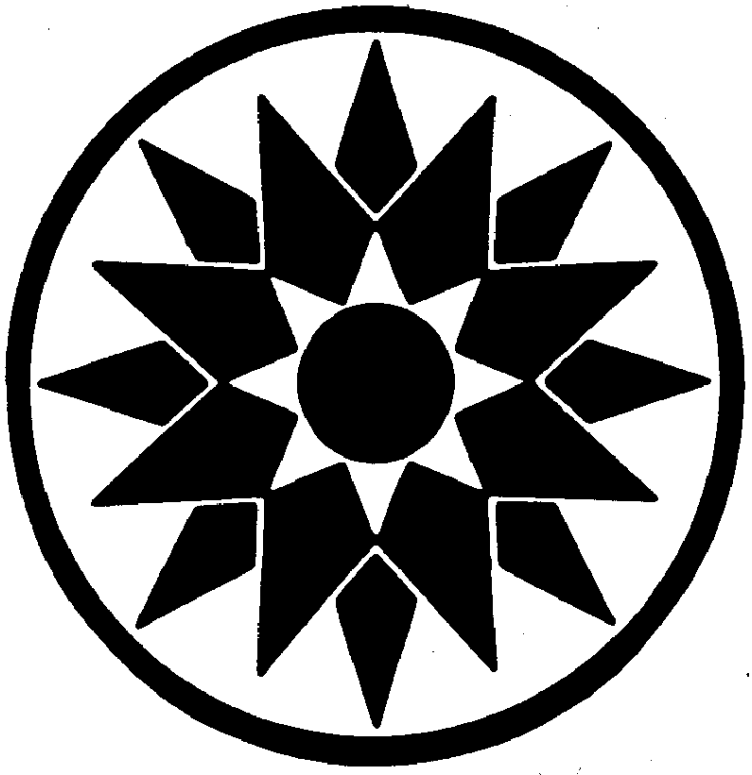
ARROW DYNAMICS

Teacher's Guide



SUNBURST
COMMUNICATIONS

ARROW DYNAMICS: THE PROGRAMMING GAME

Author: Stephen R. Shalom
Associate Professor of Political Science
William Paterson College
Wayne, New Jersey

Editors: Susan Markey
Marge Kosel
Tom Prosen

For information, or a free Microcomputer Courseware Catalog, write:
SUNBURST COMMUNICATIONS, INC.
39 WASHINGTON AVE.
PLEASANTVILLE, NEW YORK, 10570

Call toll-free: (800) 431-1934
(In New York, Alaska, Hawaii and Canada, call collect (914) 769-5030.)

ARROW DYNAMICS: THE PROGRAMMING GAME

TABLE OF CONTENTS

INTRODUCTION

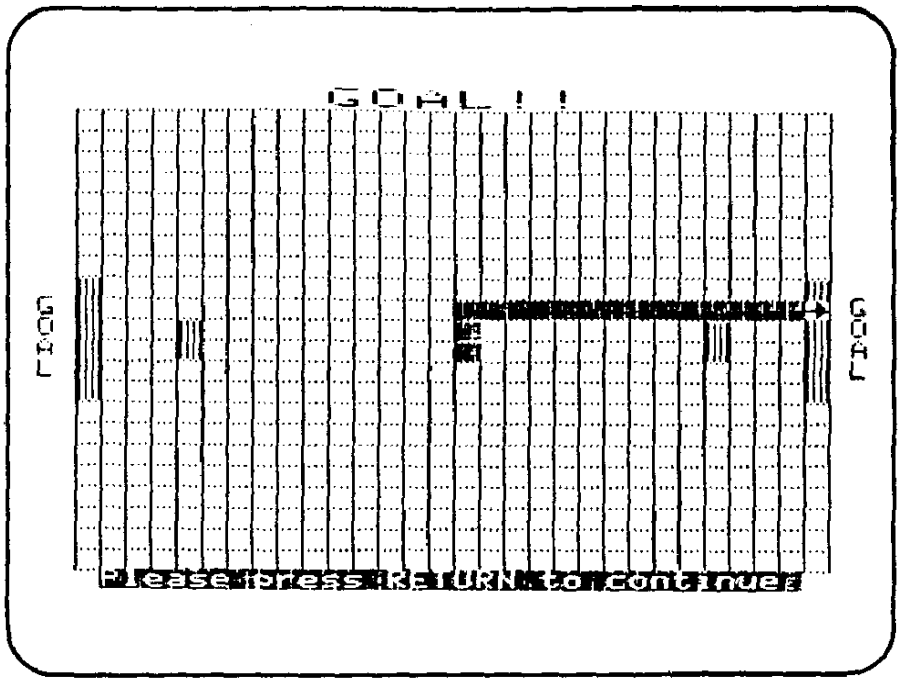
ARROW DYNAMICS: THE PROGRAMMING GAME combines the challenge of chess with the structure of programming to give students practice in logical thinking and the formulation of strategy.

The object of the game is to move an arrow across a gameboard to one of two goals using special programming statements. Obstacles, prisms and mirrors may be added to the game to increase the challenge and enhance creative thinking.

ARROW DYNAMICS can be played four ways: as a solitaire game, cooperatively with another player, competitively with another player, or as a puzzle. Each game type has a variety of player-controlled difficulty levels. The Apple version of ARROW DYNAMICS contains an editor option which allows teachers to create their own puzzles.

ARROW DYNAMICS is designed for a 48K Apple II, Apple II Plus and Apple IIe and the TI99/4A with TI BASIC. A color television or monitor is required. If you need assistance in operating your computer, check with the sections of this guide entitled "Apple II Plus: Working With The Computer" or "TI99/4A: Working With The Computer" and "What Happens If...?" or call Sunburst Communications, toll-free, at 800-431-1934.

*NOTE: The diagrams and samples used throughout this guide have been taken from the Apple version of ARROW DYNAMICS. The TI version may vary slightly in wording and appearance but the concepts are the same.



Skills: logic, strategy

Students Work With: color, sequence, direction

Grade Level: 4-12

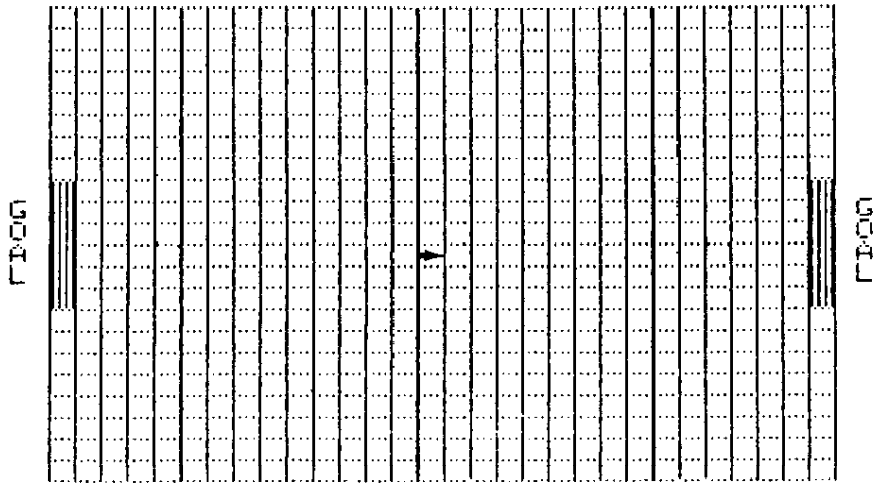
Reading Level: Grade 4 (Fry)

Time Required: 5-30 minutes

Objectives:
1. To develop the logical thought processes needed for programming.
2. To learn the structural elements of programming.
3. To improve the cognitive skills necessary for deciding and evaluating strategies.
4. To learn to work with others to reach a common goal.
5. To reinforce the concept of direction (left and right) from different perspectives.

