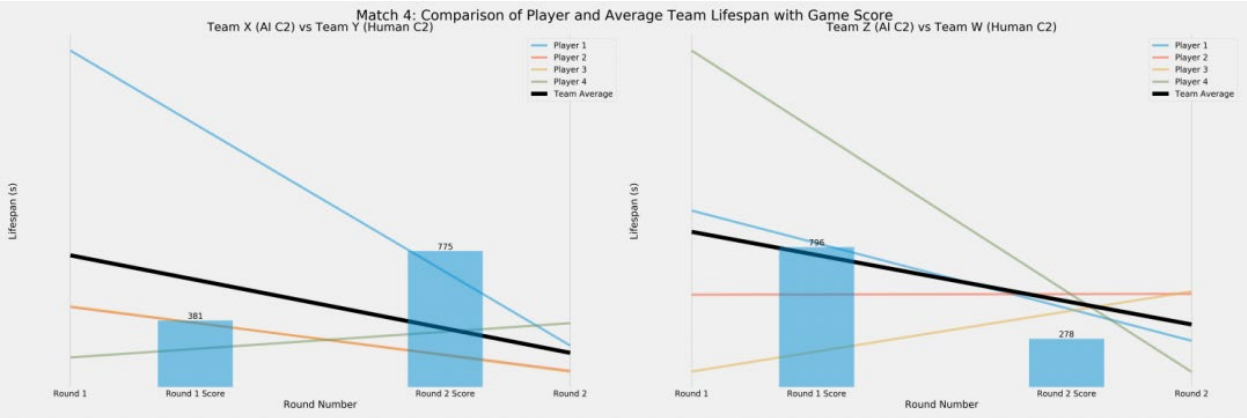


Figure 45. The lifespan (lines) of each player and the team's average along with the team's score (bars) for both rounds of match 4

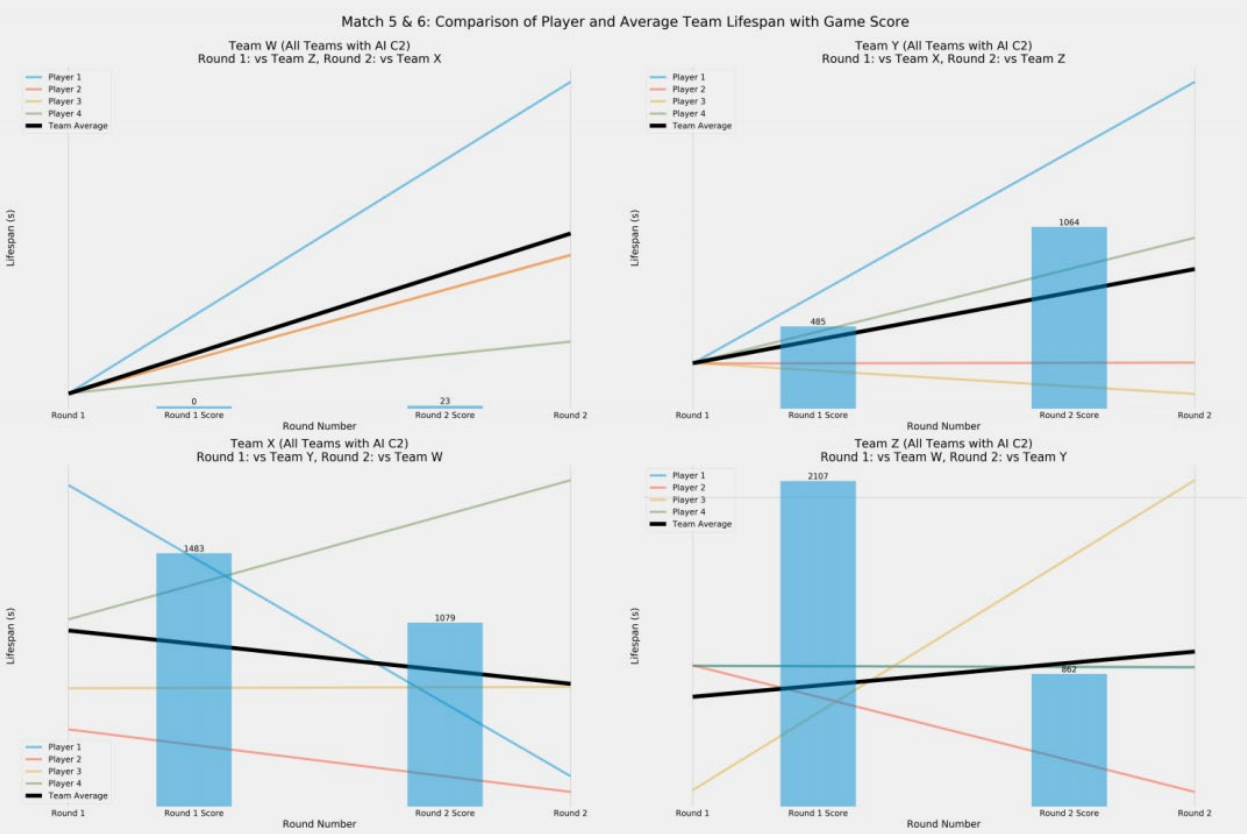


Figure 46. The lifespan (lines) of each player and the team's average along with the team's score (bars) for matches 5 and 6.

Figure 47 through Figure 49 are structured the same as the above figures, but show adherence to the commands issued by the C2 AI. Adherence is measured by how close a player was to the objective location at the time of death or issuing of a new command. A positive trend in adherence indicates the team did a better job of carrying out the commands. Teams W and Y show a weak correlation between adherence to the C2's commands and game score. This correlation can be explained by many other factors and an improvement in game score is not necessarily a result of following the C2's commands. In most cases, this finding is inverted for teams X and Z suggesting the previously mentioned correlation is even more tenuous.

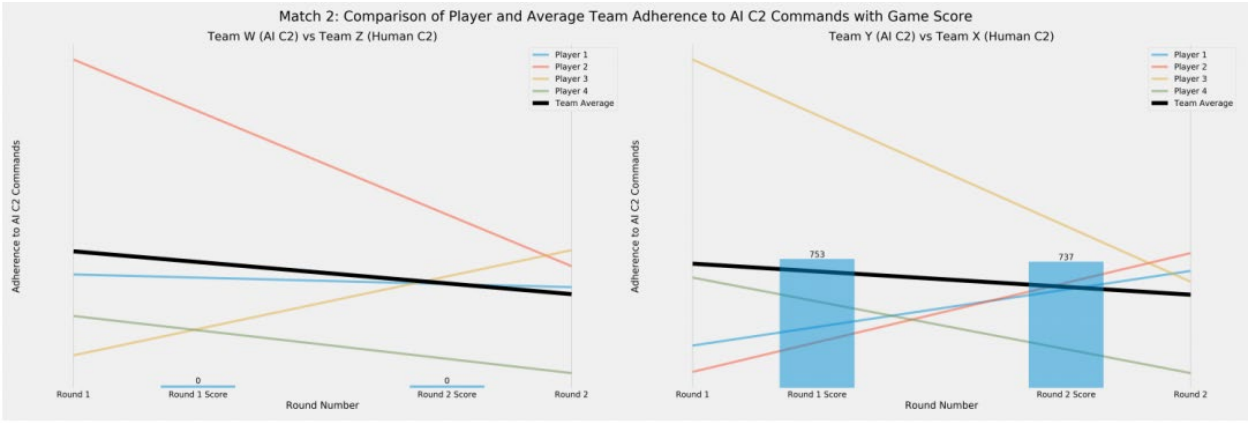


Figure 47. The adherence to C2 AI commands (lines) of each player and the team average along with the team's score (bars) for both rounds of match 2

