AD-A221 133

Research Product 90-05

Design of a Joint Service Multipurpose Arcade Combat Simulator (JMACS)

January 1990

Field Unit at Fort Benning, Georgia
Training Research Laboratory

U.S. Army Research Institute for the Behavioral and Social Sciences

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

JON W. BLADES
COL, IN
Commanding

Naval Training Systems Center

H. C. Okraski
Deputy Technical Director

E. L. LEWIS
CAPT, USN
Commanding

Technical review by

Jean L. Dyer
Franklin L. Moses

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) ARI Research Product 90-05	5. MONITORING ORGANIZATION REPORT NUMBER(S) --
6a. NAME OF PERFORMING ORGANIZATION U.S. Army Research Institute Fort Benning Field Unit 6b. OFFICE SYMBOL (If applicable) PERI-IJ	7a. NAME OF MONITORING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences
6c. ADDRESS (City, State, and ZIP Code) P.O. Box 2086 Fort Benning, GA 31905-0686	7b. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600
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.
64722D	750	517	C1

11. TITLE (Include Security Classification)
Design of a Joint Service Multipurpose Arcade Combat Simulator (JMACS)

12. PERSONAL AUTHOR(S) Evans, Kenneth L., (ARI); Marshall, Albert H., & Wolff, Ronald S. (Naval Training Systems Center); Broom, John M., & Greene, William H. (Litton Computer Services)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 85/10 TO 89/09	1990, January	

16. SUPPLEMENTARY NOTATION
Prepared in cooperation with the Naval Training Systems Center (Orlando, FL) and Litton Computer Services (Fort Benning, GA).

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP
05	06	
19	06	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Computer simulation, Instructional materials, Rifle marksmanship, Training aid. (SDW)

19. ABSTRACT (Continue on reverse if necessary and identify by block number) This report documents the design of the hardware and software used in the Joint Service Multipurpose Arcade Combat Simulator (JMACS) project. The JMACS project was an outgrowth of the Multipurpose Arcade Combat Simulator (MACS), a patented part-task weapons trainer developed by the U.S. Army Research Institute for the Behavioral and Social Sciences to overcome the training problems encountered due to insufficient facilities, ammunition, and qualified instructors. A MACS system consists of a long-distance light pen mounted to a weapon or mock-up, a microcomputer, a video monitor, and a software cartridge. Following testing of prototype systems in the Army, the Navy and Air Force became interested in potential applications of MACS technology for weapons training programs. This interest led to the initiation of the JMACS project, sponsored by the Joint Service Manpower and Training Technology Development Program. The JMACS project funded the development of a long-distance light pen (patent pending) and light pen mount by the Naval Training Systems Center, introduced MACS technology to the Air Force and Navy in a series of field tests, and supported ongoing MACS development and evaluation in the Army. Keywords:

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Seward Smith	22b. TELEPHONE (Include Area Code) (404) 545-1278 22c. OFFICE SYMBOL PERI-IJ

DD Form 1473, JUN 86 *Previous editions are obsolete.*

Research Product 90-05

Design of a Joint Service Multipurpose Arcade Combat Simulator (JMACS)

Kenneth L. Evans
U.S. Army Research Institute

Albert H. Marshall and Ronald S. Wolff
Naval Training Systems Center

John M. Broom and William H. Greene
Litton Computer Services

Field Unit at Fort Benning, Georgia,
Seward Smith, Chief

Training Research Laboratory
Jack H. Hiller, Director

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

January 1990

Army Project Number
2Q464722D750

Education and Training Systems

FOREWORD

The Joint Service Multipurpose Arcade Combat Simulator (JMACS) project was an outgrowth of the Multipurpose Arcade Combat Simulator (MACS), a patented part-task weapons trainer developed by the Fort Benning Field Unit of the U.S. Army Research Institute (ARI) to address training problems arising from insufficient facilities, ammunition, and numbers of qualified instructors. The MACS system consists of a microcomputer, video monitor, software, and long-distance light pen mounted to a weapon or mock-up. Although MACS is inexpensive, it has been found to be as effective as more costly training devices, particularly in the area of rifle marksmanship training.

Following Army testing of prototype systems, the Navy and Air Force became interested in potential training applications of MACS technology. This interest led to the initiation of the JMACS project, sponsored by the Joint Services Program (PE64722A) on Manpower and Training Technology Development. The JMACS project funded the development of an improved light pen (patent pending) and light pen mount by the Naval Training Systems Center (NTSC), introduced MACS technology to the Air Force and Navy in a series of field tests, and supported ongoing MACS development and evaluation in the Army.

This report documents the design of the hardware and software used in the JMACS project, as well as more recent developments in the areas of light pen and software design. Three organizations cooperated in its preparation: the ARI Fort Benning Field Unit, its resident scientific support contractor, Litton Computer Services, and the Advanced Simulation Concepts Laboratory of NTSC. The contributions of each organization were crucial to the overall success of the project.

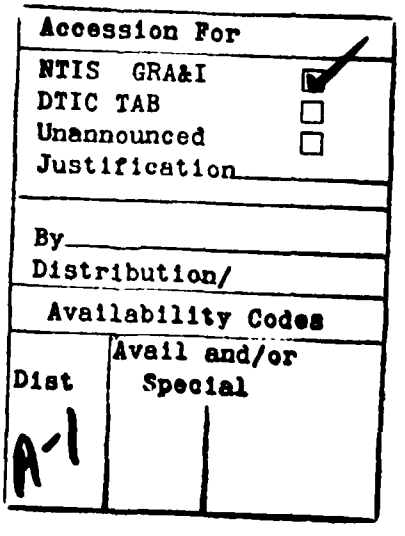
The Navy, Air Force, and Coast Guard have maintained interest in this technology through a number of demonstration projects that evolved from the original JMACS field tests. However, the Army has been most vigorous in its pursuit of the training benefits resulting from MACS use. With the support of the senior leadership of the U.S. Army Infantry School, the U.S. Army Forces Command, and the U.S. Army Training and Doctrine Command, over 400 MACS M16A1 rifle systems have been fielded Army-wide to date. It should be noted that all MACS hardware components are now available commercially, due to the efforts of the Fort Benning Training Support Center in developing MACS contract specifications. By duplicating the latest MACS software and obtaining components from commercial sources, the Fort Benning Training Support Center has been able to offer MACS M16A1 rifle systems at a total cost of $950 per system (plus shipping) to requesting units and schools.

H. C. OKRASKI
Deputy Technical Director
Naval Training Systems Center

EDGAR M. JOHNSON
Technical Director
Army Research Institute

ACKNOWLEDGMENTS

Portions of this research were sponsored by the Joint Service Manpower and Training Technology Development Program (PE64722A). Mrs. Ellen Wiley of Litton Computer Services provided the computer graphics presented in this report.



DESIGN OF A JOINT SERVICE MULTIPURPOSE ARCADE COMBAT SIMULATOR (JMACS)

CONTENTS

LIST OF FIGURES

DESIGN OF A JOINT SERVICE MULTIPURPOSE ARCADE COMBAT SIMULATOR (JMACS)

Introduction

This report documents the design of the hardware and software used in the Joint Service Multipurpose Arcade Combat Simulator (JMACS) project. The JMACS project was an outgrowth of the Multipurpose Arcade Combat Simulator (MACS), a patented part-task weapons trainer developed by the Fort Benning Field Unit of the U.S. Army Research Institute in an effort to overcome the training problems sometimes encountered due to insufficient facilities, ammunition, and numbers of qualified instructors (Schroeder, 1984). A MACS system consists of a long-distance light pen mounted to a weapon or mock-up, a microcomputer, a video monitor, and a software cartridge (see Figure 1). As targets are presented on the monitor, the light pen determines the firer's point of aim and a switch mechanism attached to the weapon's trigger sends an electrical signal to the microcomputer when each shot is fired. Performance feedback to the firer is presented on the monitor after each shot or series of shots.

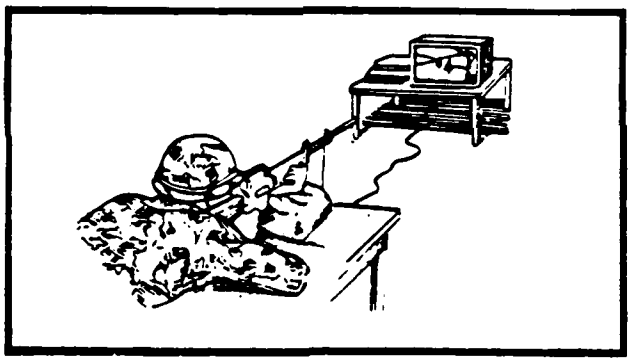


Figure 1. The Multipurpose Arcade Combat Simulator (MACS) configured as an M16A1 rifle marksmanship trainer.

MACS is capable of providing part-task training for a variety of weapon systems because the light pen can be moved from one weapon system to another accompanied by a change in the software cartridge. Further, it is relatively inexpensive to produce, because many of the components are available commercially. Due to the uniqueness of a weapons training device based on a long-distance light pen, the Army filed a U.S. patent application in 1984 that was awarded in 1986 (Schroeder, 1986). Following testing of prototype MACS systems by the U.S. Army Research Institute and the U.S. Army Infantry Board between 1982 and 1986, the U.S. Navy and U.S. Air Force became interested in the potential applications of MACS technology in their respective weapons training programs.

Navy and Air Force interest led to the initiation of a two-year Joint Service MACS (JMACS) project in fiscal year 1986 sponsored by the Joint Service Manpower and Training Technology Development Program. The JMACS project accomplished several objectives.

First, it funded the development of a long-distance light pen (patent pending) and light pen mount by engineers at the Naval Training Systems Center. Compared to the single-lens light pens used in earlier MACS prototypes, this light pen used a dual-lens focusing system that permitted its overall length and weight to be reduced by approximately one third. Further, it yielded more accurate light pen readings when in close temporal proximity to trigger switch closure, because its electronic circuitry was completely separate from that of the trigger switch. Second, the JMACS project funded the purchase of 25 MACS systems configured for the M16A1 rifle. Equipped with the Naval Training Systems Center light pen and mount, these systems were used for initial testing in the Navy and Air Force, as well as for support of the ongoing MACS development and evaluation program in the Army. Third, demilitarized M16A1 rifles were obtained from the Anniston Army Depot for the JMACS project to provide greater realism and durability than previously used mock-up rifles. Mock-ups contained lower receivers made of injection-molded plastic instead of metal. A pushbutton switch that more accurately replicated the force of pulling an actual M16A1 rifle's trigger also contributed to better realism.

Although current interest has focused on the use of MACS as an M16A1 rifle marksmanship trainer, prototype systems with demonstration software also have been developed for the M72A2 light antitank weapon (LAW), M203 grenade launcher, Mark 19 grenade machinegun, and the M136 (AT-4) LAW. Since exploratory research began on the MACS concept in 1982, over 20 developmental hardware tests, training and cost effectiveness evaluations, and informal field investigations have been conducted. Evans (1988) identified five major research trends in summarizing the history of MACS development and evaluation, including the JMACS project:

1. MACS performance is at least moderately related to both live-fire marksmanship performance and performance on a high-fidelity simulator.

2. In relatively extensive training programs over one week in duration, firers who receive MACS training in addition to standard training tend to perform slightly higher than those who receive standard training alone.

3. In less extensive training programs completed in one day, firers who receive a meaningful amount of MACS training prior to standard training tend to perform significantly higher than those who receive standard training alone.

4. The effectiveness of MACS training is greater for those individuals or groups having a lower initial level of ability.

5. MACS use in training is associated with a consistent reduction in the percentage of firers failing to meet minimum performance standards and with a significant reduction in subsequent ammunition expenditures.

In addition to these research trends, MACS use appears to have a number of intangible benefits in preparatory, remedial, and sustainment training (Evans, 1988). MACS provides precise and immediate performance feedback in a dry fire training environment, where the usual amount of performance feedback provided to instructors and firers is severely limited. With MACS it is now possible to provide a more standardized method of dry fire training and to establish more meaningful minimum performance

standards during this instruction. Another intangible benefit is that MACS training seems to be more enjoyable and interesting than conventional dry fire training methods, as student questionnaire responses have consistently indicated highly positive opinions of MACS training. Evans (1988) concluded that the benefits of MACS training exceed the costs associated with its use.

Hardware Design

All MACS hardware originally consisted of off-the-shelf components, except for the light pen and light pen mount for the M16A1 rifle. These latter two items were designed for the JMACS project by engineers at the Naval Training Systems Center.

Microcomputer

Although the Apple II series of microcomputers had been used in early MACS prototypes, the Commodore 64 microcomputer was selected later for its lower cost and its additional built-in features that included sprite graphics, a cartridge port, and light pen/joystick ports. It should be noted that the Commodore 64 is no longer manufactured. This model was replaced with the Commodore 64C, which has been found to be fully compatible with the original. However, it would be possible to adapt almost any microcomputer having graphics for MACS use, given a light pen interface and compatible software.

Monitor

A Commodore 13-inch color monitor, the most current model being the 1802, is recommended for the MACS M16A1 rifle system. A monitor having this screen size provides for the proper scaling of computer-generated targets when viewed at a distance of 10 ft (3.05 m). If larger monitors are used, the distance to the screen should be increased accordingly. However, the light pen has not been tested for accuracy beyond a distance of 25 ft (7.62 m). In reality, many color monitors or televisions having a degausser and microcomputer-compatible cable could be used.

Program Cartridge

MACS M16A1 rifle software, written in a combination of BASIC and 6510 assembly language, is adapted for use with an erasable programmable read only memory (EPROM) cartridge that is inserted in the cartridge port of the Commodore 64/64C microcomputer. The primary advantage of cartridge-based software is that it eliminates the need for, as well as the cost of, a disk drive in the hardware configuration. In addition, it speeds program loading and eliminates the need for typing loading commands on the microcomputer keyboard. Although a variety of program cartridges have been developed, the one for basic rifle marksmanship training uses three EPROM microchips (32k bytes x 8 bits each). All MACS cartridges consist of a plastic housing, a bank switch board with four sockets, and the EPROMs themselves. Obtained from the Jason-Ranheim Company of Auburn, California, the components required for one basic rifle marksmanship cartridge are one #PCCH-2 housing, one #PCC-4 bank switch board, and three #27256 EPROMS (either 12.5 or 21 V). When the bank switch board is used with 32k EPROMS, as is the case with

the basic rifle marksmanship cartridge, simple modifications must be made to the board in accordance with instructions provided by Jason-Ranheim.

Light Pen

A key component of the MACS system is its long-distance light pen (patent pending), developed for the JMACS project by the Advanced Simulation Concepts Laboratory of the Naval Training Systems Center. This light pen, like all light pens, operates by detecting a scanned dot of light on the microcomputer's monitor. The instant the scanned dot of light is detected, a voltage pulse is transmitted from the light pen to the microcomputer. Because the microcomputer controls the scanned dot movement, it can then determine the position on the monitor where the light pen is pointed. The Naval Training Systems Center long-distance light pen, unlike other light pens, focuses the image of the scanning beam on its photodetector with a dual-lens telescope, which has a focal length that limits the area that can be detected on the face of the monitor. The telescope permits the light pen to operate at distances from the monitor that are far greater than those of ordinary light pens. Specifically, the long-distance light pen has been tested successfully in the laboratory at distances up to 25 ft (7.62 m). Compared with the single-lens light pen used in earlier MACS prototypes, the use of two lenses also allows the overall length, as well as the weight, of the light pen to be reduced by approximately one third.

Unique to the design of the Naval Training Systems Center light pen is its signal amplification and noise reduction circuitry (see Figure 2). Its photodetector diode (shown as D1 in Figure 2) is a current generator. A transimpedance amplifier (A1) produces a small signal as its output. The signal at A1 is the product of the diode current and the resistor at R1. This very small signal is then filtered and AC-coupled to the first amplifier stage (A2). The signal from A1 is filtered by a second-order high-pass butterworth filter (B1). Designed to have a cut-in frequency of about 14 kHz, this filter removes any noise generated by the power supply or by ambient light sources. Any noise sources below 1 kHz are reduced significantly by a factor of about 40 decibels. Next, the signal is amplified by a simple two-stage amplifier (B2 and B3). The amplified signal is now at a level of about 1 to 3 V, depending on screen brightness. Finally, the signal is applied to the final operational amplifier (B4), which is used as a voltage comparator. The threshold level is set by a potentiometer (R13). The comparator is used to switch from 5 VDC to 0 VDC upon detecting the scanned electron beam from the monitor. The output of B4 is normally high. When the amplified signal from the photodetector is greater than the present threshold on the comparator, the comparator will go low. When observed with an oscilloscope, the output of the light pen appears as a periodic pulse train, with pulses going from a logical one to a logical zero.

Subsequent to the completion of the JMACS project, the Naval Training Systems Center modified the circuitry of the light pen slightly (see Figure 3). These design improvements were found to improve its output signal stability and ease of calibration. Documentation related to the design of the light pen's circuit board and mechanical drawings of the light pen housing are on file at the Advanced Simulation Concepts Laboratory of the Naval Training Systems Center.

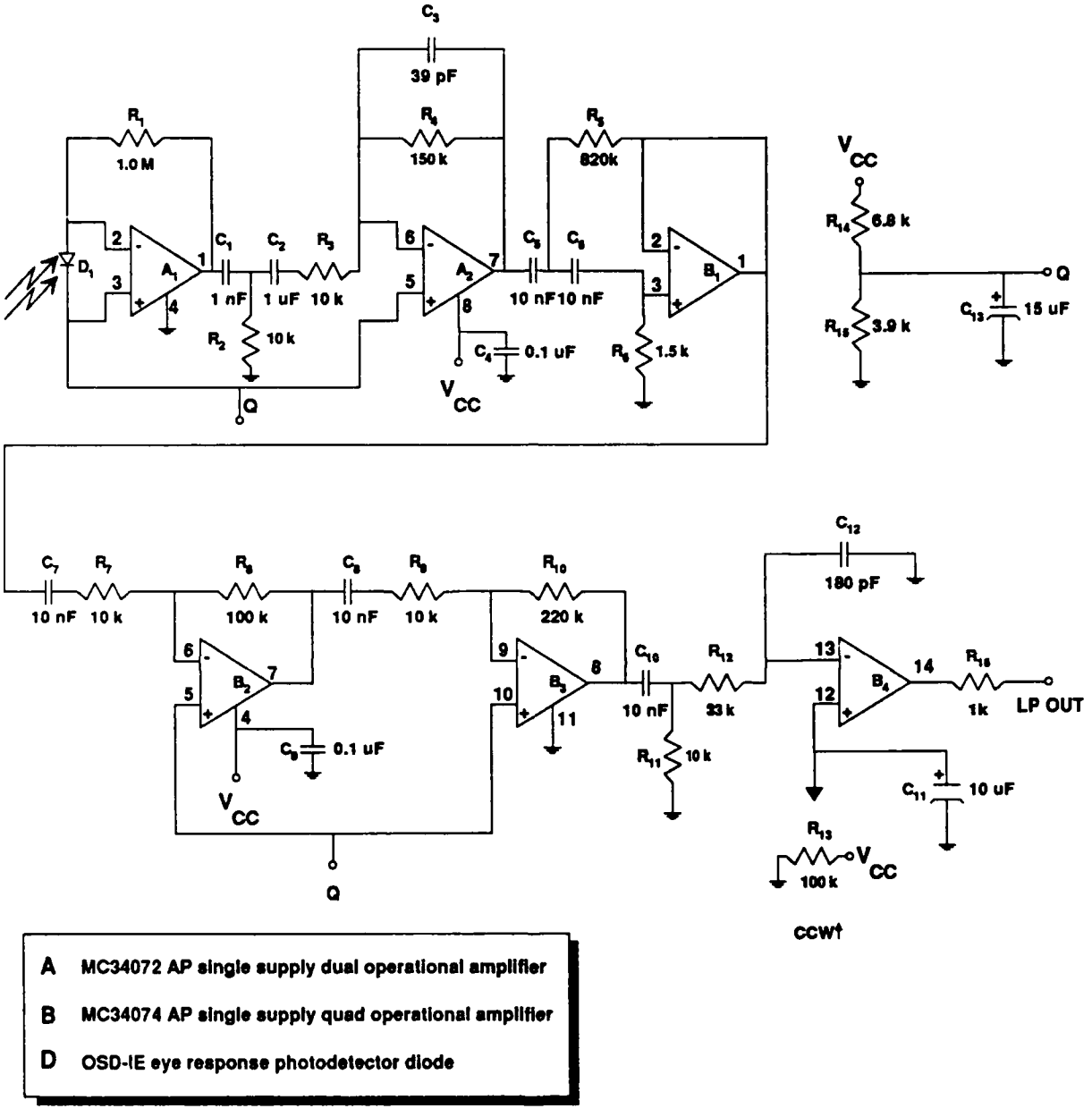


Figure 2. Schematic of the Naval Training Systems Center long-distance light pen (5 VDC power supply).

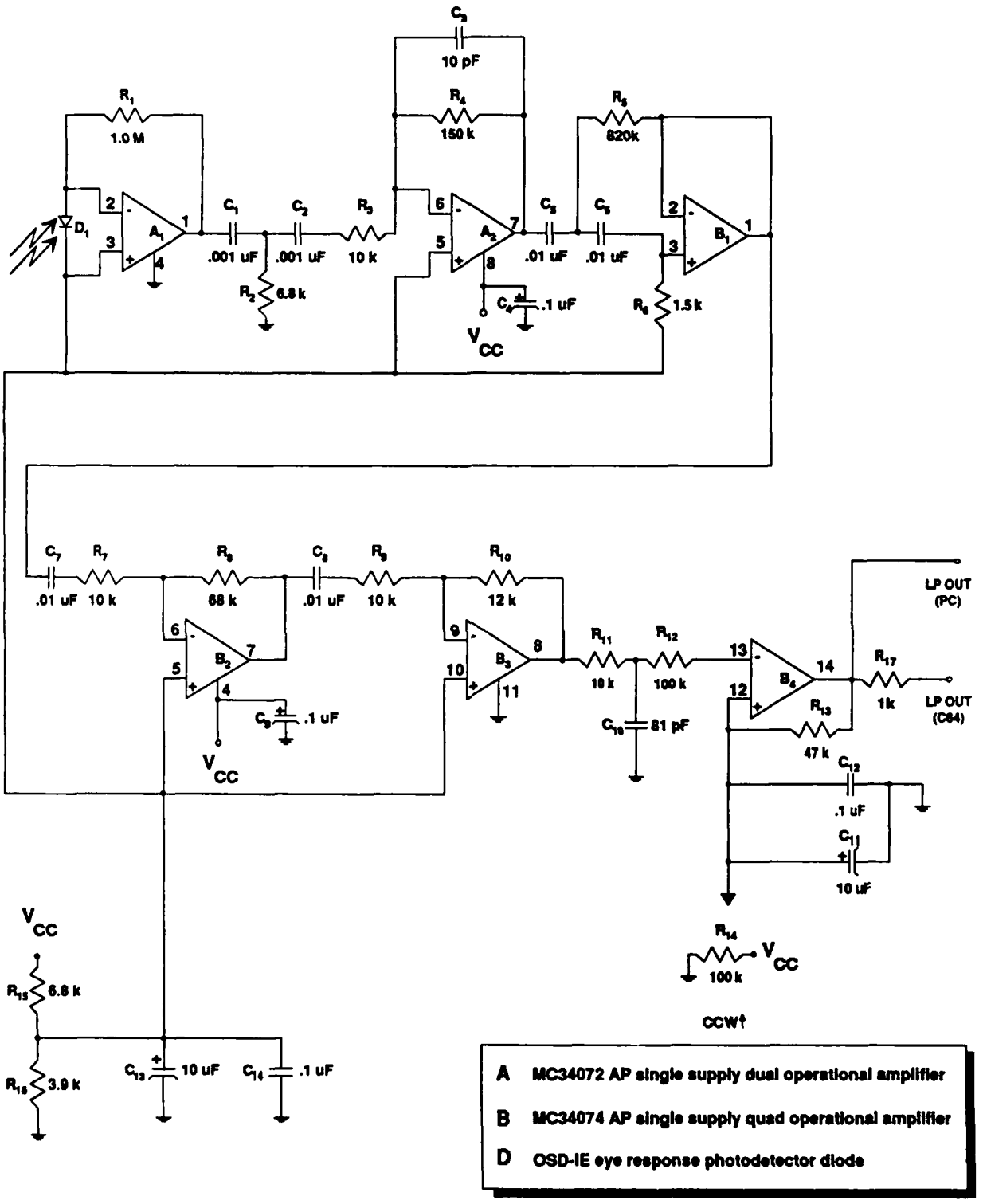


Figure 3. Schematic of the Naval Training Systems Center improved long-distance light pen (5 VDC power supply).

Light Pen Mount

The Naval Training Systems Center also designed a light pen mount for the M16A1 rifle. Made of anodized aluminum, the mount is designed to position the light pen above the barrel and in front of the front sight assembly. Although the mount is connected to the bayonet stud, to prevent rotation around the barrel, it does permit the light pen to be aligned with the sights in both the vertical and horizontal axes. Mechanical drawings of the light pen mount for the M16A1 rifle are on file at the Advanced Simulation Concepts Laboratory of the Naval Training Systems Center.

Rifle and Trigger Switch

Demilitarized M16A1 rifles were obtained from the Anniston Army Depot for the JMACS project. Retaining their metallic lower receivers, these rifles offered greater durability and realism than previously used mock-up rifles whose lower receivers were made of injection-molded plastic. Within the upper receiver of each demilitarized rifle an Arrow Hart single-pole single-throw switch (model #83090-C) was installed in an opposing position to the trigger in the lower receiver. Complete documentation on the trigger switch and its wiring has been provided in an earlier report (Hunt et al., 1987).

Software Design

MACS software for Basic Rifle Marksmanship (BRM) training evolved over a period of several years, as did MACS hardware in general (see Evans, 1988). The version of MACS BRM software that was used in the JMACS project introduced several features that had not been available in earlier versions. This fourth-generation software featured ten levels of instruction, arranged in an ascending hierarchy of difficulty. After an initial skill test determined the most appropriate starting level for each firer, the program branched automatically between levels based on firer performance. The amount and type of performance feedback also was geared to firer performance and instructional level, with performance feedback maximized in the earlier levels and gradually decreased at the higher, more difficult, levels. Seven major types of performance feedback were featured in this software:

1. crosshairs indicating the location of hits and misses

2. descriptive diagnostic scores based on normative performance standards for various marksmanship fundamentals

3. an animated replay of a shot comparing the firer's actual sight picture with the correct sight picture

4. a screen border that changed color when performance standards were not met

5. a whistle sound indicating the failure to fire within a prescribed time limit

6. summary tables of the number of hits, misses, and "no fires" obtained at each target range

7. delayed recall of the locations of previously-fired shots, shown as a collective group of shots at each target range.

This performance feedback package, together with associated internal performance standards, made self-paced MACS instruction possible for the first time. In addition to its evaluation in the Navy and Air Force as part of the JMACS project (see Evans, 1988; Heller & Evans, 1989), the fourth-generation software also was evaluated for its training effectiveness in two subsequent U.S. Army Infantry Board (USAIB) Concept Evaluation Program (CEP) tests (USAIB, 1987, 1988). With the exception of the BASIC program listing and accompanying 6510 assembler source code, which is presented in Appendix A, detailed documentation of this software has been presented in an earlier report (Hunt, Broom, & Greene, 1988).

