Less is More: Additional Information Leads to Lower Performance in Tetris Models

Catherine Sibert (siberc@rpi.edu)
Cognitive Science, Rensselear Polytechnic Institute, 110 8th Street
Troy, NY 12180 USA

Jacob Speicher (speicj@rpi.edu)
Cognitive Science, Rensselear Polytechnic Institute, 110 8th Street
Troy, NY 12180 USA

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

Abstract

Expert performers in complex tasks synthesize a wide variety of information to select the optimal choice at each decision point. For the task of Tetris, the synthesis includes information about the "next" piece in addition to the configuration of pieces currently on the board. While simple models of Tetris are capable of behavior similar to high level human players most (to reduce the combinatorial explosion in computation time) are only aware of the active piece and its possible placement positions. To explore how additional information contributes to expertise, when placing the current 'on board' piece, our model also considers placements for the "next piece" (visable to humans in the Preview Box). Though we expected this additional information to result in higher performance, we instead observed a drop in performance, and a shift in behavior away from common human patterns. These results suggest that human experts are not incorporating the additional piece information into their current decision. We speculate about the role of next piece information for expert level players.

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

Introduction

Complex task environments are, almost by definition, difficult to master and, by extension, difficult to study. In this work we focus on the complex task environment created by the dynamic, decision-making game of Tetris, which we see as the *poster child* for human studies of predictive processing (Clark,2013;Engstrom et al.,2018;Rao & Ballard,1999). Although we have made considerable progress in understanding Tetris play in our laboratory (Lindstedt & Gray,2019), human play represents a confounding of various human limitations that may well be impossible to disentangle *in vivo*. This tangle has led us to machine models of Tetris (Fahey,2015;Gabillon, Ghavamzadeh, & Scherrer,2013;Szita & Lorincz,2006) where we have focused on understanding how the configural properties of the Tetris board can be interpreted by machine models as good placements or bad placements for the currently falling zoid (i.e., Tetris piece) (Sibert, Gray, & Lindstedt,2017;Sibert & Gray,2018).

The attentive reader will note that we initiated the preceding paragraph by alluding to human predictive processing but ended that paragraph by focusing on defining characteristics of good or bad placement decisions by using machine models. This shift is possible as, unlike humans, our models do not actually rotate, transpose, or drop pieces; rather, as in an episode of Star Trek, they simply beam the piece to the desired location. This trick neatly disentangles the *where to place it* decision from the *how to get it there* one (which are the concern for Tetris studies of Predictive Processing (Lindstedt & Gray,2019)).

The current work complicates our models by observing that the classic, Nintendo Entertainment Systems (NES) version of Tetris (which is the version used in the annual *Classic Tetris World Championships, CTWC*) as played by humans, always provides the "next" zoid in a Preview Box (see Figure 1). As the goals of the machine modeling community differ from ours, no prior machine model of Tetris play uses that information.[1] Hence, our current work explores two-piece placement decisions in an attempt to determine whether and, if so, how, attempts to optimize the current placement with respect to the next placement improves the game.

A Very Brief History of Games in Research

Gray (2017) distinguishes among three ways in which modern computer games have been used in psychological research. *Gamification* represents the attempt to use features of game play for more serious work such as gamifying a social media field trial (Rapp, Cena, Gena, Marcengo, & Console,2016), modeling "professional thinking" (Nash & Shaffer,2011), or teaching helicopter flight skills (Proctor, Bauer, & Lucario,2007).

Games as Treatment Conditions represents attempts to use game play as a means of changing some aspect of human behavior, health, intelligence, and so on. Examples involving Tetris include its use to reduce "flashbacks" associated with Posttraumatic Stress Disorder (PTSD) (Holmes, James, Coode-Bate, & Deeprose,2009) or as a placebo control in a

[1] While we do not know for sure why the machine modeling community does not use that information, we do note that doing so extends the search space for moves from approximately 23 placements (between 9 and 34, depending on the zoid) up to 34^{34} placements. The addition of more complex move generation functions that allow zoids to be navigated underneath other zoids further increases the number of placements to be considered at each decision point.

study of the utility of games (Belchior et al.,2013) in expanding older adult's useful field of view (UFOV).

