The Need for Speed: Effects of Human Derived Time Constraints on Performance and Strategy in Machine Models of Tetris

Catherine Sibert (sibert@uw.edu)
Department of Psychology, Campus Box 351525
Seattle, WA 98195 USA

Wayne D. Gray (grayw@rpi.edu)
Cognitive Science, Rensselear Polytechnic Institute, 110 8th Street
Troy, NY 12180 USA

Abstract

One of the hallmarks of expert performance in complex, dynamic tasks is the ability to select and perform the appropriate action within a constantly shifting environment, often under tight time constraints. In an example task, the video game Tetris, expert players select placement positions for the active zoid and navigate them into place in increasingly short time spans. Machine models of the same task are capable of producing human-like performance patterns, but either ignore or only roughly approximate the time constraints that seem to be an integral part of human behavior. Using a set of scaled time parameters derived from a large set of human players, we trained and tested an existing machine Tetris model and observed the resultant changes in performance and behavior.

Keywords: Expertise, Reinforcement Learning, Machine Learning, Human Performance

Introduction

Expertise is marked by the ability to perform a particular task at a very high level of proficiency, and in many task domains, are capable of performing more quickly and efficiently than non experts. But while speed is often observed in conjunction with high levels of skill, it is not always clear how it contributes to performance. In this paper, we explore the relationship between speed and strategy in a complex and dynamic task environment, the video game Tetris.

Video games, and Tetris in particular, have a long history of use in research. Gray (2017) identifies three major uses of games: *Gamification* describes efforts to use a game-like environment for more serious and real world applications (Rapp, Cena, Gena, Marcengo, & Console,2016;Nash & Shaffer,2011;Proctor, Bauer, & Lucario,2007), games as *Treatment Conditions* are studies that use games to alter some aspect of human behavior (Holmes, James, Coode-Bate, & Deeprose,2009;Belchior et al.,2013), and *Game-XP* is a term referring to the use of a game as an experimental paradigm. in this work we use Tetris as a task environment to investigate the low level mechanisms that give rise to high level skilled behavior (Kirsh & Maglio,1994;Destefano, Lindstedt, & Gray,2011;Lindstedt & Gray,2019;Sibert, Gray, & Lindstedt,2017;Sibert & Gray,2018). Exploring this task could help to us understand how these skills are developed and how complex strategies are learned and used.

Tetris the Task

Tetris is a real-time, dynamic puzzle solving game that is simple in concept and can be very complex in execution. A player is presented with a sequence of game pieces, called "zoids", made up of four conjoined squares. These zoids fall from the top of the game board, and as they fall, the player navigates them into position using a series of translation and rotation maneuvers. Placed zoids form a pile at the bottom of the screen, and when a row of the pile is completely filled across the width of the screen, that row will disappear, lowering the pile and earning the player some points. The game ends if the pile reaches the top of the screen. Higher scores are achieved by clearing multiple lines (up to four) simultaneously, which encourages players to plan ahead and construct board structures that can support these more complex maneuvers. However, as lines are cleared, the zoid falling speed increases, allowing players less and less time to plan and execute their moves. At early game levels, players have a full 16 seconds for maneuvering, but that time window is slowly reduced until the fall speed of a zoid is a third of a second at the highest playable levels. Successful players must balance the score benefits of complex multi-line clears with the risk of building the pile too high and having insufficient time and space to maneuver zoids into place.

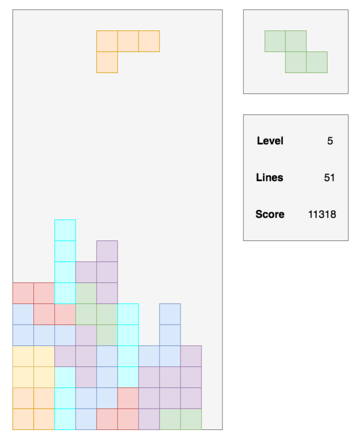

Figure 1: A Tetris game in progress. The active piece, the orange "L" is currently being placed by the player on the main game screen. The player also has access to score information, in the lower right-hand box, and one upcoming piece, the green "Z" in the upper right-hand box.