Recently a fifth generation of MACS software was developed, largely in response to shortcomings identified in the aforementioned tests and in later research (Broom et al., 1989). While similar to fourth-generation software in most respects, the fifth-generation software for BRM training differed in four areas. First, the diagnostic performance measures were revised to make them more sensitive to small firing errors and more consistent with the marksmanship fundamentals taught in U.S. Army BRM training. Second, many new information screens were added to enhance student understanding of program features in less-than-ideal training environments, where knowledgeable instructors may be unavailable in sufficient quantity. Third, one instructional level was removed from the program because its performance feedback was found to be confusing to some students. Fourth, optional instructional programs requiring the use of joysticks were revised so that the same concepts could be taught without the need for joysticks in the MACS hardware configuration. Complete documentation of fifth-generation software has been presented separately (Broom et al., 1989). A trainer's guide to accompany this software also has been developed (Purvis & Wiley, 1989).

Lessons Learned in Hardware and Software Development

In addition to previously mentioned specific improvements in light pen and software design initiated as a result of product testing and user feedback, several general lessons were learned during the course of hardware and software development that should be of interest to others working in the area of marksmanship simulation. First, simulated recoil was not incorporated into the MACS hardware configuration, despite the belief by some instructors that it is an important feature of any rifle marksmanship simulator. Consistent with earlier findings in the areas of classical conditioning and simulator fidelity, subsequent experimentation found that MACS performance did not vary as a function of recoil (Evans, 1989). Evans (1989) concluded that the accurate reproduction of recoil may be unnecessary in rifle marksmanship simulation, particularly if simulation is used in conjunction with live firing in an overall training strategy. In the case of MACS, it is estimated that the addition of recoil would at least double the cost of the system, with no expectation of any greater training effectiveness. Similar reasoning precluded the development of a realistic firing noise feature for MACS, though this issue was not investigated experimentally.

Another lesson learned was that some software features commonly found in video arcade games can dramatically increase student interest and system usage. For example, a software routine allowing high scorers to enter and display their initials was introduced in a five-target demonstration program used in MACS briefings and exhibitions. Prior to the introduction of the high scorer routine. most users would only fire the program once or twice, which was sufficient to become familiar with the major capabilities and features of the MACS rifle system. After the introduction of the high scorer routine, however, many users could be observed firing the program repeatedly (sometimes dozens of times) without any evidence of boredom. Due to the increased system usage and apparent competitive interest that resulted from the introduction of the high scorer routine in the demonstration program, a "top gun" routine for high scorers was added to the fifth-generation software for basic rifle marksmanship training (Broom et al., 1989; Purvis & Wiley, 1989).

Finally, it was learned that user acceptance sometimes can be influenced more by organizational factors than by factors solely related to the hardware and software features of a system. This was most apparent in a pair of U.S. Air Force training effectiveness evaluations conducted by Butzin as part of the JMACS project (cited in Evans, 1988). In these evaluations, a questionnaire administered to two groups of instructors yielded quite different opinions of MACS. Specifically, instructors in a one-week training program with a relatively low student throughput held much higher opinions of MACS hardware and software features than did instructors in a one-day training program having a relatively high student throughput. This disparate finding was largely attributable to the greater difficulty instructors in the one-day program experienced finding available time to integrate MACS training with their existing training.

Commercial Availability and Product Utilization

The U.S. Army Research Institute and the Naval Training Systems Center jointly provided 25 MACS systems to the Army (5 systems), Navy (10 systems), and Air Force (10 systems) for evaluation as part of the JMACS project. Since the conclusion of the project, these systems have been used for training within each service. Using the Naval Training Systems Center's light pen design, the Fort Gordon Training Support Center assembled over 100 additional MACS systems to support subsequent Army testing (USAIB, 1987, 1988). Since the conclusion of this testing, these systems also have been used in training at the U.S. Military Academy and Forts Benning, Jackson, Richardson, and Wainwright. Several of these systems were also used to support informal evaluations by services within allied nations and by other U.S. Army organizations.

Due to continuing interest in MACS technology, the U.S. Army has assembled and fielded over 300 production systems, at the present time, to requesting units and schools through the Fort Benning Training Support Center. These production systems were the first MACS systems in which all hardware components were obtained commercially. The Fort Benning Training Support Center developed contract specifications necessary to obtain preassembled light pen, mount, and replica M16A1 rifle packages, with the light pen based on the Naval Training Systems Center circuit design (see Figure 2). Two commercial firms have manufactured these packages to date: the BETAC Corporation of Arlington, Virginia, and Firearms Training Systems (FATS) of Norcross, Georgia. Commodore 64C microcomputers and 1802 monitors were obtained commercially on the open market. All

program cartridge components were obtained from the Jason-Ranheim Company of Auburn, California.

At a total cost of $950 per system (plus shipping) to requesting units and schools, the Fort Benning Training Support Center obtains all MACS hardware commercially, performs quality control checks on the equipment, duplicates a MACS BRM program cartridge from a master set of fifth-generation EPROMs (Broom et al., 1989), prints a trainer's guide (Purvis & Wiley, 1989), and then prepares the completed system for shipping. The Fort Benning Training Support Center also has Spanish language versions of the MACS BRM program and trainer's guide on file, with translation having been provided by the U.S. Army School of the Americas.

References

Broom, J. M., Champion, D. F., Greene, W. H., Martere, R. F., Purvis, J. W., & Sills, E. G. (1989). Multipurpose arcade combat simulator (MACS) basic rifle marksmanship (BRM) program (ARI Research Product 90-01). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A214 459)

Evans, K. L. (1988). Development and evaluation of the multipurpose arcade combat simulator: A research summary (ARI Research Report 1488). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD B130 099)

Evans, K. L. (1989). Effects of recoil on rifle marksmanship simulator performance (ARI Technical Report 828). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A206 984)

Heller, F. H., & Evans, K. L. (1989). Joint service multipurpose arcade combat simulator (JMACS) user guide (ARI Research Product 89-23). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A212 773)

Hunt, J. P., Broom, J. M., & Greene, W. H. (1988). Multipurpose arcade combat simulator (MACS): Year three report (ARI Research Note 88-86). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A199 468)

Hunt, J. P, Broom, J. M., Greene, W. H., Crawford, J. W., Martere, R. F., & Parish, J. R. (1987). Multipurpose arcade combat simulator (MACS): Year two report (ARI Research Note 87-34). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A184 179)

Purvis, J. W., & Wiley, E. W. (1989). Trainer's guide: Multipurpose arcade combat simulator (MACS) basic rifle marksmanship (M16 rifle) (ARI Research Product 90-03). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Schroeder, J. E. (1984). A multipurpose arcade combat simulator (MACS) (ARI Technical Report 629). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A156 795)

Schroeder, J. E. (1986). Light pen marksmanship trainer (U.S. Patent No. 4,583,950, April 22, 1986).

U.S. Army Infantry Board. (1987). Concept evaluation program (CEP) test of MACS device/LOMAH training (MACSD/LT) (USAIB Project No. 3868). Fort Benning, GA: Author.

U.S. Army Infantry Board. (1988). Concept evaluation program test of the M16 rifle Gowen South: Phase III (USAIB Project No. 3881). Fort Benning, GA: Author. (AD B124 423L)

