

To: Ian Saunter

From: Steve Grand

4 pages

Subject: Ewoks techie bit for proposal.

23rd December 1992

Technology

Brains & physiology

A fundamental feature of this game is its use of Neural Net Technology to confer lifelike and adaptable behaviour upon the characters that occupy the computer world. As far as we know, this will be the first ever commercial game to use these techniques.

Mainstream neural net research has tended to concern itself with designs aimed at solving real-world problems in Pattern-Recognition and Prediction. Very little seems to have been done towards using computer-based nets to mirror the behaviour of biological neural systems, at the level of whole organisms. However, as my [SG's] main interest lies in computer-based life forms, I have for some time been developing the basis of just such a system, and am itching to implement it in a game. The new relationship between XXX and YYY provides a heaven-sent opportunity to put these ideas into practice.

The system I have in mind uses a relatively small net (a hundred or so neurons) for each furry creature and spider, modelling only the parts of the brain dealing with decision-making and drive-reduction. The net is supplied with inputs in the form of pre-digested sensory schema (algorithms are currently much better than nets for converting a mass of sensory data into real information about the environment) and generates output which triggers action schema (sequences of events, again handled by algorithms). The processing of sensory and action schema involves many of the techniques that I developed for Rome, but the addition of a network to handle the decision-making process changes the characters from dumb, rule-following automata, into sensitive, living creatures, with minds of their own.

The net associates inputs from the external and bodily senses, to produce output which seeks to reduce the level of a currently pressing drive (e.g. the need for companionship, or food, or sleep). Actions that ultimately reduce drives will tend to result in pleasurable sensations (either directly, such as the feeling of food in the stomach, or indirectly, for example a friendly stroke or other reward from the user). Likewise, actions that increase, or fail to decrease a currently pressing drive will tend to result in pain of one sort or another. The perception of pleasure or pain alters the levels of certain 'hormones' bathing the brain, which in turn tends to amplify or attenuate the neuronal associations that led to the recent choice of actions. Thus the organism learns ways to seek pleasure and avoid pain.

The full model is somewhat more complex than this, of course, and involves systems for the direction of attention, the raising and lowering of arousal, the consolidation of concepts during dreaming, etc. The available sensory information includes, not only external senses such as vision and touch, but also internal data from a fairly complex physiology, capable of simulating exhaustion, stress, boredom, communicable disease, etc.

The psychological model for this kind of behaviour is a fairly old-fashioned form of Behaviourism, but I'm confident that it will be adequate for creating relatively unintelligent, but highly subtle and sensitive living creatures, who are capable of learning reasonably complex, but above all believable, relationships.

LFCs will do whatever it is that LFCs do, if left alone, but with the intervention of a human user, they can be taught new ideas and relationships. For example, the user will discover that by associating a picture with a word (by writing it on the blackboard in the schoolroom), the LFCs can be taught simple language. They may then respond in sensible ways when the user speaks that word, or they might use the word themselves (to the user, or to each other), perhaps to issue a request. (Note that the actual language the user chooses to teach them is immaterial - it might be French, Anglo-Saxon, or even the LFCs' own language, as deciphered from books in the LFCs' library.)

Not only will users have fun trying to make the little critters do something sensible, but they will also have plenty of opportunity (if so inclined) to study the way they function, using psychological experiments or the physiological monitoring equipment conveniently strewn around the hospital area. This is because the internal models that make the system work use valid biological metaphors for their representation, rather than "computery cons". So, for example, the user might choose to administer a drug to a LFC (or spider, if he can catch one) and learn about its mental effects by monitoring the electrical activity in various parts of the brain. Everything will work the way it should.

The sensory/action-schema engine may work on the proven, linear, rule-based principles used in Rome. However, I have been working on a design for a neural net that should be able to incorporate rule-like relationships in a 'holographic' form (i.e. expressed within the total wiring of the neurons), yet which allows these rules to be programmed-in in a more discrete, linear form. In other words, a set of 'rules for living' can be written in a human-understandable form, then passed through a compiler which modifies the wiring of the neural net in such a way as to express those rules. This is not only more consistent (and possibly more efficient) but also allows the organism to modify the initial rules through experience.

Display

Obviously, the representation of a complex environment, plus a fair number of complex objects operating in real time, involve a good deal of processor power. Consequently, the display method should not demand too much of the computer's time if the game is to run on the kind of equipment that mere mortals have available. For this reason, we believe a fairly simple, top-down viewpoint is best.

For the DOS version, there are three options for which screen mode to use (multi-mode is too bulky, too limiting and tends to suffer from the 'lowest common denominator' problem). Each mode has different advantages and disadvantages:-

1/ Low resolution, 256-colour mode.

High level of realism, using a wide choice of subtle colours, with anti-aliasing.

Allows palette cycling, which is a fast and attractive way to make a landscape 'come alive'.

Similar to the style of display available on future, CD-ROM platforms, thus improving portability of the code, graphics and concepts.

Only room for one VGA page, so bank-swapping methods can't be used.

Large amount of data to blit to the screen, so smooth scrolling is out of the question. Even burst-scrolling is likely to be unbearably slow on most machines. A complex display scheme is needed to update such a bulky screen efficiently in real-time.

2/ Low resolution, 16-colour mode.

- Very limited palette and chunky pixels - very poor sense of realism.
- Insufficient colours to allow palette-cycling.
- Small memory requirement allows bank-swapping, very smooth scroll, more room to store graphics in RAM.

3/ High resolution (probably 640x350, to allow bank-swapping), 16-colour mode.

- Very limited palette, but dithering can be used to increase available colours.
- Insufficient colours to allow palette-cycling.
- More appropriate for transferring concepts to Windows, though aspect ratio is wrong.
- Room for more than one video page, and can use hardware to help produce a reasonably smooth scroll.

We are currently working on the assumption that the low-res, 256-colour mode is best. The biggest disadvantage is the poor scrolling, and the main advantage is the sense of realism generated by 256 colours and palette-cycling.

Because all user interaction happens by clicking onto hot-spots in the landscape, there is no need for a toolbar, and so we are aiming to devote the entire screen to the landscape view. There will be a need for pop-up windows to display occasional close-ups, and it may be necessary to use pop-ups to display real-time information (e.g. from medical displays), although I am hoping that this can be displayed directly on the objects themselves.

I shall isolate the modelling code as far as possible from the display technology, to allow for different resolutions, display methods and even viewpoints in future versions.

These matters are subject to change, once initial experiments have been carried out.

Programming considerations

The main part of the program will be written in C, to optimise portability (just possibly in C++, as the design is highly object-oriented in nature), with the time-critical sections in real-mode 386 assembler (even in real mode, the 386 offers enough extra features to justify ignoring the 286).

The minimum target machine will be a 386SX/25, with VGA and mouse, 1MB or more of memory and probably 5 or 6MB of hard disc.

There will be too many sprites and tiles to fit all the graphics in memory, so these will be cached from disc (on a most-recently-used basis), and into extended memory if available.

It is hoped that hooks can be included to allow for possible future add-ons, such as new landscapes, apparatus and species.