Research Product 90-01

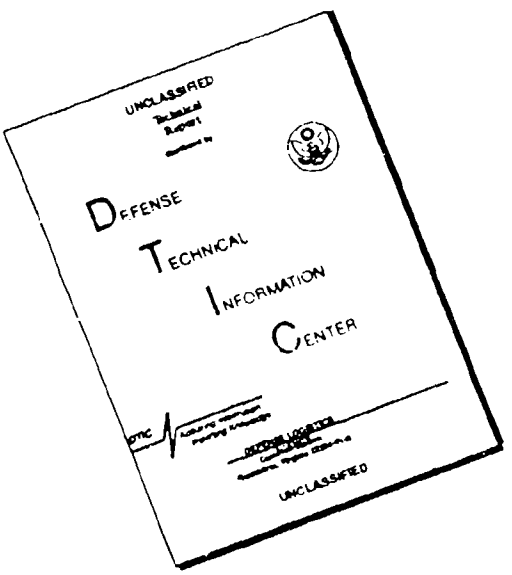
Multipurpose Arcade Combat Simulator (MACS) Basic Rifle Marksmanship (BRM) Program

October 1989

Fort Benning Field Unit
Training Research Laboratory

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

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

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

Research accomplished under contract
for the Department of the Army

Litton Computer Services

Technical review by

Kenneth L. Evans
Margaret Salter

NOTICES

FINAL DISPOSITION: This Research Product may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: This Research Product is not to be construed as an official Department of the Army document, unless so designated by other authorized documents.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS --
2a. SECURITY CLASSIFICATION AUTHORITY --	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE --	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) --	5. MONITORING ORGANIZATION REPORT NUMBER(S) Research Product 90-01
6a. NAME OF PERFORMING ORGANIZATION Litton Computer Services 6b. OFFICE SYMBOL (If applicable) --	7a. NAME OF MONITORING ORGANIZATION ARI Field Unit at Fort Benning, GA
6c. ADDRESS (City, State, and ZIP Code) 1300 Villa Street Mountain View, CA 94041	7b. ADDRESS (City, State, and ZIP Code) P.O. Box 2086 Fort Benning, GA 31905
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences 8b. OFFICE SYMBOL (If applicable) PERI-IJ	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA903-88-C-0407
8c. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 63007A; PROJECT NO. 794; TASK NO. (322) 3306; WORK UNIT ACCESSION NO. C3

11. TITLE (Include Security Classification)

Multipurpose Arcade Combat Simulator (MACS) Basic Rifle Marksmanship (BRM) Program

12. PERSONAL AUTHOR(S) Broom, John M.; Champion, David F.; Greene, William H.; Martere, Ronald F.; Purvis, Jamie W.; and Sills, Edward G.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM 88/01 TO 89/07	1989, October	92

16. SUPPLEMENTARY NOTATION

Seward Smith, Contracting Officer's Representative

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
05	06		Computer Simulation, Instructional Materials,
19	06		Rifle Marksmanship, Training Aid

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

This document contains a description of the Basic Rifle Marksmanship (BRM) program of the Multipurpose Arcade Combat Simulator (MACS) system, along with a record of the measures taken, standards used, underlying instructional methodologies, a statement of the technical limitations of the system, and copies of programming flow charts. The information is for use by personnel who may be involved in future developmental work on the MACS system and for those who must demonstrate and explain the BRM program.

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) 545-5589 22c. OFFICE SYMBOL PERI-IJ

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

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Research Product 90-01

Multipurpose Arcade Combat Simulator (MACS) Basic Rifle Marksmanship (BRM) Program

John M. Broom, David F. Champion, William H. Greene, Ronald F. Martere, Jamie W. Purvis, and Edward G. Sills
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

October 1989

Army Project Number
2Q263007A794

Education and Training

FOREWORD

The Multipurpose Arcade Combat Simulator (MACS) is an inexpensive, part-task weapons trainer that was developed to overcome the training problems encountered because of insufficient facilities and too few instructors. Because it uses mostly off-the-shelf components, MACS is substantially lower in cost than other available training devices, although it is equally effective. While current interest is focused primarily on its use as a rifle marksmanship trainer, MACS was designed so that the basic hardware could be used to provide training on a variety of weapon systems.

This report describes the Basic Rifle Marksmanship (BRM) software designed for use with the MACS system. An overview of the software is presented, followed by a discussion of the instructional design of the program. Presented in detail are discussions of the feedback provided on marksmanship fundamentals, derivation of performance standards, skill tests used to determine appropriate starting levels, and program options that may be accessed through the menu.

Patented in 1986, MACS is a product of the U.S. Army Research Institute for the Behavioral and Social Sciences Fort Benning Field Unit, which conducts research on training and training technology with particular emphasis on individual and small team skills in the Infantry arena. The research task that supports this mission is titled "Developing Training for Individual and Crew-served Weapons" and is organized under the "Training for Combat Effectiveness" program area. The U.S. Army Infantry School under a Memorandum of Understanding (9 December 1987) and the U.S. Army Training Support Center under a Training Device Need Statement for MACS approved in 1984 provided sponsorship for the MACS research program. The MACS system software described in this report is being fielded Army-wide by the Training Support Center at Fort Benning, Georgia.

EDGAR M. JOHNSON
Technical Director

MULTIPURPOSE ARCADE COMBAT SIMULATOR (MACS) BASIC RIFLE MARKSMANSHIP (BRM) PROGRAM

CONTENTS

CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

MULTIPURPOSE ARCADE COMBAT SIMULATOR (MACS)
BASIC RIFLE MARKSMANSHIP (BRM) PROGRAM

Introduction

The MACS BRM program has been under development since 1985. The BRM program is intended for school instruction in basic rifle marksmanship, as well as for remedial and sustainment training. The program has been used with particular success in helping the average and below average shooter attain proficiency in marksmanship skills (Evans, 1988). The purpose of this paper is to describe the latest version of the BRM program (Version 8901). The program's instructional design, performance measures, and rationale behind the standards are described in detail. This paper will be of interest to individuals engaged in MACS developmental work and others who are involved in demonstrating the BRM program.

The MACS system is a low-cost, computer-based, part-task weapons trainer. The system components include a weapon (e.g., a demilitarized M16 rifle) with an optical focus light pen attached to the barrel, a Commodore 64 computer, and software appropriate to the characteristics of the weapon. The system allows shooters to practice basic skills by firing at targets at scaled ranges that are displayed on a computer screen. Diagnostic feedback is provided. The program has been validated as a teaching device (Evans, 1988).

The MACS BRM program presents stationary targets only, set at ranges between 50 and 300 meters. The program includes an initial assessment of the shooter's marksmanship skill level, followed by nine progressively more difficult levels of training. Specific performance standards are set for each level. At the end of each level, the shooter's performance is compared to standards within the program. At this point, a shooter may advance to the next level, stay at the same level, or regress to a lower level.

The program is designed to be used with an instructor present; however, one instructor can monitor several systems concurrently, since the on-screen text is largely self-explanatory to the user. As long as a soldier is making satisfactory progress, the computer screen border remains black in color. The instructor's attention is drawn to any soldier having difficulty by a change in border color. The program is designed to help the instructor diagnose the shooter's problems and provide corrective guidance to the soldier. Diagnostic measures include steady position, aiming, trigger squeeze, and breath control. Shot groups for each range are displayed at the end of each firing sequence or level in the program.

The program begins with a marksmanship skill test, which is used to zero the system electronically, and to determine the appropriate starting level for the shooter. Shooters beginning at a low level receive heavily augmented feedback and, as progress is made from level to level, this feedback is gradually withdrawn. When shooters reach Level 7, they are performing with feedback that simulates that found in actual record fire tests and they are firing to record fire standards. Levels 8 and 9 take the shooters beyond basic record fire requirements and prepare them for the Advanced Rifle Marksmanship (ARM) course.

The BRM program is written in 6510 Assembler and BASIC. Flow charts, program documentation, and program listings are presented in Appendixes C through G.

Instructional Design of the BRM Program

When novices learn to shoot, they frequently flinch or blink as they pull the trigger. When flinching occurs in anticipation of the discharge of the weapon rather than following the discharge, it disturbs the lay of the weapon. Thus, in live fire, a new shooter has to learn to control any tendency to flinch and, concurrently, master the fundamentals of marksmanship that enable accurate shooting.

Because the rifle used with the MACS BRM system has no flash, recoil, or cartridge ejection when fired, no anticipatory flinch is induced. This enables the shooter to concentrate on developing the psychomotor skills needed to master the fundamentals of marksmanship. These are maintenance of a steady position, controlled breathing, correct aiming, and good trigger control (Osborne & Smith, 1985; U.S. Army Infantry School (USAIS), 1984).

Evans (1989) has demonstrated that rifle marksmanship simulators that have no recoil train as effectively as those that incorporate it; and he has pointed out that the lack of recoil is of particular advantage to the instructor because it enables errors arising from poor trigger squeeze to be identified. In live fire, such errors are often confounded with errors arising from anticipatory flinch. Poor trigger control has been shown to cause more target misses than does failure to master each of the other fundamentals (Osborne & Smith, 1985; USAIS, 1984).

The Army teaches soldiers to aim at the center of mass of a target. The rationale for this is that if soldiers always aim at the center of mass of a stationary target, they have a higher chance of hitting that target. However, the overall purpose of basic rifle marksmanship training is to enable the soldier to hit the enemy in combat. For this reason, the minimum standard set in record fire is to hit 23 out of 40 targets; the hits are credited irrespective of the location of the shots relative to center of mass. Thus, two separate criteria exist (<u>aim center of mass</u> and <u>hit the target</u>), and these criteria are not always compatible.

For example, a shot that strikes an E-type silhouette in the head is located farther from the center of mass of that target than a shot that just misses the target but is in the horizontal plane of the center of the target. By the criterion of <u>aim center of mass</u> the miss is a better located shot than is the head shot. However, by the hit or miss criteria, the head shot is clearly better. The MACS BRM program uses both criteria. Progress within a level is governed by a requirement to hit a set number of targets, either at each range or in total. Progress from level to level is determined by the accuracy of shots relative to the center of mass of the targets. Therefore, a soldier who progresses through the levels is not only hitting an adequate number of targets, but is hitting targets within reasonable proximity to center of mass.

Many scholars, and notably Smith and Smith (1966), argue that, in essence, the act of learning is the adjustment of one's activities in the

light of feedback received. Thus in learning to shoot, one fires, observes the result (target does or does not fall), makes adjustments, and fires again; the objects being to reduce discrepancies between the goal (hitting the target) and the result (bullet strike - if this is observed). Although a shooter can learn independently, the learning process is speeded by the provision of extrinsic, augmented feedback, such as sure knowledge of where the missing shot went, and guidance on what one did wrong. Bilodeau and Bilodeau (1958) have shown that for some motor skill tasks, the rate of learning is proportional to the number of trials providing extrinsic feedback, and that ten trials each with extrinsic feedback can teach as rapidly as 100 trials with feedback after every tenth trial.[1]

Bilodeau (1966) describes three types of augmented feedback that have been demonstrated to help in such tasks; these are concurrent feedback, information feedback, and learning feedback. The MACS BRM program uses all three. Concurrent feedback is artificial augmentation of the effect of a response. In the initial levels of the BRM program (Levels 1 and 2), this is provided by a cross showing bullet strike and the words HIT or MISS appearing on the screen. Information feedback is feedback given immediately after a trial and is an objective assessment of performance. In the initial levels of the program, such feedback is given after every shot and takes the form of a screen showing ratings of the shooter's performance on the four fundamentals of marksmanship (steady position, aiming, trigger squeeze and breath control), as well as a replay of the front sight post coming onto target. Learning feedback is delayed feedback provided at the end of a series of trials. The MACS BRM program provides summary screens showing overall performance at the end of each level.

Because augmented feedback is not available in actual combat, it is important that the shooter does not become dependent upon it. For example, the showing of bullet strike on the screen when the shooters miss aids in adjusting their subsequent shots, but in combat they may not know where the round impacted. Therefore, as the shooter progresses, feedback is gradually withdrawn so that in Levels 3 and 4, detailed single-shot feedback is only given for a poor shot and in Levels 5 and 6, feedback is given only in summary form. By the time shooters reach Level 7, they have learned to operate with the same amount of feedback that they would receive on a record fire range. Precise details of the feedback at each level are in Appendix A and are discussed in the descriptions of the levels (see p. 5). The withdrawal of feedback gradually) and only after the shooter has manifested a reasonable level of proficiency agrees with conclusions reached by Stammers and Patrick (1975) in their comprehensive discussion of the subject.

The MACS BRM program is a self-paced training system that enables individual shooters to progress at a rate appropriate to their initial skill

---

[1]Lavery (1962) argues that performance can contain both variable error and constant error, where variable error is the result of random factors, and constant error is the result of the trainee's behavior. He points out that provision of feedback after every trial can cause a trainee to try and correct for "variable error" and therefore concludes that feedback should only be given at the end of a block of trials. This view is not subscribed to in the design of the MACS BRM program.

levels and aptitudes for the task. When new shooters begin on the system, they are given a test that assesses the level at which they should begin. The program then may start them at any level between 1 and 5. It is only a very competent shooter, however, who begins at Level 3 or above. This test is discussed in detail subsequently.

The program is designed to enable self-pacing. To this end, information screens at the start of each level inform the shooter of the number of targets that will appear, whether they are timed or untimed targets, the position the shooter should adopt (supported or unsupported), and the standard the shooter must meet (e.g., in Levels 1 and 2, to hit two out of three targets at each range). At the start of Levels 1 and 2 only, additional screens instruct the shooter to aim at center of mass and teach what is meant by center of mass. When shooters fail to meet the standard, they are told on screen that they must refire; they are informed when they have refired successfully. Similarly, if they are sent back a level or made to repeat a level, they are told on screen that this is happening, and why it is happening.

Before a novice can use the MACS BRM program to develop marksmanship skills, three basic concepts must be understood. These are correct sight alignment, correct placement of the front sight post on the target, and what is meant by a tight shot group. However, because not all users are novices, these concepts are taught in an optional introductory program that the instructor can access via the system menu. This program is described in the section on optional programs and features.

For the expert marksman, a motivational element is present in the program. At the end of level 9, the top ten shooters may record their initials on the screen upon completion of the level. The recording of shooter's initials is designed primarily to encourage competition among those undergoing sustainment training.

The MACS BRM program is designed so that one instructor can monitor a number of shooters, firing on separate individual systems, simultaneously. This is achieved by making the program as self-explanatory as possible. As long as a shooter is making steady progress, the screen border remains black. Only when a shooter is having difficulty does the border change color. Therefore, instructors can scan rapidly down a row of monitors and determine where their individual attention would be best applied. The border color coding is as follows:

- A black border indicates that the shooter is making satisfactory progress.

- A red border indicates that the shooter is refiring targets at a particular range, or a part of a level.

- A blue border shows that the shooter has been made to repeat a level.

- A yellow border means that the shooter has been sent back a level.

- A purple border indicates either that the shooter is aiming off screen and pulling the trigger, or that the light pen is not reading properly.

When an instructor sees a border color that indicates a shooter is having difficulty, the level the shooter is on, the particular target within the level the shooter has reached, and the number of times the shooter has refired can be checked. This information is given in a small inset in the top left-hand corner of the screen. The on-screen feedback designed for the shooter is, of course, also available to the instructor to help diagnose the problem the shooter is having.

Summary of the BRM Program

A brief discussion of each of the nine levels is provided below. Optional features that may be used with some or all of these levels are described separately, as is the initial skills test. A summary level-by-level description of the program with details of feedback and standards is given in Appendix A. Program features, such as the initial skills test, are described elsewhere.

Levels 1 and 2 are designed to develop basic skills. The only difference between the two levels is that Level 1 is fired in the supported position and Level 2, unsupported. For Levels 1 and 2, detailed shot-by-shot feedback is provided. When the shooter fires at the target, HIT or MISS is indicated on screen and the bullet strike is shown by a cross. The shooter then receives a rating (EXCELLENT, GOOD, AVERAGE, BELOW AVERAGE, POOR) on three diagnostic measures (steady position, aiming, trigger squeeze) for the shot fired, and the location of the shot relative to center of mass of the target is rated. Breath control is rated O. K. or CHECK. The movements of the front sight of the rifle are replayed for a period of up to 6 seconds before trigger closure. Then the frame is briefly frozen to allow comparison of the final front sight position (shown in white) with the correct front sight position of the target (shown in black). Three targets are presented at each of six ranges (50 to 300 m). F-type targets are presented at 50 and 100 m, E-type targets at all other ranges. After each set of three shots, a screen showing the three-round shot group on a single target is provided. At the end of the 18 shots, a summary of the mean ratings for steady position, aiming, trigger squeeze, breath control, and shot location is presented. As shooters go through each level, internal algorithms determine whether they should refire targets at a given range, and at the end whether they should be permitted to advance to the next level.

Levels 3 and 4 introduce shooters to timed targets and to the need to adjust their aims for targets at different ranges. To this end, targets at different ranges are presented in a random order. Concurrently, some of the feedback is withdrawn. HIT and MISS no longer appear on the screen, and diagnostic information and replay are provided only when the shooter fires badly (rated POOR by one of the diagnostic measures or BELOW AVERAGE on shot location). The main difference between the levels is that Level 3 is fired in the supported position and Level 4, unsupported. There is a slight relaxation of standards for 250- and 300-m targets when they are fired at from the unsupported position. Information and replay are provided only when the shooter fires poorly (rated POOR by one of the diagnostic measures or BELOW AVERAGE on shot location).

Levels 5 and 6 provide practice for record fire. The number of targets at each range and their exposure times are the same as those used in record fire, but the order of target presentation is randomized. In Level 5, bullet strike is still shown by a cross for both misses and hits; in Level 6, it is shown for misses only. At these levels, diagnostic feedback and replays are no longer provided. Only two measures are taken, a count of hits and misses, and the measure of shot location. The rationale for this is that good shot location is largely the result of having a steady position, accurate aiming, and good trigger squeeze. These have been emphasized for four levels, and the shooter should by now have an understanding of the causes of poor shots. Levels 5 and 6 are both divided into two parts; the first 20 shots are fired from the supported position and the next 20, unsupported. Shooters must hit 15 out of 20 targets on the first part before proceeding to the second part, which also requires 15 out of 20 hits; thus shooters are being trained to sharpshooter standard (30/40). At the end of each level, shooters are graded according to the number of targets they hit (Expert 36/40, Sharpshooter 30/40, Marksman 23/40, or Unqualified). Because training is to sharpshooter standard, shooters are graded marksman if they have to refire one of the positions.

Whereas the criterion for getting through the two parts of Levels 5 and 6 is the number of hits obtained, the mean rating of shot location determines whether the shooter progresses to the next level, refires the same level, or regresses one level. Thus shooters who have hit the 30 out of 40 targets needed to get through a level may still be regressed one level if the mean rating of shot location, over all 40 targets, is POOR. This is reasonable because an average rating of POOR indicates that they are just nicking the targets rather than hitting them squarely.

Level 7 (record fire) is the culmination of the previous six levels of training and is an on-screen re-creation of actual record fire. It is identical to the previous level (Level 6), but feedback is reduced to that which a shooter would receive on the record fire range, i.e., targets disappear when hit, or remain until their exposure time has elapsed (indicated by a whistle) and then disappear. Bullet strike is no longer shown, and no replays or diagnostic scores are given. Level 8 (rapid record fire) takes the shooter beyond the requirements for record fire. It is identical to level 7 in structure, but the exposure times of targets are reduced to two-thirds of the time allowed for record fire as an initial step in the transition to the engagement of moving targets, in the MACS Advanced Rifle Marksmanship (ARM) program (under development).

Level 9 (combat fire) is an attack/retreat scenario. Because up to five individually timed targets may appear on the screen simultaneously, and because new targets may appear as others are being engaged, it teaches shooters to maintain an awareness of their whole fields of fire even as they engage particular targets. Individual target exposure times vary but are similar to or less than those allowed for record fire. The level is divided into two parts, the first 40 shots being fired from a supported position and the second 40, unsupported. The shooter must hit 23 out of 40 targets on the first part before proceeding to the second part, which also requires 23 out of 40 hits. If shooters do not meet this standard, they must refire the position. At the end of each position, the shooter is graded Expert, Sharpshooter, Marksman, or Unqualified. Once a shooter has reached this level, the only measure taken is hit or miss. The locations of the shots

relative to center of mass of targets are no longer considered, and therefore there is no question of the shooters being sent back a level or being made to repeat the whole level.

BRM Program Feedback

The instructional design of the BRM program incorporates feedback for each shot fired in Levels 1 and 2, and for poor shots in Levels 3 and 4. Ratings are provided for each of the four marksmanship fundamentals: steady position, aiming, breathing, and trigger squeeze. In addition, a rating of shot location is provided. The first four measures are diagnostic, and assess the shooter's performance up to the moment of firing. The shot location measure assesses bullet strike in relation to center of mass.

The diagnostic and resultant scores are obtained as follows. The light pen reads the screen constantly and maintains a running memory (stack) of 127 readings[2]. Immediately after trigger closure, the appropriate readings are extracted from the stack and the scores are calculated. These scores are then compared with criteria values to obtain the verbal ratings. The scores for each measure are described below.

The shot location score is determined by the radial distance between the location of the last light pen reading before trigger closure and the center of mass of the target.

The aiming score is determined by the radial distance between the mean point of aim (the average location of readings 45 through 7 (inclusive) before trigger closure) and the center of mass of the target. Because the light pen takes 60 readings per second, the aiming score is taken over a period of 0.65 seconds.

The steady position score is based on the standard deviation of light pen readings in the X and Y axes about the point of aim and is also taken over readings 45 though 7 before trigger closure. The greater the degree of wobble, the greater the standard deviations, and the lower the score awarded.

Since breath control cannot be directly measured, the ratings assigned to breath control are derived from the steady position score. The three highest steady position scores (EXCELLENT, GOOD, or AVERAGE) correspond to a breath control rating of O.K. The assumption is made that given a relatively steady position, breath control is adequate. If the position is relatively unsteady, as indicated by the two lowest steady position scores (BELOW AVERAGE and POOR), the breath control feedback is CHECK. The assumption is made that a poor steady position may be due to poor breath control, or to other factors. Breath control, as one of the four fundamentals of marksmanship, is emphasized in the feedback given to the soldier. Of course, the instructor observing the soldier in the act of shooting, will remain alert for other factors contributing to unsteadiness.

---

[2]This should not be confused with the replay which is based on 256 readings taken with a sampling rate of 20 readings per second.

The trigger squeeze score is based on the range of readings in the X and Y axes about the point of aim, and is taken from reading 6 before trigger closure to reading 3 after trigger closure; that is, over 0.15 seconds. It should be noted that the trigger squeeze measure is not concerned with how rapidly or steadily the trigger is squeezed, but with whether or not the point of aim is disturbed. The actual algorithms used to compute all diagnostic scores are in Appendix H.

Aiming and Shot Location

The MACS BRM program contains separate standards for firing in the supported and unsupported positions; those for the supported position are more rigorous. A supported position is a "position which utilizes something other than the body to steady the weapon" (USAIS, 1984). The unsupported position assumes no support other than the body and the ground. The difference is that in the unsupported position the weight of the rifle is borne by the forearm, wrist, and hand, whereas in the supported position some object bears this weight. Thus shooters in a standing foxhole position, who use the ground to support their elbows, are defined as firing unsupported. If the rifle is bedded onto sandbags, then firing is termed supported.

Supported position. Underlying all targets that appear on the screen is a second, unseen, and mathematically defined target area that is analogous to a bull's-eye. The rings of this bull's-eye are defined in terms of radial distances (in pixels) from the center of mass of each target, with a correction to accommodate the rectangular shape of pixels (so that the unseen bull's-eye is close to circular on the screen), and adjusted to ensure that the areas defined as EXCELLENT and GOOD correspond closely to 2-cm and 4-cm circles on an E-type silhouette target scaled for 25 m.

This bull's-eye is of constant size, irrespective of the perceived size of targets. Figure 1 shows a representation of the unseen bull's-eye on a section of screen. The small rectangles represent the pixels on the screen. The mathematically defined midpoint of the target area is shown as point 0,0. As can be seen, the target area is not centered perfectly on the midpoint, and even with a redesign of the shape of the area it could not be perfectly centered. This is because of the way the computer stores the values of odd-numbered pixels in the X axis, and is fully explained in Appendix B.

The midpoint of the mathematically defined target area is always set at the center of mass of each target seen on the screen. When a shot is fired at the target, and if the mean point is computed to be in the area designated GOOD, then the shooter receives a rating of GOOD for aiming. If the final light pen reading before trigger closure is from one of the six pixels labeled EXCELLENT, then shot location is rated as EXCELLENT. The mathematically defined target area is smaller than the 50- and 100-m targets, but larger than those at 150-m and beyond. For example, the 300-m target is only 5 pixels wide and 9 pixels high. Therefore, a shot can miss a 300-m target and still hit the underlying target area in the area rated as average. Because it is not intuitively obvious to the user how a miss can deserve a rating of AVERAGE in shot location, the shot location score is reported on the screen as a MISS. On-screen ratings of POOR through EXCELLENT are given only when the target has been hit. However, the computer maintains an internal record of the actual

rating, and this is included in the calculations for the summary feedback screens.

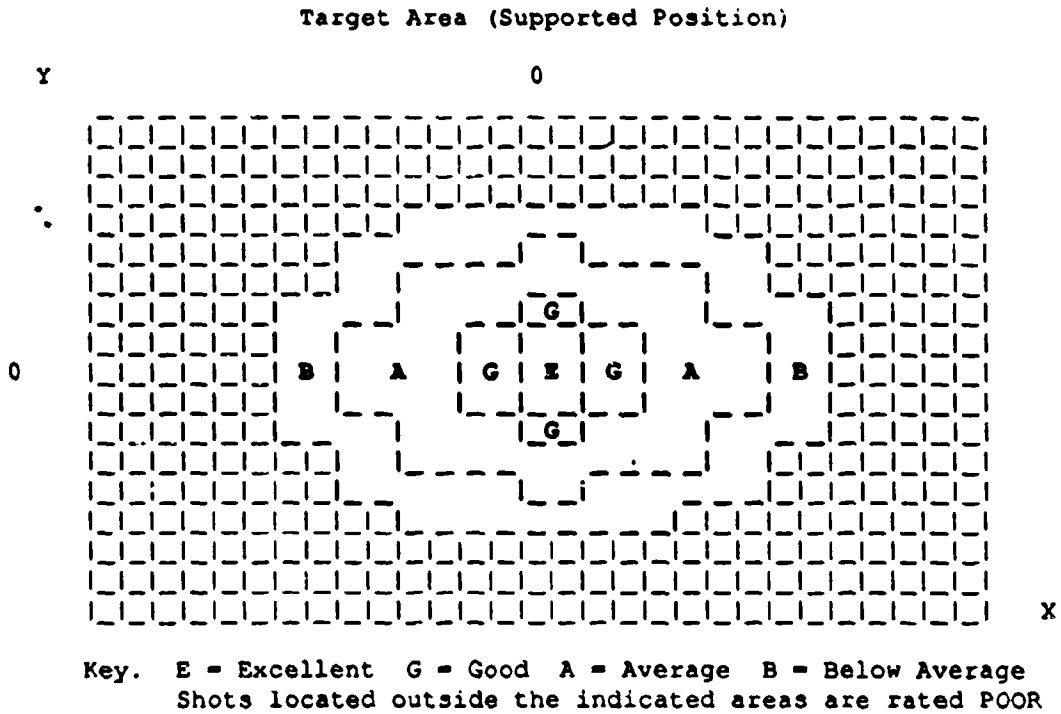


Figure 1. A representation of a 2.15 x 1.66 cm section of screen, showing the target area and the ratings that are awarded for mean points of aim and shots located at different distances from the center of mass. The point 0,0 would be at the center of mass of an observed target.[3]

The areas defined in the underlying bull's-eye target are best explained by relating them to circles on a target at 25 m. The area defined as EXCELLENT comprises 6 pixels. These 6 pixels occupy the same area as would a circle of 0.24 cm diameter on the screen. A circle of .24 cm diameter subtends an angle of 2.68 minutes of arc when viewed at 3 m, which is the eye to screen distance for using the MACS system. Viewed at 25 m, a circle of 1.95-cm diameter would also subtend an angle 2.68 minutes of arc. Thus were a shooter's shot location to be rated EXCELLENT on three consecutive shots, he could have fired the equivalent of a 2-cm, three-round shot group at 25 m. In a similar manner, we can calculate that the area defined as GOOD or better, equates to a circle of 3.73 cm diameter at 25 m, AVERAGE to a circle of 7.30 cm diameter, BELOW AVERAGE to a circle of 10.25 cm diameter, and POOR is greater than 10.25 cm.

[3]Actual pixels are longer in the Y axis than in the X axis (approximate ratio is 1.33:1); the on-screen target area is, therefore, more circular in shape than is shown.

The target areas were derived from two sources. First, 300 initial entry trainees fired a series of 250-m targets on MACS. The distances between bullet strike and the center of mass of the targets were recorded in numeric form and the mean and standard deviation (SD) of the numeric scores were calculated (Hunt et al., 1987). Arbitrarily, shots located within 1 SD either side of the mean were termed AVERAGE, shots located between 1 and 2 SDs above the mean were termed GOOD, and those over 2 SDs above the mean were termed EXCELLENT. Likewise, shots located between 1 and 2 SDs below the mean were termed BELOW AVERAGE, and those over 2 SDs below the mean were termed POOR. This yielded an underlying bull's-eye target that reflected the performance of novice shooters but was overgenerous as a training standard. The area rated GOOD or better, for example, equated to a circle of 4.64 cm diameter at 25 m.

The second source used was Field Manual 23-9 (Department of the Army, 1988), which contains guidelines for categorizing shot groups on a scaled E-type target at 25 m (fired from the supported position). These state that a 2-cm, three-round shot group "indicates no ... shooter error is involved," a 3-cm, three-round shot group represents "acceptable firing performance," a 4-cm, three-round shot group indicates that "shooter error" has occurred, and a 6-cm, three-round shot group indicates obvious error.

Using this information, a constant was added to the algorithms derived from the performance of the 300 trainees that resulted in the target area shown in Figure 1. The areas defined in Figure 1 as EXCELLENT and GOOD are the closest approximation that can be reached on the screen to circles of 2 and 3 cm diameter at 25 m. (For reasons discussed in Appendix B, odd-numbered pixels in the X axis always take the value of the pixel to their left. This means that no part of the area defined as GOOD in Figure 1 can be reduced without that part being eradicated. One cannot, for example, make the areas left and right of EXCELLENT just 1 pixel wide.) Scores of AVERAGE and below for shot location indicate that shooter error has occurred. The choice of the term AVERAGE to describe inadequate shooting was made with a view to encouraging the shooter. We may also note that the shooter who is rated average on shot location has in fact fired a little better than what was defined as average from the performance of the trainees. This is because the constant that was added to limit the area defined as GOOD, also reduced the areas designated AVERAGE and BELOW AVERAGE, in each case by approximately 40%.

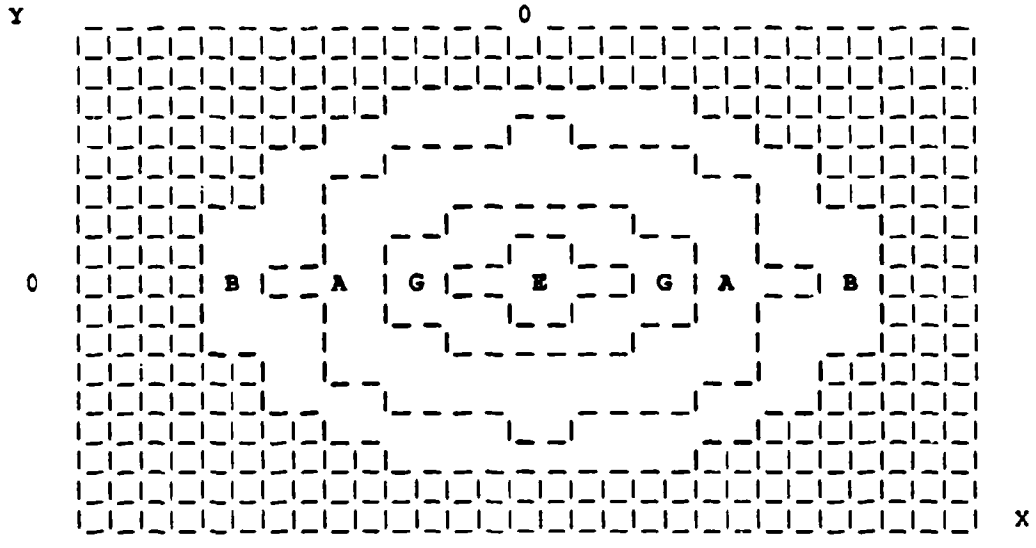
It has been inferred that if shooters could consistently score EXCELLENT on shot location with MACS, they would be placing their shots within the equivalent of a 2-cm circle at 25 m. In fact, such consistency would exceed the technical limits of the system. This is because the light pen is not that precise. A light pen is judged to be correctly calibrated if it reads to within 2 pixels (+ or -) of a given pixel. In fact, a well grounded and correctly calibrated light pen does a little better than this, in that it rarely deviates more than 1 pixel in the Y axis. Because of these deviations, a shooter who consistently fires at the precise center of mass of a target will occasionally have his shot location rated as GOOD; similarly, GOOD shots will on occasions be rated EXCELLENT. Over an entire level of shooting, these deviations cancel out, so that summary screens are more accurate than are feedback screens on individual shots.

Unsupported position. The criteria for allocating ratings of shot location and aiming in unsupported fire are less rigorous than those for

supported fire. This is because firing from an unsupported position is more difficult than firing supported. The target areas are shown in Figure 2.

As may be seen, the area rated as EXCELLENT now comprises 10 pixels on the screen. Scaled out to 25 m, this may be equated to the area encompassed by a circle of 2.52 cm diameter. The area rated as GOOD or better has been increased to 42 pixels and equates to a circle of 5.16 cm diameter at 25 m. One implication of the larger target areas is that deviations in light pen readings have a reduced impact. Therefore, a shooter who consistently aims and fires at the exact center of mass of a target should receive single shot ratings of EXCELLENT with far greater reliability than occurs in the supported position. As with the supported position, the criteria were originally developed from the measured performance of the 300 trainees. These criteria were then modified by the introduction of the same constant that was used to reduce target areas in the supported position; thus the changes made are proportional to those made for the supported position.

Target Area (Unsupported Position)



Key. E = Excellent G = Good A = Average B = Below Average
Shots located outside the indicated areas are rated POOR

Figure 2. A representation of a 2.15 x 1.66 cm section of screen, showing the target areas and ratings awarded for mean points of aim and shots located at different distances from the center of mass for unsupported fire.[4]

[4]Actual pixels are longer in the Y axis than in the X axis (approximate ratio is 1.33:1); therefore on-screen target area is more circular in shape than is shown.

Steady Position and Trigger Squeeze

Steady position ratings for each shot are made based on the sum of the standard deviations of light pen readings in the X and Y axes (corrected for pixel shape), for readings 45 through 7 before trigger closure. The sum is then compared with criteria values. The criteria values were originally established using data collected from the group of 300 trainees referred to earlier. Subsequently, the correction factor for pixel shape was added, and this had the effect of inflating the Y-axis standard deviations. Therefore, the degree of dispersion (wobble) permitted for a shooter to earn a given rating is reduced. These tightened standards agree with the recommendations of subject matter experts (SMEs). The criteria used are given in Table 1.

Table 1.

Criteria Values of Standard Deviation (SD) and Range (R) for Ratings of Steady Position (SP) and Trigger Squeeze (TQ)

	Ratings				
	Excellent	Good	Average	Below Average	Poor
Supported Position					
SP	0<SD<=2.5	2.5<SD<=5.0	5.0<SD<=8.0	8.0<SD<=10.5	10.5<SD
TQ	0< R<=2.5	2.5< R<=5.0	5.0< R<=8.0	8.0< R<=10.5	10.5< R
Unsupported Position					
SP	0<SD<=4.0	4.0<SD<=6.5	6.5<SD<= 9.5	9.5<SD<=12.0	12.0<SD
TQ	0< R<=4.0	4.0< R<=7.0	6.5< R<=11.0	9.5< R<=14.5	14.5< R

Trigger squeeze is measured over the 9/60 of a second period about the moment of firing. The readings used are from the sixth reading before to the third after trigger closure. Measurement is by taking the range of light pen readings in the X axis and in the Y axis (with the Y values corrected for pixel shape), adding the two ranges together, and comparing the resulting score with the criteria values (see Table 1). The mathematical procedures used to compute the steady position and trigger squeeze measures were adopted because of the need to minimize computer processing time. The more obvious method of computing the standard deviation and range from the radial distances between individual light pen readings and the mean point of aim was found to cause unacceptable delays in providing feedback.

In the supported position, the decision to use the same criteria for trigger squeeze as are used for steady position means that a shooter who is rated GOOD on steady position must hold the weapon even more steadily at the moment of trigger closure if he is also to obtain a GOOD on trigger squeeze. This is because the whole range of movement in the 9/60 of a second period about trigger closure must be equal to or less than the standard deviation of

movement in the preceding 36/60 of a second. These more rigorous demands mean that soldiers who wish to improve their trigger squeeze are forced to concentrate on maintaining the lay of the weapon as they pull the trigger.

In the unsupported position, the criteria for trigger squeeze and steady position diverge, the differences between them increasing with the descending order of ratings. This is done to make some allowance for the poorer novice shooter whose fundamental problem may be lack of strength or sufficient muscle tone to bear the weight of the rifle for extended periods, so that muscle fatigue induces tremor.

It should be emphasized here that MACS is a teaching device, designed to facilitate the development of the psychomotor skills necessary to fire the weapon accurately. It is <u>not</u> an analytic tool designed to measure precisely the degree to which shifts in the lay of the weapon are directly attributable to poor trigger control. In fact, although the range of movement that is measured at trigger squeeze may often be the result of poor trigger control, it may equally be nothing more than the range of movement that was present when steady position was measured. In other words, a poor trigger squeeze score can result from an unsteady position. This being acknowledged, let us also recognize that a soldier who concentrates on carefully squeezing the trigger while maintaining the point of aim, is learning to hold the weapon steadily, which is the object of the exercise.

The method of measuring trigger squeeze, and the criteria used to award ratings, were derived empirically. Earlier versions of the BRM program took the standard deviation of the movement, but SMEs judged this measure to be too insensitive. To increase the sensitivity, the range was substituted for the standard deviation. It was then found that the criteria originally derived from the performance of the IETs were too rigorous (SMEs could detect no differences in their own trigger squeeze between shots even though the ratings suggested great differences had existed). Thus, by trial and error, it was found that the steady position criteria combined with the range score gave a degree of sensitivity that was judged appropriate by SMEs.

A detailed explanation of the derivation of the steady position and aiming ratings may be found in Evans' (1988) review. The breath control rating is derived from the steady position measure. Steady position ratings from average to excellent correspond to a breath control rating of O.K., and the below average and poor ratings correspond to a breath control rating of CHECK.

BRM Program Standards

Progression through a level is determined by a simple count of the number of targets hit. A standard is set for each level, and an information screen at the start of the level tells the shooter what the standard is. The standards for each level are given in Appendix A. The instructional principle underlying this progression is that of overlearning. "Overlearning is additional practice after performance standards have been met," and its value is that it aids retention and "tends to prevent skill deterioration under stressful conditions, such as combat" (Bryan & Regan, 1972). The performance standard used to pass a soldier on record fire is that of hitting 23 out of 40 targets. The MACS BRM system achieves overlearning by demanding a higher

standard, and making shooters who fail to meet the standard refire until they succeed. Thus in Levels 1 and 2, the standard is to hit 2 targets out of 3 at each range, which equates to 27 hits with 40 shots; failure to meet the standard at a range necessitates an immediate refire of three more targets at that range. In Level 3, the standard is to hit 3 out of 4 targets at each range; this equates to 30 hits with 40 shots, which is sharpshooter standard. In Levels 5 though 8, the standard is to hit 15 out of 20 targets in each firing position (supported and unsupported), which again equates to 30 hits with 40 shots, and failure to meet the standard requires an immediate refire of all 20 targets in that position.

Progression from Level 1 through Level 4 is determined by the mean ratings that the shooter receives at the end of each level, for steady position, aiming, trigger squeeze, and shot location. For Level 5 through Level 9, only shot location is considered.

For Levels 1 to 4, a shooter must achieve a mean rating of AVERAGE or better on all four measures before he is permitted to advance to the next level. A score of BELOW AVERAGE or POOR on any one measure indicates a failure to apply at least one of the fundamentals of marksmanship correctly at that level, and therefore the shooter is made to repeat the level. A mean rating of POOR on two measures will cause the shooter to be sent back one level.

A shooter who has progressed to Level 5 has demonstrated the ability to hold the rifle steady, aim correctly, squeeze the trigger without disturbing the lay of the weapon, and, not merely to hit a sufficient number of targets at each level, but to hit them close to center of mass. He has demonstrated this ability when firing in the supported and unsupported positions, and against timed targets in both positions. Therefore, from Level 5 onward the three individual diagnostic measures are dropped, and progress is determined solely by consideration of shot location. The rationale behind this is that, to achieve an adequate score on shot location, the shooter must be applying the fundamentals of marksmanship, and therefore the individual skills are still implicitly being tested. If, at the end of a level, the mean shot location is BELOW AVERAGE, then the shooter is made to repeat the level. If shot location is POOR, then the shooter is sent back one level.

One implication of these standards is that shooters may hit a sufficient number of targets to get through a level but will only pass the level if they have on average also hit close to their centers of mass. In this sense, the standards demanded for record fire in the MACS BRM program are higher than those demanded in actual record fire, where a hit is sufficient.

BRM Program Skill Tests

When new shooters begin on the MACS BRM program, they are automatically taken into hierarchically structured marksmanship skill tests that assess their abilities and determine appropriate starting levels. The process is described below, and is shown as a flow chart in Appendix C. The flow chart for programming is given in Appendix D.

As a first step in the testing process, the shooter is told to fire three rounds from the supported position to establish an initial shot group. These

first three rounds are of key importance to the system because they are used by the computer to perform an electronic equivalent of zeroing the rifle. The system assumes that shooters are aiming at the center of mass of each target to the best of their abilities. It then computes the central point of the three-round shot group, measures the distance between this central point and the actual center of the target, and uses this distance as a constant offset value, which is applied to all subsequent shots. In effect, it moves the shot group onto target electronically rather than by mechanical adjustment of the sights.

From the shooter's viewpoint, the sequence begins with information that states to aim at the center of mass targets, and illustrates what is meant by center of mass. The shooter fires one shot at each of three scaled 250-m, E-type targets. As soon as the three shots have been fired, a feedback screen is shown that gives ratings of steady position, aiming, and trigger squeeze, an assessment of the quality of the three-round shot group (POOR through EXCELLENT) and a close up of the target showing the three-round shot group. The rating of shot group is computed in the same way as that of shot location. The difference in name arises because, for the first three shots, the radial distance of each individual shot is calculated from the shot group center (which is not yet related to target center because the system is in the process of zeroing). Whereas subsequently, radial distances are directly related to the center of mass of targets, and shot location relative to the target can be reported.

As with actual rifle fire, the tighter the shot group the better the zero. The system is able to compute a usable zero offset from a very poor initial shot group, but a limit is set. For each target, a rectangular area is mathematically defined within which all three shots must hit for the shot group to be accepted. This rectangular area would equate to an area 46 cm wide and 24 cm high on a 25-m target. If the shot group is too large to fit within the rectangle, the shooter is told the shot group is invalid, and is invited to try again. If, on the second try, the group is still invalid, either the shooter is making some gross error (such as looking over the rear sight aperture rather than through it) or the light pen is out of alignment with the bore of the rifle and the system requires adjustment. This latter possibility is indicated on screen to cue the instructor, and a program can be accessed from the menu that will let the instructor test and, if necessary, reset the light pen alignment. This program is described subsequently.

