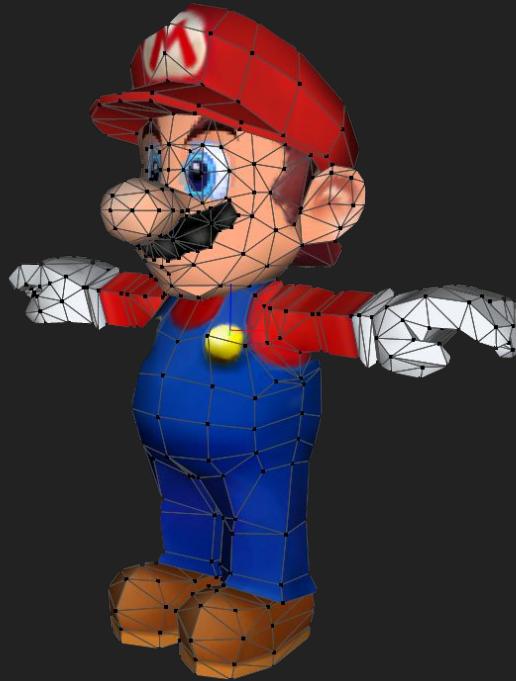


How did we get here?







Hardware
(Specialized)



Software
Rendering



Hardware
Rendering

Hardware (Specialized)



The 2600 did not use a frame buffer. Instead the video device used two 8-pixel bitmapped sprites, two 1-pixel "missile" sprites, a 1-pixel "ball", and a 40-pixel "playfield" that is drawn by writing a bit pattern for each line into a register just before the television scans that line. As each line is scanned, a game must identify the non-sprite objects that overlap the next line, assemble the appropriate bit patterns to draw for those objects, and write the pattern into the register.

Racing the Beam

The Atari Video Computer System is a book by Ian Bogost and Nick Montfort



The NES uses a custom-made Picture Processing Unit (PPU) developed by Ricoh. All variations of the PPU feature 2 kB of video RAM, 256 bytes of on-die "object attribute memory" (OAM) to store the positions, colors, and tile indices of up to 64 sprites on the screen, and 28 bytes of on-die palette RAM to allow selection of background and sprite colors. The console's 2 kB of onboard RAM may be used for tile maps and attributes on the NES board and 8 kB of tile pattern ROM or RAM may be included on a cartridge.