The constantly changing nature of Tetris allows for a wide range of player skill, with extremely low level players struggling to clear any lines at all, all the way up to extreme experts, who comfortably set up and execute multi-line clears in fractions of a second. Performance in this task is usually judged by the final game score, but that score is achieved though a complex interaction of perceptual, cognitive, and motor skills.

Tetris Models

Building models of Tetris has long been pursed in the Machine Modeling community (Robertson,2003;Fahey,2015;Szita & Lorincz,2006;Thiery & Scherrer,2009a), and these efforts have produced computer models capable of high level Tetris performance. These models are not usually developed out of any specific interest in Tetris behavior, but they use Tetris as a testbed to demonstrate the effectiveness of machine learning algorithms that optimize a large feature space.

Though the search methods and models are all different across the machine learning work, the basic model structure is fairly consistent.The researcher selects a set of features of interest, often structural aspects of the board like the pile height, or number of unfilled cells, that may be important when making placement decisions. Each of these features is assigned a numerical weight, and using these weights, the model is able to calculate a numerical score for all potential placements for any static board state. A game is played by simply selecting the highest scoring move at each decision point, and updating the game board to reflect the previous decision. A simple, yet effective, set of features was defined by Dellacherie (Fahey,2015) and used in the modeling experiments presented here. The features are described in Table 1.

Table 1: Tetris features proposed by Dellacherie, and used to construct the models used in this paper

Feature	Description
Landing Height	Height where the last zoid is added
Eroded Cells	# of cells of the current zoid eliminated due to line clears
Row Transitions	# of full to empty or empty to full horizontal transitions between cells on the board
Column Transitions	# of full to empty or empty to full vertical transitions between cells on the board
Pits	# of empty cells covered by at least one full cell
Wells	a series of empty cells in a column such that the cells to the left and right are both full

As suggested by Table 1, most machine modeling work on Tetris has focused on selecting weights for a limited set of features. These weights must be able to select moves that result in high game scores, but must do so in a nearly infinitely variable environment of potential board configurations. A machine learning method that is capable of producing a successful model of Tetris, then, is likely to also be effective in other complex task domains. One such method, and the one employed in this research, is *Cross Entropy Reinforcement Learning* (CERL), first proposed by Szita and Lorincz (2006) and modified by Thiery and Scherrer (2009a,2009b), and uses a generational search method to narrow in on the optimal feature space.

The models produced by this line of research are very effective Tetris players, clearing hundreds of thousands of lines in a game (a high scoring human clears five or six hundred), but they do so by adopting very un-human-like strategies that allow them to take advantage of significant differences between the human and model task environment. Most notably, the models are completely unconstrained by time pressure. Where human strategy often revolves around making and executing the best placement decisions in the time available, the models instantly choose which of the possible zoid placements has the highest rank (the highest number of alternative placements possible for one episode is 35). For a model player, the game ends when the feature weights encounter a sequence of zoids that it cannot place in a way that clears any lines, far different from the time constraints that limit human players. In response, models and humans develop divergent strategies. The model player emphasizes clearing single lines repeatedly over very long game spans (behavior also exhibited by low level human players), while the expert human player emphasizes setting up and executing as many multi-line clears as possible.

Because of this and other limitations, the models in their original forms are not very informative about human behavior. However, previous modeling work has found that a basic limitation, imposing a hard limit on the length of a model game, could be employed to induce more human-like strategies in the models (Sibert et al.,2017). A follow-up study showed that the human-like strategy of multi-line clears will arise naturally in models in response to a short game condition, even without explicit reinforcement of the score-seeking behavior (Sibert & Gray,2018).

