10 years ESJ
*Special edition*

# A Robot Advisor to Improve Computerized Game Play

***Dustin Tanksley, BS***
Missouri University of Science and Technology, USA
***Carson Arnold***
University of Missouri, Columbia, USA
***Abigail Wilson***
***Daniel Shank, PhD***
***Donald C. Wunsch II, PhD***
Missouri University of Science and Technology, USA

Doi:10.19044/esj.2022.v18n4p50

Submitted: 10 November 2021
Accepted: 20 January 2022
Published: 08 February 2022



*Cite As:*
Tanksley D., Arnold C., Wilson A., Shank D. & Wunsch D.C. (2022). *A Robot Advisor to Improve Computerized Game Play*. European Scientific Journal, ESJ, 18 (4), 50. https://doi.org/10.19044/esj.2022.v18n4p50

## Abstract

This paper explores using a trained machine learning agent as a robot advisor for StarCraft II. A targeted visual representation of the robot advisor decision vector advised players of superior decisions in real-time. The robot advisor provided players with the best decisions given the game state and time remaining. Study subjects had to generalize a game strategy from the robot advisor recommendations to a later game round. We sought to determine whether different advice representations (1) improved performance when an advisor is available, (2) improved subsequent performance when an advisor was not available (i.e., did carry over learning occur?), and (3) whether subjects reported that the robot advice was a positive learning experience. The research design involved a pre-test condition (play without an advisor to gauge initial performance), a test condition (subjects were randomized to receive no robot advice, single-recommendation robot advice, or multiple-recommendation robot advice), and a post-test condition (play without an advisor to gauge performance gains). Some high-performing subjects had a ceiling effect and did not improve over the three experiment rounds. After excluding subjects with a ceiling effect, subjects assigned to the single-recommendation robot advisor showed more learning across the rounds than the subjects in the control group (no robot advisor) or those assigned to the multiple-recommendation robot advisor. In the randomized test round, the single-recommendation robot advisor group outperformed no advisor group or the multiple-recommendation robot advisor group. Our project offers a research framework for evaluating the potential of robot advisors in other training scenarios.

**Keywords:** Robot advisor, game-play, StarCraft II

## Introduction

The educational value of AI agents, robot advisors, or other instructional software is under investigation. Kanda et al. (2004) found that robotic teachers are most effective when they have overcome social and technical challenges. The implementation of a robot teacher in an elementary school for three and a half months showed that a breakdown of communication between the robotic teacher and the students reflected the inability of the robot to create initial engagement, to identify misunderstandings, to be consistent and fair, and robot controller problems (Serholt 2018). The acceptance of robotic fashion advisors depended on the social skills of the robot advisor such as social intelligence, human-likeness, and knowledgeableness (Song and Kim 2020). However, a more social robot is not always the most effective one-on-one tutor. Students tutored by a robot in math showed gains dependent the student's pre-existing skill level rather than robot sociability (Konijn and Hoorn 2020). Consumer perceptions of the level of skill of an financial AI advisor software positively influenced consumer trust (Yokoi and Nakayachi 2019). When AI agent proficiency was measured by employee efficiency at two hydrocarbon processing companies, proficiency correlated with the quality and quantity of validated information available during AI agent training (Walker 2019). First-year engineering students performed better in a beginner-level computer programming course when aided by a virtual debugging advisor program that gave student feedback (Lee et al., 2018).

There is little research on the utility of AI advisors as coaches for video games, especially for real-time video games like StarCraft II, which provides complex scenarios akin to real-world situations. We developed AI advisors for a StarCraft II minigame and conducted experiments on human subjects to assess their efficacy. The choice of StarCraft II was fortuitous, as DeepMind and Blizzard Entertainment worked together to release an API for the game in 2017 (Vinyals et al., 2017). This API allows large amounts of game data to be downloaded and permits commands to be uploaded back to the game. In addition, due to the communication architecture, humans and AI robots can interact with the game simultaneously through separate interfaces.. StarCraft II is a real-time strategy game, meaning actionable information must be shared with players in an easy-to-understand format to support real-time decision-making. If not, the data would be too complex to allow players to play the game and accept inputs from a robot advisor. Furthermore, DeepMind has improved its algorithms and has reached a grand master level in ranked online matches (Vinyals et al., 2019).