https://en.wikipedia.org/wiki/Nintendo_Entertainment_System



Mode 7



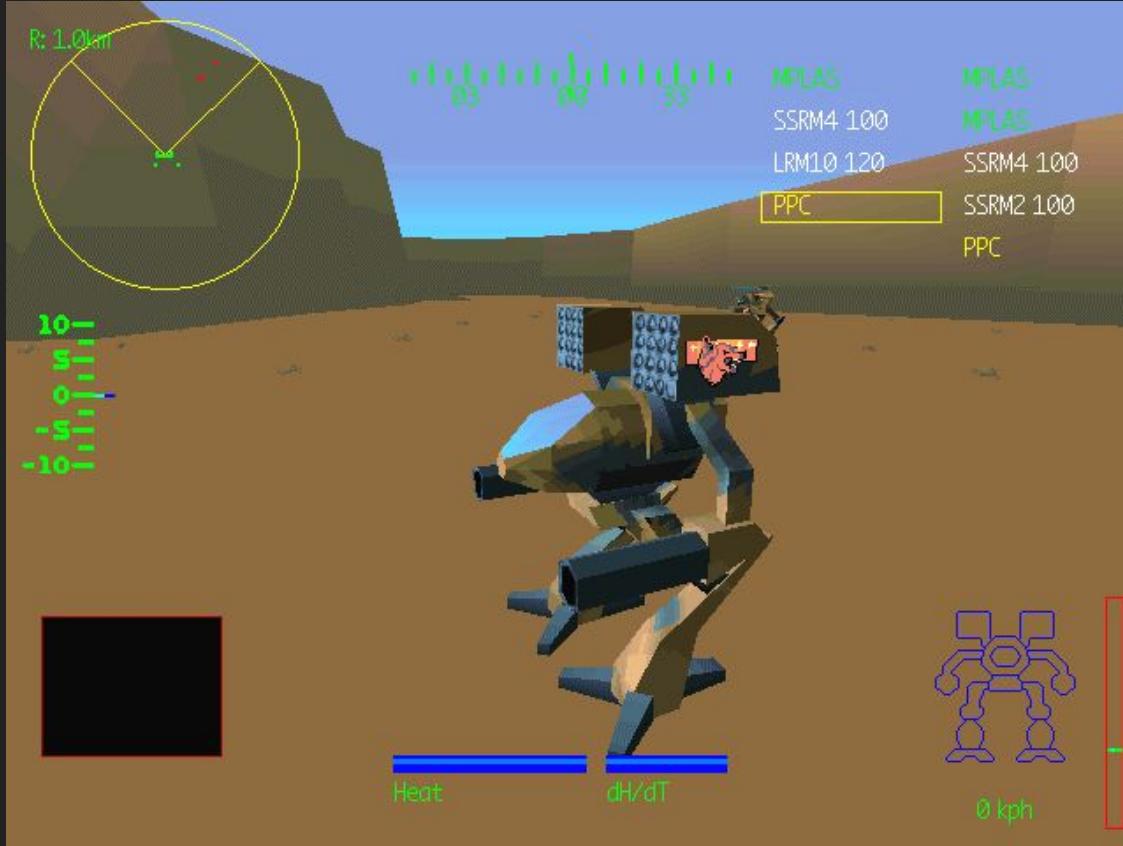
The Super FX Chip

<https://www.youtube.com/watch?v=Opzomu6mgYk&start=313&end=462>

Software Rendering



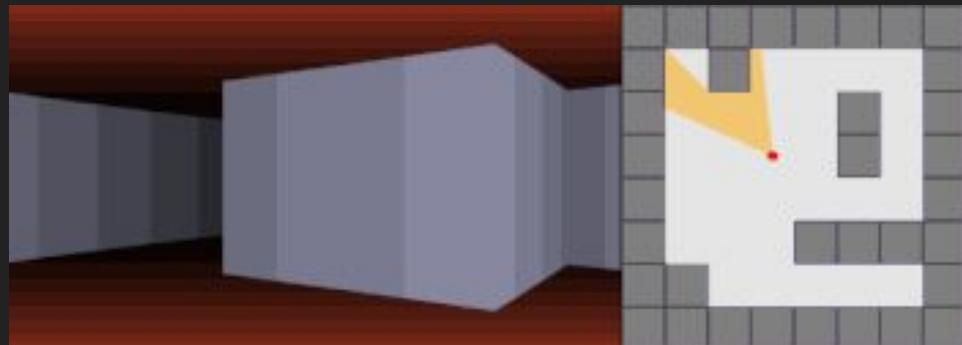
(The predecessor to Elite Dangerous)



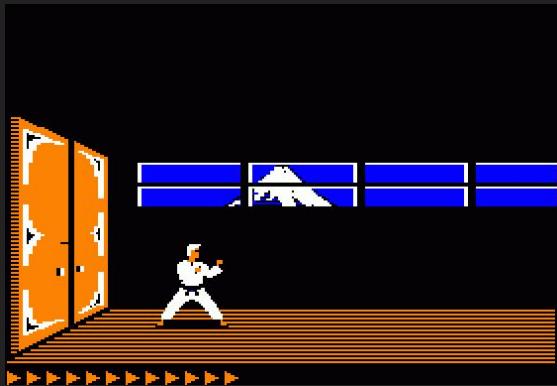
(MechWarrior 2: I lost my mind when I first played this game.)

Raycasting

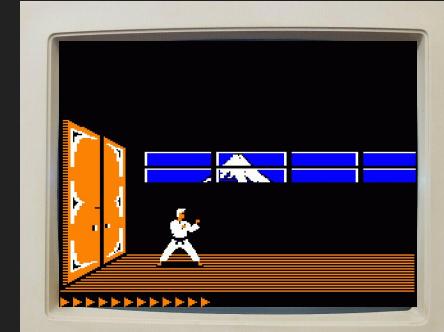
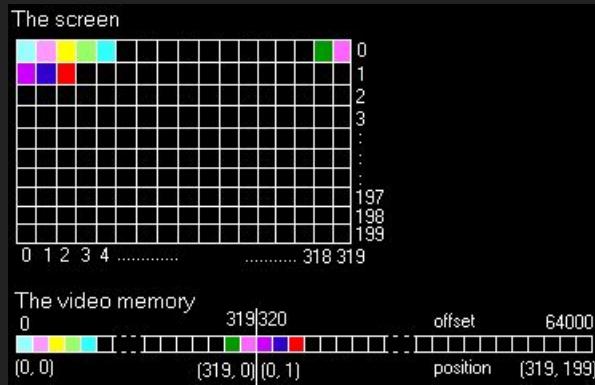
Wolfenstein 3D
Doom
Duke Nukem 3D
(and more)



