Freaktures - An exploration of the Perception of Artificial Life.

Samantha Dickinson - c1400430

Introduction to the Project

This project has been an informal exploration of the creation and human perception of digital life. I have created a supporting program in order to try out some ideas and expand my knowledge of programming.
So far in the program I have created basic digital creatures called *Freaktures* which move around the screen making decisions on where to go based on random values. I have purposely left the Freaktures without any textures or secondary animation because, based on my friends reactions even early on in the project, this seems to encourage a more natural, uninhibited use of the imagination. I found that even though I have not, as yet, coded any actual interaction between the Freaktures, people would assign human attributes and emotions to them and believe that there was some interaction going on. Based on research into other games and from the kinds of things my friends believed the Freaktures were doing, I have come up with a number of ideas to try out in my program. It is very much a work in progress which I have found very interesting and I fully intend to continue working on it in my own time.
I have concentrated on coming up with and exploring ideas, and finding my own ways of approaching and solving problems I encountered.

Introduction to the Report

This report is intended to reflect the informal nature of my project and to attempt to communicate how interesting and enjoyable I found this experience. As such, I felt an informal, perhaps light-hearted approach to presenting my report would be best. I have created a sketchbook style report which contains my full report and annotations, thoughts and ideas I have come up with during the course of my project. If possible, I recommend you read the physical version.
This is the digital handin version of my report. It contain only the text, minus my comments, annotations, diagrams and images that are in the original version.
The source code and executable files for my project have been included in the digital hand in and are on a disc which can be found at the back of this book.

Initial Inspiration

TOSHIO IWAI

During a presentation we were shown a video of Toshio Iwai's1992/1994 exhibit, *Music Insects.* I found this very interesting because of it's rather unique view of music-creation. It gives the user a visual way to create their own musical compositions, regardless of their level of experience.
The user chooses four insects from a selection of sixteen, each representing a different musical instrument. They can then select different colours and draw on the screen using a mouse, as in a

paint package. The insects move around the screen and react to the colours by either changing direction or playing a note. Certain colours represent different notes on the musical scale, others change the way the insects move. You can draw randomly to get random sounds, or you can draw in a more structured way to create something that sounds more like music, to add beats and melodies.

I find Toshio Iwai to be a very interesting artist - partly a visual artist, partly a musician, partly an inventor. He has created a large number of interactive pieces, including installations/exhibits, live Computer Generated TV programs and video games.
His work largely seems to concentrate on the integration of visual displays in both the experience of listening to and the experience of creating music - allowing a new, visual approach to composing and a whole extra dimension to the enjoyment of a piece of music.
I was most interested in the way it encourages you to experiment and play with what's available. It looks like a lot of the music creation can be down to "happy accidents" and the piece can be totally different each time!

Toshio Iwai has created a number of very interesting interactive works, I have included some links that I found useful or interesting. A lot of the information is quite old, but still relevant.

http://en.wikipedia.org/wiki/Toshio_Iwai

http://ns05.iamas.ac.jp/~iwai/

http://museum.doorsofperception.com/doors1/transcripts/iwai/iwai.html

ELECTROPLANKTON

I was filled with joy when a friend showed me *Electroplankton* on the Nintendo DS. The game was pure, utterly delightful! It was made for playing around with and having fun, like a toy. No plot. No in-depth characters. No predefined goal. Just a selection of “plankton” creatures for you to choose and play with, poke and prod – with the wonderful “side-effect” of creating rather pretty musical compositions. Of course, this was the intention of the creators of the game, not simply an unplanned side-effect, but as far as the person playing is concerned that's all it needs to be – a side effect of their interaction with the plankton. The notes and sounds have been chosen to allow harmonious compositions without the user needing any musical experience. I was pleased, but not entirely surprised to discover that Toshio Iwai was also behind this fantastic little game. It seems to be a combination of, and an expansion upon, his earlier works, *Music Insects* and *Well of Light*s.

The game is split into ten sections, each with a different type of plankton which you interact with in different ways. In some you can only affect the environment, in others you can directly affect the plankton by spinning, prodding or drawing paths for them to follow. Often, the music is created just as much by the plankton's actions and decisions as those of the person playing the game.