We used a machine-learning algorithm to train our robot advisor to advance this research. All advisor strategies were generated through self-play, with no prior human knowledge, allowing measurement of how effectively the knowledge gained by the AI agent can be transferred to human participants. This research begins to bridge the gap between human learning and AI learning in a real-time setting.

## Methods And Subjects

### *Experimental Design*

We created two robot advisors to provide StarCraft II minigame players with real-time advice. We recruited human subjects to test the effectiveness of this advisor. Subjects were instructed on playing, then played the same StarCraft II minigame three times. After the play was complete, subjects answered a questionnaire. In the first and third sessions, participants played with no advisor. In the second session, subjects were randomly assigned to one of three experimental conditions: (1) no advisor (the experimental control condition), (2) the single-recommendation advisor, or (3) the multi-recommendation advisor. Both robot advisors provided optimal playing recommendations, with one presenting a single bar showing the best advice (single-recommendation advisor) and the other presenting multiple bars showing the relative strength of each recommendation (multi-recommendation advisor).

### *The Minigame Scenario*

The Build Marines minigame from (Vinyals et al. 2017) was used for this experiment. It is one of the seven minigames created by Blizzard and DeepMind to measure performance for StarCraft II Learning Agents. In this minigame, the player must create units and buildings efficiently to maximize *Marines* produced within the time limit. Possible actions include **Train SCV**, which increases mineral collection rate; **Build Supply Depot**, which increases the maximum number of units and unlocks the barracks; **Build Barracks** the building which can train Marines; and **Train Marine**, which is the final goal of this scenario but provides no benefits other than increasing score. Additionally, a **No-Op** action is included for simply doing nothing. The action space was reduced to only actions resulting in Marine production, making it easier to train the robot advisor and display the information to the subjects.

The original scenario was modified with the time limit reduced from 15 minutes to 6 minutes to allow multiple runs of the minigame and observe whether the robot advisor influenced learning over multiple trials. The game was run on the *faster* game-play speed mode, reducing each round to 4 minutes, 17 seconds.

*The Agent*

The agent used for this experiment was an actor-critic neural network (Prokhorov and Wunsch 1997) with a reduced state-space representation. The input space was 27 values from the game, including mineral count, time in-game, number of builds and units, etc. These inputs were passed into two linear layers of 128 nodes with a 50% dropout layer, followed by an output layer for the five available actions. These actions were masked depending on whether the requirements to perform each action were met. The hyperparameters for training were a learning rate of 0.03 and a discount factor of 0.99. The agent was trained until it reached an average score of 100, which required several hundred training games. Elnabarawy et al. (2020) has shown that this method can reach super-human performance with an in-depth hyper-parameter search and additional training. However, this level of performance is irrelevant to this experiment, as reaction time determines success after an optimal strategy has been selected. The agent is robust enough to score well in the Build Marines minigame (Figure 1). It does not outperform expert human players, but it does perform at a competent level, making it ideal for training novice human players.

| Agent | Metric | Build Marines Score |
|---|---|---|
| Random Policy | Mean | <1 |
| | Max | 5 |
| Random Search | Mean | 8 |
| | Max | 46 |
| DeepMind Human Player | Mean | 138 |
| | Max | 142 |
| StarCraft Grand Master | Mean | 133 |
| | Max | 133 |
| FullyConv LSTM | Mean | 6 |
| | Max | 62 |
| Our Actor Critic Agent | Mean | 100 |
| | Max | 104 |

**Figure 1:** Comparison of our Actor-Critic model against results in DeepMind's publication introducing the StarCraft II minigames (Vinyals et al. 2017).

