General Information

Author: Greg Knauss
Language: ACTION!
Compiler/Interpreter: ACTION!
Published: Antic Vol. 6 #10 (02/ 88)

Killer Chess#

Two-player ACTION! shootout#

Killer Chess brings a new frenzy of aggression to the classic game, as you mop up the chessboard without waiting for your opponent to make moves. This type-in program is written in ACTION! and requires the ACTION! language cartridge from Optimized Systems Software, as well as an 8-bit Atari computer with at least 32K memory and a disk drive.

Unless you're a real fanatic or a tournament contender, I'll bet that you don't play much chess anymore. Let's face it, most "regular folks" find chess boring!

But now imagine a revitalized, fast-ACTION! chess-where the players don't take turns.

That's right. . . no turns. Killer Chess players make legal chess moves as fast as they can, deciding on instant strategies that they would have spent dull minutes pondering in a traditional game. Stodgy old chess becomes a fast-gun shootout.

Welcome to Killer Chess, written in ACTION! the fast, powerful programming language from Optimized Systems Software. You and your human opponent will use an Atari 8-bit computer and a pair of joysticks to battle it out in a radical new version of a traditional game

GETTING STARTED

TYPING IT IN: Insert the ACTION! cartridge into your 8-bit Atari and type in Listing 1, KILLER.ACT Type carefully; because there isn't a TYPO II for ACTION! After you have a copy of the complete program safely saved, go to the monitor by pressing [CONTROL] [SHIFT] [M] and compile the program by typing [C] [RETURN]. When the cursor starts blinking again, type [R] [RETURN] and the title page should appear.

MONTHLY DISK USERS: You can play Killer Chess without owning the ACTION! cartridge. Just insert your Antic Monthly Disk into your disk drive, remove all cartridges from your Atari (XL/XE owners should press the [OPTION] key) and turn on your Atari. When the DOS menu appears, just type [L][RETURN], then type KILLER.EXE [RETURN].

When the title screen is seen, press [START] to begin a game. When the game begins, both players will be able to simultaneously move their respective cursors around the board. With joystick 0, player 1 controls the white cursor and white pieces. With joystick 1, player 2 controls the gray cursor and gray pieces.

PLAYING KILLER CHESS

Simply place the cursor over any piece you want to move and press the joystick button. Now move the cursor over a square that would be a legal move for that piece and press the button again. If the move is illegal, the computer will tell you so -with a rather unpleasant sound- and let you try again. Otherwise the piece will be placed at the new square. If you accidentally pick up a piece and don't want to move it, just replace the cursor over the piece you selected and press the button again. The piece will be dropped.

To capture an enemy, simply make a legal move on top of it. The offending piece will be removed from play. You can capture a piece your opponent is "holding". The piece isn't actually moved until it is set down again.

To win, just land one of your characters on top of the opponent's King. To return to the title screen press [START] or wait about 10 seconds.

Killer Chess does not have castling or en passant moves, which are allowed under advanced chess rules but would be too confusing here.

ABOUT THE PROGRAM

The biggest programming problem in Killer Chess was detecting illegal chess moves. My solution is quite simple and can be applied to any chess program. The method is even fast enough to be used with BASIC.

Here's what I did: When a piece is selected, its old position is recorded. Each new position chosen by a player is also recorded. The old position is then subtracted from the new position and stored in a "delta" value, one delta for X and one for Y Delta means how much something changes. So if the new X position is 5 more than the previous one, the Delta X would be five. If the new Y position is 1 less than the old, Delta Y would be -1.

I then used IF statements to determine if the piece was allowed to move to that spot. For instance, a pawn is only allowed to move forward, so I checked to make sure that Delta X is equal to nothing but 1. If the old position was equal to its starting position, I allowed it to move an extra space-because Pawns can move two spaces on their first move.

If the Pawn's new position is on top of an opponent's piece, I allowed for a Delta Y movement of either 1 or -1. Combined with the Delta X, that would result in diagonal movement. Simple, really. It just took a bit of planning to work out the values for the special conditions of each chess piece.