Appendix A

BASIC Program Listing and 6510 Assembler Source Code for
Fourth-Generation Basic Rifle Marksmanship Software

```
10 POKE 808,237:V=53248:POKE 53281,0:POKE V+21,0:B$="  ":SL=1:EL=10
20 DIM SD%(25,3),CR%(9,3),H%(8),DA%(6,10):I1=43640:I2=43984:NU=0:EZ=0:CS=0
30 DEF FNH(X)=INT(X/256):DEF FNL(X)=X-INT(X/256)*256
40 DEF FNW(X)=PEEK(X)+PEEK(X+1)*256:DEF FNZ(X)=INT((ABS(X>0)*X)+.5):C=2
45 POKE 56579,143:POKE 56577,0
50 RK=256:EN=3400:DS=3403:RT=3406:SV=3409:BANG=3415:SC=3421:DV=3430:LC=3433
60 GP=3436:CO=3439:WH=3442:MA=3445
70 GET A$:IF A$="" THEN 70
80 IF A$=" " THEN POKE 53280,0:SYS DS:GOTO 130
90 IF A$="m" THEN SYS DS:GOSUB 3030:GOTO 130
100 B$=RIGHT$(B$,1)+A$:IF (B$<>"ca") AND (B$<>"pf") THEN 70
110 Z=1:IF B$="pf" THEN Z=2
120 SYS DS:POKE 53265,27:POKE 53272,23:PRINT"{CLR}";:POKE 49161,Z:SYS 49179
130 IF RK<256 THEN 210
140 A$="{F3}09{F6}00{F5}00{F7}00{CLR}{LBLU}{F2}0504{F1}Welcome to {RED}MACS{LBLU}{F3}00":GOSUB 2830:BC$="{F6}00"
150 POKE 53281,0:FOR I=0 TO 9:FOR J=0 TO 3:READ CR%(I,J):NEXT J,I
160 FOR I=1 TO 6:FOR J=0 TO 4:READ DA%(I,J):NEXT J,I
170 Z=0:SB=15:GOSUB 5240:RK=Z
180 IF RK=0 THEN 210
190 POKE 53265,27:POKE 53272,23:INPUT"{CLR}Date";A$:IO$=LEFT$(A$,8)+CHR$(13)
200 INPUT"Unit";A$:IO$=IO$+LEFT$(A$,20)+CHR$(13)
210 S1=32768:S2=10:POKE 929,0:CL=0:MO$="a":GOSUB 1000:IF CL>EL THEN CL=EL
220 IF CL<SL THEN CL=SL
230 GOSUB 2950:IF CL=0 THEN 210
240 IF(WI AND 16)THEN WI=INT(RND(1)*5):WI=16+((WI-(WI-4)*2))+(INT(RND(1)*8)*32)
250 POKE 929,WI:ON CL GOSUB 1480,1480,1680,1680,1990,1990,1990,1990,2330
260 MO$="a":IF (CL<=EL) AND (SL<>EL) THEN 220
270 A$="{F3}09{F6}05{CLR}{F5}05{F7}05{BLK}{F2}0502{F1}CONGRATULATIONS{F8}{F8}  You are finished{F8}  with this "
280 A$=A$+"program.{F8}{F8}  Call instructor.{F1}{F2}0723Press SPACEBAR to continue{F3}00"
290 GOSUB 2830:CL=0:POKE 198,0:RESTORE:GOTO 70
1000 IF RK=0 THEN 1020
1010 POKE 53272,23:POKE 53265,27:INPUT"{CLR}{WHT}Firer ID #";ID$:ID$=LEFT$(ID$,20)
1020 POKE 2040,34:POKE 2047,37:POKE V+14,10:POKE V+15,130:POKE V+39,0
1030 POKE V+46,12:POKE V,21:POKE V+1,145:POKE V+16,129:POKE V+23,0:POKE V+29,0
1040 POKE 53265,43:A$="{F6}00{F5}13{F7}00{CLR}{F1}{BLK}{F2}0701The following{F2}0103targets are untimed"
1050 A$=A$+"{F8}{F8} Aim at the{F8} center of the{F8} target as{F8} the black dot{F8} shows{F1}"
1060 TN=0:POKE V+21,129:A$=A$+"{F2}0723<Pull trigger to continue>{F3}00":GOSUB 2830
1070 IF PEEK(56321)=247 THEN 1100
1080 GET A$:IF A$<>"@" THEN 1070
1090 TN=-1
1100 POKE 899,42:POKE 883,60:POKE 838,0:FOR Z=834 TO 837:POKE Z,0:NEXT Z
1110 OX=0:OY=0:POKE 838,0:POKE 889,0:POKE 890,160:POKE 891,40:POKE 892,180
1120 POKE 823,0:POKE 824,0:POKE 821,17:POKE 822,0:SB=6:GOSUB 5230:S=0
1130 IF (S<>0) AND (S<>3) THEN 1150
1140 FOR Z=0 TO 3:SD%(25,Z)=0:NEXT Z:PS=INT(S/3):GOSUB 2750:GOSUB 2840
1150 POKE 876,0:POKE 877,205:X=((INT(RND(1)*75))*2)+71:IF TN THEN X=165
1160 POKE 52481,X:GOSUB 2810
1170 Z=-20352:SB=0:GOSUB 5240:POKE V+21,0:IF Z=128 THEN 1150
1180 IF S<3 THEN OX=OX+((X+11)-FNW(847)):OY=OY+(165-FNW(849))
1190 N=FNW(907):N=N+(N>128)*(N-128):GOSUB 2500:SD%(S,0)=FNZ(100-SD*C)
1195 SD%(25,0)=SD%(25,0)+SD%(S,0)
1200 GOSUB 2540:SD%(S,1)=FNZ(100-SD*C):SD%(25,1)=SD%(25,1)+SD%(S,1)
1210 GOSUB 2580:SD%(S,2)=FNZ(100-SD*C):SD%(25,2)=SD%(25,2)+SD%(S,2)
1220 GOSUB 2810:IF S<>2 THEN 1280
1230 OX=INT((OX/3)+1):OX=INT(OX/2)*2:OY=INT((OY/3)+0.5)
1235 X=OX-(OX<0)*65536:Y=OY-(OY<0)*65536
1240 POKE 834,FNL(X):POKE 835,FNH(X):POKE 836,FNL(Y):POKE 837,FNH(Y)
1250 POKE 785,FNL(LC):POKE 786,FNH(LC)
1260 FOR Z=0 TO 2:SD=USR(Z+128):SD%(Z,3)=FNZ(100-SD*C)
1270 SD%(25,3)=SD%(25,3)+SD%(Z,3):NEXT Z:Z=43640:GOSUB 4570:IF Z=128 THEN 1100
1280 S=S+1:IF S<6 THEN 1130
1290 POKE 53265,43:POKE 785,FNL(LC):POKE 786,FNH(LC)
1300 FOR Z=3 TO 5:SD=USR(Z):SD%(Z,3)=FNZ(100-SD*C):SD%(25,3)=SD%(25,3)+SD%(Z,3)
1310 NEXT Z:Z=43664:GOSUB 4570:IF Z=128 THEN 1100
1320 N=6:T=0:GOSUB 4920
1330 PS=0:B=0:E=2:GOSUB 2900:IF (Z=0) OR (EL<2) THEN CL=1:RETURN
1340 PS=1:B=3:E=5:GOSUB 2900:IF (Z=0) OR (EL<3) THEN CL=2:RETURN
1350 A$="{F3}09{F5}00{F7}00{CLR}{F1}{LBLU}{F2}0703The following{F2}0306targets are timed{F2}1211ASSUME A"
1360 A$=A$+"{F2}1114SUPPORTED{F2}1217POSITION{F1}":GOSUB 2790:GOSUB 5220
1370 POKE 838,0:POKE 889,0:POKE 890,160:POKE 891,40:POKE 892,180:POKE 876,0
1380 POKE 877,205:POKE 823,17:POKE 824,0:POKE 821,114:POKE 822,0:SB=6:GOSUB5230
1390 POKE 899,1:GOSUB 2840:B=0:E=2:T=1
1400 FOR S=B TO E
1410 SB=0:Z=-20352:GOSUB 5240:POKE V+21,0
1420 IF (Z>63) THEN SYS WH:FOR Z=0 TO 3:SD%(S,Z)=0:NEXT Z:GOTO 1440
1430 N=FNW(907):N=N+(N>128)*(N-128):GOSUB 2600
1440 GOSUB 2810:NEXT S:GOSUB 2900:IF (Z<>0) AND (EL>(T+2)) THEN 1460
1450 N=6:GOSUB 4920:CL=T+2:RETURN
1460 T=T+1:IF T=2 THEN PS=1:GOSUB 2750:GOSUB 2840:B=3:E=5:GOTO 1400
1470 N=6:GOSUB 4920:CL=5:RETURN
1480 NT=3:PS=CL-1:GOSUB 2750:GOSUB 5220:POKE 876,0:POKE 877,205:POKE 889,0
1490 POKE 890,160:TN=0:POKE 891,40:POKE 892,180:POKE 838,0:POKE 899,1
1500 IF (EZ=0) OR (CL<>1) THEN 1530
1510 FOR Z=0 TO NT*6-1:POKE 16325+Z,4:NEXT Z:POKE 16325+NT*6,255
1520 POKE 821,0:POKE 823,114:POKE 824,0:SB=3:GOSUB 5230:GOTO 1540
1530 POKE 823,114:POKE 824,0:POKE 821,225:POKE 822,0:SB=6:GOSUB 5230
1540 S1=32768:S2=10:IR=-32640-(CS*4096):H=0:TN=TN+1:RP=0:IF TN=7 THEN 1670
1550 GOSUB 2630
1560 S=PEEK(838):GOSUB 2840:A$="L="+STR$(CL)+"|T="+STR$(S+1)+"|R="+STR$(RP)
1570 GOSUB 5100:POKE I2+7+S*8,RP:X=((INT(RND(1)*65))*2)+71:AD=FNW(876)
1580 POKE AD+1,X:Y=PEEK(AD+2):POKE AD+6,255:TJ=PEEK(AD+11):TJ=TJ+(TJ>127)*256
1590 IF (TN=1) AND ((EZ=0) OR (CL<>1)) THEN POKE AD+15,X+48:POKE AD+20,255
1600 GOSUB 3480:SB=0:Z=IR:GOSUB 5240:POKE 878,0:IF Z=128 THEN 1670
1601 IF CS=0 THEN 1620
1602 A$="{F2}0423{BLK}{SWLC}{F1}Call your shot:":Z1=TI+359
1603 Z2=INT((Z1-TI)/60):A$=A$+"{F2}3423"+STR$(Z2):GOSUB 2830:A$="{F1}"
1604 IF Z2>0 THEN 1603
1605 POKE V+21,PEEK(V+21) OR 1
1620 IF PEEK(900)<128 THEN H=H+1
1630 GOSUB 3500:IF Z=128 THEN 1670
1640 S=S+1:IF S/3<>INT(S/3) THEN POKE 876,H%(2):POKE 877,H%(3):GOTO 1560
1650 N=3:Z=I1+H%(1)*8:GOSUB 4080:IF H>1 THEN BC=0:GOTO 1540
1660 GOSUB 2650:IR=-31616-(CS*4096):RP=RP+1:H=0:GOSUB 5080:GOTO 1560
1670 POKE V+21,0:GOSUB 3760:GOSUB 4920:CL=CL+A:RETURN
1680 RP=0:PS=CL-3:NT=4:GOSUB 2750:POKE 53265,43:GOSUB 5220:IR=-32640
1690 B=0:E=NT*6-1:GOSUB 3890
1700 AD=I1:FOR Z=1 TO NT*6:POKE AD,255:AD=AD+8:NEXT Z:S1=37445:S2=10
1710 FOR Z=B TO E:POKE 16325+Z,INT(PEEK(16325+Z)/NT):NEXT Z:FOR Z=1 TO 6
```

```
1720 DA%(Z,5)=0:DA%(Z,6)=0:NEXT Z:FOR Z=I1 TO I1+199 STEP 8:POKE Z,0:NEXT Z
1730 POKE 821,0:POKE 823,114:POKE 824,0:SB=3:GOSUB 5230:POKE 876,0:POKE 877,205
1740 I=0:GOSUB 2840
1750 I=I+1:IF I>NT*6 THEN 1790
1760 GOSUB 4370:POKE V+21,0:IF Z=128 THEN 1930
1770 IF Z=64 THEN SYS WH
1780 GOTO 1750
1790 FOR TN=1 TO 6:Z=I1+(TN-1)*32:N=4:GOSUB 4080:NEXT TN
1800 POKE 53265,43:RP=RP+1:ER=0:TN=1
1810 Z1=3:IF (CL=4)AND(TN>4) THEN Z1=2
1820 IF DA%(TN,5)>=Z1 THEN 1910
1830 ER=-1:IR=-31616:POKE 821,0:POKE 823,114:POKE 824,0:GOSUB 5080
1840 AD=I1+(TN-1)*32:FOR Z=1 TO 4:POKE AD,255:AD=AD+8:NEXT Z
1850 FOR Z=0 TO NT-1:POKE 16325+Z,TN-1:NEXT Z:POKE 16325+NT,255:SB=3:GOSUB 5230
1860 POKE 876,0:POKE 877,205:DA%(TN,5)=0:DA%(TN,6)=0:GOSUB 2840
1870 GOSUB 4370:POKE V+21,0:IF Z=128 THEN 1930
1880 IF Z=64 THEN SYS WH
1890 I=FNL(I+1):IF DA%(TN,6)<NT THEN 1870
1900 Z=I1+(TN-1)*32:N=4:GOSUB 4080
1910 TN=TN+1:IF TN<7 THEN 1810
1920 IF ER THEN 1800
1930 POKE V+21,0:FOR J=0 TO 3:SD%(25,J)=0:NEXT J
1940 N=0:FOR TN=1 TO 6:IF DA%(TN,6)=0 THEN 1970
1950 FOR I=1 TO DA%(TN,6):S=(TN-1)*NT+(I-1):N=N+1
1960 FOR J=0 TO 3:SD%(25,J)=SD%(25,J)+SD%(S,J):NEXT J,I
1970 NEXT TN:A=0:BC=0:IF N>0 THEN GOSUB 3790
1980 GOSUB 4920:CL=CL+A:RETURN
1990 PS=0:GOSUB 2750:GOSUB 5220:RP=0:IR=512
2000 S1=38042:S2=1:IF CL>7 THEN S1=37968:S2=3
2010 POKE 53265,43:B=0:E=15:GOSUB 3890:POKE 876,0:POKE 877,205:POKE 889,0
2020 POKE 890,160:POKE 891,40:POKE 892,180:POKE 838,0:POKE 899,20:POKE 910,0
2025 POKE 933,0:POKE 934,0
2030 POKE 912,0:TN=0:POKE 821,0:POKE 823,225:POKE 824,0:SB=3:GOSUB 5230
2040 GOSUB3950:A=Z:POKE I2+7,RP:HI=0:N=PEEK(838):IF (N=0) AND (A<>128) THEN2080
2050 A$="{F3}09{F6}00{CLR}{F5}00{F7}00{LBLU}{F2}0601SUMMARY: Supported Position":GOSUB 4190
2060 HI=Z1:MI=Z2:NF=Z3:AC=Z4:IF A=128 THEN BC=0:GOSUB 4920:RETURN
2070 FOR TN=1 TO 6:Z=I1:GOSUB 4080:NEXT TN
2080 IF HI<15 THEN IR=1536:RP=RP+1:GOSUB 5080:GOTO 2010
2090 POKE 881,AC:GOSUB 4920:BC=0
2100 PN=PEEK(912):PN=16:RP=0:PS=1:GOSUB 2750:POKE 899,40:POKE 53265,43:IR=512
2110 B=16:E=28:GOSUB 3890:POKE 876,0:POKE 877,205:POKE 889,176:POKE 890,164
2120 POKE 891,184:POKE 892,181:POKE 838,20:POKE 910,0:POKE 912,0
2125 POKE 933,2:POKE 934,0
2130 POKE 821,16:POKE 823,225:POKE 824,0:SB=3:GOSUB 5230:GOSUB 3950:A=Z
2140 POKE I2+167,RP:HI=0:N=PEEK(838)-20:IF (N=0) AND (A<>128) THEN 2180
2150 A$="{F3}09{F6}00{CLR}{F5}00{F7}00{LBLU}{F2}0501SUMMARY: Unsupported Position":GOSUB 4190
2160 HI=Z1:MI=Z2:NF=Z3:AC=Z4:IF A=128 THEN BC=0:GOSUB 4920:RETURN
2170 FOR TN=1 TO 6:Z=I1+160:GOSUB 4080:NEXT TN
2180 IF HI<15 THEN IR=1536:RP=RP+1:GOSUB 5080:TN=16:GOTO 2110
2190 POKE 881,24:GOSUB 4920
2200 POKE V+21,0:HI=0:MI=0:NF=0:A$="{F3}09{F6}00{CLR}{F5}00{F7}00{LBLU}{F2}1401FINAL SCORES"
2210 A$=A$+"{F2}1303{GRN}Hits{F2}2199{RED}Misses{YELO}{F2}2999No Fires{F8}{F8}":GOSUB 2830
2220 FOR I=1 TO 6:Z=DA%(I,5)+DA%(I,8):A$="{LBLU}{F2}0599"+STR$(I*50)+"{F2}1399{GRN}"+STR$(Z)
2230 HI=HI+Z:Z=DA%(I,6)+DA%(I,9):A$=A$+"{F2}2299{RED}"+STR$(Z):MI=MI+Z
2240 Z=DA%(I,7)+DA%(I,10):A$=A$+"{F2}3299{YELO}"+STR$(Z)+"{F8}{F8}":NF=NF+Z:GOSUB 2830:NEXT I
2250 A$="{F2}0699{LBLU}TOTAL{F2}1399{GRN}"+STR$(HI)+"{F2}2299{RED}"+STR$(MI)+"{F2}3299{YELO}"+STR$(NF)
2260 AC=INT(((AC+Z4)/2)+.5):Z4=AC:Z5=9:GOSUB 2680:A$=A$+"{F8}{F8}{LBLU}{F2}0699Penalties:"
2270 A$=A$+STR$(PEEK(912)+PN)
2280 IFPEEK(838)>0 THEN A$=A$+"{F8}{F2}0699Your accuracy was "+RIGHT$(CR$,LEN(CR$)-5)
2290 A$=A$+"{WHT}":GOSUB 2790
2300 IF Z>2 THEN BC=0:CL=CL+1:RETURN
2310 IF Z=1 THEN A=(CL>SL):BC=6-A:CL=CL+A:B$="{F2}1499a poor":GOTO 2320
2312 BC=6:B$="{F2}0599a below average"
2320 A$="{F3}09{CLR}{RED}{F1}{F8}{F8} You are being sent{F8}{C/RT}  back to level"+STR$(CL)
2322 A$=A$+"{F8}  because you have{F8}"+B$+"{F8}{F2}0699accuracy score{F1}{WHT}":GOSUB 2790:RETURN
2330 PS=0:GOSUB 2750:POKE 53280,0:RP=0:S1=37968:S2=3:GOSUB 5220:IR=640
2340 POKE 876,0:POKE 877,205:POKE 889,0:POKE 890,160
2350 TN=0:POKE 891,40:POKE 892,180:POKE 838,0:POKE 899,40:POKE 912,0
2360 POKE 823,145:POKE 824,3:POKE 821,59:POKE 822,6:SB=6:GOSUB 5230
2370 GOSUB 2840:A$="L=10|T=  0|R="+STR$(RP):GOSUB 5100:POKE 933,0:POKE 934,0
2380 POKE I2+7,RP:SB=0:Z=IR:GOSUB 5240:A=Z
2390 POKE V+21,0:POKE 53280,0:N=PEEK(838):IF N=0 THEN GOSUB 4920:RETURN
2400 B$="Supported Position":IF PS THEN B$="Unsupported Position"
2410 A$="{F3}09{F6}00{CLR}{F5}00{F7}00{LBLU}{F2}0601SUMMARY: "+B$:N=PEEK(838):GOSUB 4710
2420 IF A=128 THEN BC=0:GOSUB 4920:RETURN
2430 FOR TN=1 TO 6:Z=I1:GOSUB 4080:NEXT TN
2440 IF HI<(23+ABS(PS-1)*5) THEN IR=1664:RP=RP+1:GOSUB 5080:GOTO 2340
2450 BC=0:GOSUB 4920:RP=0:IF PS=1 THEN CL=11:RETURN
2460 PS=1:GOSUB 2750:IR=640:GOTO 2340
2500 POKE 785,FNL(DV):POKE 786,FNH(DV):AD=43008+(S AND 1)*128
2510 N1=-60*(CL<3)-45*(CL>=3):IF (CL=0) AND (T>0) THEN N1=45
2520 IF N<(N1+2) THEN SD=999:RETURN
2530 N1=(N-N1)*256+(N-19):GOTO 2560
2540 IF N<18 THEN SD=999:RETURN
2550 N1=(N-18)*256+(N-1)
2560 POKE 253,FNL(AD):POKE 254,FNH(AD):SD=USR(N1)
2570 POKE 253,FNL(AD+256):POKE 254,FNH(AD+256):SD=SD+USR(N1):RETURN
2580 POKE 253,0:POKE 254,170:SD=USR(262):POKE 253,30:SD=SD+USR(262):RETURN
2600 GOSUB 2500:SD%(S,0)=FNZ(100-SD*C):GOSUB 2540:SD%(S,1)=FNZ(100-SD*C)
2610 GOSUB 2580:SD%(S,2)=FNZ(100-SD*C):POKE 785,FNL(LC):POKE 786,FNH(LC)
2620 SD=USR(S):SD%(S,3)=FNZ(100-SD*C):RETURN
2630 H%(1)=PEEK(838):H%(2)=PEEK(876):H%(3)=PEEK(877):H%(4)=PEEK(889)
2640 H%(5)=PEEK(890):H%(6)=PEEK(891):H%(7)=PEEK(892):H%(8)=PEEK(899):RETURN
2650 POKE 838,H%(1):POKE 876,H%(2):POKE 877,H%(3):POKE 889,H%(4)
2660 POKE 890,H%(5):POKE 891,H%(6):POKE 892,H%(7):POKE 899,H%(8):RETURN
2670 Z4=SD%(S,J):Z5=J+PS*4
2680 CR$="{F2}2299{GRN}excellent":Z=5:IF Z4<CR%(Z5,0) THEN CR$="{F2}3299{BLUE}good":Z=4
2690 IF Z4<CR%(Z5,1) THEN CR$="{F2}2699{BRN}average":Z=3
2700 IF Z4<CR%(Z5,2) THEN CR$="{F2}2299{YELO}below avg":Z=2
2710 IF Z4<CR%(Z5,3) THEN CR$="{F2}3299{RED}poor":Z=1
2720 IF (CL<>0) OR (NU=0) THEN RETURN
2730 B$=STR$(Z4):Z4=LEN(B$):IF Z4<4 THEN FOR Z5=1 TO 4-Z4:B$=" "+B$:NEXT Z5
2740 CR$="{F1}{F2}2499"+MID$(CR$,6,10)+"{F2}3599"+B$+"{F1}{F8}":RETURN
2750 POKE V+21,0:A$="{F3}09{CLR}{F6}00{F5}00{WHT}{F1}"
2760 IF PS THEN A$=A$+"{F2}1108ASSUME AN{F2}0911UN":GOTO 2780
2770 A$=A$+"{F2}1208ASSUME A{F2}1111"
2780 A$=A$+"SUPPORTED{F2}1214POSITION{F1}"
2790 A$=A$+"{F2}0723<Pull trigger to continue>{F3}00":GOSUB 2830
```

```
2800 IF PEEK(56321)<>247 THEN 2800
2810 IF PEEK(56321)<>255 THEN 2810
2820 RETURN
2830 A$=A$:SYS 3424:RETURN
2840 POKE 253,FNL(S1):POKE 254,FNH(S1):POKE 251,S2:SYS SC:RETURN
2850 REM GET AVERAGE ACROSS B-E SHOTS FOR EACH DIAGNOSTIC SCORE
2860 FOR Z2=0 TO 3:SD%(12,Z2)=0:NEXT Z2
2870 FOR Z1=B TO E:FOR Z2=0 TO 3:SD%(12,Z2)=SD%(12,Z2)+SD%(Z1,Z2):NEXT Z2,Z1
2880 Z=(E-B)+1:FOR Z2=0 TO 3:SD%(12,Z2)=INT((SD%(12,Z2)/Z)+.5):NEXT Z2:RETURN
2890 REM ROUTINE TO SEE IF SHOTS B-E FALL INTO RANGE FOR ADVANCEMENT
2900 GOSUB 2860:S=12:J=3:GOSUB 2670:IF Z<>5 THEN Z=0:RETURN
2910 J=0:GOSUB 2670:Z1=Z:J=1:GOSUB 2670
2920 IF ((Z<>5) OR (Z1<4)) AND ((Z1<>5) OR (Z<4)) THEN Z=0:RETURN
2930 J=2:GOSUB 2670:IF Z<3 THEN Z=0:RETURN
2940 Z=1:RETURN
2950 A$="{F3}09{F6}00{CLR}{F5}00{F7}00{F1}{LBLU}{F2}1505LEVEL{F2}1709":B$=STR$(CL):Z=CL+(CL=10)*10
2960 IF CL=10 THEN B$="{C/RT}10"
2970 A$=A$+B$+"{GRN}{F2}0015":GOSUB 2830:POKE 823,CL:SB=18:GOSUB 5230
2980 A$="{RED}{F2}0723<Pull trigger to continue>{F3}00":GOSUB 2830
2990 IF PEEK(56321)=247 THEN RETURN
3000 IF PEEK(56321)<>127 THEN 2990
3010 GOSUB 3030:IF CL=0 THEN RETURN
3020 GOTO 2950
3030 POKE 53265,43:POKE 53272,31:POKE 198,0:A$="{F6}00{F5}00{F7}00{CLR}{GRN}Start level:"
3040 A$=A$+STR$(SL)+"{GRY1}{F2}3399ARI8806{GRN}Final level:"+STR$(EL)+"{F8}{BLUE}Wind speed: "
3050 IF (WI AND 16) THEN A$=A$+" Variable";:GOTO 3080
3060 A$=A$+STR$((WI AND 7)*5)+" MPH ":IF ((WI AND 7)=0) THEN 3080
3070 B$=STR$(INT(WI/32)+1):B$="0"+RIGHT$(B$,1):A$=A$+"{F4}"+B$
3080 A$=A$+"{BLK}{F8}{F8}":GOSUB 2830:POKE 253,SL:POKE 254,EL:POKE 823,0:SB=18:GOSUB 5230
3090 A$="{F8}{LBLU} L: Select Start/Stop Level{F8} N: New firer{F8} W: Set Wind Speed{F8}"
3100 B$="skill test":IF CL<>0 THEN B$="level"+STR$(CL)
3110 A$=A$+" G: Go to "+B$+"{F8}{F8}"
3120 IF EZ THEN A$=A$+"{GRN}(Level 1 targets are at 250 meters){F8}"
3130 IF NU THEN A$=A$+"{GRN}(Pretest diagnostic scores are numeric){F8}"
3135 IF CS THEN A$=A$+"{GRN}(Call your shot on Levels 1-2)"
3140 A$=A$+"{F2}0023{RED}Type a number or letter & press RETURN{LBLU}{F8}{F3}00":GOSUB 2830
3150 A$="":GOSUB 5140:Z=INT(VAL(A$))
3160 IF (Z>=SL) AND (Z<=EL) THEN CL=Z:MO$="m":BC=0:RETURN
3170 IF A$="nu" THEN NU=NOT(NU):GOTO 3030
3180 IF A$="ez" THEN EZ=NOT(EZ):GOTO 3030
3185 IF A$="cs" THEN CS=NOT(CS):GOTO 3030
3190 IF A$="n" THEN CL=0:RETURN
3200 IF A$="g" THEN RETURN
3210 IF A$<>"l" THEN 3300
3220 MO$="m":A$="{F2}1423{RED}& press RETURN          {F2}0024{BLUE}Start level (1-10)"
3230 GOSUB 5140:Z$=A$:SL=INT(VAL(A$)):IF (SL<1) OR (SL>10) THEN 3220
3240 IF (CL<SL) AND (CL<>0) THEN CL=SL
3250 IF SL=10 THEN EL=10:GOTO 3030
3260 A$="{F2}0024{LBLU}Final level ("+Z$+"-10)":GOSUB 5140:EL=INT(VAL(A$))
3270 IF (EL<SL) OR (EL>10) THEN 3260
3280 IF CL>EL THEN CL=EL
3290 GOTO 3030
3300 IF A$<>"w" THEN 3420
3310 A$="{F3}09{CLR}Wind Effects:{F8}{F8}Type a number (1-3) for wind speed:{F8}{F8}"
3320 A$=A$+" 1:  0 MPH{F8}{F8} 2: 10 MPH{F8}{F8} 3: 20 MPH{F8}{F8}{F3}00":GOSUB 2830
3330 A$="":GOSUB 5140:WI=VAL(A$):IF (WI<1) OR (WI>3) THEN 3330
3340 WI=(WI-1)*2:IF WI=6 THEN WI=16:GOTO 3030
3350 IF WI=0 THEN 3030
3360 A$="{F3}09{CLR}Wind Effects:{F8}{F8}Type a number (1-8) for wind direction:{F8}{F8}"
3370 A$=A$+" 1: {F4}01 (No value){F8}{F8} 2: {F4}02 (Half value){F8}{F8} 3: {F4}03 (Full value){F8}{F8}"
3380 A$=A$+" 4: {F4}04 (Half value){F8}{F8} 5: {F4}05 (No value){F8}{F8} 6: {F4}06 (Half value){F8}{F8}"
3390 A$=A$+" 7: {F4}07 (Full value){F8}{F8} 8: {F4}08 (Half value){F8}{F8}{F3}00":GOSUB 2830
3400 A$="":GOSUB 5140:IF (VAL(A$)<1) OR (VAL(A$)>8) THEN 3400
3410 WI=WI+((VAL(A$)-1)*32):GOTO 3030
3420 IF A$="ca" THEN Z=1:GOTO 120
3430 IF A$="pf" THEN Z=2:GOTO 120
3440 IF A$="rs" THEN Z=3:GOTO 120
3450 IF A$="sp" THEN Z=4:GOTO 120
3460 GOTO 3150
3470 REM SETUP FOR REPLAY: CALL ONCE BEFORE ANY TARGETS PRESENTED
3480 POKE 2041,43:POKE 2042,43:POKE V+23,6:POKE V+29,6
3490 POKE V+40,1:POKE V+41,0:RETURN
3500 N=FNW(907):N=N+(N>128)*(N-128):SYS 3448:GOSUB 2600
3510 J=0:GOSUB 2670:A$="{F3}05{CLR}{F5}00{F1}{WHT}steady pos "+CR$
3520 J=1:GOSUB 2670:A$=A$+"{F8}{WHT}trigger sq "+CR$:J=2:GOSUB 2670
3530 A$=A$+"{F2}0019{WHT}follow thr "+CR$:J=3:GOSUB 2670
3540 A$=A$+"{F8}{WHT}shot loc   "+CR$+"{HOME}":GOSUB 2810
3550 Z=TN:W=FNW(930):W=W+(W>32767)*65536:IF (CL=1) AND (EZ) THEN Z=5
3560 POKE 839,X+DA%(Z,2):POKE 840,0:POKE 841,Y+DA%(Z,3)
3570 POKE V+4,231-W:POKE V+5,127-TJ
3580 X1=ABS(244+(FNW(847)-(X+DA%(Z,2)))):Y1=ABS(143+(PEEK(849)-(Y+DA%(Z,3))))
3590 X2=PEEK(V+16) AND 254:IF FNH(X1)<>0 THEN X2=X2 OR 1
3600 GOSUB 2830:POKE V,FNL(X1):POKE V+1,FNL(Y1):POKE V+16,X2
3610 WS=128:POKE V+14,254-DA%(Z,2):POKE V+15,152-DA%(Z,3)
3620 IF (TN=1) AND ((EZ=0) OR (CL<>1)) THEN WS=192:POKE V+12,255:POKE V+13,120
3630 POKE 823,7:POKE 824,11:POKE 253,221:SYS CO:POKE 845,1:F=-1:SYS SV
3640 POKE V+21,WS OR 6:A$="{BLK}{F2}0212{F1}{RVON}replay{RVOF}{F3}00":GOSUB 2830:N=FNW(905)
3645 N=N+(N>255)*(N-255):POKE 821,N:N=N-60:POKE 833,-N*(N>0):POKE 884,1
3650 IF (PEEK(884) AND 1)=0 THEN 3680
3660 IF (PEEK(56321)=247) AND (F=0) THEN POKE 884,0:Z=0:GOTO 3750
3670 GOTO 3650
3680 SYS BANG:F=0:POKE V+21,WS OR 7
3690 A$="{BLK}{F2}0212pull trigger{F8}{C/RT}{C/RT}to continue ":GOSUB 2830:POKE 198,0:F=1:Z=1
3700 IF PEEK(56321)=247 THEN Z=0:GOTO 3750
3710 IF PEEK(56321)=127 THEN Z=128:GOTO 3750
3720 Z=Z+1:IF Z<100 THEN 3700
3730 F=F+1:IF F=2 THEN POKE V+21,WS OR 1:Z=0:GOTO 3700
3740 F=0:GOTO 3640
3750 GOSUB 2810:POKE V+21,0:RETURN
3760 A=0:BC=0:N=PEEK(838):IF N<2 THEN RETURN
3770 FOR J=0 TO 3:SD%(25,J)=0:NEXT J
3780 FOR S=0 TO N-1:FOR J=0 TO 3:SD%(25,J)=SD%(25,J)+SD%(S,J):NEXT J,S
3790 FOR J=1 TO 5:H%(J)=0:NEXT J
3800 FOR J=0 TO 3:SD%(25,J)=INT((SD%(25,J)/N)+.5):NEXT J:S=25:J=0:GOSUB 2670
3810 A$="{F3}05{F6}00{F5}00{F7}00{CLR}{LBLU}{F2}0601{F1}final: level"+STR$(CL)+"{F8}{F8}{WHT}steady pos "+CR$
3820 H%(Z)=H%(Z)+1:J=1:GOSUB 2670:A$=A$+"{F8}{WHT}trigger sq "+CR$:H%(Z)=H%(Z)+1
3830 J=2:GOSUB 2670:A$=A$+"{F8}{WHT}follow thr "+CR$:H%(Z)=H%(Z)+1
3840 J=3:GOSUB 2670:A$=A$+"{F8}{WHT}shot loc   "+CR$+"{F3}08{F1}{WHT}":H%(Z)=H%(Z)+1:POKE V+21,0
```

```
3850 GOSUB 2790:A=0:BC=0:IF N<>NT*6 THEN RETURN
3860 IF H%(1)<2 THEN 3870
3865 A=(CL>SL):BC=6-A:B$="{F2}11992 or more{F8}{F2}0999poor scores":GOTO 3881
3870 IF (H%(1)=0) AND (H%(2)=0) THEN A=1:RETURN
3880 BC=6:B$="{F2}1199a poor or{F8}{C/RT}below average score"
3881 A$="{F3}09{CLR}{RED}{F1}{F8}{F8} You are being sent{F8}{C/RT}  back to level"+STR$(CL+A)
3882 A$=A$+"{F8}  because you have{F8}"+B$+"{F1}{WHT}":GOSUB 2790:RETURN
3890 FOR Z=B TO Z+1:POKE 16325+Z,255:NEXT Z
3900 FOR Z=B TOE:Z1=INT(RND(1)*(E-B))+B
3910 IF PEEK(16325+Z1)=255 THEN POKE 16325+Z1,Z:GOTO 3940
3920 Z1=Z1+1:IF Z1>E THEN Z1=B
3930 GOTO 3910
3940 NEXT Z:RETURN
3950 GOSUB 2840:A$="L="+STR$(CL)+"|T=  0|R="+STR$(RP):GOSUB 5100
3960 TN=TN+1:IF TN=E+2 THEN Z=0:RETURN
3970 IF CL<8 THEN 4000
3980 AD=FNW(876):T=PEEK(AD):AD=AD+1
3990 FOR Z=1 TO T:Z1=INT(((PEEK(AD+5)*2)/3)+.5):POKE AD+5,Z1:AD=AD+14:NEXT Z
4000 AD=16512:IF (CL=6) OR (CL=8) THEN AD=18560
4010 IF (CL=7) OR (CL=9) THEN AD=20608
4030 Z=AD:IF PEEK(838)>=PEEK(899) THEN Z=Z+1
4040 SB=0:Z=Z+IR:POKE 53280,BC:GOSUB 5240:POKE V+21,0:IF Z=128 THEN RETURN
4050 IF Z=64 THEN SYS WH
4060 IF PEEK(878)<>0 THEN 4060
4070 Z1=(INT(RND(1)*3)+1)*750:FOR Z=1 TO Z1:NEXT Z:GOTO 3960
4080 A$="{F3}09{F6}00{CLR}{F5}13{F7}00{BLK}":GOSUB 2830
4090 POKE 251,FNL(Z):POKE 252,FNH(Z):Z=TN:IF (CL=1) AND (EZ) THEN Z=5
4100 POKE 880,Z:POKE 881,N:SYS GP:IF PEEK(824)=0 THEN RETURN
4110 A$="{F2}0402Here is your"+STR$(PEEK(824))+" round shot group"
4120 IF PEEK(823)>0 THEN A$=A$+"{F2}1003"+STR$(PEEK(823))+" shot(s) missing"
4130 WS=128:A$=A$+"{F2}1404"+STR$(Z*50)+" meters":POKE V+14,172-DA%(Z,2)
4140 POKE V+15,150-DA%(Z,3):POKE 2047,Z+32:POKE V+45,12:POKE V+46,12:Z1=0
4150 IF Z<3 THEN POKE2047,37+Z*2:Z1=192
4160 IF Z=1 THEN POKE V+12,173:POKE V+13,118:POKE 2046,40:WS=192
4170 POKE V+23,Z1:POKE V+29,Z1:POKE V+27,255:POKE V+16,0:POKE V+21,WS
4180 GOSUB 2790:POKE 53265,43:POKE V+27,0:POKE V+21,0:RETURN
4190 IF N=0 THEN RETURN
4200 Z1=0:Z2=PS*20:Z3=PEEK(838)-1:Z5=0:POKE 785,FNL(LC):POKE 786,FNH(LC)
4210 FOR Z=Z2 TO Z3:Z4=USR(Z):IF Z4<40 THEN Z5=Z4+Z5:Z1=Z1+1