Because the system is designed to teach novices, it will accept a very large shot group as valid (within the limits detailed above) and allow the shooter to proceed. The reason for this is that one cannot expect a novice to shoot a tight shot group until the fundamentals of marksmanship have been learned. MACS is designed to teach these fundamentals, and, to state the obvious, the shooter must be able to get into the system in order to learn. However, because the system is also designed to be used by experienced personnel in sustainment training, the shooter is given the option of refiring the first three shots until a tight three-round shot group is achieved. On MACS, a tight shot group is defined as one that is rated as GOOD and is equivalent to placing the three shots in a 3.7 cm circle at 25 m. Although any valid shot group will zero the system adequately for a novice, the experienced shooter may wish to continue to refire until a tight group is achieved. This is because the three-round shot group that the shooter finally

accepts is the one used to zero the system electronically for all future shots.

The shot group the shooter accepts is used also as an initial determinant of skill. If, for the measures of steady position, aiming, trigger squeeze, and shot group, shooters obtain a minimum of two ratings of EXCELLENT and two ratings of GOOD (irrespective of order), they are judged to be competent in firing at non-timed targets in the supported position. Those who do not meet this standard are routed immediately to Level 1 (Introduction to Supported Position). For shooters who meet the standard, a second three-round test is given, using the same targets but engaged from the unsupported position. The same criteria are applied. Those who do not meet the standard are directly routed to Level 2 (Introduction to Unsupported Fire). Shooters who pass the second test take a third test that comprises three timed targets at different ranges (100, 250, and 300 m), the targets to be engaged from the supported position. Failure to meet the standard on the third test takes a shooter to Level 3 (Timed Targets in the Supported Position). Shooters who meet the standard are given a fourth and final test, which is identical to the third, except that it is fired from the unsupported position. Failure to meet the standard in this test routes the shooter to Level 4 (Timed Targets in the Unsupported Position), and success takes the shooter to Level 5 (Practice Record Fire).

The four tests are hierarchically structured to select the very able shooters. The poor shooter and the novice never know that they have been tested or that the tests exist; they simply establish a shot group and go straight to Level 1. The standards demanded in the tests are deliberately set at a high level, so that only those who have no need of additional training or practice in the fundamentals of marksmanship can move straight to practice record fire. Although the system will automatically direct a shooter to a starting level that the system's internal logic deems appropriate, the instructor can override the system. If the instructor feels that a level is inappropriate, the menu can be accessed, and the shooter moved to a different level.

BRM Program Menu and Options

The program contains a menu that allows a number of additional programs and special features to be called into use. These are described below.

Light Pen Mount Alignment Program

The light pen is mounted to the barrel of the M16 demilitarized rifle by a mounting bracket. This bracket holds the light pen securely and in alignment with the bore of the weapon. Accident or rough handling can disturb the alignment, and therefore a program is provided to allow the instructor to check the alignment and realign if necessary. The alignment program provides an on-screen aiming point and a target area. The point at which the light pen is reading is shown by a black dot on the screen. If the black dot is in the target area when the sights are placed on the aiming point, and if the black dot tracks smoothly across the screen as the weapon is moved, then the light pen is correctly aligned. If the light pen is incorrectly aligned then the mounting screws must be loosened and the light pen position adjusted. The

procedures for this may be found in the Trainer's Guide (Purvis & Wiley, 1989).

Sight Alignment, Aiming, and Grouping Program

The program is designed to teach three basic concepts; these are the correct alignment of the front and rear sights, the correct placement of the front sight post on the target, and the meaning of a tight shot group. Programming flow charts for this program may be found in Appendix E. The program is divided into four parts.

The first part provides a demonstration of the correct alignment of front and rear sights, which is followed by a self-paced test in which a series of sight alignments are shown on the screen and the soldier must determine whether alignment is correct. The soldier responds by pulling the trigger on the MACS rifle when the appropriate response is highlighted on the screen. Feedback is given after each decision. The feedback acknowledges correct responses and, when responses are incorrect, tells the soldier why the answer was wrong. Soldiers must give three consecutive correct answers to move on to the next part of the program.

The second part of the program provides a demonstration of correctly aligned sights moving onto a scaled 250-m, E-type target. The demonstration is supported by on-screen text and is iterative, continuing until the soldier understands and is ready to proceed. The demonstration is followed by a self-paced test, which is structured exactly like that for sight alignment.

The third part is a test, in which a series of randomly generated sight pictures are shown. One in three of these sight pictures is correct; the remainder have poor sight alignment and/or incorrect placement of the front sight post on the target. As with the previous two tests, the soldier indicates correct or incorrect using the trigger as a response button. Feedback is given after each decision, and the shooter must give three consecutive correct answers to complete the test.

Although the primary purpose of the program is to teach concepts, because precision of sight placement is required in marksmanship, fine judgments are demanded of the soldier in the tests. The soldier is required to detect errors in sight placement that range between 1 and 3 pixels in the X axis and 1 and 2 pixels in the Y axis. At 3 m, the height of a pixel (Y axis) subtends an angle at the eye of 1.1 minutes of arc, and the width of a pixel (X axis) subtends 0.84 minutes of arc. Of the two, the Y axis displacement is the more difficult to resolve because, on occasions, the soldier must hold an image of the target in memory when making a judgment. This is because the image of the front sight post may obscure the lower part of the target. In the X axis, the soldier must discern the degree of lateral displacement and can use the sides of the target and the sides of the sight as referents. The 0.84 minutes of arc minimum displacement is well within human capabilities; indeed, Berry, Riggs, & Duncan (1950) have shown that discriminations of lateral displacement (vernier acuity) as small as 6 seconds of arc are possible even with low background luminance levels.

The fourth part of the program is designed to help the novice understand the difference between a tight shot group and a large shot group. It does

this by showing demonstrations of three-round shot groups hitting the image of a 25-m zeroing target. The referent taught is the 4-cm circle in the center of the target. Once the soldier has seen the demonstration, a self-paced test is given in which the soldier is asked to judge whether or not a series of shot groups are tight. Feedback is given after each decision. The feedback either acknowledges a correct response or, in the case of an incorrect response, demonstrates whether the three-round shot group would fit inside the 4-cm circle. The soldier must answer three consecutive questions correctly to complete the program.

The MACS BRM program has a number of optional features that an instructor can elect to use with some levels of the program. These are given on a menu that the instructor can access at the keyboard and are briefly described below.

Wind Speed

The program allows an instructor to set wind speed at 0, 10, or 20 mph, and wind direction relative to the firing range (across the range, oblique to the range, or in line with the range). This option may be selected for any and all levels of fire. To help the instructor teach soldiers the correct sight placement, a teaching screen appears at the start of the shooting sequence, which shows the correct sight placement on targets at 50, 150, and 300 m for the direction and speed of wind selected.

Call Your Shot

This option applies only to Levels 1 and 2. For each shot, it builds in a 5-s time delay (with on-screen counter) before feedback and bullet strike are shown. This allows the shooter time to announce where they believe the bullet strike will be.

Targets at 250 M

This option applies to Level 1 only. It allows the instructor to set all 18 targets to 250 m, so that the shooter can practice at a single range. It is intended for use with novice shooters who are experiencing difficulty at Level 1, and it parallels the ability instructors have in live fire of moving a novice to a 25-m range for practice. The 25-m target is a scaled-down, 250-m target, as is the target on the MACS screen, so that the two equate.

Discussion

The development of the MACS BRM program was evolutionary rather than entirely planned. It began as a diagnostic tool to help instructors determine what a shooter was doing wrong. This early program resembled Level 1 of the present BRM program. Subsequently, the diagnostic program was transformed into a teaching program. The transformation process was gradual, and as a result the system received considerable testing during development. In the process, less advanced versions than that described in this paper were demonstrated to be effective in improving rifle marksmanship (Evans, 1988).

The transformation from diagnostic tool to training device obviously necessitated many changes. For example, a new measure, aiming, was introduced

because aiming is one of the four fundamentals of marksmanship taught by the Army. New levels were added to take the soldier through to record fire standard, and the standards themselves were tightened so that they paralleled those demanded of soldiers on actual ranges. Because these changes happened sequentially rather than as a planned single modification, the reader may have formulated some questions on developmental methodology. For example, it is reasonable to ask why, given the new training role, the criteria for rating performance are still based on a modified version of trainees performance, and why no new data from trained soldiers have been collected. The answer is twofold. First, time constraints did not permit additional experimentation. Second, and from a purely practical viewpoint, the present system works, and there is no reason to believe that the quality of teaching the system provides will be significantly improved if additional research is done.

Summary and Recommendations

As has been noted, earlier versions of the MACS BRM program have been demonstrated to improve soldier performance in marksmanship. The present system is an enhanced version that reduces the load on the instructor, has greater flexibility in that the experienced user can now establish a shot group before starting, and is more in accord with Army standards for marksmanship. However, it has not been formally validated. Indications from both in-house SMEs and U.S. Army Marksmanship Unit personnel (n=6) are that the feedback the system provides is both realistic and appropriate. From this we may conclude that the current version has face validity. If there is a need to prove to the Army that soldiers trained, in part on MACS BRM will learn more rapidly, or if there is a need to demonstrate the degree of improvement that may be gained by using the system, then it is recommended that a new system validation be undertaken.

References

Berry, R. N., Riggs, L. A., & Duncan, C. P. (1950). The relation of vernier and depth discrimination to field brightness. Journal of Experimental Psychology, 40, 349.

Bilodeau, E. A., & Bilodeau, I. McD. (1958). Variable frequency of knowledge of results and the learning of a simple skill. Journal of Experimental Psychology, 55, 379-383.

Bilodeau, I. McD. (1966). Information feedback. In E. A. Bilodeau, (Ed.), Acquisition of skill (pp. 255-296). New York: Academic Press.

Bryan,G. L. & Regan, J. J. (1972). Training system design. In H. P. Van Cott, & R. G. Kincade, (Eds.) Human engineering guide to equipment design (pp. 633-666). Washington: U.S. Government Printing Office.

Department of the Army (1988). Final draft: M16A1 and M16A2 rifle marksmanship. Washington, DC: Author.

Evans, K. L. (1988). Development and evaluation of the multipurpose arcade combat simulator (MACS): A research summary (ARI Research Report 1488). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (DTIC No. AD-B130 099)

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. (DTIC No. AD-A 184 179)

Lavery, J. J. (1962). Retention of simple motor skills as a function of type of knowledge of results. Canadian Journal of Psychology, 16, 300-311.

Osborne, A. D., & Smith, S. (1985). Unit rifle marksmanship training guide (ARI Research Product 85-12). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (NTIS No. AD-B 105 422)

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

Smith, K. U., & Smith, M. F. (1966). Cybernetic principles of learning and educational design. New York: Holt, Rinehart & Winston.

Stammers, R., & Patrick, J. (1975). The psychology of training. London: Methuen.

U.S. Army Infantry School. (1984). Unit rifle marksmanship training guide (Field Circular No. 23-11). Fort Benning, GA: Author.

APPENDIX A

Summary Description of Levels

Level 1. Introduction to Supported Position

Summary description

- No time limit
- Supported position
- Single-target presentations
- Eighteen targets presented, three targets at each of six ranges
- Targets engaged in order of range: 50, 100, 150, 200, 250, and 300 m

Feedback

- Words HIT or MISS appear on screen for each shot
- Cross appears on screen to show bullet strike
- Diagnostics and replay are shown after each shot
- Screen shows three rounds on single target after each group of three shots
- Summary screen at end of level shows average diagnostic scores

Standards

- Two hits out of three shots at each range within level. Failure to meet standard at a given range results in immediate refire of 3 more targets at that range. Refires continue until standard is met
- Refire of level is required if any score on summary screen is BELOW AVERAGE or POOR

Level 2. Introduction to Unsupported Position

Identical to Level 1, with these exceptions:

- It is fired in the unsupported position
- Shooter is regressed a level if any 2 scores on summary screen are POOR

Level 3. Timed Targets in Supported Position

Summary description

- Time limit approximately one-and-one-third times that for record fire
- Supported position
- Single target presentations
- Twenty-four targets in random sequence (four presentations per range)

Feedback

- Cross appears on screen to show bullet strike
- For a hit, target disappears immediately after bullet strike is shown
- Diagnostics and replay are shown after bad shots only. Criteria for bad shot is a POOR for any score or a BELOW AVERAGE for shot location
- Audible tone when target exposure time limit has expired
- Summary screen at end of level showing average diagnostic scores
- Summary screens showing four shots at each range on single targets

Standards

- Three hits out of four shots at each range within level. For each range at which this standard is not met, a refire of four targets at each of those ranges is required. Refires continue until standard is met
- Refire of level is required if any score on summary screen is BELOW AVERAGE or POOR
- Shooter is regressed a level if any two scores on summary screen are POOR

Level 4. Timed Targets in Unsupported Position

Identical to Level 3, with these exceptions:

- It is fired in the unsupported position

- Standard is relaxed. Three hits out of four shots at 50, 100, 150, and 200 m, and two hits out of four shots at 250 m and 300 m

Level 5. Practice Record Fire I

Summary description

- Time limit same as for record fire
- Supported and unsupported positions
- Single or double presentations
- Forty targets in random sequence, 20 fired supported and 20 unsupported.

Feedback

- Cross appears on screen to show bullet strike (hit and miss)
- For a hit, target disappears after bullet strike is shown
- Audible tone when target exposure time limit has expired
- OUT OF AMMO appears on screen when 20 shots have been fired
- Interim summary of shot location (hits, misses, and no fires) given at end of each position and final summary at end of course of fire
- Summary screens showing shots at each range on a single target at end of supported position and at end of unsupported position
- Performance graded as Expert, Sharpshooter, Marksman, Unqualified

Standards

- Fifteen hits out of 20 shots in both supported and unsupported positions. Refire of position if standard not met
- On double target exposures, a penalty is recorded if further target is engaged first
- Regress one level for average shot location score of POOR. Stay at same level for mean shot location score of BELOW AVERAGE. Otherwise pass

Level 6. Practice Record Fire II

Identical to Level 5, with the exception that the cross showing bullet strike appears only after misses. Target disappears when hit.

Level 7. Record Fire

Summary description

- Time limit same as for record fire
- Supported and unsupported positions
- Single or double presentations
- Forty targets in random sequence, 20 fired supported and 20 unsupported

Feedback

- Targets disappear when hit
- OUT OF AMMO appears on screen when 20 shots have been fired
- Audible tone when target exposure time limit has expired
- Interim summary of shot location (hits, misses, and no fires) given at end of each position and final summary at end of course of fire
- Summary screens showing shots at each range on a single target at end of supported position and at end of unsupported position
- Performance graded as Expert, Sharpshooter, Marksman, Unqualified
- Cross appears on screens to show bullet strike for misses, whenever soldier has to refire level

Standards

- Fifteen hits out of 20 shots in both supported and unsupported positions. Refire of position if standard not met
- On double target exposures, a penalty is recorded if further target is engaged first
- Regress one level for average shot location score of POOR. Stay at same level for mean shot location score of BELOW AVERAGE. Otherwise pass

Level 8. Rapid Record Fire I

Identical with Level 7, except that the exposure times of targets are two-thirds of those used in record fire.

Level 9. Combat Fire

Summary description

- Time limits variable, dependent on number of targets presented
- Supported and unsupported positions
- Single, double, and multiple presentations
- Eighty targets in two attack/retreat scenarios each of 40 targets. First scenario is fired supported and second unsupported

Feedback

- Targets disappear when hit
- OUT OF AMMO appears on screen when 40 shots have been fired
- Summary of shot location (hits, misses, and no fires) given at end of supported position and at end of unsupported position
- Summary screens showing shots at each range on a single target at end of supported position and at end of unsupported position
- Performance graded as Expert, Sharpshooter, Marksman, Unqualified
- Cross appears on screens to show bullet strike for misses, whenever soldier has to refire level

Standard

- Twenty-three hits out of 40 shots in both supported and unsupported positions. Refire of position if standard not met

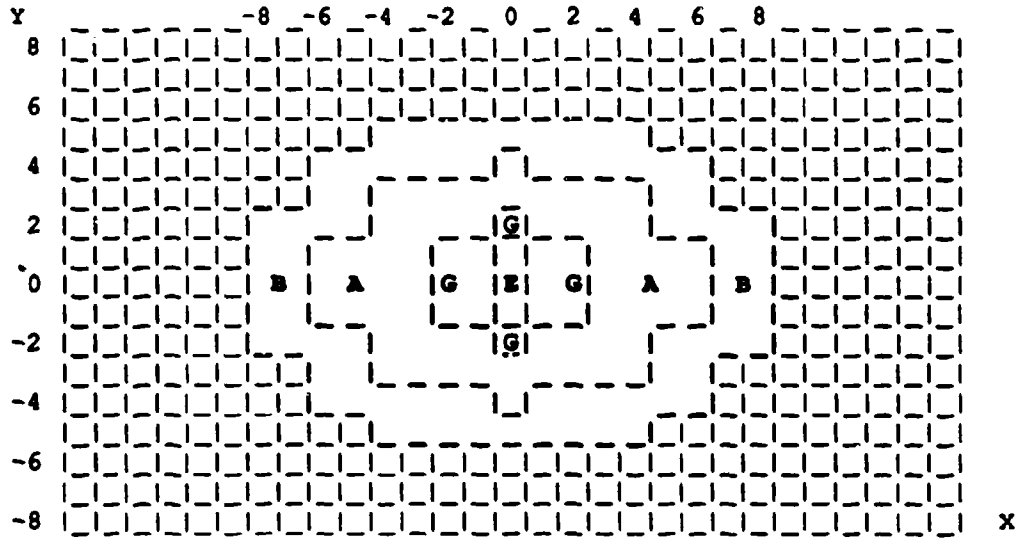
APPENDIX B

Criteria Target Areas for Aiming and Shot Location

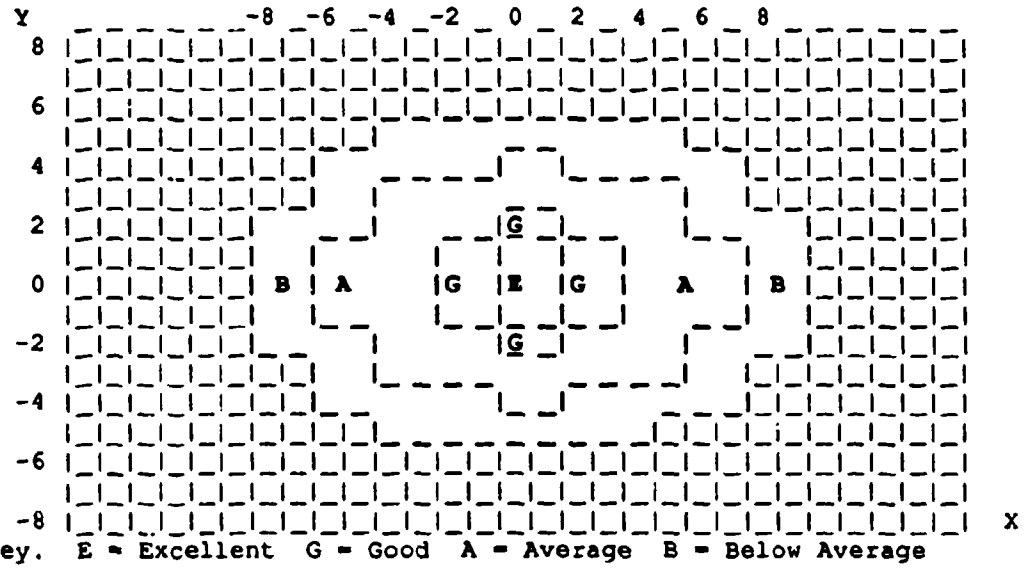
The target area that is mathematically defined for the aiming and shot location scores differs from the effective target area that was shown in the main text. For comparison purposes, both are shown in Figure B-1. The reason that the effective target area is different is as follows. The computer stores the X and Y coordinates of a pixel location each as a single byte. One byte can hold the record of any value between 0 and 255. The screen is 200 pixels high, and therefore each pixel location in the Y axis can have a separate 1 byte address. However, this is not possible in the X axis because the screen is 320 pixels wide. Because only one byte is allocated to describe the X axis location, the computer divides the number of the pixel by two before storing it. Thus, pixel number 300 is stored as pixel number 150, and when the pixel number is retrieved from memory, its stored value is doubled to recreate its original value. Unfortunately, when a number is halved and stored, it is stored as an integer so that any fractional values are lost. Thus, pixel number 301 is not stored as 150.5, but as 150; when it is retrieved, it is doubled and becomes 300. Therefore, if the light pen is focused on an odd-numbered pixel in the X axis, it is reported by the computer as the value of the pixel immediately to its left.

This system peculiarity has two implications: First, targets must be (and are) located with their centers of mass on even-numbered pixels in the X axis (were it otherwise, a score of excellent would be impossible to obtain). Second, the target areas are distorted compared with those that were mathematically defined. In defining target areas that scale to actual areas on the 25-m zeroing target, this distortion was taken into account.

Mathematically Defined Target Area (Supported Position)



Effective Target Area (Supported Position)



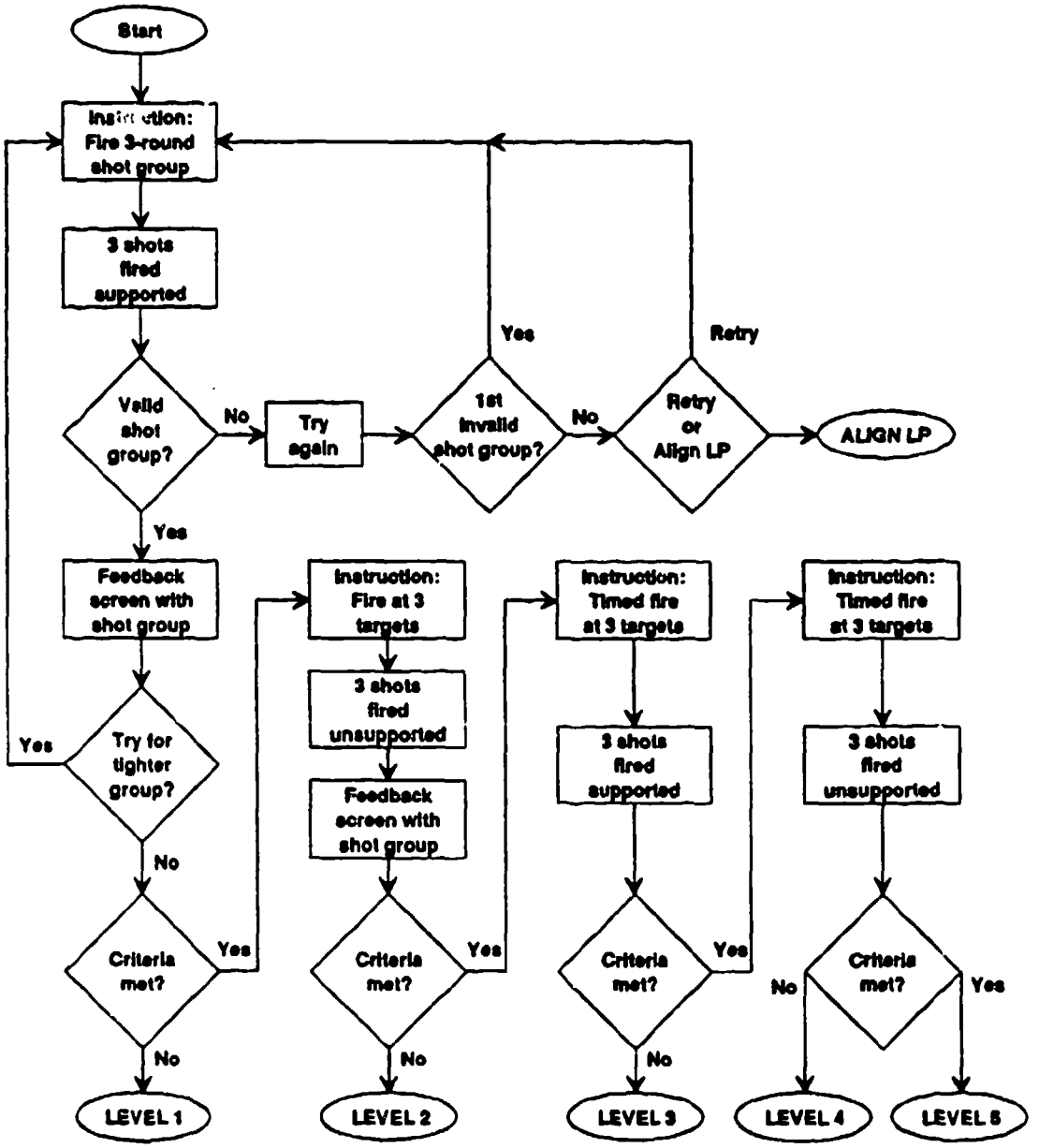
Key. E = Excellent G = Good A = Average B = Below Average
Shots located outside the indicated areas are rated POOR

Figure B-1. Comparison between mathematically defined target area, and the effective target area.[4]

[4]The effective target area is adjusted for the fact that the computer reports readings from an odd-numbered pixel in the X axis as having the same value as the pixel immediately to its left. Actual pixels are longer in the Y axis than in the X axis (approximate ratio is 1.33:1); therefore, the on-screen target area is more circular in shape than is shown.

APPENDIX C

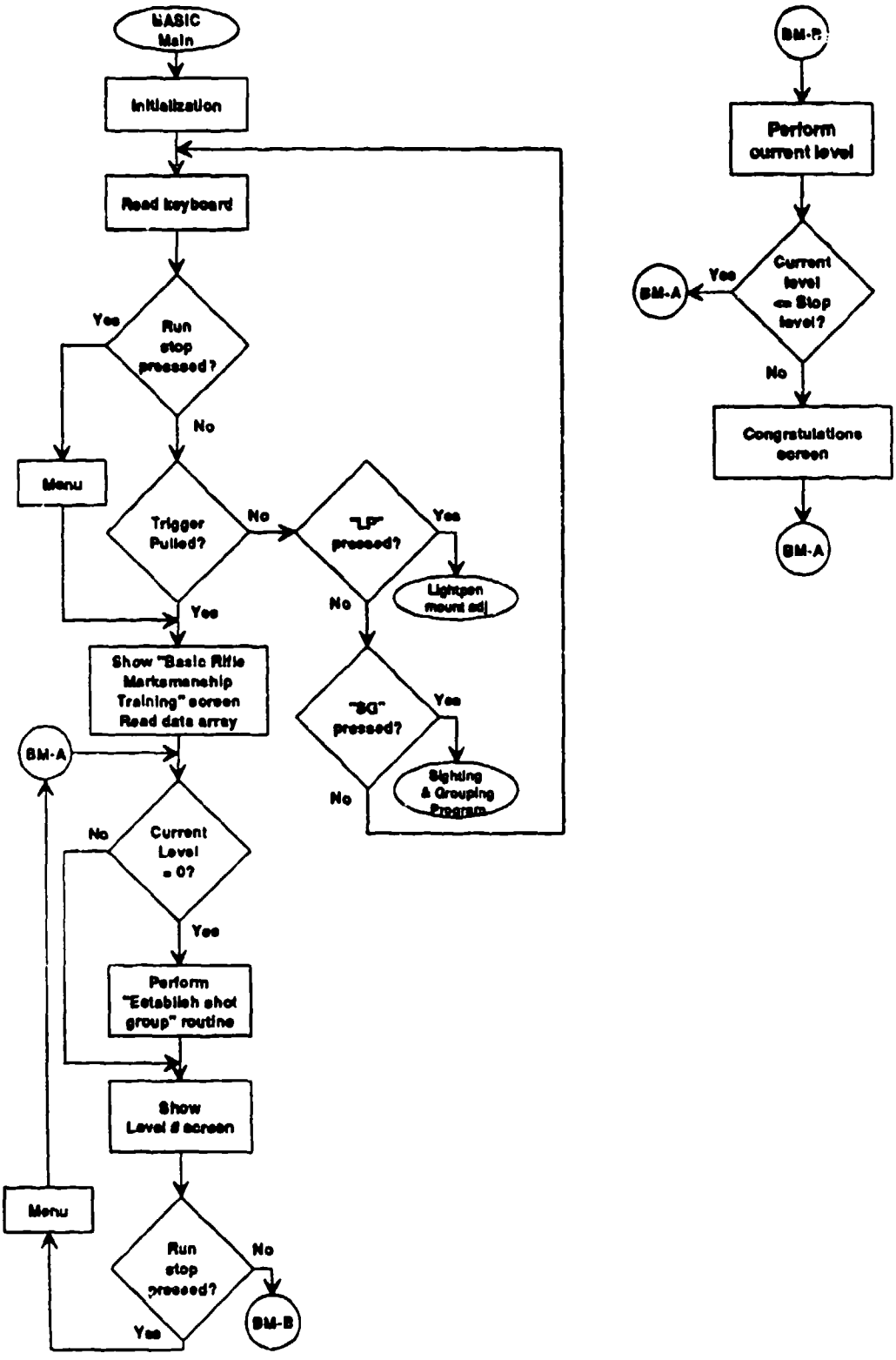
Users' Flow Chart for the Marksmanship Skill Tests[1]

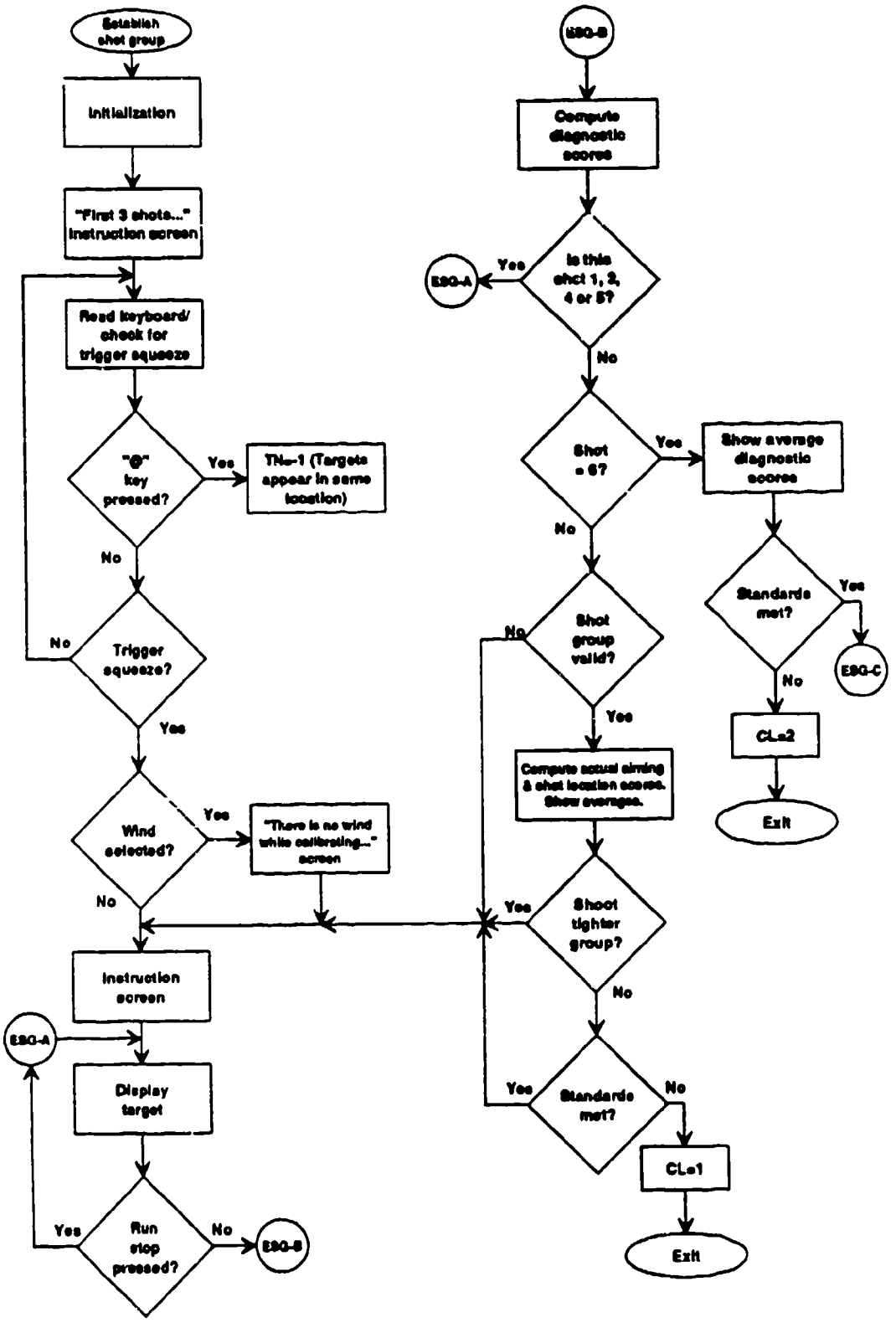


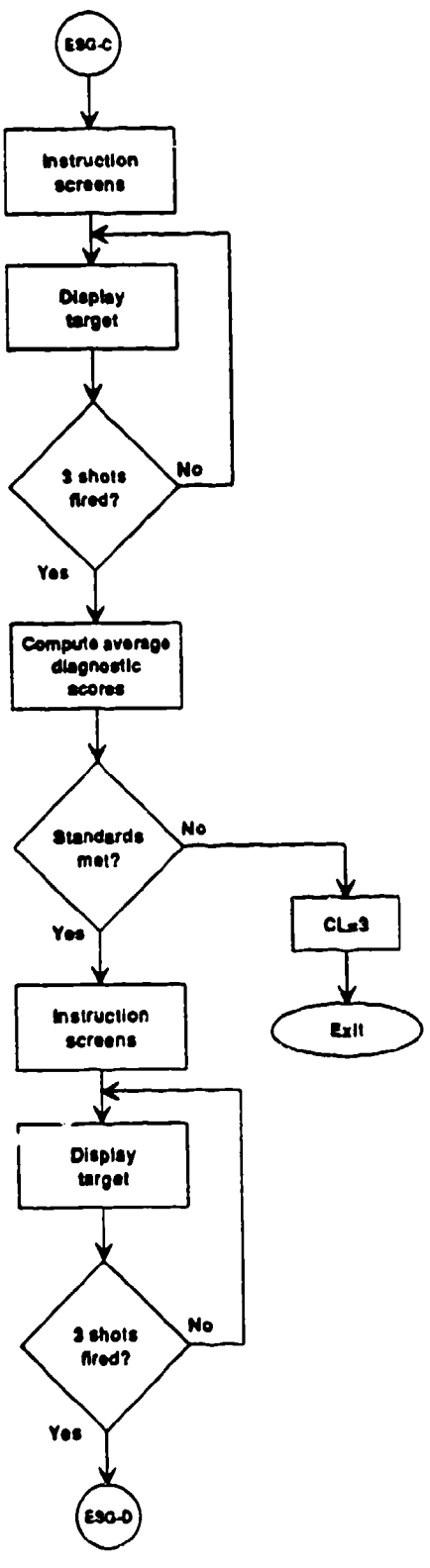
[1]Initial Skill Test Criteria - For the measures of shot location, steady position, aiming, and trigger squeeze, the firer must earn two ratings of GOOD and two ratings of EXCELLENT to pass each test. The firer does not see ratings after the second test.

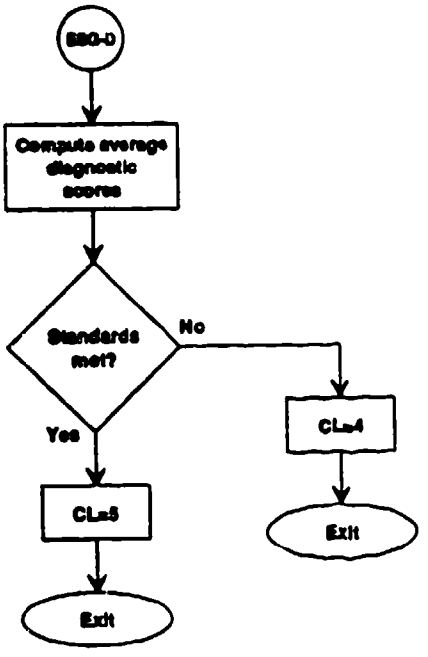
APPENDIX D

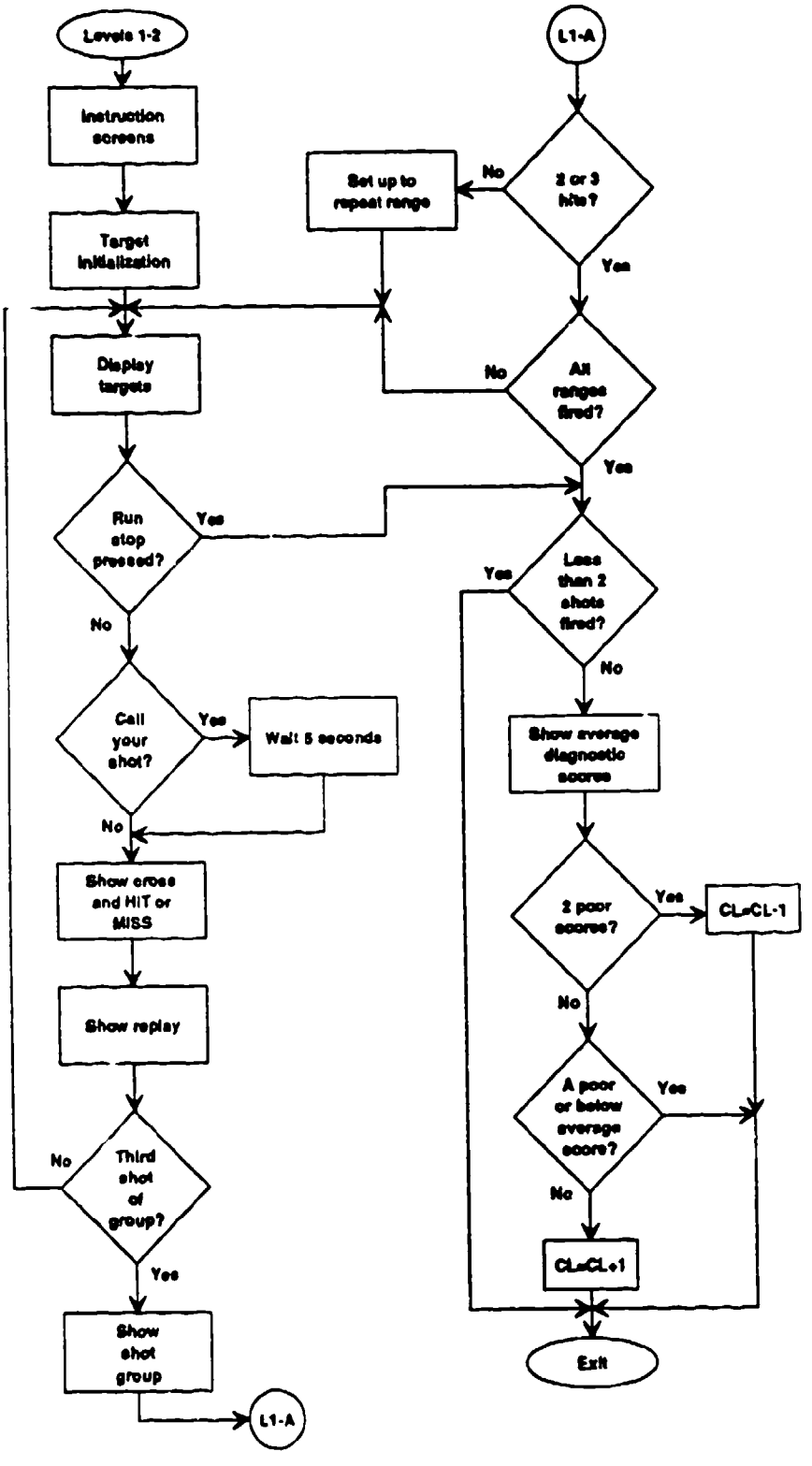
Programming Flow Chart for the BRM Program

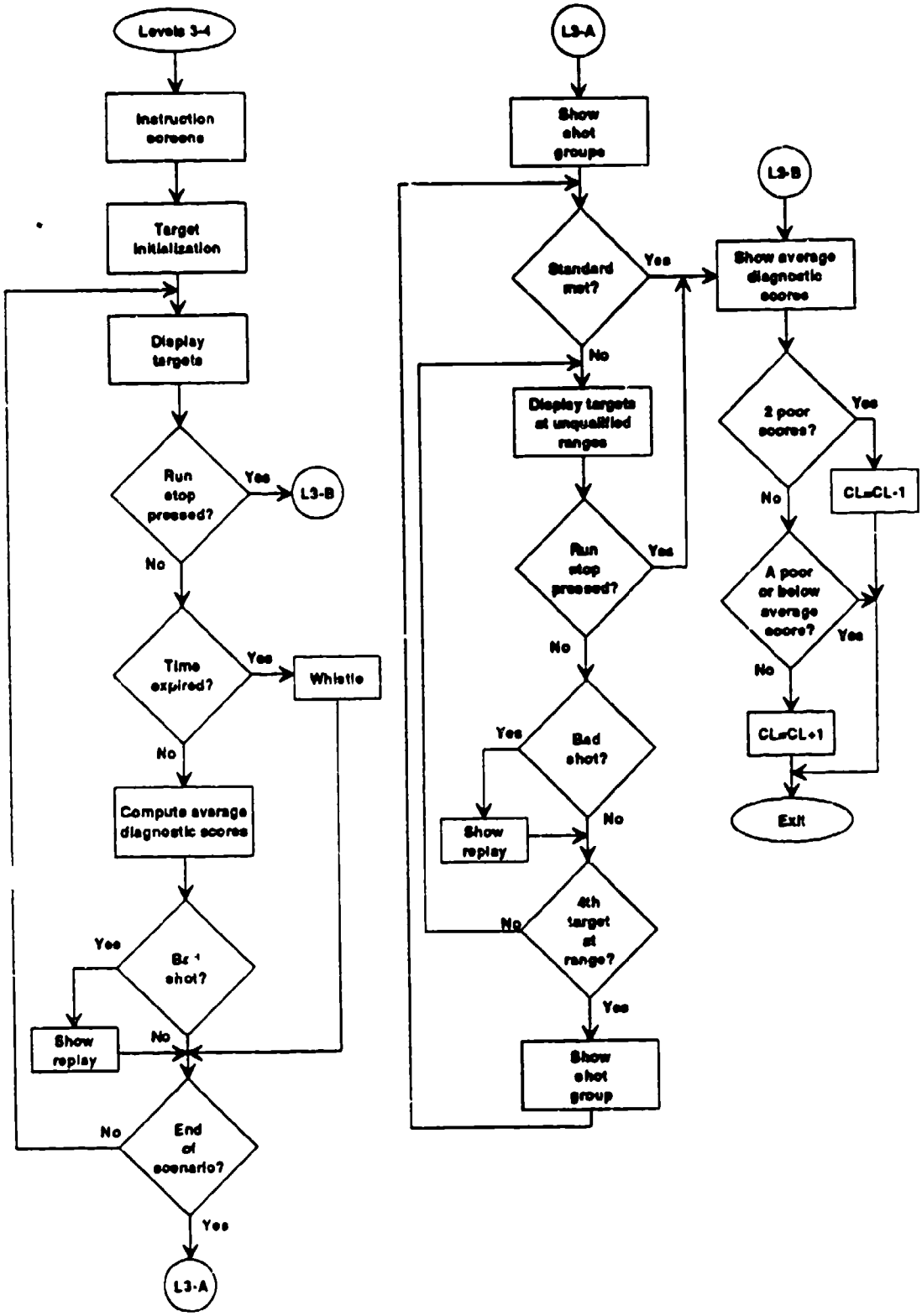


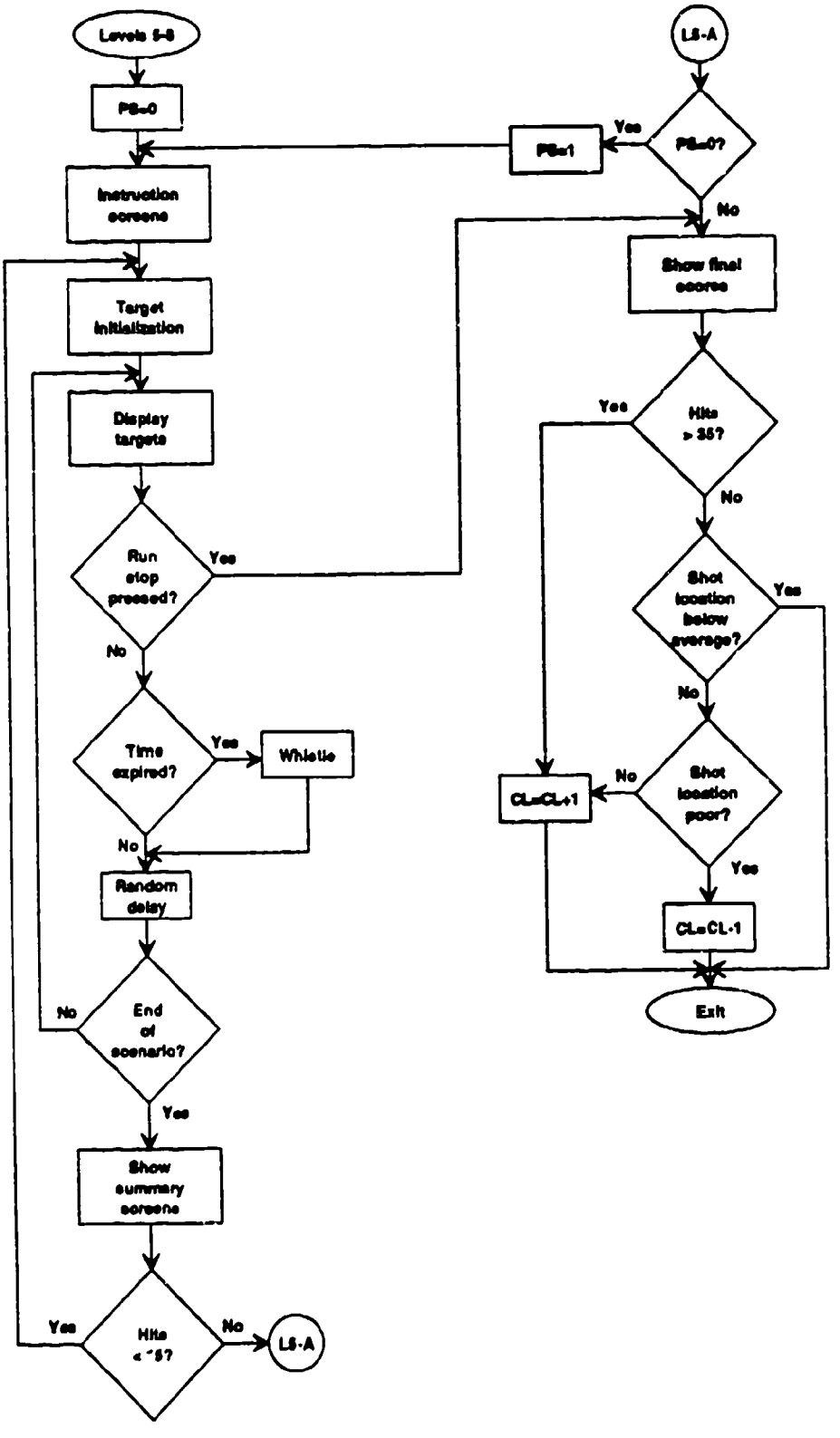

Establish shot group
Initialization
"First 3 shots..." instruction screen
Read keyboard/ check for trigger squeeze
"@" key pressed?
Yes
TN=1 (Targets appear in same location)
No
Trigger squeeze?
No
Yes
Wind selected?
Yes
"There is no wind while calibrating..." screen
No
Instruction screen
ESC-A
Display target
Run stop pressed?
Yes
No
ESC-B
ESC-B
Compute diagnostic scores
Is this shot 1, 2, 4 or 5?
Yes
ESC-A
No
Shot = 6?
Yes
Show average diagnostic scores
No
Shot group valid?
No
Yes
Compute actual aiming & shot location scores. Show averages.
Shoot tighter group?
Yes
No
Standards met?
Yes
No
CL=1
Exit
Standards met?
Yes
ESC-C
No
CL=2
Exit

