Alive Dead Media

Day 5, eh, sort of day 4 but anyway...

Bitplanes and palette

We've heard about these things already and now let's try how they are in practice

- Typical of the 16-bit computers, Commodore Amiga, Atari ST and many PC graphics modes (EGA/VGA)
- Replaced by the more straightforward but memory-hungry "chunky" modes
- Let's revisit 1st day slides again

Bitplanes

- The amount of bitplanes dictates how many colors are available, 2^n:
  - 1 -> 2 colors
  - 2 -> 4 colors
  - 3 -> 8 colors
  - 4 -> 16 colors (Atari ST maximum, PC planar modes)
  - 5 -> 32 colors
  - All the way up to 8 -> 256 colors (later Amiga models)
- By collecting one bit from each bitplane the graphics chip decides the color number

Bitplane pros and cons

+ Memory-efficient

+ Changing just one bit can change a full pixel, many can be changed quickly

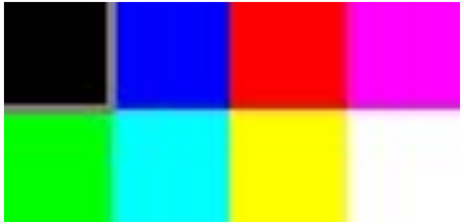
+ Planes are largely independent layers

- Hard to access individual pixels, need for bit shifting and logical operations

- Plotting a single pixel may require touching each bitplane

- Vertical placement easy, horizontal difficult

Modifiable palettes



- Most 8-bit computers only had a fixed palette (with the Amstrad CPC as a notable exception)
  - Typically 8 or 16 fixed colors
- 16-bit computers let you choose *each color* from a larger RGB set:
  - Atari ST, 3-bit RGB components -> 512 colors
  - Amiga, 4-bit RGB components -> 4096 colors
  - VGA, 6-bit RGB components -> 262144 colors
- In essence a three-dimensional RGB cube

Palette tricks

A modifiable palette lets us do certain things conveniently:

- Changing a large area to another color quickly
- Flashing the screen, fading to black or other color
- Small repetitive animations ("color cycling")
- Combined with bitplanes we can make transparent and translucent layers
- Changing the palette ("racing the beam") while the screen is drawn we get more colors on screen
- Let's see some examples again

Time to code

Next we'll try go deal with bitplanes and palettes ourselves:

- Download *bitplanerender.pde*
- Make a new sketch and a new tab with the contents
- First let's try setting the colors
- Here we have 4-bit RGB components (0..15)
  - By default the ugly IBM PC standard colors
- And four bitplanes, yielding 16 different colors in total
- Next goal: setting a full pixel to a certain color

Rehearse: binary and hex numbers

decimal	hex	binary	decimal	hex	binary
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	10	A	1010
3	3	0011	11	B	1011
4	4	0100	12	C	1100
5	5	0101	13	D	1101
6	6	0110	14	E	1110
7	7	0111	15	F	1111

Rehearse: logical operations

0011 **AND** 1010 ------- 0010	0011 **OR** 1010 ------- 1011
0011 **XOR** 1010 ------- 1001	**NOT** 10 = 01

Rehearse: bit shifting

0110 0111 **>>>** 1 ------------- 0011 0011	0110 0111 **<<** 1 ------------- 1100 1110
0110 0111 **>>>** 4 ------------- 0000 0110	0110 0111 **<<** 4 ------------- 0111 0000

