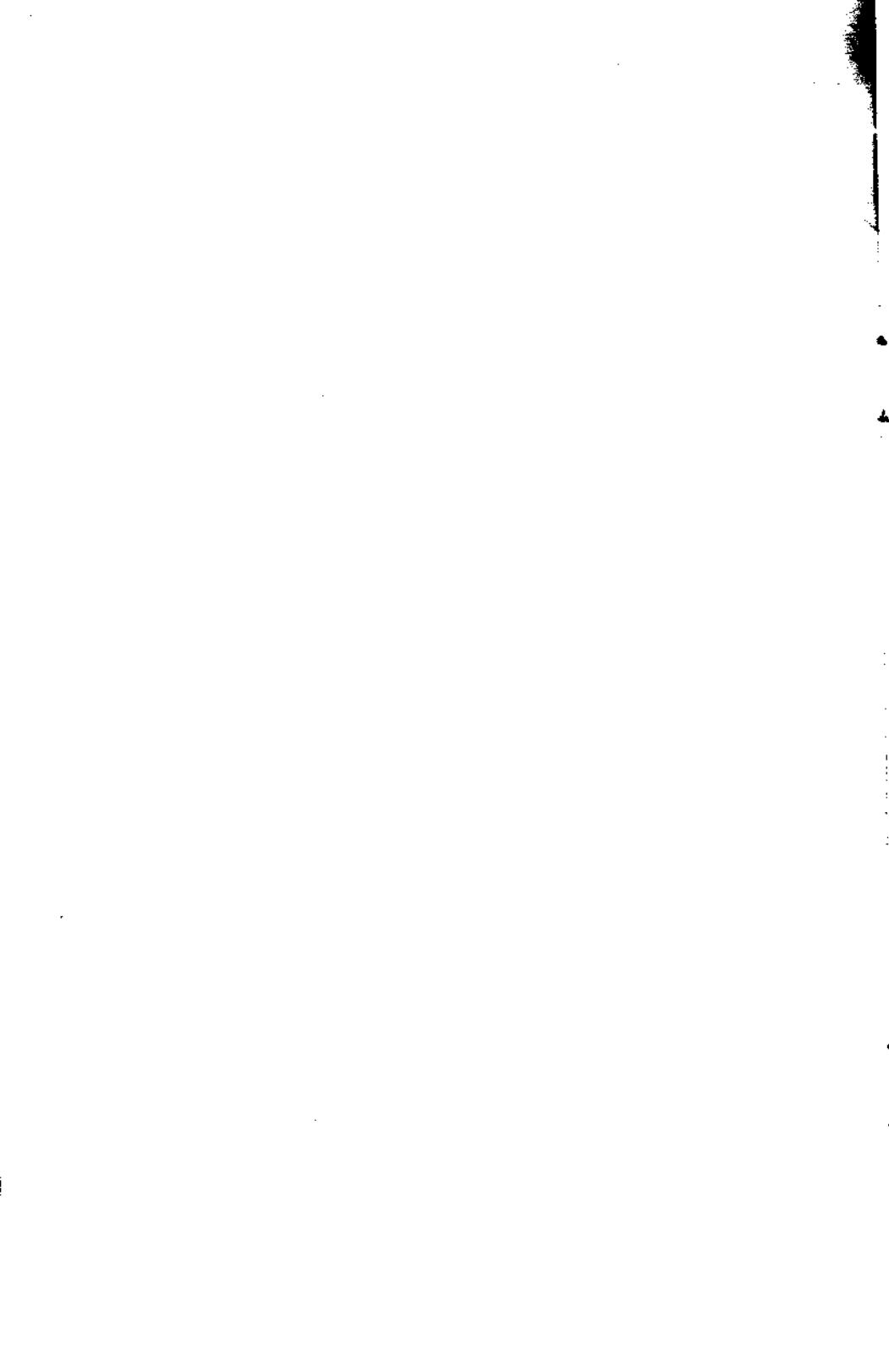# HAYDEN

Tapes

APPLE II

USER'S GUIDE TO

A BEGINNER'S GUIDE TO GAME PROGRAMS

GAME PLAYING WITH BASIC

(Tapes 1, 2, and 3)

(] — Floating point, 16K)

The program listings in the book, GAME PLAYING WITH BASIC, will occa-
sionally differ from those you will see on your video monitor. This is
because some changes were necessary:

1. To fit the general BASIC of the text to your machine's BASIC.
2. To format the programs' "menu style" with consecutive line
   numbering.
3. To add game instructions.
4. To enhance game use (for instance, by giving the user the ability
   to run through a game as many times as he or she wishes).
5. To cut out unnecessary REMs, etc., so that memory will not be
   wastefully consumed.