These results provide a strong argument for the importance of time pressure in shaping human behavior, but so far, it has only been implemented in a very simple way. In this paper, we gift our models with human-like time pressure so that rather than StarTrek-like "beaming" each zoid to its final location in a single instant, moves requiring more zoid movements cost more to execute than those which favor fewer.

Methods

As part of an ongoing exploration of human expertise in Tetris, we have collected gameplay data from over 600 subjects through a combination of laboratory studies where Tetris is played in acoustically isolated "research pods" to local and international tournaments where players compete against each other in loud and, at times, raucous events. All data was collected using the Meta-T software (Lindstedt & Gray,2015) and its successors, providing access to a huge array of game information including all board states and key presses.

As shown in Figure 2 and Table 2, players were placed into skill bins ranging from Extreme Novice to Extreme Expert based on the mean performance across their best several games. In addition to score differences between groups, players showed a clear shift in strategy from a nearly complete reliance on single line clears (by novice players) to prioritizing multi-line clears (by expert players).

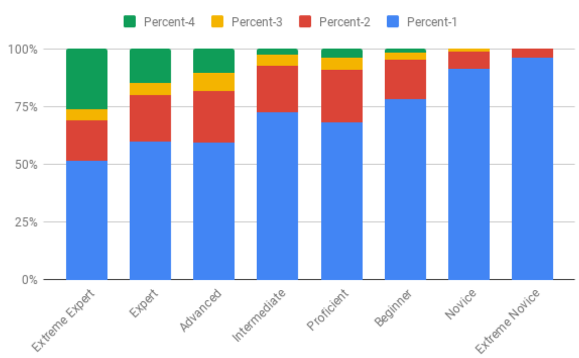

Figure 2: Behavior patterns of human players in each of the eight defined skill bins. The proportion of each bar corresponds to the proportion of each line clear type that players in each group made during their games. Players at the low end of the skill spectrum rely almost entirely on single line clears (blue), while the rate of 4-line clears (green) increases with skill.

Time Parameters

From the raw behavioral data of the human players, we created three measures that capture the execution cost of an individual move: Initial Latency, Average Latency, and Efficiency. These values were logged for all players at every decision point for all games during the experimental period. However, not all these values reflect the actual speed of players. For example, a very advanced player playing at a very slow level may start moving the zoid before a decision is made, and therefore overlapping the costs represented by the Initial Latency and Average Latency parameters (described in more detail below). To account for this, time parameters were not derived from all of a players decisions, but only those made at the Maximum Playable Level, the final level completed before the game was lost. We believe that these decisions are being made at the edge of the player's skill, but are not the random panicked moves often made during the final moments of a game. The parameters for each skill bin are listed in Table 2.

Initial Latency This parameter captures the average time needed for a player to make any key press following the appearance of a new zoid at the top of the screen. When playing to the edge of a player's ability, this initial latency likely captures the processing time to recognize the zoid, some kind of evaluation of possible placement options, the selection of the final placement, and the planning of the keypress sequence required to move the zoid into place. Initial latency is calculated by taking the average time between the zoid's appearance and the first key press for each player. The expertise group value is calculated by taking the average value for all players in the group.

Average Latency This parameter captures the speed of all key presses after the initial key press. As these key presses are likely made after a motor plan has been determined, average latency reflects the motor speed of the player without any decision making or planning. Average latency is calculated by taking the average time between key presses for all but the first key press for each player. The value for each of the 8 expertise groups is calculated by averaging the value across all players in that group

Efficiency A perfectly efficient player will make the minimum number of key presses to move the zoid into position, but humans are prone to all manner of small mistakes that require extra key presses. Zoids are rotated or translated too far and must be brought back into position, players switch paths to pursue an alternate placement, or players simply are unaware that a slightly different order of translations and rotations would require fewer key presses. The efficiency parameter is calculated by determining the optimal path to the final placement and then finding the difference in key presses between this optimal path and the one executed by the player. A player making only the required key presses when maneuvering pieces would have an efficiency of 0. The efficiency parameter is the average number of extra key presses made by a player. The expertise group value is calculated by taking the average value for all players in the group.

