GAMES AND ARTIFICAL LIFE

Artificial Intelligence has been a crucial component in video games throughout its history. The use of logical agents to provide a challenging and engaging experience is a key factor that separates the design of video games from traditional games. AI controlled opponents and virtual environments allow for the competitive aspect of game play to transfer over to a single player experience that would otherwise not be possible.

Early games could not afford to implement sophisticated AI. Limited processing power left little room for complex decision algorithms and improved graphics were seen as a better investment. As technology improved, more powerful gaming systems allowed for much more complex non-player character behavior. Designers began to experiment with more complex logical agents, focusing on creating improved pathfinding, combat tactics, learning agents, and other improvements. Attempts to create realistic AI behavior, however, have fallen short. Bethesda is one of the more notable examples of a company attempting this undertaking through its Radiant AI system, first demonstrated in The Elder Scrolls IV: Oblivion.

Oblivion and its Radiant AI were set to revolutionize the games industry and the way players view game AI. The Radiant AI "imitates the impact of resource pressure, individual needs, and player interaction on a non-player character (NPC), resulting in a dynamic, procedural reaction" (Griffin, 2010, para 2). Non-player characters possessed their own personalities and goals, and could interact with the environment to satisfy their needs. Previews of the game excited players over the prospect of such a rich and dynamic world to interact with. When the game shipped, many were disappointed to find that many of the advanced features of the Radiant AI were not present. It turned out that the full AI system ended up causing unexpected behavior, as elaborated on by Sanjeev(2006):

> 1. One character was given a rake and the goal “rake leaves”; another was given a broom and the goal “sweep paths,” and this worked smoothly. Then they swapped the items, so that the raker was given a broom and the sweeper was given the rake. In the end, one of them killed the other so he could get the proper item.
> ...3. In one Dark Brotherhood quest, the player can meet up with a shady merchant who sells skooma, an in-game drug. During testing, the NPC would be dead when the player got to him. The reason was that NPCs from the local skooma den were trying to get their fix, did not have any money, and so were killing the merchant to get it.
> ...5. In one test, after a guard became hungry and left his post in search of food, the other guards followed to arrest him. The town people looted the town shops, due to lack of guards.
>
> Bethesda worked to fix these issues, balancing an NPC’s needs against his penchant for destruction so that the game world still functions in a usable fashion. In-game there are over 1,000 different NPCs, not including randomly spawned monsters and bandits. The result is that the AI in the release version is much reduced, only featuring NPC schedules.

The behavior described above is impressive in terms of emergent game experiences, but its flaws are readily apparent. The responses of the agents are very erratic and exaggerated compared to normal human behavior (such as in case 1 and 5), and these responses in turn have negative impacts on elements of story and gameplay (case 3). While it appears that Bethesda considered all the elements necessary for proper human agents, the Radiant AI lacked in making natural choices to a human observer. The functionality is there but the psychology is not. This paper will attempt to address the issue of making human-like decisions by applying psychological principals to AI design.

To create a more believable human AI in games, there must be a fundamental shift in the perception of the AI and its place in the game world. If an agent is to be as human-like as possible, their behavior should be able to exist independent of the game itself. By separating game mechanics from the high level decisions of game entities, the design not only retains more lifelike decisions but also becomes a more flexible design that can be reused in other games. Thus, the goal of this approach is to first make an AI system that replicates human behavior at a high level, then as the high level decisions are made, lower level game specific tasks can be implemented on a game-by-game basis. The purpose of this paper is to implement and evaluate this approach based on its feasibility, performance, and effectiveness.

Human behavior is an incredibly complex thing. A proper understanding of human thought processes and the logic behind a person's behavior can take years of study, and even then research is still ongoing. Creating a perfect model of human behavior is more likely to come from a fully synthetic brain than a clever AI programmer. Luckily, a perfect representation is not needed to be a believable representation. Applying a well-researched model of human behavior to a game AI provides a base to work off to create believable AI.

