Second Life as a Simulation Environment: Rich, high-fidelity world, minus the hassles.

Vladislav D. Veksler
(vekslv@rpi.edu)
Cognitive Science Department
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180 USA

Abstract

Second Life™ is a 3D virtual world with unlimited potential as a tool for cognitive modeling. This paper discusses the many advantages of using Second Life versus other simulation environments, the aspects of cognitive modeling that this simulation environment may be appropriate for, the interface setup, and various technical issues. Two simulations are provided as examples of interfacing Second Life with cognitive models, including an example where the high-fidelity complexity and constraints of Second Life may help to distinguish between models and/or parameter values that produce varying performance in different task environments.

Keywords: cognitive modeling, Second Life, 3D virtual worlds, embodiment, ACT-R, task environment, cognitive architectures.

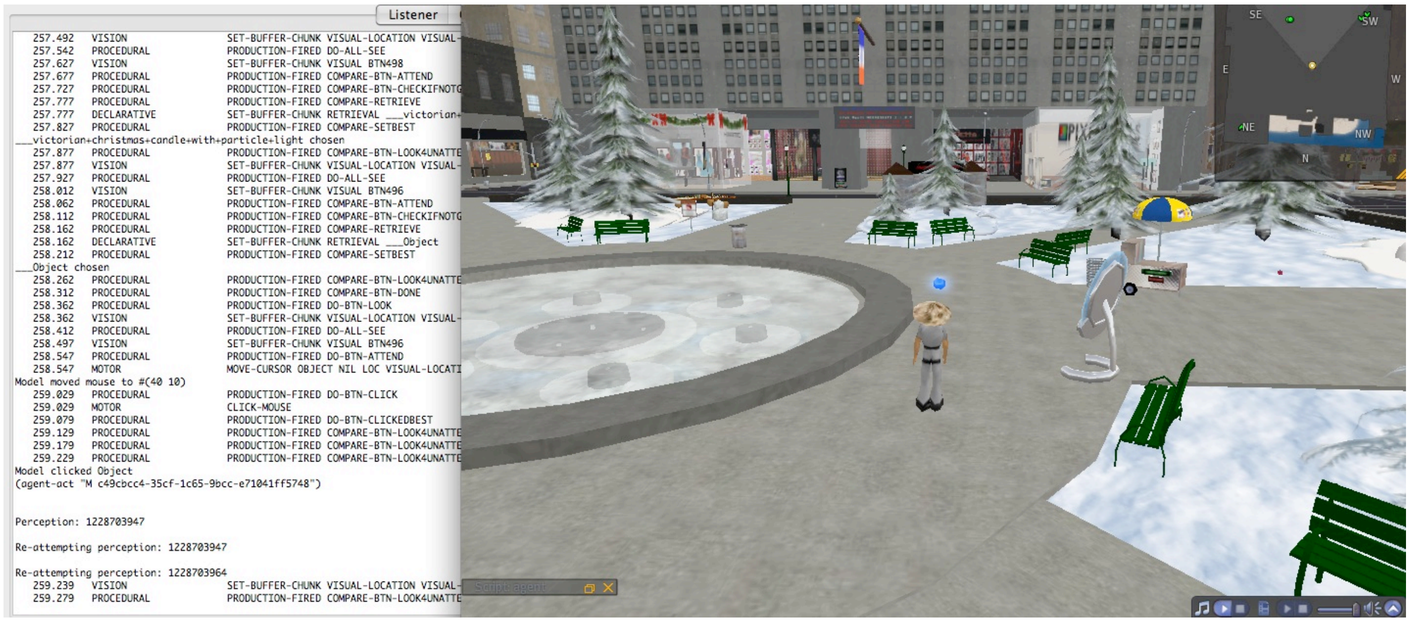
Introduction

The 3D virtual world Second Life™ offers a potential environment for training and testing cognitive models. Second Life is populated by hundreds of thousands of online users, and perhaps millions of virtual objects. This technology may be of interest to the cognitive modeling community for a variety of reasons, including scalability and scope testing, skill transfer simulations and long-term model development, emerging behavioral and social simulations, etc. Second Life provides a very rich, dynamic, and interesting world, (compared with the simple simulation environments that are typical in cognitive modeling), that is well-supported and easy to use and to redesign as needed (compared with robotics), with unlimited tasks, and the opportunity for life-long (rather than simulation-long) learning for a cognitive agent. Some Cognitive Science researchers have already begun to explore Second Life for demo and simulation purposes (e.g. Burden; Merrick & Maher, 2009; Rensselaer Polytechnic Institute, 2008), but more work with this environment is needed to take full advantage of its features.

The rest of this paper discusses the advantages of Second Life over alternative simulation environments for cognitive modeling, the types of simulations that Second Life may be appropriate for, and some key technical issues for modeling in this environment. Finally, two simulations are provided as examples of how cognitive models may be interfaced with Second Life, and how the high-fidelity complexity and constraints provided by the virtual world may be useful in distinguishing between models and/or parameter values that produce varying performance in different task environments.

Why Second Life?

The attraction of Second Life is the same as that of robotics



