

APPLICATIONS PROGRAMMING ENTERPRISE

P.O. BOX 186 • LEE'S SUMMIT, MO. 64063 • [816] 525-2735

George M. Hale/Owner

THANK YOU for ordering our original program: SONIC SATELLITE.

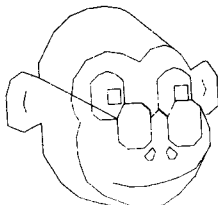
CONDITIONS OF SALE:

- (1) SONIC SATELLITE is a program of limited distribution, produced and published by A.P.E. Each purchaser is assigned a unique six-digit registration number that must be inserted in place of the digits A00000 in line 5 of the program. Use of the program without its registration number is not permitted.
- (2) NO TRANSFER of the registration number is permitted. NO RESALE of the program by the purchaser is allowed.
- (3) NO COPIES OF ANY KIND may be made of the source code, with one exception: One who has purchased the source code listing from A.P.E. is permitted to make a cassette-tape copy to alleviate the necessity of repetitively keying in the entire code. A proper registration number must be inserted into line 5 of the copy. Purchaser is not restricted as to who may use his program for its intended purpose as a video game.
- (4) ALTERATIONS to the program will not be considered to constitute a new program free from the conditions stated above. Alterations would include conversions to machine language, implementation on ROM, as well as mere changes in specific bits of source code.

DESCRIPTION OF PROGRAM:

The SONIC SATELLITE is a sinister invader who appears above your city and directs a SONIC BOMB at some portion of the target below him. When the target is struck, it is flattened! The higher the altitude from which the sound wavefront begins, the greater the destruction that can be dealt to the city below.

The city is defended by two operators of satellite-seeking rockets (hand controls #1 and #2). To stop the satellite, the coordinates of the rocket must land exactly on the coordinates of the satellite (see illustration on next page). A rocket can be quickly positioned prior to launch by rotating the knob on the hand control. Once launched, the rocket automatically follows a path toward the satellite. Pushing forward on the hand control knob accelerates the rocket. Pulling back on the knob decelerates the rocket. With the knob in its center position the rocket travels at constant speed -- which can be zero. Moving the knob toward the left or toward the right will step the rocket one increment to the left or right in addition to any other motion it has. If the rocket is aligned directly above or below the coordinates of the satellite, the controls go into a creep mode (rather than acceleration) and pushing forward or pulling backward on the knob will step the rocket one increment up or down the screen.



The satellite-seeking function of a rocket is disengaged when the rocket is in creep mode or when it has no vertical motion. It is therefore possible to stop the rocket very close to the satellite, then to use the left/right forward/backward controls to creep into overlap, a "hit". Caution! The rocket moves very slowly in creep mode. You will not always have time to zero in on the satellite in this way.

If the rocket goes off screen, a new rocket will appear for that operator at the bottom of the screen. The rockets are anti-gravity compensated, as is the satellite. There is no gravitational pull.

The hand controls are actually read by the program just after each flash of the sonic bomb and just prior to the rockets appearing on the screen.

When the satellite is hit, it will momentarily disappear and then reappear at some random location on the screen. A hit is recorded for the appropriate operator. If both operators hit simultaneously, a hit is recorded for each. A rocket disappears from its position on screen after it has scored a hit, but a new rocket will take its place in the launching area.

The trigger on the hand control is non-functional. A rocket is launched simply by accelerating it. Remember, once it is launched rotation of the knob will have no effect.

THE LENGTH OF THE GAME and the MAXIMUM POSSIBLE SCORE are determined during the introduction to the game when the program asks "# SONIC BOMBS?" Use the Bally BASIC keypad overlay when answering this question.

WHO WINS? If the entire mid-section of the city is destroyed, leaving only a few bits of rubble at the extreme edges of the screen, you may consider that the satellite has won! Otherwise, after all bombs have been released the number of hits for each player is indicated on the screen and the operator with the highest number of hits wins. The game automatically resets after a timed delay.



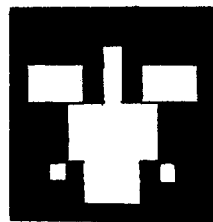
SATELLITE



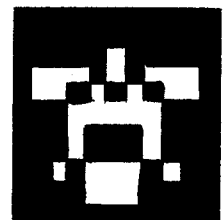
ROCKET
#1



ROCKET
#2



HIT BY #1



HIT BY #2

PROGRAM LAYOUT:

Lines	1 to 250	Program Introduction
	260 to 420	Game Control
	1000 to 1080	Subroutine: Create a City
	2000 to 2270	Subroutine: Read Hand Controls, Branch for Hit
	2500 to 2570	Subroutine: Draw Rockets
	3000 to 3370	Primary Routine: Draw Satellite, Drop Bomb, Branch to 2000, Branch to 2500, Destroy City
	5000 to 5030	Subroutine: Record Hit for Hand Control #1
	5500 to 5530	Subroutine: Record Hit for Hand Control #2
	6000 to 6070	Subroutine: Display Hit Totals at End of Game
	6030 to 6070	2nd Entry Subroutine: Reverse ("Flash") Screen

IMPORTANT NOTE:

When keying in the program from a keyboard,

DO NOT ADD SPACES TO MAKE THE PROGRAM MORE READABLE!

