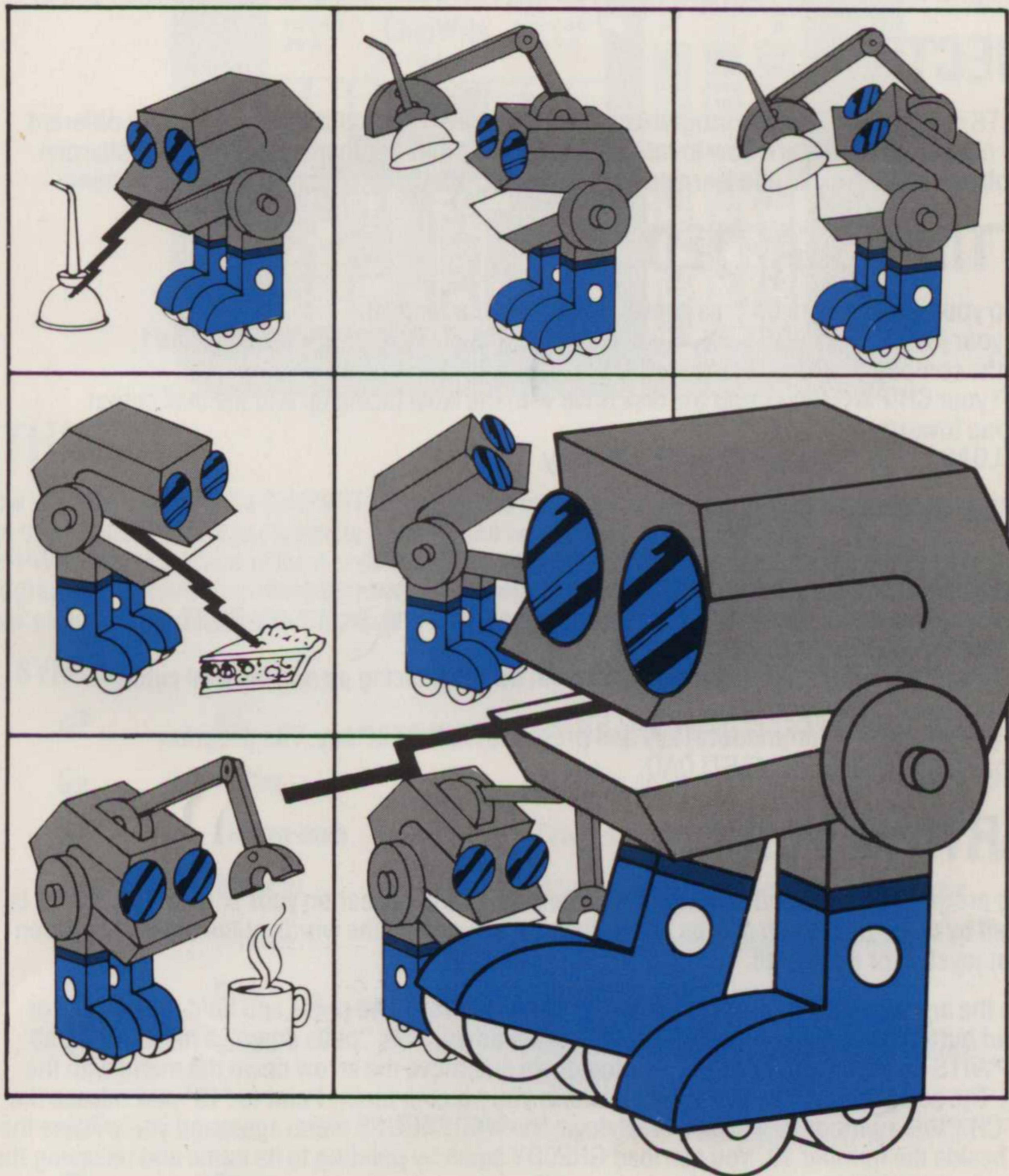


EDYX™
COMPUTER SOFTWARE

CHIPWITS

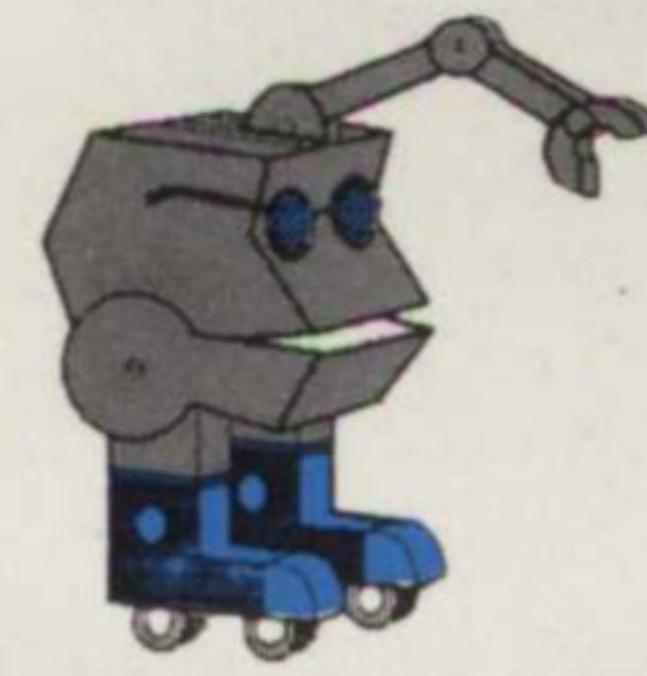
INSTRUCTION MANUAL

For the Commodore 64™



INTRODUCTION

Hello there, human. My name is Greedy and I am a CHIPWIT—a robot with a brain you can program with pictures. You can teach me to explore, to sing, to compete, to eat pie and coffee (yum), even to dance. But here's the catch—I'm only as smart as you make me.



Sixteen CHIPWITS live on this disk and you can teach each of us to play any of eight different games. Each game is an adventure with 4 to 49 rooms filled with good things and bad things. Program me to grab disks and oilcans for points, and make sure I eat plenty of pie and coffee to keep up my energy. Be sure to keep me away from those electro-crabs; they're always following me and draining my energy. I love to zap them and sometimes I get points for doing it. Bouncers are a nuisance, too. Bombs are worst of all—one BOOM, and it's all over.

I can move, feel things, see things, smell things, remember things, grab things, and zap things. My language, IBOL (Icon Based Operating Language), has a picture chip for each of these actions and it's up to you to arrange them in a program that makes me smart. Anyone can build a CHIPWIT that wanders in circles bumping into walls, but it takes practice to plug my chips into a program that lets me score big points game after game.

But enough of this chitchat. I can smell some pie and I sure am hungry. Greedville, here I come.

OBJECTIVE

CHIPWITS is a game where you program robots to explore rooms and mazes filled with different kinds of objects. As you learn how to make the robots "think" for themselves, you will sharpen your problem-solving skills and learn some of the basic principles of computer programming.

GETTING STARTED

- Set up your Commodore 64™ as shown in the Owner's Manual.
- Plug your joystick into PORT #2. If you are using a Koala Pad, plug it into PORT #1.
- Turn the computer and the disk drive ON.
- Insert your **CHIPWITS** disk into the disk drive with the label facing up and the oval cutout pointing towards the back.
- Type **LOAD "*,8,1** and press the **RETURN** key.

With EPYX FASTLOAD™ Cartridge

- Set up your Commodore 64™ as shown in the Owner's Manual.
- Insert the **FASTLOAD**™ Cartridge into the cartridge slot of your computer.
- Plug your joystick into PORT #2. If you are using a Koala Pad, plug it into PORT #1.
- Turn the computer and the disk drive ON.
- Insert your **CHIPWITS** disk in the disk drive with the label facing up and the oval cutout pointing towards the back.
- Hold down the **C=** (Commodore) key and press the **RUN/STOP** key. The program will load twice as fast with FASTLOAD.

STARTING PLAY

After the program has finished loading, the game display will appear on your screen. CHIPWITS is controlled by using pull-down menus. Move the arrow cursor to the words at the top of the screen with your joystick or Koala Pad.

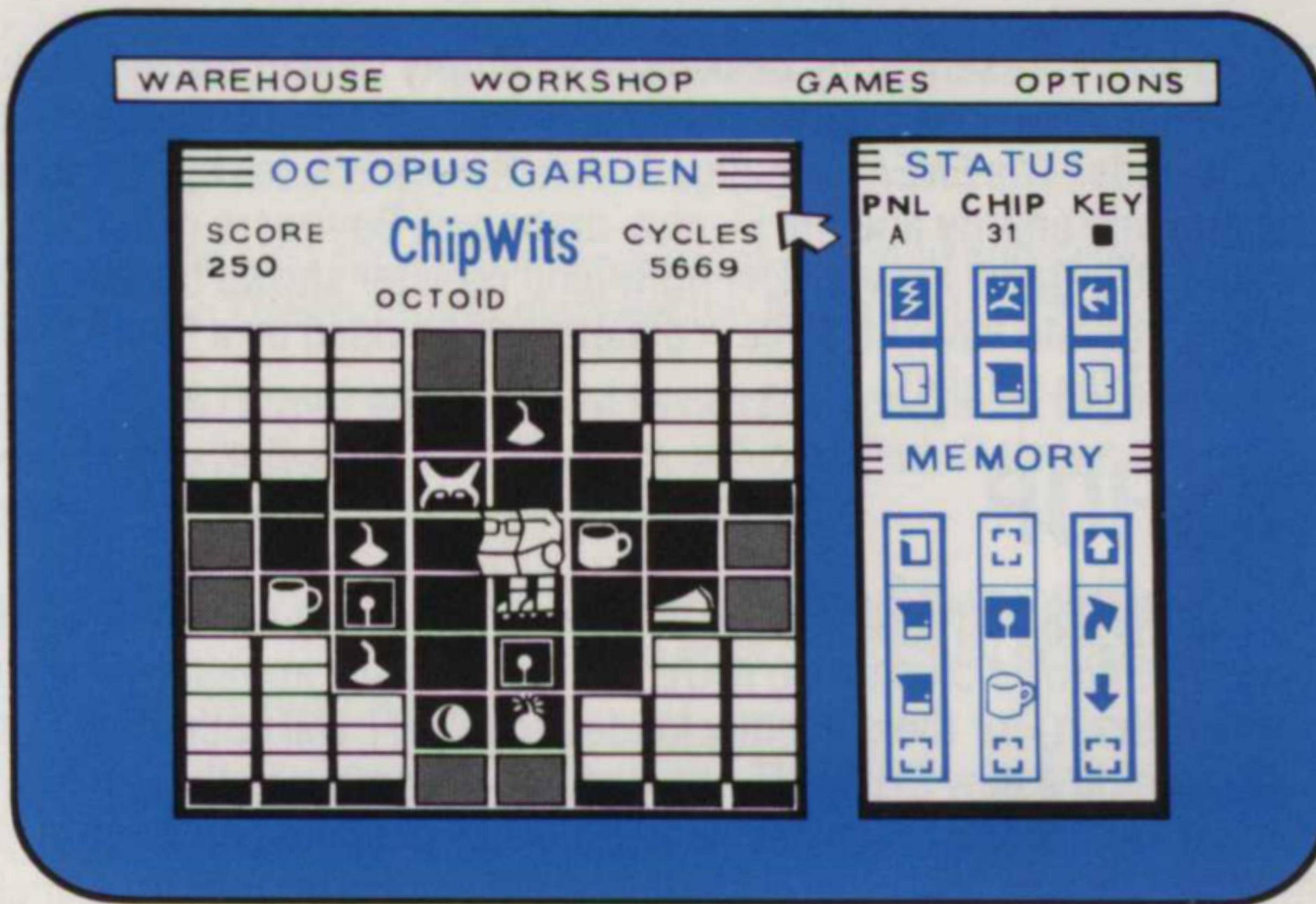
Position the arrow so that it points to the word "**WAREHOUSE**" and press and hold the joystick or Koala Pad button. Be sure to keep holding the button down. This "pulls down" a menu listing all the CHIPWITS on your disk. Keep your button down and move the arrow down the menu with the joystick. The selections on the menu turn white as you point to them. Point to "**10**" and release the button. CHIPWIT number 10 will load. Pull down the **WAREHOUSE** menu again and you will see the marker beside the number 10. You can load GREEDY again by pointing to its name and releasing the button.

To start a game, pull down the “**GAMES**” menu. At the top are two game options and at the bottom are the eight games. Some of the games are in blue. These are games in which GREEDY does well. Load **GREEDVILLE** by pointing to it and letting go of the button. Now select **START MISSION** from the **GAMES** menu and watch GREEDY play in GREEDVILLE.

GAME PLAY

GREEDVILLE has 4 rooms containing good things; GREEDY will wander around and pick them up. As you watch GREEDY play in GREEDVILLE, here are some things you should observe:

- GREEDY gets points for eating disks and oilcans.
- When GREEDY eats pie and coffee, his energy level increases. The energy gauge is on the right side of the screen.
- Every time GREEDY does something, cycles are used up. When the CYCLES count reaches zero, the game will end.
- Each room has a wall around it with a few doors opening into other rooms. Whenever GREEDY moves, feels, or looks into the next room that room appears.



“THINGS”

Now you have observed a CHIPWIT following the instructions in its program and playing a game. You can see the CHIPWIT’s program by going to the workshop (see WORKSHOP). The program tells the CHIPWIT how to behave when it sees or touches the things it encounters in each room. The CHIPWITS programming instructions use symbols to represent all good things, bad things and parts of rooms—here’s a list to acquaint you with things and their symbols.

SYMBOL	OBJECT	COMMENTS
pie	Pie	Pie is a good thing. CHIPWITS “eat” pie for energy.
coffee	Coffee	Coffee is good. CHIPWITS “eat” coffee for energy.
crab	Electro-crab	Crabs are bad. They can hurt CHIPWITS.
bouncer	Bouncer	Bouncers are bad. They can drain energy.
oil can	Oil Can	CHIPWITS “eat” oil cans for points.
disk	Disk	CHIPWITS “eat” disks for points.
bomb	Bomb	Bombs are bad. Touching a bomb destroys CHIPWITS.
wall	Wall	Running into a wall damages CHIPWITS.
floor	Floor	When no object is present, CHIPWITS are able to see or feel the floor.