```
   257.492   VISION                    SET-BUFFER-CHUNK VISUAL-LOCATION VISUAL-
   257.542   PROCEDURAL                PRODUCTION-FIRED DO-ALL-SEE
   257.627   VISION                    SET-BUFFER-CHUNK VISUAL BTN498
   257.677   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-ATTEND
   257.727   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-CHECKIFNOTG
   257.777   PROCEDURAL                PRODUCTION-FIRED COMPARE-RETRIEVE
   257.777   DECLARATIVE               SET-BUFFER-CHUNK RETRIEVAL ___victorian+
   257.827   PROCEDURAL                PRODUCTION-FIRED COMPARE-SETBEST
___victorian+christmas+candle+with+particle+light chosen
   257.877   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
   257.877   VISION                    SET-BUFFER-CHUNK VISUAL-LOCATION VISUAL-
   257.927   PROCEDURAL                PRODUCTION-FIRED DO-ALL-SEE
   258.012   VISION                    SET-BUFFER-CHUNK VISUAL BTN496
   258.062   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-ATTEND
   258.112   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-CHECKIFNOTG
   258.162   PROCEDURAL                PRODUCTION-FIRED COMPARE-RETRIEVE
   258.162   DECLARATIVE               SET-BUFFER-CHUNK RETRIEVAL ___Object
   258.212   PROCEDURAL                PRODUCTION-FIRED COMPARE-SETBEST
___Object chosen
   258.262   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
   258.312   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-DONE
   258.362   PROCEDURAL                PRODUCTION-FIRED DO-BTN-LOOK
   258.362   VISION                    SET-BUFFER-CHUNK VISUAL-LOCATION VISUAL-
   258.412   PROCEDURAL                PRODUCTION-FIRED DO-ALL-SEE
   258.497   VISION                    SET-BUFFER-CHUNK VISUAL BTN496
   258.547   PROCEDURAL                PRODUCTION-FIRED DO-BTN-ATTEND
   258.547   MOTOR                     MOVE-CURSOR OBJECT NIL LOC VISUAL-LOCATI
Model moved mouse to #(40 10)
   259.029   PROCEDURAL                PRODUCTION-FIRED DO-BTN-CLICK
   259.029   MOTOR                     CLICK-MOUSE
   259.079   PROCEDURAL                PRODUCTION-FIRED DO-BTN-CLICKEDBEST
   259.129   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
   259.179   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
   259.229   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
Model clicked Object
(agent-act "M c49cbcc4-35cf-1c65-9bcc-e71041ff5748")


Perception: 1228703947

Re-attempting perception: 1228703947

Re-attempting perception: 1228703964
   259.239   VISION                    SET-BUFFER-CHUNK VISUAL-LOCATION VISUAL-
   259.279   PROCEDURAL                PRODUCTION-FIRED COMPARE-BTN-LOOK4UNATTE
```

Figure 1. Cognitive Agent exploring a park in New York. ACT-R model controlling the agent.

