More Fun with Bounce!

24K Cassette or 32K Disk
by Joel Gluck

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 dirty 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. □
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=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=19 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 foot()
BYTE y

FOR v=0 TO 15 DO
Sound(0,255,10,15-v)
Sound(1,0,0,0-(v RSH 1))
Pauz(900)
OD
SndRst()
RETURN

PROC intro()
BYTE x

Graphics(17)
CUR5=508
Position(0,10)
PrintD6("MORE FUN WITH")
Position(0,12)
PrintD6("BOUNCE")
Position(0,14)
PrintD6("BY JOEL GLUCK")
Pauz(65000)
Pauz(65000)
Pauz(65000)
FOR x=0 TO 12 DO
fig(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
CUR5=548
Position(14,1)
PrintD6("ANALOG")
foot()
Position(11,3)
PrintD6("COMPUTING")
foot()
Position(12,5)
PrintD6("FEBRUARY")
foot()
Position(16,7)
PrintD6("1985")
foot()
Pauz(65000)
Pauz(65000)
Pauz(65000)
RETURN

PROC drawdoc()
BYTE CUR5=752

CUR5=1
PutE()
Print("Use joystick and ")
PrintE("SPACE to draw/erase.")
Print("Hit 1 for Balls, ")
PrintE("0-8 for brush speed.")
PrintE("ESC to draw screen.")
PrintE("CTRL-ESC to clear screen.")
PrintE("")
Print(""Press START to Bounce!")
RETURN
PROC clearscrn() 
BYTE a,b,g 
FOR b=1 TO 19 DO 
FOR a=1 TO 79 DO 
g=locate5(a,b) 
IF (g=2 OR CH>28) AND g<i THEN 
plot5(a,b,0) 
sound(0,0,6,4) 
IF CH>28 THEN 
pauz(300) 
FI 
FI 
g=locate5(a,39-b) 
IF (g=2 OR CH>28) AND g<i THEN 
plot5(a,39-b,0) 
sound(0,0,6,4) 
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=1222111102000111111111 
INT cx,cd,cy 
IF STIK=15 OR bflag=1 THEN 
cx=v((STIK-5)LSH 1)-1 
cy=v((STIK-5)LSH 1) % 1)-1 
IF bflag=1 THEN 
cx=x 
FI 
g:hidden 
IF TRIG THEN 
vol=4 
ELSE 
vol=10 
g=cmodex 
FI 
sound(0,10*(cy+cx)*cmodex,8+cmodexLSH 1),vol-cmodexLSH 1)) 
plot5(cx,cy,9) 
xc=cx 
yc=cy 
IF xc(i THEN 
xc=78 
FI 
IF yc(i THEN 
yc=38 
FI 
IF yc)38 THEN 
yc=1 
FI 
hidden=locate5(cx,cy) 
plot5cx,cy,1) 
FI 
RETURN 

PROC audlayball() 
BYTE i,j,k 
FOR i=0 TO 2 DO 
FOR j=k50 TO j+50+20 DO 
sound(0,i,10,15-j*6) 
pauz(100) 
OD 
OD 
sound(0,0,0,0) 
RETURN 

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,0,6) 
FI 
pauz(2000) 
SendRst() 
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,cy,2) 
movcurscr() 
audlayball() 
ELSE 
n=number() 
IF n<99 THEN 
curspeed=n*500 
FI 
CH=255 
FI 
movcurscr() 
RETURN 

PROC bouncedoc() 
CARD n 
PutE() 
n=NUM 
IF n=1 THEN 
PrintE("1 ball is bouncing.") 
ELSE 
PrintE(n) 
PrintE(" balls are bouncing.") 
FI 
PrintE("Hit digits 0-9 for speed.") 
PrintE(" 2 changes sound focus, ") 
PrintE(" 0 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 
x(num)=a 
y(num)=b 
um=num+1 
ELSE 
plot5(a,b,0) 
FI