GENERAL DESCRIPTION

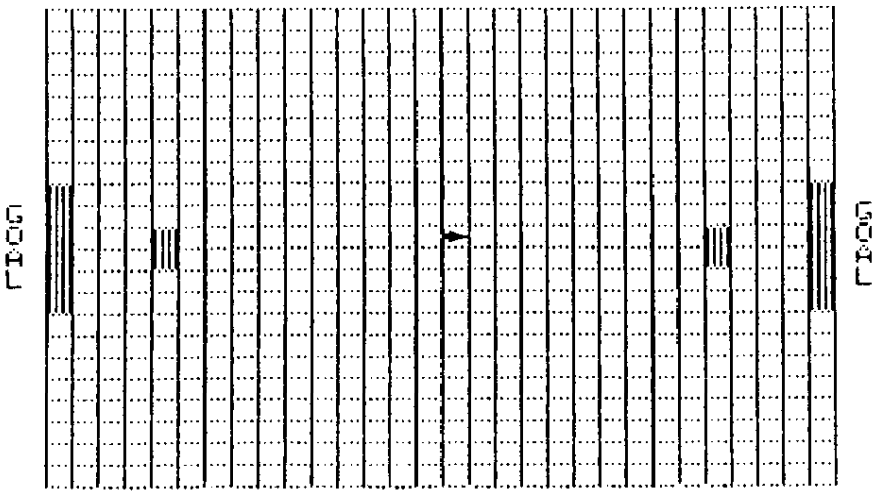
In ARROW DYNAMICS, the object of the game is to move an arrow across a board to a goal:



To do this, the player(s) must write directions for the arrow to follow. The directions are written in a series of program statements. In the example above, the player might enter a statement like this:

```
10 MOVE 15
```

Sound simple? Yes, sometimes. But what if <u>obstacles</u> were placed in the way?

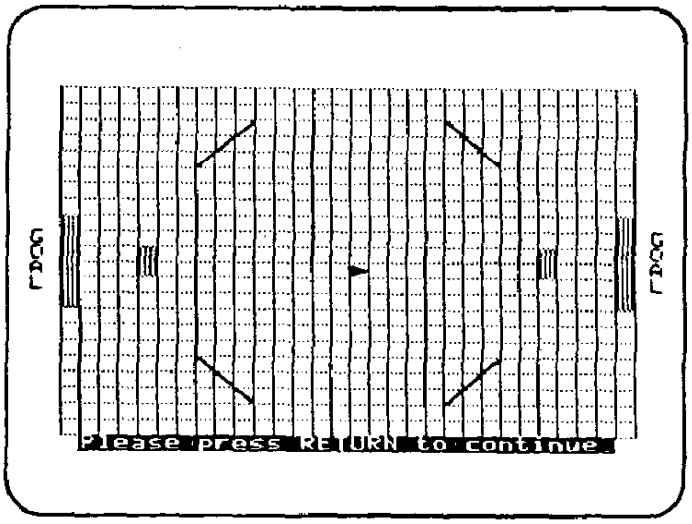


Look at the gameboard from the previous page. Using "TURN" and "MOVE" statements, the player might enter:

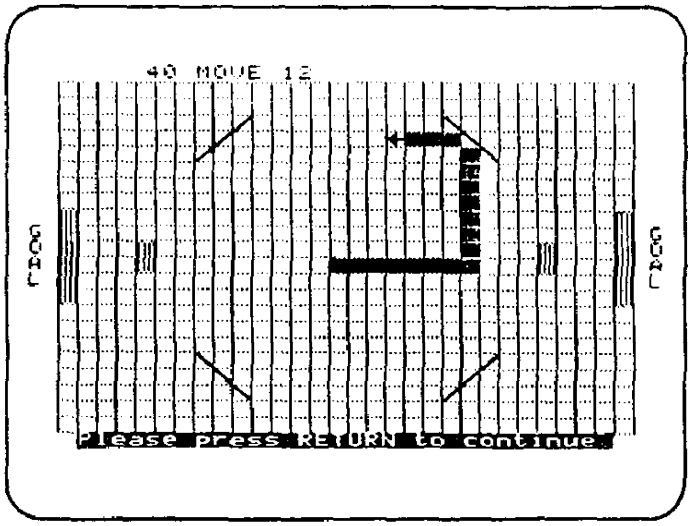
```
10 MOVE 9
20 TURN LEFT
30 MOVE 1
40 TURN RIGHT
50 MOVE 5
```

to reach a goal. (The goal can be approached from 3 sides.)

Still not so difficult, right? Well, in ARROW DYNAMICS, there are many variables that can be added to the program to increase the fun and challenge. For example, consider the following situation where the game is played with MIRRORS:



When the arrow hits a mirror, it will be deflected 90 degrees:



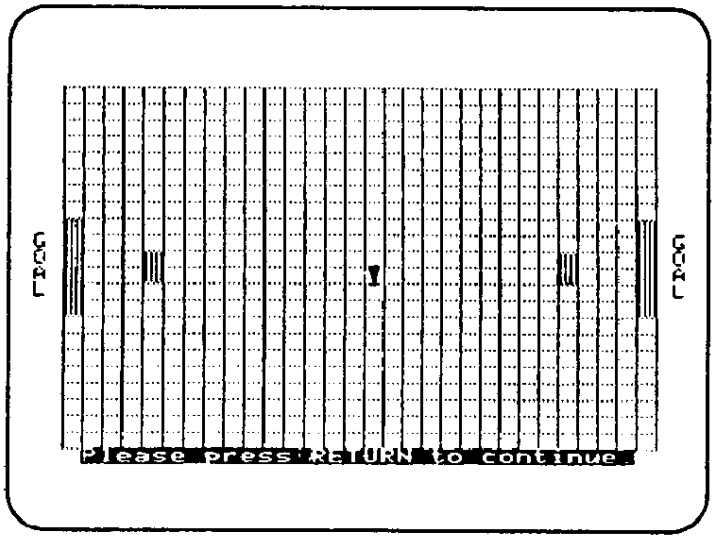
The use of mirrors can benefit the player, as well as cause problems. For example, the player can purposely lead an arrow into a mirror to eliminate the necessity of adding a TURN statement. On the other hand, mirrors can present a real problem, particularly in the 2-player, competitive game mode, where one player is trying to thwart the other player's advances towards a goal by sending the arrow in many directions.

Another feature of ARROW DYNAMICS which requires players to anticipate the unexpected and to plan ahead is computer-generated statement numbers (line numbers). When the computer chooses the line numbers, the planning becomes more difficult. For example, suppose the computer gave the player line 30 on which to enter a program statement. The player decides to:

```
30 TURN RIGHT
```

Next, it gives the player line 10. How does this new line number affect the player's plan to use the TURN RIGHT to reach the goal? What would happen if the computer selected statement number 180 instead of 10?

Finally, consider the situation where the computer has randomly given the player statement numbers. The player has thus far entered the following statements:




```
 50 TURN LEFT
 90 MOVE 3
140 TURN RIGHT
170 MOVE 11
```

The computer now gives the player line number 100. Is there a statement that can be entered which will result in moving the arrow to a goal?

Yes, with still another feature of ARROW DYNAMICS -- loop statements. (A loop is a group of instructions that are repeated.) Using a loop statement, the player can avoid waiting for that second needed line number between 100 and 140 in order to complete the program. The player could enter one statement which would perform two commands:

```
 50 TURN LEFT
 90 MOVE 3
100 REPEAT FROM 50
140 TURN RIGHT
170 MOVE 11
```

In this program, the arrow would TURN LEFT, MOVE 3, TURN LEFT, MOVE 3 TURN RIGHT, MOVE 11. Thus, as you can see, loops can make a program more efficient.

Summary

The above examples are just a few of the many variations of play available in ARROW DYNAMICS. In short, ARROW DYNAMICS can be played in four different modes:

- by one player as a solitaire game,
- with two players cooperatively working towards the same goal,
- with two players working competitively,
- or by one player working a puzzle.

In addition, each mode of the game may be played at varying levels of difficulty. To adjust the difficulty level, the following choices are offered at the beginning of the game. The game may be played:

- with either (1) small obstacles or (2) large obstacles in front of each goal,

- (1) without mirrors or prisms, (2) with mirrors, which will deflect the arrow, or (3) with prisms, which will deflect the arrow only if the arrow is a different color from the prism,

- either (1) not permitting loop statements or (2) permitting loops statements,

- with statement numbers (line numbers) ranging from (1) 10-200, (2) 10-400, or (3) 10-600. (Statement numbers are entered in multiples of 10.)

- with statement numbers selected by the player or randomly generated by the computer, depending upon which mode of play you are in.

For detailed instructions on how to play, refer to the appropriate sections of this guide. It's a good idea to begin your students in the solitaire mode; have them play the easiest options first (choose option 1 of each question). Then they can gradually increase the difficulty level until they've worked with all the variables and understand the game.

