All microcomputers have a maximum density of dots that can be displayed on their graphics screens, but some computers can also be programmed in ways that use less dense arrays of dots. For example, the BBC Microcomputer has a maximum resolution of 640 x 256; that is, it has a maximum dot density given by 256 rows each containing 640 dots. However, it can also be programmed using only 320 of these columns, a resolution of 320 x 256, or even lower resolutions if required. On a machine that is capable of providing this kind of variation, the resolution must be set at the beginning of the part of the program that produces the graphics.

Lines and curves produced by a dot-based computer graphics system are not actually continuous, as they would be if they were drawn with a pen. Instead they are paths of illuminated dots that merge more or less closely depending on the resolution of the system. One with low resolution will be able to produce only lumpy curves, and a straight line will not necessarily be perfectly straight either, since the dot positions on the screen lie in straight lines only in certain directions — such as along rows and columns, and across the diagonals of the array of dots. To plot a straight line the graphics system must illuminate the dots that lie closest to the path of the chosen line. The result can be a 'staircase' effect. Again, the higher the resolution of the system, the less noticeable such 'staircases' will be.

Because graphics are shown on a television screen, in order to give continuity of display the image must be continually 'refreshed' or 'redrawn' — otherwise it would appear only for a moment and then fade. For this reason, the image must be represented in some way in the computer, so that it can be referred to when necessary. The representation of the image is actually stored in a special area of the computer's memory known as the *screen memory*. With a monochrome display, each dot on the screen corresponds to a bit in the screen memory. An image is represented by setting the bits that correspond to illuminated dots to 1 and leaving the bits corresponding to unlit dots at 0. Thus, if a computer can maintain a monochrome display with a resolution of 256 x 192 it must have a screen memory of 256 x 192 or 49,152 bits — which is 6 kilobytes, since 1 kilobyte equals 8,192 bits.

With colour graphics more memory is required. Two bits can be used to represent four different colours, perhaps as follows:

bit 1	bit 2	colour
0	0	white
0	1	red
1	0	blue
1	1	black

To represent any four-colour image, it is necessary to assign two bits to each dot on the screen. In the same way, for eight colours three bits in the screen memory are assigned to each dot on the screen, while for 16-colour displays there are four bits per dot, and so on. So, a microcomputer that can display colours with a resolution of 160 x 256, must have a screen memory of 160 x 256 x 4 which is equivalent to 160 kilobits (20 kilobytes).

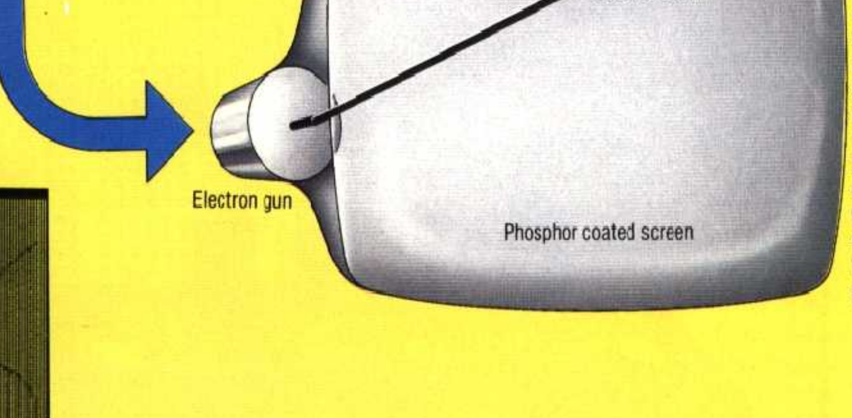
The need for screen memory explains in part why some computers are designed to work at different resolutions. Without a sufficient screen memory the computer cannot store and therefore cannot display high resolution pictures

Creating Graphics

Building images on the home computer can be easy. Most models have special commands in BASIC to help 'draw' or define objects on the screen ranging from invader-style aliens to whole pictures. Special software is also available to create animated pictures on the screen.

Image Building

Here's how signals produced by the computer get converted into images on the television screen. The TV picture is built up line by line, and the input signal which produces the illuminated portions on each line is timed so that they all fit together to form the shapes, lines and characters (see below)



The signal from the computer is fed into the television via the aerial socket. When the signal is high, the electron beam is switched on as it scans across the screen. When the signal is low it is switched off. The beam moves rapidly across the width of the screen, then zips back and starts the next line slightly lower. Hundreds of lines like this are needed to cover the entire screen. When the beam reaches the bottom of the screen it moves back to the top again and repeats the process