Using a Cognitive Model to Provide Instruction for a Dynamic Task

Jungaa Moon (jungaam@andrew.cmu.edu)
Department of Psychology
Pittsburgh, PA 15213 USA

Dan Bothell (db30@andrew.cmu.edu)
Department of Psychology
Pittsburgh, PA 15213 USA

John R. Anderson (ja+@cmu.edu)
Department of Psychology
Pittsburgh, PA 15213 USA

Abstract

The current study used the Space Fortress game (Donchin, 1989) to study the effects of training and instruction in acquisition of complex skills. The game requires flexible coordination of perceptual, cognitive and motor components in a dynamically changing environment. We examined whether effective instruction can be developed for such a task in the same way that instruction is developed for academic tasks. Instruction was developed for certain aspects of the game based on a set of explicit procedural rules in an ACT-R model that plays the game. Participants who were given these instructions were significantly better at handling those aspects of the game that the instructions targeted. The results indicate that it is possible to perform a task analysis of a dynamic task, develop explicit instructions from the analysis, and improve target skills. The results further provide implications for designing training and instructional systems for dynamic skill acquisition.

Keywords: skill acquisition; Space Fortress game; ACT-R cognitive architecture.

Introduction

There has been a considerable history of taking cognitive models for the performance of various academic tasks and building successful instructional programs based on them (Anderson, Corbett, Koedinger, & Pelletier, 1995; Ritter, Anderson, Koedinger, & Corbett, 2007). Much of this work has used computer-based instructional systems where instruction is potentially available after each step of the task. The evidence is sparser for similar success in non-academic, time-pressured tasks. One challenge in providing instructions in such tasks is that processing instruction often interferes performing the task. In a study by Fu and his colleagues (Fu, Bothell, Douglass, Haimson, Sohn, & Anderson, 2006), participants were provided with real-time auditory instructions on an Anti-Air Warfare Coordinator (AAWC – see also Zachary, Cannon-Bowers, Bilazarian, Krecker, Lardieri, & Burns, 1999) task, based on a cognitive model of the task. This resulted in better decisions but slower performance and so no net improvement. It was speculated that this was because of interference in simultaneously processing instruction and performing the task. In this research we investigated whether instruction, based on a cognitive model, but given prior to the performance of a task, would improve performance of the task.

We chose to pursue this issue within the context of the Space Fortress game, a computer-based video game. The Space Fortress game (Donchin, 1989) was developed for the learning strategy program initiated by DARPA to investigate the effectiveness of various learning strategies in complex tasks. The underlying assumption of the program was that there are learning strategies that make practice on complex tasks more efficient. Since then the game has been used in a number of skill acquisition studies to compare the effects of various training and instructional strategies on improving performance, minimizing performance decrements under dual-task conditions or facilitating the transfer of skills to a novel task (Fabiani, Buckley, Gratton, Coles, & Donchin, 1989; Frederiksen & White, 1989; Gopher, Weil, & Bareket, 1994; Ioerger, Sims, Volz, Workman, & Shebilske, 2003; Mane, Adams, & Donchin, 1989; Newell, Carlton, Fisher, & Rutter, 1989; Whetzel, Arthur, & Volz, 2008).

We have developed a cognitive model capable of performing the game and closely matching human performance (Bothell, 2010) in a modern version of the Space Fortress game developed by Destefano (2010). Perhaps because of a change from joystick navigation to key-based navigation common in modern video games, it turns out that the navigation strategy adopted by experts and incorporated in our model (as well as a model by Destefano, 2010) is different than that the optimal strategy reported by Frederiksen and White (1989). We will explore the effectiveness of off-line instruction based on our cognitive model of this navigation strategy.

The Space Fortress Game

The main goal of the Space Fortress game (Figure 1) is to maximize the total scores by navigating a ship to destroy a fortress multiple times and protecting the ownship from the fortress and mines. The player navigates the ship in the

frictionless space by rotating it in a clockwise (by holding down 'D' keyboard key) or counter-clockwise ('A' key) direction or applying thrust ('W' key) to accelerate the ship. The player needs to fly the ship so that it can fly within an area enclosed by two hexagons.

