Experimenting with Colour on the Risc PC

by Lee Calcraft

Although the good old Archimedes could only handle a maximum of 256 colours on screen at once, you could actually create pretty impressive graphics with it, by careful choice of colours, or by dithering to create the required effect. But one of the most restricting features of the 256-colour modes was the cumbersome way in which colours were defined. Working out the correct colour number and the amount of tint was far from intuitive.

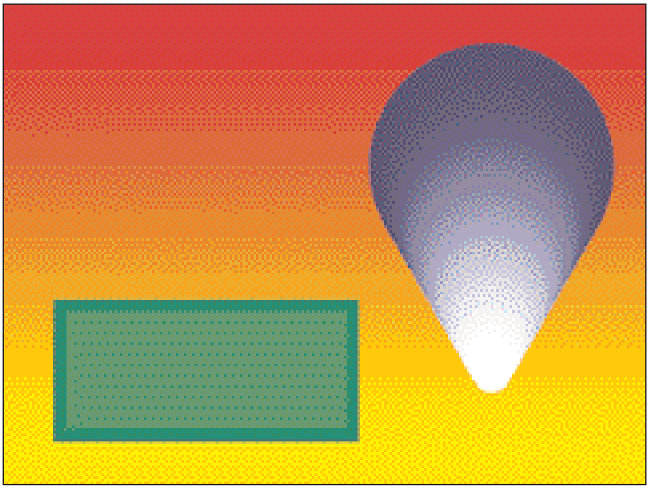

Figure 1.
Lozenge and washed cone against a yellow-red wash backdrop

One of the many blissful things about the Risc PC is that RISC OS 3.5 allows you to specify colours in a far more natural way as true combinations of red green and blue. Thus the command to set a foreground text colour is simply:

```
    COLOUR r,g,b
```

where r, g and b are integers between 0 and 255. The same is true for setting the graphics foreground colour:

```
    GCOL r,g,b
```

Thus the three line program:

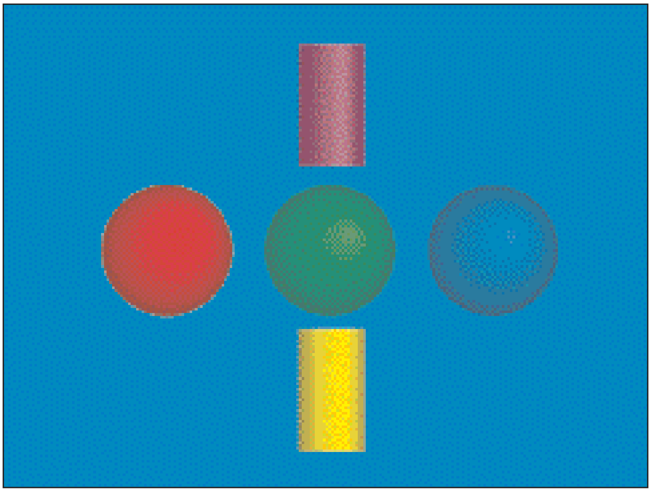

Figure 2.
Highlighted spheres and cylinders

```
    MODE "X800,Y600,C32K,EX1,EY1"
    GCOL 255,0,0
    RECTANGLE FILL 400,400,300,100
```

will put you in a 32,000-colour mode, and draw a red rectangle.

Setting the background colour for graphics is achieved by preceding the three colour values with the number 128. Thus to clear the screen to a medium blue, you could use something like the following:

```
    GCOL 128,0,100,200:CLG
```

or for black:

```
    GCOL 128,0,0,0:CLG
```

And as a quite unexpected bonus, if you use the rgb forms of the GCOL command while in a 256 or even a 16-colour mode, dithering (via ECF patterns) is automatically used in an attempt to recreate the required shade. The same is true of 32,000-colour modes, where the dithering is particularly effective. If you want to avoid dithering in 32,000-colour modes, you should only use colour numbers that are divisible by 8.

COLOUR RUNS

Because it is so easy to describe any desired colour, it is a simple matter to generate colour runs with any given colour gradually fading into any other. For example, the following short program fills the screen with a colour wash which goes from red at the top of the screen through orange to yellow at the bottom. I have used a 32,000-colour mode for this, and there is only barely perceptible banding. Using a 16-million colour mode would probably eliminate it completely - though of course you will need 2Mb of VRAM for that.

```
MODE "X800,Y600,C32K,EX1,EY1"
OFF
Y=1200
step=Y/255
FOR A=0 TO 255
 GCOL 255,A,0
 RECTANGLE FILL 0,Y-A*step,1600,10
NEXT
A=GET
```

GREEN LOZENGE

By using a similar technique, it is possible to create a wide variety of objects with a 3D feel to them. For example, a simple lozenge shape can be created by drawing a series of filled rectangles, each with the same centre, but with decreasing width and height. If you brighten the colour with each step then the object appears to be raised, while the reverse will make it appear to be sunken. For example, the following will create a bright green lozenge.

```
REM Green lozenge
X=100:Y=100
H=200:W=400
FOR A=100 TO 255
 GCOL 0,A,0
 RECTANGLE FILL X+A/4,Y+A/4,W-A/2,H-A/2
NEXT
```

WASHED CIRCLES AND SPHERES

What you can do with rectangles you can also do with circles. The following four lines create a conical shape by progressively reducing the radius and y coordinate of a succession of filled circles at the same time as changing their hue from cyan to blue:

```
FOR A=0 TO 255
 GCOL 0,255-A,255
 CIRCLE FILL 600,800-2*A,300-A
NEXT
```

It is also a simple matter to create a good approximation to a lit sphere by progressively reducing the radius of a filled circle at the same time as slowly moving the centre diagonally upwards, and lightening the colour. The following few lines illustrate the technique, to create a red sphere.

```
REM Red sphere
X=300:Y=500:R=260
FOR A=100 TO 255
 GCOL A,0,0
 CIRCLE FILL X+A/3,Y+A/4,R-A
NEXT
```

If you replace the GCOL line by:

```
GCOL 0,A,0
```

you will get a bright green sphere, while:

```
GCOL 0,2*A/3,A
```

will give a blue one. In this latter case I have added in a component of green (i.e. 2*A/3) to the highlighting to make it more effective.

Highlighted cylinders can be created by a similar approach. To make a vertical cylinder you need to draw a sequence of filled rectangles whose width, but not height, steadily decreases. At the same time just move the horizontal centre slightly to one side, and brighten the fill colour as you do it. Here s an example in red:

```
H=300:W=260
X=300:Y=800
FOR A=100 TO 255
 GCOL A,0,0
 RECTANGLE FILL X+2*A/3,Y,W-A,H
NEXT
```

READING COLOURS

In Archimedes Basic you can read the colour at any point on the screen using the POINT function. For example:

```
colour=POINT(X,Y)
```

will set the variable colour to a value representing the colour at screen coordinate X,Y. This can be useful in a graphics program where you want to pick up the colour under the pointer, or in a games program where you want to detect collisions.

This function works in a similar way in

That s about all we have room for I m afraid - more on the Risc PC next month. Meanwhile, on this month s disc there is a program which incorporates all the examples given here.