

The original Commander Keen, Commander Keen in Invasion of the Vorticons, was only released for the EGA videocard. Keen Dreams and later versions included a CGA version as well. The game play was exactly the same, sounds were the same, it was just that the graphics were CGA. Before diving into the source code, let's first get a better understanding of the CGA video hardware.

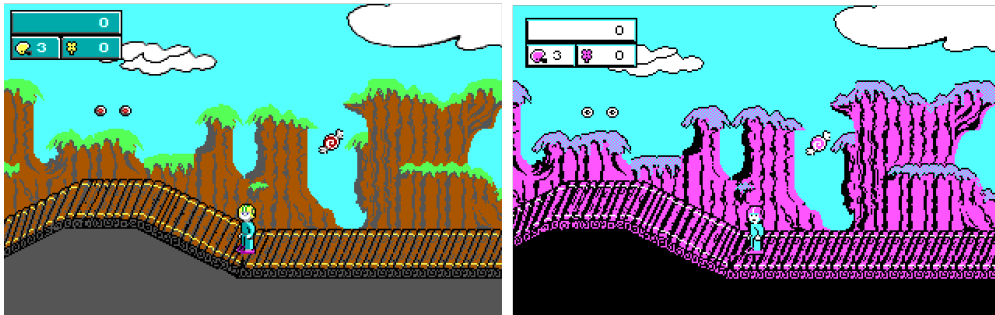


Figure 1: Keen Dreams EGA and CGA version.

Trivia : It's an ironic twist that Softdisk did not use the original Keen's engine, as the code violated the company policy by depending on 16-color EGA hardware without supporting older 4-color CGA cards!

0.1 CGA Videocard

The Color Graphics Adapter (CGA), originally also called the Color/Graphics Adapter or IBM Color/Graphics Monitor Adapter, introduced in 1981, was IBM's first color graphics card for the IBM XT.

The CGA card can be summarized by the following hardware:

- It was built around the Motorola 6845 display controller.
- The framebuffer (the VRAM) contained two memory banks of 8 kilobytes each, resulting in 16 kilobytes total.
- Character generator ROM, containing a 14-row font and two 8x8 fonts (same as used on the MDA videocard).

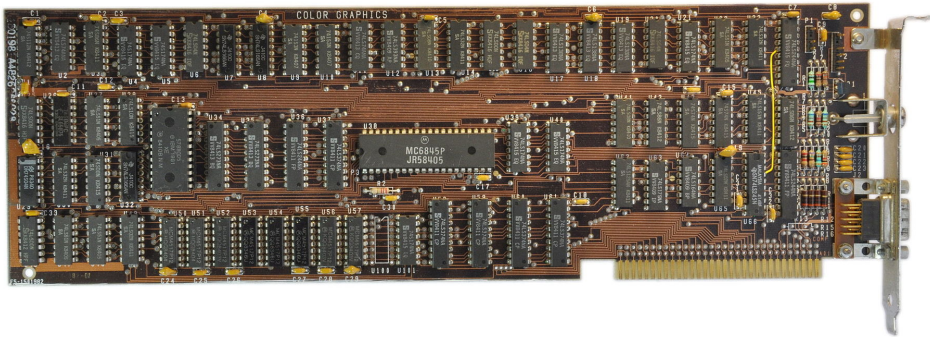


Figure 2: The CGA is a full-length 8-bit ISA card.

The CGA card has the following text and graphics modes:

Mode	Type	Format	Colors	RAM Mapping	Hz
0	text	40x25	16 (monochrome)	B8000h	60
1	text	40x25	16	B8000h	60
2	text	80x25	16 (monochrome)	B8000h	60
3	text	80x25	16	B8000h	60
4	CGA Graphics	320x200	4	B8000h	60
5	CGA Graphics	320x200	4 (monochrome)	B8000h	60
6	CGA Graphics	640x200	2	B8000h	60

Figure 3: EGA Modes available.

In the graphics mode 4, which is used by Commander Keen, each pixel is using 2 bits for color, resulting in only four colors being displayed at a time. These four colors could not be freely chosen from the 16 CGA colors, there were only two official palettes for this mode:

1. Magenta, cyan, white and background colour (black by default).
2. Red, green, brown/yellow and background colour (black by default).

The background color could be any of the 16 colors, but often it was kept black. For each mode there is a high- and low-intensity version of the palette.

