Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment

A Generative Computational Model for Human Hide and Seek Behavior

Andrew Cenkner[1] and **Vadim Bulitko**[1] and **Marcia Spetch**[2]
Departments of Computer Science[1] and Psychology[2]
University of Alberta
Edmonton, AB, T6G 2E8, CANADA
cenkner | bulitko | mspetch @ ualberta.ca

Abstract

Hiding and seeking is a cognitive ability frequently demonstrated by humans in both real life and video games. We use machine learning to automatically construct the first computational model of hide/seek behavior in adult humans in a video game like setting. The model is then run generatively in a novel environment and its behavior is found indistinguishable from actual human behavior by a panel of human judges. In doing so the artificial intelligence agent using the model appears to have passed a version of the Turing test for hiding and seeking.

Introduction

Hiding and seeking is a cognitive ability of humans studied in psychology (Talbot et al. 2009). Many video games invoke this ability in some form. For instance, competitive on-line first-person shooters such as *Counter-strike: Source* (Valve Software 2008) have players searching for members of the opposing team (e.g., snipers). Role-playing games such as *Borderlands* (Gearbox Software 2009) or *Fallout: New Vegas* (Bethesda Softworks 2011) encourage the player to explore the environment and reward them with weapons, side-quests and information on the story and environment.

To support these hide and seek activities, game developers face several challenges. First, level designers need to place loot (either statically or procedurally) in locations that would reward both casual and hardcore players. Deciding on which kinds of items to place at which locations can be made easier and more efficient by predicting, at the development stage, where the players will search and how their search patterns will be different depending on the player type (e.g., from a casual player to a completionist).

Second, level designers can also benefit from knowing *a priori* where the players will be looking for other players (e.g., in *Counter-strike: Source*) or other player's units (e.g., in *StarCraft 2* (Blizzard Entertainment 2010)). Finally, artificial intelligence (AI) developers need to develop non-playable characters that search for the player in a compelling, non-cheating way. In *Counter-strike: Source* this was accomplished by hard-coding search movement patterns from developer's experience (Booth 2004). Other games attempt to generate realistic looking seeking behavior by obfuscating AI's omniscient knowledge of the player's location. Both approaches are error-prone and require extensive tuning and testing.

Beyond video games, understanding hiding and seeking is valuable to law-enforcement agencies (e.g., predicting hiding spots for illegal substances) and the military (e.g., predicting locations of stashes of weapons, improvised explosive devices, etc.) Finally, if hiding and seeking are viewed as fundamental cognitive abilities then mastering them via a computer program/model may bring us closer to building strong Artificial Intelligence.

The rest of the paper is organized as follows. We formalize the problem and describe our performance measures in the next section. We then review the existing related work and argue that it is insufficient to solve the problem at hand. Our own approach is presented next, followed by an empirical evaluation. We then discuss the results, consider directions for future work and conclude the paper.

Problem Formulation

To be an effective tool to game developers an AI agent must be generative, so it can produce results on its own, and portable so it can produce results in new environments. In this paper we consider the problem of building a generative and portable AI agent capable of human-level hiding and seeking in a realistic three-dimensional virtual environment. While hiding or seeking, a human or an AI agent selects from a finite pre-determined set of locations. They do so by navigating in the environment, pointing at a location and indicating whether they wish to hide or seek there (Figure 1). We say that a generative computational model of hide/seek behavior is successful if it can be used within an AI agent to pass the following version of Turing test.

A panel of human judges observes an equal mixture of human and AI agents moving about in an environment and picking locations to hide or seek. The environment is novel to the agents insomuch as they have not seen the environment beforehand. After each judge views an agent they decide if the agent is in fact a human or an AI program. Their binary votes are tallied and a detection rate is averaged over a number of demonstrated hide/seek behaviors and a number of judges. An AI agent is said to have passed the Turing

Figure 1: A 3D game-like environment with possible hide and seek locations shown as tiles on the floor.

test if the detection rate is statistically indistinguishable from chance. We then say that the hide/seek model that the agent used is validated.

