



**ANALYSIS OF HOW COMMUNICATION AFFECTS
HUMAN TEAMS IN A DYNAMIC GAME**

THESIS

Jake M. Spuller, 2nd Lieutenant, USAF

AFIT-ENG-MS-17-M-071

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-MS-17-M-071

ANALYSIS OF HOW COMMUNICATION AFFECTS
HUMAN TEAMS IN A DYNAMIC GAME

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Jake M. Spuller, BS

2nd Lieutenant, USAF

March 2017

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENG-MS-17-M-071

AN ANALYSIS OF HOW COMMUNICATION AFFECTS
HUMAN TEAMS IN A DYNAMIC GAME

Jake M. Spuller, BS

2nd Lieutenant, USAF

Committee Membership:

Maj Jason M. Bindewald, PhD
Chair

Dr. Douglas D. Hodson
Member

Dr. Michael E. Miller
Member

Abstract

This research presents a new multi-agent, dynamic design to an existing tablet-based game environment that allows for the study of multi-human team communication when adapting to a changing environment. Research was conducted on designs of multi-agent environments, as well as synchronization protocols, to develop the new environment. Different types of communication were also examined to universally define how players in the environment interacted with each other.

To make the environment rules dynamic and force participant strategies to change, common strategies provided from participant answers in a previous study in the same environment were confounded. Participants worked in two-person teams to establish strategies and work together to perform optimally in the environment, and adapt to the unseen changes from the environment. Participants were divided into two groups – teams that could communicate and teams that could not communicate. This experiment was performed to examine how communication affects performance, strategy development, and strategy adaptation.

The results of the research show how teams that can communicate compare with teams that cannot communicate in strategy selection, strategy adaptation, and overall performance in the environment. These results can be applied when creating machine-agent teammates that must work with a human to adapt to changes never seen before.

Acknowledgments

I would like to thank my thesis committee, Maj Jason Bindewald, Dr. Michael Miller, and Dr. Douglas Hodson, for sharing their expertise and time to support my research. I would also like to thank Maj Bindewald for his countless hours of dedication, support, and guidance throughout my thesis and research process. Finally, I would like to thank my wife Emma, for her unwavering love and support for me every day.

Jake M. Spuller

Table of Contents

	Page
Abstract	iv
Table of Contents	vi
List of Figures	viii
List of Tables	x
1. Introduction	1
Chapter Overview	1
1.1 Department of Defense Motivation	1
1.2 Autonomy Research Motivation	2
1.3 How Human-Human Teams Apply to Human-Machine Teams	4
1.4 Research Question and Thesis	5
1.5 Methodology Introduction	7
1.6 Paper Outline	8
2. Background	9
Chapter Overview	9
2.1 Multiplayer	10
2.2 Description of Test Environment	16
2.3 Communication	21
2.4 Summary	30
3. Methodology	31
Chapter Overview	31
3.1 Multi-Agent <i>Space Navigator</i> Environment	31
3.2 Human-Subject Test	38
3.3 Summary	46
4. Analysis and Results	47
Chapter Overview	47
4.1 Comparing Convergence on Strategies between Groups	47
4.2 Overview of Strategy Change as Environment Changes between Groups	55
4.3 Overall Performance Comparison between Groups	69

4.4 Summary	74
5. Conclusion	76
Chapter Overview	76
5.1 Summary of Thesis and Investigative Questions.....	76
5.2 Summary of Results.....	76
5.3 Summary of Methodology to Answer Research Questions.....	77
5.4 Future Work.....	78
Appendix A - <i>Space Navigator</i> Interaction Study Pre-Questionnaire	82
Appendix B – <i>Space Navigator</i> Interaction Study Post-Pairing-Questionnaire	83
Appendix C – <i>Space Navigator</i> Interaction Study Post-Questionnaire	85
Works Cited	90

List of Figures

	Page
Figure 1. A visual representation of a client-server architecture [11].	11
Figure 2. A visual representation of a peer-to-peer (P2P) network [13].	12
Figure 3. An example of the <i>Space Navigator</i> environment	19
Figure 4. The different types of pings in <i>League of Legends</i> [24].	26
Figure 5. An overview of the functions of the client and server in <i>Space Navigator</i>	32
Figure 6. A behavioral diagram showing how a player draws a trajectory.	35
Figure 7. How participants responded when asked how they would divide workload in <i>Space Navigator</i>	48
Figure 8. How non-communication teams reported initial strategies.	50
Figure 9. How non-communication teams reported final strategies.	50
Figure 10. How communication teams reported initial strategies.	52
Figure 11. How communication teams reported final strategies.	53
Figure 12. Difference of proportions for colored ships.	58
Figure 13. Difference of proportions for each side of the environment.	59
Figure 14. Average number of trajectories drawn for all players across all game instances.	61
Figure 15. Average proportion change of drawn trajectories after environment change occurs.	62
Figure 16. Average number of initial trajectories for all players across all game instances.	63

Figure 17. Average proportion change of redrawn trajectories after environment change occurs.	64
Figure 18. Average number of redraws for all players across all game instances.....	65
Figure 19. Average proportion change of redraws between players after environment change occurs.	66
Figure 20. Proportion gaps for redrawn trajectories between teammates.....	67
Figure 21. Overall average team scores for experiment groups.	70
Figure 22. Average team scores across all game instances.	71
Figure 23. Average number of ships reaching destinations per team.	71
Figure 24. Average number of collisions per team across all game instances.	72
Figure 25. Average number of bonus pick-ups per team.....	73

List of Tables

	Page
Table 1. How many times participants suggested general strategies.....	36
Table 2. How players may divide workload and how to confound those strategies.....	38
Table 3. Player pair ordering for the human-subject test.....	41
Table 4. Summary of changes occurring in environment.	42
Table 5. Example of determining change in performing an action for players in a team.	56
Table 6. Correlations between redraws, scores, destinations, and collisions.	74

ANALYSIS OF HOW COMMUNICATION AFFECTS HUMAN TEAMS IN A DYNAMIC GAME

I. Introduction

Chapter Overview

The purpose of this chapter is to explain the motivation for this research from a Department of Defense point of view, the motivation in relation to autonomy overall, and align this specific research with both motivations. Next, this chapter describes the specific research question, the goals and investigative questions to answer the research question, and the expected results in answering the questions. This chapter will also introduce the methodology used to answer the research question. Finally, it outlines the remaining chapters.

1.1 Department of Defense Motivation

Humans and machines interact daily to accomplish tasks that one could not do without the other. Humans have skills that computers lack such as reasoning through new problems, whereas computers can handle complex calculations and monitor multiple objects of interest with precision and ease [1]. Successful communication helps the development of reliable strategies in teams, mitigating issues that arise and ensuring the performance of each entity in the team.

The United States Air Force is moving in the direction of human-machine teams in various fields of work with varying levels of automation in the teams. Examples include pilots understanding software in new aircrafts, autonomy in cyber defense, and remote controlled vehicles [2]. Former Air Force Chief Scientist, Dr. Mica Endsley, explained that airmen need to address the challenges of “supporting understandability of

the autonomous system...and successful methods for verification and validation of the autonomy” [2] (i.e. communication between the humans and machines). Machines that can correctly analyze data but cannot communicate that information to a human waiting to act on certain results may be rendered useless when communication is inefficient.

Without proper communication and strategy development, systems produce incorrect information which leads to mission failure [3]. An example of this is the struggling software development in the new F-35 aircraft. One problem “involves frequent timing errors between radar sensors and onboard computers during takeoff” [4], and with the software’s lack of efficient communication to the pilot, the pilot will be misinformed of his surroundings which could lead to numerous secondary problems. Dr. Endsley outlines how humans and machines should interact to create successful teams including: automate only where necessary and at the lowest level, provide automation transparency, and keep the operator in control and in the loop [2]. Designing human-machine teams that follow these guidelines and that can communicate efficiently overall will produce successful teams that become more reliable than comparable teams without autonomy.

1.2 Autonomy Research Motivation

Human-machine teams are not only utilized in the Department of Defense, but also in many modern technologies that are available to most consumers worldwide. Self-driving cars developed by Cadillac, Tesla, Mercedes-Benz, and others, for example, take enormous trust from the driver to believe the car will adapt to changing situations (e.g. construction closing lanes, other cars cutting the driver off, and different weather

conditions) and keep the driver safe. The driver cannot rely on the car to perform every detail in driving, but instead, as a spokesperson for Tesla said, “We’re not getting rid of the pilot. This is about releasing the driver from tedious tasks so they can focus and provide better input” [5]. It is critical that the human-machine team in autonomous cars communicate well so the driver knows who is in control of the vehicle or which aspects of driving the driver needs to focus on to efficiently utilize the autonomy in the vehicle. Also, when there are changes to driving or road conditions that the vehicle has never seen, it needs to be able to alert the driver of its problem or be able to solve the problem in the time it has to make the decision.

Another example of human-machine teams communicating and working together to adapt to change are in smart homes. There are numerous entities that operate within a single home, such as the thermostat, sprinkler systems, and lighting, that makes it difficult for a human to control and monitor all operations throughout an entire day. In the MavHome smart home project that utilizes an agent, “the agent’s goal is a function that maximizes comfort and productivity of its inhabitants and minimizes operation cost. In order to achieve these goals, the house must be able to predict, reason about, and adapt to its inhabitants” [6]. For example, if the occupant normally comes home from work at a certain time, and the home is programmed to turn on a television, turn on certain lights, and set the temperature, the program would need to adapt if the occupant instead came home late from working overtime and went straight to bed. The program should recognize the change and adapt to continue running the house efficiently and not wasting energy when the occupant isn’t using it. One goal of human-machine teams is to seamlessly work together as if the human was working with another reasoning human

instead of a machine. One way to develop better human-machine teams is by studying and characterizing human-human team communication and adaptation as they adjust to changes in an environment.

1.3 How Human-Human Teams Apply to Human-Machine Teams

Human-machine teams and human-human teams are similar in that they are both working together to achieve some goal. They divide tasks and workload based on strengths and weaknesses so that each entity in the team is working as optimally and efficiently as possible. An advantage humans have over machines is their ability to reason through problems never seen before, as well as unforeseen changes to current problems. Humans can communicate with their teams to derive solutions that adapt to the changes whereas machine teammates could freeze up by not knowing what to do with the unknown [7].

Studying how humans communicate their adaptive strategies to new changes may lead to more successful machine teammates. If the machine encounters an error or notices something completely different in the environment it is programmed to respond to, the machine must be able to communicate its problem so that the human can aid the machine to adapt to the change it sees. An example of a human-machine team in unmanned aerial vehicles (UAVs) could be if the UAV is suddenly struck by something in the air and needs to land immediately. It may be preprogrammed to land at many different airstrips but if there are none nearby, it needs to adapt and find some straight piece of land or completely cleared road to avoid losing the aircraft. The aircraft must communicate with the remote pilot its most up-to-date surroundings constantly and communicate details of

the environment such as elevation, wind speed and direction, and most importantly if there are people in the area that could be harmed from a landing or crash. This information will assist the pilot in making the best decision in a short amount of time based on how well the machine communicates with the human. Determining the information a remote pilot would need to handle the situation involves first going through the process with a piloted aircraft. The remote pilot could talk to the pilot of the aircraft and the communication that occurs could be applied to the remote pilot and UAV to handle the situation as if two reasoning humans were working together.

Another example in self-driving cars is if construction closed a lane with a barrier and forced traffic to merge to another lane. As a human, the driver would know to merge and avoid the barrier to avoid a head-on crash. The self-driving car may not know this, and if it does not communicate with the human that it does not know what to do and the human needs to take control as the driver, it could cause the car to crash. There are numerous different scenarios that can happen on the road, even in a single day of driving. The car must recognize what it can and cannot do immediately as it occurs to alert the driver when the car is in control or when the driver needs to be in control. Studying how humans would react in these situations among many others may lead to better results and more successful human-machine teams when the same reasoning a human would use is applied to the machine.

1.4 Research Question and Thesis

Machines in human-machine teams are programmed to do specific tasks and rely on the human to make changes if the environment they work in changes, or is dynamic.

The dynamic environment is defined as the tablet based game this research stems from, and the changes that occur throughout the game that forces players to adapt their strategies to work with the new version of the game. This is different from a static environment, which would allow a player to use one strategy throughout the game without it affecting performance negatively or positively. A dynamic environment is necessary for facilitating a change and to measure the reactions to that change from human participants. The dynamic environment used for this research is very similar to a real-world air traffic controller environment, where aircraft are safely guided by humans to land in specific locations. This environment is extremely dynamic, where unanticipated events could occur (e.g. an aircraft breaking down on a runway, forcing all aircraft utilizing that runway to reroute) and the human controller must rely on training, reason, and situational awareness to continue with the job.

Machines need to be able to adapt to changes quickly and efficiently and the problem that needs to be solved is how to build agents that can adapt to changing environments. Before building the machine agent that adapts like a human, research must study how humans adapt to the changes in an environment. Learning how the humans adapt, documenting the process, and storing the information can lead to the creation of human models that can be applied to machine teammates. The machines will then be able to reason and adapt to changes just as a human teammate would. The question this research aims to answer is how do human teams adapt their strategies to a dynamic video game environment when they can communicate versus when they cannot communicate?

Two different human groups - those who can communicate and those who cannot communicate - must be studied to compare the results communication to a baseline of no

communication. The expected result from the research that answers this question is that users who can communicate will change their strategies in different ways and to different extents than those who cannot communicate. Secondary questions that need to be answered to solve this research question include: how are users dividing the workload based on the goals and objects in this dynamic video game, and how do the performances of the groups compare in this dynamic video game? Answering these questions will lead to a solution that discovers how communication affects the differences in adaptation and strategy development in human teams that play this video game.

1.5 Methodology Introduction

The methodology for testing the research question involved first developing the single human-machine environment *Space Navigator*[8] to a multi-human environment. The multi-human teams can then work together to achieve the same goals that the human-machine teams previously worked towards accomplishing. Next, the environment needed to change at different times when the users interacted with it. Preliminary research allowed for the design of environmental changes that would target certain behaviors and strategies that humans would use within in the environment. This forces the human teams to make some change to their strategies in response to the changing environment. Third, all environmental data, human-provided answers to questionnaires, and audio recordings were analyzed to categorize the ways the human-teams interacted with the environment that suggest results and trends that human-teams make when adapting to the changing environment. Teams that can and cannot communicate are compared to decide if vocal

communication had an effect on adapting strategies to the changing environment and to what extent.

1.6 Paper Outline

This paper proceeds as follows: Chapter 2 describes similar work in this research area, defines different forms of communication in a virtual environment, and describes how this specific research is novel compared to similar work. It also defines the environment used for this research. Chapter 3 defines how the modified environment was created from its previous version. It also describes how the experiment was designed to successfully answer the research question. Chapter 4 analyzes the data from the experiment and suggests how multi-human teams adapt their strategies in a changing environment when they can and cannot communicate. Finally, Chapter 5 concludes the paper and suggests future work involving creating human-based models that can be applied to machines in the research environment.

II. Background

Chapter Overview

The presented research aims to test how communication affects performance in a new environment in response to changes in the environment. Experiments in the environment will be used to determine a baseline for performance between two people in the same, new environment when communication is allowed in one group and not allowed in another group. The purpose of this chapter is to discuss related work. Included in this is the current and most common implementations to synchronize multiplayer games. This chapter will also discuss the effects of communication on the performance of a task between humans in a team environment. This information will provide a basis to develop a methodology to test human communication in a real-time multiplayer environment. This chapter is structured in two overall sections, multiplayer and communication. The multiplayer section describes how to structure and synchronize a multiplayer game, which can be used as the tool to test the thesis in experiments. The following section describes the environment used for the experiment. Next, communication will help in understanding what to look for in humans communicating and classifying the observed results for better analysis. The analysis of the communications will help in proving or disproving the thesis.

2.1 Multiplayer

This section introduces real-time strategy games and the basic structures used to develop a multiplayer game. This section also discusses different techniques to synchronize a multiplayer game to minimize latency.

2.1.1 Real-Time Strategy

Ontañón *et al.* [9] define real-time strategy (RTS) games as “simultaneous move games, where more than one player can issue actions at the same time. Additionally, these actions are...not instantaneous...Each player has a very small amount of time to decide the next move.” All players have the ability to manipulate the environment at any time which increases the complexity and need for teams to communicate when working in an RTS environment. If the state of the environment changes, this could create a need for new strategies or a change in current strategies. This definition applies to the game *Space Navigator* as both players have the ability to draw trajectories at any given time. The game environment never pauses, which forces the players to continuously update their trajectories and strategies to perform optimally and score as high as possible.

Three challenges specifically apply to humans who are introduced to a new RTS environment and not expected to continue playing for an extended amount of time: planning, learning, and uncertainty [9]. Because the number of possible actions at a given moment in the environment is large, planning is important for the players to have an idea of what moves they should make in the next few possible situations. Planning improves through player learning. There are three basic ways to learn about a game and improve performance: *prior learning*, which is applying lessons learned from other RTS games a player may have played previously; *in-game learning*, which is a human learning in real-

time how his teammate plays or others play the game and extracting the best strategies and rejecting those that do not work; and *intergame learning*, which is learning between games what worked as a whole.

