

INNOVATIONS PROJECT “ARTIFICIAL LIFE SIMULATION” IMPLEMENTED IN MASSIVE

Peter Seager

Index

1. Introduction
 - 1.1 Summary of Project Aims
2. Literature Review
 - 2.1 A Review of Artificial Life
 - 2.2 Other Projects
 - 2.3 A Review of Massive
 - 2.4 What is Fuzzy Logic?
 - 2.5 A Review of Applications of Massive in CG
3. Project Implementation
 - 3.1 Structure of Simulation
 - 3.2 The Environment
 - 3.3 The Plants
 - 3.4 The Animals
 - 3.5 Agent Design
4. Results
 - 4.1 Comments of Animations Produced
 - 4.2 Explanation of Behaviours and Interactions Between Agents
 - 4.3 Overall Performance and Limitations
5. Conclusion
 - 5.1 Future Work
6. References
- Appendix 1
 - A1 How the Networks Work
 - A1.1 How the Animals Work

This project was aimed to create something innovative within a piece of software. The software chosen was Massive and the creation was to be an Artificial Life System.

1. Introduction

The original aim of the project was to choose a piece of existing software, take it one step further and use it in a way which it was not originally intended for. For this to work it needed to be a relatively new piece of software that had not yet been pushed to its full potential.

Massive appealed to me because it was specifically designed for one purpose, crowd simulations. This made me think about what else you could use it for. Although I had never used it before, I was aware it was a very powerful piece of software and capable of producing brilliant results within its specific field (see section 2.5).

Whilst working through tutorials the possibilities and limitations soon became apparent. It was very good at creating mass animation for crowds, but simpler tasks were sometimes challenging. This made it more of an appealing project because I truly felt I would be pushing Massive to its limits outside of crowd simulation.

After researching into previous uses of Massive (see section 2.5) I decided upon creating an artificial life system. So far the software has been used to create a huge variety of living creatures and to me the next step on from this would be to create them so they can live continuously based on an eco-system created by the user. They would then feed and drink based on the available resources in the system. To my knowledge nothing like this has been attempted before and it would be interesting to see creatures come alive, live, and die all within the same simulation.

I decided to concentrate more on the simulation side rather than looking at evolution of creatures and their adaptability. By doing this I would be able to look at the interesting behaviours produced from simple user defined rules.

I chose this route because evolution algorithms have been successfully produced in the past (see section 2.2). The aim of my project was to push Massive on step further. By keeping the project simpler it will act as more of a frame work to show what is achievable and prove something much more advanced could be produced based on the same theory and methods.

Coloured primitives will be used to represent living organisms within the simulation. Massive has created perfect animation which has been integrated seamlessly into effects shots in the past (see section 2.5), so development of the simulation would be much more worthwhile.

After I began to learn how Massive worked the idea became more of a viable option. The software is based around an agent system where each different type of character has its own brain, skeleton, and body. The agent can then be placed as many times as you wish into a scene. It can then be simulated and you see the agents and instances of agents interact with one another, and act based upon the brain and scenario given to them by the user, via the viewport.

Early on it was decided the simulation will be purely implemented with the nodes available within Massive. I decided on this approach because otherwise I could have programmed the simulation, something which has been done many times, and to a greater extent than my intention. The idea behind the project is not to produce a revolutionary system, but implement it in a different way.

1.1 Summary of Project Aims

- To show an artificial life system can be implemented within Massive
- Make the simulation a good framework that can be built upon for future work
- To replicate real life behaviours based on simple rules
- Create an animation showing this

2. Literature Review

2.1 A Review of Artificial Life

Artificial life (or Alife)

*is a field of study and art form that examines systems related to life, its processes, and its evolution through simulations using computer models, robotics, and biochemistry.*¹

This seemed relevant to my study as I am trying to make a system which replicates real life. The goal is to replicate real life, but be able to eliminate the problems associated with studies using live bacteria or animals. Experiments that were previously impossible have been and can be carried out using Artificial Life simulations.

There are different forms of Artificial Life. These are Program-Based, where complex organisms are used which have a complicated DNA language, Module Based, where modules are added to a creature which modify it's behaviour, parameter based, where organisms have been constructed with predefined behaviours which are controlled by parameters that mutate, and finally Neural Net Based where the creatures in these simulations, using Neural Nets, grow with emphasis put on them learning as they go.

¹ Anon. *Artificial Life* [online]. Available from: http://en.wikipedia.org/wiki/Artificial_life [Accessed 21 January 2007]

In a description of Artificial Life², Chris G. Langton, describes it as

the name given to a new discipline that studies 'natural' life by attempting to recreate biological phenomena from scratch within computers.

Alife complements the traditional analytic approach of traditional biology with a synthetic approach in which, rather than studying biological phenomena by taking apart living organisms to see how they work, one attempts to put together systems that behave like living organisms

This was exactly what I wanted my project to do, attempt to simulate living organisms.

A quote I found was in a description³ by Chris Adami and Titus Brown, who said “Artificial Life is often described as attempting to understand high-level behavior from low-level rules”. After reading this I realised that to create a good simulation I would have to keep the rules simple, so they can be easily analysed, but have them produce complex results.

2.2 Other Projects

Tom Ray began in 1990 by conducting research into digital evolution and looking at the results gained when natural selection is implemented in the digital form. This work led to the creation of ‘Tierra’, “a system in which self-replicating machine code programs evolved by natural selection”. It creates a virtual computer and a Darwinian operating system which allows it to evolve.

In a description of ‘Tierra’ from Tom Ray’s website⁴ he describes life on Earth as

the product of evolution by natural selection operating in the medium of carbon chemistry.