Greg "Maddog" Knauss of Rancho Palos Verdes, California is an indefatigable ACTION! language programmer

```
; KILLER CHESS
; BY GREG KNAUSS
; (c)1987, ANTIC PUBLISHING

CARD PM,CH
BYTE I,J,K,STK,PLR,LOC,CAP,OK
BYTE ARRAY X(2),Y(2),OX(2),OY(2),
     HOLD(2),PAU(2),SND(2),DIS(2)
INT  X1,Y1,DX,DY,DUM1,DUM2


PROC CURSOR()

; SHAPE FOR CURSORS
[255 129 129 129 129 129 129 255]


PROC CHRS()

; BOARD EDGE
```

```
[  0   0   0   0   0   0   0   0
   0   0   0   0  15  15  15  15
   0   0   0   0   0   0   0   0
   0   0   0   0 255 255 255 255
   0   0   0   0 240 240 240 240
  15  15  15  15  15  15  15  15
 240 240 240 240 240 240 240 240
  15  15  15  15   0   0   0   0
 255 255 255 255   0   0   0   0
 240 240 240 240   0   0   0   0

; PIECES
   0   0   0  56  56  16 124   0
   0  84 124  56  56  56 124   0
   0   6  60 124  28  28  60 126
   0  16  24 108 124  56  16 124
   0 214 254 124  56  56 124 254
   0  16  56 146 254 124  56 124

; TITLE
   0 247 108 112 112 108 246   3
   0  62 102  96  96 102  60   0
  24   0  56  24  24  24  60   0
 224  96 124 102 102 102 247   0
  56  24  24  24  24  24  60   0
   0   0  60 102 126  96  62   0
   0   0 220 102  96  96 240   0
   0   0  62  96  60   6 124   0


; "PRESS START"
   0 238 170 238 140 138   0   0
   0 238 136 206 130 238   0   0
   1 225 129 225  33 225   1   0
 255  17 123  27 219  27 255   0
 255  17  85  17  83  85 255   0
 240  16 176 176 176 180 240   0]

PROC SETUP()
GRAPHICS(18) POKE(559,0)
POKE (559,46)

; COLORS
SETBLOCK(706,2,66)
POKE(704,14) POKE(705,8)
POKE(708,10) POKE(710,4)
POKE(709,142) POKE(711,15) ;DO 711

; P/M GRAPHICS
PM=(PEEK(106)-8)*256
POKE(54279,PM/256) POKE(53277,3)
SETBLOCK(53258,2,3) ZERO(PM,1024)
POKE(623,2) PM==+512 K=51
FOR I=32 TO 95 STEP 8 DO
 FOR J=0 TO 7 DO POKE(PM+I+J+256,K)
  POKE(PM+I+J+384,K) OD K=255-K OD

; REDEFINED CHARACTERS
CH=(PEEK(106)-16)*256
```

```
MOVEBLOCK(CH,CHRS,512)
POKE(756,CH/256)

; DRAW BOARD
POSITION(5,1) PRINTD(6,"..........")
  FOR I=2 TO 9 DO POSITION(5,I)
  PRINTD(6,".        .") OD
POSITION(5,10) PRINTD(6,"..........")
POSITION(6,2) PRINTD(6,"+,-./-,+")
POSITION(6,3) PRINTD(6,"********")
POSITION(7,5) PRINTD(6,"......")
POSITION(7,6) PRINTD(6,".....")
POSITION(6,8) PRINTD(6,"********")
