

A brief intro:


Raster's display: all displays are generate line by line. (CRT)

v blank: $\frac{1}{60}$ s. \Leftarrow this is the refreshing time.

LCD & LEDs: Bitmap displays:

2D grid of pixels where each pixel is updated independently or is mapped to a particular video memory location

These pixels are display in a combination of colours. implemented by a set of 3 subpixels, which could be

8 colors (8 bit) $\Leftarrow 2^3$ colours 

patterns might differ between panels.

primary colour: Red Yellow Blue

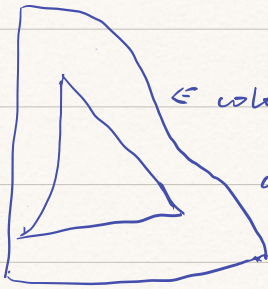
Color model: A way that human describe colors.

Additive: RGB. Closer to white when adding
i.e. Green + Blue \Rightarrow Yellow.

Subtractive: RYB. Green + Purple \Rightarrow Dark Red

mixing of colors takes away luminous, and closer to black.

CMYK is also subtractive. \Rightarrow printer.
cyan magenda yellow black



← color space: space of colors that could generate.

adjoin color model with coordinate space.

(usually integers)

RGB 16: 16 bits RGB $\Rightarrow R, G, B \in [0, 2^5 - 1]$

RGBA 16: alpha takes transparency, i.e. this pixel
↑
alpha, 1 bit. is on/off.

RGB 24: $R, G, B \in [0, 2^8 - 1]$ (True color)

standard

\Rightarrow approx 16m colors.

Human could see approx 10 m colors

\Rightarrow but these are not all in those 16m

RGB 30: deep color, $[0, 2^{10}]$

PNG files \rightarrow RGBA 32: $R, G, B, A \in [0, 2^8 - 1]$.

each pixel is 4 bytes nicely

Palette: A lookup table of pre-defined colors.

↑
LUT.

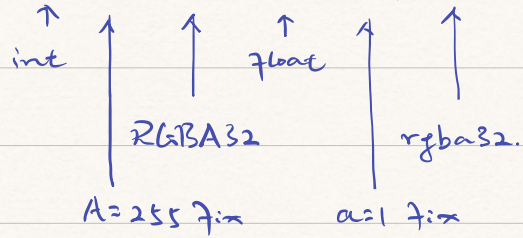
so each color could be just an index in LUT

it is much smaller than true color, but faster

respond, and it is abandoned.

rgb/rgba $\in [0, 1]$ 32 bit floats, $4 \times 4 = 16$ bytes each pixel.

in OpenGL, glColor 2i, 3i, 4i, 2f, 3f, 4f



Hex for RGBA/RGB 24 is 2 hex digit each colors.

e.g. #FF16BA ← some kind of purple/pink.

R G B