PRACTICING ON THE CORE GRAPHICS SYSTEM

G. Faconti

P. Palamidese

Rapporto interno C82-2

Istituto CNUCE

PISA

PRACTICING ON THE CORE Graphics System

G.Faconti
CNUCE - Institute of C.N.R.
Via S.Maria, 36
56100 - Pisa, Italy

P.Palamidese
CNUCE - Institute of C.N.R.
Via S.Maria, 36
56100 - Pisa, Italy

Abstract

The system described in this paper is intended to be of assistence to those people who would benefit in their works from computer graphics but do not have the necessary knowledge or familiarity with its methodological principles. This system consist of an APL implementation of the CORE Graphics System and of a user interface giving the opportunity of learning and practicing on the CORE concepts. The system is highly interactive, being ready to give explanations to the user at any time, and to teach him the use of the system. The implementation has been made under VSAPL for CMS running on an IBM 3033. The graphics output device supported is the IBM 3277 Graphics Attachment equipped with a Tektronix 619 Storage Tube and a Summagraphics ID tablet.

1 - Introduction

As a computing center providing services to a large community of researchers dealing with several fields of application, the main goal of CNUCE has always been to develop its systems following some sort of standards.
Actually, this is not true in the field of Computer Graphics where packages and portions of application programs exist to support various graphics devices following completely different approches.
In order to solve the problem, it has been decided to implement on our computers a graphics system conforming to the more recent tendencies of standardization in Computer Graphics.
Following this decision, a prototype of the CORE Graphics System has been developed and a high level user interface has been implemented with the purpose of:

- gaining detailed knowledge of the CORE System itself,
- evaluating the usability of the CORE System and its interface to several of the graphics applications within our computing center,
- being of assistence to those users who could benefit in their work from computer graphics but do not have the necessary knowledge or familiarity with its methodological principles.

In the following sections we will give some highlights on the implementation of the prototype of the CORE System and we will discuss on the capabilities provided by the user interface. In the concluding remarks some general considerations are made on the base of the feedbacks received from the users of the system.

2 - The APLCORE implementation

The CORE Graphics System is implemented in APL language under VM/CMS. Referring to the Status Report of the Graphics Standards Planning Committee of ACM/SIGGRAPH dated August 1979, it currently supports the level 2 capabilities for output, input, and dimension classes (i.e. buffered, synchronous, 3-D).

2.1 - The Language

The choice of APL as implementing language has been motivated by the following considerations:

- the tremendous power of the APL debugging facilities,
- APL is particularly designed to operate on arrays as the CORE System viewing transformations require
- the possibility of full adhere to the syntax of the CORE System,
- once a function has been defined it can be made part of a higher level function or it can be invoked interactively as part of a command language,
- the function concept of APL led almost inevitably to a modular approch in the construction of the program,
- the workspace concept of APL still reinforce the characteristic of modularity in designing applications,
- the shared variable mechanism makes it possible to clearly separate the device independent portion of the CORE System from the device drivers.

2.2 - The system components

The CORE System implementation is organized into seven APL workspaces. The APLCORE workspace contains the INITIALIZE-CORE and TERMINATE-CORE functions and acts as main driver for the system.
The remaining workspaces, named from CORE1 to CORE6, contain the device independent functions of the CORE System following the level options as follow:

Workspace Name	CORE Classes Output	Input	Dimension
CORE1	Basic	None	2-D
CORE2	Basic	None	3-D
CORE3	Basic	Synchronous	2-D
CORE4	Basic	Synchronous	3-D
CORE5	Buffered	None	2-D
CORE6	Buffered	Synchronous	2-D

The system is accessed by loading the APLCORE workspace and executing the INITIALIZE-CORE function. Depending on the argument passed to this function the proper COREx workspaces are loaded and the requested CORE System level is established.
The device drivers are developed as separate components of the system and are loaded as auxiliary processors at VSAPL loading time.
Actually, device drivers are implemented for the IBM 3277 Graphics Attachment and for the Tektronix 401x family terminals.

3 - The user interface

The user interface has been developed in order to provide the user with the capability of practicing on the CORE methodology.
The man-machine dialogue style is based on the menu type. The system presents control elements on the alphanumeric screen and responds to their activation with either alphanumeric or graphics elements on the corresponding screen.
The most important features of the interface are:

- improved human-computer interaction,
- modular structure,
- help/explanation features,
- usability by non programmers.

A considerable help in reaching these goals is due to the use of APL and to the implementation for the IBM 3277 Graphics Attachment.
APL has been found very helpful in designing a high interactive, modular and usable system. The IBM 3277 Graphics Attacnment gaves us the opportunity of eliminating interference between graphics and alphanumeric output, and of enhancing the level of interactivity through the access to the IBM 3270 Information Display System facilities.
The system is divided into five sections: introduction to the CORE Graphics System, viewing transformations, output functions, input functions, and advanced application.
A primary menu allows the user to choose the proper section which he whishes to exercise on.

3.1 - Introduction to the CORE Graphics System

The user is provided with a short script describing the objectives and the philosophy on which the CORE proposal is based, together with a few graphics displays showing the structure of a graphics package. The user has also the possibility of interrogating a glossary which gives a description of the CORE terminology, by selecting keyword parameters presented in a menu.