Uncertainty is the final challenge applicable to humans. In situations where humans cannot communicate, there is always uncertainty in what the teammate will do. The game environment is also uncertain because the future environment cannot be predicted, which forces players to think critically and does not allow a deterministic approach toward solving the environment and producing an optimal score.

2.1.2 Basic Multiplayer Structures

There are two primary ways to set up a multiplayer environment, either using a client-server architecture or a peer-to-peer (P2P) network. In a client-server architecture (Figure 1) “there is a single, authoritative server that is responsible for running the main game logic” [10]. All clients connect and send their inputs to the server. The server processes the input commands at the rate it is capable and returns to the clients an array of objects to render in the next frame processed by the client.



Figure 1. A visual representation of a client-server architecture [11].

A P2P network (Figure 2) works differently from a client-server architecture in that clients do not connect to a central server. The clients are instead described as peers who connect to each other using the network of the resource they are trying to share. Each peer that establishes a connection is responsible for sharing part of their hardware resources, such as processing power or storage capacity in return for access to the content shared and offered by the network [12].

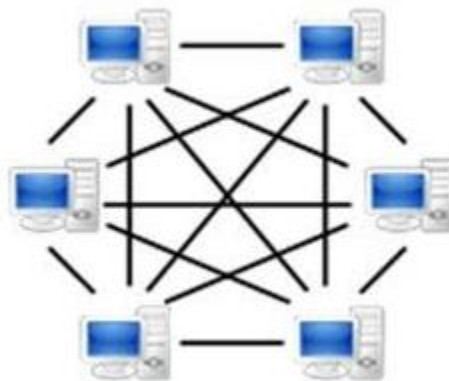


Figure 2. A visual representation of a peer-to-peer (P2P) network [13].

One advantage of using a client-server architecture over a P2P network is that all traffic can be monitored at a single point to prevent cheating which is useful in multiplayer games. Maintenance and troubleshooting can also be more efficient because the server is the single point of failure if an online game crashes. However, because multiplayer games can scale to a large number of players, the server would need to be able to handle an enormous load of requests to receive, process, and send back out to each client, which could cause the server to become a bottleneck for the network traffic and impact the latency and experience for the users. P2P networks are advantageous in splitting the work-load of the network among the peers and allow the peers to use their

own computing power to process requests. By giving more control to the peers, the peers will also have to be responsible for their own network connections, backing up and storage of data, and analysis of connections and data that are secure and unsecure. P2P networks are decentralized and difficult to administer, which can lead to more exploits, cheating, and intrusion of the software shared over the network.

2.1.3 Multiplayer Synchronization

Synchronizing an environment is a challenge in developing a multiplayer game because it is the difference of what users will be experiencing when playing a game online versus playing the game individually. A user wants to see exactly what the other users are seeing in the room they are all connected to in order to make the most accurate decisions. A decision is entirely based on the information provided and if the information is flawed, delayed, or missing the user will not perform as optimally as they would like. It is impossible for information to pass over a network instantaneously [14] and, because of this inherent delay in information, research has provided solutions to optimize the way information is passed and viewed by the user, including Trailing-State Synchronization, Bucket Synchronization, and a P2P synchronization protocol.

2.1.3.1 Trailing-State Synchronization

Trailing state synchronization (TSS) is a synchronization method developed to work primarily in a mirrored game architecture, which is a hybrid between a client-server architecture and a peer-to-peer network. “Mirrored architectures allow for improved performance compared to client-server architectures while at the same time allowing for a greater degree of centralized administration than peer-to-peer architectures” [15]. Cronin *et al.* [15] developed TSS to solve synchronization problems in the game *Quake* “which

has very frequent updates, has a need for strong consistency, and is very latency sensitive,” all which are qualities of the multiplayer environment needed for this experiment.

TSS operates by copying the current state (leading state) of a game that the user sees, and running the copied state at a delay relative to the leading state [15]. The platform on which the game runs creates multiple copies of a state, or trailing states, depending on the size and processing power of the platform to improve synchronization. In fast-paced multiplayer games, multiple commands push to the server every second and the server may process the commands in the wrong order due to latency from the users or the amount of traffic from the game. To eliminate inconsistencies, commands execute in the correct order in the copied states where there is more time to process all the input commands. If there are inconsistencies between the trailing states and the leading state, the platform updates the leading state to reflect the true state of the game environment.

2.1.3.2 Bucket Synchronization

A different synchronization method developed by Gautier and Diot [16] is known as the *bucket synchronization mechanism*. The goal of this mechanism is to “guarantee the consistency of [a] game, regardless [of] network delay” [16]. The mechanism was tested using a game the authors created called *MiMaze*, which utilizes a fully distributed architecture with no central server. All participants in the game are responsible for computing their own view of the global state that is consistent with the states of all other participants.

The bucket synchronization mechanism functions by dividing the game time of *MiMaze* into different time window intervals which are known as buckets [16].

Whenever data passes through the network it is marked with a time stamp. Each player stores the data in a local bucket associated with the time stamp of the data. While a player is not changing the global state, the data in the buckets will spill over to the current bucket. When a player is ready to send data across the network to the other players and update the global state of the game, the player executes all the commands and data in the local current bucket. The programmer or the game platform's capabilities determine and limit the size of the buckets and execution rate of the commands in each bucket.

2.1.3.3 P2P Synchronization Protocol

Synchronizing a multiplayer game over a P2P network can be difficult because synchronization occurs from multiple sources. Client-server architectures have the advantage that everything from the server is the true global state for all players and all clients must update from that server. In the P2P network, each peer is changing the state and that information must get to all other peers in the network efficiently to optimize game performance. A recent P2P Synchronization protocol handles synchronization in P2P networks by utilizing obsolescence and correlation [17].

Obsolescence in the P2P Synchronization Protocol determines the events that have become obsolete since the current state of the game. Events are considered obsolete if their actions do not alter the current game state that a player is currently viewing. Eliminating obsolete actions, whether they are received by the peer too late or simply have no effect on the current state, can allow peers that process information slower than others to accelerate their input delivery to the network.

Correlation allows the protocol to identify events that need to be processed in a specific order. Two events correlate if a change in their execution order affects the final

game state. Correlation affects game synchronization speed because if a peer receives packets of data out of order, and they are not correlated, then the peer should process the data in the order it arrived instead of waiting for other packets that are delayed.

The P2P Synchronization protocol is similar to Bucket synchronization in that information is coming from multiple peers and the data passes to other peers to create a common state among the peers playing a game. However, the P2P Synchronization Protocol differs in that it is able to reorder execution of commands only if necessary and avoid executing obsolete commands instead of blindly executing a bucket filled with commands in an arbitrary order. This is an advantage over Bucket synchronization because the processor will not waste time executing commands that are no longer needed, but it will use more time in deciding which commands are obsolete. An advantage of Trailing-State Synchronization over both Bucket synchronization and the P2P Synchronization Protocol is the assurance of the currency of the leading state and insuring that all data that have passed through the network have executed in the leading state. A disadvantage to using TSS is the amount of memory needed to store multiple copies of the current state and continuously update the current state as well as the copied states.

2.2 Description of Test Environment

The test environment, *Space Navigator* (Figure 3), is a tablet-based video game that uses touchscreen input to control objects in the environment [8]. The goal of the game is to obtain the most points over a period of five minutes. There are five objects that are operated upon in the game:

1. Spaceships - Spaceships are entities that “fly” onto the screen, and must be controlled by the user. The goal of each spaceship is to land on its assigned destination planet. Spaceships appear at time intervals set by some distribution (in the standard case, this is a uniform distribution). Each spaceship appears at a random location along the edge of the screen. In order to ensure against collisions, consecutive spaceships cannot come from overlapping locations (i.e. they must be more than a spaceship’s width away from the location of the last ships appearance). A spaceship appears on the screen moving in the direction opposite the screen edge from which it emanated.
2. Planets - Planets are fixed objects on the screen. Each spaceship is assigned a destination planet (color of the spaceship and its destination planet will match). When the edge of a spaceship overlaps the edge of the destination planet, it lands on the planet and points are given to the player accordingly.
3. Routes - A route represents the line of movement that a spaceship is assigned to follow from its current location to the destination planet. A valid route emanates from a spaceship and ends at the given spaceship’s destination planet. A valid route is drawn by pressing the finger on top of the given spaceship, dragging the pressed finger across the screen, and releasing at any given point in the environment. The line traced by the finger is recorded as the ship’s route by taking the current location of the finger and adding that point as the next coordinate to travel to for the ship, and is displayed on the screen as a red line. The ship automatically traverses its completed route once the finger is released from the screen. Routes will be followed unconditionally unless redrawn. Routes

can be redrawn by selecting a given spaceship and drawing a valid route. After a ship completes its route, it will continue in the direction of the last two points on the route in a straight line.

4. Bonuses - Bonuses are objects that can be picked up by spaceships to gain extra points. A bonus can appear randomly at any location on the screen. They appear at time intervals according to some probability distribution (in the standard case, this is a uniform distribution). A bonus is picked up by a ship if the ships edge comes into contact with the edge of the bonus. Thus, a ship can pick up a bonus by following a route that leads it into contact with the bonus. Any ship can pick up any bonus. Upon a ship picking up a bonus, the player is allotted the number of points the bonus is worth. Once it appears, a bonus does not move and does not leave the environment unless collected by a ship.
5. No-fly zones - No-fly zones are grey areas that appear on the screen. Each no-fly zone is a small rectangle that covers a portion of the screen. A spaceship that enters a no-fly zone will lose points according to the amount of time it is in the no-fly zone, for each second in the no-fly zone the player loses a set number of points. A spaceship is in a given no-fly zone when any part of the spaceship overlaps any part of the no-fly zone. No-fly zones appear at two static locations. Once it appears on the screen, a no-fly zone does not move.

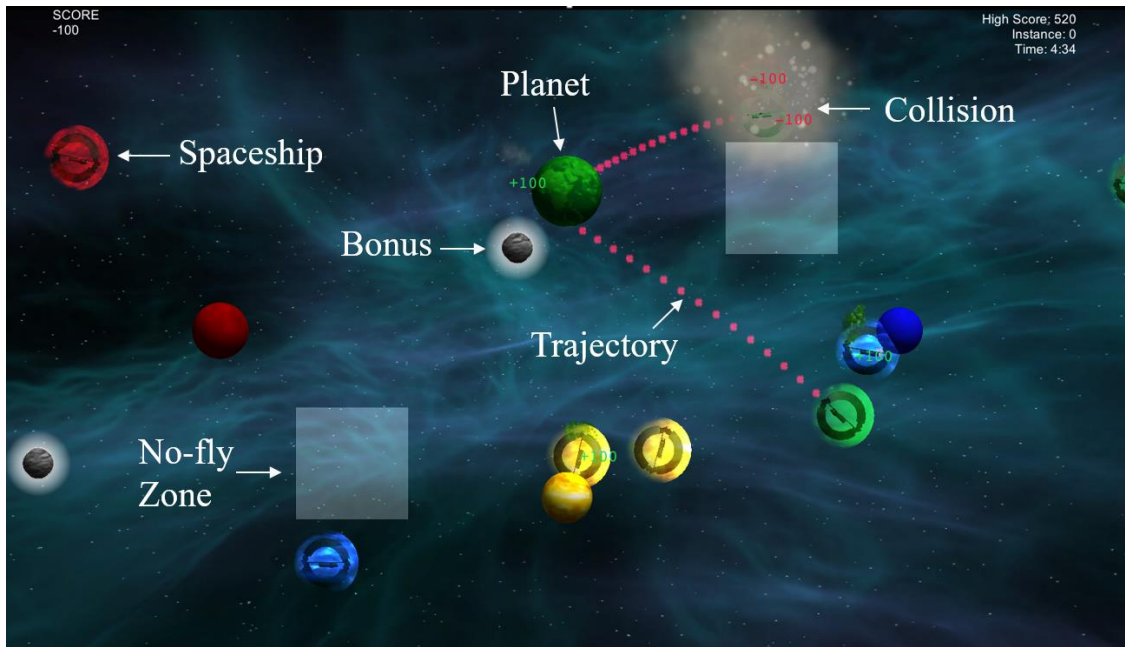


Figure 3. An example of the *Space Navigator* environment

In *Space Navigator*, points can be obtained or lost in four ways:

1. Points are **gained** for successfully landing a ship on a planet - Upon landing on a planet, a spaceship disappears and points are given to the player. All ships, no matter the color, are worth the same number of points if landed at the correct planet.
2. Points are **gained** for picking up a bonus – Points are given for each bonus picked up. All bonuses are valued at the same number of points.
3. Points are **lost** for traversing no-fly zones - Points are lost over time in no-fly zones. A set number of points is deducted from the player's score for each second a spaceship is in a no-fly zone. Each spaceship loses points at the same rate, regardless of the number of ships in the no-fly zone.
4. Points are **lost** when two or more ships collide - When two ships or more overlap on the screen, a collision occurs. All involved ships, and any routes assigned

them, are immediately removed from the screen, and points are taken away from the player. The number of points lost is equivalent to the combined number of points that would be gained had each ship landed successfully. A collision is signified by a graphic showing a small explosion.

2.2.1 Previous Work in Space Navigator

Previous studies using the *Space Navigator* environment include gathering user trajectory data to enable the creation of trajectories similar to what the user has been drawing in the game [8]. Also, this environment was involved in creating a *function-to-task design process model*, which focuses on allocating functions to either the human or machine in a human-machine team [18]. Current work in *Space Navigator* involves human reliance and compliance with a machine-agent's tasks in the environment, such as collision detection, and how the human trusts its machine-agent partner.

2.2.2 Applying Multiplayer and Synchronization to Space Navigator

Because there will only be a small number of players together in the same environment there is no need to make a massive network where peer-to-peer networks are optimal. Instead, a client-server architecture is the best approach because of the small size of the network. Also, having a central server allows all information to pass through that server, where the server can observe and save the data for later analysis. In *Space Navigator* the only inputs the users will be sending through the server will be the trajectories they draw for ships, and while a trajectory is drawn by one player, that information is also passed to prevent concurrency issues with players drawing multiple trajectories for the same ship at the same time.

Synchronization in the new *Space Navigator* multiplayer environment will model after the mirrored architecture discussed in 2.1.3.1, because of the need to optimize performance in synchronizing the environments for each client and the need to administratively analyze all the data transferred between each client playing the game. Synchronization is important because the players will develop strategies and execute those strategies based on the current environments they see. If the players see altered environments, it defeats the purpose of working together because it is as if they are playing different games. Because the information that is passing through the network is relatively lightweight (a linked-list of coordinates for the trajectories and a boolean of whether a ship is currently being drawn), the processing power of the individual tablets will be utilized to process all graphics. The tablet will simply be waiting for the other user to send a trajectory or the current user to draw a trajectory for it to push to the network. By passing as little information across the network as possible, the inherent latency of passing information will not be noticeable to the average player [14] and allow for successful experiments in collaboration in the synchronized environment.

2.3 Communication

This section defines the various ways humans communicate in both virtual worlds and the real world. This section also discusses current research in the area of collaboration between humans in virtual environments.

2.3.1 Human-Machine Communication

Humans and machines interact daily, but not necessarily in a team environment. Humans communicate with smartphones, computers, etc. as the owner, and the machine

responds to the input commands. Human-machine teaming is an alternative where both entities work together to accomplish a task where the human and machine give and take from each other. There are functions that a machine simply cannot do, such as reasoning with unknown datasets; and functions a human cannot do in a reasonable amount of time, such as complex calculations. Finding a balance between which functions humans and machines should perform may optimize teamwork efficiency.

Both human and machine entities need to work together in three levels: Action, Planning, and Meta (or maintenance) [19]. Action is task performance in real-time; planning involves allocating tasks, setting up the machine, and training the human; and maintenance is making changes to the entities as the task progresses. Communication is what makes these levels attainable and is critical as team size increases. Communication has different forms, depending on the type of machines and personalities of the humans involved. Communication between humans and machines is critical in team atmospheres, because when a problem arises for either entity, information needs to be shared to solve the problem and continue the task [20].

2.2.2 Human-Human Communication

Communication between humans and machines may gain efficiencies from understanding how and why humans communicate with other humans. The need for communication stems from a dynamic environment that requires quick decision-making and instant understanding between entities. Human-human communication will be studied in the real-time strategy environment *Space Navigator* to challenge humans in their ability to communicate effectively and perform optimally under changing circumstances.

Outside of virtual environments, humans communicate effectively when they have multiple modes of communication. Humans can simplify their means of communication as well by including context in “conjunction with language to clarify meaning,” known as *deixis* [21]. For example, when trying to describe a specific house to go to on a street, one could say the numbered address, the shape of the house, or the color. However, it is much simpler to point to the house to instantly know which house is correct.