Software Rendering



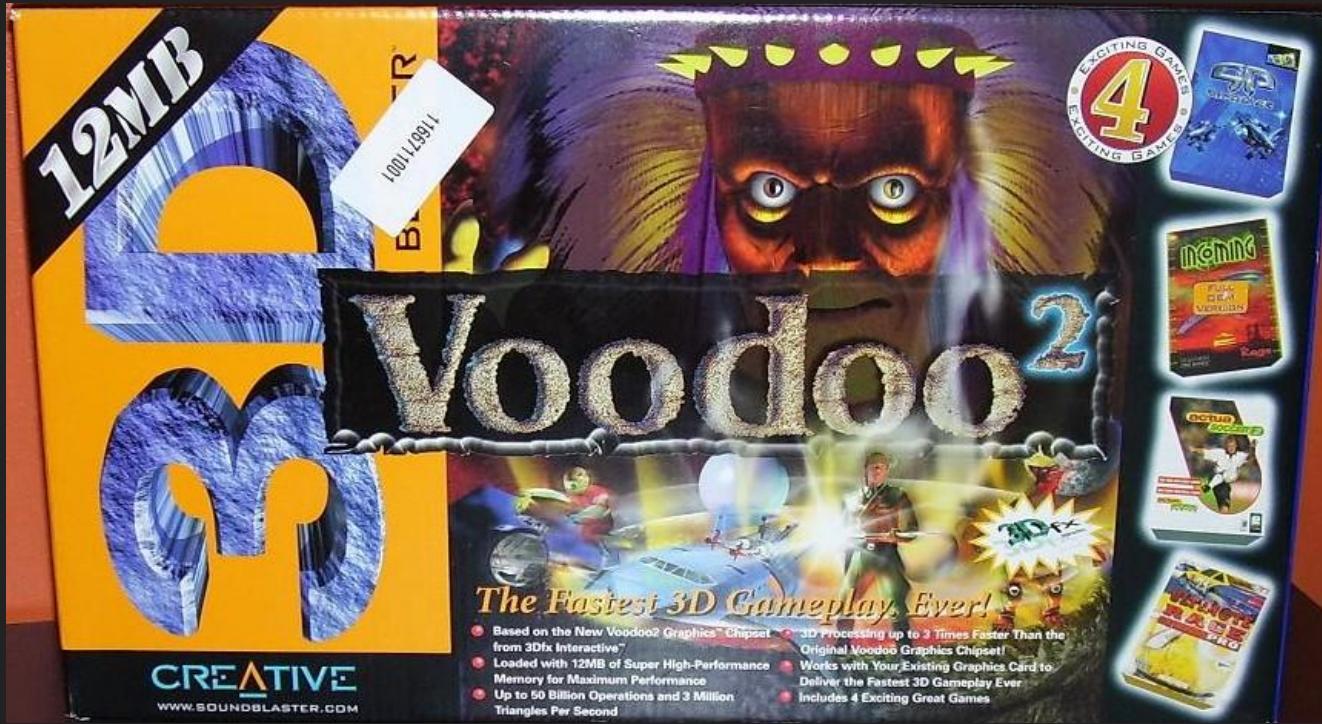
Scene built in RAM



Video Card

Hardware Rendering

Graphics Cards!

