GPUs & You

AN ABRIDGED HISTORY AND INTRO TO GRAPHICS PROGRAMMING

03/2025

Bits and Bolts

HARRY MORRIS

AGENDA

1 ABOUT ME

2 THE HISTORY

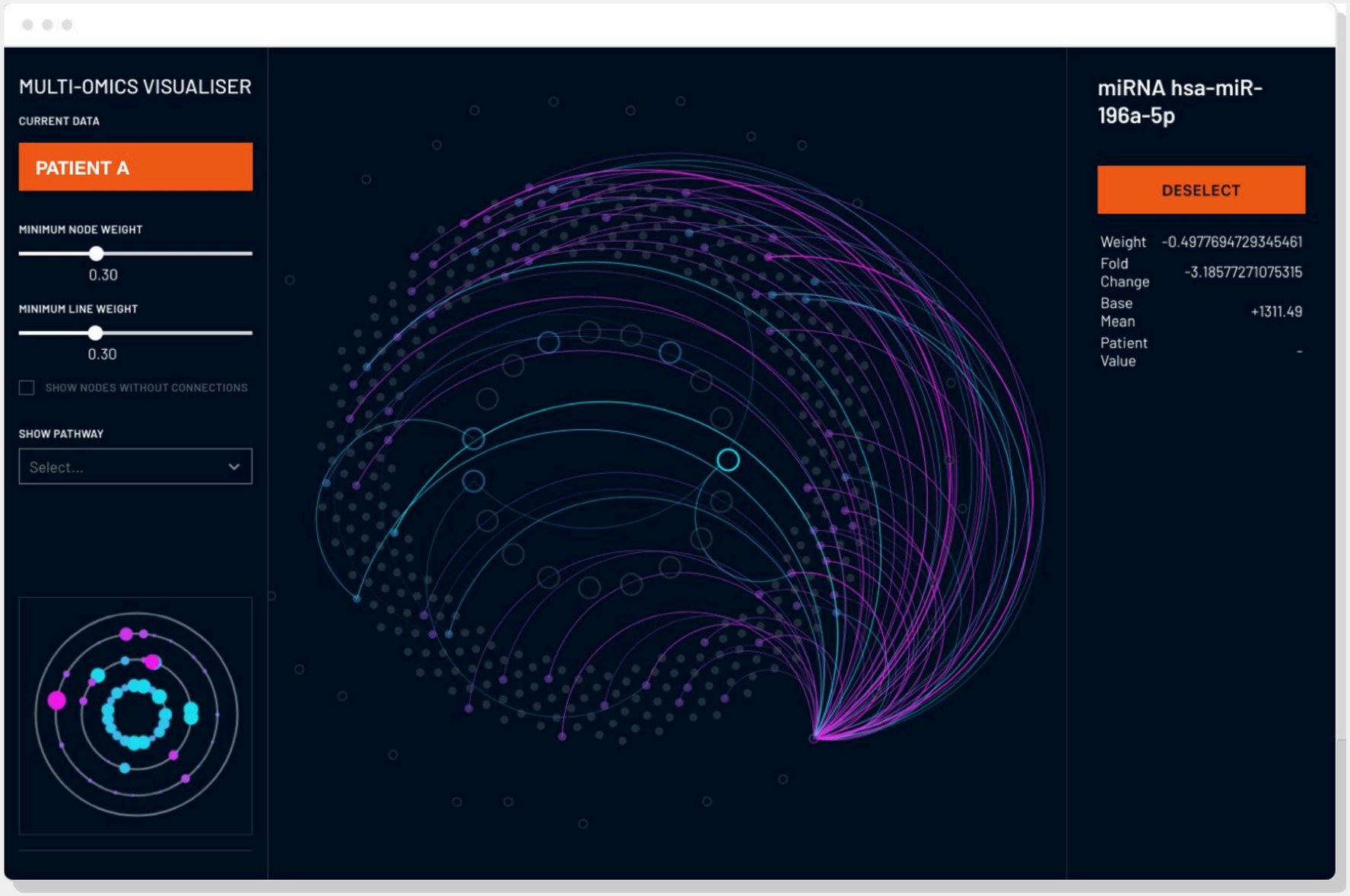
3 “LIVE” DEMO

HI, I’M HARRY

1

I always wanted
to be an *artist*.

Now I write software.



Pretty pictures with *purpose*

Drone photogrammetry in Propeller

“If you know the enemy and know yourself, you need not fear the result of a hundred battles.”

Sun Tzu, GPU expert

“To know your enemy, you must become your enemy”

Sun Tzu, responding to questions about AI replacing his role

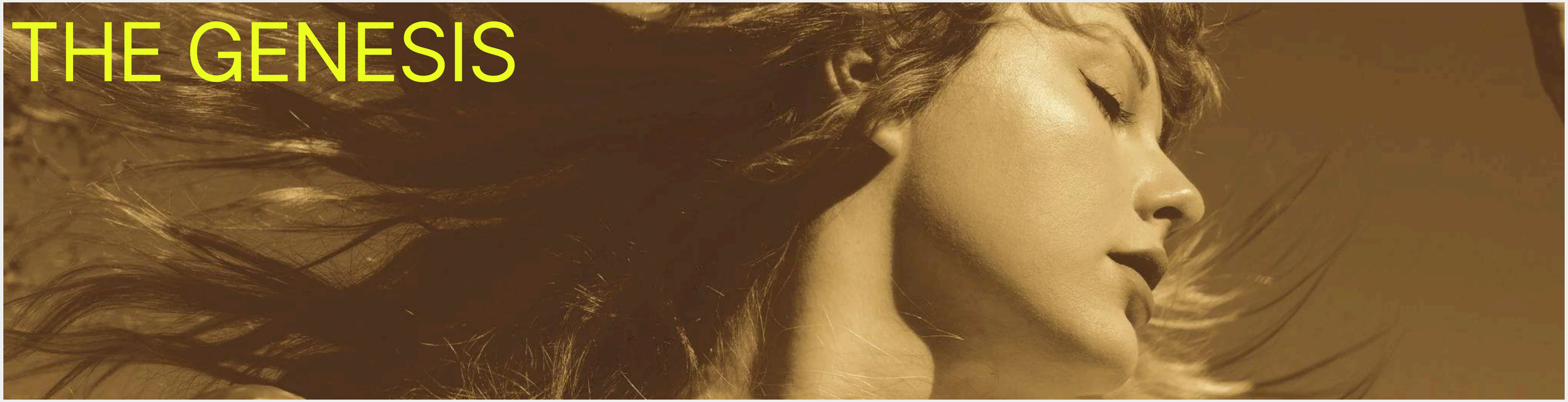
A BRIEF HISTORY OF GRAPHICS...