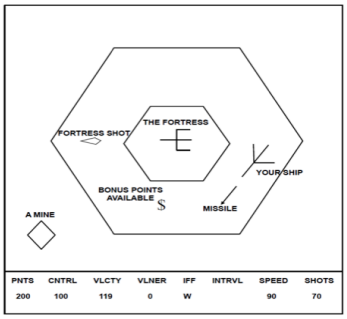

Figure 1: Schematic representation of Space Fortress game.

A fortress stationed in the smaller hexagon rotates like a turret and tracks the ship wherever it moves and fires shells at the ship if the ship stays within one of the fortress sectors (the fixed-size areas each with 10 degree angle surrounding the fortress) longer than 1000 ms. The player has to shoot the fortress with an interval of at least 250 ms between shots to increment the vulnerability of the fortress. Once the vulnerability reaches 10, one can shoot a double shot (two hits with an interval less than 250 ms) to destroy the fortress. A mine appears at a random location in the screen 5 seconds after the destruction of the previous mine and starts pursuing the ship. A mine is 'foe' if a letter associated with the mine (which appears under the IFF label at the bottom of the screen) matches with one of the three alphabet letters shown to the player in the beginning of the game. The player needs to press J key twice with a 250-400 ms interval then shoot a missile to 'destroy' the foe mine. If the letter does not match, the mine is a friend. The player needs to shoot a missile to 'energize' the friend mine. If the player fails to execute timely responses, the mine may collide with the ship. The mine identification task embeds the working memory task (Sternberg, 1966). The player also has to monitor symbols regularly flashing underneath the fortress. When the "$" symbol appears twice in a row, one can collect bonus missiles ('K' key) or bonus points ('L' key). The bonus collection task is similar to a 1-back task that requires judging whether an item matches the item one back in a sequentially presented list of items (McElree, 2001).

The total score consists of four subscores: PNTS, CNTRL, VLCTY, and SPEED. The PNTS scores are earned by destroying fortress, energizing friend mines, destroying foe mines, and collecting point bonuses. They are lost when the fortress or mine damages the ship. The CNTRL scores are accumulated as the ship flies within the hexagon area. Hitting the smaller hexagon or wrapping the space (fly the ship off the edge of the screen) causes one to lose the CNTRL scores. The VLCTY scores are accumulated from flying at an intermediate speed, or steadily lost from moving very fast or very slow. The SPEED scores are earned based on how quickly and accurately one handles mines.

Cognitive Model and the Navigation Instruction

Bothell (2010) developed a cognitive model of the Space Fortress game in ACT-R (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004), a cognitive architecture that simulates cognitive processes of a person working on a task[1]. ACT-R is based on two types of knowledge: procedural knowledge and declarative knowledge. In ACT-R, procedural knowledge is represented as productions, and declarative knowledge is represented as chunks. A production is a condition-action pair and is applied when a certain contingency (a pattern of chunks present) is met. In a dynamic task, the initial set of instructions is converted to chunks and a set of specific productions that operates upon the chunks is generated, which represents how skills are acquired in ACT-R.

The ACT-R model tries to keep the ship aimed at the fortress and orbit in a clockwise direction at the reasonably close distance (approximately 95 pixels from the fortress). The model has a set of productions that respond to various deviations from this ideal pattern to return the ship to the ideal pattern. Because the ship is moving in a clockwise direction most of the direction corrections are right turns. The productions call for a right turn whenever the ship is more than 5 degrees behind a perfect aim at the fortress and a left turn should the ship ever be more than 15 degrees ahead of a perfect aim. Maintaining a perfect aim allows the fortress to be shot at all times and also enables a near perfect circular orbiting pattern. To achieve this perfect circular pattern the rules for thrust are designed to keep the direction of the fortress at a tangent to the circular orbit. Whenever the direction of the fortress is more than 90 degrees from the radius to center a short thrust is issued. Figure 2 compares the orbit achieved by a model with these rules with the orbit achieved by an expert player.