The official UK website:
http://electroplankton.nintendods.com/flash.html

http://en.wikipedia.org/wiki/Electroplankton

BOREDOMRESEARCH

Before starting my project, I met with Vicky Isley and Paul Smith of *boredomresearch.* They build "observable phenomena of intrigue and beauty" using computer simulations based on sets of rules which are carried out in a completely automated fashion – with no external interference. The systems they create are mainly for display in festivals and galleries, in the form of self-contained installations.
The screens of the installations are a kind of portal into the digital world of the creatures, which allows the viewers to see the world but not do anything to affect it. Interestingly, however, my conversation with Vicky and Paul revealed that often people would expect them to be touch-screen interactive and press different areas until something happened which seemed to correspond to what they had just done. The viewers would imagine that they were interacting and affecting the world of the creatures when, in fact, they weren't.
Everything about the creatures created and the lives they live is automated, although strict rules are set out and must be followed. The environments, the creatures' behaviour and even patterns on the creatures themselves are generated algorithmically.

***Theatre of Restless Automata*:**

randomSeed

In this system, there are a number of simple rules to be followed by tiny machines which move around on the screen. When left running, the machines create intricate, infinitely variable images based upon the rules set down for them. Each time the system is restarted, a new and completely unique image is created.
To begin with, the rules were followed and plotted on graph paper, just to see how they could be followed and the possible outcomes. They were then translated into programmable algorithms which became randomSeed.

Ornamental Bug Garden

Each of the elements of the bug garden are created algorithmically before being composed as a final working piece. The plant-like structures in the background are procedurally generated, under the direction of it's creators, as are the patterns on the bugs that inhabit the garden. The bugs behaviour

is defined by a set of rules which are designed to create an illusion of life, rather than to actually re-create life itself. They try to spread out along the bottom of the screen, but in doing so they get pulled up on "wires" until they either get knocked off or reach the top and are forced to jump.

This piece also includes collision triggered sounds, which are randomly selected from a set of notes which would make up a chord. Just as *randomSeed* creates automated images, this creates accidental automated sound pieces which are different every time, but always in harmony.

boredomresearch are quoted as being "fascinated by how simple rules can be combined to create systems that exhibit complex behaviour." I totally agree. I also believe that a lot of the complex behaviour could be in our interpretation of the more simple behaviour defined by rules.

Biome

The Biome is a small circular window which looks into a self-contained world of moving entities. The area they exist in is huge, stretching far beyond the limits of the screen. The creatures seem to swim around, moving on and off the screen - it's quite possible to see an entity once and never again. The patterns and forms are again created algorithmically, giving almost unlimited variations. As the program is left running, increasingly complex patterns develop as any creature who's pattern lacks complexity is forced to reload it's pattern generator.
This system also includes sound, the creatures have a set of calls that correspond to certain actions and interactions with other creatures. They can only be heard when they are within a short distance of the viewing area. As several Biomes are set up in a room, viewers can hear the calls from other Biomes and often dash over to try to catch the creature that made the call before it disappears.
The creatures are programmed to react to the presence of others. For example, if one makes a threatening sound, others in the area might switch off their lights and slink away, or they might attack the offending creature with little sparks.

I found boredomresearch's projects very interesting, I liked that they were fun and intriguing even though there was no actual user interaction involved. It also showed that even very simple shapes representing the entities can provoke responses from an audience, I wondered how far this could be taken. I was surprised that such seemingly complex behaviour can be controlled by such simple rules. The size of some of the environments and the level of automation was quite impressive. Although this was not exactly the idea I was aiming for with my project, it was definitely very useful and gave me some great ideas about interactions between my own creatures. It also started me thinking about limiting the user's ability to interact with my creatures, if at all – to just give them something to watch and perhaps influence but not control.

More detailed information can be found here:

www.boredomresearch.net

SONIC ADVENTURE 1 & 2

Or more specifically, the sub-game called the *Chao Garden*. Although the actual games are very fun and a well executed move into 3D for the crazy, fast-paced Sonic games - I spent a lot more time raising little creatures called Chao in the Chao Garden than playing through the main story. While playing the main game, you can find animals and powerups to give to the Chao, and they evolve to look and behave differently based on which ones you give them and which characters you use to look after them.
So, the part of the game which doesn't have a pre-defined plot or story to follow actually provided me with the longest lasting fun. Interesting...