However, in theory, the process of evolution is neither limited to occurring on the Earth, nor in carbon chemistry. Just as it may occur on other planets, it may also operate in other media, such as the medium of digital computation. And just as evolution on other planets is not a model of life on Earth, nor is natural evolution in the digital medium.

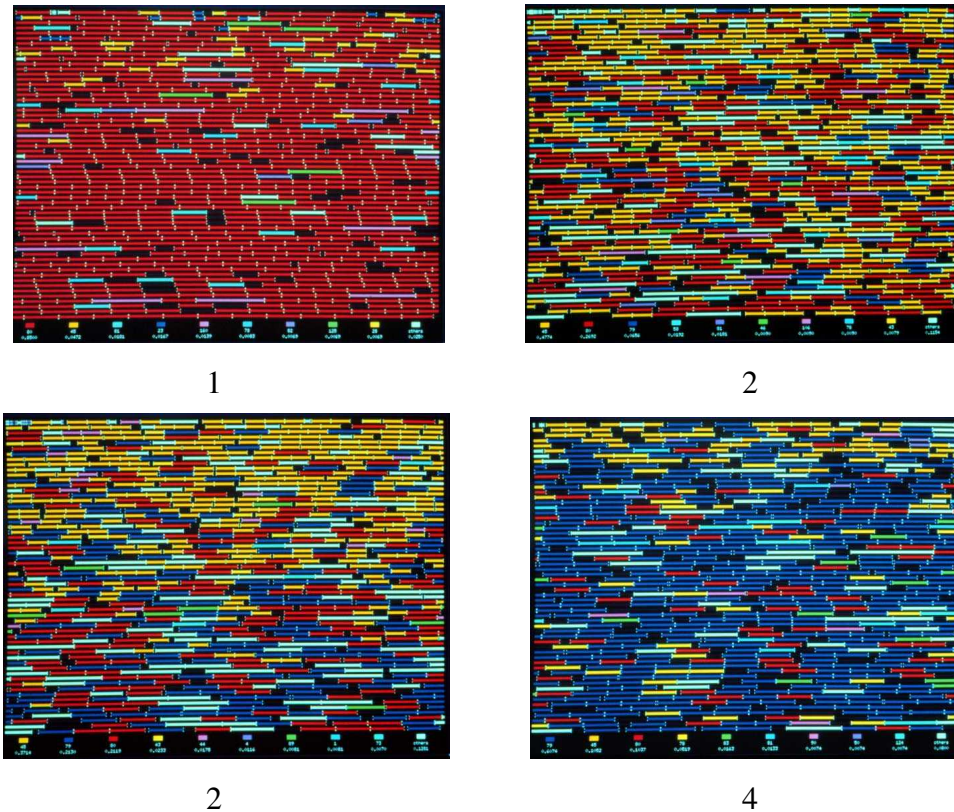
What I believe he is trying to say is that evolution is just the name given to a form of development, and this can occur anywhere, not just on Earth or on other planets, but in a computer. This is what ‘Tierra’ is especially good at. I think this is a very

² Langton, C G., *What is Artificial Life?* [online]. Available from <http://zooland.alife.org> [Accessed 21 January 2007]

³ Adami, C. Brown, T., *What is Artificial Life?* [online]. Available from www.alife7.alife.org/whatis.shtml [Accessed 21 January 2007]

⁴ Ray, T., *What is Tierra?* [online]. Available from www.his.atr.jp/~ray/tierra/whatis.html [Accessed 25 January 2007]

important concept and helped me to understand the thought process behind evolution within artificial life.



These were made (Mark Cygnus⁵) using the Artificial Life Monitor (Almond). They show an evolution race between hosts and parasites from the Tierra Synthetic Life program. Each image represents 60000 bytes, divided into 60 segments of 1000 bytes. The colours represent different creature.

In image 1, red (the host), it very common. But in image 2 you see the parasite, yellow, become more popular. Finally Immune hosts (blue) begin to take hold in image 3. And by image 4 immune hosts dominate.

Figure 2.1a

‘Tierra’ led me onto a system created by Karl Sims and his Siggraph Paper⁶ from 1994 ‘Evolving Virtual Creatures’. What his system does is create “virtual creatures that move and behave in simulated three-dimensional physical worlds”. It uses the genetic algorithm as a

⁵ Cygnus, M., *Almond Overview* [online]. Available from www.his.atr.jp/~ray/pubs/images/index.html [Accessed 25 January 2007]

⁶ Sims, K., 1994. *"Evolving Virtual Creatures"* Computer Graphics (Siggraph '94 Proceedings), July 1994, pp.15-22.

method for optimization. “A Darwinian ‘survival of the fittest’ approach is employed to search for optima in large multidimensional spaces”. This enables the system to create multiple number of creatures and then select the ones best suited to the environment and reproduce using these so a natural evolvement occurs. Previously user defined control systems were used, but with this system complete creatures are evolved, hence removing the need for user input.

What really fascinated me about this was the way in which the creatures are controlled. “A virtual ‘brain’ determines the behaviour of a creature. The brain is a dynamic system that accepts input sensor values and provides output effector values. The output values are applied as forces of torques at the degrees of freedom of the body’s joints.” This is the same way in which Massive works, which enabled me to see a similar example and that it can work.