4220 NEXT Z:Z4=0:IF Z1>0 THEN Z4=FNZ(100-(Z5/Z1)*C)
4230 A$=A$+"{F2}1303{GRN}Hits{F2}2199{RED}Misses{YELO}{F2}2999No Fires{F8}{F8}":GOSUB 2830:Z1=0:Z2=0:Z3=0
4240 FOR I=1 TO 6:AD=I1+PS*160:POKE 251,FNL(AD):POKE 252,FNH(AD)
4250 POKE 881,N:POKE 880,128+I
4260 SYS GP:A$="{LBLU}{F2}0599"+STR$(I*50)+" m":Z=PS*3:DA%(I,5+Z)=PEEK(910)
4270 DA%(I,6+Z)=PEEK(911):DA%(I,7+Z)=DA%(I,PS)-PEEK(824)
4280 IF DA%(I,7+Z)<0 THEN DA%(I,7+Z)=0
4290 Z1=Z1+PEEK(910):Z2=Z2+PEEK(911):Z3=Z3+DA%(I,7+Z)
4300 A$=A$+"{GRN}{F2}1399"+STR$(PEEK(910))+"{RED}{F2}2299"+STR$(PEEK(911))+"{YELO}{F2}3299"
4310 A$=A$+STR$(DA%(I,7+Z))+"{F8}{F8}":GOSUB 2830:NEXT I
4320 A$="{F2}0699{LBLU}TOTAL{F2}1399{GRN}"+STR$(Z1)+"{F2}2299{RED}"+STR$(Z2)+"{F2}3299{YELO}"+STR$(Z3)+"{F8}{F8}"
4330 A$=A$+"{LBLU}{F2}0699Penalties:"+STR$(PEEK(912))+"{F8}":IF PEEK(838)<=PS*20 THEN 4350
4340 Z5=9:GOSUB2680:A$=A$+"{LBLU}{F2}0699Your accuracy was "+RIGHT$(CR$,LEN(CR$)-5)+"{F8}"
4350 B$="{F2}1599{GRN}QUALIFIED{WHT}":IF Z1<15 THEN B$="{F2}1499{RED}UNQUALIFIED{WHT}"
4360 A$=A$+B$:GOSUB 2790:RETURN
4370 AD=FNW(876):TN=PEEK(AD+3):Z1=DA%(TN,6)
4380 X=((INT(RND(1)*65))*2)+71:POKE AD+1,X:Y=PEEK(AD+2):TJ=PEEK(AD+11)
4390 TJ=TJ+(TJ>127)*256:IF TN=1 THEN POKE AD+15,X+48
4400 S=((TN-1)*NT)+Z1:POKE 838,S:AD=I1+S*8:POKE AD+7,I
4420 POKE 900,255:POKE AD,255:GOSUB3480
4430 A$="L="+STR$(CL)+"|T="+STR$(I)+"|R="+STR$(RP):GOSUB 5100:POKE I2+7+S*8,RP
4440 SB=0:Z=IR:GOSUB 5240:POKE 878,0
4450 IF Z<>128 THEN DA%(TN,6)=DA%(TN,6)+1
4460 IF Z>63 THEN RETURN
4470 FOR Z=1 TO 5:H%(Z)=0:NEXT Z:N=FNW(907):N=N+(N>128)*(N-128):GOSUB 2600
4480 J=0:GOSUB 2670:A$="{F3}05{CLR}{F5}00{F1}{WHT}steady pos "+CR$:H%(Z)=H%(Z)+1
4490 J=1:GOSUB 2670:A$=A$+"{F8}{WHT}trigger sq "+CR$:H%(Z)=H%(Z)+1
4500 J=2:GOSUB 2670:A$=A$+"{F2}0019{WHT}follow thr "+CR$:H%(Z)=H%(Z)+1
4510 J=3:GOSUB 2670:A$=A$+"{F8}{WHT}shot loc   "+CR$+"{HOME}":H%(Z)=H%(Z)+1
4520 GOSUB 2810:IF (H%(1)=0) AND (Z>2) THEN 4550
4530 GOSUB 3550:IF Z=128 THEN RETURN
4540 IF I<>NT*6 THEN GOSUB 2840
4550 IF PEEK(900)<128 THEN DA%(TN,5)=DA%(TN,5)+1
4560 RETURN
4570 FOR J=0 TO 3:SD%(25,J)=INT((SD%(25,J)/3)+.5):NEXT J
4580 A=S:B=Z:S=25:J=0:GOSUB 2670:A$="{F3}05{F6}00{F5}00{F7}00{CLR}{WHT}{F1}steady pos "+CR$
4590 J=1:GOSUB 2670:A$=A$+"{WHT}trigger sq "+CR$:J=2:GOSUB 2670
4600 A$=A$+"{WHT}follow thr "+CR$:J=3:GOSUB 2670:A$=A$+"{WHT}shot loc   "+CR$:S=A:Z=B
4610 GOSUB 2830:POKE 251,FNL(Z):POKE 252,FNH(Z):POKE 880,1:POKE 881,3:SYS GP
4620 A$="":IFPEEK(823)>0 THEN A$=A$+"{BLK}{F2}1008"+STR$(PEEK(823))+" shot(s) missing"
4630 POKE 823,0:POKE 824,15:POKE 253,13:SYS CO
4640 POKE V+14,161:POKE V+15,135:POKE V+16,0:POKE V+21,128:POKE V+27,128
4650 A$=A$+"{SWLC}{WHT}{F2}0723<Pull trigger to continue>{F3}00":GOSUB 2830:B$="  "
4660 IF PEEK(56321)=247 THEN Z=0:GOTO 4700
4670 GET A$:IF A$="" THEN 4660
4680 B$=RIGHT$(B$,1)+A$:IF B$="rz" THEN Z=128:GOTO 4700
4690 GOTO 4660
4700 POKE 53265,43:POKE V+27,0:POKE V+21,0:RETURN
4710 A$=A$+"{F2}1303{GRN}Hits{F2}2199{RED}Misses{YELO}{F2}2999No Fires{F8}{F8}":GOSUB 2830:HI=0:MI=0:NF=0
4720 FOR I=1 TO 6:POKE 251,FNL(I1):POKE 252,FNH(I1):POKE 881,PEEK(838)
4730 POKE 880,128+I:SYS GP:A$="{LBLU}{F2}0599"+STR$(I*50)+" m"
4740 HI=HI+PEEK(910):MI=MI+PEEK(911):Z3=DA%(I,4)-PEEK(824):IF Z3<0 THEN Z3=0
4750 NF=NF+Z3:A$=A$+"{GRN}{F2}1399"+STR$(PEEK(910))+"{RED}{F2}2299"+STR$(PEEK(911))
4760 A$=A$+"{YELO}{F2}3299"+STR$(Z3)+"{F8}{F8}":GOSUB 2830:NEXT I
4770 A$="{F2}0699{LBLU}TOTAL{F2}1399{GRN}"+STR$(HI)+"{F2}2299{RED}"+STR$(MI)+"{F2}3299{YELO}"+STR$(NF)+"{F8}{F8}"
4780 REM A$=A$+"{F2}0799Penalties:"+STR$(PEEK(912)):GOSUB 2770:RETURN
4790 B$="{F2}1599{GRN}QUALIFIED{WHT}":IF HI<(23+ABS(PS-1)*5) THEN B$="{F2}1499{RED}UNQUALIFIED{WHT}"
4800 A$=A$+B$:GOSUB 2790:RETURN
4810 PRINT#1,"p"CHR$(104)CHR$(FNL(Z))CHR$(FNH(Z))CHR$(0):RETURN
4820 INPUT#1,ER,A$,A$,A$:IF ER<21 THEN ER=0
4830 RETURN
4840 Z=1:OPEN 8,8,8,"data":GOSUB 4820:IF ER=0 THEN 4860
4850 CLOSE 8:OPEN 8,8,8,"data,l,"+CHR$(170):RE=1:HS=0:GOTO 4880
4860 GOSUB 4810:GET#8,A$,B$:RE=ASC(A$+CHR$(0))+ASC(B$+CHR$(0))*256
4870 GET#8,A$,B$:HS=ASC(A$+CHR$(0))+ASC(B$+CHR$(0))*256
4880 RE=RE+1:HS=HS+1:GOSUB 4810:A$=CHR$(FNL(RE))+CHR$(FNH(RE))
4890 A$=A$+CHR$(FNL(HS))+CHR$(FNH(HS)):PRINT#8,A$:Z=RE:GOSUB 4810
4900 PRINT#8,IO$;ID$;CHR$(13);CHR$(FNL(RE));CHR$(FNH(RE));:CLOSE 8
4910 RC=RE:GOSUB 4820:RETURN
```

```
4920 IF (RK=0) OR ((N=0) AND (RP=0)) THEN RETURN
4930 OPEN 1,8,15:POKE V+21,0:IF RE<300 THEN 4970
4940 A$="{F3}09{F6}00{F5}00{F7}00{CLR}{F1}{F8} This disk is full{F4}{F8}{F8} Please insert{F8}{F8} another and{F8}{F8}"
4950 A$=A$+" press RETURN{F3}00":GOSUB2830
4960 GET A$:IF A$<>CHR$(13) THEN 4960
4970 A$="{F3}09{CLR}{F5}00{F7}00{RED}{SWLC}{F1}{F2}0112--- Please Wait ---{F3}00":GOSUB 2830
4980 IF (RE>299) OR ((CL=0) AND (T=0)) THEN GOSUB 4840:IF RE>299 THEN 4940
4990 OPEN 8,8,8,"data":Z=RE+1:GOSUB 4810:POKE 823,FNL(Z):POKE 824,FNH(Z)
5000 Z=-(T<>0)*128:IF CL>0 THEN Z=CL:IF (CL>4) AND (PS=1) THEN Z=Z+128
5010 POKE 880,Z:POKE 881,AC:POKE 831,ASC(MO$):Z=8192
5020 IF CL>4 THEN 5040
5030 FOR Z1=0 TO 23:FOR Z2=0 TO 3:POKE Z,SD%(Z1,Z2):Z=Z+1:NEXT Z2,Z1
5040 SYS SA:RE=FNW(823):Z=1:GOSUB 4810:A$=CHR$(FNL(RE))+CHR$(FNH(RE))
5050 A$=A$+CHR$(FNL(HS))+CHR$(FNH(HS)):PRINT#8,A$:Z=RC:GOSUB 4810
5060 PRINT#8,IO$;ID$;CHR$(13);CHR$(FNL(RE));CHR$(FNH(RE)):CLOSE 8
5070 GOSUB 4820:CLOSE 1:MO$="a":RETURN
5080 A$="{F3}09{F6}00{F5}00{F7}00{CLR}{RED}{SWLC}{F1}{F2}1110Get Ready{F3}00":GOSUB 2830
5090 FOR Z=1 TO 3000:NEXT Z:BC=2:RETURN
5100 B$="":IF ((WI AND 7)*5)=0 THEN5130
5110 B$=STR$(((WI AND 224)/32)+1)
5120 B$="{F2}3400W-"+STR$((WI AND 7)*5)+"{F4}0"+RIGHT$(B$,1)
5130 A$="{HOME}{BLK}{RVON}{SWLC}"+A$+B$+"{F6}0"+CHR$(BC+48):GOSUB 2830:RETURN
5140 A$=A$+"?    {C/LF}{C/LF}{C/LF}{C/LF}{C/LF}":GOSUB2830:B$=""
5150 GET A$:IF A$="" THEN 5150
5160 IF (A$=CHR$(13)) OR (LEN(B$)>2) THEN A$="{C/UP}{F8}":GOSUB 2830:A$=B$:RETURN
5170 IF A$<>CHR$(20) THEN 5200
5180 IF B$<>"" THEN B$=LEFT$(B$,LEN(B$)-1):A$=" {C/LF}{C/LF} {C/LF}/{C/LF}":GOSUB 2830
5190 GOTO 5150
5200 IF (A$<"0") OR (A$>"z") THEN 5150
5210 B$=B$+A$:A$=A$+" {C/LF}":GOSUB 2830:GOTO 5150
5220 FOR Z=I2 TO I2+343 STEP 8:POKE Z,255:NEXT Z:RETURN
5230 POKE 49168,SB:SYS 49162:RETURN
5240 POKE 785,10:POKE 786,192:POKE 49168,SB:Z=USR(Z):RETURN
6000 REM THE FOLLOWING IS THE CRITERION FOR GOOD,AVERAGE,BELOW AVERAGE,POOR
6010 DATA 95,90,84,79:REM SP SUP
6020 DATA 96,92,87,83:REM TS SUP
6030 DATA 95,89,82,76:REM FT SUP
6040 DATA 97,93,87,82:REM SL SUP
6050 DATA 92,87,81,76:REM SP UNS
6060 DATA 93,86,78,71:REM TS UNS
6070 DATA 93,79,64,50:REM FT UNS
6080 DATA 96,91,84,78:REM SL UNS
6090 DATA 39,35,30,26:REM HITS
6100 DATA 97,95,92,90:REM ACCURACY
6110 DATA 3,2,47,32,3,4,5,23,36,9,4,6,11,12,9
6120 DATA 4,4,11,14,6,3,2,11,15,7,2,1,11,16,6
```

```
        *=3400
;MLTEXT.3400 FOR BRM CARTRIDGE
        JMP   ENABLE
        JMP   DSABLE
        JMP   ROTATE
        JMP   SEEVAL
        JMP   NOTTRG
        JMP   BANG
        JMP   WAIT
        JMP   SCENE
        JMP   LETERS
        JMP   LETML
        JMP   STDEV
        JMP   RADERR
        JMP   SHTGRP
        JMP   COLORS
        JMP   WHISTL
        JMP   SAVEIT
        JMP   PORT
        .OPT  NOL
PRGNUM =49161
;
;              CONVENTIONS USED
;PROCEDURE XXXXX
;PURPOSE OF PROCEDURE
;NECESSARY ACTIONS BEFORE CALL
;HOW TO CALL
;WHAT USER CAN EXPECT AFTER CALL
;
        .LIB  MACROS-VARS
        .OPT  NOL
;820-1023 IS UNUSED IF NO CASSETTE
CHOICE =820
STOP   =821
HOLD1  =823
HOLD2  =824
NSRT   =825
HOLDA  =826
HOLDX  =827
NUMRD  =828
NUMR20 =829
HOLD   =831
DELAY  =832
CUR1   =833
OFFX   =834
OFFY   =836
SHOTS  =838
H1IRQ  =839
H2IRQ  =840
H3IRQ  =841
H4IRQ  =842
H5IRQ  =843
H6IRQ  =844
H7IRQ  =845
H8IRQ  =846
XVAL   =847  ;X BULLET STRIKE
YVAL   =849  ;Y BULLET STRIKE
SORTAF =851  ;# RDGS TO SORT AFTER
STSIZE =852  ;# TO PUT IN SRTBUF
SRTBUF =853
CURSRT =863
HOLDY  =864
LENSTR =865
TRAJ   =866
IRQ    =867
SSCOL  =868
SDCOL  =869
FLAGS  =870
CODE1  =871
CODE2  =872
COLOR  =873
OFFSCR =874
HAFSEC =875
TARORD =876
CRSDLA =878
CURTF  =879
HOLD3  =880
HOLD4  =881
CURTAR =882
CUR2   =883
IRQ2   =884
COPY   =885
CODE3  =886
CODE4  =887
TARPRE =888
NUMR60 =889
;2 BYTES FREE AT 891
HITSPR =893
SUMX   =894
NSIZE  =896
TIMES  =897
FSTAT  =898
MAXSHT =899
LASTSH =900
HITS   =910
MISSES =911
PENAL  =912
PINFO  =913
MAXVAL =928
WIND   =929
WDRIFT =930
SHOTRR =932
CURNUM =933
IRQ3   =935
BORCOL =936
LPCMPX =937
LPCMPY =938
;NEXT ONE AT 939

CENX   =15872
CENY   =15880
TARNUM =15888
SPECIL =15896
YDRIFT =15904
XDRIFT =15912
JIFFYS =15920
TIMEFL =15928
INSTK  =15936
CDLSTK =15944
DLSTK  =15952
;
ADD    =$B86A
CHKIN  =$FFC6
CHKOUT =$FFC9
CHRIN  =$FFCF
CLOSE  =$FFC3
CLRCHN =$FFCC
CHROUT =$FFD2
DIVIDE =$BB12
FLOAT  =$B391
GETIN  =$FFE4
LOAD   =$FFD5
MLTPLY =$BA2B
OPEN   =$FFC0
SETLFS =$FFBA
SETNAM =$FFBD
SQRT   =$BF71
SUBTRT =$B853
UNFLOT =$B1BF
V      =$D000
SID    =$D400
TEMP   =16192
XLPB20 =40960 ;UNDERNEATH ROM
YLPB20 =41472 ;UNDERNEATH ROM
XTGB20 =41984 ;UNDERNEATH ROM
YTGB20 =42496 ;UNDERNEATH ROM
XLPB60 =43008 ;UNDERNEATH ROM
YLPB60 =43264 ;UNDERNEATH ROM
XLPA60 =43520 ;UNDERNEATH ROM
YLPA60 =43550 ;UNDERNEATH ROM
XTGA60 =43580 ;UNDERNEATH ROM
YTGA60 =43610 ;UNDERNEATH ROM
INFO   =43640 ;UNDERNEATH ROM
INFO2  =43984 ;UNDERNEATH ROM
HLDBUF =44328 ;UNDERNEATH ROM
;NEXT AT 44840
        .MAC  DINC    ;DOUBLE
        INC   ?1      ;PRECISION
        BNE   ?2      ;INCREMENT
        INC   ?1+1
?2      .MND
        .MAC  ADDR    ;MOVE LOW
        LDA   #<?2    ;BYTE OF ?2
        STA   ?1      ;INTO ?1 AND
        LDA   #>?2    ;HIGH BYTE OF
        STA   ?1+1    ;?2 INTO ?1+1
        .MND
        .MAC  MOVE
        LDA   ?2
        STA   ?1
        LDA   ?2+1
        STA   ?1+1
        .MND
        .MAC  DISK    ;DISK
        LDA   #0      ;OPERATIONS
        TAX           ;LOAD"?2",8
        LDY   #?1     ;?3 IS END OF
        JSR   SETLFS  ;FILE NAME.
        LDA   #?3-?2  ;THUS LENGTH
        LDX   #<?2    ;OF FILE NAME
        LDY   #>?2    ;IF ?3-?2
        JSR   SETNAM
        LDA   #0
        .MND
        .MAC  PRINT
        LDA   #<?1
        LDY   #>?1
        JSR   $AB1E   ;PRINT
        .MND
        .MAC  PUTR
        PHA
        TYA
        PHA
        TXA
        PHA
        .MND
        .MAC  GETR
        PLA
        TAX
        PLA
        TAY
        PLA
        .MND
        .MAC  DADD
        LDA   ?1
        CLC
        ADC   ?2
        STA   ?1
        LDA   ?1+1
        ADC   ?2+1
        STA   ?1+1
        .MND
        .MAC  DSUB
        LDA   ?1
        SEC
        SBC   ?2
        STA   ?1
        LDA   ?1+1
        SBC   ?2+1
        STA   ?1+1
        .MND
        .MAC  PLOT
        LDY   #?1
        LDX   #?2
        CLC
        JSR   $FFF0
        .MND
        .MAC  DSPL
        LDA   #?2-?1
        STA   LENSTR
        LDA   #<?1
        STA   $A3
        LDA   #>?1
        STA   $A4
        JSR   LETML
        .MND
        .OPT  LIST
        .END  MACROS-VARS
        .OPT  NOL
        .LIB  IRQSTUFF
;IRQSTUFF FOR BRM CARTRIDGE
;PROCEDURE ENABLE
;PREPARES IRQ TO TAKE READINGS
;B:NONE
;C:SYS ENABLE
;A:NONE
ENABLE SEI
        LDA   #0
        STA   56334
        LDA   53265
        AND   #127
        STA   53265
        LDA   #250
        STA   53266
        LDA   #5
        STA   53274
        LDA   53273
        STA   53273
        ADDR  $314,START
        LDA   #0
        STA   CUR1    ;20 RDGS SP
        STA   CUR2    ;60 RDGS SP
        STA   IRQ
        STA   BORCOL
        STA   NUMRD   ;# READINGS B
        STA   SORTAF  ;# SORT AFTER
        LDA   #5
        STA   STSIZE
        LDA   #1
        STA   HAFSEC
        STA   H7IRQ
        CLI
        RTS
;
;PROCEDURE DSABLE
;RETURNS IRQ VECTOR TO NORMAL
;B:NONE
;C:SYS DSABLE
;A:NONE
DSABLE SEI
        LDA   #1
        STA   56334
        LDA   #240
        STA   53274
        LDA   53273
        STA   53273
        ADDR  $314,$EA31
        CLI
        RTS
;
;PROCEDURE START (IRQ)
;INTERRUPT CONTROLLER ROUTINE
;
;IRQ HAS THE FOLLOWING VALUES:
;128: TAKE READINGS BEFORE
; 64: TAKE READINGS AFTER
; 32: GET COLLISION DATA
; 16: SOUND NEEDED
;  8: WAITING FOR TRIGGER RELEASE
;  4: SHOT HAS BEEN FIRED
;  2: NEW TARGET DISABLE
;  1: NO BULLETS LEFT
;
;IRQ2 HAS THE FOLLOWING VALUES:
;128: RETURN AFTER ONE SHOT FIRED
; 64: RETURN WHEN NO TARGETS UP
; 32: NO HIT/MISS DETECTION
; 16: NO CROSS DISPLAYED
;  8: CROSS FOR MISS ONLY
;  4: NO DATA SAVE TO INFO2
;  2: DISPLAY TARGET NUMBER
;  1: REPLAY
;
;TARORD (TARGET ORDER) IS SET UP
;IN THE FOLLOWING MANNER:
;BYTE 0:# TARGETS THIS HALF SECOND
;     1:STARTING X COORDINATE
;     2:STARTING Y COORDINATE
;     3:TARGET # (FOR ID)
;     4:TARGET SPRITE POINTER
;     5:BIT 7:SET TO EXPAND X
;           6:SET TO EXPAND Y
;           5:RIGHT X
;           0-3:SPRITE COLOR
;     6:TIME LIMIT*2 (SECONDS)
```

```
;     7:DELAY (FOR DLSTK,CDLSTK)
;     8:INCREMENT (FOR INSTK)
;     9:TIME/FLIGHT (#PIXELS/BYT8)
;    10:X DRIFT
;    11:Y DRIFT
;    12:BIT 7:NOT A TARGET
;           6:REVERSE DIRECTION
;           5:TAR MORE THAN 1 SPR
;           4:DON'T INC TARPRE
;           2-0:# TARGETS
;    13:X CENTER OF MASS OFFSET/2
;    14:Y CENTER OF MASS OFFSET
;
;INFO IS SAVED AS FOLLOWS:
;BYTE 0:TARGET # (FOR ID)
;     1:BULLET X/2
;     2:BULLET Y
;     3:TARGET X/2
;     4:TARGET Y
;     5:NUMR20 (LOW)
;     6:NUMR20 (HIGH)
START  LDA   #4
       BIT   53273
       BEQ   CONTIN
       LDA   53273
       STA   53273
       JMP   $FEBC
CONTIN LDA   53265
       AND   #127
       STA   53265
       LDA   #250
       STA   53266
       LDA   53273
       STA   53273
       LDA   IRQ2
       AND   #1
       BEQ   IRQ010
       JMP   REPLAY
IRQ010 LDX   V+30     ;SPR/SPR
       LDY   V+31     ;SPRITE/DATA
       LDA   IRQ
       AND   #32      ;COLLISION
       BEQ   IRQ020
       STX   SSCOL
       STY   SDCOL
       LDA   IRQ      ;CLEAR
       AND   #223     ; COLLISION
       STA   IRQ      ; FLAG
IRQ020 LDA   IRQ
       AND   #16
       BEQ   IRQ030
;PLAY ANY SOUNDS HERE
IRQ030 LDA   IRQ      ;WAITING FOR
       AND   #8       ; TRIGGER
       BEQ   IRQ040   ; RELEASE?
       LDA   56321    ;YES:HAS IT
       CMP   #247     ; BEEN
       BEQ   IRQ040   ; RELEASED
       LDA   IRQ      ;YES:CLEAR
       AND   #247     ; FLAG
       STA   IRQ
IRQ040 LDA   CRSDLA   ;CROSS ON?
       BEQ   IRQ060
       DEC   CRSDLA
       BNE   IRQ060
       LDA   HITSPR   ;TURN OFF
       EOR   #255     ; CROSS AND
       AND   V+21     ; ANY HIT
       STA   V+21     ; TARGET LEFT
       BIT   IRQ2
       BVC   IRQ050
       CMP   #0
       BNE   IRQ050
       LDA   IRQ
       AND   #63
       STA   IRQ
       JMP   $EA31    ;NORMAL IRQ
IRQ050 LDA   #0
       STA   HITSPR
IRQ060 LDA   IRQ      ;TAKE ANY
       AND   #192     ; READINGS?
       BNE   IRQ070
       LDA   #0
       STA   CUR1
       STA   CUR2
       STA   BORCOL
       JMP   $EA31    ;NORMAL IRQ
IRQ070 LDA   $FB      ;SAVE
       PHA            ; $FB AND $FC
       LDA   $FC
       PHA
       DEC   HAFSEC   ;NEW TARGET?
       BEQ   IRQ080   ; OR TIME LM?
       JMP   IRQ250   ;NOT TIME YET
IRQ080 LDA   #30      ;RESTORE
       STA   HAFSEC   ; HALF SECOND
       MOVE  $FB,TARORD
IRQ090 LDY   #255
       LDA   IRQ      ;IS TARGET
       AND   #2       ; CHECK
       BNE   IRQ140   ; DISABLED?
       LDY   #0
       LDA   ($FB),Y  ;# TARGETS
       CMP   #255
       BNE   IRQ100
       LDA   #32
       STA   FSTAT
       DINC  TARORD
       LDA   IRQ      ;DON'T TAKE
       AND   #63      ; ANY MORE
       STA   IRQ      ; READINGS IF
       JMP   IRQ340   ; NO TARGETS
IRQ100 CMP   #254
       BNE   IRQ110
       LDA   IRQ
       ORA   #2
       STA   IRQ
       JMP   IRQ140
IRQ110 CMP   #253     ;WHISTLE?
       BNE   IRQ130
       LDA   OFFSCR   ;SET FLAG
       ORA   #1
       STA   OFFSCR
IRQ120 DINC  $FB
       JMP   IRQ090
IRQ130 CMP   #128     ;DELAY
       BCC   IRQ150
       SEC
       SBC   #1
       STA   ($FB),Y
       CMP   #128
       BEQ   IRQ120
       DEY
IRQ140 LDA   #0
IRQ150 STA   H5IRQ    ;TARGETS
       INY
       LDX   #7
       LDA   #128
       STA   H3IRQ    ;SPRITE BIT
IRQ160 LDA   H3IRQ
       EOR   #255     ;NOT SPRITE
       STA   H4IRQ    ; BIT
       LDA   V+21
       AND   H3IRQ
       BEQ   IRQ180   ;SPRITE OFF
       LDA   JIFFYS,X
       BMI   IRQ170   ;NO LIMIT
       LEC   JIFFYS,X
       BNE   IRQ170   ;MORE TIME
       LDA   H3IRQ    ;TIME LIMIT
       ORA   OFFSCR   ; HAS EXPIRED
       STA   OFFSCR
       LDA   #128
       STA   JIFFYS,X
       LDA   OFFSCR   ;HAS TIME
       EOR   #255     ; EXPIRED
       AND   V+21     ; ON ALL
       BNE   IRQ170   ; TARGETS?
       LDA   #64
       STA   FSTAT
IRQ170 JMP   IRQ230
IRQ180 LDA   H5IRQ    ;ANY TAR NEED
       BEQ   IRQ170   ; DISPLAYING?
       DEC   H5IRQ    ;ONE LESS TAR
       LDA   H4IRQ    ;CLEAR
       AND   V+23     ; EXPAND Y
       STA   V+23
       LDA   H4IRQ    ;CLEAR
       AND   V+29     ; EXPAND X
       STA   V+29
       LDA   H4IRQ    ;CLEAR
       AND   V+16     ; RIGHT X
       STA   V+16
       TXA            ;X=X*2
       ASL   A
       TAX
       LDA   ($FB),Y  ;X COORDINATE
       STA   V,X
       INY
       LDA   ($FB),Y  ;Y COORDINATE
       STA   V+1,X
       INY
       TXA            ;X=X/2
       LSR   A
       TAX
       LDA   ($FB),Y
       STA   TARNUM,X
       INY
       LDA   ($FB),Y  ;SPRITE PTR
       STA   2040,X
       INY
       LDA   ($FB),Y  ;EXPAND X?
       BPL   IRQ190
       LDA   V+29     ;YES
       ORA   H3IRQ
       STA   V+29
IRQ190 LDA   ($FB),Y  ;EXPAND Y?
       AND   #64
       BEQ   IRQ200
       LDA   V+23     ;YES?
       ORA   H3IRQ
       STA   V+23
IRQ200 LDA   ($FB),Y  ;RIGHT X?
       AND   #32
       BEQ   IRQ210
       LDA   V+16
       ORA   H3IRQ
       STA   V+16
IRQ210 LDA   ($FB),Y  ;SPRITE COLOR
       AND   #15
       STA   V+39,X
       INY
       LDA   ($FB),Y  ;TIME LIMIT
       STA   JIFFYS,X
       INY
       LDA   ($FB),Y  ;DELAY FOR
       STA   DLSTK,X  ; MOVEMENT
       STA   CDLSTK,X
       INY
       LDA   ($FB),Y  ;INCREMENTFOR
       STA   INSTK,X  ; MOVEMENT
       INY
       LDA   ($FB),Y  ;TIME OF
       STA   TIMEFL,X ; FLIGHT
       INY
       LDA   ($FB),Y  ;X DRIFT
       STA   XDRIFT,X
       INY
       LDA   ($FB),Y
       STA   YDRIFT,X
       INY
       LDA   ($FB),Y  ;SPECIAL
       STA   SPECIL,X
       AND   #16      ;COUNT AS
       BNE   IRQ220   ; TARPRE
       INC   TARPRE
       TXA
       JSR   INCNUM
       TAX
IRQ220 INY
       LDA   ($FB),Y  ;CEN X OFFSET
       STA   CENX,X
       INY
       LDA   ($FB),Y  ;CEN Y OFFSET
       STA   CENY,X
       INY
       LDA   V+21     ;TURN ON
       ORA   H3IRQ    ; SPRITE
       STA   V+21
IRQ230 LSR   H3IRQ
       DEX            ;NEXT SPRITE
       BEQ   IRQ240   ;FINISHED?
       JMP   IRQ160
IRQ240 TYA            ;SET UP
       CLC            ; TARORD
       ADC   $FB      ; FOR NEXT
       STA   TARORD   ; TIME
       LDA   $FC
       ADC   #0
       STA   TARORD+1
       LDA   IRQ2     ;SHOW TARGET
       AND   #2       ; NUMBER?
       BEQ   IRQ250
       JSR   SHONUM
IRQ250 LDA   IRQ      ;READY TO
       AND   #5       ; PROCESS?
       BEQ   IRQ270
IRQ260 JMP   IRQ340
IRQ270 LDA   IRQ
       AND   #128
       BNE   BEFORE
       JMP   AFTER
BEFORE JSR   ONSCRN
       ADDR  $FB,XLPB20
       LDA   CUR2
       AND   #127
       TAY
       LDA   SHOTS
       AND   #1
       BEQ   IRQ280
       TYA
       ORA   #128
       TAY
       INC   $FC
IRQ280 LDA   H1IRQ
       STA   XLPB60,Y
       LDA   H2IRQ
       STA   YLPB60,Y
       INY
       STY   CUR2
       DINC  NUMR60
       DEC   H7IRQ
       BNE   IRQ260
       LDA   #3
       STA   H7IRQ
       LDY   CUR1
       LDA   H1IRQ
       STA   ($FB),Y
       INC   $FC
       INC   $FC
       LDA   H2IRQ
       STA   ($FB),Y
       INY
       STY   CUR1
       DINC  NUMR20
       LDY   CURSRT
       INC   CURSRT
       CPY   #5
       BCC   IRQ290
       LDY   #0
       STY   CURSRT
IRQ290 LDA   H1IRQ
       STA   SRTBUF,Y
       TYA
       CLC
       ADC   #5
       TAY
       LDA   H2IRQ
       STA   SRTBUF,Y
       LDA   IRQ      ;HAS TRIGGER
       AND   #12      ; BEEN
       BNE   IRQ300   ; RELEASED?
       LDA   56321    ;TRIGR PULL
       CMP   #247
```

```
        BNE   IRQ300
        BIT   BORCOL  ;LIGHTPEN
        BPL   SWPULL  ; ON SCREEN?
        LDA   #4
        STA   53280
IRQ300  JMP   IRQ340
SWPULL  JSR   BANG
        LDA   #0      ;IF ANYONE
        STA   XVAL    ; HAS THE
        STA   XVAL+1  ; REVELATION
        STA   YVAL    ; THAT SORT-
        STA   YVAL+1  ; ING REALLY
        LDA   H1IRQ   ; IS THE
        BEQ   IRQ320  ; ANSWER,
        ASL   A       ; REMOVE
        LDX   #0      ; THESE LINES
        BCC   IRQ310  ; & REINSERT