Related Work

Psychologists have long researched animal hiding and seeking behavior, in particular with respect to food caches (Clayton, Dally, and Emery 2007; Clayton, Emery, and Dickinson 2006; Dally, Clayton, and Emery 2006). There have also been a number of studies on hiding and seeking behavior in children (Cornell and Heth 1986; Cornell et al. 1987). None of these studies is directly applicable to our problem as they do not consider human adults.

A more recent study (Talbot et al. 2009) did consider hiding and seeking behavior of adult humans in a game-like virtual environment. Unfortunately, it does not solve our problem for two reasons. First, the environment used in the study — a small square room devoid of any features, with a total of nine hiding/seeking locations — was too simplistic to be realistic. Second, the behavior exhibited by humans was analyzed at a coarse aggregate level (e.g., mean distance traveled to the first location from the room entrance). No attempt to computationally generate hiding/seeking behavior was made.

Some work has been done in the area of predicting player locations in first-person shooters (Hladky and Bulitko 2008; Darken and Anderegg 2008). The objective of that research was to predict human behavior in the task of hiding oneself and seeking for other players. In contrast our problem deals with the task of participants hiding and seeking for arbitrary objects. Also there have been no Turing test verified hide and seek models.

Proposed Approach

We set out to develop a computational model capable of effectively generating human-like hide and seek behaviors. The desired properties were as follows. First, we wanted our model to be developed automatically via data-mining previously collected recordings of human hide/seek behavior (we call this stage "training"). Second, we wanted the model to be portable — that is, applicable in an environment it has never seen before — without any extensive annotation of such a novel environment. Third, the model was to be stochastic in nature and capable of producing many human-like behaviors in the same environment.

We decided to break the hide/seek tasks into two specific behaviors: selecting hide/seek locations and producing a realistic trajectory between them. For each behavior, we created both a simple and an advanced strategy. We wanted to compare the advanced strategies to the simple strategies to demonstrate that our version of the Turing test is meaningful and cannot be passed trivially.

Location Selection Strategies

For the simple location selection strategy (L1) we used uniform random selection and for the advanced location selection strategy (L2) we used a stochastic selection from a previously created distribution. An agent would use its strategy (L1 or L2) to repeatedly select locations as long as desired.

Strategy L1: random selection. Locations are selected uniformly randomly from a set of all possible locations. The selections are made with replacement for the seek task and without replacement for the hide task. This was because, in the seek task, humans were found to re-visit locations frequently, but not in the hide task.

Strategy L2: data-driven selection. First, a set of probability density functions (PDFs) is created offline from a pool of sample human location selections. The goal of all our strategies is to be portable, so we had to come up with a PDF that could scale to new environments. For example, if we created a distance PDF from samples in a small environment, the PDF would never select a distance greater than the largest possible in that environment. This could be a problem if we wanted to use the PDF in a large outdoor environment. Our solution was to have each PDF model a rankable feature (able to be sorted from highest to lowest) of the sample selections. For example distance can be ranked from shortest to longest. Instead of each sample contributing its absolute value to the PDF it contributes its relative value in the ranking of all possible selections. In this way the PDF scales to rooms with any range of values for that feature.

Next, a joint probability distribution (JPD) is created online from the set of PDFs. The PDF features are selected in an attempt to be statistically independent so that the JPD is their product. When the agent is placed in the room it computes the JPD by taking the product of all the PDFs at each selection location.

Finally the selection is drawn stochastically using the JDF to weigh all possible selection location.

Movement Strategies

Once a vector of locations was picked using L1 or L2, a movement strategy was used to navigate between them. For

	Random (L1)	Data Driven (L2)
Spline (M1)	Agent 1 (A1)	Agent 2 (A2)
Data Driven (M2)	Agent 3 (A3)	Agent 4 (A4)

Table 1: Definition of agents with movement strategies on left and location selection strategies on top.

the simple movement strategy (M1) we used cubic spline interpolation between the desired locations. For the advanced movement strategy (M2) we searched a library of human trajectories and selected one that passes through the chosen locations.