PICKING THE MODEL: MASLOW'S HIERARCHY

The psychological model chosen for this project is Maslow's Heirarchy of Needs. In his A Theory of Human Motivation, Abraham Maslow detailed his numerous findings on what drives human motivation. Of note is one of his more interesting propositions, that “human needs arrange themselves in hierachies of pre-potency...the appearance of one need usually rests on the prior satisfaction of another"(Maslow, 1943, p. 370). This concept gave birth to the Maslow Hierachy, a tiered organization of human needs that demonstrate which human needs are dependent on others before they become motivational factors. The Hierarchy itself consists of five levels of human needs arranged in a pyramid. At the bottom of pyramid are Physiological needs, the most basic requirements of life such as food, water, and sleep. The next layer is Safety, ensuring that one possesses enough resources and stability in the near future to be confident of his or her future. The following three levels are, respectively, Love/Belonging, Esteem, and Self-actualization. As the bottommost needs are satisfied, higher level needs then appear to motivate the decision process. For example, someone "lacking food, safety, love, and esteem would most probably hunger for food more strongly than anything else"(Maslow, 1943, p. 373).

Maslow's Hierarchy is an excellent model for simulating human behavior in games. The model is simple enough for anyone to understand but its simplicity also makes it flexible. It already provides the conditions of high level decision making, and defines those conditions in a general enough manner to be applied to a wide variety of game types. The bottom two layers tie in extremely well with the common gameplay themes of survival against a harsh environment or enemy. Not all layers need to be implemented either, allowing games to select the particular levels their game addresses and dropping the rest without changing the basic decision model.

BUILDING THE HIERARCHY

To test the effectiveness of the Maslow-based model, an experimental game is developed using the principles of the design for the game agents. The game features a player-controlled avatar interacting with a 2D environment populated by various world objects and AI entities representing human farmers. Each agent is given a set of goals to accomplish that correspond to the first two levels of the Maslow Hierarchy, covering the most basic needs of survival and stable health. The game world is populated by houses and farmland that are property of the agents, and can be used to satisfy their needs. Both player and agents are able to alter their surroundings in a positive/constructive way and a negative/destructive way. Positive actions offer a benefit to needs, such as working farmland for food or planting new crops. Negative actions cause harm, such as destroying crops or harming other agents.

The experiment was developed using C# and the XNA 4.0 library. Ease of use and flexibility were the two main considerations when selecting language and library. The game consists of three primary classes: a main class to execute the game and update all the game objects at each frame, a Utility class that stores the condition variables used for evaluating decisions, and an umbrella class called Entity that handles information related to all in game objects. Entity is further broken down into Agent, Player, and WObject subclasses.

Agents are the game's AI entity and contain the implementation of the Maslow AI being evaluated. Each Agent possesses an instance of Utility that is used for tracking their current state, and its values are used in the logic function to make decisions. The Agent class also includes two arrays of WObjects. The first array holds all of the agent's available knowledge about the world. Whenever a new object is sensed, it is added to the knowledge array. The second array consists of all WObjects that an agent is considered the owner of. Owned property takes precedence over unowned property due to its non-existent risk. While this array is somewhat redundant and could easily be integrated to the knowledge array, separating out the owned property from the total knowledge base improves readability and understanding of the code, however does result in additional memory usage.

The Utility class contains all the important information related to making decisions for the Agents. Utility contains two sets of condition values: an available amount of utility for each condition value (for evaluating the current state of an Agent/WObject) and a return amount for how much utility can be transferred in one game iteration (for evaluating a choice once a goal state has been specified). Whenever an agent uses an object, that object will return the gained utility based on the action taken, and the values are added to the agent's own utility. The values tracked include hunger, sleep, stored food, and threat. These are just the conditions used for this experiment, and an actual game application could have many more depending on the needs of the project and how detailed the decision process should be.

Due to limited resources, the sample game focuses only on the first two levels of Maslow's Hierarchy. The first level, Physiological, uses the hunger and sleep Utility to form its decisions. While the Physiological level covers a larger range of needs, food and sleep are far more relevant to games than the others listed. The second level, Safety, relies on the food stores condition variable. Safety also covers a large range of needs, many of which could have game applications. The context of the example game made security of resources, in this case food, the best choice for second level decisions.