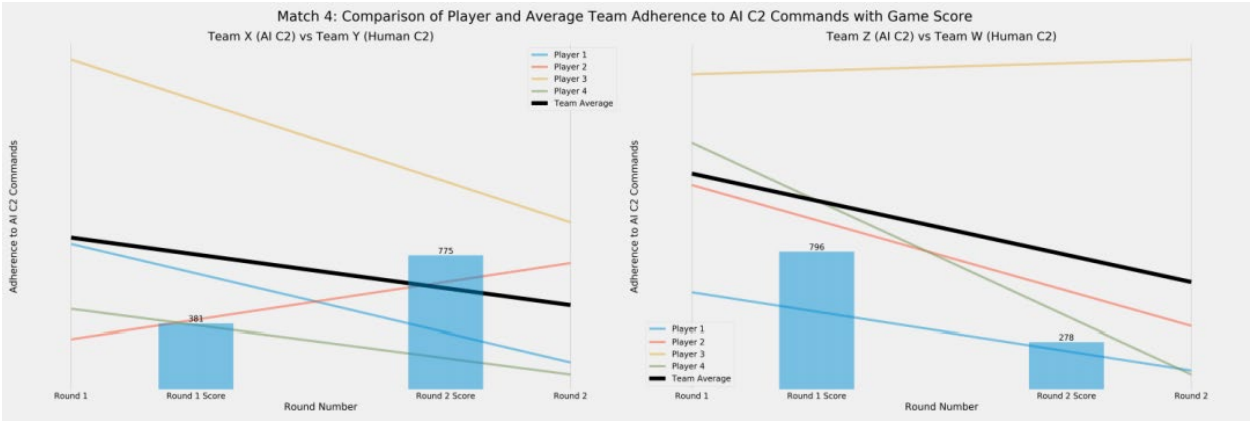


Figure 48. The adherence to C2 AI commands (lines) of each player and the team average along with the team's score (bars) for both rounds of match 4

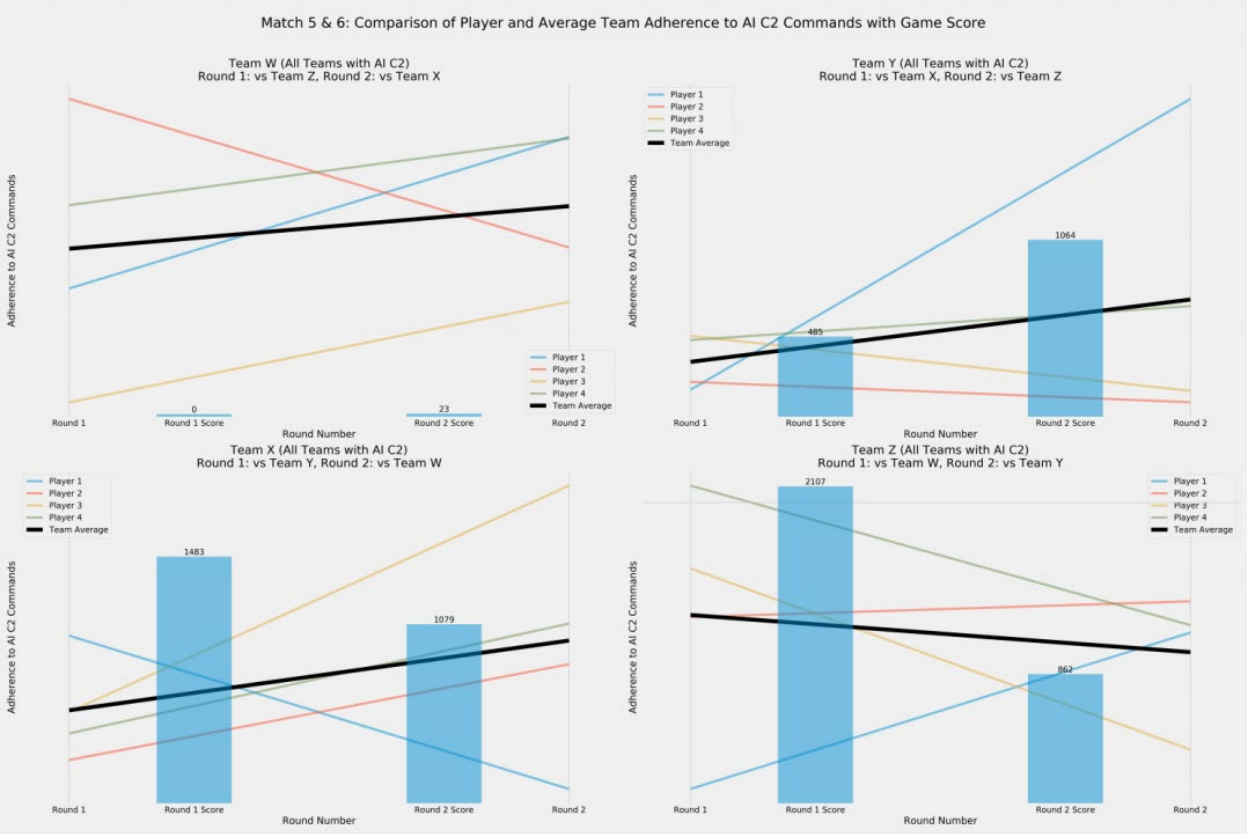


Figure 49. The adherence to C2 AI commands (lines) of each player and the team average along with the team's score (bars) for both matches 5 and 6

When examining the trends in adherence and lifespan presented in the previous figures, which are shown together in Figure 50 through Figure 52, the team's average lifespan was correlated with their adherence to the C2 commands, with the exception of Teams X and Z in matches 5 and 6. It is possible lifespan was increased because the C2 AI sent players to the same place giving the teams a numbers advantage. This could be a reflection of the behavior seen in training, the C2 AI achieves a higher K / D ratio by keeping units in close proximity leading to a higher unit lifespan.

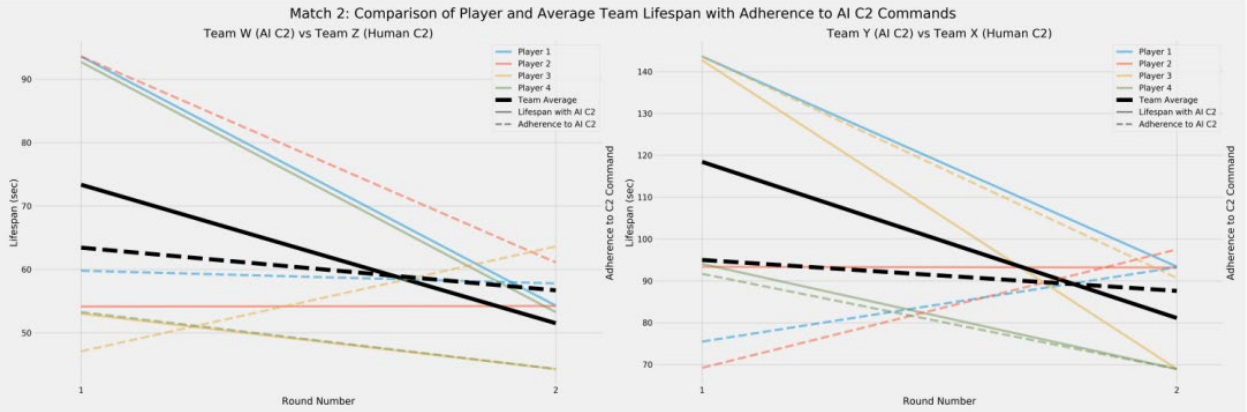


Figure 50. The lifespan of each player and the team average is shown by the solid lines and similarly the adherence to C2 AI commands is shown by the dashed lines for match 2

