Antopolis Report

By Johnathan August Fisher and John Luke Tyler

Executive Summary

In our project we decided to make a simulation of an ant colony. When we say a simulation we actually mean a game/simulation. This was achieved by making the game in a real time strategy format. A real time strategy game is the type in which you can select units and move them or make them do various tasks. In our game/simulation you control an ant colony. The part that makes it a game is the part that you can select and assign tasks to ants. The simulation is the part where the ants will sometimes do their own tasks and find their own food. The ants will also do "smart" things and gang up on enemies. Our first project was supposed to be a form of AI but that failed miserably as it was far beyond our skills. Since AI was out of question we decided to make a simulation. The reason it is part game is because we have been premature game programmers for about four years and because we thought incorporating the user to control parts of the game would be interesting.

Main Paper

"Can we make a premature artificial intelligence system simulating the life of an ant colony?" Now, while this may seem like an extremely daunting task to undertake, we felt that we were up to the challenge. Let me just say that we were very, very wrong. Not only was it harder to create a neural network so that our "virants" (virtual ants) could learn than we expected, but we were also going in blind to a project that we had almost no research about. In the end, we were not able to achieve the results we hoped we would, and could not, in fact, create a simulation of an ant colony to the best of our abilities.

We went into the programming of the simulation without any previous planning or research, which, we must admit, we will never do again. This was an extremely bad way to go about programming, as it kept us from finding errors and other such things in our code. We also didn’t have much of an idea about how ant life works, and usually guessed. That is, until, we did extensive research on both ant colonies themselves and something known as “swarm intelligence.” This helped us make our simulation much faster and also make our simulation much more accurate and detailed.

The results we gained from our project, however, will greatly help is in future projects that we may endeavor upon. We learned much more about one of our favorite programming languages, Game Maker Language (GML), and how it relates to Visual Basic (VB) and Star Logo (SL) in more ways that we thought. Unfortunately we did not gain as much information from our actual simulation as we would have hoped, but it helped us broaden our knowledge of the programming techniques and systems that went into creating simulations such as ours. One thing we did learn, however, was how to go about creating a better neural network so that our virants could learn faster and more accurately. Again, our actual programming was buggy and we found that our skills were not up to par to undertake a project as such, but we are still extremely happy we undertook the project. We have learned a few important lessons such as, research before you attempt to program.

Ants have always been a topic of interest for both scientists and researchers alike. Not only do ants have a developed self-organization system, but they also serve as a great study guide for many of today's computational science projects. These fairly "simple" creatures have led to many discoveries not only in the biological world but also in the algorithmic world. One of the most prominent and popular outcomes from studying ants is known as the "Swarm Intelligence."

Swarm Intelligence can be basically thought of as such: Self-organization through direct and indirect interactions. Now, to understand this, one must understand that humans generally have a "centralized mind set;" meaning that they would like to assign the order and coordination of activities to a central command. However, social insects, such as ants, think in a much different way. They build vast networks and "bridges" that other insects of their kind may follow, and with each insect following a trail like this, the trail becomes much stronger.

This brings us to the topic of self-organization. With self-organization, the behavior of the group is often unpredictable, simply because of the many collective interactions with all of the individuals. When one walks out and sees an ant traveling across the ground in search of food, it is believed that the ant is just randomly moving in various directions. This is definitely true, but it is a good way to find food and other such resources. Both randomness and errors help the ant in finding food and resources, and when this food is found, it lays a trail back so that other ants can follow it. So, unlike having a "master" controlling everything, the ants self-organize themselves so that the job can get done much quicker. This creates a certain "mind set shift" within each individual, since instead of them proceeding to do a much bigger job each individual does a much tinier job. The more individuals, the quicker the job gets done.

Now, wait a minute, wouldn't this kind of system result in less control and a better possibility for things to go *out* of control? It is not impossible, to say the least, but if everyone is working together and in the same mind set, then it will run much more smoothly. This swarm intelligence tactic can be used to save thousands to millions of dollars within companies, but many of the managers of said companies have a problem accepting this swarm intelligence. Eric

Bonabeau, one of the leading researchers of swarm intelligence, can be quoted saying: "My experience trying to 'sell' the concepts of swarm intelligence to the commercial world is that managers would rather live with a problem they can't solve than with a solution they don't fully understand or control. So the mind set is a big barrier to adoption." However, he goes on to say that if we, as humans, do not eventually accept this swarm intelligence, then we are, in short, screwed. It is not simply possible to control using traditional, centralized, hierarchical command where there are billions of dynamically changing and communicating entities. Eventually, as more and more entities are added to the fray, one person will not be able to manage it all. This will lead into a system of chaos, and then swarm intelligence must be accepted or it could very well lead to a company's destruction.

The main goal of most current research on swarm intelligence today is to find a way to make it so that the self-organizing process will not get out of control. It is quite possible that the self-organization could get out of control and lead to a dangerous system of self-preservation and chaos, but studies being done are trying to help limit this possibility. Before self-organization can be fully implemented on a worldwide scale, it must first be tested in a much more controlled environment, such as that of a computer system. Unfortunately, simulations can only be so accurate, and it is quite possible that there still will be problems with it when it is more accepted in the world.

