

AVON



CAMELOT

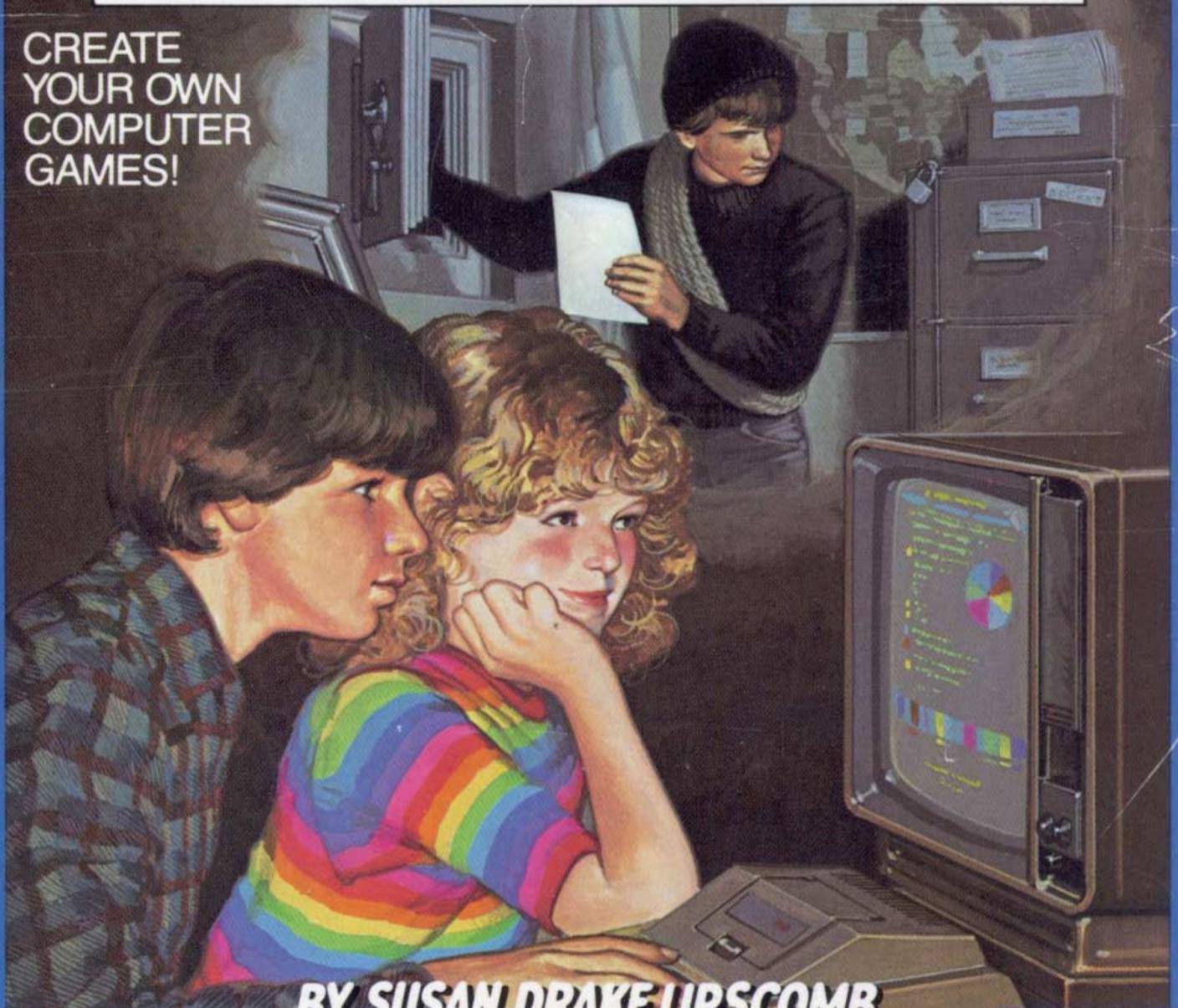
NEW! From the Authors of
BASIC FUN WITH GRAPHICS

87486
\$2.95

BASIC FUN

WITH **ADVENTURE**
GAMES

CREATE
YOUR OWN
COMPUTER
GAMES!



BY SUSAN DRAKE LIPSCOMB
AND MARGARET ANN ZUANICH

NEWSOM

BASIC FUN
WITH ADVENTURE
GAMES

Other Avon Camelot Books by
Margaret Ann Zuanich and
Susan Drake Lipscomb

BASIC FUN

BASIC BEGINNINGS

BASIC FUN WITH GRAPHICS: THE APPLE® COMPUTER WAY
BASIC FUN WITH GRAPHICS: THE ATARI® COMPUTER WAY
BASIC FUN WITH GRAPHICS: THE IBM/PC® COMPUTER WAY

MARGARET ANN ZUANICH and **SUSAN DRAKE LIPSCOMB** have a unique combination of skills that contributed to the creation of **BASIC FUN WITH GRAPHICS**. Margaret Zuanich's experience in the computer field has included everything from programming to management consulting. She earned her Master's Degree in Business and is now involved in computer systems training. Susan Lipscomb holds a Master's Degree in Education and has spent fourteen years in the area of language and learning disabilities. They both live in Palo Alto, California.

Avon Books are available at special quantity discounts for bulk purchases for sales promotions, premiums, fund raising or educational use. Special books, or book excerpts, can also be created to fit specific needs.

For details write or telephone the office of the Director of Special Markets, Avon Books, Dept. FP, 1790 Broadway, New York, New York 10019, 212-399-1357.

BASIC FUN
WITH ADVENTURE
GAMES

SUSAN DRAKE LIPSCOMB
AND MARGARET ANN ZUANICH

Illustrated by Ted Enik

AN AVON



CAMELOT BOOK

5th grade readability has been determined by using the Fry Readability Scale.

BASIC FUN WITH ADVENTURE GAMES is an original publication of Avon Books. This work has never before appeared in book form.

AVON BOOKS
A division of
The Hearst Corporation
1790 Broadway
New York, New York 10019

Copyright © 1984 by Margaret Ann Zuanich and
Susan Drake Lipscomb
Published by arrangement with the authors
Library of Congress Catalog Card Number: 84-45138
ISBN: 0-380-87486-4

All rights reserved, which includes the right to reproduce this book or portions thereof in any form whatsoever except as provided by the U.S. Copyright Law. For information address Avon Books.

First Camelot Printing, July, 1984

CAMELOT TRADEMARK REG. U.S. PAT. OFF. AND IN OTHER COUNTRIES, MARCA REGISTRADA, HECHO EN U.S.A.

Printed in the U.S.A.

OP 10 9 8 7 6 5 4 3 2 1

**This book is dedicated to Slaton, Bill, Mark,
Jeff and John . . . whose interests and creativ-
ity have fueled our own imaginations.**

TABLE OF CONTENTS

Introduction	9	
How to Use <i>BASIC Fun With Adventure Games</i>		11
1 CIA: A Text Adventure Game	13	
2 Game Enhancements: Make the Program Look Professional	20	
3 The Pieces of the Puzzle: The Parts of a Text Adventure	30	
4 The BASIC Program: Subroutines for Text Adventures	38	
5 Write Your Own Game: Coding BASIC Statements	47	
6 Game Testing: Get Rid of Program Bugs		56
7 Game Documentation: The Final Polish		63
Appendix A Description of the General Purpose Subroutines in CIA	67	
Appendix B Dictionary of Variables for CIA		71
Appendix C Program Listing for CIA		73
Appendix D Sample Run for CIA		95

A disk containing the adventure program used in this book is available for Apple and IBM microcomputer users from the authors:

Computer Alternatives
4025 Laguna Avenue
Palo Alto, CA 94306

The price is \$10.00 (California residents add 6½% sales tax). Specify IBM or Apple version with your order.

INTRODUCTION

You are a CIA agent. Your job is to prove that the Russian Ambassador to the U.S. is a spy. You have two hours to find the evidence in his apartment. Good luck! Here is an adventure game that challenges you to overcome an attack dog, avoid deadly hazards, and find a hidden room while you collect evidence needed to solve the mystery.

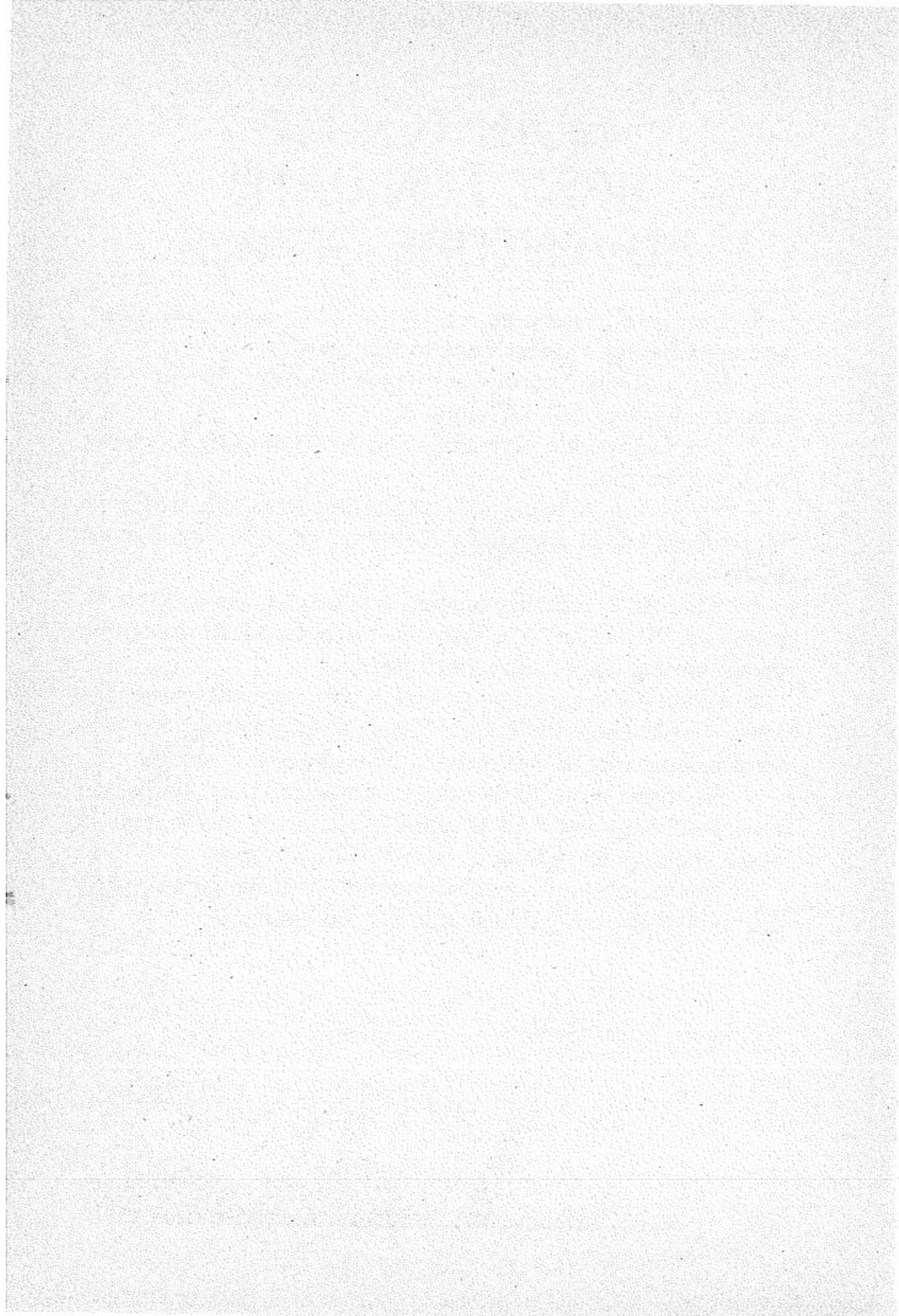
BASIC Fun With Adventure Games contains more than an exciting adventure game. It includes a step-by-step process for developing and writing a game of your own. Using the simple instructions, you can add enhancements to this game, or use it as a model for your own adventure.

The instructions lead you through the process of developing a plot, defining the solution, and determining the objects, obstacles, and hazards of your game. You are then shown how to translate your ideas into a BASIC program using the CIA program as a model.

BASIC Fun With Adventure Games doesn't write the program for you, but provides you with the information needed to develop your own program. The instructions assume that the reader has some knowledge and experience with the programming language BASIC and some familiarity with adventure games. The complete CIA program is given for the IBM-PC and Apple computers. It can also be used on other microcomputers with minor changes in some of the BASIC programming statements unique to individual computers. These are left to the reader to modify with the help of his computer's reference manual.

HOW TO USE BASIC FUN WITH ADVENTURE GAMES

1. Get the CIA program up and running using the instructions in Chapter 1 under How to Put Up CIA.
2. Run and enjoy your game. Try to solve the mystery using only the information in Chapter 1.
3. See if some of your friends want to try the game. Try not to give them hints.
4. If you want to learn more about how this game works, or more about adventure games in general, read through the rest of the book.
5. Chapter 2 contains suggestions for changes to the CIA program. This is an easy first step toward understanding the programming used in adventure games.
6. When you are ready to develop your own adventure, follow the instructions in Chapter 3, which show step by step how to develop a plot and add clues, hazards, and obstacles.
7. Chapters 4 and 5 describe how the program works and how to change it for your game. Follow the instructions in these chapters to write your own BASIC program.
8. Chapters 6 and 7 give information on testing, improving, and polishing your game so it looks professional.



1 CIA

A Text Adventure Game



THE PLOT

You are a CIA agent. The department has received an anonymous tip that the Russian Ambassador to the United States, Vladimir Griminski, is passing classified U.S. Defense Department information to the KGB. Your job is to obtain evidence implicating Ambassador Griminski as receiving classified information so that the United States can have him expelled.

The Ambassador is attending a meeting at the Russian Embassy tonight. You have two hours to search his apartment before he is expected to return. Your adventure starts in your Headquarters office. You can then proceed to Griminski's apartment. You must be extremely cautious, as some clues may be booby-trapped. When you think you have sufficient evidence to incriminate the Ambassador, call Headquarters. Your Control will ask you to list your evidence and tell you if it is sufficient.

Can you accomplish your mission before Ambassador Griminski returns to his apartment? Good luck!

HOW TO PUT UP CIA

1. Type in the CIA program, line by line, exactly as it appears in Appendix C. Enter the entire line, starting with a line number, before hitting RETURN. When entering PRINT and DATA statements use the same spacing shown in the listing.