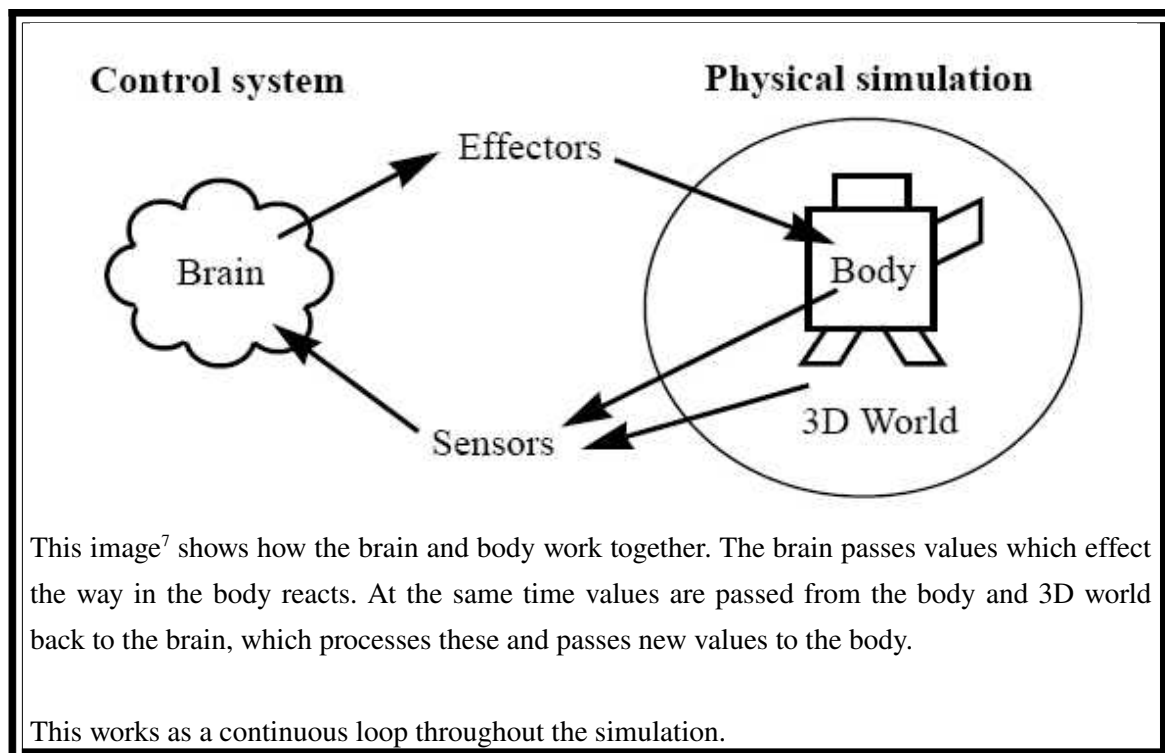


Figure 2.2b

An example of a Module based Artificial Life simulation (see section 3) is Techno-Sphere. This is a real-time 3D modelled world where users can release their animal. The simulation uses a combination of algorithms to decide which animals survive based on how the eco-system reacted and their interaction and components. It is also capable of simple evolution.

On a visit to the ‘The National Museum of Photography Film and TV’, in Bradford, in 2005 I saw this simulation first hand. Created by Jane Prophet and Gordon Selley in 1995 it was

⁷ Sims, K., 1994. [*"Evolving Virtual Creatures"*](#) Computer Graphics (Siggraph '94 Proceedings), July 1994, pp.15-22.

intended for the internet and ran until successfully until 2002, when it was terminated. Recently, in December 2006, it was released back onto the internet.

2.3 A Review of Massive

Originally it was written by Stephen Regelous at the request of Peter Jackson when he was making ‘The Lord of the Rings’ Trilogy. They struck a deal which would enable the software to be written for use in the films and Weta (the production house working on ‘The Lord of the Rings’), but Stephen Regelous would retain ownership, hence Massive was set up. At the outset with it was seen as a way to easily simulate crowd sequences for feature films and commercials.

It is a node based system where by connecting various nodes you can set rules for the agents to follow. Each node performs a calculation based upon Fuzzy-Logic (see section 2.4)