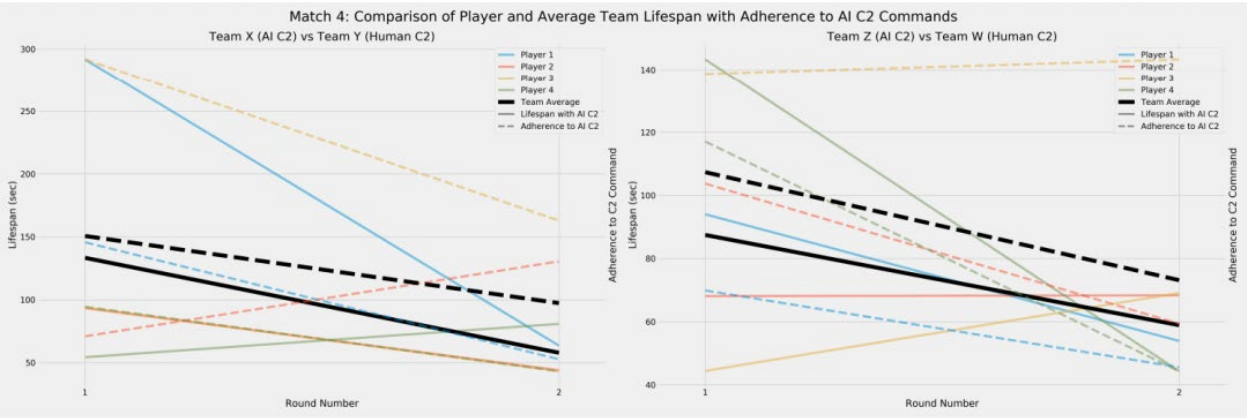


Figure 51. The lifespan of each player and the team average is shown by the solid lines and similarly the adherence to C2 AI commands is shown by the dashed lines for match 4

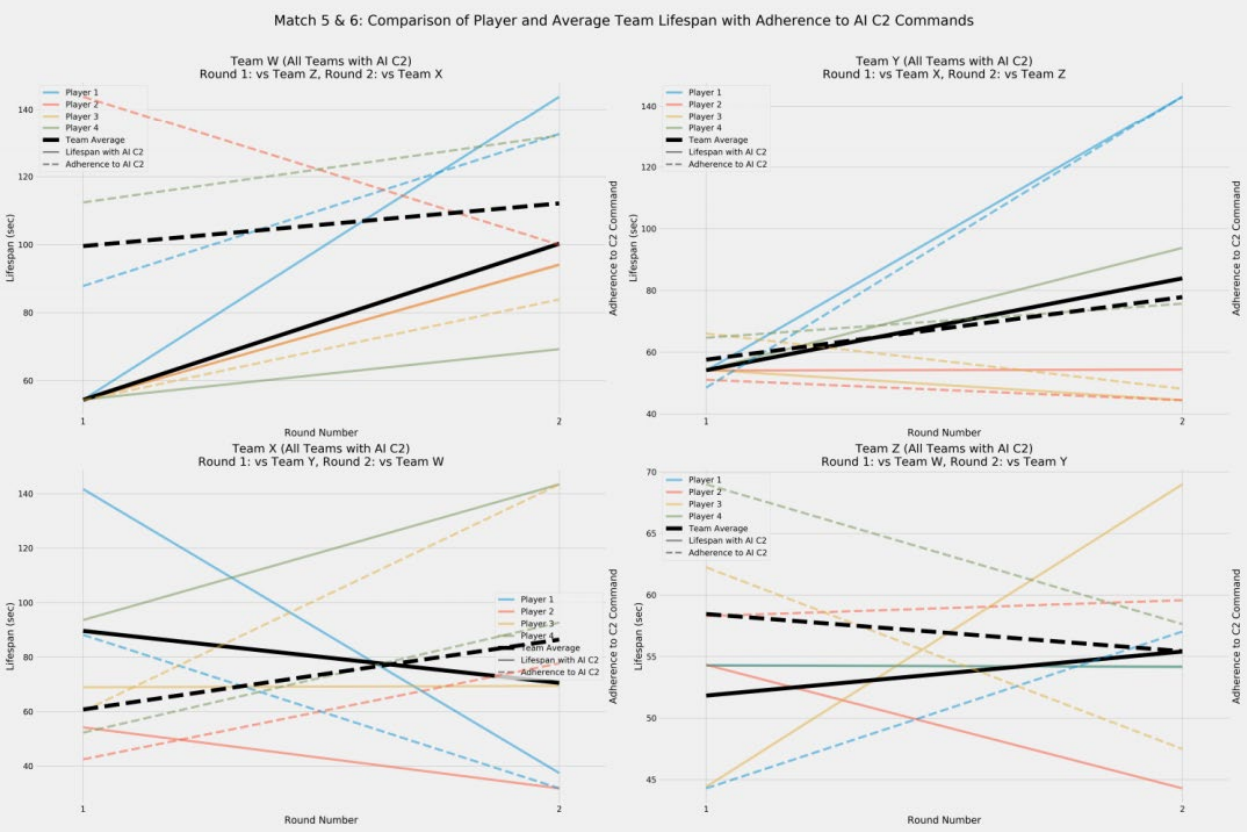


Figure 52. The lifespan of each player and the team average is shown by the solid lines and similarly the adherence to C2 AI commands is shown by the dashed lines for matches 5 and 6

APPENDIX B: SEPTEMBER 2019 TEST ANALYSIS

Summary and Qualitative Observations

This document provides a brief overview of the September 2019 AT2B playtest. Qualitative observations noted below lead to and are confirmed by quantitative measurements.

The team determined neither AI composition, ISR with C2 and ISR only, benefited expert players. The discussion below can be summarized by saying experts did not need additional guidance; players with sufficient knowledge brought advanced strategies and were able to exploit their opponent's weaknesses faster than the AI's.

This suggests pursuing a co-learning AI strategy where the AI is able to leverage advanced human strategies to guide players might have the potential to yield better results. The team can assume skilled human players will exhibit sophisticated strategies and tactics. Useful AI must learn from these strategies. An AI capable of recognizing when a team's approach fails and suggesting a new tactic or helping a team further exploit their current strategy may create an advantage for AI-assisted teams.

The team conducted four follow up experimental runs with novice level players. These players were selected for their inexperience with video games, first person shooters, and the AT2B testbed. One team played with AI assistance against a team of similar skill without AI assistance. After two games, the AI was swapped between teams. The team receiving AI assistance won 3 out of 4 games. While this is a small sample size, it does indicate that novice players benefit from AI assistance.

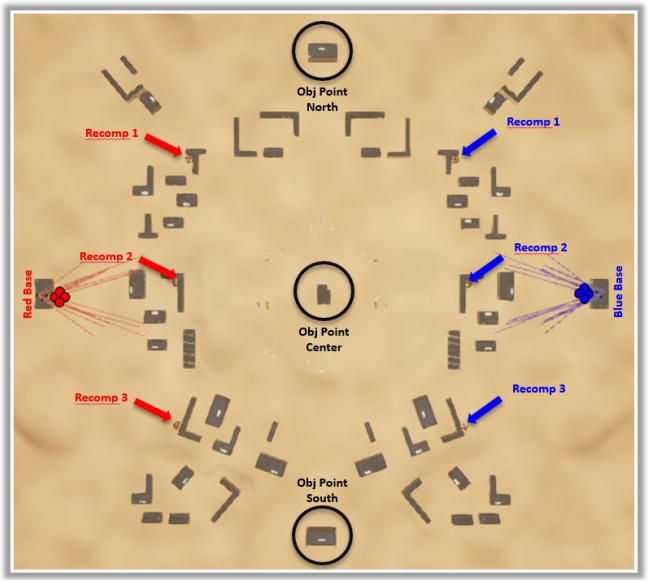


Figure 53. A-Teams September 2019 Testbed Play Area Map

During the playtest, see Figure 53, the team observed nearly all combat occurring at or between the three control points. While players transited through a large portion of the map, nearly all combat was contained in three specific areas. A majority of the buildings and terrain acted merely as scenery.