2. Once you have the program entered into your computer, type RUN and press RETURN. The program should display the opening lines of the game at CIA Headquarters and a prompt "NOW WHAT?", waiting for your instructions.

3. Refer to Appendix D for a sample run of CIA. Yours should look the same.

4. To check programming statements for bugs, type LIST and the lines you want to review, and press RETURN. The computer will list the program lines you specify for you to check and edit as needed.

5. When you're ready to save, use the SAVE command. To retrieve the program from the disk use the LOAD command.

6. If you have difficulties making the program work, check your program, line by line, particularly for typographical errors in spelling or punctuation. CIA was written, run, and playtested on the Apple IIe, the Apple II Plus, and the IBM-PC using DOS 1.0, 1.1 and 2.0.

7. All responses must be in upper case for the program to recognize input. On the IBM and the Apple IIe, press the CAPS LOCK key at the beginning of your game play.

RUNNING CIA

1. Follow How to Put Up CIA to enter the game into your computer.
2. To run CIA, type RUN and the program name you used to SAVE the program onto the disk.
3. The program will display a description of your office at Headquarters and prompt "NOW WHAT?". You are ready to play the game.
4. All response inputs to CIA must be in upper case for the program to recognize them.
5. The program understands two kinds of input: (1) two-word sentences of a verb and either a noun or a direction, separated by a blank space, such as "TAKE PISTOL" or "GO NORTH" and (2) one-word commands such as "QUIT" or "HELP".

PLAYING CIA

MOVING: To move from one location to another, type GO and the direction in which you wish to move. In CIA there are four directions: NORTH, SOUTH, EAST and WEST. When you are in a location for the first time, examine it carefully. LOOK at different objects in the room description for more information.

GATHERING CLUES: You are permitted to carry up to six items at one time. In this adventure you need to TAKE or GET some objects that will count as incriminating evidence against the Ambassador. You also need some to overcome a few hazards. The game may allow you to TAKE or GET certain extraneous items, but they will only take up valuable space in your inventory. It is wise to DROP certain objects after they have no apparent continuing value. In this particular adventure, TAKING or DROPPING objects will not usually change the description of the room, so you need to remember where you have left something, in case you should have future use of it.

ENDING A SUCCESSFUL MISSION: When you think you have gathered sufficient evidence, CALL HEAD-QUARTERS. Your Control will answer the phone and ask you to list the objects you have obtained that will incriminate the Ambassador. To end the listing of items, hit the RETURN key. If you are correct, your mission will be accomplished. If not, you will be instructed to continue your investigation.

VERB SUMMARY: The program understands 25 simple verbs and has a vocabulary of over 75 nouns. The verbs allow the player to perform some action such as "GO",

“DROP”, or “TAKE”. The nouns refer to objects found in the rooms of the game. The verbs are:

CALL	EAT	LOOK	READ	TALK
CHEW	GET	OFF	RUN	UNLOCK
CRAWL	GO	ON	SHOOT	UNSCREW
DRINK	LIFT	OPEN	SPRAY	UNWRAP
DROP	LOAD	PUSH	TAKE	WALK

COMMAND SUMMARY: The commands are used to perform an action independent of an object.

HELP	Gives player one clue for each location of the game.
INVENTORY	Lists all the items the player is currently carrying.
LOOK	Gives the room description of the player's location.
QUIT	Allows player to quit playing the game and gives total score in this play of the game.
RESTART	Allows player to end current game and return to beginning of the adventure to start again.
SCORE	Gives total number of evidence points accumulated in current game.
TIME	Gives total time elapsed in current game.

HINTS FOR PLAYING CIA

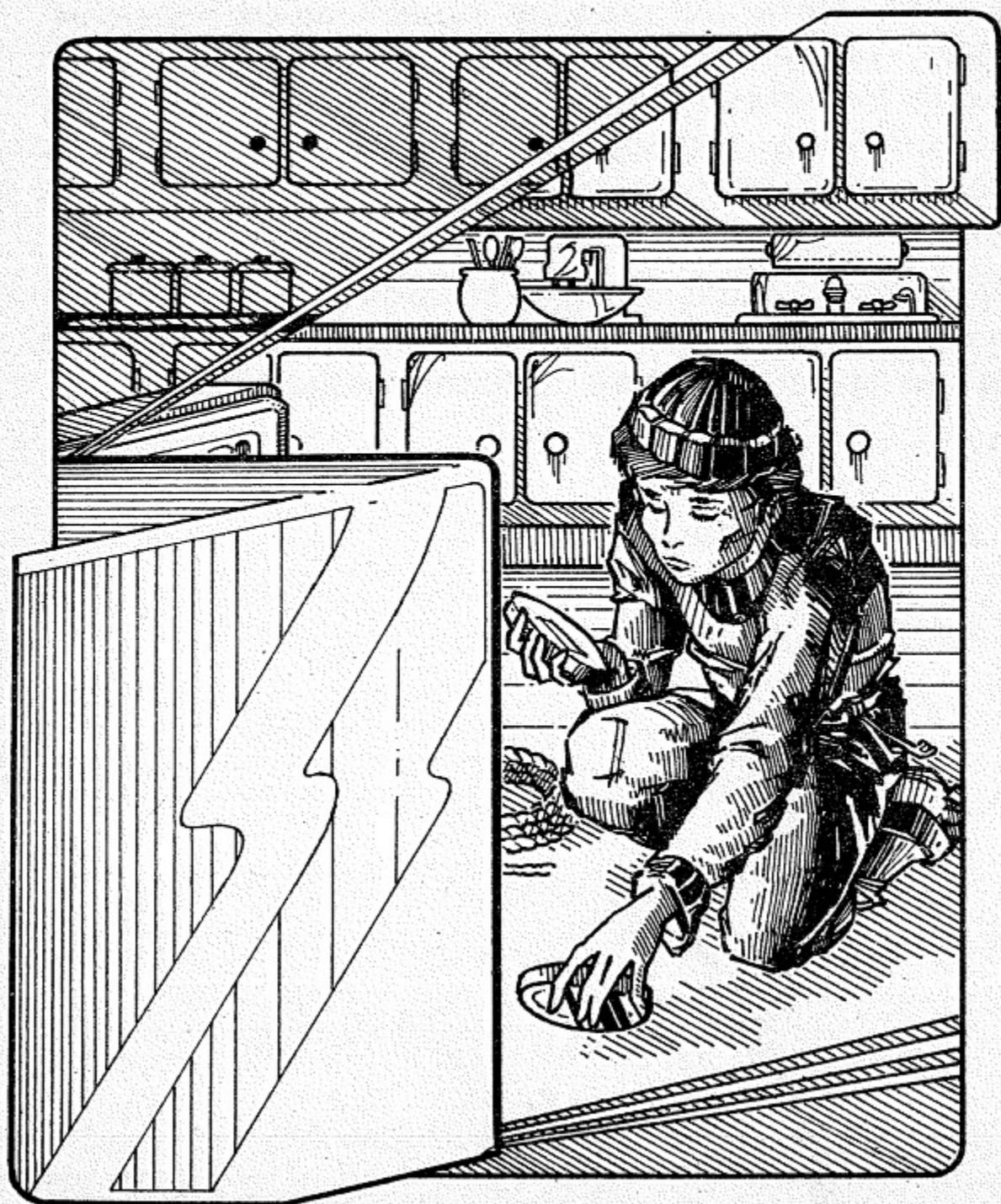
THE TIME ELEMENT: In this text adventure, you have two hours in which to complete your mission. Most actions you perform will take about three minutes of investigative time.

STRATEGY: Start out by exploring the places to which you can move and read the location descriptions carefully. You may find it helpful to draw a map as you go along. In doing so you will become familiar with the locations, hazards, and some of the objects available. Once you have an overview of the game you will be ready to start playing to win.

HAZARDS: There are some hazards scattered throughout the game. If you plan ahead, you should have the necessary items available to surmount the obstacles.

2 GAME ENHANCEMENTS

Make the Program Look
Professional



CHANGING CIA

Once you've played CIA and hopefully solved it, you're ready to learn more about how the CIA program works. A good way to start is to make changes to the program which improve the game and make it more challenging and exciting. This chapter contains some changes for you to try. Simply enter the programming statements as shown in each section. Remember to test each change thoroughly before adding the next one. After you have our changes working, try making some of your own.

ADD OBSTACLES

Add other obstacles to CIA. For instance, make the safe in the bedroom blow up if the player attempts to open it without the right combination.

The dialogue is:

```
NOW WHAT ? OPEN SAFE
```

```
COMBINATION 10-5-9
```

```
YOU'VE HAD IT!!...THE SAFE IS BOOBY  
TRAPPED AND EXPLODES!!  
END OF GAME
```

The program changes in the OPEN verb routine are:

```
2445 PRINT  
2446 PRINT "YOU'VE HAD IT!!...TH  
      E SAFE IS BOOBY      TRAPPED  
      AND EXPLODES!!"  
2447 PRINT "END OF GAME": END
```

ADD DOCUMENTATION

Make the program self-documenting. Put the essential information from the written documentation into PRINT statements that are executed when the program starts up.

The program prints:

```
WELCOME TO CIA
```

```
YOU ARE A CIA AGENT. THE DEPARTMENT HAS  
JUST RECEIVED A TIP THAT THE RUSSIAN  
AMBASSADOR. VLADIMIR GRIMINSKI. IS  
PASSING CLASSIFIED INFORMATION TO THE  
KGB.
```

```
YOU HAVE 2 HOURS WHILE THE AMBASSADOR IS  
GONE TO COLLECT EVIDENCE. YOU START IN  
YOUR OFFICE AND THEN PROCEED TO HIS  
APARTMENT. GOOD LUCK!
```

```
HIT RETURN TO START
```

The program statements are:

1. Add a call to the subroutine in the main program.

```
206 GOSUB 1300
```

2. Add the subroutine to print the documentation.

```
1300 HOME  
1301 PRINT "          WELCOME TO  
CIA": PRINT  
1302 PRINT "YOU ARE A CIA AGENT.  
THE DEPARTMENT HAS JUST REC  
EIVED A TIP THAT THE RUSSIAN  
AMBASSADOR. VLADIMIR GRI  
MINSKI. IS PASSING CLAS  
SIFIED INFORMATION TO THE  
KGB.": PRINT
```

```
1303 PRINT "YOU HAVE 2 HOURS WHI  
LE THE AMBASSADOR ISGONE TO  
COLLECT EVIDENCE. YOU START  
IN YOUR OFFICE AND THEN PRO  
CEED TO HIS APARTMENT.  
GOOD LUCK!": PRINT  
1304 INPUT "HIT RETURN TO START"  
:A$: RETURN
```

IBM users: Replace line 1300 with:

1300 CLS: KEY OFF

ADD COMMANDS

Add a command that prints a list of the verbs available to the player.

The dialogue is:

```
NOW WHAT ? VERBS
```

```
I CAN UNDERSTAND THE FOLLOWING VERBS:
```

```
LOOK  
GET  
TAKE  
GO  
CRAWL  
WALK  
OPEN  
READ  
DROP  
CALL  
UNSCREW  
SPRAY  
PUSH  
LOAD  
RUN
```

DRINK
EAT
CHEW
UNWRAP
TALK
SHOOT
UNLOCK
ON
OFF

NOW WHAT ?

The programming statements are:

1. Increase the number of commands in the DATA statement on line 5099.

```
5099 DATA 8
```

2. Add the command keyword, "VERBS" to the DATA statement on line 6021.

```
6021 DATA "HELP","QUIT","INVENTORY","LOOK","TIME"  
      ,"SCORE","RESTART","VERBS"
```

3. Add VERBS to the command routine by adding line number 1200 to the multiple GOTO on line 925.

```
925 ON C GOTO 930,934,940,1000,1030,1060,1100,1200
```

4. Add the VERBS routine starting at line 1200.

```
1200 PRINT "I CAN UNDERSTAND THE  
      FOLLOWING VERBS:"  
1205 FOR I = 1 TO NV  
1210 PRINT VA$(I)  
1215 NEXT  
1220 RETURN
```

ADD OBJECTS

Add an object and some additional clues to a room. We've put a freezer in the Kitchen, Room 8, and added some objects inside which give the player additional information.

The dialogue is:

NOW WHAT ? GO NORTH

YOU ARE IN THE APARTMENT KITCHEN WHICH SHIMMERS WITH POLISHED CHROME APPLIANCES AND BUTCHER BLOCK COUNTERS. A LONG CABINET ABOVE THE STAINLESS STEEL SINKS IS CLOSED. A FREEZER STANDS IN THE CORNER.

NOW WHAT ? OPEN FREEZER

OPENED

NOW WHAT ? LOOK FREEZER

THIS IS A SMALL WHITE FREEZER. INSIDE ARE CONTAINERS OF FROZEN YOGURT, CAVIAR AND FROZEN HERRING. A BOX SITTING ON THE RACK ON THE DOOR IS LABELED 'FILM'.

NOW WHAT ?

The programming statements are:

1. Change the description for Room 8 in DATA statement 6008.

```

6008 DATA "YOU ARE IN THE APARTM
      ENT KITCHEN WHICH SHIMMERS
      WITH POLISHED CHROME
      APPLIANCES AND BUTCHER BL
      OCK COUNTERS. A LONG CABINE
      T ABOVE THE STAINLESS STEELS
      INKS IS CLOSED. A FREEZER S
      TANDS IN THE CORNER.",0,0,
      7,0

```

2. Increase the number of objects in the DATA statement on line 6000.

```

6000 DATA 60

```

3. Add the freezer to the object DATA statements.

```

7060 DATA "FREEZER","THIS IS A SMALL WHITE FREEZER
      .",8,0,0,5

```

4. Change the OPEN Verb Routine to add the freezer and reveal the contents.

```

2415 IF N = 60 THEN PRINT "OPEN
      ED":OD$(60) = OD$(60) + "
      INSIDE ARE CONTAINERS OF FR
      OZEN YOGURT, CAVIAR AND FROZ
      EN HERRING. A BOX SITTING O
      N THE RACK ON THE DOOR IS L
      ABELED 'FILM'.": RETURN

```

ADD COLOR

If you have an IBM-PC, you might want to add color to the screen display.

The programming statements are:

```

501 SCREEN 1: COLOR 9.15

```

ADD SCREEN DISPLAY

Add a special screen display for the start of the game—like a Title Page—and one for the conclusion of the adventure.