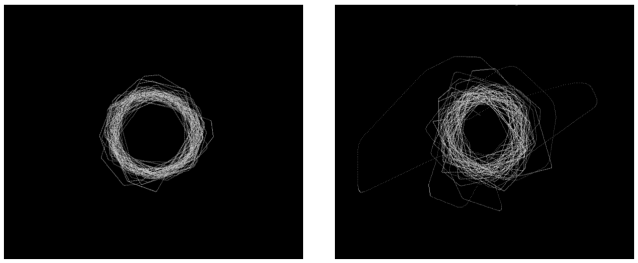

Figure 2: Orbit achieved by model (left) and expert player (right).

[1] The runnable model and a detailed description of that model and its behavioral correspondence are available at http://act-r.psy.cmu.edu/publications/pubinfo.php?id=974. We also have placed there the experimental software that runs with human subject or model and the detailed point structure of the game.

Method

Participants

81 participants completed the four one-hour sessions. Participants were aged between 18 and 40 and had normal or corrected-to-normal vision.

Apparatus

Participants played the Space Fortress game using a computer keyboard to make inputs while the Space Fortress game was displayed on a monitor. The version of the Space Fortress game used in the study was the Pygame Space Fortress (Destefano, 2010) based on the Python programming language.

Procedure

Each session consisted of 17 three-minute games. In the first session, participants read a default instruction describing the basic rules of the Space Fortress game before they started the first game. In the beginning of each game, a screen with three alphabet letters for mine identification was displayed. At the end of each game, total score and subscores earned in that game were displayed.

Design

Participants were randomly assigned to one of the four conditions defined by crossing 'instruction' (whether they received the navigation instruction) and 'training' (whether they played training games in the first two sessions).

Instructional Intervention

After the fifth game in the first session, instruction participants received a written 'navigation instruction' based on the ACT-R model (in addition to the default instruction all participants read before the first game started)[2].

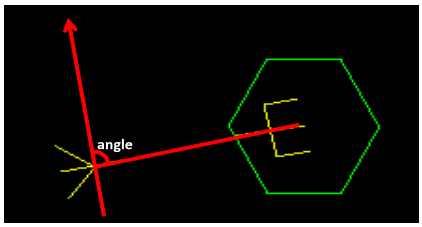

Figure 3: Illustration of the second rule

The instruction described the following set of rules. First, the ship always has to make its tip point straight at the fortress. This can be achieved by constantly rotating the ship in a clockwise direction (by tapping the D key) to the appropriate degrees and overcorrecting when necessary (by tapping the A key). The second rule assumes that the first rule is in effect and the ship is reasonably close to the fortress. When the ship points straight at the fortress, two vectors are formed (Figure 3). One of the vectors describes the direction the ship is aimed which should be the fortress. The second vector is the direction in which the ship is drifting. When the angle formed between these two vectors just exceeds 90 degrees, one has to apply thrust by tapping the W key. Following the vector addition rules, thrust applied at this moment keeps the ship's trajectory tangent to the circular orbit it is traveling. Constantly applying thrust at the right moment keeps the ship stay in an orbit without drifting away from the fortress.

In addition to the procedural rules above, the instruction also provided a couple of additional pieces of information based on the ACT-R model. First, the instruction emphasized the importance of maintaining a close distance between the ship and the fortress. Staying relatively close to the fortress makes aiming at the fortress easier (Frederiksen and White, 1989). Second, the instruction taught participants to 'tap' the keys instead of holding them down for an extended amount of time.

Training Intervention

Training games was designed for training participants in the first two sessions so that all games involved the basic components of navigation and dealing with the fortress but gradually introduced additional tasks (mine handling and bonus collection) and increased the speed of game objects (mines and fortress). Both to allow contrast with a control condition and to make sure participants did not lose track of the final task they would have to perform we interspersed 20 'baseline games' (identical to the standard games) across the 4 sessions. Note that non-training participants played standard games instead of training games.