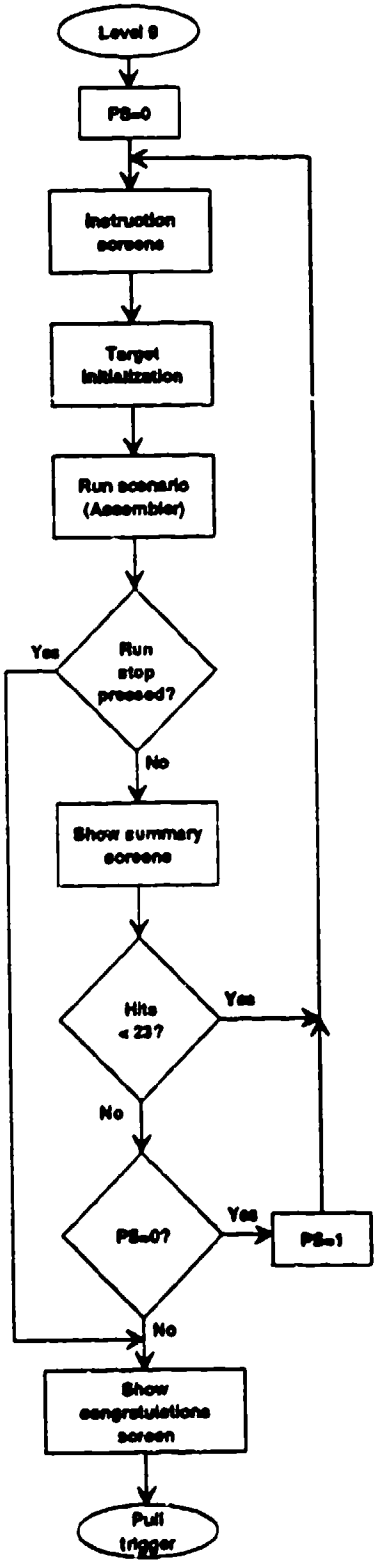

ESQ-C
Instruction screens
Display target
3 shots fired?
No
Yes
Compute average diagnostic scores
Standards met?
No
CL=3
Exit
Yes
Instruction screens
Display target
3 shots fired?
No
Yes
ESQ-D

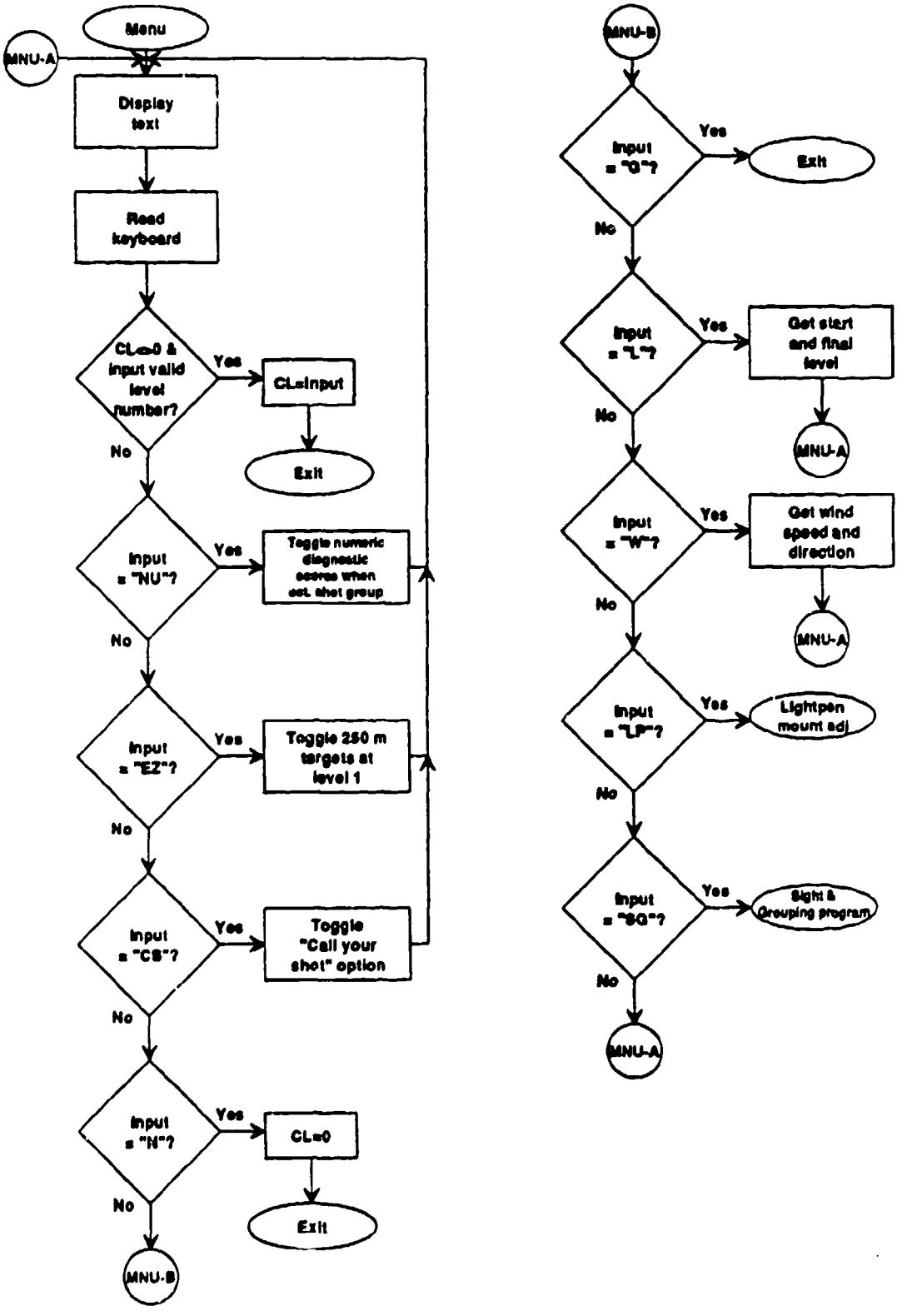

ESQ-D
Compute average diagnostic scores
Standards met?
No
CL=4
Exit
Yes
CL=5
Exit

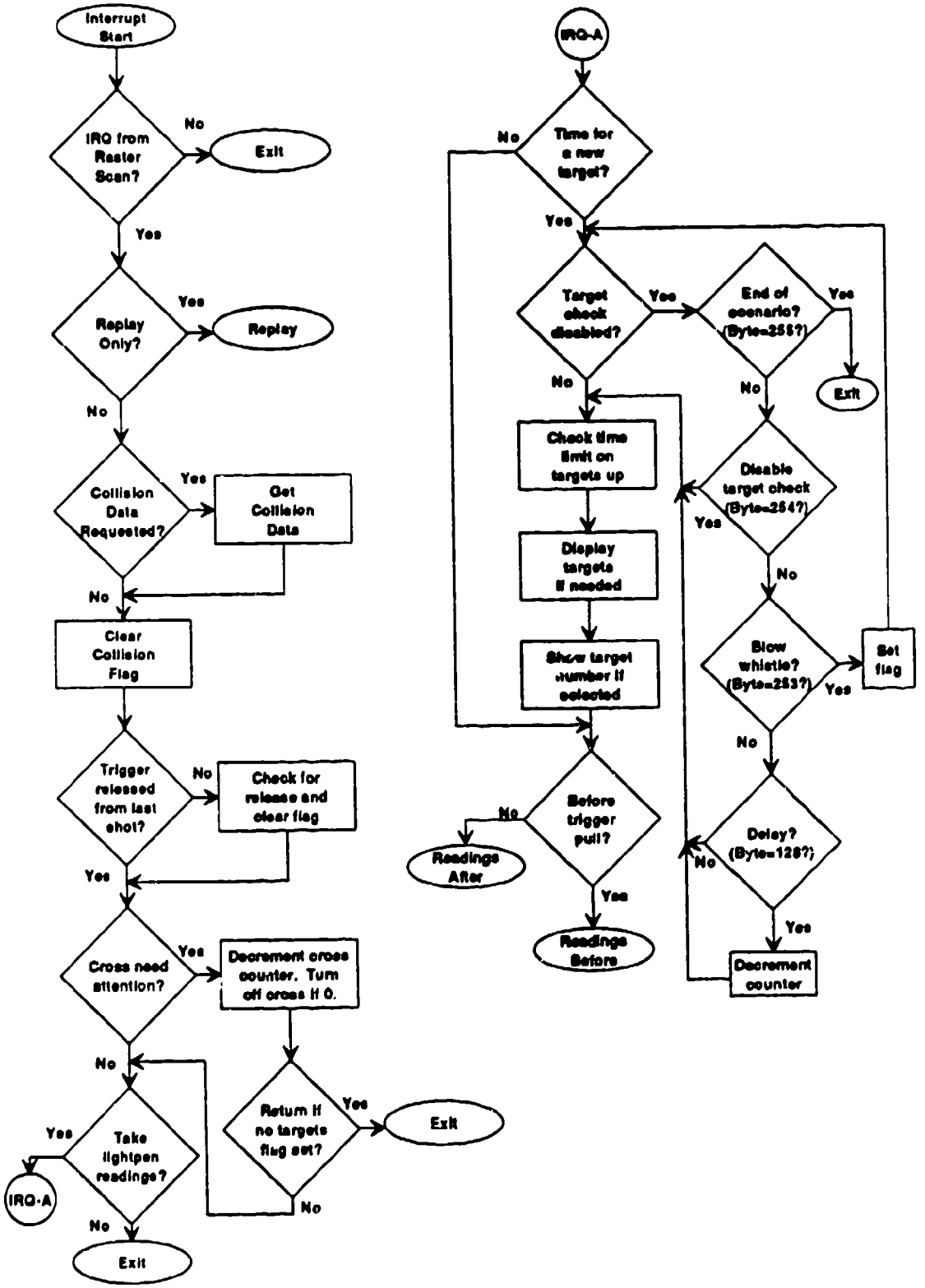

Levels 1-2
Instruction screens
Target initialization
Display targets
Run stop pressed?
Yes
No
Call your shot?
Yes
Wait 5 seconds
No
Show cross and HIT or MISS
Show replay
Third shot of group?
No
Yes
Show shot group
L1-A
L1-A
2 or 3 hits?
No
Set up to repeat range
Yes
All ranges fired?
No
Yes
Less than 2 shots fired?
Yes
No
Show average diagnostic scores
2 poor scores?
Yes
CL=CL-1
No
A poor or below average score?
Yes
No
CL=CL+1
Exit

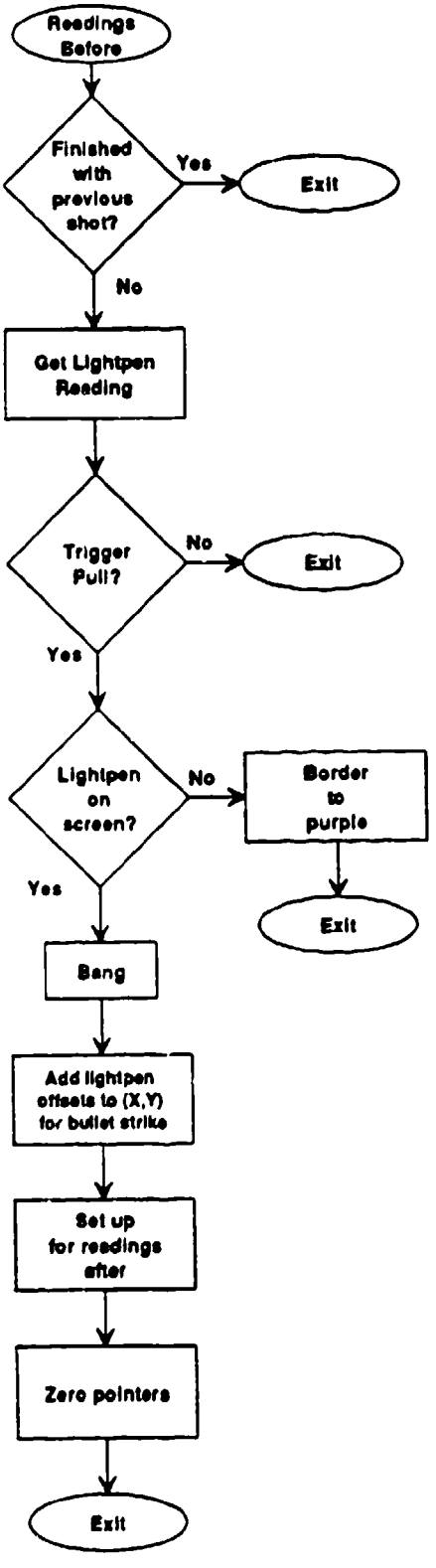

Levels 3-4
Instruction screens
Target initialization
Display targets
Run stop pressed?
Yes
L3-B
No
Time expired?
Yes
Whistle
No
Compute average diagnostic scores
Bad shot?
Yes
Show replay
No
End of scenario?
No
Yes
L3-A
L3-A
Show shot groups
Standard met?
Yes
No
Display targets at unqualified ranges
Run stop pressed?
Yes
No
Bad shot?
Yes
Show replay
No
4th target at range?
No
Yes
Show shot group
L3-B
Show average diagnostic scores
2 poor scores?
Yes
CL=CL-1
No
A poor or below average score?
Yes
No
CL=CL+1
Exit

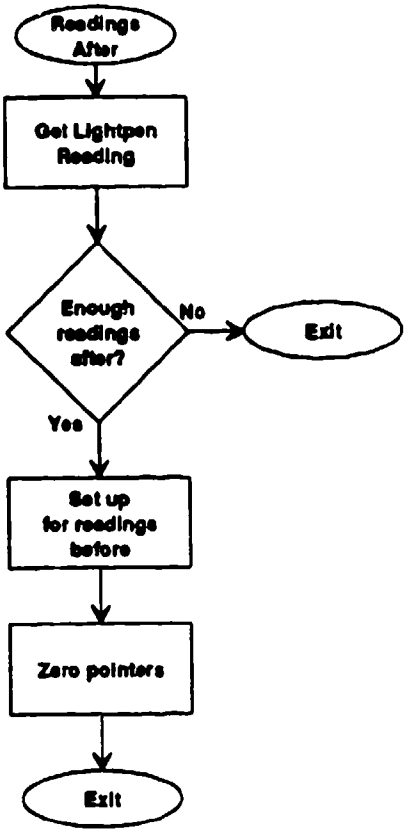

Levels 5-8
PS=0
Instruction screens
Target initialization
Display targets
Run stop pressed?
Yes
No
Time expired?
Yes
Whistle
No
Random delay
End of scenario?
No
Yes
Show summary screens
Hits < 55?
Yes
No
LS-A
LS-A
PS=0?
Yes
PS=1
No
Show final scores
Hits > 35?
Yes
No
Shot location below average?
Yes
No
Shot location poor?
No
CL=CL+1
Yes
CL=CL-1
Exit

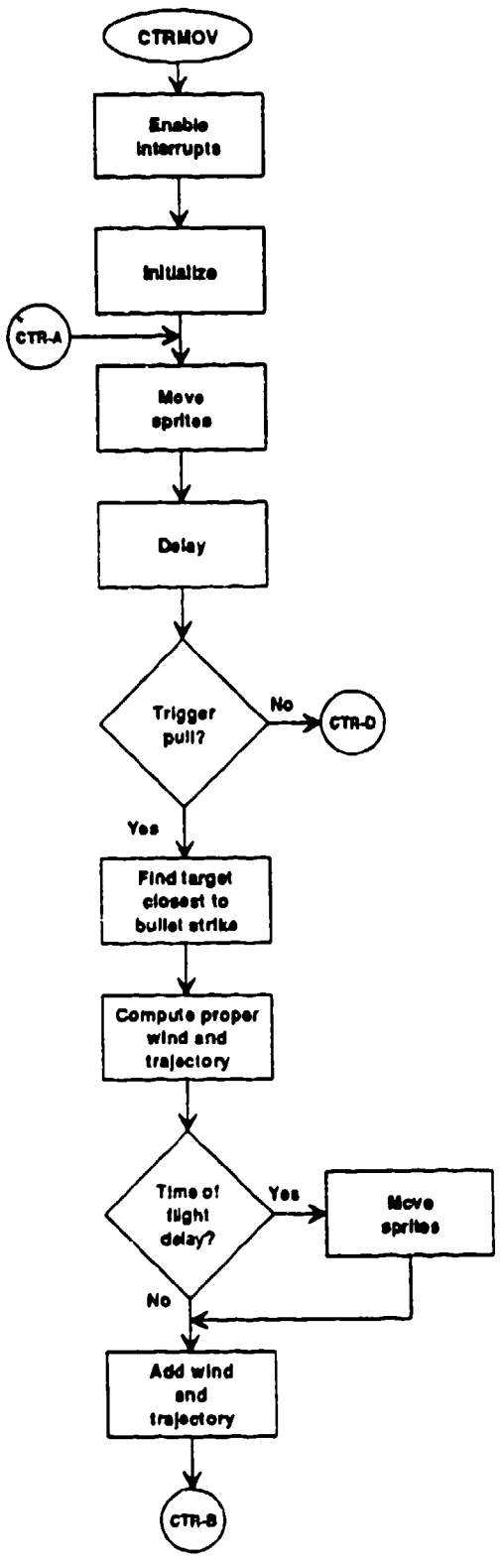

Level 9
PS=0
Instruction screens
Target initialization
Run scenario (Assembler)
Run stop pressed?
Yes
No
Show summary screens
Hits < 23?
Yes
No
PS=0?
Yes
PS=1
No
Show congratulations screen
Pull trigger

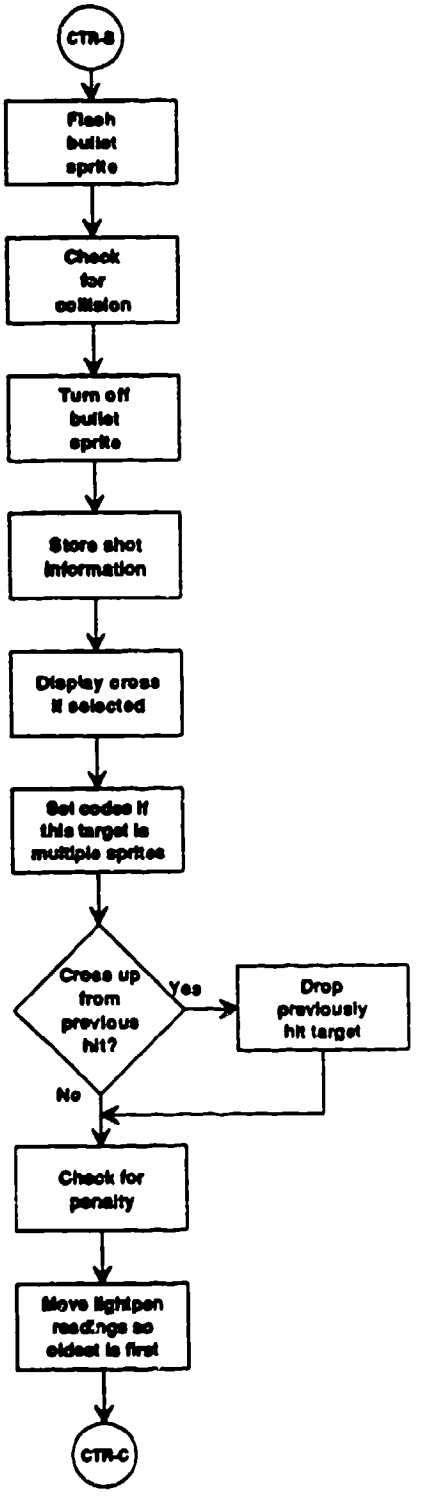

Menu
MNU-A
Display text
Read keyboard
CL=0 & input valid level number?
Yes
CL=input
Exit
No
Input = "NU"?
Yes
Toggle numeric diagnostic scores when est. shot group
No
Input = "EZ"?
Yes
Toggle 250 m targets at level 1
No
Input = "CS"?
Yes
Toggle "Call your shot" option
No
Input = "N"?
Yes
CL=0
Exit
No
MNU-B
MNU-B
Input = "Q"?
Yes
Exit
No
Input = "L"?
Yes
Get start and final level
MNU-A
No
Input = "W"?
Yes
Get wind speed and direction
MNU-A
No
Input = "LP"?
Yes
Lightpen mount adj
No
Input = "SG"?
Yes
Sight & Grouping program
No
MNU-A

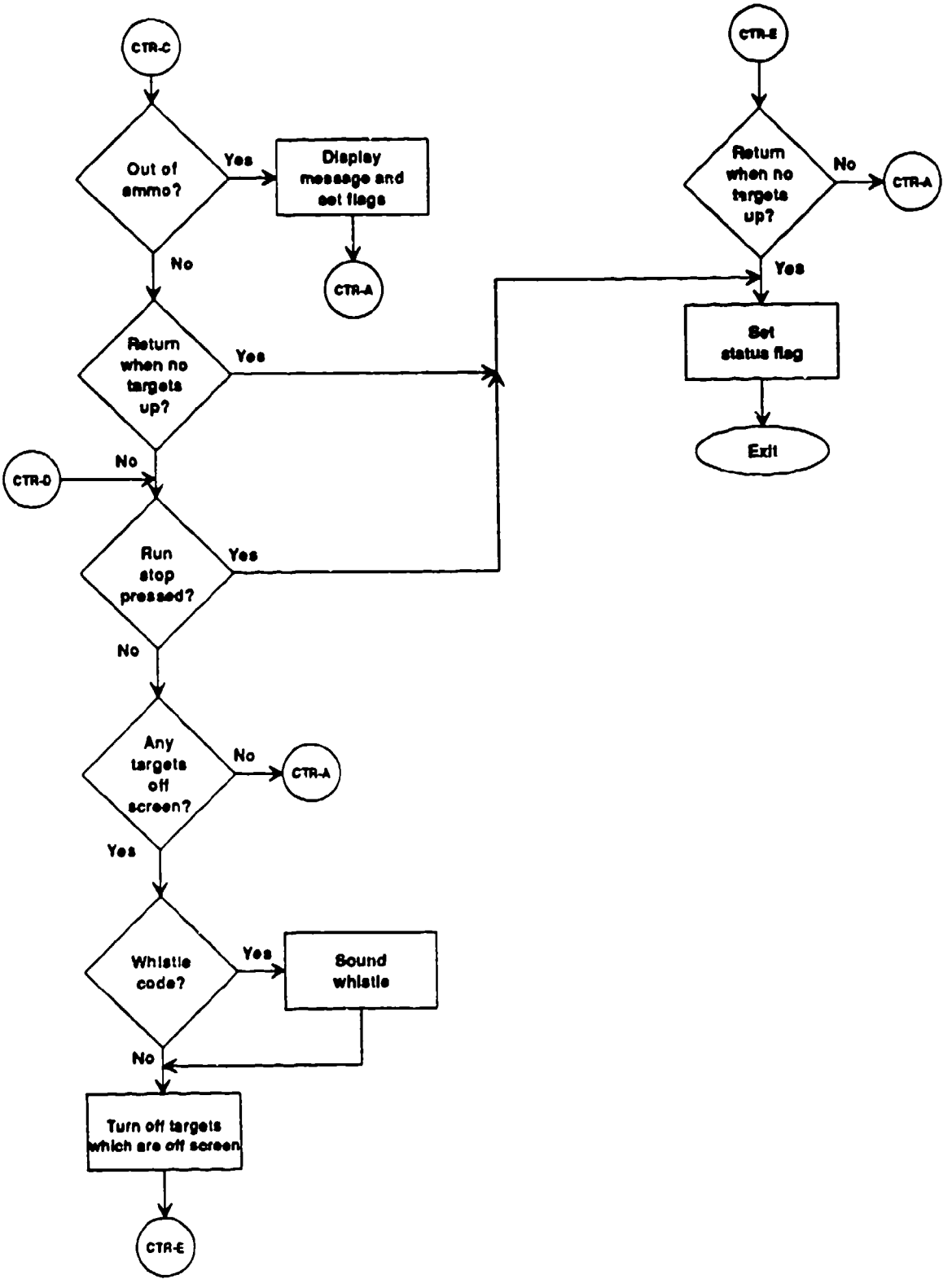

Interrupt Start
IRQ from Raster Scan?
No
Exit
Yes
Replay Only?
Yes
Replay
No
Collision Data Requested?
Yes
Get Collision Data
No
Clear Collision Flag
Trigger released from last shot?
No
Check for release and clear flag
Yes
Cross need attention?
Yes
Decrement cross counter. Turn off cross if 0.
No
Take lightpen readings?
Yes
IRQ-A
No
Exit
Return if no targets flag set?
Yes
Exit
No
IRQ-A
Time for a new target?
No
Yes
Target check disabled?
Yes
End of scenario? (Byte=255?)
Yes
Exit
No
No
Check time limit on targets up
Display targets if needed
Show target number if selected
Disable target check (Byte=254?)
Yes
No
Blow whistle? (Byte=253?)
Yes
Set flag
No
Delay? (Byte=128?)
No
Yes
Decrement counter
Before trigger pull?
No
Readings After
Yes
Readings Before

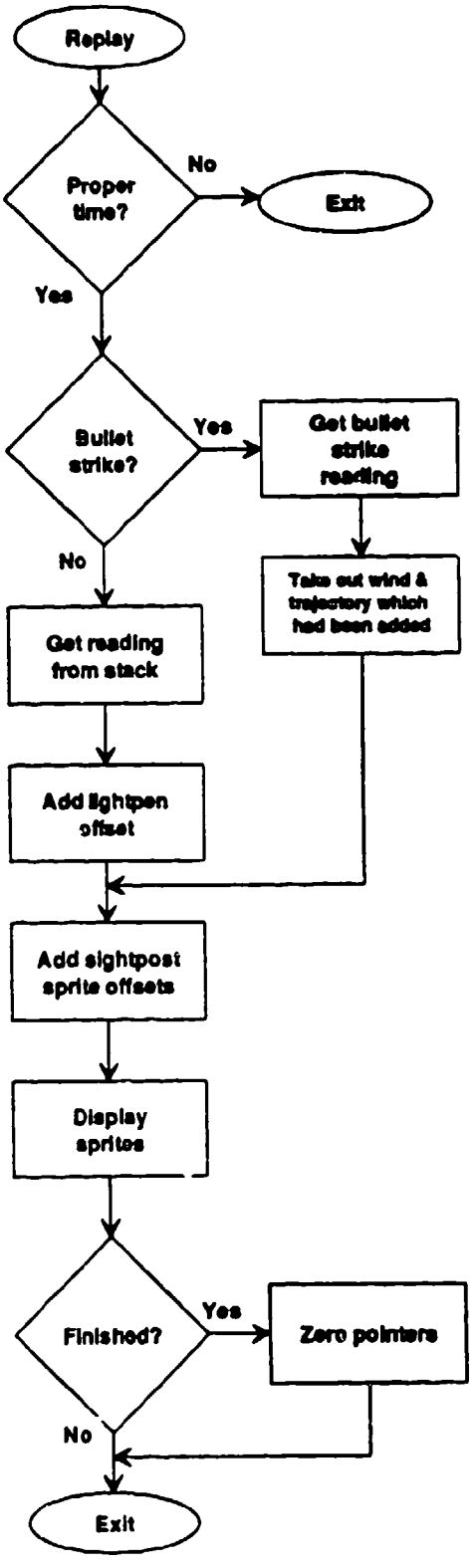

Readings Before
Finished with previous shot?
Yes
Exit
No
Get Lightpen Reading
Trigger Pull?
No
Exit
Yes
Lightpen on screen?
No
Border to purple
Exit
Yes
Bang
Add lightpen offsets to (X,Y) for bullet strike
Set up for readings after
Zero pointers
Exit


Readings After
Get Lightpen Reading
Enough readings after?
No
Exit
Yes
Set up for readings before
Zero pointers
Exit


CTRMOV
Enable interrupts
Initialize
CTR-A
Move sprites
Delay
Trigger pull?
No
CTR-D
Yes
Find target closest to bullet strike
Compute proper wind and trajectory
Time of flight delay?
Yes
Move sprites
No
Add wind and trajectory
CTR-B

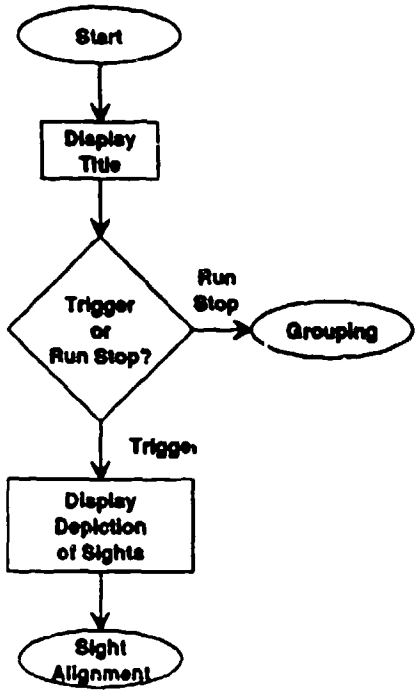

CTR-B
Flash bullet sprite
Check for collision
Turn off bullet sprite
Store shot information
Display cross if selected
Set codes if this target is multiple sprites
Cross up from previous hit?
Yes
Drop previously hit target
No
Check for penalty
Move lightpen readings so oldest is first
CTR-C

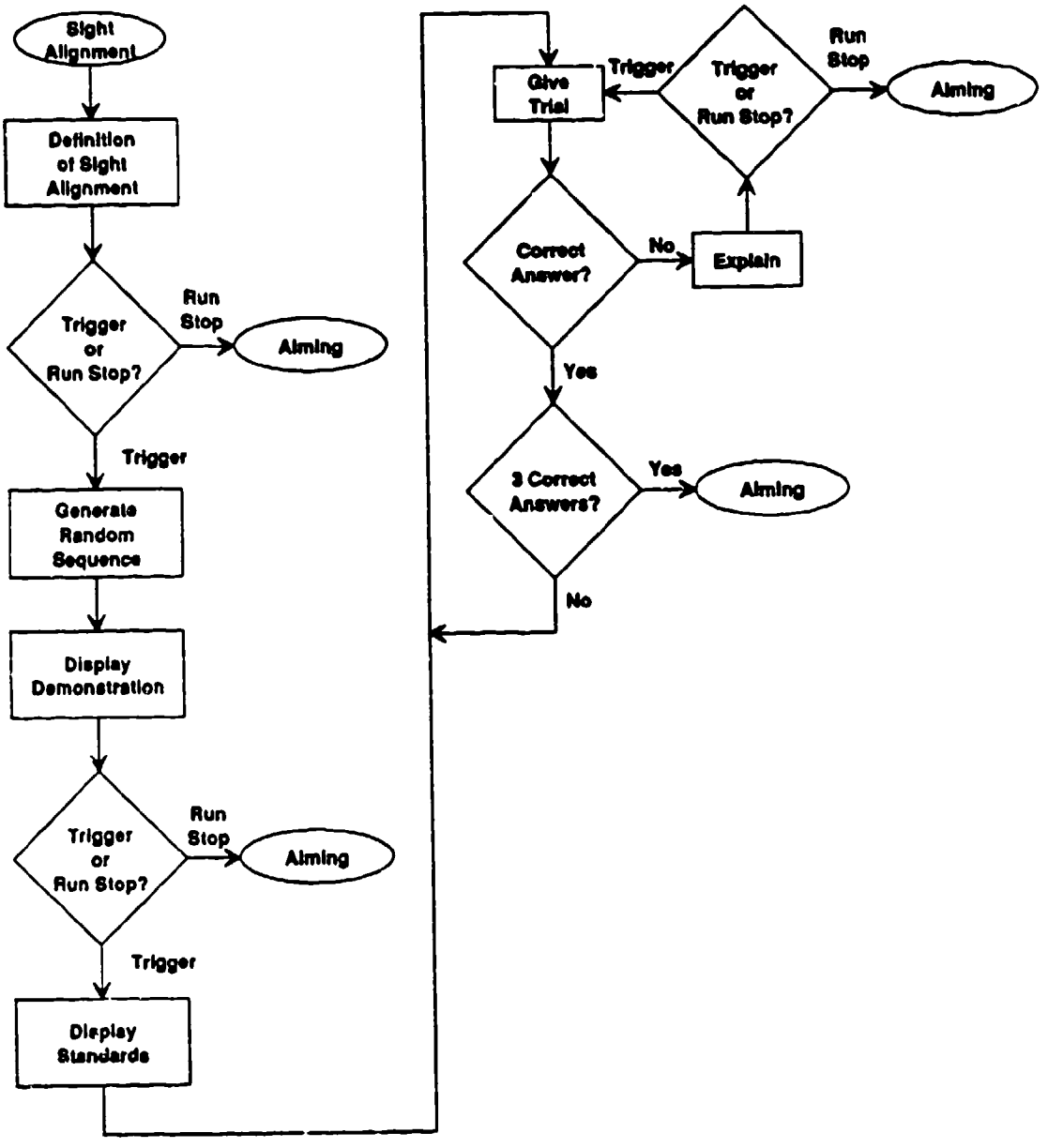

CTR-C
Out of ammo?
Yes
Display message and set flags
CTR-A
No
Return when no targets up?
Yes
No
CTR-D
Run stop pressed?
Yes
No
Any targets off screen?
No
CTR-A
Yes
Whistle code?
Yes
Sound whistle
No
Turn off targets which are off screen
CTR-E
CTR-E
Return when no targets up?
No
CTR-A
Yes
Set status flag
Exit

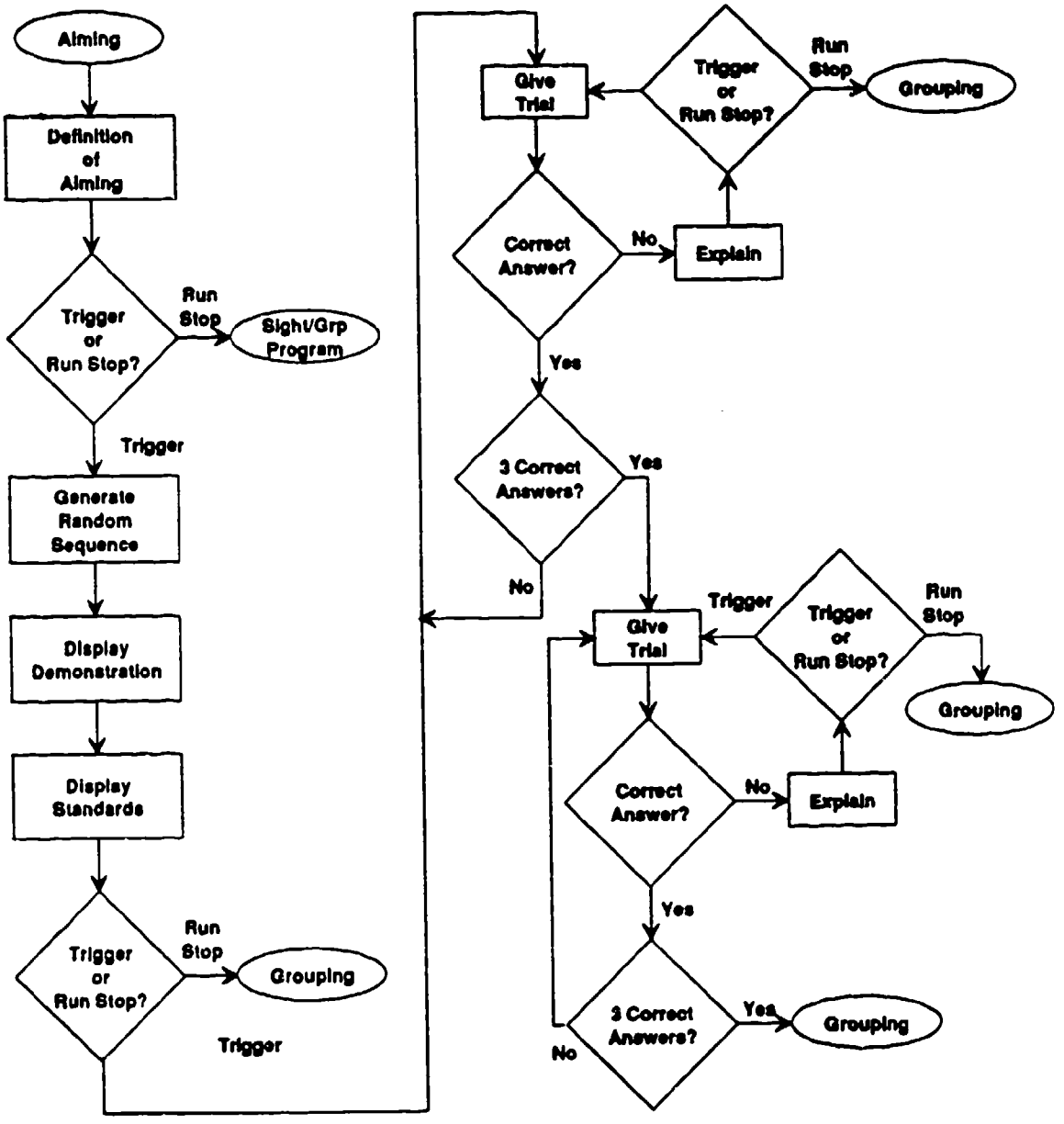

Replay
Proper time?
No
Exit
Yes
Bullet strike?
Yes
Get bullet strike reading
Take out wind & trajectory which had been added
No
Get reading from stack
Add lightpen offset
Add sightpost sprite offsets
Display sprites
Finished?
Yes
Zero pointers
No
Exit

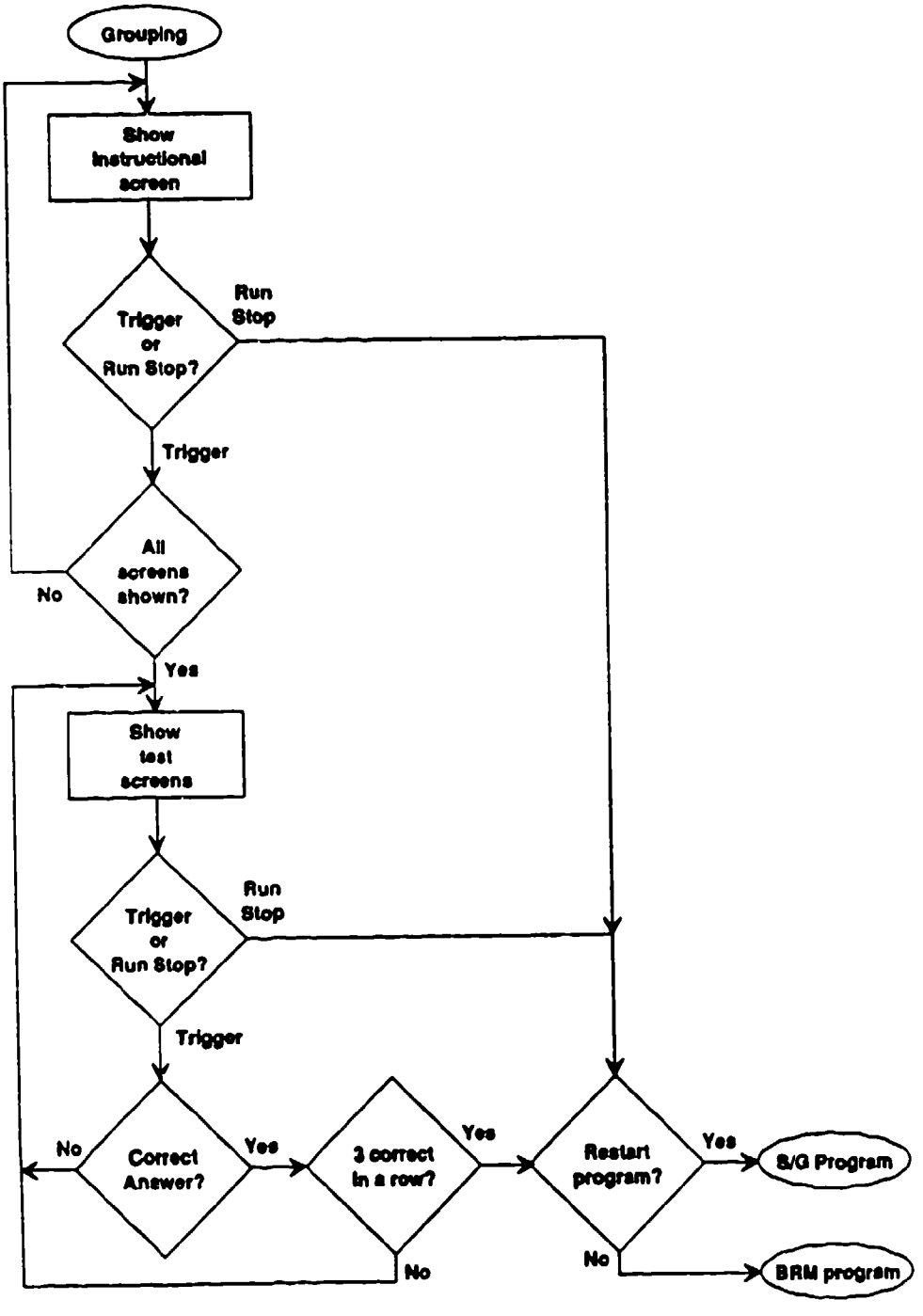
APPENDIX E

Programming Flow Chart for the Sight Alignment and Grouping Program




Sight Alignment
Definition of Sight Alignment
Trigger or Run Stop?
Run Stop
Aiming
Trigger
Generate Random Sequence
Display Demonstration
Trigger or Run Stop?
Run Stop
Aiming
Trigger
Display Standards
Give Trial
Trigger
Trigger or Run Stop?
Run Stop
Aiming
Correct Answer?
No
Explain
Yes
3 Correct Answers?
Yes
Aiming
No


Aiming
Definition of Aiming
Trigger or Run Stop?
Run Stop
Sight/Grp Program
Trigger
Generate Random Sequence
Display Demonstration
Display Standards
Trigger or Run Stop?
Run Stop
Grouping
Trigger
Give Trial
Trigger or Run Stop?
Run Stop
Grouping
Correct Answer?
No
Explain
Yes
3 Correct Answers?
Yes
No
Give Trial
Trigger
Trigger or Run Stop?
Run Stop
Grouping
Correct Answer?
No
Explain
Yes
3 Correct Answers?
Yes
Grouping
No


Grouping
Show instructional screen
Trigger or Run Stop?
Run Stop
Trigger
All screens shown?
No
Yes
Show test screens
Trigger or Run Stop?
Run Stop
Trigger
No
Correct Answer?
Yes
3 correct in a row?
Yes
No
Restart program?
Yes
S/G Program
No
BRM program

APPENDIX F

BRM Program Documentation

Basic Hierarchy Chart.
Lines 10-310: Main program driver.
Lines 1000-2890: Subroutines for firing levels.
    Lines 1000-1640: Establish shot group/extended skills tests procedure.
    Lines 1650-1930: Firing levels 1 and 2.
    Lines 1940-2300: Firing levels 3 and 4.
    Lines 2310-2730: Firing levels 5-8.
    Lines 2740-2890: Firing level 9.
Lines 3000-5860: Supporting subroutines.
Lines 6000-6140: Data.

