Q. What is the Applesoft collision counter, and how do you use it with shapes in a program?

A. The collision counter is a one-byte memory location ($EA = 234 decimal) whose purpose is to detect when a shape drawn in the current HCOLOR has overdrawn a pixel of the same color on the Hi-Res screen. After DRAWing a shape, the counter will be equal to the number of points on the screen that were intercepted by the drawn shape.

It is important to realize that this includes the situation where you are DRAWing a shape with HCOLOR=0 (black) on a black background; the collision counter will show the number of collisions equal to the number of points in the shape, even though nothing is visible on the screen.

If instead of DRAWing the shape, you use XDRAW, then the current HCOLOR has no effect on the collision counter result. Instead, the collison counter returns the count of those pixels converted from "off" to "on" by the XDRAW action.

The collision counter can be used in Hi-Res games, or even to simulate the SCRN function of Lo-Res graphics, which is otherwise missing in Hi-Res.

A simple way to experiment with the collision counter is with the following program. This program uses the now famous "one-dot-shape" to create a radar screen-like "scanner." The arm of the scanner is created by updating the SCALE and ROT values before DRAWing or XDRAWing the shape.

The first program (Listing 1) DRAWs the scanning line, and then checks the collision counter afterward to see if any dots on the screen were intercepted. If so, the program prints the contents of the collision counter and waits for a keypress. The line is erased each time by DRAWing in black, which has the effect of erasing any dots the scan line encounters.

For the second program (Listing 2), XDRAW is used, and the collision counter is checked after the line is erased with a second XDRAW action. With XDRAW, the collision counter will always have a high count after drawing a line on a black background. The test for a collision is done by changing the entire path back to black. The collision counter will be zero if the entire line is converted (no dots left "on"). If any dots are left on after the XDRAW, the collision counter will be greater than zero, and our program will detect this. Depending on your program, and whether the background is white or black, different approaches will be required. Notice how the use of XDRAW leaves the background undisturbed by the scanning line.

With a SCALE value of 1 and a ROT value of 0, you can XDRAW a single-dot shape anywhere on the screen twice, and then look at the collision counter to see if that dot was turned on. This can be used the way SCRN is in Lo-Res. Listing 3 is a more interesting demo of using the single-dot shape: a program called the "Hungry Dot," taken in slightly abbreviated form from a demo program of the same name on the SoftSwitch disk. Hungry Dot bounces a single dot around on the Hi-Res screen. Whenever the dot hits something, as detected by the collision counter, there's an "explosion," and the offending screen dot is eaten. The velocity of the hungry dot is then recalculated, and the process continues. Should the dot make it to the outside screen border, an Applesoft "ILLEGAL QUANTITY ERROR" is trapped, and the dot is reset to the center of the screen. The program is quite hypnotic — I hope you enjoy it!

LISTING 1: Collision Counter Demo 1

```
10  POKE 768,1: POKE 769,0: POKE 770,4
20  POKE 771,0: POKE 772,4: POKE 773,0
30  POKE 232,0: POKE 233,3: REM  $E8,E9=$300
40  HGR : HCOLOR= 3
50  HOME
60  GOSUB 190
70  REM  'DRAW' SCANNER
80  SCALE= 80
90  FOR R = 0 TO 63
100  ROT= R
110  HCOLOR= 3: DRAW 1 AT 140,80: REM  DRAW L
     INE
120  VTAB 22: PRINT  PEEK (234);"  "
130  IF  PEEK (234) > 0 THEN  GET A$: IF A$ =
     "X" THEN
140  FOR I = 1 TO 25: NEXT I: REM  DELAY LOOP
150  HCOLOR= 0: DRAW 1 AT 140,80: REM  ERASE
     LINE
160  NEXT R
170  HCOLOR= 3: GOSUB 190: REM  RE-CREATE FIE
     LD
180  GOTO 90
190  REM  CREATE FIELD
200  HPLOT 140,100: HPLOT 140,101
210  RETURN
```

LISTING 2: Collision Counter Demo 2

```
10  POKE 768,1: POKE 769,0: POKE 770,4
20  POKE 771,0: POKE 772,4: POKE 773,0
30  POKE 232,0: POKE 233,3: REM  $E8,E9=$300
40  HGR : HCOLOR= 3
50  HOME
60  GOSUB 180
70  REM  'XDRAW' SCANNER
80  SCALE= 80
90  FOR R = 0 TO 63
100  ROT= R
110  XDRAW 1 AT 140,80: REM  XDRAW 1ST IMAGE
120  FOR I = 1 TO 25: NEXT I: REM  DELAY LOOP
130  XDRAW 1 AT 140,80: REM  'ERASE' IMAGE
140  VTAB 22: PRINT  PEEK (234);"  "
150  IF  PEEK (234) > 0 THEN  GET A$: IF A$ =
     "X" THEN
160  NEXT R
170  GOTO 90
180  REM  CREATE FIELD
190  HPLOT 140,100: HPLOT 140,101
200  RETURN
```