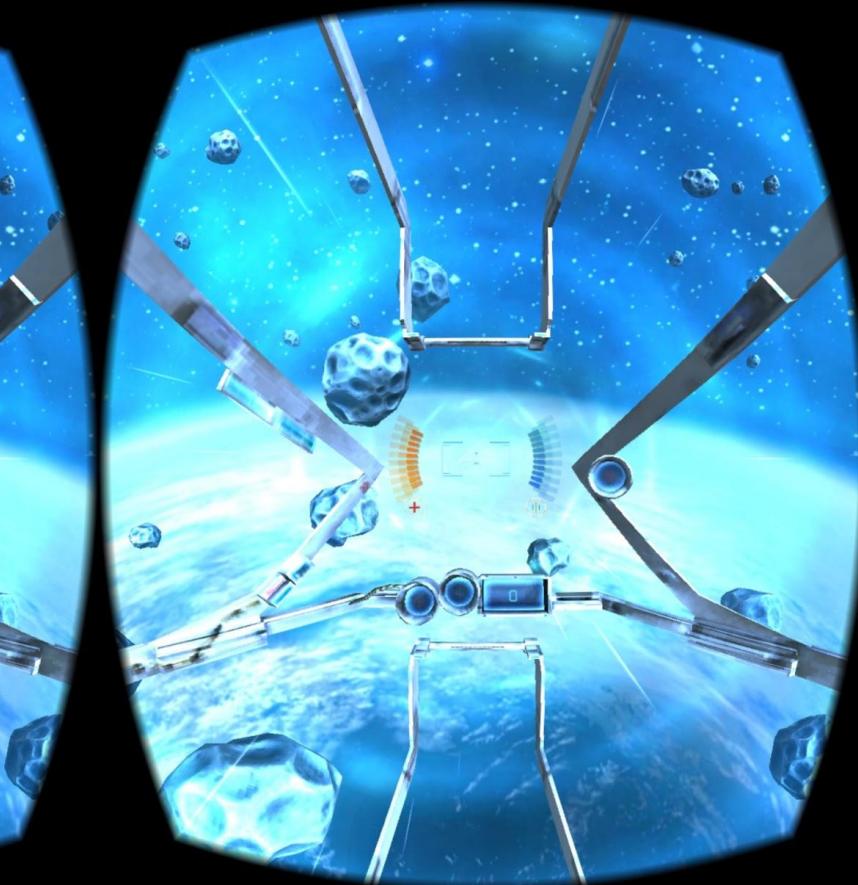










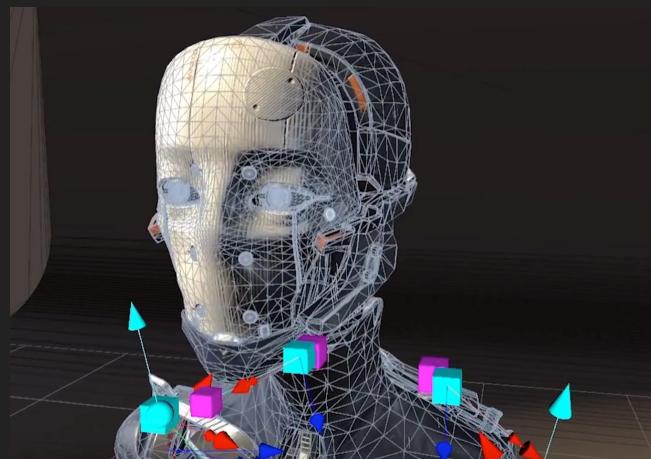
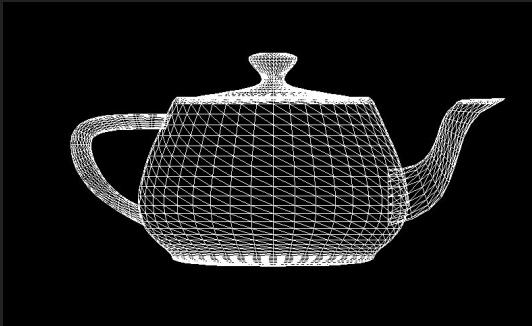
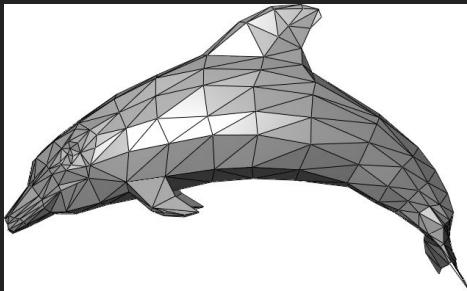


How does this all work?