*The Advising Interface*

The evaluation for each game step was normalized, so each possible action was converted to a weighted score such that together they summed to one. This weighted score was averaged with previous steps, such that the stored output was 99% of the previous output and 1% of the new evaluation. This value was updated every 200 milliseconds (5 times per second). Frequent updating smooths out the rapid transitions of the evaluation, requiring about 14 seconds of a continuous high output to transition from one preferred action to another. Over a minute, the previous output would have decayed to 5% of the total value. The only difference between the two advisors was that the multiple-recommendation advisor displayed all the values for all five actions. In contrast, the single-recommendation advisor showed only the top preferred action. As illustrated in Figure 2, The multi-recommendation advisor primarily recommends the player build more barracks based on the dominant need, with a relatively small secondary recommendation to train more workers. These bars are smoothly adjusted as the relative weight of the recommended changes. The single-recommendation advisor does not show a secondary recommendation and shifts to another recommendation when it becomes preferred.

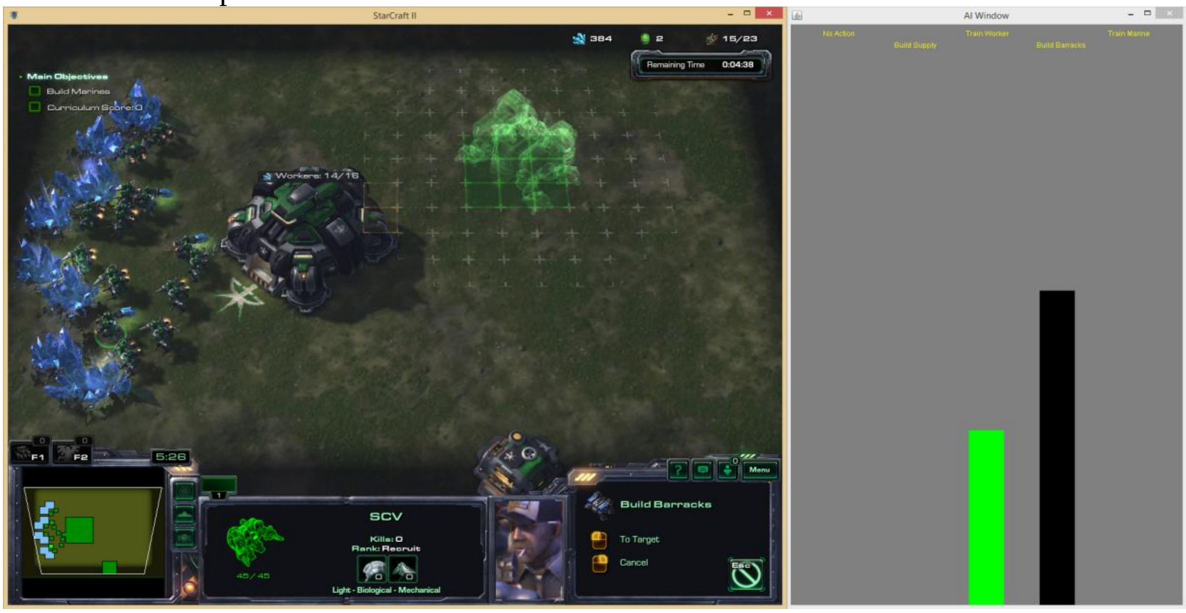


**Figure 2.** An example user interface with recommendations to Build Barracks and TrainWorkers.

*Human Subjects*

Subjects were recruited through two methods that included seventeen subjects contacted through listservs (paid $10 for their participation) and sixteen unpaid subjects who volunteered through the psychology research pool (receiving credit in their Introduction to Psychology course). The thirty-three subjects included ten women and twenty-three men, ages 18 to 28 (mean 21.0). Students majored in computer science/engineering (10), other types of engineering (15), other non-engineering majors (7), or were undeclared (1).

The majority (26) had never played StarCraft II before. Subjects were randomly assigned to conditions, eleven subjects per condition.

*Procedure*

Subjects gave informed consent upon arrival in the Psychology Research Laboratory. Participants were given a StarCraft II reference sheet (available on request) and then directed to a computer monitor. Subjects reviewed the reference sheet, and a research assistant answered any questions about the game. They watched a brief video tutorial on the StarCraft II minigame twice.

Subjects were asked to move to another computer, where they played through each of the three rounds of the minigame in succession. The robot advisor window was always in the right third of the screen, with the minigame in the left two-thirds (Figure 2). In rounds with no advising, the advisor window remained blank. Upon completion of the final round, subjects filled out a short post-study questionnaire, including their perceptions of each round, demographics, and previous experience with StarCraft (available on request). We recorded the number of minerals, workers, barracks, supply depots, and *Marines* (the goal) for each round.