The threat metric stands separate, as it doesn't particularly fit into a specific level. Rather, it modifies the decision process of the other levels based upon the perceived risk of an action, object, or person. An agent eating its own food carries zero threat: there is no chance of a conflict for this action. However, eating another agent's food carries the risk of being seen, and the other agent retaliating in defense. Thus, unowned property has some level of threat. Threat helps make decisions based on the state of the agent, used to evaluate the risks of one option over another. Agents will not take a risky option unless a no-risk option is not available. This element of risk evaluation helps prevent some of the flaws of the Radiant AI. Violence is much closer to a last resort, and agents will fight to death only in situations where it would be reasonably be expected to do so.

These components combine together to form the decision logic of the agents. An agent will first check to see if its immediate Physiological needs are met. If the agent determines it is not in a stable state, it will attempt to satisfy its needs. It will look first for sources of food and rest that are owned by the agent. The greater the immediate need, the greater the risk the agent will take to satisfy its need, such as if an owned source of food cannot be reached in time, a nearby unowned source will be picked in order to ensure survival. Agents will first prioritize owned sources and efficient sources (those that give/possess the greatest utility), as long as they are within a reasonable range. Once

Physiological needs are met, the agent attempts to secure long term food sources. This is achieved through working farmland and planting new crops. The agent will work the farmland available to it until it runs out of its stored utility. If the amount of farmland is insufficient for long term stable survival, the agent will being planting new crops that will turn into farmland to fulfill the needs. When both levels are seen as being in a stable state, the agent stops acting. In a further experiment, it would be worthwhile to code some third level Social actions for the agent to perform, such as wandering around the game world and talking to other agents.

PUTTING MASLOW TO THE TEST

After the game's completion, it is run through several tests of agent realism and performance metrics. The game, as seen in Figure 1, uses simple colored blocks to identify different objects in the world. The red square is the player himself. Blue squares are agents and brown squares are agent homes. The dark green squares are farmland, with the lighter green representing land associated with a particular house. This land can be turned into farms if the agent chooses to do so.

The first run of the game revealed an oversight of the agent design that produced unnatural behavior: every agent took the same actions at the same time, causing the entire village to be synchronized. Every agent and object contained the same default values and use rates, which caused the identical behavior across agents. This prompted the addition of additional randomization of agent properties. Such properties included movement

