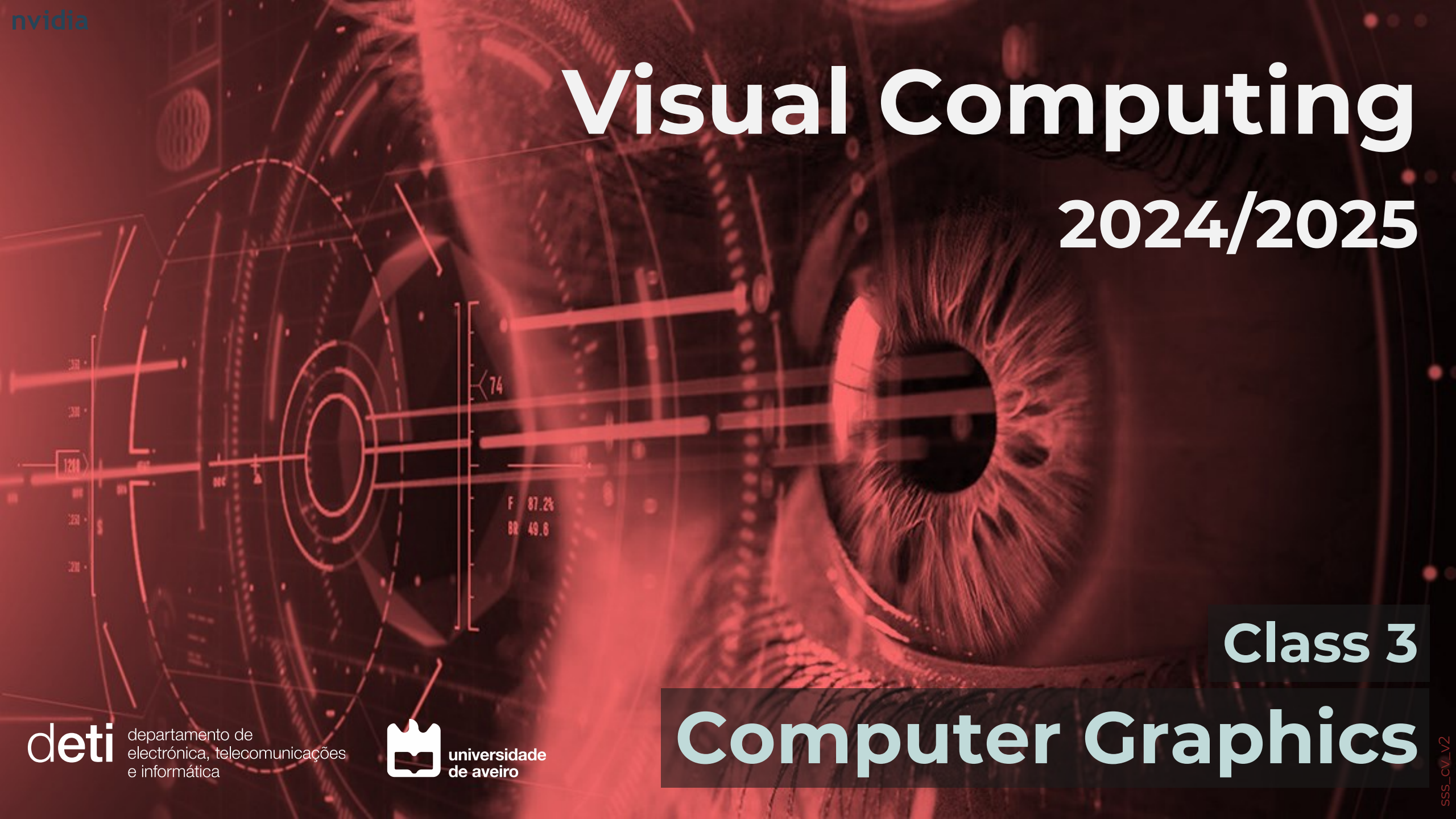
nvidia

Visual Computing

2024/2025



Class 3

Computer Graphics

deti departamento de electrónica, telecomunicações e informática

universidade de aveiro

Today’s Agenda

Let’s get into CG!