POSITION(6,9) PRINTD(6,"+,-./-,+")
POSITION(7,11) PRINTD(6,"......")
POKE(53250,96) POKE(53251,128)
POKE(559,46)

; WAIT FOR [START]
I=0 DO POKE(54282,0) POKE(53273,I)
 I==+3 UNTIL PEEK(53279)=6 OD

; DRAW PIECES
POSITION(6,2) PRINTD(6,"+*    *+")
POSITION(6,3) PRINTD(6,",*    *,")
POSITION(6,4) PRINTD(6,"-*    *-")
POSITION(6,5) PRINTD(6,".*    *.")
POSITION(6,6) PRINTD(6,"/*    */")
POSITION(6,7) PRINTD(6,"-*    *-")
POSITION(6,8) PRINTD(6,",*    *,")
POSITION(6,9) PRINTD(6,"+*    *+")
POSITION(7,11) PRINTD(6,"      ")
RETURN


PROC MAIN()

; GAME LOOP
DO

SETUP()

X(0)=6 Y(0)=5 X(1)=13 Y(1)=6
HOLD(0)=0 HOLD(1)=0 PAU(0)=0 PAU(1)=0
PLR=1

; PLAYER TURN LOOP
DO

; ALTERNATE PLAYERS
PLR=1-PLR

; RESET THESE FOR EACH TURN
X1=0 Y1=0 POKE(77,0)
IF PAU(PLR)=0 THEN SOUND(PLR,0,0,0)
 FI

; MOVE WHICH WAY???
STK=STICK(PLR)
```

```
IF STK=14 OR STK=10 OR STK=6 THEN
 Y1=-1 FI
IF STK=13 OR STK=9 OR STK=5 THEN Y1=1
 FI
IF STK=11 OR STK=10 OR STK=9 THEN
 X1=-1 FI
IF STK=7 OR STK=6 OR STK=5 THEN X1=1
 FI

; KEEP PLAYER ON BOARD
LOC=LOCATE(X(PLR)+X1,Y(PLR)+Y1)
IF LOC<10 THEN X1=0 Y1=0 FI

; MOVE CURSOR
IF Y1<>0 THEN
 ZERO(PM+128*PLR+16+8*Y(PLR),8) FI
X(PLR)==+X1 Y(PLR)==+Y1
POKE(53248+PLR,8*X(PLR)+48)
MOVEBLOCK(PM+128*PLR+16+8*Y(PLR),
 CURSOR,8)

; WAIT! HE'S PLACING A PIECE!
IF HOLD(PLR)>0 AND STRIG(PLR)=0 AND
 PAU(PLR)=0 THEN CAP=0 OK=0 DX=0 DY=0

; SOMETHING TO CAPTURE!
 IF LOC<>32 THEN CAP=1 FI

; FIND DELTA VALUES
 DUM1=X(PLR)
 DUM2=OX(PLR)
 DX=DUM1-DUM2

 DUM1=Y(PLR)
 DUM2=OY(PLR)
 DY=DUM1-DUM2

; FLIP FOR PLAYER 2
 IF PLR=1 THEN DX=-DX DY=-DY FI

; IS IT LEGAL???

; PAWN
 IF HOLD(PLR)=1 THEN
  IF DX=1 AND DY=0 AND CAP=0 THEN
   OK=1 FI
  IF DX=2 AND DY=0 AND CAP=0 AND
   OX(PLR)=7+PLR*5 THEN OK=1 FI
  IF DX=1 AND (DY=1 OR DY=-1) AND
   CAP=1 THEN OK=1 FI FI

; ROOK
 IF HOLD(PLR)=2 THEN
  IF (DX<>0 AND DY=0) OR (DX=0 AND
   DY<>0) THEN OK=1 FI FI

; KNIGHT
 IF HOLD(PLR)=3 THEN
  IF (DX=2 AND DY=1) OR (DX=-2 AND
   DY=1) THEN OK=1 FI
  IF (DX=2 AND DY=-1) OR
   (DX=-2 AND DY=-1) THEN OK=1 FI
  IF (DX=1 AND DY=2) OR (DX=-1 AND
   DY=2) THEN OK=1 FI
  IF (DX=1 AND DY=-2) OR
   (DX=-1 OR DY=-2) THEN OK=1 FI FI

; BISHOP
 IF HOLD(PLR)=4 AND (DX=DY OR DX=-DY)
  THEN OK=1 FI

; QUEEN
 IF HOLD(PLR)=5 THEN
  IF DX=DY OR DX=-DY THEN OK=1 FI