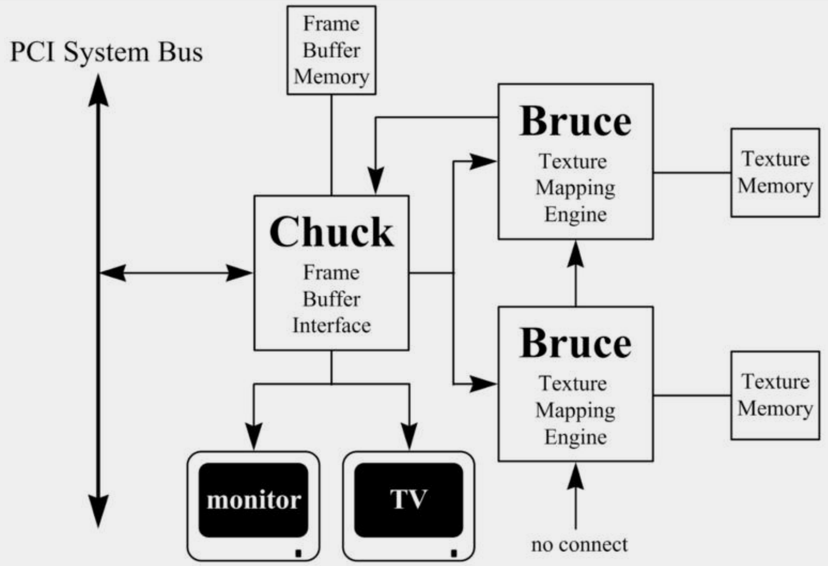
(If they were Taylor Swift's eras)

2

The ‘Fearless’ Era

90’s-Early 2000s

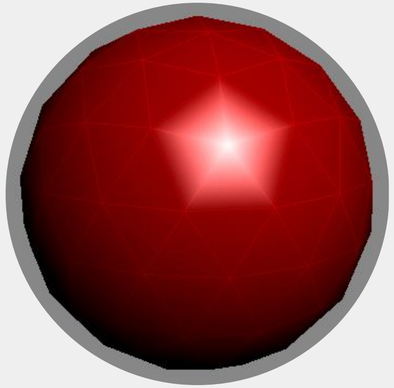


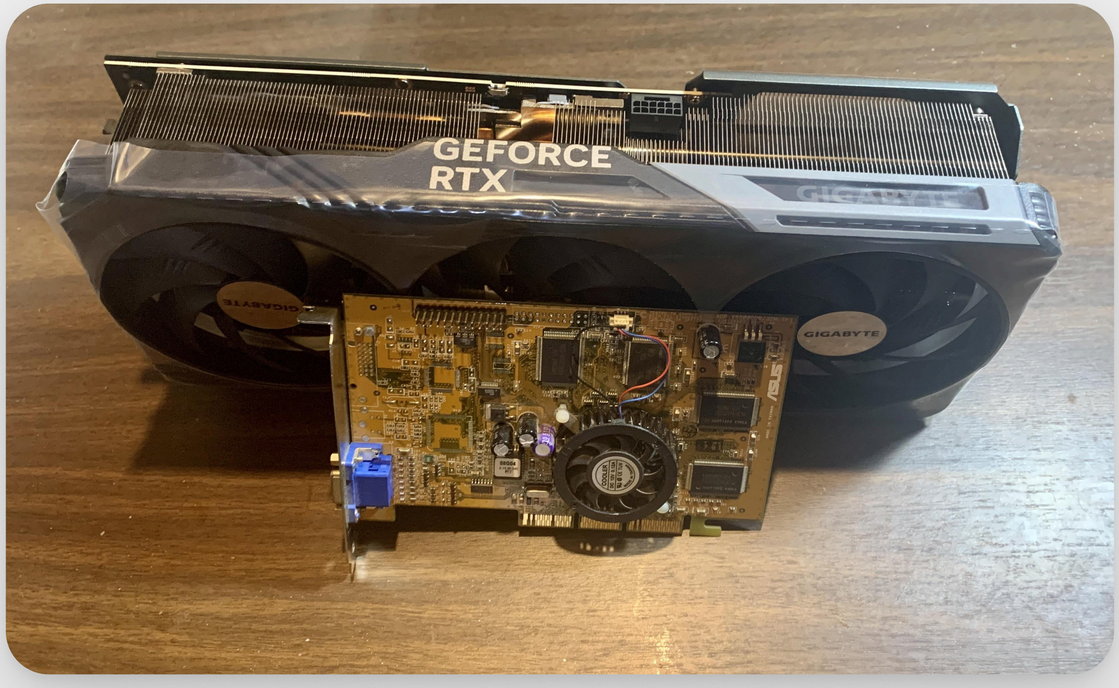


3Dfx Voodoo 2

“Graphics Accelerators”

- A framebuffer (800×600) with depth-tests and Gouraud shading
- Texturing units
- Not a complete solution to turn data into graphics