Results

All participants performed the same 20 baseline games (1, 5, 9, 13, and 17th game in each session). Reflecting the wide range of individual differences, the scores earned on the first baseline game (entry scores) ranged from -4597 to 1930, with mean -2182 and SD 1537.

We performed an analysis of covariance on the 19 baseline games after the first, using the first baseline game as a covariate. This revealed a significant benefit of instruction ($t(76) = 1.73$, $p < 0.05$, one-tailed) and no effect of training ($t(76) = .40$). As training intervention failed to have significant effects (including interactions with instruction) in the more detailed analyses that follow, we will just average over this variable from this point forward.

Figure 4 shows the difference between the instruction group and no instruction group as a function of games. To help appreciate the effect of ability on performance in this task we have divided the population into two halves – those who had the higher scores and those who had the lower scores in the first baseline game. Although there might appear to be an interaction with ability, there is not a significant interaction between ability and instruction ($F(1,72) = 1.12$) or a significant three-way interaction with

[2] The navigation instruction is also available at http://act-r.psy.cmu.edu/publications/pubinfo.php?id=974.

these variables and games (F(18,1296) = 0.65).

The navigation instruction concerned optimal control of the ship for purpose of shooting at the fortress. The total score, which is not a very sensitive measure of what has been learned, consists of four subscores which are more focused in what they represent. Table 1 shows the effect of instruction on the total score and subscores in 17 games in session 4. It also shows the maximum possible on each score. The actual maximum for PNTS cannot be exactly placed because it depends on some random factors that vary from game to game. What is apparent by comparing the actual scores with the maximum scores is that the CNTRL and VLCTY components of the total measure are near their maximum already while the PNTS and SPEED scores are not. The PNTS and SPEED score also show much greater variation among participants.

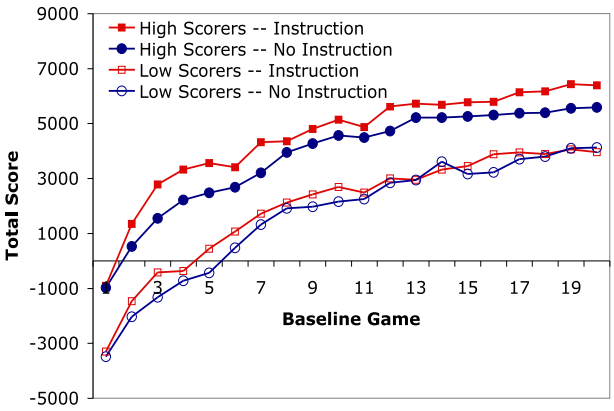

Figure 4: The total scores in 20 baseline games.

Instruction has a significant effect on only the CNTRL subscore. An effect on CNTRL would be expected because control of ship is one of the components targeted by the instruction. Instruction should also have a positive effect on the destruction of the fortress. This is represented in the PNTS measure but PNTS also reflects other components of the task and so is not a pure measure of fortress-handling.

Table 1: ANCOVAs on score measures in session 4.

Score measures	Mean (No-Inst)	Mean (Inst)	Max	Standard Deviation	F	*p*
total	4660	4992	>11000	1443	1.98	.163
PNTS	604	768	>5000	697	2.06	.155
CNTRL	931	996	1080	117	8.46	.005
VLCTY	1017	986	1260	165	0.93	.337
SPEED	2206	2322	3600	577	1.24	.269
IR	1513	1761	4780	578	6.05	.016
NIR	3238	3250	>6000	1077	.00	.953

Since the subscore measures do not provide enough resolution to investigate the influence of the instruction on the relevant parts of the task, we created more specified subscores: Instruction-Relevant (IR) and Non-Instruction-Relevant (NIR) scores by classifying the measures into two categories. The purpose of creating these measures was to determine if the instruction indeed influenced performance in the measures that it was targeted to improve. Specifically, due to the hierarchical relations between maintaining an optimal orbiting pattern and destroying a fortress (Frederikson & White, 1989), the instruction taps both the CNTRL and the 'fortress-handling' part of the PNTS, which belong to the IR components. The 'non-fortress-handling' part of the PNTS (earned by handling mines, collecting point bonuses, and avoiding ship damage), the SPEED (earned by handling mines), and the VLCTY (earned by maintaining an intermediate ship speed) were classified as the NIR components since they were not directly concerned with the instruction.