Game-XP refers to the use of game play itself as an experimental or quasi-experimental paradigm. The earliest example of using Tetris for Game-XP (that we are aware of) was for exploring the concept of epistemic or complementary action (Kirsh & Maglio,1994;Destefano, Lindstedt, & Gray,2011). Of course, our past work (cited earlier) as well as the work presented in this paper provide other examples of the use of Tetris for these purposes; that is, an experimental paradigm which we use to seek insights into the low level mechanisms that contribute to skilled performance in dynamic tasks.

Tetris the Task

During a game of Tetris, players navigate a series of pieces, called "zoids", as they fall from the top of the screen into a pile at the bottom of the screen. When a row within the pile becomes full (all ten cells contain a part of a placed zoid) it vanishes, lowering the pile and earning points for the player. More points are earned if more lines are cleared simultaneously, with up to four lines able to be cleared in a single move. The game ends when the pile reaches the top of the screen. A game in progress can be seen in Figure 1.

Though the basic task is simple to understand, game difficulty increases as the player plays. The player is limited in the actions that they can take to move a zoid: zoids can be translated one cell left and right, or rotated 90 or 180 degrees (depending on the zoid) using a single button press, and a complete movement usually requires several button presses. As the game progresses, the pieces fall more quickly, meaning that players must make placement choices and navigate the zoids in increasingly short time periods. At the start of the game, it takes 16 seconds for a zoid to fall from the top of the screen to the bottom. At level 29 (considered by top players to be the "kill screen", and the highest playable level) pieces fall in a third of a second.

In addition to managing the ever increasing game speed, players must weigh the risks and benefits of making different types of line clears. Clearing a single line is fairly simple to do, and most low level players focus on clearing one line at a time in order to prolong the game as long as possible. However, from a purely points based perspective, this is a poor strategy. Setting up and executing a single 4 line clear (or a Tetris) is worth 7.5 times as many points as clearing a single line four times. Because the speed component of Tetris will eventually force any game to end, most high level players adopt a strategy that emphasizes making 4 line clears early and often.

Tetris is a complex, dynamic task in that the task state is constantly changing independent of any action taken by the player. Pieces will fall even if the player presses no buttons. In this kind of environment, taking no action requires a decision to do nothing, and the series of decisions made by the player at each zoid placement result in the final game score. Performance in Tetris is judged by this final game score, but because of the constant and varied game state, it is difficult to know what contributes to that performance.

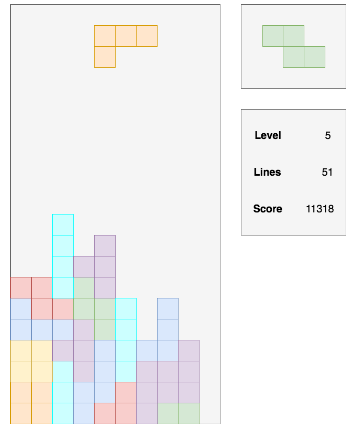

Figure 1: A Tetris game in progress. The active piece, the orange "L" is currently being placed by the player on the main game screen. The player also has access to score information, in the lower right-hand box, and one upcoming piece, the green "Z" in the upper right-hand box.

Tetris Models

Human play of Tetris is a test of human limits in dynamic decision-making and action and provides an excellent example of predictive processing (Clark,2013;Engstrom et al.,2018;Rao & Ballard,1999). Deciding where best to place a zoid becomes increasingly time-limited as the rate of fall increases. Likewise, the time available for the player to move the zoid to the chosen location also decreases.

Despite the complexity of human behavior in a task like Tetris, it is possible to build simple models capable of high level performance. Most of these come from the machine learning community, where Tetris is a popular test case for feature search algorithms.

These models function by defining a set of board features (selected by the researcher) that are believed to be important when making placement decisions. An early and commonly used set of features, defined by Dellacherie (Fahey,2015), is provided in Table 1. These are the features that we use to build the models used in this study.[2]

The models play Tetris by assigning each feature a numerical weight, the magnitude and sign of the weight indicates how desirable or undesirable a particular feature is. For a given move placement, the model generates all possible zoid positions and evaluates each one by multiplying the weight of each feature against the value produced by that move. These feature scores are added together to form a total move score, and the model ultimately selects and executes the placement with the highest move score.

The feature weights remain constant during a game, so the challenge of building a high performing model lies in choos-

[2]See Sibert et al. (2017) and Sibert and Gray (2018) for a fuller story.

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

ing an optimal set of weights from a large search space. We employ the Cross-Entropy Reinforcement Learning method proposed by Szita and Lorincz (2006) and modified by Thiery and Scherrer (2009a,2009b).