– embodiment (minus the many hassles of robotics, discussed below). A large portion of human cognitive abilities is the result of the complexities and consistencies of our environment. Thus, a dynamic, rich world, with physical laws and consistent object properties may provide for more fidelity than simpler simulation environments, and thus, more useful models of cognition.

Second Life's complexity and constraints may help to avoid some 'false positives', as well as 'misses' in cognitive modeling. A false positive may occur when a cognitive model accounts for human data in a simplified task environment, but cannot scale in the real world. A miss may occur when a cognitive model cannot fit human data without the added complexity and constraints of the real world; thus, the use of a simplistic simulation environment may cause for the model to be incorrectly dismissed.

Second Life vs Robotics

If real-world fidelity is so important, why not just use the real world? There are many limitations to working with robots in the physical environment, versus simulated agents in virtual reality. In addition to the financial expenses, one major problem is that robotics work involves disproportionally more work on the 'body' as opposed to the 'mind'. In the end, a slight change to the task environment (e.g. taking a driving robot off-road) may require changes in both sensory and motor mechanisms.

While biological agents are endowed with appropriate sensory-motor systems for their world, and virtual agents for theirs, robotic agents are in no way equipped to handle the dynamics of the real world. For example, the number of sensors on today's robots, compared to the amount of sensors that a biological cognitive agent might have, is simply laughable. Virtual worlds like Second Life provide for environmental complexity and fidelity, as well as proportionally suitable sensory-motor abilities of virtual agents. Said simply, by using Second Life as opposed to a robot platform, researchers may be able to focus on cognitive research, and avoid unnecessary investments of time and finances.

Other Simulation Environments

Many other virtual simulation environments exist, and may be used for cognitive modeling. Some of the alternatives have better graphics, which can be very useful for demo purposes, some have a faster interface for brain-body communication, etc. However, due to the sheer size of the Second Life user community, due to its steadily increasing popularity, it makes for a much richer, ever-growing world. Additionally, the commercial value of Second Life is reflected in greater expansion of its technical capability and technical support. Using Second Life over a less popular simulation environment may be equated to using the World Wide Web over a Bulletin Board System.

What in the world of Cognitive Modeling is Second Life good (and not good) for?

Second Life may NOT be employed for modeling millisecond response times, nor is it appropriate for large-scale parameter exploration. Rather, Second Life is best used for modeling performance, and learning curves. Specifically, Second Life is best employed for (1) testing the scope of models' learning/decision-making mechanisms in complex and dynamic, distractor-full environment, (2) modeling adaptation and skill transfer, and (3) social modeling.

Complexity and Constraints

Spatial navigation is a prime example of a task that requires the complexity and constraints of Second Life for cognitive modeling. When modeling navigation, researchers often unrealistically represent the environment as a flat grid of adjacent spaces (e.g. Braga & Araujo, 2003; Voicu & Schmajuk, 2002). Some alternatives may be to include two-way or one-way wormholes. Different models may thrive in different environments, and so the choice of task-environment is not trivial. Second Life may be employed to provide realistic uncertainties and constraints. Although Second Life bears many geographic properties (e.g. if space A can be reached from space B, usually this means that space B can be reached from space A), it also provides many realistic uncertainties (e.g. object B may be in view when approaching object A from the East, but not from the North; object C may be dynamic, sometimes to be found in proximity with A, and sometimes in proximity with B, etc.). The use of a high-fidelity environment may help to deduce high-fidelity cognitive models and parameter sets.

Task Variety and Skill Transfer

Second Life may be used for simulations of a variety of tasks, from playing with building blocks, to maze-running, to soccer, etc. Most tasks are very easy to set up, require no programming or 3D modeling background, and are reusable by other researchers. Of great importance is the fact that an agent may 'live' and develop in this rich world, learning new skills along the way. The multitude of tasks can also help in modeling skill-transfer – an important qualification of human intelligence. A cognitive agent may adopt their soccer skills to hockey, walking skills to driving, and block-building skills to tower of Hanoi, Tetris, and sculpting.