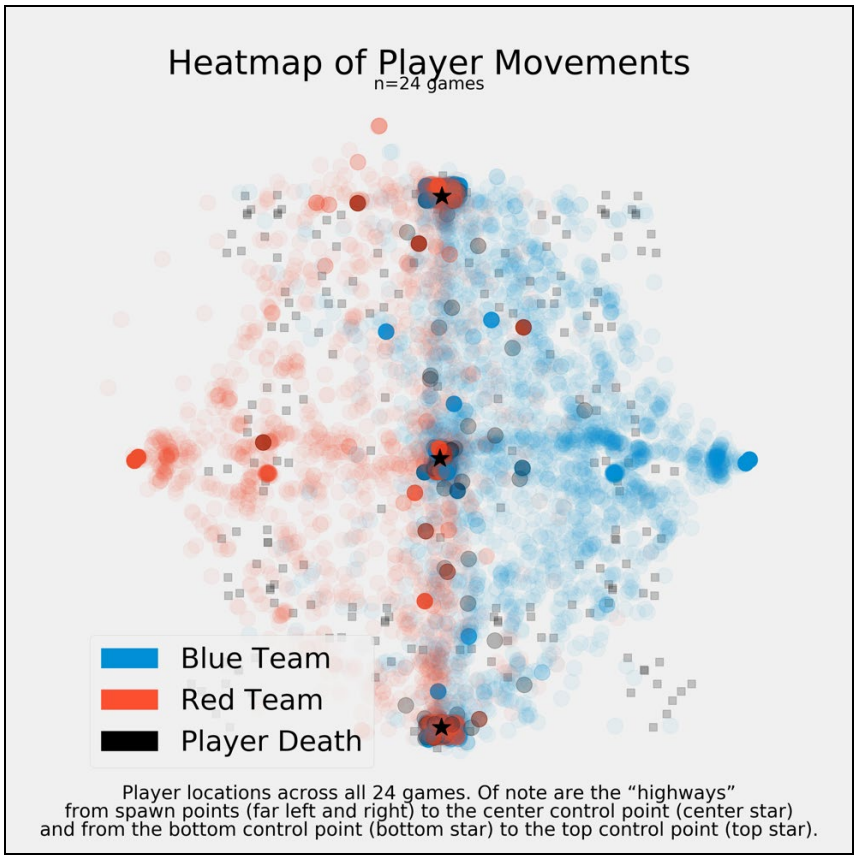


Figure 54. Heatmap of Player Movements

It was originally assumed the ISR platform would be useful at scouting for opposition forces, see Figure 54. In practice, this produced diminishing returns. Since all players fought at the control points, the real benefit lay in knowing which specific part of the building the opponent was occupying. With this knowledge, for example, a well-placed rocket could dislodge the opponent, see Figure 55. This revealed a design flaw in the ISR AI: less movement, not more, allowed the UAVs to provide better information. An ISR AI optimized for sitting on top of control points, not scouting for enemy locations, could be more advantageous for this experiment.

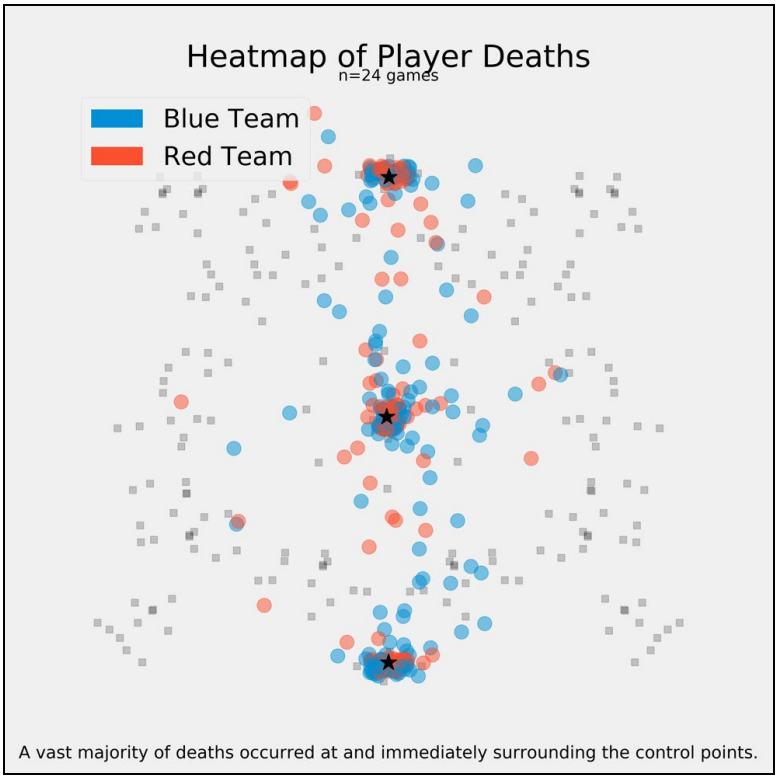


Figure 55. Heatmap of Player deaths

The team also noticed players were able to find the next active control point just as reliably as the ISR platform. Players would often split the team sending one group to capture the active point and the remaining members to find the next active control point.

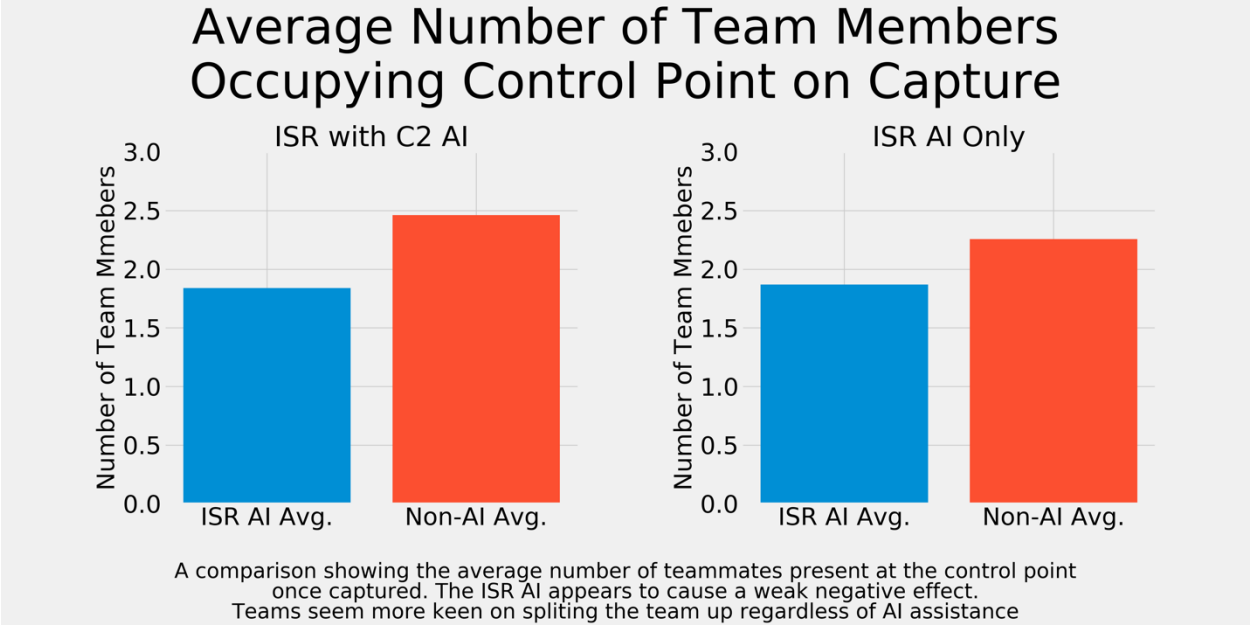


Figure 56. Average number of Team Members Occupying Control Point on Capture

There are two probable explanations. First, the light UAVs were only slightly faster than the players so they could not obtain and relay intelligence (intel) in time for the team to adjust their strategy. Second, capturing the control points was always more important to the teams than maximizing points scored per opportunity. There was no incentive for the team to wait for intel from the ISR before acting.

