Assessing Interfaces Supporting Situational Awareness in Multi-agent Command and Control Tasks

Donald Kalar[1,2] and Collin Green[2]

[1] Department of Psychology
San Jose State University, San Jose CA, USA
[2] Human Systems Integration Division
NASA Ames Research Center, Moffett Field, CA, USA
donald.j.kalar@nasa.gov, collin.b.green@nasa.gov

Abstract. Here, we describe our efforts to uncover design principles for multi-agent supervision, command, and control by using real-time strategy (RTS) video games as a source of data and an experimental platform. Previously, we have argued that RTS games are an appropriate analog for multi-agent command and control [3] and that publicly-available data from gaming tournaments can be mined and analyzed to investigate human performance in such tasks [5]. We outline additional results produced by mining public game data and describe our first foray into using RTS games as an experimental platform where game actions (e.g., clicks, commands) are logged and integrated with eye-tracking data (e.g., saccades, fixations) to provide a more complete picture of human performance and a means to assess user interfaces for multi-agent command and control. We discuss the potential for this method to inform UI design and analysis for these and other tasks.

Keywords: Situation Awareness, Automation, RTS, Gaze Tracking, User Interfaces.

1 Investigating Multi-agent Command and Control Using Real-Time Strategy Games

Robotic sensor and on-board computing capabilities are advancing at an accelerated pace, enabling an increasing number of scenarios where robots can be deployed as autonomous or semi-autonomous agents rather than needing to be controlled via closed-loop tele-operation. These transitions allow the human operator to take on the role of commander or supervisor over multiple agents and increases the overall task execution rate for a fixed number of human users. Effective supervisory control systems require user interfaces that are compatible with the task demands and limitations of human memory, perception, cognition, attention, and actions. Optimized systems will maximize task execution while minimizing errors, allowing the human to command and control a greater number of autonomous agents at an acceptable level of performance.

R. Shumaker (Ed.): VAMR/HCII 2013, Part I, LNCS 8021, pp. 277–284, 2013.

One approach to developing candidate interfaces to support these multi-agent command and control scenarios is to investigate other domains that have similar or overlapping task characteristics [3]. This allows for more informed decisions when building interfaces, and can help to identify complications or limitations before building and testing high-fidelity systems. Real-Time Strategy (RTS) video games are one analog domain that can inform many aspects of how a human operator commands and controls multiple semi-autonomous agents in a dynamic environment under constraints. RTS games, as they are commonly played, constitute relatively a complex multi-agent supervision and control problem: A player will command ten to 200 units at a time, including as many as 20 different types with different characteristics and unique capabilities. The units will be deployed in service of several overarching strategic tasks (e.g., resource collection, construction, defense, scouting, offense) with multiple complex subtasks composing or supporting each task. To enable players to successfully manage the game (and even enjoy doing so) game user interface (UI) designers have–through careful design, evaluation, user research, and iteration–built RTS game UIs that are effective in this context.

Can the designs and design principles that are effective in RTS games be applied to supervisory control of multiple robots? Some evidence suggests that they can: at least one robotic control interface has been built to mimic a conventional RTS game and shown to be effective for robotic command and control [4]. However, merely duplicating interface and interaction designs from RTS games is not an adequate basis for building interfaces for all multi-agent command and control tasks. RTS games are distinctly *dissimilar* to real robotic control tasks in that RTS games do not include complexities like uncertainty and latency, both of which can have important consequences for human-robotic interaction (e.g., see [2]). Yet, RTS games can provide some design inspiration for robotic command and control interfaces and they also constitute a good platform for investigating the how interfaces, interactions, and properties of human performance and cognition may shape multi-robot control.

Like many real-world multi-agent tasks, these games demand that their players maintain high levels of situational awareness (SA) throughout the entire course of the game. Players must maintain a basic economy by acquiring and managing raw materials, monitor unit construction, explore terrain to monitor their opponent and environment, all while staging offensive attacks and defending their own bases and resources. Despite the high cognitive demand of these games, devoted players develop high levels of expertise, not unlike professional athletes. Video games and video game players are also of interest due to findings suggesting that game play may boost cognitive function, though these findings are still an open matter for investigation [1].

In summary, there is reason to believe that RTS games may be useful to researchers in human-robotic interaction inasmuch as they provide a platform for evaluating interface design and for assess human performance in a context that is similar (though not identical) to supervisory control of multiple robotic agents. In the remainder of this paper we outline our progress in this direction.

We have used publicly-available game archives to obtain data that help to shape and refine hypotheses about supervisory control and situational awareness, and we have recently begun collecting enhanced game data (user game actions plus user eye movement data) in an effort to better understand multi-robot HRI.