Implementing Time Pressure in Models

To capture the effect of time pressure on how long movements could take, we calculated the mean movement time for each type of movement at each of our 8 levels of player expertise (Table 2). In the no time pressure condition, the best placement is calculated and the move is instantly made (This is how all previous Machine Learning Models have worked). The time parameters add the additional step of determining if the move chosen can be made in the time allocated The time cost of a move is calculated by the following formula.

$$TimeCost = InitLat + AvgLat(Path * Eff) \quad (1)$$

The estimated path length (Path) is multiplied by the efficiency parameter (Eff). This extended path length is multiplied by the average latency parameter (AvgLat). Finally, the

Skill Bin	Initial Latency (ms)	Average Latency (ms)	Efficiency
Extreme Expert	79.28	162.11	0.50
Expert	70.56	173.79	0.57
Advanced	96.12	221.72	0.98
Proficient	272.26	433.31	1.64
Intermediate	151.24	311.03	1.16
Beginner	411.29	536.51	1.88
Novice	448.76	668.98	2.06
Extreme Novice	571.37	804.34	2.62

Table 2: Time parameters for each skill bin. Initial and Average latency are measured in milliseconds, and Efficiency is measured in key presses. The Criterion Score is a score based metric reflecting the average of the highest scoring games achieved by players in each group.

initial latency parameter (InitLat) is added to determine the overall time cost (TimeCost).

This time cost is then compared with the zoid drop speed at the current game level. If the move time cost is greater than the time available, the model must choose an alternate move (that does not raise the pile past the top of the screen), and if there are no alternates are possible, the game ends.

Time parameters were implemented into the models in two ways: first, by adding each level of constraints to a high performing model previously trained with no time constraints, and second, by training a model with each set of human-derived time parameters.

Models were trained using the CERL method, reinforced for high score. The model with no time parameters was limited to 525 piece games (a reasonable human game length), but the other models were not explicitly limited in length and relied on the time parameters to restrict game length. The training process ended when the variation of feature weight values dropped below 0.01, and were considered to have converged.

After training, the models were tested by playing a set of ten games using a pre-selected set of game seeds (that would produce the same sequence of zoids). The models were evaluated on their performance, reflected in the scores of these test games, and their behavior, measured by the proportion of line clear types made by the models.

Results

Imposed Time Parameters

Imposing time constraints on a previously unconstrained model caused a significant drop in performance. This performance drop increased as the timecost (see equation 1) became greater. Models slowed to the speed of Proficient players scored hardly any points at all. (See Table 3) The behavior of the model as measured by line clear types remained largely consistent (Table 4), with an emphasis on 4 line clears, until the model was unable to execute line clears of any kind.

Skill Level	Game Length	Lines Cleared	Score
No Time Parameters	409.9 (155.00)	153.1 (67.20)	203766 (133487)
Extreme Expert	278.9 (67.45)	94.4 (26.93)	71878 (32742.14)
Expert	271.1 (64.84)	91.5 (25.92)	68416 (30855.40)
Advanced	231.2 (37.77)	75.9 (15.08)	45000 (12673.27)
Intermediate	116 (51.51)	49.5 (20.14)	23688 (16916.47)
Proficient	41.5 (7.29)	1.6 (1.96)	72 (93.90)
Beginner	24.4 (4.30))	0 (0)	0 (0)
Novice	24.8 (6.03)	0.4 (0.97)	16 (38.64)
Extreme Novice	25.1 (4.95)	0.6 (0.97)	24 (38.64)

Table 3: Performance of models with *Imposed Time Parameters*. Performance is measured by average game length (number of episodes), average lines cleared, and average score. Standard deviations are listed in parentheses.