2.3.2 Types of Virtual Communication

There are numerous ways that humans can communicate in person. From talking and using gestures and body language in common languages and cultures, to pointing and showing pictures when there is a language barrier, humans can be very creative when attempting to convey a message. We have passed that creativity into technology as we are able to communicate through text messages with simple acronyms or strings of emojis (a picture that represents an emotion, concept, landmark, etc.) and we are able to send our locations, contact information, and other items about ourselves. These types of communication can be applied when communicating in a dynamic virtual environment as well [21]. When in need, humans tend to find a way to convey a message, and research has been performed to classify the types of communication humans operate with in virtual environments, these include: automated, immersive and expressive, emerging and attention-focusing, and environment-modifying.

2.3.2.1 Automated Communication

When a voice option is not available to communicate in an environment, it may be too time-consuming or a burden for a player to convey a message (e.g. typing a long

message on a touch screen in an action-packed game). *Automated communication* is produced by the game developer that supplies pre-programmed messages that are useful in the game and easily accessible by the player. The two forms of automated communication are *constrained* and *context-sensitive* [22]. Constrained automated communication requires “the player to decide what information to share with teammates, accounting for context manually, [and] activating it through a game mechanic” [22]. An example of constrained automated communication is if the game had a certain button that sent the message “Help Me!” to all teammates. Conversely, context-sensitive automated communication is able to pull context from a player’s avatar or any other relevant information and use the game engine to process and include it in the automated message. In the previous example of the message “Help Me” it would be the job of the teammates to interpret who said the message and where the player needs help. With context included in the message, it could look like “[player]: Help Me [Here],” which makes it easier for the teammates to help the player by defining the player and location.

2.3.2.2 Immersive and Expressive Communication

Immersive communication is cosmetic and communicative in the game environment. It is expressing a player’s character or avatar in a dramatic way that relates to the fiction of the environment while simultaneously conveying a message [22]. For example, if two players are facing each other and Player 1 sees an enemy behind Player 2, Player 1 could force his avatar to start trembling or scream and point, which may cause Player 2 to look behind and see the enemy.

Expressive communication is different than immersive because it may or may not relate to the fiction in the game. Instead, “it includes mechanics that support players in

sharing information about themselves. Expressive mechanics are used for planning and sharing emotions” [22]. In both planning and emotional expressive communication, the action used can either be mechanical (altering the state of the game in a meaningful way) or cosmetic (does not alter the state but changes the way the environment is perceived). Planning is conveying future intent by a player in the game environment. A visual example is if a player has a sword wielded or if the sword is put away, indicating whether the player is about to take action or not. Emotional expressive communication uses player gestures to convey how they feel without using words. This includes examples such as high-fiving, thumbs-up, etc. to show alliance with another player, whereas taunting, joking, or glaring would show hostility. These expressions can also be used to grab attention [23], which can be useful when a task needs to be done specifically but there are no common forms of communication available such as text or voice.

2.3.2.3 Emerging and Attention-Focusing Communication

Emergent cooperative communication mechanics are ways to communicate that were not specifically designed for communication, but have been adopted by the community and understood as having meaning [22]. Players who are new to the environment may not understand the meaning of emergent mechanics.

Getting the attention of other players is the first step in communicating something about an environment. If a player has important information they are sending out, but nobody is receiving and responding to it, it is as if the player did not communicate at all. Attention-focusing cooperative communication mechanics can either be unbounded or semantically imbued [22] [23]. An example of this mechanic is with map pings, which “allow players to identify locations to one another by placing icons on a map” [22]. For

example, in the game *League of Legends*, a multiplayer online battle arena (MOBA) game where two teams push into enemy territory to destroy the enemy base, players may ping on the map with a location that needs attention (Figure 4) but it is up to the players to decide what the ping means (whether to avoid or move towards the location, if enemies are there, etc.). Conversely, if a player pings onto an enemy turret, the ping states to attack the turret, and if the player pings on an allied turret, the ping states to defend the turret. These pings are performed the same way, but because they carry context they are semantically imbued.



Figure 4. The different types of pings in *League of Legends* [24].

Semantically imbued messages do not always have to involve a coordinate of a map. The game *Portal 2* is a cooperative puzzle game where players work together to escape a room. Certain actions must be done synchronously, so a player may initiate a timer ping, which starts a countdown that is displayed for both players so that they can both do a task at the same instance [23]. Built-in actions are designed to optimize performance according to the game developers. However, when they do not suffice in

conveying messages, emerging communication mechanics may be substituted to employ the same messages in a better way.

2.3.2.4 Environment-Modifying Communication

The final form of communication in a virtual environment is by modifying the environment itself. This could involve taking items from a player's inventory and leaving them out in the environment to form a path that leads to a desired location for other players to follow or moving items to form a wall to prevent players from going somewhere. Toups *et al.* [22] researched forty games and the forms of communication used in them. In their research, environment-modifying communication was seen the least. However, this could be prominent in the game *Space Navigator*. Players are able to draw trajectories in the environment and each point in the trajectory remains on the screen until the ship passes that point. By modifying the environment, and drawing ship trajectories that resemble arrows or words, players may be able to communicate successfully when no other form of communication is available. This form of communication may be rare in games with much more dynamic worlds because of the time it takes to modify the environment and convey a complex message. But in *Space Navigator*, it may be efficient because the only input players can make is drawing trajectories and a word or shape could be drawn that still accomplishes the goal of drawing a ship's trajectory to its proper planet.

2.3.3 Current Collaboration Research

A research team is “working to advance an understanding of players' in-game use of cooperative communication mechanics to support more engaging design as well as learning through gameplay” [23]. The experiment involved three pairs of people, where

each person in the pair knew the other prior to the experiment. Players of different skill levels were asked to complete 3 puzzles in *Portal 2* and rate the difficulty of each puzzle. The research team is first validating that the new puzzles they created are difficult for beginners and easy for experts. Their initial results showed that their hypothesis was correct. The goal is to use these levels in further experiments that pair experts and beginners together which would require communication between the two to accomplish the puzzle. The team's future goal is to understand the most effective means of communication between pairs and how cooperative communication mechanics support gameplay.

Another research group identified the rarity of collaborative planning with contingencies in gameplay [25]. Having previously worked with disaster response teams, the team determined that the essential characteristics of planning are gathering information and developing intelligence (distinguishing between what is true or false based on reports from sources), emergent objectives (changing plans efficiently in response to new intelligence), and collaboration (synthesizing perspectives) [25].

To support collaborative gameplay and make it successful, the team suggests three baseline game design patterns: shared expressive workspaces, executing plans, and emergent goals [25]. Shared workspaces allow all members of a team to bring diverse opinions and perspectives to a single environment to compare and contrast ideas on goals and contingency plans. Executing plans is a simple design but it ensures a deterministic approach in situations that are similar. If plans are executed exactly as planned in the same environment in the same state, these plans can be evaluated on how successful they are and changes can be applied to future occurrences of the same situations. Finally,

emergent goals relates to reevaluating strategies and adapting goals when introduced to new intelligence. This also allows players the ability to recover after failures and still succeeding instead of simply failing and having to start all over again.

2.3.4 Communication Applied To Space Navigator

There are multiple ways for people to communicate through a game environment as outlined previously. From automated communication and emotional expressions, to focusing attention and modifying the environment, humans will find a way to communicate depending on the environment they are in. Current research is studying how humans communicate using cooperative communication mechanics when the pair knows each other and there is an overall solution to the problem (i.e. in *Portal 2* players need to get from point A to point B in each puzzle). Research in *Space Navigator* will instead study the communication between two strangers and observe the performance when humans can communicate versus when they cannot communicate to create a baseline in the communication in the *Space Navigator* environment. Humans could converge on a specific strategy or on a specific form of communication within the environment, or the results could be completely scattered and diverse, but this will not be known until human subject experiments are performed which are outlined in Chapter 3. Also, *Space Navigator* does have objectives and sub-objectives, but there is no true end state, the game ends after a set time determined prior to playing. There are multiple solutions to the game and observing the strategies and results of gameplay from the humans along with their communication will be helpful in establishing the baseline on performance. The future goal of these results is to determine the best forms of communication, which are defined by how well the player perform as a team, and apply

those to a human-machine team so that the human can trust and communicate with a machine-agent just as the player would with another human.

2.4 Summary

This chapter presented the basic implementations of multiplayer games and current methods to synchronize games across multiple players. Real-time strategy games were discussed to give background on the domain of *Space Navigator*. The *Space Navigator* environment was described in detail to understand the environment of the experiment. Multiple forms of communication for multiplayer games were defined as well to help categorize the forms of communication that will be observed in testing. Finally, this chapter presented current research in collaborative communication mechanics and how this research and its experiments are different both in the experiment set up and the future goals in utilizing the results.

III. Methodology

Chapter Overview

The purpose of this chapter is to explain how to answer the research question “How do human machine teams adapt to a changing environment when they can communicate versus when they cannot communicate?” This chapter describes how *Space Navigator* developed from a single player environment to a multi-agent environment to support an experiment that aims to answer the research question. It will also discuss the rule changes made to the environment to force strategy adaptation by players in the game. It will then include a detailed outline of the human-subject test performed using the new *Space Navigator* environment to study how humans adapt to changing environments when they can and cannot vocally communicate. The expected results for this experiment are that human teams that can communicate will produce higher scores in the environment, the teams that can communicate will establish more efficient and organized strategies than the teams that cannot communicate, and that the teams who can communicate will adapt to changes in the environment differently than the teams who cannot communicate.

3.1 Multi-Agent *Space Navigator* Environment

To utilize the *Space Navigator* environment to test the hypothesis, the environment needed features for multi-agent playability and the environment needed new game rules to force players to adapt their strategies. This section will outline the design of the multi-agent environment, the design of the environment changes that will force

players to adapt their strategies, and the initial results that validate the environment changes.

3.1.1 Multi-Agent Design

The new multi-agent design for *Space Navigator* allows two human players to connect to each other and interact within the same environment in real-time. The multi-agent design uses a client-server architecture (Figure 5), where all players use client machines to play the game and connect to a central server to pass and receive all game state information. The client machine process graphics because the computations involved are too large to pass over a network in real-time. Whenever a client draws a trajectory, the client is responsible for sending that trajectory information to the server so that the server can update the client's partner. The client will also be responsible for storing the information about the environment to a local hard drive because the client's partner does not need to handle that information.

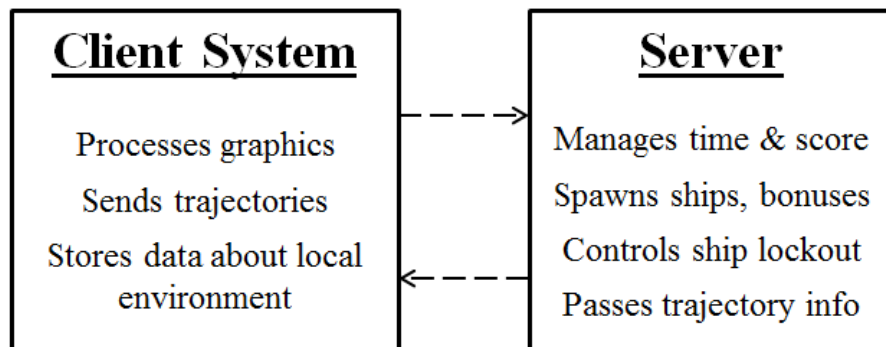


Figure 5. An overview of the functions of the client and server in *Space Navigator*.

The server is a local, stand-alone machine that does not connect to external networks and hosts its own local network using a wireless router. Client machines receive local IP addresses to identify each other and the server, which allows all *Space Navigator* games

to connect and run on the local network. Shown in Figure 5, the server is responsible for managing the time remaining in a game and the scores for both players so that they can base decisions off similar information. Also, the server will spawn ships and bonuses in equal locations for both clients to maintain a synchronized environment. The server will keep information on who is drawing on a specific ship so that it can lockout the other client in the case that two partners want to draw on the same ship and avoid a problem. Finally, the server will pass trajectory information from one client to the other so that the environment updates in real-time.

3.1.1.1 Synchronization

The first problem to solve in creating the multi-agent environment was synchronization. Synchronization is difficult in *Space Navigator* because there are multiple ships and bonuses spawning based on timers, ships move on specific trajectories drawn in real time, and scores must be updated based on specific events in the environment. Photon Unity Networking (PUN) [26] uses functions called Remote Procedure Calls (RPCs) that are efficient for the synchronization of *Space Navigator*. RPCs are effective for events that occur when triggered by some action. For example, in *Space Navigator*, whenever the score changes the new score is updated on the server for the specific game instance, and that change is sent out to the specific clients connected to that room to update local scores. RPCs execute on all clients in fractions of a second and give each player on the client machines the most updated score from the server. The server also manages the time to account for unforeseen delays in loading the game environment from the individual tablets. As time changes, an RPC informs the clients in

the room, updating their current time. The server handles spawning ships and bonuses the same way, except with different wait times for each object.

A trajectory in *Space Navigator* is an array list of coordinates that are stored in the ship object. Whenever a player touches a ship on the client machine, the ship stops moving so that the player is able to define a new trajectory by dragging his finger across the screen which outlines a new path. The software stores the points created in this process as an array of linked points. When the player releases his finger from the client machine, the ship begins to move along the trajectory at a constant speed. Once the ship reaches the end of its trajectory path, the ship continues in a straight line determined by the last two points in the trajectory.

Ships are relatively large data objects that carry 3D graphical information as well as their properties of trajectory, speed, and color. Synchronizing the entire object in near real-time over the network is not possible due to the size of an entire Ship object. To simplify the synchronization of the ships' locations and trajectories, the server only sends the trajectory information across the network.

3.1.1.2 Mutual Exclusion

The second problem to solve was to ensure that each ship in all client environments was mutually exclusive. Mutual exclusion means that if there are two actions possible (Player A draws or Player B draws), only one action can occur at a time, and the other action must be prevented. A trajectory can be changed at any time by any player, however, if Player A is currently drawing a trajectory for a ship, that ship cannot be selected by Player B until Player A has finished drawing the trajectory by releasing his finger. RPCs are used again to synchronize the trajectories for each ship in the

environment, and an overview is shown in the behavioral diagram in Figure 6 [27]. The client and server are entities that communicate with each other. The figure denotes communication with dashes. The actions inside solid, rounded rectangles are actions the player or server performs without contacting the other entity. Each action has a numbered rule (e.g. Rule 1). The diagram flows using the arrows and if any action does not hold true, the sequence returns to Rule 1.

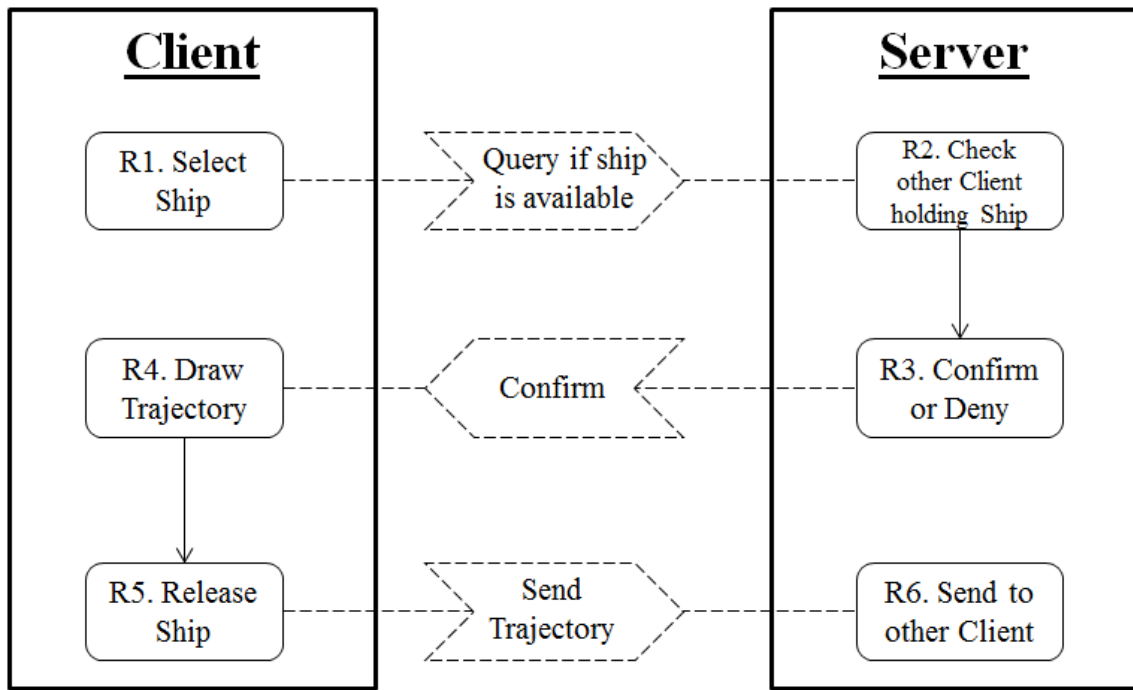


Figure 6. A behavioral diagram showing how a player draws a trajectory.