Making Models More Human-Like

While traditional machine learning models are capable of high level performance, several important changes are made to the task environment that encourages models to adopt unhuman-like strategies in order to do well.

First, models tend to be unconstrained by the time pressure that is a major component of human gameplay. Second, models are reinforced for line clearing behavior, which encourages a strategy that primarily clears single lines. This is a viable strategy in the very long term (as, for example, used in Sibert and Gray, 2018), but only yields mediocre performance during the restricted time scale of a human game. Third, humans have access to additional information, like the upcoming zoid, that is not incorporated into the model decision making process.

Efforts have been made to explore how these environmental factors impact behavior. When trained on games of restricted length, models reinforced for line clearing behavior performed at a low-scoring but stable score level, while models reinforced for score reached higher scores but not as consistently. At their best, the score-reinforced models performed at the level of high performing student players, while line-reinforced models performed closer to intermediate level student players (Sibert et al.,2017).

This behavioral and strategy split was also observed in the absence of a reinforcement criteria when comparing models trained on restricted games against models trained on games of unrestricted length. The best long-game models far outperformed the short-game models by clearing single lines far beyond the point that the human game becomes unplayable. When restricted to human-length games, models adopted the higher scoring strategy of executing multiple line clears (worth far more points than a series of single line clears) early and often (Sibert & Gray,2018).

Whereas these prior studies focused on addressing the time pressure and reinforcement criteria aspects of the human Tetris environment, the current study aims to look at a third major difference between models and humans: humans have access to upcoming zoid information that models lack. Initial eye-tracking explorations (e.g., (Gray, Hope, Lindstedt, & Sangster,2015)) into human behavior show increased fixations on the next zoid box for higher level players, suggesting that this information is an important aspect of advanced play. Prior studies adjusting the model game environment led to performance levels, but only to the level of advanced human players (when equating for game length), suggesting that aspects of truly expert gameplay are still beyond the models. We hypothesized that allowing the models to consider the upcoming zoid when making placement decisions would result in higher performance. This ability to do *Two-Piece lookaheads*, thereby optimizing placements for 2 zoids rather than just 1, should also promote an increase in multiple line clears, as the models will have an increased capacity to plan ahead.

Methods

Model Development

Using the Dellacherie feature set (described in Table 1) and the cross-entropy reinforcement learning (CERL) method, we developed two models, a *One-Piece Lookahead* model and a *Two-Piece Lookahead* model.

Both models were trained on short games (a maximum of 525 zoids[3]) and were reinforced for high score. Both of these environmental conditions have encouraged more human-like behavior in our previous modeling studies. The models were developed using the same iterative CERL method (described in more detail in (Sibert et al.,2017), which can be summarized as a process that generates a set of candidate models with each model playing a single game of Tetris. The highest performing models are averaged together to create the starting point for generating the next set of candidate models. At each generation, 100 candidate models are tested, and the 10 best models were used to create the averaged model. In previous studies, this process was repeated 80 times, but here we implemented a halting condition: when the variance of the feature weights in the top performing models reached an acceptable threshold (below 0.01), the model was considered to have reached conversion and the search ended. Models tended to converge between 30 and 40 generations, greatly reducing the search time required for development.

The critical difference between the models was the amount

[3]Note that although these games are short for Machine Models, for the 300+ humans who have played an hour or more of Tetris in our laboratory, 525 zoids is the most zoids ever played by any human.

of lookahead information incorporated into the decision-making process. The *One Piece* models only information about one zoid at a time, and have no knowledge about what might be coming next in the sequence. It generates all possible placements for that zoid, and each placement is given a score by combining feature weights with the value of those features that result from the placement (i.e., if the placement creates a new pit, the score for that placement will change by the weight of the pit feature, and so on). At each placement, the model selects and executes the highest scoring move.

Two Piece models, by contrast, have access to the active zoid as well as the next zoid. Rather than calculate a score for each zoid placement, the Two Piece model evaluates the score for each pair of moves (adding together the score for the first and second move). This might cause the model to choose the second or third best move for the first zoid in order to allow a much higher scoring placement for the second zoid. Adding this capability greatly increases training time, not just in the greater computation time required for each game, but also by increasing the number of generations for convergence from approximately 30, for the One Piece models, to over 50 for the Two Piece models. However, we expected that this initial training cost would be compensated by better model performance.