        INX           ; THE CALL TO
IRQ310  CLC           ; GETXY IN
        ADC   OFFX    ; CTRMOV
        STA   XVAL    ; ---
        TXA           ; ---
        ADC   OFFX+1  ; ---
        STA   XVAL+1  ; ---
        LDA   H2IRQ   ; ---
        CLC           ; ---
        ADC   OFFY    ; ---
        STA   YVAL    ; ---
IRQ320  LDA   IRQ
        AND   #127
        ORA   #76
        STA   IRQ
        LDA   #0      ;FORMERLY 3
        STA   SORTAF
        LDA   #10
        STA   NUMRD
        JSR   INFOFB
        LDY   #5
        LDA   NUMR20
        STA   ($FB),Y
        STA   LASTSH,Y
        INY
        LDA   NUMR20+1
        STA   ($FB),Y
        STA   LASTSH,Y
        LDA   NUMR60
        STA   LASTSH+7
        LDA   NUMR60+1
        STA   LASTSH+8
        LDA   #0
        STA   CUR1
        STA   CUR2
        STA   NUMR20
        STA   NUMR20+1
        STA   NUMR60
        STA   NUMR60+1
IRQ340  PLA
        STA   $FC
        PLA
        STA   $FB
        JMP   $EA31   ;NORMAL IRQ
AFTER   LDY   CUR2    ;STACK PTR
        INC   CUR2
        LDA   53267   ;X LOCATION
        STA   H1IRQ
        STA   XLPA60,Y
        LDA   53268   ;Y LOCATION
        STA   H2IRQ
        STA   YLPA60,Y
        LDA   SORTAF
        BEQ   IRQ360
        LDY   CURSRT
        INC   CURSRT
        CPY   #5
        BCC   IRQ350
        LDY   #0
        STY   CURSRT
IRQ350  LDA   H1IRQ
        STA   SRTBUF,Y
        TYA
        CLC
        ADC   #5
        TAY
        LDA   H2IRQ
        STA   SRTBUF,Y
        DEC   SORTAF
IRQ360  DEC   NUMRD   ;# READINGS
        BNE   IRQ380  ; TO TAKE
        LDA   IRQ
        EOR   #192
        STA   IRQ
        LDA   #0      ;ZERO STACK
        STA   CUR1    ;PTR IF DONE
        STA   CUR2
        STA   CURSRT
        LDA   #1
        STA   H7IRQ
        BIT   IRQ2
        BMI   IRQ370
        BVC   IRQ380
        LDA   V+21
        AND   #254
        BNE   IRQ380
IRQ370  LDA   IRQ
        AND   #63
        STA   IRQ
IRQ380  JMP   IRQ340
ONSCRN  LDA   53267   ;X READINGS
        STA   H1IRQ
        LDX   53268   ;Y READINGS
        STX   H2IRQ
        CMP   LPCMPX
        BNE   ONSC30
        CPX   LPCMPY
        BNE   ONSC30
        BIT   BORCOL
        BMI   ONSC20
        INC   BORCOL
        LDA   BORCOL
        CMP   #30
        BCS   ONSC10
        RTS
ONSC10  LDA   53280
        ORA   #128
        STA   BORCOL
;ONSC20 LDA   #0
;       STA   H1IRQ
;       STA   H2IRQ
ONSC20  RTS
ONSC30  STA   LPCMPX
        STX   LPCMPY
        BIT   BORCOL
        BPL   ONSC40
        LDA   BORCOL
        STA   53280
ONSC40  LDA   #0
        STA   BORCOL
        RTS
INFOFB  LDA   #0
        STA   $FC
        LDA   SHOTS
        ASL   A
        ASL   A
        ASL   A
        ROL   $FC
        CLC
        ADC   #<INFO
        STA   $FB
        LDA   $FC
        ADC   #>INFO
        STA   $FC
        RTS
INCNUM  LDX   CURNUM+1
        INX
        CPX   #10
        BCC   INC020
        LDX   CURNUM
        INX
        CPX   #10
        BCC   INC010
        LDX   #0
INC010  STX   CURNUM
        LDX   #0
INC020  STX   CURNUM+1
        RTS
SHONUM  LDY   #0
        LDX   #80     ;LEAD SPACE
        LDA   CURNUM
        BEQ   SHN010
        ASL   A
        ASL   A
        ASL   A
        TAX
SHN010  LDA   NUMDAT,X
        STA   @256,Y
        INX
        INY
        CPY   #8
        BCC   SHN010
        LDA   CURNUM+1
        ASL   A
        ASL   A
        ASL   A
        TAX
SHN020  LDA   NUMDAT,X
        STA   @256,Y
        INX
        INY
        CPY   #16
        BCC   SHN020
        RTS
NUMDAT .BYTE 195,153,145,137,153,153,
195,255,231,231,199,231,231,231,129
       .BYTE 255,195,153,249,243,207,
159,129,255,195,153,249,227,249,153
       .BYTE 195,255,249,241,225,153,
128,249,249,255,129,159,131,249,249
       .BYTE 153,195,255,195,153,159,
131,153,153,195,255,129,153,243,231
       .BYTE 231,231,231,255,195,153,
153,195,153,153,195,255,195,153,153
       .BYTE 193,249,153,195,255,255,
255,255,255,255,255,255,255,255,255
       .END  IRQSTUFF
       .LIB  REPLAY
;REPLAY FOR 10-LEVEL BRM PROGRAM
REPLAY  DEC   H7IRQ
        BEQ   RPL010
        JMP   RPL090
RPL010  LDA   #3
        STA   H7IRQ
        LDY   CUR1
        INY
        CPY   STOP    ;LAST RDG?
        BNE   RPL020
        LDA   YVAL
        BEQ   RPL050
        SEC
        SBC   TRAJ
        STA   H6IRQ
        LDA   XVAL
        BEQ   RPL050
        SEC
        SBC   WDRIFT
        TAX
        LDA   XVAL+1
        SBC   WDRIFT+1
        STA   H5IRQ
        TXA
        LDX   H5IRQ
        DEY
        JMP   RPL040
RPL020  DEY
        LDA   12928,Y ;BULY
        BEQ   RPL050
        CLC
        ADC   OFFY
        STA   H6IRQ
        LDX   #0
        LDA   12672,Y ;BULX
        BEQ   RPL050
        ASL   A
        BCC   RPL030
        LDX   #1
RPL030  CLC
        ADC   OFFX
        STA   H4IRQ
        TXA
        ADC   OFFX+1
        STA   H5IRQ
        LDA   H4IRQ
RPL040  SEC
        SBC   H1IRQ   ;TARX L
        STA   H4IRQ   ;BULX-TARX
        LDA   H5IRQ
        SBC   H2IRQ   ;TARX H
        STA   H5IRQ
        LDA   H4IRQ
        CLC
        ADC   #231    ;254-23 SPOST
        STA   V+2     ; OFFSET
        LDA   H5IRQ
        ADC   #0
        BNE   RPL060
        LDA   V+16
        AND   #253
        STA   V+16
        JMP   RPL070
RPL050  LDA   #0
        STA   V+2
        STA   V+3
        LDA   V+16
        AND   #253
        STA   V+16
        JMP   RPL080
RPL060  LDA   V+16
        ORA   #2
        STA   V+16
RPL070  LDA   H6IRQ   ;BULY
        SEC
        SBC   H3IRQ   ;BULY-TARY
        CLC
        ADC   #127    ;152-25 SPOST
        STA   V+3     ; OFFSET
RPL080  INY
        STY   CUR1
        CPY   STOP
        BNE   RPL090
        LDA   #0
        STA   IRQ2
RPL090  JMP   $EA31   ;NORMAL IRQ
        .END  REPLAY
;
;PROCEDURE ROTATE
;TAKES READINGS BEFORE TRIGGER
;  PULL AND MOVES THEM SO THAT
;  THE OLDEST READINGS ARE FIRST.
;C:SYS ROTATE
;A:RDGS IN BUFFERS ARE ADJUSTED.
ROTATE  LDA   1
        PHA
        LDA   #54
        STA   1
        LDA   LASTSH+6
        BEQ   ROT030
        LDY   LASTSH+5
        BEQ   ROT030
        ADDR  $FB,XLPB20
        ADDR  $FD,YLPB20
        LDX   #0
        LDA   SHOTS
        AND   #1
        BEQ   ROT010
        INC   $FC
        INC   $FE
ROT010  LDA   ($FB),Y
        STA   HLDBUF,X
        LDA   ($FD),Y
        STA   HLDBUF+256,X
        INY
        INX
        BNE   ROT010
        LDY   #0
ROT020  LDA   HLDBUF,Y
```

```
        STA   ($FB),Y
        LDA   HLDBUF+256,Y
        STA   ($FD),Y
        INY
        BNE   ROT020
ROT030  LDA   LASTSH+7
        ASL   A
        BEQ   ROT070
        LDA   LASTSH+8
        BNE   ROT040
        BCC   ROT070
ROT040  LDA   #<XLPB60
        STA   $FB
        STA   $FD
        LDA   #>XLPB60
        STA   $FC
        STA   $FE
        INC   $FE
        LDA   LASTSH+7
        AND   #127
        TAY
        LDA   SHOTS
        AND   #1
        BEQ   ROT050
        LDA   $FB
        CLC
        ADC   #128
        STA   $FB
        STA   $FD
        LDA   $FC
        ADC   #0
        STA   $FC
        STA   $FE
        INC   $FE
ROT050  LDA   ($FB),Y
        STA   HLDBUF,X
        LDA   ($FD),Y
        STA   HLDBUF+128,X
        INY
        TYA
        AND   #127
        TAY
        INX
        BPL   ROT050
        LDY   #0
ROT060  LDA   HLDBUF,Y
        STA   ($FB),Y
        LDA   HLDBUF+128,Y
        STA   ($FD),Y
        INY
        BPL   ROT060
ROT070  PLA
        STA   1
        RTS
NOTTRG  LDA   56321    ;REPEAT
        CMP   #255     ; UNTIL
        BNE   NOTTRG   ; TRIGGER IS
        RTS            ; RELEASED
BANG    LDA   #15
        STA   54296
        LDA   #10
        STA   54277
        LDA   #30
        STA   54273
        LDA   #128
        STA   54276
        LDA   #129
        STA   54276
        RTS
WHISTL  LDY   #23
        LDA   #0
WHSL10  STA   SID,Y
        DEY
        BPL   WHSL10
        LDA   #18
        STA   DELAY
WHSL20  LDA   #180
        STA   SID+1
        LDA   #46
        STA   SID
        LDA   #15
        STA   SID+5
        STA   SID+24
        LDY   #7
WHSL30  LDX   #255
WHSL40  DEX
        BNE   WHSL40
        DEY
        BNE   WHSL30
        LDA   #170
        STA   SID+1
        LDA   #6
        STA   SID
        LDA   #21
        STA   SID+4
        LDY   #30
WHSL50  LDX   #255
WHSL60  DEX
        BNE   WHSL60
        DEY
        BNE   WHSL50
        DEC   DELAY
        BNE   WHSL20
        LDY   #23
        LDA   #0
WHSL70  STA   SID,Y
        DEY
        BPL   WHSL70
        RTS
;
;PROCEDURE WAIT
;CAUSES A DELAY
;B:LDY WITH AN APPROPRIATE DELAY
;C:JSR WAIT
;A:NONE
WAIT    LDX   #255
WAIT1   DEX
        BNE   WAIT1
        DEY
        BNE   WAIT1
        RTS
SCENE   LDA   $FB      ;CHIP #/BANK
        STA   $DFFF
        LDY   #0
        LDA   ($FD),Y
        STA   $FB      ;CRUNCH CODE
        DINC  $FD
        LDA   #43
        STA   53265
        LDA   #29
        STA   53272
        ADDR  $A3,1024
SCL010  LDA   ($FD),Y
        STA   ($A3),Y
        DINC  $FD
        DINC  $A3
        LDA   $A3
        CMP   #<2024
        BNE   SCL010
        LDA   $A4
        CMP   #>2024
        BNE   SCL010
        ADDR  $A3,8192
SCL020  LDA   $FB      ;CRUNCH CODE
        STA   ($A3),Y
        DINC  $A3
        LDA   $A3
        CMP   #<16192
        BNE   SCL020
        LDA   $A4
        CMP   #>16192
        BNE   SCL020
        ADDR  $A3,8192
SCL030  LDA   ($FD),Y
        CMP   $FB      ;CRUNCH CODE
        BNE   SCL040
        DINC  $FD
        LDA   ($FD),Y
        STA   $A3
        DINC  $FD
        LDA   ($FD),Y
        CLC
        ADC   #>8192
        STA   $A4
        JMP   SCL050
SCL040  STA   ($A3),Y
        DINC  $A3
SCL050  DINC  $FD
        LDA   $A3
        CMP   #<16192
        BNE   SCL030
        LDA   $A4
        CMP   #>16192
        BNE   SCL030
        LDA   #59
        STA   53265
        LDA   #32      ;RAM
        STA   $DFFF
        RTS
;PROCEDURE SEEVAL
;MOVE VALUES FROM UNDER ROM
; SINCE SHOTS WILL HAVE ALREADY
; BEEN INCREMENTED, THIS ROUTINE
; MOVES THE HIGH BUFFER IF SHOTS
; IS EVEN.  IT MOVES THE LOW
; BUFFER IF SHOTS IS ODD
SEEVAL  LDA   1
        PHA
        LDA   #54      ;SELECT RAM
        STA   1        ;AT $A000
        ADDR  $FB,XLPB20
        LDY   #0
        LDA   SHOTS
        AND   #1
        BNE   SEE010
        INC   $FC
SEE010  LDA   ($FB),Y
        STA   12672,Y
        INY
        BNE   SEE010
        INC   $FC
        INC   $FC
SEE020  LDA   ($FB),Y
        STA   12928,Y
        INY
        BNE   SEE020
        PLA
        STA   1
        RTS
;PROCEDURE COLORS
;CHANGES COLOR OF PART OF BITMAP
;POKE HOLD1 WITH START LINE
;  POKE HOLD2 WITH # OF LINES
;  POKE $FD WITH THE COLOR
COLORS  ADDR  $FB,1024
        LDX   HOLD1    ;START LINE
        BEQ   COL020
COL010  JSR   ADD40
        DEX
        BNE   COL010
COL020  LDX   HOLD2    ;HOW MANY
COL030  LDY   #39
COL040  LDA   $FD      ;COLOR
        STA   ($FB),Y
        DEY
        BPL   COL040
        JSR   ADD40
        DEX
        BNE   COL030
        RTS
ADD40   LDA   $FB      ;ADD 40 TO
        CLC            ; $FB FOR
        ADC   #40      ; NEXT LINE
        STA   $FB
        LDA   $FC
        ADC   #0
        STA   $FC
        RTS
        .LIB  MATHFUNCTIONS
;FUNCTION STDEV
;PUT INTO FLOATING POINT REGISTER:
; START BYTE IN STACK * 256 +
; END BYTE IN STACK
STDEV   JSR   $B1BF
STDL10  LDA   100
        STA   NSIZE
        LDA   101
        STA   TIMES
        SEC
        SBC   100
        CLC
        ADC   #1
        CMP   #2
        BCS   STDL20
        JMP   STDL40
STDL20  PHA
        LDA   #0
        TAY
        JSR   $B391    ;FLOAT
        JSR   $BBC7    ;SAVE
        LDA   #0
        STA   SUMX
        STA   SUMX+1
STDL30  LDA   1
        PHA
        LDA   #54
        STA   1
        LDY   TIMES
        LDA   ($FD),Y
        STA   HOLD
        CLC
        ADC   SUMX
        STA   SUMX
        LDA   SUMX+1
        ADC   #0
        STA   SUMX+1
        PLA
        STA   1
        LDY   HOLD
        LDA   #0
        JSR   $B391    ;FLOAT
        JSR   $BC0F    ;PUT IN FA2
        LDY   HOLD
        LDA   #0
        JSR   $B391    ;FLOAT
        JSR   $BA2B    ;MULTIPLY
        LDA   #$5C
        LDY   #0
        JSR   $BA8C    ;LOAD FA2
        JSR   $B86A    ;ADD
        JSR   $BBC7    ;SAVE FA1
        DEC   TIMES
        LDY   TIMES
        CPY   NSIZE
        BCS   STDL30
        PLA
        STA   NSIZE
        LDY   SUMX
        LDA   SUMX+1
        JSR   $B391    ;FLOAT
        JSR   $BC0F    ;PUT IN FA2
        LDY   SUMX
        LDA   SUMX+1
        JSR   $B391    ;FLOAT
        JSR   $BA2B    ;MULTIPLY
        JSR   $BC0F    ;PUT IN FA2
        LDA   #0
        LDY   NSIZE
        JSR   $B391    ;FLOAT
        JSR   $BB12    ;DIVIDE
        LDA   #$5C
        LDY   #0
        JSR   $BA8C    ;LOAD FA2
        JSR   $B853    ;SUBTRACT
        JSR   $BC0F    ;PUT IN FA2
        LDA   #0
        LDY   NSIZE
        DEY
        JSR   $B391    ;FLOAT
        JSR   $BB12    ;DIVIDE
        LDA   102      ;SIGN FOR FA1
        AND   #127
        STA   102      ;ABS
        JSR   $BF71    ;SQR
```

```
STDL40 RTS
RADERR JSR  $B1BF    ;SHOT #
       LDX  SHOTS
       LDA  101
       AND  #127
       STA  SHOTS
       JSR  INFOFB
       STX  SHOTS
       LDA  1
       PHA
       LDA  #54
       STA  1
       BIT  101
       BPL  RER020
       LDA  $FB      ;MAKE SURE
       CLC           ; VALUE WITH
       ADC  #<344    ; OFFSET IS
       STA  $FD      ; STORED IN
       LDA  $FC      ; RECORD
       ADC  #>344    ; KEEPING
       STA  $FE      ; BUFFER
       LDY  #1
       LDA  ($FB),Y ;BULLET X
       ASL  A
       STA  HOLD1
       LDX  #0
       BCC  RER010
       INX
RER010 STX  HOLD1+1
       LDA  HOLD1
       CLC
       ADC  OFFX
       STA  HOLD1
       LDA  HOLD1+1
       ADC  OFFX+1
       STA  HOLD1+1
       LDA  HOLD1
       LSR  HOLD1+1
       ROR  A
       STA  ($FB),Y
       STA  ($FD),Y
       INY
       LDA  ($FB),Y ;BULLET Y
       CLC
       ADC  OFFY
       STA  ($FB),Y
       STA  ($FD),Y
RER020 LDY  #3
       LDA  ($FB),Y ;TARGET X
       LDY  #1
       SEC
       SBC  ($FB),Y ;BULLET X
       STA  HOLD1
       LDA  #0
       SBC  #0
       STA  HOLD1+1
       ASL  HOLD1    ;X-X*2
       ROL  HOLD1+1
       LDY  #4
       LDA  ($FB),Y ;TARGET Y
       LDY  #2
       SEC
       SBC  ($FB),Y ;BULLET Y
       STA  HOLD3
       LDA  #0
       SBC  #0
       STA  HOLD3+1
       PLA
       STA  1
       LDY  HOLD1
       LDA  HOLD1+1
       JSR  FLOAT
       JSR  $BC0F    ;MOVE TO FA2
       LDY  HOLD1
       LDA  HOLD1+1
       JSR  FLOAT
       JSR  $BA2B    ;MULTIPLY
       JSR  $BBC7    ;SAVE
       LDY  HOLD3
       LDA  HOLD3+1
       JSR  FLOAT
       JSR  $BC0F    ;MOVE TO FA2
       LDY  HOLD3
       LDA  HOLD3+1
       JSR  FLOAT
       JSR  $BA2B    ;MULTIPLY
       LDA  #$5C
       LDY  #0
       JSR  $BA8C    ;LOAD FA2
       JSR  $B86A    ;ADD
       JSR  $BF71    ;SQR
       RTS
       .END  MATHFUNCTIONS
       .LIB  SHOTGROUPS
;PROCEDURE SHTGRP (SHOTGROUPS)
;POKE $FB WITH START ADDRESS.
;  POKE HOLD3,TAR# (ADD 128 IF
;  ONLY HIT/MISS INFO IS DESIRED.)
;  POKE HOLD4 WITH # OF SHOTS
;  AT END, HITS, MISSES ARE SET,
;  HOLD1 WILL HAVE # UNGRAPHABLE
;  SHOTS.  HOLD2 WILL HAVE TOTAL
;  # SHOTS FIRED AT THIS TARGET.
SHTGRP LDA  1
       PHA
       LDA  #54      ;SELECT RAM
       STA  1        ; AT $A000
       LDA  #0
       STA  HITS
       STA  MISSES
       STA  HOLD1
       STA  HOLD2
       LDA  HOLD3    ;SEARCH
       AND  #127     ; TAR #
       STA  CODE1
SHG010 LDY  #0
       LDA  ($FB),Y ;TAR #
       AND  #127
       CMP  CODE1
       BNE  SHG040
       INC  HOLD2
       LDA  ($FB),Y
       BMI  SHG020
       INC  HITS
       JMP  SHG030
SHG020 INC  MISSES
SHG030 BIT  HOLD3
       BPL  SHG050
SHG040 JMP  SHG100
SHG050 INY
       LDA  ($FB),Y ;BULX
       LDX  #0
       ASL  A
       BCC  SHG060
       INX
SHG060 STA  XVAL     ;BULX*2
       STX  XVAL+1
       LDY  #3
       LDA  ($FB),Y ;TARX
       LDX  #0
       ASL  A
       BCC  SHG070
       INX
SHG070 STA  YVAL     ;TARX*2
       STX  YVAL+1
       LDA  XVAL     ;BULX-TARX
       SEC
       SBC  YVAL
       STA  XVAL
       LDA  XVAL+1
       SBC  YVAL+1
       STA  XVAL+1
       LDA  XVAL
       CLC
       ADC  #140     ;172-24
       STA  XVAL
       LDA  XVAL+1
       ADC  #0
       STA  XVAL+1
       BEQ  SHG080
       LDA  XVAL
       CMP  #<320
       BCC  SHG080
       INC  HOLD1
       JMP  SHG100
SHG080 LDY  #2
       LDA  ($FB),Y ;BULY
       LDY  #4
       SEC
       SBC  ($FB),Y ;TARY
       CLC
       ADC  #100     ;150-50
       STA  YVAL
       CMP  #200
       BCC  SHG090
       INC  HOLD1
       JMP  SHG100
SHG090 JSR  BITPLT
       INC  YVAL
       JSR  BITPLT
       DINC XVAL
       JSR  BITPLT
       DEC  YVAL
       JSR  BITPLT
SHG100 LDA  $FB      ;ADD 8 FOR
       CLC           ; NEXT TIME
       ADC  #8
       STA  $FB
       LDA  $FC
       ADC  #0
       STA  $FC
       DEC  HOLD4
       BEQ  SHG110
       JMP  SHG010
SHG110 PLA
       STA  1
       RTS
BITPLT STY  HOLDY
       LDA  #0
       STA  $FD
       STA  $FE
       STA  $A4
       STA  YVAL+1
PL020  LDA  YVAL
       AND  #248     ;INT(Y/8)*320
       LDX  #3
PL030  ASL  A
       ROL  $A4
       DEX
       BNE  PL030
       STA  $A3
       LDX  #5
PL040  LDA  $FD
       CLC
       ADC  $A3
       STA  $FD
       LDA  $FE
       ADC  $A4
       STA  $FE
       DEX
       BNE  PL040
       LDA  XVAL     ;8*INT(X/8)
       AND  #248
       CLC
       ADC  $FD
       STA  $FD
       LDA  XVAL+1
       ADC  $FE
       CLC
       ADC  #>8192 ;+8192
       STA  $FE
       LDA  YVAL     ;Y AND 7
       AND  #7
       TAY
       LDA  XVAL     ;X AND 7
       AND  #7
       STA  $A3
       LDA  #7
       SEC
       SBC  $A3      ;7-(X AND 7)
       TAX
       LDA  #1
       CPX  #0
       BEQ  PL060
PL050  ASL  A        ;2^(7-(X
       DEX           ; AND 7))
       BNE  PL050
PL060  ORA  ($FD),Y
       STA  ($FD),Y
       LDY  HOLDY
       CLC
       RTS
       .END  SHOTGROUPS
       .LIB  LETTERS
;F1 TOGGLE SIZE (1X, 2X)
;F2 SET CURSOR (X,Y NEXT 4 BYTES)
;F3 SPECIAL INSTRUCTIONS
;    BIT 0 SET:BLANK SCREEN
;    BIT 0 CLR:UNBLANK SCREEN
;    BIT 1 SET:ENTER TEXT MODE
;    BIT 1 CLR:ENTER GRAPHICS MODE
;    BIT 2 SET:SET UPPER CASE MODE
;    BIT 2 CLR:NO CHANGE
;    BIT 3 SET:SET LOWER CASE MODE
;    BIT 3 CLR:NO CHANGE
;F5 SCREEN COLOR (IN NEXT 2 BYTES)
;F6 BORDER COLOR (IN NEXT 2 BYTES)
;F7 CHARACTER COLOR (NEXT 2 BYTES)
;F8 RETURN
LETERS LDA  71
       STA  $FD
       LDA  72
       STA  $FE
       LDY  #0
       LDA  ($FD),Y ;LENGTH OF
       BNE  LET010
       RTS
LET010 STA  LENSTR   ;STRING
       INY
       LDA  ($FD),Y ;ADDRESS OF
       STA  $A3      ;STRING NOW
       INY           ;IN $A3 AND
       LDA  ($FD),Y ;$A4
       STA  $A4      ;---
LETML  LDY  #0
       STY  FLAGS    ;CLEAR FLAGS
       LDA  53272    ;WHICH SET?
       AND  #2
       BEQ  LET020   ;SET 1
       LDA  #4       ;SET 2
       STA  FLAGS
LET020 LDA  PRGNUM
       BNE  LET025
       JSR  DSABLE
LET025 LDA  56334    ;TURN OFF
       AND  #254     ;KEYSCAN
       STA  56334    ;TIMER
       LDA  1        ;SWITCH
       AND  #251     ;IN
       STA  1        ;CHARS
LET030 LDA  #0
       STA  HOLDA
       LDA  214      ;LINE NUMBER
       STA  $FE      ;LINE*256
       ASL  A        ;LINE*2
       ASL  A        ;LINE*4
       ASL  A        ;LINE*8
       ASL  A        ;LINE*16
       BCC  LET040
       INC  HOLDA
LET040 ASL  HOLDA
       ASL  A        ;LINE*32
       BCC  LET050
       INC  HOLDA
LET050 ASL  HOLDA
       ASL  A        ;LINE*64
       BCC  LET060
       INC  $FE
LET060 STA  $FD      ;LINE*320
       LDA  HOLDA
       CLC
       ADC  $FE
       STA  $FE
       LDA  211      ;COLUMN
```

```
        ASL   A       ;COLUMN*2
        ASL   A       ;COLUMN*4
        ASL   A       ;COLUMN*8
        BCC   LET070
        INC   $FE
        CLC
LET070  ADC   $FD
        STA   $FD     ;L*320+C*8
        LDA   $FE
        ADC   #32     ;+8192
        STA   $FE
LET080  LDA   ($A3),Y
        STY   HOLDY
        CMP   #142    ;SET 1
        BNE   LET090
        LDA   FLAGS
        AND   #251
        STA   FLAGS
        JMP   LET600
LET090  CMP   #14     ;SET 2
        BNE   LET100
        LDA   FLAGS
        ORA   #4
        STA   FLAGS
        JMP   LET600
LET100  CMP   #133    ;F1-ENLARGE
        BNE   LET110  ;OR
        LDA   FLAGS   ;NORMAL
        EOR   #1
        STA   FLAGS
        JMP   LET600
LET110  CMP   #137    ;F2-CURSOR
        BNE   LET160
        JSR   BYTE2
        BCC   LET130
LET120  JMP   LET610
LET130  CMP   #40