The display for the Title Page could look like this:

```
*****      *****      ***
*****      *****      *****
**           **           *** **
**           **           **  **
**           **           **  **
**           **           *****
**           **           *****
**           **           **  **
**           **           **  **
**           **           **  **
*****      *****      **  **
*****      *****      **  **
```

The programming statements are:

```
1400 HOME
1402 PRINT : PRINT : PRINT : PRINT
1404 PRINT TAB( 8) "*****      *****      ***"
1406 PRINT TAB( 8) "*****      *****      *****"
1408 PRINT TAB( 8) "**           **           *** **"
1410 PRINT TAB( 8) "**           **           **  **"
1412 PRINT TAB( 8) "**           **           **  **"
1414 PRINT TAB( 8) "**           **           *****"
1416 PRINT TAB( 8) "**           **           *****"
1418 PRINT TAB( 8) "**           **           **  **"
1420 PRINT TAB( 8) "**           **           **  **"
1422 PRINT TAB( 8) "**           **           **  **"
1424 PRINT TAB( 8) "*****      *****      **  **"
1426 PRINT TAB( 8) "*****      *****      **  **"
1430 FOR I = 1 TO 5000: NEXT I
1435 RETURN
```

IBM users: replace line 1400 with:

```
1400 CLS: KEY OFF
```

ADD A SLEUTH RATING

Print a sleuth rating as part of the player's score by changing the score routine. The categories could be:

SCORE	RATING
0	AMATEUR SLEUTH — GO BACK FOR FIELD TRAINING
10	INTERMEDIATE SLEUTH — POUND THE BEAT SOME MORE
20	ADVANCED SLEUTH — YOU STILL NEED AN ASSISTANT
30	EXPERT OPERATIVE — YOU CAN HANDLE ANY MISSION ALONE
40	WORLD RENOWNED OPERATIVE — YOU WILL BE ELEVATED TO CONTROL

The programming statements in the SCORE Command Routine are:

```
1073 PRINT "YOUR CURRENT SCORE I  
S: "S  
1074 PRINT "YOUR RANK IS": PRINT  
  
1075 S = INT (S / 10) + 1  
1080 ON S GOTO 1085,1087,1089,10  
91,1093  
1082 PRINT "INVALID SCORE S=":S  
1085 PRINT "AMATEUR SLEUTH - GO  
BACK FOR FIELD TRAINING  
": RETURN
```

```
1087 PRINT "INTERMEDIATE SLEUTH
      - FOUND THE BEAT     SOME MOR
      E": RETURN
1089 PRINT "ADVANCED SLEUTH - YO
      U STILL NEED AN     ASSISTAN
      T": RETURN
1091 PRINT "EXPERT OPERATIVE - Y
      OU CAN HANDLE ANY   MISSION
      ALONE": RETURN
1093 PRINT "WORLD RENOWNED OPERA
      TIVE - YOU WILL BE ELEVATED T
      O CONTROL": RETURN
```

ADD OTHER FEATURES

Some other features experienced programmers might try are:

- A. Intersperse your program with full-screen location drawings or animated subroutines of the player's movements. Outstanding text adventures can stand on their own without color, graphics, or sound, but you might consider adding these enhancements if they are part of your programming skills.
- B. Add SAVE GAME and RESTORE GAME features to your adventure. SAVE GAME stores the player's current status on a disk, and RESTORE reads this information back into the program. This feature allows a player to pause in the middle of a game.

PIECES OF THE PUZZLE

Now that you know something about the CIA adventure, you may be interested in the pieces of the puzzle that go together to design an adventure game.

THE PLOT

A plot, such as the one described in Chapter 1, is the starting point. No matter how imaginative, an adventure story requires a fairly logical situation in which the player needs to move from location to location, gathering items for a specific goal. Game players quickly get bored with adventures that rely mainly on pointlessly roaming through locations and hacking monsters.

THE GOAL

Once you have a reasonable plot in mind, decide the goal of the adventure: treasure, points, rescue, arrest, or whatever. You may revise your goal later, but a definite objective will help you design your game play.

The goal of CIA is for the player to “get” four items, worth 10 points each, which definitely incriminate the Ambassador. Some of the items, like the phone bill, prove the evidence couldn’t have been “planted” by someone else.

THE OUTLINE

Next, write an outline of the adventure’s features and jot down all the possibilities that come to mind. List each

room or area that the player may enter during the game and decide on a time frame for the adventure.

Use the outline to write down all of your ideas for the game. Don't try to eliminate any at first; just put down everything that comes to mind. Try moving from room to room and visualize objects that would be found in each room. Add hazards or obstacles and the way each can be overcome as you move through the rooms. Also make note of any objects needed to overcome the obstacles.

The categories used in the CIA outline were: Location Rooms, Suspects, Time, Obstacles and Clues, How They Will Be Overcome or Acquired, Tools Needed to Get Through Obstacles, False Clues, Booby Traps and Hazards, Evidence Needed to Apprehend, and Other Items Which Advance Game and Score.

The outline for CIA looked like this:

Location:

Ambassador Griminski's NY apartment

Rooms:

headquarters
front steps
foyer
drawing room
dining room
kitchen
bedroom with study area
bathroom

Suspects:

Suspects inconsequential to game. Can be added later.

Time:

Friday evening, 9:30 P.M.

Obstacles and Clues:

locked doors
locked drawers
hidden panel
safe
hidden closet for computer with phone modem
microfilm
camera
phone numbers
phone bills
alarm system
attack dog

How They Will Be Overcome or Acquired:

need combination for safe
need key to unlock door
need to disable alarm system
need to find computer
need to drug dog

Tools Needed to Get Through Obstacles:

flashlight
wire cutters
screwdriver
keys

False Clues:

briefcase
umbrella
gun in case

Booby Traps and Hazards:

poison drink
armed umbrella

cyanide capsules

Griminski returns prematurely and eliminates you.

Evidence Needed to Apprehend:

letter containing phone number for computer modem

program for Defense Department computer

camera

microfilm

Other Items Which Advance Game and Score:

none

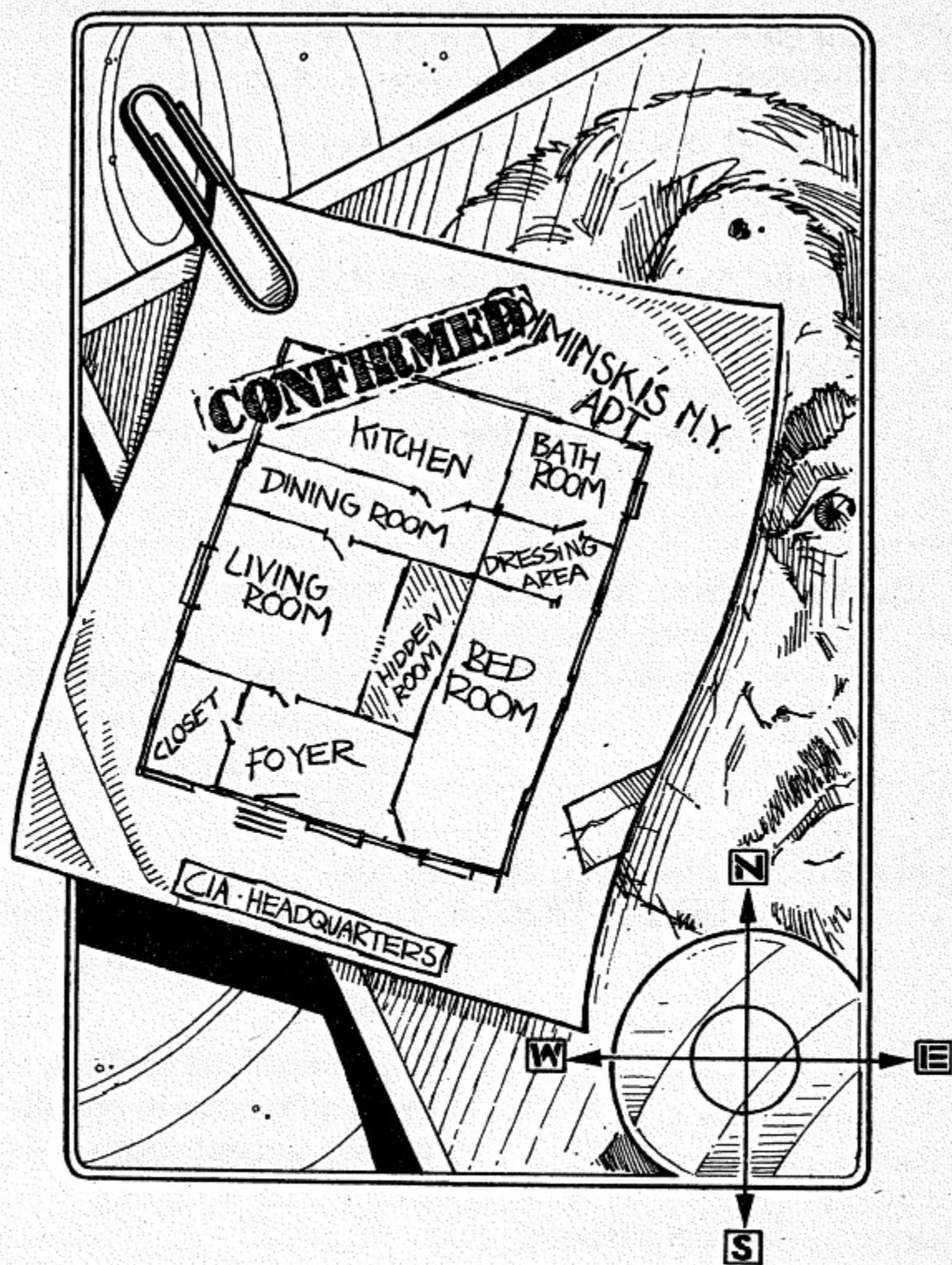
Keep your list handy to jot down additional ideas as they come to you.

THE MAP

Now you're ready for the fun part: drawing a map of the adventure. Just draw how each location or room is connected, and plant the winning items in your map. This is the game map for CIA. (*see opposite page.*)

THE ROOMS

Starting with location #1 on your map, give a general description of each room. Picture the movements of the player in your mind as you do this. Add relevant clues and obstacles which obscure the clues as you go along. Use your outline and add booby traps, hazards, or steps the player must go through to move about the setting. A hazard should be solvable and the outcome sensible, not random. Players resent getting killed off by chance. Finally, include some extraneous objects, but not too



south —foyer—#3
west —nothing

Objects: sofa
coffee table
desk
pad
 READ PAD = password to logon computer
drawer
 OPEN DRAWER = paper with combination to briefcase
pigeonhole
bill
 READ BILL = regular phone bill for Griminski
phone
number
 READ NUMBER = regular phone number
fireplace
panel
 PUSH PANEL = door to hidden room east pops open

Write down the specific information for your adventure, location by location, using this outline form. Add ideas as you think of them and delete those that unnecessarily complicate the play. When you've exhausted your ideas, write up your descriptions in short sentences that will flow together as the player reads along. Then you're ready to think about the BASIC program for your adventure.

4 The BASIC Program

Subroutines for Text Adventures



GENERAL PURPOSE ROUTINES

The BASIC program for CIA was designed to be used easily for different adventures. It consists of a short main program, and subroutines that are called as needed. Most of the subroutines are general purpose routines which can be used for any text adventure. The Verb Subroutine is tailored to the specific clues and hazards within CIA.

All of the information regarding the rooms, objects, and actions available to a player is stored in DATA statements and read into arrays when the game begins. This information is then read and changed as the game progresses.

To use the program for a different adventure, just include the General Purpose Subroutines without any changes, change the information in the DATA statements, and substitute the Verb Subroutines needed for your particular game.

This chapter describes the General Purpose Subroutines, gives the format for the DATA statements, and describes how the Verb Subroutine is written. Also see Appendix A, which contains more detailed information on the General Purpose Subroutines, and Appendix B, which is a list of the program variables in CIA.

DIMENSION STATEMENTS (lines 100-115) set the maximum sizes for each subscripted variable. The maximum size may need to be changed for your game. See Appendix B for the definitions and sizes of the variables in CIA.

INITIALIZATION STATEMENTS (lines 150-160) set the

initial values for program variables not included in the DATA statements. See Appendix B for definitions of these variables.

MAIN PROGRAM (lines 200-260) controls the flow of the program by calling each of the subroutines as needed.

INPUT ROUTINE (lines 300-345) accepts input from the player and finds the blank space separating the verb from the noun in the input string.

PARSING ROUTINE (lines 400-470) matches the player's input with the array of verbs and nouns the program uses.

NEW ROOM ROUTINE (lines 500-510) prints a description of the room as the player enters it.

MOVE ROUTINE (lines 600-650) determines the room number corresponding to the direction input by the player.

READ DATA ROUTINE (lines 700-780) reads game information from the DATA statements into arrays at the beginning of each game.

COMMAND ROUTINE (lines 900-1120) matches the player's input with the array of acceptable commands and performs the action requested.

DATA STATEMENT FORMATS

Most of the information specific to this particular mystery is put in DATA statements. The program reads this data into arrays to start the game.

The format of the DATA statements and the way the data elements are connected form the basis of the program's structure and can be used for any adventure. The pieces of the adventure put in DATA statements are:

Rooms

Help statements

Objects

Verbs

Commands

The first DATA statements, lines 5097-6000, tell the program how many items are in each group as follows:

5097 24 Verbs

5098 10 Rooms and Help statements

5099 7 Commands

6000 59 Objects

The formats are:

ROOMS

Description: Include up to 240 characters describing the room and any relevant objects visible as the player enters the room. Enclose in quotes.

Exits: Room numbers for each of the 4 exits in the order North, East, South, West. Separate by commas. If there is no exit in a direction use a 0.

The DATA statement for Room 1 is:

```
6001 DATA "YOU ARE IN YOUR OFFICE AT THE CIA. ON THE  
SHELVES ARE TOOLS YOU'VE USED IN PAST MISSIONS  
AMBASSADOR GRIMINSKI'S APARTMENT IS NORTH.",2,  
0,0,0
```

HELP STATEMENTS

Include up to 240 characters which offer a hint to a player for each room. Enclose in quotes.

The Help statement for Room 1 is:

```
6011 DATA "YOU'LL NEED SOME TOOLS TO GET INTO THE APA  
RTMENT."
```

COMMANDS

Single words which perform an action independent of any object. Enclose in quotes and separate by commas.

The Command DATA statement for CIA is:

```
6021 DATA "HELP","QUIT","INVENTORY","LOOK","TIME","SC  
ORE","RESTART"
```

VERBS

Verbs available to the player to perform actions on the objects in the game. Enclose in quotes and separate by commas. The Verb DATA statement for CIA is:

```
6031 DATA "LOOK","GET","TAKE","GO","CRAWL","WALK","OP  
EN","READ","DROP","CALL","UNSCREW","SPRAY"  
,"PUSH","LOAD","RUN","DRINK","EAT","CHEW","UNWRAP",  
"TALK","SHOOT","UNLOCK","ON","OFF"
```

OBJECTS

Keyword: Single word used to identify an object. Enclose in quotes.