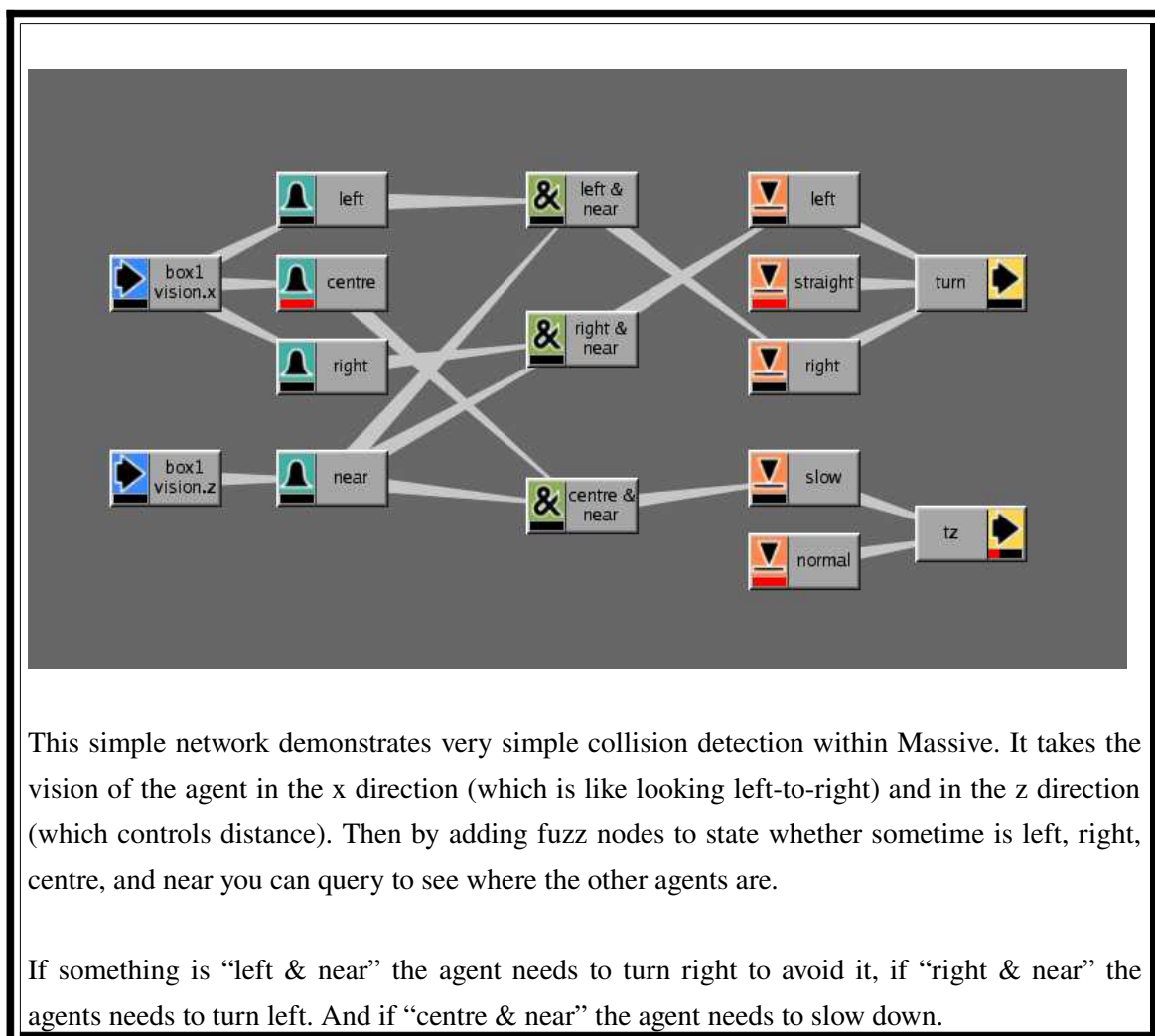
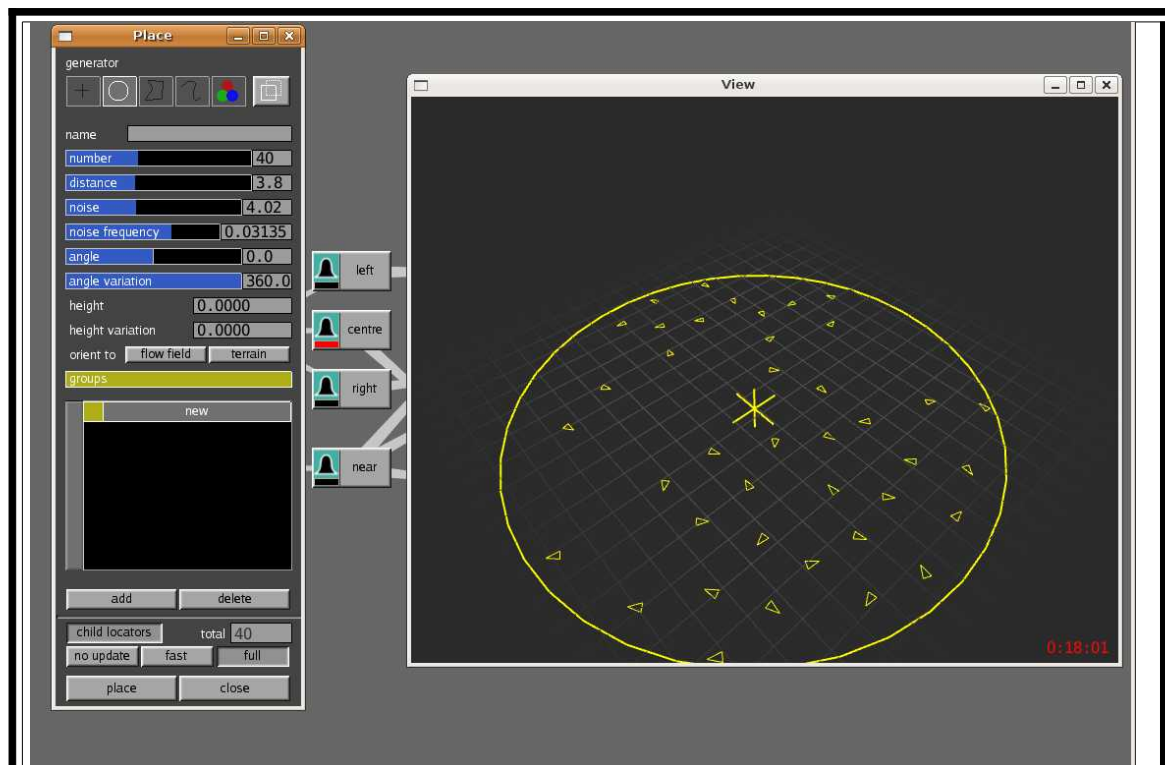


Figure 2.3a

Each agent has its own brain, which is made up of a network like the one above. In a scene you can have a multiple number of agents, and multiple instances of the same agent. The agents are placed using the place tool.



To place instances of the agent the place tool is used. There are different place tools. A locator, circle, custom shape, curve, or by colour. Once the method is decided upon there are a number of options to choose from, such as, number of agents, distance apart, noise, noise frequency, angle, angle frequency, height, height variation, and orientation.

By using a combination you can achieve anything from formation patterns to sporadic placement.