Now, to switch gears, I will dive into the way that ants survive in our world. Ants have always been seen as one of the best portrayer of swarm intelligence, and that award has been correctly given. Not only do they operate extremely well using self-organization with billions of individuals, but they also seem to have one of the best and most maintained systems to date. Each ant works for the better of the colony, and in turn each gets an equal share of position.

Collecting food, one of the most important duties that an ant may have, looks to be a seemingly easy task for ants. One ant may go out, randomly skitter around for awhile, and when it hits food, instantly rush back to the hive, leaving a pheromone trail behind. As other ants pick up the trail, they in turn travel it and proceed to leave their own trails, making the trail much stronger. Then some of the ants may travel the trail and find a much shorter and faster way to get to the food, thus laying down a speedier route for all ants to travel by. This makes a "river," so to speak, of ants that are all going to a definite food source, which helps collect the food much faster. In the end, it seems that no matter where ants are, they will always be able to find a source of food and nourishment, however far from the hive they may be.

Ants, being insects, develop by complete metamorphosis; passing through both larval and pupal stages before they finally reach adult hood. The larval stage is completely helpless and must be tended to. Food is given to the larvae through a process called trophallaxis, which involves an ant regurgitating formerly digested food to feed the young ones. The difference between queens and workers is determined at this stage of feeding. One thing that must be monitored when an ant is developing in its larvae and pupal stages is its temperature, which is maintained by moving the different larvae and pupae to different "caring stations" within the brood. Finally, after almost all metamorphosis is complete, the ant will begin to grow as it works. In some ant species, there are different “classes” of ants; which are minor, media, and major. Minor class ants do most of the foraging, while the major class ants do more of the digging and protection, and media is somewhere in the middle. Sometimes, the media class isn’t even achieved by the ant colony, so the minor and major classes have an extremely evident difference. The developing stage may be one of the most important stages in an ant’s life, as it chooses exactly what they will do for the rest of their life.

A simulation isn’t an easy thing to do for someone as inexperienced in programming as one if us. At first we didn't record this fact and we wanted to make a crack at it anyway. As we soon discovered, simulations are far beyond our programming skills and because of this new fact we decided to make it a game/simulation in which the user could only control some of the parts of the game while the ants would work on their own and together to do other tasks in the game that an ant colony normally would do. The game part seemed like a good idea because we have been programming games for about three years now. As far as the game goes, we had decided it would be a real time strategy game. The main reason we wanted to program a real time strategy game is because it fits perfectly with the task we are trying to achieve. Since we were making a game we needed a programming language that would work well for a game. Thus we chose Game Maker Language (GML). We have been using GML for about three years now and it is by far our favorite language to use for games. GML is a language that is embedded with a program known as Game Maker. Game Maker is a lot like most visual languages but a little bit easier to use. It also object oriented which helps a lot when making games. Although GML isn't a very widely known language, it is still very efficient and fun to use. It works like any other language using syntax and functions. Game Maker also has two ways of functioning. You can use the drag and drop functions that have pieces of code inside of them that does small basic functions. The drag and drop functions are for beginners because they can't do very advanced things. The other method is to do it by code. The code is for the more advanced users because it requires all text. The coding is much more powerful as it can use all of the functions of GML. Using the code is a lot better because you can do much more with it. The drag and drop has limited capabilities compared to the code. One of the greatest things about GML is that it can import DLL's and files that were written in a different language. This is helpful because GML is definitely a limited language when it comes to really advanced and powerful stuff. There are a lot of DLL's that were made by people to extend GML's capabilities. Some one can even make their own.

Since our project was kind of a failure we had decided not to try and make artificial intelligence. Instead we decided to just try and simulate ant life. Our main goal was to make a path finding system so the ants could go and follow "pheromones" to their destination. The main reason we wanted to do this is because that is one of the really unique things about ants. We had made a decent path finding system that worked for what we wanted to do. Of course it wasn't perfect but it worked and that is precisely what we needed. We also tried to make our virants "smart" by making them react to things different ways given different circumstances. The ants won't fight if they have no chance or if they are greatly outnumbered, instead they will run back to the anthill and get backup. As this is a smart thing to do it also has some flaws because the armies will sometimes get in huge battles and then retreat after there is no point of retreating. Although our virants make stupid mistakes they are still just as dumb as any other robot because without AI, nothing that is robotic can actually be intelligent. Robots have to be told exactly what to do or else they will, inevitably, lock up.

Acknowledgments

Steven Schum- For helping us all the way through with our AiSC project and for teaching us a lot of the programming concepts we used.

Irene Lee- For opening our eyes unto the world of swarm intelligence and giving us some definite topics to research.

References

1. Ant- Wikipedia, the Free Encyclopedia. 02 April 2005. 04 April 2005 <http://en.wikipedia.org/wiki/Ants>
2. Bonabeau, Eric. Interview with Derrick Story. 02 February 2003. 04 April 2005 <http://www.openp2p.com/pub/a/p2p/2003/02/21/bonabeau.html>
3. Overmars, Mark. Game Maker Documentation. October 2004. 04 April 2005 <http://www.gamemaker.nl/doc/html/index.html>
4. Eberhart, Russel, James Kennedy, and Yuhui Shi. Swarm Intelligence. 11 July 2002. 01 April 2005 <http://www.swarmintelligence.org/SIBook/SI.php>
5. The Source for Information About Ant Conlony Life and Founding. 2002. 01 April 2005 <http://www.antcolony.org/>.