This research helped me to decide on my approach to this project and to give me a lot of inspiration and ideas on things to try out.
To begin with, I was looking at making a visual music creation program similar to *Music Insects*, but later my focus shifted to creating automated images and then finally to looking into the actual creatures that would be making these images. I have not included any sound in my project, but would not rule it out at a later stage.
I love that games can be incredibly fun without the need for a high level of detail in either the graphics or the plot. I find it a bit of a shame that so many people making games seem to overlook this. I believe that if the reason for playing is simply to experiment and enjoy then the detail need only be in the possibilities created. That is one of the driving *ideas* behind my project.

Goals

My original aim for this project was to create an intriguing and fun mini game involving small creatures moving and interacting in a basic environment. This broke down into the following goals:

- To explore the creation and manipulation of digital creatures with emergent Artificial Intelligence.
- To have the creatures interact with each other, their environments and objects within those environments.
- To eventually add user interaction, either with the environments or the creatures themselves.
- To learn more on the theory and use of the C++ programming language.

During the course of the project it became apparent that these goals would need to be broken down further, with clear but flexible targets to be set. I became more concerned with exploring ideas and learning things for myself. I decided it was important to learn my own creative approach to solving problems, particularly with programming, and to ask for help when I needed it with my own methods, rather than trying to take bits and pieces of other people's methods and “crow-bar” them into my own work and way of thinking.
This project is far more about the journey than the destination. As such, I decided it would be best

to keep my overall aim in mind, but to keep the details flexible, allowing for changes in the actual content and production of my program and setting the goals for each stage or problem as I encountered it.

My goals shifted and expanded. My overall aim became to observe and explore the way we as people perceive life where there is none, and to come up with ideas on how to incorporate this into an intriguing and fun mini game. An important aspect of this was to begin creating the mini game, in order to try out my ideas while learning in a practical way about C++ programming.

My specific goals were as follows:

- To learn to use C++ more effectively and come up with my own solutions to problems I encountered.
- To create simple shapes representing the Freaktures which randomly position and orient themselves within set boundaries (i.e. The edges of the window).
- To have the Freaktures move around within these boundaries.
- To produce different types of movement which can later be used to represent e.g. The mood of the Freakture.
- To create coloured trails for the Freaktures.
- To implement changes of size or speed, to be later used in conjunction with other factors such as eating food or reacting to other Freaktures nearby.
- To observe people's interpretations of the actions of the Freaktures and come up with new ideas based on their perceptions.
- To then add actual interaction:
    - Firstly to get the Freaktures interacting with each other, based on mood or personality.
    - Secondly to get the Freaktures interacting with the environment or objects within the environment.
    - Lastly to include some user interaction.
- To allow the death of old Freaktures and the birth of new ones?

Achievements and thoughts

Learning C++

While I have by no means learned everything about using C++, this project has definitely improved both the extent of my knowledge and my ability to apply that knowledge. I find it a lot easier to understand what's being said in lectures and even know about some aspects before they are covered in the lectures. I now find I understand enough to make finding out more about what I don't understand less of a problem. I seem to have a better understanding of any error messages I get and how to fix them, and feel I have also come a long way in working out logical errors. This is, however, an area that I know I still need to work on and there are still some bugs that need to be fixed or at least explained before I'm happy to move on to the next stage of my program.
I have been experimenting with the use of classes, a new concept to me, including the vector

template class. This was very useful for storing my Freaktures so that they could be easily traversed and updated with the possibility of creating new ones and deleting old ones during run-time, should I so wish.

I'm sure that my program isn't the most efficient piece of coding ever, nor is it the neatest. This is largely down to the experimental way I approached the programming side of the project, and given the simple nature of the program I don't feel that, at this stage, it's too much of a problem. Efficiency and neatness came second in my priority list to experimenting and having fun, this worked for me and I found it an effective way to learn new things. I have been implementing my own solutions rather than using other people's solutions to similar but fundamentally different problems.

Create and initialise the Freaktures

The first step was to create a Freakture. Just one, and assign it positional and rotational values. Next, I had to work out how to create any given number of Freaktures, and assign values to their positions and so on. I created a Freakture Factory class in which I could create and initialise a whole bunch of Freaktures, stored in a vector template class.
Once I was sure that these were working, and that I could update or retrieve the information as and when I needed to, I set about giving the Freaktures a visual form.
I created a Renderer class to handle all of the drawing. Each Freakture was drawn as a small quad which was more pointed in one corner to show it's orientation. Putting the positional information of each Freakture into the drawing loop, I made the little quads appear in random positions and with random orientations on the screen. The starting positions were set to be within the limits of the window.