2 Mining and Analysis of Public RTS Game Archives

RTS Games such as *Starcraft II* often generate 'replay' files containing a timestamped log of every player input for later playback. These replay files are routinely posted online for general viewing, and it is especially common for expert-level tournaments to make these match replay files available. This provides a large and rich dataset for meta-analysis. In previous work, we analyzed 220 *Starcraft II* matches from the Championship Bracket at the Major League Gaming Pro Circuit 2011 in Orlando [5]. We investigated how different factors impacted instantaneous changes in player Actions Per Minute (APM) whenever a new unit or set of units were selected by the player in the game (see Figure 1). APM was chosen because is the standard heuristic for quantifying the skill-level of a player, and has been consistently correlated with winning games [6]. Expert players are able to maintain a rate of hundreds of APM throughout an entire match.

While the data are very noisy, the huge datasets allowed us to extract several small but interesting signals. For example, euclidean distance between sequential unit selections was not predictive of changes in APM, but changes between spatial clusters (ROIs) of actions did show an increase in APM. Players were faster to act when selecting larger groups of units, but slower to act if those groups were heterogeneous (i.e., multiple unit types in a single selected group). Players were much faster in issuing commands to mobile units when compared to stationary units. As expected, the general trend was that winners had greater APM than losers, but interestingly that increase in APM was only significantly different from zero immediately prior to a unit selection, not after selecting a new unit.

One major limitation to this approach is that, while these findings provide small hints as to how the players are interacting with the game interface, these data are necessarily ambiguous. A player can select a unit in order to gain SA (selecting allows the user to inspect the unit's current status), or the selection may be part of a series of commands based on existing SA.

3 RTS Games as a Platform for HRI Experiments

In order to better understand how a player uses the interface to support game play, we are recruiting participants of varying skill level to play the game while having their game actions logged and their eye movements tracked and recorded. The user interface provides a number of controls to support various aspects of SA, as well as interfaces to support command execution. These controls are spatially distinct, so we can know how frequent and how long players of varying

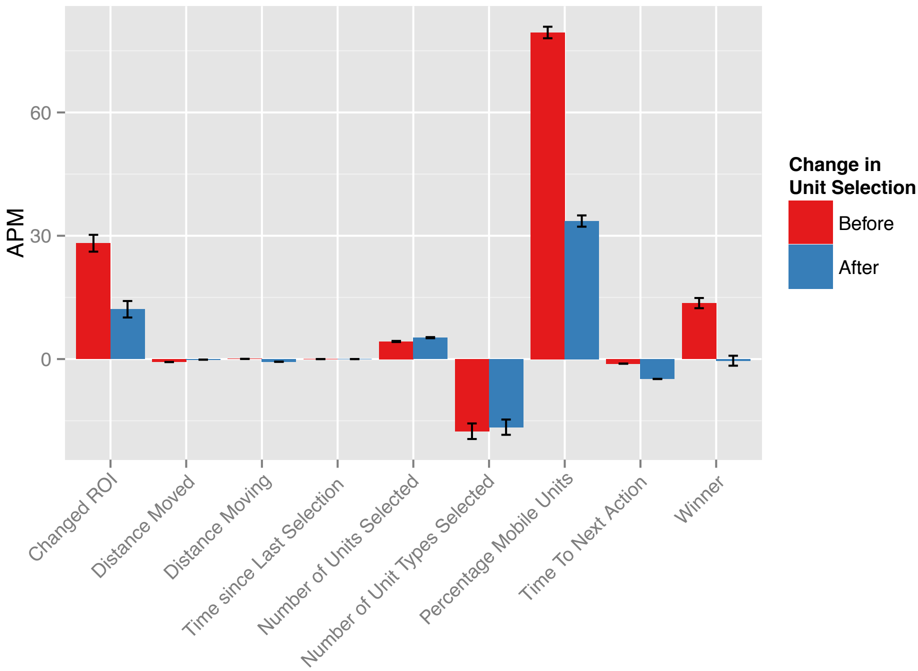

Fig. 1. Regression model output predicting changes in instantaneous Actions Per Minute (APM) as a result of different game transitions or states. Red bars indicate the estimated APM using a one-second window prior to unit selection; blue bars are the estimated APM using a one-second after selection window.