Compare the cassette programs with those in the book. The changes are
useful for study. They demonstrate some ways in which programs can be
modified to improve formatting and information flow.

The comments that follow on each game or puzzle are meant to be mind
teasers — to point out ways you can alter game programs or learn effec-
tive programming techniques from them.

The combination of the discussion and documentation in Spencer's GAME
PLAYING WITH BASIC and the up and running games on these tapes will give
you the chance to explore your machine's capabilities and test and
enhance your understanding of fundamental program structure.

GAME PLAYING WITH BASIC, Tape 1

1. COIN TOSS, COIN BY COIN
      (Page 31; starts line 200, ends line 380)

   A simple game; program No. 2 shows some elaborations on it.
   1. Change the program so it will sum up the number of tosses and
      total heads and total tails.
   2. Note that our BASIC uses RND (∅) instead of RND (1). Try RND
      (1) and see what happens. Do you know why?
   3. Print out the answers in columns so they don't run off the
      screen or set limits to the number of tosses.

2. COIN TOSS WITH SUMMARY
      (Page 32; starts line 400, ends line 630)

   1. We have added input, so the user controls the number of tosses.
   2. Add toss total to results.
   3. Can you "fix" the coin so one side will come up more than the
      other?

3. ROLLING ONE DIE
      (Page 33; starts line 650, ends line 790)

   A simple, random integer generation program, but very useful for
   many applications.
   1. Add a printout summarizing results.

4. ROLLING TWO DICE
      (Page 33; starts line 800, ends line 970)

   1. Can you work out the probabilities of each number combination
      (2 to 12) — perhaps summarizing the results of any number of
      rolls? This could be done by revising the next program (No. 5,
      SEVEN/ELEVEN).
   2. At least add a summation of results.
   3. Develop a dice game with odds and payoffs.
   4. Roll three or more dice.

5. SEVEN/ELEVEN
      (Page 34; starts line 1000, ends line 1190)

   1. Work up the odds for other numbers or number combinations.

6. MOUSE IN A MAZE
      (Page 35; starts line 1200, ends line 1710)

   1. Change the position of the cheese (can you?).
   2. Change the mouse entry point.
   3. Have mice enter at different points.

7. A RANDOM WALK
      (Page 39; starts line 1730, ends line 2010)

      1. Sam's range is phenomenal. What is the largest number of steps
         possible?
      2. Revise the program to generate x consecutive walks and print the
         totals together.
      3. Alter the size of the bridge and/or Sam's steps.

8. BATTLE OF THE NUMBERS
      (Page 40; starts line 2030, ends line 2660)

      1. See the explanation of computer invincibility on page 45 of
         GAME PLAYING WITH BASIC. Where is this built into the program?

9. DEALING A POKER HAND
      (Page 46; starts line 2680, ends line 3170)

      1. Can this program deal more than one hand without dealing the
         same card twice? If not, can it be amended to do so?
      2. Can the deck be "stacked"?
      3. DIM ERR appears if play Poker then go through index to it again.
         This can be fixed by shifting DIM statement. Can you do this?

10. TIC-TAC-TOE
      (Page 52; starts line 3200, ends line 3850)

      1. Revise the game so the player has first choice.
      2. GAME PLAYING WITH BASIC, page 55, has revision suggestions.

GAME PLAYING WITH BASIC, Tape 2

1. KNIGHT'S TOUR
      (Page 56; starts line 200, ends line 750)

      The knight has a specific pattern for moving in chess.
      1. Can you see where move limits are set in the program?
      2. This program would be more instructive if some summation of
         moves were included after each tour and sequence of tours.
      3. It is possible to complete the tour (but the chances are small
         using random moves). Can a controlled program be built from
         this to show what any full tour would be starting from any
         position?