This meant teams usually opted to split up to prevent the opponent from capturing a control point. This also increased their odds of capturing the next control point even if they sacrificed maximizing the points they scored. This behavior discouraged symbol re-composition. Conversely, since symbol intel was rare, it is possible players would have been more eager to respec with better intel and thus maximize points scored. Lower points scored per control point captured may have resulted from a lack of intel.

Quantitative Measurements and Findings

Before exploring individual game metrics, the team sought to determine if the ISR AI had a statistically significant impact by calculating the p-values of the AI configurations. Calculation of p-values used the difference between the teams' final score in each game. Since the sample size was limited, the team assumed the distribution of final score differences are normally distributed with unknown variance (with each team combination playing against each other the expected difference in final score should be symmetric about the mean). To reject the null hypothesis requires, on average, the mean difference in final score to be non-zero, a positive difference indicating AI

assistance improves team performance and a negative difference the opposite. Otherwise, the team cannot definitively reject the null hypothesis.

Table 2. P-Value / Test Results

Test Structure	P-Value	Null hypothesis rejected?	Average Score Difference
Intific's ISR AI	0.36	No	-157.0
Aptima's C2 AI	1.83e-6	Yes	-2101.5
C2 AI & ISR AI	1.77e-5	Yes	-2104.5

It was determined the ISR AI had a statistically weak negative impact (assuming an alpha of 0.05 to account for the small sample size of each test) on a team's performance, which will be discussed in more detail later. The p-values of Aptima's C2 AI and the pairing of the C2 and ISR AI are also shown in Table 2 for comparison. The p-values of those latter two combinations clearly indicate a rejection of the null hypothesis.

Next, the team evaluated whether the combinations of the AIs (ISR, C2, and ISR & C2) produce the same effects on player gameplay. This is done by calculating the one-way analysis of variance (ANOVA). The resulting p-value is 1e-4, rejecting the null hypothesis and indicating that at least two of the means of the final score difference are different.

To determine which test structures vary differently the team performed a pairwise comparison. The results are shown in Table 3. In this case, rejecting indicates the two test structures differ significantly in score difference.

Table 3. Adjusted P-Value / Test Results

Test Structure	Adjusted P-Value	Reject?
C2 AI vs. ISR AI	0.001	Yes
C2 AI vs. C2 & ISR AI	0.9	No
ISR AI vs C2 & ISR AI	0.001	Yes

This confirms qualitative observations and player feedback that the C2 AI had statistically significant negative impact and the ISR AI had a plausible weak negative

impact on player performance. The focus of this assessment will now shift to a discussion of ISR AI game metrics.

A presumed near-optimal ISR strategy is placing a UAV on top of each control point. This guarantees the ISR knows the active control point immediately and gives a 33% chance of learning the control point symbol immediately. It also provides intel of the opponent's locations within the buildings. Keeping the symbol-spying heavy UAV over the center control point guaranteed the fastest time to discover the symbol as it minimizes the distance to the other two control points. The team estimates the baseline scripted ISR policy employed at least a ~75% optimal strategy as it did not place the symbol-spying UAV over the center control point or perform OPFOR searches around the control point.

As the scripted ISR behavior was close to optimal, the team compared the ISR AI against the scripted ISR to determine how close the AI got to optimal, with the understanding that there is margin to improve upon the scripted ISR.

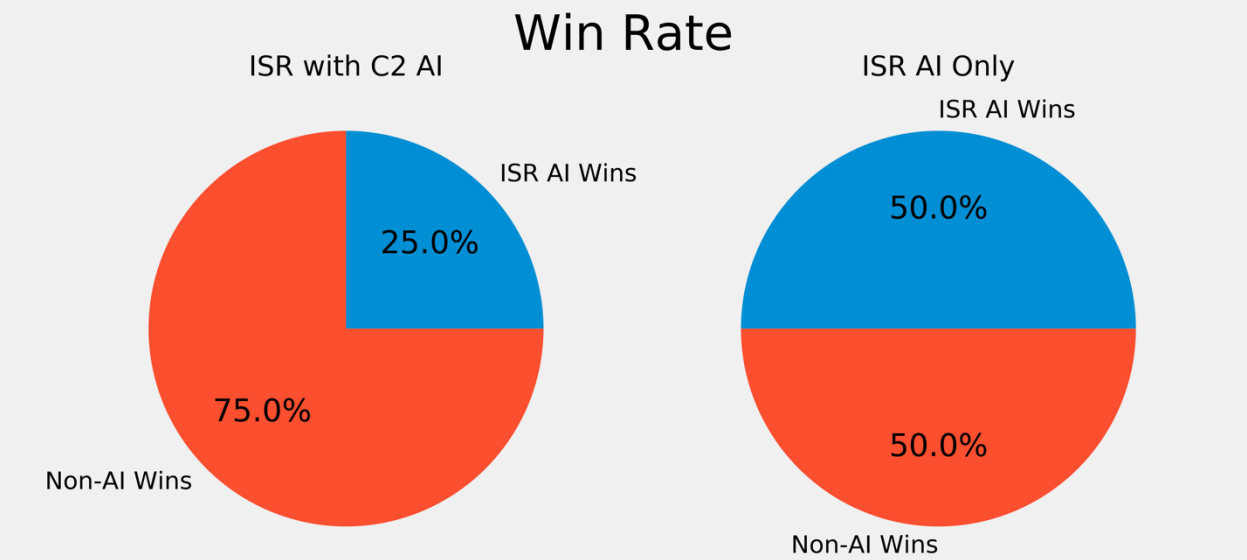


Figure 57. Win Rate Comparison

Four rounds of four matches each, totaling 16 games, were played with the ISR (eight games were played with the C2 AI only and are not included in this analysis). Half of those 16 games isolated the ISR AI (ISR AI with no C2 AI) against a non-AI assisted team and the other half were played with both an ISR and C2 AI against a non-AI assisted team. On the left of Figure 57 is the win rate breakdown of the ISR and C2 AI assisted teams. The right of Figure 57 shows the win rate breakdown of the ISR AI only (no C2) assisted teams.

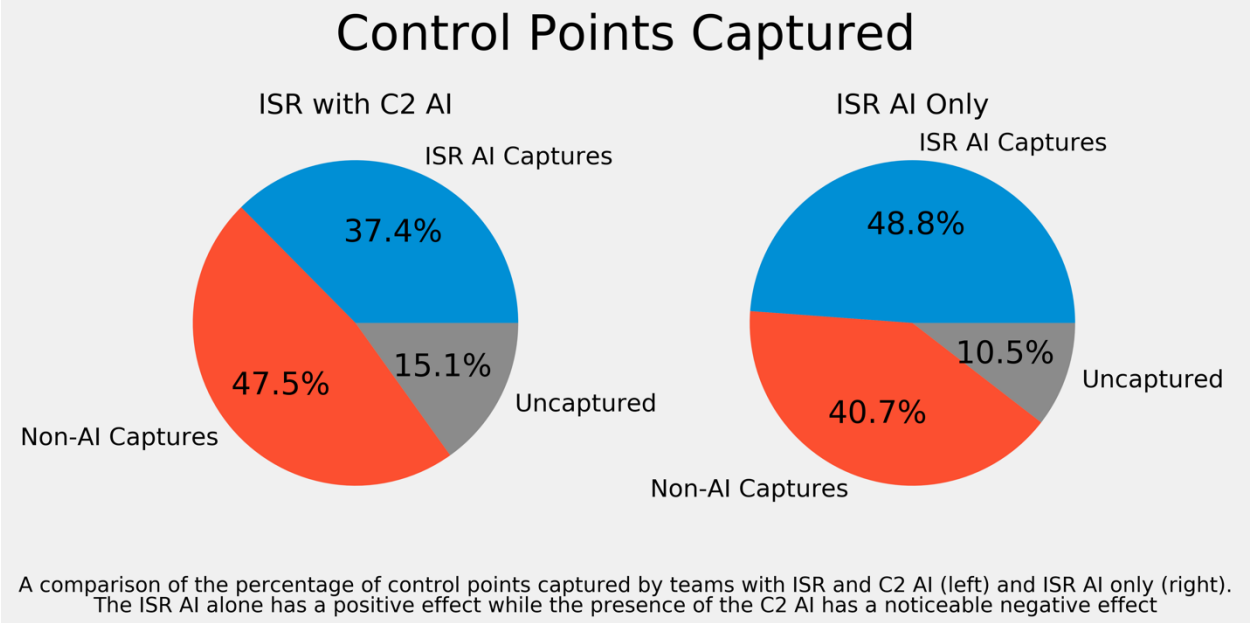


Figure 58. Control Points Captured

The same trend noted above with the win rates applies also to control points captured. Figure 58 shows that the ISR AI only assisted team actually captured more control points than the non-AI assisted team. The C2 AI paired with the ISR AI swung the advantage in the opposite direction.