## Results

Across all three experimental conditions, subjects improved with subsequent game rounds. In the Round One pre-test, an average of 14.2 *Marines* was produced. In Round Two, where some subjects had advisors, an average of 16.7 *Marines* was made. In Round Three post-test, an average of 18.7 *Marines* were created. The control condition showed linear learning over each round going from 14.6 to 16.7 to 18.8, increasing by about 2 per round and a total increase of +4.2 (Figure 3). The *Marines* produced in the single-recommendation advisor condition went from 12.5 to 16.8 to 18.2 over the three rounds, for a total increase of +5.6. The multi-recommendation advisor condition went from 15.4 to 16.7 to 19.2 for a total gain of +3.8 *Marines*. The single-recommendation advisor showed the greatest increase across rounds, due in part to the subjects in that condition having lower initial scores in the pre-test round.

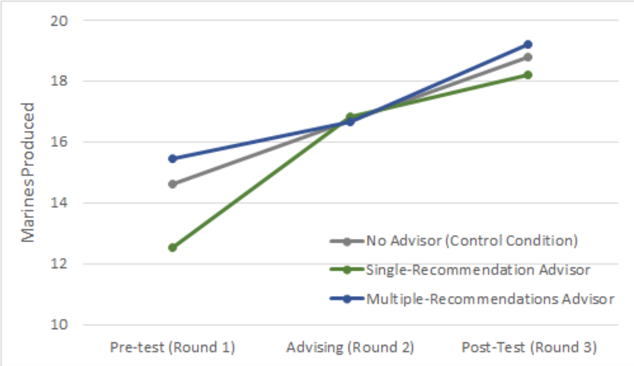


**Figure 3:** Marines produced by round and by experimental condition.

The highest number of *Marines* obtained across all rounds was 26, suggesting that the 13 subjects who scored 20 or more *Marines* on the pre-test round had less room to improve based on AI robot advisor. Our second analysis removed these 13 subjects with a ceiling effect. Figure 4 shows performance across rounds with the different conditions. In this analysis, the control group went up 6.7 from pre-test to post-test, whereas the single-recommendation group increased by 7.7 and the multiple-recommendation group increased by 7.3.

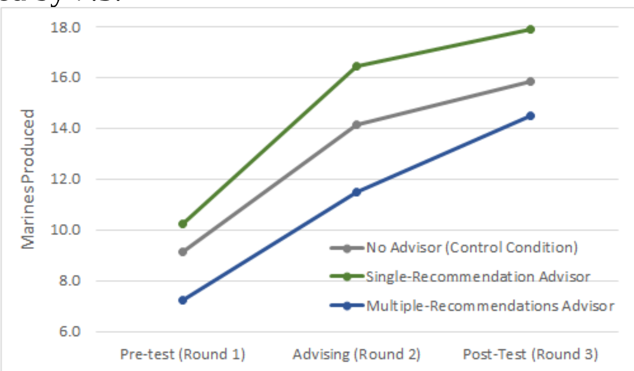


**Figure 4:** *Marines* produced by round and experimental condition with 13 ceiling effect subjects removed.

Figure 5 shows a third analysis of the data which removed 15 subjects who had no substantial increase in *Marines* from pre-test to post-test (a change of 2 or less was considered a non-substantial change; 4 of these subjects were in the control condition). After deleting 15 subjects who failed to improve game-play across rounds, the learning curves for those who improved are shown in Figure 5. These subjects show substantial improvements across rounds, with differences in improvement related to the experimental condition. The no advisor control condition shows an average increase of 7.1 *Marines*, whereas the single-recommendation shows an average of 12.5 *Marines*, and the multiple-recommendation advisor shows an average of 8.3 *Marines*. Two analyses (Figures 5 and 6) show that the robot advisors increased learning compared to the no-advisor control condition, with the single-recommendation advisor showing the greatest improvement.