Model Testing

Both models were tested using performance metrics (measured by game score) and behavior metrics (measured by types of line clears executed). Though only two models were developed, we had a total of four testing conditions. Because Lookahead was an environmental condition, it could be turned on or off for a developed model during testing. All tests were conducted on both models in both conditions: One Piece model with One Piece tests (same as training), One Piece model with Two Piece tests (alternate test condition), Two Piece model with One Piece tests (alternate test condition), and Two Piece model with Two Piece tests (same as training).

For the performance test, models were run through ten Tetris games. The zoid sequences of these games were generated using one set of ten random seeds (111, 222, 333, and on to 101010) to ensure that the models were tested in a controlled and equal environment.[4] Each model plays through this set of 10 games twice, once with only the current zoid (One Piece lookahead), and once with the current and next zoid (Two Piece lookahead).

Model performance was measured in three ways: the high score, the mean score, and the criterion score. The high score is the best score achieved on any game, and the mean is the average score of all ten games. The criterion score is a metric developed for evaluating human player skill (Lindstedt & Gray,2019), and is calculated by averaging the scores of the top four games in a testing period (for human players, this testing period is one hour, for models it is the set of ten games). The criterion score reduces the influence of a single unusually high or unusually low score on the overall measurement of player skill.

Model behavior was evaluated using the same ten test games, but rather than looking at a numerical score, the models were measured by the proportion of line clear types made during the game. Of all lines cleared during a game, some percentage are cleared using single line clears, some by two line clears, three line clears, and four line clears. The pattern of line clear types is a good measure of how the model behaves, as truly machine models tend to clear predominantly single lines, and high level humans try to emphasize 4 line clears.

Results

Table 2 shows the performance results for models trained in the One-Piece condition, tested in both the One-Piece and Two-Piece conditions. All scores were higher during testing with One-Piece lookahead (the same as the training condition), though the scaling scoring system of Tetris makes the score differences look larger than the actual performance differences that they reflect (line clears of all types are worth more points when executed at higher levels, meaning the rate of score accumulation increases as the game progresses).

Comparing the "native" training positions in the one-piece model (left column in Table 2) versus the native training position of the two-piece model (right column in Table 3) shows that the two-piece model performs worse than the one-piece model. Perhaps more surprising is the massive drop in performance when the Two Piece model is tested in the One Piece condition (left column in Table 3). These extremely low scores (left column in Table 3) represent very few line clears and in several games, these Two Piece models made no points at all.

Table 2: One-piece lookahead models tested in either the one or two piece lookahead condition

Testing Condition	One Piece	Two Piece
High Score	406000	200560
Mean Score	203766	161472
Criterion Score	323740	187965

Table 3: Two-piece lookahead models tested in either the one and two piece lookahead condition

Testing Condition	One Piece	Two Piece
High Score	1600	326180
Mean Score	220	132818
Criterion Score	540	229565

[4]See the discussion in Sibert & Gray, 2018, of the surprising differences in the variability of model performance across different random seeds.