Figure 2.3b

When you run the simulation the agents will interact with one another based upon the rules you set them (See video “simpleExample.m2v” with digital and CD hand in)

2.4 What is Fuzzy Logic?

The general idea of fuzzy logic is to represent continuously valued inputs such as temperature as fuzzy values such as cold, warm, and hot, and then combine them in rules to control outputs. The possible output values are represented with fuzzy values, such as stop, slow, and fast.⁸

⁸ Anon., *Introduction to Fuzzy Logic* [Massive Documentation]. Available from nccastaff.bournemouth.ac.uk/jmacey/Massive/index.html [Accessed throughout project duration]

Taking the example of an air conditioning⁹ unit, if it is hot, then increase fan speed. Or in a more complete form:-

Inputs: *temperature*

Outputs: *Fan Speed*

Fuzzy values representing temperature: *cold, warm, hot*

Fuzzy output values for fan speed: *stop, slow, fast*

Rules:

If **hot** then **fast**

If **warm** then **slow**

If **cold** then **stop**

This can be seen in the above example (section 2.3).

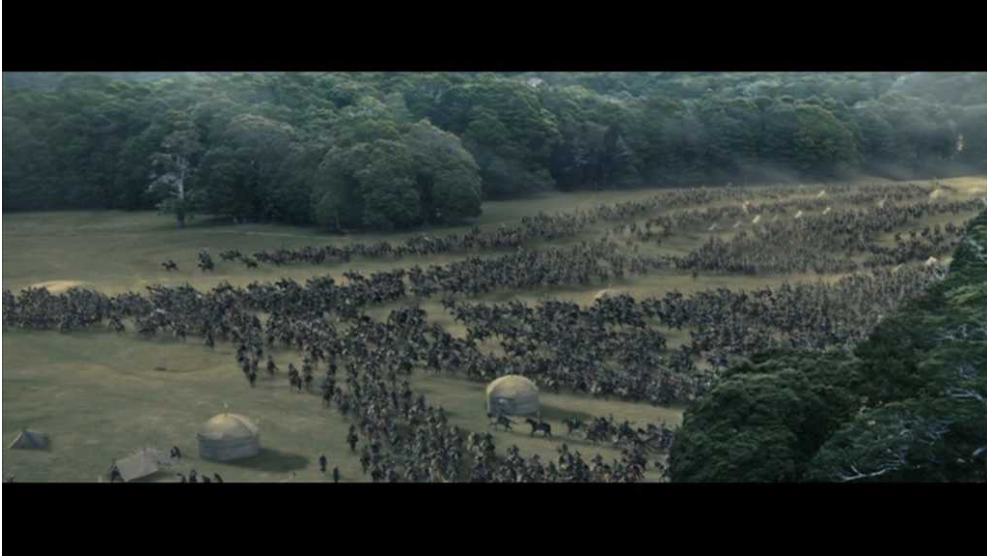
By using the logic in Massive controls can easily be added so agents gradually react without the need for expressions every time. The majority of calculations used in Massive are based around this method you can see what an advantage this holds over C/C++ style programming.

2.5 A Review of Applications of Massive in CG

The first commercial project it was used on was the Radiohead video for 'Go To Sleep'. It was then used on commercials such as the Nike advert 'The Other Game', and the Playstation advert 'Mountain'.

Dan Smiczek of Rhythm And Hues soon realised the software's potential. While they had already adopted it for use on 'The Lion, The Witch And The Wardrobe' he realised they could also utilize it for use on 'Electra' when a scene with hundreds of snakes was required. This was the first time it had been used in such a way and proved it is a very adaptable piece of software. After this it was used on such films as 'Poseidon' and 'X-Men: The Last Stand'. It was again pushed to another level, this time by Weta for 'King Kong'. As well as creating shots with crowds of people it was used to "generate cars, buses, trains...and circa-1930's traffic patterns". Also to "animate birds, insects...boats...and the people sitting in them to people on the streets reacting, falling down and running." This once again showed how it can be pushed to a new level.

⁹ Anon., *Introduction to Fuzzy Logic* [Massive Documentation]. Available from nccastaff.bournemouth.ac.uk/jmacey/Massive/index.html [Accessed throughout project duration]



1



2



3

1. In this shot from ‘The Lord of the Rings’¹⁰ – Return Of The King’. Massive was used to create the horses and riders.
2. This shot, also from ‘The Lord Of The Rings – Return Of The King’, Massive was used to create the Orks as they begin their assault.
3. And this shot, from ‘King Kong’¹¹, Massive was used to create the people walking the streets and the moving cars.

In all three of the above images Massive has been integrated into the scene seamlessly in previous projects, showing that realism is no longer the key with Massive. Something more revolutionary needed to be produced to once again show new uses for the software.

Figure 2.5a

While using the software again, but this time on ‘The Fast and the Furious: Tokyo Drift’, Dan Smiczek said “Massive is the only tool that lets us animate so many characters with this level of organic realism.”

At this stage I decided that creating a life system was the next logical step for usages in Massive. It was Dan Smiczek’s description of his characters having “organic realism” that led me to the idea.

3. Project Implementation

The type of Artificial Life I will be implementing will consist of Module based and Parameter based. There is a mix between the two because there will be a brain for each agent, which acts as the Module added to the life form. Then the parameters within that brain for the Parameter based part of my model.

3.1 Structure of Simulation

For the implementation I decided to take advantage of Massive’s agent system. This means each element (weather, herbivores, carnivores, and plants) can be set up as a separate agent with its own brain and body. Although they are their own object they will still interact with one another. This also makes it better for implementation and alterations as you can work with just one element at a time.

