

General Information

Author: Joel Gluck

Language: ACTION!

Compiler/Interpreter: ACTION!

Published: Analog #27 (02/ 85)

More Fun with Bounce!#

Way back in the golden days, when issues of **ANALOG Computing** were still numbering in the teens, I wrote a program called **Bounce**. It appeared in the **Our Game** column in issue 15. At that time, I was fiddling with a fun new language for the Atari ? Action! by Optimized Systems Software. I was thinking that a version of **Bounce** in Action! would be a worthwhile project.

Not long after I had that thought I discovered, to my amusement, that someone had beaten me to it. The friendly folks at **ANALOG Computing** told me one day that a certain David Plotkin had submitted a little ditty called [Bounce in Action!](#), which later appeared in issue 20.

However, David's idea of a better **Bounce** program was different from mine. His improvements consisted of adding GTIA color and, of course, speed (with Action!) to the original design. I *enjoyed* playing with David's program, and I was *pleased* that someone else was as enthusiastic about **Bounce** as I was. . . I simply had another idea that had to be tried.

To me, the next natural step for **Bounce** is to add more discs?-having multiple moving objects at your command makes **Bounce** about a million times more fun than the original. Of course, Action! is the only high-level language for the Atari that is fast enough to do a multiple-object **Bounce** effectively.

First there was **Bounce**, then **Bounce in Action!**, and now I give you **More Fun with Bounce (MFB)** for short).

Other improvements.#

I had other upgrading in mind, too. Tops on the list was user-friendliness. **MFB** lets you move the cursor around freely without upsetting the walls or the discs already laid down. Drawing or erasing occurs only when your joystick trigger is held down. To switch between the two functions (drawing: and erasing). simply hit the SPACE BAR.

Another user-friendly feature is the amount of control over cursor speed available. For a slow cursor (to maintain fine drawing control), hit a lower digit key like 3 or 4. For high-speed drawing, hit 7 or 8. Cursor speed 9 is for maniacs only.

Laying down the spheres is simplicity itself. just hit the B key. A disc appears at the cursor?s position, while the cursor itself moves to the right (so you can keep laying down more). Note that?when drawing, erasing. moving or placing balls? the cursor performs automatic wraparound should it go off the edge of the screen.

Even **Bounce**?s screen-clearing feature has been improved upon. In **MFB**, when you hit ESC, instead of the whole screen clearing, only the discs disappear. This lets you keep your old wall patterns. If you'd like to clear everything, just hit CTRL-ESC. To remove individual discs, draw or erase over them with the cursor.

Let ?er rip!#

To start things bouncing. hit START. (If you forgot to lay discs down, the program automatically returns you to the drawing mode.) Immediately, the playing field fills with red goop (to be eaten away by the bouncing balls), and the number of objects bouncing appears in the text window.

Again, as in drawing mode. you have a number of options. For starters, you have complete control over disc speed. Simply hit digit keys 0 through 9, 9 being the fastest. Keep in mind, however, that the fewer objects you have on screen, the faster they will go (this is a natural by-product of the limited processing speed of your Atari computer). One or two discs on the screen at speed 9 move so fast that they are more of a blur than an object.

You may notice as the balls are bouncing that only one of them is actually making bouncing sounds; the others are silent. To change the "sound focus", hit the S key. This allows you to make different balls audible, one at a time. If you keep hitting S, you'll finally return sound focus to the original disc. This effect is easier to see if you have only a few objects on the screen

An improvement I've always wanted to add to **Bounce** is to give the user some direct control over the bouncing sphere. In **MFB**, this feature exists and is called "nudging". When you hit the N key, the ball that the sound focus is on gets nudged. The effect of this is distinct, yet simple ? it causes the ball to react as if a vertical wall were momentarily placed directly ahead of it. Essentially, the ball bounces off of a ghost wall.

Nudging is fun (as is holding down the N key for repeated nudgings) and, also, useful if there is a red area on the screen where no ball has been. You can direct one over to that area by nudging it. Note: it is best to practice nudging with only a few objects and at a slow speed. Also note: you can nudge different spheres by changing the sound focus.

When you want to change your wall configuration or the number of bouncing objects, hit SELECT to get back to the drawing mode. To start with a fresh screen, just hit CTRL-ESC.

