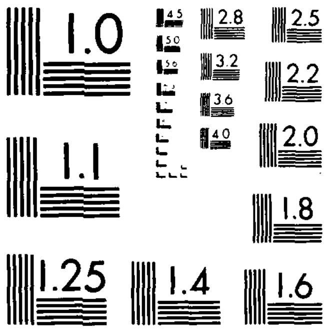
AD-A184 621 TECHNOLOGY APPLICATIONS BRANCH GRAPHICS PROJECTS(U) 1/1
ARMY TRADOC ANALYSIS COMMAND FORT LEAVENWORTH KS
R H KAEDING ET AL JUL 87 TRAC-F-TM-0687

UNCLASSIFIED F/G 12/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Technical Memorandum TRAC-F-TM-0687
July 1987

TRADOC Analysis Command-Fort Leavenworth (TRAC-FLVN)
Operations Directorate
Fort Leavenworth, Kansas 66027-5200

Technology Applications Branch Graphics Projects

by

Tim Daniels

Robert H Kaeding

ACN 48722

ADA184621

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TRAC-F-TM-0687	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION TRAC-FLVN 6b. OFFICE SYMBOL (If applicable) ATRC-FOC	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Commander TRAC-FLVN (ATTN: ATRC-FOC) Ft Leavenworth, KS 66027-5200	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Technology Applications Branch Graphics Projects

12. PERSONAL AUTHOR(S)
R. H. Kaeding/Tim Daniels

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final Report	FROM 4-87 TO 7-87	1987 Jul	9

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Technology Applications Branch-Graphics Team (TAB-GT), Operations Support Directorate, TRADOC Analysis Command (TRAC) is located at Fort Leavenworth, Kansas. TAB-GT is responsible for graphics interface software development and maintenance, application software modification when compatibility problems are encountered, and assisting application programmers where a graphics interface is desirable. Currently the majority of application software packages maintained by TRAC and requiring graphics capabilities utilize a package known as the Kellner Graphics Interface Package (KGIP) named for its author Mr. Al Kellner, TRAC-WSMR. This document discusses three major projects undertaken by TAB-GT to meet the previously mentioned responsibilities which led to the modification of the KGIP and eventual evolution of an even more sophisticated graphics interface package compatible with any TRAC VAX/Ramtek hardware configuration.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL R. H. Kaeding/Tim Daniels	22b. TELEPHONE (Include Area Code) (913) 684-3617	22c. OFFICE SYMBOL ATRC-FOC

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE

CONTENTS

ABSTRACT

The Technology Applications Branch - Graphics Team (TAB-GT), Operations Support Directorate, TRADOC Analysis Command (TRAC) is located at Fort Leavenworth, Kansas. TAB-GT is responsible for graphics interface software development and maintenance, application software modification when compatibility problems are encountered, and assisting application programmers where a graphics interface is desirable. Currently the majority of application software packages maintained by TRAC and requiring graphics capabilities utilize a package known as the Kellner Graphics Interface Package (KGIP) named for its author Mr Al Kellner, TRAC-WSMR. This document discusses three major projects undertaken by TAB-GT to meet the previously mentioned responsibilities which led to the modification of the KGIP and eventual evolution of an even more sophisticated graphics interface package compatible with any TRAC VAX/Ramtek hardware configuration.

1. Purpose. The Technology Applications Branch - Graphics Team (TAB-GT), Operations Directorate, TRADOC Analysis Command (TRAC) at Fort Leavenworth Kansas has several responsibilities. These include recommending graphics hardware and software procurements, graphics interface software development and maintenance, applications software modification (when compatibility problems are encountered), and assisting application programmers where a graphics interface is desirable. This memorandum discusses three major undertakings (one of which remains ongoing) regarding these responsibilities which led to the evolution of a sophisticated graphics interface package.

2. Consolidate Graphics Software. This task partially solved a pressing software problem. TRAC currently has software that uses two different graphics packages. Neither of these packages are the current industry standard. These packages are the Color Graphics Interface Package (CGIP) and the Kellner Graphics Interface Package (KGIP). CGIP adheres very nearly to the old Tektronix IGL standard, while the KGIP adheres to no industry standard. Both packages work solely with Ramtek hardware. The problem was that neither package had the ability to handle the wide variety of possible Ramtek hardware configurations. A temporary solution was to change the application software so that CGIP or KGIP would work with Ramtek hardware at various sites. Previous attempts at a permanent fix to this problem consisted of duplicating these graphics packages in their entirety, and changing a very small portion thereof for KGIP or CGIP to function correctly on a given Ramtek configuration (i.e. a separate stylized graphics package had to accompany each application package for each hardware configuration that might be encountered). This resulted in numerous versions of each graphics package and consequently a file management nightmare.

a. Approach. For each graphics package, CGIP and KGIP, TAB-GT sorted into categories of redundant, modified, and unique, the modules in the several versions of that package. We kept only one version of the redundant modules in a common library for that particular graphics package. We made smaller libraries for the modules unique to each iteration of that particular graphics package.

b. Result. The result of this operation yielded for each graphics package one main library of modules, plus several sub-libraries. Application software run on any Ramtek configuration would require software from the one main library and, for a specific Ramtek configuration, it would require software from one or more of the smaller libraries. For the user's convenience, option files were developed to facilitate application software linking. There was (no longer necessary after completion of paragraph 4 phase) a single, mnemonic option file for each graphics package. The result of this project was a drastic reduction in the number of graphics modules on-line (65 percent) due primarily to the elimination of redundant modules, and a set of graphics libraries whose contents pointed out how a graphics package needs to distinguish among Ramtek hardware configurations.