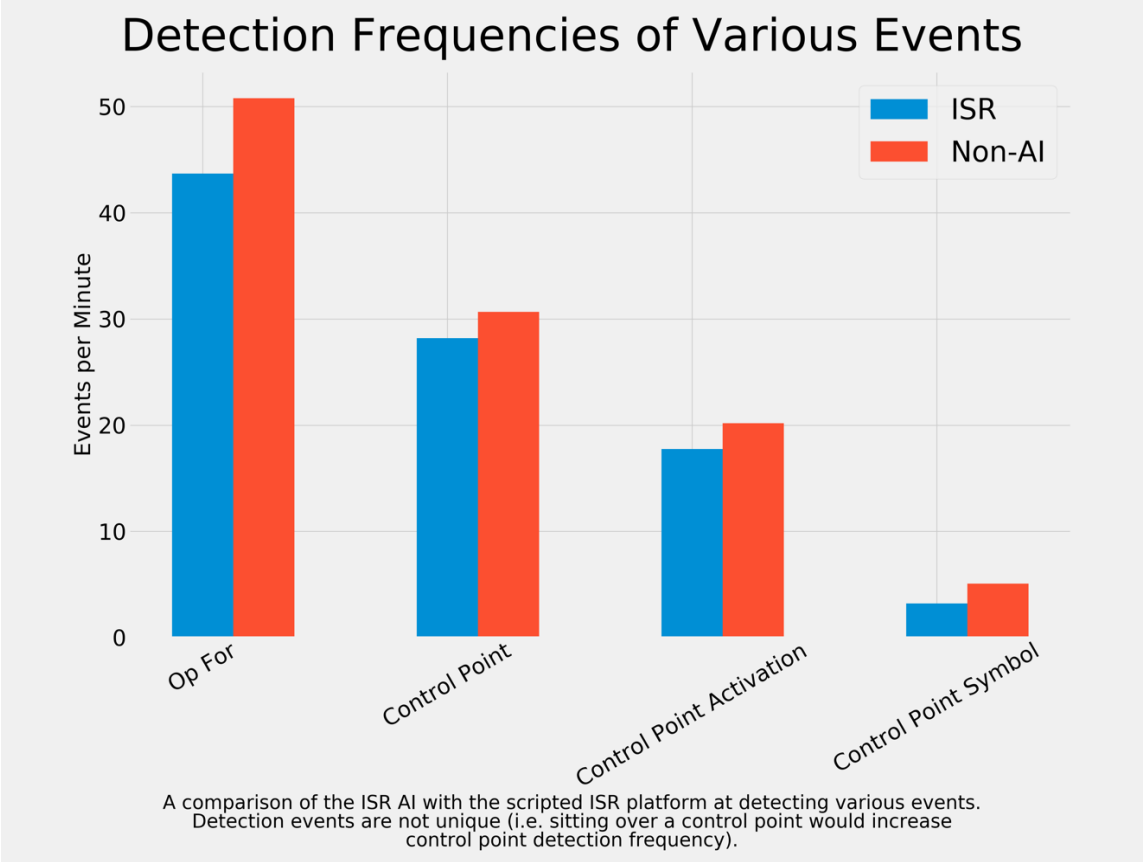


Figure 59. Detection Frequencies of Various Events

Figure 59 compares the detection of non-unique OPFOR locations, (inactive) control points, active control points, and control point symbols between the ISR AI and scripted ISR. These events are not unique, multiple detections of the same active control point are counted. Figure 59 shows how the scripted agent is predisposed to hover over control points. The ISR AI, favoring mobility, was incentivized early in training to search for OPFOR locations. Later in training control point detection was emphasized. Unfortunately, there was only a marginal improvement in the ISR AI's ability to detect active control points in the amount of time allotted. The ISR performed 86%, 92%, 88%, and 63% as effectively as the baseline scripted ISR at detecting op for locations, (inactive) control points, active control points, and control point symbols respectively.

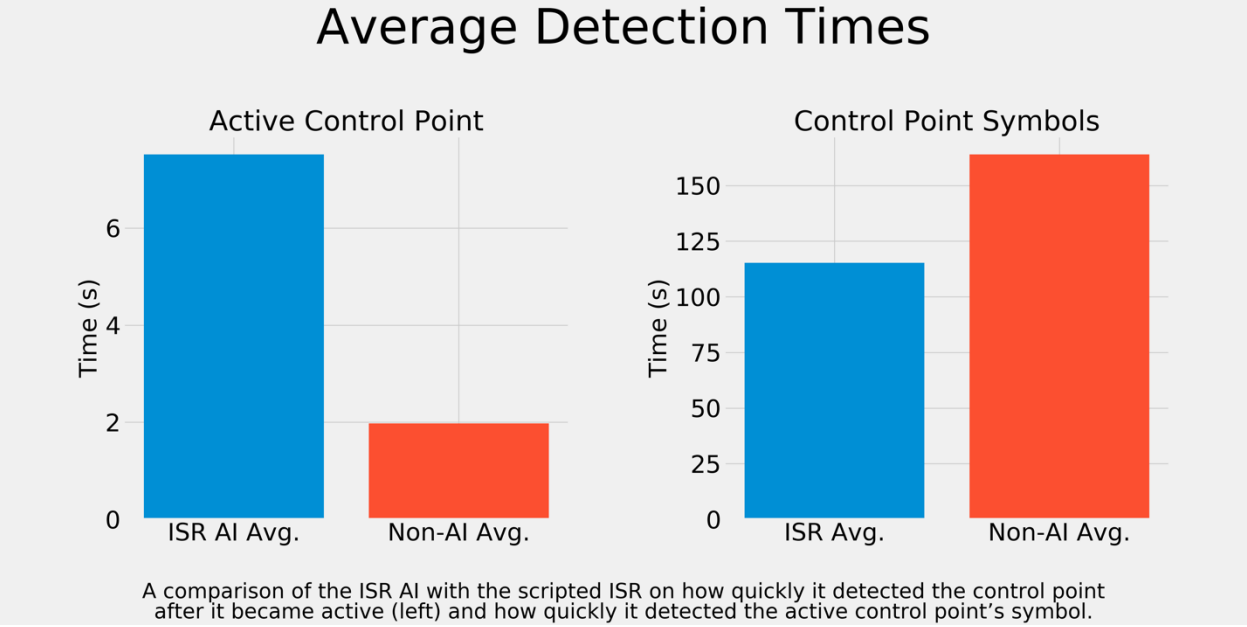


Figure 60. Average Detection Times

Figure 60 compares how quickly the ISR AI and baseline detected newly active control points and discovered the symbol. The ISR AI was overly focused on attempting to find OPFOR players causing it to operate 26% as effectively as the baseline. Again, this was because the AI was originally incentivized to find OPFOR locations. The area the UAV needs to cover to find an OPFOR target is far larger and very sparse compared to the control point locations which were static and known. With more time to iterate on the proper reward function the ISR's performance can be improved.

The ISR was better than the baseline at determining the correct symbol; it was 142% more effective. This actually indicates the AI was able to learn the importance of using the symbol-detecting UAV for its primary purpose. It is likely the AI redirected the symbol-detecting UAV in response to detecting active control points. With an ISR AI prioritized for discovering the active control point the symbol-detecting UAV should become even more effective.