Palette 1		Palette 2	
low intensity	high intensity	low intensity	high intensity
0 - Background	- Background	0 - Background	0 - Background
2 - Green	10 - Bright Green	3 - Cyan	11 - Bright Cyan
4 - Red	12 - Bright Red	5 - Magenta	13 - Bright Magenta
6 - Brown	14 - Yellow	7 - Bright Grey	15 - White

Figure 4: CGA color palettes.

The default palette when switching to Mode 4 is palette 2 with high intensity, which is used by Commander Keen. Beside the limitation in number of colors, there are several other challenges to overcome:

- The display memory was interlaced.
- The display memory was not sufficient to support double buffering.

0.1.1 Memory architecture and Interlacing

The first complication of CGA is that the display memory in graphics modes was interlaced. Normally, video memory is strictly linear: the next row of display data corresponds to the next row of pixels. But with CGA, the next row of display data corresponded to the row of pixels two rows down. This continued until the end of the screen and only with the second half of display data were the in-between rows addressed. So the first half of display memory was for rows 0, 2, 4, etc., until the end of the screen and the second half of CGA RAM was for rows 1, 3, 5, etc. This added calculation steps to most CGA graphics operations if the programmer wanted to avoid visual artifacts when updating the screen.

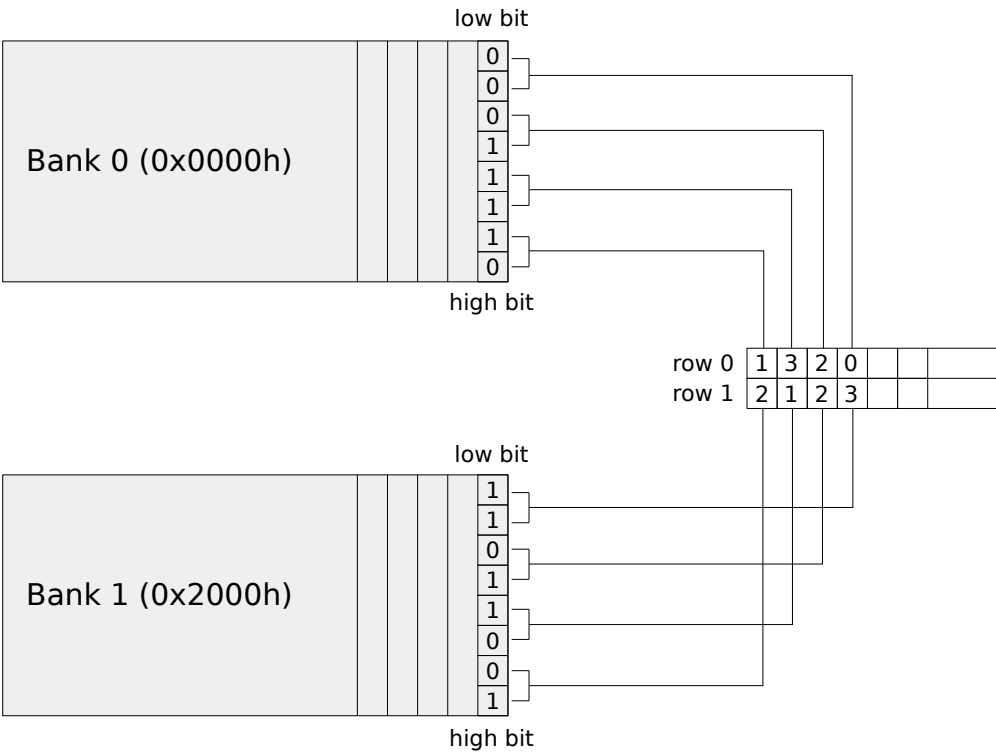


Figure 5: CGA interlaced memory.

Another complication of CGA is the size of VRAM. Each graphics modes requires all 16KiB of memory, so it is not possible to have double buffering in VRAM.

0.1.2 Double buffering

0.2 TO BE INCLUDED

The organization of the memory, 16K bytes by 8 bits, gave rise to one of the problems of CGA: In 80-column text mode, the adapter had to fetch 160 bytes per line- 80 characters and 80 attributes- leaving no time for the processor to read from or write to the display. Any program that went directly to the display memory quickly filled the screen with "snow." The read and write routines in the BIOS waited for horizontal retrace before accessing the memory. The scroll routine in the BIOS, which had to move lots of data, simply turned off the display, did the move, and then turned the display back on. There was a noticeable blink as the screen scrolled, but it was less objectionable than the blizzard that would have

occurred otherwise.