Description: Include up to 240 characters which describe the object to the player. Enclose in quotes.

Object codes: These numbers enable the program to use objects in the generalized routines. They are:

Room Number—The number of the room where the object is located at the start of the game.

Link—The number of another object hidden by this one. A LOOK at this object will print the description of the linked object. Used when one object is inside or hidden by another.

Take Value—The number of points this object is worth towards solving the mystery. In CIA, four objects have a value of 10; the rest have no score value.

Take Code—This number determines the results of the player attempting to take the object. The codes are:

- 1 - Player can take the object, e.g., tools on shelf in office.
- 2 - Object is hidden; player can't use until some action is performed on another object first. For example, gum inside the safe in the bedroom can't be taken until safe is opened.
- 3 - Object too large to carry, e.g., sofa.
- 4 - Object not one that can be carried, e.g., wall or panel.
- 5 - Object can't be opened or used until some other action performed on it, e.g., door can't be opened until it is unlocked.

The DATA statement for Object 6 is:

```
7006 DATA "SCREWDRIVER"."AN ALL-PURPOSE SCREWDRIVER  
WITH COLLAPSIBLE HANDLE.",1,7,0,1
```

(At the start of the game, the screwdriver is in Room 1, it is linked to Object 7, it has no score value, and it can be taken.)

VERB SUBROUTINE

The Verb Subroutine (lines 2000-4110) consists of a series of short subroutines which perform the necessary action for each verb the program recognizes. These subroutines are called by the multiple GOTO statement on line 2010. The verbs LOOK, GET, GO, and TAKE are used in most adventures, and can be included without change. The rest of the routine must be tailored for the verbs specific to your adventure. However, the format for these verbs is essentially the same: an IF . . . THEN statement to check for the object number, and then print a message or change a program variable as required.

The following examples explain how specific verbs are used:

1. The LOOK routine gives the player the opportunity to examine any object in any room by printing the object's description stored in OD\$. It then prints the description of the object, if any, linked to the one the player named.

For example, the bottle in the kitchen holds capsules. A LOOK BOTTLE prints the description of the bottle and also the description of the capsules. This allows you to hide clues and obstacles behind objects and requires that the player LOOK through the room in order to search for the clues.

The program statements are:

```
2100 PRINT OD$(N)
2110 IF OL(N) = 0 THEN RETURN
2115 N = OL(N)
2120 IF OM(N) = R THEN 2100
2125 GOTO 2110
```

2. The verbs EAT and CHEW are only valid in Room 8, Kitchen, and Room 9, Bedroom. The only objects in the game that can be eaten are the cyanide pills, number 39 or 54, and the gum, number 45. Unfortunately for the player, trying to eat or chew these objects results in the player's demise. An attempt to eat or chew anything else results in the message YOU CAN'T EAT object.

The program statements are:

```
3500 IF N = 39 OR N = 54 THEN PRINT "YOU FOOL! THESE  
ARE CYANIDE CAPSULES.", "YOU FALL WRITHING TO THE  
FLOOR AND", "DIE IN AGONY ... END OF GAME.": END  
3505 IF N = 45 THEN PRINT "YOU IDIOT! THE GUM IS A F  
LASTIC", "EXPLOSIVE. YOU HAVE JUST BLOWN", "YOURSEL  
F TO SMITHEREENS!! ... END OF GAME.": END  
3510 PRINT "YOU CAN'T ";V$:" ";N$: RETURN
```

Now you can see that the program used for CIA can be easily adapted to other adventures and mysteries. The next chapter shows you how to translate the ideas for your own mystery into DATA statements and Verb Subroutines.

5 Write Your Own Game

Coding BASIC Statements



DATA STATEMENTS

Now that you have some idea of the structure of the BASIC program for CIA, you are ready to translate the game information from Chapter 2 into DATA statements and Verb routines.

ROOMS

Number the rooms consecutively, using the map as a guide. Write a DATA statement for each room from the room outline in Chapter 2 as follows:

Descriptions

- Write up the room descriptions in short sentences that will flow together as the player moves through the adventure.
- Include any relevant visible objects or clues in the description.
- Add any hazards and obstacles encountered as the player enters the room.
- Describe the exits giving the room names and directions.

Exits

- Enter the room numbers for each exit. If a direction has no exit, enter 0.

For example, for Room 5 in CIA, until you find the hidden computer room, there are only 2 exits: the Dining Room #7 is north and the Entry #3 is south. The DATA statement is:

```
6005 DATA "YOU ARE IN THE DRAWING ROOM. A DESK IS HER  
E. A SOFA AND A COFFEE TABLE ARE IN FRONT OF THE  
FIREPLACE SET INTO THE PANELED EAST WALL. THE  
DINING ROOM IS NORTH.",7,0,3,0
```

OBJECTS

Using the room outline from Chapter 2, number the objects consecutively, starting with the objects in the first room and continuing through the entire map. Any object that the player will ever need to use with a verb must be listed and numbered separately. Code a DATA statement for each object as follows:

Keyword

- Use a single word which will identify the object, e.g., TABLE.

Description

- Use short sentences which describe the object to the player.

Room Number

- Use the room number where the object is at the start of the game.

Object Link

- Enter the number of another object linked to this one. For example, in Room 9, Object #42 PAPER is linked to Object #43 COMBINATION. If the player says LOOK PAPER, the description of Object 43 is printed also.

Take Value

- Enter a 0 if the object does not help solve the mystery.
- Enter a 10 if the object is needed to solve the mystery.

Take Code

- Enter a number from 1 to 5 which corresponds to the results of a player attempting to TAKE the object. See Data Statement Formats, in Chapter 4 for a description of the codes.

The DATA statement for the PHONE in Room 5 is:

```
7024 DATA "PHONE", "THIS IS A BEIGE PUSH BUTTON DESK  
PHONE.", 5, 25, 0, 4
```

(PHONE is the keyword, and the description describes the PHONE to the player. It is in Room 5, it is linked to Object 25, it has no Take Value, and the player can't take it.)

HELP COMMANDS

Enter up to 240 characters for each room. This clue should aid the player by providing additional information regarding clues and hazards within the room.

- Provide one HELP statement for each room.
- Make the print statements complete sentences.

The HELP statement for Room 5 is:

```
6015 DATA "THERE IS MORE HERE THAN MEETS THE EYE."
```

VERBS

List all the verbs necessary to play the game. This includes moving through the location, disarming hazards, avoiding obstacles, and solving the mystery. Number the verbs consecutively and enter in the DATA statement in the same order.

The verbs for CIA are:

```
6031 DATA "LOOK", "GET", "TAKE", "GO", "CRAWL", "WALK", "OP  
EN", "READ", "DROP", "CALL", "UNSCREW", "SPRAY",  
"PUSH", "LOAD", "RUN", "DRINK", "EAT", "CHEW", "UNWRAP",  
"TALK", "SHOOT", "UNLOCK", "ON", "OFF"
```

COMMANDS

List the one-word commands necessary to operate your game, and number them consecutively. Enter the commands in a DATA statement in the same order.

The commands for CIA are:

```
6021 DATA "HELP", "QUIT", "INVENTORY", "LOOK", "TIME", "SCORE", "RESTART"
```

The program reads the DATA statements in order and stores the information in arrays. The program expects the groups of DATA statements to be in the order:

Rooms
Help Statements
Commands
Verbs
Objects

Within each group, the DATA statements for Rooms, Objects and Help must be in numerical order. For example, Room 5 should be the 5th statement in the Room DATA statements, Object 49 should be the 49th statement in the Object group, etc.

The order for Commands and Verbs corresponds to the line numbers in the multiple GOTO statements in the Command and Verb routines. For example, since the first line number in the multiple GOTO statement on line 2010 in the Verb routine points to the LOOK subroutine, starting at line number 2100, the Verb LOOK must be the first verb in the DATA statement.

VERB SUBROUTINES

Translate the hazards and obstacles in each room outline in Chapter 2 into a BASIC routine. Most hazards and obstacles can be put into IF . . . THEN statements.

For Room 3, the information taken from the room outline:

READ LETTER = the phone bill made out to modem number in name of Grim, different address

is coded in the READ routine as:

```
2500 IF R = 3 AND N = 16 THEN PRINT "THE TELEPHONE BILL IS MADE", "OUT TO 322-9678 -V.GRIM, P.O. X", "GRAND CENTRAL STATION NYC", "THE AMOUNT IS $247.36 FOR LONG", "DISTANCE CHARGES TO WASHINGTON D.C.": RETURN
```

The logic for the obstacles in Room 5 is translated as follows:

READ PAD = password to log on computer

is coded in the READ routine as:

```
2505 IF N = 20 THEN PRINT "YOU CAN JUST MAKE OUT THIS MESSAGE", "HEL-ZXT.93.ZARF.1": RETURN
```

READ BILL = regular phone bill for Griminski

is coded in the READ routine as:

```
2510 IF N = 23 THEN PRINT "THE BILL IS MADE OUT TO 322-8721", "AMBASSADOR VLADIMIR GRIMINSKI", "14 PARKSIDE AVENUE - NYC.", "THE BILL IS FOR $68.34 FOR MOSTLY", "LOCAL CALLS.": RETURN
```

READ NUMBER = regular phone number

is coded in the READ routine as:

```
2515 IF N = 25 THEN PRINT "322-8721": RETURN
```

Some of the hazards need more complicated routines. The attack dog hazard in Room 3 is translated this way:

SPRAY DOG = If you have DRUG, the dog is drugged and falls at your feet.

If you don't have DRUG, you can't move in any direction except south.

requires the following program statements:

Check the array of objects the player is carrying (CA):

```
2910 FOR I = 1 TO IC
2915 IF CA(I) = 10 THEN 2930
2920 NEXT
```

If DRUG, Object 10, is not in the player's inventory, then print the message:

```
2925 PRINT "YOU HAVE NOTHING TO SPRAY WITH": RETURN
```

If DRUG is in the player's inventory, then print the message:

```
2930 PRINT "THE DOG IS DRUGGED AND FALLS HARMLESSLY", "
      AT YOUR FEET."
```

Set the array of room exits (RE) to the correct rooms,

since now the player can move north, east and west, as well as south:

```
2935 RE(3,1) = 5:RE(3,2) = 9:RE(3,4) = 4
```

Change the description of Room 3 (RD\$):

```
2940 RD$(3) = LEFT$(RD$(3),176) + " THE DRUGGED DOG  
IS ON THE FLOOR."
```

Change the description of the dog (OD\$):

```
2945 OD$(14) = "THE FIERCE DOBERMAN LIES DRUGGED ON T  
HE FLOOR."
```

Drop DRUG from the array of player's inventory:

```
2947 CA(I) = CA(IC):IC = IC - 1  
2948 PRINT "THE DRUG IS USED UP AND NO LONGER IN YO  
UR INVENTORY."
```

And return to the main program:

```
2950 RETURN
```

Translating the OPEN drawer obstacle in Room 5:

OPEN DRAWER = paper with combination to brief-
case

requires the following statements in the OPEN verb rou-
tine to print the message OPENED, put Object 57 in the
object link array (OL) for Object 21 and RETURN:

```
2411 IF N = 21 THEN PRINT "OPENED":OL(21) = 57:OT(57)
      = 1: RETURN
```

The hidden room behind the panel in the Drawing Room:

PUSH PANEL = door to hidden room east pops open
requires the following statements in the PUSH routine:

Print the message:

```
3110 PRINT "THE PANEL POPS OPEN TO REVEAL THE", "ENTRAN  
      CE TO A PREVIOUSLY HIDDEN ROOM."
```

Add the hidden room to the array of exits (RE):

```
3115 RE(5,2) = 6
```

Change the description of Object 26, PANEL:

```
3120 OD$(26) = LEFT$(OD$(26),35) + "A      HIDDEN ROOM  
      CAN BE SEEN BEHIND ONE      PANEL."
```

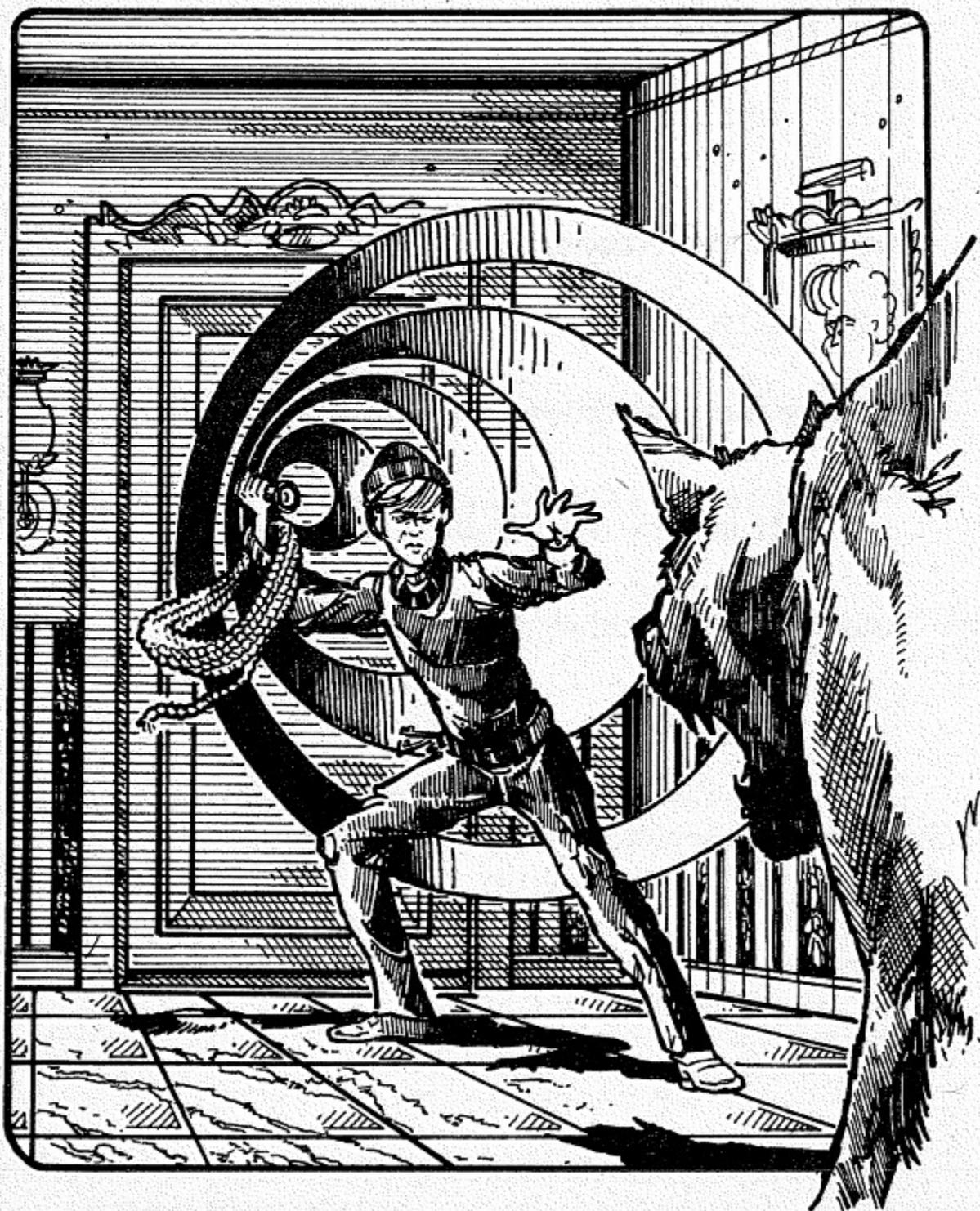
And RETURN to the main program:

```
3125 RETURN
```

Obviously, the structure and complexity of the Verb Subroutines are determined by the difficulty of the obstacles and hazards in a game. A wise choice would be to start with a fairly simple game, and then add enhancements and improvements as you become more familiar with the program.