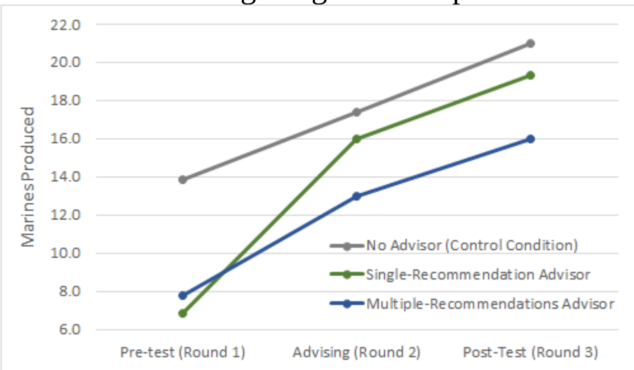


**Figure 5:** *Marines* produced by round and experimental condition with 18 non-learning cases removed.

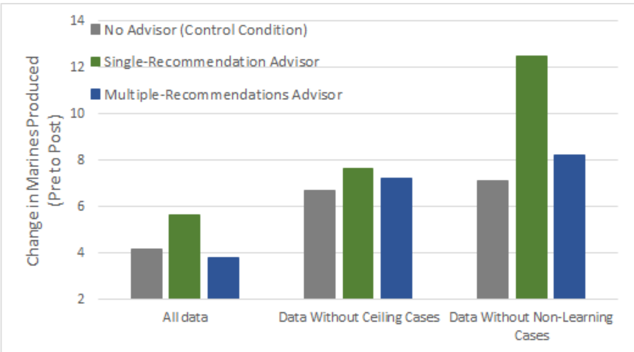


**Figure 6:** Changes in *Marines* produced from pre-test to post-test grouped by condition and analysis

Additionally, we graphed the same performance as in Figure 3 in Figure 7, based on subjective self-report of the subjects from the follow-up questionnaire. Because this assessment was made after the experiment, subjects may have skewed their perception of their performance based on hindsight bias. However, we found moderate correlations between actual performance and perceived performance for each round (0.40, 0.35, and 0.43, respectively). Based on a 1 to 10 point rating scale, subjects in all conditions perceived that they performed at the 4.1 to 5.5 level on the first round, better on round 2 (6.2 to 7.1), and still better on round 3 (6.9 to 8.1). There is a slight variation in total gain from pre- to post-test, with the control condition increasing 2.5, the single-recommendation condition increasing 2.8, and the multiple-recommendation condition increasing 2.7. However, both advisor conditions are in a concave shape in Figure 7, with round 2 increasing more than a linear progression would suggest. Finally, subjects reported that the single-recommendation robot advisor was moderately helpful at 4.3 (again on a scale of 1 to 10). In contrast, the multiple-recommendation robot advisor was perceived to be more helpful at 6.4.

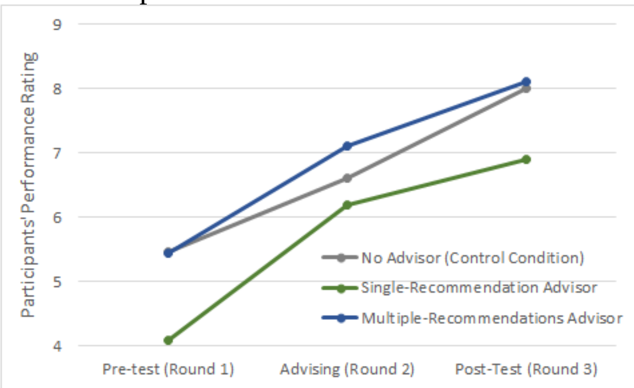


**Figure 7:** Subjective (self-reported) subject performance by round and experimental condition.

## Discussion And Conclusions

Across all the study subjects, the robot advisor had little impact on game performance. The single-recommendation advisor outperformed the control condition, but the multi-recommendation advisor did not (Figure 6). However, when subjects who started at a high initial level are removed from the analysis, or subjects who failed to improve are removed, a different pattern emerges (Figure 6). While subjects in all conditions improve each round, conditions with a robot advisor show greater learning than those without. Learning is especially marked for the single-recommendation advisor, where subjects showed the most improvement. This observation suggests that the focus on a single recommendation may have made it easier for subjects to learn the strategy provided by the robot advisor. In contrast, the multiple-recommendation robot advisor may have added unneeded complexity.