Action! listing.

```
;More Fun with BOUNCE
;by Joel Gluck
;for ANALOG COMPUTING

BYTE ARRAY xx(256),yy(256),
            xd(256),yd(256)
BYTE xc,yc,hidden,cmode,TIME=20,
      RANDOM=53770, CONSOL=53279,
      CURSC=708,CH=764,NEWCOL=710,
      dist=[0],audball=[0]
CARD num=[0],curspeed=[1500],
      ballspeed=[900]
CARD ARRAY linept(48)

PROC gr5init()
  CARD scrn=88
  BYTE line,BALLCOL=709,WALLCOL=710

  Graphics(5)
  FOR line=0 TO 47 DO
    linept(line)=scrn+20*line
  OD
```

```
BALLCOL=$0C
WALLCOL=$94
RETURN
```

```
PROC plot5(BYTE x,y,col)
  BYTE POINTER pixel
  BYTE ARRAY colfil= [0 85 170 255],
                    mask= [63 207 243 252],
                    mask2= [192 48 12 3]
```

```
  pixel = linept(y)+(x RSH 2)
  pixel^ = pixel^ & mask(x & 3)
           % (colfil(col)
           & mask2(x & 3))
```

```
RETURN
```

```
BYTE FUNC locate5(BYTE x,y)
  BYTE POINTER pixel
  BYTE ARRAY mask= [192 48 12 3]
```

```
  pixel = linept(y)+(x RSH 2)
RETURN((pixel^ & mask(x & 3)) RSH
      ((x & 3) XOR 3) LSH 1))
```

```
PROC hline(BYTE y,c)
  BYTE i
```

```
  FOR i = 0 TO 79 DO
    plot5(i,y,c)
  OD
```

```
RETURN
```

```
PROC vline(BYTE x,c)
  BYTE i
```

```
  FOR i = 0 TO 47 DO
    plot5(x,i,c)
  OD
```

```
RETURN
```

```
PROC pauz(CARD p)
```

```
CARD i
  FOR i = 1 TO p DO
    OD
```

```
RETURN
```

```
PROC f16(BYTE x,y)
```

```
  BYTE g,a,b
```

```
  g=Locate(x,y)
  IF g=32 THEN
    RETURN
```

```
  FI
  g==+128
  NEWCOL=15
  b=y
  DO
    color=0
    plot(x,b)
    b=-1
```

```

    color=g
    plot(x,b)
    IF b=2 THEN
        EXIT
    FI
    Sound(0,b,8,8)
    pauz(700+x*50)
OD
a=x
DO
    color=0
    Plot(a,b)
    a==+1
    color=g
    Plot(a,b)
    IF a=19 THEN
        EXIT
    FI
    Sound(0,a,8,8)
    pauz(700+x*50)
OD
color=0
Plot(a,b)
SndRst()
RETURN

PROC colburst(BYTE x,y)
    BYTE g,c,a
    g=Locate(x,y)
    IF g=32 THEN
        RETURN
    FI
    g=g+128
    NEWCOL=(Rand(16) LSH 4) % 10
    color=g
    a=x-1
    IF a>13 THEN
        a=0
    FI
    Plot(x,a)
    Drawto(x,y)
    FOR c = 0 TO 15 DO
        Sound(0,0,4,15-c)
        pauz(400)
    OD
    color=0
    plot(x,0)
    Drawto(x,y)
    SndRst()
RETURN

PROC dropkick(BYTE x,y)
    BYTE g,h,a,b

    g=Locate(x,y)
    IF g=32 THEN
        RETURN
    FI
    g==+128

```

```

NEWCOL=152
b=y
DO
    color=0
    Plot(x,b)
    b==+1
    color=g
    Plot(x,b)
    IF b=23 THEN
        EXIT
    FI
    Sound(0,b+10+(x LSH 1),10,8)
    Sound(1,b+20+(x LSH 1),10,8)
    pauz(400)
OD
SndRst()
h=0
NEWCOL=159
a=x
DO
    color=h
    Plot(a,b)
    h=Locate(a+1,b-1)
    a==+1
    b== -1
    color=g
    Plot(a,b)
    IF a=18 OR b=1 THEN
        EXIT
    FI
    Sound(0,a-x,8,(b RSH 1))
    pauz(800)
OD
color=0
Plot(a,b)
SndRst()
RETURN

PROC foo()
    BYTE v

    FOR v=0 TO 15 DO
        Sound(0,255,14,15-v)
        Sound(1,0,8,8-(v RSH 1))
        pauz(500)
    OD
    SndRst()
RETURN

PROC intro()
    BYTE x

    Graphics(17)
    CURSC=$08
    Position(0,10)
    PrintD(6, "MORE FUN WITH")
    Position(0,12)
    PrintD(6, "B O U N C E ! ")
    Position(0,14)