6 Game Testing

Get Rid of Program Bugs



DEBUGGING YOUR GAME

Now that your game is up and running, it is time to debug and playtest your adventure. This stage is essential in writing a smooth, professional-quality game. We recommend using the steps described in this chapter.

PLAY THE GAME

Play the program logically. Move from room to room, looking at objects, collecting some, dropping others, and resolving obstacles. Change the code temporarily so the program doesn't go back to the beginning each time you get killed or run out of time. Eliminate all the program bugs. At this point, don't labor over the screen format of your output. Concentrate on the programming logic.

Here are two sample test runs of CIA:

NOW WHAT ? UNSCREW ALARM

THE ALARM SYSTEM IS OFF

NOW WHAT ? UNLOCK DOOR

UNLOCKED

NOW WHAT ? GO NORTH

I CAN'T GO THAT DIRECTION

NOW WHAT ? OPEN DOOR

OPENED

NOW WHAT ?

The first one shows the program logic working appropriately: you can't go north until you unscrew the alarm, unlock the door, and open the door:

THIS IS THE MARBLED FOYER OF THE
AMBASSADOR'S APARTMENT. THERE IS A
TABLE IN THE CORNER. THE MASTER BEDROOM
IS EAST, THE DRAWING ROOM IS NORTH AND
A CLOSET WEST. A FIERCE DOG CHARGES
TO ATTACK.

NOW WHAT ? LOOK DOG

THE SAVAGE DOBERMAN LEAPS TOWARD YOU
WITH BARED FANGS. HE WILL NOT LET YOU
PAST HIM.

NOW WHAT ? TALK DOG

THAT WON O O O : RETURN
IT BITES THE GUN OUT OF YOUR HAND!

NOW WHAT ?

This run shows a programming bug when you try to talk to the dog. It was necessary to trace your error and change the code so the program would run correctly.

TEST YOUR ADVENTURE

Once the program is running smoothly in "normal" play, use unusual commands to test your adventure. You want to be sure the program logic can handle all possibilities. Try using crazy verb-noun combinations, taking hid-

den items before they are revealed, taking things you don't expect the player to want, and moving in directions that are not possible. Make sure the game gives the player reasonable replies to these inputs.

NOW WHAT ? GO NORTH

YOU ARE IN THE DRAWING ROOM. A DESK IS HERE. A SOFA AND A COFFEE TABLE ARE IN FRONT OF THE FIREPLACE SET INTO THE PANELED EAST WALL. THE DINING ROOM IS NORTH.

NOW WHAT ? LOOK DESK

THE LARGE OAK DESK HAS A BLOTTER AND PEN SET ON IT. A PHONE IS HERE. A BLANK NOTE PAD IS BY THE PHONE. THE DESK HAS A PIGEONHOLE AND ONE DRAWER IN IT.

NOW WHAT ? READ PAD

YOU CAN JUST MAKE OUT THIS MESSAGE
HEL-ZXT.93.ZARF.1

NOW WHAT ? EAT PAD

YOU CAN'T EAT PAD

NOW WHAT ? BURN PAD

I DON'T KNOW HOW TO BURN

CLEAN UP THE OUTPUT

Clean up the program output. Check all possible spelling errors, correct typographical errors, and format your output so it is easy to read on the screen. It may be tedious, but your friends will not have your patience in overlooking this aspect of the game when you are ready to ask them to playtest your adventure.

IT WON'T HELP

NOW WHAT ? OPEN CASE

COMBINATION ? 2-2-2

SORRY - YOU DON'T HAVE THE RIGHT COMBINATION

NOW WHAT ? BOMB CASE

I DON'T KNOW HOW TO BOMB

NOW WHAT ? SHOOT CASE

THAT JUST MAKES A BIG MESS!

NOW WHAT ?

Here the PRINT statement in line 2435 needs to be spaced so the word "combination" isn't split on the screen.

PLAYTEST YOUR GAME

Enlist the help of several friends to playtest your game and listen to their criticism. Sit next to them and take

notes about their reactions. Some of the criticism will be bad. Just ignore it. Some of it will be good and you can incorporate the ideas immediately. While it is hard not to give your friends hints and clues, it's important to know if they can follow the game logic and arrive at a solution on their own.

EVALUATE YOUR GAME

Evaluate your game by answering these questions:

Is your adventure challenging?

Is the goal obvious and meaningful, yet the outcome uncertain?

Does the game have multiple goal levels, like score-keeping or time limits?

Does the program have surprises and hidden information that can be revealed? Do you need to include more of these features?

Can other players figure out how to overcome the obstacles and discover the objectives of your adventure without getting so frustrated they lose interest and quit?

Do you need more hints in the game itself?

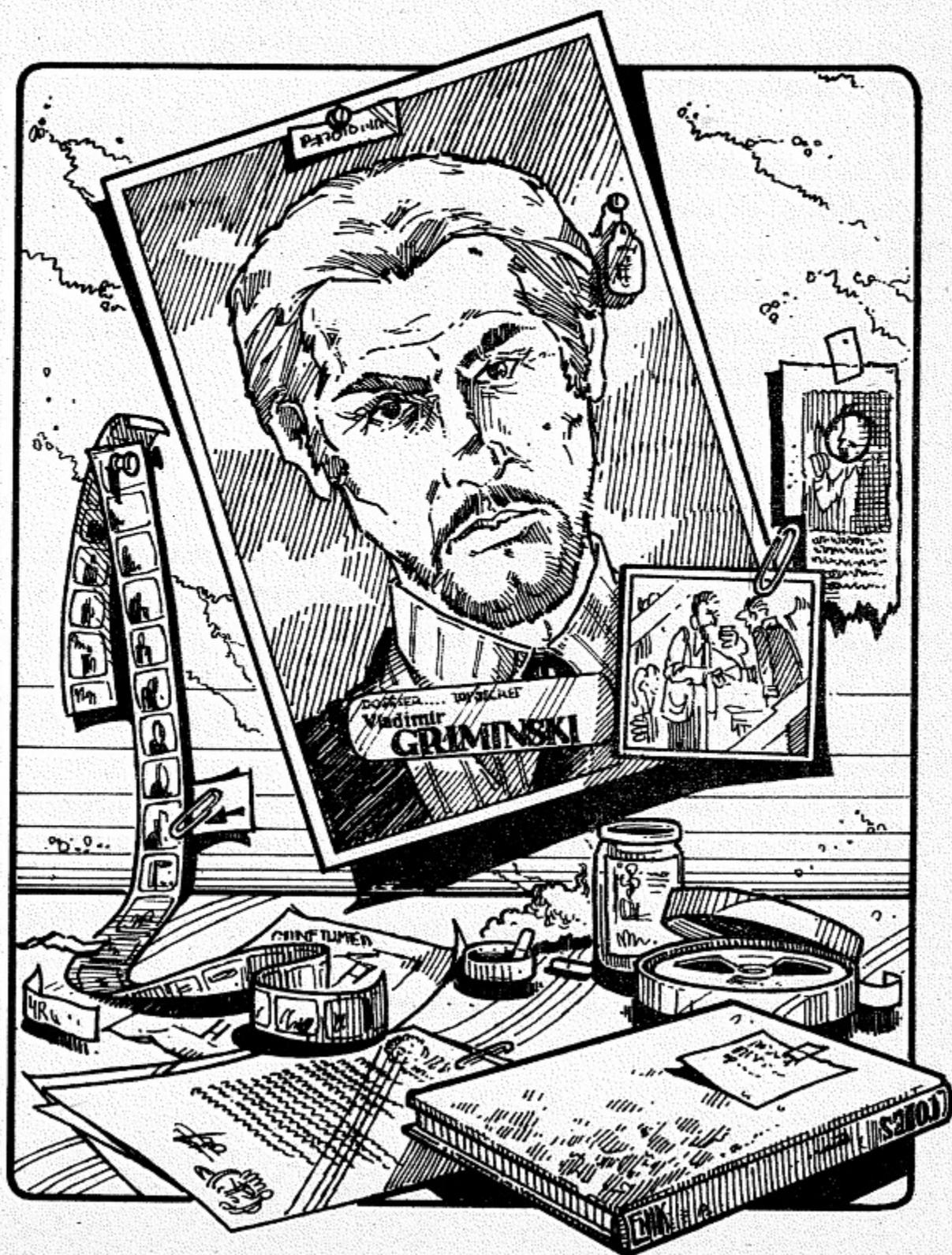
IMPROVE YOUR GAME

Revise and improve your game. Playtest it until you think it's perfect. Then, polish your adventure. You will get sick of it, but this is the final element that makes a

computer game outstanding. You know a good one when you play it, and you also know when a programmer stopped before his game was polished.

7 Game Documentation

The Final Polish



DOCUMENTATION

The information a player needs to understand a game—its objectives and playing features—is called the documentation. This is often the most slighted aspect of an otherwise good program. Without adequate documentation the player will be confused or frustrated at best, or will give up altogether on the adventure.

The documentation can be written into the program itself as a separate series of routines on background and instructions, or can be discussed separately in the form of a pamphlet or manual. Well-written text adventures usually provide some features of both methods.

The documentation of an adventure should include the following features:

INTRODUCTION

As an introduction to the program, the documentation first needs to discuss the plot or background to the game, the physical setting (caves, apartment, outer space, etc.), the time of the action (prehistoric, 21st century, present day, etc.), and what is required of the player to solve the adventure.

For CIA, this was done in the section of Chapter 1 called The Plot. It could have been included as an introductory subroutine to CIA, or in a separate booklet with illustrations if we were marketing the adventure on a disk or cassette. Notice how the illustrations help the player visualize the setting of the adventure.

INSTRUCTIONS

Next the documentation needs to include instructions for the user about communicating with the program. For CIA these were covered in the Moving, Gathering Clues, Ending a Successful Mission, Verb Summary, and Command Summary sections of Playing CIA in Chapter 1. They could also have appeared in a directions or instructions subroutine in the program.

HINTS

Hints for playing the game should also be covered in the documentation. CIA addresses for these under Time Element, Strategy, and Hazards in Hints for Playing CIA. These suggestions help the user avoid nasty surprises or unnecessary frustration in playing the adventure. Gamers resent not being properly "briefed" before an adventure.

PROGRAM INFORMATION

Finally, include some practical information in the documentation about loading, running, and restarting your game. Each adventure has some unique features, and not all players are seasoned gamers. This is covered in How To Put Up CIA and Running CIA in Chapter 1.

PACKAGING EXTRAS

If you have an artistic bent, the program documentation can be an opportunity to really heighten the appeal of your game. It can add to the intrigue and enjoyment of playing the adventure to provide pictures, photographs, maps, and playing boards and magnetic pieces. Sample evidence could include police reports, bills, airline tickets, fake medication, clothing or hair samples, fingerprints, and receipts.

Scoring techniques can be augmented with checklists, notebooks, and elaborate scoring procedures.

The efforts you expend on the documentation of your adventure may well be the key to whether others are seriously interested in your adventure. A rich imagination and a sense of humor should add a special flair to your programs, which makes them fun to write and exciting to play.

Appendix A

Description of the General Purpose Subroutines in CIA

INPUT SUBROUTINE (300-345)

- Accepts input from the keyboard and puts it in variable R\$.
- Finds the position of the first blank space in R\$ and stores it in variable B. If the input has only 1 word, sets B = 0.
- Increments variable TT, total time, by 3.
- If TT exceeds ET, the maximum time allowed for the game, game ends.

PARSING SUBROUTINE (400-470)

- Assumes the first word in R\$ is a verb and the second is a noun.
- Puts the first word in variable V\$.
- Finds a match in the array of verbs, VA\$, and sets V = position in VA\$ of the word matched.
- If no matching verb in VA\$, prints I DON'T KNOW HOW TO V\$, sets V = 0, and returns.
- Puts the second word in variable N\$.
- Finds match in array of nouns, OA\$, checks that the object is either in the current room or in inventory, and sets N = position in OA\$ of the word matched.
- If no matching noun in OA\$, prints IT WON'T HELP, sets N = 0, and returns.

NEW ROOM (500-510)

- Clears screen and prints the room description when player enters a new room.

MOVE SUBROUTINE (600-650)

- Determines the room number corresponding to the direction input by the player.
- Finds a match in D\$, the array of directions, with N\$.
- Array RE contains the room numbers for each exit in the current room.
- Sets $R = RE(I,J)$; if $RE(I,J) = 0$, then prints I CAN'T GO IN THAT DIRECTION.

READ DATA (700-780)

- Reads information from the DATA statements needed for the game into arrays.

COMMANDS (900-1120)

- Finds a match in CA\$, the array of commands, with player's input.
- If no match, prints I CAN'T UNDERSTAND and returns.
- If match, uses an expanded GOTO statement (line 900) to execute the appropriate code.
- The commands used in CIA are:
 - HELP (930-931) Prints H\$, the help message, for the current room.
 - QUIT (934-937) Prints the time elapsed, the score, and ends game.
 - INVENTORY (940-965) Prints a list of the objects being carried, from array CA.
 - LOOK (1000-1000) Prints a description of the current room.