Movement

I have given the Freaktures two types of movement, one moves in straight lines and makes sharp turns at random intervals, the other has a much smoother curve which is randomly changed every so often.
Before adding in the timer, the movement was very erratic and frenzied. It was very funny to watch, but also slightly irritating as it was hard to focus on each Freakture. I added the timer to allow me to define how fast the Freaktures would move and make the program a bit more consistent if viewed on different machines.

Setting a heading.

The X of the heading is given by the cosine of the angle.
The Y of the heading is given by the sine of the angle.
These values then had to be normalized in order to get the actual heading.

Normalizing

To normalize the heading you need to find the length of that heading by finding the square root of the sum of each component (i.e. the sqrt of (x*x)+(y*y)).
Next, you need to divide each component of the heading by the length of the heading.
(See sketchbook version)

(Freakture: My implementation of this still seems to be a bit buggy, but I'm working on it.)

Sharp movement.

This function generates a random number between 0 and 1. If it is below a certain value (which can vary depending on how often I want it to change), the Freakture generates a new rotation and uses it to set a new heading. Otherwise, it carries on with the heading it already has. The turning is instantaneous, so the Freakture looks like it's darting about.

Smoothly curving movement.

In this function, the random angle is generated in a similar way to before, but is used to set a target heading instead of the current heading. The current heading is set over time by interpolating between the previous heading and the target heading using the linear interpolation equation:

$$\mathbf{P=P1 + t(P2-P1)}$$

Where **P1** and **P2** are the points to blend between and **t** is the blending factor, a number between 0 and 1.

Of course, once the Freaktures started moving, they were no longer bound by the edges of the window as they were when initialised. I tried two methods of fixing this:

Reversing the headings.

I ran a check on each Freakture to find whether or not it was within the viewable area. If it wasn't, it's heading was reversed, making it head in the opposite direction. However, due to the random nature of the Freakture's movement, the fact that each Freakture could decide at any given moment to alter its direction meant that, while they generally stayed within the confines of the window, eventually they would all leave. This, I believe is because it was possible for them to move outside of the viewable area, then have the heading reversed as planned, but before coming back into view, change direction so it would be reversed again and head away from the viewable area and so on until it was too far away to ever get back.

One attempt to fix this caused the Freaktures to zoom off the screen upon nearing the edges. This was very interesting because when I showed my friends they felt that the little dots were in danger of being sucked off the edge against their will! They would cheer when one "narrowly avoided" getting too close to the edge and cry out when another was pulled off. It was quite tense when there was only one left and it was getting closer and closer to the edge!

Head to centre.

As the first method didn't seem to work too well, I came up with another one. Instead of just reversing their heading, but leaving their behaviour the same, I decided to switch to a different behaviour which would make the Freaktures unable to randomly change course while outside of the boundaries. Instead they would head to the centre until they were back in the viewable area, where the normal method of movement was reinstated and they were able to turn freely.
To do this I used the equation:

D=P2-P1

where P2 is the centre and P1 is the current position.

Once normalized, this gave the heading to be used to make the offending Freakture move towards the centre.
Although these values have been normalized, the Freakture seems to speed up when it heads towards the centre. Brilliantly, viewers have interpreted this as it making a decision to jump back off the edge. It's quite a happy accident, and a really nice effect which I would like to incorporate in the final program.

Coloured Trails

Upon creation, each Freakture is assigned random colour values to be used for the trail. As each Freakture's position is updated, their last position is pushed onto an array. All of the values in all of the arrays are then used to draw large dots of the relevant colour to the screen. These dots are close enough together to appear as a solid line.

Similar to the idea of BoredomResearch's *randomSeed*, I wanted my Freaktures to be able to draw unique pictures as they moved around. I decided to cap the length of the trails so the images would be ever shifting and changing from one moment to the next. This made it much more interesting to watch, and added to the illusion of interaction. You could see more clearly if one appeared to be circling another, for example. Once actual interaction has been implemented, the trails will be useful to check how it's all working, as they emphasize the Freaktures movements and speed.