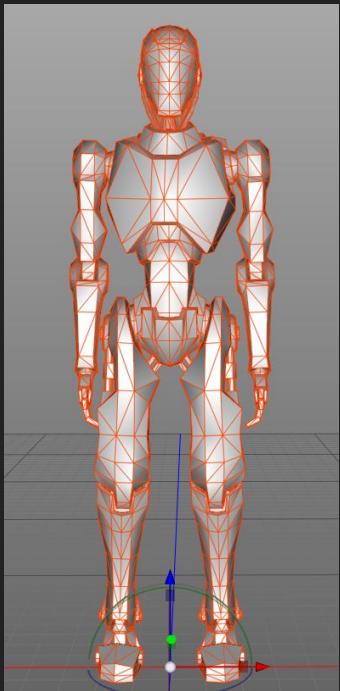
Triangles!

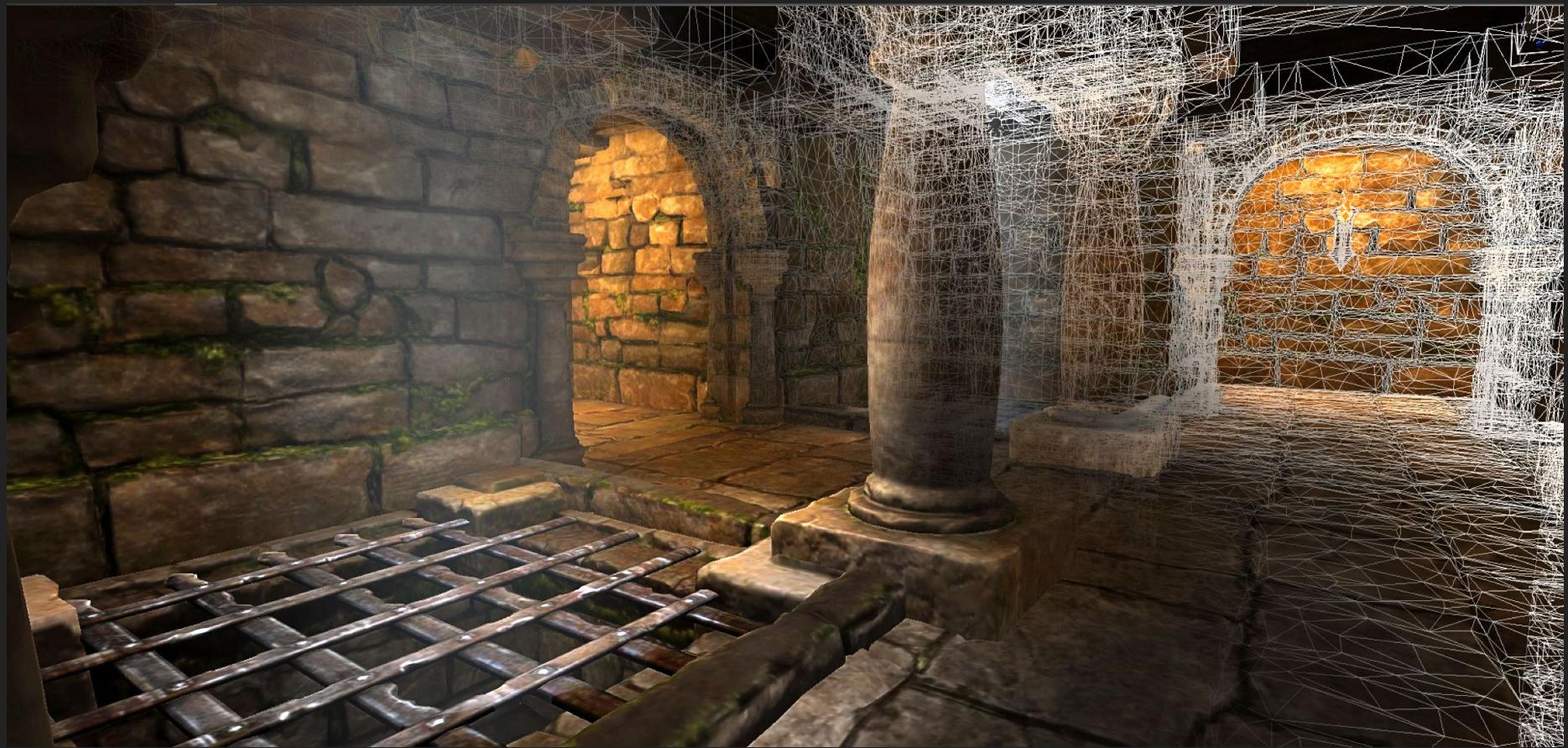


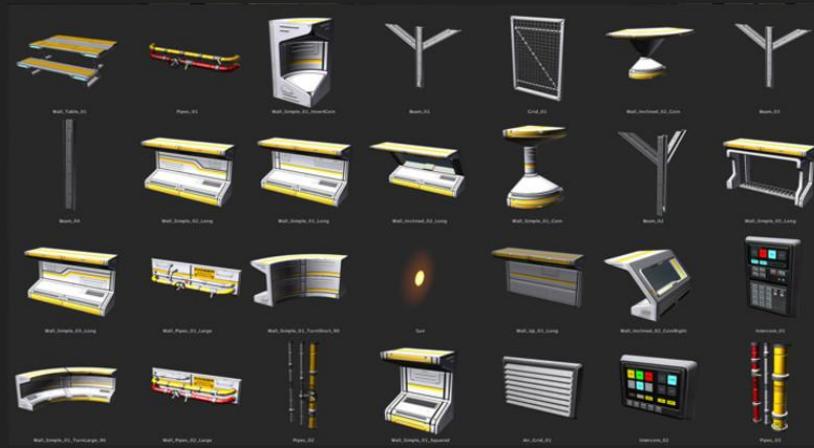
Lots of Triangles!



...and Textures







Hardware Rendering

Load assets to Video Card RAM

Run Game Logic
Tell Video Card what to draw

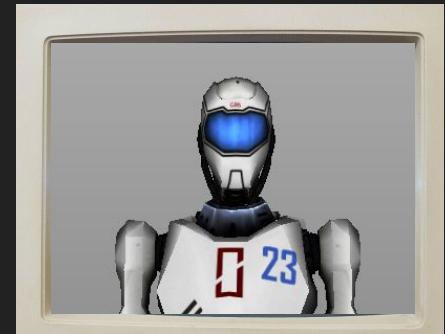
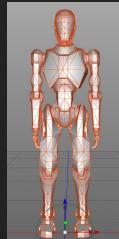