Door

Doors allow CHIPWITS to pass from room to room. When a CHIPWIT is standing on a door square, the display changes to the next room only after the CHIPWIT looks, feels, grabs, or zaps ahead into the room.

STATUS DISPLAY

You have seen that a CHIPWIT uses energy when it is playing a game. CHIPWITS are equipped with an energy gauge that displays the robot's energy level at all times. CHIPWITS also have two other gauges, or registers—a range finder and a damage meter—both of which appear in the status display.

The window on the right side of the screen contains the status display. Three symbols are displayed. From left to right, they are **DAMAGE**, **ENERGY**, and **RANGE**. Below each symbol is an object that represents a number.

All numbers in CHIPWITS are represented by beakers—an empty beaker means 0 and a full one means 7. The **DAMAGE** meter shows how damaged the CHIPWIT is from bumping into things. When this reaches full the game ends. The **ENERGY** gauge shows how much energy is left in the CHIPWIT. Almost every action uses some energy and the Electro-crabs and Bouncers can drain energy as well. When this reaches “empty” the game is over. The righthand register shows the **RANGE** to the last object the CHIPWIT looked for and saw. All three registers can be used by a CHIPWIT’s program (see USING THE REGISTERS for details).

THE WORKSHOP

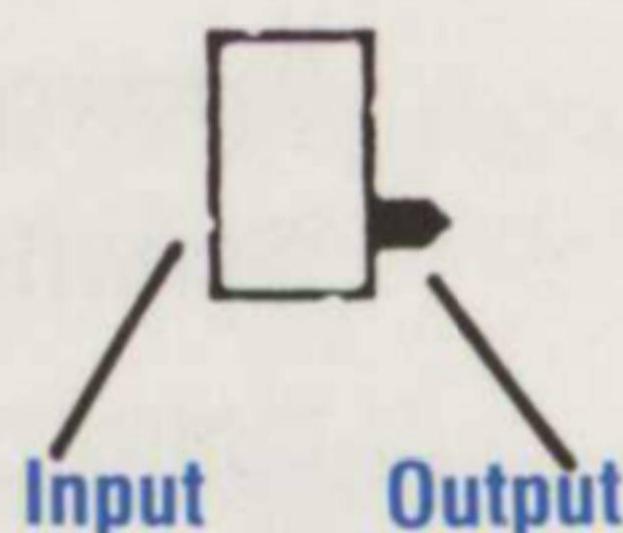
You can program your CHIPWITS in the **WORKSHOP**. To enter, just pull down the **WORKSHOP** menu and select **ENTER**. Several windows containing IBOL instructions will appear on the right side of the screen. The Main Panel (or Brain) for the currently loaded CHIPWIT will appear on the left side of the screen. If you are programming a new CHIPWIT, the panel may contain nothing more than a **GO** chip (a traffic light) in the upper left corner which designates the starting location for that panel. A white chip outline, called the **Chip Cursor**, will appear in the bottom right corner of the panel.

CHIPS AND PANELS

Every CHIPWIT needs a “program”, or set of instructions, to tell it what to do. The panel you see on the left side of your workshop screen contains a set of “chips” with instructions for your CHIPWIT. The instructions are written in a language called IBOL (Icon Based Operating Language).

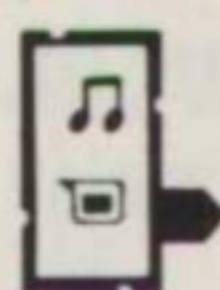
The program for a CHIPWIT consists of the set of chips on its **Main Panel** (or panel **A**) and nine **Subpanels** (labelled **B** through **J**). Each of these panels can hold up to forty different chips (arranged in a 5 by 8 matrix). The chips are instructions which are executed in an order determined by their positions and connections. Program control is passed from one chip to the next through **Output** wires. Each chip has an output wire pointing to the next chip to be executed.

Chip



Each chip must contain an IBOL instruction, or **Operator**. The instruction tells CHIPWIT what to do next, some examples of instructions are **Look**, **Feel** and **Move**. Some Operators also require that an additional component, or **Argument**, be included within the chip. For example, a chip instructing a CHIPWIT to **Sing** must also contain the number of the note to be sung (remember, the numbers inside a CHIPWIT are represented by beakers).

Operator



Argument

Some Operators, such as **Pick up** and **Zap**, need no Arguments. (See the quick reference card on the last page for a complete list of Operators and their Arguments.)

MAKING A CHIP

Using your joystick or Koala Pad, move the arrow cursor until it points to one of the empty chip locations on the **Workshop panel**. Press the joystick button or Koala Pad button to make the selection. The **Chip Cursor** will move to that location and you will be ready to make a chip.

Now, point to and select the desired **Operator** from the Operator Window on the right side of the screen. The Operator you selected will appear inside the chip you are constructing, and at least one Output will be attached.

Note: No Output wire will appear for the Loop and Boomerang Operators; two Output wires will appear for any branching Operator. (See OPERATORS.)

Next, if the Operator selected also requires an **Argument**, point to and select the desired Argument from the Argument Window. The computer displays the Arguments whenever you select an Operator.

Finally, if the direction of the Output wire(s) need to be changed, point to the Output wire to be moved and select it by pressing the button on your joystick or Koala Pad. Release the button, then select the new location by pointing to it and pressing the button again.

Note: If you move the Chip Cursor to a new socket before a chip is completed, that chip will simply "fall off" the panel. If you move the Chip Cursor to a socket currently occupied by a chip and select a new Operator, that chip will be reprogrammed.

MOVING CHIPS

Sometimes you need to move a chip. You can do this by pointing to the chip you'd like to move and holding down the button. While you're holding the button, move the cursor to the place you'd like the chip to go and release the button. If you drag a chip to a spot containing another chip, the old one will be thrown away.

DELETING CHIPS

To remove and discard a chip from any panel, first move the Chip Cursor to the location of the unwanted chip, and then hold down the button and drag the cursor off the panel. Release the button when the arrow is off the panel and the chip cursor disappears.

CHANGING PANELS

To select the Main Panel (**A**) or any of the Subpanels (**B** through **J**), point to and select one of the letters located directly above the workshop panel.

EDITING PANELS

Pull down the **WORKSHOP** menu. Options on this menu allow you to cut, copy, paste or clear the panel currently being displayed.

- The **Cut** option erases the current panel and saves it, enabling you to **Paste** the contents on another panel.
- To **Paste**, simply select another panel by pointing to a letter (**A** through **J**), then select **Paste** from the **WORKSHOP** menu.
- **Copy** allows you to copy a panel without erasing its contents. You can then **Paste** the copy on another panel.
- **Clear** simply clears the current panel, erasing its contents without storing them.

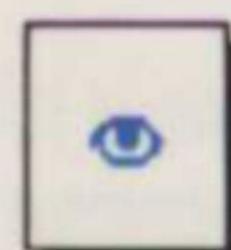
SAVING CHIPWITS

To save the current CHIPWIT, first pull down the **WORKSHOP** menu and select the **SAVE CHIPWIT** option. The area of the screen normally used to display a panel will be replaced with a "Name" window.

- If you don't wish to make any changes, simply point to the **OK** window and press the button on your joystick or Koala Pad.
- If you want to enter a new name for the CHIPWIT (up to 12 characters), just type the name on your keyboard.
- Next, select the names of the games for which the CHIPWIT was designed.
- When you're finished, click the **OK** window or press the **RETURN** key.

To bring a previously programmed CHIPWIT back into the Workshop, just pull down the **WORKSHOP** menu and select the name of the desired CHIPWIT.

OPERATORS



Look for...

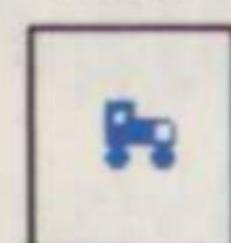
Safely concealed behind each CHIPWIT's polarized crystal eyeglasses are two sophisticated vision systems. The component under the right lens is used to seek and detect objects directly ahead. A CHIPWIT may look for any **Thing**, or the object on top of the **Thing Stack** (see MEMORY).



This chip, for example, instructs a CHIPWIT to look directly ahead for a Door.



If the CHIPWIT sees a door, the TRUE branch will pass control to the chip on the right. If there is no Door ahead, or any other object is seen, the FALSE branch will pass control to the chip located directly below.



Move or Turn

What might first appear to be a pair of high-topped roller skates is, in fact, a rectilinear magnetic drive system that each CHIPWIT uses to move forward and backward, or to turn right and left in 45 degree increments. A CHIPWIT will try to execute any **Move** (or the Move on top of the **Move Stack**)—even if it means running into something.



The following chip, for instance, instructs a CHIPWIT to turn 45 degrees to the right. Control is then passed to the chip on the right.



Note: If a CHIPWIT tries to move into some **Things**, damage may occur.

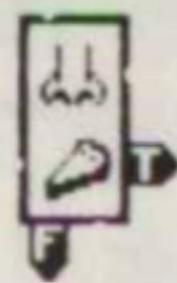


Smell for...

Many of the objects a CHIPWITS encounters in its adventures will have a specific odor. (Needless to say, Pie smells much better than any Bouncer!) So to help a CHIPWIT quickly detect the presence of **Things** in a room, state-of-the-art olfactory sensing pads have been built into the CHIPWIT's ankles. By executing just one chip, a whole room can be sniffed for any **Thing** (or the object on top of the **Thing Stack**).



If your CHIPWIT needed a little energy, for example, you might include a chip like this in one of its panels.



When this chip is executed, the CHIPWIT smells for Pie in the current room. Here, control is then passed to the chip on the right if Pie is detected, or to the chip below if not.

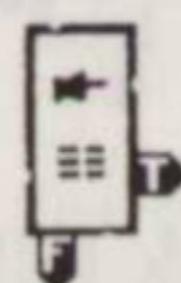


Feel for...

After a few head-on collisions with a wall, any CHIPWIT can tell you that it's important to feel for **Things** while exploring. So, a sensor that pops out of its head enables each CHIPWIT to "feel" for any **Thing** (or the object on top of the **Thing Stack**) directly ahead.

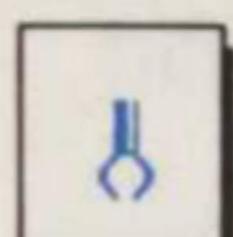


This chip, for example, tells a CHIPWIT to feel for a Wall in front of it.



In this case, if a Wall is detected, the TRUE branch transfers control to the chip on the right. If no Wall is touched, or some other **Thing** is "felt", the FALSE branch passes control to the chip directly below.

Note: Feeling for Bombs can be disastrous.



Pick up

A CHIPWIT can pick up most **Things** with its electro-hydraulic manipulator, which automatically appears through the cranial access port when needed. Picking up useful objects can have beneficial effects. Pie and Coffee, for instance, will increase a CHIPWIT's energy level. But trying to grab Electro-Crabs or Bombs can really hurt!

The manipulator is activated with this chip.



Only the object directly in front of a CHIPWIT can be picked up.



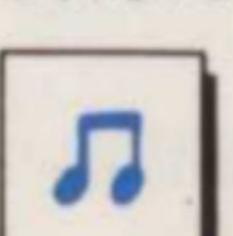
Zap

Every CHIPWIT has a plasma beam generator to defend itself from Electro-crabs and Bouncers. This generator, also called the "**Zapper**", automatically appears through the cranial access port when needed. In some games, points are awarded for zapping bad things. In one game, however, you'll lose points for destroying things with the zapper.

The zapper is activated with this chip.



This instructs the CHIPWIT to zap the first object directly in its path, even if it is across the room. Good objects will be zapped if you're not careful, so it's usually a good idea to look for bad objects before you use the zapper.

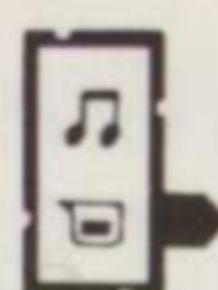


Sing

You can talk to a CHIPWIT through the keyboard, but your CHIPWIT can also "talk" back. In fact, it can sing any one of eight different notes on command. What talent! A **Number** is used to tell the CHIPWIT just what note to sing.

Notes	0	1	2	3	4	5	6	7
	C	D	E	F	G	A	B	C

For example, this chip tells a CHIPWIT to sing an **F**.



Singing doesn't seem to have any effect on most **Things**, although Electro-Crabs have been known to become quite angry.

Coin Flip

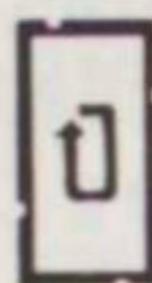
In several of the games, you may need to have your CHIPWIT "flip a coin" to decide what to do next. Say you want to find an object, but you don't know where it is—it could be to the left, but on the other hand it might be to the right. Use the Coin Flip instruction to make the decision.



When this chip is executed, the CHIPWIT will pick TRUE or FALSE at random. Half of the time, control will pass to the TRUE branch. The other half of the time, the program will follow the FALSE branch.

Loop

When your program reaches the end of a panel, you'll need a way to switch control back to the start chip. The Loop instruction tells the CHIPWIT to go back to the start chip of the current panel and repeat the panel. This instruction can be used on the Main Panel or any Subpanel, and it looks like this:



When a CHIPWIT encounters this instruction, control will transfer back to the start chip of the current panel and the program will continue from there.

Junction

As you program your CHIPWITS, you'll occasionally need a way to connect two chips that aren't next to each other. The junction chip can be placed anywhere on any panel to link parts of your program together. Like other chips, the pointer can be changed to pass control in any of four directions: left, right, up or down.



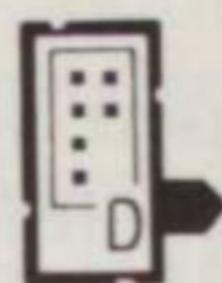
When a CHIPWIT program encounters this chip, control will pass automatically to the next chip on the right. Junction chips may be placed next to other junction chips.