Strategy M1: spline interpolation. Given a vector of locations, a cubic spline interpolation was used to construct a trajectory through them. If a segment between points a and b of the resulting trajectory intersected an obstacle in the environment then A* was used to construct a new valid trajectory between a and b and its middle point was inserted in the vector of locations between a and b. The spline fitting was then repeated for the new vector of locations. The process stopped when the resulting trajectory fit the environment. This strategy was complete in the sense that given enough additional locations inserted in the trajectory it would degenerate into A*. In other words if there was a trajectory between a and b this strategy would find one.

Strategy M2: data-driven trajectory shaping. Given a vector of locations $[l_1, \ldots, l_n]$ to traverse, we considered them sequentially. For each pair of locations (l_i, l_{i+1}), we translated, rotated and linearly scaled every recorded trajectory in the trajectory library so that it connected l_i and l_{i+1}. The quality of each such fit was determined by a linear combination of the amount of scaling required and the continuity at connection points. A stochastic selection from the highest quality trajectories was used for navigation.

Agents

To test the effectiveness of these strategies we created four agents. Each agent's model was composed of one strategy for the movement behavior and one strategy for the location selection behavior. The breakdown of the models used for each agent can be seen in Table 1.

Five pools of videos were created: one of each agent performing the hide/seek task and one of humans performing the hide/seek task. The judges were divided into four groups: one for each agent. The judges then viewed a mixture of videos from the human pool and from their agent pool only. This means a judge never saw videos of behavior produced by agents of different types. In this way our Turing test compares agents against humans and not directly to each other. Each judge received a score based on the number of videos they correctly labeled. This score was tied to the particular agent the judge was assigned, forming a list of scores for each agent. The agents were then compared to each other using the average of their list of scores.

Empirical Evaluation

We used the following implementation of the hide and seek tasks in our experiment. Each participant was briefed with a description of the task and trained on how to control a first-person avatar with the *Half-Life 2* (Valve Corporation 2004) mouse-and-keyboard controls in a small specifically designed training environment. The participants were then put in one of two virtual environments built with *Hammer* (Valve Coorporation 1996) and executed with *Source* engine (Valve Coorporation 2007) and asked to perform the hide and seek tasks. In the hide task the participant was asked to hide three objects and to "make your objects difficult for other people to find." In the seek task the participant was asked to pick locations until three previously hidden items were found. The seek task was limited to 1 minute, while the hide task was not time limited. Participants were free to move about the room, but had to wait a delay of 1 second between selecting locations. The selection locations were black tiles on the floor (Figure 1).

The two environments were: an office style room modeled after an existing laboratory and a simple rectangular room (Figure 2). The former contained 73 designated locations for hiding and seeking, shown as black floor tiles. The latter had 75 locations. Both environments had realistic lighting and office furniture.

In our experiment we pursued two objectives to demonstrate that the version of Turing test we used is indeed meaningful. The objectives are (i) showing that the task is cognitively rich enough that a simple AI agent would fail the test and (ii) that the human judges are given enough information to render an informed judgment. An example of a test that violates condition (i) would be "to sit in the chair" since a statically placed agent model would easily pass the test. An example of violating condition (ii) would be withholding the actual contents of a conversation from the judge in the original Turing test and, instead, showing them only a light when the agent is using their teletype.

We satisfied these conditions by presenting the judges with a video of a top-down view (Figure 2) of the agent (human or AI) moving about in the environment and selecting locations. We demonstrated that a non-trivial, manually designed AI agent was reliably distinguished from humans. This shows that the test is not passed trivially, and that the judges were given enough information to correctly label a trivial agent.

The study was carried out in three parts: data collection, model/agent training and judging.

Data Collection. The dataset we used to create our models was the collective recordings of 1071 human participants in virtual environments. Our participants were recruited from a first-year course in psychology.

