INTRODUCTION

Congratulations! By purchasing the Oasis Systems "light Pen" you have acquired one of the most fascinating accessories possible for your TRS-80 (T.M.). Light pen input will add a whole new dimension to your computer, especially for games. The programs included on the cassette, and the listings in a later section of this manual, show many of the basic concepts of light pen input. We're sure, however, that in a short while many other possibilities will occur to you. The TRS-80 Light pen is truly an "Imagination Amplifier". We at Oasis Systems hope that this innovative new product will provide you with many hours of enjoyment.

BASIC LIGHT PEN USE

After opening the package and examining the contents you should notice the following things about the light pen. The body of the light pen is connected via a short length of cable to a connector which is similar to the connectors used to connect to the power supply, video monitor and cassette recorder of the TRS-80. In fact the light pen is designed to connect to the same place as the cassette recorder. The light pen has been designed to take advantage of some the internal circuitry of the TRS-80. This results in its small size and its ability to operate from a single 9 volt battery.

The heart of the light pen is the precision photosensor, which is located in the end opposite from the cable. Contrary to what you might gather from its name, the light pen does not produce light but, rather, senses it. Special circuitry built into the body of the light pen enables it to respond only to light emitted from the video monitor of your TRS-80. The internal circuitry will compensate for changes in room light during normal use, but the sensitity of the light pen can be reduced if the ambient room light is too bright. When you point the light pen at a source of light on the video screen the light pen will send a signal to the TRS-80 telling it that it "sees" light. With suitable programming the light pen can be made to act as a pointer so that you can point to information on your video monitor and thereby indicate to the computer what it should do. There are many uses for a light pen, some of which are illustrated by the progroms included on the cassette. Many are quite practical and others are just plain fun.

STARS

A simple but quite fascinating game called "STARS" is provided on the cassette included with the light pen. See the section on "Loading Cassette Programs" and try it out. "Stars" is self prompting with instructions on how to play. "Stars" shows how the light pen can be used as a pointer and should stimulate your imagination towards using the light pen in other games.

PROGRAMMING BASICS

Unlike many other peripheral attachments for computers, the light pen is simplicity in itself. The light pen performs no other function than to inform the computer when it senses light. What it can do is limited only by the sophistication of the programs written for it. However, there are some important considerations you should understand first.

When we look at the screen of the video monitor we see continuous light, but those of us who are savvy about how television works, know that the light is not really continuous. Television creates images by scanning a point of light across the television screen. So rapidly does this point of light scan that to our mere human eyes it appears to be everywhere at once. The photosensor of the light pen, however, is not so easily fooled. It sees a "pulse" of light every time the point scans past wherever the light pen is pointing. The video monitor is designed to scan from the top to the bottom of the screen 60 times a second or once every 1/60 of a second. To prevent this from becoming a problem the light pen makes use of some of the TRS-80's built in circuitry called a "latch". A latch is a device that stores information. A latch can be either SET (ON) or RESET (OFF) and therefore has two 'states'. The TRS-80 is designed such that it can RESET the latch and the light pen can SET it. In addition the TRS-80 can determine the state of the latch at any time. The light pen SETS the latch every time it senses a pulse of light from the video monitor. Once the latch is SET it stays in that state until it is RESET by the TRS-80. Therefore to detect whether or not the the light pen is "looking" at light we need only to RESET the latch, wait at least 1/60 of a second and then check the latch to see if it is SET. Consider the following LEVEL 2 program.

```
10 CLS
20 SET(64,24):SET(65,24)
30 OUT 255,0
40 FOR X=0 TO 6: NEXT X
50 IF (INP(255) AND 128)=0 GOTO 70
60 PRINT @ 0,"ON ";: GOTO 30
70 PRINT @ 0,"OFF";: GOTO 30
```

The program clears the screen (statement 10) and puts a spot of light in the center of the screen (20). Statement 30 causes the latch to be RESET (to know why this is so you would have to know a bit more about the internal operation of the TRS-80 than we have time to go into now). Statement 40 delays execution for about 1/60 of a second as required. Statement 50 reads the the state of the latch and decides which message (ON or OFF) to print. Type in the program, connect the light pen, and RUN it. You will find that pointing the light pen at the spot of light causes the 'ON' message to be displayed in the upper left corner of the screen otherwise the 'OFF' message will show.