Subpanel

Every CHIPWIT contains one Main Panel (**A**) and nine additional Subpanels (**B** through **J**). You can program a CHIPWIT that uses only the Main Panel (GREEDY is a good example), but eventually you'll want to create more complicated programs that use more than one panel. The Subpanels give you plenty of extra room for programming. There are just two rules to remember:

- 1) A Subpanel can be "called", or accessed, **only** from the Main Panel. Subpanels cannot call each other—you can't access a Subpanel from another Subpanel.
- 2) When a Subpanel is finished, control **always** returns to the Main Panel and the program continues at the location after the chip that called the Subpanel.

Logically, each Subpanel might be used to perform just a single activity—searching a room, counting objects or moves, or perhaps singing a special tune. But whatever its function, a Subpanel is accessed from the Main Panel with a chip like this.

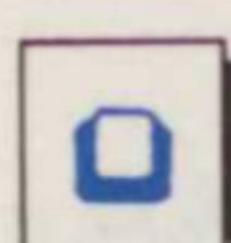
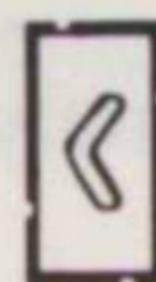


In this example, the program will jump to the beginning of Subpanel **D** and when finished (see “Boomerang” below), control will be returned to the Main Panel and the chip immediately on the right.



Boomerang

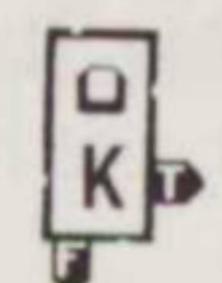
The Boomerang instruction is used only on Subpanels, when you want the program to return to the Main Panel. When a CHIPWIT program encounters the Boomerang, it jumps back to the Main Panel and continues at the location after the chip that called the Subpanel. A Subpanel can have more than one Boomerang chip.



Compare key

A CHIPWIT can be programmed for every situation he'll encounter, but there may be times when you'll want to control his actions yourself. The Compare key instruction allows you to have manual control. In fact, you can even program a CHIPWIT to be operated completely from the keyboard.

This chip, for example, tells a CHIPWIT to find out if the letter **K** has been pressed.



In this example, if the letter **K** has been pressed, the TRUE branch transfers control to the chip on the right. If that letter has not been pressed, the FALSE branch passes control to the chip directly below.

ADVANCED PROGRAMMING

Each of your CHIPWITS has a memory. To the right of the game room is a window into the CHIPWIT's brain. At the top of the window you can see the numbers of the current panel and chip being executed, as well as a letter showing the last key pressed on the keyboard. The bottom of the window is a display of three kinds of memory.

In one game, your CHIPWIT may need to remember the moves it has made since entering a room to quickly return to a door. In still another game, your CHIPWIT may need to count up to a hundred objects. So to enable each CHIPWIT to easily store, retrieve, and manipulate **Numbers**, **Moves**, and **Things**, each has been provided with three “pockets,” called **Stacks**.

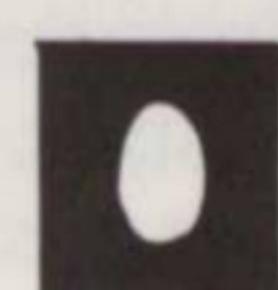
Stacks



Numbers



Moves



Things

USING MEMORY

An argument is saved or “pushed” on the top of a Stack using the **Save Thing**, **Save Move**, or **Save Number** Operator.



Save Number



Save Move



Save Thing

Each of the Stacks will hold up to **256** arguments. When an argument is placed on the top of any

Stack, the arguments already in that Stack (if any) are pushed down, and the new argument will then occupy the top position. If more than 256 arguments are pushed on a Stack, the argument on the bottom will simply “fall out” and be “forgotten”.

Likewise, an argument can be removed or “popped” from any Stack using the **Pop** Operator. Only the top argument is discarded, and the remaining arguments in the Stack, if any, move “up” one position.

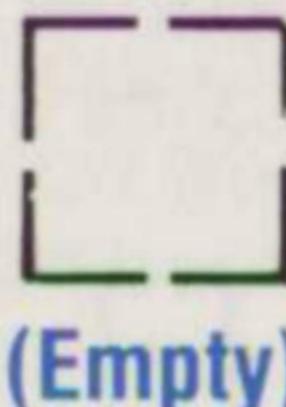


Pop Move
Stack



Pop Thing
Stack

A special symbol is used to indicate that a Stack is empty. This particular argument is very useful in comparisons (see below), but it **cannot** be pushed onto or popped from any of the Stacks.



(Empty)

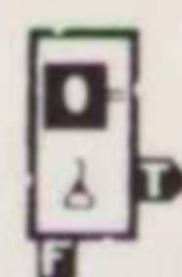
COMPARISONS

If you need to look at something you saved on a stack, use a “compare” instruction. Say you need to find out if the number on top of the Number Stack is equal to zero. To do that, you would use the **Compare Number** operator with the number zero as the argument.

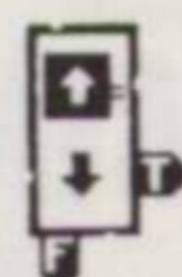


In this case, if the number at the top of the Number Stack is zero, the TRUE branch transfers control to the chip on the right. If the number is not zero, the FALSE branch passes control to the chip directly below.

Using the **Compare Thing**, **Compare Move**, and **Compare Number** Operators, a CHIPWIT can check an argument against the top of any Stack, and branch (TRUE) if they are identical (i.e., Equal To).



Compare
Thing

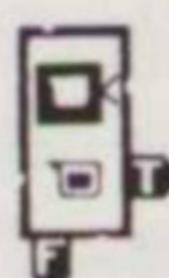


Compare
Move

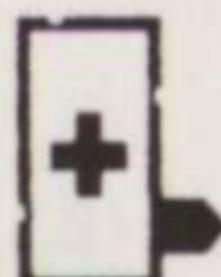


Compare
Number

A special Operator, **Compare Number Less Than**, branches (TRUE) if the number on the Number Stack is less than the Number argument.



Two additional Operators, **Increment** and **Decrement**, allow you to perform simple arithmetic. You can use them to add or subtract from the number on top of the Number Stack. **Increment** adds one to the number on top of the Number Stack. **Decrement** subtracts one from the number at the top of the Number Stack. However, if an attempt is made to increment the top of the Number Stack beyond seven, it “rolls around” and becomes a zero. Likewise, if it is decremented below zero, it becomes seven.



Increment



Decrement

Each Stack is dynamically updated when a Mission is in progress and its top three arguments can be viewed in the **Status/Memory Window**.

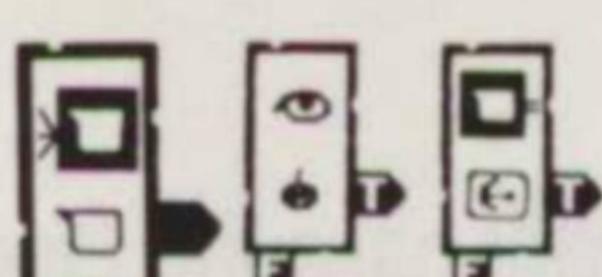
CYCLES AND ENERGY

Every CHIPWIT action consumes cycles and energy. Your CHIPWITS always start each game with a full energy tank and a certain number of cycles. Energy can be replenished by eating pie or coffee; cycles cannot be replenished—once used, cycles are lost. The number of cycles available to a CHIPWIT vary from 6,000 in Greedville to 20,000 in Mystery Matrix, Memory Lanes and Octopus Garden. If a CHIPWIT runs out of cycles or energy, the game will end. Refer to the **IBOL Reference Card** on the last page for a list of the number of cycles consumed by each instruction.

USING THE REGISTERS

Any instruction that saves, compares or uses a number can incorporate one of the registers as an argument instead. This becomes particularly useful if you need to keep track of a CHIPWIT's damage or energy levels, or if you try to program a CHIPWIT for Boomtown (where you need to use the range finder to tell you the distance to bombs).

The following sequence of chips, for example, compares the range to a bomb with the number on top of the number stack.



If you push the number **0** onto the number stack, then look for a bomb, the compare instruction will tell you whether the range register has dropped to **0** (meaning the CHIPWIT is right next to a bomb).

OPTIONS

The options menu allows you to **quit** the game or turn on the **Debug Mode**. Pull down the **OPTIONS** menu. Some of the items are red. These items are disabled; they can't be selected.

DEBUGGING

When you finish writing a CHIPWIT program and you start a game to see how the program works, you'll probably see ways you can improve the program. And you'll want to follow the program's execution to see how you can improve it. This is when you use the Debug mode.

Start a game, then pull down the **OPTIONS** menu. Choose **DEBUG ON**. This gives you a look at your CHIPWIT's program as it operates. The white cursor outlines the chip that is executing. Try some of the other OPTIONS menu items; **SLOW** will allow you watch CHIPWIT perform slowly, **SEE ROBOT** displays the CHIPWIT on the debug screen. If you select **STEP**, GREEDY waits for you to press the button before the next chip executes. Select **DEBUG OFF** to see the game room again. You may turn on the Debug Mode as many times as you wish during a game.

GAMES

The following descriptions provide an overview of the games included in CHIPWITS. The Things that appear in each game and their respective point values are indicated.

Name: GREEDVILLE **Difficulty:** Easy

Objective: To pick up as many “good” Things as possible.

Description: Four confusingly connected rooms filled randomly with objects having only positive point values (“good” Things).

Point Assignments:

■ = 100 ▽ = 50

Cycles: 6,000

Name: CHIPWIT CAVES **Difficulty:** Easy

Objective: To pick up as many “good” Things as possible, and zap Electro-Crabs.

Description: Eight connected rooms (the walls are arranged to spell out the word CHIPWITS).

Point Assignments:

 = 100  = 50

 = 50

Cycles: 10,000

Name: DOOM ROOMS **Difficulty:** Average

Objective: To survive as long as possible by avoiding or zapping Electro-crabs and Bouncers.

Description: 12 interconnected rooms populated with only “bad” things (i.e., Electro-Crabs, Bouncers).

Point Assignments:

 = 100  = 250

Cycles: 10,000

Name: PEACE PATHS **Difficulty:** Average

Objective: To pick up as many “good” Things as possible without harming Electro-Crabs and Bouncers.

Description: Very long connected “hallway” with isolated rooms on either side containing randomly distributed “good” and “bad” Things.

Point Assignments:

 = 150  = -30
 = 20  = -60

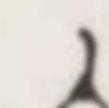
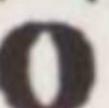
Cycles: 16,000

Name: MYSTERY MATRIX **Difficulty:** Hard

Objective: To discover the best pathway through the matrix by deciphering objects arranged as “signposts”. Wherever there are three objects lined up in a row, they form an arrow—the pair of identical objects are the shaft of the arrow, and the third object is the arrowhead.

Description: 100 connected rooms, each having four identically located doors.

Point Assignments:

 = 250  = 20
 = 150  = 40

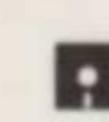
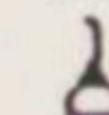
Cycles: 20,000

Name: MEMORY LANES **Difficulty:** Hard

Objective: To move from one end to the other within a maze as many times as possible.

Description: 10 room “horseshoe” maze populated with a variety of “bad” Things and one high-point Thing (disk) at either end.

Point Assignments:

 = 250  = 30
 = 30  = 60

Cycles: 20,000

Name: OCTOPUS GARDEN **Difficulty:** Extremely hard

Objective: To retrieve the “good” Things (disks) located at the ends of eight “arms” of the octopus maze.

Description: 44 rooms divided into eight hallways that connect only in the central room where the CHIPWIT starts the game.

Point Assignments:

 = 250	 = 10
 = 50	 = 20

Cycles: 20,000

Name: BOOMTOWN **Difficulty:** Extremely hard

Objective: To pick up disks while avoiding contact with bombs.

Description: A series of connecting rooms with clusters of bombs surrounding disks.

Point Assignments:

 = 250	 = 40
---	--

Cycles: 12,000

CONTINUING PLAY

You may pull down the menus and select menu items at any time during the course of play—even right in the middle of a game. A game can be started or ended as many times as you like. CHIPWITS even includes an option on the **GAME** menu which allows you to repeat playing a game—the **SERIES** option. When you select **SERIES**, the currently loaded CHIPWIT will start playing the currently loaded game and repeat playing it until you tell it to stop. When the first game ends, it will start another game. As the **SERIES** continues, an average score will be calculated and displayed at the end of each game.

GLOSSARY

Argument—A part of an IBOL instruction; the Argument serves as the object of the action to be performed by the CHIPWIT. For example, in the instruction “Look for pie”, **look** is the action to be performed and **pie** is the object (or Argument).

Beaker—A symbol representing a number from zero to seven; for example, an empty beaker represents zero and a full beaker represents the number seven.

Bouncer—A bad thing, or object, that spins in place and drains energy from any adjacent CHIPWIT.

Chip—A receptacle containing the parts of an IBOL instruction. Chips usually have one or two output wires to connect them to other chips.

CHIPWIT—A short, squat robot equipped with a programmable brain, sophisticated vision system, rectilinear magnetic drive and other mechanical systems designed for use in GREEDVILLE and other adventures.

Cranial access port—An opening on the portion of a CHIPWIT considered to be its “head.” The access port slides open to allow the CHIPWIT to use its electro-hydraulic manipulator (or “grabber”) and plasma beam generator (sometimes called a “zapper”).

Debug—The method used by a CHPWIT programmer to eliminate bugs, or errors, from an IBOL program. The Debug mode allows you to follow a program’s execution step by step.

Decrement—An IBOL instruction to subtract the number one from the beaker on top of the Number Stack.

Electro-crab—A mechanical thing in the shape of a crab; dreaded for its ability to move and attack a CHIPWIT.

Electro-hydraulic manipulator—A device that extends from a CHIPWIT's cranial access port to pick up an object directly in front of the CHIPWIT.