As the subjects performed their tasks, their avatar's locations and orientations were recorded once per second. Additionally, their location selections were recorded as well, whenever they selected a tile. Overall, more than 5142 trajectories were recorded, each containing between 6 and 125 data points.

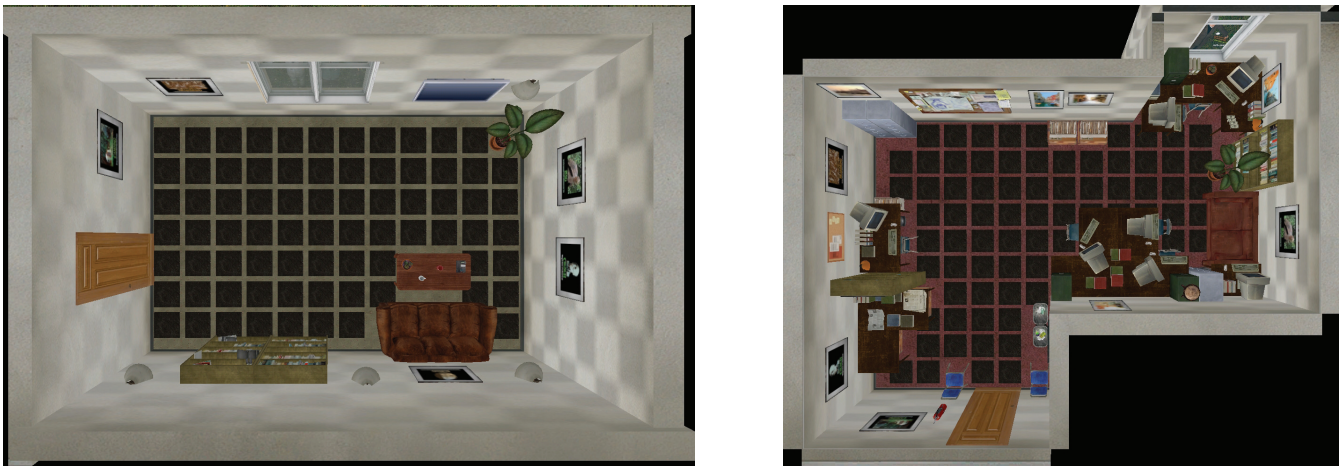

Figure 2: The two environments used in our study (a top-down view).

Model/Agent Training. Each agent first creates a list of location selections using either L1 or L2. In the hide task a list of 3 selections is made. In the seek task a list of selections is created that is long enough to ensure that the task times out before running out of locations.

We implemented the L1 strategy by drawing uniformly random tile ID numbers.

In our implementation of the L2 strategy we created 3 PDFs. The first PDF was how likely a tile was to be selected given its Euclidean distance from the last location. The second PDF was a tile's likelihood given angle rotation required. The angle rotation was calculated as the amount an agent would need to rotate to face the tile, standing on the last tile, with their back facing the second last tile. The angle was normalized between -$180°$ and $180°$. The third PDF was a tile's likelihood given the last time it was selected. For example a tile that was selected 2 selections ago would be less likely to be selected than a tile that had never been selected. Since all the values of these PDFs were rankable, as they were scalars, they were portable to new environments. When asked to make a tile selection the L2 strategy calculated the distance, angle, and last time selected for each tile. These values were then used in their respective PDFs to get three likelihood values for each tile. Each tile was then assigned a final likelihood as the product of the three. The tile selection was drawn from the final distribution.

Next, the list of selections, created by L1 or L2, was passed to the movement strategy. The movement strategy, M1 or M2, then fit a trajectory from the player starting area, near the room's door, within selecting distance of each sequential location in the list and back to the door.