2. GUESS THE NUMBER
      (Page 62; starts line 800, ends line 1140)

      1. The game can be enhanced by adding words of encouragement to
         the losing player.

2. Note the manner in which the number choices are compared to see which is closest to the computer's choice. This is a handy technique for many games and other applications.
3. Change the program so the winner is the player who chooses numbers farthest away from the computer's numbers (avoidance).

3. PRIME NUMBERS 1
(Page 64; starts line 1200, ends line 1540)

Prime numbers are useful in analyzing problems about divisibility.
1. If you carry a high dimension array, see what is the largest prime number you can generate using the sieve of Eratosthenes.
2. Can you set this program so columns are in exact alignment?
3. The REDIM'D ARRAY statement is a function of the menu format. Can you redo the menu to fix this?

4. PRIME NUMBERS 2
(Page 66; starts line 1600, ends line 1800)

1. Can you "translate" this program and determine how odd number primes are identified by it?
2. See 3 above for Prime Numbers 1.

5. CHINESE REMAINDER THEOREM
(Page 68; starts line 1850, ends line 2060)

1. Study your primes and alter this "trick" to increase or decrease the victim's range of choice.
2. Does your BASIC have a modulus function (see page 69)? If so, revise this program to use it.

6. PERFECT NUMBERS
(Page 69; starts line 2100, ends line 2290)

1. The real test — can you modify this program to generate a third perfect number? How about a fourth and fifth?

7. FIBONACCI NUMBERS
(Page 71; starts line 2320, ends line 2520)

1. Just misses fitting on the screen. What ways can it be changed to do so?
2. The printout for this can be varied — one column, many columns, slow roll, spacing with pauses, etc.
3. How large a Fibonacci can you produce?

8. SQUARE NUMBERS
(Page 76; starts line 2550, ends line 2720)

1. This program does not produce all pairs of square numbers less than 100. Can you determine why? (The printout on page 76 shows more.)

2. Can you change it to produce them all? Can you increase the range?
3. How does increasing or decreasing the range effect speed of execution?

9. ARMSTRONG NUMBERS
    (Page 77; starts line 2750, ends line 2920)

    1. Can you write a similar program to find Armstrong numbers for any n digits?

GAME PLAYING WITH BASIC, Tape 3

1. SLOT MACHINE
    (Page 81; starts line 300, ends line 890)

    This program doesn't build in house odds, so it is fairly honest. We have added a counter as a time delay before each bar, cherry, orange printout, so the answers don't come up too fast. It is in line 475 and has a value of 1000. You may change this to alter the pause, however.
    1. Obvious changes are in language and spacing. To personalize the game, you can have the player enter his/her name and be identified by it occasionally. The displays could be centered, with the screen cleared before they appear.
    2. Symbols can be arranged in PRINT statements so that the bars, cherries, oranges are shown rather than described. Can you do it?
    3. This program does not show zeroes in the decimal places. Can you revise it to do so?
    4. Some odds could be built in to vary the payoff (for example, $1.00 for oranges, $5.00 for cherries, and $10.00 for bars).

2. BLACKJACK
    (Page 82; starts line 910, ends line 2250)

    This is a simple version of a popular game. Study how the deck is set up and you will understand the base for many different card games. Possible alterations beyond this are legion:
    1. Dealer does not cut off cards if player continues to draw after 21 points are in hand. Change program to stop this.
    2. House wins when player and dealer tie. Change as you see fit (player wins, nobody wins).
    3. Program states "House stands pat," when house has over 21 points. Add loop for House and player to indicate "busted."
    4. Revise program for wagering. Have bets placed after each card.
    5. Identify cards to player as "first card," "second card," etc.
    6. Program does not show suits. Can it?
    7. This game will trigger a "REDIM'D ARRAY ERROR" after a number of repeats through the index. There are a number of ways to fix this (perhaps by putting DIM statement in front of index).