Lines 10-310: Main program driver.
    Lines 10-70: Initialize variables.
    Lines 80-120: Lines 80-310 form the main loop of the program. At this time, the BRM cartridge opening screen is being shown alternatively with the high scorer screen (after someone has fired). These lines poll the keyboard for the spacebar, the break key, or the "LP" or "SG" key combinations which enter the 9-level scenario, go to the menu, or run the light pen mount adjustment, or sighting and grouping programs respectively.
    Lines 130-220: Line 130 transfers control to line 230 if variables have been initialized and the option of record keeping has been established. Otherwise, these lines welcome the user to MACS, and initialize array variables.
    Lines 230-310: These lines send the firer first to the zeroing routine, and then to the appropriate level of fire. Upon completion of the exercises, the congratulations screen is displayed and control returns to the top of the program at line 80.

Lines 1000-1640: Establish shot group/extended skills tests procedure.
    Lines 1000-1010: If the record keeping option has been specified, the user must now input the firer's identification number.
    Lines 1020-1120: These lines show the instructional zeroing screen and check for the trigger to be pulled, or the "@" key to be pressed. If the trigger is pulled, the variable TN is set to 0 and the zero targets are shown with a random horizontal (X) value. If the "@" key is pressed, TN is set to -1 and the zero targets are displayed with the same X value (165). The machine language routine which draws expanded targets from the sprite data (referenced by the BASIC variable DT) is called to draw the 150-meter target (expanded to 50 meters). Line 1050 calls this routine and shows a center dot representing center of mass of the target.
    Lines 1130-1150: These lines initialize the variables needed by the interrupt routines and copy the zero target data from its permanent location on the EPROM to its useful place in RAM. This routine is referenced by the BASIC variable DA.
    Lines 1160-1200: If the shot number is 0 or 3, i.e., if the user is about to fire the first shot of either firing position, the accumulator for the average diagnostic scores (for the summary) is set to 0 and the firer is instructed to assume the proper position. The high resolution graphic screen must also be displayed at this point.

Lines 1210-1220: The appropriate X coordinate (either 165 or random, as determined by the value of TN) is put into the target data.

Line 1230: The target control and lightpen reading routine is called. If the routine returns to BASIC as a result of the break key's being pressed, control returns to line 1210 and the firer engages another target without advancing the shot counter.

Line 1240: Compute the diagnostic scores.

Lines 1250-1270: If the shot count is less than 3, the lightpen offset x and y values (OX and OY) are incremented. Each (X,Y) coordinate for bullet strike is examined to make sure it is within a range left and right of the target and immediately below. This range is the same as in the calibration program, assuming the calibration cross is the center of mass of the 250 meter target now being fired at. If a shot is determined to be out of range, the flag BC is set to 1.

Lines 1280: The accumulators for the average steady position and trigger squeeze scores are adjusted. Only these two scores can be computed now because no lightpen offset has been determined.

Lines 1290-1440: These lines are executed only if the shot count is at 2 (after the third shot has been fired). If a shot was determined to be out of range, the message "INVALID SHOT GROUP, TRY AGAIN" is displayed. If this is not the first invalid shot group, the additional prompt "CHECK CALIBRATION (SEE MANUAL)" is shown. The firer must then repeat the calibration process at line 1130 or press the stop key and enter the calibration program. If all shots were in range, the lightpen offsets can now be determined since the zeroing process has been completed, and the aiming and shot location scores for the first three shots can be computed (SYS 3427). The three-round shot group and summary diagnostic scores are shown. If the summary scores meet the standard, the firer fires three more shots. If the standard was not met, the program returns to line 230.

Line 1450: If the shot count is less than 6, the program loops back to line 1160.

Line 1460-1470: The shot location scores for shots 3-5 are computed and the three-round shot group and summary diagnostic scores are shown.

Lines 1480-1490: The program determines a starting level or continues to the extended skill test. If no further testing is required, control is returned to line 230.

Lines 1500-1540: Variables for the extended skill test are set up and the introductory screens displayed.

Lines 1550-1640: Either three or six more shots are fired in this loop for the extended skill test. These lines proceed in similar manner to the regular skill test with the exception of line 1570, which must now consider the possibility of a no fire. Control is returned to line 230 upon completion.

Lines 1650-1930: Firing levels 1 and 2.

Lines 1650-1660: The targets are shown at all ranges and the expanded target with the "bull's eye" circles is now shown.

Lines 1670-1680: Variable initialization.

Lines 1690-1710: If the current level is 1 and the "EZ" option has been selected, only the data for 250 meter targets is moved. The line which loads all other targets is skipped.

Line 1720: This line moves the data for all targets from EPROM.

Lines 1730-1780: All variables are initialized, and the hires scenario and status line are displayed. A random X coordinate and the no time limit

code are recorded in the data for the machine language routines and the trajectory and Y coordinate are obtained from the data for use in future BASIC routines. If the 50 meter target is being engaged, the second sprite is prepared.

Lines 1790: The machine language routine target and lightpen control is called here. POKE 878,0 turns off the cross delay counter so that the cross will remain on the screen and not disappear after one half second. If the break key is pressed during engagement, control is transferred to line 1930 and the subroutine is exited.

Lines 1800-1840: If the "call your shot" option is in effect, the program displays the message "CALL YOUR SHOT: 5." Each second from 5 to 0 is counted down and displayed. When the timer reaches 0, the crosshair is displayed.

Line 1850-1870: If a hit is detected, the prompt "HIT" is shown at the bottom left of the screen. If a miss is reported, the word "MISS" appears. The replay routine is called with a check for the break key.

Line 1880: The shot counter is incremented and if it is not the third shot at any target, the target data pointer is restored to the same target and the program loops back to line 1750.

Line 1890-1910: At this point in the routine, three shots have been fired at a target. The shot group is displayed and if the number of hits is 2 or 3, the next target is engaged (beginning at line 1730). If this success comes after an earlier failure, the message 'STANDARD MET, CONTINUE." is displayed.

Line 1920: This line indicates the firer did not pass the standards. The number of repetitions counter (RP) is incremented, the "YOU DID NOT MEET THE STANDARD. PREPARE TO REFIRE." message is displayed, the border is changed to red, the number of hits (H) is set to 0, and control is sent to line 1750 for replay of the same target.

Line 1930: All sprites are turned off, summary diagnostic scores are displayed, the next level of fire is determined, and control is returned to line 280.

Lines 1940-2300: Firing levels 3 and 4.

Lines 1940-2060: The proper instructional screens are displayed, all variables are initialized and the target data is moved to RAM in random order.

Lines 2070-2110: These lines control the firing for the first round of targets. The work horse of this routine is the subroutine beginning at line 4850, which returns (in the variable Z) 128 if the break key is pressed during the exercise, 64 for a no fire, or 0 to proceed. When all targets have been engaged once, the next set of lines are executed to handle any necessary repetitions.

Line 2120: The summary shot groups for all targets engaged are shown.

Lines 2130-2240: Shots are refired on the targets for which the firer did not qualify. Again, the main routine is the subroutine beginning at line 4370.

Lines 2250-2300: The average diagnostic scores are computed and displayed and the next level of fire is determined. Control is returned to line 280.

Lines 2310-2730: Firing levels 5-8.

Lines 2310-2410: Instructional screens are displayed, all variables are initialized and the target data for the appropriate level (supported position) are moved to RAM.

Lines 2420-2460: The main routine at subroutine 4510 is called in line 2420. A summary of hit, miss, no fire, accuracy, and penalties is shown after firing. If the break key was pressed during engagement, the subroutine is exited. After shot groups are shown, the program checks to see if the standards have been met and either returns to line 2380 or continues to the unsupported position.

Lines 2470-2530: Instructional screens are displayed, all variables are initialized and the target data for the appropriate level (supported position) are moved to RAM. The main routine at subroutine 4510 is called in line 2530.

Lines 2540-2580: After the hit, miss, no fire, accuracy, and penalty summary screen is shown, the shot groups for all targets engaged are shown and records for the unsupported position are saved to disk if requested. If standards have not been met, control returns to line 2500.

Lines 2590-2730: The hit, miss, no fire, accuracy, and penalty summary screen is shown for both firing positions and the next level of fire is determined.

Lines 2740-2890: Firing level 9.

Lines 2740-2800: Instructional screens are displayed, all variables are initialized and the target data for level 9 are moved from EPROM to RAM. The high resolution graphics screen and status line are displayed.

Lines 2810-2890: The machine language routine controlling the target engagement and lightpen collection is entered. Due to the unique timing which is utilized in this level, all targets are displayed by this machine language routine, therefore, very little is done in BASIC. The positions are repeated (with shot groups and record storage to disk in between) until standards are met or the break key is pressed. After successful completion, the current level (CL) variable is set to 10 to signify the conclusion of the program and control is returned to line 280.

Lines 3000-3040: These lines perform the calculations for the diagnostic scores. For the steady position score, SD(0,S) is initially set to -1. Assume there are n readings. If n is greater 45, SD(0,S) is set tc examine the readings from -45 to -7. If n is less than 45 but greater than 6, SD(0,S) is set for readings n to -7. The range of readings for the aiming score is set to the same value as that of the steady position. For the trigger squeeze measure, SD(2,S) is set to -1. If there are more than 5 readings, it is changed so that readings -6 to -1 are examined. The machine language routine (MA) is then called to compute the scores.

Lines 3050-3060: Certain variables are saved into the array H%.

Lines 3070-3080: Variables which were saved in lines 3050-3060 can be restored by calling this routine.

Lines 3090-3180: With the numeric diagnostic scores in the SD array, this routine returns the appropriate descriptive diagnostic score. It is most often called at line 3090, but in certain circumstances, may be called at line 3100 with the diagnostic score in Z4 and the subscript for criterion comparison in Z5. If the current level is 0 (zeroing procedure) and the NU variable (a flag for numeric scores) is not equal to 0, both the descriptive and numeric diagnostic scores are displayed simultaneously.

Lines 3190-3290: The ASSUME (proper) POSITION message is displayed. If PS is 0, the position is supported. For a 1 in PS, it is unsupported. If B$ is not the null string, it is displayed above the ASSUME POSITION message. This routine is sometimes called at line 3250 to display the <PULL TRIGGER TO CONTINUE> message, at line 3260 to check for trigger pull, or at line 3280 to check for trigger release.

Line 3300: This procedure prints the string in A$ to the graphics screen by calling the appropriate machine language routine.

Line 3310: The hires graphics screen at location S1 of chip/bank S2 is displayed.

Lines 3320-3350: The average diagnostic scores from B to E are computed.

Lines 3360-3410: Determines the appropriate next level of fire based on skill test performance.

Lines 3420-3480: These lines display the introductory screen to each level. If the trigger is pulled, the corresponding level is entered. If the break key is pressed, control is transferred to the menu screen.

Lines 3490-3930: The menu screen. This routine displays the options available to the user, including "L" (selecting start and stop levels), "N" (returns a new firer to the calibration routine), "W" (sets wind speed), "CS" (selects or deselects the "call your shot" option), "G" (returns to the level where the program was interrupted), and "EZ" (show only 250 meter targets on level 1).

Lines 3940-3970: Sprite initialization for the machine language replay routine.

Lines 3980-4290: Display diagnostic scores and replay for levels 1-4.

Lines 3980-4040: Display diagnostic scores. If a miss was detected, the word "MISS" is shown in place of a shot location score.

Lines 4050-4290: Show replay. The replay target is positioned so that its center of mass is at the point (254,176). The sight post which represents the perfect sight picture is adjusted for wind and trajectory. The replay message is displayed and the center of the screen is colored green by the machine language routine referenced by the variable CO. The replay is interrupt driven, and lines 4150-4170 wait for the replay to finish at least once before the trigger can be pulled. Lines 4200-4280 control the pause when the "Correct Sight Placement/Your Sight Placement" and "Pull trigger to continue" messages are displayed, waiting for either trigger pull, the break key, or the time limit to expire before showing the replay again.

Lines 4300-4450: Display final diagnostic scores for levels 1-4 and determine the next firing level.

Line 4300: If less than two shots have been fired, the summary scores are not displayed.

Lines 4310-4340: Compute the average diagnostic scores.

Lines 4350-4390: Display the final scores.

Lines 4400-4450: Determine the next firing level. If there is more than one poor score and the current level is greater than 1, the firer must regress a level. If there is a poor score or a score which is below

average, the firer must repeat the level. Otherwise, the firer attempts the next level.

Lines 4460-4510: A stack beginning at 16325 is used to store target numbers before the machine language routine referenced by the variable RM is called. By randomizing the numbers in the stack, the RM procedure can take the corresponding target from the data stored on the EPROM and place it in RAM. Before calling this subroutine, the variables B and E must be set to the beginning and ending point in the stack. This technique is used to accommodate those circumstances where two or more sets of targets must be randomized. For levels 5-8, the supported targets must be dealt with before the unsupported targets so that they are not all randomized together. This subroutine randomizes the stack so that the RM procedure can be called.

Lines 4520-4620: Main scenario routine for levels 5-8.

Lines 4520-4580: These lines perform the initialization process which varies for each level. The variable AD considers the different feedback for hit or miss (cross always, cross only for misses, or cross never) and whether or not the records should be saved to the disk buffer (remember that only the first attempt is saved).

Lines 4590-4610: The machine language target control routine is called, the whistle is blown if a target is left standing after the time limit has expired, and the procedure waits for the cross to disappear before continuing.

Line 4620: A random delay before the next screen is displayed.

Lines 4630-4690: Display shot groups. The target is positioned so that its center of mass is at (174,150) and the shots are displayed by the machine language routine referenced by the variable GP. The shot location score is returned in Z4. This result was once displayed at the bottom of the shot group screen.

Lines 4700-4850: Display the summary screen of hits, misses, no fires, shot location, and penalties for levels 5-8. It will also display the messages "Standard met" or "Standard not met."

Lines 4860-5050: Main scenario routine for levels 3-4.

Lines 4860-4920: All variables are initialized and the high resolution graphic screen with status line is displayed. The machine language routine which displays the targets is called.

Line 4930-4940: Hit, miss, or no fire is determined. If the break key was pressed or a no fire was recorded, the subroutine returns to line 2090.

Lines 4950-5050: The diagnostic scores are displayed and the replay is shown if a bad shot is determined. A bad shot is indicated by a poor score in any diagnostic measure or a below average or poor score for shot location.

Lines 5060-5150: Display diagnostic scores and shot group for the skill test.

Lines 5160-5240: Display summary screen for level 9.

Lines 5250-5270: Display four fundamentals of rifle marksmanship.

Lines 5520-5540: The "YOU DID NOT MEET THE STANDARD. PREPARE TO REFIRE." message is displayed and the border is changed to red to signify more target presentations due to a failure to meet standards.

Lines 5550-5560: The message "STANDARD MET. CONTINUE." is shown when a student has failed a standard, and then successfully meets a standard.

Lines 5570-5600: These lines display the status line after adding the current wind speed and direction.

Lines 5610-5680: This subroutine prompt the user for input at the menu screen. The menu is displayed on the high resolution graphic screen, necessitating a more complex means of obtaining the input. Only two characters or numbers are accepted.

Line 5690: The record keeping buffer is initially set for no fires.

Line 5700: This subroutine calls machine language routines from chip 0, bank 1. The variable SB should be set to the low byte of the address of the routine desired.

Line 5710: This routine performs the same function as in line 5700, except parameters can be passed through the variable Z.

Lines 5720-5730: The graphic depiction of the diagnostic scores ("bull's eye" target) and the target ranges are shown.

Lines 5740-5810: Displays examples of the proper offset for wind effects.

Lines 5820-5860: Checks the number of hits and determines if the firer was an expert, marksman, sharpshooter, or unqualified. If H%(0)=1, the firer had an earlier repetition and therefore can be rated only as high as sharpshooter.

Lines 6000-6120: Data.