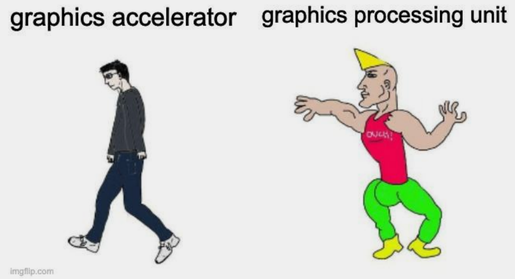


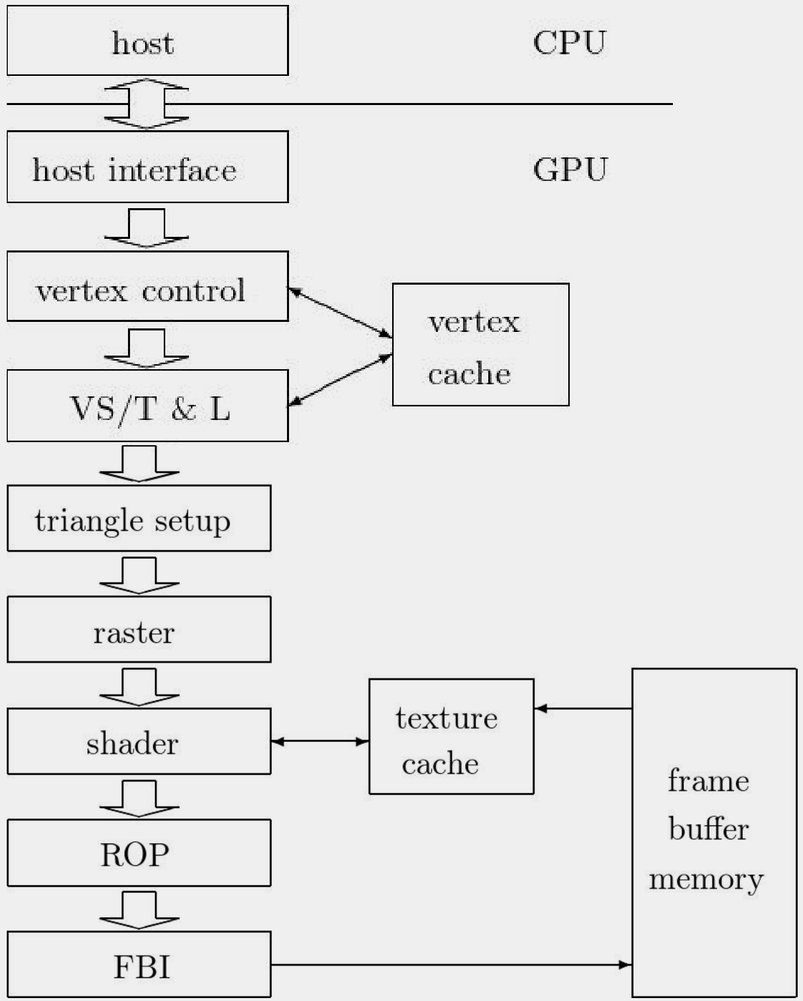
u/SaberHaven

Hardware *evolved*

Geforce256 coined the *“worlds first GPU”*

Hardware transform & lighting, instead of software, marking the transition to “Graphics Processing Unit”





Fixed-function pipelines

No control over the vertex transform & lighting

No control over the rasterization and shading of triangles

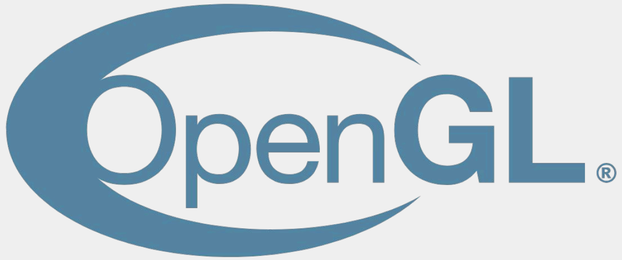
Some extensions for greater visual fidelity, but still no control