PRE-GAME OPTIONS

Before you begin playing ARROW DYNAMICS, several questions must be answered in order to set up the initial play conditions of each game. The options selected will directly affect the difficulty level of the game.

Selection of Game Mode

To begin, load the program into the computer (for instructions, see page 34 for the Apple, 35 for the TI). The following menu will appear:

```
CHOOSE A GAME:
1.  SOLITAIRE
2.  2 PLAYER, COOPERATIVE
3.  2 PLAYER, COMPETITIVE
4.  PUZZLE
5.  INSTRUCTIONS
6.  END
```

(Apple version only)

Option #1 - solitaire - allows the game to be played by one player.

Option #2 - 2 player, cooperative - lets two players work cooperatively towards the same goal.

Option #3 - 2 player, competitive - pits two players against each other, each trying to get to a different goal.

Option #4 - puzzle - is played by one player. Here, the player must figure out a missing statement from a program to move the arrow to a goal.

Option #5 - instructions - gives brief, onscreen instructions.

Option #6 - end - ends the game.

Begin ARROW DYNAMICS in the solitaire mode and learn the intricacies of the program before trying other game modes.

Entering a Name

After selecting the game mode, the player will be asked to enter his/her name. There is a maximum of 15 characters on the Apple (14 on the TI). In the 2-player modes, each player may enter his or her name. If the option to enter names is waived (by pressing RETURN or ENTER), the computer will identify players as Player 1 and Player 2.

Selection of Statement Numbers

In the solitaire mode, the statement numbers (line numbers) for the program statements may be:

- selected by the player or
- randomly generated by the computer.

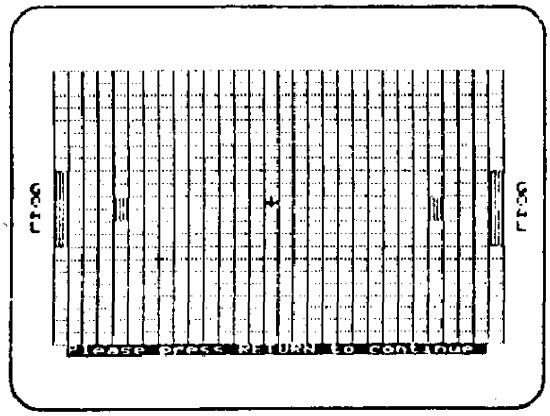
It is easier for the player to enter the statement numbers because the player can sequentially build a program to reach a goal. If the computer randomly generates the line numbers, this creates an interesting challenge; the player can't start at the beginning of a program and build towards the end. For example, the computer may ask the player to enter a statement for line number 50 first. Next, the computer may ask for line number 120 then 70, then 10, then 130, etc.

In the 2-player modes, the line numbers are always generated at random by the computer. Players do not have the option of choosing the line numbers they wish to enter. Also in the 2-player games, the players take turns entering statements.

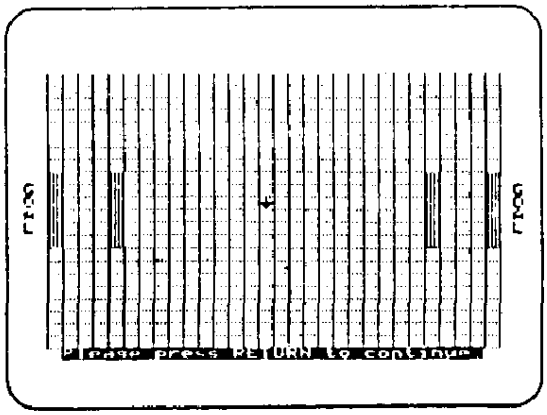
In the puzzle mode, the player is given a program with one statement missing. The player must supply a single missing statement that will direct the arrow to a goal when the program is RUN.

Obstacles: Small or Large

The ARROW DYNAMICS Board is a rectangular grid, 30 squares across and 22 squares high. Six squares in the middle of the left hand column and six squares in the middle of the right hand (30th) column are designated as GOALS. In the competitive 2-person game, the goals are labeled GOAL A and GOAL B (on the TI) or with the players' names (on the Apple). In all other modes, the goals are both simply labeled GOAL. In front of each goal are obstacles. The player(s) must choose which size obstacle to use. Below are examples of SMALL OBSTACLES and LARGE OBSTACLES.



SMALL OBSTACLES



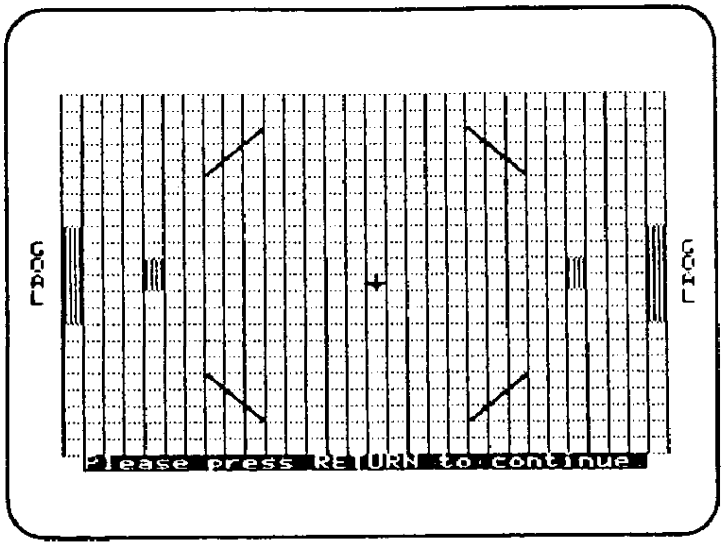
LARGE OBSTACLES

Mirrors and Prisms

The players may choose which they want to work with:

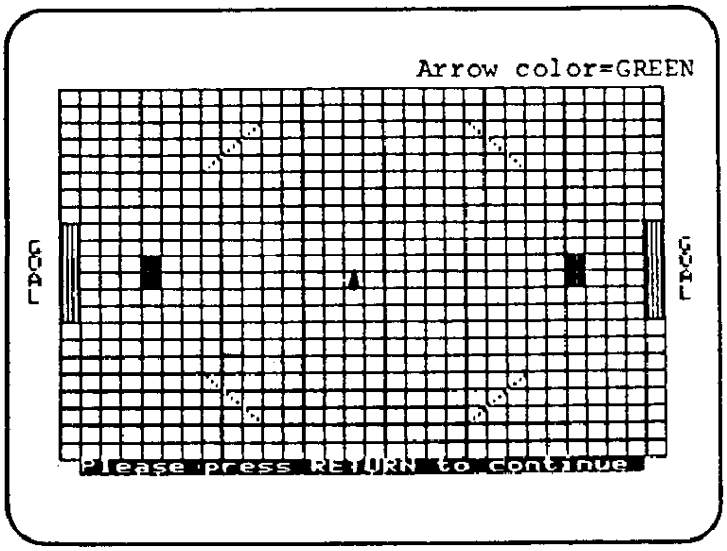
1) no mirrors or prisms
2) mirrors
3) prisms

In Option 2, there are four MIRRORS, each consisting of three diagonally connected squares with a line running through them at a 45 degree angle. The four mirrors are always located symmetrically with the center of the board, as shown below:



If an arrow hits a mirror from any direction, it will be deflected 90 degrees. When the arrow is adjacent to and facing a mirror, its next step places it on the mirror, turned 90 degrees; on the next step, it will be adjacent to and facing away from the mirror.

Option 3 is played with PRISMS. These are the same as the mirrors except that the prisms in the upper left and lower right areas of the board are one color and the other two are another color.



A prism which is a different color from the arrow deflects the arrow just as a mirror does. A prism which is the same color as the arrow allows the arrow to pass through undeflected as if there were no structure there at all. NOTE: If you end a move directly on a prism, you cannot then change the color of the arrow in order to pass it through the prism. The color must be changed prior to landing on the prism.

