INTERACTIVE GRAPHICS AS A VEHICLE FOR THE ENHANCEMENT OF HUMAN CREATIVITY

Ronald M. Baecker
University of Toronto

Abstract

A research computing environment which is particularly well suited to the design and implementation of interactive graphics systems has been developed at the University of Toronto. Key components in the environment are a large minicomputer, calligraphic and digital video displays, a digitizing tablet, the UNIX operating system, the high-level programming language C, a high-level general-purpose "device-independent" graphics package, and a first-rate user community. Users of this environment have developed with relative ease highly responsive interactive graphics systems which become dynamic media to aid creative problem solving. Three such programs are described briefly: a system for dynamic modelling; a system for producing movies of astronomical phenomena; and a system for newspaper page layout.

UTILISATION DE SYSTÈMES GRAPHIQUES INTERACTIFS POUR AMÉLIORER LES APTITUDES CRÉATRICES CHEZ L'HOMME

Résumé

Un cadre informatique de recherche qui convient particulièrement bien à la structure et au mode d'utilisation des systèmes graphiques interactifs a été mis au point à l'université de Toronto. Parmi les composants clés de ce cadre on retrouve un mini-ordinateur, des affichages vidéo calligraphiques et numériques, un tableau de numérisation, le système UNIX, le langage de programmation C évolué, un bloc graphique d'usage général, évolué et non-tributaire du type d'appareil, et un ensemble d'usagers de première catégorie. Les usagers de ce cadre ont mis au point sans trop de difficulté des systèmes graphiques interactifs très sensibles qui servent de moyens dynamiques pour aider à résoudre des problèmes de façon innovatrice. On y décrit brièvement trois de ces programmes: système de simulation dynamique; système de production de films représentant des phénomènes astronomiques; et système de mise en page pour les journaux.

INTERACTIVE GRAPHICS AS A VEHICLE FOR THE ENHANCEMENT OF HUMAN CREATIVITY

Ronald M. Baecker
Dynamic Graphics Project
Computer Systems Research Group
University of Toronto

I. INTRODUCTION

We shall describe a research computing environment which is particularly well suited to the design and implementation of interactive graphics systems. It has enabled us to develop with relative ease highly responsive interactive graphics systems which become dynamic media to aid creative problem solving. We shall describe the environment, which may be contrasted with our previous one reported upon at the Fourth Conference [Baecker 75], present brief introductions to three typical application systems, and discuss some future research directions.

II. THE ENVIRONMENT

The key components include a large minicomputer, calligraphic and digital video displays, and a digitizing tablet, the UNIX operating system, the high-level programming language C, a high-level general-purpose "device-independent" graphics package, and a first-rate user community.

A. The Hardware

The computer is the DEC PDP-11/45. The displays are a high-performance refreshed calligraphic display, a direct view storage tube, and a low resolution digital colour video frame buffer. The input device is a tablet with stylus and cursor.

Our Three Rivers Computer Company Graphic Display Processor has several outstanding features. It includes a state-of-the-art digital vector generator capable of synthesizing lines at 30 nanoseconds per point, and hence of drawing 100,000 short vectors (less than 1/4 inch) flicker-free. It has three vector formats, short (one byte), medium (one word), and long (two words). It draws all characters as subroutines composed of vectors. Because of the last three facts, it is by far the most cost-effective commercially available display for working with text and free-hand curves. Although originally designed for use with an 11/05 as a satellite graphics terminal, it has been integrated into our 11/45 with special memory mapping hardware. The display is refreshed from double port memory in the address space of the 11/45, hence the display file can be updated rapidly for real-time graphics and animation. Because of the extra port on the memory, the high refresh bandwidth does not saturate the Unibus. The display possesses a simple and relatively clean instruction set [Rosen 74].

The frame buffer is currently limited to 256X256 resolution and to a choice from among 256 hard-wired colours. A programmable colour map which will allow selection of 256 logical colours from a space of a billion physical colours is almost installed. A novel video display processor which would significantly enhance the device's effective resolution and versatility has been functionally designed, but neither logic design nor implementation has been carried out.

The tablet, used together with a stylus or a four-button cursor, was selected as the most versatile single input device. It is of course augmented by a keyboard.

B. The Operating System

The operating system is the Bell Laboratories UNIX system [Ritchie 74a], a robust time-sharing system which provides very elegant and congenial tools for program preparation, input-output, file management, and multi-process cooperation.

UNIX consists of a rich set of programs including good text editing, document formatting, and typesetting systems, a variety of high-level languages, two compiler-compilers, and a macroprocessor. Most system software including the bulk of the operating system is written in C. Hence UNIX has proved both reliable and malleable. File, device, and interprocess I/O are compatible. The user can choose to accept the default I/O scheme, which is clean and invisibly buffered, or can develop his own buffering scheme and optimize it at will. There is a hierarchic file system with mountable directories, as well as reasonable file storage, archival, retrieval, and protection mechanisms. Asynchronous processes can exist concurrently and can communicate via "pipes". A sophisticated programmable command interpreter allows multitasking, foreground and background execution, and the synergistic coupling of software tools [Kernighan 76].