Technical Setup and Complications

The Second Life programming language, LSL, is required for interfacing Second Life objects with cognitive models. Although the basic algorithm is relatively simple (capture and send sensory information to model; perform any actions returned by the model), some complications are bound to arise.

Land Ownership

There are many parts of the Second Life world where new objects cannot be created because the landowner does not allow this. There are parts of the world, called sandboxes, where users are encouraged to build and script their objects; however, objects usually cannot remain in most sandboxes for longer than a few hours. Thus, sandboxes may be fine for building models and running short simulations, but not for longer lifespans or more controlled simulations. One alternative is to buy land. Another may be to connect an object to an avatar (see a lengthier discussion of this in the Region Restrictions section below).

One last alternative is to use land that may be offered for research purposes by a university or a private research institution. For example, the Second Life AI Laboratory (SLAIL) provides booth size spaces for free to anyone undertaking research in AI (cognitive modeling included), and particularly AI in virtual worlds, providing a permanent exhibition, meeting and collaboration space for the community. The space may be found on Daden Cays in Second Life – (http://slurl.com/secondlife/Daden%20Cays/152/44/22; for more details visit http://knoodl.com/ui/groups/ArtificialIntelligenceGroup/wiki/SLAIL).

Region Restrictions

Scripted objects in Second Life are restricted from entering certain regions. If a modeling simulation requires travel beyond known open regions, it may be necessary to use an avatar (an avatar is a representation of a human user in Second Life, and only exists as long as the user is logged on). One simple way to resolve this issue is to attach the object interfaced with a cognitive model to an avatar. For example, the neon-blue sphere floating above the avatar's head in Figure 1 is an object scripted to interact with a cognitive model. For demo purposes the scripted object can be made see-through, tiny, or made to look like an article of clothing (e.g. a hat).

Firewall Issues

When a computer running a cognitive model is using DHCP, or if it is behind a firewall, a dedicated web server is necessary for interfacing the model with the Second Life world (Figure 2). Alternatively, LSL scripts can answer HTTP requests from the cognitive agent directly through their XML-RPC service (XML-RPC is a standard for XML structure for sending function calls to remote systems). This latter route is sometimes unreliable and may be deprecated ("Category:LSL XML-RPC - Second Life Wiki," 2009), but may be faster than the setup shown in Figure 2, depending on the speed of the researcher-owned web servers.

Asynchronous HTTP Calls

A question may arise when a cognitive model sends a command to its Second Life 'body' (e.g. "move toward the fountain", "raise left arm .2m", "push the block object"), and receives information back about the state of the world, as to the time of the state. The model may require information as to whether its last action has been performed, and whether the HTTP responses are in order. This is easily resolved by sending a timestamp along with the last performed action from the body script to the model.