Greedville—An adventure designed for the beginning CHIPWIT programmer. Greedville contains no Bouncers, Bombs or Electro-crabs.

Greedy—The most famous CHIPWIT. Known for an insatiable appetite, Greedy prefers Greedville to all other adventures.

IBOL—Icon Based Operating Language; a set of symbols that can be used to form instructions, or commands, for a CHIPWIT.

Increment—An IBOL instruction to add the number one to the beaker on top of the Number Stack.

Olfactory—An adjective used when referring to the sense of smell.

Operator—A part of an IBOL instruction; the Operator is the action to be performed by the CHIPWIT. For example, in the instruction "Look for pie", **look** is the action to be performed (or Operator) and **pie** is the object of that action.

Panel—Part of CHIPWIT's brain; chips are installed on the Main Panel and Subpanels to form a CHIPWIT's program.

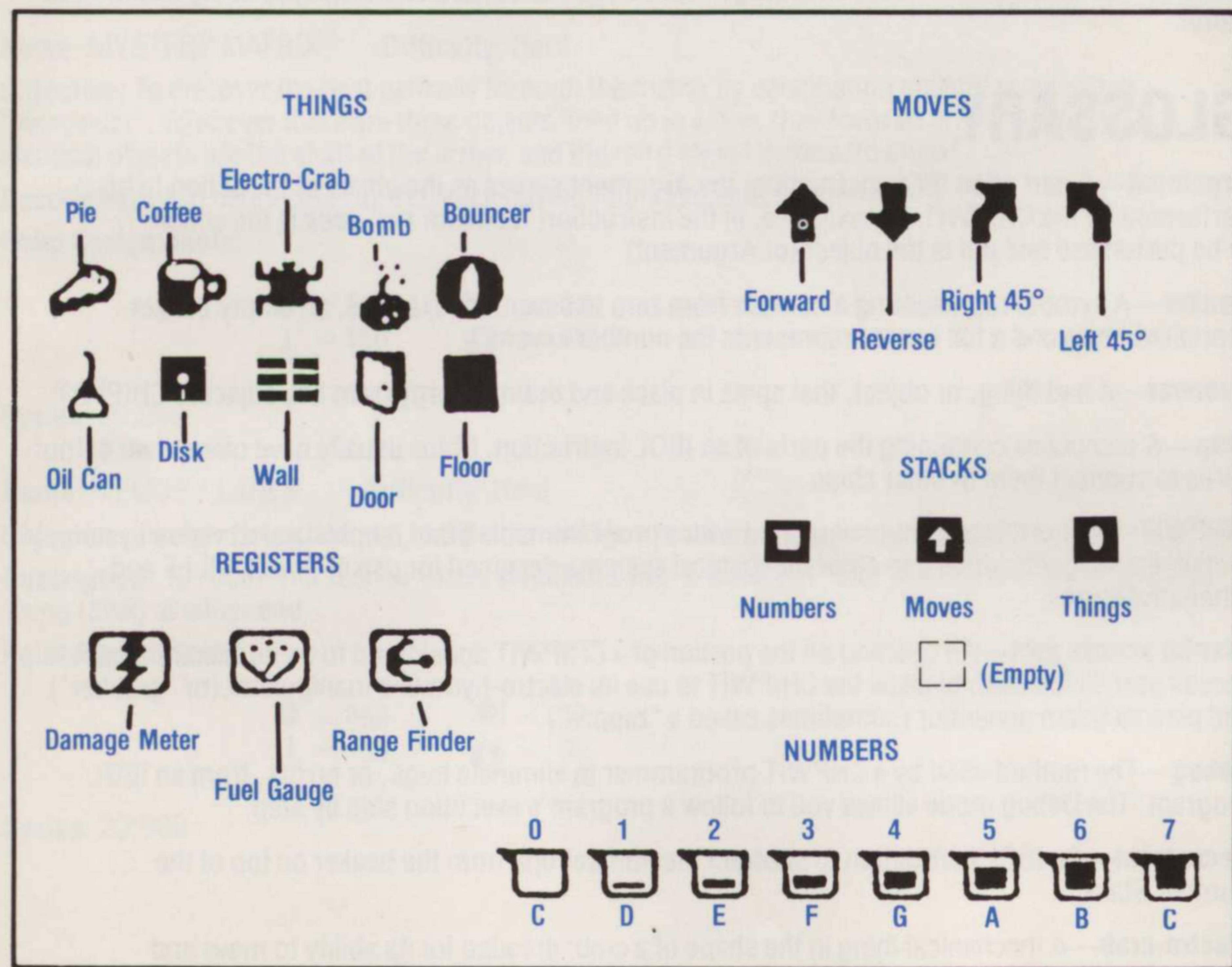
Plasma beam generator—A device that extends from a CHIPWIT's cranial access port to zap the first object in a CHIPWIT's path.

Pop—An IBOL instruction to remove the top element of a stack and discard it.

Socket—Any location in a Main Panel or Subpanel; a place where a chip may be installed.

Stack—A series of memory locations where moves, numbers or things may be stored and retrieved.

Thing—An object that appears in a CHIPWIT adventure. **Good things** include pie, coffee, oilcans and disks. **Bad things** include Bouncers, Bombs and Electro-crabs.



IBOL Reference Card

Operator	Icon	Action	Legal Arguments	Cycles
LOOK		Look for a specific Thing directly ahead and branch T/F	Things	2
SMELL		Smell for a specific Thing in current room and branch T/F	Things	2
FEEL		Feel for a specific Thing directly ahead and branch T/F	Things	4
COMPARE KEY		Compare keypress with specific key and branch T/F	Keys ABCDEFGHIJKLMNOP OPQRSTUVWXYZ	1
MOVE		Move forward, backward, or turn 45°	Moves	5
PICK UP		Pick up the Thing directly ahead	None	5
ZAP		Discharge plasma beam directly ahead	None	7
SING		Sing a note	Notes Registers	2
COIN FLIP		Perform a random branch T/F	None	1
SUBPANEL		Jump to Subpanel and continue execution	Subpanels B C D E F G H I J	1
LOOP		Repeat from start of current Subpanel	None	1
BOOMERANG		Return to Main Panel & continue execution	None	1
SAVE THING		Push argument onto top of Thing Stack	Things	1
COMPARE THING		Compare argument with Thing on top of Thing Stack & branch	Things	1
SAVE MOVE		Push argument onto top of Move Stack	Moves	1
COMPARE MOVE		Compare argument with Move on top of Move Stack & branch	Moves	1
SAVE NUMBER		Push argument onto top of Number Stack	Numbers Registers	1
COMPARE NUMBER =		Compare argument (equal) with Number on Top of Number Stack & branch T/F	Numbers Registers	1
COMPARE NUMBER <		Compare argument (less than) with Number on Top of Number Stack & branch T/F	Numbers Registers	1
INCREMENT		Increment Number Stack by 1 (with wrap around)	None	1
DECREMENT		Decrement Number Stack by 1 (with wrap around)	None	1
POP		Remove top element from specific Stack	Any Stack	1
JUNCTION		Continue program execution in direction of Output wire	None	0

LIMITED WARRANTY

Epyx, Inc., warrants to the original purchaser of this Epyx software product that the medium on which this computer program is recorded is free from defects in materials and workmanship for a period of ninety (90) days from the date of purchase. This Epyx software program is sold "as is," that is without express or implied warranty of any kind, and Epyx is not liable for any losses or damages of any kind resulting from use of this program. Epyx agrees for a period of ninety (90) days to either repair or replace, at its option, free of charge, any Epyx software product, postage paid, with proof of date of purchase, at its Factory Service Center.

This warranty is not applicable to normal wear and tear. This warranty shall not be applicable and shall be void if the defect in the Epyx software product has arisen through abuse, unreasonable use, mistreatment or neglect. THIS WARRANTY IS IN LIEU OF ALL OTHER EXPRESS WARRANTIES AND NO OTHER REPRESENTATION OR CLAIMS OF ANY NATURE SHALL BE BINDING ON OR OBLIGATE EPYX. ANY IMPLIED WARRANTIES APPLICABLE TO THIS SOFTWARE PRODUCT, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED TO THE NINETY (90) DAY PERIOD DESCRIBED ABOVE. IN NO EVENT WILL Epyx BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGE RESULTING FROM POSSESSION, USE OR MALFUNCTION OF THIS Epyx SOFTWARE PRODUCT.

Some states do not allow limitations as to how long an implied warranty lasts and/or the exclusion or limitation of incidental or consequential damages so the above limitations and/or exclusions or limitation of liability may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.



1043 Kiel Court, Sunnyvale, CA 94089
Commodore 64 is a trademark of Commodore Business
Machines, Inc.

© 1985 Epyx, Inc.