If you should add spaces, you will run out of memory before the entire program is keyed in. SONIC SATELLITE uses nearly all of the 1800 memory spots available as user RAM. If you wish to try modifications of the program, you should probably regain memory by eliminating the print statements in the introduction. The command PRINT SZ, entered directly from the keypad, will tell you how much memory is still unused.

TIPS ON PROGRAMMING:

SONIC SATELLITE is written in rather straightforward Bally BASIC. It is intended to serve as an example of the kind of game that you might create, given sufficient practice, patience, and time.

Immediately after you have printed the program on cassette tape, key in a RUN statement (no line number necessary). The program then starts to run and encounters the :RETURN in line 6. Nothing further is recorded on the tape, although the recorder will not stop until you turn it off. Thereafter, when the program is read back from tape the game will start automatically after it is entered.

Whenever possible, IF statements are constructed to be false the majority of the time. This reduces execution time, making the program run faster. In a subroutine like .FIRE involving many conditional branches, mostly true IF statements would be very detrimental to speed.

SONIC SATELLITE uses only the variable names A through Z (except O and M). By comparison the array variables (@(1),@(2),...@(8),etc.) use more memory and consequently execute more slowly.

TIPS (continued)

Two overlapping subroutines have been used in .SCOR; the computer doesn't care as long as it encounters a RETURN statement after the GOSUB branch to the subroutine.

In order to conserve memory, spaces have been left out where they are not essential. Look at lines 2020 and 2030 for example: The computer word GOTO can immediately follow the number 47, for the system readily distinguishes numeric from non-numeric information. In line 2030, however, it would not be wise to run the variable name R into the word GOTO. (Wouldn't it even be more difficult for YOU to decide what RGOTO meant?) Compare also lines 3200 and 3210. There are many other examples throughout the program.

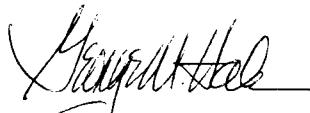
In our TV games, the leading blank within quotes in a PRINT statement is deliberate and should be inserted. It helps out those of you who have sets with an enlarged picture area -- images that have expanded to be bigger than the screen.

ALWAYS USE COMPUTER WORDS WHEN THEY APPEAR IN A PROGRAM. This not only conserves memory, but a computer may not recognize a computer word if you actually spell it out! Although we haven't done so in the game, you can use Bally computer words like BOX (i.e., P O BOX 186) within the text of an INPUT or PRINT statement to save a few bits of memory in your own applications.

A PRINT statement without an argument -- that is, without any text -- will print a blank line. We use this in our "blank line" subroutine, lines 40-60. TV=13 will also generate a blank line (skip to next line) but it uses more memory than the single computer word PRINT.

Line numbers use memory. They can be partially eliminated by stacking several instructions on one line, as in line 70. This is our time delay subroutine. Note that the return statement has been kept separate as line 80 in order to avoid confusing the computer.

Finally, the Bally Computer in our possession has a couple of idiosyncrasies. Perhaps yours does too. Ours must warm up for 10 to 15 minutes before we can be fairly certain that it will operate for a prolonged period without crashing. You don't want a system crash after you have spent hours keying in a lengthy program! We like to print out partially completed versions to tape cassette once in a while while constructing a program so that we don't lose everything in a crash. Also, our system seems to have an uncertainty of +/- 1 in the addressing of its video display under some conditions of execution. To avoid problems in SONIC SATELLITE, we turn the .RCKT display on in line 3260, then off again immediately in line 3270. To have some fun, try interchanging lines 3250 and 3260, then using the hand control. The rockets paths should be ~~quite~~ evident.



George M. Hale

SONIC SATELLITE
by George M. Hale

```
1 .
2 .
3 .
4 .
5 .APE#A00000
6 :RETURN
7 CLEAR ;C=255;BC=C
8 FOR A=16TO 23
9 &(A)=0;NEXT a
10 N=2;GOSUB 40
20 PRINT " SONIC SATELLI
TE"
30 GOTO 90
40 FOR A=1TO N
50 PRINT ;NEXT A
60 RETURN
70 FOR A=1TO 100;NEXT A
80 RETURN
90 NT=3
100 GOSUB 70
110 CLEAR ;GOSUB 40
120 PRINT " THIS IS YOUR
CITY"
130 R=-40
140 GOSUB 1000
150 N=3;GOSUB 40
160 PRINT " YOUR ROCKETS"
170 H=R;I=R;L=20;J=-L
180 GOSUB 2500
190 N=5;GOSUB 40
200 PRINT " "
210 PRINT " YOUR ENEMY"
220 S=1;E=1
230 &(16)=16;&(17)=C
240 &(18)=C;&(20)=32
250 &(21)=30;GOSUB 3000
260 CLEAR ;N=2;GOSUB 40
270 P=0;Q=0
280 INPUT " # SONIC BOMBS
?"S;IF S<1S=1
290 GOSUB 40
300 PRINT " BEGIN BATTLE!
"
310 GOSUB 70
320 CLEAR ;E=0;GOSUB 1000
330 I=R;K=0;H=R;F=0
340 FOR B=1TO S
350 GOSUB 3000
360 NEXT B
370 GOSUB 6000
380 &(23)=0;&(19)=0
```