As formulated for the September playtest, it is likely the ISR task's search space did not require an AI to solve. A scripted ISR was sufficient. The team believes a few tweaks can be made to the testbed to facilitate an ISR AI that can answer an interesting question.

Analysis of AI training

To facilitate the training of the ISR AI, the team trained it alongside a C2 AI.

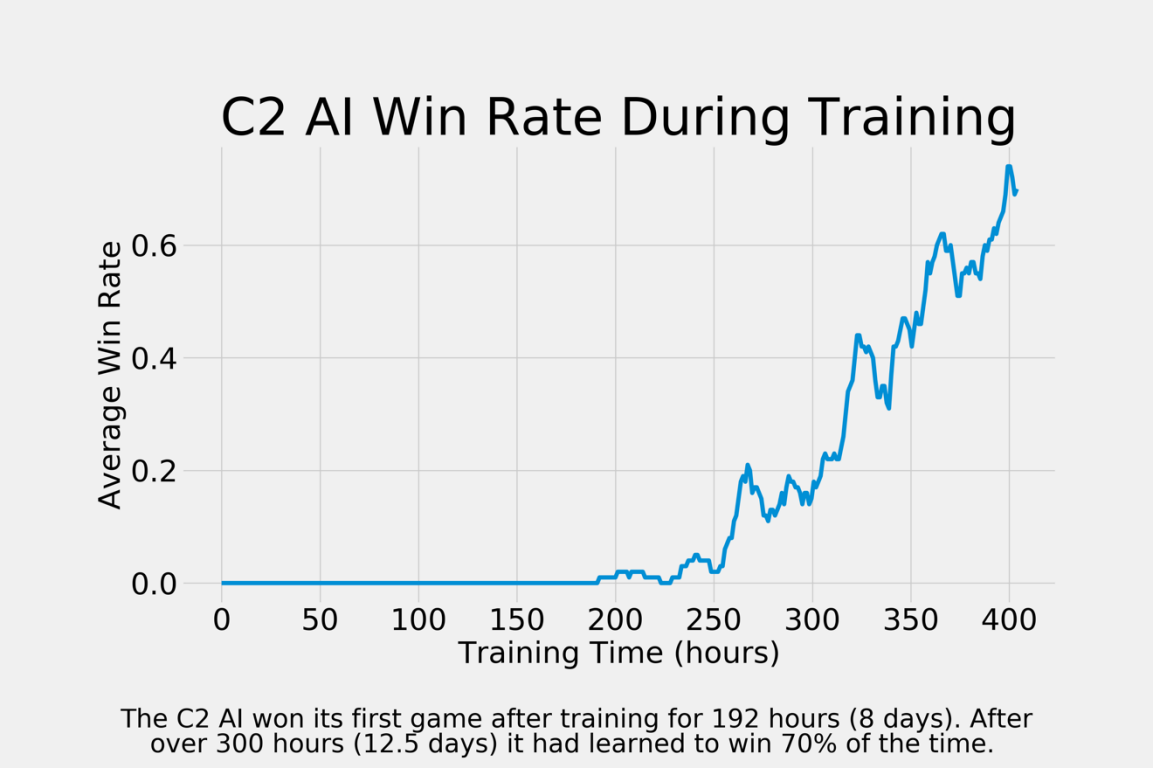


Figure 61. C2 AI Win Rate During Training

Figure 61 is included to demonstrate the difficulty in gathering enough training data on a new environment for the AI to learn robust strategies in a short time frame. The C2 AI won its first game after training for 192 hours (8 days). After over 400 hours (16.5 days) it had learned to win 70% of the time against a random aggressive opponent.

The AT2B Domination game and the strategies required to win are clearly complex. However, it is also clear that this approach has potential if allowed to mature. With enough lead time and ability to iterate on reward functions, hyperparameters, and fine-tuning action and observation spaces the C2 and ISR AIs should converge on competitive strategies. With a formulated training regimen, the C2 ISR may be able to learn a host of strategies and execute them in response to the opponent's strategy.

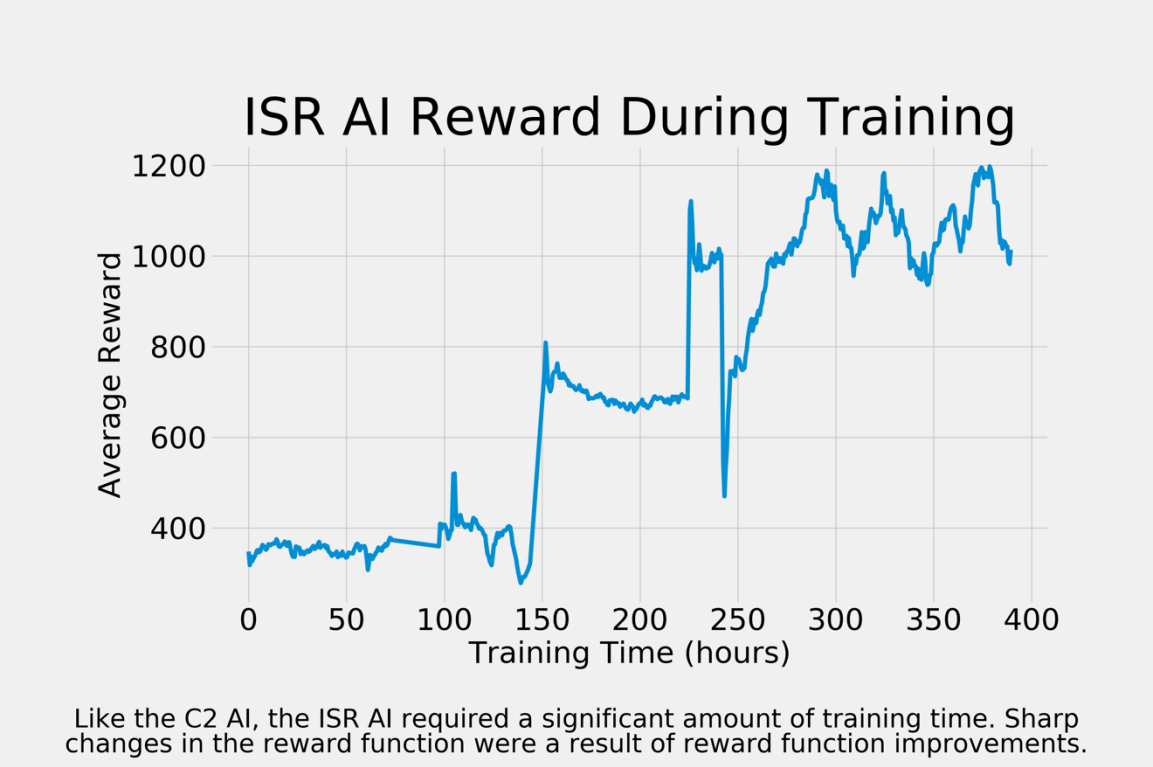


Figure 62. ISR AI Reward vs Training Time

Figure 62 shows ISR AI reward during training. While the C2 AI faces a complex and dynamic environment forcing complex decisions, the ISR AI instead faces an extremely large action space. Limiting the action space of the ISR AI to the three control points drastically reduces the action space but severely limits the learnable strategies.

The ISR AI certainly needs a reward function prioritizing control point intel. It would also benefit from a richer observation space. It was unable to learn a model of when control points would become active. With more information, it could learn this model and then handle multiple goals, control point detection versus OPFOR hunting, depending on the control points' status.

APPENDIX C: SEPTEMBER 2019 TEST TELEMETRY DETAILS

The following documentation describes the data structure of the telemetry generated by the AT2B testbed as of the September 2019 test build.

Root folders named 29Palms* contain full session dumps from each mission trial.

Each root folder contains a subfolder named ‘PawnInfo’ and a collection of optional additional folders.