Whenever a ship is selected by Player A, the ship's unique identifier is sent to the server which passes the information to Player B to duplicate the actions of Player A. First, the environment destroys the ship's previous trajectory. Second, the server sends an RPC to Player B to hold the specific ship in its current location. Third, an RPC updates Player B that Player A is holding the ship. During the time Player A is drawing a trajectory for that ship, Player B is not able to select that ship. Fourth, the server copies

Player A's trajectory and sends an update to Player B's ship in real-time. Finally, when Player A finishes the trajectory, the ship begins moving on its trajectory on both players' tablets and either player may now select it. To discern between each player's trajectories, players will always see their trajectory colored red and the other players' drawn trajectories colored blue by their local environment.

3.1.2 Dynamic Environment Design

In a previous experiment utilizing the single player environment of *Space Navigator*, participants answered the following question after completing multiple instances of the game dealing with compliance with an automated agent: "If you were required to divide the workload of *Space Navigator* between two people, what are some examples of how you would do that?" 43 participants completed the survey question: 26 female and 17 male. The mean age was 23.3 years with a standard deviation of 5.00 years, a median age of 21 years, and a range of 18-38 years. Six participants gave two suggestions resulting in 49 total suggestions in dividing workload.

Table 1. How many times participants suggested general strategies.

Division of Workload	Number of Participant Suggestions
Divide physical environment in half	8
Divide ships by color	26
Divide trajectory drawing and collision detection	11
Other	4

The four general categories for dividing workload are shown in Table 1 along with the number of times a participant suggested each strategy. The “Other” category did not fit into a general strategy or the answers were too vague to categorize, such as using “one person as an agent.” The responses to the survey question were typed by participants and subjectively placed into these categories based on interpretation. Examples of responses were “one person should set up routes for each ship and the other should watch for any possible [collisions] and make changes to their routes” and “assign each person two colors to control.”

The categorization of the responses resulted in 53.1% of participants suggesting that workload be divided among players by colors of the ships, while only 16.3% suggested dividing by sides of the screen and 22.4% suggested detecting collisions and managing trajectories. These preliminary results verify the most common strategies participants will use when dividing workload in *Space Navigator* and will be used to create the instances where rules change in the game environment. This should persuade all participant teams to develop a new strategy at some point in the experiment unless a new strategy emerges that was not confounded in any of the instances, such as one player dominating and doing all of the work, which is not confoundable.

To characterize how human teams adjust strategy to a changing environment, the environment must change in a way that requires the players to adjust. The environment will change by confounding the ways in which humans naturally divide workload. Since there are multiple possible strategies players can use as teams in *Space Navigator*, Table 2 presents several ways to confound the potential strategies and force adjustments. It shows initial ideas on dividing workload which allowed the design of game instances that

confound these strategies and force the players who use them to develop a new strategy. Not all participants will divide workload evenly due to varying personalities and skill in the game so the environment cannot stop one teammate from handling all the workload, i.e. altering the authority of players in the game.

Table 2. How players may divide workload and how to confound those strategies

Division of Workload	Confounding the Strategy
Divide physical environment in half	Spawn all ships on one side of the environment
Divide ships by color	Color all ships the same
Divide trajectory drawing and collision detection	Reduce collision score deduction

For each instance, there were only minor changes made to the source code which allowed for the overall structure of the game to be similar to the normal game settings. Making only slight changes to the game environment allows for the isolation and study of those particular changes and how humans would adapt specifically to those changes.

3.2 Human-Subject Test

With the completion of a multi-agent, changing environment, humans were able to play *Space Navigator* together. The humans worked together to perform optimally as a team but communication was limited to vocals and/or any creative ways that the players came up with in communicating within the environment. The humans were subjected to rules changes that forced them to adapt their strategies from the familiar, normal settings

of the game. This section of the chapter outlines the human-subject test performed using the new multi-agent *Space Navigator* environment.

3.2.1 Subjects

16 subjects were recruited for this experiment. The population included a mix of active duty military and Department of Defense civilian students and professors. Subjects were first recruited from the population of AFIT students and faculty.

The only physical requirements for inclusion were that participants were able to hold and operate a tablet computer and not be color blind. A simple color-blindness test was administered to determine color vision. No special subjects were involved and no compensation was given to the subjects. The time commitment for each subject was one approximately two hour session. The principal and associate investigators recruited subjects in person using snowball sampling. Finally, the age range for all subjects were from 22 years old on up.

3.2.2 Equipment

The only equipment the subjects interacted with were the tablet computers and headsets with microphone. The subjects primarily used the tablet computer for interacting with the study environment. The headsets served the purposes of cancelling noise when communication is not allowed and recording audio through an attached microphone throughout the study.

3.2.3 Description of Experiment

Four sessions with four participants each took place. Each session was the same, except for that in half of the sessions vocal communication between teammates was

allowed and in the other half, vocal communication was not allowed. The sessions were randomly assigned and designated as:

- a. Group X: No verbal communication between teammates allowed
- b. Group Y: Verbal communication between teammates allowed

Each session consisted of two phases. The first phase was used for establishing a baseline for all players. The second phase allowed all participants in the session to pair with each other in the form of a round-robin.

Phase 1: Establish Minimal Baseline

The purpose of this phase was to briefly familiarize users with the game environment equally. The experiment took place with 4 users. Each user had their own tablet to interact with the Space Navigator environment. The experimenter described the rules of the game and defined the terminology of the game to the group. The experimenter also assigned each user a player letter to ease the process of pairing partners. Afterwards, all users were sent to their own space where they were not able to see each other. The users first started a five minute instance of the game individually whenever they were ready to start.

Phase 2: Round Robin

The four users were placed in their own spaces where they were not able to see or hear the other user. The users were prompted to wait for the game environment to load on the individual tablets. Once the users were connected to the correct partner on their tablets and both users were ready to begin, the game counted down to a start and the users began to play in the same game environment for a 5 minute instance. The users in Group

X were not able to communicate with each other while the users in Group Y were able to communicate with the partner they were cooperating with in the game environment. User pairings are shown in Table 3.

Table 3. Player pair ordering for the human-subject test

	Player Pairing		Instance Order	Player Pairing		Instance Order
1 st Round	A	B	1,2,3	C	D	1,2,3
2 nd Round	A	C	1,4,5	B	D	1,4,5
3 rd Round	A	D	1,6,7	B	C	1,6,7

Each participant was assigned a random designator as either Player A, B, C, or D. To begin pairing the partners, Player A and Player B worked together and Player C and Player D worked together first. For each pairing, the team of players cooperated in 3 complete game instances. Each game instance started when both players were ready. The first game instance between two players was always Instance 1, where the game was defined as “normal” to establish a base of communication between the players. Each game instance afterwards had a significant change from the previous game instance and was not seen by either player beforehand.

Each player was paired with a different partner to allow different types of players to work together. Players had different styles and personalities when playing the game and this round-robin style pairing allowed different combinations that could affect the ways humans communicate and adapt to the different rule changes they see. Also, the players never saw the same instance twice, besides Instance 1, because the players should not have set strategy for the rule change they are about to encounter. The goal of the experiment was to see how the teams adapted to new environments, which was why 7

different instances were created. Each team saw a completely new rule change after their first, baseline game together. All players got to work in all 7 instances throughout the experiment.

The users were able to see each other's drawn trajectories and all aspects of the game were synchronized. The users were encouraged to vocalize what they were thinking and told that all things said by the user would be recorded for future analysis. The goal for the users was to collectively attain the highest score possible. Each pairing of users played together for three complete game instances. A complete game instance was an uninterrupted 5 minute game where pauses were not allowed. Each complete game had some significant rule change to the environment, defined as Instance 1-7 and summarized in Table 4.

Table 4. Summary of changes occurring in environment.

Instance Number	Description of Change
1	No change to the normal game rules of <i>Space Navigator</i> .
2	North, south, and east planets are rearranged into a triangle on the right half of the screen.
3	Only blue ships can pick up bonuses.
4	Bonuses are worth 250 points
5	All ships are colored yellow.
6	All ships spawn from the left side of the screen
7	Collisions are worth 0 points.

With exception to Instance 1, the changes to the environment occurred when half of the game-time was remaining in the game instance. The players were not notified

about the specific change to the rules that occurred and simply needed to observe and adapt to the change for themselves.

1. Instance 1 - Game environment was considered “normal.” The planets were placed in their usual locations of north, south, east, and west, all equidistant from each other. The ships spawned at a rate of 1 ship every 2 seconds. The ships spawned from a random side on the screen with a straight trajectory to a random location. Each planet had its own color and each ship was assigned with a random color that belonged to a planet to designate which planet the ship should travel to. Bonuses spawned at a rate of 1 bonus every 10 seconds in random locations in the environment. Any ship was able to pick up any bonus. Two no-fly-zones were placed in static locations and any ship that passes through the no-fly-zone decreased the total score for each second the ship was in the no-fly-zone.
2. Instance 2 - All the aspects of the game were the same as Instance 1 except that the north, south and east planets were arranged in a triangle on the right half of the environment. The west planet remained in the same location.
3. Instance 3 - All the aspects of the game were the same as Instance 1 except that the bonuses were colored blue, and only blue ships could pick up the bonuses.
4. Instance 4 - All the aspects of the game were the same as Instance 1 except that the point value for the bonuses increased from the normal value. Normally, the bonuses were worth 50 points for a pick up. This instance increased the bonus value to 250 points, two and a half times as much as a ship reaching its planet.
5. Instance 5 - All aspects of the game were the same as Instance 1 except that the ships all spawned as the same color, yellow.

6. Instance 6 - All aspects of the game were the same as Instance 1 except that all ships spawned from the left side of the screen.
7. Instance 7 - All aspects of the game were the same as Instance 1 except that collisions between ships deducted 0 points from the total score. Normally, ships that collide deduct 100 points for each ship in the collision.

3.2.4 Data Gathering

Throughout a game instance, a number of data items were stored in order to garner insight into user gameplay. First, each game was associated with a specific user. Secondly, at the point when a route is entered, a snapshot of the current environment was taken:

1. Number and location of each bonus
2. Number and location of each other spaceship already present
3. Presence and location of no-fly zone
4. Location of destination planet
5. Location and cardinal direction of selected spaceship
6. Whether or not each other spaceship has a valid route already
7. Observation time stamp

These data were captured as a series of vectors pertaining to a gridded representation of the game space. The screen was broken down into a 20 x 15 grid, with each square representing a portion of the screen. The presence (or lack thereof) of each of the different items was recorded as a vector for each input snapshot. Additionally, the user's

response vector (route) was recorded, and correlated with the given environment snapshot.

Participants who were not able to communicate with their teammate had all audio recorded, as they were encouraged to voice their opinions and thoughts while playing the game. Participants who were able to communicate with their teammate had their audio conversations recorded to document behavior in communicating with their assigned teammate participant. Communications that were documented include:

1. What participants say to each other
2. Forms of communication that emerge in the game environment
3. Division of work

These data observed were documented through audio recordings and snapshots taken in the environment, and organized into categories that fit the descriptions of how the players communicated.

Throughout the experiment, participants were asked to complete questionnaires. Before the experiment began, participants completed a questionnaire to collect basic demographical information (Appendix A). Next, when each participant completed three game instances with a partner, the participant was asked to complete a questionnaire regarding only those three instances and that partner (Appendix B). The questions involved rating the partner on overall cooperation, communication, and strategy development. The participant also rated specific aspects of their partner's communication such as: quantity (how much information they gave), quality (how truthful was the information communicated), relation (how relevant was the communication), and manner (how clear was the communication). The participant also answered questions that asked

about initial strategies in each of the three instances and how those strategies changed throughout each instance.

After each participant went through the round-robin pairing, and all game instances were completed, each participant ranked their partners against each other on the same qualities that they rated each partner previously (cooperation, communication, and strategy development) (Appendix C). Participants were also asked about the forms of communication they used to potentially see if new forms emerged specific to this environment. The participants were asked to state their most effective and least effective strategies. Finally, participants were asked to complete a personality test to determine the types of players within the teams (Appendix D).

3.3 Summary

This chapter described the development and evolution the *Space Navigator* environment. Moving from a single-player to a multi-agent game allows for the study of multiple humans playing together, and also facilitates the possibility of adding machine agents to the games as well. This chapter also defined the *Space Navigator* environment and the human-subject test to be performed using it. The human-subject test will analyze the way humans cooperate, communicate, and adapt their strategies to rule changes in an environment.

IV. Analysis and Results

Chapter Overview

The purpose of this chapter is to explain how the results obtained from performing the experiment outlined in the previous chapter answer the thesis question and investigative questions. The results will focus on how both experiment groups converged on strategies, how both experiment groups changed in reaction to a change in the environment, and finally how the groups compared in overall performance of the *Space Navigator* game.

4.1 Comparing Convergence on Strategies between Groups

Before analyzing how both of the experiment groups adapted to changes in the environment, it is important to first look at how the groups perform in the environment under normal conditions. All participants played a standard game of *Space Navigator* without a partner for 5 minutes allowing them to acclimate to the touch screen controls, understand the rules of the environment, and develop their own strategies. After this baseline game of *Space Navigator*, participants were asked “How would you divide the workload of *Space Navigator* among two players?” The distribution of the provided answers is shown in Figure 7.

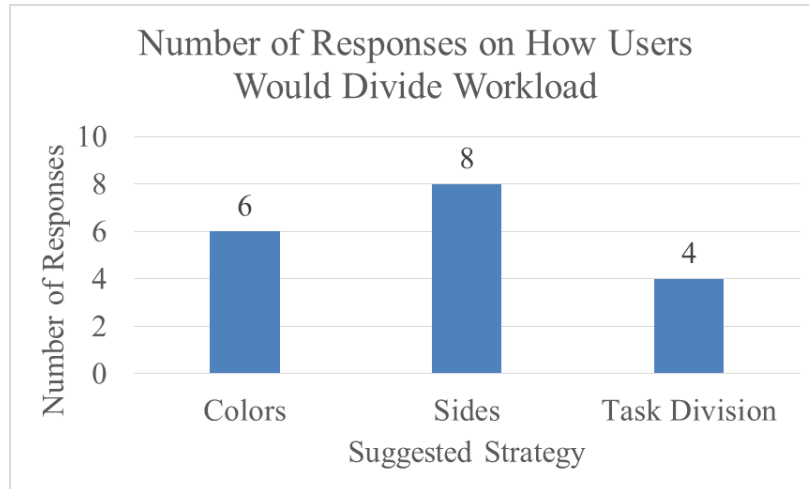


Figure 7. How participants responded when asked how they would divide workload in *Space Navigator*.

Participants answered the question by writing out their answers and two participants suggested two strategies which results in 18 suggested strategies. The strategies are defined as follows: *colors* means the participant suggested that each player only draws trajectories for two of the four available colors (e.g. only red and blue ships or only red and green ships), *sides* means that the participant suggested that each player only draws trajectories for ships on a designated side of the screen (e.g. left side only or top side only), *task division* means that the participant suggested that each player should focus on either the task to draw initial trajectories for all ships or the task to avoid collisions for all ships.

Under normal game conditions, ships spawn randomly on one of four sides of the screen and ships are randomly colored one of the four available colors. Dividing by either colors or sides is favorable because those strategies stand out as dividing the work evenly among two players. Dividing the work by task does not present an immediate clear division of workload, because the strategy involves more understanding of the rules and

point system of *Space Navigator* not developed by playing alone for five minutes.

Participants initially suggested the *colors* and *sides* strategies due to the strategies' simplicity, visual aspect, and lack of need for advanced in-game knowledge.

4.1.1 Non-Communication Group

In each grouped run of the experiment, two teams of two participants played 9 total game instances, resulting in 18 total team and game instance configurations. Having two sets of non-communication groups presents 36 total data points of teams reporting both initial and final team strategies. Figure 7 depicts the strategy suggestions among all participants. Within that data, the participants who could not communicate with their partners had 4 participants suggest *sides*, 3 participants suggest *colors*, and 1 participant suggest tasks. After answering the question about dividing workload, participants played 9 games of *Space Navigator*, every 3 games switching to a new partner. The participants reported what initial team strategies they used for each game. If there was a change in their strategy during the game at any time, the participants also reported whether or not a change occurred in the environment. Figure 8 shows how the teams of participants in the non-communication group reported their initial strategies, while Figure 9 shows how the same teams reported their final strategies.

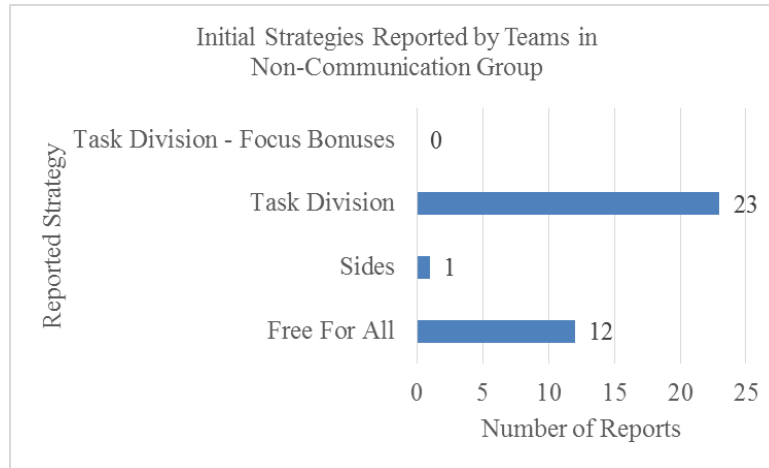


Figure 8. How non-communication teams reported initial strategies.