Part #07007D-60

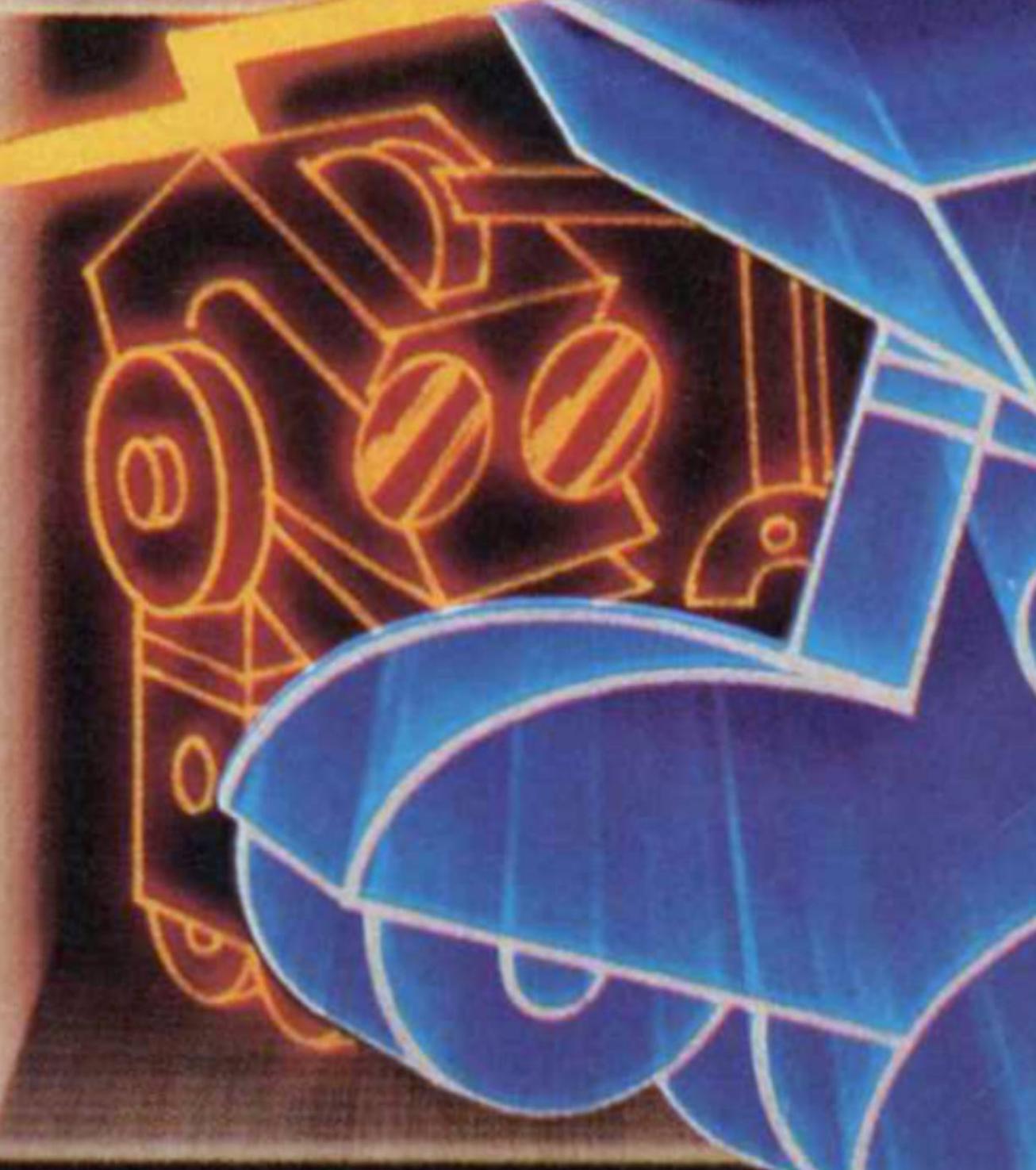
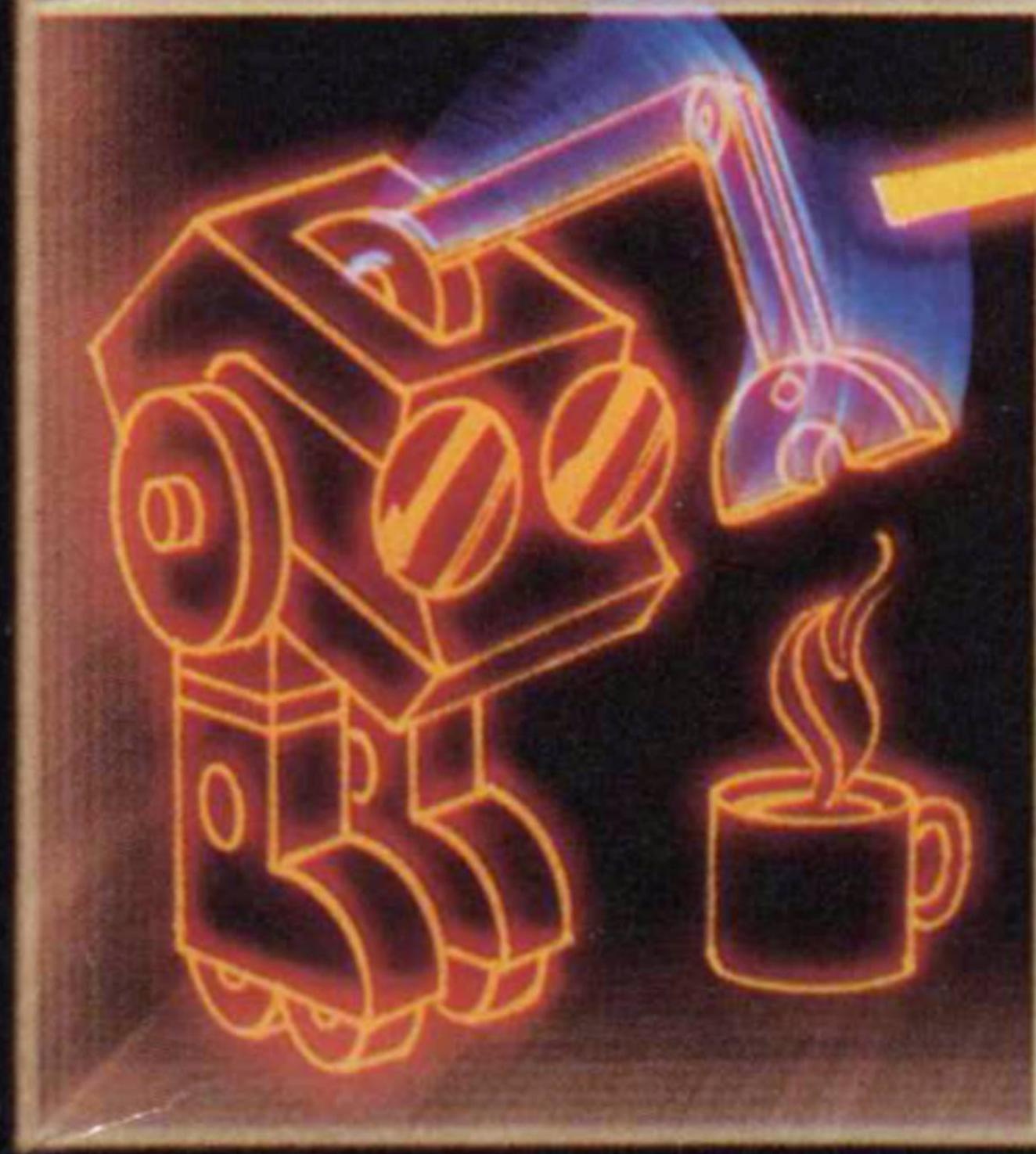
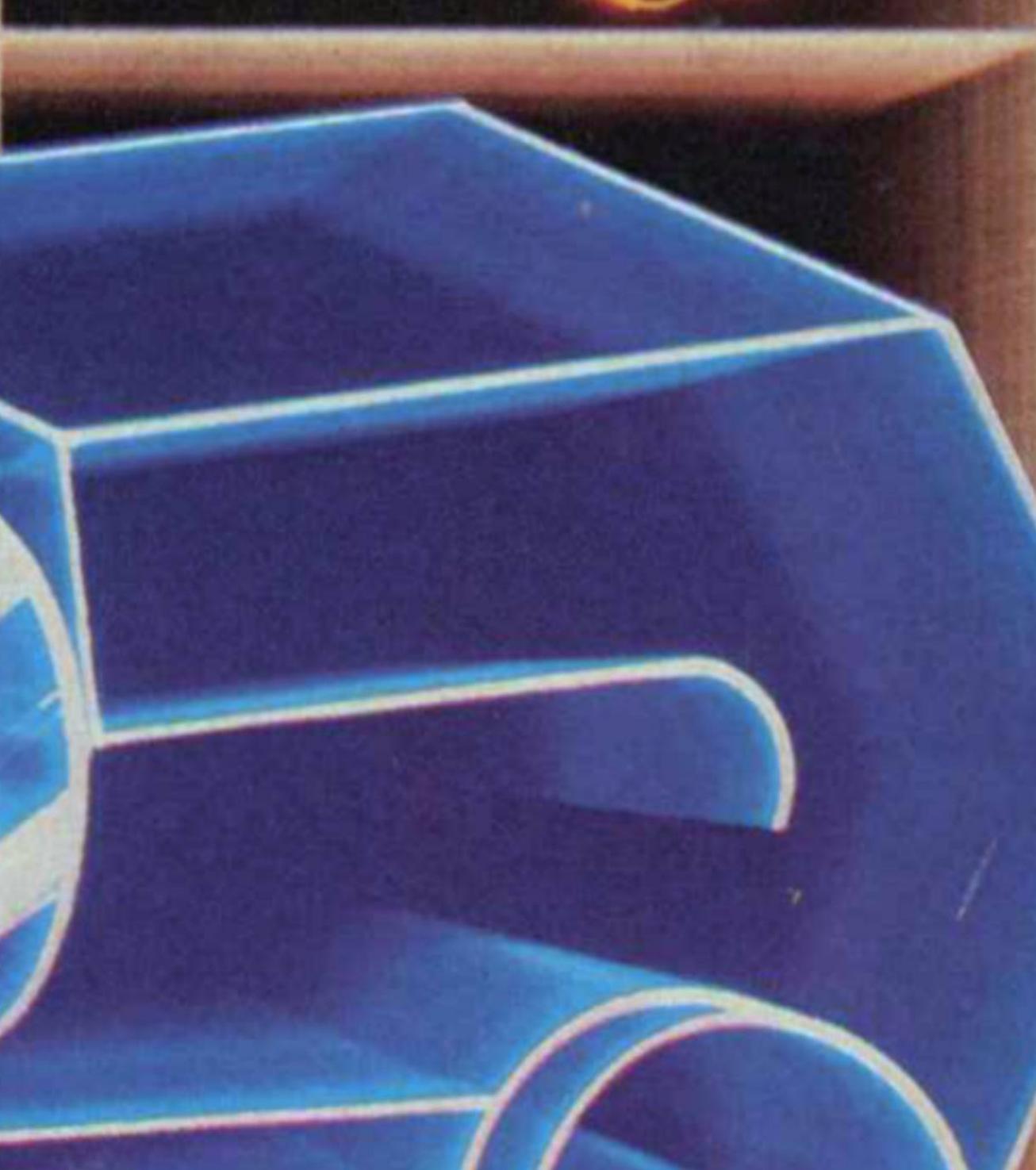
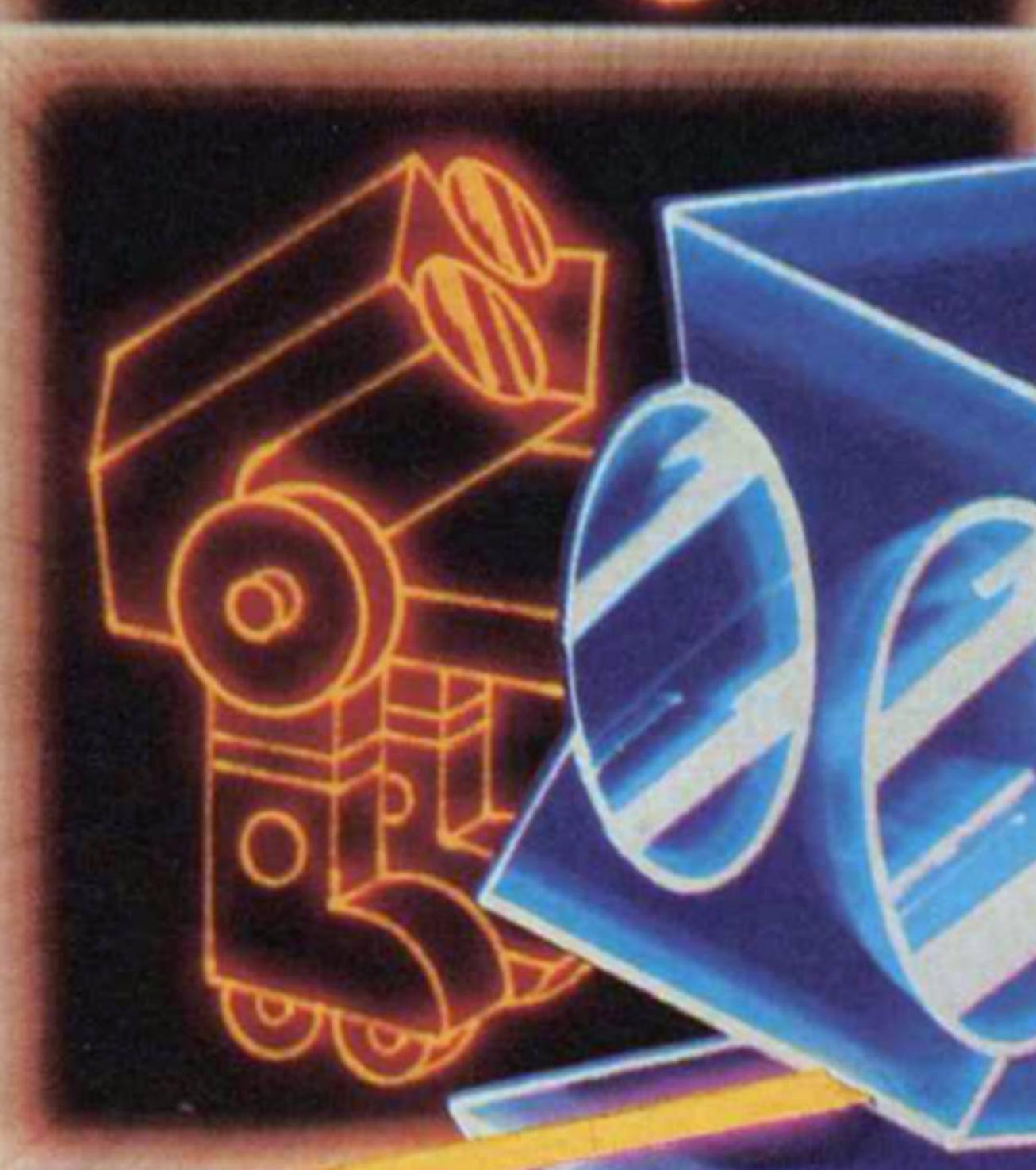
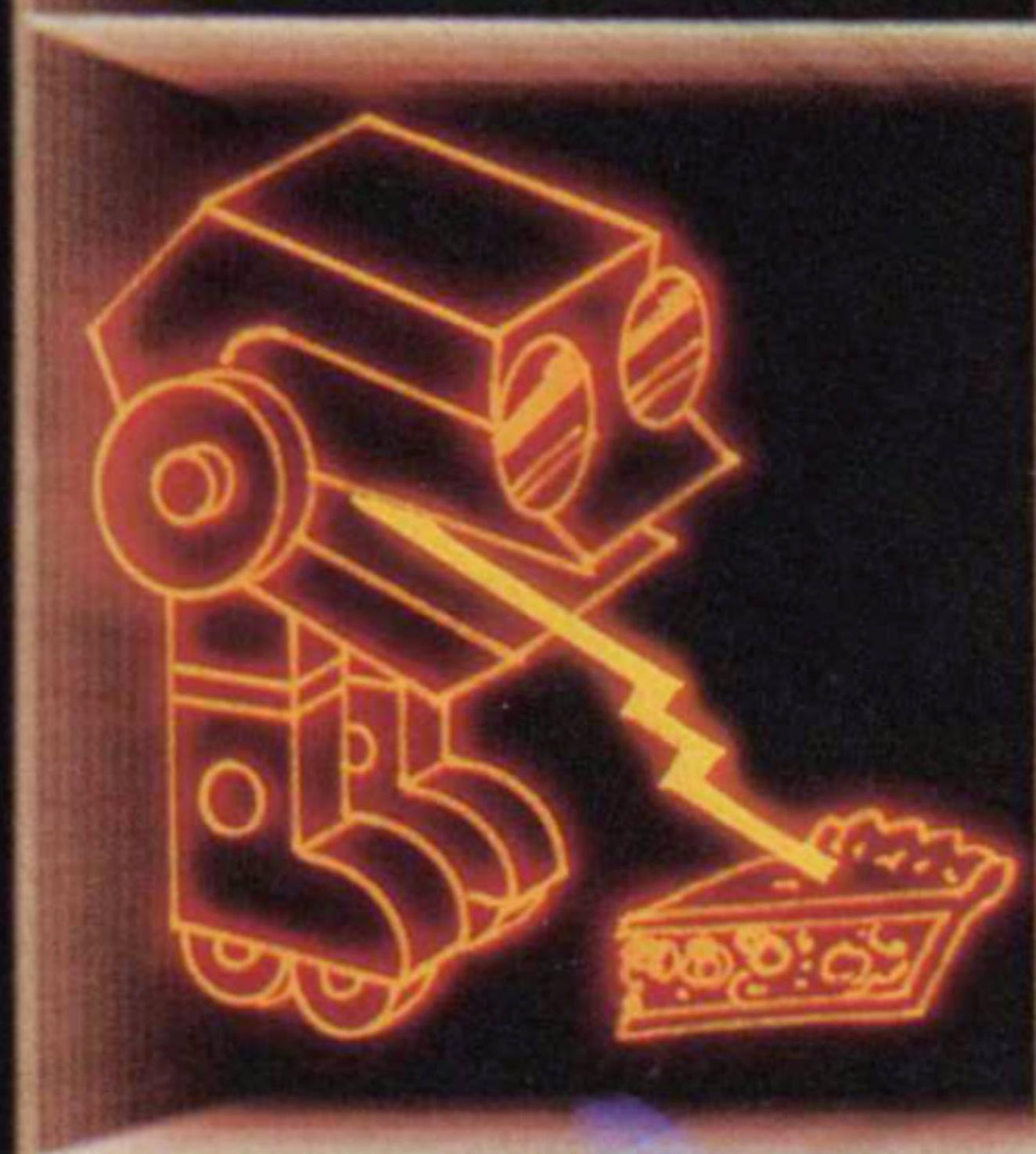
PROOF-OF-PURCHASE
PRODUCT #707D
Epyx
COMPUTER SOFTWARE