  IF (DX<>0 AND DY=0) OR (DX=0 AND
   DY<>0) THEN OK=1 FI FI

; KING
 IF HOLD(PLR)=6 THEN
  IF (DX=1 AND DY=1) OR (DX=0 AND
   DY=1) OR (DX=-1 AND DY=1) THEN
   OK=1 FI
  IF (DX=1 AND DY=0) OR (DX=-1 AND
   DY=0) THEN OK=1 FI
  IF (DX=1 AND DY=-1) OR (DX=0 AND
   DY=-1) OR (DX=-1 AND DY=-1) THEN
   OK=1 FI FI

; CAN'T CAPTURE OWN PIECES OR
; BORDER
 IF LOC>128*PLR+41 AND
  LOC<128*PLR+127 OR LOC<10 THEN OK=0
  FI

; DIDN'T MOVE
 IF DX=0 AND DY=0 THEN OK=1 FI

; MAKE SURE JUMPS WEREN'T MADE,
; EXCEPT BY KNIGHT
 IF HOLD(PLR)<>3 THEN
  I=OX(PLR) J=OY(PLR)
  X1=0 Y1=0
  IF DX<0 THEN X1=-1 FI
  IF DX>0 THEN X1=1 FI
  IF DY<0 THEN Y1=-1 FI
  IF DY>0 THEN Y1=1 FI
  IF PLR=1 THEN X1=-X1 Y1=-Y1 FI
  IF (DX<-1 OR DX>1) OR (DY<-1 OR
   DY>1) THEN
    DO
     I==+X1 J==+Y1
     K=LOCATE(I,J)
     IF K<>32 THEN OK=0 FI
     UNTIL (I=X(PLR)-X1 AND
      J=Y(PLR)-Y1) OR K<10 OD FI FI

; LEGAL MOVE!
 IF OK=1 THEN
```

```
  COLOR=32 PLOT(OX(PLR),OY(PLR))
  COLOR=HOLD(PLR)+128*PLR+41

;  QUEEN ME!
  IF HOLD(PLR)=1 AND
   X(PLR)=7*(1-PLR)+6 THEN
   COLOR=128*PLR+46 FI

;  KILL OTHER PLAYERS HOLD IF THAT'S
;  WHAT WAS CAPTURED
  IF X(PLR)=OX(1-PLR) AND
   Y(PLR)=OY(1-PLR) THEN
   HOLD(1-PLR)=0
   POSITION(11*(1-PLR)+4,2)
   PRINTD(6," ") FI

;  WHO'D HE LAND ON??
  K=LOCATE(X(PLR),Y(PLR))

;  WHOEVER IT WAS, KILL HIM
  PLOT(X(PLR),Y(PLR))
  COLOR=32 PLOT(11*PLR+4,2)

;  A KING DIED!
  IF K-128*(1-PLR)-41=6 THEN EXIT FI
  HOLD(PLR)=0
  SND(PLR)=100*PLR+100 DIS(PLR)=14 FI

;  ILLEGAL MOVE...
  IF OK=0 THEN SND(PLR)=255
   DIS(PLR)=2 FI
PAU(PLR)=5 FI

; PICK UP PIECE
IF HOLD(PLR)=0 AND STRIG(PLR)=0 AND
 PAU(PLR)=0 AND LOC<>32 AND
 LOC>128*PLR+41 AND LOC<128*PLR+127
 THEN

; Grab HOLD
 HOLD(PLR)=LOC-128*PLR-41
 OX(PLR)=X(PLR) OY(PLR)=Y(PLR)
 COLOR=LOC PLOT(11*PLR+4,2)
 SND(PLR)=100*PLR+100 DIS(PLR)=10
 PAU(PLR)=5 FI

; DELAY
FOR CH=1 TO 2000 DO OD

; PAUSE FOR HUMANS
IF PAU(PLR)>0 THEN PAU(PLR)==-1
 SOUND(PLR,SND(PLR),DIS(PLR),
 PAU(PLR)*2) FI

; NEXT PLAYER
OD

; VICTORY ROUTINE
SNDRST() ZERO(PM,256) COLOR=32
```

```
FOR I=2 TO 9 DO FOR J=6 TO 13 DO
 LOC=LOCATE(J,I) IF LOC>128*(1-PLR)
 AND LOC<128*(1-PLR)+127 THEN
 PLOT(J,I) FI OD OD PLOT(4,2)
 PLOT(15,2)

; PAUSE
CH=0 DO CH==+1 FOR I=1 TO 100 DO OD
 UNTIL CH=7500 OR PEEK(53279)=6 OD

; START NEW GAME
OD
```