3. ROLL THE DICE
   (Page 92; starts line 3000, ends line 3410)

   Obviously, the first thing you can change is the rules. Add more
   chances to win or lose, depending on your fancy (lines 3150-3180).
   1. What happens if you enter $\emptyset$ rolls? Why? Can you change it?
   2. Dice rolling routine can go into many games as it is.
   3. Can you load the dice (rolling routine begins in line 3340)?
   4. Again, there's a pause before each printout, which can be fixed.
   5. Some folks may want totals so they don't have to work at addi-
      tion (that's what computers are for!). Can you add them to the
      display? You might also try a summary of the number of times
      you have won and lost on any series of rolls.

4. TOWER OF HANOI
   (Page 102; starts line 4500, ends line 4700)

   1. The number of moves required is $2^n - 1$, so the number gets large
      quickly. At what point do you deliver numbers in scientific
      notation? Can you avoid this?
   2. This is a simple counting program. Try the disc-moving program
      suggested on page 103.


5. 15 PUZZLE
   (Page 106; starts line 4750, ends line 5870)

   1. This program sums the counts for each position in the puzzle.
      If they are odd, the arrangement is impossible. If they are
      even, the arrangement is possible.
   2. Do the answers change if the number of squares is increased or
      decreased (keeping the configuration as an even-celled square)?
   3. Can you see in the listing where the counting is done?
   4. Why does the program seek the specific "16" positions asked for
      in lines 5530-5590?