Immediate mode rendering

The ‘Red’ Era

2000’s

Programmable pipelines



OpenGL 2+

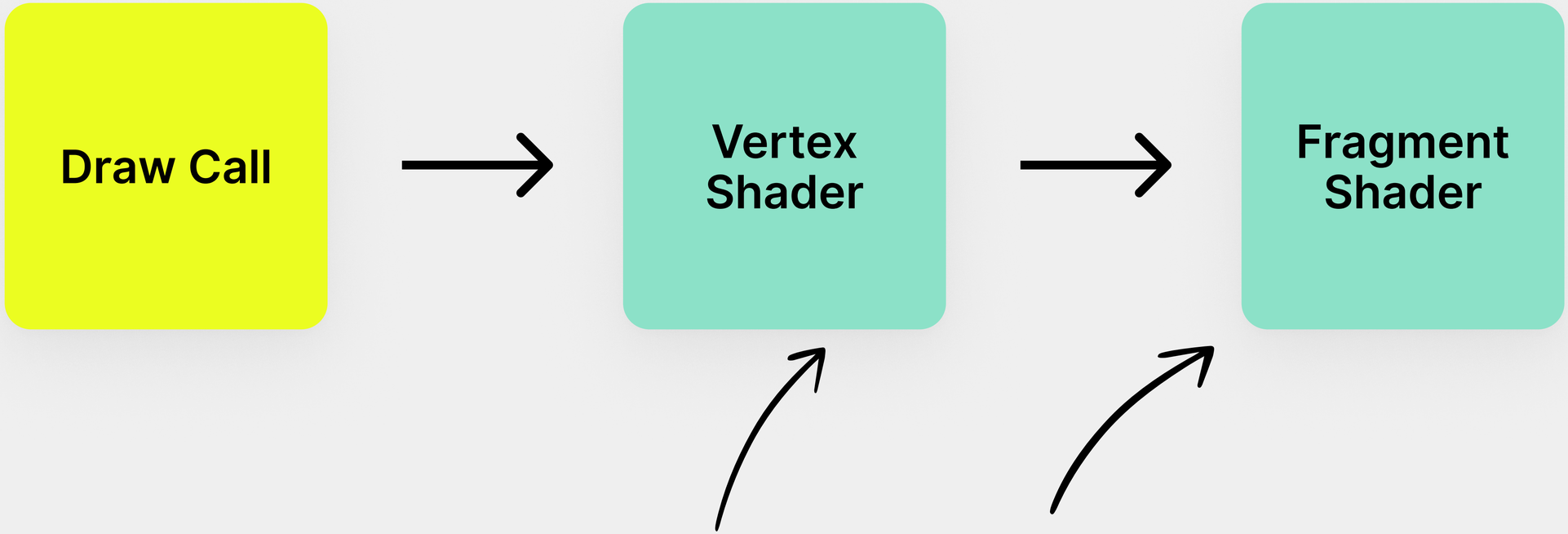
DirectX 8+

Shader model 1, 2, 3..

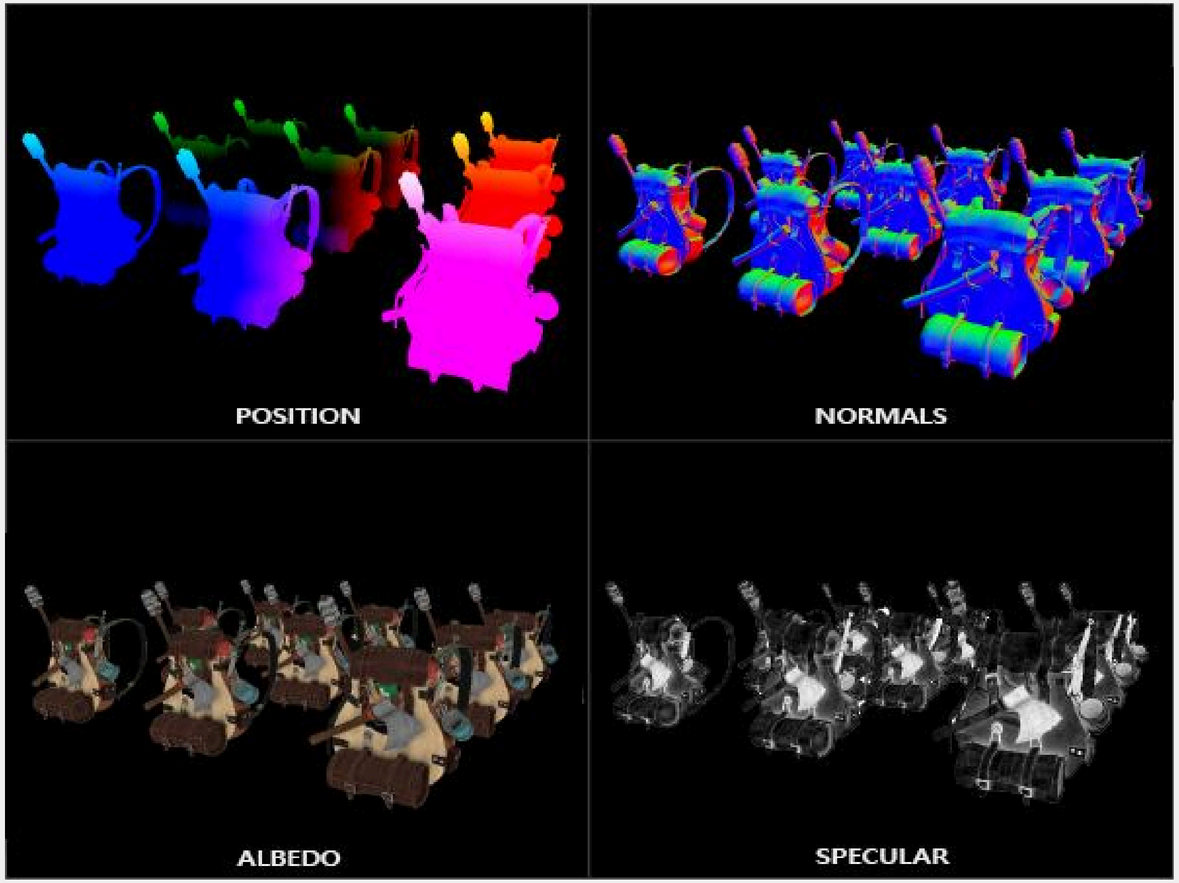
Let’s get ready to draw

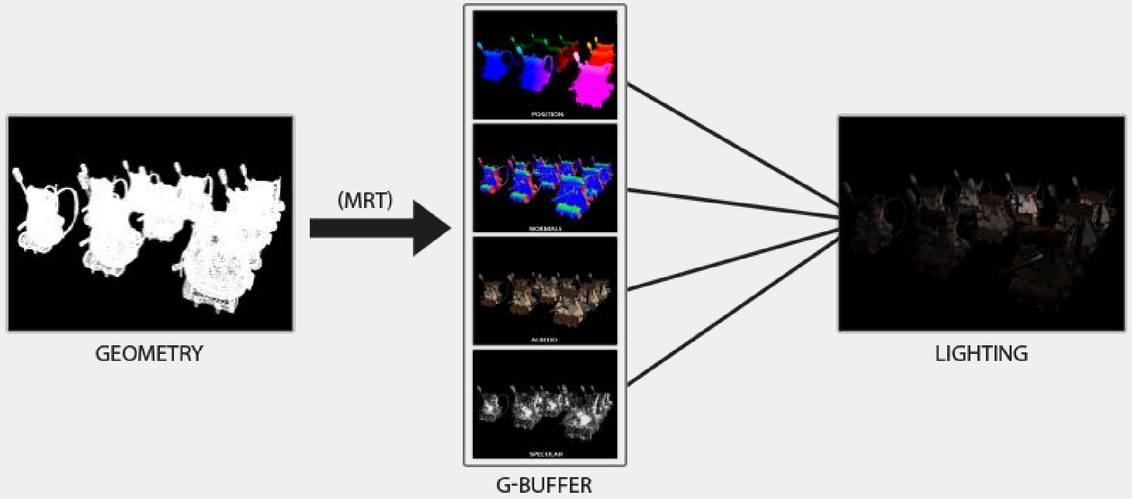
Now let’s draw...