3.2 - Viewing transformations

On the 3277 screen, a menu is presented showing the CORE viewing parameters and a selection of objects that can be displayed on the graphics screen.
The viewing parameters may be modified by positioning the alphanumeric cursor and entering a numeric value for any or all of the viewing parameters. Objects may be selected by pointing the light pen over the object name which will then be highlighted.
A save/restore facility for both viewing parameters and selected objects is implemented through the use of programmed functions keyboard.
Once selection is made and parameters are set, the depression of the enter key will cause a threeviews drawing to be displayed on the graphics screen showing the relationship between viewing parameters and selected objects. In this way the user can immediately see the effects produced on the display of an object and understand the meaning of the viewing parameters.

3.3 - Output functions

This section of the system will introduce the user into the concepts of view surface, pictures segmentation and naming, and lines and text drawing primitives, together with primitives and segments attributes. Swiching between various menus, the user has the capability of setting the value of the attributes and of executing the CORE output functions by selection through programmed functions keyboard or light pen. When the invoked function requires additional parameters the user is prompted to introduce them by keyboard entry.
During execution of a function the CORE error handling mechanism will eventually display standard error messages together with "suggestions" on the operation to perform in order to correct the error itself. If the setting of the parameters and the sequence of operations is found to be consistent, the system responds by visualizing the results of the operations on the graphics screen.

3.4 - Input functions

Within this section the user has the capability of swiching between various menus in order to establish the proper setting of parameters and execute the CORE input primitives.
Input functions support the following devices:

- Buttons (the 3277 PFKs)
- Keyboard (the 3277 keyboard itself)
- Locator (the cursor control)
- Pick (the cursor control)
- Stroke (the Summagraphics ID tablet).

No support is actually provided for the valuator device.
In order to provide the user with some kind of control over the manner in which input devices are operating, echo types are defined for any of the supported devices.
As for the output functions, any error condition detected by the CORE error handling mechanism is notified to the user on the alphanumeric screen with a short description of the error so that it can be recovered by user intervention.

3.5 - Advanced application

Within this section the user is provided with the capability of execute a purposely limited set of high level commands for building and displaying solid objects.
The objects handled by the system are considered to be composed of a set of points (vertices) connected by lines (edges). All geometrical solids (i.e. cubes, prisms, etc.) which fall naturally into this representation, may then be defined. Analytical surfaces may also be represented in this way, provided they are evaluated in a given set of points.
A simple "modelling system" has been defined in order to describe the objects. They are basically represented with two matrices: one containing the coordinates of the vertices of the objects and the other one describing the way in which vertices are connected.
The commands the user may execute, are build on top of the CORE System in the sense that they perform operations by calling the CORE functions.
When a command is executing, a trace of the calls to these functions is displayed on the alphanumeric screen together with their arguments. In this way the user can control the program flow and see directly the effects produced on the graphics screen.
Commands are provided to generate and erase simple or complex objects, and to position objects into the graphics screen.
In addition, inquiry commands are also available.

4 - Conclusions

Although the CORE System implementation is operational and of good performance, it is not intended for production works. It lacks in portability for both CPUs and graphics devices: in particular it doesn't actually support any kind of plotter device.
On the other end, the user interface has demonstrated to be very helpful for our users in learning and practicing on the CORE concepts and for developing simple applications. The man-machine dialogue has also proved to be well designed and very friendly to the user.
APL is found to be profitable used as implementing language for interactive systems. The only remark we can make is that APL interpreter cannot handle asynchronous device interrupts so that we were forced to limit the CORE Graphics System implementation to synchronous events. However taking into account the goals stated at the beginning of this work and the intermediate performance graphics devices at our disposal, a higher level implementation of the CORE System will probably be inefficient.

5 - References

1 - Status Report of the Graphics Standards Planning Committe of ACM/SIGGRAPH, Computer Graphics, Volume 11, Number 3, Fall 1977

2 - Status Report of the Graphics Standards Planning Committe of ACM/SIGGRAPH, Computer Graphics, Volume 13, Number 3, August 1979

3 - J.C.Michener and A.van Dam, "A functional overview of the CORE System with Glossary", Computing Surveys, Volume 10, Number 4, December 1978.

4 - J.C.Michener and J.Foley, "Some major issues in the design of the Core Graphics System", Computing Surveys, Volume 10, Numer 4, December 1978.

5 - W.M.Newman and R.F.Sproull, "Principles of Interactive Computer Graphics", Second Edition, McGraw-Hill 1979

6 - S.Trispel and K.Gunther, "User guidance in interactive systems", Proceedings of Eurographics 81, Darmstadt, September 1981

7 - J.Warner and N.Kiefhaber, " The DIGRAF implementation of the proposed GSPC standard", Proceedings of Eurographics 79, Bologna, October 1979

8 - H.Hvistendahl, "IGL - A structured language for interactive graphics", Proceedings of Eurographics 79, Bologna, October 1979

9 - V.Casarosa and S.Trumpy, "Conersational building and display of solid objects", Proceedings of the Sixth International APL Users Conference, Anaheim, May 1974

10 - APL Language, IBM, GC-3847

11 - VSAPL for CMS: Terminal User's Guide, IBM, SH20-9067

12 - VSAPL for CMS and TSO: Writing Auxiliary Processors, IBM, SH20-9068

13 - VSAPL Extended Editor and Full Screen Manager, IBM, SH20-3441

14 - IBM 3277 Display Station with Graphics Attachment, IBM, GA33-3039

15 - Form 16 for 2000 series, Users Manual, Summagraphics Corporation

STAMPATO PRESSO IL
SERVIZIO TECNOGRAFICO
DEL CNUCE