Range of Statement Numbers

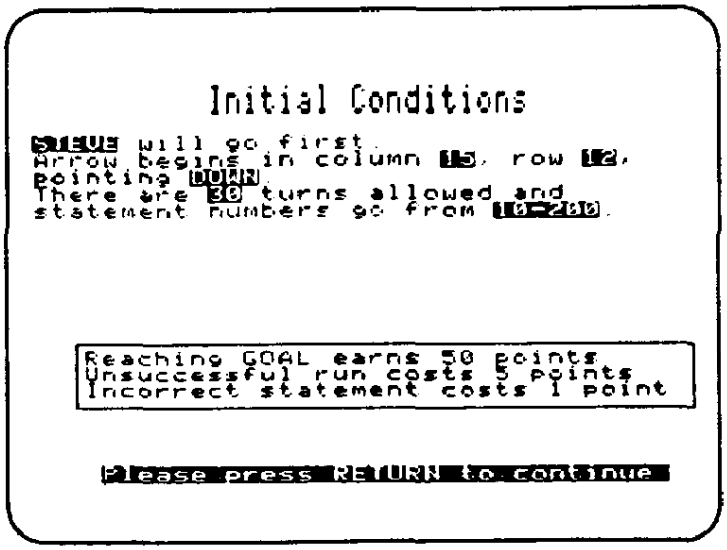
Players may choose the maximum number of chances to move their arrow to a goal. This is done by the number of statement numbers (line numbers) they select. Statement numbers range from 10-200, 10-400 and 10-600.

If the players select a range from 10-200, the player receives 30 turns. The 10-400 range gives the player 60 turns and the 10-600 range gives the player 90 turns.

Note that selecting the highest number of turns doesn't mean the game will be easier. Other factors can increase the difficulty of the game (particularly if the computer randomly chooses the statement numbers).

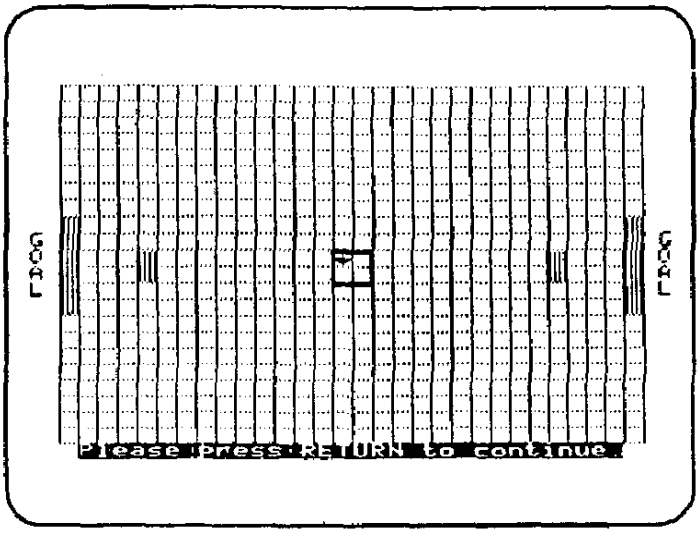
Initial Conditions

Once the pre-game selections are made, the players will be given the Initial Conditions of the game. These consist of the location of the arrow, its direction, and the number of moves left. The initial color of the arrow will be indicated if playing with prisms. In a 2-player mode, the player with the first move will be identified. At the bottom of the screen, the player will be shown the scoring structure as indicated below. (The scoring structure is not shown on the TI version.)



MOVEMENT OF THE ARROW

The arrow's starting position is random and begins in one of the four central squares of the board.



The initial direction in which the arrow points (up, down, right or left) is chosen at random. In the versions of the game without prisms, the arrow is colored white (on the Apple) or green (on the TI). In the prism version, the color of the arrow is chosen by the computer at random and will match the color of two of the prisms (on the TI). (In the Apple version, the arrow is white and the color of the arrow is printed at the top of the screen.)

The arrow moves horizontally or vertically. If it hits the edges of the board or an obstacle, it will rebound and reverse direction In (Diagram #1) below, the last statement was MOVE 17. The arrow moved 15 spaces and hit the edge, the arrow reversed direction (using 1 MOVE) and moved 1 more space to complete the command, MOVE 17.

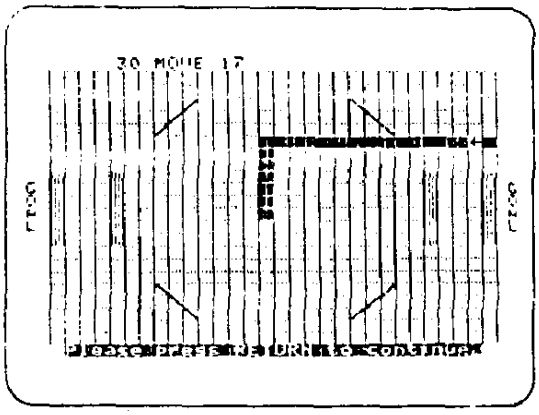


Diagram #1

If the arrow bumps into a mirror from any direction, it will be deflected 90 degrees. When the arrow is adjacent to and facing a mirror (Diagram #1), its next move places it on the mirror, turned 90 degrees (Diagram #2); on the next move, it will be adjacent to and facing away from the mirror (Diagram #3).

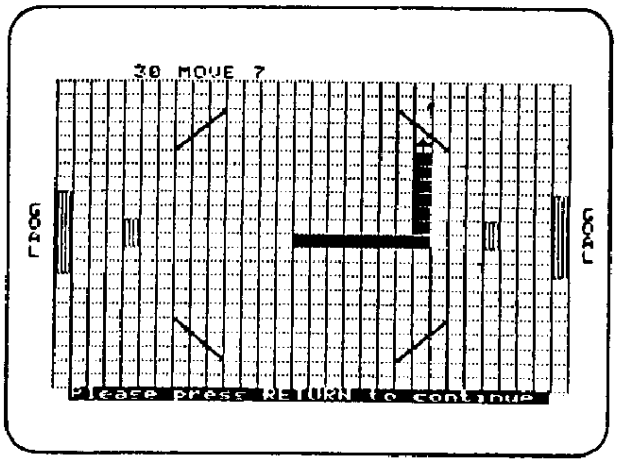


Diagram #1

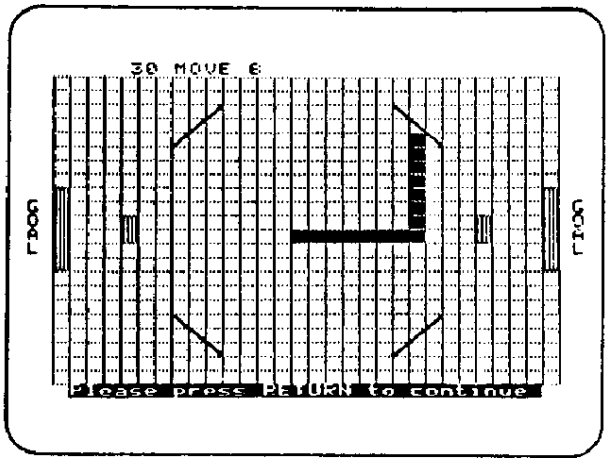


Diagram #2

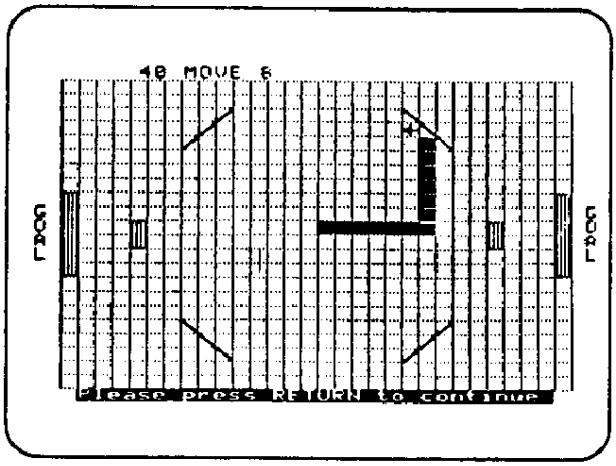


Diagram #3

A prism which is a different color from the arrow deflects the arrow just as a mirror does. A prism which is the same color as the arrow allows the arrow to pass through undeflected as if it was not there.

THE PLAYER'S TURN

In the solitaire game, either the player or the computer picks the statement numbers. In the other versions, the computer randomly picks the statement numbers. Also in the 2-player games, the computer randomly chooses who will go first and the players then alternate turns. In the solitaire mode, the player continues playing until the game is over. Generally, a player's turn consists of three distinct parts:

1. First, the computer will pick a random statement number (see page 16 for section on Random Events).

2. The player then may enter a statement for the indicated line or choose any or all of the following three options:

   - LIST THE PROGRAM

   - GET HELP

   - DISPLAY THE BOARD

If you choose to LIST THE PROGRAM, the statement number to be entered is indicated with a question mark. The board and the program listing cannot be displayed on the screen at the same time. Listing the program displays on the screen the starting conditions (initial arrow location, direction, and, if appropriate, color), the number of moves left, the permissible range of statement numbers, and all the statements entered in sequential order.