APPENDIX G

MACS BRM Cartridge Program Listings

```
10 POKE 808,237:V=53248:POKE 53281,0:POKE V+21,0:B$="  ":SL=1:EL=9:BC=-1
20 DIM SD(3,25),CR(9,3),M%(8),DA%(6,10),CR$(5)
30 I1=43640:I2=43904:NV=0:EZ=0:CS=0
40 DEF FNH(X)=INT(X/256):DEF FNL(X)=X-INT(X/256)*256
50 DEF FNW(X)=PEEK(X)+PEEK(X+1)*256:DEF FNE(X)=INT((ABS(X>0)*X)+.5)
60 BX=256:BM=3400:DS=3403:RT=3406:SV=3409:BANG=3415:SC=3421:MA=3424:LC=3430
70 GP=3433:CO=3436:WM=3439:HS=3442:DT=3445:T=0
80 IF PEEK(56321)=127 THEN SYS DS:GOSUB 3490:GOTO 130
90 IF PEEK(56321)=247 THEN GOSUB 326C:POKE 53280,0:SYS DS:GOTO 130
91 IF BX=256 THEN 100
92 T=T+2:IF T<800 THEN 100
93 IF (T AND 1)=1 THEN SYS 49186:T=0:GOTO 80
94 POKE 871,3:SYS HS:A$="{LGRN}{F2}0823<Pull trigger to begin>":GOSUB 3300:T=1:GOTO 80
100 GET A$:IF A$="" THEN 80
110 B$=RIGHT$(B$,1)+A$:S=-(B$="lp")-(B$="sg")*2:IF S=0 THEN 80
120 SYS DS:POKE 53265,27:POKE 53272,23:PRINT"{CLR}";:POKE 4*161,S:SYS 49179
130 IF BX<256 THEN 230
140 A$="{F3}09{F6}00{F5}00{F7}00{CLR}{F2}1302{F1}{GRN}M A C A{F2}0905{LBLU}Basic Rifle{F2}0803Marksmanship{F2}1211"
150 A$=A$+"Training!RED}{F3}00":GOSUB 3300
160 POKE 53281,0:FOR I=0 TO 9:FOR J=0 TO 3:READ CR(I,J):NEXT J,I
170 FOR I=1 TO 6:FOR J=0 TO 4:READ DA%(I,J):NEXT J,I
180 FOR I=1 TO 5:READ CR$(I):NEXT I
190 REM S=0:SB=15:GOSUB 5710:BX=S
200 BX=0:FOR I=1 TO 500:NEXT I:IF BX=0 THEN 230
210 POKE 53265,27:POKE 53272,23:INPUT"{CLR}Date";A$:IO$=LEFT$(A$,8)+CHR$(13)
220 INPUT"Unit";A$:IO$=IO$+LEFT$(A$,20)+CHR$(13)
230 POKE 914,0:BL=0:CL=0:MO$="a":GOSUB 1000:IF CL>BL THEN CL=BL
240 IF CL<SL THEN CL=SL
250 GOSUB 3420:BC=BL:IF CL=0 THEN 230
260 IF(WI AND 16)THEN WI=INT(RND(1)*5):WI=16+((WI-(WI=4)*2))+(INT(RND(1)*8)*32)
270 POKE 914,WI:ON CL GOSUB 1650,1650,1940,1940,2310,2310,2310,2310,2740
280 MO$="a":IF (CL<=EL) AND (SL<>EL) THEN 240
290 A$="{F3}09{F6}05{CLR}{F5}05{F7}05{BLK}{F2}0504{F1}CONGRATULATIONS{F8}{F8}  You are finished{F8} with this "
300 A$=A$+"program.{F8}{F8}  Call instructor.{F1}":GOSUB 3250
310 POKE 53280,0:SYS 49186:RESTORE:POKE 198,0:GOTO 80
1000 TN=0:M%(0)=0:S1=32766:S2=0:IF BX=0 THEN 1020
1010 POKE 53272,23:POKE 53265,27:INPUT"{CLR}{WHT}Firer ID #";ID$:ID$=LEFT$(ID$,20)
1020 GOSUB 5250:POKE 53265,43:A$="{F6}00{F5}13{F7}00{CLR}{BLK}{F1}{F2}0701First 3 shots"
1030 A$=A$+"{F2}0803establish shot group{F2}0105Aim center of mass."
1040 A$=A$+"{F2}0517White dot shows{F2}1319center.":GOSUB 3300
1050 POKE 880,3:SYS DT:FOR S=12196 TO 12199:POKE S,15:NEXT S:POKE 1524,1
1060 A$="{F2}0723<Pull trigger to continue>{F3}00":GOSUB 3300
1070 IF PEEK(56321)=247 THEN 1100
1080 GET A$:IF A$<>"@" THEN 1070
1090 TN=-1
1100 GOSUB 3280:IF WI=0 THEN 1130
1110 A$="{F3}09{F5}00{F7}00{F1}{GRN}{F2}0904There is no{F2}0607wind while you{F2}0610are establishing"
1120 A$=A$+"{F2}0913shot group.{LGRN}{F1}":GOSUB 3250
1130 BC=0:POKE 899,42:POKE 883,60:POKE 838,0:FOR S=834 TO 837:POKE S,0:NEXT S
1140 OX=0:OY=0:POKE 838,0:POKE 889,0:POKE 890,160:POKE 891,40:POKE 892,160
1150 POKE 823,0:POKE 824,0:POKE 821,17:POKE 822,0:SB=6:GOSUB 5700:S=0
1160 IF (S<>0) AND (S<>3) THEN 1210
1170 FOR S=0 TO 3:SD(S,25)=0:NEXT S:PS=INT(S/3)
1180 B$="{F1}{F2}0303{GRN}Next 3 shots help{F2}0106decide start level.":IF PS THEN 1200
1190 B$="{F1}{F2}0301{GRN}Fire one shot per{F2}1304target.{F2}0007Targets are untimed."
1200 GOSUB 3190:GOSUB 3310
1210 POKE 876,0:POKE 877,205:X=((INT(RND(1)*75))*2)+71:IF TN THEN X=165
1220 POKE 52481,X:GOSUB 3250
1230 S=-20352:SB=0:GOSUB 5710:POKE V+21,0:IF S=128 THEN 1210
1240 N=FNW(907):N=N+(N>128)*(N-128):GOSUB 3000
1250 IF S>2 THEN 1280
1260 X1=(X+11)-FNW(867):Y1=165-FNW(869):OX=OX+X1:OY=OY+Y1:SD(1,S)=Z
1270 IF (X1>-73) OR (X1<-148) OR (Y1>-17) OR (Y1<-47) THEN BC=1
1280 SD(0,25)=SD(0,25)+SD(0,S):SD(2,25)=SD(2,25)+SD(2,S)
1290 GOSUB 3280:IF S<>2 THEN 1450
1300 OX=INT((OX/3)+1):OX=INT(OX/2)*2:OY=INT((OY/3)+0.5)
1310 IF (BC=0) OR (EC=0) THEN 1370
1320 A$="{F3}09{CLR}{RED}{F5}00{F7}00{F1}{F2}0204Invalid shot group{F2}1107try again":IF M%(0)=0 THEN1340
1330 A$=A$+"{F2}0310or check lightpen{F2}0511mount alignment{F2}0816{see manual}"
1340 A$=A$+"{F1}{LGRN}":M%(0)=1:GOSUB 3250
1350 IF BR THEN S=1:GOTO 120
1360 GOTO 1130
1370 X=OX-(OX<0)*65536:Y=OY-(OY<0)*65536
1380 POKE 834,FNL(X):POKE 835,FNH(X):POKE 836,FNL(Y):POKE 837,FNH(Y)
1390 SD(1,0)=SD(1,0):SYS 3427:FOR S=0 TO 2:SD(1,25)=SD(1,25)+SD(1,S)
1400 SD(3,25)=SD(3,25)+SD(3,S):NEXT S:S=43640:GOSUB 5060
1410 A$="{F3}05{CLR}{F5}00{F7}00{F1}{WHT}shot group"+CR$+"{BWLC}{LBLU}{F2}0304Would you like to{F2}1107try for a"
1420 A$=A$+"{F2}0610tighter group?{F1}{F2}0523{GRN}Pull trigger to select answer{F2}0017{F3}08"
1430 GOSUB 3300:SB=16:GOSUB 5710:POKE 53265,43:IF S THEN 1130
1440 B=0:S=2:GOSUB 3370:S=2:IF (S=0) OR (SL<2) THEN CL=1:RETURN
1450 S=S+1:IF S<6 THEN 1160
1460 POKE 53265,43:FOR S=3 TO 5:SD(1,25)=SD(1,25)+SD(1,S)
1470 SD(3,25)=SD(3,25)+SD(3,S):NEXT S:S=43664:GOSUB 5060
1480 N=6:T=0
1490 B=3:S=5:GOSUB 3370:IF (S=0) OR (SL<3) THEN CL=2:RETURN
1500 B$="{F1}{F2}0101{GRN}You have done well!{F2}1304Now try{F2}0607timed targets."
1510 PS=0:GOSUB 3190:GOSUB 5690
1520 POKE 838,0:POKE 889,0:POKE 890,160:POKE 891,40:POKE 892,180:POKE 876,0
1530 POKE 877,205:POKE 823,17:POKE 824,0:POKE 821,114:POKE 822,0:SB=6:GOSUB5700
1540 POKE 899,42:GOSUB 3310:B=0:E=2:T=1
1550 FOR S=B TO E
1560 SB=0:S=-20352:GOSUB 5710:POKE V+21,0
```

```
1570 IF E>63 THEN SYS WH:FOR I=0 TO 3:SD(I,S)=CR(I+PS*4,3):NEXT I:GOTO 1590
1580 M=FNW(907):N=N+(N>128)*(N-128):GOSUB 3000
1590 GOSUB 3280:NEXT I:GOSUB 3370:IF (E<>0) AND (EL>(T+2)) THEN 1610
1600 N=6:CL=T+2:B$="":GOTO 1630
1610 T=T+1:IF T=2 THEN PS=1:B$="":GOSUB 3190:GOSUB 3310:B=3:E=5:GOTO 1550
1620 N=6:CL=5:B$="{F2}1005Well done!"
1630 A$="{F3}09{CLR}{F5}00{F7}00{GRN}{F1}"+B$+"{F2}0308You will start at{F2}1511. . .{F3}00"
1640 GOSUB 3100:FOR D=1 TO 800:NEXT D:RETURN
1650 GOSUB 5720:GOSUB 5740:NT=3:PS=CL-1:B$="{F1}{F2}0002{GRN}Targets are untimed."
1660 B$=B$+"{F2}0005Standard: Hit 2 of 3{F2}0308at each distance.":GOSUB 3190
1670 GOSUB 5690:POKE 876,0:POKE 877,205:POKE 889,0
1680 POKE 890,160:TM=0:POKE 891,40:POKE 892,180:POKE 838,0:POKE 899,42
1690 IF (EE=0) OR (CL<>1) THEN 1720
1700 FOR I=0 TO NT*6-1:POKE 16325+I,4:NEXT I:POKE 16325+NT*6,255
1710 POKE 821,0:POKE 823,114:POKE 824,0:SB=3:GOSUB 5700:GOTO 1730
1720 POKE 823,116:POKE 824,0:POKE 821,225:POKE 822,0:SB=6:GOSUB 5700
1730 S1=32768:S2=8:IR=-32640-(CS*4096):M=0:TM=TM+1:RP=0:IF TM=7 THEN 1930
1740 GOSUB 3050
1750 S=PEEK(838):GOSUB 3310:A$="L="+STR$(CL)+"|T="+STR$(S+1)+"|R="+STR$(RP)
1760 GOSUB 5570:POKE I2+7+S*8,RP:X=((INT(RND(1)*65))*2)+71:AD=FNW(876)
1770 POKE AD+1,X:Y=PEEK(AD+2):POKE AD+6,255:TJ=PEEK(AD+11):TJ=TJ+(TJ>127)*256
1780 IF (TM=1) AND ((EE=0) OR (CL<>1)) THEN POKE AD+15,X+40:POKE AD+20,255
1790 GOSUB 3950:SB=0:E=IR:GOSUB 5710:POKE 878,0:IF S=128 THEN 1930
1800 IF CS=0 THEN 1850
1810 A$="{F2}0423{BLK}{SWLC}{F1}Call your shot:":S1=TI+359
1820 I2=INT((S1-TI)/60):A$=A$+"{F2}3423"+STR$(I2):GOSUB 3300:A$="{F1}"
1830 IF I2>0 THEN 1820
1840 POKE V+21,PEEK(V+21) OR 1
1850 A$="{BLK}{F1}{F2}0423                    {F2}0123"+CHR$(142)
1860 B$="miss":IF PEEK(900)<>28 THEN M=M+1:B$="hit"
1870 A$=A$+B$:GOSUB 3300:GOSUB 3980:IF S=128 THEN 1930
1880 S=S+1:IF S/3<>INT(S/3) THEN POKE 876,M%(2):POKE 877,M%(3):GOTO 1750
1890 M=3:I=11+M%(1)*8:GOSUB 4630:IF M<2 THEN 1920
1900 IF RP>0 THEN BC=BL:GOSUB 5530
1910 GOTO 1730
1920 GOSUB 3070:IR=-31616-(CS*4096):RP=RP+1:M=0:GOSUB 5520:GOTO 1750
1930 POKE V+21,0:GOSUB 4300:CL=CL+A:RETURN
1940 RP=0:PS=CL-3:NT=4:GOSUB 5740
1950 A$="{F3}09{F6}00{CLR}{F5}00{F7}00{GRN}{F1}{F2}030324 timed targets.":IF CL=4 THEN 1980
1960 A$=A$+"{F2}0007Standard: Hit 3 of 4{F2}0310at each distance."
1970 A$=A$+"{F2}0614Replay for bad{F2}0917shots only.{F1}{LGRN}":GOSUB 3250:B$="":GOTO 2010
1980 A$=A$+"{F2}1107Standard:{F2}0010Hit 3 of 4 at ranges{F2}0012between 50 and 200 m"
1990 A$=A$+"{F2}0715Hit 2 of 4 at{F2}0717250 and 300 m{F1}{LGRN}":GOSUB 3250
2000 B$="{F1}{GRN}{F2}0604Replay for bad{F2}0907shots only."
2010 GOSUB 3190:POKE 53265,43:GOSUB 5690:IR=-32640:POKE 899,42
2020 B=0:E=NT*6-1:GOSUB 4460
2030 AD=I1:FOR I=1 TO NT*6:POKE AD,255:AD=AD+8:NEXT I:S1=37445:S2=8
2040 FOR I=B TO E:POKE 16325+I,INT(PEEK(16325+I)/NT):NEXT I:FOR I=1 TO 6
2050 DA%(I,5)=0:DA%(I,6)=0:NEXT I:FOR I=I1 TO I1+199 STEP 8:POKE I,0:NEXT I
2060 POKE 821,0:POKE 823,114:POKE 824,0:SB=3:GOSUB 5700:POKE 876,0:POKE 877,205
2070 I=0:GOSUB 3310
2080 I=I+1:IF I>NT*6 THEN 2120
2090 GOSUB 4860:POKE V+21,0:IF S=128 THEN 2250
2100 IF S=64 THEN SYS WH
2110 GOTO 2080
2120 FOR TM=1 TO 6:I=I1+(TM-1)*32:N=4:GOSUB 4630:NEXT TM:F=-1
2130 POKE 53265,43:RP=RP+1:ER=0:TM=1
2140 IF DA%(TM,1)>=(3+((CL=4) AND (TM>4))) THEN 2210
2150 ER=-1:IR=-31616:POKE 821,0:POKE 823,114:POKE 824,0:IF F THEN GOSUB 5520
2160 AD=I1+(TM-1)*32:FOR I=1 TO 4:POKE AD,255:AD=AD+8:NEXT I
2170 FOR I=0 TO NT-1:POKE 16325+I,TM-1:NEXT I:POKE 16325+NT,255:SB=3:GOSUB 5700
2180 POKE 876,0:POKE 877,205:DA%(TM,5)=0:DA%(TM,6)=0:GOSUB 3310
2190 GOSUB 4860:POKE V+21,0:IF S=128 THEN 2250
2200 IF S=64 THEN SYS WH
2210 I=FNI(I+1):IF DA%(TM,6)<NT THEN 2190
2220 S=I1+(TM-1)*32:N=4:GOSUB 4630:F=0
2230 TM=TM+1:IF TM<7 THEN 2140
2240 IF ER THEN 2130
2250 POKE V+21,0:FOR J=0 TO 3:SD(J,25)=0:NEXT J
2260 M=0:FOR TM=1 TO 6:IF DA%(TM,6)=0 THEN 2290
2270 FOR I=1 TO DA%(TM,6):S=(TM-1)*NT+(I-1):M=M+1
2280 FOR J=0 TO 3:SD(J,25)=SD(J,25)+SD(J,S):NEXT J,I
2290 NEXT TM:A=0:BL=0:IF M>0 THEN GOSUB 4330
2300 CL=CL+A:RETURN
2310 GOSUB5740:A$="{F3}09{F6}00{CLR}{F5}00{F7}00{GRN}{F1}{F2}040240 timed targets{F2}0205Randomly presented"
2320 A$=A$+"{F2}0308single or double.{F2}021220 shots/supported"
2330 A$=A$+"{F2}001520 shots/unsupported{F1}{LGRN}":GOSUB 3250
2340 PS=0:B$="{F1}{GRN}{F2}0502Fire at closest{F2}0404targets first or"
2350 B$=B$+"{F2}0006penalty is recorded.{F2}0108Standard: 15 of 20.":GOSUB 3190
2360 GOSUB 5690:RP=0:IR=512
2370 M%(0)=0:S1=35050:S2=2:IF CL>7 THEN S1=37968:S2=2
2380 POKE 53265,43:B=0:E=15:GOSUB 4460:POKE 876,0:POKE 877,205:POKE 889,0
2390 POKE 890,160:POKE 891,40:POKE 892,180:POKE 838,0:POKE 899,20:POKE 910,0
2400 POKE 918,0:POKE 919,0
2410 POKE 912,0:TM=0:POKE 821,0:POKE 823,225:POKE 824,0:SB=3:GOSUB 5700
2420 GOSUB4520:A=2:POKE I2+7,RP:HI=0:M=PEEK(838):IF (M=0) AND (A<>128) THEN2460
2430 A$="{F3}09{F6}00{CLR}{F5}00{F7}00{LBLU}{F2}0601SUMMARY: Supported Position":GOSUB 4700
2440 HI=21:MI=12:MF=23:AC=24:IF A=128 THEN BL=0:RETURN
2450 FOR TM=1 TO 6:I=I1:GOSUB 4630:NEXT TM
2460 IF HI<15 THEN IR=1936:RP=RP+1:M%(0)=1:GOSUB 5520:GOTO 2380
2470 S4=AC:BC=BL
2480 PM=PEEK(912):TM=16:RP=0:PS=1:B$="{F1}{GRN}{F2}1104Standard:{F2}0707Hit 15 of 20."
2490 GOSUB 3190:POKE 899,40:POKE 53265,43:IR=512
2500 B=16:E=28:GOSUB 4460:POKE 876,0:POKE 877,205:POKE 889,176:POKE 890,164
2510 POKE 891,184:POKE 892,181:POKE 838,20:POKE 910,0:POKE 912,0
2520 POKE 918,2:POKE 919,0
2530 POKE 821,16:POKE 823,225:POKE 824,0:SB=3:GOSUB 5700:GOSUB 4520:A=2
2540 POKE I2+167,RP:HI=0:M=PEEK(838)-20:IF (M=0) AND (A<>128) THEN 2580
2550 A$="{F3}09{F6}00{CLR}{F5}00{F7}00{LBLU}{F2}0501SUMMARY: Unsupported Position":GOSUB 4700
```

```
2560 MI=Z1:MI=Z2:MF=Z3:IF A=128 THEN BL=0:RETURN
2570 A=Z4:FOR TM=1 TO 6:Z=11+160:GOSUB 4630:NEXT TM
2580 IF MI<15 THEN IR=1536:RP=RP+1:M%(0)=1:GOSUB 5520:TM=16:GOTO 2500
2590 POKE V+21,0:Z4=A:AC=(AC+Z4)/2
2600 MI=0:MI=0:MF=0:A$="{F3}09{F6}00{CLR}{F5}00{F7}00{LBLU}{F2}1401FINAL SCORES"
2610 A$=A$+"{F2}1303{GRN}Hits{F2}2199{RED}Misses{YELO}{F2}2999No Fires{F8}{F8}":GOSUB 3300
2620 FOR I=1 TO 6:Z=DA%(1,5)+DA%(I,8):A$="{LBLU}{F2}0399"+STR$(I*50)+"{F2}1399{GRN}"+STR$(Z)
2630 MI=MI+Z:Z=DA%(I,6)+DA%(I,9):A$=A$+"{F2}2299{RED}"+STR$(Z):MI=MI+Z
2640 Z=DA%(I,7)+DA%(I,10):A$=A$+"{F2}3299{YELO}"+STR$(Z)+"{F8}{F8}":MF=MF+Z:GOSUB 3300:NEXT I
2650 A$="{F2}0699{LBLU}TOTAL{F2}1399{GRN}"+STR$(MI)+"{F2}2299{RED}"+STR$(MI)+"{F2}3299{YELO}"+STR$(MF)
2660 Z4=AC:Z5=9:GOSUB 3100:A$=A$+"{F8}{F8}{LBLU}{F2}0699Penalties:"
2670 A$=A$+STR$(PEEK(912)+PN)+"{F8}{F2}0699Overall shot location: "
2680 A$=A$+RIGHT$(CR$,LEN(CR$)-5)+"{F8}":GOSUB 5820:A$=A$+CR$+"{LGRN}":GOSUB 3250
2690 IF (Z>2) OR (MI>35) THEN BL=0:CL=CL+1:RETURN
2700 IF Z=1 THEN A=(CL>BL):BL=6-A:CL=CL+A:B$="{F2}1599poor.":GOTO 2720
2710 BL=6:B$="{F2}0699below average."
2720 A$="{F3}09{CLR}{RED}{F1}{F8}{F8} You are being sent{F8}(C/RT)  back to level"+STR$(CL)+"{F8}{F2}0899because"
2730 A$=A$+" your{F8}{F2}0399shot location was{F8}"+B$+"{F1}{LGRN}":GOSUB 3250:RETURN
2740 P8=0:GOSUB5740:SD(0,0)=0:SD(1,0)=0
2750 B$="{F1}{GRN}{F2}090240 targets.{F2}0605Attack/Retreat{F2}1108scenario.":GOSUB 3190
2760 POKE 53280,0:RP=0:S1=37968:S2=2:GOSUB 5690:IR=4736:M%(0)=0
2770 POKE 876,0:POKE 877,205:POKE 889,0:POKE 890,160
2780 TM=0:POKE 891,40:POKE 892,160:POKE 838,0:POKE 889,40:POKE 912,0
2790 POKE 823,145:POKE 824,3:POKE 821,59:POKE 822,6:BB=6:GOSUB 5700
2800 GOSUB 3310:A$="L= 9|T=  0|R="+STR$(RP):GOSUB 5570:POKE 918,0:POKE 919,0
2810 POKE 12+7,RP:BB=0:Z=IR:GOSUB 5710:A=Z
2820 POKE V+21,0:POKE 53280,0:M=PEEK(838):IF M=0 THEN RETURN
2830 B$="Supported Position":IF PS THEN B$="Unsupported Position"
2840 A$="{F3}09{F6}00{CLR}{F5}00{F7}00{LBLU}{F2}0601SUMMARY: "+B$:M=PEEK(838):GOSUB 5160
2850 IF A=128 THEN BL=0:RETURN
2860 FOR TM=1 TO 6:Z=11:GOSUB 4630:NEXT TM
2870 IF MI<23 THEN IR=3712:RP=RP+1:M%(0)=1:GOSUB 5520:GOTO 2770
2875 SD(0,0)=SD(0,0)+MI:SD(1,0)=SD(1,0)+Z4
2880 BC=BL:RP=0:IF PS=0 THEN PS=1:B$="":GOSUB 3190:IR=4736:GOTO 2770
2890 CL=10:Z4=SD(1,0)/2:Z5=9:GOSUB 3100:POKE 785,FNL(HS):POKE 786,FNH(HS)
2900 SD(0,0)=SD(0,0)+(Z/10)-(((MI AND 96)=64)-((MI AND 32)*5))*(MI AND 7)
2910 POKE 871,1:Z=USR(SD(0,0)):IF PEEK(924)>10 THEN RETURN
2920 POKE 871,0:SYS MS:POKE 871,2:Z=USR(SD(0,0)):RETURN
3000 SD(0,S)=-1:IF N>6 THEN SD(0,S)=(N-45)*256+(N-7):IF N<45 THEN SD(0,S)=N-7
3010 SD(1,S)=SD(0,S)
3020 SD(2,S)=-1:IF N>5 THEN SD(2,S)=(N-6)*256+(N-1)
3030 POKE 785,FNL(MA):POKE 786,FNH(MA):Z=USR(S)
3040 RETURN
3050 M%(1)=PEEK(838):M%(2)=PEEK(876):M%(3)=PEEK(877):M%(4)=PEEK(889)
3060 M%(5)=PEEK(890):M%(6)=PEEK(891):M%(7)=PEEK(892):M%(8)=PEEK(899):RETURN
3070 POKE 838,M%(1):POKE 876,M%(2):POKE 877,M%(3):POKE 889,M%(4)
3080 POKE 890,M%(5):POKE 891,M%(6):POKE 892,M%(7):POKE 899,M%(8):RETURN
3090 Z4=SD(J,S):Z5=J+PS*4
3100 Z=5:IF Z4>CR(Z5,0) THEN Z=4
3110 IF Z4>CR(Z5,1) THEN Z=3
3120 IF Z4>CR(Z5,2) THEN Z=2
3130 IF (Z4>CR(Z5,3))OR(Z4<0) THEN Z=1
3140 CR$=CR$(Z):IF (CL<>0) OR (MU=0) THEN RETURN
3150 B$=STR$(39-LEN(STR$(Z4))):B$=RIGHT$(B$,2)
3160 CR$="{F1}{F2}2499"+MID$(CR$,6,10)+"{F8}{F2}"+B$+"99"+STR$(Z4)+"{F8}{F1}":RETURN
3170 B$=STR$(Z4):Z4=LEN(B$):IF Z4<4 THEN FOR Z5=1 TO 4-Z4:B$=" "+B$:NEXT Z5
3180 CR$="{F1}{F2}2499"+MID$(CR$,6,10)+"{F2}3599"+B$+"{F1}{F8}":RETURN
3190 POKE V+21,0:A$="{F3}09{CLR}{F6}00{F5}00":IF B$="" THEN A$=A$+"{WHT}{F1}{F2}0008":B$="{F8}{F8}":GOTO3210
3200 A$=A$+B$:GOSUB 3300:A$="{F1}{WHT}{F2}0012":B$="{F8}{F1}{F8}{F1}"
3210 IF PS THEN A$=A$+"{F2}1199ASSUME A/W"+B$+"{F2}0999UN":GOTO 3230
3220 A$=A$+"{F2}1299ASSUME A"+B$+"{F2}1199"
3230 A$=A$+"SUPPORTED"+B$+"{F2}1299POSITION"
3240 A$=A$+"{F1}"+B$+"{LGRN}{F2}0823<Pull trigger to begin>{F3}00":GOSUB 3300:GOTO 3260
3250 A$=A$+"{F2}0723<Pull trigger to continue>{F3}00":GOSUB 3300
3260 BR=0:IF PEEK(56321)=127 THEN BR=-1:GOTO 3280
3270 IF PEEK(56321)<>247 THEN 3260
3280 IF PEEK(56321)<>255 THEN 3280
3290 RETURN
3300 A$=A$:SYS 49982:RETURN
3310 POKE 253,FNL(S1):POKE 254,FNH(S1):POKE 251,S2:SYS SC:RETURN
3320 REM GET AVERAGE ACROSS B-E SHOTS FOR EACH DIAGNOSTIC SCORE
3330 FOR Z2=0 TO 3:SD(Z2,12)=0:NEXT Z2
3340 FOR Z1=0 TO 3:FOR Z2=B TO E:SD(Z1,12)=SD(Z1,12)+SD(Z1,Z2):NEXT Z2,Z1
3350 Z=(E-B)+1:FOR Z2=0 TO 3:SD(Z2,12)=SD(Z2,12)/Z:NEXT Z2:RETURN
3360 REM ROUTINE TO SEE IF SHOTS B-E FALL INTO RANGE FOR ADVANCEMENT
3370 FOR J=1 TO 5:M%(J)=0:NEXT J:GOSUB 3330
3380 FOR J=0 TO 3:S=12:GOSUB 3090:M%(Z)=M%(Z)+1:NEXT J
3390 IF (M%(1)>0) OR (M%(2)>0) OR (M%(3)>0) OR (M%(5)<2) THEN Z=0:RETURN
3410 Z=1:RETURN
3420 A$="{F3}09{F6}00{CLR}{F8}00{F7}00{F1}{LBLU}{F2}1505LEVEL{F2}1709":B$=STR$(CL)
3430 A$=A$+B$+"{GRN}{F2}0015":GOSUB 3300:POKE 823,CL:BB=21:GOSUB 5700
3440 A$="{LGRN}{F2}0723<Pull trigger to continue>{F3}00":GOSUB 3300
3450 IF PEEK(56321)=247 THEN RETURN
3460 IF PEEK(56321)<>127 THEN 3450
3470 GOSUB 3490:IF CL=0 THEN RETURN
3480 GOTO 3420
3490 POKE 53265,43:POKE 53272,31:POKE 198,0:A$="{F6}00{F5}00{F7}00{CLR}{GRN}Start level:"
3500 A$=A$+STR$(BL)+"{DOWN}{F2}3399AA18902{GRN}Final level:"+STR$(EL)+"{F8}{BLUE}Wind speed: "
3510 IF (WI AND 16) THEN A$=A$+" Variable";:GOTO 3540
3520 A$=A$+STR$((WI AND 7)*5)+" MPH ":IF ((WI AND 7)=0) THEN 3540
3530 B$=STR$(INT(WI/32)+1):B$="0"+RIGHT$(B$,1):A$=A$+"{F4}"+B$
3540 A$=A$+"{BLK}{F8}{F8}":GOSUB 3300:POKE 253,SL:POKE 254,EL:POKE 823,0:BB=21:GOSUB 5700
3550 Z=154-((SL>1) OR (EL<9))*4:A$=CHR$(Z)+"{F8} L: Select Start/Final Level{F8}{LBLU}"
3553 Z$="{LBLU}":IF (WI AND 23) THEN Z$="{YELO}"
3555 A$=A$+" N: New fire{F8}"+Z$+" W: Set Wind Speed{F8}"
3560 B$="on ":Z$="{LBLU}":IF CS THEN B$="off ":Z$="{YELO}"
3570 A$=A$+Z$+"CS: Turn "+B$+CHR$(34)+"Call your shot"+CHR$(34)+" option{F8}{LBLU}"
3580 A$=A$+"LP: Light pen mount adjustment{F8}SG: Sighting and grouping program{F8}"
3590 GOSUB 3300:B$="250 meters":Z$="{LBLU}":IF EI THEN B$="all ranges":Z$="{YELO}"
```

```
3600 A$=Z$+"E1: Set level 1 targets for "+B$+"{F8}{LBLU}"
3610 B$="start":IF CL<>0 THEN B$="level"+STR$(CL)
3620 A$=A$+" G: Go to "+B$:B$="number or letter":IF CL=0 THEN B$="letter"
3630 A$=A$+"{F2}0023{RED}Select "+B$+" & press RETURN{LBLU}{F8}{F3}00":GOSUB 3300
3640 A$="":GOSUB 5610:E=INT(VAL(A$))
3650 IF (E>=SL) AND (E<=EL) AND (CL>0) THEN CL=E:MO$="m":BL=0:RETURN
3660 IF A$="MU" THEN MU=NOT(MU):GOTO 3490
3670 IF A$="E1" THEN E1=NOT(E1):GOTO 3490
3680 IF A$="CS" THEN CS=NOT(CS):GOTO 3490
3685 IF A$="EG" THEN EG=NOT(EG):GOTO 3490
3690 IF A$="N" THEN CL=0:RETURN
3700 IF A$="Q" THEN RETURN
3710 IF A$<>"L" THEN 3800
3720 MO$="m":A$="{F2}0723{RED}number & press RETURN          {F2}0024{BLUE}Start level (1-9)"
3730 GOSUB 5610:E$=A$:SL=INT(VAL(A$)):IF (SL<1) OR (SL>9) THEN 3720
3740 IF (CL<SL) AND (CL<>0) THEN CL=SL
3750 IF SL>9 THEN SL=9:GOTO 3490
3760 A$="{F2}0024{LBLU}Final level ("+E$+"-9)":GOSUB 5610:EL=INT(VAL(A$))
3770 IF (EL<SL) OR (EL>9) THEN 3760
3780 IF CL>EL THEN CL=EL
3790 GOTO 3490
3800 IF A$<>"W" THEN 3920
3810 A$="{F3}09{CLR}Wind Effects:{F8}{F8}Type a number (1-3) for wind speed:{F8}{F8}"
3820 A$=A$+" 1:  0 MPH{F8}{F8} 2: 10 MPH{F8}{F8} 3: 20 MPH{F8}{F8}{F3}00":GOSUB 3300
3830 A$="":GOSUB 5610:WI=VAL(A$):IF (WI<1) OR (WI>3) THEN 3830
3840 WI=(WI-1)^2:IF WI=4 THEN WI=16:GOTO 3490
3850 IF WI=0 THEN 3490
3860 A$="{F3}09{CLR}Wind Effects:{F8}{F8}Type a number (1-8) for wind direction:{F8}{F8}"
3870 A$=A$+" 1: {F4}01 (No value){F8}{F8} 2: {F4}02 (Half value){F8}{F8} 3: {F4}03 (Full value){F8}{F8}"
3880 A$=A$+" 4: {F4}04 (Half value){F8}{F8} 5: {F4}05 (No value){F8}{F8} 6: {F4}06 (Half value){F8}{F8}"
3890 A$=A$+" 7: {F4}07 (Full value){F8}{F8} 8: {F4}08 (Half value){F8}{F8}{F3}00":GOSUB 3300
3900 A$="":GOSUB 5610:IF (VAL(A$)<1) OR (VAL(A$)>8) THEN 3900
3910 WI=WI+((VAL(A$)-1)*32):GOTO 3490
3920 E=-(A$="LP")-(A$="SG")*2:IF E>0 THEN 120
3930 GOTO 3640
3940 REM SETUP FOR REPLAY: CALL ONCE BEFORE ANY TARGETS PRESENTED
3950 POKE 2041,43:POKE 2042,43:POKE 2043,43:POKE 2044,43:POKEV+23,6:POKE V+29,6
3960 POKE V+40,1:POKE V+41,0:POKE V+42,0:POKE V+43,1:POKE V+6,46:POKE V+7,164
3970 POKE V+8,126:POKE V+9,164:RETURN
3980 M=FNM(907):M=M+(M>128)*(M-128):GOSUB 3000:HI=PEEK(900)<128
3990 J=0:GOSUB 3090:A$="{F3}05{CLR}{F5}00{F1}{WHT}steady pos"+CR$:X1=E:E$=MID$(CR$,6,1)
4000 J=1:GOSUB 3090:A$=A$+"{WHT}aiming":E1$=CR$:X2=E
4010 CR$="{F2}3099"+E$+"check":IF X1>2 THEN CR$="{F2}3699"+E$+"ok"
4020 B$="{WHT}breath con"+CR$
4030 J=2:GOSUB 3090:B$=B$+"{WHT}trigger sq"+CR$:GOSUB 3280
4040 J=3:GOSUB 3090:IF HI=0 THEN CR$="{F2}3299{RED}miss"
4044 IF (X1>3)AND(X2>3)AND(Y1>3)AND(HI=0) THEN E1$=CR$(3):SD(1,E)=CR(1+PS*4,2)
4045 CR$="{F1}{F2}0008{LBLU}shot loc"+CR$:A$=A$+E1$+B$
4050 E=TN:M=FNM(915):M=M+(M>32767)*65536:IF (CL=1) AND (EZ) THEN E=5
4060 POKE 839,X+DA%(E,2):POKE 840,0:POKE 841,Y+DA%(E,3)
4070 POKE V+4,231-M:POKE V+5,175-TJ
4080 X1=ABS(244+(FNM(847)-(X+DA%(E,2)))):Y1=ABS(191+(PEEK(849)-(Y+DA%(E,3))))
4090 X2=PEEK(V+16) AND 230:IF FNM(X1)<>0 THEN X2=X2 OR 1
4100 GOSUB 3300:POKE V,FNL(X1):POKE V+1,FNL(Y1):POKE V+16,X2
4110 WS=128:POKE V+14,254-DA%(E,2):POKE V+15,200-DA%(E,3)
4120 IF (TN=1) AND ((EZ=0) OR (CL<>1)) THEN WS=192:POKE V+12,255:POKE V+13,166
4130 POKE 823,8:POKE 824,17:POKE 253,221:SYS CO:POKE 845,1:F=-1:SYS SV
4140 POKE V+21,WS OR 6:A$="{BLK}{F2}0217{F1}{RVON}replay{RVOF}{F3}00":GOSUB 3300:M=FNM(915)
4150 M=M+(M>255)*(M-255):POKE 821,M:M=M-60:POKE 833,-M*(M>0):POKE 884,1
4160 IF (PEEK(884) AND 1)=0 THEN 4190
4170 IF (PEEK(56321)=247) AND (F=0) THEN POKE 884,0:E=0:GOTO 4290
4180 GOTO 4160
4190 SYS BANG:F=0:POKE V+21,WS OR 31
4200 A$="{BLK}{F2}0217correct   your{F8}  sight     sight{F8}  placement placement"+CR$
4210 POKE 823,8:POKE 824,2:POKE 253,0:SYS CO:GOSUB 3300:F=1:E=1
4220 IF PEEK(56321)=247 THEN E=0:GOTO 4290
4230 IF PEEK(56321)=127 THEN E=128:GOTO 4290
4240 E=E+1:IF E<50 THEN 4220
4250 F=F+1:IF F<>2 THEN 4280
4260 A$="{BLK}{F2}0017  pull trigger  {F8}  to continue     {F8}                    "
4270 GOSUB 3300:POKE V+21,WS OR 1:E=0:GOTO 4220
4280 F=0:POKE 823,8:POKE 824,2:POKE 253,221:SYS CO:GOTO 4140
4290 GOSUB 3280:POKE V+21,0:RETURN
4300 A=0:BL=0:N=PEEK(838):IF N<2 THEN RETURN
4310 FOR J=0 TO 3:SD(J,25)=0:NEXT J
4320 FOR S=0 TO N-1:FOR J=0 TO 3:SD(J,25)=SD(J,25)+SD(J,S):NEXT J,S
4330 FOR J=1 TO 5:M%(J)=0:NEXT J
4340 FOR J=0 TO 3:SD(J,25)=INT((SD(J,25)/N)+.5):NEXT J:S=25:J=0:GOSUB 3090
4350 A$="{F3}05{F6}00{F5}00{F7}00{CLR}{LBLU}{F2}0400{F1}summary: level"+STR$(CL)+"{F8}{F8}{WHT}steady pos"+CR$
4355 M%(E)=M%(E)+1:X1=E:E$=MID$(CR$,6,1)
4360 J=1:GOSUB 3090:A$=A$+"{F8}{WHT}aiming"+CR$:M%(E)=M%(E)+1
4365 CR$="{F2}3099"+E$+"check":IF X1>2 THEN CR$="{F2}3699"+E$+"ok"
4366 A$=A$+"{F8}{WHT}breath con"+CR$
4370 J=2:GOSUB 3090:A$=A$+"{F8}{WHT}trigger sq"+CR$:M%(E)=M%(E)+1
4380 J=3:GOSUB 3090:A$=A$+"{F8}{LBLU}shot loc"+CR$+"{F3}C8{F1}{LGRN}":M%(E)=M%(E)+1:POKE V+21,0
4390 GOSUB 3250:A=0:BL=0:IF N<>NT*6 THEN RETURN
4400 IF M%(1)<2 THEN 4420
4410 A=(CL>SL):BL=6-A:B$="two of{F8}{C/RT}your summary scores{F8}{F2}1099were poor.":GOTO4440
4420 IF (M%(1)=0) AND (M%(2)=0) THEN A=1:RETURN
4430 BL=6:B$="one of{F8}{C/RT}your summary scores{F8} {C/RT}was poor or below{F8}{F2}1299average."
4440 A$="{F3}09{CLR}{RED}{F1}{F8}{F8} You are being sent{F8}{C/RT}  back to level"+STR$(CL+A)
4450 A$=A$+"{F8}   because "+B$+"{F1}{LGRN}":GOSUB 3250:RETURN
4460 FOR I=B TO E+1:POKE 16325+I,255:NEXT I
4470 FOR I=B TOE:I1=INT(RND(1)*(E-B))+B
4480 IF PEEK(16325+I1)=255 THEN POKE 16325+I1,I:GOTO 4510
4490 I1=I1+1:IF I1>E THEN I1=B
4500 GOTO 4480
4510 NEXT I:RETURN
4520 GOSUB 3310:A$="L="+STR$(CL)+":T=  0:R="+STR$(RP):GOSUB 5570
```

```
4530 TN=TN+1:IF TN=Z+2 THEN Z=0:RETURN
4540 IF CL<0 THEN 4570
4550 AD=FNW(876):T=PEEK(AD):AD=AD+1
4560 FOR Z=1 TO T:Z1=INT(((PEEK(AD+1)*2)/3)+.5):POKE AD+5,Z1:AD=AD+14:NEXT Z
4570 AD=-16512*(CL=5)-18560*(CL=6)-20608*(CL>6)+2048*((CL>6) AND (RP>0))
4580 Z=AD:IF PEEK(630)>=PEEK(899) THEN Z=Z+1
4590 BB=0:Z=Z+IR:POKE 53280,BC:GOSUB 5710:POKE V+21,0:IF Z=128 THEN RETURN
4600 IF Z=64 THEN SYS NN
4610 IF PEEK(878)<>0 THEN 4610
4620 Z1=(INT(RND(1)*3)+1)*730:FOR Z=1 TO Z1:NEXT Z:GOTO 4530
4630 A$="{F3}09{F6}00{CLR}{F5}13{F7}00{BLK}":GOSUB 3300:POKE 785,FNL(GP):POKE 786,FNH(GP)
4640 POKE 251,FNL(Z):POKE 252,FNH(Z):Z=TN:IF (CL=1) AND (Z5) THEN Z=5
4650 POKE 880,Z:POKE 881,N:Z4=USR(0):IF PEEK(824)=0 THEN RETURN
4660 A$="{F2}0402Here is your"+STR$(PEEK(824))+" round shot group":Z4=Z4/PEEK(824)
4670 IF PEEK(823)>0 THEN A$=A$+"{F2}0903"+STR$(PEEK(823))+" shot(s) off screen"
4680 A$=A$+"{F2}1404"+STR$(Z*30)+" meters"
4690 GOSUB 3250:POKE 53265,43:RETURN
4700 IF N=0 THEN RETURN
4710 Z1=0:Z2=PB*20:Z3=PEEK(838)-1:Z4=0:POKE 785,FNL(GP):POKE 786,FNH(GP)
4720 A$=A$+"{F2}1303{GRN}Hits{F2}2199{RED}Misses{YELO}{F2}2999No Fires{F8}{F8}":GOSUB 3300:Z1=0:Z2=0:Z3=0
4730 AD=Z1+PB*160:FOR I=1 TO 6:POKE 251,FNL(AD):POKE 252,FNH(AD)
4740 POKE 881,N:POKE 880,128+I
4750 Z4=Z4+USR(0):A$="{LBLU}{F2}0599"+STR$(I*30)+" m":Z=PB*3:DA%(I,5+Z)=PEEK(910)
4760 DA%(I,6+Z)=PEEK(911):DA%(I,7+Z)=DA%(I,PB)-PEEK(824)
4770 IF DA%(I,7+Z)<6 THEN DA%(I,7+Z)=0
4780 Z1=Z1+PEEK(910):Z2=Z2+PEEK(911):Z3=Z3+DA%(I,7+Z)
4790 A$=A$+"{GRN}{F2}1399"+STR$(PEEK(910))+"{RED}{F2}2299"+STR$(PEEK(911))+"{YELO}{F2}3299"
4800 A$=A$+STR$(DA%(I,7+Z))+"{F8}{F8}":GOSUB 3300:NEXT I:Z4=Z4/N
4810 A$="{F2}0699{LBLU}TOTAL{F2}1399{GRN}"+STR$(Z1)+"{F2}2299{RED}"+STR$(Z2)+"{F2}3299{YELO}"+STR$(Z3)+"{F8}{F8}"
4820 A$=A$+"{LBLU}{F2}0699Penalties:"+STR$(PEEK(912))+"{F8}"
4830 Z=9:GOSUB 3100:A$=A$+"{LBLU}{F2}0699Shot location: "+RIGHT$(CR$,LEN(CR$)-5)+"{F8}"
4840 B$="{F2}1499{GRN}Standard met{LGRN}":IF Z1<15 THEN B$="{F2}1299{RED}Standard not met{LGRN}"
4850 A$=A$+B$:GOSUB 3250:RETURN
4860 AD=FNW(876):TN=PEEK(AD+3):Z1=DA%(TN,6)
4870 X=((INT(RND(1)*65))*2)+71:POKE AD+1,X:Y=PEEK(AD+2):YJ=PEEK(AD+11)
4880 YJ=YJ+(YJ>127)*256:IF TN=1 THEN POKE AD+15,X+48
4890 Z=((TN-1)*NT)+Z1:POKE 830,Z:AD=Z1+Z*8:POKE AD+7,I
4900 POKE 900,255:POKE AD,255:GOSUB3950
4910 A$="L="+STR$(CL)+"{T="+STR$(I)+"{R="+STR$(RP):GOSUB 5370:POKE I2+7+Z*8,RF
4920 BB=0:Z=IR:GOSUB 5710:POKE 878,0
4930 HI=(PEEK(900)<128):IF Z<>128 THEN DA%(TN,6)=DA%(TN,6)+1
4940 IF Z>63 THEN RETURN
4950 FOR Z=1 TO 5:H%(Z)=0:NEXT Z:N=FNW(907):N=N+(N>128)*(N-128):GOSUB 3000
4960 J=0:GOSUB 3090:A$="{F3}05{CLR}{F5}00{F1}{WHT}steady pos"+CR$:H%(Z)=H%(Z)+1:X1=Z
4970 Z$=MID$(CR$,6,1):J=1:GOSUB 3090:A$=A$+"{WHT}aiming":Z1$=CR$:H%(Z)=H%(Z)+1
4980 X2=Z:CR$="{F2}3099"+Z$+"check":IF X1>2 THEN CR$="{F2}3699"+Z$+"ok"
4990 B$="{WHT}breath con"+CR$:J=2:GOSUB 3090:B$=B$+"{WHT}trigger sq"+CR$
5000 H%(Z)=H%(Z)+1:J=3:GOSUB 3090:IF HI=0 THEN CR$="{F2}3299{RED}miss"
5005 CR$="{F1}{F2}0008{LBLU}shot loc"+CR$:H%(Z)=H%(Z)+1:Y1=Z
5006 IF (X1>3)AND(X2>3)AND(Y1>3)AND(HI=0) THEN Z1$=CR$(3):SD(1,8)=CR(1+PB*4,2)
5010 A$=A$+Z1$+B$:GOSUB 3200:IF (H%(1)=0) AND (Z>2) AND (HI) THEN 5040
5020 GOSUB 4050:IF Z=128 THEN RETURN
5030 IF I<>NT*6 THEN GOSUB 3310
5040 IF HI THEN DA%(TN,5)=DA%(TN,5)+1
5050 RETURN
5060 FOR J=0 TO 3:SD(J,25)=SD(J,25)/3:NEXT J
5070 A=Z:B=Z:Z=25:J=0:GOSUB 3090:A$="{F3}05{F6}00{F5}00{F7}00{CLR}{WHT}{F1}steady pos"+CR$:X1=Z
5080 Z$=MID$(CR$,6,1):J=1:GOSUB 3090:A$=A$+"{WHT}aiming"+CR$
5081 CR$="{F2}3099"+Z$+"check":IF X1>2 THEN CR$="{F2}3699"+Z$+"ok"
5083 A$=A$+"{WHT}breath con"+CR$
5090 J=2:GOSUB 3090:A$=A$+"{WHT}trigger sq"+CR$:J=3:GOSUB 3090
5100 Z=A:X=B:GOSUB 3300:POKE 785,FNL(GP):POKE 786,FNH(GP):POKE 251,FNL(Z)
5110 POKE 252,FNH(Z):POKE 880,5:POKE 881,3:Z=USR(0):A$=""
5120 IF PEEK(823)>0 THEN A$="{BLK}{F2}0921"+STR$(PEEK(823))+" shot(s) off screen"
5130 POKE 823,0:POKE 824,15:POKE 253,13:SYS CO:A$=A$+"{SWLC}{LGRN}":GOSUB 3250
5140 POKE 53265,43:RETURN
5160 A$=A$+"{F2}1303{GRN}Hits{F2}2199{RED}Misses{YELO}{F2}2999No Fires{F8}{F8}":GOSUB 3300
5170 POKE 785,FNL(GP):POKE 786,FNH(GP):HI=0:MI=0:NT=0:Z4=0
5180 FOR I=1 TO 6:POKE 251,FNL(I1):POKE 252,FNH(I1):POKE 881,PEEK(838)
5190 POKE 880,128+I:Z4=Z4+USR(0):A$="{LBLU}{F2}0599"+STR$(I*30)+" m"
5200 HI=HI+PEEK(910):MI=MI+PEEK(911):Z3=DA%(I,4)-PEEK(824):IF Z3<0 THEN Z3=0
5210 NT=NT+Z3:A$=A$+"{GRN}{F2}1399"+STR$(PEEK(910))+"{RED}{F2}2299"+STR$(PEEK(911))
5220 A$=A$+"{YELO}{F2}3299"+STR$(Z3)+"{F8}{F8}":GOSUB 3300:NEXT I:Z4=Z4/PE[illegible]8)
5230 A$="{F2}0699{LBLU}TOTAL{F2}1399{GRN}"+STR$(HI)+"{F2}2299{RED}"+STR$(MI)+"{F2}3299{YELO}"+STR$(NT)+"{F8}{F8}{F8}"
5240 GOSUB 5620:A$=A$+CR$:GOSUB 3250:RETURN
5250 A$="{F3}09{F6}00{F5}00{F7}00{CLR}{RED}{SWLC}{F1}{F2}03024 fundamentals of{F2}0209rifle marksmanship"
5260 A$=A$+"{GRN}{F2}0109{F1}. {F1}Steady Position{F2}0112{F1}. {F1}Aiming{F2}0115{F1}. {F1}Breath Control"
5270 A$=A$+"{F2}0118{F1}. {F1}Trigger Squeeze{F1}{LGRN}":GOSUB 3250:RETURN
5520 A$="{F3}09{F6}00{F5}00{F7}00{CLR}{RED}{SWLC}{F1}{F2}0407You did not meet{F2}0710the standard."
5530 A$=A$+"{F2}0213Prepare to refire.{F1}{LGRN}{F2}0823<Pull trigger to begin>{F3}00":BC=2
5540 GOSUB 3300:GOSUB 3260:RETURN
5550 A$="{F3}09{F6}00{F5}00{F7}00{CLR}{GRN}{SWLC}{F1}{F2}0709Standard met.{F2}1113Continue.{F3}00":GOSUB 3300
5560 FOR Z=1 TO 800:NEXT Z:RETURN
5570 B$="":IF ((HI AND 7)*5)=0 THEN5600
5580 B$=STR$(((HI AND 224)/32)+1)
5590 B$="{F2}3400W="+STR$((HI AND 7)*5)+"{F4}0"+RIGHT$(B$,1)
5600 A$="{HOME}{BLK}{RVON}{SWLC}"+A$+B$+"{F6}0"+CHR$(BC+48):GOSUB 3300:RETURN
5610 A$=A$+"?       (C/LF){C/LF}{C/LF}{C/LF}{C/LF}{C/LF}":GOSUB 3300:B$=""
5620 GET A$:IF A$="" THEN 5620
5630 IF (A$=CHR$(13)) OR (LEN(B$)>3) THEN A$="{C/UP}{F8}":GOSUB 3300:A$=B$:RETURN
5640 IF A$<>CHR$(20) THEN 5670
5650 IF B$<>"" THEN B$=LEFT$(B$,LEN(s$)-1):A$="{C/LF}{C/LF} {C/LF}/{C/LF}":GOSUB 3300
5660 GOTO 5620
5670 IF (A$<"0") OR (A$>"z") THEN 5620
5675 IF (A$>="a") AND (A$<="z") THEN A$=CHR$(ASC(A$) OR 128)
5680 B$=B$+A$:A$=A$+"_{C/LF}":GOSUB 3300:GOTO 5620
5690 FOR Z=12 TO I2+383 STEP 8:POKE Z,255:NEXT Z:RETURN
5700 POKE 49168,BB:SYS 49162:RETURN
5710 POKE 785,10:POKE 786,192:POKE 49168,BB:Z=USR(Z):RETURN
```

```
5720 POKE 251,1:BB=12:GOSUB 5700:A$="":GOSUB 3250:POKE V+21,0
5730 S1=32768:S2=7:GOSUB 3310:GOSUB 3260:RETURN
5740 IF WI=0 THEN RETURN
5750 A$=STR$((WI AND 7)*5):B$=STR$(INT(WI/32)+1):B$="0"+RIGHT$(B$,1)
5760 A$="{F3}09{F5}13{F7}00{CLR}{BLK}{F1}{F2}0601WIND:"+A$+" mph {F4}"+B$:B$="{F2}0705Offset needed"
5770 X=INT(WI/32):IF (X<>0) AND (X<>4) THEN 5800
5780 B$="{F2}0204(wind towards you)":IF X=0 THEN B$="{F2}0004(wind away from you)"
5790 B$=B$+"{F2}0407No offset needed"
5800 A$=A$+B$+"{F1}{F2}3111300 m{F2}19}3150 m{F2}0715$50 m":GOSUB 3300
5810 POKE 251,2:BB=12:GOSUB 5700:A$="{BLK}":GOSUB 3250:POKE V+21,0:RETURN
5820 CR$="{F2}1799{GRN}Expert":IF HI<36 THEN CR$="{F2}1699{LGRN}Sharpshooter"
5830 IF HI<30 THEN CR$="{F2}1699{YELO}Marksman"
5840 IF HI<23 THEN CR$="{F2}1499{RED}Unqualified"
5850 IF (H%(0)=1)AND(HI>22) THEN CR$="  {YELO}Marksman (because you had to refire)"
5860 RETURN
6000 REM THE FOLLOWING IS THE CRITERION FOR GOOD,AVERAGE,BELOW AVERAGE,POOR
6010 DATA 2.5,5,8,10.5:REM SP SUP
6020 DATA 1.6,3.5,6.5,9:REM AIM SUP
6030 DATA 2.5,5,8,10.5:REM TS SUP
6040 DATA 1.6,3.5,6.5,9:REM SL SUP
6050 DATA 4,6.5,9.5,12:REM SP UNS
6060 DATA 2,4.5,8,11:REM AIM UNS
6070 DATA 4,7,11,14.5:REM TS UNS
6080 DATA 2,4.5,8,11:REM SL UNS
6090 DATA 39,35,30,26:REM HITS
6100 DATA 1.5,3.5,6.5,9:REM ACCURACY
6110 DATA 3,2,47,34,3,4.5,23,37,9,4,6,11,13,9
6120 DATA 4,4,11,15,6,3,2,11,15,7,2,1.11,16,6
6130 DATA "{F2}3299{RED}poor","{F2}2299{ORNG}below avg","{F2}2699{YELO}average","{F2}3299{LGRN}good"
6140 DATA "{F2}2299{GRN}excellent"
```

```
        *=3400
;BRM.3400.TXT FOR BRM CARTRIDGE
        JMP   ENABLE
        JMP   DSABLE
        JMP   ROTATE
        JMP   SEEVAL
        JMP   HOTTRG
        JMP   BANG
        JMP   WAIT
        JMP   SCENE
        JMP   MATH
        JMP   ADJLOC
        JMP   RADERR
        JMP   SHTGRP
        JMP   COLORS
        JMP   WHISTL
        JMP   HISCOR
        JMP   EXPTAR
        .OPT  NOL
PRGRUN =49161
SELSUB =49162
;
;              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
HSRT   =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
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
SOLOAD =891
HITSPR =893
SUMX   =894
HSIZE  =895
TIMES  =897
FSTAT  =898
MAXSHT =899
LASTSH =900
HITS   =910
MISSES =911
PENAL  =912
MAXVAL =913
WIND   =914
WDRIFT =915
SHOTRK =917
CURNUM =918
IRQ3   =920
BORCOL =921
LPCMPX =922
LPCMPY =923
RANK   =924
INITS  =925
;NEXT ONE AT 928
CENX   =15872
CENY   =15880
TARNUM =15888
SPECIL =15896
YDRIFT =15904
XDRIFT =15912
JIFFYS =15920
TIMRPL =15928
INSTR  =15936
CDLSTR =15944
DLSTR  =15952
LENSTR =16383
;
ADD    =$B86F
CHKIN  =$FFC6
CHKOUT =$FFC9
CHRIN  =$FFCF
CLOSE  =$FFC3
CLRCHN =$FFCC
CHROUT =$FFD2
DIVIDE =$BB12
FAISC  =$BBC7
FAIFA2 =$BC0F
FA1MEM =$BBD4
FLOAT  =$B391
GETIN  =$FFE4
LOAD   =$FFD5
MEMFA1 =$BBA2
MEMFA2 =$BA8C
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 =-43530 ;UNDERNEATH ROM
XTGA60 =-43580 ;UNDERNEATH ROM
YTGA60 =-43610 ;UNDERNEATH ROM
INFO   =-43640 ;UNDERNEATH ROM
INFO2  =-43984 ;UNDERNEATH ROM
HLDBUF =-44328 ;UNDERNEATH ROM
INIBUF =-44840 ;UNDERNEATH ROM
;NEXT AT 44920
LETBUF =-56789 ;UNDERNEATH ROM
        .MAC  DINC    ;DOUBLE
        INC   ?1      ;PRECISION
        BNE   ?2      ;INCREMENT
        INC   ?1+1
?2      .RND
        .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
        .RND
        .MAC  MOVE
        LDA   ?2
        STA   ?1
        LDA   ?2+1
        STA   ?1+1
        .RND
        .MAC  PRINT
        LDA   #<?1
        LDY   #>?1
        JSR   $AB1E   ;PRINT
        .RND
        .MAC  DADD
        LDA   ?1
        CLC
        ADC   ?2
        STA   ?1
        LDA   ?1+1
        ADC   ?2+1
        STA   ?1+1
        .RND
        .MAC  PLOT
        LDY   #?1
        LDX   #?2
        CLC
        JSR   $FFF0
        .RND
        .MAC  DSPL
        LDA   #?2-?1
        STA   LENSTR
        LDA   #<?1
        STA   $A3
        LDA   #>?1
        STA   $A4
        JSR   LETHL
        .RND
        .OPT  LIST
        .END
        .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   $6334
        LDA   $3265
        AND   #127
        STA   $3265
        LDA   #250
        STA   $3266
        LDA   #5
        STA   $3274
        LDA   $3273
        STA   $3273
        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   $6334
        LDA   #240
        STA   $3274
        LDA   $3273
        STA   $3273
        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/BYTE)
;    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  M5IRQ   ;TARGETS
      INY
      LDX   #7
      LDA   #128
      STA   M3IRQ   ;SPRITE BIT
IRQ160 LDA  M3IRQ
      EOR   #255    ;NOT SPRITE
      STA   M4IRQ   ; BIT
      LDA   V+21
      AND   M3IRQ
      BEQ   IRQ180  ;SPRITE OFF
      LDA   JIFFYS,X
      BMI   IRQ170  ;NO LIMIT
      DEC   JIFFYS,X
      BNE   IRQ170  ;MORE TIME
      LDA   M3IRQ   ;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  M5IRQ   ;ANY TAR NEED
      BEQ   IRQ170  ; DISPLAYING?
      DEC   M5IRQ   ;ONE LESS TAR
      LDA   M4IRQ   ;CLEAR
      AND   V+23    ; EXPAND Y
      STA   V+23
      LDA   M4IRQ   ;CLEAR
      AND   V+29    ; EXPAND X
      STA   V+29
      LDA   M4IRQ   ;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   ($FB),Y ;EXPAND X?
      BPL   IRQ190
      LDA   V+29    ;YES
      ORA   M3IRQ
      STA   V+29
IRQ190 LDA  ($FB),Y ;EXPAND Y?
      AND   #64
      BEQ   IRQ200
      LDA   V+23    ;YES?
      ORA   M3IRQ
      STA   V+23
IRQ200 LDA  ($FB),Y ;RIGHT X?
      AND   #32
      BEQ   IRQ210
      LDA   V+16
      ORA   M3IRQ
      STA   V+16
IRQ210 LDA  ($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   M3IRQ   ; SPRITE
      STA   V+21
IRQ230 LSR  M3IRQ
      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   SHOWUM
IRQ250 LDA  IRQ
      BMI   BEFORE
      JMP   AFTER
BEFORE AND  #5      ;READY TO
      BEQ   IRQ270  ; PROCESS?
IRQ260 JMP  IRQ340
IRQ270 JSR  CNSCRN
      ADDR  $FB,XLPB20
      LDA   CUR2
      AND   #127
      TAY
      LDA   SHOTS
      AND   #1
      BEQ   IRQ280
      TYA
      ORA   #128
      TAY
      INC   $FC
IRQ280 LDA  M1IRQ
      STA   XLPB60,Y
      LDA   M2IRQ
      STA   YLPB60,Y
      INY
      STY   CUR2
      DINC  NUMR60
      DEC   M7IRQ
      BNE   IRQ260
      LDA   #3
      STA   M7IRQ
      LDY   CUR1
      LDA   M17RQ
      STA   ($FB),Y
```

```
        INC   $FC
        INC   $FC
        LDA   H2IRQ
        STA   ($FB),Y
        INY
        STY   CUR1
        DEC   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       ;WAS TRIGGER
        AND   #12       ; BEEN
        BNE   IRQ300    ; RELEASED?
        LDA   56321     ;TRIGR PULL
        CMP   #247
        BNE   IRQ300
        BIT   $ORCOL    ;LIGHTPEN
        BPL   $NPULL    ; ON SCREEN?
        LDA   #4
        STA   53280
IRQ300  JMP   IRQ340
$NPULL  JSR   BANG
        LDA   #0        ;IF ANYONE
        STA   XVAL      ; HAS THE
        STA   XVAL+1    ; REVELATION
        STA   YVAL      ; THAT SORT-
        STA   YVAL+1    ; ING REALLY
        LDA   H1IRQ     ; IS THE
        BEQ   IRQ320    ; ANSWER,
        ASL   A         ; REMOVE
        LDX   #0        ; THESE    ES
        BCC   IRQ310    ; & REINSERT
        INX             ; THE CALL TO
IRQ310  CLC             ; GETXY IN
        ADC   OFFX      ; CTRMOV
        STA   XVAL      ; ---
        TXA             ; ---
        ADC   OFFX+1    ; ---
        STA   XVAL+1    ; ---
        LDA   H2IRQ     ; ---
        CLC             ; ---
        ADC   OFFY      ; ---
        STA   YVAL      ; ---
IRQ320  LDA   IRQ
        AND   #127
        ORA   #76
        STA   IRQ
        LDA   #0        ;FORMERLY 3
        STA   SORTAF
        LDA   #10
        STA   NUMRD
        JSR   INFOFB
        LDY   #5
        LDA   NUMR20
        STA   ($FB),Y
        STA   LASTSH,Y
        INY
        LDA   NUMR20+1
        STA   ($FB),Y
        STA   LASTSH,Y
        LDA   NUMR60
        SEC
        SBC   #1
        STA   LASTSH+7
        LDA   NUMR60+1
        SBC   #0
        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   $HOTS
        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   INC02C
        LDX   CURNUM
        INX
        CPX   #10
        BCC   INC010
        LDX   #0
INC010  STX   CURNUM
        LDX   #0
INC02C  STX   CURNUM+1
        RTS
$HONUM  LDY   #0
        LDX   #80       ;LEAD SPACE?
        LDA   CURNUM
        BEQ   $HN010
        ASL   A
        ASL   A
        ASL   A
        TAX
$HN010  LDA   NUMDAT,X
        STA   4256,Y
        INX
        INY
        CPY   #8
        BCC   $HN010
        LDA   CURNUM+1
        ASL   A
        ASL   A
        ASL   A
        TAX
$HN020  LDA   NUMDAT,X
        STA   8256,Y
        INX
        INY
        CPY   #16
        BCC   $HN020
        RTS
NUMDAT .BYTE
195,153,165,137,153,153,195,255,231,231,19
9,231,231,231,129
       .BYTE
255,195,153,249,243,207,159,129,255,195,15
3,249,227,249,153
       .BYTE
195,255,249,241,225,153,128,249,249,255,12
9,159,131,249,249
       .BYTE
153,195,255,195,153,159,131,153,153,195,25
5,129,153,243,231
       .BYTE
231,231,231,255,195,153,153,195,153,153,19
5,255,195,153,153
       .BYTE
193,249,153,195,255,255,255,255,255,255,25
5,255,255
       .END
       .LIB  REPLAY
;REPLAY FOR RAM CARTRIDGE
REPLAY DEC   H7IRQ
       BEQ   RPL010
       JMP   RPL090
RPL010 LDA   #3
       STA   H7IRQ
       LDY   CUR1
       INY
       CPY   $TOP      ;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   14848,Y ;BULY
       BEQ   RPL050
       CLC
       ADC   OFFY
       STA   H6IRQ
       LDX   #0
       LDA   14592,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 $POST
       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   $61RQ   ;BULY
        SEC
        SBC   $3IRQ   ;BULY-TARY
        CLC
        ADC   #175    ;200-25 SPOST
        STA   V+3     ; OFFSET
RPL080  INY
        STY   CTR1
        CPY   STOP
        BNE   RPL090
        LDA   #0
        STA   IRQ2
RPL090  JMP   $EA31   ;NORMAL IRQ
        .END
;
;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
        LDY   #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
WOTTRG  LDA   $6321   ;REPEAT
        CMP   #255    ; UNTIL
        BNE   WOTTRG  ; 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   #10
        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     ;CHG $/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   14592,Y
        INY
        BNE   SEE010
        INC   $FC
        INC   $FC
SEE020  LDA   ($FB),Y
        STA   14848,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
;MATHFUNCTIONS FOR BRM CARTRIDGE
MATH    JSR   UNPLOT
        LDA   47
        STA   $FB
        LDA   48
        STA   $FC
MAT010  LDY   #0
        LDA   ($FB),Y
        BNE   MAT020
        RTS             ;NO SD ARRAY
MAT020  CMP   #'S
        BNE   MAT030
        INY
        LDA   ($FB),Y
        CMP#  #'D
        BEQ   MAT040
MAT030  INY             ;ADD OFFSET
        LDA   ($FB),Y   ; TO NEXT
        CLC             ; ARRAY
        ADC   $FB       ; (BYTES 2-
        TAX             ; 3) TO $FB
        INY
        LDA   ($FB),Y
        ADC   $FC
        STX   $FB
        STA   $FC
        JMP   MAT010
MAT040  LDA   $FB       ;ASSUME A
        CLC             ; 2-DIM'D
        ADC   #9        ; ARRAY
        STA   HOLDAD    ; WHERE 1ST
        LDA   $FC       ; DIM HOLDS
        ADC   #0        ; EACH SCORE
        STA   HOLDAD+1
        LDX   101       ;SCORE #
        BEQ   MAT060
MAT050  LDA   HOLDAD    ;ASSUMING
        CLC             ; 1ST DIM
        ADC   #20       ; IS 4,
        STA   HOLDAD    ; ADDING 20
        LDA   HOLDAD+1  ; (4*5) WILL
        ADC   #0        ; BE THE
        STA   HOLDAD+1  ; NEXT SCORE
        DEX
        BNE   MAT050
;
;STEADY POSITION SCORE
;
MAT060  LDA   #128      ;DOUBLE X
        STA   HOLD      ; READING
        JSR   MXA010
        SEC
        SBC   $FB
        CLC
        ADC   #1
        CMP   #2
        BCS   MAT070
        LDA   #255
        TAY
        JSR   FLOAT
        JMP   MAT080
MAT070  PHA
        JSR   STDEV
        LDX   HOLDAD
        LDY   HOLDAD+1
        JSR   FAIMEM
        INC   $FE
        LDA   $FB
        STA   MSIZE
        LDA   $FC
        STA   TIMES
        LDA   #0        ;NO DOUBLE
        STA   HOLD      ; Y READING
        PLA
        JSR   STDEV
        JSR   FA1FA2
        LDA   #<CONST
        LDY   #>CONST
        JSR   MEMFA1
        JSR   MLTPLY
        JSR   FA1FA2
        LDA   HOLDAD
        LDY   HOLDAD+1
        JSR   MEMFA1
        JSR   ADD
MAT080  LDX   HOLDAD
        LDY   HOLDAD+1
        JSR   FAIMEM
;
;AIMING SCORE
;
        LDA   #128      ;DOUBLE X
        STA   HOLD      ; READING
        JSR   NEXTAR
        SEC
        SBC   $FB
        CLC
        ADC   #1
        CMP   #2
        BCS   MAT090
        LDA   #255
        TAY
        JSR   FLOAT
        JMP   MAT110
MAT090  STA   MSIZE
        JSR   AVG
        TYA             ;AVG X LOW
        CLC
        ADC   OFFX
        TAY
        TXA             ;AVG X HIGH
        ADC   OFFX+1
        TAX
        TYA
        CLC
        ADC   WDRIFT
        STA   TEMP
        TXA
        ADC   WDRIFT+1
        STA   TEMP+1
        LDX   #0
        LDA   LASTSH+3  ;TARGET X
        ASL   A
        BCC   MAT100
        INX
MAT100  SEC
        SBC   TEMP      ;BULLET X
        TAY
        TXA
        SBC   TEMP+1
        STY   HOLD1
        STA   HOLD1+1
        JSR   FLOAT
        JSR   FA1FA2
        LDY   HOLD1
        LDA   HOLD1+1
        JSR   FLOAT
        JSR   MLTPLY
        LDX   HOLDAD
        LDY   HOLDAD+1
        JSR   FAIMEM
        INC   $FE
        LDA   #0        ;NO DOUBLE
        STA   HOLD      ; Y READING
        JSR   AVG
        TYA
        CLC
        ADC   OFFY
        CLC
        ADC   TRAJ
        STA   TEMP+2
        LDA   LASTSH+4  ;TARGET Y
        SEC
        SBC   TEMP+2    ;BULLET Y
        TAY
        LDA   #0
        SBC   #0
        STY   HOLD1
        STA   HOLD1+1
        JSR   FLOAT
        JSR   FA1FA2
        LDY   HOLD1
        LDA   HOLD1+1
        JSR   FLOAT
        JSR   MLTPLY
        JSR   FA1FA2
        LDA   #<CONSQ
        LDY   #>CONSQ
        JSR   MEMFA1
        JSR   MLTPLY
        JSR   FA1FA2
        LDA   HOLDAD
        LDY   HOLDAD+1
        JSR   MEMFA1
        JSR   ADD
        LDA   102       ;FA1 SIGN
        AND   #127
        STA   102       ;ABS
        JSR   SQRT
MAT110  LDX   HOLDAD
        LDY   HOLDAD+1
        JSR   FAIMEM
;
;TRIGGER SQUEEZE SCORE
;
        JSR   NEXTAR
        CMP   $FB
        BNE   MAT120
        LDA   #255
        TAY
        JSR   FLOAT
        JMP   MAT130
MAT120  JSR   RANGE
        LDA   MSIZE
        PHA
        LDA   TIMES
        PHA
        INC   $FE
        LDA   #0        ;NO DOUBLE
        STA   HOLD      ; Y READING
        JSR   RANGE
        ADDR  $FD,YLPA60
        LDA   #0
        STA   $FB
        LDA   #2
        STA   $FC
        JSR   RAN010
        JSR   FA1FA2
        LDA   #<CONST
        LDY   #>CONST
        JSR   MEMFA1
        JSR   MLTPLY
        JSR   FA1FA2
        PLA
        STA   TIMES
        PLA
        STA   MSIZE
        LDA   #128      ;DOUBLE X
        STA   HOLD      ; READING
        ADDR  $FD,XLPA60
        JSR   RAN010
        JSR   ADD
MAT130  LDX   HOLDAD
        LDY   HOLDAD+1
        JSR   FAIMEM
;
;SHOT LOCATION SCORE
;
        LDA   LASTSH
        CMP   #255
        BNE   MAT140
        TAY
        JSR   FLOAT
        JMP   MAT150
MAT140  JSR   NEXTAR
        JSR   RADERR
MAT150  LDX   HOLDAD
        LDY   HOLDAD+1
        JSR   FAIMEM
        LDA   TEMP      ;X AVG
        LSR   TEMP+1
        ROR   A
        LDY   TEMP+2    ;Y AVG
        JSR   FLOAT
        RTS
NEXTAR  LDA   HOLDAD    ;SET UP
        CLC             ; FOR NEXT
        ADC   #5        ; ELEMENT
        STA   HOLDAD    ; IN ARRAY
        LDA   HOLDAD+1  ; ---
        ADC   #0        ; ---
        STA   HOLDAD+1  ; ---
MXA010  ADDR  $FD,XLPB60
        LDA   SHOTS     ;WHICH
        AND   #1        ; LIGHTPEN
        BNE   MXA020    ; BUFFER?
        LDA   $FD
        CLC
        ADC   #128
        STA   $FD
        LDA   $FE
        ADC   #0
        STA   $FE
MXA020  LDA   HOLDAD
        LDY   HOLDAD+1
        JSR   MEMFA1
        JSR   UNPLOT
        LDA   100
        STA   $FB
        STA   MSIZE
        LDA   101
        STA   $FC
        STA   TIMES
        RTS
MUL2    LDX   #0
        BIT   HOLD
        BPL   ML2010
        ASL   A
        BCC   ML2010
        INX
ML2010  RTS
;CONSTANTS (AND CONSTANTS SQUARED)
;FOR SCALING THE Y COORDINATE
;(320/23.5)/(200/19.5)
;1.33 = 129,170,61,112,164
;1.76 = 129,225,71,176,20
;320/200
;1.6  = 129,76,204,204,205
;2.56 = 130,163,215,10,61
CONST   .BYTE 129,76,204,204,205
CONSQ   .BYTE 130,163,215,10,61
ADJLOC  LDA   71
        STA   HOLDAD
```

```
        LDA   72
        STA   HOLDAD+1
        LDA   1
        PHA
        LDA   #54
        STA   1
        LDY   #23
ADJ010  LDA   INFO,Y
        STA   TEMP,Y
        DEY
        BPL   ADJ010
        PLA
        STA   1
        LDX   #0
        STX   HOLDX
ADJ020  LDA   HOLDAD
        LDY   HOLDAD+1
        JSR   MEMFA1
        JSR   UNPLOT
        LDA   100       ;X VALUE
        ASL   A
        LDY   #0
        BCC   ADJ030
        INY
ADJ030  CLC
        ADC   OFFX
        STA   LASTSH+1  ;BULLET X
        TYA
        ADC   OFFX+1
        LSR   A
        ROR   LASTSH+1
        LDA   101       ;Y VALUE
        CLC
        ADC   OFFY
        STA   LASTSH+2  ;BULLET Y
        LDX   HOLDX
        LDA   TEMP+3,X
        STA   LASTSH+3
        LDA   TEMP+4,X
        STA   LASTSH+4
        JSR   RADERR
        LDX   HOLDAD
        LDY   HOLDAD+1
        JSR   FA1MEM
        LDA   HOLDAD    ;SET UP
        CLC             ; FOR
        ADC   #10       ; SHOT LOC
        STA   HOLDAD    ; SCORE
        LDA   HOLDAD+1
        ADC   #0
        STA   HOLDAD+1
        LDX   HOLDX
        LDA   TEMP+1,X  ;BULLET X
        ASL   A
        LDY   #0
        BCC   ADJ040
        INY
ADJ040  CLC
        ADC   OFFX
        STA   TEMP+1,X
        TYA
        ADC   OFFX+1
        LSR   A
        ROR   TEMP+1,X
        LDA   TEMP+1,X
        STA   INFO+1,X
        STA   INFO2+1,X
        STA   LASTSH+1
        LDA   TEMP+2,X  ;BULLET Y
        CLC
        ADC   OFFY
        STA   INFO+2,X
        STA   INFO2+2,X
        STA   LASTSH+2
        JSR   RADERR
        LDX   HOLDAD
        LDY   HOLDAD+1
        JSR   FA1MEM
        LDA   HOLDAD    ;SET UP
        CLC             ; FOR NEXT
        ADC   #10       ; AIMING
        STA   HOLDAD    ; SCORE
        LDA   HOLDAD+1
        ADC   #0
        STA   HOLDAD+1
        LDA   HOLDX
        CLC
        ADC   #8
        STA   HOLDX
        CMP   #24
        BCS   ADJ050
        JMP   ADJ020
ADJ050  RTS
STDEV   PHA
        LDA   #0
        TAY
        JSR   FLOAT
        JSR   FA1SC
        LDA   #0
        STA   SUMX
        STA   SUMX+1
        STA   HOLD3
        STA   HOLD4
STDL10  LDA   1
        PHA
        LDA   #54
        STA   1
        LDY   TIMES
        LDA   (8FD),Y
        TAX
        BIT   HOLD4
        BMI   STDL20
        STA   HOLD3
        LDA   #128
        STA   HOLD4
        TXA
STDL20  CMP   HOLD3
        BEQ   STDL30
        LDA   HOLD4
        ORA   #64
        STA   HOLD4
        TXA
STDL30  JSR   MUL2
        STA   HOLD1
        STX   HOLD2
        CLC
        ADC   SUMX
        STA   SUMX
        LDA   SUMX+1
        ADC   HOLD2
        STA   SUMX+1
        PLA
        STA   1
        LDY   HOLD1
        LDA   HOLD2
        JSR   FLOAT
        JSR   FA1FA2
        LDY   HOLD1
        LDA   HOLD2
        JSR   FLOAT
        JSR   MLTPLY
        LDA   #85C
        LDY   #0
        JSR   MEMFA2
        JSR   ADD
        JSR   FA1SC
        DEC   TIMES
        LDY   TIMES
        BMI   STDL40
        CPY   MSIZE
        BCS   STDL10
STDL40  PLA
        STA   MSIZE
        BIT   HOLD4
        BVC   STDL50
        LDA   #0
        TAY
        JSR   FLOAT
        RTS
STDL50  LDY   SUMX
        LDA   SUMX+1
        JSR   FLOAT
        JSR   FA1FA2
        LDY   SUMX
        LDA   SUMX+1
        JSR   FLOAT
        JSR   MLTPLY
        JSR   FA1FA2
        LDA   #0
        LDY   MSIZE
        JSR   FLOAT
        JSR   DIVIDE
        LDA   #85C
        LDY   #0
        JSR   MEMFA2
        JSR   SUBTRT
        JSR   FA1FA2
        LDA   #0
        LDY   MSIZE
        DEY
        JSR   FLOAT
        JSR   DIVIDE
        LDA   102       ;FA1 SIGN
        AND   #127
        STA   102       ;ABS
        JSR   SQRT
        RTS
AVG     LDA   #0
        STA   SUMX
        STA   SUMX+1
        STA   HOLD1
        STA   HOLD1+1
        LDA   1
        PHA
        LDA   #54
        STA   1
        LDY   #FC
AVG010  LDA   (8FD),Y
        JSR   MUL2
        CLC
        ADC   SUMX
        STA   SUMX
        TXA
        ADC   SUMX+1
        STA   SUMX+1
        DEY
        BMI   AVG020
        CPY   #FB
        BCS   AVG010
AVG020  PLA
        STA   1
AVG030  LDA   SUMX+1
        BNE   AVG040
        LDA   MSIZE
        CMP   SUMX
        BCC   AVG040
        BNE   AVG050
AVG040  LDA   SUMX
        SEC
        SBC   MSIZE
        STA   SUMX
        LDA   SUMX+1
        SBC   #0
        STA   SUMX+1
        DINC  HOLD1
        JMP   AVG030
AVG050  LDY   HOLD1
        LDX   HOLD1+1
        RTS
RANGE   LDA   #0
        STA   MSIZE
        LDA   #255
        STA   TIMES
RAN010  LDA   1
        PHA
        LDA   #54
        STA   1
        LDY   #FC
RAN020  LDA   (8FD),Y
        CMP   TIMES
        BCS   RAN030
        STA   TIMES
RAN030  CMP   MSIZE
        BCC   RAN040
        STA   MSIZE
RAN040  DEY
        BMI   RAN050
        CPY   #FB
        BCS   RAN020
RAN050  PLA
        STA   1
        LDA   MSIZE
        SEC
        SBC   TIMES
        JSR   MUL2
        TAY
        TXA
        JSR   FLOAT
        RTS
RADERR  LDX   #0
        LDA   LASTSH+1  ;BULLET X
        ASL   A
        BCC   RAD010
        INX
RAD010  STA   CODE3
        STX   CODE4
        LDX   #0
        LDA   LASTSH+3  ;TARGET X
        ASL   A
        BCC   RAD020
        INX
RAD020  SEC
        SBC   CODE3
        STA   CODE3
        TXA
        SBC   CODE4
        STA   CODE4
        LDY   CODE3
        JSR   FLOAT
        JSR   FA1FA2
        LDY   CODE3
        LDA   CODE4
        JSR   FLOAT
        JSR   MLTPLY
        JSR   FA1SC
        LDX   #0
        LDA   LASTSH+2  ;BULLET Y
        SEC
        SBC   LASTSH+4  ;TARGET Y
        BCS   RAD030
        DEX
RAD030  STA   CODE3
        STX   CODE4
        LDY   CODE3
        LDA   CODE4
        JSR   FLOAT
        JSR   FA1FA2
```

```
        LDY   CODE3
        LDA   CODE4
        JSR   FLOAT
        JSR   MLTPLY
        JSR   FA1FA2
        LDA   #<CONSQ
        LDY   #>CONSQ
        JSR   MEMFA1
        JSR   MLTPLY
        LDA   #$5C
        LDY   #0
        JSR   MEMFA2
        JSR   ADD
        LDA   102       ;FA1 SIGN
        AND   #127
        STA   102       ;ABS
        JSR   SQRT
        RTS
        .END
        .LIB  SHOTGROUPS
;SHOTGROUPS IN BRM.3400.TXT
;PROCEDURE EXPTAR
;POKE HOLD3,TAR#.  THE CENTER OF
; THE TARGET IS PLACED AT 160,100
; IF THE CENTER OF MASS OFFSETS
; ARE CHANGED AGAIN, A NEW START
; MUST BE CALCULATED AS:
; (160-OFF*FACTOR,100-OFF*FACTOR).
; EXPDATA IS SPRITE POINTER,
; STARTX,STARTY,FACTOR.
EXPTAR LDA   #0
        STA   XVAL+1
        STA   $5D
        STA   CHOICE
        LDA   HOLD3
        AND   #127
        SEC
        SBC   #1
        ASL   A
        ASL   A
        TAY
        LDA   EXPDAT+1,Y
        STA   XVAL
        STA   HOLDX
        LDA   EXPDAT+2,Y
        STA   YVAL
        LDA   EXPDAT+3,Y
        STA   CODE2
        LDA   EXPDAT,Y
        LDY   #6
EXP010  ASL   A
        ROL   $5D
        DEY
        BNE   EXP010
        STA   $5C
EXP020  LDY   #0
EXP030  LDA   #3
        STA   HOLDA
EXP040  LDA   #128
        STA   CODE1
EXP050  LDA   ($5C),Y
        AND   CODE1
        BEQ   EXP070
        LDA   CODE2
        STA   CODE3
        STA   CODE4
EXP060  JSR   BITPLT
        DINC  XVAL
        DEC   CODE3
        BNE   EXP060
        LDA   XVAL
        SEC
        SBC   CODE2
        STA   XVAL
        LDA   XVAL+1
        SBC   #0
        STA   XVAL+1
        LDA   CODE2
        STA   CODE3
        INC   YVAL
        DEC   CODE4
        BNE   EXP060
        LDA   YVAL
        SEC
        SBC   CODE2
        STA   YVAL
EXP070  LDA   XVAL
        CLC
        ADC   CODE2
        STA   XVAL
        LDA   XVAL+1
        ADC   #0
        STA   XVAL+1
        LSR   CODE1
        BNE   EXP050
        INY
        CPY   #63
        BCS   EXP080
        DEC   HOLDA
        BNE   EXP040
        LDA   HOLDX
        STA   XVAL
        LDA   #0
        STA   XVAL+1
        LDA   YVAL
        CLC
        ADC   CODE2
        STA   YVAL
        JMP   EXP030
EXP080  LDA   $5C       ;$0 METER
        CMP   #<2496    ; TARGET?
        BNE   EXP090
        LDA   $5D
        CMP   #>2496
        BNE   EXP090
        ADDR  $5C,2560
        LDA   #161