The first attempt at creating the M1 strategy was to fit a cubic spline through all the 3-space coordinates of the given list of locations. This proved to be ineffective as the agent ended up standing right on top of each location before selecting it. In practice most participants do not walk over their selected location after selecting it, and almost none stood directly on top of the location before selecting it. To remedy this problem, instead of making M1 fit a spline through the actual selection locations, we made it fit a spline through points beside the actual locations. The points beside the actual locations were created by running an A* trajectory through the actual locations and taking the first point along the A* trajectory within a reachable distance of the actual location. The reachable distance is the furthest the avatar can be from an object and still activate it. In our experiment the reachable distance was approximately the height of the avatar (2 meters). In other words, an agent using the M1 strategy, did not quite travel to each location selection. Instead it came within a 2 meters and started heading to the next location. The cubic spline interpolation was done by parameterizing the desired x and y coordinates with respect to time and performing a one dimensional cubic spline interpolation on each. The agents' yaw was set to the direction it was traveling.

The trajectories mentioned in data collection were used to create the library of human movements for the M2 strategy. Each trajectory was divided into the segments between tile selections. When M2 was asked to create a trajectory from a to b it rotated, translated and scaled each of these segments so the endpoints lined up with a and b. Next, a cubic spline interpolation was fit to each segment, and traced from a to b. If a segment intersected an object or wall it was discarded, and the rest were ranked based on how well they fit. The quality of fit was determined as a linear combination of the amount the segment stretched and the difference between the final yaw of the last segment and starting yaw of the new segment. If we have x as target length over the segment length, the stretching was calculated as the maximum of x and $1/x$. In this way a segment half the desired length was of the same quality as a segment twice the desired length. Finally, M2 uniformly picked a segment from the top 10 highest quality segments. The larger the library of human movements was the more fluid the segment transitions appeared. If all the segments are discarded M2 falls back on spline interpolation for that segment.

An environment novel to an agent is an environment that the agent has never seen, and, in particular, not trained on.

An agent must be able to perform on novel environments in order to be portable. All of these strategies are portable. L1 and M1 do not have any training data. L2 scales its PDFs to fit the environment given and M2 scales the trajectories in its library to fit the environment. Therefore all of the agents discussed are portable, and provided with a traversal map and a list of selection locations can perform in a new environment.

Judging. For this part of the study we ran 96 human participants recruited from a first-year course in psychology. Each participant was briefed with a description of the task and was asked to judge videos of agents in 4 trials: hiding and seeking in each of the two rooms. Each trial consisted of 5 training videos followed by 10 test videos. The judges knew that the training videos were of human behavior. The judges were told that some test videos may be of a computer and some may be of humans with no particular proportion given. The proportion was approximately half. The judges were not told that there may be multiple agents. Each video was played at double speed and was between 3 and 44 seconds long. At the end of each test video the judge labeled it as "Human" or "Computer" which queued in the next video. The judges were also able to re-watch the most recent video before labeling it. In total each judge labeled 40 videos, 20 for hide and 20 for seek. Each judge's score out of 20 for hide and 20 for seek was calculated as the number of correct labelings. These scores were tied to the agent the judge was assigned. The judges were not told their scores.

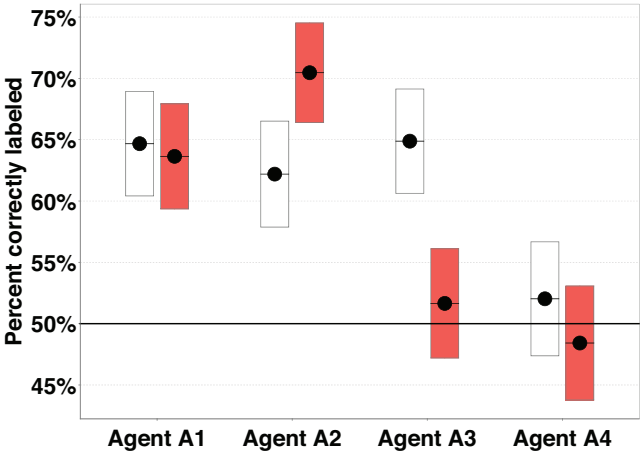

Figure 3: 95% significance boxes for correct agent labeling rates. Empty boxes correspond to the seek task. Filled boxes correspond to the hide task.