If you choose to GET HELP, the allowable statements you may use are displayed (see Entering Program Statements on page 17 for a detailed description of these statements).

If you select DISPLAY THE BOARD, the board will be displayed with the arrow in its starting position.

3. After an acceptable statement has been entered, the player may:

   - RUN THE PROGRAM
   - END THE TURN
   - DISPLAY THE SCORE SO FAR
   - DISPLAY THE BOARD in its starting position.

If the program is RUN, the board is displayed and the program statements executed in order of the statement numbers. To assist the players in following the running of the program, the statements are printed on the top of the screen as they are executed. Execution takes place at a relatively slow speed, leaving a trail behind as the arrow moves.

In the 2-player games, selecting END THE TURN transfers play to the other player. In the solitaire mode, after selecting END THE TURN, the player may then list the program or pick the next statement number (unless the computer is randomly selecting the statement number, in which case the next statement number is given). The player must always END THE TURN before the next statement can be entered.

The other two options, DISPLAY THE SCORE or DISPLAY THE BOARD, will show the score or display the board and when finished will return to the same menu.

RANDOM EVENTS

In those versions of the game in which the computer picks the statement numbers, occasionally the computer will randomly inform the player that he or she may pick the statement number. In the 2-player games, the computer will occasionally inform the player that the chance to enter a statement must be forfeited. These options occur infrequently.

If the player is allowed to pick the statement number, he or she may choose a number that has been picked before, just as the computer may occasionally select a previously chosen statement number. The player may also first LIST the program before choosing the statement number.

The arrow may reach a goal from the top or bottom as well as from the front. The arrow need not reach the goal in its last move; the goal is reached if at any point during a RUN of the program the arrow lands on one of the goal squares.

All levels of ARROW DYNAMICS accept these five statements:

1. MOVE N (where N is a whole number from 1 to 30) - moves the arrow N squares in the direction in which the arrow is pointing. (Numbers from 1 to 9 may be entered as either 1, 2, 3, etc. or 01, 02, 03, etc.)

2. TURN RIGHT - rotates the arrow clockwise 90 degrees from the direction in which it was pointing.

3. TURN LEFT - rotates the arrow counter-clockwise 90 degrees from the direction in which it was pointing.

4. A BLANK LINE - erases the statement that you previously entered (with the same statement number) or skips the line. To do this, press RETURN (if you have an Apple) or ENTER (if you have a TI).

5. A SLASH (/) - leaves unchanged any previously entered statement with the same statement number.

When using PRISMS, two additional statements are permitted:

6. COLOR=PURPLE or COLOR=GREEN (on the Apple)

   COLOR=BLUE or COLOR=RED (on the TI)

   These statements change the color of the arrow. If the arrow is already the indicated color, the statement will have no effect.

Finally, in the versions of the game that permit LOOPS (a LOOP is a group of instructions that are repeated) the following statement is allowed:

7. REPEAT FROM S - (where "S" is a statement number of a non-blank statement which is lower than the statement number of the entered statement).

For example, in the program:

```
10 TURN RIGHT
20 MOVE 5
30 REPEAT FROM 10
40 TURN LEFT
```

the arrow would TURN RIGHT, MOVE 5, TURN RIGHT, MOVE 5, TURN LEFT.

The REPEAT FROM statement causes the program to repeat all the instructions from statement S up to, but not including, the REPEAT FROM statement. After repeating these statements, the program continues with the next statement after the REPEAT FROM.

No other REPEAT FROM statement may appear between statement S and the entered statement (see following examples), nor may the entered statement itself fall between another REPEAT FROM statement and the latter's S. (In other words, "nesting" or embedding of REPEAT FROM statements is forbidden.) It is permissible, however, to erase or change statement S -- or any statement between S and the REPEAT FROM statement -- on future turns.

For example, if the program thus far, is:

```
40 MOVE 2
50 REPEAT FROM 40
80 MOVE 5
90 TURN RIGHT
150 MOVE 10
```

and you are asked to enter statement 130, then REPEAT FROM 80 or REPEAT FROM 90 are permissible; all other REPEAT FROMs would be incorrect. For example, REPEAT FROM 40 or REPEAT FROM 50 would embed one loop within another:

(Note: This is an incorrect program.)

```
40 MOVE 2
50 REPEAT FROM 40
80 MOVE 5
90 TURN RIGHT
130 REPEAT FROM 40
150 MOVE 10
```

REPEAT FROM 130 or REPEAT FROM 150 (or any number greater than 130) reference statement numbers that are not smaller than the entered statement (130). REPEAT FROM 10 (or 20, 30, 60, 70, 110, 120 or 140) reference blank statements.

If, in another example, the program consists of the statements:

```
40 MOVE 2
60 TURN LEFT
120 REPEAT FROM 60
150 TURN RIGHT
```

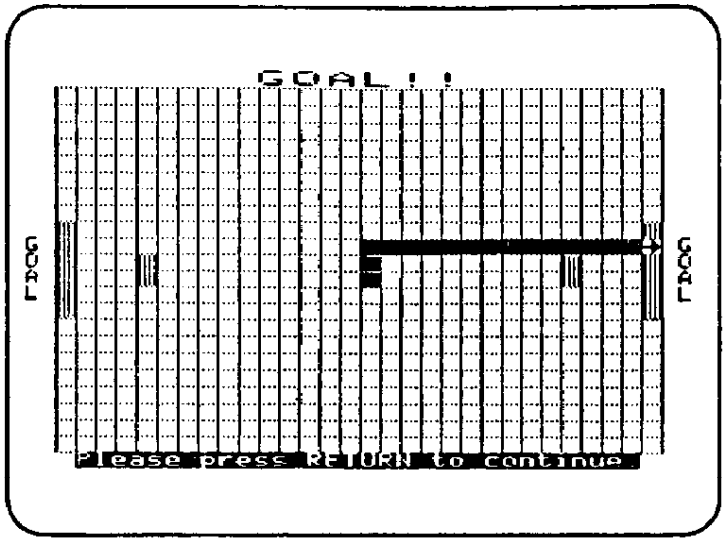
and statement 90 must be entered, no REPEAT FROM statement is permissible since statement 90 falls between statement 120 and its S, 60.

A player types in an allowable statement and then presses the ENTER (on the TI) or RETURN (Apple). Before the key is pressed, the statement may be corrected or changed by using the back arrow key. Statements must be typed precisely as listed on the previous pages.

The computer checks for errors in the syntax of entered statements and rejects those that are incorrect. The player may continue entering statements until a correct statement is accepted; however, there is a one-point penalty charged for each unacceptable statement that is entered.

THE END OF THE GAME

The game ends when a goal is reached or when the maximum number of turns has been completed, whichever comes first. The maximum number of turns is set at the beginning of the game to 30, 60, or 90, depending on whether the statement numbers range from 10-200, 10-400 or 10-600. If the player reaches a goal, the word GOAL!! flashes at the top of the screen (on the TI, music will also play).



When the score is displayed during the game, the number of turns taken and the number remaining are indicated.

At the end of the game, the score and the number of turns taken are displayed. In the 2-player competitive game, the game is won by the player with the higher score. In the 2-player cooperative game and the solitaire game, the player(s) try to beat their previous score and to reach a self-determined goal (score) in as few turns as possible.

After the game is over, the program may still be RUN or LISTed again.

The computer will then print:

DO YOU WANT TO:
1. PLAY AGAIN
2. END

If you select #1, PLAY AGAIN, you will be returned to the following menu:

CHOOSE A GAME:
1. SOLITAIRE
2. 2 PLAYERS, COOPERATIVE
3. 2 PLAYERS, COMPETITIVE
4. PUZZLE
5. INSTRUCTIONS
6. END ——(Apple version only)

If you select #2, END, the computer will let you: 1) use a new diskette or 2) give instructions on how to shut off the computer (on the Apple version only).

(Note: For instructions on how to stop the program at any point in the game, see page 36.)