        BCS   LET140
        STA   211     ;X
LET140  JSR   BYTE2
        BCS   LET120
        CMP   #25
        BCS   LET150
        STA   214
        LDY   #4
        STY   210
        ASL   A
        ASL   A
        ASL   A
        STA   HOLDA
        STA   209
MUL5    LDA   HOLDA
        CLC
        ADC   209
        STA   209
        LDA   210
        ADC   #0
        STA   210
        DEY
        BNE   MUL5
LET150  JMP   LET600
LET160  CMP   #134    ;F3-SPC INSTR
        BEQ   LET170
        JMP   LET250
LET170  JSR   BYTE2
        BCC   LET180
        JMP   LET610
LET180  LDA   1       ;VIC CHIP
        ORA   #4      ;BACK IN FOR
        STA   1       ;A SECOND
        LDA   HOLDA
        AND   #1
        BEQ   LET190
        LDA   53265   ;BLANK SCREEN
        AND   #239
        STA   53265   ;SCREEN
        JMP   LET200
LET190  LDA   53265   ;UNBLANK
        ORA   #16     ;SCREEN
        STA   53265
LET200  LDA   HOLDA
        AND   #2
        BEQ   LET210
        LDA   53265   ;SET
        AND   #223    ;TEXT
        STA   53265   ;MODE
        LDA   53272
        AND   #247
        STA   53272
        JMP   LET220
LET210  LDA   53265   ;SET
        ORA   #32     ;BIT MAP
        STA   53265   ;MODE
        LDA   53272
        ORA   #8
        STA   53272
LET220  LDA   HOLDA
        AND   #4
        BEQ   LET230
        LDA   53272   ;SET UPPER
        AND   #253    ;CASE
        STA   53272
        LDA   FLAGS
        AND   #251
        STA   FLAGS
LET230  LDA   HOLDA
        AND   #8
        BEQ   LET240
        LDA   53272   ;SET LOWER
        ORA   #2      ;CASE
        STA   53272
        LDA   FLAGS
        ORA   #4
        STA   FLAGS
LET240  LDA   1       ;VIC CHIP
        AND   #251    ;OUT AGAIN
        STA   1
        JMP   LET600
LET250  CMP   #18     ;REVERSE
        BNE   LET260  ;ON
        LDA   FLAGS
        ORA   #2
        STA   FLAGS
        JMP   LET600
LET260  CMP   #146    ;REVERSE
        BNE   LET262  ;OFF
        LDA   FLAGS
        AND   #253
        STA   FLAGS
        JMP   LET600
LET262  CMP   #138    ;F4-ARROWS
        BNE   LET270
        JSR   BYTE2
        BCC   LET264
LET263  JMP   LET610
LET264  SEC
        SBC   #1
        CMP   #8
        BCS   LET263
        ASL   A
        ASL   A
        ASL   A
        CLC
        ADC   #<ARROWS
        STA   $FB
        LDA   #>ARROWS
        ADC   #0
        STA   $FC
        JMP   LET520
LET270  CMP   #135    ;F5-COLOR
        BNE   LET290
        JSR   BYTE2
        BCC   LET280
        JMP   LET610
LET280  LDA   #240
        STA   CODE1
        JSR   ADJSCR
        JMP   LET600
LET290  CMP   #139    ;F6-BORDER
        BNE   LET310  ;COLOR
        JSR   BYTE2
        BCC   LET300
        JMP   LET610
LET300  TAX
        LDA   1       ;SWITCH IN
        ORA   #4      ;VIC CHIP FOR
        STA   1       ;A SECOND
        STX   53280   ;BORDER
        LDA   1       ;SWITCH OUT
        AND   #251    ;VIC CHIP
        STA   1       ;AGAIN
        JMP   LET600
LET310  CMP   #147    ;CLEAR SCREEN
        BNE   LET340
        LDA   #<8192
        STA   $FD
        LDA   #>8192
        STA   $FE
        LDY   #0
LET320  LDA   #0
        STA   ($FD),Y
        DINC  $FD
        LDA   $FD
        CMP   #<16192
        BNE   LET320
        LDA   $FE
        CMP   #>16192
        BNE   LET320
LET330  LDA   #>1024
        STA   210
        LDA   #0
        STA   209
        STA   211
        STA   214
        JMP   LET600
LET340  CMP   #29     ;RIGHT ARROW
        BNE   LET350
        JSR   TLINE
        JMP   LET600
LET350  CMP   #17     ;DOWN ARROW
        BNE   LET360
        JSR   LET620
SKIP    JMP   LET600
LET360  CMP   #145    ;UP ARROW
        BNE   LET370
        LDA   FLAGS
        AND   #1
        STA   $FF
BIG1    LDA   214
        BEQ   SKIP
        DEC   214
        LDA   209
        SEC
        SBC   #40
        STA   209
        LDA   210
        SBC   #0
        STA   210
        SBC   #0
        STA   210
        DEC   $FF
        BEQ   BIG1
        JMP   LET600
LET370  CMP   #157    ;LEFT ARROW
        BNE   LET390
        LDA   FLAGS
        AND   #1
        STA   $FF
BIG2    LDA   211
        BNE   LET380
        LDA   214
        BEQ   SKIP
        DEC   214
        LDA   #40
        STA   211
        LDA   209
        SEC
        SBC   #40
        STA   209
        LDA   210
        SBC   #0
        STA   210
LET380  DEC   211
        DEC   $FF
        BEQ   BIG2
        JMP   LET600
LET390  CMP   #13
        BNE   LET410
LET400  LDA   #40
        STA   211
        JSR   TLINE
        JMP   LET600
LET410  CMP   #136    ;F7-CHAR COL
        BNE   LET430
        JSR   BYTE2
        BCC   LET420
        JMP   LET610
LET420  LDA   #15
        STA   CODE1
        LDA   HOLDA
        ASL   A
        ASL   A
        ASL   A
        ASL   A
        STA   HOLDA
        JSR   ADJSCR
        JMP   LET600
LET430  CMP   #140    ;F8-RETURN
        BEQ   LET400
        CMP   #19     ;HOME
        BNE   LET440
        JMP   LET330
LET440  LDX   #0
LET450  LDA   SPCIAL,X
        BEQ   LET470
        CMP   ($A3),Y
        BEQ   LET460
        INX
        INX
        JMP   LET450
LET460  INX
        LDA   SPCIAL,X
        STA   646
        JMP   LET600
LET470  LDA   ($A3),Y
        CMP   #96
        BCS   LET480
        AND   #191    ;X<96
        JMP   LET500
LET480  CMP   #161
        BCS   LET490
        AND   #223    ;96<X<161
        JMP   LET500
LET490  AND   #127    ;161<X<256
        ORA   #64
LET500  STA   $FB
        LDA   #0
        STA   $FC
        LDX   #3      ;*8
LET510  ASL   $FB
        ROL   $FC
        DEX
        BNE   LET510
        LDA   $FC
        CLC
        ADC   #$D0
        STA   $FC
        LDA   FLAGS   ;WHICH SET?
        AND   #4
        BEQ   LET520  ;SET 1
        LDA   $FC     ;SET 2
        CLC
        ADC   #8
        STA   $FC
LET520  LDA   209
        CLC
        ADC   211
        STA   $C3
        LDA   210
        ADC   #0
        STA   $C4
        LDX   #8
        LDA   646
        ASL   A
        ASL   A
        ASL   A
```

```
        ASL   A
        STA   COLOR
LET530  LDY   #0
        LDA   ($FB),Y ;CHARS
        STA   HOLDA
        LDA   FLAGS
        BEQ   LET550
        AND   #1
        BEQ   LET540
        JSR   DOUBLE
LET540  LDA   FLAGS
        AND   #2
        BEQ   LET550
        LDA   HOLDA
        EOR   #255
        STA   HOLDA
        LDA   HOLDA+1
        EOR   #255
        STA   HOLDA+1
LET550  LDA   HOLDA
        STA   ($FD),Y ;BIT MAP
        LDA   ($C3),Y ;COLOR
        AND   #15
        ORA   COLOR
        STA   ($C3),Y
        LDA   FLAGS
        AND   #1
        BEQ   LET570
        LDA   214
        CMP   #24
        BEQ   LET570
        INY
        LDA   HOLDA
        STA   ($FD),Y
        LDA   ($C3),Y ;COLOR
        AND   #15
        ORA   COLOR
        STA   ($C3),Y
        LDY   #40
        STA   ($C3),Y
        INY
        STA   ($C3),Y
        LDY   #8
        LDA   HOLDA+1
        STA   ($FD),Y
        INY
        STA   ($FD),Y
        DINC  $FB
        DINC  $FD
        DINC  $FD
        DEX
        BNE   LET560
        INC   211
        DEC   $FE
        LDA   $FD
        SEC
        SBC   #64
        STA   $FD
        LDA   $FE
        SBC   #0
        STA   $FE
        JMP   LET590
LET560  CPX   #4
        BNE   LET580
        LDA   $FD
        CLC
        ADC   #56
        STA   $FD
        LDA   $FE
        ADC   #1
        STA   $FE
        JMP   LET530
LET570  DINC  $FD   ;BIT MAP
        DINC  $FB   ;CHARS
        DEX
        BEQ   LET590
LET580  JMP   LET530
LET590  JSR   TLINE
LET600  LDY   HOLDY
        INY
        DEC   LENSTR
        BEQ   LET610
        JMP   LET030
LET610  LDA   1
        ORA   #4
        STA   1
        LDA   56334
        ORA   #1
        STA   56334
        LDA   PRGNUM
        BNE   LETRET
        JSR   ENABLE
LETRET  RTS
TLINE   INC   211
        LDA   211
        CMP   #40
        BCC   LET640
        LDA   #0
        STA   211
LET620  LDA   FLAGS
        AND   #1
        STA   $FF
LET630  LDA   214
        CMP   #24
        BEQ   LET640
        INC   214
        LDA   209
        CLC
        ADC   #40
        STA   209
        LDA   210
        ADC   #0
        STA   210
        DEC   $FF
        BEQ   LET630
LET640  RTS
ADJSCR  LDY   #<1024
        STY   $FD
        LDY   #>1024
        STY   $FE
ADJ01   LDY   #0
        LDA   ($FD),Y
        AND   CODE1
        ORA   HOLDA
        STA   ($FD),Y
        DINC  $FD
        LDY   $FD
        CPY   #<2024
        BNE   ADJ01
        LDY   $FE
        CPY   #>2024
        BNE   ADJ01
        RTS
DOUBLE  TXA
        PHA
        LDA   HOLDA
        LDX   #0
        STX   HOLDA
        STX   HOLDA+1
        TAX
        LDA   #128
        STA   CODE1
        LDA   #192
        STA   CODE2
DBL1    TXA
        AND   CODE1
        BEQ   DBL2
        LDA   CODE2
        ORA   HOLDA
        STA   HOLDA
DBL2    LSR   CODE1
        LSR   CODE2
        LSR   CODE2
        BNE   DBL1
        LDA   #192
        STA   CODE2
DBL3    TXA
        AND   CODE1
        BEQ   DBL4
        LDA   CODE2
        ORA   HOLDA+1
        STA   HOLDA+1
DBL4    LSR   CODE2
        LSR   CODE2
        LSR   CODE1
        BNE   DBL3
        PLA
        TAX
        RTS
BYTE2   JSR   UPDATE
        BCS   BYL20
        SEC
        SBC   #'0
        ASL   A
        STA   HOLDA
        LDY   #4
BYL10   CLC
        ADC   HOLDA
        DEY
        BNE   BYL10
        STA   HOLDA
        JSR   UPDATE
        BCS   BYL20
        SEC
        SBC   #'0
        CLC
        ADC   HOLDA
        STA   HOLDA
        CLC
BYL20   RTS
UPDATE  LDY   HOLDY
        INY
        DEC   LENSTR
        BNE   UP10
        SEC
        RTS
UP10    LDA   ($A3),Y
        STY   HOLDY
        CLC
        RTS
        .OPT  NOL
SPCIAL  .BYTE 144,0,5,1,28,2,159,3,156,4,
30,5,31,6
        .BYTE 158,7,129,8,149,9,150,10,
151,11,152,12
        .BYTE 153,13,154,14,155,15,0,0
ARROWS  .BYTE 0,24,60,126,24,24,24,24,0,
24,120,56,104,96,192,192
        .BYTE 0,4,6,255,255,6,4,0,0,192,
192,96,104,56,120,24
        .BYTE 0,24,24,24,24,126,60,24,0,
3,3,6,22,28,30,24
        .BYTE 0,32,96,255,255,96,32,0,0,
24,30,28,22,6,3,3
        .END  LETTERS
        .LIB  RECORDKEEPING
SAVEIT  JSR   POSREC
        LDA   HOLD3   ;LEVEL #
        JSR   CHROUT
        LDA   WIND    ;WIND
        JSR   CHROUT
        LDA   HOLD    ;MODE (A/M)
        JSR   CHROUT
        LDA   SHOTRK  ;# SHOTS
        JSR   CHROUT
        LDA   HOLD3   ;LEVEL #
        AND   #127
        CMP   #5
        BCC   SAV010
        LDA   HOLD4   ;ACCURACY
        JSR   CHROUT  ; SCORE
        LDA   PENAL   ;PENALTIES
        JSR   CHROUT
SAV010  LDX   #0      ;FIRST 20
        JSR   SEEINF  ; SHOTS
        LDA   HOLD3
        AND   #127
        BNE   SAV020
        LDX   #48     ;SKILL TEST
        JSR   SAVREC
        LDX   #24
        JMP   SAV070
SAV020  CMP   #5
        BCS   SAV040
        LDX   #144    ;LEV 1-4
        JSR   SAVREC
        DINC  HOLD1   ;2ND RECORD
        JSR   POSREC
        LDA   HOLD3
        CMP   #3
        BCS   SAV030
        LDX   #72
        JMP   SAV070
SAV030  LDX   #16     ;LEV 3-4
        LDY   #144    ;SHOTS 19-20
        JSR   SAV130
        LDX   #1
        JSR   SEEINF
        LDX   #32     ;SHOTS 21-24
        JSR   SAVREC
        LDX   #96
        JMP   SAV070
SAV040  CMP   #10
        BEQ   SAV060
        BIT   HOLD3
        BPL   SAV050
        LDX   #1      ;SHOTS 21-40
        JSR   SEEINF
SAV050  LDX   #160
        JSR   SAVREC
        JMP   SAV090
SAV060  LDX   #160    ;SHOTS 1-20
        JSR   SAVREC
        LDX   #1      ;SHOTS 21-40
        JSR   SEEINF
        DINC  HOLD1   ;2ND RECORD
        JSR   POSREC
        LDX   #160
        JSR   SAVREC
        JMP   SAV090
SAV070  LDY   #0
SAV080  LDA   $192,Y  ;SAVE
        JSR   CHROUT  ; DIAGNOSTICS
        INY
        DEX
        BNE   SAV080
SAV090  JSR   CLRCHN
        RTS
SEEINF  LDA   1
        PHA
        LDA   #54
        STA   1
        LDY   #159
        CPX   #0
        BNE   SAV110
SAV100  LDA   INFO2,Y
        STA   TEMP,Y
        DEY
        CPY   #255
        BNE   SAV100
        JMP   SAV120
SAV110  LDA   INFO2+160,Y
        STA   TEMP,Y
        DEY
        CPY   #255
        BNE   SAV110
SAV120  PLA
        STA   1
        RTS
SAVREC  LDY   #0
SAV130  LDA   TEMP,Y
        JSR   CHROUT
        INY
        DEX
        BNE   SAV130
        RTS
POSREC  JSR   CLRCHN
        LDX   #1
        JSR   CHKOUT
        LDA   #'P     ;"P"
        JSR   CHROUT
        LDA   #104    ;CHR$(104)
        JSR   CHROUT
        LDA   HOLD1   ;CHR$(RECLO)
```

```
        JSR   CHROUT
        LDA   HOLD1+1 ;CHR$(RECHI)
        JSR   CHROUT
        LDA   #0      ;CHR$(0)
        JSR   CHROUT
        JSR   CLRCHN
        LDX   #8
        JSR   CHKOUT
        RTS
        .END  RECORDKEEPING
PORT    LDX   #0
PO010   LDA   LOGON,X
        BEQ   PO020
        JSR   SEND
        INX
        BNE   PO010
PO020   LDX   #0
PO030   LDA   OFFX,X
        JSR   SEND
        INX
        CPX   #4
        BCC   PO030
        LDX   #0
PO040   LDA   LASTSH,X
        JSR   SEND
        INX
        CPX   #5
        BCC   PO040
        LDA   #128
        LDX   LASTSH+8
        BNE   PO050
        LDX   LASTSH+7
        BMI   PO050
        TXA
PO050   STA   STOP
        JSR   SEND
        LDA   SHOTS
        LDX   #128
        AND   #1
        BEQ   PO060
        LDX   #0
PO060   TXA
        CLC
        ADC   STOP
        STA   STOP
        LDA   1
        PHA
        LDA   #54
        STA   1
PO070   LDA   XLPB60,X
        JSR   SEND
        LDA   YLPB60,X
        JSR   SEND
        INX
        CPX   STOP
        BNE   PO070
        LDA   #10
        JSR   SEND
        LDX   #0
PO080   LDA   XLPA60,X
        JSR   SEND
        LDA   YLPA60,X
        JSR   SEND
        INX
        CPX   #10
        BCC   PO080
        PLA
        STA   1
        RTS
LOGON   .BYTE 202,205,194,0
SEND    PHA
SEND10  BIT   56577  ;WAIT FOR
        BVS   SEND10 ; DAR CLEAR
        ROR   A      ;HIGH NYBBLE
        ROR   A
        ROR   A
        ROR   A
        ORA   #128   ;SET DA
        STA   56577
SEND20  BIT   56577  ;WAIT FOR
        BVC   SEND20 ; DAR SET
        LDA   #0     ;CLEAR DA
        STA   56577
SEND30  BIT   56577  ;WAIT FOR
        BVS   SEND30 ;DAR CLEAR
        PLA
        AND   #15    ;LOW NYBBLE
        ORA   #128   ;SET DA
        STA   56577
SEND40  BIT   56577  ;WAIT FOR
        BVC   SEND40 ; DAR SET
        LDA   #0     ;CLEAR DA
        STA   56577
        RTS
        .END
        *=32768
;MLCH0BK1.TXT
        JMP   CTRMOV
        JMP   RANDOM
        JMP   DODATA
        JMP   BEGIN
        JMP   GTINFO
        JMP   REKEEP
        JMP   DESCRP
        JMP   GETXY
        .OPT  NOL
        .LIB  CTRMOV
CTRMOV  JSR   ENABLE
        LDA   #0
        STA   HITSPR
        STA   NUMR20
        STA   NUMR20+1
        STA   NUMR60
        STA   NUMR60+1
        STA   FSTAT
        JSR   $B1BF
        LDA   101
        STA   IRQ
        LDA   100
        STA   IRQ2
CTR010  LDA   #0
        STA   OFFSCR
CTR020  JSR   MOVESP
        JSR   PAUSE
        BCS   CTR030
        JMP   CTR380
CTR030  LDA   SORTAF
        BNE   CTR030
;       JSR   GETXY   ;NO MORE SORT
        JSR   CLOSES
        STA   CURTAR
        TAX
        BNE   CTR040
        JMP   CTR110
CTR040  LDA   SPECIL,X
        STA   SPECIL
        LDA   #0
        STA   WDRIFT+1
        LDA   WIND
        AND   #96
        BEQ   CTR070
        LDA   XDRIFT,X
        ASL   A       ;RANGE*8
        ASL   A
        ASL   A
        STA   XDRIFT
        LDA   WIND    ;GET SPEED
        AND   #7
        CLC           ;ADD TARGET
        ADC   XDRIFT  ; OFFSET
        TAY
        LDA   WIND    ;FULL OR
        AND   #32     ; HALF VALUE?
        BNE   CTR060
        LDA   WTABLE,Y
        LSR   A
        LSR   A
        LSR   A
        LSR   A
        JMP   CTR070
CTR060  LDA   WTABLE,Y
        AND   #15
CTR070  STA   WDRIFT
        BIT   WIND    ;NEGATIVE
        BPL   CTR080  ; DIRECTION?
        DEC   WDRIFT+1
        EOR   #255
        STA   WDRIFT
        DINC  WDRIFT
CTR080  LDA   YDRIFT,X
        STA   TRAJ
        LDA   TIMEFL,X
        BEQ   CTR110
        STA   CURTF
        LDA   INSTK,X
        BEQ   CTR110
        LDX   HOLD1
        LDA   V,X
CTR090  STA   HOLD2
        JSR   MOVESP
        JSR   PAU010
        LDA   OFFSCR
        BEQ   CTR100
        EOR   #255
        AND   V+21
        STA   V+21
        LDA   #0
        STA   OFFSCR
CTR100  LDA   V+21
        AND   HOLD3
        BEQ   CTR110
        LDX   HOLD1
        LDA   V,X
        CMP   HOLD2
        BEQ   CTR090
        DEC   CURTF
        BNE   CTR090
CTR110  LDA   XVAL
        CLC
        ADC   WDRIFT
        STA   XVAL
        STA   V
        LDA   XVAL+1
        ADC   WDRIFT+1
        STA   XVAL+1
        LDA   YVAL
        CLC
        ADC   TRAJ
        STA   YVAL
        STA   V+1
        LDA   V+16
        AND   #254
        LDX   XVAL+1
        BEQ   CTR120
        ORA   #1
CTR120  STA   V+16
        LDA   V+28    ;TURN OFF
        AND   #254    ; MULTICOLOR
        STA   V+28    ; FOR SPR 0
        LDA   #34     ;BULLET
        STA   2040
        LDA   #12     ;MED GRAY
        STA   V+39
        LDA   #0
        STA   SSCOL
        LDA   IRQ2    ;NO CHECK
        AND   #32     ; FOR
        BNE   CTR150  ; COLLISION?
        LDA   V+21    ;TURN ON FOR
        ORA   #1      ; COLLISION
        STA   V+21    ; CHECK
        LDX   #2
CTR130  LDA   IRQ     ;CHECK FOR
        ORA   #32     ; COLLISION
        STA   IRQ
CTR140  LDA   IRQ
        AND   #32
        BNE   CTR140
        DEX
        BNE   CTR130
        LDA   V+21    ;TURN OFF
        AND   #254    ; BULLET
        STA   V+21
CTR150  LDA   #0
        STA   $FC
        LDA   SHOTS
        ASL   A
        ASL   A
        ASL   A
        ROL   $FC
        CLC
        ADC   #<INFO
        STA   $FB
        LDA   $FC
        ADC   #>INFO
        STA   $FC
        LDA   SSCOL
        AND   #254
        TAY
        LDA   #2
        STA   CODE3   ;2^(HIT SPR#)
        LDX   #1      ;HIT SPR#
CTR160  TYA
        STX   CODE2
        AND   CODE3
        CLC
        BNE   CTR170
        INX
        ASL   CODE3
        BCC   CTR160
        LDX   #0
        STX   CODE2
        LDX   CURTAR
CTR170  LDA   TARNUM,X
        BCC   CTR180
        ORA   #128
        JMP   CTR190
CTR180  INC   HITS
CTR190  LDY   #0
        STA   TARNUM
        STA   ($FB),Y ;TARGET ID
        STA   LASTSH,Y
        INY
        LDA   XVAL+1
        LSR   A
        LDA   XVAL
        ROR   A
        STA   ($FB),Y ;BULX/2
        STA   LASTSH,Y
        INY
        LDA   YVAL
        STA   ($FB),Y ;BULY
        STA   LASTSH,Y
        INY
        LDA   SPECIL,X
        STA   SPECIL
        LDA   CENX,X
        STA   CENX
        LDA   CENY,X
        STA   CENY
        TXA
        ASL   A       ;FOR SPR XY
        PHA
        LDA   V+16
CTR200  LSR   A       ;PUT RIGHT
        DEX           ; X IN
        BPL   CTR200  ; CARRY
        PLA
        TAX
        LDA   V,X
        ROR   A       ;RIGHT X
        CLC           ;ADD CENTER X
        ADC   CENX    ; OFFSET
        STA   ($FB),Y ;TARX/2
        STA   LASTSH,Y
        INY
        LDA   V+1,X
        CLC           ;ADD CENTER Y
        ADC   CENY    ; OFFSET
        STA   ($FB),Y ;TARY
        STA   LASTSH,Y
        LDA   IRQ2    ;IS THIS SHOT
        AND   #4      ; A REPEAT?
        BNE   CTR220
```

```
        LDA   $FB      ;STORE IN
        CLC            ; 2ND INFO
        ADC   #<344    ; BUFFER
        STA   $FB      ; (DISK
        LDA   $FC      ; RECORDS
        ADC   #>344    ; ONLY HAVE
        STA   $FC      ; ORIGINAL
        LDA   SHOTS    ; SHOT)
        CLC
        ADC   #1
        STA   SHOTRK
        LDY   #7
CTR210  LDA   LASTSH,Y
        STA   ($FB),Y
        DEY
        BPL   CTR210
CTR220  LDA   XVAL     ;PUT UP CROSS
        SEC
        SBC   #10
        STA   V
        LDA   XVAL+1
        SBC   #0
        TAX
        LDA   V+16
        AND   #254
        CPX   #0
        BEQ   CTR230
        ORA   #1
CTR230  STA   V+16
        LDA   YVAL
        SEC
        SBC   #9
        STA   V+1
        LDA   #42
        STA   2040
        LDA   V+28     ;MULTICOLOR
        ORA   #1       ; MODE ON
        STA   V+28
        LDA   #1       ;WHITE
        STA   V+39
        LDA   IRQ2     ;DON'T
        AND   #16      ; DISPLAY
        BNE   CTR250   ; CROSS?
        LDA   IRQ2
        AND   #8
        BEQ   CTR240
        LDA   LASTSH
        BPL   CTR250
CTR240  LDA   V+21
        ORA   #1
        STA   V+21
CTR250  LDA   SSCOL    ;WAS CLOSEST
        AND   HOLD3    ; TARGET HIT?