Now let’s draw...



Deferred Rendering



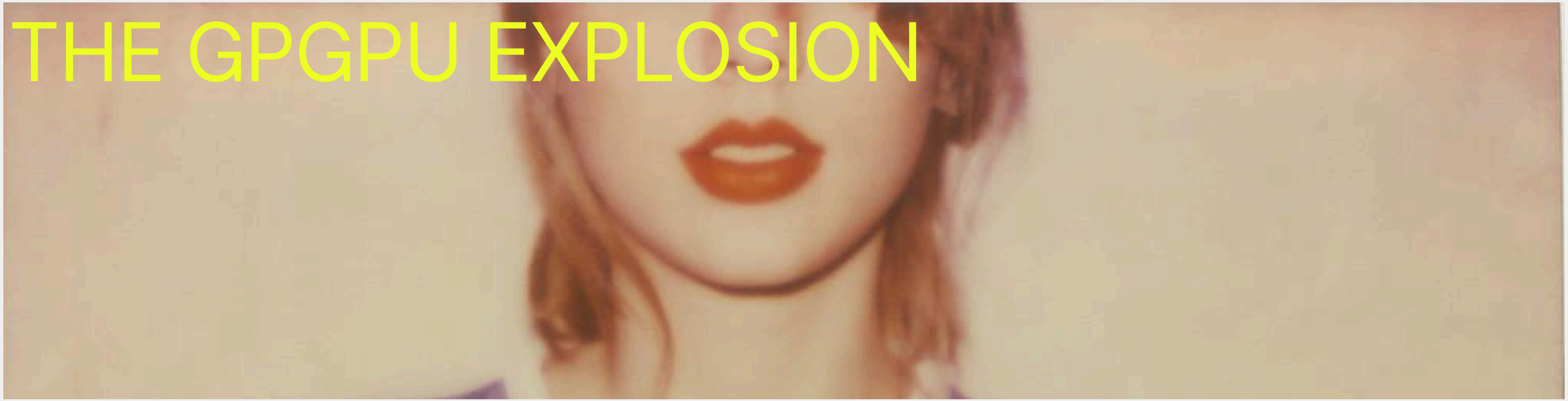


From https://learnopengl.com/Advanced-Lighting/Deferred-Shading

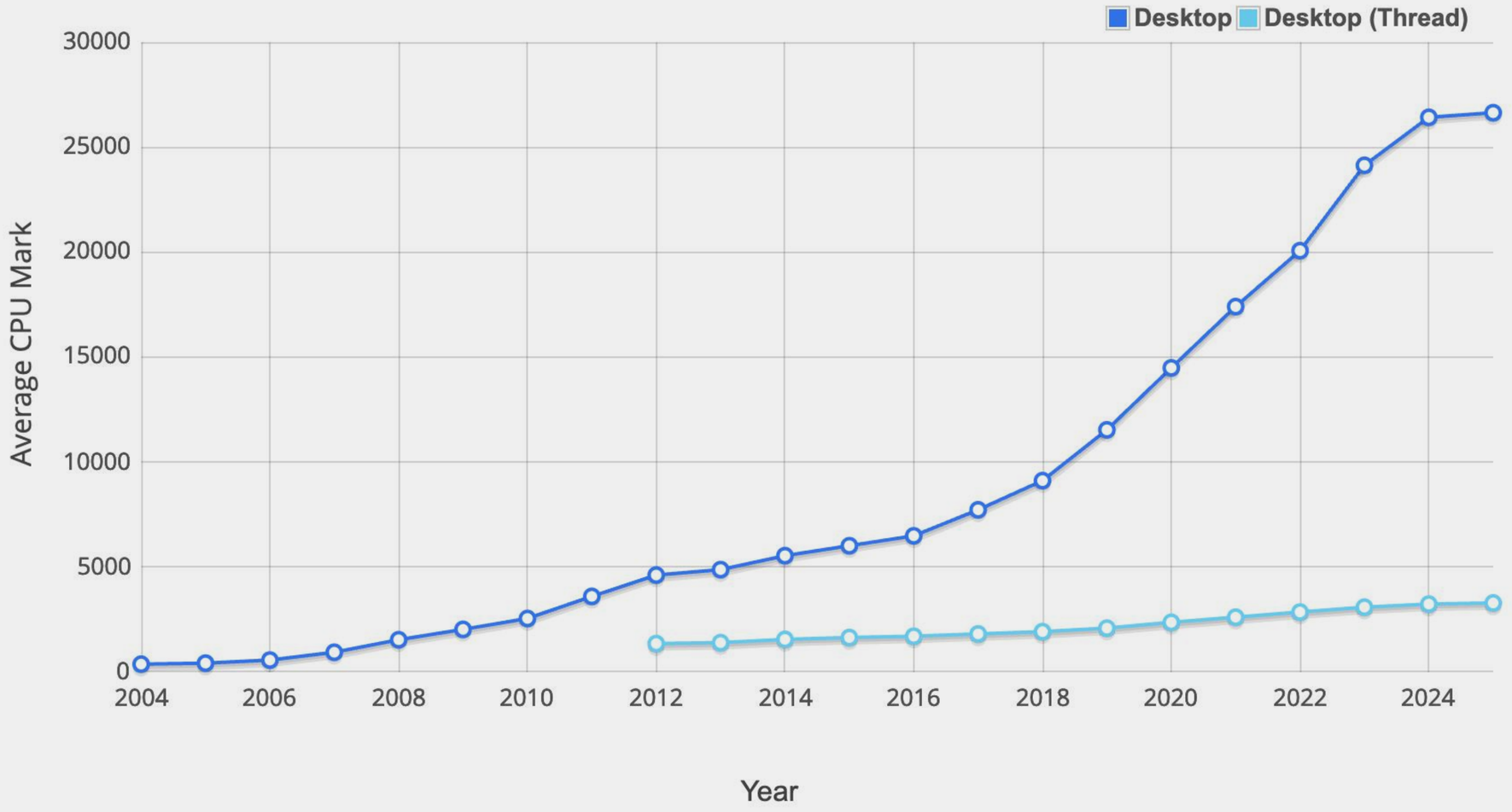

SCREEN SPACE
AMBIENT OCCLUSION

The ‘1989’ Era

Late 2000’s - Early 2010s



CPUs slow down



From https://www.cpubenchmark.net/year-on-year.html

CUDA

Compute Unified Device Architecture

PhysX (2008ish)

Bitcoin (2009ish)

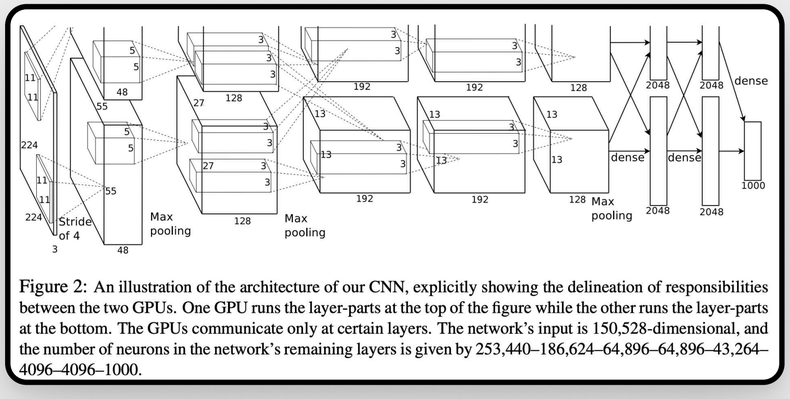
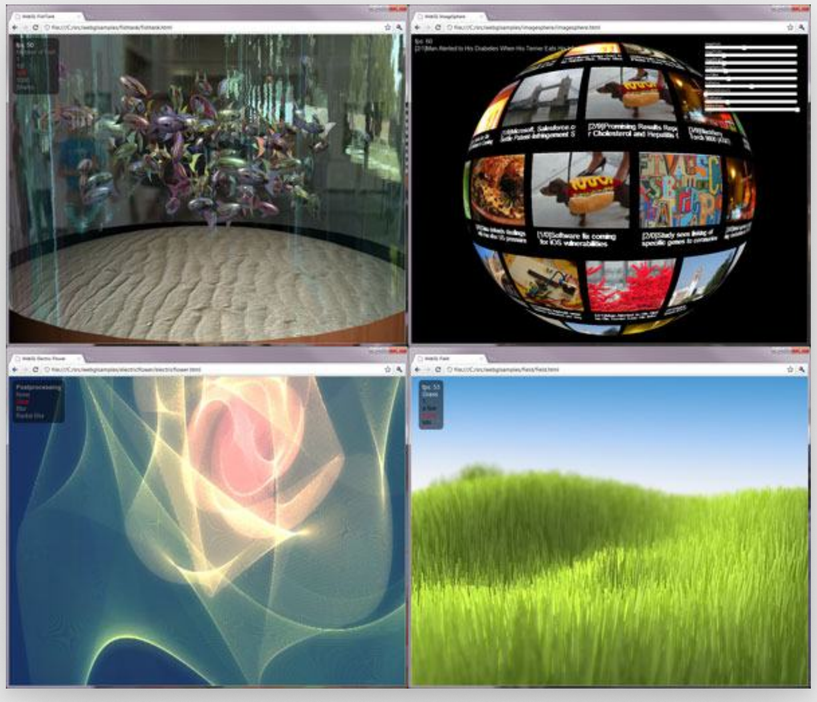


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

AlexNet (2011)



From https://github.com/WebGLSamples/WebGLSamples.github.io

WebGL

2011-ish

Based on OpenGL ES

mandelbulb_ by EvilRyu

The ‘Lover’ Era

Mid 2010s - Early 2020s

Shaders got pretty serious here

...and a lot more sensible here

Shader Model 1 + 2

Programmable vertex and pixel shaders. **DirectX 8**
Direct X 9 later adding flow control and texture access

2000-2003

Shader Model 3

Dynamic flow control, WAY more instructions, instancing and multiple render targets
Direct X 9c

2004-2005

Shader Model 4 + 5

Unified shader architecture, compute shaders
Direct X 10 + 11

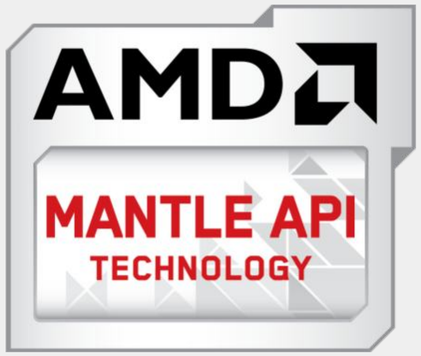
2006-2015

Shader Model 6

Ray tracing, AI stuff, mesh shaders, multithreaded shaders
Direct X 12

2015+

Get low (level)



Mantle (2013)

Metal (2014)

DirectX 12 (2015)

Vulkan (~2015)