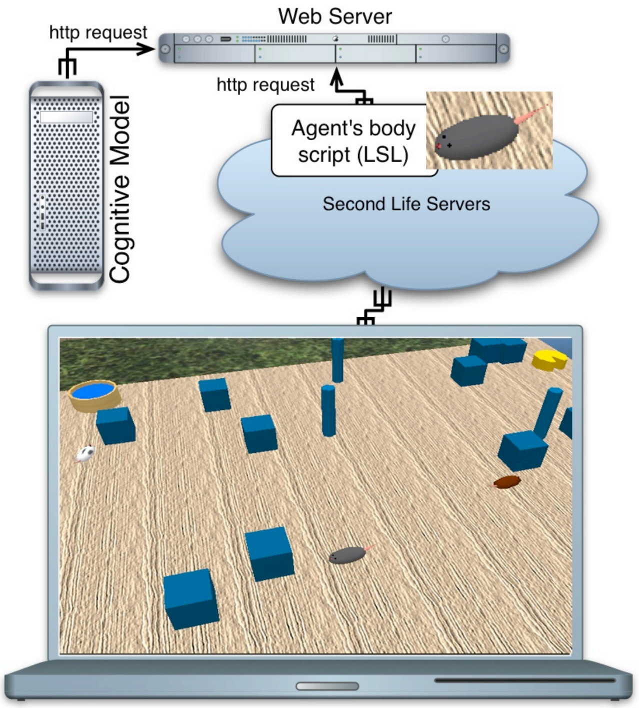


Figure 2. Second Life setup for models on DHCP or behind a firewall. Simulation shown at bottom has 3 models exploring a maze with cheese and water.

Memory Issues

Second Life scripts are relatively restricted in memory (16KB total for Byte-code, Stack, Free Memory, and Heap). This may be a serious restriction for collecting data about the state of the agent and keeping a copy of the prior state (prior state information may be necessary to avoid sending unchanged information to the model, saving both speed and bandwidth). This is not an issue when world-state contains only the last taken action plus the names of a few surrounding objects, but becomes an issue when collecting all possible information (object id, name, description, position, direction, velocity, dimensions, etc.) for a large number of objects.

Scanning

Other complications may arise in the way that a model in Second Life may be allowed to scan around for nearby objects. The scan is performed as a sphere, rather than a cylinder. This may take unnecessarily long for a large radius. A smaller radius may be scanned for a simulated sense of smell, but for long-distance vision, scanning must be restricted from a sphere to a smaller cone.

Speed

The greatest complication is that the perception-action protocol can take a relatively long time. This, of course, depends on the setup of scanning and HTTP requests. The greater bottleneck seems to be the maximum rate of HTTP requests (capped at 25 requests in 20 seconds). The assumption in modern cognitive architectures (e.g. Anderson & Lebiere, 1998) is that visual information is used at most 10 times per second (50ms for attention shift, and 50ms for attending the information). Thus, it seems that Second Life vision is about 10 times slower than may be desired for real-time cognitive models. This is not a major problem for interacting with static objects or other (similarly retarded) models, but it is a problem nonetheless. However, the technical support enjoyed by the Second Life community carries the promise of near-future solutions for these issues.

Specifics of Sample Simulations

Simulation 1

The first simulation was attempted to examine how a cognitive architecture may be interfaced with Second Life. The ACT-R cognitive architecture (Anderson & Lebiere, 1998) was connected with a Second Life script through an intermediary web server, as displayed in Figure 2. A scripted object was created in Second Life that would scan the world every few seconds, and send the state of the world via an HTTP call to the intermediary web server. On the ACT-R side, a cognitive model, in a perceive-think-act loop, would request an updated world-state from the intermediary web server, decide upon an action, and send motor commands back to the server.

Second Life Setup

The Second Life scripted object was attached to an avatar for greater exploratory capabilities (without an avatar scripted objects are restricted from many lands). The script performed a regular scan of nearby objects with a radius of 2m. If less than 5 objects were detected, the radius was increased, and another scan was re-initiated, until at least 5 objects were detected. Much more information was collected and transferred to the ACT-R model than was necessary for this simulation (e.g. object position, velocity, size, etc.), as this helped to examine the technical limitations of the setup. In addition, information sent to ACT-R included a timestamp, and the latest received motor command.

ACT-R Interface and Model

ACT-R visual and motor components were interfaced for Second Life in the following manner. Lisp functions were added to send out motor commands to the intermediary web server, and to retrieve world-state from the server. The ACT-R visual information (visicon) was filled with Button objects, each Button containing the name of its corresponding Second Life object found in the world-state list from the server. Upon clicking one of the button objects, a command would be issued, via a call to the web server, to move toward the corresponding object.