        BNE   CTR260   ;YES
        LDX   CURTAR
        BEQ   CTR260
        LDA   SPECIL,X
        AND   #64      ;REVERSE
        BEQ   CTR260   ; DIRECTION?
        LDA   INSTK,X
        EOR   #255
        CLC
        ADC   #1
        STA   INSTK,X
CTR260  LDA   SPECIL
        BPL   CTR265
        JMP   CTR340
CTR265  AND   #32      ;IS THIS
        BEQ   CTR290   ; TARGET MORE
        LDA   #128     ; THAN 1 SPR?
        STA   CODE1
        LDX   #7
CTR270  LDA   V+21
        AND   CODE1
        BEQ   CTR280
        LDA   TARNUM,X
        CMP   TARNUM
        BNE   CTR280
        LDA   CODE1
        ORA   CODE3
        STA   CODE3
CTR280  LSR   CODE1
        DEX
        BNE   CTR270
CTR290  LDA   HITSPR   ;TURN OFF ANY
        EOR   #255     ; TARGET PRE-
        ORA   #1       ; VIOUSLY HIT
        AND   V+21
        STA   V+21
        LDX   LASTSH
        LDA   IRQ2
        AND   #16
        BEQ   CTR300
        CPX   #128
        BCC   CTR310
        JMP   CTR320
CTR300  LDA   IRQ2     ;SPECIAL
        AND   #8       ; CROSS?
        BEQ   CTR330
        CPX   #128     ;MISS?
        BCS   CTR330
CTR310  LDA   IRQ2     ;IF RET AFTER
        AND   #128     ; EACH TARGET
        BNE   CTR320   ; LEAVE IT UP
        LDA   CODE3    ;TURN OFF
        EOR   #255     ; HIT TARGET
        AND   V+21     ; NOW
        STA   V+21
CTR320  LDA   #0
        STA   HITSPR
        JMP   CTR340
CTR330  LDA   CODE3    ;SET UP TO
        ORA   #1       ; TURN OFF
        STA   HITSPR   ; HIT TARGET
        LDA   #0       ;STOP HIT
        LDX   CODE2    ; TARGET
        STA   INSTK,X
        LDA   #30
        STA   CRSDLA
CTR340  BIT   TARNUM
        BVS   CTR370
        LDX   #7       ;CHECK FOR
        LDA   #128     ; PENALTIES
        STA   CODE1
CTR350  LDA   V+21     ;SPRITE MUST
        AND   CODE1    ; BE ON
        BEQ   CTR360
        LDA   TARNUM   ;HIT TARGET
        AND   #127
        CMP   TARNUM,X
        BCC   CTR360   ;MUST BE LESS
        BEQ   CTR360   ; OR EQUAL
        INC   PENAL
        JMP   CTR370
CTR360  LSR   CODE1
        DEX
        BPL   CTR350
CTR370  JSR   ROTATE
        INC   SHOTS
        LDA   IRQ
        AND   #251
        STA   IRQ
        SEC
        JSR   CKSPCL
        BCS   CTR430
        LDA   SHOTS
        CMP   MAXSHT
        BCC   CTR380
        LDA   IRQ
        ORA   #1
        STA   IRQ
        JMP   CTR020
CTR380  LDA   IRQ
        AND   #192
        BEQ   CTR430
        LDA   56321
        CMP   #127
        BNE   CTR390
        LDA   #128     ;BREAK KEY
        STA   FSTAT    ; FLAG
        JMP   CTR430
CTR390  LDA   OFFSCR
        BNE   CTR400
        JMP   CTR020
CTR400  AND   #1       ;WHISTLE
        BEQ   CTR420   ; FLAG
        LDA   V+21     ;SPRITE ON?
        AND   #254
        BEQ   CTR410
        JSR   WHISTL
CTR410  LDA   OFFSCR   ;CLEAR FLAG
        AND   #254
        STA   OFFSCR
CTR420  LDA   OFFSCR
        EOR   #255
        AND   V+21
        STA   V+21
        CLC
        JSR   CKSPCL
        BCS   CTR430
        JMP   CTR010
CTR430  LDA   IRQ
        AND   #15
        STA   IRQ
        MOVE  $FB,TARORD
        LDY   #0
        LDA   ($FB),Y
        CMP   #254
        BNE   CTR440
        DINC  TARORD
CTR440  LDA   #0
        LDY   FSTAT
        JSR   $B391
        RTS
CKSPCL  LDA   IRQ2     ;RETURN WHEN
        BMI   CKSP10   ; NO TARGETS
        AND   #64      ; UP?
        BEQ   CKSP40
        LDA   V+21
        AND   #254
        BNE   CKSP40
CKSP10  BCC   CKSP30
CKSP20  LDA   IRQ
        AND   #192
        BNE   CKSP20
CKSP30  SEC
        RTS
CKSP40  CLC
        RTS
MOVESP  LDY   #14
        LDX   #7
        LDA   #128
        STA   HOLD
MOV010  LDA   V+21
        AND   HOLD
        BEQ   MOV060
        LDA   INSTK,X ;STATIONARY?
        BEQ   MOV060
        DEC   CDLSTK,X ;MOVE IT?
        BNE   MOV060
        LDA   HOLD
        AND   OFFSCR
        BNE   MOV060
        LDA   DLSTK,X
        STA   CDLSTK,X
        LDA   HOLD     ;CHECK
        AND   V+16     ; RIGHT X
        BEQ   MOV020
        LDA   #1
MOV020  STA   HOLDX
        LDA   INSTK,X ;ADD
        STA   HOLDY    ; INCREMENT
        CLC
        ADC   V,Y
        STA   V,Y
        LDA   #0
        BIT   HOLDY    ;INCREMENT<0?
        BPL   MOV030
        LDA   #255
MOV030  ADC   HOLDX
        BEQ   MOV040
        LDA   HOLD
        ORA   V+16
        STA   V+16
        LDA   V,Y
        CMP   #55
        BCS   MOV050
        JMP   MOV060
MOV040  LDA   HOLD
        EOR   #255
        AND   V+16
        STA   V+16
        LDA   V,Y
        CMP   #10
        BCS   MOV060
MOV050  LDA   HOLD
        ORA   OFFSCR
        STA   OFFSCR
MOV060  DEY
        DEY
        LSR   HOLD
        DEX
        BNE   MOV010
        RTS
PAUSE   LDA   IRQ      ;TRIG PULL
        AND   #4
        BEQ   PAU010
        SEC
        RTS
PAU010  LDA   #$66
PAU020  SEC
        SBC   #1
        BNE   PAU020
        CLC
        RTS
CLOSES  LDA   #255
        STA   HOLDA    ;LOW DISTANCE
        LDA   #128
        STA   CODE1    ;SPRITE CODE
        STA   TARNUM   ; FOR MISS
        LDA   XVAL+1
        LSR   A
        LDA   XVAL
        ROR   A
        STA   HOLDX    ;BULLET X
        LDA   #0
        STA   HOLD1
        STA   HOLD3
        LDX   #14
CLO010  LDA   V+21     ;SPRITE ON?
        AND   CODE1
        BEQ   CLO050
        LDA   V+1,X
        SEC
        SBC   YVAL
        BCS   CLO020
        EOR   #255
        CLC
        ADC   #1
CLO020  STA   HOLDY    ;TARY-BULY
        LDA   V+16
        AND   CODE1
        CLC
        BEQ   CLO030
        SEC
CLO030  LDA   V,X
        ROR   A
        SEC
        SBC   HOLDX
        BCS   CLO040
        EOR   #255
        CLC
        ADC   #1
CLO040  CLC
        ADC   HOLDY    ;XDIF+YDIF
        ROR   A        ;(XDIF+YDIF)/2
        CMP   HOLDA
        BCS   CLO050
        STA   HOLDA
        STX   HOLD1
        LDA   CODE1
        STA   HOLD3
CLO050  LSR   CODE1
        DEX
```

```
        DEX
        BNE   CLO010
        LDA   HOLD1
        LSR   A
        RTS
WTABLE .BYTE 0,16,17,33,49,66,66,0
        .BYTE 0,17,33,50,66,83,115,0
        .BYTE 0,17,49,66,99,116,148,0
        .BYTE 0,33,66,83,116,148,181,0
        .BYTE 0,33,66,99,148,181,215,0
        .BYTE 0,49,83,132,181,231,248,0
        .END  CTRMOV
        .OPT  NOL
        .LIB  DATAITEMS
;DATAITEMS FOR BRM PROGRAM
DODATA JSR   WHCDAT
        LDA   #<DITEMS
        CLC
        ADC   STOP
        STA   STOP
        LDA   #>DITEMS
        ADC   STOP+1
        STA   STOP+1
        ADDR  $FD,52480
        LDY   #0
DOD010 LDA   ($FB),Y
        STA   ($FD),Y
        DINC  $FB
        DINC  $FD
        LDA   $FB
        CMP   STOP
        BCC   DOD010
        LDA   $FC
        CMP   STOP+1
        BCC   DOD010
        RTS
;PROCEDURE RANDOM
;POKE STOP WITH THE STARTING LOC
; IN STACK (X=START-16325)
; AND 16325-XXXXX WITH THE DATA
; ITEMS.  POKE HOLD1,HOLD2 WITH
; ADDRESS OFFSET OF DATA TO BE
; MOVED.
RANDOM LDX   STOP
        ADDR  $FD,52480
RAN010 JSR   WHCDAT
        LDA   16325,X
        BMI   RAN020
        JSR   SEARCH
        INX
        BNE   RAN010
RAN020 LDY   #0
        LDA   #255
        STA   ($FD),Y
        RTS
SEARCH STA   $A3
        STX   $A4
        LDY   #0
        LDX   #0
        CMP   #0
        BEQ   RAN060
RAN030 LDA   ($FB),Y
        DINC  $FB
        TAX
RAN040 LDA   $FB
        CLC
        ADC   #14
        STA   $FB
        LDA   $FC
        ADC   #0
        STA   $FC
        DEX
        BNE   RAN040
        LDA   ($FB),Y
        BPL   RAN050
        DINC  $FB
RAN050 DEC   $A3
        BNE   RAN030
RAN060 LDA   ($FB),Y
        STA   $A3
        STA   ($FD),Y
        INY
RAN070 LDX   #14
RAN080 LDA   ($FB),Y
        STA   ($FD),Y
        INY
        DEX
        BNE   RAN080
        DEC   $A3
        BNE   RAN070
        LDA   ($FB),Y
        BPL   RAN090
        STA   ($FD),Y
        INY
RAN090 TYA
        CLC
        ADC   $FD
        STA   $FD
        LDA   $FE
        ADC   #0
        STA   $FE
        LDX   $A4
        RTS
WHCDAT LDA   #<DITEMS
        CLC
        ADC   HOLD1
        STA   $FB
        LDA   #>DITEMS
        ADC   HOLD1+1
        STA   $FC
        RTS
;ZERO DATA (0-16)
DITEMS .BYTE 1,0,150,1,37,12,255,0,0,0,
4,0,0,6,15,254,255
;EXTENDED PRETEST DATA (17-113)
        .BYTE 1,141,159,  1, 41,204, 10,  0,
 0,  0,  1,254,  0, 12, 36,254
        .BYTE 1,107,150,  4, 37, 12, 16,  0,
 0,  0,  4,  0,  0,  6, 15,254
        .BYTE 1, 93,140,  5, 38, 12, 18,  0,
 0,  0,  5,  1,  0,  6, 16,254
        .BYTE 1,193,159,  1, 41,204, 10,  0,
 0,  0,  1,254,  0, 12, 36,254
        .BYTE 1, 75,150,  4, 37, 12, 16,  0,
 0,  0,  4,  0,  0,  6, 15,254
        .BYTE 1,149,140,  5, 38, 12, 18,  0,
 0,  0,  5,  1,  0,  6, 16,254,255
;LEVELS 1-4 (114-224)
        .BYTE 2,  0,169,  1, 39,204,  8,  0,
 0,  0,  0,255, 32, 24, 32
        .BYTE 0,169,  1, 40,204,  8,  0,  0,
 0,  0,255,  32,  0, 32,254
        .BYTE 1,  0,159,  2, 41,204, 11,  0,
 0,  0,  1,254,  0, 12, 36,254
        .BYTE 1,  0,170,  3, 35, 12, 13,  0,
 0,  0,  2,254,  0,  6, 12,254
        .BYTE 1,  0,160,  4, 36, 12, 16,  0,
 0,  0,  3,255,  0,  6, 14,254
        .BYTE 1,  0,150,  5, 37, 12, 19,  0,
 0,  0,  4,  0,  0,  6, 15,254
        .BYTE 1,  0,140,  6, 38, 12, 21,  0,
 0,  0,  5,  1,  0,  6, 16,254,255
;LEVELS 5-9 (225-912)
        .BYTE 1,243,152,  2, 41,204,  8,  0,
 0,  0,  1,254,  0, 12, 36,254
        .BYTE 1,130,160,  3, 35, 12, 10,  0,
 0,  0,  2,254,  0,  6, 12,254
        .BYTE 2, 48,165,  1, 39,204,  6,  0,
 0,  0,  0,255, 32, 24, 32
        .BYTE   96,165,  1, 40,204,  6,  0,
 0,  0,  0,255, 48,  0, 32,254
        .BYTE 1,130,160,  3, 35, 12, 10,  0,
 0,  0,  2,254,  0,  6, 12,254
        .BYTE 1,230,115,  6, 38, 12, 16,  0,
 0,  0,  5,  1,  0,  6, 16,254
        .BYTE 1,243,152,  2, 41,204,  8,  0,
 0,  0,  1,254,  0, 12, 36,254
        .BYTE 1,130,160,  3, 35, 12, 10,  0,
 0,  0,  2,254,  0,  6, 12,254
        .BYTE 2, 48,165,  1, 39,204,  6,  0,
 0,  0,  0,255, 32, 24, 32
        .BYTE   96,165,  1, 40,204,  6,  0,
 0,  0,  0,255, 48,  0, 32,254
        .BYTE 1,130,160,  3, 35, 12, 10,  0,
 0,  0,  2,254,  0,  6, 12,254
        .BYTE 1,210,145,  4, 36, 12, 12,  0,
 0,  0,  3,255,  0,  6, 14,254
        .BYTE 1,194,130,  5, 37, 12, 14,  0,
 0,  0,  4,  0,  0,  6, 15,254
        .BYTE 2,243,152,  2, 41,204, 18,  0,
 0,  0,  1,254,  0, 12,36
        .BYTE   194,130,  5, 37, 12, 18,  0,
 0,  0,  4,  0,  0,  6, 15,254
        .BYTE 1,210,145,  4, 36, 12, 12,  0,
 0,  0,  3,255,  0,  6, 14,254
        .BYTE 1,194,130,  5, 37, 12, 14,  0,
 0,  0,  4,  0,  0,  6, 15,254
        .BYTE 2,243,152,  2, 41,204, 18,  0,
 0,  0,  1,254,  0, 12, 36
        .BYTE   210,145,  4, 36, 12, 18,  0,
 0,  0,  2,254,  0,  6, 12,254
        .BYTE 1,149,140,  5, 38, 12, 18,  0,
 0,  0,  5,  1,  0,  6, 16,254,255
;LEVELS 10-14 (913-1585)
;SCREEN 1
        .BYTE 1,147,115,  6, 38, 12,  8,  0,
 0,  0,  5,  1,  1,  6, 16,136,140
;SCREEN 2
        .BYTE 1,233,115,  6, 38, 12, 14,  0,
 0,  0,  5,  1,  1,  6, 16,  0
        .BYTE 1,131,130,  5, 37, 12, 12,  0,
 0,  0,  4,  0,  1,  6, 15,  0
        .BYTE 1, 79,115,  6, 38, 12, 10,  0,
 0,  0,  5,  1,  1,  6, 16,138,138
;SCREEN 3
        .BYTE 2,171,130,  5, 37, 12,  8,  0,
 0,  0,  4,  0,  2,  6, 15
        .BYTE    65,145,  4, 36, 12,  8,  0,
 0,  0,  3,255,  2,  6, 14,136,134
;SCREEN 4
        .BYTE 2, 13,130,  5, 37, 44, 16,  0,
 0,  0,  4,  0,  2,  6, 15
        .BYTE    59,145,  4, 36, 12, 16,  0,
 0,  0,  3,255,  2,  6, 14,131
        .BYTE 1,211,160,  3, 35, 12, 12,  0,
 0,  0,  2,254,  1,  6, 12,131
        .BYTE 1,103,145,  4, 36, 12,  8,  0,
 0,  0,  3,255,  1,  6, 14,136,140
;SCREEN 5
        .BYTE 1,253,160,  3, 35, 12,  8,  0,
 0,  0,  2,254,  1,  6, 12,  0
        .BYTE 1, 43,152,  2, 41,204,  6,  0,
 0,  0,  1,254,  1, 12, 36,134,134
;SCREEN 6
        .BYTE 1,181,160,  3, 35, 12, 12,  0,
 0,  0,  2,254,  1,  6, 12,131
        .BYTE 1,251,152,  2, 41,204,  8,  0,
 0,  0,  1,254,  1, 12, 36,131
        .BYTE 2, 41,165,  1, 39,204,  4,  0,
 0,  0,  0,255, 33, 24, 32
        .BYTE    89,165,  1, 40,204,  4,  0,
 0,  0,  0,255, 49,  0, 32,132,130
        .BYTE 3, 47,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  3, 12, 36
        .BYTE   189,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  3, 12, 36
        .BYTE    19,160,  3, 35, 44, 20,  0,
 0,  0,  2,254,  3,  6, 12,140
        .BYTE 3,107,152,  2, 41,204,  8,  0,
 0,  0,  1,254,  2, 12, 36
        .BYTE   207,165,  1, 39,204,  8,  0,
 0,  0,  0,255, 34, 24, 32
        .BYTE   255,165,  1, 40,204,  8,  0,
 0,  0,  0,255, 50,  0, 32,136,130
        .BYTE 5, 29,165,  1, 39,204, 20,  0,
 0,  0,  0,255, 36, 24, 32
        .BYTE    77,165,  1, 40,204, 20,  0,
 0,  0,  0,255, 52,  0, 32
        .BYTE    91,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  4, 12, 36
        .BYTE   153,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  4, 12, 36
        .BYTE   255,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  4, 12, 36,144
        .BYTE 1,221,160,  3, 35, 12,  4,  0,
 0,  0,  2,254,  1,  6, 12,132,134
;SCREEN 9
        .BYTE 1,205,152,  2, 41,204, 12,  0,
 0,  0,  1,254,  1, 12, 36,  0
        .BYTE 1,101,160,  3, 35, 12, 10,  0,
 0,  0,  2,254,  1,  6, 12,  0
        .BYTE 1,205,160,  3, 35, 12,  8,  0,
 0,  0,  2,254,  1,  6, 12,136,140
;SCREEN 10
        .BYTE 1,111,160,  3, 35, 12,  8,  0,
 0,  0,  2,254,  1,  6, 12,  0
        .BYTE 1,255,145,  4, 36, 12,  6,  0,
 0,  0,  3,255,  1,  6, 14,134,134
;SCREEN 11
        .BYTE 3,241,160,  3, 35, 12, 16,  0,
 0,  0,  2,254,  3,  6, 12
        .BYTE    99,145,  4, 36, 12, 16,  0,
 0,  0,  3,255,  3,  6, 14
        .BYTE   215,145,  4, 36, 12, 16,  0,
 0,  0,  3,255,  3,  6, 14,132
        .BYTE 1,151,130,  5, 37, 12, 12,  0,
 0,  0,  4,  0,  1,  6, 15,140,138
;SCREEN 12
        .BYTE 1,101,130,  5, 37, 12,  8,  0,
 0,  0,  4,  0,  1,  6, 15,132
        .BYTE 1,239,130,  5, 37, 12,  4,  0,
 0,  0,  4,  0,  1,  6, 15,132,136
;SCREEN 13
        .BYTE 2,147,115,  6, 38, 12, 14,  0,
 0,  0,  1,254,  0, 12, 36
        .BYTE   130,160,  3, 35, 12, 16,  0,
 0,  0,  2,254,  0,  6, 12,254
        .BYTE 1,130,160,  3, 35, 12, 16,  0,
 0,  0,  2,254,  0,  6, 12,254
        .BYTE 2,243,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  0, 12, 36
        .BYTE   210,145,  4, 36, 12, 20,  0,
 0,  0,  3,255,  0,  6, 14,254
        .BYTE 2,130,160,  3, 35, 12, 24,  0,
 0,  0,  2,254,  0,  6, 12
        .BYTE   210,145,  4, 36, 12, 18,  0,
 0,  0,  3,255,  0,  6, 14,254
;ATTACK/RETREAT 1 (913-1586)
;SCREEN 1
        .BYTE 1,147,115,  6, 38, 12,  8,  0,
 0,  0,  5,  1,  1,  6, 16,136,140
;SCREEN 2
        .BYTE 1,233,115,  6, 38, 12, 14,  0,
 0,  0,  5,  1,  1,  6, 16,  0
        .BYTE 1,131,130,  5, 37, 12, 12,  0,
 0,  0,  4,  0,  1,  6, 15,  0
        .BYTE 1, 79,115,  6, 38, 12, 10,  0,
 0,  0,  5,  1,  1,  6, 16,138,138
;SCREEN 3
        .BYTE 2,171,130,  5, 37, 12,  8,  0,
 0,  0,  4,  0,  2,  6, 15
        .BYTE    65,145,  4, 36, 12,  8,  0,
 0,  0,  3,255,  2,  6, 14,136,134
;SCREEN 4
        .BYTE 2, 13,130,  5, 37, 44, 16,  0,
 0,  0,  4,  0,  2,  6, 15
        .BYTE    59,145,  4, 36, 12, 16,  0,
 0,  0,  3,255,  2,  6, 14,131
        .BYTE 1,211,160,  3, 35, 12, 12,  0,
 0,  0,  2,254,  1,  6, 12,131
        .BYTE 1,103,145,  4, 36, 12,  8,  0,
 0,  0,  3,255,  1,  6, 14,136,140
;SCREEN 5
        .BYTE 1,253,160,  3, 35, 12,  8,  0,
 0,  0,  2,254,  1,  6, 12,  0
        .BYTE 1, 43,152,  2, 41,204,  6,  0,
 0,  0,  1,254,  1, 12, 36,134,134
;SCREEN 6
        .BYTE 1,181,160,  3, 35, 12, 12,  0,
 0,  0,  2,254,  1,  6, 12,131
        .BYTE 1,251,152,  2, 41,204,  8,  0,
 0,  0,  1,254,  1, 12, 36,131
        .BYTE 2, 41,165,  1, 39,204,  4,  0,
 0,  0,  0,255, 33, 24, 32
        .BYTE    89,165,  1, 40,204,  4,  0,
 0,  0,  0,255, 49,  0, 32,132,130
        .BYTE 3, 47,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  3, 12, 36
        .BYTE   189,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  3, 12, 36
        .BYTE    19,160,  3, 35, 44, 20,  0,
 0,  0,  2,254,  3,  6, 12,140
        .BYTE 3,107,152,  2, 41,204,  8,  0,
 0,  0,  1,254,  2, 12, 36
        .BYTE   207,165,  1, 39,204,  8,  0,
 0,  0,  0,255, 34, 24, 32
        .BYTE   255,165,  1, 40,204,  8,  0,
 0,  0,  0,255, 50,  0, 32,136,130
        .BYTE 5, 29,165,  1, 39,204, 20,  0,
 0,  0,  0,255, 36, 24, 32
        .BYTE    77,165,  1, 40,204, 20,  0,
 0,  0,  0,255, 52,  0, 32
        .BYTE    91,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  4, 12, 36
        .BYTE   153,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  4, 12, 36
        .BYTE   255,152,  2, 41,204, 20,  0,
 0,  0,  1,254,  4, 12, 36,144
        .BYTE 1,221,160,  3, 35, 12,  4,  0,
 0,  0,  2,254,  1,  6, 12,132,134
;SCREEN 9
        .BYTE 1,205,152,  2, 41,204, 12,  0,
 0,  0,  1,254,  1, 12, 36,  0
        .BYTE 1,101,160,  3, 35, 12, 10,  0,
 0,  0,  2,254,  1,  6, 12,  0
        .BYTE 1,205,160,  3, 35, 12,  8,  0,
 0,  0,  2,254,  1,  6, 12,136,140
;SCREEN 10
        .BYTE 1,111,160,  3, 35, 12,  8,  0,
 0,  0,  2,254,  1,  6, 12,  0
        .BYTE 1,255,145,  4, 36, 12,  6,  0,
 0,  0,  3,255,  1,  6, 14,134,134
;SCREEN 11
        .BYTE 3,241,160,  3, 35, 12, 16,  0,
 0,  0,  2,254,  3,  6, 12
        .BYTE    99,145,  4, 36, 12, 16,  0,
 0,  0,  3,255,  3,  6, 14
        .BYTE   215,145,  4, 36, 12, 16,  0,
 0,  0,  3,255,  3,  6, 14,132
        .BYTE 1,151,130,  5, 37, 12, 12,  0,
 0,  0,  4,  0,  1,  6, 15,140,138
;SCREEN 12
        .BYTE 1,101,130,  5, 37, 12,  8,  0,
 0,  0,  4,  0,  1,  6, 15,132
        .BYTE 1,239,130,  5, 37, 12,  4,  0,
 0,  0,  4,  0,  1,  6, 15,132,136
;SCREEN 13
        .BYTE 2,147,115,  6, 38, 12, 14,  0,
```

```
 0,  0,  5,  1,  2,  6, 16
   .BYTE    63,130,  5, 37, 12, 14,  0,
 0,  0,  4,  0,  2,  6, 15,  0
   .BYTE 1,237,115,  6, 38, 12, 12,  0,
 0,  0,  5,  1,  1,  6, 16,140,140
;SCREEN 14
   .BYTE 1,105,115,  6, 38, 12,  8,  0,
 0,  0,  5,  1,  1,  6, 16,138,255
       .END  DATAITEMS
       .LIB  BEGIN
BEGIN  LDA   #0
       STA   53280
       STA   53281
       LDA   #21
       STA   53272
       LDA   #27
       STA   53265
       PRINT MSG1
       LDA   #'2
       JSR   CKANS
       STA   512
       STA   PINFO
       LDA   #58
       JSR   CKANS
       STA   513
       STA   PINFO+1
       LDY   #0
       JSR   DTOA
       BEQ   BEGIN
       CMP   #13
       BCS   BEGIN
       LDY   #5
       JSR   RARROW
       LDA   #'/
       STA   PINFO+2
       LDA   #'4
       JSR   CKANS
       STA   512
       STA   PINFO+3
       LDA   #58
       JSR   CKANS
       STA   513
       STA   PINFO+4
       LDY   #0
       JSR   DTOA
       BEQ   BEGIN
       CMP   #32
       BCS   BEGIN
       LDY   #6
       JSR   RARROW
       LDA   #'/
       STA   PINFO+5
       LDA   #58
       JSR   CKANS
       STA   PINFO+6
       LDA   #58
       JSR   CKANS
       STA   PINFO+7
       LDA   #13
       STA   PINFO+8
       JSR   $FFD2   ;CHROUT
       JSR   $FFD2   ;CHROUT
       JSR   $FFD2   ;CHROUT
BGL10  PRINT MSG2
       LDY   #9
       JSR   GETANS  ;A-E OK
       CPY   #11
       BNE   BGL15
       LDA   521
       CMP   #'A
       BCC   BGL15   ;<A
       CMP   #'F
       BCC   BGL20   ;<F
BGL15  JSR   CLRLNE
       JMP   BGL10
BGL20  PRINT MSG3
       LDY   #11
       JSR   GETANS
       CPY   #14
       BEQ   BGL25
       CPY   #13
       BNE   BGL30
       LDA   PINFO+11
       CMP   #'1     ;<'1'
       BCC   BGL30
       CMP   #58     ;>'9'
       BCS   BGL30
       STA   PINFO+12
       LDA   #'0
       STA   PINFO+11
       LDA   #13
       STA   PINFO+13
       JMP   BGL50
BGL25  LDY   #11
       JSR   DTOA
       BCC   BGL40
BGL30  JSR   CLRLNE
       JMP   BGL20
BGL40  BEQ   BGL30
       CMP   #11
       BCS   BGL30
BGL50  PRINT MSG4
       LDY   #14
       JSR   GETANS
       CPY   #16
       BNE   BGL60
       LDA   526
       CMP   #'1
       BCC   BGL60
       CMP   #'3
       BCC   BGL70
BGL60  JSR   CLRLNE
       JMP   BGL50
BGL70  LDX   #23
       LDY   #9
       CLC
       JSR   $FFF0   ;SET CURSOR
       PRINT MSG5
BGL80  JSR   $FFE4   ;GETIN
       CMP   #0
       BEQ   BGL80
       CMP   #'Y
       BEQ   BGL90
       CMP   #'N
       BNE   BGL80
       JMP   BEGIN
BGL90  RTS
GETANS JSR   $FFCF    ;CHRIN
       STA   512,Y
       STA   PINFO,Y
       INY
       CPY   #20
       BEQ   GETL10
       CMP   #13
       BNE   GETANS
GETL10 LDA   #13
       JSR   $FFD2    ;CHROUT
       RTS
CLRLNE LDA   #145     ;UP ARROW
       JSR   $FFD2    ;CHROUT
       LDY   #38
CLRL10 LDA   #32      ;SPACE
       JSR   $FFD2    ;CHROUT
       DEY
       BNE   CLRL10
       LDA   #13
       JSR   $FFD2    ;CHROUT