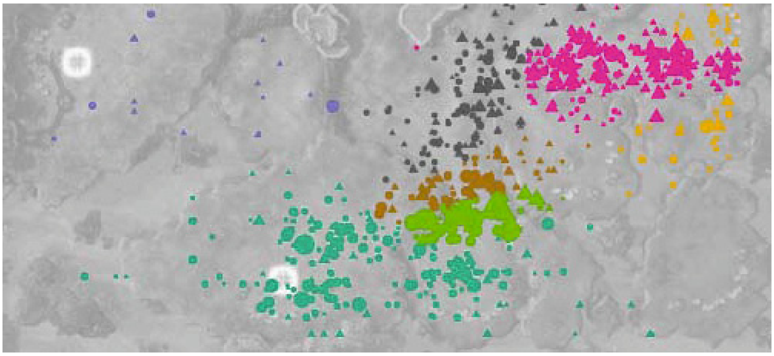

Fig. 2. An example of regions of interest (ROIs) generated from player actions during a game. Each color denotes a different ROI, as extracted by a standard expectation-maximization clustering algorithm. There were observed changes in player action rate when transitioning between ROIs, but no observed differences for action rate based solely on Euclidean distance (see Figure 1).

skill level scan away from the main view (the game map) to monitor unit status, game resources, regions that are outside of view from the main display, and command interfaces (see Figure 3). We predict qualitative differences between more expert and novice players in how frequently they sample these supporting controls and the total length of time they fixate on these controls rather than the main display.

Fig. 3. An annotated UI screenshot. The user is presented with a main overhead display of units in the environment (top). A mini map (bottom, left) provides a condensed view of the entire gameplay area, which is too large to view in the main display. A unit status window (bottom, center) provides information about currently selected units in the game, including details such as consumables status and task progress. The action menu (bottom right) provides a set of buttons to issue commands to the selected units, dynamically changing the set of buttons available based on what actions are available to issue the current selection.

It is worth noting that the addition of eye movement data collection to game logging may be especially useful for understanding how situational awareness (SA) is acquired and maintained. RTS game UIs (like many other UIs) conflate the acquisition of SA and the use of SA in issuing commands: the selection of a unit in an RTS game might be intended to reveal that unit's identity and status (acquire/update SA), to enable the issuance of a new command to that unit (apply SA), or both. Game log data only reveal that a unit was selected and what command (if any) was issued to that unit. However, eye movement data can reveal whether unit selection is followed by sampling of unit status information, examination of command options, or other eye movements associated with SA acquisition or SA use.

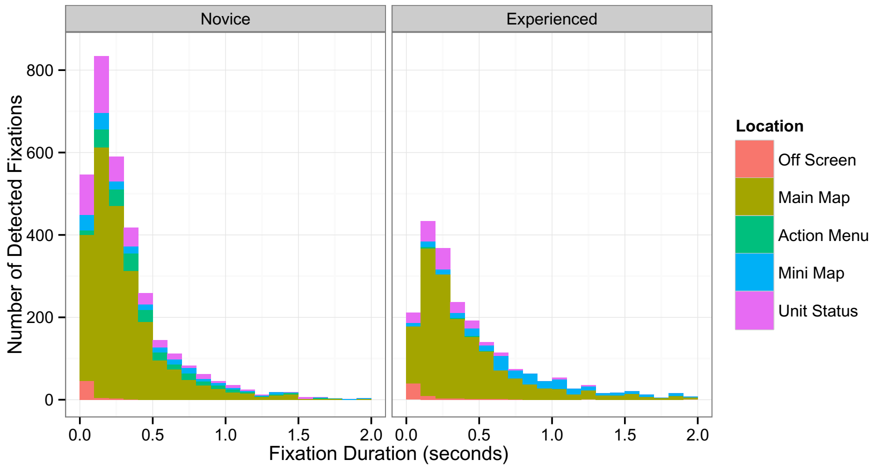

Fig. 4. Measured fixations for a single game. Gaze fixation locations and durations are plotted in this stacked histogram for both a novice and an experienced player for a single game. While the total observed fixation time during the games are almost identical (both just over seventeen minutes of detected gaze), the experienced player tends to fixate for longer durations while the novice player scans the UI at a much faster rate. The experienced player also effectively ignores elements of the UI (i.e., the Action Menu), while the novice is still reliant on the information in that display.

4 Future Experiments with RTS Games

Our initial explorations in RTS game data are promising. A number of compelling hypotheses have been generated from the analysis of one sample of RTS game archive data and our pilot experiment with just two users is already showing great promise for quantifying differences in the play of novice and expert users (c.f., left and right plots, Figures 4 & 5) . The combination of a rich and engaging user task with high-fidelity human performance data gather makes the RTS game experiment platform a good choice for further investigations of HRI, generally. For example, some of the dimensions upon which a multi-agent control task might vary are easily explored in an RTS game experiment. As a candidate study, in the near future we hope to use this platform to examine how the number and diversity of units affects human performance in simple tasks like guiding a team of units between points on a map. RTS games (and game-editing utilities built into them) make it possible to vary the terrain, the size of the robot team, and the relative homogeneity or heterogeneity of the team. Metrics such as task time, path efficiency, or avoidance of hazards (i.e. enemies in an RTS game) can be used to evaluate task performance, and game log and eye movement data can reveal differences in supervision and control strategies for different users and different robot team compositions.