The equations below were used for the calculation of the IR and NIR. Note that the CNTRL and PNTS are based on the default Space Fortress score calculation:

IR = CNTRL + fortress-handling-PNTS

fortress-handling-PNTS = 100*(The number of fortresses destroyed) – 50*(The number of ship damage from fortress)

NIR = total – IR

The results (also in Table 1) confirmed the positive influence of the instruction on the IR scores but not on the NIR scores.

Having shown that the instruction does have an effect on the score measures, we then looked at various measures of performance that contributed to the scores. The purpose of this set of analyses was to investigate how the instruction influenced participants' behaviors in more fine-grained measures and to identify which of them contributed to the higher IR scores. See Table 2 for the result of analyses (a and b) as well as the detailed description of the measures (c). As in the previous analyses, we performed ANCOVAs with training and instruction factors as independent variables and total score in the first game in session 1 as a covariate. For every variable we will look at there was a strong relationship between it and the first game total score in session 1, usually at the .001 level.

Table 2: ANCOVAs on performance measures in session 4.

(a) Ship Control	Mean (No-Inst)	Mean (Inst)	Standard Deviation	F	*p*
Wrapped	1.16	.51	1.42	4.91	.030
Hit Small Hexagon	0.61	0.55	0.57	0.24	.623
Distance	157.02	137.51	30.56	12.19	.001

(b) Fortress Handling	Mean (No-Inst)	Mean (Inst)	Standard Deviation	F	*p*
Fortresses Destroyed	7.94	9.74	4.67	4.22	.043
Hit Fortress	99.11	122.35	59.49	4.17	.045
Shots at Fortress	147.02	170.39	66.72	3.12	.081
Aiming Accuracy	.63	.68	.11	6.42	.013
Aimed Percentage	0.50	0.58	0.15	12.79	.001

(c) Measure	Description
Wrapped	The number of times the ship flied off the edge of the screen
Hit Small Hexagon	The number of times the ship hit the small hexagon
Distance	The average distance in pixels between the ship and the fortress
Fortresses Destroyed	The number of fortresses destroyed
Hit Fortress	The number of times the fortress was hit by missiles
Shots at Fortress	Number of times the fortress was shot at
Aiming Accuracy	Hit Fortress/Shots at Fortress
Aimed Percentage	Ratio of time spent while the ship was aimed at the fortress to the total time spent while the ship was alive

Table 2a presents analysis of three measures relevant to control of the ship – how often it wrapped around the space, how often it hit the inner hexagon, and how close it stayed to the fortress. The first two are measures of lack of control while the third is a measure of good control in that it is easier to aim at the fortress the closer one is to it. The first and third measures show significant benefits of instruction.

Table 2b presents five measures relevant to success at destroying the fortress – number of fortresses destroyed, number of shots that hit the fortress, number of shots taken at the fortress, aiming accuracy, and aiming percentage. All of these measures show a benefit of instruction and in four cases their effects reach statistical significance at the .05 level and the other effect is at the .10 level (note these are two-tailed tests). Instructed participants destroyed 23% more fortresses. It is worth noting that the difference in number of fortresses destroyed is worth 180 points – approximately the difference between the two groups on the PNTS in Table 1. However, because of the noise of other factors that difference is not significant for PNTS in Table 1 but is significant here. Instruction participants also showed higher aiming accuracy and spent larger proportion of time keeping the ship aimed at the fortress. Recall that the instruction encouraged participants to apply the second rule when the two conditions are satisfied: 1) the ship always has to make its tip point straight at the fortress and 2) the ship should be reasonably close to the fortress. Table 2a (distance) and b (aimed percentage) results suggest that instructed participants successfully met those conditions.