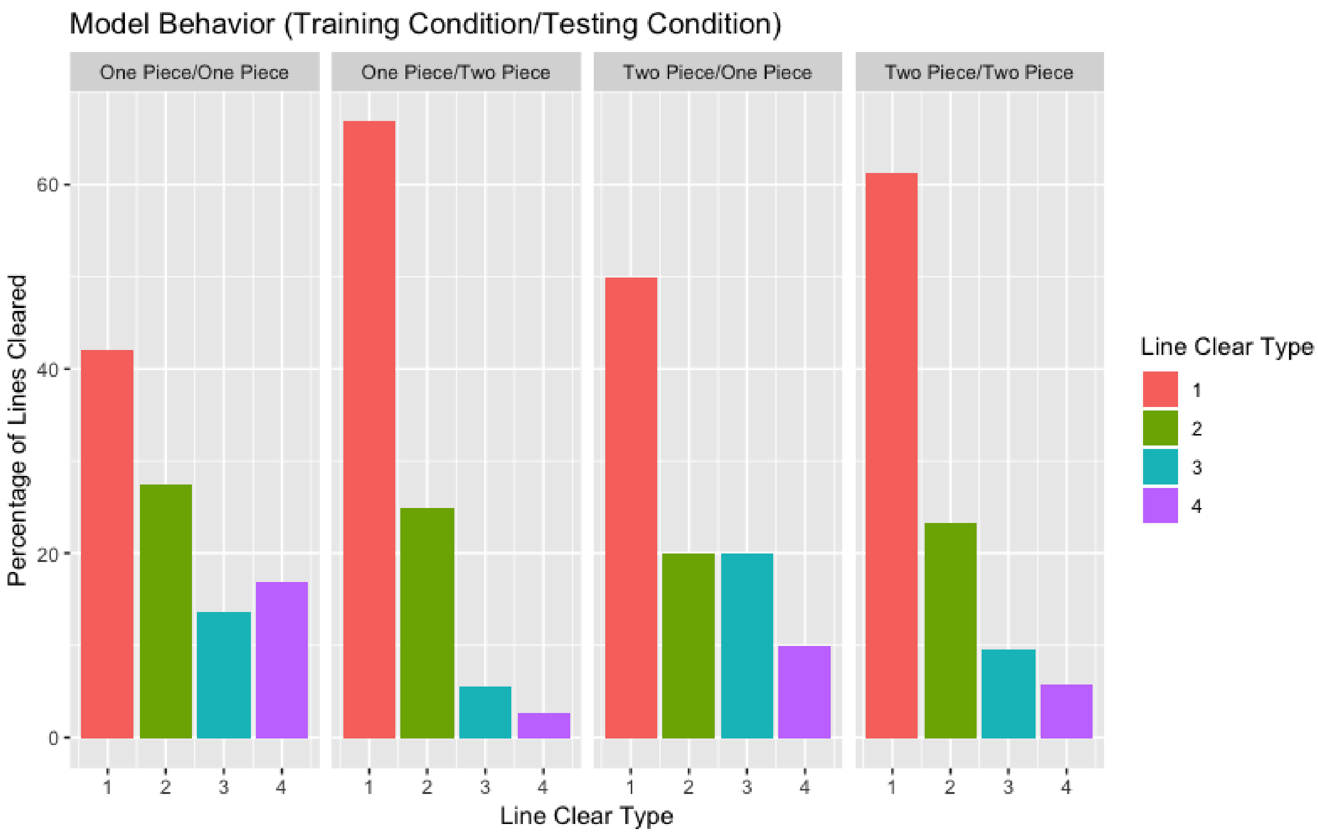

Figure 2: The behavior of models as represented by the proportion of each type of line clear made. Each set of bars represents a training/testing condition pair.

Figure 2 shows the percentage of each type of line clear averaged through the ten test games. The percentage of line clear type indicates the proportion of lines cleared using each type of clear to the total lines cleared during a game. Typical machine performance is characterized by a very high percentage of single line clears, and steadily lower percentages of each type of multiple line clear. High level human players have a more U-shaped pattern, with the highest percentage of lines being from 4 line clears, followed by 1 line clears and two line clears, with the fewest lines from 3 line clears.

The behavior pattern produced by the One Piece/One Piece condition (one-piece lookahead model playing one-piece lookahead games) is not quite the same shape as human experts, but represents a significant behavior shift toward human-like behavior. The behavior pattern has a U-shape that is similar to good human players, with more 4-line clears than 3-line clears.

Both models trained in the Two Piece condition show the much more typical machine pattern, with high percentages of single line clears, and progressively lower percentages of higher order line clears. The results from the Two Piece/One Piece model are representative of significantly fewer lines cleared, and are not as robust as the results from the other conditions.

The most unexpected result comes when the One Piece model is tested in the Two Piece condition. This resulted in lower performance, but also in a significant behavioral shift away from a humanlike pattern and toward the machine pattern.

Discussion

We expected that providing models with more information would improve model performance, and encourage a behavior pattern with higher levels of long term planning. Instead, we found that more zoid information led to lower model performance, and less human-like behavior. While there were not huge differences in performance between the one piece model tested with one piece lookahead and the two piece model tested with two piece lookahead (compare Tables 2 and 3), there were significant performance drops when a one piece model was tested with two piece lookahead. Removing two piece lookahead from the two piece model led to an even larger drop in performance, where the models were barely able to clear any lines.

The changes in the patterns of model behavior were also unexpected. Successful human players display a distinct pattern of line clear types, prioritizing four line clears. Single line clears are the next most frequent, followed by double line clears, and very low frequencies of three line clears. Many of our previous modeling efforts have tried to encourage models to follow similar patterns. During these experiments (Sibert & Gray,2018), we found that the model's behavior was determined by the training condition, and the pattern of line clears would persist in alternate testing conditions.