Another interesting trend is that the subjects who got the single-recommendation advisor rated their performance significantly lower, despite achieving similar or even superior performance. This observation could reflect that these subjects had a better understanding of the robot advisor's strategy and did not perceive that they had adopted the optimal approach.

Limitations of this research include ceiling effects (some subjects scored high on the pre-test round) and subject heterogeneity (variability in participant ability and learning trajectory). Future research can address these limitations by recruiting more subjects and designing a minigame scenario that fosters less initial variability in subject performance.

These results are quite promising given the novelty of this research. Teaching new skills to naïve human subjects is usually done with a combination of didactic instruction, worked examples, and practice opportunities. Our research begins to address the question as to whether traditional teaching methods should be supplemented with robot advisors.

Our test scenario required subjects to play while advised of the optimal move with no formal instruction on the underlying strategy. The evaluation was done under game conditions that included real-time variability and time pressures. Nevertheless, subjects could use the knowledge gained from the advising round to improve performance in the post-test round. Given that we have used a machine learning method that generated its advice based on self-play, we have demonstrated the potential for robot advisors to learn from game-play and transfer knowledge gained to human players in real-time.

**Acknowledgments**

Partial support for this research was received from the Missouri University of Science and Technology Intelligent Systems Center, The Missouri University of Science and Technology Chancellor's Distinguished Fellowship, the U.S. Department of Education Graduate Assistance in Areas of National Need program, the Mary K. Finley Missouri Endowment, the National Science Foundation, the Lifelong Learning Machines program from DARPA/Microsystems Technology Office, and the Army Research Laboratory (ARL). It was accomplished under Cooperative Agreement Number W911NF-18-2-0260. The views and conclusions contained in this document are those of the authors. They should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation.

**Conflicts of Interest**

The authors have no conflicts of interest to disclose.

**Human Studies**

The Institutional Review Board of the Missouri University of Science and Technology approved these studies.

**References:**

1. Elnabarawy, I., Arroyo, K., & Wunsch II, D. C. (2020). StarCraft II Build Order Optimization using Deep Reinforcement Learning and Monte-Carlo Tree Search. Retrieved at https://arxiv.org/pdf/2006.10525.pdf

2. Kanda, T., Hirano, T., Eaton, D., & Ishiguro, H. (2004). Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial. *Human-Computer Interaction*, *19*(1/2), 61–84.
3. Konijn, E. A., & Hoorn, J. F. (2020). Robot tutor and pupils' educational ability: Teaching the times tables. *Computers & Education*, *157*, retrieved at https://doi.org/10.1016/j.compedu.2020.103970.
4. Lee, V. C. S., Yu, Y. T., Tang, C. M., Wong, T. L., & Poon, C. K. (2018). ViDA: A virtual debugging advisor for supporting learning in computer programming courses. *Journal of Computer Assisted Learning*, *34*(3), 243–258.
5. Prokhorov, Danil V., and Donald C. Wunsch. (1997). Adaptive Critic Designs. *IEEE Transactions on Neural Networks* 8 (5): 997–1007.
6. Serholt, S. (2018). Breakdowns in children's interactions with a robotic tutor: A longitudinal study. *Computers in Human Behavior*, *81*, 250–264.
7. Song, S. Y., & Kim, Y. (2020). Factors influencing consumers' intention to adopt fashion robot advisors: Psychological network analysis. *Clothing and Textiles Research Journal,* doi:10.1177/0887302X20941261
8. Vinyals, Oriol, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, et al. (2017). StarCraft II: A New Challenge for Reinforcement Learning. *ArXiv:1708.04782 [Cs]*,
9. Vinyals, Oriol, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, et al. (2019). Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning. *Nature* 575 (7782): 350–54.
10. Walker, A. (2019). Streamline critical information searches with advanced data techniques. *Hydrocarbon Processing*, *98*(1), 23–24.
11. Yokoi, R., Nakayachi, K. (2019). The effect of shared investing strategy on trust in artificial intelligence. *Japanese Journal of Experimental Social Psychology*, *59*(1), 46-50.