The PawnInfo folder contains the master pawn key file ‘PawnInfoDatas.csv’ and a series of ‘PawnInfoPositionKeyFrames’ files.

PawnInfoDatas Files

PawnInfoDatas structure:

Name	Id	Faction	FireTeamId	FireTeamRole
Entity name stored as [sublevel] : [ue4 actor name]	Index of the actor starting with 0	The faction the actor belongs to	Optional team id	Optional fire team role (requires team id to be set to be meaningful)

PawnInforPositionKeyFrames Files

PawnInfoPositionKeyFramesfiles have a numeric suffix matching the index number contained in the PawnInfoDatas file so PawnInfoPositionKeyFrames7.csv corresponds to the entity listed with id 7 in the PawnInfoDatas file.

PawnInfoPositionKeyFrames file structure:

Time	PosX	PosY	PosZ
Time seconds	z location cm	y location cm	z location cm

Events Folders

Event folders are named for a specific event stream and contain a single file with the same name as the folder. Event files are only written to disk if events with the matching signature were triggered during a trial. So if one of the 29Palms folders is missing a given event it means that an event of that type did not trigger during the trial.

The first cell of the first row of an event file is reserved as a version number. The values in column zero are the times in seconds from the beginning of the trail that a given event triggered. The remainder of the first row contains labels for the values below the corresponding column.

The value types in each column are documented below:

- Int_: a signed 32bit integer
- UInt64_: an unsigned 64bit integer
- Float_: a 32bit floating point number
- Bool_: a Boolean value (true, false)
- Name_: A text label store at runtime using the UE4 FName data type
- Vector_: a set of 3 32bit floating point numbers <X,Y,Z>
- LocationIndex_: a helper type that indexes the entity data set
  - AfterActionPathManager or PawnInfoDatabase depending on mode
  - Can encode a collection of 1 or more entities
  - packs as ':' delimited list <actor>|<location keyframe index>
    - example output:
      - 6|700:12|15
      - The 700th key-frame for entity #6 and the 15th key-frame for entity #12
  - -1 is used to indicate the null index for optional entries
- UnknownEnum a helper type that allows for any user defined enumeration to be included as a supported type
  - See structFactionEnum for an example of how to use this type.

For example ‘WeaponFiredEvent / WeaponFiredEvent.csv’.

0	PawnLocations	HitLocation	TargetKilled
322.617004	106|6:96|85:96|85	-15548.885742:-15012.152344:2440.132568	TRUE
326.265015	97|88:107|9:107|9	-15889.333984:-14622.547852:2753.480957	TRUE
327.108002	102|88:106|6:-1|-1	-15664.178711:-14887.779297:2641.036621	FALSE
327.216003	102|88:106|6:-1|-1	-15667.833984:-14885.497070:2641.036621	FALSE

So the above data represents 4 event captures of the WeaponFiredEvent type with type signature version 0

Pawnlocations a `LocationIndex_ type:` read as <shooter>|<location>:<intended target>|<location>:<optional entity hit>:<location>

So 102|88:106|6:-1|-1 => entity PawnInfoPositionKeyFrames102 at location at row 88 shot at PawnInfoPositionKeyFrames106 at location at row 6 and missed.

HitLocation a Vector_ type: read as x y z in centimeters

TargetKilled a Bool_: type read as TRUE or FALSE

Below is a complete listing for all event layouts:

WeaponFiredEvent	Triggers when a weapon is fired
`LocationIndex<3>`	`PawnLocations index 0 is shooter, 1 target, 2 entity hit or -1 if nothing was hit or the round impacted the static environment`
`Vector`	`HitLocation the round stopped in the environment x:y:z in cm from world origin`
`Bool`	`TargetKilled if an entity was hit did it survive`

CharacterStanceEvent	Triggers when an entity changes stance
StanceEnum	Standing, crouched, prone
LocationIndex<1>	CharacterInfo

CharacterDeathEvent	Triggers when an entity dies
LocationIndex<1>	CharacterInfo

CharacterReloadEvent	Triggers when an entity reloads a weapon
Int	WastedRoundCount
Int	MagazineCount
LocationIndex<1>	CharacterInfo

ExplosionEvent	Triggers when a grenade explodes
Vector	location
Int	Radius in centemeters

UAVSensorFeed	Triggers over time as the UAV repositions its sensor
Float	Sensor field of view relative to x axis
Float	Sensor field of view relative to y axis
Vector	Sensor location
Vector	Sensor view direction

C2CommandEvent	Triggers when the C2 AI script sends a command
Int	Team number
Int	Member number
Name	Name of the command
Name	Name node the command targets, or name of the class the respec to (in case of respec event)
Name	Rotation Yaw to target when AI unit arrives, or which symbol to respec to.

DominationControlStateEvent	Triggers when a control points state changes
Int	Point number (0 = North, 1 = Central, 2 = South)
Name	State of the control point
Int	Symbol of the control point (always 1 in 1 symbol games)
Int	Number of blue team members inside the building
Int	Number of red team members inside the building
Float	Score multiplier of the claiming / capturing team

DominationScoreUpdate	Triggers when a scoring phase ends
Int	Blue team’s score
Int	Red team’s score

UAVKnowledgeEvent	Triggers when a UAV's knowledge of the game changes
Int	Team number
Int	Member number
LocationIndex<1>	Location index of the UAV
List<Int>	The Pawn IDs of spotted enemies
List<Vector>	List of vectors (location) of spotted enemies
Int	Which capture point is seen by the UAV (-1 for no point, 0 for north control, 1 for central control, 2 for south control)
Int	Activation status of the control point (-1 if no point, 0 if deactivated, 1 if activated)
Int	The control point's symbol if known (0 if no known symbol)
Float	Time since the point has been activated.

CharacterRespecEvent	Triggers when a unit respecs
Int	Team number
Int	Member number
Int	Loadout (1 is shotgun, 2 is rifle, 3 is rocket)
Int	Symbol

LIST OF SYMBOLS, ABBREVIATIONS AND ACRONYMS

AAR	After Action Review
AI	Artificial Intelligence
ANOVA	Analysis of Variance
API	Application Programming Interface
AT2B	A-Teams Test Bed
ATAK	Android Tactical Assault Kit
A-Teams	Agile Teams
AWS	Amazon Web Services
BFC	Boston Fusion Corporation
BLUFOR	Blue Force
C2	Command and Control
CI / CD	Continuous Integration / Continuous Deployment
CO	Commander
CP	Control Point
DARPA	Defense Advanced Research Projects Agency
EW	Electronic Warfare
FPS	First-Person Shooter (videogame)
GTAI	Game-Theoretic AI
HMT	Human-Machine Teaming
HRL	Hughes Research Laboratories
HSR	Human Subjects Research
HST	Hide and Strike Testbed
HUD	Heads Up Display
IED	Improvised Explosive Device
Intel	Intelligence
IRB	Institutional Review Board

ISR	Intelligence, Surveillance, and Reconnaissance
K/D	Kill-to-Death
KIA	Killed in Action
LSTM	Long Short-Term Memory
MC	Monte Carlo
MITRE	The MITRE Corporation
ML	Machine Learning
MoE	Measures of Effectiveness
MoP	Measures of Performance
OPFOR	Opposing Force
OPTEMPO	Operational Tempo
PI	Principal Investigator
PII	Personal Identifiable Information
PM	Program Manager
PROTEUS	Persistent, Resilient Operations Testbed for Expeditionary Urban Scenarios (DARPA program)
R&D	Research and Development
Respec	Respecification
RL	Reinforcement Learning
ROTC	Reserve Officers' Training Corps
SDK	Software Development Kit
SME	Subject Matter Expert
TA	Technical Area
TTP	Tactics, Techniques, and Procedures
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UI	User Interface
USMC	United States Marine Corps
UX	User Experience

UxVs	Unmanned Vehicles
VTB	Virtual Test Bed
WebGL	Web Graphics Library
WLU	Wonderful Life Utility