Trained Time Parameters

Models trained with time constraints were able to score points (see Table 5), even when playing very slowly, though faster models were unsurprisingly higher scoring than slower models. These scores were also roughly equivalent to the average score of human players in the group from which the time parameters were derived. The model behavior also shows a strategy change, with slower models relying almost completely on single line clears, while faster models started shifting toward a multi-line clear strategy. (See Table 4). However, the rate of multi line clears for these models was lower than those of models trained without time parameters.

Discussion

Imposing time pressure on an already trained model resulted in a significant score drop. Indeed, the model was unable to perform at all once slowed to the speed of our mid range players. Unlike humans and unlike models trained at different speeds (Table 6), models trained with no time pressure (Table 4 did not change strategies with changes in drop speed.

In contrast, training models with time parameters produced successful models at all speed levels, and roughly reflected the scores and strategies of the human players in the corresponding expertise groups, with a few notable exceptions. The Extreme Expert group had lower performance than the Expert group because the former has a higher initial latency parameter, making them, on paper, slower. We believe this slowdown is reflective of a higher decision quality by extreme expert players, but as the models all have the same de-

Skill Level	1 Line	2 Line	3 Line	4 Line
No Parameters	20.51% (3.44)	25.48% (7.51)	20.00% (11.36)	34.01% (7.86)
Extreme Expert	20.55% (5.37)	26.63% (9.23)	21.62% (22.34)	31.20% (6.33)
Expert	21.25% (6.03)	26.15% (8.32)	20.51% (11.60)	32.09% (6.27)
Advanced	21.84% (6.47)	27.66% (10.48)	19.52% (10.65)	31.00% (8.65)
Intermediate	22.60% (7.68)	21.21% (12.60)	25.17% (10.66)	31.02% (16.43)
Proficient	73.33% (41.31)	26.67% (41.31)	0% (0)	0% (0)
Beginner	0% (0)	0% (0)	0% (0)	0% (0)
Novice	100% (0)	0% (0)	0% (0)	0% (0)
Extreme Novice	100% (0)	0% (0)	0% (0)	0% (0)

Table 4: Behavior of models with *Imposed Time Parameters*. Behavior is measured by the average percentage of total lines that are cleared with each line clear type (1, 2, 3, or 4). Standard deviations are shown in parentheses.

Skill Level	Game Length	Lines Cleared	Score
Extreme Expert	378 (110.42)	134.3 (44.34)	80350 (45401.84)
Expert	392.1 (90.54)	139.5 (36.03)	95552 (46846.02)
Advanced	337.3 (31.26)	118 (12.45)	32434 (7209.18)
Intermediate	170.1 (64.80)	50.7 (26.05)	23196 (19153.34)
Proficient	115 (80.24)	30.7 (31.50)	4582 (5652.35)
Beginner	134.4 (34.89)	38.6 (13.75)	4514 (3031.51)
Novice	94.2 (34.58)	23.6 (13.66)	2134 (1869.31)
Extreme Novice	66.4 (18.77)	14.1 (7.82)	928 (606.06)

Table 5: Performance of models with *Trained Time Parameters*. Performance is measured by average game L=length (number of episodes), average lines cleared, and average score. Standard deviations are listed in parentheses. For all test games, the game length was unlimited but dependent on the model's speed.

cision ability, the small speed difference results in a slightly lower score. In addition, players in the Extreme Novice group scored much lower than their corresponding model, likely because even when very slow, the models are making better quality decisions than the players. These deviations at the extreme ends of the skill range suggest that a player's skill level is determined by a combination of their speed and their decision quality, rather than just by speed alone.

There were two more deviations between the performance and behavior of the human and the model. First, the Proficient model scored much lower than the corresponding human players. Second, although the Advanced model achieved a score that was close to its human counterparts, it relied primarily on clearing single rather than multiple lines (see Table 4).

The above two cases demonstrate the pitfalls of our model search method. While hundreds of potential models are tested and evaluated during training, the development process is run only once. Under most circumstances, we consider the wide breadth and length of the search sufficient to prevent overfitting, the CERL method can sometimes converge on a locally optimal area of the feature space. Additional noise introduced into the search, or perhaps another search method altogether could possibly mitigate some of these problems and produce more robust models.