General Discussion

The current study provided a set of explicit procedural rules to improve ship navigation performance in the Space Fortress game. The results suggest the positive effect of the instruction in earning more scores as well as improving performance in the measures associated with ship control and fortress handling. The benefits of the instruction were almost exclusively on the ship control and fortress handling.

These effects of instruction are striking in two ways. First they are focused, seeming to target all aspects of the performance that they addressed and having little effect on anything else. This seems clear evidence that one can take a cognitive model of successful performance, identify critical features in its performance, and communicate these features to learners – even for a skill that has such a strong perceptual motor component. Second, it is striking that a one-time instruction given after the fifth game in the first session continued to have a positive effect in the fourth session. One might expect the effect would have been stronger had these instructions been repeated.

Given the difficulty of the task and relatively little time-on-task (compared to some other Space Fortress studies with more than 20 hours of practice), it is possible that participants with different abilities were in the different stages of skill acquisition by the end of the practices. Assuming the three stages of skill acquisition (Fitts & Posner, 1967), some participants with initially low ability might have exited the study somewhere between the declarative and associative stages while some other participants with initially high ability might have been in the autonomous stage by the end. Despite this, the study provided a promising result: instructing a set of explicit procedural rules on how to perform a major task led to improvement in the targeted task.

Finally, we should note that the instructional effect was not that large. In Figure 4 we can see that it is dwarfed by the ability differences in the experiment. We noted that differences in entry performance were related to prior experience with video games. On the other hand, Figure 4 also shows that these ability differences are dwarfed in turn by the effect of 4 sessions of practice. Both the ability and practice effects reflect the importance of the perceptual-motor components of this task. Participants with more experience at game playing had such skills practiced and the game provided targeted practice of those skills.

Transfer of learning is one of the major issues in skill acquisition studies. Many studies (e.g., Ball, Berch, Helmers, Jobe, Leveck, & Marsiske, 2002; Dustman, Emmerson, Steinhaus, Shearer, & Dustman, 1992) show specificity of training-induced learning – positive effects on the trained task but minimal transfer to novel tasks. On the other hand it has been shown that video game experience leads to improvement in a number of basic cognitive tasks (Boot, Kramer, Simons, Fabiani, & Gratton, 2008; Green, Pouget, & Bavelier, 2010) and the experience can be transferred to real-world tasks (Gopher, Weil, & Bareket, 1994). In light of these two classes of results it needs to be studied how instructional interventions can be designed to provide a generalizable benefit.

Based on the previous success of intelligent tutoring systems in academic tasks, there have been some efforts to bring some of the components of intelligent tutors to real-time, dynamic tasks. For example, individuals trained with intelligent agents as team-training partners showed

improved Space Fortress performance comparable to those trained with human partners (Whetzel et al., 2008). Partner agent protocol (Ioerger et al., 2003) based on cognitive task analysis by Frederikson and White (1989) significantly improved Space Fortress performance compared to individual training. Especially participants who worked with expert level agents showed the most benefit compared to those with novice or intermediate level agents. The explanation is that demonstrating optimal behavior facilitates learning since the learners can model the correct behaviors of their partners. Those results suggest that the instructional intervention can be potentially integrated with training protocols in order to instruct learners to model optimal behaviors of the cognitive model.

The current instructional approach suggests one possible way to instruct learners in dynamic task: provide procedural instruction based on a task analysis of the targeted skills. One can further investigate 1) how to improve the instruction to have transferable benefit and 2) how to optimally integrate the instruction into the task so that the learners can effectively process the instruction.

Acknowledgments

This work was supported by ONR grant N00014-09-1-0402 to Wayne Gray & John Anderson. Correspondence concerning this article should be addressed to Jungaa Moon (jungaam@andrew.cmu.edu).

References

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. Psychological Review, 111(4), 1036-1060.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive Tutors: Lessons learned. The Journal of the Learning Sciences, 4, 167-207.