EDYX™
COMPUTER SOFTWARE

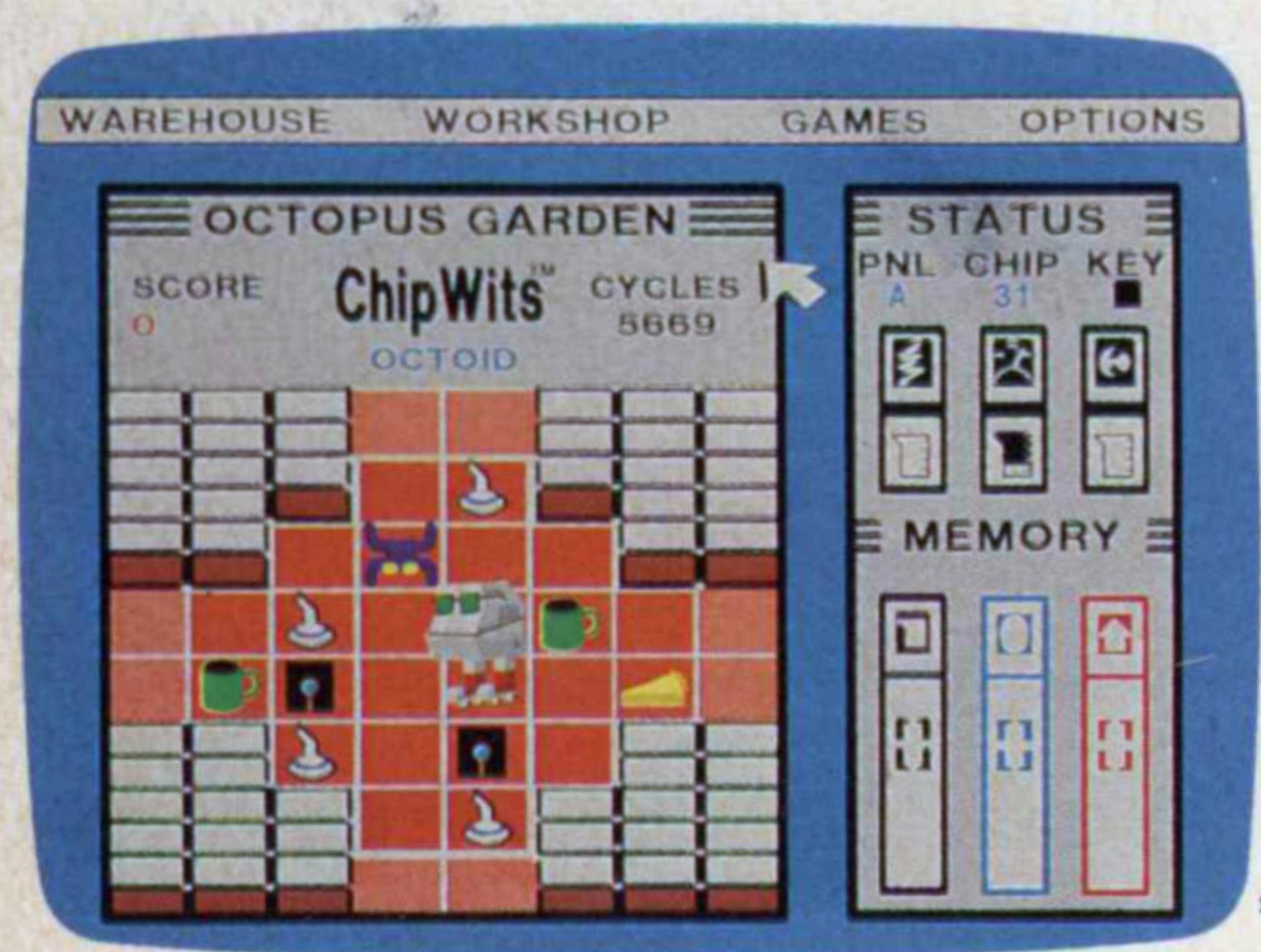
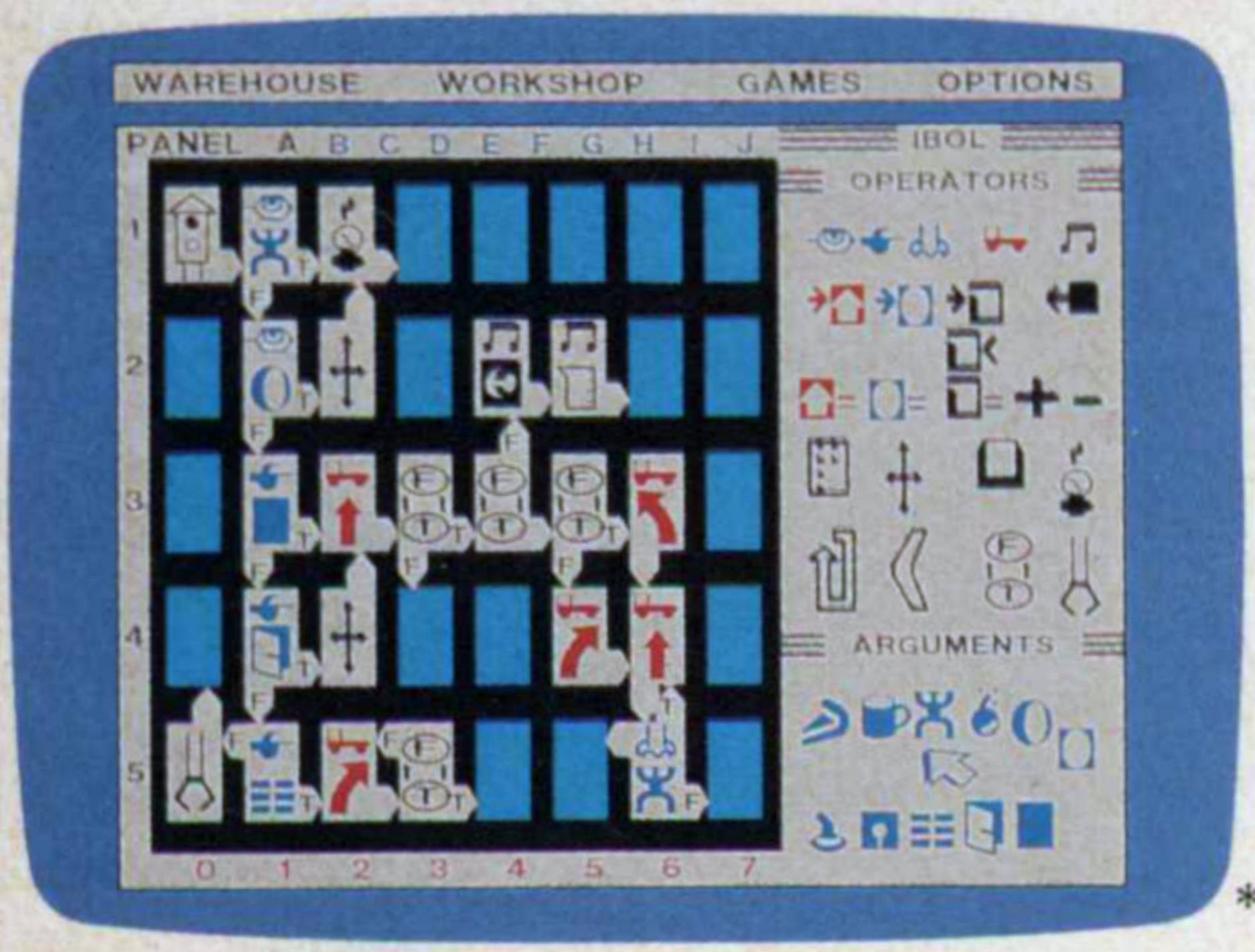
Learning Fun Series™

CHIPWITS™



**COMMODORE
64**
Disk
64K

CHIPWITS™



- 16 Different CHIPWITS Robots to Program.
- 8 Challenging Adventures—From Easy to Difficult.
- Easy-to-Recognize Icons Identify All 16 Robot Functions.
- Multiple Displays Make Programming Concepts Easy to Understand.
- Joystick or KolaPad™ Controlled.
- One Player.

RATING:

EDYX™
COMPUTER SOFTWARE

CHIPWITS is a trademark of EPYX, Inc.
©1985 Epyx, Inc., Sunnyvale, CA

And you thought understanding computers would be difficult. Well it's not if you have the right teacher—a teacher who makes learning a lot of fun. That's why you need Chipwits, the lovable little robots that teach the basic principles of computer programming.

Start in the workshop, where you program your Chipwit for adventures. Should he look for pie first? Or zap that Electro-crab over in the corner? Programming a Chipwit is easy, because the workshop has pictures that represent the actions he can perform. You can make a Chipwit move, eat a piece of pie or even sing a song!

When your program is ready, send Chipwit on any of eight different adventures. Watch your Chipwit try to solve the Mystery Matrix, untangle Octopus Garden or explore Chipwit Caves. If he makes a mistake, simply return to the workshop and change his program. The more he learns, the more you learn, too!

Chipwits. Who would have thought learning could be so much fun?

EPYX CODING

To help you choose the game that's right for you, EPYX games are coded to indicate the degree of emphasis on "Action," "Strategy" or "Learning." These symbols will appear on the front of all EPYX packages:

Action	Strategy	Learning

*Graphics simulate the Commodore 64™ version of game.

TIRED OF WAITING FOREVER FOR YOUR PROGRAMS TO LOAD?



Introducing The Fast Load Cartridge™ For Your Commodore 64™



You're tired of waiting three minutes for your programs to load. But it's no use glaring at your disk drive. Calling it names won't help, either. It was born slow—a lumbering hippo. You need the *Fast Load Cartridge* from EPYX. *Fast Load* transforms your Commodore 64 disk drive from a lumbering hippo into a leaping gazelle. With *Fast Load*, programs that once took minutes to load are booted up in a matter of seconds.

Fast Load plugs into the cartridge port of your Commodore 64 and goes to work automatically. Unlike other products, the *Fast Load Cartridge* even works with protected disks.

So why waste time waiting for your disks to load? Say good-bye to the hippo and load your disks up to five times faster with the *Fast Load Cartridge* from EPYX.

- Loads disks up to five times faster
- Plugs right into your computer
- Easy to use, no loading needed

LIMITED WARRANTY

Epyx, Inc., warrants to the original purchaser of this Epyx software product that the medium on which this computer program is recorded is free from defects in materials and workmanship for a period of ninety (90) days from the date of purchase. This Epyx software program is sold "as is," that is without express or implied warranty of any kind, and Epyx is not liable for any losses or damages of any kind resulting from use of this program. Epyx agrees for a period of ninety (90) days to either repair or replace, at its option, free of charge, any Epyx software product, postage paid, with proof of date of purchase, at its Factory Service Center.

This warranty is not applicable to normal wear and tear. This warranty shall not be applicable and shall be void if the defect in the Epyx software product has arisen through abuse, unreasonable use, mistreatment or neglect. THIS WARRANTY IS IN LIEU OF ALL OTHER EXPRESS WARRANTIES AND NO OTHER REPRESENTATION OR CLAIMS OF ANY NATURE SHALL BE BINDING ON OR OBLIGATE EPYX. ANY IMPLIED WARRANTIES APPLICABLE TO THIS SOFTWARE PRODUCT, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED TO THE NINETY (90) DAY PERIOD DESCRIBED ABOVE. IN NO EVENT WILL Epyx BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGE RESULTING FROM POSSESSION, USE OR MALFUNCTION OF THIS Epyx SOFTWARE PRODUCT.

Some states do not allow limitations as to how long an implied warranty lasts and/or the exclusion or limitation of incidental or consequential damages so the above limitations and/or exclusions or limitation of liability may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

EPYX WARRANTY REGISTRATION

Thank you for purchasing an EPYX computer game! Please complete this warranty registration so that we may keep you informed of new products and promotions from EPYX.

Name _____ Phone (_____) _____

Address _____

City _____ State _____ Zip _____

(S1) Date Purchased _____ (S2) Game Purchased _____

(S3) My Computer is: Apple (1) Atari (3) Adam (4) Commodore 64 (5) IBM (6)
 Macintosh (8) Other (7) : _____

(S10) Have you filed a warranty card for a previous EPYX purchase? Yes No

(S4) From what type of store was game purchased?

Department Electronics Store Toy Store Computer Store Software Store
 Record Store Discount Mail Order Audio/Video Store Other

(S5) How long have you owned your computer?

less than 1 year (1) 1-2 years (2) 2-3 years (3) 3-4 years (4) 4 or more years (5)

(S6) Do you plan to purchase another computer? Yes No

Type: Apple (1) Atari (3) Adam (4) Commodore (5) IBM (6) Macintosh (8) Other (7)

(S7) Your occupation: Professional Trade Student
 Clerical Managerial/Business Other: _____

(S8) Please give Purchaser's: Sex _____ Age: less than 10 (A) 11-14 (B) 15-18 (C)
 19-24 (D) 25-35 (E) 35+ (F)

(S9) Person who plays game most if other than the purchaser:

Sex _____ Age: less than 10 (A) 11-14 (B) 15-18 (C)
 19-24 (D) 25-35 (E) 35+ (F)

(S13) How many computer games do you own? _____ (S14) How many EPYX games do you own? _____

(S15) Do you prefer Action games or Strategy/Role Playing Games ?