C. The Language

The high-level language is C [Ritchie 74b], which provides good problem description and program structuring tools and also produces efficient and trustworthy code. It contains an adequate set of modern control structures including WHILE ... DO ..., DO ... WHILE ..., SWITCH, and recursive procedures, and of modern data structures including arrays, structures, and pointers. There is no block structure. It contains a macro preprocessor for parameter substitution and file inclusion. The compiler is reliable and generates efficient code which makes good use of the PDP-11 addressing capabilities. There is a source language debugger. We have added additional tools for program paragraphing, syntax checking, and graphical debugging.

D. The Graphics Support System

The graphics package is GPAC [Reeves 76], a set of C-callable subroutines which provides good tools for "device-independent" picture creation, management, and transformation, for real-time input, for picture archiving, and for real-time movie playback. The system allows the high-level creation and manipulation of both line drawing and area shaded pictures on a variety of refresh, storage, and raster scan displays. Real-time graphical input facilities are available for stylus tracking and inking, light button interactions, and device polling. A standard format for representing pictures allows graphical modules to be coupled and combined in the style of [Kernighan 76], and facilitates picture archiving and the use of multiple devices for soft display and hard copy.

E. The User Community

There is great excitement, productivity, and creativity in the environment. We have attracted a remarkable collection of graduate and undergraduate students. They appreciate the use of the first good interactive computing facility on campus. They thrive on their easy and unhassled access to the system; we do no accounting and respond to demands for system resources that exceed capacity by an informal protocol consisting of good will and mild prodding by "system hacks." Resources are shared whenever possible. For example, four processes can obtain display file space concurrently and therefore share the screen; it is not unusual for someone using the display to find superimposed on it someone else's picture for a few seconds as the latter obtains just a quick look to see if his program is working. UNIX has facilitated the production of modular tools which are then re-used and coupled in various ways to bootstrap our productivity; we are constantly amazed at how much has been accomplished in 2 years.

III. SOME APPLICATIONS

We shall now give brief descriptions of three application systems, which will be demonstrated with film and video tape at the Conference. These three are only illustrative of current developments; several others could have been selected as well. [Crossey 77] [Fono 77] [Grossman 77] [Tuori 77] [Wright 77]

A. SLOGO

SLOGO [Duff 76] is a conversational process-oriented discrete stochastic simulation language and system, with integrated animation capabilities which allow the execution of any model to be visualized immediately as a moving picture portraying the simulated system.

In a typical Slogo model, distinct classes of objects are identified and then procedurally described. Multiple instances of each class can be activated and then exist concurrently. These instances are called processes. Thus, in a model of a centrally dispatched taxicab system [Hauer 77], there are usually simultaneously active many cab processes, many passenger processes, and one dispatcher process. Processes can start, suspend, and restart the execution of other processes, can read and write the state of other processes, can suspend their own execution for a fixed or an indefinite period of time, and can request and release units of resources. Resources function as queueing mechanisms and also as the process synchronization tools usually known as semaphores.

The most novel feature of SLOGO is the ability it gives the model builder to visualize what is happening in the simulated system. As the model executes, a movie depicting the system is computed and displayed.

For example, in the animation of the taxicab system [Baecker 76a], passenger origins and destinations are represented by numbered crosses and diamonds. Taxis waiting for a customer, enroute to pick up a fare, or carrying a customer to a destination are represented by pictures of taximeters with upright flags, horizontal flags, and flags down, respectively. These symbols move around the screen, picking up and discharging passengers. Changes in system behavior caused by different dispatching strategies are visualizable in the animation, which can later be replayed and analyzed at will.

B. KEPLER

KEPLER [Pike 76] is a system for producing movies of astronomical phenomena. The user is provided a concise but powerful language for specifying the motion, orientation, and field of view of a hypothetical camera. The system can compute movies varying in realism from a two-dimensional display of planet trajectories to a full three-dimensional portrayal with occlusion and phases. Unlike SLOGO, in which all user input is entered at the keyboard, KEPLER also uses tablet pointing and inking. Basic system control is exercised by pointing at light buttons, camera motions are specified using a written observer language, and motion dynamics are described by sketching waveforms.

C. NEWSWHOLE

NEWSWHOLE [Tilbrook 76] is a system for newspaper page layout. An editor uses NEWSWHOLE to specify the placement and typesetting controls for stories, pictures, and ads. He works with an abstract diagrammatic representation of the page in which constituent text is suppressed. Almost all input is given with the cursor and tablet. Stories are selected, their heading style is defined, and they are positioned on the page. New stories flow around existing ones; space is automatically contracted or expanded so that a story fits. Pictures are represented by rectangles, which, once positioned, can be moved, shrunk, or expanded by pointing at the center, a side, or a corner, and then by pulling or pushing. Moving objects stop when they encounter an obstacle, and moving vertical story borders jump from column rule to column rule. Groups of headings and banks of text can be adjusted under constraints so that space is given from one story to another. The system is highly interactive, blazingly fast, and congenially robust [Baecker 76b].

