

Comments from the Editors

A Questionnaire

To determine the current status of some of the work being done by our readers I have put together a questionnaire. The questionnaire itself will be found following this short note. I would appreciate hearing from all persons currently doing some work in the area of computer graphics, or from anyone wishing to contribute information on a system in use. I will try to put all responses together (assuming someone out there responds) and summarize what is being done, as indicated by the returned questionnaires.

For those of you not familiar with the publication, the Graduate School of Design at Harvard University produces a newsletter entitled CONTEXT. It is a product of the Laboratory for Computer Graphics and Spatial Analysis from the above group. This group has a distribution system for submitted programs, and also markets their own software. The areas covered by the newsletter include (but are not limited to): CAD, computer cartography and spatial analysis techniques. If you feel you might like contact with people in the above or related areas, please contact the group.

Please complete the questionnaire and return to:

A. P. Lucido
Rm. 4 Computer Garage
Ames Lab
Iowa State University
Ames, Iowa 50010

(Please, all jokes about if we diagonal park our computer at the computer garage should be put on separate sheets!) If you run out of room answering the questionnaire, feel free to use additional sheets of paper for your responses.

Anthony P. Lucido

- | | | |
|-------------|--------------------|-------------|
| 1. Computer | Graphics device(s) | Type of use |
|-------------|--------------------|-------------|
-
2. Languages being used for graphics:
 3. Graphics devices under development:
 4. Graphics languages under development:
 5. Other graphics research:
 6. Any documents available, and who to contact:
 7. Remarks:

Data Structures

An introductory discussion about data structures is given below. The reasons for the remarks that follow are to outline an area of interest for SIGGRAPH and to encourage contributions for publication and further discussion.

It is widely acknowledged that the organization of data is of primary importance in any real application of computers, whether or not graphics is involved. The way in which data is structured determines the elegance or otherwise of programs and algorithms that access and manipulate the data, the time taken to execute these programs, and the storage space required to store the data. For applications involving interactive computer graphics there are additional critical factors, including response times to user actions and display frame-rates, that depend on the complexity of data structures and display file organizations. Therefore data must be organized to allow necessary searching and accessing of related data to be performed efficiently. This leads to two conclusions: that data structures are problem dependent; and also that there is no universal or optimal data structure. Consequently, data-structure design is an art and requires experienced people to make the best designs. Difficulties arise in balancing various compromises and tradeoffs for a given application. Furthermore, a data structure, which describes the way a programmer views his data, must be mapped into physical storage, hence several more difficulties arise, again relating to efficiency considerations.

There appears to be two major problems. First, it is not easy to explain or document data structures or physical storage structures; one reason being because there does not exist a convenient mathematical notation for describing data structures and data structure operations. This makes it difficult to discuss and compare different designs for the same application. Also, although one can teach certain basic ideas such as list structures, hash coding, etc., to new programmers, it is much harder to teach them "Data-Structure Design Principles" when one gets beyond the basic ideas. Secondly, implementing data structures can be difficult. One reason is because of the limitations of programming languages; most programming languages support only a very limited set of data types such as integer, real, logical and complex, (although PL/I is more powerful and allows pointer manipulations and STRUCT definitions). Because of these limited data description facilities, it is usually quite difficult to cater to all the hardware idiosyncrasies of interactive graphical equipment and one often has to resort to numerous routines coded in machine/assembly language.

Many people have tackled these two problems with varying degrees of success. This is not the place to review the various contributions, but permit me to mention one approach to the second problem of limited language facilities that we are working on at New York University. We are developing an extensible programming language, called EX.GRAF, that simplifies programming for graphics. The graphical extensions fit into the general language framework very neatly, and allow a user to write programs making full use of available graphical facilities. Normal arithmetic computational facilities are included also. Specifically in EX.GRAF, one can define new data types and structures that are suitable for graphical problems. One can then define variables to have these types or structures and subsequently use these variables in statements. It is particularly interesting that graphical data types can be defined so that operations with

variables having graphical data types can be created. Consider some examples: a point can be treated as a single entity rather than as three numbers; a table of coordinate data in display format can be created; then the structure of a display format table can be altered or a table can be linked to other tables, or the coordinate data can be accessed, or the coordinate values can be changed. Transformations of graphical elements can be specified and hierarchical image structures can be constructed. List processings can be defined and one could do vector string manipulations like character string manipulations if it was useful. After all these features have been provided, a programmer still has to design his data structures of course, all that a language can do is help him implement designs (just as a programming language helps one implement an algorithm.)

The language is currently being implemented on an Adage AGT-30. A description of the language should appear in a paper soon.

From the preceding remarks it should be clear that the range of interest in data structures is wide. Contributions, comments and suggestions on any aspect of "data structures" are welcome. Some graphical orientation is desirable, although 90% or more of a "graphical" application involves non-graphical work. We will try to disseminate any information you wish to send us.

Robin Williams, Assistant Editor
Data Structures

Miscellaneous Comments from the Editor

The last newsletter was incorrectly dated as the Winter 72 newsletter. T'aint so. It was the Fall 72 issue and this one is the Winter 72.

The monthly bulletin discussed in the last newsletter hasn't gotten started because our dues haven't been officially modified yet. Hopefully we'll be on the air by February.

Alas, my supply of artwork that I've been using in the newsletters has been exhausted. I could use more if I could get it in small enough size and shape for simple reproduction in the newsletter.

Also I could use more articles for the next newsletter. I'm holding some back for then but I could definitely use more. So send me articles, status reports, thoughts, gripes, etc.

Deadlines for Copy

Deadlines for copy in the Spring and Summer issues are March 12 and June 4, respectively. The issues should be in member's hands within 6-7 weeks of these dates.