3.2 The Environment

I looked at the weather patterns for the United Kingdom and decided the system would be based on the south of England because there is enough change in the weather over a year without there being extreme changes.

¹⁰ *The Lord of the Rings*, 2003. Film Directed by Peter JACKSON. USA: New Line Productions

¹¹ *King Kong*, 2005, Film Directed by Peter JACKSON. USA: Universal Productions

	Spring	Summer	Autumn	Winter
Max Temp (degrees C)	13.0 – 14.2	20.9 – 22.2	14.4 – 15.5	7.9 – 10.3
Min Temp (degrees C)	4.8 – 8.2	11.3 – 13.6	7.2 – 11.2	2.4 – 7.0
Mean Temp (degrees C)	8.8 – 10.3	16.0 – 17.7	10.7 – 13.0	5.0 – 8.7
Sunshine (hours)	461 – 569	591 – 726	321 – 393	175 – 217
Rainfall (mm)	161 – 190	107 – 160	211 – 260	201 – 250
Days of rainfall >= 0.2 mm	36 – 40	28 – 33	37 – 41	46 – 48
Days of rainfall >= 1mm	26 -29	20 – 25	31 – 33	33 - 35

Figure 3.2a

The data in Figure 3.2a was taken from the Met office¹². It is the mapped averages on the weather in the South of England from 1971 – 2000.

I decided to implement the weather system based on the four seasons. This way I could keep it manageable, and you would be able to see the changes in season and weather, thus producing clearer results.

For the water levels there is a box which gets translated up for down based whether it is sunny, raining, or overcast. The terrain being used was modelled in Maya and has varying height, so when the water levels rise, there is still parts which are not submerged.

To transmit the weather there is a ‘sun’ for them to react to. In Massive, after a day of trying, I could not find a way to pass a variable from one agent to another. As the plants needed to grow or die based on the current short-term weather there needed to be a way to transmit the weather information. The agent vision seemed the most logical way to do so. It enables you to set objects colours then the agents can recognise the colour values. The ‘sun’ is included within the weather agent, and if it is sunny the colour changes one way, and if it is raining it

¹² Anon., 1971-2000 mapped averages [online]. Available from <http://www.metoffice.gov.uk/climate/uk/averages/19712000/mapped.html> [Accessed 5 February 2007]

will change in another direction. The plants will be fully grown if the value lies in the middle, then with either too much rain or too much sun they will gradually die.

3.3 The Plants

They will be based around a group of plants called 'embryophytes'. These the most common types of plant and include trees, flowers, ferns, and mosses. Like most plants they photosynthesis to gain energy, so growth is based around the weather. Their main habitat is on land so this will be ideal for the simulation.

The original plan was to use Massive's blend shapes for the growth of the plants. Unfortunately it did not seem to work on an agent with multiple instances of itself, so had to translate the plants along the Y-axis.

3.4 The Animals

The animals will be kept simple by using a carnivore and a herbivore. These are both quite primitive, but form the basis of a lot of animals seen in the wild. These will work well because the aim is to have a good basic simulation rather than spending time on unnecessary detail.

An animal develops behaviours in two ways, by instinct or by learning. For the simulation they will know their behaviour from instinct. This is the best method because the simulation will be set over a long period of time, so the animals will move quickly, and in real time have a short life span. To have the creature learn their behaviour would be more appropriate for a slower simulation as you would be able to closely analyse the results.

The behaviours to be implemented are going to be primitive. This will enable me to prove the theory that simple rules can produce complex results. In the wild most animal behaviour is very primitive. They live, eat, drink, and die. These are at the core of any living creature, whether it be an insect, mammal, or fish. By doing this it will help keep the piece a general example and a good foundation.

3.5 Agent Design

The weather agent is based on an overall timer that counts through a year, which lasts 120 seconds. This triggers the seasons, which are 30 seconds each, which triggers the day timer so there is a sense of night and day. This is necessary because when calculating the sunlight, it cannot be sunny twenty four hours a day, so during the night it can only be overcast or rain. Random values are then produced which decide if it is sunny, raining, or overcast. The amount of sun is based on the average hours of sunshine for each season in the south of England. The simulation will always take the amount of sunshine into account first. So if it is not sunny then it will either be raining or overcast. By basing it directly on the hours of sunshine it is easier to alter the weather pattern.

For the animals the first control added was to enable them to move along the terrain and avoided one another. Again this was done using the vision control. By giving the herbivores and carnivores their own colour, they can look out for it and by slowing down and changing direction can avoid one another. A similar technique was used for the plants and water, they were given their own colour, so for example, the herbivores are attracted to the colour of the plants and water if they are hungry or thirsty respectively. As they close in on the objects, by using the vision, they slow down as to avoid intersections. When they eat and drink a variable is used to store how much food and drink they have taken in. If they are not hungry or thirsty they will avoid the plants and water.

The carnivores were implemented in the same way, but look for herbivores when they were hungry, instead of plants. Due to this, a control was added to the herbivores which meant if a carnivore was feeding from it, it would stop and then die when the carnivore had finished eating.

See Appendix 1 for an example of how an agent is implemented

4. Results

4.1 Comments of Animations Produced

The final animation is a compilation of simulations produced, each with different numbers of Carnivores and Herbivores, to see differences in results produced. By doing this there is a range of varying results to study. I edited them together so it brings it viewers' attention to the interesting behaviours gained from the simulation.

4.2 Explanation of Behaviours and Interactions Between Agents

From the final video you can see a lot of interesting behaviours from both the Carnivores and Herbivores. At the start of the animation you can see them spreading out and searching for food and drink. All the time they are avoiding one another and the plants, unless the Herbivores are eating them.