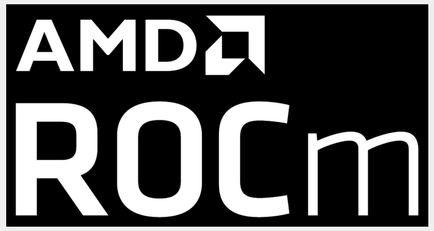
WebGPU (2023???)

xkcd.com #927

Because there *weren't enough ways* to program a GPU yet

OpenCL (2009)

SYCL (2014)



ROCm (2016)

oneAPI (2020)

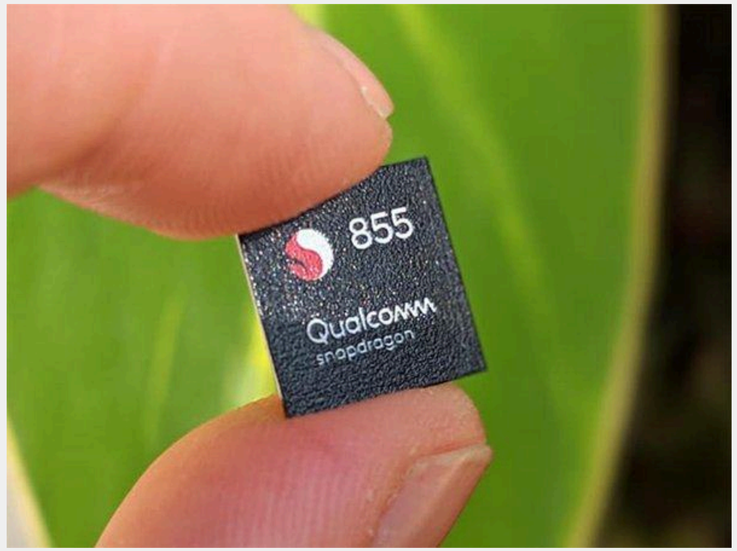
The ‘Tortured Poets’ Era

2020s - ???

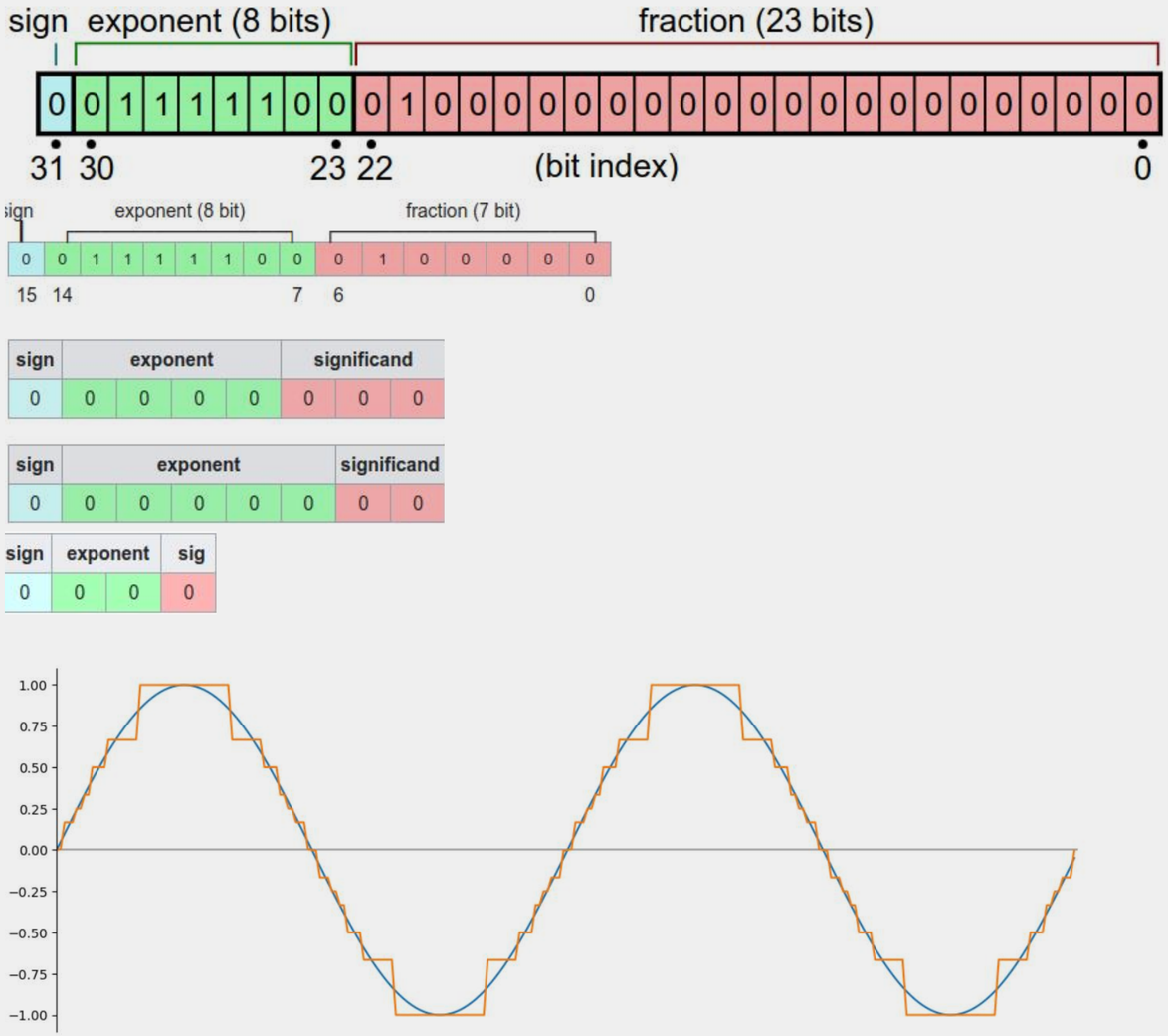
Dedicated AI processing + edge AI

Volta (2017) introduced Tensor Cores optimised for matrix multiplication and mixed precision