Study Results

The correct labeling rates in Figure 3 are the percentage of videos the judges correctly labeled, with the boxes indicating the 95% confidence intervals. The intervals displayed are the Wilson score intervals, used when there is a chance of the mean being close to the boundaries (0% or 100%). The closer an agent is to 50% the closer it is to passing the Turing test. If one confidence bar occurs completely under another bar we can conclude that the first is correctly labeled less often than the second with greater than 95% confidence.

In the hide task, we can see that when comparing the agents with the same location strategy, the agents implementing the M2 strategy were correctly labeled significantly less than their M1 counterparts (A3's box is under A1's box and A4's is under A2's). This implies that the M2 strategy made a significant difference in the hide task. It is interesting to note the same is not true when comparing agents with the same movement strategy (A2's box is not under A1's and A4's is not under A3's).

In the seek task, A4 was correctly labeled significantly less than A1, A2 and A3. This implies that the M2 and L2 strategies made a significant difference when used together in the seek task.

The results above indicate that A4 is correctly labeled as an AI agent less frequently than A1 with 95% confidence. The following analysis allows us to derive an even better confidence bound. To do this we performed a Student's t-test on the scores judges received out of 20. A distribution of the judge scores for A1 and A4 can be found in Figure 4. This score can be approximated as a normally distributed random variable since it is the sum of individual Bernoulli trials. The number of degrees of freedom was 44 in both the hide and seek tasks (24 participants judged A1 and 22 judged A4). The t-test produced a t value of 3.563 in the hide task and 2.723 in the seek task. As a result, we claim A4 was correctly labeled less often then A1 with confidence of 99.9992% for the hide task and 99.08% for the seek task.

Discussion

In the hide task, the lower correct labeling rates of the agents implementing M2 indicates that, in our experiment, the advanced movement strategy made a significant difference. The L2 strategy produced no significant difference in the hide task, but did in the seek task. We attribute this to the limited number of location selections (3) in the hiding task. The location selection strategy plays a much smaller part in the hiding task because 3 choices are often not enough to draw an informed guess to the identity of the agent.

Agent A4 performs significantly better than Agent A1 in both hiding and seeking. The mean near 50% in both hiding and seeking indicates agent A4 has passed the Turing test we set out to pass. That is to say, judges do no better than chance against agent A4.

We can go even further in saying that judges can do no better than chance against agent A4. It is possible, although unlikely, that in the general Turing test a bimodal distribution in judge scores can appear. This would lead to average judge score of 50%, and an incorrect inference that an agent is indistinguishable. For example lets say all human videos start with one turn clockwise and all computer videos start with one turn counterclockwise, but are otherwise indiscernible. If this is the case, every judge can trivially sort the two types of videos into two bins, clockwise and counter clockwise. However, they will not be sure which bin to label human and which to label agent. This will result in half of the judges making the correct guess and labeling 100% of videos correctly and the other half making the wrong guess

and labeling 0% correct. This creates a bimodal distribution in which the mean is 50%, but the agent is fully distinguishable from a human. Any judge who is given this "tell" will be able to correctly label the agent from a human every time.

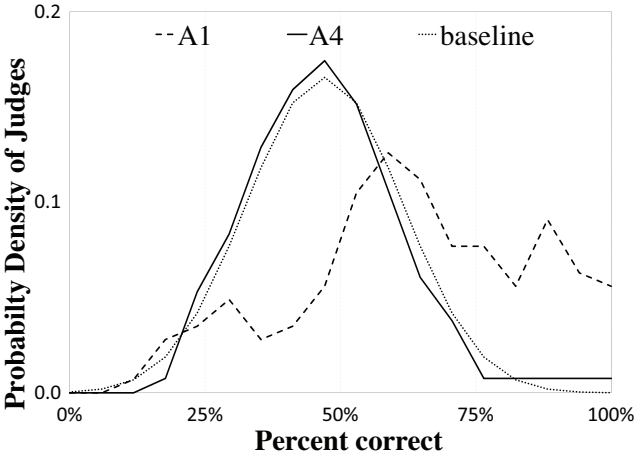

Figure 4: Distribution of judge scores.