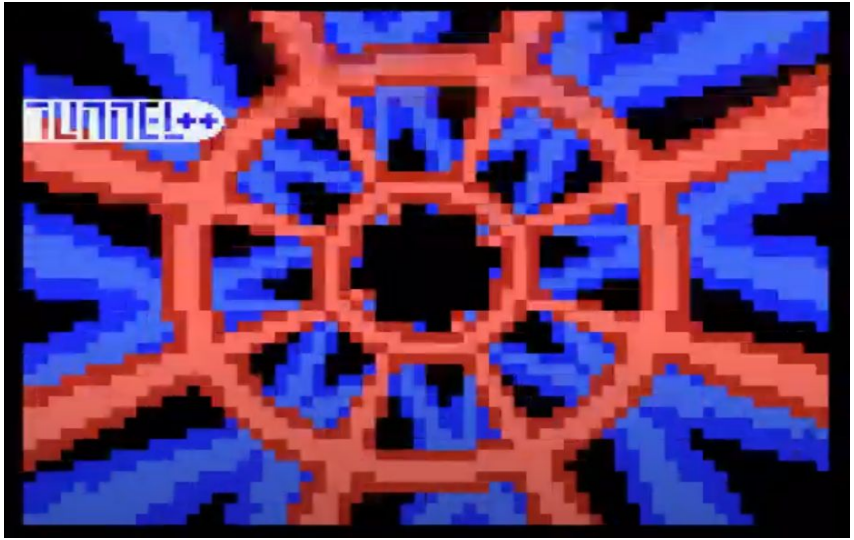
Screen buffer structure (bitplanes)

Bitplane 0	0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000 0000 0000 0000	… (640x480 bits = 9600 *int* numbers)
Bitplane 1	0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000 0000 0000 0000	… (640x480 bits = 9600 *int* numbers)
Bitplane 2	0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000 0000 0000 0000	… (640x480 bits = 9600 *int* numbers)
Bitplane 3	0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0000 0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000 0000 0000 0000	… (640x480 bits = 9600 *int* numbers)

Effects: 3D Starfield

Here in *Plan-B* by Sonic PC (1993)

Effects: Tunnel



Avaakkus by Lieves!Tuore (1998) on the MSX

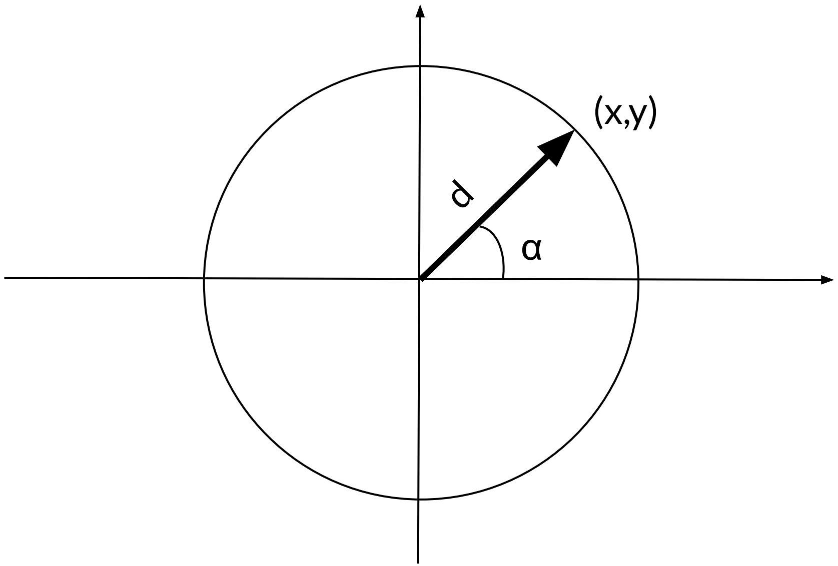
Cartesian and polar coordinates

$x = d * \cos \alpha$
$y = d * \sin \alpha$

$d = \sqrt{x^2 + y^2}$
$\alpha = \arctan y/x$

In **Processing**, square root is **sqrt()**, but **dist()** does what you probably want

For arctan there is **atan2(y,x)**



I.e. from point coordinates to angles/distances and back

Takeaways

- Consumer computer graphics are more than 40 years old
- Many different competing and disappeared paradigms
- Ingenious game and demo programmers have explored the hardware to do the impossible
- Not just tech: economy, politics, popular culture, trends and the community affect things
- After this course I wish you…
    - Know more about the history of computer graphics
    - Can analyze old software – why did it look like this?
    - Got programming experience that is applicable elsewhere too