The ACT-R model employed to examine this interface was the Goal-Proximity Decision model (Veksler, Gray, & Schoelles, 2009). The details of the model are not provided here, as this is tangential to the focus of this paper. The general idea of the model is that it attends all objects in the visicon, and clicks the object with the greatest strength of association to the current goal (plus or minus noise). The strength of association between objects, in turn, is updated based on experienced temporal proximity of those objects.

Simulation Results

The Second Life script was first limited to find only the objects that belonged to its owner, which comprised sixteen randomly distributed boxes that served as navigational landmark (Figure 3). The model was presented with each of the sixteen objects as its goal, one at a time, until it successfully found each object.

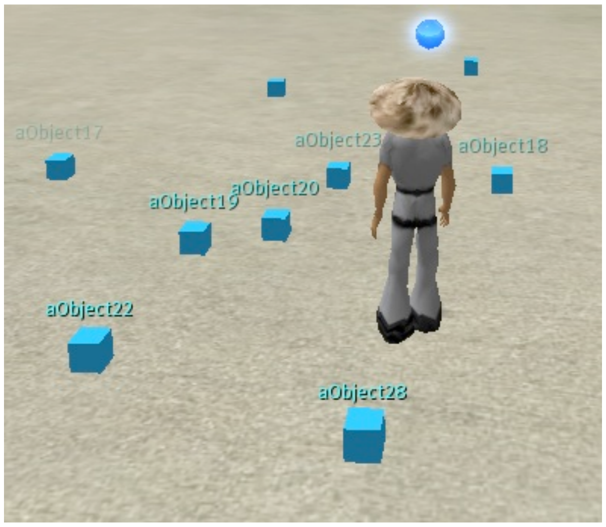


Figure 3. Second Life simulation. Controlled environment, with researcher-owned objects.

Upon the successful completion of this exercise, the scanning restrictions were removed, allowing the model to 'see' all objects within its scanning radius. The model was moved to an object-rich region, Washington Square Park in Manhattan (Figure 1), and manually given various objects as its goals (e.g. fountain, bench, store). Although the model was able to successfully navigate most of the region, some distant objects were unreachable due to the chosen scanning procedure. Thus long-distance vision, as discussed in the Scanning section above, may be necessary for most exploratory agents.

Simulation 2

The purpose of the second simulation was to examine whether Second Life can be set up to help distinguish between sets of model parameters for a Reinforcement Learning model. Reinforcement Learning (Sutton & Barto, 1998) is a widely implemented model of trial-and-error behavior. The specific form of Reinforcement Learning that was implemented in this model was a closed-form version of the ACT-R decision/utility-learning mechanism. The model chose which object to approach based on the utility of that object, given the specific goal (plus or minus noise). Upon reaching its goal, the model updated the utility of all encountered objects for reaching that goal based on the ACT-R utility-learning equation (Bothell, 2008).

The details of the model are relatively tangential to the focus of this work. What is important, however, is that there are several free parameters in this model (e.g. exploratory noise, learning rate, etc.), and that the same parameter values may cause different behavior for different task environments. Thus a high-fidelity task environment, like Second Life, may be necessary to distinguish between different parameter sets.

Different Task Environments

Parameter searches were performed with the model using three different maze structures. Each maze contained sixteen available spaces for the model to explore. The mazes were rated according to the average difficulty of finding each possible maze location from each possible starting point, by means of a random walk. The easy, medium, and difficult mazes required on average 181.39, 369.83, and 793.79 steps, respectively. The easy and medium difficulty mazes were set up in a grid-like fashion, with bidirectional movement allowed between any two neighboring locations. The easy maze, shown in Figure 4A, allows movement in all eight directions to its neighboring cells (N, NE, E, SE, S, SW, W, and NW). The medium difficulty maze, shown in Figure 4B, allows movement in four directions (N, E, S, W). The difficult maze, shown in Figure 4C, was set up with unidirectional and bidirectional connections, without regard for grid consistency.