TIME (1030-1030) Prints TT, elapsed time.
SCORE (1060-1073) Calculates the total score
for the evidence being carried.
RESTART (1100-1120) Initializes all the variables
to start a new game.

VERB SUBROUTINES

This section of the program consists of a subroutine for each Verb that the program understands. The multiple GOTO statement (line 2010) branches to the appropriate subroutine for the verb input by the player. Each subroutine contains the logic needed to perform the action requested by the player, as well as any obstacles or hazards encountered when that Verb is used. Some verbs, such as LOOK and TAKE, can be used with any objects. Others, such as EAT and DRINK, are only appropriate with a few objects.

Each subroutine processes the request if possible and returns to the main program. If a request does not make sense, such as EAT TABLE, a message is printed. In this case, YOU CANNOT EAT A TABLE.

Some verb subroutines are essentially the same for every adventure. Others must be changed for the hazards and obstacles for a particular adventure. The following subroutines are general:

TAKE and GET (2200-2270) Allow the player to take an object from a room and carry it. The action performed on an object is dependent on the value of OT, the object take code, as follows:

- 1 - Player can take the object.
- 2 - Object is hidden from view, e.g., behind a closed door or inside another object. Object can't be taken until some action is performed to expose it.

- 3 - Object is too large or heavy to carry, e.g., sofa.
- 4 - Object can't be moved, e.g., wall or panel.
- 5 - Object can't be taken or used until some action performed; e.g., the door must be unlocked before it can be opened.

If an object can be taken, then it is put in the player's inventory by incrementing the variable IC by one, setting the array CA(IC) to the object number, and setting the object's room number OM to 100.

If the player already has 6 objects, prints YOU CAN'T CARRY ANYTHING ELSE.

GO (2300-2300) Calls the MOVE subroutine and returns.

DROP (2600-2630) Looks for a match in the player's inventory array CA, for the object in N\$. If a match is found:

- Takes object number out of array CA.
- Decreases IC by 1.
- Changes the room number OM for the object to the current room.
- Prints DROPPED and returns.

If there is no match, prints YOU AREN'T CARRYING A N\$ and returns.

Appendix B

Dictionary of Variables for CIA

TERM	DEFINITION
VA\$(30)	Verbs available in the game
CA\$(10)	Commands available in the game
D\$(4)	Directions available to move D\$(1) = NORTH D\$(2) = EAST D\$(3) = SOUTH D\$(4) = WEST
CA(10)	Object numbers of the objects being carried
RD\$(10)	Room descriptions
RE(10,4)	Room numbers of exits in each of the 4 directions
OA\$(75)	Object keywords
OD\$(75)	Object descriptions
OM(75)	Object room number
OL(75)	Object number linked to this one
OV(75)	Object take value
OT(75)	Object take code
IC	Number of objects being carried
TT	Total time elapsed
ET	Total time available to player before end of game
R	Current room number
N\$	Keyword of object input by player
N	Position in OA\$, object array, of current object
V\$	Verb input by player

V	Position in VA\$, verb array, of current verb
C	Position in command array of current command
B	Location in input string of first space
A\$	Local string variable available to use in routines
G	Flag used to determine if Griminski has returned G = 0 Griminski not there G = 1 Griminski returned and has been shot
LI(10)	Stores the object numbers named by the player as evidence

Appendix C

Program Listing for CIA

APPLE USERS - Type in program listing exactly as shown.

IBM USERS - Replace line 500 in the program listing with:

500 CLS: KEY OFF

```
100 DIM VA$(30),CA$(10),D$(4),CA(10)

105 DIM RD$(10),RE(10,4)
110 DIM OA$(75),OD$(75),OM(75),OL(75
    ),OV(75),OT(75)
115 DIM LI(10)
150 IC = 0:TT = 0:ET = 1000
155 RESTORE
160 G = 0
197 REM
198 REM ----- MAIN PROGRAM -----
199 REM
200 GOSUB 700 REM READ FILE
209 R = 1
210 GOSUB 500: REM NEW ROOM
215 PRINT
220 GOSUB 300 REM INPUT
225 PRINT
230 IF B > 0 THEN 245
235 GOSUB 900 REM COMMAND
240 GOTO 215
245 GOSUB 400: REM PARSING
250 IF V = 0 OR N = 0 THEN 215
255 GOSUB 2000 REM ACTION
```

```

260 GOTO 215
297 REM
298 REM -----INPUT ROUTINE--
299 REM
300 R$ = ""
305 TT = TT + 3
310 IF TT > ET THEN PRINT "SORRY...
      YOU RAN OUT OF TIME": END
315 INPUT "NOW WHAT ? ";R$
320 PRINT
325 FOR B = 1 TO LEN (R$)
330 IF MID$ (R$,B,1) = " " THEN RETURN

335 NEXT
340 B = 0
345 RETURN
397 REM
398 REM -----PARSING-----
400 V$ = LEFT$ (R$,B - 1): REM VERB

405 FOR V = 1 TO NV
410 IF V$ = VA$(V) THEN 435
415 NEXT
420 PRINT "I DON'T KNOW HOW TO ";V$
425 V = 0
430 RETURN
435 L = LEN (R$): REM NOUN
440 N$ = MID$ (R$,B + 1,L)
445 FOR N = 1 TO NN
450 IF N$ = OA$(N) AND (OM(N) = R OR
      OM(N) = 100) THEN RETURN
455 NEXT
460 PRINT "IT WON'T HELP "
465 N = 0
470 RETURN
497 REM
498 REM -----NEW ROOM-----
499 REM
500 HOME

```

```

505 PRINT RD$(R)
510 RETURN
597 REM
598 REM -----MOVE ROUTINE-----
599 REM
600 FOR J = 1 TO 4
605 IF D$(J) = N$ THEN 625
610 NEXT
615 GOTO 645
625 IF RE(R,J) = 0 THEN 645
630 R = RE(R,J)
635 GOSUB 500
640 RETURN
645 PRINT "I CAN'T GO THAT DIRECTION
"
650 RETURN
697 REM
698 REM ----- READ DATA -----
699 REM
700 REM
720 READ NV,NR,NC,NN
730 FOR I = 1 TO NR: READ RD$(I)
735 FOR J = 1 TO 4: READ RE(I,J): NEXT
: NEXT
736 FOR I = 1 TO 10: READ H$(I): NEXT
740 FOR I = 1 TO NC: READ CA$(I): NEXT
742 FOR I = 1 TO NV: READ VA$(I): NEXT
745 FOR I = 1 TO NN
750 READ OA$(I),OD$(I),OM(I),OL(I),O
V(I),OT(I)
755 NEXT
765 D$(1) = "NORTH":D$(2) = "EAST"
770 D$(3) = "SOUTH":D$(4) = "WEST"
780 RETURN
897 REM

```

```

898 REM ----- COMMANDS -----
899 REM
900 FOR C = 1 TO NC
905 IF R$ = CA$(C) THEN 925
910 NEXT
915 PRINT "I CAN'T UNDERSTAND ";R$
920 RETURN
925 ON C GOTO 930,934,940,1000,1030,
    1060,1100
926 PRINT "INVALID COMMAND CODE"C: RETURN

929 REM --- COMMAND - HELP ---
930 PRINT H$(R)
931 RETURN
932 REM --- COMMAND - QUIT ---
934 INPUT "ARE YOU SURE YOU WANT TO
QUIT? ";R$
935 IF R$ = "NO" THEN RETURN
936 GOSUB 1030
937 GOSUB 1060
938 END
939 REM --- COMMAND - INVENTORY
940 IF IC = 0 THEN PRINT "YOU AREN'
T CARRYING ANYTHING": RETURN
945 PRINT "YOU HAVE"
950 FOR I = 1 TO IC
955 PRINT OA$(CA(I))
960 NEXT
965 RETURN
999 REM --- COMMAND - LOOK ---
1000 PRINT RD$(R): RETURN
1029 REM --- COMMAND - TIME ---
1030 PRINT "ELAPSED TIME IS "TT" MIN
UTES.": RETURN
1059 REM --- COMMAND - SCORE ---
1060 IF IC = 0 THEN PRINT "YOU AREN
'T CARRYING ANYTHING": RETURN
1065 S = 0

```

```

1070 FOR I = 1 TO IC
1071 S = S + OV(CA(I))
1072 NEXT
1073 PRINT "YOU HAVE "S" POINTS FOR
      EVIDENCE.": RETURN
1099 REM --- COMMAND - RESTART ---

1100 INPUT "ARE YOU SURE YOU WANT TO
      RESTART ";R$
1110 IF R$ = "YES" THEN 150
1120 PRINT "SINCE YOU DON'T WANT TO
      RESTART": RETURN
1997 REM
1998 REM --- VERB ROUTINES ---
1999 REM
2000 IF OT(N) < > 2 THEN 2010
2001 PRINT "YOU CAN'T "V$" "N$" YET.
      ": RETURN
2010 ON V GOTO 2100,2200,2200,2300,2
      300,2300,2400,2500,2600,2700,280
      0,2900,3100,3200,3300,3400,3500,
      3500,3600,3700,3800,3900,4000,41
      00
2011 PRINT "INVALID VERB NUMBER "V: RETURN

2099 REM ---- VERB - LOOK ----
2100 PRINT OD$(N)
2110 IF OL(N) = 0 THEN RETURN
2115 N = OL(N)
2120 IF OM(N) = R THEN 2100
2125 GOTO 2110
2199 REM ---- VERBS - TAKE, GET ---
      -
2200 K = OT(N)
2205 ON K GOTO 2210,2270,2240,2260,2
      270
2206 PRINT "INVALID TAKE CODE FOR OB
      JECT "OA$(N),OT(N): RETURN

```

```

2210 IF IC < 6 THEN 2220
2215 PRINT "YOU CAN'T CARRY ANYTHING
ELSE": RETURN
2220 IF OM(N) = 100 THEN PRINT "YOU
ALREADY HAVE IT": RETURN
2230 PRINT "TAKEN"
2232 OM(N) = 100
2234 IC = IC + 1
2236 CA(IC) = N
2238 RETURN
2240 PRINT "SILLY, THAT'S TOO HEAVY
TO CARRY": RETURN
2260 PRINT "THAT'S RIDICULOUS!": RETURN
2270 PRINT "YOU CAN'T TAKE "N$" YET.
": RETURN
2299 REM ---- VERB - GO ----
2300 GOSUB 600: REM MOVE ROUTINE
2310 RETURN
2399 REM ---VERB - OPEN ---
2400 IF N = 12 THEN 2420
2401 IF N = 18 THEN 2430
2402 IF N = 44 THEN 2440
2403 IF N = 49 THEN 2450
2410 IF N = 17 THEN PRINT "YOU STAB
YOURSELF WITH THE TIP", "WHICH I
S A POISONED DART.", "YOU ARE RUS
HED TO THE HOSPITAL", "END OF GAM
E": END
2411 IF N = 21 THEN PRINT "OPENED":
OL(21) = 57:OT(57) = 1: RETURN
2412 IF N = 37 THEN PRINT "OPENED":
OL(37) = 38:OT(38) = 1:RD$(8) =
LEFT$(RD$(8),169) + "OPEN": RETURN
2419 PRINT "A "N$" CAN'T BE OPENED":
RETURN
2420 IF OT(12) = 4 AND OT(13) = 4 THEN
PRINT "OPENED":RE(2,1) = 3: RETURN

```

```

2422 IF OT(12) = 5 THEN PRINT "THE
DOOR IS LOCKED.": RETURN
2424 IF OT(13) = 5 THEN PRINT "YOU
DIDN'T DISCONNECT THE ALARM.", "I
T GOES OFF AND THE POLICE COME A
ND", "ARREST YOU...END OF GAME.":
END
2426 PRINT "CAN'T GET THROUGH DOOR Y
ET": RETURN
2430 INPUT "COMBINATION ";C$
2432 IF C$ = "2-4-8" THEN PRINT "OP
ENED":OD$(18) = OD$(18) + "PARTS
OF AN RR-13 RIFLE ARE INSIDE TH
E PADDED CASE.": RETURN
2435 PRINT "SORRY - YOU DON'T HAVE T
HE RIGHT COMBINATION": RETURN
2440 INPUT "COMBINATION ";C$
2442 IF C$ = "20-15-9" THEN PRINT "
OPENED":OL(44) = 45:OT(45) = 1:O
D$(44) = OD$(44) + " INSIDE I
S.": RETURN
2445 PRINT "SORRY - COMBINATION ISN'
T RIGHT": RETURN
2450 PRINT "OPENED":OT(51) = 1:OL(49
) = 51:RD$(10) = LEFT$(RD$(10)
,184) + "OPEN": RETURN
2499 REM --- VERB - READ ---
2500 IF R = 3 AND N = 16 THEN PRINT
"THE TELEPHONE BILL IS MADE", "OU
T TO 322-9678 -V.GRIM, P.O. X", "
GRAND CENTRAL STATION NYC", "THE
AMOUNT IS $247.36 FOR LONG", "DIS
TANCE CHARGES TO WASHINGTON D.C.
": RETURN
2505 IF N = 20 THEN PRINT "YOU CAN
JUST MAKE OUT THIS MESSAGE", "HEL
-ZXT.93.ZARF.1": RETURN

```

```

2510 IF N = 23 THEN PRINT "THE BILL
      IS MADE OUT TO 322-8721", "AMBAS
      SADOR VLADIMIR GRIMINSKI". "14 PA
      RKSIDE AVENUE - NYC.", "THE BILL
      IS FOR $68.34 FOR MOSTLY", "LOCAL
      CALLS.": RETURN
2515 IF N = 25 THEN PRINT "322-8721
      ": RETURN
2520 IF N = 30 THEN PRINT "322-9678
      ": RETURN
2525 IF N = 42 THEN PRINT "20-15-9"
      : RETURN
2530 IF N = 56 THEN PRINT "322-8721
      ": RETURN
2590 PRINT "NOTHING TO READ": RETURN

2599 REM --- VERB - DROP ---
2600 FOR I = 1 TO IC
2605 IF N = CA(I) THEN 2620
2610 NEXT
2615 PRINT "YOU AREN'T CARRYING A "N
      $: RETURN
2620 OM(CA(I)) = R
2622 CA(I) = CA(IC)
2625 IC = IC - 1
2630 PRINT "DROPPED": RETURN
2699 REM --- VERB - CALL ---
2700 IF N = 53 AND (R = 5 OR R = 6 OR
      R = 9) THEN 2710
2708 IF N < > 53 THEN PRINT "IT'S
      NO USE TO CALL ";N$: RETURN
2709 PRINT "YOU'RE NOT NEAR A PHONE
      ": RETURN
2710 PRINT "RING...RING"
2711 PRINT "HELLO, AGENT. THIS IS Y
      OUR CONTROL"
2712 PRINT "SPEAKING"
2713 PRINT "LIST YOUR TANGIBLE EVIDE
      NCE".