SCORING

Points are awarded or deducted in three situations:

1. When a goal is reached.
   When the arrow reaches a goal, 50 points are awarded and the game ends. In the 2-player competitive game, the player whose goal is reached (not necessarily the player who ran the program) receives the 50 points. Remember that players try to get to their own goal. In all other modes of the game, the 50 points are awarded regardless of which goal is reached.

2. When the run of the program is unsuccessful.
   If, during a run of the program the arrow fails to land on a goal, a 5-point-penalty is charged to the player who ran the program and the player's turn ends.

3. When a program statement is incorrectly entered.
   Each incorrectly entered statement causes a 1-point-penalty.

A player's final score is the sum of all the points awarded or deducted. In the cooperative game, the points of the two players are combined. In the competitive game, the game is won by the player with the highest score (usually, though not necessarily, the player whose goal was reached).

THE PUZZLE

ARROW DYNAMICS contains a game mode that lets students play the game as a puzzle. Each puzzle consists of a list of statements -- with one statement missing. The player is given the statement number and must enter the correct statement in order to move the arrow to one of the goals. (Some puzzles may have more than one correct answer.) As in all the other modes, in the puzzle mode, the arrow can reach the goal at the end of or during the execution of any statement. (For example, if there are 5 program lines, the arrow doesn't have to hit the goal at the end of the fifth program line; it can reach the goal any time during the running of the program.)

ARROW DYNAMICS comes with nine puzzles - three puzzles stored at each of three levels (easy, medium, difficult).

At the easy level, there are small obstacles, no loops, no mirrors and no prisms.

At the medium level, there are large obstacles, loops and mirrors.

At the difficult level, there are large obstacles, loops and prisms.

The player begins by selecting the level. (On the Apple, the player then selects the number of the puzzle within that level (1, 2, 3).) Next, a program is listed - with a question mark next to the missing statement number. For example:

```
          PROGRAM LISTING
Arrow begins in column 15, row 12,
pointing RIGHT.
You have 5 tries left.

  10 TURN LEFT
  40 MOVE 2
  70 TURN RIGHT
 110 TURN RIGHT
 130 (?)
 150 MOVE 15
```

Then, the player presses ENTER or RETURN to get the following screen:

(NAME), do you want to:

1. ENTER STATEMENT 130 (the missing statement #)
2. LIST THE PROGRAM
3. GET HELP
4. DISPLAY THE BOARD

If a statement is entered that does not result in a program that directs the arrow to a goal, the player is then asked:

```
DO YOU WANT TO:

1.  TRY THIS PUZZLE AGAIN
2.  END
```

Players have up to five tries to complete the puzzle. After five unsuccessful tries, the correct answer is displayed. For example:

```
GAME OVER!

SCORE =  -25 POINTS

ONE ANSWER IS:
          130 TURN LEFT
NUMBER OF TRIES USED = 5
```

You are then shown this menu:

```
DO YOU WANT TO:

1.  RUN YOUR PROGRAM
2.  LIST YOUR PROGRAM
3.  RUN CORRECT PROGRAM
4.  LIST CORRECT PROGRAM
5.  PLAY AGAIN
6.  END
```

Players can either RUN or LIST their program so they can study the consequences of their incorrect statements. They may also RUN or LIST the correct program. If they elect to PLAY AGAIN, they will be returned to the following menu:

```
CHOOSE A GAME:

1.  SOLITAIRE
2.  2-PLAYERS, COOPERATIVE
3.  2-PLAYERS, COMPETITIVE
4.  PUZZLE
5.  INSTRUCTIONS
6.  END                    —(Apple version only)
```

If players enter the correct statement into the puzzle, their scores are reported and they are then returned to a menu where they can choose to RUN the program again, LIST it, play ARROW DYNAMICS again, or stop playing.

THE PUZZLE EDITOR

The Apple version of ARROW DYNAMICS allows you to design puzzles to fit your special classroom needs. Your diskette comes with nine puzzles - three puzzles on each level. You may add up to 6 puzzles on each of the three levels.

How to use the EDITOR:

Remove the write protection sticker on the upper right-hand corner of your diskette. Insert the diskette in the disk drive and turn on your computer and monitor. Proceed to the Main menu:

```
CHOOSE A GAME:
1.  SOLITAIRE
2.  2-PLAYERS, COOPERATIVE
3.  2-PLAYERS, COMPETITIVE
4.  PUZZLE
5.  INSTRUCTIONS
6.  END              (Apple version only)
```

When the menu appears, hold the Control key down (CTRL) and press the T key (CTRL T) to get the Editor menu (this will not work on the TI):

```
DO YOU WANT TO:
1.  SEE A PUZZLE
2.  ADD A PUZZLE
3.  DELETE A PUZZLE
4.  (UN)RESTRICT PUZZLE
5.  RETURN TO MAIN PROGRAM
```

See a Puzzle

Select option #1, See a Puzzle, to see the puzzles stored on the diskette. First, you must select the difficulty level you want and the puzzle number. Then, the computer will display the program listing of your selected puzzle, including the difficulty level selected, the puzzle number, the status of the puzzle ("restricted" [students may not play the puzzle] or "unrestricted"), and the answer to the puzzle.

For example, if you choose the easy level, you would get a status report on how many puzzles are available on that level. You would then choose, say, Puzzle #1, and the following screen will appear (see next page):

```
EASY puzzle # 1 is unrestricted.
Answer:  TURN LEFT.
There are small obstacles, no
loops, and no mirrors or prisms.

10  TURN LEFT
40  MOVE 2
70  TURN RIGHT
110 TURN RIGHT
130 (?)
150 MOVE 15
```

Press RETURN to get back to the Puzzle menu. You may ask to see as many puzzles as you wish.

Add a Puzzle

The ARROW DYNAMICS diskette comes with three puzzles at each of the three levels. There is room for up to nine puzzles (the three originals plus six more) at each level. To create your own puzzle, select option #2 from the Puzzle menu, ADD A PUZZLE. Then, select the difficulty level (easy, medium, difficult) where you would like to store the puzzle. (If you select ADD A PUZZLE and there are already nine puzzles stored on the level, you will be given the message: FILE FULL. IF YOU WISH TO ADD A PUZZLE, PLEASE DELETE A PUZZLE FIRST.)

You will then create a puzzle by writing an ARROW DYNAMICS program in the solitaire mode. Then, you will see this menu:

```
EDITOR, YOU MAY
1.  PICK THE STATEMENT NUMBER
2.  LIST THE PROGRAM FIRST
```

Enter the statement number in multiples of 10 and continue playing as if you were in the Solitaire game. Once you have completed your program, RUN the program. When it reaches a GOAL, you will be asked:

```
DO YOU WANT TO:
1.  CONTINUE EDITING
2.  LIST THE PROGRAM
3.  PICK LINE # TO REMOVE
```

If you want to change the program, select #1 to continue editing. You can select #2 to review your new program. Once you are ready to select the line # number to remove, select #3. The statement you select to remove will be replaced by a question mark (?) when the puzzle is played. This will be the statement that must be figured out by the person solving the puzzle. Once you have completed the new puzzle, you will be returned to the Puzzle menu. *NOTE: If the program you write goes beyond 30 statements, you may receive a message telling you that your program is too long and requesting that you shorten it.

Delete a Puzzle

By selecting option #3, Delete a Puzzle, you may eliminate a puzzle from the diskette. Only puzzles written by you, the editor, may be deleted; the original three puzzles on each level cannot be deleted (although they may be restricted [students can't play them] - see the following section on restricting puzzles). Deleting unwanted puzzles creates room for adding new puzzles.

To delete a puzzle, simply select the difficulty level of the puzzle you wish to delete. Then the computer will tell you which (indicated by numbers) may be deleted. (Remember, you cannot delete the original three puzzles.) Type the number of the puzzle you wish to delete and press RETURN to go back to the Puzzle menu.

(Un)restrict a Puzzle

The puzzles in ARROW DYNAMICS are designed so you have the option of whether or not your students can play a particular puzzle. A "restricted" puzzle does not allow the students access to the puzzle. An "unrestricted" puzzle allows them to play the puzzle.

Select option #4, and choose the level. The computer will report the numbers of puzzles on that level and give you a status report on each puzzle (whether it is presently restricted or unrestricted). Enter the number of the puzzle you want to change. The computer will then report, for example,

EASY PUZZLE 1 WAS RESTRICTED.
NOW IT IS UNRESTRICTED.