        STA   XVAL
        STA   HOLDX
        LDA   #0
        STA   XVAL+1
        LDA   #6C
        STA   YVAL
        JMP   EXP020
EXP090  RTS
EXPDAT  .BYTE 39,113,66,2
        .BYTE 41,116,26,4
        .BYTE 35,127,61,3
        .BYTE 36,116,40,4
        .BYTE 37,105,25,5
        .BYTE 38, 94, 4,6
;
;PROCEDURE SHTGRP (SHOTGROUPS)
;POKE $FB WITH START ADDRESS.
; POKE HOLD3,TAR# (ADD 128 IF
; ONLY HIT/MISS INFO IS DESIRED.)
; POKE HOLD4,WITH # OF SHOTS.
; AT END: HITS, MISSES ARE SET,
; HOLD1 WILL HAVE # UNGRAPHABLE
; SHOTS, HOLD2 WILL HAVE TOTAL
; # SHOTS FIRED AT THIS TARGET
SHTGRP  LDA   #255
        STA   CHOICE
        LDA   #0
        STA   HITS
        STA   MISSES
        STA   HOLD1
        STA   HOLD2
        STA   HOLDY
        LDA   HOLD3
        AND   #127
        TAX
        SEC
        SBC   #1
        ASL   A
        ASL   A
        TAY
        LDA   EXPDAT+3,Y
        STA   CODE2
        STX   CODE1
        CPX   #3
        BCS   SHT010
        LSR   CODE2
SHT010  LDY   #100
        LDA   #255
SHT020  STA   TEMP,Y
        DEY
        BPL   SHT020
        LDA   1
        PHA
        LDA   #54
        STA   1
SHT030  LDY   #0
        LDA   ($FB),Y
        AND   #127
        CMP   CODE1
        BNE   SHT070
        INC   HOLD2
        LDA   ($FB),Y
        BMI   SHT060
        INC   HITS
        JMP   SHT050
SH...J  INC   MISSES
SHT050  LDY   #4
SHT060  LDA   ($FB),Y
        STA   LASTSH,Y
        DEY
        BPL   SHT060
        PLA
        PHA
        STA   1
        LDX   #<TEMP+100
        LDY   #>TEMP
        JSR   FA1MEM
        JSR   RADERR
        JSR   FA1FA2
        LDA   #<TEMP+100
        LDY   #>TEMP
        JSR   MEMFA1
        JSR   ADD
        LDA   #54
        STA   1
        BIT   HOLD3
        BPL   SHT080
SHT070  JMP   SHT160
SHT080  LDA   #0
        STA   XVAL
        STA   XVAL+1
        STA   YVAL
        STA   YVAL+1
        LDA   LASTSH+3 ;TARX
        LDX   #0
        ASL   A
        BCC   SHT090
        INX
SHT090  STA   $FD
        STX   $FE
        LDA   LASTSH+1 ;BULX
        LDX   #0
        ASL   A
        BCC   SHT100
        INX
SHT100  SEC
        SBC   $FD
        STA   $FD
        TXA
        SBC   $FE
        STA   $FE
        LDX   CODE2
SHT110  LDA   XVAL
        CLC
        ADC   $FD
        STA   XVAL
        LDA   XVAL+1
        ADC   $FE
        STA   XVAL+1
        DEX
        BNE   SHT110
        LDA   XVAL
        CLC
        ADC   #160
        STA   XVAL
        TAX
        LDA   XVAL+1
        ADC   #0
        STA   XVAL+1
        CMP   #2
        BCS   SHT150
        CMP   #0
        BNE   SHT120
        CPX   #10
        BCC   SHT150
        BCS   SHT130
SHT120  CPX   #54
        BCS   SHT150
SHT130  LDY   HOLDY
        LSR   A
        TXA
        ROR   A
        STA   TEMP,Y
        LDA   LASTSH+2 ;BULY
        SEC
        SBC   LASTSH+4 ;TARY
        STA   $FD
        LDA   #0
        SBC   #0
        STA   $FE
        LDX   CODE2
SHT140  LDA   YVAL
        CLC
        ADC   $FD
        STA   YVAL
        LDA   YVAL+1
        ADC   $FE
        STA   YVAL+1
        DEX
        BNE   SHT140
        LDA   YVAL
        CLC
        ADC   #100
        STA   YVAL
        TAX
        LDA   YVAL+1
        ADC   #0
        STA   YVAL+1
        BNE   SHT150
        CPX   #176
        BCS   SHT150
        CPX   #24
        BCC   SHT150
        LDY   HOLDY
        INY
        TXA
        STA   TEMP,Y
        INY
        STY   HOLDY
        JMP   SHT160
```

```
SHT150 INC   HOLD1
SHT160 LDA   $FB
       CLC
       ADC   #8
       STA   $FB
       LDA   $FC
       ADC   #0
       STA   $FC
       DEC   HOLD4
       BEQ   SHT170
       JMP   SHT030
SHT170 PLA
       STA   1
       BIT   HOLD3
       BPL   SHT180
       RTS
SHT180 LDY   #0
SHT190 LDX   #0
       LDA   TEMP,Y
       CMP   #255
       BEQ   SHT240
       STA   $FB
       ASL   A
       BCC   SHT200
       INX
SHT200 SEC
       SBC   #1
       STA   XVAL
       STA   HOLDA
       TXA
       SBC   #0
       STA   XVAL+1
       STA   HOLDX
       LDA   TEMP+1,Y
       STA   $FC
       SEC
       SBC   #1
       STA   YVAL
       LDA   #3
       STA   CODE3
       STA   CODE4
SHT210 JSR   BITPLT
       DEMC  XVAL
       DEC   CODE3
       BNE   SHT210
       LDA   HOLDA
       STA   XVAL
       LDA   HOLDX
       STA   XVAL+1
       INC   YVAL
       LDA   #3
       STA   CODE3
       DEC   CODE4
       BNE   SHT210
       INY
       INY
       TYA
       TAX
SHT220 LDA   TEMP,X
       CMP   #255
       BEQ   SHT190
       CMP   $FB
       BNE   SHT230
       LDA   TEMP+1,X
       CMP   $FC
       BNE   SHT230
       INC   TEMP,X
       INC   TEMP+1,X
SHT230 INX
       INX
       JMP   SHT220
SHT240 JSR   EXPTAR
       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   $A3     ;7-(X AND 7)
       TAX
       LDA   #1
       CPX   #0
       BEQ   PL060
PL050  ASL   A       ;2^(7-(X
       DEX           ; AND 7))
       BNE   PL050
PL060  BIT   CHOICE
       BPL   PL070
       ORA   ($FD),Y
       JMP   PL080
PL070  EOR   ($FD),Y
PL080  STA   ($FD),Y
       LDY   HOLDY
       RTS
       .END
;PROCEDURE HISCOR
;THIS PROCEDURE CALLES THE HISCOR
; PROCEDURE ON CH0BK1.  THE ONLY
; REASON IT IS HERE IS BECAUSE
; THE INITIALS BUFFER IS UNDER
; ROM AND ROUTINES RUNNING OFF
; THE EPROM CANNOT BANK OUT ROM
; WITHOUT CRASHING.  IT MOVES
; THE INITIALS BUFFER TO TEMP
; BEFORE CALLING AND MOVES IT
; BACK AFTER THE CALL.  CODE1
; AND FA1 MUST BE SET (IF APP.)
; BEFORE CALLING THIS ROUTINE.
HISCOR LDA   1
       PHA
       LDA   #54
       STA   1
       LDY   #79
HIS010 LDA   INIBUF,Y
       STA   TEMP,Y
       DEY
       BPL   HIS010
       PLA
       STA   1
       LDA   #24
       STA   49168
       JSR   SELSUB
       LDY   #79
HIS020 LDA   TEMP,Y
       STA   INIBUF,Y
       DEY
       BPL   HIS020
       RTS
       .OPT  LIST
       .END
       *=32768
;MLCH0BK1.TXT
       JMP   CTRMOV
       JMP   RANDOM
       JMP   DODATA
       JMP   GETXY
       JMP   HELPSC
       JMP   REKEEP
       JMP   YESNO
       JMP   DESCRP
       JMP   HISCOR
       .LIB  MACROS-VARS
       .END
       .OPT  NOL
BANG   =3415
DSABLE =3403
ENABLE =3400
MOTTRG =3412
ROTATE =3406
WAIT   =3418
WHISTL =3436
PRGNUM =49161
LSTMSL =49185
       .LIB  CTRMOV
;CTRMOV IN MLCH0BK1.TXT
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   CTR410
CTR030 LDA   SORTAF
       BNE   CTR030
;      JSR   GETXY   ;NO :HERE 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
       DEMC  WDRIFT
CTR080 LDA   YDRIFT,X
       STA   TRAJ
       LDA   TIMEPL,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   $HOTS
        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+15
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   $FC       ; ORIGINAL
        LDA   SHOTS     ; SHOT)
        CLC
        ADC   #1
        STA   SHOTRX
        LDY   #7
CTR210  LDA   LASTSH,Y
        STA   ($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       ;S 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 HIT AFTER
        AND   #128      ; EACH TARGET
        BNE   CTR320    ; LEAVE IT UP
        LDA   CODE3     ;TURN OFF
        EOR   #255      ; HIT TARGET
        AND   V+21      ; BOM
        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
        LDA   SHOTS
        CMP   MAXSHT
        BCC   CTR400
        LDA   IRQ
        ORA   #1
        STA   IRQ
        LDY   #87
CTR380  LDA   NOAMMO,Y
        STA   15776,Y
        DEY
        BPL   CTR380
        LDY   #10
        LDA   #13
CTR390  STA   1972,Y
        DEY
        BPL   CTR390
CTR400  SEC
        JSR   CKSPCL
        BCS   CTR460
        LDA   IRQ
        AND   #1
        BEQ   CTR410
        JMP   CTR020
CTR410  LDA   IRQ
        AND   #192
        BEQ   CTR460
        LDA   $6321
        CMP   #127
        BNE   CTR420
        LDA   #128      ;BREAK KEY
        STA   PSTAT     ; FLAG
        JMP   CTR460
CTR420  LDA   OFFSCR
        BNE   CTR430
        JMP   CTR020
CTR430  AND   #1        ;WHISTLE
        BEQ   CTR450    ; FLAG
        LDA   V+21      ;SPRITE ON?
        AND   #254
        BEQ   CTR440
        JSR   WHISTL
CTR440  LDA   OFFSCR    ;CLEAR FLAG
        AND   #254
```

```
        STA   OFFSCR
CTR450  LDA   OFFSCR
        EOR   #255
        AND   V+21
        STA   V+21
        CLC
        JSR   CKSPCL
        BCS   CTR460
        JMP   CTR010
CTR460  LDA   IRQ
        AND   #15
        STA   IRQ
        MOVE  $FB,TARORD
        LDY   #0
        LDA   ($FB),Y
        CMP   #254
        BNE   CTR470
        DINC  TARORD
CTR470  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   HCLDX
        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   #666
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
NOAMMO  .BYTE
60,102,102,102,102,102,60,0,102,102,102,10
2,102,102,60,0,126
        .BYTE
24,24,24,24,24,24,0,0,0,0,0,0,0,0,60,102
,102,102,102,102
        .BYTE
60,0,126,96,96,120,96,96,96,0,0,0,0,0,0,0,
0,0,24,60,102,126
        .BYTE
102,102,102,0,99,119,127,107,99,99,99,0,99
,119,127,107,99,99
        .BYTE
99,0,60,102,102,102,102,102,60,0
        .END
        .OPT  NOL
        .LIB  DATAITEMS
;DATAITEMS FOR BRN PROGRAM
DODATA  JSR   WHCDAT
        LDA   #<DITEMS
        CLC
        ADC   STOP
        STA   STOP
        LDA   #>DITEMS
        ADC   STOP+1
        STA   STOP+1
        ADDR  $FD,$2480
        LDY   #0
DOD010  LDA   ($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,$2480
RAN010  JSR   WHCDAT
        LDA   16325,X
        BMI   RAN020
        JSR   SEARCH
        INX
        BNE   RAN010
RAN020  LDY   #0
        LDA   #255
        STA   ($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
1,0,150,5,37,12,255,0,0,0,4,0,0,6,15,254,2
55
;EXTENDED PARTEST 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, 10,  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, 10,  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, 34
```

```
   .BYTE    0,169,  1, 40,204,  8,  0,
0,  0,  0,255, 32,  0, 34,254
   .BYTE 1,  0,159,  2, 41,204, 11,  0,
0,  0,  1,254,  0, 12, 37,254
   .BYTE 1,  0,170,  3, 35, 12, 13,  0,
0,  0,  2,254,  0,  6, 13,254
   .BYTE 1,  0,160,  4, 36, 12, 16,  0,
0,  0,  3,255,  0,  6, 15,254
   .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, 37,254
   .BYTE 1,131,160,  3, 35, 12, 10,  0,
0,  0,  2,254,  0,  6, 13,254
   .BYTE 2, 49,165,  1, 39,204,  6,  0,
0,  0,  0,255, 32, 24, 34
   .BYTE   97,165,  1, 40,204,  6,  0,
0,  0,  0,255, 48,  0, 34,254
   .BYTE 1,131,160,  3, 35, 12, 10,  0,
0,  0,  2,254,  0,  6, 13,254
   .BYTE 1,231,115,  6, 38, 12, 16,  0,
0,  0,  5,  1,  0,  6, 16,254
   .BYTE 1,219,145,  4, 36, 12, 12,  0,
0,  0,  3,255,  0,  6, 15,254
   .BYTE 1,195,130,  5, 37, 12, 14,  0,
0,  0,  4,  0,  0,  6, 15,254
   .BYTE 2, 49,165,  1, 39,204,  4,  0,
0,  0,  0,255, 32, 24, 34
   .BYTE   97,165,  1, 40,204,  6,  0,
0,  0,  0,255, 48,  0, 34,254
   .BYTE 1,219,145,  4, 36, 12, 12,  0,
0,  0,  3,255,  0,  6, 15,254
   .BYTE 1,195,130,  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, 37
   .BYTE  195,130,  5, 37, 12, 18,  0,
0,  0,  4,  0,  0,  6, 15,254
   .BYTE 1,219,145,  4, 36, 12, 12,  0,
0,  0,  3,255,  0,  6, 15,254
   .BYTE 2,131,160,  3, 35, 12, 20,  0,
0,  0,  2,254,  0,  6, 13
   .BYTE  231,115,  6, 38, 12, 20,  0,
0,  0,  5,  1,  0,  6, 16,254
   .BYTE 2,243,152,  2, 41,204, 16,  0,
0,  0,  1,254,  0, 12, 37
   .BYTE  219,145,  4, 36, 12, 16,  0,
0,  0,  3,255,  0,  6, 15,254
   .BYTE 3, 49,165,  1, 39,204, 12,  0,
0,  0,  0,255, 32, 24, 34
   .BYTE   97,165,  1, 40,204, 12,  0,
0,  0,  0,255, 48,  0, 34
   .BYTE  243,152,  2, 41,204, 12,  0,
0,  0,  1,254,  0, 12, 37,254
   .BYTE 1,131,160,  3, 35, 12, 10,  0,
0,  0,  2,254,  0,  6, 13,254
   .BYTE 1,195,130,  5, 37, 12, 16,  0,
0,  0,  4,  0,  0,  6, 15,254
   .BYTE 3, 49,165,  1, 39,204, 16,  0,
0,  0,  0,255, 32, 24, 34
   .BYTE   97,165,  1, 40,204, 16,  0,
0,  0,  0,255, 48,  0, 34
   .BYTE  151,160,  3, 35, 12, 16,  0,
0,  0,  2,254,  0,  6, 13,254
   .BYTE 1,243,152,  2, 41,204, 10,  0,
0,  0,  1,254,  0, 12, 37,254
   .BYTE 1,131,160,  3, 35, 12, 12,  0,
0,  0,  2,254,  0,  6, 13,254
   .BYTE 3, 49,165,  1, 39,204, 16,  0,
0,  0,  0,255, 32, 24, 34
   .BYTE   97,165,  1, 40,204, 16,  0,
0,  0,  0,255, 48,  0, 34
   .BYTE  219,145,  4, 36, 12, 16,  0,
0,  0,  3,255,  0,  6, 15,254
   .BYTE 1,243,152,  2, 41,204, 10,  0,
0,  0,  1,254,  0, 12, 37,254
   .BYTE 2,131,160,  3, 35, 12, 24,  0,
0,  0,  2,254,  0,  6, 13
   .BYTE  219,145,  4, 36, 12, 24,  0,
0,  0,  3,255,  0,  6, 15,254
   .BYTE 1,231,115,  6, 38, 12, 18,  0,
0,  0,  5,  1,  0,  6, 16,254
   .BYTE 2,243,152,  2, 41,204, 16,  0,
0,  0,  1,254,  0, 12, 37
   .BYTE  131,160,  3, 35, 12, 16,  0,
0,  0,  2,254,  0,  6, 13,254
   .BYTE 1,131,160,  3, 35, 12, 16,  0,
0,  0,  2,254,  0,  6, 13,254
   .BYTE 2,243,152,  2, 41,204, 20,  0,
0,  0,  1,254,  0, 12, 37
   .BYTE  219,145,  4, 36, 12, 20,  0,
0,  0,  3,255,  0,  6, 15,254
   .BYTE 2,131,160,  3, 35, 12, 24,  0,
0,  0,  2,254,  0,  6, 13
   .BYTE  195,130,  5, 37, 12, 24,  0,
0,  0,  4,  0,  0,  6, 15,254
   .BYTE 2,243,152,  2, 41,204, 18,  0,
0,  0,  1,254,  0, 12, 37
   .BYTE  219,145,  4, 36, 12, 18,  0,
0,  0,  3,255,  0,  6, 15,254
;ATTACK/RETREAT 1 (913-1806)
;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, 16,138,130
;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, 15,136,134
;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, 15,131
   .BYTE 1,211,160,  3, 35, 12, 12,  0,
0,  0,  2,254,  1,  6, 13,131
   .BYTE 1,103,145,  4, 36, 12,  8,  0,
0,  0,  3,255,  1,  6, 15,136,140
;SCREEN 5
   .BYTE 1,253,160,  3, 35, 12,  8,  0,
0,  0,  2,254,  1,  6, 13,  0
   .BYTE 1, 43,152,  2, 41,204,  6,  0,
0,  0,  1,254,  1, 12, 37,134,134
;SCREEN 6
   .BYTE 1,181,160,  3, 35, 12, 12,  0,
C,  0,  2,254,  1,  6, 13,131
   .BYTE 1,251,152,  2, 41,204,  8,  0,
0,  0,  1,254,  1, 12, 37,131
   .BYTE 2, 41,165,  1, 39,204,  4,  0,
0,  0,  0,255, 33, 24, 34
   .BYTE   89,165,  1, 40,204,  4,  0,
0,  0,  0,255, 49,  0, 34,132,130
   .BYTE 3, 47,152,  2, 41,204, 20,  0,
0,  0,  1,254,  3, 12, 37
   .BYTE  189,152,  2, 41,204, 20,  0,
0,  0,  1,254,  3, 12, 37
   .BYTE   19,160,  3, 35, 44, 20,  0,
0,  0,  2,254,  3,  6, 13,140
   .BYTE 3,107,152,  2, 41,204,  8,  0,
0,  0,  1,254,  2, 12, 37
   .BYTE  207,165,  1, 39,204,  8,  0,
0,  0,  0,255, 34, 24, 34
   .BYTE  255,165,  1, 40,204,  8,  0,
0,  0,  0,255, 50,  0, 34,136,130
   .BYTE 5, 29,165,  1, 39,204, 20,  0,
0,  0,  0,255, 36, 24, 34
   .BYTE   77,165,  1, 40,204, 20,  0,
0,  0,  0,255, 52,  0, 34
   .BYTE   91,152,  2, 41,204, 20,  0,
0,  0,  1,254,  4, 12, 37
   .BYTE  153,152,  2, 41,204, 20,  0,
0,  0,  1,254,  4, 12, 37
   .BYTE  255,152,  2, 41,204, 20,  0,
0,  0,  1,254,  4, 12, 37,144
   .BYTE 1,221,160,  3, 35, 12,  4,  0,
0,  0,  2,254,  1,  6, 13,132,134
;SCREEN 9
   .BYTE 1,205,152,  2, 41,204, 12,  0,
0,  0,  1,254,  1, 12, 37,  0
   .BYTE 1,101,160,  3, 35, 12, 10,  0,
0,  0,  2,254,  1,  6, 13,  0
   .BYTE 1,205,160,  3, 35, 12,  8,  0,
0,  0,  2,254,  1,  6, 13,136,140
;SCREEN 10
   .BYTE 1,111,160,  3, 35, 12,  8,  0,
0,  0,  2,254,  1,  6, 13,  0
   .BYTE 1,255,145,  4, 36, 12,  6,  0,
0,  0,  3,255,  1,  6, 15,134,134
;SCREEN 11
   .BYTE 3,241,160,  3, 35, 12, 16,  0,
0,  0,  2,254,  3,  6, 13
   .BYTE   99,145,  4, 36, 12, 16,  0,
0,  G,  3,255,  3,  6, 15
   .BYTE  215,145,  4, 36, 12, 16,  0,
0,  0,  3,255,  3,  6, 15,132
   .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
        .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    YSN040
DISKON  LDA    #1
        JSR    CLOSE
        JSR    CLRCHN
        ADDR   $A3,S1
        LDA    #S2-S1
        STA    LENSTR
        JSR    LETML
YESNO   ADDR   $A3,S2
        LDA    #S3-S2
        STA    LENSTR
        JSR    LETML
        LDX    #'Y
        JSR    GKEY
        BCS    YSN010
        ADDR   $A3,S3
        LDA    #S4-S3
        STA    LENSTR
        JSR    LETML
        LDX    #'N
        JSR    GKEY
        BCC    YESNO
YSN010  LDA    #S5-S4
        STA    LENSTR
        LDY    #0
        LDA    $FD
        CMP    #'N
        BEQ    YSN020
        DEY
        TYA
        JSR    FLOAT
        ADDR   $A3,S4
        JMP    YSN030
YSN020  TYA
        JSR    FLOAT
        ADDR   $A3,S5
YSN030  JSR    LETML
YSN040  LDY    #255
        JSR    WAIT
        LDY    #255
        JSR    WAIT
        LDY    #255
        JSR    WAIT
        JSR    NOTTRG
        RTS
GKEY    LDA    #$FF
        STA    $A3
        LDA    #$80
        STA    $A4
        STX    $FD
GKEY1   JSR    GETIN
        CMP    #0
        BNE    GKEY2
        DEC    $A3
        BNE    GKEY4
        DEC    $A4
        BNE    GKEY4
        CLC
        RTS
GKEY2   CMP    #'Y
        BEQ    GKEY3
        CMP    #'N
        BNE    GKEY1
GKEY3   STA    $FD
        SEC
        RTS
GKEY4   LDA    56321
        CMP    #247
        BNE    GKEY1
        SEC
        RTS
        .OPT   NOL
S1      .BYTE 137,'0617',196,'O YOU WANT TO
KEEP RECORDS?',17,17,13
```

```
82      .BYTE 32,133,'    ',5,217,197,211,'
      ',191,133
        .BYTE 157,157,'NO',165,13
83      .BYTE 32,133,'
',191,157,157,157,133,'YES ',133,'      '
        .BYTE 5,206,207,133,165,13
84      .BYTE 5,133,'
',217,197,211,'    '
85      .BYTE 5,133,'          ',206,207,'
'

        .OPT   LIST
        .END
        .OPT   NOL
        .LIB   DESCRIPTIONS
DESCRP  ADDR   $FB,DESC
        LDX    HOLD1
        BEQ    DES050
        LDY    #0
DES010  DEX
        BEQ    DES020
        LDA    ($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    TEMP,X
        INX
        DEY
        BNE    DES030
        LDA    ($FB),Y
        STA    HOLD1
        INY
DES040  LDA    ($FB),Y
        STA    TEMP,X
        INX
        INY
        DEC    HOLD1
        BNE    DES040
        STX    LENSTR
        ADDR   $A3,TEMP
        JSR    LETML
        RTS
DES050  INC    HOLD1
        LDA    HOLD1
DES060  CMP    $FD     ;START
        BNE    DES070
        LDX    #14     ;LIGHT BLUE
        STX    646
DES070  CMP    $FE     ;FINAL
        BCC    DES080
        BEQ    DES080
        LDX    #0      ;BLACK
        STX    646
DES080  LDX    #32
        STX    TEMP
        STX    TEMP+3
        ORA    #48
        STA    TEMP+1
        CMP    #58
        BNE    DES090
        LDA    #'1
        STA    TEMP
        LDA    #'0
        STA    TEMP+1
DES090  LDA    #':
        STA    TEMP+2
        LDY    #0
        LDA    ($FB),Y
        PHA
        TAY
DES100  LDA    ($FB),Y
        STA    TEMP+3,Y
        DEY
        BNE    DES100
        PLA
        TAY
        INC    $FB
        CLC
        ADC    $FB
        PHA
        LDA    $FC
        ADC    #0
        PHA
        LDA    $FD
        PHA
        LDA    $FE
        PHA
        LDA    #13
        STA    TEMP+4,Y
        INY
        INY
        INY
        INY
        INY
        ADDR   $A3,TEMP
        STY    LENSTR
        JSR    LETML
        PLA
        STA    $FE
        PLA
        STA    $FD
        PLA
        STA    $FC
        PLA
        STA    $FB
        INC    HOLD1
        LDA    HOLD1
        CMP    #10
        BEQ    DES110
        JMP    DES060
DES110  RTS
DESC    .BYTE
31,201,'NTRODUCTION/',211,'UPPORTED
',208,'OSITION'
        .BYTE
33,201,'NTRODUCTION/',213,'NSUPPORTED
',208,'OSITION'
        .BYTE 32,212,'IKED
',212,'ARGETS/',211,'UPPORTED
',208,'OSITION'
        .BYTE 34,212,'IKED
',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 17,210,'APID ',210,'ECORD
',198,'IRE'
        .BYTE 11,195,'OMBAT ',198,'IRE'
        .END
        .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
        .LIB   HELPSCREENS
;PROCEDURE HELPSC
;POKE APPROPRIATE NUMBER IN $FB
; BEFORE CALLING
HELPSC  LDA    $FB
        CMP    #1
        BEQ    HLP010
        JMP    HLP040
;TARGET ARRAY ($FB-1)
HLP010  DSPL   HMS010,HMS020
        LDY    #7
HLP020  LDA    #12
        STA    V+39,Y
        TYA
        CLC
        ADC    #34
        STA    2040,Y
        DEY
        BNE    HLP020
        LDA    #24
        STA    V+16
        LDA    #224
        STA    V+23
        STA    V+29
        LDY    #13
HLP030  LDA    $PLOC1,Y
        STA    V+2,Y
        DEY
        BPL    HLP030
        LDA    #254
        STA    V+21
        RTS
HLP040  CMP    #2
        BEQ    HLP050
        RTS
;WIND INSTRUCTIONS ($FB=2)
HLP050  LDA    #39     ;50 METER
        STA    2044    ; TARGET IN
        LDA    #40     ; SP 4-5
        STA    2045
        LDA    #35     ;150 METER
        STA    2046    ; IN SP 6
        LDA    #38     ;300 METER
        STA    2047    ; IN SP 7
        LDY    #3
        LDA    #12
HLP060  STA    V+43,Y
        DEY
        BPL    HLP060
        LDY    #3      ;EIGHT POST
HLP070  LDA    #63     ; IN SP 1-3
        STA    2040,Y
        LDA    #0
        STA    V+39,Y
        DEY
        BNE    HLP070
        LDA    #62
        STA    V+23
        STA    V+29
        LDA    #136
        STA    V+16
        LDA    #0
        STA    V+28
        LDY    #13
HLP080  LDA    $PLOC2,Y
        STA    V+2,Y
        DEY
        BPL    HLP080
        LDX    #0      ;50 M
        JSR    GETWND
        CLC
```

```
        ADC   V+2
        STA   V+2
        LDX   #16       ;150 M
        JSR   GETWND
        CLC
        ADC   V+4
        STA   V+4
        LDX   #40       ;300 M
        JSR   GETWND
        CLC
        ADC   V+6
        STA   V+6
        LDA   #254
        STA   V+21
GETWND  LDA   WIND
        AND   #96
        BEQ   GTW020
        STX   $FB
        LDA   WIND
        AND   #7
        CLC
        ADC   $FB
        TAX
        LDA   WIND
        AND   #32
        BNE   GTW010
        LDA   WTABLE,X
        LSR   A
        LSR   A
        LSR   A
        LSR   A
        JMP   GTW020
GTW010  LDA   WTABLE,X
        AND   #15
GTW020  BIT   WIND
        BMI   GTW030
        EOR   #255
GTW030  RTS
SPLOC1  .BYTE
166,170,226,160,10,150,50,140,40,169,88,16
9,130,159
SPLOC2  .BYTE
72,178,173,158,17,131,48,169,96,169,185,17
0,29,140
MMS010  .BYTE
134,'09',144,147,135,'13',133,137,'0706',2
12,'ARGET RANGES'
        .BYTE
133,137,'07215',207,'M',137,'14191',207,20
7,'M'
        .BYTE
137,'201815',207,'M',137,'25172',207,207,'
M'
        .BYTE
137,'301625',207,'M',137,'35143',207,207,'
M',144
MMS020  =*
        .END
        .LIB  HIGHSCORE
;PROCEDURE HISCOR
;ALL OF THESE ROUTINES ARE CALLED
; FROM HISCOR IN THE 3400-8191
; MACHINE LANGUAGE AREA.
HISCOR  LDX   CODE1
        BNE   HIS010
        JSR   GETINI    ;CODE1=0
        RTS
;ROUTINE TO GET LAST FIRER'S
; RANKING.  SCORE SHOULD BE IN
; FA1 AND CODE1=1.  RANK WILL BE
; RETURNED IN THE VARIABLE RANK.
HIS010  DEX
        BNE   HIS060
        LDA   #1
        STA   RANK
        LDY   #3
HIS020  LDX   #0
HIS030  LDA   TEMP,Y
        CMP   97,X
        BCC   HIS050
        BNE   HIS040
        INX
        INY
        CPX   #5
        BCC   HIS030
        BCS   HIS050
HIS040  LDA   RANK
        ASL   A
        ASL   A
        ASL   A
        CLC
        ADC   #3
        TAY
        INC   RANK
        CPY   #83
        BCC   HIS020
HIS050  RTS
HIS060  DEX
        BEQ   HIS070
        JMP   HIS130
;ROUTINE TO STORE INITIALS AND
; PRESENT A LITTLE DEMO.  SET
; CODE1=2, PUT SCORE IN FA1 AND
; PUT RANKING IN RANK BEFORE
; CALLING.
HIS070  LDA   RANK
        STA   HOLD1
        LDA   #11
        STA   RANX
        JSR   HIS130
        LDA   #43       ;SIGHT POST
        STA   2047
        LDA   #0
        STA   V+16
        STA   V+28
        LDA   #128
        STA   V+23
        STA   V+29
        LDA   #1
        STA   V+46
        LDX   #0
HIS080  TXA
        ASL   A
        TAY
        LDA   GUNSPL,Y
        STA   V+14
        LDA   GUNSPL+1,Y
        STA   V+15
        LDA   #128
        STA   V+21
        LDY   #1
        JSR   HSWAIT
        INX
        CPX   #10
        BCC   HIS080
HIS090  STX   HOLD2
        TXA
        SEC
        SBC   #1
        ASL   A
        TAY
        LDA   GUNSPL,Y
        STA   V+14
        LDA   GUNSPL+1,Y
        STA   V+15
        LDY   #1
        JSR   HSWAIT
        JSR   HSBUMP
        JSR   BANG
        JSR   HIS140
        LDY   #3
        JSR   HSWAIT
        LDX   HOLD2
        CPX   HOLD1
        BEQ   HIS100
        DEX
        BNE   HIS090
HIS100  STX   RANK
        DEX
        TXA
        ASL   A
        ASL   A
        ASL   A
        TAY
        LDX   #0
HIS110  LDA   INITS,X
        STA   TEMP,Y
        INX
        INY
        CPX   #3
        BCC   HIS110
        LDX   #0
HIS120  LDA   97,X
        STA   TEMP,Y
        INX
        INY
        CPX   #5
        BCC   HIS120
        LDA   #0
        STA   V+21
        JSR   HIS140
        LDY   #8
        JSR   HSWAIT
        RTS
;ROUTINE WHICH PRINTS THE INITIALS
; FROM TEMP.  SET CODE1=3 AND PUT
; RANKING IN RANK BEFORE CALLING.
; IF THIS ROUTINE IS CALLED AT
; HIS140, ONLY THE INITIALS ARE
; PRINTED (THE SCREEN IS NOT
; CLEARED AND 'HIGH SCORE' IS NOT
; REPRINTED).
HIS130  ADDR  $A3,PINMS1
        LDA   #PINMS2-PINMS1
        STA   LENSTR
        JSR   LETML
HIS140  LDY   #BOARD1-PINMS2
        STY   LENSTR
HIS150  LDA   PINMS2-1,Y
        STA   8192-1,Y
        DEY
        BNE   HIS150
        LDY   #7
        LDX   #0
        LDA   #1
        STA   HOLD
        LDA   HOLD
HIS160  CMP   RANK
        BNE   HIS170
        LDA   #5
        STA   8192,Y
HIS170  LDA   TEMP,X
        STA   8192+4,Y
        LDA   TEMP+1,X
        STA   8192+5,Y
        LDA   TEMP+2,X
        STA   8192+6,Y
        TXA
        CLC
        ADC   #8
        TAX
        TYA
        CLC
        ADC   #12
        TAY
        INC   HOLD
        LDA   HOLD
        CMP   #11
        BCC   HIS160
        ADDR  $A3,8192
        JSR   LETML
        RTS
GETINI  ADDR  $A3,BOARD1
        LDA   #BOARD2-BOARD1
        STA   LENSTR
        JSR   LETML
        LDA   #15
        STA   HOLD
INI010  ADDR  $A3,BOARD2
        LDA   #BOARD3-BOARD2
        STA   LENSTR
        JSR   LETML
        DEC   HOLD
        BNE   INI010
        ADDR  $A3,BOARD3
        LDA   #BOARD4-BOARD3
        STA   LENSTR
        JSR   LETML
        LDA   #3
        STA   HOLD
INI020  ADDR  $A3,BOARD4
        LDA   #BOARD5-BOARD4
        STA   LENSTR
        JSR   LETML
        DEC   HOLD
        BNE   INI020
        ADDR  $A3,BOARD5
        LDA   #BOARDE-BOARD5
        STA   LENSTR
        JSR   LETML
        JSR   DSABLE
        LDA   #0
        STA   V+23
        STA   V+29
        STA   V+38
        STA   198
        STA   HOLD1
        LDA   #1
        STA   V+39
        STA   V+28
        LDA   #42       ;CROSS
        STA   2040
INI030  LDX   #0
        LDA   $3267     ;X READING
        ASL   A
        STA   XVAL
        BCC   INI040
        INX
INI040  STX   XVAL+1
        LDA   $3266     ;Y READING
        STA   YVAL
        LDA   XVAL
        CLC
        ADC   OFFX
        STA   XVAL
        LDA   XVAL+1
        ADC   OFFX+1
        STA   XVAL+1
        LDA   XVAL
        SEC
        SBC   #10
        STA   XVAL
        LDA   XVAL+1
        SBC   #0
        STA   XVAL+1
        LDA   YVAL
        CLC
```

```
        ADC   OFFY
        STA   YVAL
        LDA   #0
        ADC   OFFY+1
        STA   YVAL+1
        LDA   YVAL
        SEC
        SBC   #9
        STA   YVAL
        LDA   YVAL+1
        SBC   #0
        STA   YVAL+1
        LDX   XVAL
        LDA   XVAL+1
        BNE   INI060
        CPX   #33
        BCS   INI050
        LDX   #33
INI050  STX   XVAL
        JMP   INI080
INI060  CPX   #58
        BCC   INI070
        LDX   #57
INI070  STX   XVAL
INI080  LDX   YVAL
        LDA   YVAL+1
        BNE   INI090
        CPX   #44
        BCS   INI100
INI090  LDX   #44
        JMP   INI110
INI100  CPX   #173
        BCC   INI110
        LDX   #172
INI110  STX   YVAL
        LDA   XVAL
        STA   V
        LDA   XVAL+1
        AND   #1
        STA   V+16
        LDA   YVAL
        STA   V+1
        LDA   #1
        STA   V+21
        LDA   XVAL
        SEC
        SBC   #33
        STA   XVAL
        LDA   XVAL+1
        SBC   #0
        STA   XVAL+1
        LDX   #0
INI120  LDA   XVAL+1
        BNE   INI130
        LDA   XVAL
        CMP   #41
        BCC   INI140
INI130  LDA   XVAL
        SEC
        SBC   #41
        STA   XVAL
        LDA   XVAL+1
        SBC   #0
        STA   XVAL+1
        INX
        JMP   INI120
INI140  STX   XVAL
        LDX   #0
        LDA   YVAL
        SEC
        SBC   #44
INI150  CMP   #33
        BCC   INI160
        SEC
        SBC   #33
        INX
        JMP   INI150
INI160  STX   YVAL
        ADDR  $FD,$864
        LDX   XVAL
        BEQ   INI180
INI170  LDA   $FD
        CLC
        ADC   #40
        STA   $FD
        LDA   $FE
        ADC   #0
        STA   $FE
        DEX
        BNE   INI170
INI180  LDX   YVAL
        BEQ   INI200
INI190  LDA   $FD
        CLC
        ADC   #<1280
        STA   $FD
        LDA   $FE
        ADC   #>1280
        STA   $FE
        DEX
        BNE   INI190
INI200  JSR   ADDRFB
        LDY   #15
INI210  LDA   ($FD),Y
        STA   ($FB),Y
        DEY
        BPL   INI210
        LDA   $FB
        CLC
        ADC   #<320
        STA   $FB
        LDA   $FC
        ADC   #>320
        STA   $FC
        LDA   $FD
        CLC
        ADC   #<320
        STA   $FD
        LDA   $FE
        ADC   #>320
        STA   $FE
        LDY   #15
INI220  LDA   ($FD),Y
        STA   ($FB),Y
        DEY
        BPL   INI220
        LDA   $6321
        CMP   #247
        BEQ   INI230
        LDA   HOLD1
        BNI   INI240
        JMP   INI320
INI230  JSR   BANG
INI240  LDA   HOLD1
        AND   #127
        STA   HOLD1
        LDX   YVAL
        CPX   #3
        BCC   INI290
        DEC   XVAL
        LDA   XVAL
        CMP   #255
        BNE   INI280
        LDA   HOLD1
        BNE   INI250
        JMP   INI320
INI250  JSR   ADDRFB
        LDY   #15
        LDA   #0
INI260  STA   ($FB),Y
        DEY
        BPL   INI260
        LDA   $FB
        CLC
        ADC   #<320
        STA   $FB
        LDA   $FC
        ADC   #>320
        STA   $FC
        LDY   #11
        LDA   #0
INI270  STA   ($FB),Y
        DEY
        BPL   INI270
        LDA   #255
        LDY   #15
        STA   ($FB),Y
        DEY
        STA   ($FB),Y
        DEY
        STA   ($FB),Y
        DEY
        STA   ($FB),Y
        DEY
        STA   ($FB),Y
        LDY   #7
        STA   ($FB),Y
        DEY
        STA   ($FB),Y
        DEY
        STA   ($FB),Y
        DEY
        STA   ($FB),Y
        DEC   HOLD1
        JMP   INI320
INI280  CMP   #5
        BNE   INI290
        LDA   #32
        JMP   INI310
INI290  LDX   #7
        LDA   #0
INI300  CLC
        ADC   YVAL
        DEX
        BNE   INI300
        CLC
        ADC   XVAL
        CLC
        ADC   #193
INI310  LDY   HOLD1
        STA   INIT8,Y
        INC   HOLD1
        CPY   #2
        BCS   INI340
INI320  LDA   $6321
        CMP   #255
        BNE   INI320
        LDX   #64
        LDA   #255
INI330  SEC
        SBC   #1
        BNE   INI330
        DEX
        BNE   INI330
        JMP   INI030
INI340  LDA   $6321
        CMP   #255
        BNE   INI340
        LDA   #0
        STA   V+21
        STA   198
        RTS
ADDRFB  LDA   HOLD1
        ASL   A
        ASL   A
        ASL   A
        ASL   A
        ASL   A
        CLC
        ADC   #<15440
        STA   $FB
        LDA   #>15440
        ADC   #0
        STA   $FC
        RTS
HSWAIT  PHA
        TXA
        PHA
        LDA   #255
        TAX
HSW010  SEC
        SBC   #1
        BNE   HSW010
        DEX
        BNE   HSW010
        DEY
        BNE   HSW010
        PLA
        TAX
        PLA
        RTS
HSBUMP  TXA
        PHA
        SEC
        SBC   #1
        ASL   A
        ASL   A
        ASL   A
        PHA
        TAY
        CPY   #72
        BEQ   HSB020
        LDX   #8
HSB010  LDA   TEMP,Y
        STA   TEMP+8,Y
        INY
        DEX
        BNE   HSB010
HSB020  LDX   #3
        PLA
        TAY
        LDA   #32
HSB030  STA   TEMP,Y
        INY
        DEX
        BNE   HSB030
        PLA
        TAX
        RTS
GUNSPL .BYTE
97,79,249,79,97,103,249,103,97,127,249,127
        .BYTE 97,151,249,151,97,175,249,175
PINMS1 .BYTE
134,'05',147,135,'00',136,'00',133,28
        .BYTE 137,'0102LEVEL 9 HIGH SCORES'
PINMS2 .BYTE 133,14,137,'0306',30,' 1
',137,'2206',30,' 2    '
        .BYTE 137,'0309',30,' 3
',137,'2209',30,' 4    '
        .BYTE 137,'0312',30,' 5
',137,'2212',30,' 6    '
        .BYTE 137,'0315',30,' 7
',137,'2215',30,' 8    '
        .BYTE 137,'0318',30,' 9
',137,'2218',30,'10    ',134,'08'
BOARD1 .BYTE
5,134,'09',147,135,'13',136,'13  '
```

```
        .BYTE
176,192,192,192,192,178,192,192,192,192,17
8,192,192,192,192
        .BYTE
178,192,192,192,192,178,192,192,192,192,17
8,192,192,192,192
        .BYTE 178,192,192,192,192,176,140
BOARD2 .BYTE '  ',221,'    ',221,'
',221,'    ',221,'    ',221,'    '
        .BYTE 221,'    ',221,'    ',221,140
BOARD3 .BYTE '
',173,192,192,192,192,177,192,192,192,192,
177,192,192,192
        .BYTE
192,177,192,192,192,192,177,192,192,192,19
2,177,192,192,192
        .BYTE
192,177,192,192,192,192,189,137,'0004'
BOARD4 .BYTE '
',171,192,192,192,192,219,192,192,192,192,
219,192,192,192
        .BYTE
192,219,192,192,192,192,219,192,192,192,19
2,219,192,192,192
        .BYTE
192,219,192,192,192,192,179,140,140,140,14
0
BOARD5 .BYTE
137,'0402',133,193,29,29,29,194,29,29,29,1
95,29,29,29
        .BYTE
196,29,29,29,197,29,29,29,198,29,29,29,199
,137,'0406'
        .BYTE
200,29,29,29,201,29,29,29,202,29,29,29,203
,29,29,29
        .BYTE
204,29,29,29,205,29,29,29,206,137,'0410'
        .BYTE
207,29,29,29,208,29,29,29,209,29,29,29,210
,29,29,29
        .BYTE
211,29,29,29,212,29,29,29,213,137,'0414'
        .BYTE
95,29,29,29,214,29,29,29,215,29,29,29,216,
29,29,29
        .BYTE 217,29,29,29,218,29,29,29,32
        .BYTE 151,137,'0220',197,'NTER
YOUR',140,'  INITIALS'
        .BYTE
137,'2622',175,32,175,32,175,134,'00'
BOARDE =*
        .END
        .OPT  LIST
        .END
        *=49152
        .OPT  NOL
        .LIB  MACROS-VARS
        .END
        .OPT  NOL
ENABLE =$3400
DSABLE =$3403
SCENE  =$342C
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 CHOBK1
; BY POKING ADDRESS FROM BASIC
SELSUB LDA   #1
       STA   $DFFF
       JSR   32768
       LDA   #32
       STA   $DFFF
       RTS
       JMP   BRM       ;BRM
       JMP   PRGSEL    ;PRGSEL
       JMP   LETERS
       JMP   LETHL
       JMP   OPENSC
ESTART STX   $D016
       JSR   $FDA3     ;IOINIT
       JSR   $FD50     ;RAMTAS
       JSR   $FD15     ;RESTOR
       JSR   $FF5B     ;CINT
       CLI
       JSR   $E453     ;COPYVECTORS
       JSR   $E3BF     ;INIT
       ADDR  $FD,57344
       ADDR  STOP,LETBUF
       ADDR  $A3,39515
       JSR   MOVEIT-16384
       LDY   #204
       LDA   #0
STU010 STA   819,Y
       DEY
       BNE   STU010
       LDY   #79
STU020 LDA   INIDAT-16384,Y
       STA   INIBUF,Y
       DEY
       BPL   STU020
BRM    LDA   #0
       STA   53280
       STA   53281
       STA   $DFFF
       LDA   #63
       STA   53265
       ADDR  $FD,49152
       ADDR  STOP,FIN
       ADDR  $A3,$8000
       JSR   MOVEIT-16384
       JMP   NEWLOC
NEWLOC LDA   #$34      ;DISABLE RUN
       STA   808       ; STOP KEY
       LDA   #128      ;DISABLE CDR
       STA   657       ; SHIFT KEYS
       LDA   #<16384   ;BASIC START
       STA   641       ;OS BOTTON L
       LDA   #>16384
       STA   44        ;BASIC BOT H
       STA   642       ;OS BOTTON H
       LDA   #<16385
       STA   43        ;BASIC BOT L
       ADDR  55,40960  ;BASIC TOP
       LDX   #0
       JSR   PREPAR
       JSR   OPENSC
       JMP   STU060
PRGSEL LDA   PRGNUM
       CMP   #1        ;CALIBRATION
       BNE   STU030
       LDX   #CALIB-DATA
       JSR   PREPAR
       JMP   STU050
STU030 CMP   #2        ;SIGHT PIC
       BNE   STU040
       LDX   #SITPIC-DATA
       JSR   PREPAR
       JMP   STU050
STU040 LDX   #SEROSC-DATA
       JSR   PREPAR
       JMP   STU060
STU050 LDA   #21        ;TEXT SCREEN
       STA   53272
       LDA   #27
       STA   53265
       LDA   #147       ;CLR SCREEN
       JSR   $FFD2
STU060 LDA   #32        ;RAM
       STA   $DFFF
       LDA   #0         ;CLEAR KEY-
       STA   198        ; BOARD BUF
       JSR   $A871
       JMP   $A7AE      ;RUN
OPENSC JSR   DSABLE
       LDA   #0
       STA   PRGNUM
       STA   198
       STA   V+16
       STA   V+21
       STA   V+23
       STA   V+28
       STA   V+29
       STA   V+39
       LDA   #36
       STA   2040
       ADDR  $FD,35350
       LDA   #0
       STA   $FB        ;CHIP 0/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
       RTS
PREPAR LDA   DATA,X
       STA   101
       AND   #63
       STA   $DFFF
       INY
       LDA   DATA,X
       STA   $FD
       INX
       LDA   DATA,X
       STA   $FE
       INX
       LDA   DATA,X
       STA   STOP
       BIT   101
       BPL   STU070
       STA   45
STU070 INX
       LDA   DATA,X
       STA   STOP+1
       BIT   101
       BPL   STU080
       STA   46
STU080 INX
       LDA   DATA,X
       STA   $A3
       INX
       LDA   DATA,X
       STA   $A4
       INX
       JSR   MOVEIT
       BIT   101
       BVC   PREPAR
       RTS
MOVEIT LDY   #0
STU090 LDA   ($A3),Y
       STA   ($FD),Y
       DINC  $A3
       DINC  $FD
       LDA   $FD
       CMP   STOP
       BNE   STU090
       LDA   $FE
       CMP   STOP+1
       BNE   STU090
       RTS
LETERS JSR   LETSB1
       JSR   57344
       JSR   LETSB2
       RTS
LETHL  LDY   LENSTR
STU100 LDA   ($A3),Y
       STA   LETBUF,Y
       DEY
       CPY   #255
       BNE   STU100
       ADDR  $A3,LETBUF
       JSR   LETSB1
       JSR   57347
       JSR   LETSB2
       RTS
LETSB1 JSR   DSABLE
       SEI
       LDA   1
       AND   #253
       STA   1
       RTS
LETSB2 LDA   1
       ORA   #7
       STA   1
       JSR   ENABLE
       RTS
DATA   .BYTE 3       ;BRM.3400
       .WORD 3400,7916,32768
       .BYTE 2       ;BRM.SPRITES
       .WORD 2176,3008,34210
       .BYTE 4       ;BRM.BAS (1)
       .WORD 16384,24576,32768
       .BYTE 5       ;BRM.BAS (2)
       .WORD 24576,32768,32768
       .BYTE 198     ;BRM.BAS (3)
       .WORD 32768,37979,32768
CALIB  .BYTE 194     ;CALIBRATION
       .WORD 16384,17555,32768
SITPIC .BYTE 9       ;SA.3400
       .WORD 3400,4017,40343
       .BYTE 7       ;SA.SPRITES
       .WORD 2176,2816,40320
       .BYTE 201     ;SIGNTALIGN
       .WORD 16384,23493,32768
SEROSC .BYTE 7       ;SERO.3400
       .WORD 3400,4486,39171
       .BYTE 7       ;SERO.SPRITES
       .WORD 2176,2239,40257
       .BYTE 203     ;SERO.BAS
       .WORD 16384,19707,32768
       .LIB  INTRO
;INTRO FOR BRM CARTRIDGE
START  LDA   $A2
       AND   #3
       BNE   INL030
       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
INL030  LDA   $A1
        AND   #2
        BEQ   INL060
        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   #200
        STA   V+1      ;Y: SPRITE 0
        LDA   #1
        STA   V+16     ;RIGHT X
        STA   V+21     ;SPRITE CTRAG
        LDA   #0
        STA   $A1
        LDA   CTRREG
        ORA   #8
        STA   CTRREG
        ADDR  $A3,1336
        LDX   #9
INL040  LDY   #6
        LDA   #33
INL050  STA   ($A3),Y
        DEY
        BPL   INL050
        LDA   $A3
        CLC
        ADC   #40
        STA   $A3
        LDA   $A4
        ADC   #0
        STA   $A4
        DEX
        BNE   INL040
        LDA   #<13960
        STA   $FD
        LDA   #<13968
        STA   $C3
        LDA   #>13968
        STA   $FE
        STA   $C4
        LDA   #5
        STA   GUNDL
INL060  LDA   CTRREG
        AND   #1
        BEQ   INL120
        LDA   $A2      ; SCROLL
        AND   #7
        BNE   INL120
        LDY   #0
INL070  LDA   15880,Y
        STA   15872,Y
        INY
        BNE   INL070
INL080  LDA   16136,Y
        STA   16128,Y
        INY
        CPY   #56
        BNE   INL080
        LDY   #7
INL090  LDA   ($FB),Y
        CMP   #254
        BEQ   INL100
        STA   16184,Y
        DEY
        BPL   INL090
        LDA   $FB
        CLC
        ADC   #8
        STA   $FB
        LDA   $FC
        ADC   #0
        STA   $FC
        JMP   INL120
INL100  LDA   #0
        STA   16184,Y
        DEY
        BPL   INL100
        DEC   SCRCT1
        BNE   INL120
        LDA   #3
        STA   SCRCT2
        EOR   CTRREG
        STA   CTRREG
        LDY   #39
        LDA   #33
INL110  STA   1984,Y
        DEY
        BPL   INL110
INL120  LDA   CTRREG
        AND   #2
        BEQ   INL140
        LDY   #252     ;BLINK
INL130  LDA   PTTB-1,Y
        STA   15919,Y
        DEY
        BNE   INL130
        LDA   #180
        STA   SCRCT1
        LDA   CTRREG
        EOR   #66
        STA   CTRREG
INL140  LDA   CTRREG
        AND   #8
        BEQ   INL160
        LDY   #0       ;KICK LEFT
        JSR   SPRUPD
INL150  LDA   ($C3),Y
        STA   ($FD),Y
        INY
        CPY   #240
        BNE   INL150
        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   INL160
        LDA   #10
        STA   GUNDL
        LDA   CTRREG
        EOR   #24
        STA   CTRREG
INL160  LDA   CTRREG
        AND   #16
        BEQ   INL170
        JSR   SPRUPD
        DEC   GUNDL    ;GUN DELAY
        BNE   INL170
        LDA   CTRREG
        EOR   #48
        STA   CTRREG
        LDA   #5
        STA   GUNDL
        LDA   #<13952
        STA   $FD
        LDA   #<13960
        STA   $C3
        LDA   #>13960
        STA   $FE
        STA   $C4
INL170  LDA   CTRREG
        AND   #32
        BEQ   INL210
        LDA   #240     ;KICK RIGHT
        JSR   SPRUPD
INL180  LDA   ($FD),Y
        STA   ($C3),Y
        DEY
        CPY   #255
        BNE   INL180
        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   INL210
        LDA   CTRREG
        AND   #223
        STA   CTRREG
        LDA   #0
        STA   V+21     ;SPRITE CTRAG
        ADDR  $A3,1336
        LDX   #9
INL190  LDY   #6
        LDA   #177
INL200  STA   ($A3),Y
        DEY
        BPL   INL200
        LDA   $A3
        CLC
        ADC   #40
        STA   $A3
        LDA   $A4
        ADC   #0
        STA   $A4
        DEX
        BNE   INL190
INL210  LDA   CTRREG
        AND   #64
        BEQ   INL270
        DEC   SCRCT1   ;DELAY FOR
        BNE   INL270
        DEC   SCRCT2
        BNE   INL230
        LDA   CTRREG
        EOR   #65
        STA   CTRREG
        ADDR  $FB,FN16
        LDY   #40
        STY   SCRCT1
        DEY
        LDA   #177
INL220  STA   1984,Y
        DEY
        BPL   INL220
        JMP   INL250
INL230  LDA   SCRCT2
        AND   #1
        BEQ   INL240
        LDA   CTRREG
        EOR   #66
        STA   CTRREG
        JMP   INL270
INL240  LDA   #60
        STA   SCRCT1
INL250  LDY   #247
        LDA   #0
INL260  STA   15912,Y
        DEY
        BNE   INL260
INL270  JMP   $EA31
SPRUPD  INC   V        ;X: SPRITE 0
        INC   V
        INC   V
        RTS
FN16    .BYTE
126,96,96,120,56,96,96,0,0,0,60,102,102,10
2,60,0,0,0,124
        .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
124,120,108,102,0,0,24,0,56,24,24,60,0,0,1
4,24,62,24,24,24
        .BYTE
0,0,56,24,24,24,24,60,0,0,0,60,102,126,96,
60,0,0,0,0,0,0
        .BYTE
0,0,0,99,115,127,107,99,99,99,0,0,0,60,6,6
2,102,62,0,0,0
        .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
```

```
PTTB    .BYTE
0,102,60,255,60,102,0,0,0,102,60,255,60,10
2,0,0,0,102
        .BYTE
60,255,60,102,0,0,0,0,0,0,0,0,0,0,124,102,
102,124,96
        .BYTE
96,96,0,0,0,102,102,102,102,62,0,0,56,24,2
4,24,24,60
        .BYTE
0,0,56,24,24,24,24,60,0,0,0,0,0,0,0,0,0,0,
24,126,24,24
        .BYTE
24,14,0,0,0,124,102,96,96,96,0,0,24,0,56,2
4,24,60,0,0
        .BYTE
0,62,102,102,62,6,124,0,0,62,102,102,62,6,
124,0,0,60
        .BYTE
102,126,96,60,0,0,0,124,102,96,96,96,0,0,0
,0,0,0,0,0
        .BYTE
0,0,24,126,24,24,24,14,0,0,0,60,102,102,10
2,60,0,0,0
        .BYTE
0,0,0,0,0,0,0,96,96,124,102,102,124,0,0,0,
60,102,126
        .BYTE
96,60,0,0,0,62,102,102,62,6,124,0,24,0,56,
24,24,60,0
        .BYTE
0,0,124,102,102,102,102,0,0,0,0,0,0,0,0,0,
0,102,60,255
        .BYTE
60,102,0,0,0,102,60,255,60,102,0,0,0,102,6
0,255,60,102,0,0,0
        .END
FIN     =*
INIDAT .BYTE 202,205,194,135,157,0,0,0
                    ;JMB 78.5
        .BYTE 215,200,199,135,153,0,0,0
                    ;WHG 76.5
        .BYTE
205,193,195,135,152,102,102,102
        ;MAC 76.2
        .BYTE
193,210,201,135,150,204,204,205
        ;ARI 75.4
        .BYTE
204,195,211,135,148,153,153,154
        ;LCS 74.3
        .BYTE
215,200,199,135,146,153,153,154
        ;WHG 73.3
        .BYTE
202,205,194,135,144,102,102,102
        ;JMB 72.2
        .BYTE
205,193,195,135,140,153,1 3,154
        ;MAC 70.3
        .BYTE
204,195,211,135,136,102,102,102
        ;LCS 68.2
        .BYTE 193,210,201,135,130,51,51,51
                    ;ARI 65.1
        .OPT  LIST
        .END
      *=57344
        .OPT  NOL
;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
CODE1   =16376
CODE2   =16377
COLOR   =16378
FLAGS   =16379
HOLDA   =16380
HOLDY   =16382
LENSTR =16383
        .MAC  ADDR
        LDA   #<?2
        STA   ?1
        LDA   #>?2
        STA   ?1+1
        .MND
        .MAC  DINC
        INC   ?1
        BNE   ?2
        INC   ?1+1
?2      .MND
        JMP   LETERS
        JMP   LETML
LETERS  LDA   ?1
        STA   $FD
        LDA   ?2
        STA   $FE
        LDY   #0
        LDA   ($FD),Y  ;LENGTH OF
        BNE   LET010   ; STRING
        RTS
LET010  STA   LENSTR
        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   $3272    ;WHICH SET?
        AND   #2
        BEQ   LET020   ;SET 1
        LDA            ;SET 2
        STA
LET020  LDA            ;SWITCH IN
        AND   #251     ; CHARACTER
        STA   1        ; ROM
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   LET690
LET090  CMP   #14      ;SET 2
        BNE   LET100
        LDA   FLAGS
        ORA   #4
        STA   FLAGS
        JMP   LET690
LET100  CMP   #133     ;F1-ENLARGE
        BNE   LET110   ;OR
        LDA   FLAGS    ;NORMAL
        EOR   #1
        STA   FLAGS
        JMP   LET690
LET110  CMP   #137     ;F2-CURSOR
        BNE   LET160
        JSR   BYTE2
        BCC   LET130
LET120  JMP   LET700
LET130  CMP   #40
        BCS   LET140
        STA   ?1       ;
LET140  JSR   ??2
        BCS
        CMP
        BCS       1
        ST-   .
        L     #
        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   LET690
LET160  CMP   #136     ;F3-SPCINSTR
        BEQ   LET170
        JMP   LET310
LET170  JSR   BYTE2
        BCC   LET180
        JMP   LET700
LET180  LDA   1        ;VIC CHIP
        ORA   #4       ;BACK IN FOR
        STA   1        ;A SECOND
        LDA   HOLDA
        AND   #1
        BEQ   LET190
        LDA   53265    ;BLANK SCRN
        AND   #239
        STA   53265    ;SCREEN
        JMP   LET200
LET190  LDA   53265    ;UNBLANK
        ORA   #16      ;SCREEN
        STA   53265
L   00  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   #8
        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
        LDA   HOLDA    ;CLEAR PART
        AND   #16      ; OF SCREEN?
        BEQ   LET250
        JSR   BYTE2
        BCS   LET300
        LDX   HOLDA    ;START LINE
        JSR   BYTE2    ;LAST LINE
        BCS   LET300
        TXA
        TAY
        LDA   #<8192
        STA   $FD
        LDA   #>8192
        STA   $FE
        CPY   #0
        BEQ   LET260
        CPY   #25
        BCS   LET290
LET250  LDA   $FD
        CLC
        ADC   #<320
        STA   $FD
        LDA   $FE
        ADC   #>320
        STA   $FE
        DEY
        BNE   LET250
```

```
LET260 LDY   #0
       LDA   #0
LET270 STA   ($FD),Y
       INY
       BNE   LET270
       INC   $FE
LET280 STA   ($FD),Y
       INY
       CPY   #64
       BCC   LET280
       LDA   $FD
       CLC
       ADC   #64
       STA   $FD
       LDA   $FE
       ADC   #0
       STA   $FE
       INX
       CPX   #25
       BCS   LET290
       CPX   HOLDA
       BCC   LET260
       BEQ   LET260
LET290 JMP   LET690
LET300 JMP   LET700
       JMP   LET690
LET310 CMP   #18      ;REVERSE
       BNE   LET320   ;ON
       LDA   FLAGS
       ORA   #2
       STA   FLAGS
       JMP   LET690
LET320 CMP   #146     ;REVERSE
       BNE   LET330   ;OFF
       LDA   FLAGS
       AND   #253
       STA   FLAGS
       JMP   LET690
LET330 CMP   #138     ;F4-ARROWS
       BNE   LET360
       JSR   BYTE2
       BCC   LET350
LET340 JMP   LET700
LET350 SEC
       SBC   #1
       CMP   #8
       BCS   LET340
       ASL   A
       ASL   A
       ASL   A
       CLC
       ADC   #<ARROWS
       STA   $FB
       LDA   #>ARROWS
       ADC   #0
       STA   $FC
       JMP   LET610
LET360 CMP   #135     ;F5-COLOR
       BNE   LET380
       JSR   BYTE2
       BCC   LET370
       JMP   LET700
LET370 LDA   #240
       STA   CODE1
       JSR   ADJSCR
       JMP   LET690
LET380 CMP   #139     ;F6-BORDER
       BNE   LET400   ;COLOR
       JSR   BYTE2
       BCC   LET390
       JMP   LET700
LET390 TAX
       LDA   1        ;SWITCH IN
       ORA   #4       ; VIC CHIP
       STA   1        ; A SECOND
       STX   $3280    ;BORDER
       LDA   1        ;SWITCH OUT
       AND   #251     ; VIC CHIP
       STA   1        ; AGAIN
       JMP   LET690
LET400 CMP   #147     ;CLEAR SCRN
       BNE   LET420
       LDA   #<8192
       STA   $FD
       LDA   #>8192
       STA   $FE
       LDY   #0
LET410 LDA   #0
       STA   ($FD),Y
       DINC  $FD
       LDA   $FD
       CMP   #<16192
       BNE   LET410
       LDA   $FE
       CMP   #>16192
       BNE   LET410
LET420 LDA   #>1024
       STA   210
       LDA   #0
       STA   209
       STA   211
       STA   214
       JMP   LET690
LET430 CMP   #29      ;RIGHT ARROW
       BNE   LET440
       JSR   TLINE
       JMP   LET690
LET440 CMP   #17      ;DOWN ARROW
       BNE   LET450
       JSR   LET710
SKIP   JMP   LET690
LET450 CMP   #145     ;UP ARROW
       BNE   LET460
       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
       DEC   $FF
       BEQ   BIG1
       JMP   LET690
LET460 CMP   #157     ;LEFT ARROW
       BNE   LET480
       LDA   FLAGS
       AND   #1
       STA   $FF
BIG2   LDA   211
       BNE   LET470
       LDA   214
       BEQ   SKIP
       DEC   214
       LDA   #40
       STA   211
       LDA   209
       SEC
       SBC   #40
       STA   209
       LDA   210
       SBC   #0
       STA   210
LET470 DEC   211
       DEC   $FF
       BEQ   BIG2
       JMP   LET690
LET480 CMP   #13
       BNE   LET500
LET490 LDA   #40
       STA   211
       JSR   TLINE
       JMP   LET690
LET500 CMP   #136     ;F7-CHAR COL
       BNE   LET520
       JSR   BYTE2
       BCC   LET510
       JMP   LET700
LET510 LDA   #15
       STA   CODE1
       LDA   HOLDA
       ASL   A
       ASL   A
       ASL   A
       ASL   A
       STA   HOLDA
       JSR   ADJSCR
       JMP   LET690
LET520 CMP   #140     ;F8-RETURN
       BEQ   LET490
       CMP   #19      ;HOME
       BNE   LET530
       JMP   LET420
LET530 LDX   #0
LET540 LDA   SPCIAL,X
       BEQ   LET560
       CMP   ($A3),Y
       BEQ   LET550
       INX
       INX
       JMP   LET540
LET550 INX
       LDA   SPCIAL,X
       STA   646
       JMP   LET690
LET560 LDA   ($A3),Y
       CMP   #96
       BCS   LET570
       AND   #191     ;X<96
       JMP   LET590
LET570 CMP   #161
       BCS   LET580
       AND   #223     ;96<X<161
       JMP   LET590
LET580 AND   #127     ;161<X<256
       ORA   #64
LET590 STA   $FB
       LDA   #0
       STA   $FC
       LDX   #3       ;*8
LET600 ASL   $FB
       ROL   $FC
       DEX
       BNE   LET600
       LDA   $FC
       CLC
       ADC   #$D0
       STA   $FC
       LDA   FLAGS    ;WHICH SET?
       AND   #4
       BEQ   LET610   ;SET 1
       LDA   $FC      ;SET 2
       CLC
       ADC   #8
       STA   $FC
LET610 LDA   209
       CLC
       ADC   211
       STA   $C3
       LDA   210
       ADC   #0
       STA   $C4
       LDX   #8
       LDA   646
       ASL   A
       ASL   A
       ASL   A
       ASL   A
       STA   COLOR
LET620 LDY   #0
       LDA   ($FB),Y  ;CHARS
       STA   HOLDA
       LDA   FLAGS
       BEQ   LET640
       AND   #1
       BEQ   LET630
       JSR   DOUBLE
LET630 LDA   FLAGS
       AND   #2
       BEQ   LET640
       LDA   HOLDA
       EOR   #255
       STA   HOLDA
       LDA   HOLDA+1
       EOR   #255
       STA   HOLDA+1
LET640 LDA   HOLDA
       STA   ($FD),Y  ;BIT MAP
       LDA   ($C3),Y  ;COLOR
       AND   #15
       ORA   COLOR
       STA   ($C3),Y
       LDA   FLAGS
       AND   #1
       BEQ   LET660
       LDA   214
       CMP   #24
       BEQ   LET660
       INY
       LDA   HOLDA
       STA   ($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   LET650
       INC   211
       DEC   $FE
       LDA   $FD
       SEC
       SBC   #64
       STA   $FD
       LDA   $FE
       SBC   #0
       STA   $FE
       JMP   LET680
LET650 CPX   #4
       BNE   LET670
       LDA   $FD
```

```
        CLC
        ADC   #56
        STA   $FD
        LDA   $FE
        ADC   #1
        STA   $FE
        JMP   LET620
LET660  DINC  $FD       ;BIT MAP
        DINC  $FB       ;CHARS
        DEX
        BEQ   LET680
LET670  JMP   LET620
LET680  JSR   TLINE
LET690  LDY   HOLDY
        INY
        DEC   LENSTR
        BEQ   LET700
        JMP   LET030
LET700  RTS
TLINE   INC   211
        LDA   211
        CMP   #40
        BCC   LET730
        LDA   #0
        STA   211
LET710  LDA   FLAGS
        AND   #1
        STA   $FF
LET720  LDA   214
        CMP   #24
        BEQ   LET730
        INC   214
        LDA   209
        CLC
        ADC   #40
        STA   209
        LDA   210
        ADC   #0
        STA   210
        DEC   $FF
        BEQ   LET720
LET730  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
0,4,6,255,255,6,4,0,0,192,192,96,104,56,12
0,24
        .BYTE
0,24,24,24,24,126,60,24,0,3,3,6,22,28,30,2
4
        .BYTE
0,32,96,255,255,96,32,0,0,24,30,28,22,6,3,
3
        .OPT  LIST
        .END
```