Our biggest surprise was the behavior of the time trained models. Though models with faster time parameters displayed higher percentages of 4-line clears, these percentages were lower than the human players in the skill groups from which those parameters were derived, and also lower than the model trained without time parameters. We believe this discrepancy is caused by the implementation of time pressure as a global factor, that is, the models have the same time constraints for a full game and these parameters dictate what is considered a 'good' move based on the extent of the model's ability. This leads the consideration of a 'good' move to be moves that are successful at the final level of the game, where line clears of any type are worth more points. Even highly skilled players, who primarily pursue the score based strategy, shift to the lines based strategy in a bid for survival (Sibert & Gray,2018) at the end of a game. With global time parameters, a move made at level 1 would be given the same score as a move made at level 15, even though at level 15 the time constraints might make that move impossible. Gifting the models with an awareness of the current level might moderate the other features and produce time based models more representative of human behavior.

Conclusions

The deliberate practice framework (Ericsson, Krampe, & Tesch-Römer,1993) identifies practice as the single best predictor of a player's eventual level of expertise. Though the exact definition of *deliberate practice* has proven difficult to pin down, it is most often agreed to be effortful execution of the target task at the very edge of the player's skill. In this way, the current implementation of the model time constraints as global parameters may be forcing the model training into a kind of deliberate practice, and may be forcing our models to use strategies at the highest speed levels that humans would

Skill Level	1 Line	2 Line	3 Line	4 Line
Extreme Expert	34.42% (6.42)	34.48% (5.68)	21.64% (5.34)	9.46% (8.77)
Expert	24.38% (4.13)	40.75% (6.26)	24.42% (5.49)	10.45% (5.10)
Advanced	88.49% (3.66)	10.33% (2.75)	1.18% (1.63)	0% (0)
Intermediate	22.79% (6.49)	34.33% (16.01)	19.24% (9.79)	23.64% (14.56)
Proficient	91.36% (7.17)	8.64% (7.17)	0% (0)	0% (0)
Beginner	88.33% (7.18)	11.19% (7.15)	0.48% (1.51)	0% (0)
Novice	93.28% (8.81)	6.06% (8.31)	0.67% (2.11)	0% (0)
Extreme Novice	90.88% (10.94)	7.45% (8.85)	1.67% (5.27)	0% (0)

Table 6: Behavior of models with *Trained Time Parameters*. Behavior is measured by the average percentage of total lines that are cleared with each line clear type (1, 2, 3, or 4). Standard deviations are shown in parentheses. For all test games, the game length was unlimited but dependent on the model's speed.

not. That is, the (Sibert & Gray,2018) observations show that, while experts mostly favor a multi-line strategy, when playing at the limits of their expertise, humans revert to strategies that favor single-line clears. Our models, in contrast, are more dogged. Once they acquire a strategy they keep with it to the very end. By only practicing, or training (in the model case), under high time constraints, a player may fail to learn or master the alternate strategies that are more successful at earlier levels.

The models provide only a rough approximation of human behavior, but the observed contrasts between them and human players helps to shed light on the relationship between the time-based game constraints and the strategies employed by players. A safe, survival based strategy is best when the game is hard (because of low player skill or high speeds), but true expert players are able to flexibly switch strategies to best suit the current game state. A single, unchanging strategy may be functional in complex, dynamic task environments, but is unlikely to allow for the highest levels of performance.

Acknowledgments

The work was supported, in part, by grant N000141712943 to Wayne D. Gray from the Office of Naval Research, Dr. Ray Perez, Project Officer. The time parameters were originally codified by Mitchell Mellone as a senior thesis project.

References

Belchior, P., Marsiske, M., Sisco, S. M., Yam, A., Bavelier, D., Ball, K., et al. (2013). Video game training to improve selective visual attention in older adults. *Computers in Human Behavior*, *29*(4), 1318 - 1324.