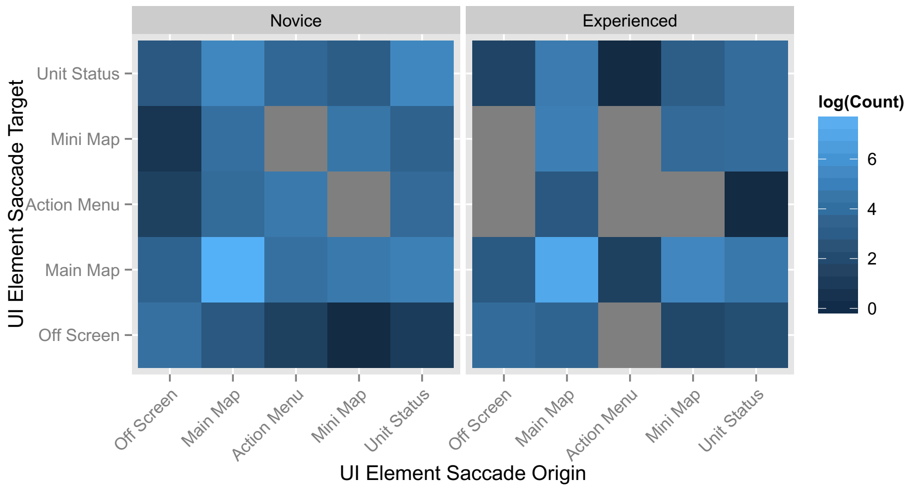

Fig. 5. Saccade origin and target frequency by UI element for a novice and an experienced player, log transformed. While the vast majority of saccades both originate and terminate in the Main Map region of the UI, there are other interesting scan patterns in the data. The novice spends more time scanning off of the display screen than the experienced player, presumably due to a difference in familiarity with keyboard issued commands. The experienced player also very rarely ever scans to the Action menu (as also illustrated in Figure 4), as that information can be determined from game state. Neither the novice nor the expert scan between the Mini Map and Action Menu UI elements. This makes sense, as the information and interactions provided in those controls do not directly relate to each other, regardless of the sophistication of the player.

Another issue that may be amenable to investigation with this approach concerns multi-agent command and control as one component of a complex environment involving several tasks. The UI and user strategies that best support multi-agent supervision when performed as an isolated task may not be the same as those that best support it when performed concurrently with another task (like communicating with another human). Asking participants to play RTS games in multi-tasking experiments may aid us in understanding how control interfaces should be altered for busy workplaces.

A related issue concerns collaborative control and human-human communication. RTS game support multiplayer matches where a two or more players can control teams of units in a single environment and can act in a collaborative or adversarial (or ambiguous) manner with other human or computer players. Future experiments may investigate verbal (or other) communication patterns among human players and may also reveal changes in how SA is acquired and maintained when multiple humans operate robotic teams in a shared environment.

In general, it seems promising to exploit the similarity between RTS games and multi-agent command and control even though these are not strictly identical domains. The availability of archive data, the existence of an population of expert users, and the ease of implementing interesting experiments all support fast and effective data collection. The resulting data will inform our understanding of interface design, our understanding of human performance and strategy, and our understudying of situational awareness in complex tasks.

References

1. Walter, R., Boot, D.P.: Blackely, and Daniel J Simons. Do action video games improve perception and cognition? Frontiers in Psychology 2 (2011)
2. Ellis, S.R., Yeom, K., Adelstein, B.D.: Human control in rotated frames: anisotropies in the misalignment disturbance function of pitch, roll, and yaw. In: Proceedings of the Human Factors and Ergonomics Society (2012)
3. Green, C., Kalar, D.: Design inspiration for interactions between humans and multiple robots. Presentation at the 4th IEEE Conference on Space Mission Challenges for IT (2011)
4. Jones, H.L., Snyder, M.: Supervisory control of multiple robots based on a real-time strategy game interaction paradigm. In: Proc. of the 2001 IEEE International Conference on Systems, Man, and Cybernetics (2001)
5. Kalar, D., Green, C.: Understanding situational awareness in multi-unit supervisory control through data-mining and modeling with real-time strategy games. In: 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE (2012)
6. Lewis, J.M., Trinh, P., Kirsh, D.: A corpus analysis of strategy video game play in Starcraft: Brood War. In: Miyake, N., Peebles, D., Cooper, R.P. (eds.) Proceedings of the 34th Annual Conference of the Cognitive Science Society, pp. 671–676 (2011)