```

```

2715 EV = 0:LL = 0: FOR I = 1 TO IC:L
      I(I) = 0: NEXT
2717 GOSUB 2780
2719 IF EV > = 40 THEN 2730
2720 PRINT "I'M SORRY YOU HAVE INSUF
      FICIENT", "EVIDENCE FOR A CONVICT
      ION. CALL ME", "WHEN YOU HAVE MO
      RE INFORMATION": RETURN
2730 PRINT "FANTASTIC JOB!!"
2731 PRINT "WE'LL BE OVER IN A FLASH
      !"
2732 PRINT "TO ARREST THE SUSPECT"
2733 TT = TT + 6: IF TT > ET THEN 310

2734 PRINT " -----"
2735 PRINT "AMBASSADOR GRIMINSKI ARR
      IVES HOME AT", "10:30 TO FIND OPE
      RATIVES WAITING", "TO ARREST HIM.
      "

2736 PRINT " -----"
2737 PRINT "YOU ARE HANDSOMELY REWAR
      DED FOR YOUR", "CLEVER SLEUTHING.
      "

2738 PRINT "YOU SOLVED THE MYSTERY I
      N ";TT;" MINUTES."
2739 END
2779 REM GET EVIDENCE
2780 INPUT R$
2781 IF R$ = "" THEN RETURN
2782 FOR I = 1 TO IC
2783 IF R$ = UA$(CA(I)) THEN 2786
2784 NEXT
2785 PRINT "YOU'RE NOT CARRYING A "R
      $: GOTO 2780
2786 FOR J = 1 TO LL: IF LI(J) = CA(
      I) THEN 2788: NEXT
2787 GOTO 2789
2788 PRINT "YOU ALREADY SAID ";R$: GOTO
      2780

```

```

2789 EV = EV + OV(CA(I)):LL = LL + 1:
      LI(LL) = CA(I)
2790 GOTO 2780
2799 REM ----- VERB - UNSCREW --
2800 IF N = 13 THEN 2810
2808 PRINT "YOU CAN'T UNSCREW A ";N$

2809 RETURN
2810 FOR I = 1 TO IC
2815 IF OA$(CA(I)) = "SCREWDRIVER" THEN
      2830
2820 NEXT
2825 PRINT "YOU HAVE NOTHING TO UNSC
      REW WITH": RETURN
2830 PRINT "THE ALARM SYSTEM IS OFF"
      :OT(13) = 4:OD$(13) = "THE ALARM
      IS DISABLED": RETURN
2899 REM ----- VERB - SPRAY -----
2900 IF N = 14 OR N = 10 THEN 2910
2909 PRINT "YOU CAN'T SPRAY A "N$: RETURN

2910 FOR I = 1 TO IC
2915 IF CA(I) = 10 THEN 2930
2920 NEXT
2925 PRINT "YOU HAVE NOTHING TO SPRA
      Y WITH": RETURN
2930 PRINT "THE DOG IS DRUGGED AND F
      ALLS HARMLESSLY", "AT YOUR FEET."

2935 RE(3,1) = 5:RE(3,2) = 9:RE(3,4) =
      4
2940 RD$(3) = LEFT$(RD$(3),176) + "
      THE DRUGGED DOG IS ON THE FLOO
      R."
2945 OD$(14) = "THE FIERCE DOBERMAN L
      IES DRUGGED ON THE FLOOR."
2947 CA(I) = CA(IC):IC = IC - 1
2948 PRINT "THE DRUG IS USED UP AND
      NO LONGER IN YOUR INVENTORY."

```

```

2950 RETURN
3099 REM ---- VERB - PUSH ----
3100 IF N = 26 THEN 3110
3109 PRINT "IT DOESN'T DO ANY GOOD T
O PUSH A "N$: RETURN
3110 PRINT "THE PANEL POPS OPEN TO R
EVEAL THE", "ENTRANCE TO A PREVIO
USLY HIDDEN ROOM."
3115 RE(5,2) = 6
3120 OD$(26) = LEFT$(OD$(26),35) +
"A HIDDEN ROOM CAN BE SEEN B
EHIND ONE PANEL."
3125 RETURN
3199 REM ----- VERB - LOAD -----
3200 IF N = 28 THEN 3210
3205 PRINT "CAN'T LOAD A "N$: RETURN

3210 IF OM(28) = 6 THEN PRINT "THE
PROGRAM IS ALREADY LOADED": RETURN

3215 PRINT "THAT WON'T HELP YOU"; RETURN

3299 REM --- VERB - RUN ---
3300 IF N = 28 THEN 3310
3309 PRINT "YOU CAN'T RUN A ";N$: RETURN

3310 IF OT(31) = 5 OR OT(32) = 5 OR
OT(33) = 5 THEN PRINT "THE COMP
UTER CAN'T RUN THE PROGRAM YET."
: RETURN
3311 OT(28) = 1
3312 PRINT "THE PROGRAM DIALS A WASH
INGTON D.C.", "NUMBER. A MESSAGE
APPEARS ON THE", "MONITOR."
3313 INPUT "PLEASE LOG IN ";C$
3314 IF C$ = "HEL-ZXT.93.ZARF.1" THEN
3320
3317 IF G = 0 THEN 3329

```

```

3318 PRINT "INVALID LOGON CODE": RETURN

3320 PRINT "THE FOLLOWING MESSAGE AP
PEARS ON THE","MONITOR: THIS IS
THE UNITED STATES","DEFENSE DEP
ARTMENT'S TOP SECRET","ACCOUNT F
OR WEAPONS DEVELOPMENT AND","RAD
AR RESISTANT AIRCRAFT DATA. ALL
","INFORMATION IS CLASSIFIED."
3322 RETURN
3329 G = 2
3330 PRINT : PRINT "INVALID LOGON CO
DE - THE SCREEN GOES BLANK.
YOU HEAR FOOTSTEPS GRIMINSKI
LOOMS IN THE DORWAY WITH AN 8-M
M LUGAR IN HAND. YOU'D BETTER H
AVE BROUGHT THE PPK-3 PISTOL FRO
M THE DEPARTMENT OR YOU'RE F
INISHED!"
3331 GOSUB 300
3332 IF B = 0 THEN 3335
3333 GOSUB 400
3334 IF V# = "SHOOT" AND (N = 8 OR N
= 58) THEN 3805
3335 PRINT "IT'S HOPELESS! GRIMINS
KI FIRES....","YOU CRUMPLE TO TH
E FLOOR...END OF GAME": END
3399 REM --- VERB - DRINK ---
3400 IF N = 36 THEN PRINT "YOU ARE
POISONED, STAGGER TO THE PHONE",
"AND CALL THE AMBULANCE...END OF
GAME": END
3405 PRINT "YOU CAN'T DRINK "N#: RETURN
3499 REM --- VERBS - CHEW,EAT
3500 IF N = 39 OR N = 54 THEN PRINT
"YOU FOOL! THESE ARE CYANIDE CA
PSULES.", "YOU FALL WRITHING TO T
HE FLOOR AND", "DIE IN AGONY ...
END OF GAME.": END

```

```

3505 IF N = 45 THEN PRINT "YOU IDIO
      T! THE GUM IS A PLASTIC", "EXPLO
      SIVE. YOU HAVE JUST BLOWN", "YOU
      RSELF TO SMITHEREENS!! ... END O
      F GAME.": END
3510 PRINT "YOU CAN'T ";V#;" ";N#: RETURN

3599 REM --- VERB - UNWRAP ---
3600 IF N = 45 THEN PRINT "THE WRAP
      PER CONCEALS A TINY STRIP OF", "M
      ICROFILM.":OT(46) = 1: RETURN
3609 PRINT "IT DOESN'T HELP TO UNWRA
      P "N#: RETURN
3699 REM --- VERB - TALK ---
3700 IF N = 14 THEN PRINT "HE DOESN
      'T SPEAK ENGLISH!": RETURN
3705 PRINT "THAT WON'T HELP YOU": RETURN

3799 REM --- VERB - SHOOT ---
3800 IF N = 8 OR N = 14 OR N = 58 THEN
      3805
3804 PRINT "THAT WON'T HELP": RETURN

3805 FOR I = 1 TO IC
3806 IF CA(I) = 8 THEN 3811
3807 NEXT
3808 IF R = 6 AND G = 2 THEN PRINT
      "YOU DON'T HAVE THE PISTOL", "ANY
      THING ELSE TAKES TOO MUCH TIME."
      : GOTO 3335
3809 PRINT "YOU HAVE NOTHING TO SHOO
      T WITH": RETURN
3811 IF R = 3 AND (N = 8 OR N = 14) THEN
      PRINT "THE DOG BITES YOUR HAND!
      ": RETURN
3814 IF R < > 6 THEN PRINT "THAT J
      UST MAKES A BIG MESS!": RETURN
3815 IF G < > 2 THEN 3804

```

```

3816 PRINT "YOUR SHOT GRAZES HIS FOR
EHEAD. HE", "CRASHES TO THE FLOOR
R UNCONSCIOUS.", "YOU HAVE TIME T
O GATHER ADDITIONAL", "EVIDENCE T
O APPREHEND HIM.":G = 1
3817 RD$(6) = LEFT$(RD$(6),201) + "
GRIMINSKI IS LYING UNCONSCIOUS O
N THE FLOOR": RETURN
3899 REM --- VERB - UNLOCK ---
3900 IF N = 12 THEN 3920
3919 PRINT "YOU CAN'T ";V$;" A ";N$:
RETURN
3920 FOR I = 1 TO IC
3921 IF OA$(CA(I)) = "KEY" THEN 3930

3922 NEXT
3925 PRINT "YOU HAVE NOTHING TO USE
TO UNLOCK", " THE DOOR ": RETURN

3930 PRINT "UNLOCKED":OT(12) = 4: RETURN
3999 REM --- VERB - ON ---
4000 IF (N > = 31 AND N < = 33) THEN
4020
4010 PRINT "YOU CAN'T TURN ON A ";N$
: RETURN
4020 IF N = 31 THEN M = 137
4021 IF N = 32 THEN M = 57
4022 IF N = 33 THEN M = 111
4023 OD$(N) = LEFT$(OD$(N),M) + "ON
"
4025 PRINT "ON":OT(N) = 3: RETURN
4099 REM --- VERB - OFF ---
4100 IF (N > = 31 AND N < = 33) THEN
4110
4109 PRINT "YOU CAN'T TURN OFF A ";N
$: RETURN
4110 PRINT "OFF":OT(N) = 5
4111 IF N = 31 THEN M = 137

```

4112 IF N = 32 THEN M = 57
 4113 IF N = 33 THEN M = 111
 4114 OD\$(N) = LEFT\$(OD\$(N),M) + "OF
 F."
 4115 RETURN
 5097 DATA 24
 5098 DATA 10
 5099 DATA 7
 6000 DATA 59
 6001 DATA "YOU ARE IN YOUR OFFICE AT
 THE CIA. ON THE SHELVES ARE T
 OOLS YOU'VE USED IN PAST MISS
 IONS. AMBASSADOR GRIMINSKI'S A
 PARTMENT IS NORTH.",2,0,0,0
 6002 DATA "YOU ARE AT 14 PARKSIDE AV
 ENUE THE ENTRANCE TO AMBAS
 SADOR GRIMINSKI'S SMALL BUT
 ELEGANT BACHELOR APARTMENT. Y
 OU SEE A HEAVY WOODEN DOOR WITH
 A NOTICE ON IT WARNING OF A
 N ALARM SYSTEM.",0,0,1,0
 6003 DATA "THIS IS THE MARBLED FOYER
 OF THE AMBASSADOR'S APAR
 TMENT. THERE IS A TABLE IN
 THE CORNER. THE MASTER BEDROOM I
 S EAST, THE DRAWING ROOM IS NORT
 H AND A CLOSET WEST. A FIERCE
 DOG CHARGES TO ATTACK.",0,0,2
 ,0
 6004 DATA "YOU ARE IN THE FRONT HALL
 CEDAR CLOSET. HEAVY OVERCOATS A
 ND A TRENCHCOAT ARE HANGING U
 P. BOOTS ARE ON THE FLOOR AND O
 THER ITEMS ARE IN THE CORNER.",0
 ,3,0,0
 6005 DATA "YOU ARE IN THE DRAWING RO
 OM. A DESK IS HERE. A SOFA AND
 A COFFEE TABLE ARE IN FRONT OF

- THE FIREPLACE SET INTO THE P
 ANELED EAST WALL. THE DINING RO
 OM IS NORTH.",7,0,3,0
- 6006 DATA "YOU CAN SEE A MICROCOMPUT
 ER AND PHONE MODEM AND MONITOR
 ON A TABLE AGAINST THE EAST
 WALL OF THIS OVER-SIZED CLOSET. A
 PHONE IS BY THE COMPUTER. A CH
 AIR AND SHELVES ARE HERE.",0,
 0,0,5
- 6007 DATA "YOU ARE STANDING IN A SMA
 LL FORMAL DINING ROOM. THE
 TABLE SEATS SIX GUESTS.
 A SIDEBOARD WITH A TRAY ON IT I
 S AGAINST THE EAST WALL. THE KI
 TCHEN IS TO THE NORTH.",8,0,5,0
- 6008 DATA "YOU ARE IN THE APARTMENT
 KITCHEN WHICH SHIMMERS WITH POL
 ISHED CHROME APPLIANCE
 S AND BUTCHER BLOCK COUNTERS. A
 LONG CABINET ABOVE THE STAINLES
 S STEEL SINKS IS CLOSED.",0,0,7,0
- 6009 DATA "THIS IS AMBASSADOR GRIMI
 NSKI'S BEDROOM. A BED AND BEDSID
 E TABLE ARE HERE. A SAFE IS
 IN THE WALL ABOVE THE BUREAU.
 THE BATHROOM AND DRESSING AREA A
 RE TO THE NORTH.",10,0,0,3
- 6010 DATA "YOU ARE IN A COMBINED BA
 THROOM AND DRESSING AREA.
 THE AMBASSADOR'S CLOTHES
 ARE HANGING NEATLY ON RODS AND
 OPEN SHELVES HOLD TOWELS AND SWE
 ATERS. THE MEDICINE CABINET IS
 CLOSED.",0,0,9,0
- 6011 DATA "YOU'LL NEED SOME TOOLS TO
 GET INTO THE APARTMENT."

