jhhl@panix.com

Henry Lowengard (.us)

CYBERGOMI: HERE TODAY, GONE TOMORROW

For my part of the poster session, I demonstrated my animation program Vapor Paint and showed a video tape of some of the animations and VRML objects made with the program.

Vapor Paint is written with a lot of philosophy in mind - ideas about how drawn art should be created on a computer. Here are a few of the ideas built into this program's design and interface:

- An "infinite" canvas should be used, with floating point coordinates so that a drawing may be as detailed as possible.

- All colors are 48 bit RGB with 16 bits of transparency.

- As much screen space as possible should be reserved for the working drawing and navigation around that space into the much larger canvas space should be natural and unconfusing.

- The drawing commands are edited in a schematic space much like the wireframe sketches of a 3D modelling program, and later rendered into frames (or other "products" like audio files or VRML descriptions) by a variety of renderers This "work screen" can easily be customized to show the same data in a variety of ways, both to aid in animating and to speed up its drawing on slow systems.

- The basic data structure corresponds to a marking gesture, with the coordinates of the path of that gesture (X,Y), a depth (Z), a variable width along that path (R), a variable "pen pressure" (M), and a time stamp for each vertice in that gesture(S). This structure is derived and edited by interpreting a more usual series of mouse movements, or via macro programs Then at rendering time provides the information to create various textured lines and areas.

- Certain of the editing functions use the timestamps built into every vertice to control how the editing gesture is mapped to the object, rather than simply mapping length to length.

- There are facilities for laying out and sketching in the same conceptual space as that in which the animations are built

- The gestures are organized into key frames, which are organized into sequences. All time is given in floating point "Frames", and the renderers can merge several subframes together to provide motion blur.

- The rendering view is also animated as a series of view key frames The view can be rotated and re-scaled and corresponds to the final aspect of the finished frames.

- All the elements of the animation - views, key frames, colors, gestures, processes and more - are available for editing either by hand or by macro and can be navigated, reordered, selected, labelled or hidden. The hiding feature is especially useful for cutting down on screen clutter and making test animations quickly.

- Instead of the usual pull-down menus and pop-up windows, Vapor Paint uses large forms with its own interface elements. This tends to be faster and less confusing to work with: no piles of windows all over the screen in various stages of activity and relevance.

- The macro language is heavily used to provide services usually found in custom files. Macros can do everything that can be done by mouse and keyboard and also be called automatically during the rendering process, to aid in filling out a form, asynchronously via the keyboard or for interpreting mouse movements into a custom editing or creation tool

- Vapor Paint also has a hypertextual help system built in for reference

Vapor Paint currently only runs on the (extinct) Commodore Amiga computer. It is very compact and flexible, having been written in 68000 assembler. An entire, uncompressed bootable system, complete with ARexx interpreter (the macro language), help files and example files fits on a standard 880K floppy disk and can run in as little as 1MB on operating systems dating back to the late 1980's. Nevertheless, I've written a renderer in very generic C which runs on SGI and NeXT equipment (and probably anything else with a C compiler). A modeller will be harder to write because of the conflicting graphic APIs native to the various operating systems.

Vapor Paint allows me to create animations with a particular, identifiable "look" to them. Armed with my drawing tablet and trusty Personal Animation Recorder card, I can create video animations with perfect timing and good color in a very short time. I can also use its internal structural organization as a database for more general graphic work, to visualize coordinates taken from other sources, edit them and convert them to other forms.

More information and an program package will (eventually) be found on my home page, <http://www.echonyc.com/~jhhl>.