CPU + RAM



Video Card

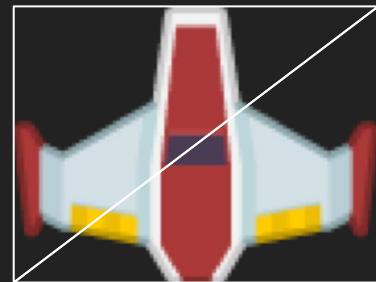


Display



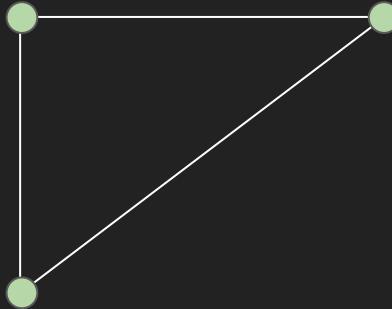
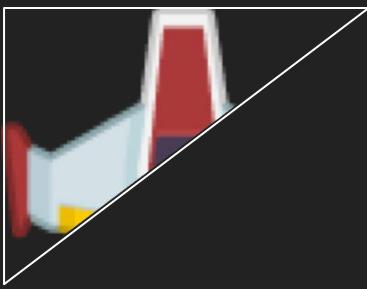
2D is really 3D
(in modern times)

2D Sprite Made of 2 Triangles



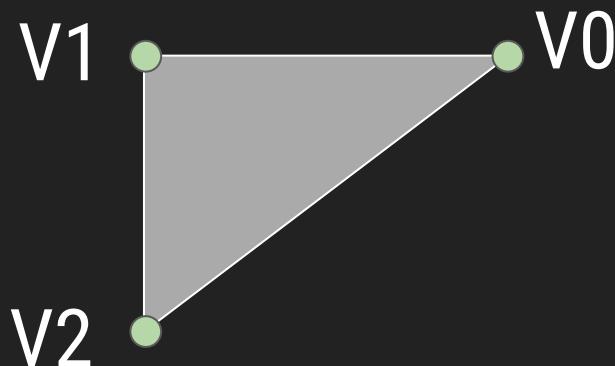
A Triangle Has 3 Vertices

(each point is a vertex)