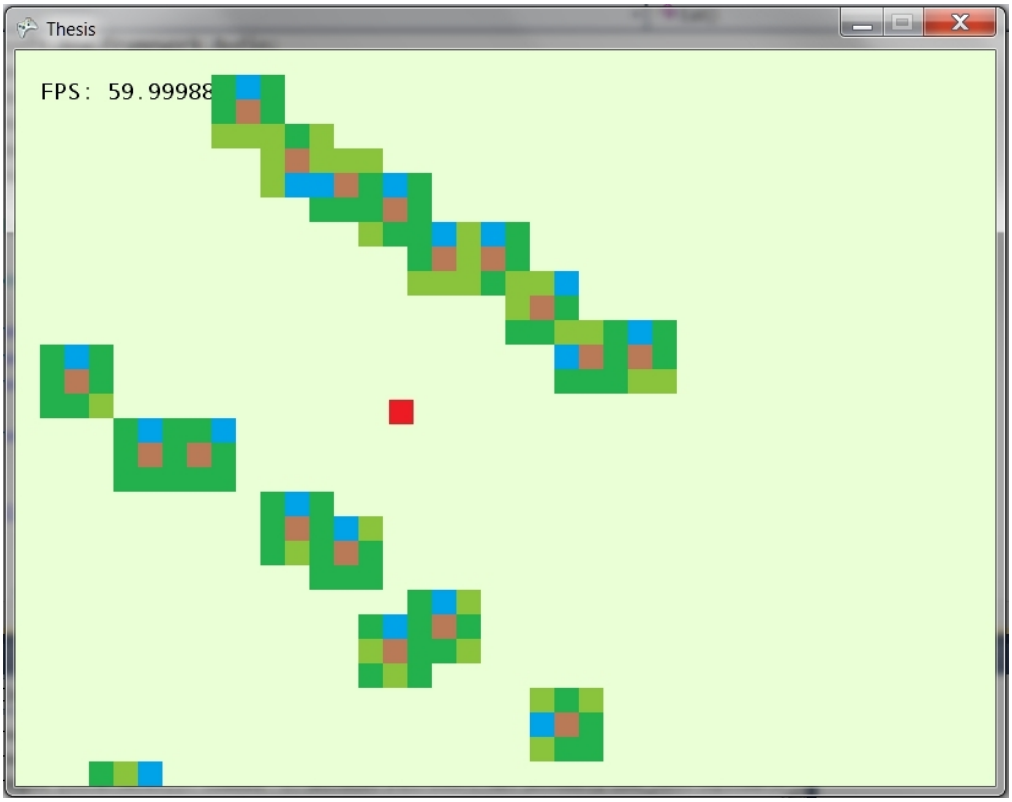

Figure 1: Game in progress

speed, work rate, resource usage, and others. This change also supports the overall theme of creating a realistic representation of human behavior. While this model assumes all humans follow the decision making outlined by the Maslow Hierarchy, individual

requirements for each need vary from person to person. With reasonable variance of need requirements added, the simulation is run undisturbed for a period of time. The agents kept themselves in a steady state, working their own farms for food and sleeping when necessary. As expected from the stable game environment of this test, there were no unusual or unexpected behaviors and agents remained self-sufficient.

With property-owning agents working as intended, the next test sought to examine the agent behavior when resources were not so readily available. Additional agents were added to the scene, but were not given a home or farmland. A few unowned pieces of farmland were also randomly placed about the map for these agents to use as a risk-free option. When the game first started, the 'homeless' agents wandered, looking for possible unowned food sources to minimize risk. When these agents became hungry, those that had found an open food sources used them up as much as possible. Agents that did not find an open food source or whose available sources weren't sufficient were then faced with choosing among all known food sources owned by other agents. Agents prioritized food sources where they would least likely be caught by other agents, such as when the owner was eating or sleeping, then running off once spotted by the owner. No agents seemed to reach a state where violence became a necessity, instead doing well enough by taking food at opportunistic intervals and looking for the open farmland when possible. These agents began falling into a set routine, taking familiar paths and stopping at the same sets of fields for food. An eventual set schedule produced sufficient food sources while minimizing risk, and the game world stayed stable. While the development of individual routines can be seen as a benefit of the AI's decision making, the effect is static in appearance to the player. If the agents perform the same set of actions at regular intervals, the Maslow AI resembles traditional scripted AI. Resolving the issue of regular schedules is a mark against the Maslow AI, as simple systems will then require additional components to keep the agent behaviors varied and interesting.

Player interaction with the game world is the true test of the merit of the Maslow Hierarchy. Providing player agency and having agents respond to player behavior is a key factor in both creating an engaging game experience and in creating a realistic AI.

Several games were played using two different types of player actions: growing open farmland for agents to use, and attacking agents/property. Agent responses to player actions were then observed.

The beneficial farmland creating player had a large impact for the first few farms created, then quickly suffered diminishing effects from each new one. New farms altered the routine of the homeless agents to take advantage of the new resource, as they were zero risk compared to the option of theft. However, once there were enough open farms to feed all agents, further fields had no additional effect. Limitations on the game world and its available mechanics and systems are an important consideration for beneficial player actions. Whenever a player does something to benefit an agent, the action taken moves an agent further up the Hierarchy, as he is helping satisfy a need. Thus, positive actions are inherently limited by the highest implemented level of the hierarchy. As higher levels are more difficult to implement, rewards for positive actions are also more difficult to include.

The harmful player produced much more engaging results. An initial attack was made against an agent's farm and home. The agent first tried minor attacks to scare the player away, like they would an unwelcome agent on their property. After continued attacks and destroyed property, the agent retaliated in full force, as failure to do so would spell the end of his livelihood. The agent is killed by the player along with remaining property. Attempts to destroy another agent's property are immediately met with desperate attacks, due to the increased threat associated with the player after having killed the other agent. As more agents were killed by the player, the remaining agents began to flee in terror whenever the player was around because of his incredibly high perceived threat, as seen in Figure 2. Negative player actions are much more accustomed to the Maslow Hierarchy AI for similar reasons why positive actions are more constrained. Harmful actions move an agent downward on the hierarchy, where the bottommost levels are the most likely to be supported and the most commonly found levels in gaming. Since one cannot move off the hierarchy, the behavior stays believable as the bottom level is already supported by the AI system. The Maslow AI thus functions much stronger with destructive player actions and agents and environments that are intended to be attacked by the player.