If all the judges were equal and none could tell the difference all the scores would be left purely up to chance. This means some would score higher than 10 out of 20, some lower, but most would be centered around 10. The shape of the purely chance distribution is included as the baseline in Figure 4 for comparison. The closer a distribution is to the baseline the closer it is to matching pure chance. The data in Figure 4 suggests that agent A4 is normally distributed about a mean of 50%, and is close to the baseline. Conversely, the distribution for agent A1, is closer to a uniform shape than it is to a normal shape. The probability mass near the 100% side of the distribution for agent A1 shows that there exist judges that can consistently label A1 correctly. It is the absence of these expert judges in the distribution of agent A4 that suggests that agent A4 does not consistently exhibit any "tells" to human judges.

Future Work and Conclusion

Hiding and seeking is viewed as a fundamental cognitive ability of humans and animals and has several applications in video game design. This paper made the following contributions. We proposed a first computational generative model of hide and seek behavior in adult humans. The model is built automatically by data-mining observed human behavior. We implemented a model within an AI agent and demonstrated its validity via a restricted version of the Turing test. Additional complexity in the model was shown to improve performance on the Turing test.

Future work will investigate biasing effects of agent behavior on judge's perception of human hide/seek behavior. It will also be of interest to incorporate our model into a combat agent in an on-line game such as *Counter-strike: Source*.

Acknowledgments

We appreciate the funding by the National Science and Engineering Council as well as contributions from Eric Legge, Craig Anderson, Matthew Brown, and the IRCL group members. We also appreciate Valve for developing software and making it available for academic researchers.

References

Booth, M. 2004. The official Counter-Strike bot. http://www.gdcvault.com/play/1013625/.

Clayton, N.; Dally, J.; and Emery, N. 2007. Social cognition by food-caching corvids. the western scrub-jay as a natural psychologist. *Philosophical Transactions of the Royal Society B: Biological Sciences* 262:507–522.

Clayton, N.; Emery, N.; and Dickinson, A. 2006. The prospective cognition of food caching and recovery by western scrub-jays (aphelocoma californica). *Comparative Cognition & Behavior Reviews* 1:1–11.

Cornell, E. H., and Heth, C. D. 1986. The spatial organization of hiding and recovery of objects by children. *Child Development* 57:603–615.

Cornell, E. H.; Heth, C. D.; Broda, L. S.; and Butterfield, V. 1987. Spatial matching in 1- to 4- year-old children. *Developmental Psychology* 23:499–508.

Dally, J.; Clayton, N.; and Emery, N. 2006. The behavior and evolution of cache protection and pilferage. *Animal Behaviour* 72:13–23.

Darken, C., and Anderegg, B. 2008. Particle filters and simulacra for more realistic opponent tracking. In *Game AI Programming Wisdom 4*. Hingham, Massachusetts: Charles River Media, Inc. 419–428.

Hladky, S., and Bulitko, V. 2008. An evaluation of models for predicting opponent locations in first-person shooter video games. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games*, 39–46. Perth, Australia: IEEE.

Bethesda Softworks. 2011. Fallout: New Vegas. http://fallout.bethsoft.com/.

Blizzard Entertainment. 2010. Starcraft 2. http://www.starcraft2.com/.

Gearbox Software. 2009. Borderlands. http://www.borderlandsthegame.com/.

Valve Coorporation. 1996. Hammer. http://developer.valvesoftware.com/wiki/Valve_Hammer_Editor/.

Valve Coorporation. 2007. Source Engine. http://developer.valvesoftware.com/wiki/Source_Engine/.

Valve Software. 2008. Counter-Strike: Source. http://store.steampowered.com/app/240/.

Talbot, K. J.; Legge, E. L.; Bulitko, V.; and Spetch, M. L. 2009. Hiding and searching strategies of adult humans in a virtual and a real-space room. *Learning and Motivation* 40:221–233.

Valve Corporation. 2004. Half-Life 2. http://www.half-life2.com/.