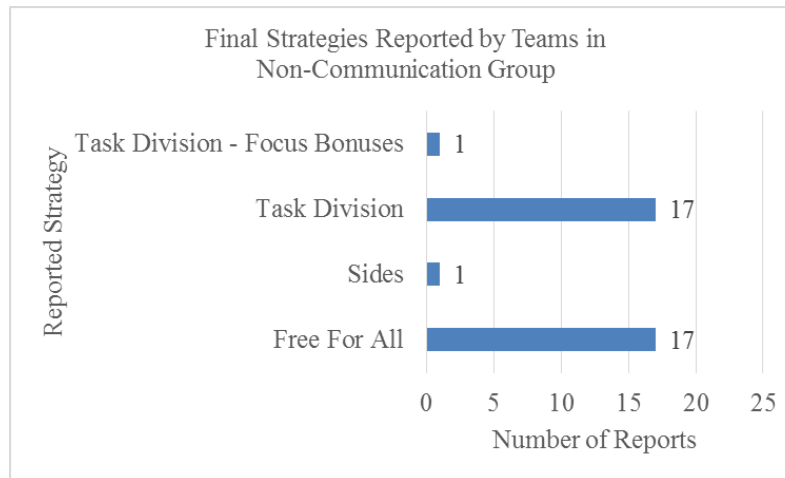


Figure 9. How non-communication teams reported final strategies.

Two new strategies emerged from the initial suggested strategies after the participants played with their partners: *task division – focus bonuses* and *free for all*. *Task division –focus bonuses* is defined by the participants as following the *task division* strategy, but also to actively try to pick up any bonus they see in the environment. This strategy emerged only one time and the team reported this strategy after the environment change occurred that altered the point value of bonuses to make them worth five times their original value. The second new strategy that emerged was *free for all*. This strategy

is defined as actively using no other defined strategy. Many participants reported their strategies as “just play,” “*free for all*,” or that they did not use a strategy.

While 87.5% of participants had intentions to divide the workload by either colors or sides, *sides* composed only 2.8% (1/36) of the reported final strategies and the colors strategies was never reported. Using the strategies of *colors* and *sides* requires communication to decide who is controlling which half of the screen or which two colors, something that the non-communication group never established. Reports from some of the participants include “[My partner] changed a lot of my decisions” and “I no longer trusted [my partner].” Without communication, the group converged on the strategies of *task division* and *free for all*.

The *task division* strategy was reported 63.9% as an initial strategy and 47.2% as a final strategy. The *free for all* strategy was reported 33.3% as an initial strategy and 47.2% as a final strategy. Combining these strategies together make up 97.2% and 94.4% of all reported initial and final strategies, respectively. Lack of communication and reports of frustration from the participants caused them to disregard strategy and play however was in their best interest which created the unforeseen *free for all* strategy. Participants within the teams that established a *task division* strategy reported not wanting to “step on [their] partner’s toes” and to “sit back and make corrections.” Although the initial question on dividing workload before playing with partners only had one response suggesting a *task division* strategy, the non-communication group of participants heavily converged on both the *task division* strategy and on a *free for all* strategy due to lack of communication, organization, and trust.

4.1.2 Communication Group

The communication group responded in similar proportions to the non-communication group on how they would divide workload in the *Space Navigator* environment among two players, with the exception of two more suggestions for the *task division* strategy. Three participants responded with the *colors* strategy, four participants responded with the *sides* strategy, and three participants responded with the *task division* strategy (two of these participants suggested *task division* as well as another strategy). Figure 10 shows how the teams of participants in the communication group reported their initial strategies and Figure 11 shows how the same teams reported their final strategies.

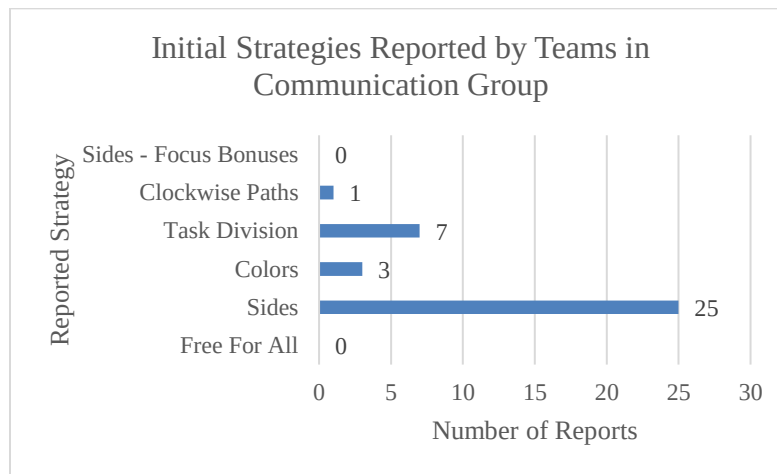


Figure 10. How communication teams reported initial strategies.



Figure 11. How communication teams reported final strategies.

Two new strategies emerged from the communication group. The strategy *sides – focus bonuses* is similar to the *task division – focus bonuses*, where the team continued their strategy of splitting the work by sides of the screen, but made picking up the bonuses a priority. This also happened in the game instance where bonuses were worth five times their original value. The strategy *clockwise paths* involved forcing the ships to move in a clockwise direction to avoid them colliding straight into each other. This was a strategy that a team tried only one time to decide if it was effective and ultimately it was not used again throughout the experiment.

The communication group was more persistent in using the strategies they devised earlier throughout the experiment and the majority of teams converged to using the *sides* strategy. Communication enabled these teams' strategies to persist even though their strategies were not ideal. The *sides* strategy was reported 69.4% as an initial strategy and 52.8% as a final strategy. The decrease in reports for *sides* as a final strategy results from the game environment changing and some teams deciding another strategy was more practical for the change in the environment. The *task division* strategy was the

second most reported strategy for the communication group at 19.4% for both the reported initial and final strategies. And third most was *colors* at 8.3% of the reported initial strategies and 11.1% of the reported final strategies.

Participants in the communication group had fewer reported issues with trust and disorganization among their respective partners. Their responses were clear as to what their strategies were. During the experiment, the partners would take some time before starting the game to discuss what each of their strategy ideas were, and then agree on the best strategy among the partners. The *sides* strategy was the most dominant because a rectangular environment, such as *Space Navigator*, can be divided before anything is shown on the screen. The participants can focus their vision and workspace to half of the environment whereas a single player must focus on the entire screen. With the *colors* strategy, partners are forced to survey the entire screen for their color, and when drawing trajectories for a specific color, it may interfere with their partner's workspace and cause collisions or miscommunication issues among the partners. The *task division* strategy relates back to the non-communication group in not "stepping on each other's feet" while playing. The initial trajectory drawer can focus on ships on the outer part of the environment by beginning to guide the ships to the planets. As the ships get closer to the planets, the collision avoider focuses on the smaller area near the planets to get the ships to their planets safely. In this strategy, the partners are dividing the screen space, but in a slightly different way with different task operations as compared to only drawing a trajectory for a ship to its corresponding planet. Ultimately, the communication group participants heavily favored the *sides* strategy due to the ease of dividing a rectangular environment evenly among two players.

4.1.3 Conclusion

Participants in both the non-communication and communication groups had similar proportions of answers on how they would divide workload in *Space Navigator* among two players before ever playing with a partner. However, the groups converged differently on the strategies used most frequently. The non-communication group favored the *task division* strategy and the *free for all* strategy because they could not communicate their ideas and form a working relationship with their partner. If no strategy was working, the participants in the group played as if they were playing by themselves. The communication group was much more organized and favored dividing the screen by *sides* because of the even divide in workspace and workload for each player.

4.2 Overview of Strategy Change as Environment Changes between Groups

Determining how specific strategies change as the environment changes is often a subjective process that relies heavily on the answers provided by participants. For the purpose of this study, objective measures were created to determine trends in how participants reacted to changes and to prioritize specific areas that should be further examined in future research. Based on the three most popular strategies in *Space Navigator* – *colors*, *sides*, and *task division* – the following categories were examined: draws on each colored ship, draws on each side of the screen, and general trajectory draws (initial trajectories and redrawn trajectories).

Each category used the following method of calculation to determine a change in player performance or behavior after a change occurred in the environment:

- For each experiment group, each team of players for every game instance provided data needed for analysis (i.e. for one experiment group, there are 9 game instances with two teams playing each game instance, creating a total of 18 teams per experiment group).
- The teams divided into two columns, Player A and Player B.
- Next, analysis counted the number of actions for the specific category being analyzed for each player.
- Then, the proportion of Player A's vs Player B's action can be calculated by dividing their count by the total number actions.
- This process is done once for the time period before a change occurs, and a second time for the time period after the change occurs.
- Finally, the change in proportion [28] of a player taking actions is the absolute value of the difference between the same player's proportion before and after the change occurred in the environment.

Table 5. Example of determining change in performing an action for players in a team.

Before Change Occurs					After Change Occurs					Diff. of Proportions
Player A	Player B	Total	Proportion Player A		Player A	Player B	Total	Proportion Player A		
10	15	25	0.4		15	10	25	0.6		0.2

Table 5 shows an example of determining how a team's change in proportions among its player's is calculated for game instances. The difference of 0.2 represents the change for both players. Player A had a 20% increase in the number of events and Player B had a 20% decrease in the number of events. Therefore, the difference of proportions

for a team is defined as 0.2 in this example, because both players moved their proportion 20%.

After all teams were separated and difference of proportions were determined, an average difference of proportions was calculated for all teams in both the non-communication group and the communication group. An assumption of a normal distribution among the difference of proportions was made to define the 95% confidence intervals. These intervals were used to determine if the data between groups were significantly different and further investigation was required. The remainder of this section analyzes those averages and how they compare among groups.

4.2.1 Difference of Proportions Comparison for Colored Ships

The first objective measure to examine was how players adapt their trajectories based on the colors of the ships across all game instances. The expected result was that no color would be more dominant than another and that players would not suddenly increase or decrease drawing a specific color after a change occurred in the environment unless the change specifically targeted ship colors. Figure 12 shows the results of how teams' proportions changed as the environment changed dependent on the color of the ship.

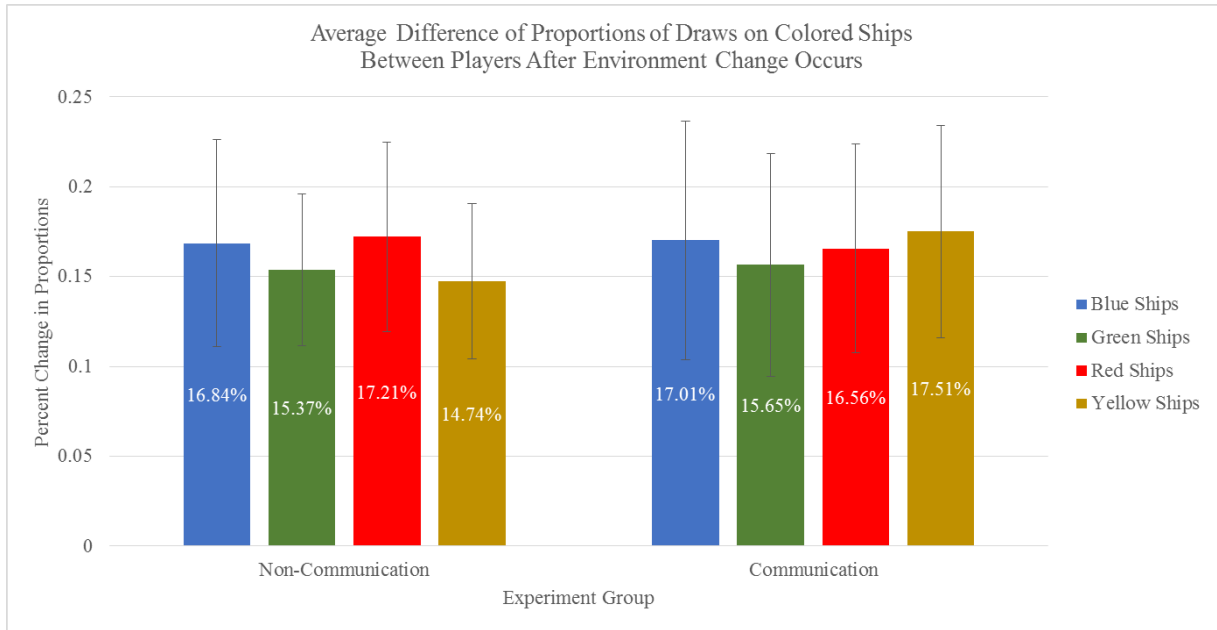


Figure 12. Difference of proportions for colored ships.

Analyzing how teams changed their proportions for specific colors showed that there was no significant difference between the teams that could communicate and the teams that could not communicate. Although teams are changing their proportions slightly on every color, an average of 16.4% per color, these changes are statistically similar and consistent between the two different experiment groups. There is a greater increase in the difference of proportions between the experiment groups for the yellow ships because there was a specific game instance that changed the rules so that only yellow ships were spawned into the environment. The communication teams could decide how to divide the work among ships of all the same color more efficiently in the short amount of time given to adjust to the change. Running this experiment with more participants, however, should decrease the difference since the communication group favored dividing the work by sides and the change in ship colors should not affect that strategy because ships are still spawning from all sides. Because of the overall similarity

between the groups, the overall adaption to change based on colored ships was not different than expected and other objective measures were examined to analyze how players adapted overall to the changing environments.

4.2.2 Difference of Proportions Comparison for Sides in the Environment

After examining the colored ships in the environment, a similar comparison was used for the four different sides of the environment: left, right, top, and bottom. Again, the goal was to determine if players are favoring or rejecting one side of the environment after a change occurs. The expected result was that players would not begin favoring different sides after an environment change occurred unless forced to by the specific environment change.

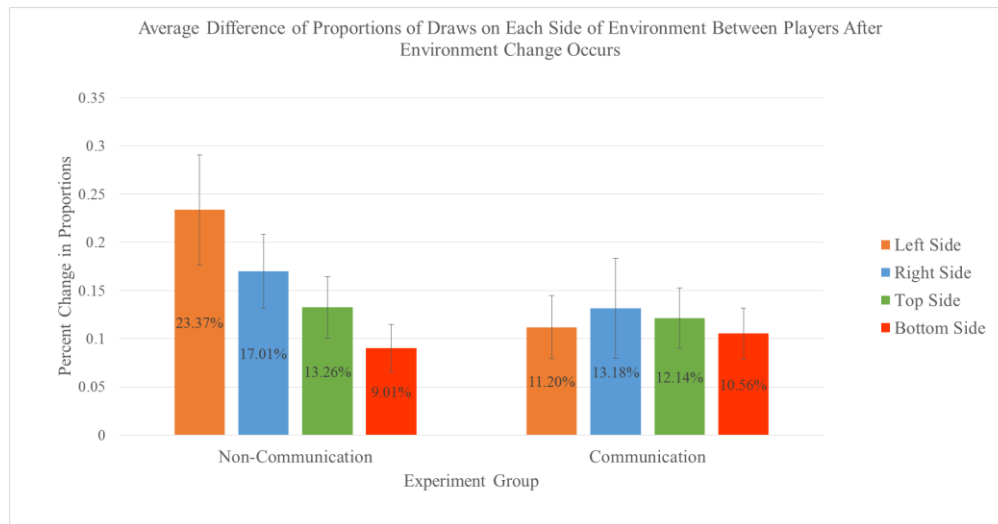


Figure 13. Difference of proportions for each side of the environment.

Similar to the colored ships comparison, there are no statistically significant differences between groups in how the teams draw on the top, bottom, or right sides of the environment due to no overlap of the 95% confidence intervals. The average

difference of proportions per side was 16.6%. However, Figure 13 shows that the average change for the left side is 12.57% higher for the non-communication than the communication group, which is statistically significant with no overlap of the 95% confidence intervals. The teams that could communicate needed to make some adjustment, affecting the proportion of left side draws for each player more than before the change occurred when all ships spawned on the left side of the screen.

4.2.3 Difference of Proportions Comparison for Total Trajectories, Initial Trajectories, and Redrawn Trajectories

After dividing the environment physically by both colors and sides, the final objective measure to examine was how players were drawing their trajectories. Along with measuring the proportion changes among all trajectories, initial trajectories, and redrawn trajectories; the average totals among each group were measured as well. Trajectories, however, occur in every game instance as they are the only action a player can take when playing *Space Navigator*. It is important to understand how players prioritize their actions and how those actions affect team strategies because they are the only effect a player has on the environment.

Total trajectory draws were measured by counting any time a player drew a trajectory for a ship, whether it was the initial trajectory for the ship or a redrawn trajectory.