Ball, K., Berch, D. B., Helmers, K. F., Jobe, J. B., Leveck, M. D., & Marsiske, M. (2002). Effects of cognitive training interventions with older adults: A randomized controlled trial. Journal of the American Medical Association, 288, 2271–2281.

Bothell, D. (2010). Modeling Space Fortress: CMU Effort [PowerPoint slides]. Retrieved from http://act-r.psy.cmu.edu/workshops/workshop-2010/schedule.html

Boot, W., Kramer, A., Simons, D., Fabiani, M., and Gratton, G. (2008). The effects of video game playing on attention, memory, and executive control. Acta Psychologica, 129, 387-398.

Destefano, M. (2010). The mechanics of multitasking: the choreography of perception, action, and cognition over 7.05 orders of magnitude. Unpublished doctoral dissertation, Rensselaer Polytechnic Institute, Troy.

Donchin, E. (1989). The learning strategies project. Acta Psychologica, 71, 1-15.

Dustman, R. E., Emmerson, R. Y., Steinhaus, L. A., Shearer, D. E., & Dustman, T. J. (1992). The effects of videogame playing on neuropsychological performance of elderly individuals. Journal of Gerontology, 47, 168–171.

Fabiani, M., Buckley, J., Gratton, G., Coles, M. G. H., & Donchin, E. (1989). The training of complex task performance. Acta Psychologica, 71, 259-299.

Fitts, P. M., & Posner, M. I. (1967). Human Performance. Monterey, CA: Brooks/Cole.

Frederiksen, J. R., & White, B. Y. (1989). An approach to training based upon principled task decomposition. Acta Psychologica, 71, 89-146.

Fu, W.-T., Bothell, D., Douglass, S., Haimson, C., Sohn, M.-H., & Anderson, J. R. (2006), Toward a real-time model-based training system. Interacting with Computers, 18(6), 1216-1230.

Gopher, D., Weil, M., & Bareket, T. (1994). Transfer of skill from a computer game trainer to flight. Human Factors, 36(3), 387-405.

Green, C. S., Pouget, A., & Bavelier, D. (2010). A general mechanism for learning with action video games: Improved probabilistic inference. Current Biology, 20, 1573-15792.

Ioerger, T.R., Sims, J., Volz, R.A, Workman, J., & Shebilske, W.L. (2003). On the use of intelligent agents as partners in training systems for complex tasks. Proceedings of the 25th Annual Meeting of the Cognitive Science Society, Boston, MA.

Mane, A. M., Adams, J. A., & Donchin, E. (1989). Adaptive and part-whole training in the acquisition of a complex perceptual-motor skill. Acta Psychologica, 71, 179-196.

McElree, B. (2001). Working memory and focal attention. Journal of Experimental Psychology, 27(3), 817-835.

Newell, K. M., Carlton, M. J., Fisher, A. T., & Rutter, B. G. (1989). Whole-part training strategies for learning the response dynamics of microprocessor driven simulators. Acta Psychologica, 71, 197-216.

Ritter, S., Anderson, J. R., Koedinger, K. R., & Corbett, A. (2007) Cognitive Tutor: Applied research in mathematics education. Psychonomic Bulletin & Review, 14, 249-255.

Sternberg, S. (1966). High-speed scanning in human memory. Science, 153(3736), 652-654.

Whetzel, J. H., Arthur, W. Jr., & Volz, R. A. (2008). The effectiveness and efficacy of intelligent agents as team training partners in the acquisition of complex skills in a gaming environment. In H. F. O'Neil & R. Perez (Eds.), Computer games and team and individual learning. Amsterdam, The Netherlands: Elsevier.

Zachary, W., Cannon-Bowers, J., Bilazarian, P., Krecker, D., Lardieri, P., & Burns, J. (1999). The advanced embedded training system (AETS): An intelligent embedded tutoring system for tactical team training. International Journal of Artificial Intelligence in Education, 10, 257-277.