Sight Picture Program Documentation

```
10 POKE608,234:V=53248:CT=821:D1=823:D2=824:LP=825:HP=826:SC=3406:SD=822
20 TG=827:YP=828:BC=0:POKEV+21,0:POKEV+16,0:POKEV+39,1
22 DIMFR(7,3),FR(7,3),TS(7,3):FORJ=40TO46:POKEV+J,0:NEXTJ
30 A$="{F3}09{CLR}{F5}00{F6}00{F1}{GRN}{F2}1605MACS"
35 A$=A$+"{LBLU}{F2}1111Basics of{F2}1413Aiming{F3}00":GOSUB1060:GOSUB1000:IF(BR)THEN4999
40 A$="{CLR}{$WLC}{LBLU}{F1}{F2}0104Correct Application{F2}0207of {WHT}Sight Alignment{LBLU}"
42 A$=A$+"{F2}0110and {WHT}Aiming{LBLU} Improves{F2}0113Marksmenship Skills"
43 GOSUB1060:GOSUB1003:IF(BR)THEN4999
80 RESTORE:A$="{F3}09{CLR}{F5}00{F6}00{F1}{GRN}{F2}1605MACS"
82 A$=A$+"{LBLU}{F2}0309Sight Alignment{F2}1711and{F2}1413Aiming{F2}1315Program{F3}00":GOSUB1060
91 POKE2040,43:POKEV+39,2:POKEV+27,7:POKEV,173:POKEV+1,136
92 FORI=0TO7:FORJ=0TO3:READFS(I,J):NEXTJ:NEXTI:GOSUB1000:IF(BR)THEN4999
93 S1=36624:S2=11:GOSUB1160:A$="{BLK}":GOSUB1060:TOSUB1003
95 A$="{CLR}{$WLC}{BLK}{F5}13{F1}{F2}0503Sight Alignment{F2}1805is{F2}0610Placing Tip of"
96 A$=A$+"{F2}0612Front Sight Post{F2}0214in {RED}Exact Center {BLK}of"
97 A$=A$+"{F2}0116Rear Sight Aperture":GOSUB1060:BR=-1:GOSUB1003:IF(BR)THEN200
102 S1=39777:S2=11:GOSUB1160:A$="{$WLC}{BLK}{F1}{F2}0401Demonstration of{F2}0503Sight Alignment"
104 GOSUB1060:POKE53280,BC:L=0
105 GOSUB1170:POKEMS,2:POKECT,6:POKE2041,40:POKE2042,41:POKEV+40,0:POKEV+41,0
153 FORI=0TO7:POKEV+2,DX(I):POKEV+3,DY(I):POKEV+4,DX(I)+24:POKEV+5,DY(I)
155 POKE58D,DD(I):POKED1,161:POKED2,139:POKEV+21,1:C$="{LGRN}"
160 FORJ=0TO250:NEXTJ:POKEV+21,7:IFI>2ORL>0THENC$="{BLK}"
161 FORJ=0TO150:NEXTJ
162 A$=C$+"{$WLC}{F2}0824<Pull TRIGGER to Continue>":GOSUB1060:POKELP,255
165 FORJ=0TO400:IFPEEK(TG)<>0THEN170
166 IFJ=>200THENPOKEV+21,6
168 NEXTJ:POKEV+21,0:NEXTI:L=L+1:GOTO105
170 IFPEEK(TG)=127THENPOKEV+21,0:GOTO208
175 IFPEEK(TG)=1THENPOKEV+21,0:A$="{F2}0824                    "
177 POKE53281,13:A$="{CLR}{$WLC}{BLK}{F1}{F2}0401Test Your Skill:{F2}1204Identify"
178 A$=A$+"{F2}0507Sight Alignment{F2}1113STANDARD:"
179 A$=A$+"{F2}03163 Correct Answers{F2}1219In a Row":GOSUB1060:GOSUB1000
180 EX(C)=161:EY(C)=139:POKETG,0:LV=1:PX=161:PY=139:POKEV+21,0
181 FORI=0TO2:S1=39777:S2=11:GOSUB1160:POKE53280,BC:POKEMS,2
182 POKECT,6:POKEV+2,TX(I):POKEV+3,TY(I):POKEV+4,TX(I)+24:POKEV+5,TY(I)
184 POKE58D,TD(I):POKED1,EX(I):POKED2,EY(I):GOSUB1060:POKEV+21,6:POKELP,254
186 IFPEEK(LP)<>0THEN186
188 A$="{$WLC}{BLK}{F1}{F2}0402Are These Sights{F2}1204Aligned?":GOSUB1060:GOSUB1119
189 IF(BR)THENPOKEV+21,0:GOTO208
200 IFAN<>C(I)THENGOSUB1090:POKEV+21,7:GOSUB1250:GOSUB1170:GOTO180
202 GOSUB1070:NEXTI:POKEV+21,0:POKE53280,0
204 A$="{CLR}{$WLC}{BLK}{F1}{F2}1006Well Done!{F2}0711Standards met"
206 GOSUB1060:GOSUB1003
208 FORI=0TO7:FORJ=0TO3:READFS(I,J):NEXTJ:NEXTI
209 A$="{CLR}{$WLC}{BLK}{F5}13{F1}{F2}0703Proper Aiming{F2}1805is{F2}0510Aligning Sights{F2}0912with Target"
211 GOSUB1060:BR=-1:GOSUB1003:IF(BR)THEN360
212 A$="{CLR}{$WLC}{BLK}{F1}{F2}0401Demonstration of{F2}0703Proper Aiming"
213 A$=A$+"{F2}0418250 meter Target"
214 GOSUB1060:POKE2047,42:POKEV+46,12:POKEV+14,181:POKEV+15,139:L=0
216 GOSUB1170:POKEMS,6:POKECT,6:POKE2041,34:POKE2042,35:POKE2043,36:POKE2044,37
217 POKE2045,38:POKE2046,39:POKETG,0
218 FORI=0TO7:POKEV+2,DX(I):POKEV+3,DY(I):POKEV+4,DX(I)+24:POKEV+5,DY(I)
219 POKEV+6,DX(I)+48:POKEV+7,DY(I):POKEV+8,DX(I):POKEV+9,DY(I)+21
220 POKEV+10,DX(I)+24:POKEV+11,DY(I)+21:POKEV+12,DX(I)+48:POKEV+13,DY(I)+21
221 POKE58D,DD(I):POKED1,154:POKED2,126:POKEV+21,129:C$="{LGRN}":IFI>2ORL>0THENC$="{BLK}"
222 FORJ=0TO200:NEXTJ:POKEV+21,255:FORJ=0TO150:NEXTJ
223 A$=C$+"{$WLC}{F2}0824<Pull TRIGGER to Continue>":GOSUB1060:POKELP,255
225 FORJ=0TO450:IFPEEK(TG)<>0THEN229
226 IFJ=>250THENPOKEV+21,254
228 NEXTJ:POKEV+21,0:NEXTI:L=L+1:GOTO216
229 IFPEEK(TG)=127THENPOKEV+21,0:GOTO360
230 IFPEEK(TG)=1THENPOKEV+21,0:A$="{F2}0824                    "
232 POKE53281,13:A$="{CLR}{$WLC}{BLK}{F1}{F2}0401Test Your Skill:{F2}0404Identify Correct"
240 A$=A$+"{F2}1407Aiming{F2}1113STANDARD:"
250 A$=A$+"{F2}03163 Correct Answers{F2}1219In a Row":GOSUB1060:GOSUB1000
260 EX(C)=156:EY(C)=126:POKETG,0:LV=2:PX=156:PY=126
270 FORI=0TO2:A$="{CLR}":GOSUB1060:POKE53280,BC:POKEMS,6:POKEV+21,128:POKECT,6
280 POKEV+2,TX(I):POKEV+3,TY(I):POKEV+4,TX(I)+24:POKEV+5,TY(I)
281 POKEV+6,TX(I)+48:POKEV+7,TY(I):POKEV+8,TX(I):POKEV+9,TY(I)+21
282 POKEV+10,TX(I)+24:POKEV+11,TY(I)+21:POKEV+12,TX(I)+48:POKEV+13,TY(I)+21
286 POKE58D,TD(I):POKED1,EX(I):POKED2,EY(I):GOSUB1060:POKEV+21,254:POKELP,254
288 IFPEEK(LP)<>0THEN288
310 A$="{CLR}{$WLC}{BLK}{F1}{F2}0902Is This Aim{F2}1204Correct?":GOSUB1060:GOSUB1110
320 IF(BR)THENPOKEV+21,0:GOTO360
330 IFAN<>C(I)THENGOSUB1090:POKEV+21,255:GOSUB1250:POKETG,0:GOSUB1170:GOTO260
340 GOSUB1070:NEXTI:POKE53280,0:POKEV+21,0
350 A$="{CLR}{$WLC}{BLK}{F1}{F2}0706Standards Met{F2}0415Now Try These...":GOSUB1060:GOSUB1003
360 RESTORE:FORI=0TO7:FORJ=0TO3:READFS(I,J):NEXTJ:NEXTI
365 GOSUB1170:POKE53281,13:A$="{CLR}{$WLC}{BLK}{F1}{F2}0501Test Your Skill{F2}0604in Identifying"
370 A$=A$+"{F2}0107Sight Alignment and{F2}0710Proper Aim{F2}1115STANDARD:"
380 A$=A$+"{F2}03183 Correct Answers{F2}1221In a Row"
382 IF(BR)THENA$="{CLR}":GOSUB1060:GOTO400
383 POKE2041,40:POKE2042,41:POKE2043,42:POKEV+42,11
385 FORJ=0TO2:X(J)=EX(J)+(INT(RND(1)*7)+17):Y1(J)=EY(J)+INT(RND(1)*3)
390 NEXTJ:EX(C)=161:EY(C)=139:X1(C)=161:Y1(C)=139:POKETG,0:LV=3
391 C1=INT(RND(1)*3):IFC=0THENC1=C+C1
392 IFC=3THENC1=C-C1
393 EX(C1)=161:EY(C1)=139
395 P1=161:P2=139:P3=161:P4=139
400 FORI=0TO2:S1=39777:S2=11:GOSUB1160:POKE53280,BC:POKEMS,2:POKECT,6
410 POKEV+2,TX(I):POKEV+3,TY(I):POKEV+4,TX(I)+24:POKEV+5,TY(I)
412 POKE58D,TD(I):POKED1,EX(I):POKED2,EY(I):POKEV+21,6:POKELP,254
415 IFPEEK(LP)<>0THEN415
420 POKEV+6,X1(I):POKEV+7,Y1(I):POKEV+21,14
440 A$="{$WLC}{BLK}{F1}{F2}0902Is This Aim{F2}1204Correct?":GOSUB1060:GOSUB1110
```

```
475 IF(BR)THENPOKEV+21,0:GOTO300
480 IFAN<>C(I)THENGOSUB1090:GOSUB1330:GOSUB1170:GOTO385
490 GOSUB1070:POKEV+21,0:NEXTI:POKE53280,0
500 A$="{CLR}{SWLC}{BLK}{F1}{F2}1006Well Done!{F2}0510Sight Alignment{F2}1012and Aiming{F2}0314"
510 A$=A$+"Program Completed":GOSUB1060:BR=-1:GOSUB1000
530 GOTO4999
1000 A$="{SWLC}{F2}0124        <Pull TRIGGER to Begin>      ":GOSUB1060:BR=-1:GOTO1010
1003 A$="{SWLC}{F2}0024        <Pull TRIGGER to Continue>      ":GOSUB1060:BR=-1
1010 IFPEEK(56321)=247THENBR=0:GOTO1030
1020 IFPEEK(56321)<>127THEN1010
1030 IFPEEK(56321)=247THEN1030
1040 A$="{SWLC}{F2}0124                                          ":GOSUB1060:RETURN
1060 A$=A$:SYS49182:RETURN
1070 A$="{SWLC}{RED}{F1}{F2}0515You are Correct{BLK}":GOSUB1060:GOSUB1003
1080 A$="{F1}{F2}0515              ":GOSUB1060:RETURN
1090 POKE53280,2:A$="{SWLC}{RED}{F1}{F2}0715You are Wrong":GOSUB1060:RETURN
1110 BR=-1:A$="{SWLC}{BLK}{F2}0617Pull {RED}TRIGGER{BLK} to {RED}SELECT{BLK} Answer":GOSUB1060
1112 A$="{SWLC}{BLK}{F1}{F2}0412YES{F2}3412  {GRY2}{C/LF}{C/LF}{F1}NO":GOSUB1060
1115 FORJ=0TO300:IFPEEK(TG)<>0THENAN=0:BR=0:GOTO1140
1116 NEXTJ
1120 A$="{F2}0412{F1}   {C/LF}{C/LF}{C/LF}{F1}{SWLC}{GRY2}YES{BLK}{F1}{F2}3412NO":GOSUB1060
1125 FORJ=0TO300:IFPEEK(TG)<>0THENAN=1:BR=0:GOTO1140
1126 NEXTJ
1130 IFPEEK(TG)=0THEN1110
1140 IFPEEK(TG)=127THENBR=-1
1150 A$="{SWLC}{BLK}{F2}0617                                    ":GOSUB1060:RETURN
1160 HB=INT(Z1/256):LB=Z1-(HB*256):POKE253,LB:POKE254,HB:POKE251,Z2
1165 SYS52C:RETURN
1170 FORI=0TO7:P(I)=0:NEXTI:FORI=0TO7
1180 P=INT(RND(1)*8):IFP(P)=1THEN1180
1190 P(P)=1:DX(I)=FS(P,0):DY(I)=FS(P,1):E1(I)=FS(P,2):E2(I)=FS(P,3)
1200 DD(I)=P:NEXTI:FORI=0TO2
1210 P=INT(RND(1)*8):IFP(P)=0THEN1210
1220 P(P)=0:TX(I)=DX(P):TY(I)=DY(P):EX(I)=E1(P):EY(I)=E2(P):TD(I)=DD(P):C(I)=1
1230 NEXTI:C=INT(RND(1)*3):C(C)=0:RETURN
1250 H$="":V$="":M$="{BLK}":L$="":B$="":SP$="{RED}{F2}":OX=PX-EX(I):OY=PY-EY(I)
1251 IFLV=1THENSA$="{F2}0521Sight Alignment":F$="{SWLC}{BLK}{F1}{F2}0617Front Sight is"
1252 IFLV=2THENSA$="{F2}1421Aiming":F$="{SWLC}{BLK}{F1}{F2}1417Aim is"
1253 A$=F$+SP$
1260 IFOX>0THENH$="Left"
1270 IFOX<0THENH$="Right"
1280 IFOY>0THENV$="High"
1290 IFOY<0THENV$="Low"
1292 IFH$=""ANDV$=""THENA$="{BLK}{SWLC}{F1}{F2}1317This is{F2}1319{RED}CORRECT{BLK}"+SA$:GOTO1310
1295 IFV$=""ANDH$<>""THENH$="{BLK}{F2}1121of Center"
1296 M1=LEN(H$):M2=LEN(V$):M3=M1+M2+1:L$=" ":IFM1=0ORM2=0THENM3=M3-1:L$=""
1300 M3=20-M3:B$=STR$(M3):B$=RIGHT$(B$,2)+"19"
1310 A$=A$+B$+V$+L$+H$+M$:GOSUB1060:GOSUB1003:IF(BR)THEN4999
1320 RETURN
1330 SA$="{SWLC}{BLK}{F1}{F2}0215Sight Alignment is":F1$="{F2}0619Front Sight is":G=0:G=0
1332 C1$="and":C2$="but"
1333 F2$="{F2}0219Front Sight is not":OK$="{F2}0917Correct ":OF$="{F2}0717Incorrect "
1335 POKEV+21,15:FORJ=0TO750:NEXTJ
1340 A1=P1-EX(I):A2=P2-EY(I):S1=EX(I)-X1(I):S2=EY(I)-Y1(I)
1350 IF(A1=0)AND(A2=0)THENSA$=SA$+OK$:G=1
1355 IF(A1<>0)OR(A2<>0)THENSA$=SA$+OF$:G=1
1360 IF(S1=-20)AND(S2=0)THENSP$=P1$:C$=C2$:IFG=1THENC$=C1$
1365 IF(S1<>-20)OR(S2<>0)THENSP$=P2$:C$=C2$:IFG=1THENC$=C1$
1370 A$=SA$+C$+SP$+"{F2}0221Centered on Target":GOSUB1060
1375 GOSUB1003:IF(BR)THEN4999
1380 A$="{CLR}":GOSUB1060:POKEV+21,0:RETURN
4999 SYS3403:POKEV+27,0:POKE49161,3:SYS49179
5000 DATA141,139,159,139,141,119,159,137,161,119,161,141,161,119,163,141
5010 DATA181,139,163,139,181,159,163,141,161,159,161,137,141,159,163,137
6000 DATA134,126,156,126,134,106,152,124,154,106,154,127,174,106,153,127
6010 DATA174,126,152,126,174,146,156,128,154,146,154,124,134,146,156,124
```

```
        *=3400
; BA.3400.TXT FOR SIGHT ALIGNMENT
; AND PLACEMENT PROGRAM
        JMP   ENABLE
        JMP   DSABLE
        JMP   SCENE
        .OPT  NOL
        .LIB  MACROS-VARS
        .OPT  NOL
;820-1023 IS UNUSED IF NO CASSETTE
CTR     =820
CTR1    =821
DIR     =822
ENDX    =823
ENDY    =824
LPON    =825
TEMP    =826
TR      =827
YMANS   =828
CHOICE  =830
STOP    =831
HOLD1   =833
HOLD2   =834
HERT    =835
HOLDA   =836
HOLDX   =837
HVHRD   =838
HVHRB   =839
HOLD    =841
CUR     =843
OFFX    =844
OFFY    =846
SHOTS   =848
SPNUM   =849
CURHOL  =850
H1IRQ   =851
H2IRQ   =852
XVAL    =853   ;X BULLET STRIKE
YVAL    =855   ;Y BULLET STRIKE
SORTAF  =857   ;# RDGS TO SORT AFTER
STSIZE  =858   ;# TO PUT IN SRTBUF
CURSRT  =859
SSNUM   =860
CODE1   =875
CODE2   =876
COLOR   =877
HOLDY   =878
FLAGS   =879
LENSTR  =880
TRAJ    =881
CNT     =882
CURCNT  =883
INCX    =884
OVIRQ   =886
IRQRGT  =887
NUMSP   =888
DELAY   =889
RPLIRQ  =891
EXTRAS  =893
H3IRQ   =895
H4IRQ   =896
H5IRQ   =897
H6IRQ   =898
H7IRQ   =899
RIGHT   =900
OVFLO   =901
TIME    =902
XHVAFT  =904
KEEPON  =906
SUMX    =907
NSIZE   =909
DSPLAY  =910
PSSNUM  =911
MAXVAL  =919
SPOSTX  =920
SPOSTY  =922
DEV     =923
TRACK   =925
PRDIAM  =926
;NEXT ONE AT 940
BUF1    =2049
BUF2    =2099
TIMES   =2149
SRTBUF  =2150 ;16 BYTES
SEROSF  =2166
; N2XT ONE =2176
DEVBUF  =51150
RDGSB1  =51200
RDGSB2  =51712
XBUF    =52224
YBUF    =52736
V       =53248
SID     =54272
SCRMEM  =31960
SCREND  =32960
CHKIN   =$FFC6
CHKOUT  =$FFC9
CHKIN   =$FFCT
CLOSE   =$FFC3
CLRCHN  =$FFCC
CHROUT  =$FFD2
FLOAT   =$B391
GETIN   =$FFE4
LOAD    =$FFD5
OPEN    =$FFC0
SETLFS  =$FFBA
SETNAM  =$FFBD
        .MAC  DINC    ;DOUBLE
        INC   ?1      ;PRECISION
        BNE   ?2      ;INCREMENT
        INC   ?1+1
?2      .MND
        .MAC  DDEC    ;DOUBLE
        LDA   ?1      ;PRECISION
        BEQ   ?2      ;DECREMENT
        DEC   ?1+1
?2      DEC   ?1
        .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  PRINT
        LDA   #<?1
        LDY   #>?1
        JSR   $AB1E   ;PRINT
        .MND
        .MAC  PLOT
        LDY   #?1
        LDX   #?2
        CLC
        JSR   $FFF0
        .MND
        .MAC  DISK    ;DISK
        LDA   #6      ;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    ;IS ?3-?2
        JSR   SETNAM
        LDA   #0
        .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  DEPL
        LDA   #?2-?1
        STA   LENSTR
        LDA   #<?1
        STA   $A3
        LDA   #>?1
        STA   $A4
        JSR   LETML
        .MND
        .END
        .OPT  NOL
;
;PROCEDURE ENABLE
;PREPARES IRQ TO TAKE READINGS
;B:NONE
;C:SYS ENABLE
;A:NONE
ENABLE SEI
        LDA   #0
        STA   $6334
        LDA   $3265
        AND   #127
        STA   $3265
        LDA   #250
        STA   $3266
        LDA   #5
        STA   $3274
        LDA   $3273
        STA   $3273
        ADDR  $314,START
        LDA   #0
        STA   LPON
        CLI
        RTS
;
;PROCEDURE DSABLE
;RETURNS IRQ VECTOR TO NORMAL
;B:NONE
;C:SYS DSABLE
;A:NONE
DSABLE SEI
        LDA   #1
        STA   $6334
        LDA   #240
        STA   $3274
        LDA   $3273
        STA   $3273
        ADDR  $314,$EA31
        CLI
        RTS
;
;PROCEDURE START (IRQ)
;IRQ ROUTINE TO TAKE PEN READINGS
;B,C,A:SEE AFTER, BEFORE
START   LDA   #4
        BIT   $3273
        BEQ   CONTIN
        LDA   $3273
        STA   $3273
        JMP   $FEBC
CONTIN  LDA   $3265
        AND   #127
        STA   $3265
        LDA   #250
        STA   $3266
        LDA   $3273
        STA   $3273
        LDA   LPON
        BPL   GETLP
        BEQ   CONT1
        JMP   MOVE
CONT1   JMP   $EA31
;
;PROCEDURE GETLP (IRQ)
;TAKES PEN READINGS FOR YES/NO
;B:POKE LP,0
;C:POKE LP,N WHERE 0<N<128
;A:(X/2) READINGS STORED IN XPIX
;  Y READINGS STORED AT YPIX.
GETLP   JSR   TRGGR
        LDA   TR
        BEQ   OUTLP1
        LDA   #0
        STA   LPON
OUTLP1  JMP   $EA31  ; RETURN
;
;PROCEDURE TRGGR
;
TRGGR   LDA   $6321
        CMP   #247
        BNE   TRG01
        LDA   #0
        STA   LPON
        LDA   #1
        STA   TR
        JMP   TRG03
TRG01   CMP   #127
        BNE   TRG02
        LDA   #0
        STA   LPON
        LDA   #127
        STA   TR
        JMP   TRG03
TRG02   LDA   #0
        STA   TR
TRG03   RTS
;PROCEDURE MOVE (IRQ)
;MOVES SIGHTS INTO ALIGNMENT
;B:POKE CTR,#SPRITES:POKE LP,0
; :POKE STARTING,ENDING X,Y
; :POKE DELAY INTO CTR1
; :POKE DIRECTION INTO DIR (0-7)
;C:POKE LP,N
;       WHERE 127<N<255 FOR TEST
;             N=255 FOR DEMO
;A:POKE LP,0:RETURNS TO PROGRAM
MOVE    LDA   LPON    ;TEST OR DEMO?
        CMP   #255    ; DEMO
        BNE   MOVE0   ; TEST
```

```
        JSR   TRGCR  : CHECK TRGGR
MOVE0   LDA   CTR1   ;DELAY
        CMP   #1     ; =0?
        BCS   MOVE1
        JMP   OUTMV
MOVE1   LDA   CTR
        ASL   A
        TAX
        INX
        STX   TEMP
SPMOV   LDA   #7     ;RESTORE CTR1
        STA   CTR1   ; DELAY
        LDA   DIR
        CMP   #1
        BEQ   SPMV01
        BCC   SPMV00
        CMP   #3
        BEQ   SPMV03
        BCC   SPMV02
        CMP   #5
        BEQ   SPMV05
        BCC   SPMV04
        CMP   #7
        BEQ   SPMV07
        BCC   SPMV06
SPMV00  JSR   ADDX
        JSR   CHKX
        JMP   OUTMV
SPMV01  JSR   ADDX
        JSR   ADDY
        JSR   CHKX
        JMP   OUTMV
SPMV02  JSR   ADDY
        JSR   CHKY
        JMP   OUTMV
SPMV03  JSR   SUBX
        JSR   ADDY
        JSR   CHKX
        JMP   OUTMV
SPMV04  JSR   SUBX
        JSR   CHKX
        JMP   OUTMV
SPMV05  JSR   SUBX
        JSR   SUBY
        JSR   CHKX
        JMP   OUTMV
SPMV06  JSR   SUBY
        JSR   CHKY
        JMP   OUTMV
SPMV07  JSR   ADDX
        JSR   SUBY
        JSR   CHKX
        JMP   OUTMV
ADDX    LDX   TEMP
ADDX1   DEX
        LDA   V,X
        CLC
        ADC   #1
        STA   V,X
        DEX
        CPX   #2
        BCS   ADDX1
        RTS
SUBX    LDX   TEMP
SUBX1   DEX
        LDA   V,X
        SEC
        SBC   #1
        STA   V,X
        DEX
        CPX   #2
        BCS   SUBX1
        RTS
CHKX    LDA   V+2
        CMP   ENDX
        BNE   CXOUT
        LDA   #0
        STA   LPON
CXOUT   RTS
ADDY    LDX   TEMP
ADDY1   LDA   V,X
        CLC
        ADC   #1
        STA   V,X
        DEX
        DEX
        CPX   #2
        BCS   ADDY1
        RTS
SUBY    LDX   TEMP
SUBY1   LDA   V,X
        SEC
        SBC   #1
        STA   V,X
        DEX
        DEX
        CPX   #2
        BCS   SUBY1
        RTS

CHKY    LDA   V+3
        CMP   ENDY
        BNE   CYOUT
        LDA   #0
        STA   LPON
CYOUT   RTS
OUTMV   DEC   CTR1
        JMP   $8A31
SCENE   LDA   $FB       ;CHIP 0/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   #32       ;RAM
        STA   $DFFF
        LDA   #59
        STA   53265
        RTS
        .OPT  LIST
        .END
```