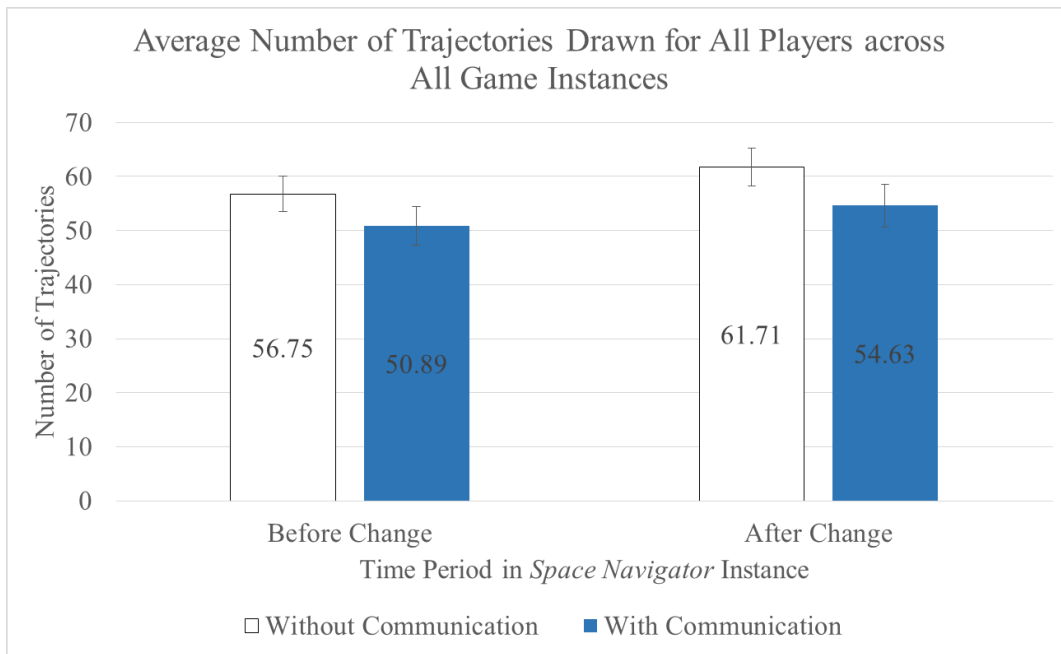


Figure 14. Average number of trajectories drawn for all players across all game instances.

Figure 14 shows that on average, the participants in the non-communication group draw more trajectories than the participants in the communication group. Participants in the communication group were able to establish their strategies and have better understanding of what their partner was doing, and therefore did not need to adjust as much as the participants in the non-communication group. Also, the number of trajectories drawn increased in both groups after the change occurred in the environment. As change happens, it takes time to adapt to the new rules which would force both groups to adjust from their established way of playing to a different way to play.

The proportion change average for drawn trajectories is comparably low (5.14%) to the average amounts of change in proportions between players for the colored ships

(16.4%) and sides of the environment (16.6%). Figure 15 shows that the average proportion change of drawn trajectories after a change occurs is 5.14% between both groups, meaning players are not likely to change more than about 5% of how many trajectories they are drawing in a given game instance in relation to their partner.

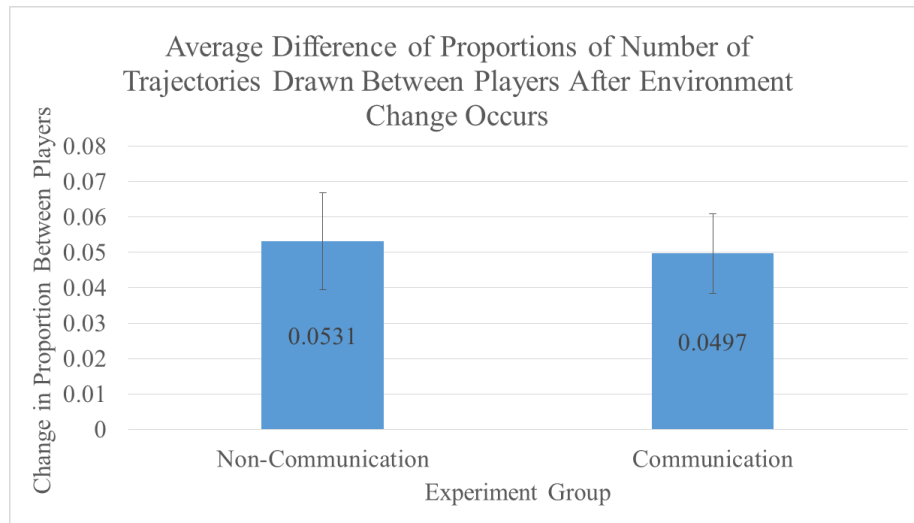


Figure 15. Average proportion change of drawn trajectories after environment change occurs.

Total trajectories drawn can be broken down into two further categories, initial trajectories and redrawn trajectories. Initial trajectories are defined as trajectories drawn on ships that have never been acted upon by either player. Redrawn trajectories are defined as trajectories drawn on ships that have been drawn on previously by either partner. Initial trajectories were similar across both experiment groups and in both time periods of the game (Figure 16).

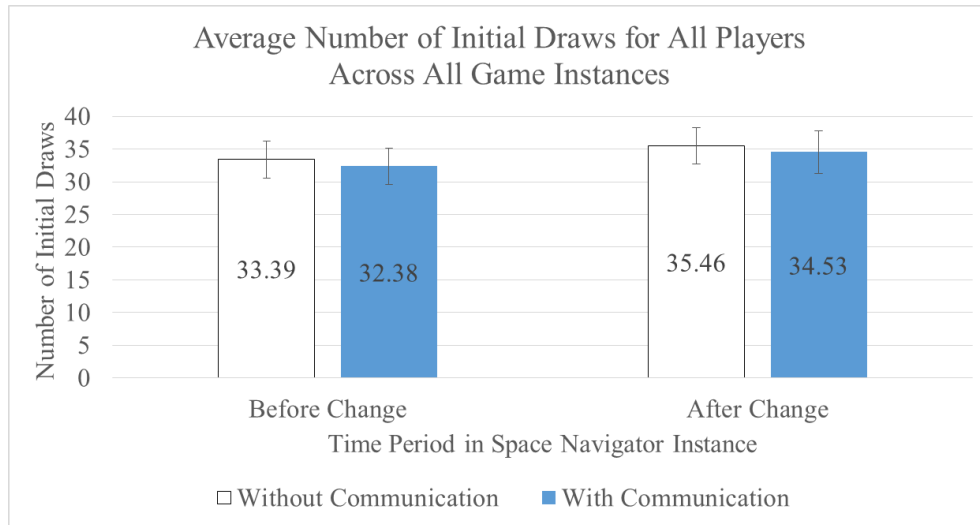


Figure 16. Average number of initial trajectories for all players across all game instances.

Ships spawn at a constant rate of one ship every two seconds. Unless the ship crashes or flies off the screen without being acted upon, there will be an initial trajectory. Players often will draw ship trajectories as soon as they can to start the ship on a more optimal path early, regardless of group or what is happening in the environment, which is why the initial trajectories are similar for both experiment groups and time periods in the environment.

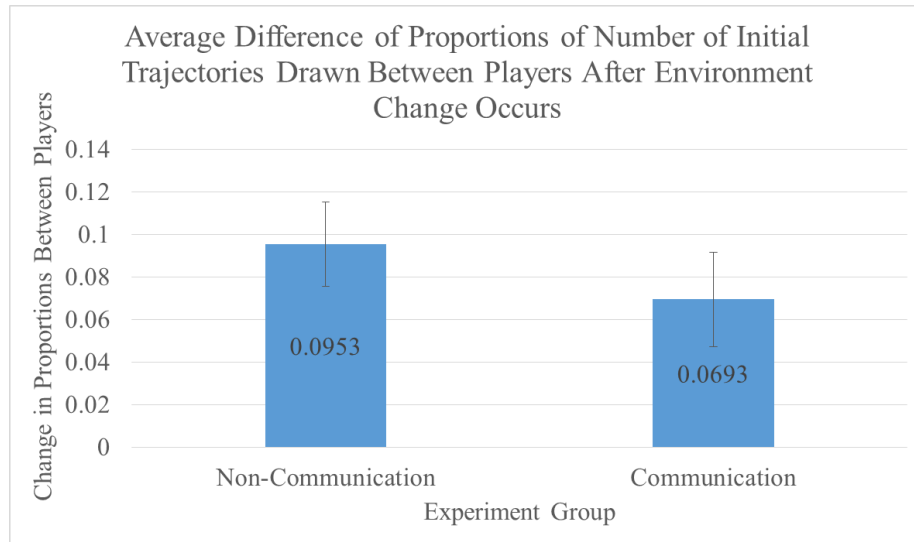


Figure 17. Average proportion change of redrawn trajectories after environment change occurs.

The players who could communicate established more even workloads by favoring division by sides and were drawing similar numbers of initial trajectories whereas the teams who could not communicate moved towards their *task division* strategies, where one player was becoming the initial trajectory drawer and the other was focused on redrawing trajectories to avoid collisions. Using a larger sample size than in this experiment would likely see this gap between experiment groups increase and potentially cause the confidence intervals to no longer overlap, making the change in initial draws between teammates a significant difference (Figure 17).

The other type of trajectory to examine was the redrawn trajectories. On average, the total number of redrawn trajectories increased after a change occurred in the environment (Figure 18). Again, this was due to teams adjusting to the new change in the environment and modifying their old, comfortable strategy slightly to accommodate for the change. However, there is a significant difference between experiment groups within each time period of the game instance.

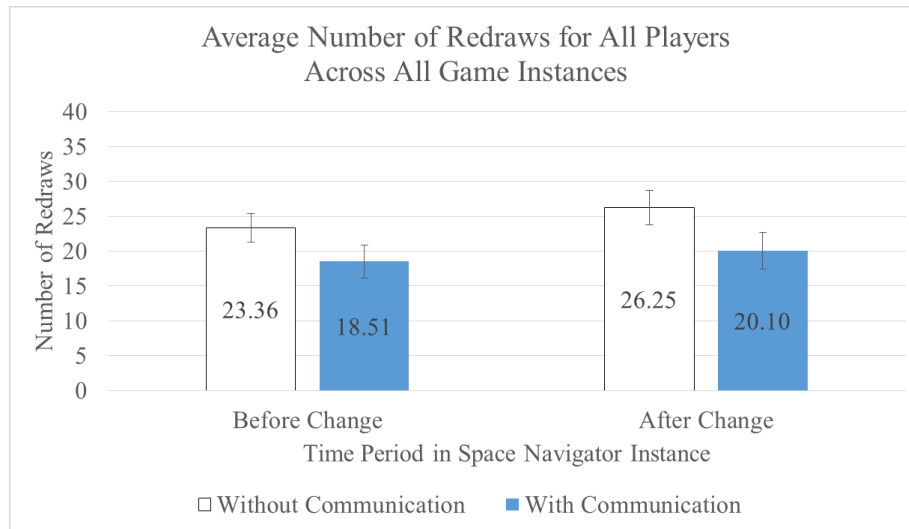


Figure 18. Average number of redraws for all players across all game instances.

Players who could communicate had established their areas of focus before the game instances had started. Ships with trajectories outside of the player’s focus caused the player to ignore those ships, even if they were on a collision course, because there was some trust that their partner would take care of their own ships. In the non-communication group, players played as if they were playing alone and looked for all collisions throughout the screen or had established the *task division* strategy, which caused those teams to redraw more in the effort to prevent more collisions.

Because the teams in the non-communication group were establishing which partners draw initial trajectories and which avoids collisions before the change occurred in the environment, there was a 7.27% proportion change between teammates in that group. The communication group, on the other hand, had an average proportion change of 12.54%, with a 95% confidence interval overlap of less than 0.001. Given the small sample size for this experiment, if more tests were to be performed under the same

conditions, the confidence intervals would likely no longer overlap to give a statistically significant difference.

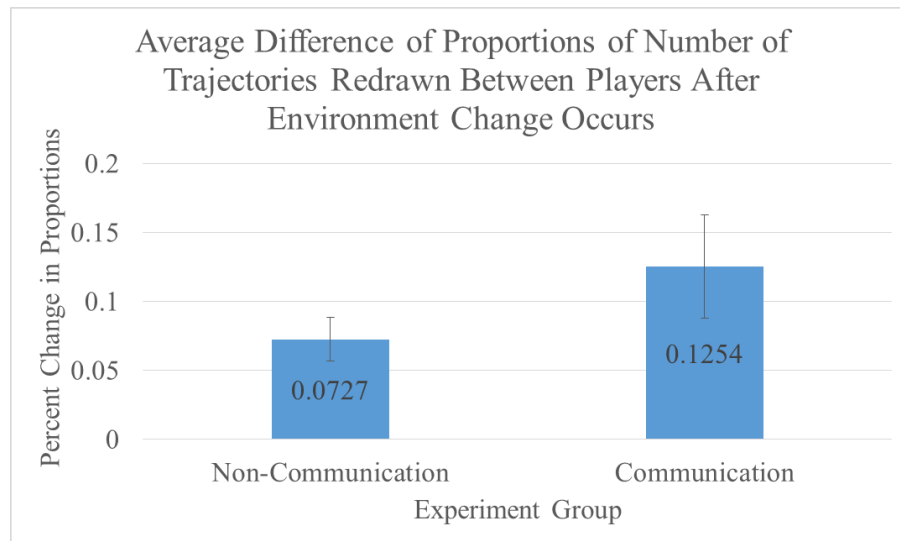


Figure 19. Average proportion change of redraws between players after environment change occurs.

Because the confidence interval overlap is small, it was important to understand how the proportions were changing. Figure 19 shows that the participants in each experiment group are changing the proportion of redraws they are performing, but the question becomes are those proportions closing the gap between the teammates or increasing the gap? For example, seeing a proportion change of 10% could mean that in one situation players are dividing redraws where Player A performs 60% of the redraws and Player B performs 40% before the change occurs. After the change Player A then performs 50% of the redraws and Player B performs 50%. In another situation, with the same proportion change of 10% and Player A performing 60% and Player B performing 40%, the proportions could move in the other direction and after the change occurs Player A could perform 70% of the redraws and Player B could perform 30%. In both situation there is a

proportion difference of 10% for both players. However, the gap between the players' proportions is 0% in the first situation and 40% in the second situation.

This gap is defined as the Proportion Gap. The proportion gap is important to understanding if players are converging (proportion gap becomes smaller) or diverging (proportion gap becomes bigger) on the amount of workload for a specific category. With only two players working together as a team in this experiment, the proportion gap is centered at 50%, meaning that dividing the proportion gap in half will give the proportion each player is above and below the 50% workload mark.

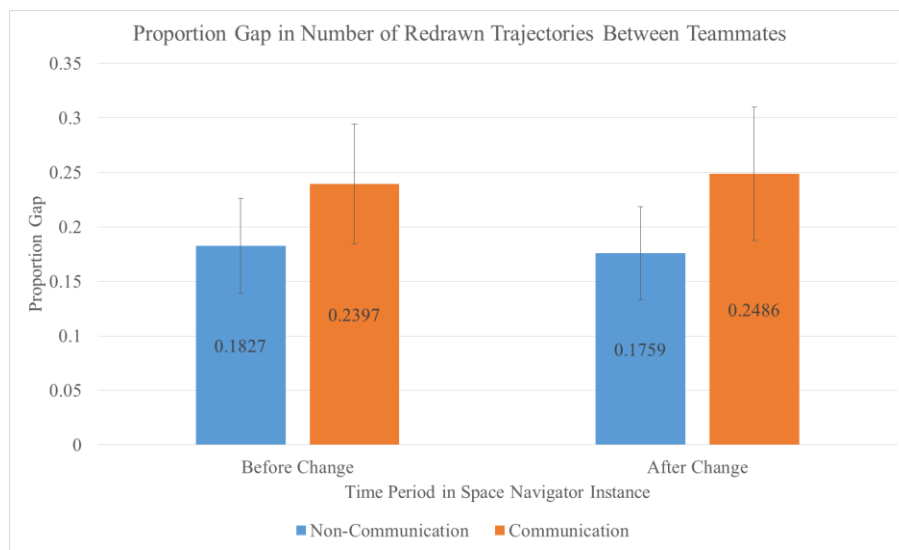


Figure 20. Proportion gaps for redrawn trajectories between teammates.

Figure 20 shows how the proportion gaps were defined before and after a change occurred in the environment. The communication group had a larger proportion gap than the non-communication group overall, meaning one player is doing more of the majority of redraws for the team. However, as the environment changes, the proportion gap for redraws were on average only 0.79% different from the proportion gap before the change occurred. The only cause for the proportions to change (shown in Figure 19) without the

proportion gap growing or shrinking (Figure 20) is if the players switch who is drawing more than half of the redraws in the game.