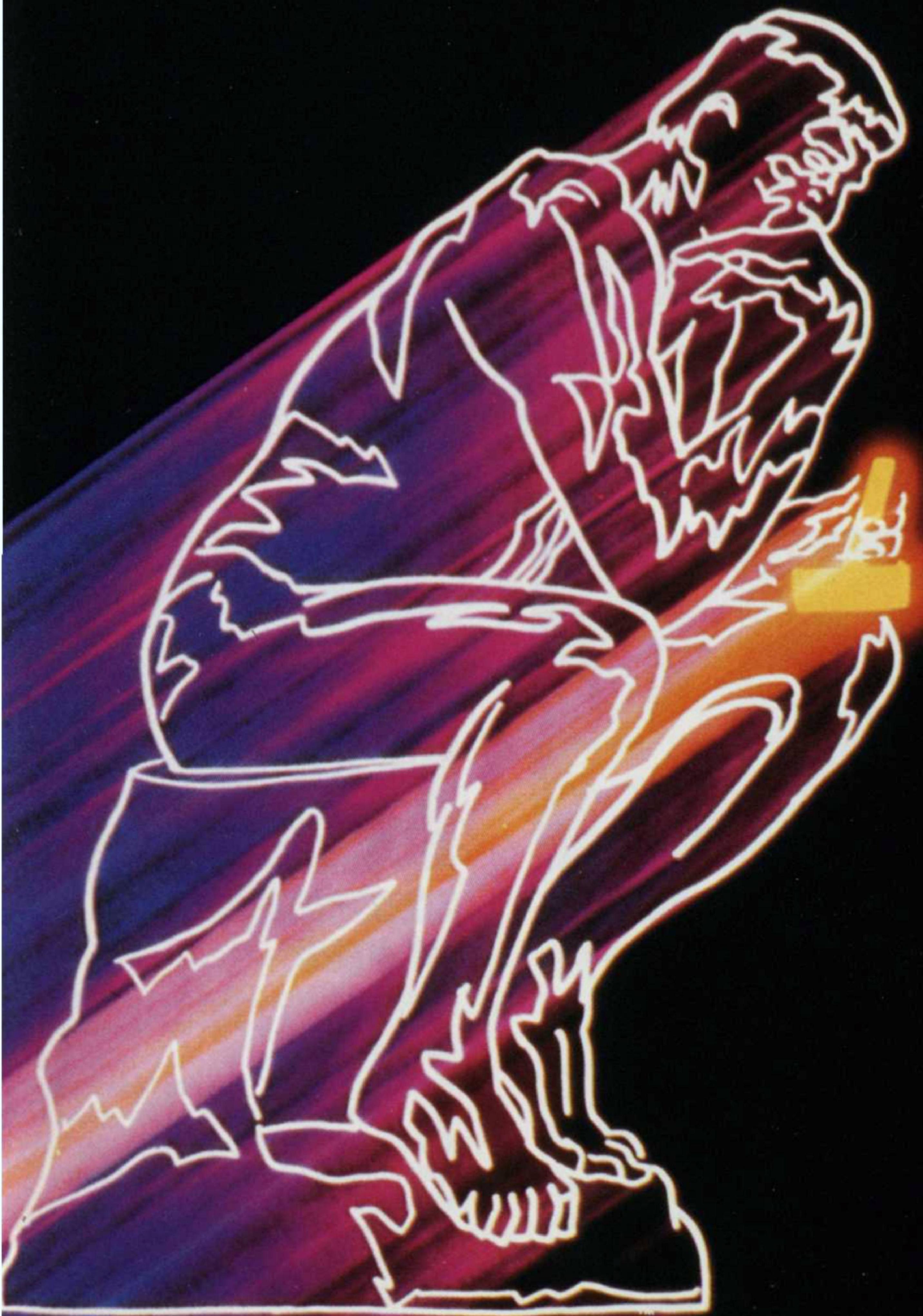
(S11) What is your favorite computer game? _____

(S12) What computer magazine do you read? _____

EPYX

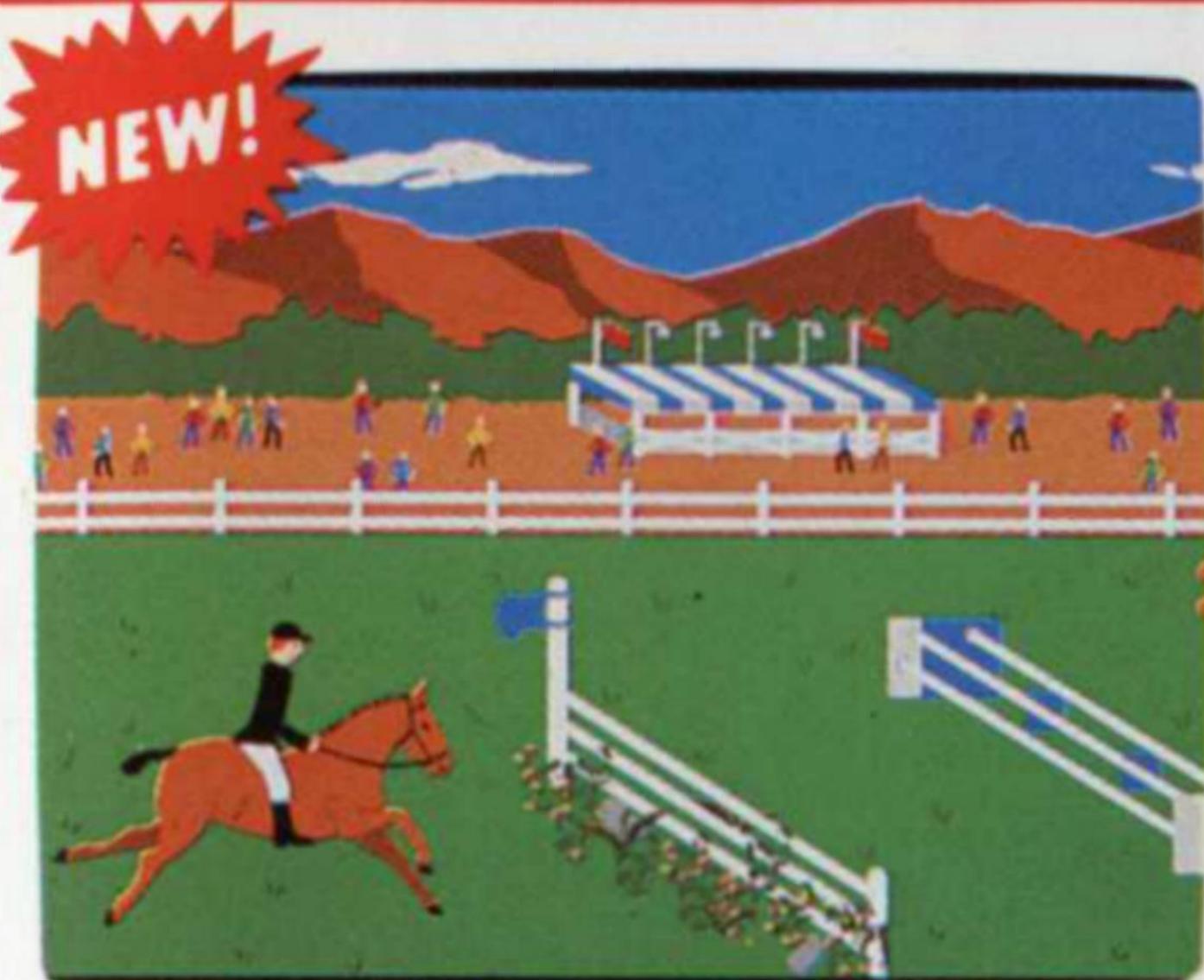
COMPUTER SOFTWARE

Strategy Games for the Action-Game Player™



**CONSUMER SOFTWARE
CATALOG
SPRING 1985**

ACTION-STRATEGY GAMES™



SUMMER GAMES II™

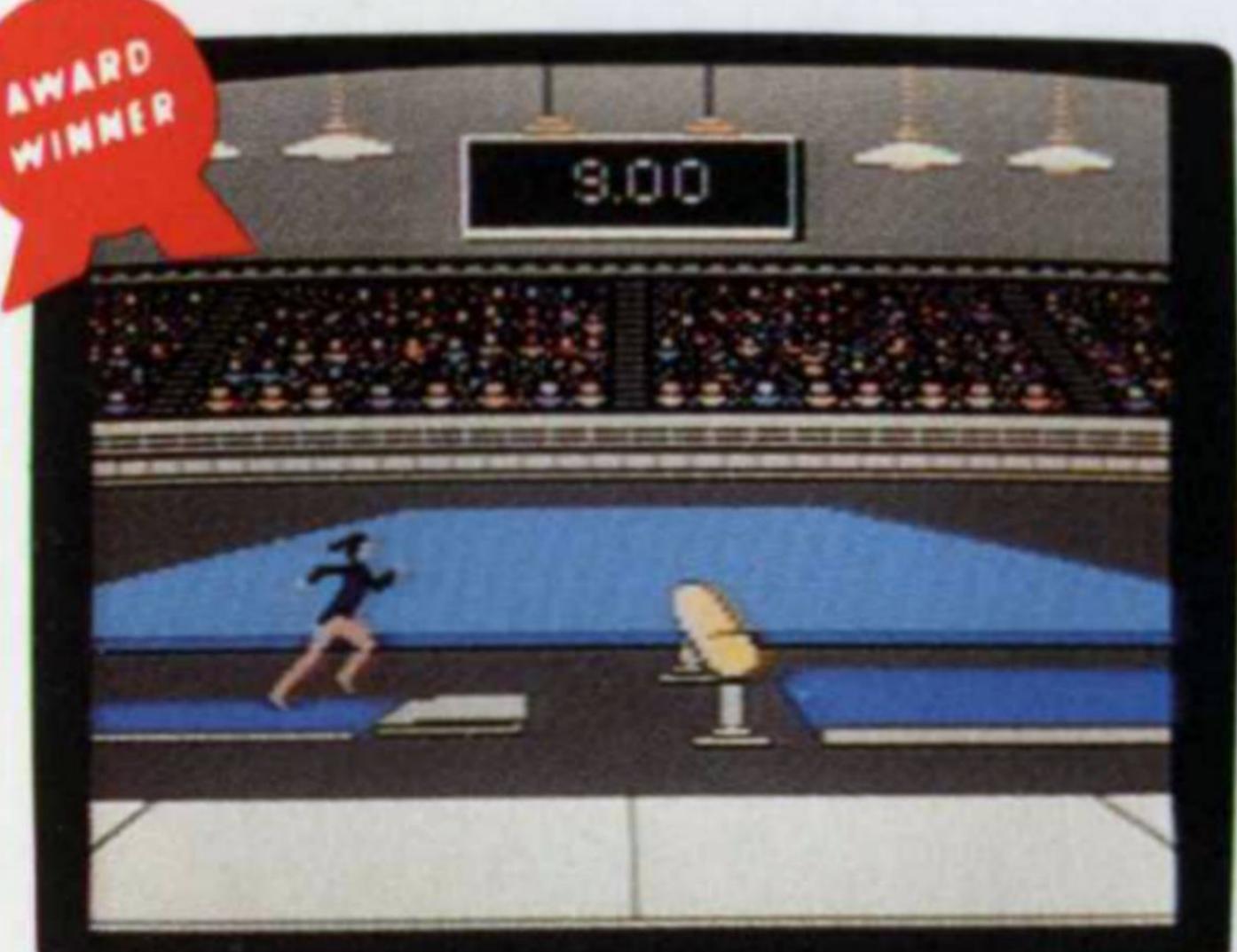
Eight new Olympic-style events let you compete against opponents or against the computer in kayaking, cycling, fencing, diving, track and field, gymnastics, equestrian events and more. Great graphics and sound effects combine with realistic animation and joystick control to help get you ready for 1988. Opening ceremony, awards ceremonies and your choice of countries are all included. Another chance to "Go for the Gold!"

Joystick Controlled

One to Eight Players

Disk: Apple, Commodore 64,
IBM PC and PCjr

Available Spring 1985



SUMMER GAMES™

You're an Olympic athlete competing in eight key events at the Summer Games. How well can you score in track, swimming, diving, skeet shooting, pole vaulting, gymnastics and more. So realistic, there's even an opening ceremony, world records and awards presentations. Change into your running shoes and "Go for the Gold!"

Joystick Controlled

One to Eight Players

Disk: Apple, Atari, Commodore 64



TWO-ON-TWO SPORTS™

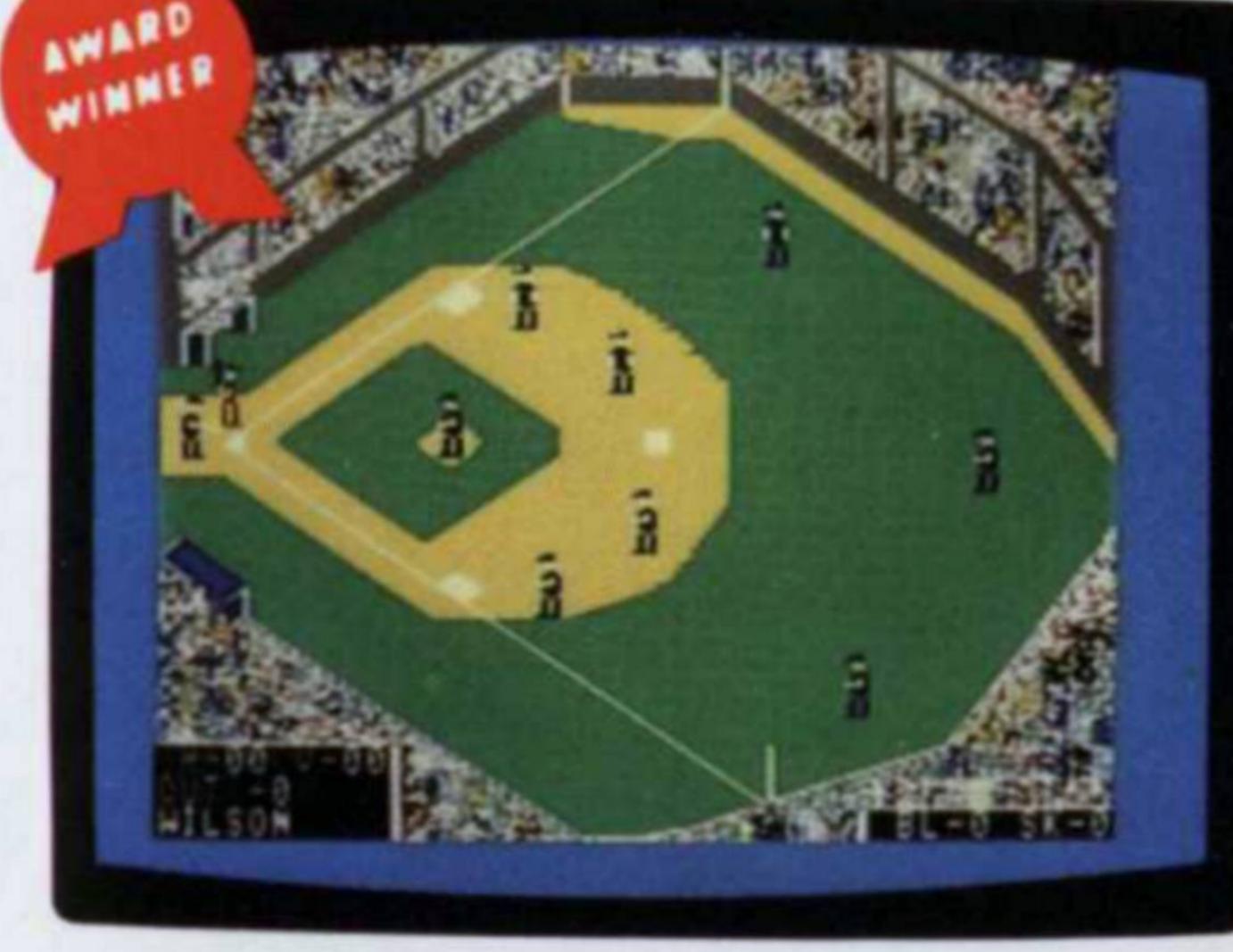
With *Two-On-Two Sports* you and a friend can compete as teammates in "sand lot" versions of volleyball, football, soccer, and baseball, or play alone against the computer and the computer will provide you with a teammate as well as your opponents. Whether it's spiking the ball, intercepting a pass, kicking a goal or hitting the long ball, this game isolates key elements of each sport and lets you go Two-on-Two.