The Herbivores avoid the Carnivores at a greater distance than they avoid the plants and water because they are a danger to them. As in real life the Herbivores rarely get away because they cannot move as fast as the Carnivores.

It is sometimes quite hard to see the Herbivores eating the plants as the simulation runs rather quickly, but when they are near to a plant and slow down, they are eating it.

The Carnivores eating the Herbivores is much clearer. When a Carnivore gets close enough to a Herbivore they will begin eating it, then the Herbivore disappears when the Carnivore has finished with it.

The other thing seen is the animals dying when they do not have enough food or water within their system. The dying is represented by the animal's disappearance. This method was used because they have no animation so simulating a death was not possible.

4.3 Overall Performance and Constraints

I think the simulation is good overall because it proves such a system can be implemented in Massive, so succeeds in its aim to be a framework for a much more advanced project. As stated in section 4.2, interesting behaviours which relate to real world scenarios have been regularly seen.

The main constraint is you cannot simulate it from Massive in real-time. I was hoping this would be possible so you could run the simulation indefinitely. Unfortunately Massive slows down after only a few calculations so it became apparent I would have to render out however many frames I wanted for the simulation.

Each time the simulation is run, the same results will be produced unless you alter the weather, agent placement, or agent numbers. This happens because Massive is designed to create the perfect simulation for your project, so the average user does not want different results every time.

Also there is no sense of reproduction at the moment, so eventually all the animals die, although the plants and environment live on.

5. Conclusion

The original aim of the project was completed by taking Massive and implementing an Artificial Life system, something which has pushed it one more step toward its potential. I think it acts as a good framework which could be built upon. It proves something much more complicated can be produced. To achieve more rules interaction needed to be added (see section 5.1).

The implementation has been achieved purely with the nodes available in Massive. I have been very successful in this, although for future development and due to some limitations I came across, some custom programming may be necessary to take the project to the next step.

As well as this, interesting and real life behaviours have been shown in the final animation (see section 4.2).

I think I was successful in my intentions for the project. I have achieved all my initial goals, but would have liked a simulation where the agents interacted slightly more. The animals do not stick together or interact with each other on a social level. It would have been good if the animals formed groups and stuck together, but this is something that can be explored as it would take a lot of research into animal behaviour.

Working with Massive has been a good learning curve as I have learnt a new way of programming and constructing rule sets. I will look at this approach more often when I am trying to solve problems in the future.

During the implementation of this project I came across a few problems. This included not being able to pass variables between agents. By stopping and thinking about what I was trying to achieve, I found another way around this. Problem solving for this project has been different to anything I have done before because I was not very familiar with the software before, so was often finding its limitations and needing to work around these.

I tried to implement reproduction but found it very difficult. Massive is designed to work with the number of agents you specify at placement, so trying to get it to alter this is very difficult. With some custom programming I am sure it is achievable, along with some of the other features I tried to add, such as altering weather patterns. To add custom programming would have been impractical because I was trying to use only Massive's nodes, and the time of the project would not have been enough to start looking into Massive plug-ins or alterations.

If I had time to do this project again I would try and produce an aesthetically pleasing piece. Although this was never my intention I think it would produce an exciting animation if you saw animated animals chasing after one another, eating, or drinking. Then you could have different animations for an array of actions. I would also look into including an evolving terrain with extreme weather. The terrain would very slowly evolve if the simulation was adapted for a longer period of time. This would replicate the Earth's surface more, but would also require the simulation to be set over a larger area. Extreme weather is becoming more common as the Earth's atmosphere is changing, so when creating an artificial life system based on the Earth I think it would be good to see how the creatures adapt to it.

5.1 Future Work

There is a lot of scope to expand this simulation as it is a framework for something much more advanced, but the main addition I would like to make is in reproduction. Although I made an attempt, the time of the project was not enough to get this included. By adding this, the simulation would last a lot longer, and could even run indefinitely. Also patterns of animal

behaviours would be seen, such as rising and falling numbers based on the resources and the other species.

As in section 5, social interaction would make the simulation much more interesting and you would be able to study the different groups the animals make, and how they behave as a group as well as an individual.

I think more species of animals and plants would also add to the simulation. Different species could have different amount of energy when used as a food source. Some the plants could contain poison, so the animals learn not to eat them.

Smaller features to be added include age (so the animals do not live until they run out of food or water, or get eaten), more plant independence (so the plants do not grow and die all at the same time), and drowning (so if the animals get submerged in the water they die).

6. References

Adami, C. Brown, T., What is Artificial Life? [online]. Available from www.alife7.alife.org/whatis.shtml [Accessed 21 January 2007]

Anon. Artificial Life [online]. Available from: http://en.wikipedia.org/wiki/Artificial_life [Accessed 21 January 2007]

Anon., *Introduction to Fuzzy Logic* [Massive Documentation]. Available from <http://nccastaff.bournemouth.ac.uk/jmacey/Massive/index.html> [Accessed throughout project duration]

Anon., 1971-2000 mapped averages [online]. Available from <http://www.metoffice.gov.uk/climate/uk/averages/19712000/mapped.html> [Accessed 5 February 2007]

Cygnus, M., Almond Overview [online]. Available from www.his.atr.jp/~ray/pubs/images/index.html [Accessed 25 January 2007]

King Kong, 2005, Film Directed by Peter JACKSON. USA: Universal Productions

Langton, C G., What is Artificial Life? [online]. Available from <http://zooland.alife.org> [Accessed 21 January 2007]

Ray, T., What is Tierra? [online]. Available from www.his.atr.jp/~ray/tierra/whatis.html [Accessed 25 January 2007]

Sims, K., 1994. "[Evolving Virtual Creatures](#)"
Computer Graphics (Siggraph '94 Proceedings), July 1994, pp.15-22.