A color television game for use
with the Bally Arcade or Bally
Home Library Computer;
programmed in Bally BASIC.

User Serial No. A00009
replaces A00000 in line 5.

Illustrated as program will
appear when LIST command is used.

```
1000 .CITY
1010 FOR G=1TO 160STEP 3
1020 BOX G-81,-33,RND (8),
RND (20),1
1030 FOR N=1TO 2
1040 BOX G-81,-25-RND (12)
,2,2,2
1050 NEXT N
1060 NEXT G
1070 BOX -1,R,159,8,1
1080 RETURN
```

```
2000 .FIRE
2010 IF JxJ>6400GOTO 2110
2020 IF I>47GOTO 2110
2030 IF I<R GOTO 2110
2040 IF X=J IF Y=I GOTO 50
00
2050 IF Y=I GOTO 2090
2060 K=K+2xJY(1)
2070 IF X=J K=K-JY(1)
2080 J=J+Kx(X-J)÷(Y-I)
2090 J=J+JX(1)
2100 GOTO 2120
2110 K=0;I=R
2120 I=I+K
2130 IF I=R J=KN(1)÷2
2140 IF LxL>6400GOTO 2240
2150 IF H>47GOTO 2240
2160 IF H<R GOTO 2240
2170 IF X=L IF Y=H GOTO 55
00
2180 IF Y=H GOTO 2220
2190 F=F+2xJY(2)
2200 IF X=L F=F-JY(2)
2210 L=L+Fx(X-L)÷(Y-H)
2220 L=L+JX(2)
2230 GOTO 2250
2240 F=0;H=R
2250 H=H+F
2260 IF H=R L=KN(2)÷2
2270 RETURN
```

SONIC SATELLITE
by George M. Hale

```
390 FOR B=1TO 20
400 GOSUB 70
410 NEXT B
420 GOTO 260

2500 .RCKT
2510 BOX J,I+2,1,3,3
2520 BOX J,I-1,5,4,3
2530 BOX J,I-3,3,2,3
2540 BOX L,H+2,1,3,3
2550 BOX L,H-1,5,5,3
2560 BOX L,H-2,3,4,3
2570 RETURN

5000 .#1
5010 P=P+1;G=1
5020 GOSUB 6030
5030 GOTO 2110

5500 .#2
5510 Q=Q+1;G=1
5520 GOSUB 6030
5530 GOTO 2240

6000 .SCOR
6010 N=2;GOSUB 40
6020 PRINT " 1:",P;PRINT ;
PRINT " 2:",Q
6030 &(23)=C;BC=C
6040 FOR N=1TO 6
6050 &(19)=16÷N
6060 BOX 0,0,161,88,3;NEXT
N
6070 RETURN

3000 .SATL
3010 G=0;D=-36
3020 IF E=1GOTO 3060
3030 X=RND (149)-75
3040 Y=RND (57)-19
3050 GOTO 3070
3060 D=-43;X=-3;Y=36
3070 BOX X,Y,11,11,1
3080 BOX X-3,Y+2,3,2,2
3090 BOX X+3,Y+2,3,2,2
3100 BOX X,Y-3,7,1,2
3110 T=1;NT=1
3120 IF X>0T=-T
3130 FOR Z=1TO 88
3140 U=X+(Z-10)×T÷2
3150 W=Z
3160 &(19)=Z+10
3170 &(22)=Z×2
3180 IF Y<0W=2×Z
3190 V=Y-5-Z;BC=V÷2×8+191
3200 IF V<D GOTO 3350
3210 IF S>1IF V<-23GOTO 33
00
3220 BOX U,V,W,1,1
3230 BOX U,V,W,1,2
3240 IF E>0GOTO 3290
3250 GOSUB 2000
3260 GOSUB 2500
3270 GOSUB 2500
3280 IF G=1GOTO 3300
3290 NEXT Z
3300 &(23)=C;Z=0
3310 BOX X,Y,11,11,2
3320 BOX U,V-Z,W,2×Z+1,2
3330 Z=Z+1
3340 IF Z<7GOTO 3320
3350 &(23)=0;NT=3
3360 &(22)=0;&(19)=0
3370 RETURN
```

Applications Programming
Enterprise
P.O. BOX 186
Lee's Summit, MO. 64063