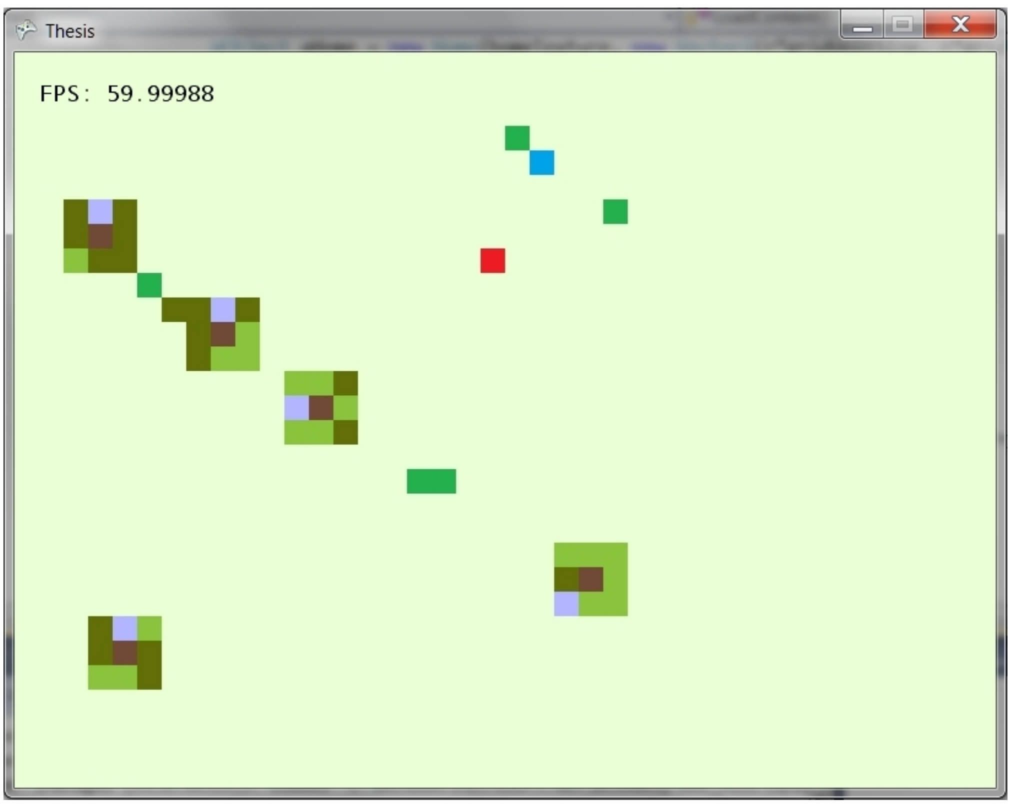

Figure 2: The lone surviving agent fleeing the player in terror.
The pink, dark brown, and dark green squares are destroyed homes, farmland, and agents

REVIEWING THE THEORY

While the program produced mostly desired results, the experiment as a whole resulted in some mixed observations as to the benefit of the approach. For the evaluation, it is important to observe positive and negative aspects from the standpoint of a player playing the game as well as the programmer involved in its creation. For the Maslow approach to be valid, it must display the desired effects of a more believable AI while keeping the additional development time required low enough to be worth the added benefit.

The additional difficulties in programming the Maslow AI and its needed components result in a substantial drawback for the flexibility of the model. In order to even implement the approach, the game's mechanics must necessitate two or more levels of the hierarchy for the AI to act upon. A game such as a First Person Shooter would have a difficult time integrating more than one level. Even if a shooter could incorporate more than one level, the decisions made between levels would have very little game impact compared to the decisions made within a level, such as combat tactics. The hierarchy also doesn't address any of the inner-level decision making problems, and the inclusion of the condition metrics can complicate the inner workings of the AI more than simplify them. The sample game could have ignored the Maslow approach and simply had the agents move to locations at scripted or semi-random intervals, drastically reducing the development time of the game while arguably not sacrificing much in return.