6. BURIED TREASURE
   (Page 113; starts line 6040, ends line 6570)

   1. It is best to have a grid at hand so you can plot your guesses.
      You can, of course, alter the number of tries offered.
   2. Even when you find the treasure, the program won't reveal all
      the squares it occupied. Can you make it do so?
   3. Program states "1 tries left" when $S = 1$. Change it so it will
      read "You have 1 try left."
   4. Bury two treasures. Add traps (fatal) at some locations,
      which will end game and protect the treasure.
   5. Add a grid to the screen.
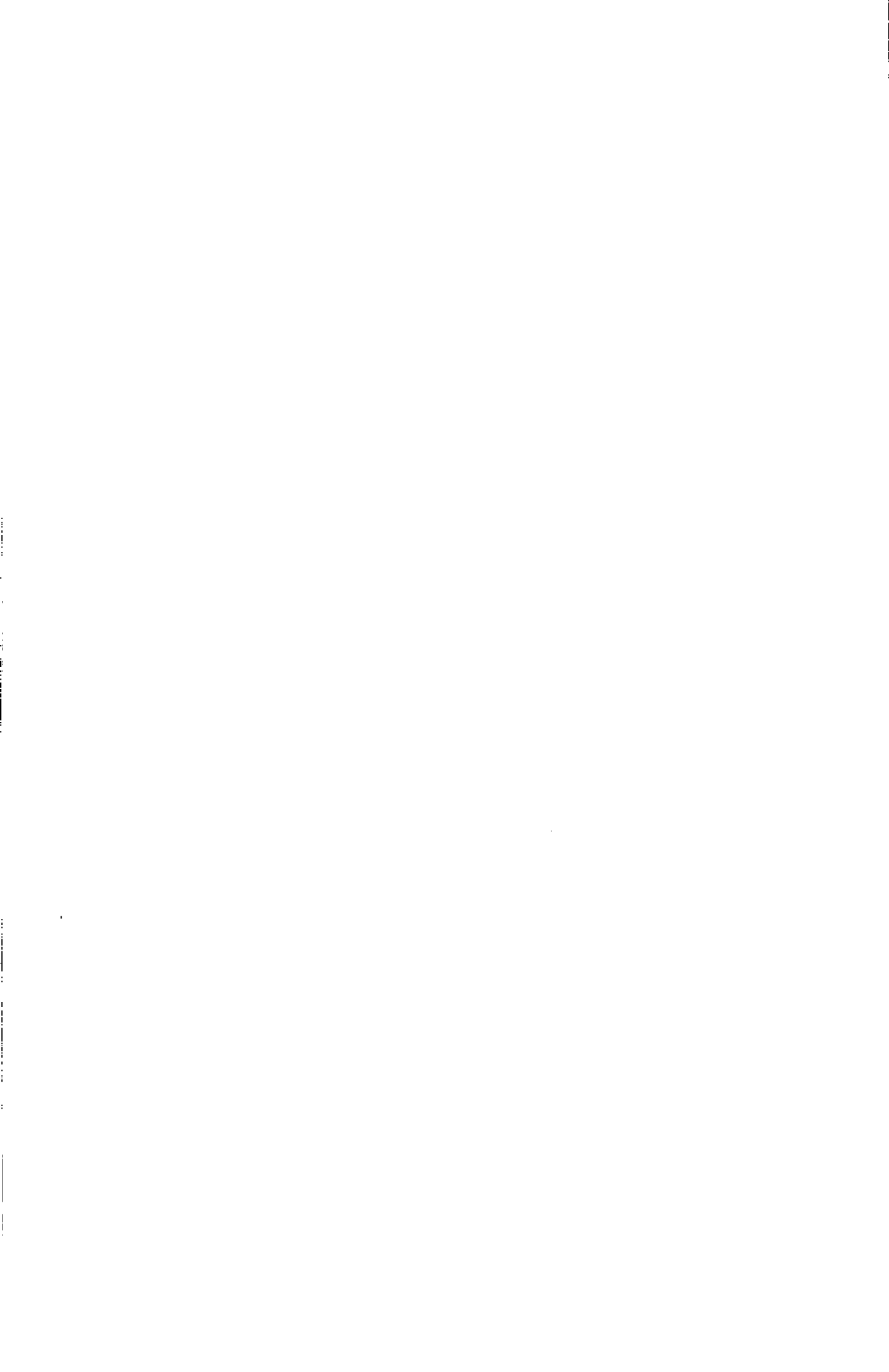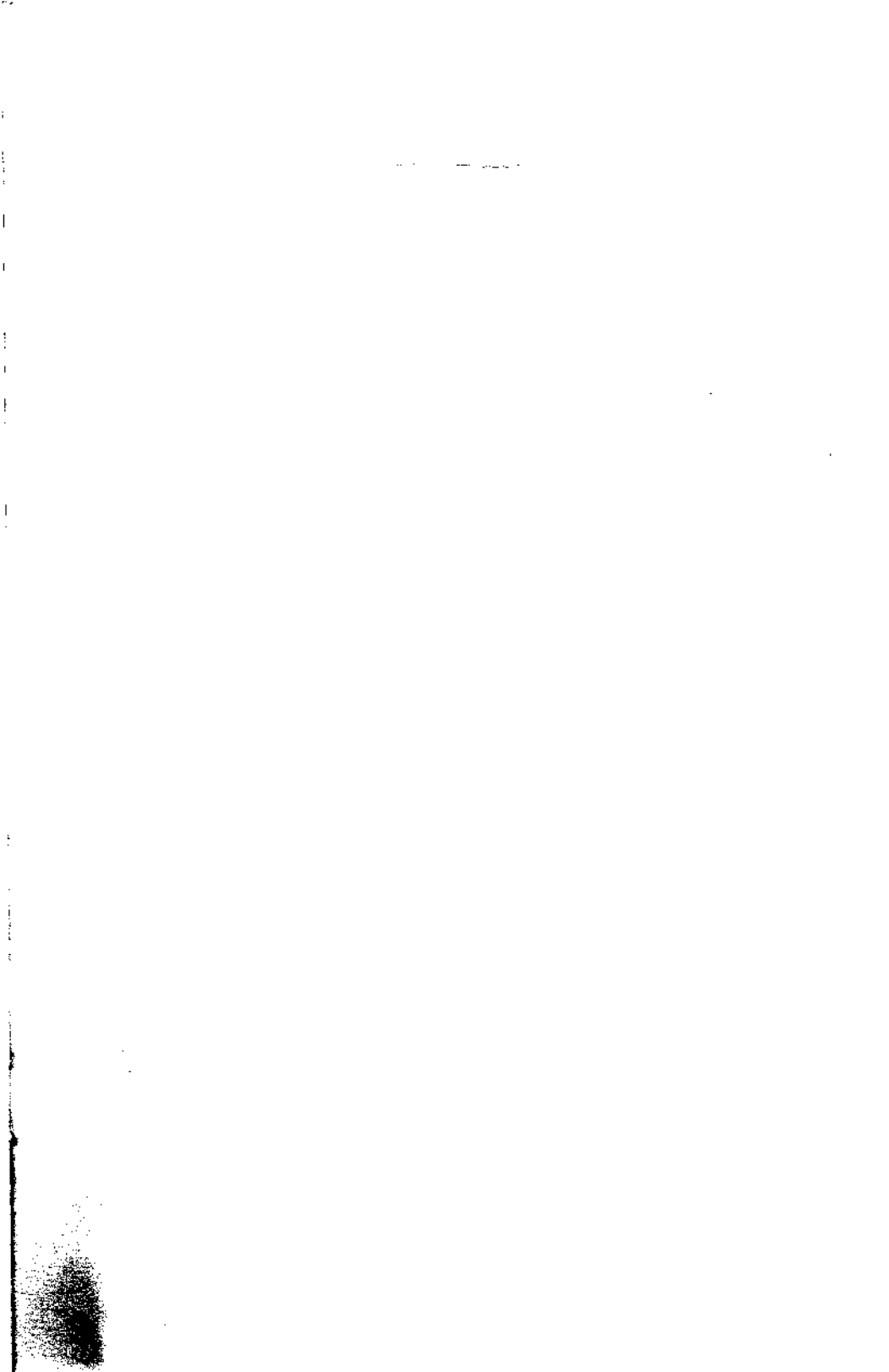
Recreation, Education, and Skills Programs from
GAME PLAYING WITH BASIC
Donald D. Spencer