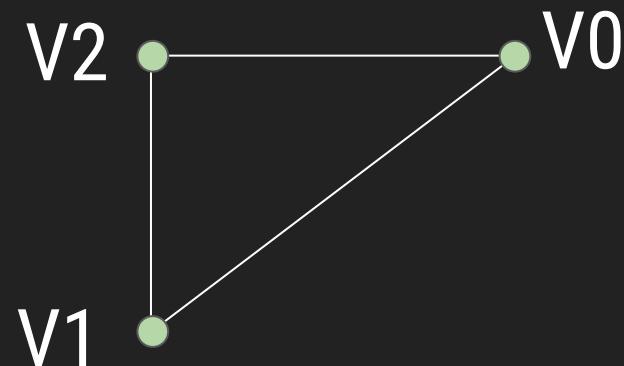


Triangles are 1 sided!

(the side is determined by order of the vertices)



Counter-Clockwise
(Front Facing)

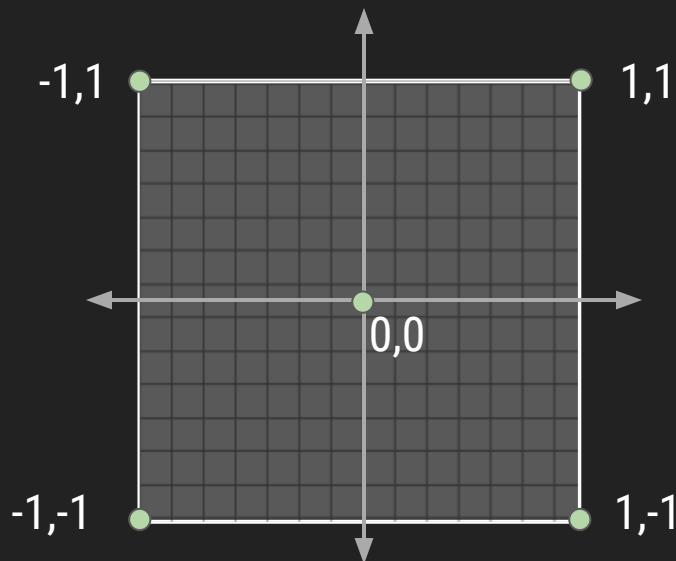


Clockwise
(Back Facing)

Texture Coordinates



Model Coordinates vs Texture Coordinates



Every Frame
Send Instructions on
What to Draw

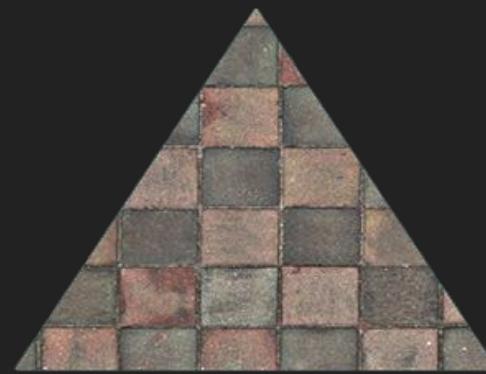
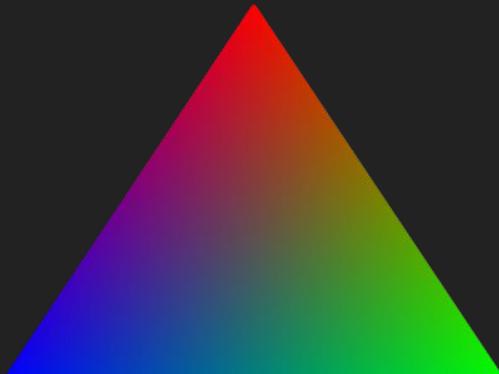
Shaders!

Vertex Shader

Translates vertices to screen positions.

Fragment Shader

For each pixel, determines what color to draw.
Interpolated by distance from vertices.
(might also grab a pixel/color from a texture)



Let's draw a triangle!