IV. SOME FUTURE RESEARCH DIRECTIONS

SLOGO could enhance significantly the ability of scientists, engineers, and planners to visualize the implications of embedding various assumptions or strategies in a model of a complex system. KEPLER allows astronomers and educators to produce and debug animated films of astronomical phenomena with an investment of only a few hours of their time. NEWSWHOLE will make possible highly responsive editorial control over the form of newspaper pages. But these systems are still only prototypes. What can we do towards bridging the gap between laboratory toys and cost-effective tools that do indeed enhance human creativity?

Our thoughts along these lines lead towards three directions of endeavor. We want to develop mechanisms to transport systems we have developed to other comparable minicomputer environments. We are working on strategies for transforming viable programs into cheap turnkey systems, possibly based on microprocessors. We are also looking towards conversational extensible graphics languages [Baecker 76c] as possible implementation environments to facilitate the production of future systems.

ACKNOWLEDGMENTS

The success of our environment is in large part due to the skillful and enthusiastic participation of many individuals. We owe special thanks to Bill Buxton, Tom Duff, Greg Hill, Tom Horsley, Les Mezei, Rob Pike, Bill Reeves, Dave Sherman, Mike Tilson, and Martin Tuori.

REFERENCES

[Baecker 75] Ronald M. Baecker, A Computer Animation Facility for Research and Educational Filmmaking: Design and Application, Proc. of the Fourth NRC Man-Computer Communications Conference, 1975, pp. 19.1-19.11.

[Baecker 76a] Ronald M. Baecker, Greg Hill, and Martin Tuori, Computer Animated Simulation of Taxi Dispatching

Strategies, 10 minute 16 mm b&w sound film, Dynamic Graphics Project, CSRG, Univ. of Toronto, 1976.

[Baecker 76b] Ronald M. Baecker, David Tilbrook, and Martin Tuori, NEWSWHOLE, 10 minute b&w sound 3/4" video cartridge, Dynamic Graphics Project, CSRG, Univ. of Toronto, 1976.

[Baecker 76c] Ronald M. Baecker, A Conversational Extensible System for the Animation of Shaded Images, Computer Graphics, Vol. 10, No. 2, Summer 1976, pp. 32-39.

[Crossey 77] Sheila Crossey, Source-Level Interactive Graphical Debugging, M.Sc. Thesis, Dept. of Computer Science, Univ. of Toronto, 1977 (expected).

[Duff 76] Thomas Douglas Selkirk Duff, Simulation and Animation, M.Sc. Thesis, Dept. of Computer Science, Univ. of Toronto, 1976.

[Fono 77] Ilan Fono, An Interactive Digital Logic Simulator, M.A.Sc. Thesis, Dept. of Electrical Engineering, Univ. of Toronto, 1977 (expected).

[Grossman 77] Gary Grossman, Interactive Dynamic Display of Electrocardiographic Data, M.Sc. Thesis, Dept. of Computer Science, Univ. of Toronto, 1977 (expected).

[Hauer 77] Ezra Hauer, Ronald M. Baecker, and Paul D. Bunt, Computer Animated Simulation of Taxi Dispatching Strategies, presented at the Annual Meeting of the Transportation Research Board, January 1977.

[Kernighan 76] Brian W. Kernighan and P.J. Plauger, Software Tools, Addison-Wesley, 1976.

[Pike 76] Robert Pike, Computer Animation of Astronomical Phenomena: The KEPLER System, Report submitted to the McLaughlin Planetarium of the Royal Ontario Museum, Dynamic Graphics Project, CSRG, Univ. of Toronto, 1976.

[Reeves 76] William T. Reeves, A Device-Independent, General-Purpose Graphics System in a Minicomputer Time-Sharing Environment, M.Sc. Thesis, Dept. of Computer Science, Univ. of Toronto, 1976.

[Ritchie 74a] Dennis M. Ritchie and Ken Thompson, The UNIX Time-Sharing System, CACM, Volume 17, Number 7, July 1974, pp. 365-375.

[Ritchie 74b] Dennis M. Ritchie, C Reference Manual, Bell Telephone Laboratories, Murray Hill, N.J., 1974.

[Rosen 74] Brian Rosen, Graphic Display Processor Programmers Guide, Dept. of Computer Science, Carnegie-Mellon Univ., 1974.

[Tilbrook 76] David M. Tilbrook, A Newspaper Pagination

System, M.Sc. Thesis, Dept. of Computer Science, Univ. of Toronto, 1976.

[Tuori 77] Martin Tuori, Tools and Techniques for Computer-Aided Animation, M.Sc. Thesis, Dept. of Computer Science, Univ. of Toronto, 1977 (expected).

[Wright 77] Sandy Wright, Animated Symbolic Language Teaching System, M.Sc. Thesis, Dept. of Computer Science, Univ. Of Toronto, 1977 (expected).