The following Level II Basic subroutine does everything needed to determine if the light pen is sensing light.

```
9000 OUT 255,0
9010 FOR Z=0 TO 6: NEXT Z
9020 LP=(INP(255) AND 128)
9030 RETURN
```

If your program calls this subroutine, it will cause the variable LP to be set to 0 if the light pen is "looking" at darkness, or to 128 if the light pen sees light. We will make use of this basic subroutine in developing more sophisticated programs.

SOME PROGRAMMING EXAMPLES

The basic subroutine as shown above is quite easy to use. Consider the game of STARS where the instructions said to "POKE HERE TO START --> *". Implemented in BASIC, this could be performed by the following program.

```
10 CLS
30 PRINT @ 384, "POKE HERE TO START --> ";CHR$(140);CHR$(15);
40 GOSUB 9000: IF LP=0 GOTO 40
50 PRINT CHR$(24);" ";
60 GOSUB 9000: IF LP=0 GOTO 100
70 PRINT CHR$(24);CHR$(140);: GOTO 40
100 CLS: PRINT "OUCH! YOU POKED ME"
```

Examining the program, you will notice a few curious points. Statement 30 prints the initial message and turns off the cursor so it will not be in the way. Statement 40 waits for the light pen to sense light. When it does the program gets to statement 50. Then to be sure that the proper area was poked we backspace and turn off the spot and check again to see if the light pen sees dark. If it does the program prints the message at 100, otherwise it turns the spot back on and resumes waiting at 40. This illustrates an important point. It is possible for the pen to receive input by sensing light from areas that we did not intend such as accidently touching the pen to a line of text. It is important to build in double checks to make sure what was expected has occured. "Stars", for instance, checks four times before deciding that the person playing the game wishes to ZAP a particular star. That is why the stars pulse before they explode.

Consider another application for the light pen called "Menu Selection". Most of us have used programs that required us to choose amoung various options. The game of "Star Trek", which is popular in most computer circles, is an example. The player is requested to "TYPE 1 TO FIRE PHASORS,TYPE 2 TO FIRE PHOTON

TORPEDOES, TYPE 3 TO SCAN QUADRANT FOR LIFE" etc. Wouldn't it be great if your computer had separate keys labeled with each option. That way you wouldn't need to remember which number stood for what. The game might seem a bit more real and a lot less computer-like as a result. We can, as you must have guessed, use the light pen for a similar function. Consider if the screen showed something like this.

```
*    FIRE PHOTON TORPEDOES

*    FIRE PHASORS

*    SCAN FOR LIFE IN QUADRANT

*    SURRENDER
```

We could play the game by merely touching the light pen to the "*" next to the appropriate command. Perhaps certain commands would call up other "Menus" of commands for us to select amoung.

We don't have space here to present a complete Star Trek program, but the following program has all the needed ideas to allow you to do it yourself. This program is called "Imaginary Restaurant". You use the light pen to select what you would like to eat from the "Bill of Fare".