The Maslow approach is not proven to be without merit, however. The game and the experience it attempts to provide need to suit the design goals of the Maslow AI in order to be used properly. Taking the Maslow Hierarchy approach requires a game that puts realistic human behavior as a key aspect of its design goals. It is not coincidental that the games that inspired the project are also the ones that most benefit. Oblivion, with its focus on breathing world that the player can immerse themselves in, can make excellent use of the Hierarchy for the residents of its world and avoid the self-destructive and irrational habits of the Radiant AI. The experimental game itself functions very much like an simplified version of Radiant, tracking multiple statistics and making choices to improve overall well being and satisfying player goals. Oblivion and games like it focus on interacting with human-like agents on a regular basis as part of its gameplay, justifying the design expense of the Maslow approach.

CONCLUSION

Using Maslow's Hierachy as a design paradigm is still in a very early stage. The experiment acted as an example of how to use the technique and to show that it can be used to design a game AI. However the simplicity of the experiment, both in terms of its use of only the first two levels of Maslow's Hierarchy and with its bare presentation and mechanics as a game itself, does not give sufficient evidence of the effectiveness of this design approach. Implementation difficulties hurt the potential for the Maslow model to be used in simple games and many game genres. It still shows potential for games the take advantage of the psychological influence on its AI design, but there is no evidence yet to support its use in those games.

Discovering the potential of this new AI approach requires a more involved case study. Applying the Maslow design to a professional game that could utilize its advantages would provide the best case to observe the effects on development cost and added player experience. This scenario is nearly impossible to accomplish in an academic setting, so it will likely be up to the games industry itself to experiment with the concepts detailed in this paper. If nothing else, the idea of turning to psychology to improve game AI can be a valuable lesson for the future of innovative game design, even if Maslow's Hierarchy isn't the approach to handle it.

WORKS CITED

Griffin, E. (2010, November 14). Why Can't Open Worlds Be More Closed? *Bitmob.* Retrieved from http://bitmob.com/articles/why-cant-open-worlds-be-more-closed.

Maslow, A.H. (1943). A Theory of Human Motivation. *Psychological Review, 50,* 370-396.

Sanjeev, A. (2006, October 23). Radiant AI – This is awesome and freaky. Message posted to http://thoughtoutflux.wordpress.com/2006/10/23/radiant-ai-this-is-awesome-and-freaky/.

BIBLIOGRAPHY

Bernhaupt, R.. (2010) *Evaluating User Experience in Games Concepts and Methods*. New York, NY: Springer.

Dragert, C., Kienzle, J. & Verbrugge, C. (2011). Toward high-level reuse of statechart-based AI in computer games. *Proceeding of the 1st international workshop on Games and software programming (GAS '11),* 25-28. doi: 10.1145/1984674.1984684

Griffin, E. (2010, November 14). Why Can't Open Worlds Be More Closed? *Bitmob.* Retrieved from http://bitmob.com/articles/why-cant-open-worlds-be-more-closed.

Laird, J.E. (2002). Research in human-level AI using computer games. *Communications of the ACM, 45*(1), 32-35. doi: 10.1145/502269.502290

Lecky-Thompson, G.W. (2008). *AI and Artificial Life In Video Games.* Boston: Charles River Media/Cengage Technology.

Livingstone, D. (2006). Turing's test and believable AI in games. *Computers in Entertainment, 4*(1). doi: 10.1145/1111293.1111303

Maslow, A.H. (1943). A Theory of Human Motivation. *Psychological Review, 50,* 370-396.

Sanchez-crespo, D. (2003). *Core Techniques and Algorithms in Game Programming.* Berkeley: New Riders.

Sanjeev, A. (2006, October 23). Radiant AI – This is awesome and freaky. Message posted to http://thoughtoutflux.wordpress.com/2006/10/23/radiant-ai-this-is-awesome-and-freaky/.

Stene, S.B. & Yildirim, S. (2008). A survey on the need and use of AI in game agents. *Proceedings of the 2008 Spring simulation multiconference (SpringSim '08),* 124-131. Retrieved from http://delivery.acm.org/10.1145/1410000/1400565/p124-yildirim.pdf?ip=137.54.19.75&acc=ACTIVE%20SERVICE&CFID=70640482&CFTOKEN=91281662&__acm__=1322864127_42e9c530cbde387d4e95ae228d7aefdf.