

General Information

Author: Michael Mitchell

Language: ACTION!

Compiler/Interpreter: ACTION!

Published: ANTIC Vol. 4, #3 (07/ 85)

DARK STAR#

Fly the Darkstar and wipe out enemy alien bases. There are two versions of this game, one is written in BASIC and the other in ACTION!. Both will run on any Atari of all memory sizes, with either disk or cassette. But the ACTION! Listing requires the ACTION! cartridge from Optimized Software Systems. Antic Disk subscribers will find a run-time file of the ACTION! version on their disk. Go to DOS and load DARKSTAR.EXE with the L function.

The year is 2001, you just graduated from high school and you're bored. all your friends headed for college or fulltime jobs, they seemed to have things planned out pretty well. You had dreams of medical school, but there was just no money for that kind of thing. So, what do you do? You join the U.S.A.F.

IN LUCK Things could be worse. It turns out you have a remarkable aptitude as a fighter pilot. But just as you soar to the top of your class, aliens from the planet Spectra land on Earth and begin installing military bases all over the planet. Because of your stupendous flying abilities, you are selected to pilot the Air Force's new secret weapon: Darkstar. Your mission is to destroy as many enemy bases as possible.

FLIGHT INSTRUCTIONS The controller on Darkstar is surprisingly primitive for a new secret weapon, and very similar to an antique Atari joystick. Just move the crude device in the direction you wish to fly the plane. The object is to pass your jet over as many of the enemy bases as possible, spraying them with your wake of radioactive wastes. You're doing fine until you encounter one major problem: the controls have jammed. As a result, Darkstar continues to spray wastes nonstop, thus preventing you from crossing your own path. If you do touch the glowing contrail, your jet will be instantly damaged. Darkstar can withstand three blasts of radioactive waste. Upon the third blast, Darkstar will be terminated-as will you, the pilot!

NOT ENOUGH As if you didn't have enough problems, some wimpy, kneejerk ecology group is up in arms just because your radioactive trail has permanently rendered uninhabitable a large number of small farms plus the entire state of Missouri. Reacting to the pressure, the President interrupts his vacation and orders NASA to erect a deadly force field around your area of operation, effectively converting your flight into a kamikaze mission.

MERIT SYSTEM But you're in this for the glory, and you will receive 30 merit points for each alien base destroyed. However, you can keep obtaining new Darkstars at the end of each mission, simply by pressing the joystick button. If you somehow keep flying until all the aliens flee back to Spectra, maybe the scientists will figure out a way you can escape through the force field

TYPING IT IN Listing 1 is the ACTION! version of Darkstar and Listing 2 is the BASIC version. Although both games are similar in structure, they are not identical. The BASIC version has a simpler title screen, a different explosion routine and-more importantly-is much slower so your scores will probably be higher. If you have the ACTION! cartridge, type in Listing 1, SAVE it and then compile and RUN it. Those with BASIC should type in Listing 2 and check it with TYPO II. SAVE a copy before RUNning it.

ACTION! ANALYSIS The source code is pretty well remarked and consists of only five procedures:

1. PROC WAIT() Pauses according to the CARDinal value passed within the parameter.
2. PROC TITLE() Prints the title, then rotates the screen colors.
3. PROC BOX() Randomly draws enemy bases.
4. PROC MAIN() The heart of the program. It sets graphics to mode seven, checks for collision, checks the joystick, and moves the player.
5. PROC START() Since MAIN() is called within itself, START() is used to isolate the initial TITLE() call.