This program is a bit longer than the previous one so try to take it a little at a time and be sure you understand why each statement is important and what it does.

```
5 CLEAR 1200
10 DIM A$(4,4),B(3)
20 B(0)=192
30 B(1)=320
40 B(2)=448

99 REM * READ DATA INTO ARRAYS
100 FOR X=0 TO 3: FOR Y=0 TO 3
110 READ A$(X,Y):NEXT Y,X
120 X=0

199 REM * DISPLAY A SCREEN GROUP
200 CLS
210 PRINT A$(X,0):PRINT:PRINT
220 FOR Y=1 TO 3
230 PRINT CHR$(140);"   ";A$(X,Y):PRINT
240 NEXT Y

299 REM * WAIT FOR LIGHT PEN TO SENSE LIGHT
300 GOSUB 9000: IF LP=0 GOTO 300
309 REM * FIND WHICH STAR WAS TOUCHED
310 FOR W=0 TO 2
```

```
320 PRINT @ B(W)," ";
330 GOSUB 9000: IF LP=0 GOTO 400
340 PRINT @ B(W),CHR$(140);
350 NEXT W
360 GOTO 300

399 REM * GET HERE IF PROPER ONE LOCATED, DOUBLE CHECK
400 FOR S=0 TO 1
410 PRINT @ B(W),CHR$(140);
420 GOSUB 9000: IF LP=0 GOTO 300
430 PRINT @ B(W)," ";
440 GOSUB 9000: IF LP=128 PRINT @ B(W),CHR$(140);: GOTO 300
450 NEXT S
460 PRINT @ B(W),CHR$(140);"<=";
470 S=0
480 S=S+1: GOSUB 9000: IF LP=128 GOTO 470
490 IF S < 5 GOTO 480

499 REM * GET HERE IF DOUBLE CHECK PASSES
500 IF X=0  X=W+1: GOTO 200
510 CLS
520 PRINT @ 192,"VERY GOOD! WE AT THE IMAGNARY";
530 PRINT " RESTAURANT HOPE YOU ENJOY YOUR "
540 PRINT A$(X,W+1);" FOR ";A$(0,X)
550 PRINT:PRINT:PRINT
560 PRINT "BY THE WAY YOUR CHECK TOTALS - $37.50"
570 FOR X=0 TO 5000: NEXT X: GOTO 120

599 REM * DATA AREA
600 DATA "WELCOME TO THE IMAGINARY RESTAURANT - PLEASE SELECT"
610 DATA "BREAKFAST"
620 DATA "LUNCH"
630 DATA "DINNER"
640 DATA "FOR BREAKFAST WE ARE SERVING"
650 DATA "HAM, EGGS, AND SAUSAGE SPECIAL"
660 DATA "PANCAKES AND FRIED POTATOES"
670 DATA "COFFEE AND DANISH"
680 DATA "OUR LUNCH SPECIALS ARE"
690 DATA "HAMBURGER, COKE, AND FRIES"
700 DATA "ROAST BEEF SANDWICH"
710 DATA "VEGETARIAN SPECIAL (RICE AND BEAN CURD)
720 DATA "FOR DINNER WE ARE PLEASED TO OFFER"
730 DATA "EXPENSIVE STEAK"
740 DATA "OVERPRICED SEAFOOD"
750 DATA "ECONOMY SOUP AND SALAD"
```

Note:Remember to include subroutine at 9000.

Statements 5 through 40 reserve string space and dimension the arrays "A$" and "B". Array B is then initialized to the values 192,320, and 448 which correspond to areas on the screen that will be changed with the "PRINT @" command. Statements 100 through 110 read the contents of the "DATA" statements into array A$. The array

is initialized to contain messages that the program will display. A$ is a two dimensional array organized so that the messages are put into groups of 4. Each group corresponds to the "X" dimension of the array and the particular message in a group corresponds to the "Y". Statement 120 selects the first group by setting X=0. Statements 200 through 240 display a group of messages on the screen. Message 0 (Y=0) of each group is displayed as a prompting message at the top of the screen. Messages 1-3 are displayed as "choices" with an target spot preceeding each one. The first group of messages appears as follows.

```
WELCOME TO THE IMAGINARY RESTAURANT -  PLEASE SELECT

*  BREAKFAST

*  LUNCH

*  DINNER
```

The program expects the user to select one of the three choices it offers by pointing to the associated spot with the light pen. The program waits at statement 300 until the pen detects light. When this occurs the program assumes that the user is pointing to one of the three spots. Statements 310 through 350 try to determine which one by turning each off in turn. The program assumes the light pen will see darkness when it finds the correct one. As a precaution the program will get to statement 360 if this test fails. The program branches to statement 400 when it thinks it has located the indicated spot. Statements 400 through 450 perform a double check on this assumption by turning the suspect spot on and off several times and checking for the correct response. If this double check fails, the program branches back to statement 300. If the check passes, the program displays an arrow "<=" next to the spot and waits for the light pen to see darkness meaning the user has lifted the pen from the screen. Statement 500 checks if there are more choices to be displayed. If there are the program sets X to the group requested and branches back to 200. If there are not the program terminates by printing the final selection.

The "Imaginary Restaurant" does not offer the user a very wide choice of food but it easy to expand the program because of its modularity. Perhaps the "Fast Food" restaurants of the future will allows customers to walk up to the counter and use a light pen to select what they want in this fashion.

THE MAGIC SUBROUTINE

Imaginary restaurant is a rather specialized program as you probably noticed. If light pen input is ever going to be painless we will need to develop a subroutine to do the hard work for us. Before we get into exactly how we are going to write such a

subroutine let's define how we would like it to work.