6012 DATA "MAYBE YOUR TOOLS WILL HELP YOU."

6013 DATA "SOMETHING FROM YOUR OFFICE COULD BE HELPFUL NOW."

6014 DATA "FIRST IMPRESSIONS CAN BE DECEIVING."

6015 DATA "THERE IS MORE HERE THAN MEETS THE EYE."

6016 DATA "RUNNING A PROGRAM IS ALWAYS INTERESTING."

6017 DATA "I CAN'T HELP YOU HERE."

6018 DATA "BE SUSPICIOUS OF ITEMS IN SMALL BOTTLES."

6019 DATA "THINGS ARE OFTEN NOT WHAT THEY SEEM."

6020 DATA "DON'T OVERLOOK THE OBVIOUS."

6021 DATA "HELP", "QUIT", "INVENTORY", "LOOK", "TIME", "SCORE", "RESTART"

6031 DATA "LOOK", "GET", "TAKE", "GO", "CRAWL", "WALK", "OPEN", "READ", "DROP", "CALL", "UNSCREW", "SPRAY", "PUSH", "LOAD", "RUN", "DRINK", "EAT", "CHEW", "UNWRAP", "TALK", "SHOOT", "UNLOCK", "ON", "OFF"

7001 DATA "NORTH", "IT DOESN'T HELP", 100, 0, 0, 4

7002 DATA "EAST", "IT DOESN'T HELP", 100, 0, 0, 4

7003 DATA "SOUTH", "IT DOESN'T HELP", 100, 0, 0, 4

7004 DATA "WEST", "IT DOESN'T HELP", 100, 0, 0, 4

7005 DATA "SHELVES", "SHELVES FOR WEAPONS AND TOOLS LINE THE WALL NEXT TO YOUR DESK. THERE ARE NUMEROUS ITEMS WHICH MAY HELP Y

- OU ON YOUR ASSIGNMENT.",1,6,0
,3
- 7006 DATA "SCREWDRIVER","AN ALL-PURPOSE SCREWDRIVER WITH COLLAPSIBLE HANDLE.",1,7,0,1
- 7007 DATA "BOMB","A MARK MX HIGH-INTENSITY SMOKE BOMB",1,8,0,1
- 7008 DATA "PISTOL","AN AUTOMATIC PPK-3 PISTOL",1,9,0,1
- 7009 DATA "KEY","A SKELETON KEY",1,10,0,1
- 7010 DATA "DRUG","A SMALL CAN OF INSTANT-KNOCKOUT DRUG",1,11,0,1
- 7011 DATA "GUN","A MARK 3K HARPOON GUN WITH GRAPPLE AND LINE.",1,12,0,1
- 7012 DATA "DOOR","THE HEAVY DOOR IS PAINTED BLACK. A BRASS KEY HOLE AND DOORKNOB ARE HERE. YOU CAN SEE THE CIRCULAR HOLES ON EITHER SIDE OF THE DOOR WHICH MUST RADIATE AN ELECTRONIC ALARM BEAM.",2,13,0,5
- 7013 DATA "ALARM","THE ALARM IS SCREWED INTO PLACE.",2,14,0,5
- 7014 DATA "DOG","THE SAVAGE DOBERMAN LEAPS TOWARD YOU WITH BARED FANGS. HE WILL NOT LET YOU PAST HIM.",3,15,0,4
- 7015 DATA "TABLE","THE VENETIAN FRONT HALL TABLE HAS A TORTOISE SHELL LETTER TRAY ON IT FOR BUSINESS CARDS AND MAIL. THERE IS A LETTER IN THE TRAY.",3,16,0,1
- 7016 DATA "LETTER","THIS IS APPARENTLY A TELEPHONE BILL THAT HAS BEEN PAID AND IS BEING SENT TO THE TELEPHONE COMPANY",3,17,0,1

- 7017 DATA "UMBRELLA", "THERE IS A BLACK BUSINESSMAN'S UMBRELLA WITH A POINTED END.", 4, 18, 0, 1
- 7018 DATA "BRIEFCASE", "THERE IS A BLACK LEATHER BRIEFCASE WITH A COMBINATION LOCK.", 4, 0, 0, 1
- 7019 DATA "DESK", "THE LARGE OAK DESK HAS A BLOTTER AND PENSET ON IT. A PHONE IS HERE. A BLANK NOTE PAD IS BY THE PHONE. THE DESK HAS A PIGEONHOLE AND ONE DRAWER IN IT.", 5, 0, 0, 1
- 7020 DATA "PAD", "ALTHOUGH THE NOTE PAD IS BLANK YOU CAN SEE THE INDENTATION OF WRITING ON IT.", 5, 0, 0, 1
- 7021 DATA "DRAWER", "THIS IS A STANDARD PULL DESK DRAWER.", 5, 0, 0, 4
- 7022 DATA "PIGEONHOLE", "THE PIGEONHOLE HAS A PAID BILL IN IT.", 5, 0, 0, 4
- 7023 DATA "BILL", "THE BILL IS FROM THE TELEPHONE COMPANY.", 5, 0, 0, 1
- 7024 DATA "PHONE", "THIS IS A BEIGE PUSHBUTTON DESK PHONE.", 5, 25, 0, 4
- 7025 DATA "NUMBER", "THE TELEPHONE NUMBER IS PRINTED ON THE BASE.", 5, 0, 0, 4
- 7026 DATA "PANEL", "THE PANELS ARE TONGUE-IN-GROOVE. ONE OF THE PANELS SEEMS MORE WORN THAN THE OTHERS.", 5, 0, 0, 4
- 7027 DATA "SHELVES", "THERE ARE SOFTWARE PROGRAMS AND BLANK DISKS AND MANUALS ON THE SHELVES.", 6, 0, 0, 4
- 7028 DATA "PROGRAM", "ONE PROGRAM IS FOR COMMUNICATING WITH THE U.S

DEFENSE DEPARTMENT'S MAIN
 FRAME COMPUTER.",6,0,10,5

7029 DATA "PHONE","THIS IS A STANDAR
 RD DESK-TYPE DIAL TELEPHON
 E. THE RECEIVER IS SET INTO A
 MODEM.",6,30,0,4

7030 DATA "NUMBER","THE TELEPHONE N
 UMBER IS PRINTED ON THE BASE.",
 6,0,0,1

7031 DATA "COMPUTER","THIS IS A STAN
 DARD BUSINESS TYPE OF MICRO
 COMPUTER WITH A KEYBOARD AND A
 PROGRAM IN ONE OF THE DISK DRI
 VES. THE ON/OFF SWITCH IS OFF."
 ,6,0,0,5

7032 DATA "MONITOR","THIS IS A HI-RE
 S COLOR MONITOR. THE ON/OFF
 SWITCH IS OFF.",6,0,0,5

7033 DATA "MODEM","THE PHONE MODEM I
 S ONE THAT CAN USE AN AUTOMATIC
 DIALING COMMUNICATIONS P
 ROGRAM. THE ON/OFF SWITCH IS OF
 F.",6,0,0,5

7034 DATA "TRAY","THE SILVER TRAY HO
 LDS A DECANTER PARTIALLY
 FILLED WITH CLARET.",7,0,0,1

7035 DATA "DECANTER","THE DECANTER I
 S OF ETCHED CRYSTAL. IT PROBAB
 LY HOLDS SOME CLARET",7,0,0,1

7036 DATA "CLARET","AN AMBER LIQUID"
 ,7,0,0,1

7037 DATA "CABINET","THIS IS A FAIR
 LY STANDARD KITCHEN CABINE
 T.",8,0,0,4

7038 DATA "BOTTLE","A BOTTLE OF CAP
 SULES ARE HERE.",8,39,0,2

7039 DATA "CAPSULE","THE CAPSULES

- ARE ELONGATED AND HAVE A SLIGHT AROMA OF BURNT ALMONDS.",8,0,0,1
- 7040 DATA "TABLE", "THE BEDSIDE TABLE HAS A PHONE ON IT. A PIECE OF PAPER AND A LAMP ARE HERE.",9,0,0,3
- 7041 DATA "PHONE", "THERE IS A NUMBER PRINTED ON THE PHONE.",9,0,0,4
- 7042 DATA "PAPER", "A PIECE OF MONOGRAMMED WRITING PAPER",9,43,0,1
- 7043 DATA "COMBINATION", "THERE IS A COMBINATION WRITTEN ON IT.",9,0,0,4
- 7044 DATA "SAFE", "THIS IS A STANDARD COMBINATION SAFE.",9,0,0,4
- 7045 DATA "GUM", "A PACK OF STICK TYPE PEPPERMINT GUM. EACH STICK IS WRAPPED IN PAPER.",9,0,0,2
- 7046 DATA "MICROFILM", "THE MICROFILM HAS BEEN DEVELOPED BUT YOU CAN'T SEE IT WITHOUT SPECIAL EQUIPMENT. NEVERTHELESS IT'S PRETTY CERTAIN WHAT YOU HAVE FOUND.",9,0,10,2
- 7047 DATA "SHELVES", "A VERY SOPHISTICATED CAMERA IS ON ONE OF THE SHELVES.",10,0,0,4
- 7048 DATA "CAMERA", "THIS CAMERA IS USED TO PHOTOGRAPH DOCUMENTS ON MICROFILM.",10,0,10,1
- 7049 DATA "CABINET", "THIS IS A LARGE MIRRORED BATHROOM CABINET.",10,0,0,4

- 7050 DATA "BUREAU", "A WALL SAFE IS
SET INTO THE WALL ABOVE THE LOW
MAHOGANY CARVED BUREAU.", 9, 0, 0,
3
- 7051 DATA "BOTTLES", "BOTTLES OF
FIXER AND PHOTOFLO ARE ON THE S
HELVES." , 10, 52, 0, 2
- 7052 DATA "TANK", "THERE IS A FILM
DEVELOPING TANK AND A FILM APR
ON AND TANK COVER HERE TOO.", 10,
0, 0, 2
- 7053 DATA "HEADQUARTERS", "HEADQUART
ERS", 100, 0, 0, 4
- 7054 DATA "CAPSULES", "THE CAPSULES A
RE ELONGATED AND HAVE A SLIGHT
AROMA OF BURNT ALMONDS.", 8, 0, 0,
1
- 7055 DATA "SIDEBOARD", "A LARGE ORNAT
E SIDEBOARD WITH A BEVEL
ED GLASS MIRROR DOMINATES THE
EAST WALL.", 7, 34, 0, 4
- 7056 DATA "NUMBER", "THE NUMBER IS
PRINTED ON THE PHONE", 9, 0,
0, 1
- 7057 DATA "PAPER", "THE NUMBERS 2-4
-8 ARE WRITTEN ON A PIECE OF PAPE
R ON THE TOP OF THE DRAWER.", 5, 0
, 0, 2
- 7058 DATA "GRIMINSKI", "THE WHITE-HA
IRED MAN IS DRESSED IN EVEN
ING CLOTHES", 6, 0, 0, 4
- 7059 DATA "CORNER", "YOU ARE LOOKING
AT THE CORNER OF THE CLOSET."
, 4, 17, 0, 4

Appendix D

Sample Run for CIA

IRUN

YOU ARE IN YOUR OFFICE AT THE CIA. ON THE SHELVES ARE TOOLS YOU'VE USED IN PAST MISSIONS. AMBASSADOR GRIMINSKI'S APARTMENT IS NORTH.

NOW WHAT ? LOOK SHELVES

SHELVES FOR WEAPONS AND TOOLS LINE THE WALL NEXT TO YOUR DESK. THERE ARE NUMEROUS ITEMS WHICH MAY HELP YOU ON YOUR ASSIGNMENT.

AN ALL-PURPOSE SCREWDRIVER WITH COLLAPSIBLE HANDLE.

A MARK MX HIGH-INTENSITY SMOKE BOMB

AN AUTOMATIC PPK-3 PISTOL

A SKELETON KEY

A SMALL CAN OF INSTA-KNOCKOUT DRUG

A MARK 3K HARPOON GUN WITH GRAPPLE AND LINE.

NOW WHAT ? GET SCREWDRIVER

TAKEN

NOW WHAT ? GO NORTH

YOU ARE AT 14 PARKSIDE AVENUE THE ENTRANCE TO AMBASSADOR GRIMINSKI'S SMALL BUT ELEGANT BACHELOR APARTMENT. YOU SEE A HEAVY WOODEN DOOR WITH A NOTICE ON IT WARNING OF AN ALARM SYSTEM.

NOW WHAT ? OPEN DOOR

THE DOOR IS LOCKED.

NOW WHAT ? UNLOCK DOOR

YOU HAVE NOTHING TO USE TO UNLOCK
THE DOOR

NOW WHAT ? GO SOUTH

YOU ARE IN YOUR OFFICE AT THE CIA. ON
THE SHELVES ARE TOOLS YOU'VE USED IN
PAST MISSIONS. AMBASSADOR GRIMINSKI'S
APARTMENT IS NORTH.

NOW WHAT ? TAKE KEY

TAKEN

NOW WHAT ? GO NORTH

YOU ARE AT 14 PARKSIDE AVENUE THE
ENTRANCE TO AMBASSADOR GRIMINSKI'S
SMALL BUT ELEGANT BACHELOR APARTMENT.
YOU SEE A HEAVY WOODEN DOOR WITH A
NOTICE ON IT WARNING OF AN ALARM SYSTEM.

NOW WHAT ? UNLOCK DOOR

UNLOCKED

NOW WHAT ? OPEN DOOR

YOU DIDN'T DISCONNECT THE ALARM.
IT GOES OFF AND THE POLICE COME AND
ARREST YOU...END OF GAME.

YOU ARE A CIA AGENT

Your mission is to prove that the Russian Ambassador to the U.S. is a spy. But don't get caught! Between you and the evidence are some deadly hazards and booby-trapped clues.

Once you master the game, your next challenge is to create an even better adventure with your own ideas. *BASIC Fun with Adventure Games* will show you how. Using the simple instructions, you can add new twists to this game, or use it as a model for creating your own adventure game. You'll learn how to develop a plot, define the solution, and determine all the objects, obstacles, and hazards—and then how to translate your ideas into a BASIC program.

With *BASIC Fun with Adventure Games* the possibilities are endless. All you need is imagination and the know-how to put your ideas into a BASIC program!

AVON



CAMELOT

PUBLISHED BY AVON BOOKS

009-012



ISBN 0-380-87486-5