```

```

PrintD(6, "BY JOEL GLUCK")
paуз(65000)
paуз(65000)
paуз(65000)
FOR x=0 TO 12 DO
    f16(12-x,10)
OD
FOR x=0 TO 12 DO
    colburst(x,12)
OD
FOR x=0 TO 12 DO
    dropkick(12-x,14)
OD
CURSC=$48
Position(14,1)
PrintD(6, "ANALOG")
foo()
Position(11,3)
PrintD(6, "COMPUTING")
foo()
Position(12,5)
PrintD(6, "FEBRUARY")
foo()
Position(16,7)
PrintD(6, "1985")
foo()
paуз(65000)
paуз(65000)
paуз(65000)
RETURN

```

```

PROC drawdoc()
    BYTE CURS=752

    CURS =1
    PutE()
    Print("Use joystick and ")
    PrintE("SPACE to draw/erase.")
    Print("Hit B for balls, ")
    PrintE("0-9 for brush speed.")
    Print("ESC clr balls; ")
    PrintE("CTRL-ESC clr screen.")
    Print("Press START to Bounce!")
RETURN

```

```

PROC clearscrn()
    BYTE a,b,g

    FOR b=1 TO 19 DO
        FOR a=1 TO 78 DO
            g=Locate5(a,b)
            IF (g=2 OR CH>28) AND g>1 THEN
                plot5(a,b,0)
                Sound(0,b,6,4)
                IF CH=28 THEN
                    пауз(300)
                FI
            FI
            g=Locate5(a,39-b)

```

```

    IF (g=2 OR CH>28) AND g>1 THEN
        plot5(a,39-b,0)
        Sound(0,b,6,4)
        IF CH=28 THEN
            pauz(300)
        FI
    FI
OD
Sound(0,0,0,0)
OD
IF CH>28 OR hidden=2 THEN
    hidden=0
FI
RETURN

```

```

PROC movecursor(BYTE bflag)
    BYTE g,STIK=632,TRIG=644, vol
    BYTE ARRAY v=[2 2 2 0 2 1 1 1 0 2 0
                  0 0 1 1 1 1 2 1 0 1 1]
    INT cxd,cyd

```

```

IF STIK<15 OR bflag=1 THEN
    cxd=v((STIK-5) LSH 1)-1
    cyd=v((STIK-5) LSH 1) % 1)-1
    IF bflag=1 THEN
        cxd=2
    FI
    g=hidden
    IF TRIG THEN
        vol=4
    ELSE
        vol=10
        g=cmode*3
    FI
    Sound(0,(xc+yc)*cmode,
          8+(cmode LSH 1),
          vol-(cmode LSH 1))
    plot5(xc,yc,g)
    xc==+cxd
    yc==+cyd
    IF xc<1 THEN
        xc=78
    FI
    IF xc>78 THEN
        xc=1
    FI
    IF yc<1 THEN
        yc=38
    FI
    IF yc>38 THEN
        yc=1
    FI
    hidden=locate5(xc,yc)
    plot5(xc,yc,1)
FI
RETURN

```

```

PROC audlayball()

```

```

BYTE i,j,k

FOR j=0 TO 2 DO
  FOR i=j*50 TO j*50+20 DO
    Sound(0,i,10,15-j*6)
    pauz(100)
  OD
OD
Sound(0,0,0,0)
RETURN

BYTE FUNC number()
  BYTE n,v

  v=CH
  Open(2,"K:",4,1)
  n=GetD(2)
  Close(2)
  CH=v
  IF n>47 AND n<58 THEN
    RETURN(57-n)
  ELSE
    RETURN(99)
  FI

PROC audcmode()
  BYTE n

  FOR n=1 TO 5 DO
    IF cmode THEN
      Sound(0,100-n*10,10,4)
    ELSE
      Sound(1,150-n*10,10,4)
      Sound(0,5-n,8,6)
    FI
    pauz(2000)
    SndRst()
    pauz(1000)
  OD
RETURN

PROC cursor()
  BYTE n

  IF CH<>255 THEN
    IF CH=33 THEN
      cmode==XOR 1
      audcmode()
    ELSEIF CH=28 OR CH=156 THEN
      clearscrn()
    ELSEIF CH=21 THEN
      hidden=2
      plot5(xc,yc,2)
      movecursor(1)
      audlayball()
    ELSE
      n=number()
      IF n<99 THEN
        curspeed=n*500
      ENDIF
    ENDIF
  ENDIF