The PRINT@ command in level II basic is a very convenient way to display messages anywhere on the screen. Using this command enables us to refer to any location on the screen by using a value between 0 and 1023 as the starting location for the print. Imaginary restaurant made extensive use of the PRINT@ in setting up the display and in turning the light pen targets on and off. Suppose we had a subroutine which would accept a list of locations on the screen where each location on the screen was described by a value from 0-1023. Let us further suppose that that this subroutine would display a light pen target at each of these locations, wait for the light pen to touch one of them, determine which one was touched, and return a value which indicated which one was touched. This will be our magic light pen subroutine. Let us design this subroutine such that the list of locations is contained in an array called "LST" and that the first element of this array, LST(0), will contain a value indicating how many locations are in the list. Therefore if we wish to display three targets on the screen at locations 120, 430, and 522 we would set up the array as follows.

```
100     LST(0) = 3
110     LST(1) = 120
120     LST(2) = 430
130     LST(3) = 522
```

We would then call our subroutine. (Let's assume it starts at line 9100). We will design the subroutine to set a variable called "SCAN" equal to the list element index (in this case 1 - 3) which corresponds to the target location touched by the light pen. If for example the light pen had touched the target displayed at location 430 then SCAN would equal 2 upon returning from the subroutine.

Now just to show you what good guys we really are, we have included this subroutine for you to start using immediatly.

```
9100    'MAGIC LIGHT PEN SUBROUTINE
9110    L=LST(0)
9120    C$=CHR$(140)
9130    FOR I=1 TO L            :' DISPLAY TARGETS ON SCREEN
9140    PRINT@ LST(I),C$;
9150    NEXT I
9160    GOSUB 9000              :' WAIT FOR LIGHT INPUT FROM PEN
9170    IF LP=0 GOTO 9160
9180    SCAN=1
9190    PRINT@ LST(SCAN)," ";   :' DETERMINE WHICH TARGET WAS TOUCHE
9200    GOSUB 9000
9210    IF LP=0 GOTO 9260
9220    PRINT@ LST(SCAN),C$;
9230    SCAN=SCAN+1
9240    IF SCAN <= L GOTO 9190
```

```
9250    GOTO 9160
9260    PRINT@ LST(SCAN),C$;    :' TARGET FOUND, DOUBLE CHECK
9270    GOSUB 9000
9280    IF LP=0 GOTO 9160
9290    CNT=2
9300    PRINT@ LST(SCAN)," ";
9310    GOSUB 9000
9320    PRINT@ LST(SCAN),C$;
9330    IF LP <> 0 GOTO 9180
9340    GOSUB 9000
9350    IF LP=0 GOTO 9160
9360    CNT=CNT-1
9370    IF CNT <> 0 GOTO 9300
9380    PRINT@ LST(SCAN)-2,"=>"; :' DISPLAY PROMPT ARROWS
9390    PRINT@ LST(SCAN)+1,"<=";
9400    GOSUB 9000              :' WAIT FOR PEN TO BE LIFTED
9410    IF LP <> 0 GOTO 9400
9420    RETURN
```

remember to include the subroutine at 9000 (described on page 3). The magic light pen subroutine uses it to do the actual light pen input.

Here is a short example program which uses the magic subroutine.

```
100     DIM LST(4)
110     LST(0) = 3
120     LST(1) = 266
130     LST(2) = 394
140     LST(3) = 522
150     CLS
160     PRINT "WHO WAS THE SECOND PRESIDENT OF THE U.S. ?"
170     PRINT@ LST(1)+4,"THOMAS JEFFERSON"
180     PRINT@ LST(2)+4,"JOHN ADAMS"
190     PRINT@ LST(3)+4,"PAUL HARVEY"
200     GOSUB 9100
210     CLS
220     IF SCAN = 2 PRINT "THAT IS CORRECT!"
230     IF SCAN <> 2 PRINT "I'M SORRY YOU ARE WRONG!"
240     STOP
```

The subroutine makes use of varaibles named SCAN, I, LP, CNT, C$, and the array LST so don't use them elsewhere unless you don't mind having their values changed. Also remember to DIMension the array LST and to put the proper values into it before doing a GOSUB 9100. The string variable C$, which is assigned a value at statement 9120, determines what character will be used as the light pen target. currently it is set to a graphic character but you can change it to something else if you like.

MAZE CRAZE

"Maze Craze" is a level II program provided on the cassette. It allows the player to wander around in a three dimensional maze. The program needs only the light pen for input and is self prompting with the rules of play. Try the program and see if you can escape from the maze. Here's a hint: try making a map as you proceed.