Neural Engine (2017)



Tensor Accelerators (2019)



From Towards Data Science

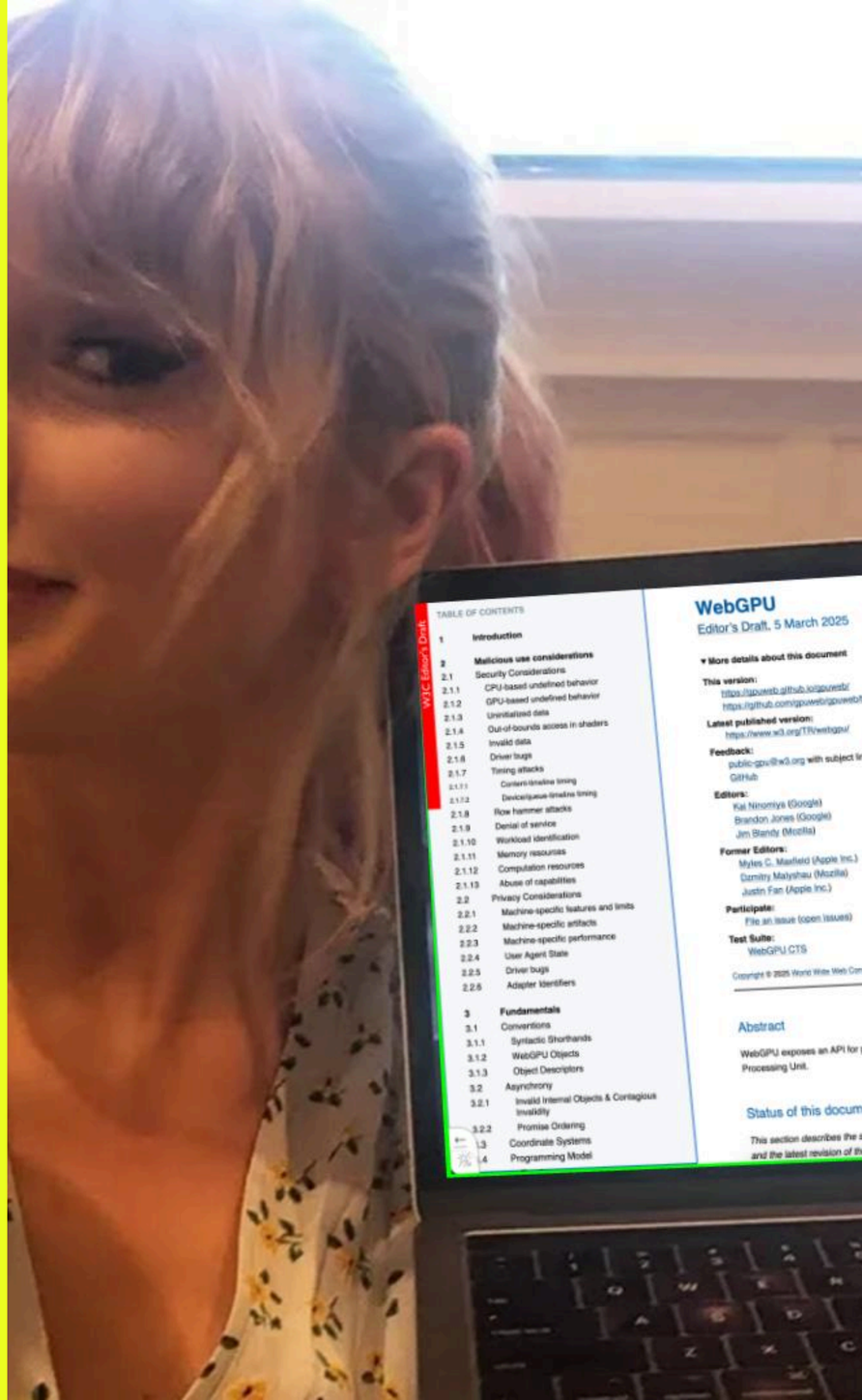
Lower-precision

“The DeepSeek R1 FP4 scored 99.8% of what the 8-bit version scored on the MMLU benchmark. That’s almost no difference in how smart the AI is, but a huge difference in how fast and cheap it runs.”

THE PART WHERE I EMBARRASS MYSELF

Lets look at some code

3



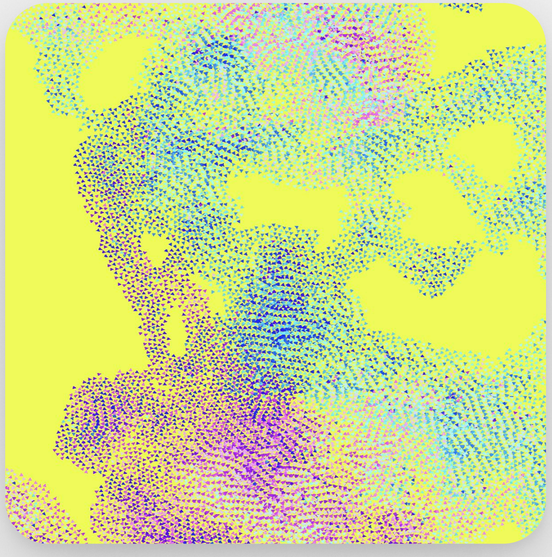
WHAT ARE WE COOKING UP?

A colourful triangle.

Let’s draw a triangle - the hello world of graphics.

Points & instancing.

GPUs are really good at doing a lot of things at the same time, if you talk to it in *just* the right way.



Boids.

Boids is a classic simulation demonstrating emergent behaviour. It’s also a highly parallelizable problem and looks sweet.

ty :>

You know what? I'm just gonna say it.
I'm not even that fond of Taylor Swift