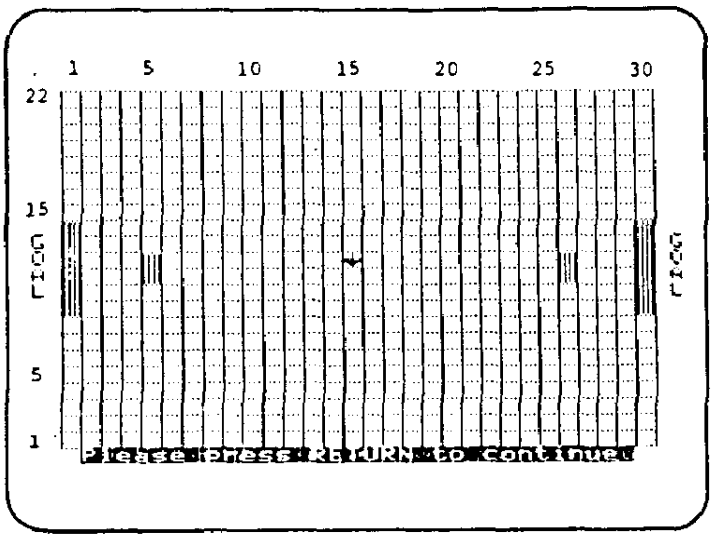
A restricted puzzle - unlike a deleted puzzle - remains on the diskette. A restricted puzzle may be unrestricted by using the same procedure.

Return to Main Program

Finally, by selecting option #5 - Return to Main Program you may return to the main program where you may select which game of ARROW DYNAMICS you wish to play.

SOME POINTS OF STRATEGY

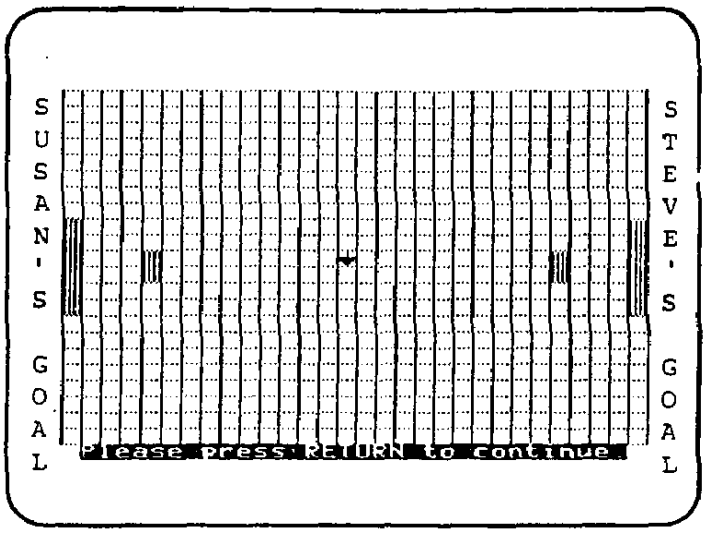
MOVE statements are additive. That is, MOVE 7 followed by MOVE 3 will have the same effect as MOVE 10. The board is 22 squares up and down, and the arrow rebounds off its edges, so any combination of statements adding up to 44 moves (2 x 22) with the arrow pointing in a vertical direction will return the arrow to the square and direction it was in before moving the 44 (assuming that no obstacles, goals, mirrors, or prisms were in the way). Likewise, since there are 30 squares across the board, 60 horizontal moves will return the arrow to its initial position (again, assuming it hits no obstacles, goals, mirrors, or prisms). If the arrow is moving horizontally between the obstacles, 40 moves will return it to its initial position. (Note, however, that you cannot exceed 30 moves in one statement. Therefore, 40 moves, for example, would require two program statements.)



Using this information can make it much easier to figure out where the arrow will be after a sequence of statements, without the necessity of counting square by square on the board. For example, if the arrow is pointing up, the sequence MOVE 15, MOVE 10, MOVE 20 has the net effect of moving the arrow upward 1 square (15+10+20=45, and 45-44=1). Or, if the arrow is pointing toward the right edge, the sequence MOVE 17, MOVE 16, MOVE 18, MOVE 15 will move the arrow 6 squares to the right (17+16+18+15=66, and 66-60=6).

There are two ways to turn the arrow around: by means of turn statements (i.e. TURN LEFT, TURN LEFT or TURN RIGHT, TURN RIGHT) or by bouncing the arrow off an edge or obstacle. If the arrow is adjacent to and pointing toward an edge, MOVE 1 will turn it around. In general, if there are N squares between the arrow and an edge or obstacle that it is facing, MOVE 2N+1 will put the arrow in the same square facing in the opposite direction. Moving 22 in an up or down direction will put the arrow in the square symmetrical about a horizontal axis to where it began and facing in the opposite direction. Moving 30 does a similar thing if the arrow is pointing left or right in a position not obstructed by an obstacle.

Example 1. Two-person competitive game, small obstacles, no mirrors or prisms, the arrow begins in column 15, row 11, pointing down. You are trying to reach the right-hand goal.



The program so far is:

```
120 TURN RIGHT
180 MOVE 20
```

and it is your turn to enter statement 90. What do you enter?

(Note: You can try out all of these examples by playing the solitaire mode and electing to pick the statement numbers yourself. If the starting position and direction of the arrow are not the same as given in the example, use the first few statements to get to the initial conditions of the example.)

If you simply enter MOVE 2 or MOVE 3, the arrow will be in line with the goals and beneath the small obstacles, but the program will move the arrow into your opponent's (the left-hand) goal. This is because a downward-facing arrow that turns right will be facing the left goal. However, if instead you enter MOVE 20 or MOVE 21,the arrow will be in the same location -- in line with the goals and beneath the small obstacles -- but pointing up rather than down. Now the TURN RIGHT instruction in statement 120 will turn the arrow toward your goal.

Example 2. Two-player cooperative game, large obstacles and mirrors, the arrow begins in column 15, row 11, pointing up. The program thus far is:

```
60 MOVE 5
80 TURN RIGHT
140 TURN LEFT
170 MOVE 5
```

and you must enter statement 90. What do you enter?

By moving the arrow 15 squares to the edge of the board with statement 90 you will have put the arrow in the same column as

the goal. However, since the arrow is facing the right edge, after the left turn in statement 140 it will be pointing away from the goal. But if statement 90 is MOVE 16 instead of MOVE 15, the arrow will be pointing away from the right edge and the left turn will head it toward the goal.

Try to give yourself as many different ways of winning as possible.

Example 3. Solitaire, small obstacles, no mirrors or prisms, statement numbers go from 10 to 200, the arrow begins in column 15, row 11, pointing down. The program so far is:

```
80 MOVE 21
200 MOVE 30
```

and you must enter statement 50. What do you enter?

MOVE 2 will place the arrow so that it is in line with the goals and beneath the obstacles. Then if statement numbers 60, 70, or 80 come up next, TURN LEFT or TURN RIGHT will head the arrow toward a goal and either statement 80 will move the arrow to that goal or, if 80 came up and was changed to a TURN statement, statement 200 will move the arrow to the goal. But if statements 90 through 200 come up, you cannot win.

On the other hand, if you entered MOVE 3 for statement 50, then you will still win if 60, 70, or 80 come up next, but you will win as well if any statement number between 90 and 190 comes up. This is because statements 50 and 80 together (MOVE 3 and MOVE 21) move the arrow 24 squares, which puts it in line with the goals and above the obstacles; then the arrow can be turned with any statement from 90 to 190 and be moved into the goal with 200. (Note that MOVE 21 for statement 50 will have the same effect.)

Play the odds.

Example 4. Two-player cooperative game, small obstacles, no mirrors or prisms, statement numbers go from 10 to 200, the arrow begins in column 15, row 11, pointing up. The program so far is just:

```
170 MOVE 20
```

and you must enter statement 90. What do you enter?

The obvious choices are (a) to move the arrow so that it is in line with the goals and unblocked by the obstacles (e.g., MOVE 2), or (b) to turn the arrow left or right. Option (a) will enable you to win on the next turn if the next statement number to come up is from 100 to 160 (and a TURN statement is entered); thus there are 7 possible statement numbers that will allow you to win. Option (b) will let you win if the next statement number is 10 through 80 (and MOVE 2 is entered); here there are 8 chances to win. Therefore, option (b) is the better choice since it gives the better chance of winning.