The Lord of the Rings, 2003. Film Directed by Peter JACKSON. USA: New Line Productions

Appendix 1

A1. How the Networks Work

A1.1 How the Animal Agents Work

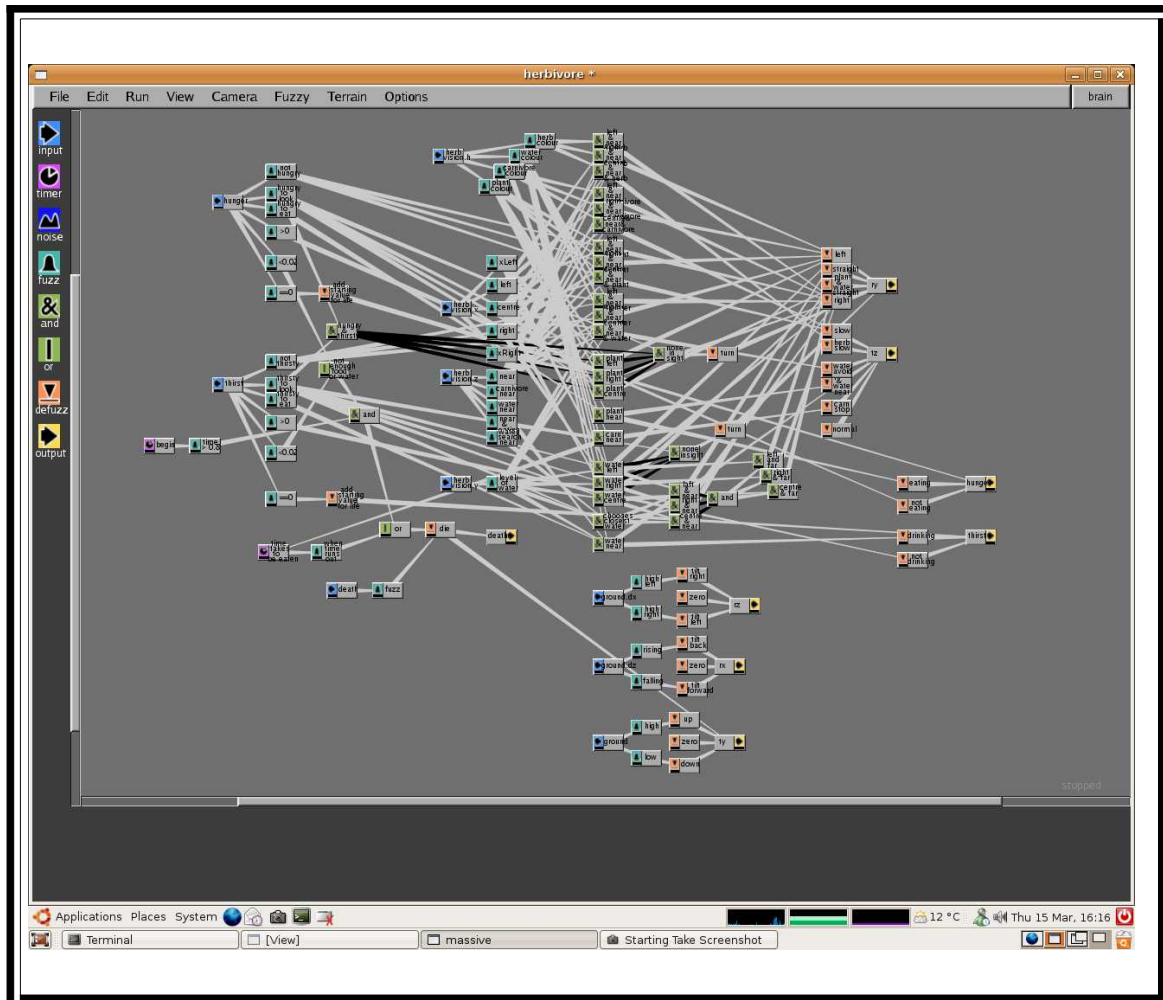


Figure A1.1a

The network for the Herbivore can be seen in Figure A1.1a. It is based around the vision for that agent and the agent's hunger and thirst levels. While the agent is neither hungry nor thirsty, it will move around avoiding the plants, water, and other carnivores. When its hunger or thirst reaches a predefined level and it becomes hungry or thirsty it will go in search of food and water.

If the Herbivore is hungry and thirsty, then thirst is the priority as an animal can survive longer without food than it can without water. It will find the water closest to it based on the direction and proximity of the colour related to the water. Once it has found this it will move

towards it, still avoiding other animals and plants. When it arrives it will stop at the water's edge and drink until its thirst is fully quenched.

It looks for food in a similar way, but instead of finding the closest watering hole, it looks for the closest plant based upon the plants colour and distance. Once found the Herbivore goes in search of the plant and feeds off it until it is no longer hungry.

There are two different levels of thirst and hunger. One where it will actively seek out water and food, and the other is if it comes across either it will eat or drink until it is fully replenished.

As well as this there is a stabilising section which keeps the Herbivore moving with the Terrain. This is necessary because otherwise they would move along the centre grid.

All the controls in the simulation work in a gradual manner. For example there is no one point where the Herbivore will become hungry. It will slowly become hungrier as the food value decreases.

The Carnivore is implemented in the same way, but instead of searching out plants when hungry they will hunt Herbivores.