With all the configurations of participants for the experiment, there were 24 total games that had a change in the environment. In 10 of those 24 games (41.7%) there was no shift in which player was drawing a majority of the redraws. And in those 10 games, 7 (70%) occurred within the non-communication group, meaning that if a teammate in the non-communication group was doing more than half of the redraws, they would likely continue doing so after a change occurred in the environment. In the other 14 games, players switched roles in who was drawing the majority of redraws. This supports how the non-communication group converged to the *task division* strategy as the partners would likely continue in their roles to draw either the minority or majority of redraws, whereas the communication group didn't stay constant in who controlled the majority of redraws because those partners in those teams had to focus on their own areas and redraw to avoid their collisions in those areas.

4.2.4 Conclusion

Analyzing the division of workload on colored ships showed that teams in both experiment groups did not change their strategies to favor one colored ship over another. Similarly, the division of workloads on the sides showed that the non-communication group did not favor any specific sides after a change occurred. The communication group, because of their preference to the *sides* strategy, had already established workload proportion for the right and left sides so there was less change in that proportion after a change occurred because those teams would continue using the *sides* strategy. The non-communication group preferred to continue using the *task division* strategy through

changes in the environment because they established one partner to perform more of the redraws than the other partner.

Overall, teams reported to not changing their strategies unless specifically forced to by the environment. Out of the 72 total game instances between all participants, there were 19 (26.4%) game instances where a team reported changing their strategy through a questionnaire. Only 1 out of the 19 reported strategy changes occurred in a game instance where the environment did not change, meaning that teams were not changing their original strategies unless forced to by some change in the environment.

4.3 Overall Performance Comparison between Groups

The final objective measurement analyzed for this experiment was overall performance in the *Space Navigator* environment. The four ways to gain or lose points in *Space Navigator* are by ships reaching their destination, picking up bonuses, flying through no fly zones, and ships colliding into each other. Overall, participants did not develop strategies around avoiding no fly zones and some teams that could communicate mentioned to their partners to ignore the no fly zones and to not worry about flying through them. Because of the low priority of no fly zones, the time spent in them was not compared among the experiment groups. Instead the following four categories of performance were examined for each game instance: total team scores, total team collisions, total team bonus pick-ups, and total number of ships reaching their corresponding planets for each team.

The expected result on overall performance was that the teams that could communicate would perform better overall in each of the four categories of performance

than the teams who could not communicate. Performing better than another team is defined respectively for each category as having a higher score, fewer collisions, more bonus pick-ups, and more ships reaching their destinations. Each game instance must be examined individually and compared to the same game instance for all participants because different aspects of game instances affected performance results in different ways. For example, game instances 6 and 10 had higher average scores because bonuses were worth more points and collisions were worth no points, respectively. The results of the experiment were different than the expected results, as the teams who could not communicate performed better in all four categories of performance.

Beginning with scores, the overall team scores for the non-communication group were higher on average than the scores for the communication group (Figure 21).

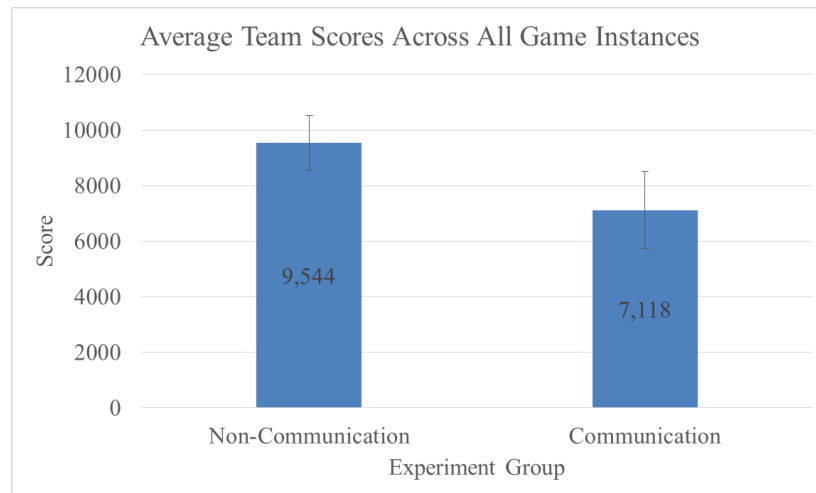


Figure 21. Overall average team scores for experiment groups.

The teams in the non-communication group also scored higher on average in all game instances than the teams in the communication group (Figure 22).

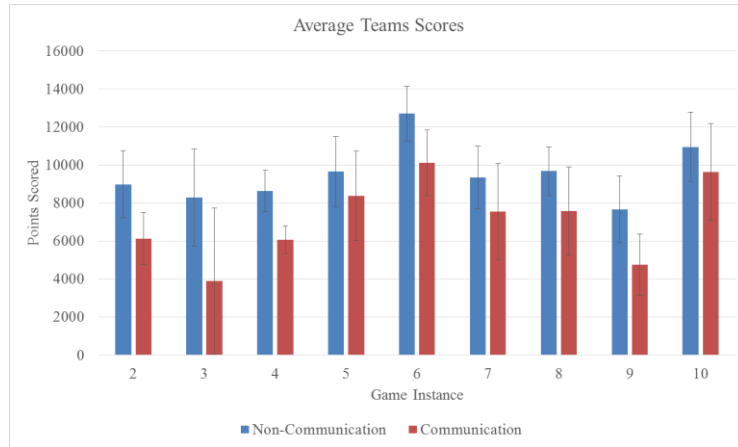


Figure 22. Average team scores across all game instances.

An important factor for the disparity in scores between the groups was the difference in the number of ships reaching their destinations for each team (Figure 23).

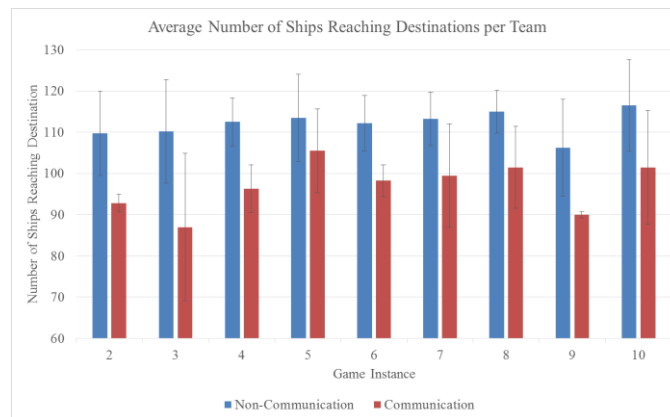


Figure 23. Average number of ships reaching destinations per team.

The average difference between the non-communication and communication group was 15.2 ships per game instance. In all game instances, a ship reaching its destination planet was worth 100 points, and the point value was never modified by the rule changes in the environment. Having 15.2 more ships reach their destination increases the team's scores by 1520 points. The average difference in total scores for each game was 2425.8 points. The teams in the non-communication group were able to gain a better total score because

the ships reaching their destinations accounted for 62.7% of the difference in score on average. In *Space Navigator*, the key for ships reaching their destinations is to avoid collisions. The only ways a ship can leave the environment are if it collides with another ship, flies off the screen (either by a user drawing the ship off the screen or by the ship flying off the screen without interference) or reaches its destination. If there are fewer collisions in a game, more ships will be available to reach their destinations.

The non-communication group also had ships collide less in their games than the communication group. On average, the communication group had 4.4 more collisions than the non-communication group (see Figure 24).

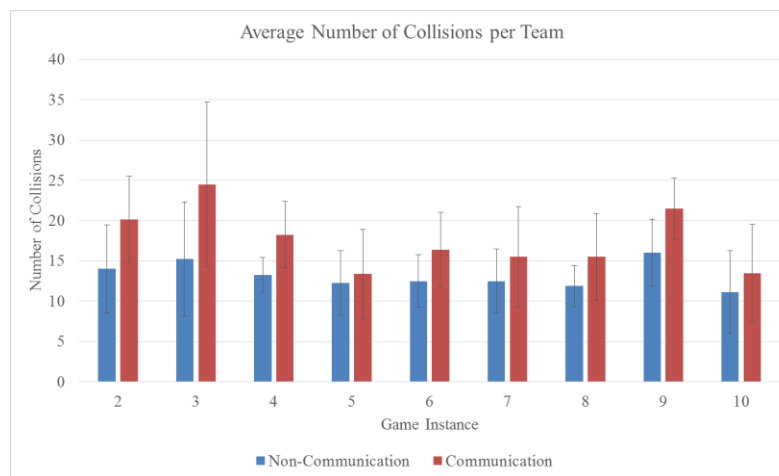


Figure 24. Average number of collisions per team across all game instances.

A collision typically involves two ships colliding (although there are rare instances when three or more ships can collide depending on the timing of the computer processing the code, for the purpose of this experiment the assumption is that a collision involves only two ships) which results in 200 points lost for a collision. This difference in collisions caused an average drop of 880 points for the communication group, which accounts for another 36.3% of the difference in scores between the groups. Combining

the ships reaching their destinations and the number of collisions accounts for 99% of the average total score difference between each experiment group.

The average number of bonus pick-ups, the final performance category, for both groups was 27.2 bonuses out of thirty (without counting Game Instance 4 – where only blue ships could pick up bonuses). Although the difference between groups was small, the non-communication group still had the advantage in the average number of bonus pick-ups per team (Figure 25).

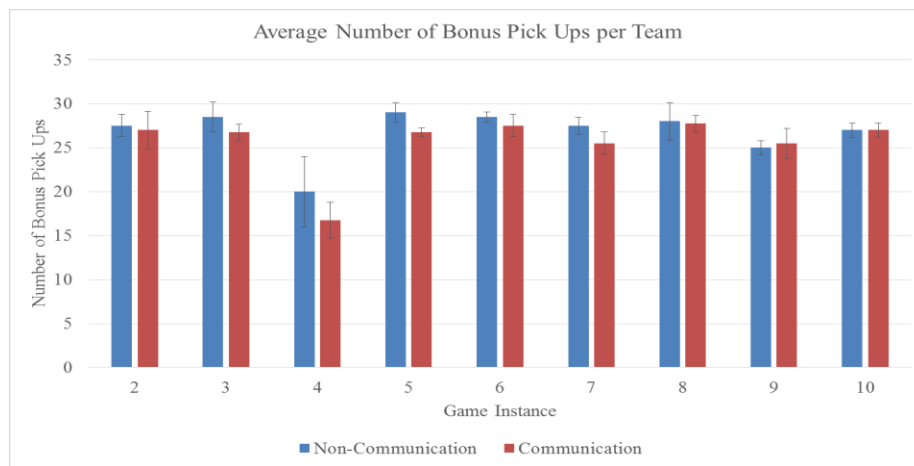


Figure 25. Average number of bonus pick-ups per team.

The non-communication teams' performances were overall better than the communication teams' because they avoided more collisions and were then able to get ships to their destination planets. For all teams in the experiment, collisions, number of ships reaching their destinations, and scores were strongly correlated and statistically significant as shown in Table 6. Redraws correlated with collisions, ships reaching destinations and scores, but could not be determined statistically significant. These trends, however, could become significant given the opportunity to perform the experiment with a larger sample size.

Table 6. Correlations between redraws, scores, destinations, and collisions.

Correlation	Correlation Coefficient	P-Value
Redraw x Collision	-0.1624	0.1729
Redraw x Destination	0.1855	0.1187
Redraw x Score	0.1689	0.1562
Collision x Destination	-0.8591	< 0.001
Collision x Score	-0.8831	< 0.001
Destination x Score	0.8500	< 0.001

4.3.1 Conclusion

The teams in the non-communication group out-performed the teams in the communication group in every scoring category that was measured. The lack of communication caused those teams to merge towards the *task division* strategy which showed trends of being the overall optimal strategy for two human partners in *Space Navigator*. The teams in the communication group instead wanted to divide the work as evenly as possible, and physically dividing the rectangle environment in half was the simplest solution for those teams. This strategy ultimately backfired on the teams that could communicate because the participants were focusing on moving ships out of their designated areas instead of ensuring the ships didn't collide. Having fewer ships collide meant having more ships available to reach their destination planets, which ultimately increased the average scores for all teams in the experiment.

4.4 Summary

This chapter discussed the results from the experiment outlined in the Methodology chapter. Three conclusions came from analyzing the results which should

be used as a focus in future experiments in studies that involve applying human-human teams to human-machine teams. The first conclusion was that each of the experiment groups converged on a strategy, but on strategies of different types – the communication group converged on *sides* and the non-communication group converged on *task division*. Second, all teams in the experiment did not change their strategies through each game instance unless they were forced to by some change in the environment. Finally, the non-communication group of teams performed better than the communication group because they used a more optimal strategy for scoring the most points in *Space Navigator*, whereas the communication group of teams were more focused on evenly balancing the workload based on the physical space of the environment.

V. Conclusion

Chapter Overview

The purpose of this chapter is to summarize the methodology to develop and create the experiment to answer the thesis question, and the conclusions from the results of the experiment. This chapter will also discuss potential directions for future work that can expand on the trends and results of this experiment.

5.1 Summary of Thesis and Investigative Questions

The question this research aimed to answer was how do human teams adapt to a changing environment when they can communicate versus when they cannot communicate? Secondary questions that needed to be answered to solve this research question included: how are users dividing the workload, and how do the performances of the groups compare in the environment?

Answering these questions could aid in the development of machine-agent teammates that partner with a human to complete a task. Virtual environments that human-machine teams work in can change in an instant, and the machine-agent needs to be able to adapt to the changes to continue completing the task with the human. It is important that the human and machine communicate effectively when changes occur so the human can know what the machine agent understands about the changed environment and what parts of the workload the human will have to take over in the event of a change.

5.2 Summary of Results

The results from the experiment developed three conclusions based on the thesis question and investigative questions. First, all teams in the experiment, regardless if the

teams could communicate or not, did not change their strategies through each game instance unless they were forced to by some change in the environment. Second, each of the experiment groups converged on a strategy, but on strategies of different types – the communication group converged on *sides* and the non-communication group converged on *task division*. Third, the non-communication group of teams performed better than the communication group because they used the optimal *task division* strategy for scoring the most points in *Space Navigator*, whereas the communication group of teams were more focused on evenly balancing the workload based on the physical space of the environment by using the *sides* strategy.

5.3 Summary of Methodology to Answer Research Questions

The dynamic environment in *Space Navigator* simulated unforeseen situations that a machine-agent may have to face when partnered with a human. *Space Navigator* was also a single player game that paired a partner with a machine-agent partner and did not have the capabilities to have two humans play together as a team. This required research in developing a multiplayer game. After examining the two best options, client-server and peer-to-peer, the client-server architecture proved best for the design of this experiment due to it having a central location for all data to pass through in order to analyze the data passed between teammates and due to its simplicity for a small-scale game situation. Synchronizing the game environment between players was important because players needed to make quick decisions based on current information and lagging data would have negatively affected those decisions. Therefore, the multiplayer game used a mirrored architecture where each player's tablet mirrored what the server

told them was the game environment, and players sent simple commands across the network to decongest network traffic.

After the development of the multi-agent environment, the dynamic changes were applied to force the teams to adapt their strategies. Preliminary results in a previous experiment showed the different ways humans would potentially divide the workload of the environment, which were categorized into three main strategies: *sides* (each player takes a side), *colors* (each player takes two of the colored ships), and *task division* (one player draws initial trajectories and the other player avoids collisions). Rule changes were applied to the environment predicting that the human teams in the sample for the experiment would follow these trends. The rule changes were specifically made to confound these main strategies so that no matter what strategy the team took, they would eventually have to adapt at some point in the experiment. The partners rotated into different teams so that each team of participants would need to adapt to a change that neither partner had seen before.

5.4 Future Work

This experiment ran with a small number of participants that showed some statistically significant data, but some data also showed only trends that may lead to further conclusions. The sample size was relatively small. With more studies using the same experiment but with a larger sample size, the trends could start to separate themselves to become statistically significant. This can be used to better examine the differences in adaptation between the two experiment groups.

Throughout the course of running the experiment and after looking back at the design after gathering results, there were some weaknesses and limitations to the experiment. First, it was difficult to schedule groups of four participants to run the experiment at the same time and commit to the hour and a half it took to run the experiment. Scheduling multiple participants simultaneously makes increasing the sample size more difficult with a limited amount of time. With a larger time window to find more participants, the sample size could easily double to create better data to analyze in the results section. Second, there was a shorter time window to run the functional experiment with participants because creating and debugging the multi-agent game alone took much longer than anticipated. There were hardware issues with the tablets (overheating) and the router (dropping packets) that caused significant delays. With the system fully functioning now, there should only be a need for software changes for future experiments that utilize the multi-agent game. Finally, this study was very observational because a multi-agent study dealing with communication in this environment had never been done before at AFIT. Before participants ran through the experiment, we were not sure what kind of results to expect, which made narrowing down the scope of analysis difficult and was approached with a trial-and-error tactic. Now that there are some insights on what players are doing within the game, and how they are performing, there are some possible directions future experiments can focus towards and better design experiments that expand on the observations shown in the Results chapter.

To enhance the analysis of adaptation and change, change point detection algorithms [29] could be used to determine specific instances of change in player behavior through all game instances using the different parameters that players can act

upon in the game environment (colored ships, number of trajectories, etc.). Another data point to examine with a further sample size was the overall performance between groups. The non-communication group could have been overall better at video games which gave them an edge in performing well in the game environment. This could have also led them to the *task division* strategy. A larger sample size could diversify the types of participants which could either solidify the current results on performance or create a different conclusion.