Changes in speed and size

I have programmed in the ability to change the speed and the size of the creatures. This is done quite statically at the moment, but once I have included some interaction between Freaktures and their environments this will be more fully and dynamically integrated into their behaviour.
For example, if a Freakture eats enough, it could increase in size. When this increase in size occurs, the Freakture could slow down as it moves. Or, if one Freakture is angry or frightened it might speed up to catch or get away from another Freakture.

Interaction

So far, time and commitment to other projects has not allowed for me to implement any actual interaction of any kind. I have set up a number of devices which will allow me to expand my program and add these aspects in the future. Currently, however, I am more concerned with removing any and all bugs which are in the program at present. I don't think it's good practise to build on something that is not working entirely in the way I expect it to.
Rather peculiarly, I have noticed that even going as far as to program actual interaction between the Freaktures is not essential to the human perception of interaction and life-like behaviour.
Once I am happy that the program is in full working order, I will add interaction between each of the Freaktures as well as between Freaktures and their environments. I have decided that I will eventually allow the user to interact with the environments in an attempt to influence the "lives" of the Freaktures, but I will not allow them to actually interact directly with the Freaktures.

Birth and Death

I think that including the creation and destruction of Freaktures can only add to the perception of them being alive, as it's a very natural part of any life. I think adding these things would allow greater depth and give the user more meaningful interaction and empathy with the Freaktures. This would hopefully add to the excitement and enjoyment of the game.

Conclusion

Overall, I have really enjoyed this project – it has given me a chance to think about different ideas and explore a field I'm interested in but would otherwise have been unable to try out in depth. While I would have liked to have gotten further with the actual Freaktures program, my focus shifted slightly and my project became more about the ideas and possibilities I was discovering, something a lot more experimental. I am still happy with the program I have produced, especially as I feel I have discovered and learned a lot while making it.
So far, I have created my own little program with Freaktures moving around. I think that it's interesting to watch and easy to interpret in your own way. The trails add to this experience and can sometimes be quite mesmerizing in the same way that screensavers often are. Now, I want to add

more detail and make it more fun!

I have spent a lot of time trying to get the basic movement and framework set up, which hasn't left me with enough time to go on to actually add the more interesting interaction before the project deadline. It does mean, however that I have a fairly expandable program which I can continue to work on.
There are still some strange bugs in the program, such as the fact that the Freaktures move differently when there are more of them to when there is only one. This is not a direct problem, but I am concerned that I don't know what's causing it. I want to make sure I have a solid base to build on in order to more effectively program the behaviour system.

Some of the ideas that have come to me during this project are the importance of simplicity and using a persons own creativity and imagination to your advantage. The imagination can be quite a powerful thing. It seems to me that the less you rigidly define in terms of what a person should be seeing, the more they will actually see for themselves and the more easily they will believe what they think they see.
I specifically chose in my program not to add creature textures – either stationary or animated. I made this decision largely because of how annoying I find it in games when the animation does not match what you believe the character is or should be doing. I don't want that kind of incongruence to be apparent in my program. This doesn't mean I think that all detail should be taken out – just the superfluous information which could be potentially distracting or conflicting. I noticed that, similar to the example I gave when discussing *boredomresearch*, even without any actual interaction coded into my program, people would still imagine it was there. I find it interesting that we seem to look for patterns and then assign them whatever meaning is most relevant to our own experiences.

Further Ideas for Implementation

The next stages of making my program will mostly involve adding different types of interaction. Some of my ideas on what to include in these areas are as follows:
To add interaction between the Freaktures based on a combination of their inherent personality which does not change and their mood which can change depending on the situation. The basis for this system has been put into place, but needs to be implemented fully and expanded upon. I have also had an idea for getting the Freaktures to follow the trails left by others, which can be linked to other behaviours or desires.
I may add some level of interaction with the environment, but I don't want to take away from the abstract quality the current non-environment lends the project. I may simply use different colours to represent changing terrain or perhaps just simple shapes to represent objects placed in the environment for the Freaktures to react to.
When I add user interaction, I would like it to be quite creative and experimental. I want them to feel they can influence the lives of the Freaktures but not directly control them – for that would surely take the fun out of it. I had the idea of allowing the user to draw in specific colours and have

the Freaktures react differently depending on what colour they were moving over. This could be used, for example, to allow the user to draw pieces of food for the Freaktures, or to draw blue areas of “water” which would change the way the Freaktures move.
Overall, I would like to try to implement some of the interactions that my friends thought were going on, without breaking the illusion that they are.