HOW ARROW DYNAMICS FITS INTO THE CLASSROOM

ARROW DYNAMICS teaches students a fundamental learning skill -- to think logically. Logical thinking is indeed a skill that can be applied to any subject area, whether it be language arts, mathematics, science, philosophy, social studies, programming, or any other discipline.

In the area of programming, ARROW DYNAMICS teaches students to communicate with the computer in a language the computer understands. Once students master this skill and understand the importance of sequence and structure, they can apply these skills to the learning of more formal programming languages.

In the area of problem-solving (required in the learning of any subject area), ARROW DYNAMICS lets students practice many strategies, such as:

* problem finding
* information gathering
* flexibility
* looking for a pattern or sequence
* analyzing
* restating the problem
* scanning for clues
* estimating, predicting and projecting
* working backward
* examining assumptions
* identifying multiple solutions

In addition, when students play ARROW DYNAMICS independently, they learn the important skill of thinking on their own. Conversely, when students play ARROW DYNAMICS cooperatively with another player, they learn an equally important skill -- working together.

About the Worksheets

There are two master worksheets provided which may be copied and distributed to your students. The STUDENT RECORD WORKSHEET may be used to record student progress.

The GAME PLAN WORKSHEET will be helpful for students who are first learning the game. Students should first draw on the grid the initial starting conditions of the game: the location and direction of the arrow and the location of prisms or mirrors, if used. Then, they should plot their moves on the worksheet before entering program statements into the computer.

APPLE: WORKING WITH THE COMPUTER

1. Turn on the television or monitor.

2. Insert the diskette into the disk drive with the label facing up and on the right.

3. Close the door to the disk drive.

4. Turn on the Apple II. (The on-off switch is on the back left side of the computer.)

5. You will see a red light on the disk drive turn on. If the disk drive light does not turn off in about 10 seconds, turn the Apple off and make sure your diskette is placed correctly in the disk drive.

6. SUNBURST will appear on the screen followed by the program name.

7. Follow directions given in the program.

Shutting Off the System

1. Remove the diskette from the disk drive and return it to its place of storage.

2. Turn off the Apple.

3. Turn off the television or monitor.

TI99/4A: WORKING WITH THE COMPUTER

1. Turn on the television or monitor.
2. Turn on the computer.
3. Choose the tape you wish to use.
4. Insert the tape in the tape recorder (the label of the program you want should be facing up).
5. Press the "1" key for TI BASIC.
6. Type OLD CSI and press the ENTER key.
7. Follow the directions on the screen.
8. Type RUN and press ENTER.

"WHAT HAPPENS IF...?" -- SUNBURST COURSEWARE AND WARRANTY

1. What happens if a program will not load or run?
   Call us on our toll-free number and we will send you a new tape or diskette.

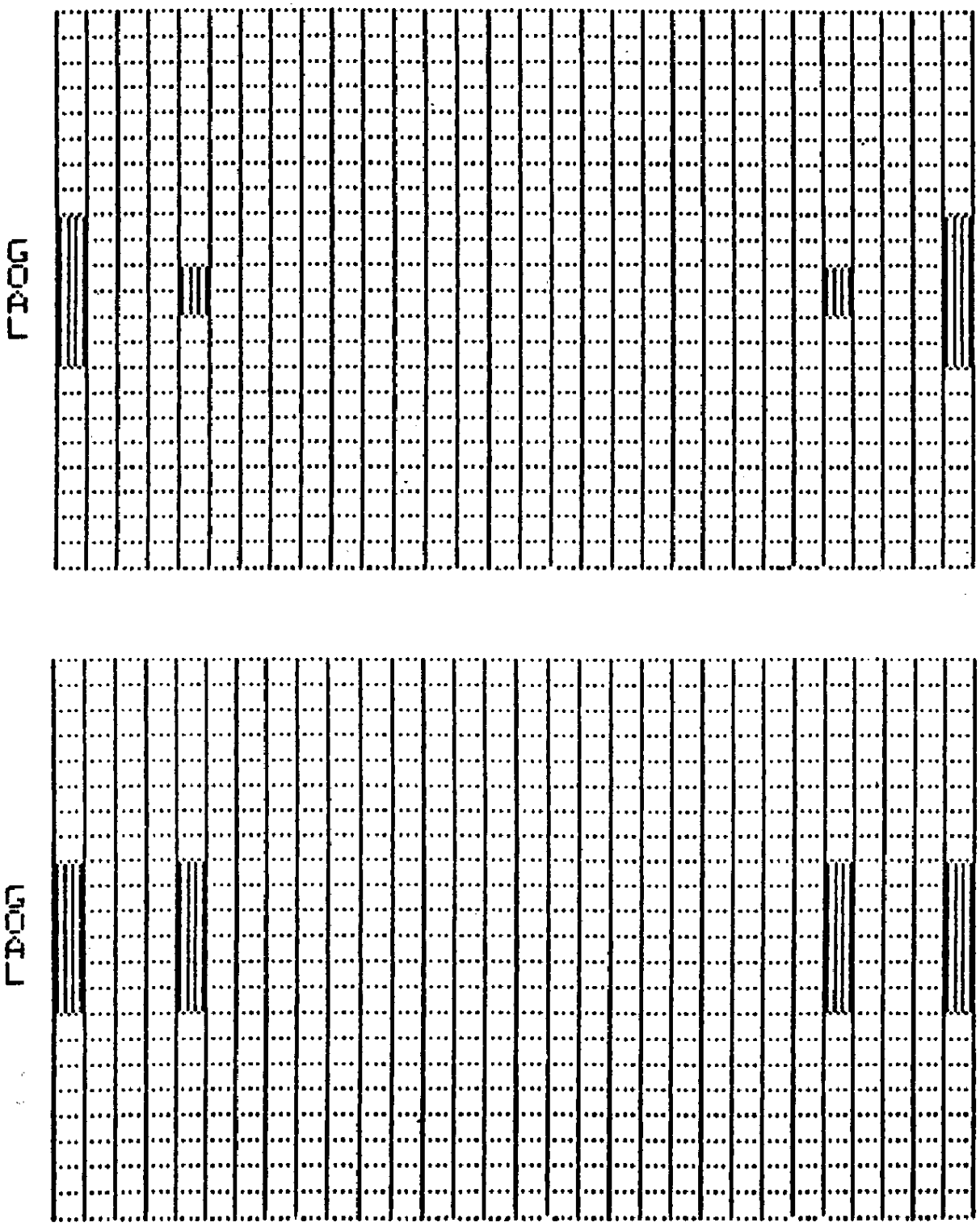
2. What if I find an error in the program?
   We have thoroughly tested the programs that SUNBURST carries so we hope this does not happen. But if you find an error, please note what you did before the error occurred. Also, if a message appears on the screen, please write the message down. Then fill out the evaluation form or call us with the information. We will correct the error and send you a new tape or diskette.

3. What happens if the courseware is accidentally destroyed?
   SUNBURST has a lifetime guarantee on its courseware. Send us the product that was damaged and we will send you a new one.

4. How do I stop a program in the middle to go on to something new?
   The Apple version can be ended at any time by holding the Control button and pressing the E key. To change diskettes, select the End option on the menu and insert a new diskette. On the TI, type FCTN 4 and the program will be interrupted. Then, a new tape may be loaded or ARROW DYNAMICS may be started over again by typing RUN.

5. Can I copy this program?
   The material on the diskette or cassette is copyrighted. You may not copy the courseware.

ARROW DYNAMICS: THE PROGRAMMING GAME
GAME PLAN WORKSHEET



ARROW DYNAMICS: THE PROGRAMMING GAME
STUDENT RECORD WORKSHEET

Student's Name(s) Player 1: _____

Player 2: _____

Class _____ Date _____

1. Game played (circle one): solitaire,
   2-person, cooperative
   2-person, competitive
   puzzle

2. Options chosen (circle appropriate option(s) played):

   small obstacles
   large obstacles
   mirrors
   prisms

3. Statement numbers were (circle one):

   chosen by player
   chosen by computer

4. Statement number range was (circle one):

   10-200
   10-400
   10-600

Final Score:

Player 1 Score = _____ points

Number of turns completed _____

Player 2 Score = _____ points

Number of turns completed _____