Grouping Program Documentation

```
10 DIM XY%(23,1):V=53248:SB=3409:SM=3412:CI=3415
20 DEF FNH(X)=INT(X/256):DEF FNL(X)=X-INT(X/256)*256:DEF FNR(X)=INT(RND(0)*X)
30 FOR I=0 TO 23:READ XY%(I,0),XY%(I,1):NEXT I
40 CO$="{F1}{GRN}:F2}1320CORRECT":POKE 53281,0
45 C1$="{BLK}{F3}162021{F1}{F2}0520YES{F1}{GRY3}{F2}3020no{LBLU}":C2$="{GRY3}{F3}162021{F2}0520yes{F1}{BLK}{F2}3020NO{LBLU}"
50 POKE 2040,34:POKE 2041,34:POKE 2042,34:POKE V+29,7:POKE V+23,7
60 POKE V+39,5:POKE V+40,5:POKE V+41,5:POKE V+28,0:POKE V+16,0:POKE V+21,0
70 A$="{F6}00{F3}09{CLR}{F5}01{F7}00{F1}{F2}1205{GRN}GROUPING{LBLU}{F2}0210Good shot grouping"
80 A$=A$+"{F2}0513is important to{F2}0616marksmanship":GOSUB 1000:GOSUB 1010
90 S=0:GOSUB 1050:A$="{F3}170005{F1}{F2}0300Tight shot group:"
100 A$=A$+"{F2}1102Shots hit{F2}0304 close together. {F1}{F3}08":GOSUB 1000
110 POKE V,100:POKE V+1,126:POKE V+2,112:POKE V+3,136:POKE V+4,126:POKE V+5,118
120 POKE 823,7:SYS SB
130 A$="{F1}{F2}1321 TIGHT ":GOSUB 1000:GOSUB 1010
140 S=0:GOSUB 1050:A$="{F3}170005{F1}{F2}0301Large shot group:"
150 A$=A$+"{F2}0103Shots hit far apart{F3}08{F1}":GOSUB 1000
160 POKE V,236:POKE V+1,119:POKE V+2,178:POKE V+3,153:POKE V+4,96:POKE V+5,123
170 POKE 823,7:SYS SB
180 A$="{F1}{F2}1321 LARGE ":GOSUB 1000:GOSUB 1010
190 S=0:GOSUB 1050:A$="{F3}170005{F1}{F2}0401Tight shot group"
200 A$=A$+"{F2}0103fits in 4 cm circle{F1}{F3}08":GOSUB 1000
210 POKE V,96:POKE V+1,118:POKE V+2,138:POKE V+3,126:POKE V+4,120:POKE V+5,110
220 POKE 823,7:SYS SB
230 POKE 864,88:POKE 865,0:POKE 866,73:POKE 863,28:SYS CI:FOR D=1 TO 1000:NEXT
240 POKE 823,7:POKE 843,112:POKE 845,122:POKE 847,104:POKE 849,150:SYS SM
250 GOSUB 1010:S=0:GOSUB 1050:A$="{F3}170005{F1}{F2}0500If only 2 shots"
260 A$=A$+"{F2}0302can be seen, fire{F2}0204another shot group{F3}08":GOSUB 1000
270 POKE V,0:POKE V+1,0:POKE V+2,145:POKE V+3,130:POKE V+4,111:POKE V+5,126
280 POKE 823,7:SYS SB:GOSUB 1010
290 I2=0:NC=0
300 IF NC=3 THEN 380
310 I1=FNR(8):IF (NC=2) AND ((I2 AND 15)=0) THEN I1=FNR(4)
320 IF (NC=2) AND ((I2 AND 240)=0) THEN I1=FNR(4)*4
330 IF (I2 AND (2^I1))>0 THEN I1=I1+1:I1=I1+(I1>7)*I1:GOTO 330
340 I2=(I2 OR (2^I1)):GOSUB 1160:IF I THEN NC=NC+1
350 IF (I2 AND 240)=240 THEN I2=I2 AND 15
360 IF (I2 AND 15)=15 THEN I2=I2 AND 240
370 GOTO 300
380 A$="{F3}09{CLR}{F1}{F2}1011Very good!{F3}0615Standards met.{F3}08":GOSUB 1000:GOSUB 1010
390 A$="{F6}00{F3}09{CLR}{F5}01{F7}00{F1}{GRN}{F2}0205Would you like to:"
400 C1$="{F1}{BLK}{F2}0615^{F2}3115 {F2}0008Begin{F8}BRM{F8}program{GRY3}{F2}2508Restart{F2}2510this{F2}2512program"
410 C2$="{F1}{BLK}{F2}0615 {F2}3115^{F2}2508Restart{F2}2510this{F2}2512program!GRY3}{F2}0008Begin{F8}BRM{F8}program"
420 POKE V+21,0:GOSUB 1000:GOSUB 1060:IF I=0 THEN POKE 49161,2:SYS 49179
430 POKE 53272,23:POKE 53265,27:PRINT"{CLR}":END:SYS 49176
1000 A$=A$:SYS 49102:RETURN
1010 A$="{LBLU}{F3}162424{F2}0724<Pull trigger to continue>":GOSUB 1000:BR=0
1020 IF PEEK(56321)=127 THEN BR=-1:GOTO 1030
1025 IF PEEK(56321)<>247 THEN 1020
1030 IF PEEK(56321)<>255 THEN 1030
1035 IF BR THEN 390
1040 POKE V+21,0:RETURN
1050 POKE 822,S:SYS 3406:POKE 53272,31:RETURN
1060 A$="{LBLU}{F3}162424{F2}0524<Pull trigger to select answer>":GOSUB 1000
1070 A$=C1$:GOSUB 1000:I=0:BR=0
1080 IF PEEK(56321)=247 THEN I=-1:GOTO 1140
1085 IF PEEK(56321)=127 THEN BR=-1:GOTO 1140
1090 I=I+1:IF I<50 THEN 1080
1100 A$=C2$:GOSUB 1000:I=0
1110 IF PEEK(56321)=247 THEN I=0:GOTO 1140
1115 IF PEEK(56321)=127 THEN BR=-1:GOTO 1140
1120 I=I+1:IF I<50 THEN 1110
1130 GOTO 1070
1140 IF PEEK(56321)<>255 THEN 1140
1145 IF BR THEN 390
1150 RETURN
1160 S=0:GOSUB 1050:I3=FNR(2)*99
1170 I4=0:X=0:Y=0:FOR J=0 TO 2:I=FNR(3)+5
1180 IF (I4 AND (2^I))>0 THEN I=I+1:I=I+(I>7)*(I-5):GOTO 1180
1190 I4=(I4 OR (2^I)):POKE V+J*2,XY%(I1*3+I-5,0)+I3:X=X+XY%(I1*3+I-5,0)
1200 POKE V+J*2+1,XY%(I1*3+I-5,1):Y=Y+XY%(I1*3+I-5,1):NEXT J
1210 X=INT(X/3)+I3:Y=INT(Y/3)
1220 A$="{F3}170005{F1}{F2}0501Is this a tight{F2}1103shot group?{F1}{F3}08":GOSUB 1000
1230 POKE 823,7:SYS SB:GOSUB 1060
1240 IF ((X1>3) AND (I=0)) OR ((X1<4) AND (I=-1)) THEN 1260
1250 I=-1:A$=CO$:GOSUB 1000:GOTO 1310
1260 I=0:NC=0:S=0:GOSUB 1050:A$="{F1}{F2}1302{RED! WRONG {LBLU}{F3}171621"
1270 IF X1>3 THEN A$=A$+"{F2}0618Group will fit{F2}1120in circle{F1}":GOTO 1290
1280 A$=A$+"{F2}0618Group does not{F2}0720fit in circle{F1}"
1290 POKE V+21,7:POKE 864,X-20:POKE 865,0:POKE 866,Y-46
1300 POKE 863,28:GOSUB 1000:SYS CI:POKE 53280,2
1310 GOSUB 1010:POKE 53280,0:RETURN
8000 DATA 112,146,152,150,132,106,117,112,105,164,132,105
8010 DATA 93,146,146,195,142,100,98,193,152,182,128,129
8020 DATA 129,105,106,99,119,84,119,126,147,124,151,116
8030 DATA 120,124,136,129,131,125,120,162,136,167,110,184
```

```
        *=3400
;ZEROTEMP.TXT
        .OPT   NOL
        JMP    ENABLE
        JMP    DSABLE
        JMP    SCENE
        JMP    SHBANG
        JMP    MOVESP
        JMP    CIRCLE
STOP    =820
PICTUR  =822
CODE1   =823
CODE2   =825
CODE3   =827
CODE4   =829
HOLD1   =831
HOLD2   =833
HOLD3   =835
HOLD4   =837
XVAL    =839
YVAL    =841
X1      =843
Y1      =845
X2      =847
Y2      =849
DX      =851
DY      =853
STEPN   =855
STEPD   =857
CSTEPN  =859
COUNT   =861
ANGLE   =862
RADIUS  =863
CENX    =864
CENY    =866
HLD1FP  =868
HLD2FP  =873
;NEXT ONE AT 878
LENSTR  =16383
LETNL   =49185
V       =53248
JOYSTK  =56320
        .MAC   DINC
        INC    ?1
        BNE    ?2
        INC    ?1+1
?2      .END
        .MAC   ADDR
        LDA    #<?2
        STA    ?1
        LDA    #>?2
        STA    ?1+1
        .END
ENABLE  CLI
        RTS
DSABLE  RTS
;SCENE IS CALLED TO DISPLAY ONE
;OF THE PICTURES ON THE EPROM
;PUT THE PICTURE NUMBER IN
;THE LOCATION "PICTUR" AND CALL
;THIS ROUTINE, OR USE THE MACRO
;SCRN PICNUM.
SCENE   LDA    #43
        STA    V+17
        LDA    #29
        STA    V+24
        ADDR   $A3,1024
SLP10   LDY    #0
SLP20   LDA    #177
        STA    ($A3),Y
        DINC   $A3
        LDA    $A3
        CMP    #<2024
        BNE    SLP20
        LDA    $A4
        CMP    #>2024
        BNE    SLP20
        LDX    PICTUR
        BNE    SLP30
        LDA    #10
        STA    $DFFF
        ADDR   $FD,$8000
        ADDR   $A3,8192
        ADDR   STOP,16192
        JSR    MOVEIT
        JMP    SLP70
SLP30   ADDR   $A3,8192
        LDA    #0
        TAY
SLP40   STA    ($A3),Y
        DINC   $A3
        LDX    $A3
        CPX    #<16192
        BCC    SLP40
        LDX    $A4
        CPX    #>16192
        BCC    SLP40
        LDA    #11
        STA    $DFFF
        LDX    PICTUR
        DEX
        BNE    SLP50
        ADDR   $FD,32768 ;TOP 1/3
        ADDR   $A3,10752
        ADDR   STOP,13631
        JSR    MOVEIT
        JMP    SLP70
SLP50   DEX
        BNE    SLP60
        ADDR   $FD,35648 ;MID 1/3
        ADDR   $A3,10752
        ADDR   STOP,13631
        JSR    MOVEIT
        JMP    SLP70
SLP60   ADDR   $FD,38848 ;BOT 1/3
        ADDR   $A3,11392
        ADDR   STOP,12991
        JSR    MOVEIT
SLP70   LDA    #32
        STA    $DFFF
        RTS
MOVEIT  LDY    #0
MOVE1   LDA    ($FD),Y
        STA    ($A3),Y
        DINC   $FD
        DINC   $A3
        LDA    $A3
        CMP    STOP
        BNE    MOVEL1
        LDA    $A4
        CMP    STOP+1
        BNE    MOVEL1
        RTS
SHBANG  LDA    #128
        STA    CODE2
SHB010  LDA    CODE1
        AND    CODE2
        BEQ    SHB020
        LDY    #255
        JSR    WAIT
        LDY    #255
        JSR    WAIT
        LDY    #255
        JSR    WAIT
        LDA    V+21
        ORA    CODE2
        STA    V+21
        JSR    BANG
SHB020  LSR    CODE2
        BNE    SHB010
        RTS
MOVESP  LDA    #0
        STA    CSTEPN
        LDA    X2
        SEC
        SBC    X1
        TAX
        STX    DX
        LDA    #0
        SBC    #0
        STA    DX+1
        BPL    MVS010
        TXA
        EOR    #255
        CLC
        ADC    #1
        TAX
MVS010  STX    HOLD1
        LDA    Y2
        SEC
        SBC    Y1
        TAX
        STX    DY
        LDA    #0
        SBC    #0
        STA    DY+1
        BPL    MVS020
        TXA
        EOR    #255
        CLC
        ADC    #1
        TAX
MVS020  STX    HOLD2
        CPX    HOLD1
        BCC    CASE1
        JMP    CASE2
CASE1   LDA    #1
        BIT    DX+1
        BPL    MVS030
        LDA    #255
MVS030  STA    COUNT
        LDA    HOLD2
        STA    STEPN
        LDA    HOLD1
        STA    STEPD
MVS040  LDA    COUNT
        STA    HOLD3
        LDA    #0
        STA    HOLD3+1
        LDX    #1
        BIT    DY+1
        JSR    CKSTEP
        JSR    DOMOVE
        LDA    X1
        CLC
        ADC    COUNT
        STA    X1
        CMP    X2
        BNE    MVS040
        RTS
CASE2   LDA    #1
        BIT    DY+1
        BPL    MVS050
        LDA    #255
MVS050  STA    COUNT
        LDA    HOLD1
        STA    STEPN
        LDA    HOLD2
        STA    STEPD
MVS060  LDA    #0
        STA    HOLD3
        LDA    COUNT
        STA    HOLD3+1
        LDX    #0
        BIT    DX+1
        JSR    CKSTEP
        JSR    DOMOVE
        LDA    Y1
        CLC
        ADC    COUNT
        STA    Y1
        CMP    Y2
        BNE    MVS060
        RTS
CKSTEP  BMI    CKS010
        LDA    CSTEPN
        CLC
        ADC    STEPN
        STA    CSTEPN
        CMP    STEPD
        BCC    CKRTS
        SEC
        SBC    STEPD
        STA    CSTEPN
        INC    X1,X
        INC    HOLD3,X
CKRTS   RTS
CKS010  LDA    CSTEPN
        CMP    STEPN
        BCC    CKS020
        JMP    CKS030
CKS020  DEC    X1,X
        DEC    HOLD3,X
        LDA    CSTEPN
        CLC
        ADC    STEPD
        STA    CSTEPN
CKS030  LDA    CSTEPN
        SEC
        SBC    STEPN
        STA    CSTEPN
        RTS
DOMOVE  LDA    #128
        STA    CODE2
        LDX    #14
DOM010  LDA    CODE1
        AND    CODE2
        BEQ    DOM020
        LDA    V,X
        CLC
        ADC    HOLD3
        STA    V,X
        LDA    V+1,X
        CLC
        ADC    HOLD3+1
        STA    V+1,X
DOM020  LSR    CODE2
        DEX
        DEX
        BPL    DOM010
        LDY    #32
        JSR    WAIT
        RTS
BANG    LDA    #15
        STA    54296
        LDA    #10
        STA    54277
        LDA    #30
        STA    54273
        LDA    #128
        STA    54276
        LDA    #129
        STA    54276
        RTS
WAIT    LDX    #255
WAIT1   DEX
        BNE    WAIT1
```

```
        DEY
        BNE   WAIT1
        RTS
CIRCLE  LDA   #0
        STA   ANGLE
        LDA   #255
        STA   CODE1
CIR010  LDA   #0
        LDY   ANGLE
        JSR   $B391     ;FLOAT
        JSR   $BC0F     ;FA1 TO FA2
        LDA   #<PID180
        LDY   #>PID180
        JSR   $BBA2     ;MEM TO FA1
        JSR   $BA2B     ;MULTIPLY
        LDX   #<HLD1FP
        LDY   #>HLD1FP
        JSR   $BBD4     ;FA1 TO MEM
        JSR   $E26B     ;SINE
        LDX   #0
        JSR   CRCOMP
        LDA   #<HLD1FP
        LDY   #>HLD1FP
        JSR   $BBA2     ;MEM TO FA1
        JSR   $E264     ;COSINE
        LDX   #2
        JSR   CRCOMP
        LDX   #0
        STX   HOLD1
CIR020  LDX   HOLD1     ;PLOT
        LDA   CENX      ; FIRST
        CLC             ; QUADRANT
        ADC   X2,X      ; VALUES
        STA   X1
        LDA   CENX+1
        ADC   X2+1,X
        STA   X1+1
        LDA   CENY
        CLC
        ADC   DX,X
        STA   Y1
        JSR   BITPLT
        LDX   HOLD1     ;PLOT
        LDA   CENX      ; SECOND
        SEC             ; QUADRANT
        SBC   X2,X      ; VALUES
        STA   X1
        LDA   CENX+1
        SBC   X2+1,X
        STA   X1+1
        JSR   BITPLT
        LDX   HOLD1     ;PLOT
        LDA   CENY      ; THIRD
        SEC             ; QUADRANT
        SBC   DX,X      ; VALUES
        STA   Y1
        JSR   BITPLT
        LDX   HOLD1     ;PLOT
        LDA   CENX      ; FOURTH
        CLC             ; QUADRANT
        ADC   X2,X      ; VALUES
        STA   X1
        LDA   CENX+1
        ADC   X2+1,X
        STA   X1+1
        JSR   BITPLT
        INC   HOLD1
        INC   HOLD1
        LDA   HOLD1
        CMP   #2
        BEQ   CIR020
        INC   ANGLE
        INC   ANGLE
        LDA   ANGLE
        CMP   #46
        BCS   CIR030
        JMP   CIR010
CIR030  RTS
CRCOMP  STX   HOLD1
        JSR   $BC0F     ;FA1 TO FA2
        LDA   #0
        LDY   RADIUS
        JSR   $B391     ;FLOAT
        JSR   $BA2B     ;MULTIPLY
        LDX   #<HLD2FP
        LDY   #>HLD2FP
        JSR   $BBD4     ;FA1 TO MEM
        JSR   $B1BF     ;UNFLOAT
        LDA   HOLD1
        EOR   #2
        TAX
        LDA   101
        STA   X2,X
        LDA   100
        STA   X2+1,X
        LDA   #<HLD2FP
        LDY   #>HLD2FP
        JSR   $BBA2     ;MEM TO FA1
        JSR   $BC0F     ;FA1 TO FA2
        LDA   #<YXRAT
        LDY   #>YXRAT
        JSR   $BBA2     ;MEM TO FA1
        JSR   $BA2B     ;MULTIPLY
        JSR   $B1BF     ;UNFLOAT
        LDX   HOLD1
        LDA   101
        STA   DX,X
        RTS
PID180  .BYTE 123,142,250,53,17
YXRAT   .BYTE 128,192,210,13,41
BITPLT  LDA   #0
        STA   $FD
        STA   $FE
        STA   $A4
        LDA   X1+1
        BEQ   PL020
        CMP   #2
        BCS   PL010
        LDA   X1
        CMP   #<320
        BCC   PL020
PL010   SEC
        RTS
PL020   LDA   Y1
        CMP   #200
        BCS   PL010
        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   X1        ;8*INT(X/8)
        AND   #248
        CLC
        ADC   $FD
        STA   $FD
        LDA   X1+1
        ADC   $FE
        CLC
        ADC   #>8192    ;+8192
        STA   $FE
        LDA   Y1        ;Y AND 7
        AND   #7
        TAY
        LDA   X1        ;X AND 7
        AND   #7
        STA   $A3
        LDA   #7
        SEC
        SBC   $A3       ;7-(X AND 7)
        TAX
        LDA   #1
        CPX   #0
        BEQ   PL060
PL050   ASL   A         ;2^(7-(X
        DEX             ; AND 7))
        BNE   PL050
PL060   BIT   CODE1
        BPL   PL070
        ORA   ($FD),Y
        JMP   PL080
PL070   EOR   ($FD),Y
PL080   STA   ($FD),Y
        RTS
        .OPT  LIST
        .END
```

APPENDIX H

Mathematical Formulae

The basic mathematical formulae used to calculate MACS diagnostic scores are as follows:

Steady Position	SD(X=-45..-7) + SD(Y=-45..-7)
Aiming	RD(Target,X,AVG(X=-45..-7),TargetY,AVG(Y=-45..-7))
Trigger Squeeze	RNG(X=-6..+3) + RNG(Y=-67..+3)
Location	RD(TargetX,BulletX,TargetY,BulletY)

SD is an abbreviation for standard deviation; thus, the steady position score is the standard deviation of the X axis readings for the 45th to 7th reading before trigger closure. This is added to the standard deviation of the Y axis readings.

RD is an abbreviation for radial distance. It is simply the Pythagorean formula: the square root of the difference in the X values, squared, plus the difference in the Y values, squared. In the aiming measure, the average X and Y values (AVG) over the time window are subtracted from the target X and Y values. In the shot location measure, the bullet strike X and Y values (defined as the last reading taken before trigger squeeze, or -1) are subtracted from the target X and Y values.

RNG is an abbreviation for range. It is simply the difference between the maximum value and the minimum value. The time window for the trigger squeeze score is from 6 readings before trigger closure through 3 readings after trigger closure.

A factor of 1.6 is multiplied to the Y axis scores in each of the above formulae to compensate for the greater length of actual pixels in the Y axis. The mathematical standards are shown in Table H-1 below. S signifies the actual score.

Table H-1

Standards in the BRM Program for Diagnostic Scores and Shot Location

Supported Postion	Excellent	Good	Average	Below Average	Poor
Steady Position	$0 \le S \le 2.5$	$2.5 < S \le 5$	$5 < S \le 8$	$8 < S \le 10.5$	$10.5 < S$
Aiming	$0 \le S \le 1.6$	$1.6 < S \le 3.5$	$3.5 < S \le 6.5$	$6.5 < S \le 9$	$9 < S$
Trigger Squeeze	$0 \le S \le 2.5$	$2.5 < S \le 5$	$5 < S \le 8$	$8 < S \le 10.5$	$10.5 < S$
Shot Location	$0 \le S \le 1.6$	$1.6 < S \le 3.5$	$3.5 < S \le 6.5$	$6.5 < S \le 9$	$9 < S$

Unsupported Postion	Excellent	Good	Average	Below Average	Poor
Steady Position	$0 \le S \le 4$	$4 < S \le 6.5$	$6.5 < S \le 9.5$	$9.5 < S \le 12$	$12 < S$
Aiming	$0 \le S \le 2$	$2 < S \le 4.5$	$4.5 < S \le 8$	$8 < S \le 11$	$11 < S$
Trigger Squeeze	$0 \le S \le 4$	$4 < S \le 7$	$7 < S \le 11$	$11 < S \le 14.5$	$14.5 < S$
Shot Location	$0 \le S \le 2$	$2 < S \le 4.5$	$4.5 < S \le 8$	$8 < S \le 11$	$11 < S$

Breath control is "OK" if steady position is excellent, good or average. "Check" is displayed if steady position is below average or poor. The breath control score is presented in the same color as the steady position score.

END

FILMED

1-90

DTIC