The purpose of designing the experiment to run with human teams was to apply the human interactions and behaviors to a machine-agent that can work with a human in the same manner as a human in a similar, unpredictable environment. Using the *Space Navigator* environment as an example, humans could decide what strategy they are best at, and give a command to the machine-agent to coincide with that strategy. If the human wanted to use the *sides* strategy, for example, they could tell the machine-agent to control a specific side while the human controls the other side. Also, if the human preferred drawing initial trajectories, the machine-agent could handle only the collisions so that the partners are not overlapping work with each other. As the environment changes, preliminary studies could focus on how the human would tell the machine-agent to adapt to certain types of change for that environment, and as the machine-agent's artificial intelligence grows, the machine-agent could then begin to detect certain queues from the environment that tell it to shift to a different strategy without the need for any human interaction. The development of the machine-agent is specific to the *Space Navigator* environment, so to create a teammate that responds well in a general environment, more studies should be performed in different games such as *Portal 2* to understand the human-

human and human-machine team interactions in general as opposed to in a specific environment. The more general of an environment these machine-agents can adapt to, the more applicable they will become when needed in real-world scenarios that the United States Air Force operates in every day.

Appendix A - *Space Navigator* Interaction Study Pre-Questionnaire

1. Participant Number (Assigned By Researcher):
2. Name:
3. Email Address:
4. Age:
5. Handedness: Left Right Ambidextrous
6. Gender: Male Female
7. Number of previous *Space Navigator* studies you have participated in:
8. How much experience do you have with the following:
 - a. In a given year, on how many days do you interact with the following types of devices?

	Never Daily	<1 per month	1-3 per month	1-2 per week	3-6 per week	
Laptop computer	0	1	2	3	4	5
Tablet computer	0	1	2	3	4	5
Smart phone	0	1	2	3	4	5
Desktop computer	0	1	2	3	4	5
Gaming consoles	0	1	2	3	4	5

- a. In a given year, on how many days do you play video games of the following type?

	Never Daily	<1 per month	1-3 per month	1-2 per week	3-6 per week	
Any Kind	0	1	2	3	4	5
Simulation (SimCity)	0	1	2	3	4	5
Battle-Arena (LoL, Overwatch)	0	1	2	3	4	5
Role-Playing (WoW)	0	1	2	3	4	5
Action (Mario, Donky Kong)	0	1	2	3	4	5
First Person Shooter (Halo)	0	1	2	3	4	5
Strategy (Civilization)	0	1	2	3	4	5
Puzzle (Tetris)	0	1	2	3	4	5
Casual (Angry Birds)	0	1	2	3	4	5
Music (Guitar Hero)	0	1	2	3	4	5
Sports (Madden Football)	0	1	2	3	4	5
Board (Ticket to Ride)	0	1	2	3	4	5
Card (Poker, Pinochle)	0	1	2	3	4	5

Appendix B – *Space Navigator* Interaction Study Post-Pairing-Questionnaire

1. Participant Number:

2. How would you rate your partner on **cooperation** (0 being not cooperative and 10 being very cooperative):

	Not Cooperative									Very
Cooperative										
	0	1	2	3	4	5	6	7	8	9
	10									

3. How would you rate your partner on **communication** (0 being they did not communicate at all and 10 being they communicated constantly):

	Not Communicative									Very
Communicative										
	0	1	2	3	4	5	6	7	8	9
	10									

4. How would you rate your partner on the following aspects of **communication** (0 extremely poor and 10 extremely good):

a. Quantity (they gave as much information as needed and no more)

	Very Poor									Very
Good										
	0	1	2	3	4	5	6	7	8	9
	10									

b. Quality (they communicated information in a truthful manner)

	Very Poor									Very
Good										
	0	1	2	3	4	5	6	7	8	9
	10									

c. Relation (their communications were relevant)

	Very Poor									Very
Good										
	0	1	2	3	4	5	6	7	8	9
	10									

d. Manner (they communicated clearly, avoiding obscurity and ambiguity)

	Very Poor									Very
Good										

0 1 2 3 4 5 6 7 8 9
10

5. How would you rate your partner on **strategy development** (0 being extremely poor and 10 being extremely good):

Very Poor Very
Good
0 1 2 3 4 5 6 7 8 9
10

6. What was your initial strategy for the **FIRST** game instance?
7. Did your strategy change during the **FIRST** game instance? Why or why not? If your strategy did change, when did you make the change?
8. What was your initial strategy for the **SECOND** game instance?
9. Did your strategy change during the **SECOND** game instance? Why or why not? If your strategy did change, when did you make the change?
10. What was your initial strategy for the **THIRD** game instance?
11. Did your strategy change during the **THIRD** game instance? Why or why not? If your strategy did change, when did you make the change?

Appendix C – *Space Navigator* Interaction Study Post-Questionnaire

1. Participant Number:

2. What was the most challenging aspect of playing *Space Navigator* well?

3. Rank each partner on **cooperation** (1 being the most cooperative and 3 being the least):

First Partner ____ Second Partner ____ Third Partner ____

4. Rank each partner on **communication** (1 being they communicated with you the most and 3 being the least):

First Partner ____ Second Partner ____ Third Partner ____

5. Rank each partner on the following aspects of **communication** (1 being the best and 3 being the worst):
 - a. Quantity (they gave as much information as needed and no more)
First Partner ____ Second Partner ____ Third Partner ____

 - b. Quality (they communicated information in a truthful manner)
First Partner ____ Second Partner ____ Third Partner ____

 - c. Relation (their communications were relevant)
First Partner ____ Second Partner ____ Third Partner ____

 - d. Manner (they communicated clearly, avoiding obscurity and ambiguity)
First Partner ____ Second Partner ____ Third Partner ____

6. Rank each partner on **strategy development** (1 meaning they developed the best strategies and 3 being the worst):

First Partner ____ Second Partner ____ Third Partner ____

7. What forms of communication did you use with your partner? And what other forms of communication would you have liked to use with your partners (if any)?

8. Which strategy did you find most effective?

9. Which strategy did you find least effective?

10. (Only for those who **were** able to verbally communicate) How did communication with a partner affect you in developing strategies and performing well?

Very Hurtful					Neutral					Very Helpful
	0	1	2	3	4	5	6	7	8	9
10										

11. (Only for those who **were not** able to verbally communicate) How did lack of communication with a partner affect you in developing strategies and performing well?

12. What outside factors may have influenced your ability to perform well?

On the following pages, there are phrases describing people's behaviors. Please use the rating scale below to describe how accurately each statement describes *you*. Describe yourself as you generally are now, not as you wish to be in the future. Describe yourself as you honestly see yourself, in relation to other people you know of the same sex as you are, and roughly your same age. So that you can describe yourself in an honest manner, your responses will be kept in absolute confidence. Please read each statement carefully, and then fill in the bubble that corresponds to your reply.

87

8. Make plans and stick to them.	☐	☐	☐	☐	☐
9. Dislike myself.	☐	☐	☐	☐	☐
	Very Inaccurate	Moderately Inaccurate	Neither Inaccurate nor Accurate	Moderately Accurate	Very Accurate
10. Respect others.	☐	☐	☐	☐	☐
11. Insult people.	☐	☐	☐	☐	☐
12. Seldom feel blue.	☐	☐	☐	☐	☐
13. Don't like to draw attention to myself.	☐	☐	☐	☐	☐
14. Carry out my plans.	☐	☐	☐	☐	☐
15. Am not interested in abstract ideas.	☐	☐	☐	☐	☐
16. Make friends easily.	☐	☐	☐	☐	☐
17. Tend to vote for liberal political candidates.	☐	☐	☐	☐	☐
18. Know how to captivate people.	☐	☐	☐	☐	☐
19. Believe that others have good intentions.	☐	☐	☐	☐	☐
	Very Inaccurate	Moderately Inaccurate	Neither Inaccurate nor Accurate	Moderately Accurate	Very Accurate
20. Do just enough work to get by.	☐	☐	☐	☐	☐
21. Find it difficult to get down to work.	☐	☐	☐	☐	☐
22. Panic easily.	☐	☐	☐	☐	☐
23. Avoid philosophical discussions.	☐	☐	☐	☐	☐
24. Accept people as they are.	☐	☐	☐	☐	☐
25. Do not enjoy going to art museums.	☐	☐	☐	☐	☐
26. Pay attention to details.	☐	☐	☐	☐	☐
27. Keep in the background.	☐	☐	☐	☐	☐

28. Feel comfortable with myself.	☐	☐	☐	☐	☐
29. Waste my time.	☐	☐	☐	☐	☐
	Very Inaccurate	Moderately Inaccurate	Neither Inaccurate nor Accurate	Moderately Accurate	Very Accurate
30. Get back at others.	☐	☐	☐	☐	☐
31. Get chores done right away.	☐	☐	☐	☐	☐
32. Don't talk a lot.	☐	☐	☐	☐	☐
33. Am often down in the dumps.	☐	☐	☐	☐	☐
34. Shirk my duties.	☐	☐	☐	☐	☐
35. Do not like art.	☐	☐	☐	☐	☐
36. Often feel blue.	☐	☐	☐	☐	☐
37. Cut others to pieces.	☐	☐	☐	☐	☐
38. Have a good word for everyone.	☐	☐	☐	☐	☐
39. Don't see things through.	☐	☐	☐	☐	☐
40. Feel comfortable around people.	☐	☐	☐	☐	☐
41. Have little to say.	☐	☐	☐	☐	☐

Works Cited

- [1] A. F. Norcio and J. Stanley, "Adaptive Human-Computer Interfaces: A Literature Survey and Perspective," *IEEE Trans. Syst. Man Cybern.*, vol. 19, no. 2, pp. 399–408, 1989.
- [2] M. R. Endsley, "Autonomous Horizons: System Autonomy in the Air Force - A Path to the Future," vol. 1: Human-A, p. 27, 2015.
- [3] I. V. Karpov, L. M. Johnson, and R. Miikkulainen, "Evaluating team behaviors constructed with human-guided machine learning," *2015 IEEE Conf. Comput. Intell. Games, CIG 2015 - Proc.*, pp. 292–298, 2015.
- [4] B. Gertz, "F-35 software problems," *Washington Times*, Mar-2016.
- [5] A. M. Kessler, "Elon Musk Says Self-Driving Tesla Cars Will Be in the U.S. by Summer," *New York Times*, New York, New York, USA, Mar-2015.
- [6] D. J. Cook *et al.*, "MavHome : An Agent-Based Smart Home," pp. 1–4.
- [7] D. C. Penn, K. J. Holyoak, and D. J. Povinelli, "Darwin ' s mistake : Explaining the discontinuity between human and nonhuman minds," pp. 109–178, 2008.
- [8] J. M. Bindewald, G. L. Peterson, and M. Miller, "Clustering-Based Real-Time Player Modeling," pp. 1–12.
- [9] S. Ontanon, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, "A survey of real-time strategy game AI research and competition in starcraft," *IEEE Trans. Comput. Intell. AI Games*, vol. 5, no. 4, pp. 293–311, 2013.
- [10] Y. W. Bernier, "Latency Compensating Methods in Client / Server In-game Protocol Design and Optimization," *Proc. 15th Games Dev. Conf.*, vol. 98033, no. 425, pp. 1–10, 2001.

- [11] GigaTribe, “What is peer-to-peer (p2p)?” [Online]. Available:
www.gigatribe.com/en/help-p2p-intro.
- [12] R. Schollmeier, C. Networks, and T. Universität, “A Definition of Peer-to-Peer Networking for the Classification of Peer-to- Peer Architectures and Applications,” pp. 2–3, 2002.
- [13] GigaTribe, “What is peer-to-peer (p2p)?” .
- [14] M. Claypool and K. Claypool, “Latency and player actions in online games,” *Commun. ACM*, vol. 49, no. 11, p. 40, 2006.
- [15] E. Cronin, A. R. Kurc, B. Filstrup, and S. Jamin, “An efficient synchronization mechanism for mirrored game architectures,” *Multimed. Tools Appl.*, vol. 23, pp. 7–30, 2004.
- [16] L. Gautier and C. Diot, “Design and evaluation of MiMaze a multi-player game on the Internet,” *Proceedings. IEEE Int. Conf. Multimed. Comput. Syst. (Cat. No.98TB100241)*, pp. 1–15, 1998.
- [17] S. Ferretti, “A synchronization protocol for supporting peer-to-peer multiplayer online games in overlay networks,” *Proc. Second Int. Conf. ...*, vol. M, no. 7, pp. 83–94, 2008.
- [18] J. M. Bindewald, M. E. Miller, and G. L. Peterson, “A function-to-task process model for adaptive automation system design,” *Int. J. Hum. Comput. Stud.*, vol. 72, no. 12, pp. 822–834, 2014.
- [19] L. Bottou, “From Machine Learning to Machine Reasoning,” *Arxiv Prepr. arXiv11021808*, p. 15, 2011.
- [20] J.-M. Hoc, “From human-machine interaction to human-machine cooperation,”

Ergonomics, vol. 43, no. 7, pp. 833–843, 2000.

- [21] D. Vaddi, Z. O. Toups, I. Dolgov, R. R. Wehbe, and L. E. Nacke, “Investigating the Impact of Cooperative Communication Mechanics on Player Performance in Portal 2,” in *Graphics Interface Conference 2016*, 2016, pp. 41–48.
- [22] Z. O. Toups, J. Hammer, W. a. Hamilton, A. Jarrah, W. Graves, and O. Garretson, “A Framework for Cooperative Communication Game Mechanics from Grounded Theory,” *Proc. CHIPlay '14*, pp. 257–266, 2014.
- [23] D. Vaddi, R. R. Wehbe, Z. O. Toups, S. N. Stahlke, R. Koroluk, and L. E. Nacke, “Validating Test Chambers to Study Cooperative Communication Mechanics in Portal 2,” *Proc. CHI Play 2015*, pp. 721–726, 2015.
- [24] “League of Legends.” .
- [25] Z. O. Toups, W. A. Hamilton, C. Keyes-Garcia, S. Perez, and R. Stanton, “Collaborative Planning Gameplay from Disaster Response Practice,” *Proc. 2015 Annu. Symp. Comput. Interact. Play - CHI Play '15*, pp. 715–720, 2015.
- [26] “Photon Unity Networking.” [Online]. Available: <https://www.photonengine.com/en/PUN>. [Accessed: 22-Aug-2016].
- [27] L. Sterling and K. Taveter, *The Art of Agent-Oriented Modeling*. Cambridge, MA: The MIT Press, 2009.
- [28] “Comparing Two Population Proportions with Independent Samples,” *Pennsylvania State University*, 2017. [Online]. Available: <https://onlinecourses.science.psu.edu/stat500/node/55>. [Accessed: 04-Feb-2017].
- [29] “Change Point Detection in Time Series with R and Tableau,” *R-Bloggers*. [Online]. Available: <https://www.r-bloggers.com/change-point-detection-in-time->

series-with-r-and-tableau/. [Accessed: 12-Jan-2017].

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 23-03-2017		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) Sept 2015-March 2017	
4. TITLE AND SUBTITLE Analysis of How Communication Affects Human Teams in a Dynamic Game				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Spuller, Jake M, 2d Lt, USAF				5d. PROJECT NUMBER 17V276	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way Wright-Patterson AFB, OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENG-MS-17-M-071	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR Computational Cognition and Machine Intelligence (CCMI) Program James Lawton, Program Manager, DSN: 426-5999, email: james.lawton.1@us.af.mil Air Force Office of Scientific Research 875 N Randolph St, Ste. 325 Arlington, VA 22203				10. SPONSOR/MONITOR'S ACRONYM(S) AFOSR/RTA	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Distribution Statement A. Approved for Public Release; Distribution Unlimited					
13. SUPPLEMENTARY NOTES This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT This research presents a new multi-agent, dynamic design to an existing tablet-based game environment that allows for the study of multi-human team communication when adapting to a changing environment. Research was conducted on designs of multi-agent environments, as well as synchronization protocols, to develop the new environment. Different types of communication were also examined to universally define how players in the environment interacted with each other. To make the environment rules dynamic and force participant strategies to change, common strategies provided from participant answers in a previous study in the same environment were confounded. Participants worked in two-person teams to establish strategies and work together to perform optimally in the environment, and adapt to the unseen changes from the environment. Participants were divided into two groups – teams that could communicate and teams that could not communicate. This experiment was performed to examine how communication affects performance, strategy development, and strategy adaptation. The results of the research show how teams that can communicate compare with teams that cannot communicate in strategy selection, strategy adaptation, and overall performance in the environment. These results can be applied when creating machine-agent teammates that must work with a human to adapt to changes never seen before.					
15. SUBJECT TERMS Human-Machine Teaming, Adaptive Automation, Dynamic Environment, Multi-Agent Environment, Virtual Communication					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 105	19a. NAME OF RESPONSIBLE PERSON Maj Jason M. Bindewald,
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code) (937) 255-3636 x4614