Michael Mitchell is a San Francisco high school student who wrote the upcoming Antic Public Domain release Rainbow DOS and is sysop of Twilight Zone BBS at (415) 755-0375.

```

; DARKSTAR, LISTING 1
; BY MICHAEL MITCHELL
; (c) 1985, ANTIC PUBLISHING

BYTE WSYNC=54282,VCOUNT=54283,
      CLR=53274,CTR,CLR1=53270,
      CHGCLR=[0],INCCLR1,S,INCCLR,
      CLR2=53271,A,B
CARD ML,SC,SP,Q,I

PROC WAIT(CARD N); MAIN DELAY
  FOR I=0 TO N+N
  DO OD
RETURN

PROC TITLE( ) ; PRINT TITLE AND
SC=0 ML=0 ; SCROLL THE COLORS
GRAPHICS(2+16)
POSITION(5,2)
  PRINTDE(6,"dArKsTaR")
POSITION(8,4)
  PRINTDE(6,"By")
POSITION(1,6)
  PRINTDE(6," MiChAeL mItChELl")
POSITION(0,09)
  PRINTDE(6,"pReSs FiRe To BeGiN!")
DO
  FOR CTR=1 TO 10
  DO
    INCCLR=CHGCLR INCCLR1=CHGCLR
  DO
    S=STRIG(0)
    IF S=0 THEN RETURN FI
    WSYNC=0
    CLR=INCCLR CLR1=INCCLR1
    CLR2=INCCLR+10
    INCCLR==+1 INCCLR1== -1
    UNTIL VCOUNT&128
  OD
  OD
  CHGCLR==+1
OD

```

```

PROC BOX() ; DRAWS THE ENEMY
A=RAND(150)+3 B=RAND(74)+3 COLOR=1
PLOT(A,B)
DRAWTO(A+2,B) DRAWTO(A+2,B+2)
DRAWTO(A,B+2) DRAWTO(A,B)
RETURN

PROC MAIN() ; THE MAIN ROUTINE
INT XX=[1],YY=[0],SS,X,Y,Q
BYTE Z,E,E1,A1,B1,D,C=[0]
BYTE A,B
X=50 Y=50

GRAPHICS(7) COLOR=2 ; DRAW BORDER
SNDRST()
PLOT(1,1)
DRAWTO(158,1) DRAWTO(158,79)
DRAWTO(1,79) DRAWTO(1,1)
BOX()
; LOOK FOR COLLISION
DO
Z=LOCATE(X,Y)
IF Z=1 THEN ; ENEMY HAS BEEN HIT
FOR E=1 TO 20
DO WAIT(50) SOUND(0,E,08,10)
SETCOLOR(2,E,10)
OD
SNDRST() BOX()
SC==+10 SETCOLOR(2,0,0)
FI
IF Z=2 THEN ; YOU HAVE BEEN HIT
FOR D=1 TO 35
DO COLOR=C
SOUND(0,D,8,10) C==+1
SETCOLOR(2,D,C) SETCOLOR(0,C,D)
SETCOLOR(1,A1,B1)
IF C=4 THEN C=1 FI
A1=RAND(153) B1=RAND(78) PLOT(X,Y)
DRAWTO(A1,B1)
OD
ML==+1 SNDRST()
IF ML>2 THEN ; CHECK FOR MEN LEFT
ML=0 GRAPHICS(2+16)
POSITION(4,4)
PRINTD(6,"GAME OVER")
POSITION(4,5)
PRINTD(6,"SCORE: ") PRINTBDE(6,SC)
FOR X=0 TO 242 STEP 2
DO
WAIT(500) SOUND(0,X+1,10,10)
SOUND(1,X+2,10,10)
SOUND(2,X+3,10,10)
SOUND(3,X+4,10,10)
SETCOLOR(0,X,10)
OD
WAIT(32000) ; DELAY
SC=0 TITLE()
FI
MAIN()

```

FI

Q==+1 SETCOLOR(1,Q,14)

COLOR=2 PLOT(X,Y)

SS=STICK(0) ; READ THE JOYSTICK

IF SS=14 THEN XX=0 YY=-1

ELSEIF SS=13 THEN XX=0 YY=1

ELSEIF SS=11 THEN XX=-1 YY=0

ELSEIF SS=7 THEN XX=1 YY=0

FI

WAIT(350) ; CHANGE WAIT VALUE FOR

; FASTER OR SLOWER SPEEDS

X==+XX Y==+YY

POKE(53279,5) ; KEYBOARD SOUND

OD

RETURN

PROC START()

TITLE()

MAIN()

RETURN