```



```

        FI
    FI
    CH=255
    FI
    movecursor(0)
RETURN

```

```

PROC bouncedoc()
    CARD n

    PutE()
    n=num
    IF n=1 THEN
        PrintE("1 ball is bouncing.")
    ELSE
        PrintC(n)
        PrintE(" balls are bouncing.")
    FI
    PrintE("Hit digits 0-9 for speed.")
    Print("S changes sound focus, ")
    PrintE("N nudges ball.")
    Print("Press SELECT to Draw again.")
RETURN

```

```

PROC process(BYTE a,b)
    BYTE g

    g=locate5(a,b)
    IF g=2 THEN
        IF num<200 THEN
            xx(num)=a
            yy(num)=b
            num==+1
        ELSE
            plot5(a,b,0)
        FI
    ELSEIF g=0 THEN
        plot5(a,b,1)
    FI
RETURN

```

```

PROC ballinit()
    BYTE a,b

    CURSC=$44
    num=0
    FOR b=1 TO 19 DO
        FOR a=1 TO 78 DO
            process(a,b)
            process(a,39-b)
        OD
    OD
    FOR a=0 TO num DO
        xd(a)=Rand(2) LSH 1
        yd(a)=Rand(2) LSH 1
    OD
RETURN

```

```

PROC moveball(BYTE n)

```

```

BYTE g,pa,pb

g=locate5(xx(n)+xd(n)-1,yy(n)+yd(n)-1)
IF g<2 THEN
  plot5(xx(n),yy(n),0)
  xx(n)=xx(n)+xd(n)-1
  yy(n)=yy(n)+yd(n)-1
  plot5(xx(n),yy(n),2)
  IF n=audball THEN
    dist==+1
  FI
  RETURN
ELSE
  pb=locate5(xx(n),yy(n)+yd(n)-1)
  pa=locate5(xx(n)+xd(n)-1,yy(n))
  IF n=audball THEN
    IF dist THEN
      Sound(0,170-((38-dist) LSH 2),10,8)
      Sound(1,((38-dist) LSH 2),10,8)
    FI
    dist=0
    TIME=0
  FI
  IF pa>1 THEN
    xd(n)=2-xd(n)
    IF pb>1 THEN
      yd(n)=2-yd(n)
      RETURN
    ELSE
      plot5(xx(n),yy(n),0)
      yy(n)=yy(n)+yd(n)-1
      plot5(xx(n),yy(n),2)
      RETURN
    FI
  ELSEIF pb>1 THEN
    yd(n)=2-yd(n)
    plot5(xx(n),yy(n),0)
    xx(n)=xx(n)+xd(n)-1
    plot5(xx(n),yy(n),2)
    RETURN
  ELSEIF Rand(2) THEN
    xd(n)=2-xd(n)
  ELSE
    yd(n)=2-yd(n)
    RETURN
  FI
FI
RETURN

PROC cleanup()
  BYTE a,b

  FOR b=1 TO 19 DO
    FOR a=1 TO 78 DO
      IF locate5(a,b)=1 THEN
        plot5(a,b,0)
      FI
      IF locate5(a,39-b)=1 THEN
        plot5(a,39-b,0)
      FI
    DO
  DO

```

```
        FI
    OD
OD
RETURN
```

```
PROC bounce()
    CARD i
    BYTE n

    ballinit()
    bouncedoc()
    audball=0
    dist=0
    IF num THEN
        DO
            FOR i=0 TO num-1 DO
                moveball(i)
                IF CH<>255 THEN
                    IF CH=62 THEN
                        audball==+1
                        IF audball=num THEN
                            audball=0
                        FI
                        dist=0
                    ELSEIF CH=35 THEN
                        xd(audball)=2-xd(audball)
                    ELSE
                        n=number()
                        IF n<99 THEN
                            ballspeed=n*n*100
                        FI
                    FI
                    CH=255
                FI
                IF CONSOL=5 THEN
                    EXIT
                FI
            OD
            pauz(ballspeed)
            IF TIME THEN
                SndRst()
            FI
            UNTIL CONSOL=5
            OD
            SndRst()
        FI
    cleanup()
RETURN
```

```
PROC MFWB()

    intro()
    gr5init()
    hline(0,3)
    hline(39,3)
    vline(0,3)
    vline(79,3)
    DO
        drawdoc()
```

```
xc=39
yc=19
hidden=locate5(xc,yc)
cmode=1
plot5(xc,yc,1)
DO
  cursor()
  CURSC=TIME
  pauz(curspeed)
  Sound(0,0,0,0)
  pauz(curspeed)
  UNTIL CONSOL=6
OD
plot5(xc,yc,hidden)
CH=255
bounce()
OD
RETURN
```