3. <u>Construct OMNI for Graphics Education.</u> The Ramtek 9400 series software course uses three basic teaching materials: an instruction book, a Ramtek reference manual, and a software tool called OMNI. The instruction manual is organized by type of graphics instructions, and is the primary source of lecture material and lab exercises. The Ramtek reference manual shows the general Ramtek instruction format, and details about each individual instruction, but does not relate them. OMNI is required to do the lab exercises for the class.

a. Need. At Ramtek in Santa Clara, OMNI is an executable program written in "C" that enables direct communications with the Ramtek hardware. Its purpose is to teach through application the individual Ramtek instruction formats, and the interaction between them. OMNI does not use any graphics package to communicate with the Ramtek system. This is an advantage in two ways. The independence from such packages means that the test exercises for class will not be clouded with a layer of ambiguity by a complex graphics package. Also, the information taught is strictly Ramtek related, without concern for a third party graphics package: this is a benefit, since the students' exposures to the many graphics packages available varies widely. To use OMNI to send instructions to the Ramtek, you do the following things.

(1) Prepare a text file, containing OMNI code. This OMNI code is a human-oriented representation of Ramtek instructions, made from a set of mnemonics and hex, decimal, and ASCII constants.

(2) Run OMNI. Select an input file of OMNI code, and a Ramtek workstation. OMNI will input this file, and then convert the OMNI code to actual Ramtek instructions, and send those instructions to the selected Ramtek workstation.

(3) Observe the Ramtek response to the converted OMNI code. Valid instructions will produce proper Ramtek responses, to draw graphics, text, images, etc. Improperly formatted instructions or sequences of instructions will produce an error message on the VAX user terminal. These messages are documented in the course instruction manual and in the Ramtek reference manual. OMNI can also read back data from the Ramtek system, after issuing a request for such data. This works with keyboard reads, tablet reads, cursor reads, and reads from the Ramtek memories for display lists, video lookup tables, etc.

b. Development. Since OMNI is not a supported software package, Ramtek would not give or sell it to TRAC. The student takes home only the Ramtek course book and reference manual. All the examples in the course book are based on OMNI. Without OMNI there is no easy way for anyone learning the Ramtek graphics to directly apply his knowledge of the machine or to relate to the examples in the Ramtek software instruction manual. TAB-GT wrote a FORTRAN version of OMNI at TRAC thus providing our analysts with a research tool for further experimentation with the unexplored features of the Ramtek, and novice Ramtek graphics aspirants with the last piece of the puzzle to aid in self-instruction. TAB-GT has made OMNI available to users on the TRAC-FLVN VAX's by defining a system logical called "OOMNI". A user can execute a command file by entering @OOMNI:OMNI, and then activate OMNI by entering OMNI <file> /DEV=<Ramtek device-name>.

4. Fort Leavenworth Improved KGIP (FLIK). Currently the majority of TRAC's application software uses the Kellner Graphics Interface Package to communicate with the Ramtek graphics display hardware. Though the consolidation effort (paragraph 2) served to eliminate redundancy in the many hardware specific versions of KGIP, subtle differences were still required for specific hardware compatibility. TAB-GT designed and implemented several features which make the interface package (newly named FLIK) more "intelligent/flexible" and efficient. Our ultimate goal was to create a single interface package which would allow any application software to execute on any VAX/Ramtek configuration regardless of the hardware configuration for which it was intended. Several of these features are discussed in moderate detail in the following paragraphs.

a. Hardware Configuration Determination Part I. Using OMNI, TAB-GT researched the methods for a host to read back from a Ramtek some information on the machine's configuration and implemented these concepts in CGIP and FLIK, so that now each graphics package can determine some of the characteristics of the hardware to which it is attached thus reducing some of the user's burden caused by changing among Ramtek hardware and sites.

This configuration information is returned in two forms from the Ramtek. All systems give the host the set of normal format parameters, including refresh resolution while the MC68000 systems also return to the host more detailed information on the chassis configuration. Using OMNI showed us that reading the latter information caused undesirable color changes, which we were able to correct when implementing in CGIP and KGIP.

b. Hardware Configuration Determination Part II. Each Ramtek hardware chassis may drive more than one graphics station. A graphics station usually consists of a color monitor with its own unique picture, a graph tablet, and an optional keyboard. Since all of the configuration information necessary to provide FLIK with the required level of intelligence is not available through the read back, TAB-GT took the following approach.