Other Inspiration and Research

PACMAN:
Worth mentioning because the old games were great! So simple, but so addictively fun! This is a very good illustration of the point I was trying to make about simplicity. Now that it has been turned into a 3D platform game, it just doesn't work as well. I actually find it really annoying - they've taken a character who's simple design is mainly down to graphical limitations of the time and turned him into a sphere with arms and legs when we are now capable of creating much more detailed and interesting characters! I think it just takes the charm out of what was a nice retro character.

DEGENETRON:
The made up games for promotion of the modern game Grand Theft Auto: Vice City which is set in the 1980's. They successfully capture the retro gaming style, where interesting “characters” are represented by coloured squares and stories made up for why you're moving these coloured squares around. It's taken to extremes, but is actually a rather good representation/interpretation of old games. And although a little frustrating due to awkward controls, they're really fun to play.

CREATURES:
An old game a friend loaned to me during this project. It was worth looking at and quite useful, but I found it a little too complicated without the tutorial working (due to the age of the game). I think it suffered quite heavily from trying to add too much detail. The animation didn't always match up to what you expected the creature to be doing and it was quite frustrating not being able to do exactly what you wanted but being able to do a whole load of things you weren't particularly interested in. Of course, this game could be just right for someone else, but it didn't quite offer what I would be looking for in this kind of game.

A WHOLE BUNCH OF INTERNET GAMES AND TOYS AND GENERAL THINGS OF INTEREST:

http://www.aaronkoblin.com/work/faa

http://onecm.com/sketches/mycelium

www.sodaplay.com/constructor/index.htm

www.peterhowell.me.uk/html/05-processing_03.html

http://users.design.ucla.edu/~mflux/manifest/
www.eusc.net/media/writer/index.html

http://rpi.edu/~mcdonk/code/density/

http://www.galenmcgee.com/egg/LinesBetweenEx/applet/

http://users.design.ucla.edu/~mflux/p5/hashcollison2/applet/

www.coolbubble.com/programs/particlemorph.htm

http://chir.ag/stuff/sand

http://www.changar.com/archives/go.html

http://www.flong.com/yellowtail/

SPORE

I only found out about this recently, so it hasn't had much influence in the design of my project so far. I watched an amazing half-hour demonstration video and it looks unbelievably good! It's something that I will be looking into more in my own time.
The game is mostly about evolving your own creatures from a microscopic level right up to animals that can travel the universe. The sheer size of the game is incredible. Yet it allows so much control over all the little details, without seeming too complex. It's like a giant sandbox, where you can just sit and play around to see what happens. You get to create your own animals and then see how well they cope in a world populated by other animals designed by other people.
I'm very much looking forward to hearing more and playing it when it's eventually released.

Www...

TECHNICAL INFORMATION:

robthebloke.org

http://www.dam.org/bell/

http::ncca.bournemouth.ac.uk/jmacey/index.html

http://www.ncrg.aston.ac.uk/~cornfosd/graphics/opengl/openglman.html

http://www.libsdl.org/index/php

http://www.codersource.net/c++_vector_stl.html

http://www.msoe.edu/eecs/cese/resources/stl/iterator.htm

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vclang/thml/vcsmpstruct.asp

MORE TOSHIO IWAI LINKS:

http://www.imomus.com/dailyphoto190801.html

Tenori-On video:

http://www.youtube.com/w/Toshio-Iwai-(TENORI-ON)-@-Artfutura05?v=WQq2aXvIsz4&search=japn

compositions on the table (very similar to one of the areas in elecroplankton):

http://www2.kah-bonn.de/1/34/0e.htm

http://www.iamas.ac.jp/interaction/i97/artist_Iwai.html

http://www.we-make-money-not-art.com/archives/006917.php

MORE BOREDOMRESEARCH:

I was shown some other interesting research which involved creating your own creature online using any number of a set of body parts and then letting it grow and mate and evolve based on how good it would be in given situations. For example, if a creature you created had a lot of eyes and legs, it might be good at seeing predators and running away, increasing it's chances of survival. If, however, your creature had only a mouth, and no limbs it would probably die quite quickly through being unable to move and find food!

Some links that Vicky and Paul gave me are:

www.sodaplay.com

www.processing.org