MACHINE LANGUAGE PROGRAMMING

Until now we have considered only programs written in LEVEL 2 BASIC because we expect that it is what the majority of people will want to use for programming with the light pen. The ideas needed for programming with the light pen are all incorporated in the section on "Programming Basics". All we need do is explain how to perform the I/O from machine language programs. Consider the following subroutine (Note: This machine language program is presented using ZILOG standard Z-80 mnemonics).

```
LPEN:
; SUBROUTINE TO READ LIGHT PEN STATUS AND RETURN
; AC = 0 IF 'NO LIGHT' OR 128 IF LIGHT ;
        PUSH    HL
        LD      HL,1500
        LD      A,0
        OUT     (255),A
LOOP:
; DELAY FOR 1/60 OF A SECOND
        DEC     HL
        LD      A,H
        OR      L
        JP      NZ,LOOP
        IN      A,(255)
        AND     128
        POP     HL
        RET
```

This simple subroutine is sufficient to perform all needed handling of light pen input. We recommend, however, that anyone planning to write machine language programs for the light pen should be throughly familiar with assembly language programming.

Note: Interested readers can obtain a source listing of the program "Stars" by sending $2.00 to cover postage,reproduction costs and handling to Oasis Systems.

LOADING CASSETTE PROGRAMS

All of the programs mentioned in the text are available on the cassette included with the light pen. Since more than one

program is included, a guide is provided to help locate programs on the cassette. To locate a particular program first find the correct title entry in the guide. The guide tells the "side" and "index count" where the program is located. Insert the cassette with the proper side up and rewind the tape. Set the tape counter to zero and then fast forward the tape until the tape counter equals the "index count" specified in the guide.

Some versions of the level II TRS-80 have a 110 ohm resistor installed on the cassette input circuit which may require the recorder volume control to be set higher than normal when reading the light pen program cassette. Initially a volume setting of "6" for basic programs and "7" for machine language programs should suffice, however the user is advised that some experimentation may be required to determine the proper setting for the volume on their particular recorder.

SIDE A (Machine Language Programs)

Index Count	Program Title	File Name
5	Stars	STAR
25	Stars (copy)	STAR
45	Maze Craze	"M"
75	Maze Craze (copy)	"M"

SIDE B (Level II Basic Programs)

Index Count	Program Title	File Name
5	Imaginary Restaurant	"I"
25	Imaginary Restaurant (copy)	"I"
45	Magic Subroutine	"S"
55	Magic Subroutine (copy)	"S"

LOADING MACHINE (SYSTEM) PROGRAMS

Machine language programs may be loaded into a Level II computer with the use of the "SYSTEM" command. Type "SYSTEM" and then "Enter". The prompt "*?" will be displayed on the screen. Position the tape to the beginning of the desired program. Type the "File Name" of the program and then press "Enter". The program will load and the "*?" prompt will be displayed again. Type a slash (/) followed by "Enter" and the program will execute.

WHAT TO DO ABOUT PROBLEMS

Should you experience problems with your light pen the following suggestions could be of help. If the light pen refuses to work at all the problem could be a weak or drained battery. Try replacing the battery and then try one of the simple programs such as the "ON/OFF" test on page 2. If the light pen still refuses to

operate try turning down the lights in the area of the video screen. Adjusting the brightness and contrast controls on the video monitor also may be of help. The light pen is most sensitive to the video screen when the contrast and brightness are adjusted for a bright clear image with a dark background.

WARRANTY RETURN OF DEFECTIVE PRODUCTS

Your Oasis Systems light pen is warranted to be free from defects in materials and workmanship for a period of 30 days following the date of purchase. Any light pen product manufactured by Oasis Systems which fails during the 30 day warranty period as a result of normal use will be replaced at no charge provided the light pen is returned postpaid to Oasis Systems during the warranty period. Any malfunctioning light pen, returned to Oasis Systems during the warranty period, which in the judgement of Oasis Systems has not been subjected to electrical or mechanical abuse, will be replaced or repaired and returned regardless of cause of malfunction.

This warranty is made in lieu of all other warranties expressed or implied and is limited in any case to the repair or replacement of the light pen involved. All correspondence and warranty return items sholud be addressed to the following:

OASIS SYSTEMS
ATTEN: WARRANTY RETURN
2765 REYNARD WAY
SAN DIEGO, CA 92103