Available on three tapes are 27 ready-to-run programs taken from GAME PLAYING WITH BASIC — a popular guide for learning computerized game-playing and for sharpening your programming skills. All 27 programs are fully documented and explained in the guide. The user of both the tapes and the guide will learn how to combine ideas for game-playing and computer programming to do some interesting and fun things.

GAME PLAYING WITH BASIC, Tape 1
The 10 programs on Tape 1 are: **Mouse in a Maze; Battle of the Numbers; Dealing a Poker Hand; Tic-Tac-Toe; Coin Toss; Rolling One Die; Rolling Dice; Seven/Eleven; and A Random Walk.** #00201 (PET); #00202 (TRS-80 Level I); #00203 (TRS-80 Level II); #00204 (Apple II).

GAME PLAYING WITH BASIC, Tape 2
The 9 programs on Tape 2 are number games: **Guess the Number; Prime Numbers; Chinese Remainder Theorem; Perfect Numbers; Fibonacci Numbers; Square Numbers; Armstrong Numbers; and Knight's Tour** (simulates a classical chessboard problem). #00301 (PET); #00303 (TRS-80 Level II); #00304 (Apple II).

For TRS-80 Level I users: Tape 2 version for your machine consists of these 8 programs: **Guess the Number; Prime Numbers; Chinese Remainder Theorem; Perfect Numbers; Fibonacci Numbers; Armstrong Numbers; Slots; Roll the Dice.** #00302 (TRS-80 Level I).

GAME PLAYING WITH BASIC, Tape 3
The 8 programs on Tape 3 are divided into gambling games (**Slot Machines, Blackjack, Roll the Dice**); puzzles (**Tower of Hanoi, The 15 Puzzle,** and **Buried Treasure**); and magic squares (**Odd-Cell Magic Square, 4 x 4 Magic Square**). #00401 (PET); #00403 (TRS-80 Level II); #00404 (Apple II).

Other Hayden Computer Program Tapes that are compatible with the KIM, PET, TRS-80 Level I, TRS-80 Level II, Apple II, and Exidy's Sorcerer machines . . .

HOW TO BUILD A
  COMPUTER-CONTROLLED
  ROBOT
SARGON: A COMPUTER
  CHESS PROGRAM

GENERAL MATH
COMPLEX AND MATRIX MATH
INTRODUCTORY ENGINEERING MATH
THE FIRST BOOK OF KIM

HAYDEN BOOK COMPANY, INC.
Rochelle Park, New Jersey

If special loading instructions are necessary, see back of insert in cassette container.