Joystick Controlled

One or Two Players

Disk: Commodore 64

Available Spring 1985



THE WORLD'S GREATEST BASEBALL GAME™

Pick your major league lineup using actual major league baseball players and team statistics. Then, watch the action unfold against an opponent or the computer. Two modes let you choose between managing and controlling your team, or just managing. *The World's Greatest Baseball Game* is everything you could ever want. Hot dogs and peanuts are not included!

Joystick Controlled

One or Two Players

Disk: Apple, Commodore 64



PITSTOP II™

The first auto race game that gives you a chance to go head-to-head against your competitor. Now, more than ever, your racing strategy and your pit crew's speed and performance, combined with your skill on the track, will determine the winner. Step up to *Pitstop II*, because auto racing is not a solo sport.

Joystick Controlled

One or Two Players

Disk: Apple, Atari, Commodore 64,
IBM PC and PCjr



IMPOSSIBLE MISSION™

As a member of the Anti-Computer Terrorist (ATC) Squad, your mission is to reach the infamous Elvin, who is holding the world hostage with the threat of nuclear annihilation. Elvin is hidden in his data base installation protected by robot defenders. Can you penetrate his complex, piece together the hidden puzzles, decode his computer system, and abort his plans? The survival of the world is in your hands!

Joystick Controlled

One Player

Disk: Apple, Commodore 64



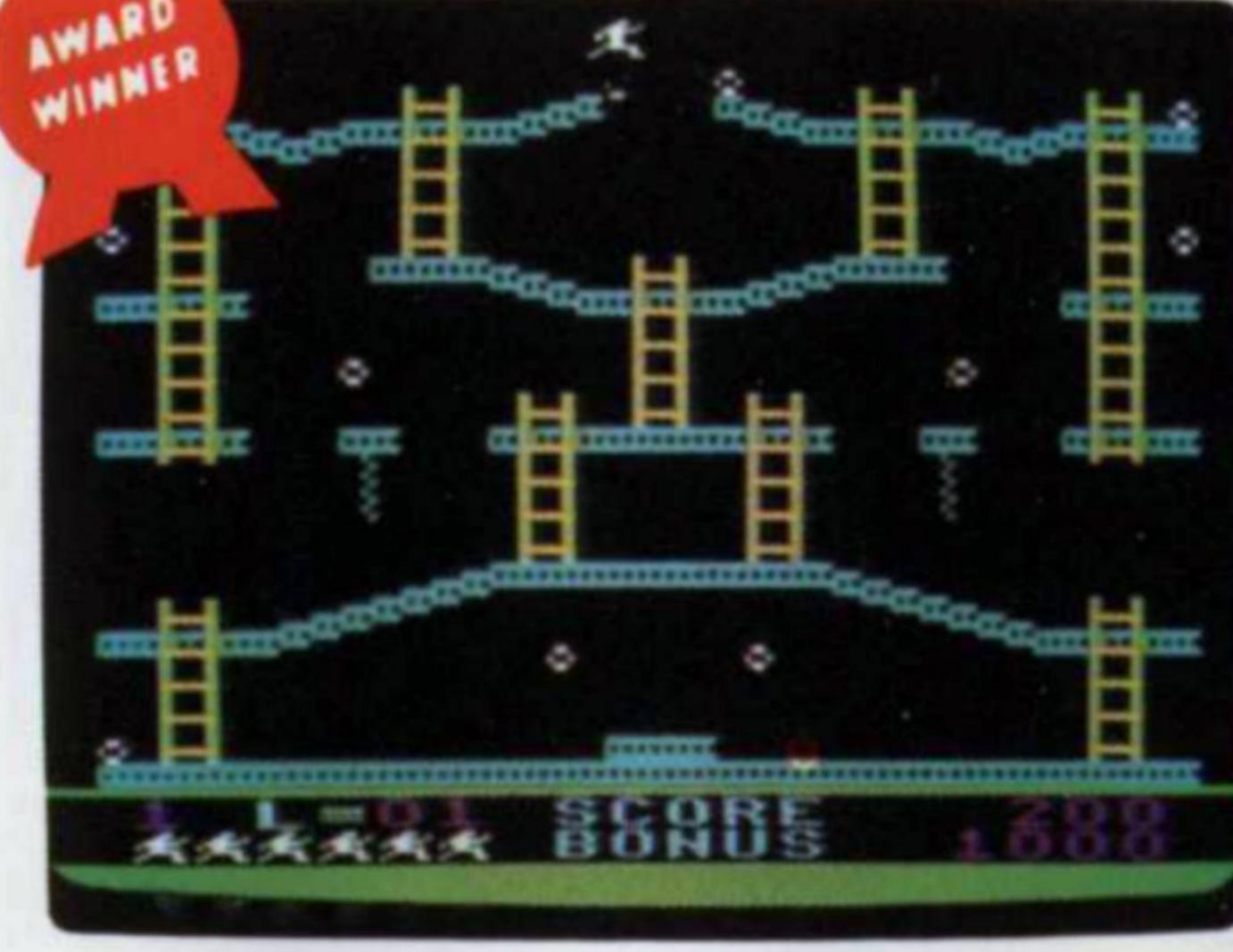
BREAKDANCE™

Now anyone can Breakdance. Just grab your joystick to control your dancer in Poppin', Moonwalking, Headspinning, Up Rocking and more. *Breakdance* is four games in one, each offering a unique and different challenge. You can play it alone or against a friend, either way... Breakin' is now made easy with *Breakdance*!

Joystick Controlled

One or Two Players

Disk: Commodore 64



JUMPMAN®

Only Jumpman's quick reflexes and unparalleled speed can save the Jupiter Command Center from the fiendish Alienators. Leap girders, climb ropes and scale ladders to diffuse the planted bombs on all 30 screens. Watch out! Enemy bullets, robots and birdmen are bent on your destruction! How long can you hold out?

Joystick Controlled

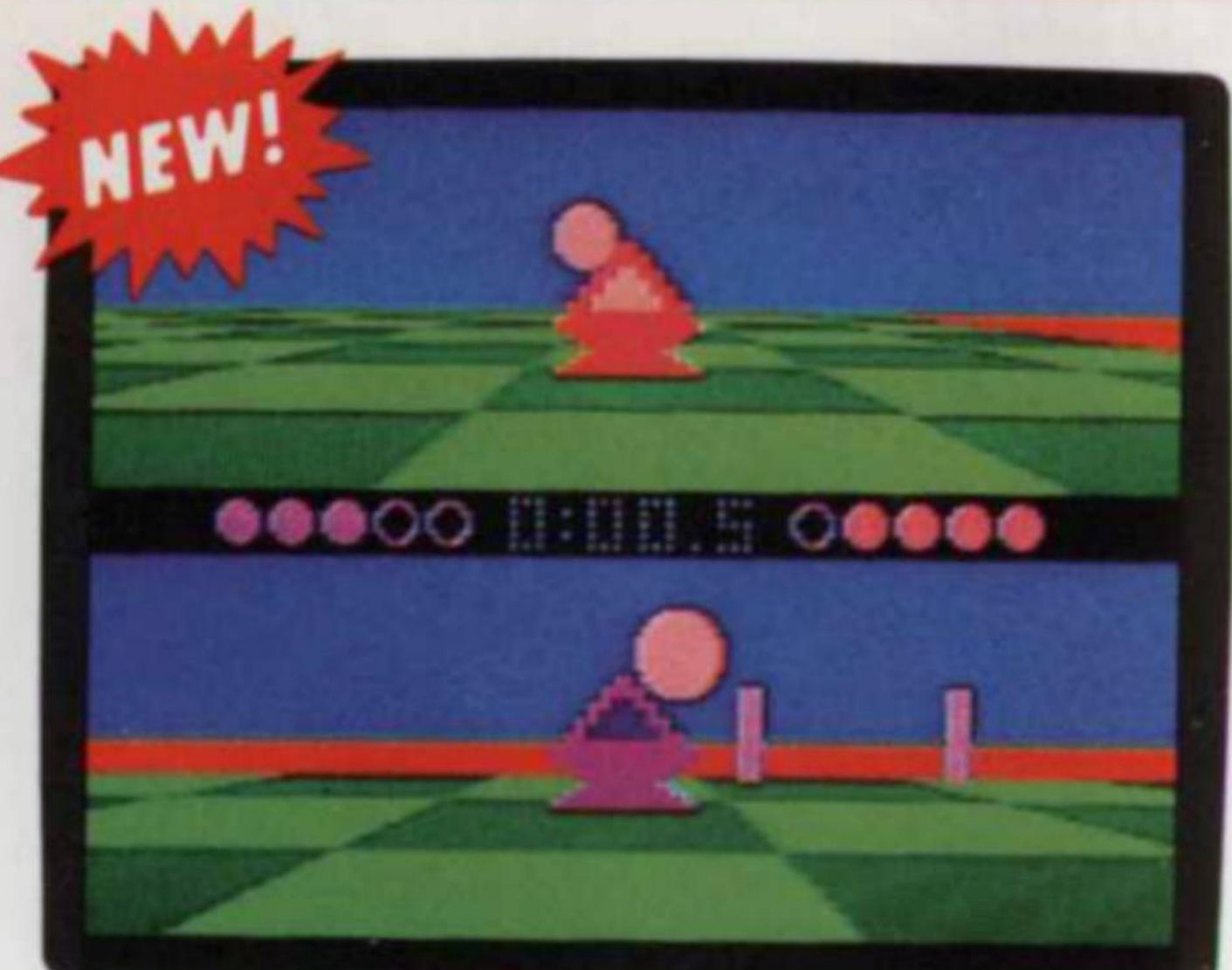
One to Four Players

Disk: Apple, Atari, Commodore 64

Cassette: Atari



LUCASFILM GAMES™



BALLBLAZER™

© LFL 1984



**RESCUE ON
FRACTALUS!™**

Your mission is to fly your *Valkyrie Fighter* through the *Jaggi* defenses and rescue the downed *Ethercorps* pilots. Sounds easy, but don't let it fool you. It's tough enough just to navigate the mountains and canyons of *Fractalus*, but try doing it while destroying enemy gun emplacements or dodging suicide saucers. We supply the Long Range Scanner, *Dirac Mirror Shield* and *Anti-Matter Bubble Torpedoes*... YOU supply the skill and guts!

Joystick Controlled
One or Two Players
Disk: Atari, Commodore 64



Joystick Controlled

One Player

Disk: Atari, Commodore 64



Computer

One Player

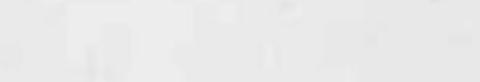
Disk: Atari, Commodore 64



Computer

One Player

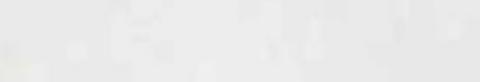
Disk: Atari, Commodore 64



Computer

One Player

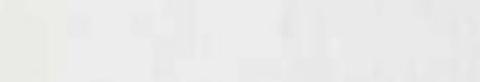
Disk: Atari, Commodore 64



Computer

One Player

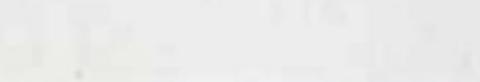
Disk: Atari, Commodore 64



Computer

One Player

Disk: Atari, Commodore 64



Computer

One Player

Disk: Atari, Commodore 64



Computer

One Player

Disk: Atari, Commodore 64



Computer

One Player

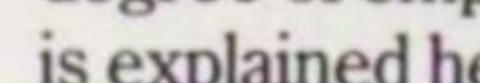
Disk: Atari, Commodore 64



Computer

One Player

Disk: Atari, Commodore 64



Computer

One Player

Disk: Atari, Commodore 64



Computer

One Player

Disk: Atari, Commodore 64



Computer

One Player

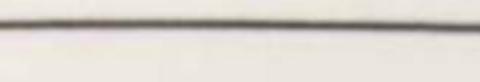
Disk: Atari, Commodore 64



Computer

One Player

Disk: Atari, Commodore 64



Computer

One Player

Disk: Atari, Commodore 64

EDYX™
COMPUTER SOFTWARE

CHIPWITS™

for the Commodore 64™

© 1985, EPYX, Inc., Sunnyvale, CA 94089



STRATEGY GAMES FOR THE ACTION-GAME PLAYER™

EDYX™
COMPUTER SOFTWARE