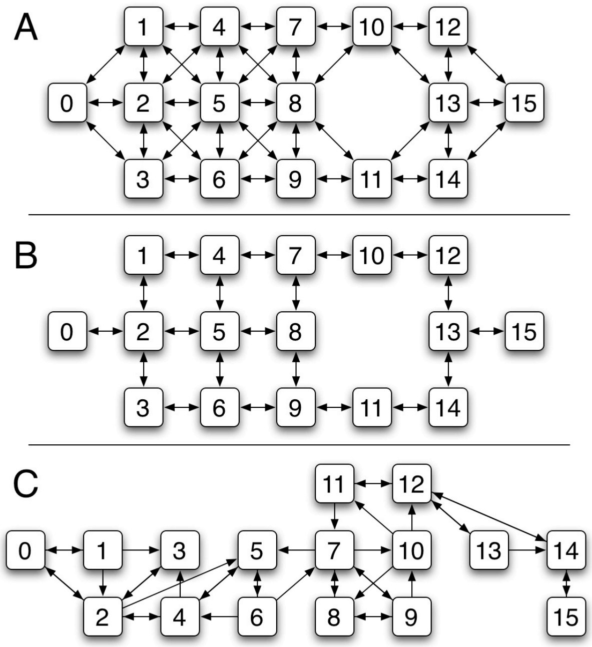


Figure 4. Sample navigation task environments. Numbered boxes signify locations, arrows signify the directions in which an agent may travel.

Different Parameter Sets

The model ran through each of the three tasks 60 times for each parameter set (noise was varied between .01 and 30, learning rate was .001 and .2). Each model run consisted of five bins, where the model had to reach sixteen goal in each bin (every possible location was set as a goal, in random order). The best performance (as measured by the average number of steps taken by the model to reach a goal in bin 5) for each maze was achieved with a different set of values for the free parameters in the model. The best parameters for the easy maze (paramsEasy) was achieved when the noise parameter was set to 5 and the learning rate was .1, for the medium difficulty maze (paramsMed) when the noise parameter was 25 and the learning rate was .1, and for the difficult maze (paramsHard) when the noise parameter was 15 and the learning rate was .01. A 3x3 two-way ANOVA revealed a significant interaction effect of ParameterSet × MazeDifficulty, $F(4, 531) = 115.42$, $p < .001$, a significant effect of ParameterSet, $F(2, 531) = 167.60$, $p < .001$, and a significant effect of MazeDifficulty, $F(2, 531) = 346.52$, $p < .001$. Post-hoc Tukey HSD comparisons revealed significant differences between the performance of all three parameter sets at the $p < .05$ level.

Second Life Simulation

Given the differences between paramsEasy, paramsMed, and paramsHard on the three types of task environments, it may be appropriate, and even essential, to establish which parameter set is best in a high-fidelity task environment. A Second Life simulation was set up as a proof of concept. Figure 2 is an accurate representation of the modeling setup, with a connection through the intermediary web server, with the models being represented as mice in a maze, with random poles and boxes (serving as landmarks), and three possible goals: swiss cheese, cheddar cheese, and water bowl. The complexity of the task, as well as its fidelity, was augmented with a greater number of objects and the presence of dynamic objects (other mice). The model was minimally altered so as to receive perceptual information from Second Life, and send motor commands back (the perception and action functions from Simulation 1 were reused).

The focus here is (1) to point out that choosing a task environment for cognitive modeling is not trivial (2) that Second Life, in theory, is an appropriate environment for task simulations, and (3) that Second Life, in practice, can be successfully interfaced with cognitive models. On the latter point, the model ran once with each of the three parameter sets, each run consisting of ten bins, where each bin comprised finding the three goals, one at a time, in random order. Early results (see Figure 5) suggest that the three parameter sets eventually converge, but this may take an extremely long time (≈27 walks through the maze, which, at worst, is almost 3000 steps). The average number of steps taken to reach a goal is 35.9 for paramsMed, 99.4 for paramsEasy, and 148.5 for paramsHard.