(1) Define workstation data. TAB-GT defined a data set that is necessary and sufficient to describe a graphics workstation. We then designed and developed the standalone software to perform a one time installation of FLIK creating this data set for the specific hardware setup at each Ramtek site. The data resides on a single disk-file for each CPU using the graphics packages. That file describes all of the stations on Ramteks connected to that CPU.

(2) Access this data from FLIK. TAB-GT defined a set of COMMON blocks (contained only in the graphics interface package itself), and a module to read into them the station descriptions mentioned above. This allowed us to completely eliminate the libraries of parallel modules in KGIP that had previously been necessary to handle various Ramtek configurations. Consequently, FLIK is all located on one library, and tests, whenever necessary, the current station's description to determine how to interface with the Ramtek hardware. Once we enabled CGIP and KGIP to read this information, these packages assumed some of the burden that had previously been placed on the application software. CGIP and KGIP now work properly, despite differences in screen resolution.

(3) Advantages. Without these upgrades, the user, and consequently the software needs to consider such things as: how to drive that station's picture, which memory control processor to use, which memory group to connect to, which memory planes to write to, which cursor to use, which lookup table to use, is the lookup table for four or eight-bit color system, choice of cursor-controller: tablet or joystick, memory resolution, memory depth, keyboard availability.

c. Intended hardware configuration determination. All application software must first call an initialization routine to interface with the Ramtek display hardware. TAB-GT modified the FLIK initialization routine to accept information concerning the configuration of the hardware on which the application package was intended to be run. This enables FLIK to interface with software designed to be run on a variety of VAX/Ramtek configurations.

d. Reduce host graphics load. This project is still ongoing. A smart graphics processor such as the Ramtek can handle a considerable portion of the computing demand for graphics display processes. Normally these algorithms are implemented on the host because of ease of implementation in a known high-level programming language. Many of the graphics processors now available, such as the Ramtek 9000 series at TRAC, do not provide a high-level programming language. Consequently, to offload graphics burdens from our host computers, we must turn to several complex solutions. These are dependent on the type of graphics operations involved.

(1) Symbology. For symbology, made of ICONs, save these line drawings in display list memory on the Ramtek, with only a directory of those ICONs on the host. From the host, draw the picture by its reference number only.

(2) Text. For text at continuously variable size and rotation, treat text as ICONs, and store and draw as symbols, as described above.

(3) Menus. It is possible to download menu processing from host to Ramtek, as follows. For display of menus, move the menu description from

host to Ramtek memory and write a display list program to accept cursor points. Find the menu box selected by the points, then backlight that box and return the box selection to host.

(4) Point data. Download these data directly without conversion to a polygon description by the host. Put these point data directly into Ramtek refresh memory. Then use a display list program to do the polygon fill to scale the image up or down by a non-integer increment (scaling to integer increments is available now on the MC68000 Ramtek using the copy-image and multiply instruction)..

(5) Image data. You can't download saved image displays to the Ramtek. You need a graphics engine with a hard disk attached. The only possible way to expedite transfer of entire images from host to Ramtek and back is with double-buffering techniques, which TAB-GT has tested and implemented in the Vector-in-Commander Input Preprocessor (VIP). This technique better than halves the time to transfer images from host to Ramtek.

Note that most of these more elaborate downloading schemes require writing display-list programs for the Ramtek. The only language available for this purpose is that used in OMNI, the teaching tool described in paragraph 3. TAB-GT plans to write such a program in OMNI to test the concept and, after modification, implement it in a graphics package.

There is one significant problem with display-list programming TAB-GT plans to encounter. The "object" code to download into a Ramtek as a display-list program is full of branch addresses, and memory-location addresses for load/store operations. There are also references to segments of the display-list RAM called display-lists 0-15. These references are all built into the "object" code, and therefore the "object" code is not relocatable. This precludes repositioning of one or more such display-list programs in the Ramtek when a user requires a different combination of such programs. Also, the creator of the program needs to work in absolute addresses and references, which makes the creation of these programs a real chore.

Currently, neither the KGIP nor CGIP package handles display-list programming. We propose an elaborate solution to the problem, whereby a body of modules could be added to either package that would permit the defining of a display-list program, and subsequent loading. Also, within the program, there would be other modules that would do branching and memory references that referred to relocatable addresses. The graphics packages would handle external references to other display-lists also. These would be handled in symbol tables for external and local references. This is a major project, but it will enable a graphics package to generate display-list software to run on the Ramtek, without the usual inconvenience to the user.

4. Summary. The Technology Applications Branch - Graphics Team in an effort to better meet the graphics needs of the TRADOC Analysis Command

undertook a series of projects which led to the development of a sophisticated graphics interface package. This package evolved from one currently embedded in the vast majority of TRAC application software known as the Kellner Graphics Interface Package (KGIP). To preserve recognition for the original author yet distinguish the new package due to its significant advancements the name was changed to the Fort Leavenworth Improved KGIP (FLIK). FLIK is designed to interface with any software application package run on any VAX/Ramtek hardware configuration regardless of the configuration on which it was intended to be run. This flexibility allows the novice applications programmer to "hardwire" his screen size related draws and use either routines requiring screen coordinate input or virtual coordinate input (though the latter is still strongly encouraged).

END

10-87

DTIC