Destefano, M., Lindstedt, J. K., & Gray, W. D. (2011). Use of complementary actions decreases with expertise. In L. Carlson, C. Hölscher, & T. Shipley (Eds.), *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (p. 2709-2014). Austin, TX: Cognitive Science Society.

Ericsson, K. A., Krampe, R. T., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological review*, *100*(3), 363–406. Available from http://doi.apa.org/psycinfo/1993-40718-001

Fahey, C. P. (2015, 09). *Tetris AI.* Available from http://www.colinfahey.com/tetris/ ([Online; accessed 2015-Jan-30])

Gray, W. D. (2017). Games-XP: Action Games as Cognitive Science Paradigm. *Topics in Cognitive Science*, *9*(2), 289–307.

Holmes, E. A., James, E. L., Coode-Bate, T., & Deeprose, C. (2009, 01). Can playing the computer game "Tetris" reduce the build-up of flashbacks for trauma? A proposal from cognitive science. *PLoS ONE*, *4*(1), e4153. Available from http://dx.doi.org/10.1371%2Fjournal.pone.0004153

Kirsh, D., & Maglio, P. (1994). On Distinguishing Epistemic from Pragmatic Action. *Cognitive Science*, *18*(4), 513–549. Available from http://doi.wiley.com/10.1207/s15516709cog1804{_}1

Lindstedt, J. K., & Gray, W. D. (2015). Meta-T: Tetris as an experimental paradigm for cognitive skills research. *Behavior Research Methods*, *47*(4), 945-965. Available from http://link.springer.com/10.3758/s13428-014-0547-y

Lindstedt, J. K., & Gray, W. D. (2019). Distinguishing experts from novices by the mind's hand and mind's eye. *Cognitive Psychology*, *109*, 1 - 25. Available from http://www.sciencedirect.com/science/article/pii/S0010028518300756

Nash, P., & Shaffer, D. W. (2011). Mentor modeling: the internalization of modeled professional thinking in an epistemic game. *Journal of Computer Assisted Learning*, *27*(2), 173-189.

Proctor, M. D., Bauer, M., & Lucario, T. (2007). Helicopter flight training through serious aviation gaming. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, *4*(3), 277-294. Available from http://dms.sagepub.com/content/4/3/277.abstract

Rapp, A., Cena, F., Gena, C., Marcengo, A., & Console, L. (2016). Using game mechanics for field evaluation of prototype social applications: a novel methodology. *Behaviour & Information Technology*, *35*(3), 184-195. Available from http://dx.doi.org/10.1080/0144929X.2015.1046931

Robertson, J. (2003, Nov). A brief history of Tetris AI. Available from http://liquidnarrative.csc.ncsu.edu/index.php/the-news/the-ln-blog/145-a-brief-history-of-tetris-ai?tmpl=component&print=1&layout=default&page=

Sibert, C., & Gray, W. D. (2018). The Tortoise and the Hare: Understanding the influence of sequence length and variability on decision making in skilled performance. *Computational Brain & Behavior*, *1*(3-4), 215–227.

Sibert, C., Gray, W. D., & Lindstedt, J. K. (2017). Interrogating feature learning models to discover insights into the development of human expertise in a real-time, dynamic decision-making task. *Topics in Cognitive Science*, *9*, 1-21. Available from http://dx.doi.org/10.1111/tops.12225

Szita, I., & Lorincz, A. (2006). Learning Tetris using the noisy cross-entropy method. *Neural Computation*, *18*(12), 2936-2941.

Thiery, C., & Scherrer, B. (2009a). Building controllers for Tetris. *ICGA Journal*, *32*, 3–11. Available from http://hal.archives-ouvertes.fr/inria-00418954/

Thiery, C., & Scherrer, B. (2009b). Improvements on learning Tetris with cross-entropy. *ICGA Journal*, *32*(1), 23-33.