       LDA   #145     ;UP ARROW
       JSR   $FFD2    ;CHROUT
       RTS
DTOA   LDA   513,Y
       SEC
       SBC   #48
       CMP   #10
       BCS   DTOA30
       LDA   512,Y
       SEC
       SBC   #48
       CMP   #10
       BCS   DTOA30
DTOA10 STA   $FF
       LDX   #9
DTOA20 CLC
       ADC   $FF
       BCS   DTOA30
       DEX
       BNE   DTOA20
       INY
       CLC
       ADC   512,Y
       SEC
       SBC   #48
       CLC
DTOA30 RTS
CKANS  STA   MAXVAL
CKANS1 JSR   $FFE4   ;GETIN
       CMP   #0
       BEQ   CKANS1
       CMP   #'0
       BCC   CKANS1
       CMP   MAXVAL
       BCS   CKANS1
       JSR   $FFD2   ;CHROUT
       RTS
RARROW LDA   #29
       JSR   $FFD2   ;CHROUT
       DEY
       BNE   RARROW
       RTS
GTINFO LDA   #16
       LDY   #0
       STA   (71),Y
       INY
       LDA   #<PINFO
       STA   (71),Y
       INY
       LDA   #>PINFO
       STA   (71),Y
       RTS
MSG1   .BYTE  147,142,13,5,'WELCOME TO
MACS',13,13
       .BYTE 'ENTER DATE (USE LEADING
ZEROS): ',13,13
       .BYTE '     --      --    19--',13
       .BYTE '   MONTH   DAY   YEAR'
       .BYTE 13,145,145,'    ',0
MSG2   .BYTE 'COMPANY (A-E): ',0
MSG3   .BYTE 'BATTALION (1-10): ',0
MSG4   .BYTE 'BRIGADE (1-2): ',0
MSG5   .BYTE 'IS THIS CORRECT (Y/N)?',
13,0
       .END  BEGIN
       .LIB  REKEEP
REKEEP LDA   #0
       JSR   SETNAM
       LDA   #1
       LDX   #8
       LDY   #15
       JSR   SETLFS
       JSR   OPEN
       BCS   NODISK
       LDX   #1
       JSR   CHKOUT
       BCC   DISKON
NODISK LDA   #1
       JSR   CLOSE
       JSR   CLRCHN
       LDA   #0
       TAY
       JSR   FLOAT
       JMP   REC5
DISKON LDA   #1
       JSR   CLOSE
       JSR   CLRCHN
       ADDR  $A3,S1
       LDA   #S2-S1
       STA   LENSTR
       JSR   LETML
REC1   ADDR  $A3,S2
       LDA   #S3-S2
       STA   LENSTR
       JSR   LETML
       LDX   #'Y
       JSR   GKEY
       BCS   REC2
       ADDR  $A3,S3
       LDA   #S4-S3
       STA   LENSTR
       JSR   LETML
       LDX   #'N
       JSR   GKEY
       BCC   REC1
REC2   LDA   #S5-S4
       STA   LENSTR
       LDA   $FD
       SEC
       SBC   #'N
       BEQ   REC3
       LDY   #1
       LDA   #0
       JSR   FLOAT
       ADDR  $A3,S4
       JMP   REC4
REC3   TAY
       JSR   FLOAT
       ADDR  $A3,S5
REC4   JSR   LETML
REC5   LDY   #255
       JSR   WAIT
       LDY   #255
       JSR   WAIT
       LDY   #255
       JSR   WAIT
       JSR   NOTTRG
       RTS
GKEY   LDA   #$FF
       STA   $A3
       LDA   #$80
       STA   $A4
       STX   $FD
GKEY1  JSR   GETIN
       CMP   #0
       BNE   GKEY2
       DEC   $A3
       BNE   GKEY5
       DEC   $A4
       BNE   GKEY5
       CLC
       RTS
GKEY2  CMP   #'Y
       BEQ   GKEY4
       CMP   #'N
       BNE   GKEY1
GKEY4  STA   $FD
       SEC
       RTS
GKEY5  LDA   56321
       CMP   #247
       BNE   GKEY1
       SEC
       RTS
       .OPT  NOL
S1     .DBYTE $1311,$8511,$1111,$1111,
$1111,$1120,$2020,$85C4
       .BYTE 'O YOU WISH TO KEEP
RECORDS?',17,17,13
S2     .DBYTE $2085,$2020,$2020,$05D9,
$C5D3,$2020,$2020
       .DBYTE $2020,$2020,$9785,$9D9D,
$4E4F,$910D
S3     .DBYTE $2085,$2020,$2020,$9720,
$2020,$9D9D,$9D85,$5945
       .DBYTE $5320,$8520,$2020,$2020,
$2020,$05CE,$CF85,$910D
S4     .DBYTE $0585,$2020,$2020,$2020,
$2020,$D9C5,$D320
       .BYTE '    '
S5     .DBYTE $0585,$2020,$2020,$2020,
$2020,$20CE,$CF20
       .BYTE '    '
       .OPT  LIST
       .END  REKEEP
       .OPT  NOL
       .LIB  DESCRIPTIONS
```

```
DESCRP ADDR  $FB,DESC
       ADDR  $A3,16192
       LDX   HOLD1
       BEQ   DES050
       LDY   #0
DES010 DEX
       BEQ   DES020
       LDA   ($FB),Y
       CLC
       ADC   #1
       CLC
       ADC   $FB
       STA   $FB
       LDA   $FC
       ADC   #0
       STA   $FC
       JMP   DES010
DES020 LDA   #40
       SEC
       SBC   ($FB),Y
       LSR   A
       TAY
       LDX   #0
DES030 LDA   #32
       STA   16192,X
       INX
       DEY
       BNE   DES030
       LDA   ($FB),Y
       STA   HOLD1
       INY
DES040 LDA   ($FB),Y
       STA   16192,X
       INX
       INY
       DEC   HOLD1
       BNE   DES040
       STX   LENSTR
       JSR   LETML
       RTS
DES050 INC   HOLD1
       LDA   HOLD1
DES060 CMP   $FD     ;START
       BNE   DES070
       LDX   #14     ;LIGHT BLUE
       STX   646
DES070 CMP   $FE     ;FINAL
       BCC   DES080
       BEQ   DES080
       LDX   #0      ;BLACK
       STX   646
DES080 LDX   #32
       STX   16192
       STX   16195
       ORA   #48
       STA   16193
       CMP   #58
       BNE   DES090
       LDA   #'1
       STA   16192
       LDA   #'0
       STA   16193
DES090 LDA   #':
       STA   16194
       LDY   #0
       LDA   ($FB),Y
       PHA
       TAY
DES100 LDA   ($FB),Y
       STA   16195,Y
       DEY
       BNE   DES100
       PLA
       TAY
       DINC  $FB
       CLC
       ADC   $FB
       PHA
       LDA   $FC
       ADC   #0
       PHA
       LDA   $FD
       PHA
       LDA   $FE
       PHA
       LDA   #13
       STA   16196,Y
       INY
       INY
       INY
       INY
       INY
       STY   LENSTR
       JSR   LETML
       PLA
       STA   $FE
       PLA
       STA   $FD
       PLA
       STA   $FC
       PLA
       STA   $FB
       INC   HOLD1
       LDA   HOLD1
       CMP   #11
       BNE   DES060
       RTS
DESC   .BYTE 31,201,'NTRODUCTION/',211,
'UPPORTED ',208,'OSITION'
       .BYTE 33,201,'NTRODUCTION/',213,
'NSUPPORTED ',208,'OSITION'
       .BYTE 32,212,'IMED ',212,
'ARGETS/',211,'UPPORTED ',208,'OSITION'
       .BYTE 34,212,'IMED ',212,
'ARGETS/',213,'NSUPPORTED ',208,
'OSITION'
       .BYTE 22,208,'RACTICE ',210,
'ECORD ',198,'IRE ',201
       .BYTE 23,208,'RACTICE ',210,
'ECORD ',198,'IRE ',201,201
       .BYTE 11,210,'ECORD ',198,'IRE'
       .BYTE 19,210,'APID ',210,'ECORD
',198,'IRE ',201
       .BYTE 20,210,'APID ',210,'ECORD
',198,'IRE ',201,201
       .BYTE 11,195,'OMBAT ',198,'IRE'
       .END  DESCRIPTIONS
       .LIB  GETXYSORT
;GETXY AND SORT
;PROCEDURE GETXY
;DETERMINES BULLET STRIKE
;B:NONE
;C:SYS GETXY
;A:PUTS THE BULLET STRIKE (X AND Y
;  VALUES) IN XVAL AND YVAL.
GETXY  LDA   STSIZE
       STA   NSRT
       JSR   SORT
       LDA   STSIZE  ;FIND THE
       LSR   A       ; MEDIAN OF
       TAX           ; THE SRTBUF
       LDA   SRTBUF,X
       ASL   A       ;*2
       STA   XVAL    ;SET UP FOR
       LDA   #0      ; ADDITION OF
       BCC   GXY10   ; OFFSET X
       LDA   #1      ;BULLET RIGHT
GXY10  STA   XVAL+1
       DADD  XVAL,OFFX
       LDA   STSIZE  ;FIND MEDIAN
       LSR   A       ; OF THE
       CLC           ; Y HALF OF
       ADC   STSIZE  ; THE SRTBUF
       TAX
       LDA   SRTBUF,X
       STA   YVAL    ;Y MEDIAN
       LDA   #0      ;SET UP FOR
       STA   YVAL+1  ;OFFSET ADD Y
       DADD  YVAL,OFFY
       RTS
;
;PROCEDURE SORT
;SORTS NSRT ELEMENTS IN SRTBUF
;B:POKE NSRT,(# OF ELEMENTS)
;C:SYS SORT
;A:ELEMENTS IN SRTBUF ARE SORTED
SORT   ADDR  $FD,SRTBUF
       LDY   NSRT
       BEQ   SRT30
       DEY
       BEQ   SRT30
       STY   HOLD
       LDY   #0
SRT10  LDA   ($FD),Y
       INY
       CMP   ($FD),Y
       BCC   SRT20   ;DON'T SORT
       TAX           ;SWAP X VALS
       LDA   ($FD),Y
       DEY
       STA   ($FD),Y
       INY
       TXA
       STA   ($FD),Y
       TYA           ;SWAP CORRE-
       CLC           ; SPONDING Y
       ADC   NSRT    ; VALUES
       TAY
       DEY
       LDA   ($FD),Y
       TAX
       INY
       LDA   ($FD),Y
       DEY
       STA   ($FD),Y
       INY
       TXA
       STA   ($FD),Y
       TYA
       SEC
       SBC   NSRT
       TAY
SRT20  CPY   HOLD
       BNE   SRT10
       LDY   #0
       DEC   HOLD
       BNE   SRT10
SRT30  RTS
       .END  GETXYSORT
       .OPT  LIST
       .END
       *=49152
; STARTUP.TXT FOR M16 BRM CART
; STARTUP/SUBS (1646 BYTES)
; 49152-50797     (0-0) 32768-34413
; PRINT FILES (4785 BYTES)
; 16384-21168     (0-0) 36141-40925
; MLPRINTFILES (34 BYTES)
; 3400-3433       (0-0) 40926-40959
; MLCH0BK1 (4793 BYTES)
; NOMOVE          (0-1) 32768-37560
; BRM.SCENARIO.3 (2918 BYTES)
; INDIRECT COPY   (0-1) 38042-40959
; OPENING.SCENARIO (4165 BYTES)
; INDIRECT COPY   (0-2) 32768-36932
; SIGHT PICTURE (2856 BYTES)
; 16384-19239     (0-2) 36933-39788
; CALIBRATION (1171 BYTES)
; 16384-17554     (0-2) 39789-40959
; BRM.3400 (4782 BYTES)
; 3400-8181       (0-3) 32768-37549
; BRM.SCENARIO.4 (2992 BYTES)
; INDIRECT COPY   (0-3) 37968-40959
; BRM.BAS (1) (8192 BYTES)
; 16384-24575     (1-0) 32768-40959
; BRM.BAS (2) (8192 BYTES)
; 24576-32767     (1-1) 32768-40959
; BRM.BAS (3) (2733 BYTES)
; 32768-35500     (1-2) 32768-35500
; ZERO.BAS (1) (2834 BYTES)
; 16384-19217     (1-2) 38126-40959
; ZERO.BAS (2) (8192 BYTES)
; 19218-27409     (1-3) 32768-40959
; ZERO.PICTURE.1 (8000 BYTES)
; 8192-16191      (2-0) 32768-40767
; ZERO.PICTURE.2 (8000 BYTES)
; 8192-16191      (2-1) 32768-40767
; BRM.SCENARIO.1 (4677 BYTES)
; INDIRECT COPY   (2-2) 32768-37444
; BRM.SCENARIO.2 (3039 BYTES)
; INDIRECT COPY   (2-2) 37445-40483
; ZERO.3000 (4689 BYTES)
; 3000-7688       (2-3) 32768-37456
; BRM.SPRITES (832 BYTES)
; 2176-3007       (2-3) 37457-38288
; ZERO.SPRITES (576 BYTES)
; 2176-2751       (2-3) 38289-38864
; ZERO.50090 (2095 BYTES)
; 50090-52184     (2-3) 38865-40959
       .OPT  NOL
CTRREG =1000
SCRCT1 =1001
SCRCT2 =1002
GUNDL  =1003
       .WORD ESTART-16384,$FEBC
       .BYTE 195,194,205,56,48
PRGNUM .BYTE 0
;
;PROCEDURE SELSUB
; SELECTS SUBROUTINE ON CH0BK1
; BY POKING ADDRESS FROM BASIC
SELSUB LDA   #1
       STA   $DFFF
       JSR   32768
       LDA   #32
       STA   $DFFF
       RTS
       JMP   BRM     ;BRM
       JMP   PRGSEL  ;PRGSEL
ESTART STX   $D016
       JSR   $FDA3   ;IOINIT
       JSR   $FD50   ;RAMTAS
       JSR   $FD15   ;RESTOR
       JSR   $FF5B   ;CINT
       CLI
       JSR   $E453   ;COPY VECTORS
       JSR   $E3BF   ;INIT
BRM    LDA   #0
       STA   53280
       STA   53281
       STA   $DFFF
       LDA   #43
       STA   53265
       ADDR  $FD,49152
       ADDR  STOP,FIN
       ADDR  $A3,$8000
       JSR   MOVEIT-16384
       JMP   NEWLOC
NEWLOC LDA   #234     ;DISABLE STOP
       STA   808      ; KEY
       LDA   #128     ;DISABLE CDR/
       STA   657      ; SHIFT KEYS
       LDA   #<16384 ;BASIC START
       STA   641      ;OS BOTTOM L
       LDA   #>16384
       STA   44       ;BASIC BOT H
       STA   642      ;OS BOTTOM H
       LDA   #<16385
       STA   43       ;BASIC BOT L
       ADDR  55,40960 ;BASIC TOP
       LDA   #0
       STA   PRGNUM
       STA   198
       STA   V+16
       STA   V+21
       LDX   #0
       JSR   PREPAR
       JSR   DSABLE
       LDA   #0
       STA   V+39
       STA   V+28
       STA   V+29
       STA   V+23
       LDA   #34
```

```
        STA   2040
        ADDR  $FD,32768
        LDA   #2
        STA   $FB      ;CHIP #/BANK
        JSR   SCENE
        SEI
        ADDR  $314,START
        ADDR  $FB,FM16
        LDA   #40
        STA   SCRCT1
        LDA   #1
        STA   CTRREG
        LDA   #0
        STA   $A1
        STA   $A2
        CLI
        JMP   MVL050
PRGSEL  LDA   PRGNUM
        CMP   #1       ;CALIBRATION?
        BNE   MVL010
        LDX   #CALIB-DATA
        JSR   PREPAR
        JMP   MVL040
MVL010  CMP   #2       ;PRINT FILES?
        BNE   MVL020
        LDX   #PFILES-DATA
        JSR   PREPAR
        JMP   MVL040
MVL020  CMP   #3       ;ZRO&SCHANGE?
        BNE   MVL030
        LDX   #ZEROSC-DATA
        JSR   PREPAR
        JMP   MVL050
MVL030  LDX   #SITPIC-DATA
        JSR   PREPAR
MVL040  LDA   #21      ;TEXT SCREEN
        STA   53272
        LDA   #27
        STA   53265
        LDA   #147     ;CLEAR SCREEN
        JSR   $FFD2
MVL050  LDA   #32      ;RAM
        STA   $DFFF
        LDA   #0
        STA   198
        JSR   $A871
        JMP   $A7AE    ;RUN
PREPAR  LDA   DATA,X
        STA   101
        AND   #63
        STA   $DFFF
        INX
        LDA   DATA,X
        STA   $FD
        INX
        LDA   DATA,X
        STA   $FE
        INX
        LDA   DATA,X
        STA   STOP
        BIT   101
        BPL   PREP20
        STA   45
PREP20  INX
        LDA   DATA,X
        STA   STOP+1
        BIT   101
        BPL   PREP30
        STA   46
PREP30  INX
        LDA   DATA,X
        STA   $A3
        INX
        LDA   DATA,X
        STA   $A4
        INX
        JSR   MOVEIT
        BIT   101
        BVC   PREPAR
        RTS
MOVEIT  LDY   #0
MVL060  LDA   ($A3),Y
        STA   ($FD),Y
        DINC  $A3
        DINC  $FD
        LDA   $FD
        CMP   STOP
        BNE   MVL060
        LDA   $FE
        CMP   STOP+1
        BNE   MVL060
        RTS
DATA    .BYTE 3       ;BRM.3400
        .WORD 3400,8182,32768
        .BYTE 11      ;BRM.SPRITES
        .WORD 2176,3008,37457
        .BYTE 4       ;BRM.BAS (1)
        .WORD 16384,24576,32768
        .BYTE 5       ;BRM.BAS (2)
        .WORD 24576,32768,32768
        .BYTE 198     ;BRM.BAS (3)
        .WORD 32768,35501,32768
CALIB   .BYTE 194     ;CALIBRATION
        .WORD 16384,17555,39789
PFILES  .BYTE 0       ;MLPRINTFILES
        .WORD 3400,3434,40926
        .BYTE 192     ;PRINT FILES
        .WORD 16384,21169,36141

ZEROSC  .BYTE 11      ;ZERO.3000
        .WORD 3000,7689,32768
        .BYTE 11      ;ZERO.50090
        .WORD 50090,52185,38865
        .BYTE 11      ;ZERO.SPRITES
        .WORD 2176,2752,38289
        .BYTE 6       ;ZERO.BAS (1)
        .WORD 16384,19218,38126
        .BYTE 199     ;ZERO.BAS (2)
        .WORD 19218,27410,32768
SITPIC  .BYTE 194     ;SIGHT PICTURE
        .WORD 16384,19240,36933
        .LIB  INTRO
;INTRO.TXT FOR BRM CARTRIDGE INTRO
START   LDA   $A2
        AND   #3
        BNE   INL040
        ADDR  $A3,1226
        LDX   #12
INL010  LDY   #3
INL020  LDA   ($A3),Y
        CLC
        ADC   #16
        STA   ($A3),Y
        DEY
        BPL   INL020
        LDA   $A3
        CLC
        ADC   #40
        STA   $A3
        LDA   $A4
        ADC   #0
        STA   $A4
        DEX
        BNE   INL010
INL040  LDA   $A1
        AND   #2
        BEQ   INL070
        LDA   #15      ;GUN BANG
        STA   54296
        LDA   #10
        STA   54277
        LDA   #30
        STA   54273
        LDA   #128
        STA   54276
        LDA   #129
        STA   54276
        LDA   #15
        STA   V        ;X: SPRITE 0
        LDA   #208
        STA   V+1      ;Y: SPRITE 0
        LDA   #1
        STA   V+16     ;RIGHT X
        STA   V+21     ;SPRITE CTRRG
        LDA   #0
        STA   $A1
        LDA   CTRREG
        ORA   #8
        STA   CTRREG
        ADDR  $A3,1334
        LDX   #9
INL050  LDY   #6
        LDA   #33
INL060  STA   ($A3),Y
        DEY
        BPL   INL060
        LDA   $A3
        CLC
        ADC   #40
        STA   $A3
        LDA   $A4
        ADC   #0
        STA   $A4
        DEX
        BNE   INL050
        LDA   #<13960
        STA   $FD
        LDA   #<13968
        STA   $C3
        LDA   #>13968
        STA   $FE
        STA   $C4
        LDA   #5
        STA   GUNDL
INL070  LDA   CTRREG
        AND   #1
        BEQ   INL130
        LDA   $A2      ; SCROLL
        AND   #7
        BNE   INL130
        LDY   #0
INL080  LDA   15880,Y
        STA   15872,Y
        INY
        BNE   INL080
INL090  LDA   16136,Y
        STA   16128,Y
        INY
        CPY   #56
        BNE   INL090
        LDY   #7
INL100  LDA   ($FB),Y
        CMP   #254
        BEQ   INL110
        STA   16184,Y
        DEY
        BPL   INL100

        LDA   $FB
        CLC
        ADC   #8
        STA   $FB
        LDA   $FC
        ADC   #0
        STA   $FC
        JMP   INL130
INL110  LDA   #0
        STA   16184,Y
        DEY
        BPL   INL110
        DEC   SCRCT1
        BNE   INL130
        LDA   #3
        STA   SCRCT2
        EOR   CTRREG
        STA   CTRREG
        LDY   #39
        LDA   #33
INL120  STA   1984,Y
        DEY
        BPL   INL120
INL130  LDA   CTRREG
        AND   #2
        BEQ   INL150
        LDY   #248     ;BLINK
INL140  LDA   PSBTB-1,Y
        STA   15911,Y
        DEY
        BNE   INL140
        LDA   #180
        STA   SCRCT1
        LDA   CTRREG
        EOR   #66
        STA   CTRREG
INL150  LDA   CTRREG
        AND   #8
        BEQ   INL170
        LDY   #0       ;KICK LEFT
        JSR   SPRUPD
INL160  LDA   ($C3),Y
        STA   ($FD),Y
        INY
        CPY   #240
        BNE   INL160
        LDA   $C3
        CLC
        ADC   #64
        STA   $C3
        LDA   $C4
        ADC   #1
        STA   $C4
        LDA   $FD
        CLC
        ADC   #64
        STA   $FD
        LDA   $FE
        ADC   #1
        STA   $FE
        DEC   GUNDL
        BNE   INL170
        LDA   #10
        STA   GUNDL
        LDA   CTRREG
        EOR   #24
        STA   CTRREG
INL170  LDA   CTRREG
        AND   #16
        BEQ   INL180
        JSR   SPRUPD
        DEC   GUNDL    ;GUN DELAY
        BNE   INL180
        LDA   CTRREG
        EOR   #48
        STA   CTRREG
        LDA   #5
        STA   GUNDL
        LDA   #<13952
        STA   $FD
        LDA   #<13960
        STA   $C3
        LDA   #>13960
        STA   $FE
        STA   $C4
INL180  LDA   CTRREG
        AND   #32
        BEQ   INL220
        LDY   #240     ;KICK RIGHT
        JSR   SPRUPD
INL190  LDA   ($FD),Y
        STA   ($C3),Y
        DEY
        CPY   #255
        BNE   INL190
        LDA   $C3
        CLC
        ADC   #64
        STA   $C3
        LDA   $C4
        ADC   #1
        STA   $C4
        LDA   $FD
        CLC
        ADC   #64
        STA   $FD
        LDA   $FE
        ADC   #1
```

```
        STA   $FE
        DEC   GUNDL
        BNE   INL220
        LDA   CTRREG
        AND   #223
        STA   CTRREG
        LDA   #0
        STA   V+21     ;SPRITE CTRRG
        ADDR  $A3,1334
        LDX   #9
INL200  LDY   #6
        LDA   #177
INL210  STA   ($A3),Y
        DEY
        BPL   INL210
        LDA   $A3
        CLC
        ADC   #40
        STA   $A3
        LDA   $A4
        ADC   #0
        STA   $A4
        DEX
        BNE   INL200
INL220  LDA   CTRREG
        AND   #64
        BEQ   INL280
        DEC   SCRCT1   ;DELAY PSB
        BNE   INL280
        DEC   SCRCT2
        BNE   INL240
        LDA   CTRREG
        EOR   #65
        STA   CTRREG
        ADDR  $FB,FM16
        LDY   #40
        STY   SCRCT1
        DEY
        LDA   #177
INL230  STA   1984,Y
        DEY
        BPL   INL230
        JMP   INL260
INL240  LDA   SCRCT2
        AND   #1
        BEQ   INL250
        LDA   CTRREG
        EOR   #66
        STA   CTRREG
        JMP   INL280
INL250  LDA   #60
        STA   SCRCT1
INL260  LDY   #247
        LDA   #0
INL270  STA   15912,Y
        DEY
        BNE   INL270
INL280  JMP   $EA31
SPRUPD  INC   V        ;X: SPRITE 0
        INC   V
        INC   V
        RTS
FM16    .BYTE 126,96,96,120,96,96,96,0,0,
0,60,102,102,102,60,0,0,0,124
        .BYTE 102,96,96,96,0,0,0,0,0,0,0,
0,0,124,102,102,124,102,102,124
        .BYTE 0,0,0,60,6,62,102,62,0,0,0,
62,96,60,6,124,0,0,24,0,56,24
        .BYTE 24,60,0,0,0,60,96,96,96,60,
0,0,0,0,0,0,0,0,0,124,102,102
        .BYTE 124,120,108,102,0,0,24,0,
56,24,24,60,0,0,14,24,62,24,24,24
        .BYTE 0,0,56,24,24,24,24,60,0,0,
0,60,102,126,96,60,0,0,0,0,0,0,0
        .BYTE 0,0,0,99,119,127,107,99,99,
99,0,0,0,60,6,62,102,62,0,0,0,0
        .BYTE 124,102,96,96,96,0,0,96,96,
108,120,108,102,0,0,0,62,96,60
        .BYTE 6,124,0,0,0,102,127,127,
107,99,0,0,0,60,6,62,102,62,0,0,0
        .BYTE 124,102,102,102,102,0,0,0,
62,96,60,6,124,0,0,96,96,124,102
        .BYTE 102,102,0,0,24,0,56,24,24,
60,0,0,0,124,102,102,124,96,96
        .BYTE 0,0,0,0,0,0,0,0,126,24,24,
24,24,24,24,0,0,0,124,102,96,96
        .BYTE 96,0,0,0,60,6,62,102,62,0,
0,24,0,56,24,24,60,0,0,0,124,102
        .BYTE 102,102,102,0,0,24,0,56,24,
24,60,0,0,0,124,102,102,102,102
        .BYTE 0,0,0,62,102,102,62,6,124,
254,254,254,254,254,254,254,254
PSBTB   .BYTE 0,102,60,255,60,102,0,0,0,
102,60,255,60,102,0,0,0,102,60,255
        .BYTE 60,102,0,0,0,0,0,0,0,0,0,0,
124,102,102,124,96,96,96,0,0,0
        .BYTE 124,102,96,96,96,0,0,0,60,
102,126,96,60,0,0,0,62,96,60,6,124
        .BYTE 0,0,0,62,96,60,6,124,0,0,0,
0,0,0,0,0,0,60,102,96,60,6,102
        .BYTE 60,0,0,0,124,102,102,124,
96,96,0,0,60,6,62,102,62,0,0,0,60
        .BYTE 96,96,96,60,0,0,0,60,102,
126,96,60,0,0,96,96,124,102,102,124
        .BYTE 0,0,0,60,6,62,102,62,0,0,0,
124,102,96,96,96,0,0,0,0,0,0,0,0
        .BYTE 0,0,0,24,126,24,24,24,14,0,
0,0,60,102,102,102,60,0,0,0,0,0
        .BYTE 0,0,0,0,124,102,102,124,
102,102,124,0,0,0,60,102,126,96,60
        .BYTE 0,0,0,62,102,102,62,6,124,
0,24,0,56,24,24,60,0,0,0,124,102
        .BYTE 102,102,102,0,0,0,0,0,0,0,
0,0,0,102,60,255,60,102,C 0,0,102
        .BYTE 60,255,60,102,0,0,0,102,60,
255,60,102,0,0
FIN     =*
        .END  INTRO
        .OPT  LIST
        .END
```