These results are not interpretable without more data, nor, even if the trend should continue, could we assume that the medium difficulty maze shown in Figure 4B may be used in place of high-fidelity task environments. Instead, the suggestion is that these task environments should be used in combination: one to quickly test many models and parameter values, the other to test whether a model could scale up to the complexities of dynamic and uncertain worlds.

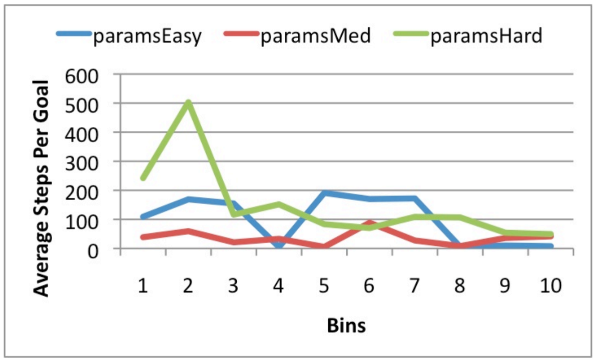


Figure 5. Second Life simulation results from three different sets of parameter values.

Summary

In sum, Second Life may be an important tool for cognitive modeling. It provides a better balance of real-world complexity and constraints than simpler simulation environments, less hassle and financial investment than robotics work, and it stands out from other 3D virtual world with a rich, massive-multiuser environment, and extensive technical support. The Second Life environment may be easily interfaced with cognitive architectures, as described in Simulation 1, or with closed form models, as described in Simulation 2. As Simulation 2 suggests, Second Life modeling work can help to answer questions as to the fidelity of various cognitive mechanisms and/or parameter values whose performance may vary in different task environments.

Future work will address rigorous statistical comparison of model performance in Second Life versus other task environments. Other plans include implementation of long-distance visual scanning coupled with head-movements, and exploration of a greater variety of tasks (e.g. soccer, building blocks, hide and seek).

Acknowledgements

I would like to thank Wayne D. Gray and Michael J. Schoelles for their theoretical and editing contributions to this work. The work was supported, in part, by grant N000140710033 to Wayne Gray from the Office of Naval Research, Dr. Ray Perez, Project Officer. Much of this work was carried out on space provided at the Second Life AI Laboratory (SLAIL) on Daden Cays in Second Life - http://slurl.com/secondlife/Daden%20Cays/152/44/22.

References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

Bothell, D. (2008). ACT-R 6.0 Reference Manual: Working Draft. Retrieved April 13th, 2009, from http://act-r.psy.cmu.edu/actr6/reference-manual.pdf

Braga, A. P. S., & Araujo, A. F. R. (2003). A topological reinforcement learning agent for navigation. *Neural Computing & Applications, 12*(3-4), 220-236.

Burden, D. J. H. Deploying embodied AI into virtual worlds. *Knowledge-Based Systems, In Press, Corrected Proof*.

Category:LSL XML-RPC - Second Life Wiki. (2009). *Second Life Wiki* Retrieved April 13, 2009, from http://wiki.secondlife.com/wiki/Category:LSL_XML-RPC

Merrick, K., & Maher, M. L. (2009). Motivated Learning from Interesting Events: Adaptive, Multitask Learning Agents for Complex Environments. *Adaptive Behaviour, 17*(1), 7-27.

Rensselaer Polytechnic Institute. (2008). Bringing Second Life To Life: Researchers Create Character With Reasoning Abilities Of A Child. *ScienceDaily*. Retrieved April 14, 2009, from http://www.sciencedaily.com/releases/2008/03/080310112704.htm

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT Press.

Veksler, V. D., Gray, W. D., & Schoelles, M. J. (2009). *Goal-Proximity Decision Making: Who needs reward anyway?* Paper presented at the 31st Annual Meeting of the Cognitive Science Society, CogSci 2009.

Voicu, H., & Schmajuk, N. (2002). Latent learning, shortcuts and detours: a computational model. *Behavioural Processes, 59*(2), 67-86.