Adding two-piece lookahead to a high performing one-piece model (see the second set of bars in Figure 2) caused a large shift in model behavior, changing from the u-shaped pattern similar to high level humans to a sloped pattern consisting of primarily single line clears and very few four line clears. Both two-piece models displayed similar behavior patterns, but because the two-piece model tested using one-piece lookahead cleared almost no lines, few conclusions can be drawn from its pattern of line clear types.

Looking at the episode-level behavior of the models, we believe that the drop in performance and change in behavior patterns is caused by the model constantly making a suboptimal decision about the current zoid placement in service to a better placement for the upcoming zoid. However, once the upcoming zoid becomes the current zoid, there is a new upcoming zoid that may change the best placement. That is, the model is always planning to make a better move, but rarely follows through. Though we thought having additional zoid information would lead to the model making better moves, the short term optimization at the level of one or two pieces came at the cost of the generalization offered by the one piece model.

Based on these models, we can guess that if humans are incorporating upcoming zoid information into their placement decisions, it is not by making choices to facilitate specific placements for the next zoid. We do have some evidence (Gray et al.,2015) that players, particularly expert players, frequently fixate the next zoid box as they play, strongly suggesting that this information is being used in some way.

We theorize that a Tetris placement involves two stages: the decision phase and the movement phase. At low speeds, movement can be initiated before a final decision is made, but at high levels, speed is the limiting factor in performance, and placement decisions must be extremely rapid in order to maximize the available movement time. Rather than make decisions about the current zoid when it appears on the game board, we now interpret our model results as suggesting that expert players offload the decision phase to the previous episode, making a decision about the zoid placement while the zoid is still in the Next box. Once the piece appears on the Board, the player can initiate the movement phase for that zoid (now the "current" zoid in our terminology) while simultaneously initiating the decision phase for the upcoming zoid (i.e., the one that is now in the Preview box). Hence, expert players do not try to optimize two-piece placements but do try to optimize one-piece placements. The extra time for making these one-piece optimization is especially important at higher levels of Tetris; whereas maximum drop time is 16s at level 0, that decreases to 2s by level 9, to 1s by level 16, and to 0.67s at level 19. This explanation is compatible with component 2 of Lindstedt & Gray's (2019) Principal Component Analysis which suggested that better players make their placement decisions prior to moving the zoid.

We have not yet formally tested this hypothesis, but some expert Tetris players have already performed an informal experiment on their own. At 2018's Classic Tetris World Championship, 16 players engaged in a novel, "no-next box" tournament which began play at level 18 (where it takes 1s for a zoid to drop from top-to-bottom). Although most of these players had secured a slot in the next day's playoffs for the Classic Tetris tournament, only one player scored over 30,000 points in this no-next box match with a few players scoring no points at all. The behavior of the players, usually characterized by high percentages of four line clears, was almost entirely single line clears. No four line clears were executed during the entire no-next box tournament.

Overall, the results of these models suggest that in complex, dynamic tasks, where there is rarely a single objectively correct action, the most successful behavior pattern must be general. Adding additional information serves to make model behavior more specific, which may be more optimal for a single decision point, but will be less successful over a long series of decisions. Additional zoid information, then, is likely not used to modulate individual zoid placement decisions. Instead, observation of expert players suggests that it is used to shift the time demands of a placement decision and allow more time to execute movements, making gameplay possible at very high levels. Further experiments may be able to explore how upcoming zoid information is incorporated by high level players, but the more machine-like approach of systematically exploring all options is clearly not the answer.

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.

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.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, *36*(3), 181-204.

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.

Engstrom, J., Bargman, J., Nilsson, D., Seppelt, B., Markkula, G., Piccinini, G. B., et al. (2018). Great expectations: a predictive processing account of automobile driving. *Theoretical Issues in Ergonomics Science*, *19*(2), 156-194.

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

Gabillon, V., Ghavamzadeh, M., & Scherrer, B. (2013). Approximate dynamic programming finally performs well in the game of Tetris. In *Advances in neural information processing systems* (Vol. 26, pp. 1754–1762). Available from http://media.nips.cc/nipsbooks/nipspapers/paper_files/nips26/881.pdf

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

Gray, W. D., Hope, R. M., Lindstedt, J. K., & Sangster, M.-D. (2015). Gaze transitions as clues to expert-novice strategy differences in a dynamic video game. In *ETVIS 2015 - First Workshop on Eye Tracking and Visualization.* IEEE VIS.

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. (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

Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*(1), 79-87.

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

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.