- CG Evolution and Trends
- CG Libraries
- OpenGL
- The Graphics Pipeline
- Hands On

https://blogs.commons.georgetown.edu/cctp-820-fall2016/2016/11/02/520/

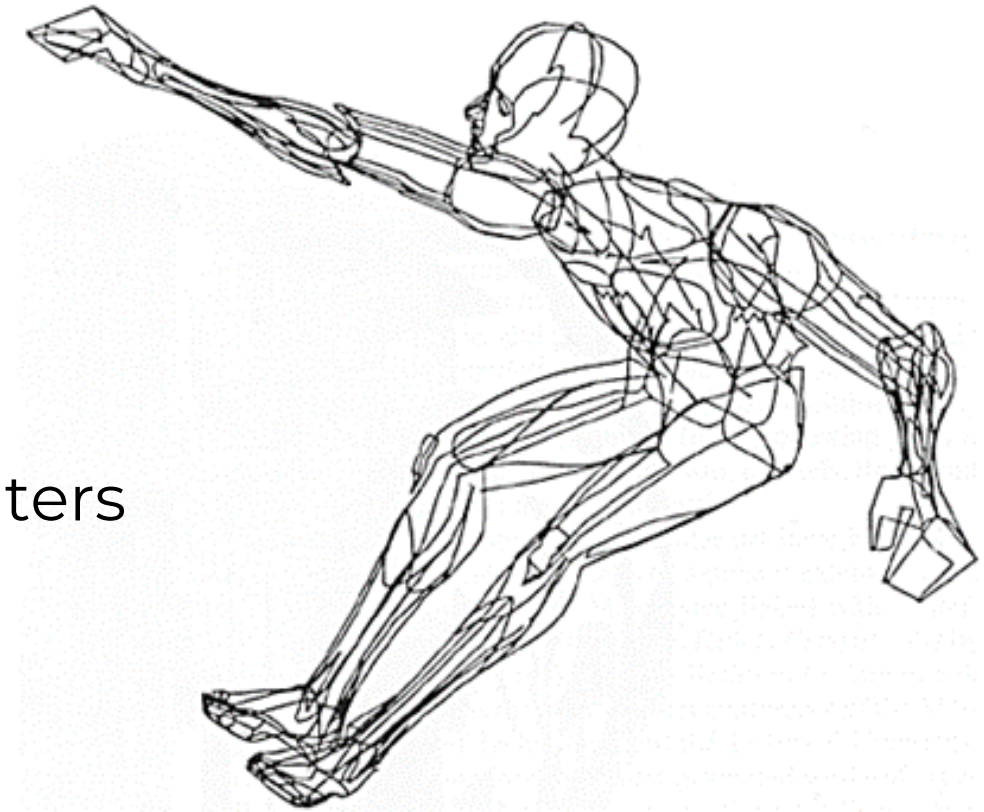
Computer Graphics: 1950 – 1960

Earliest days of computing

- Pen plotters
- Simple calligraphic displays

Issues

- Cost of display refresh
- Slow, unreliable, expensive computers

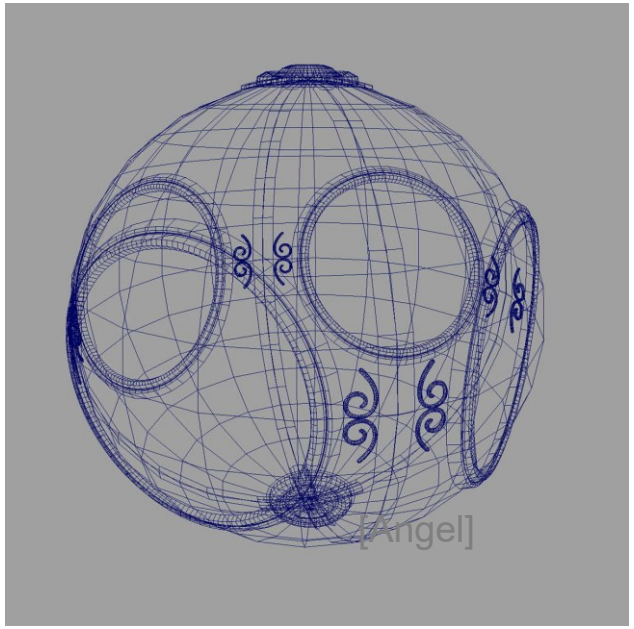


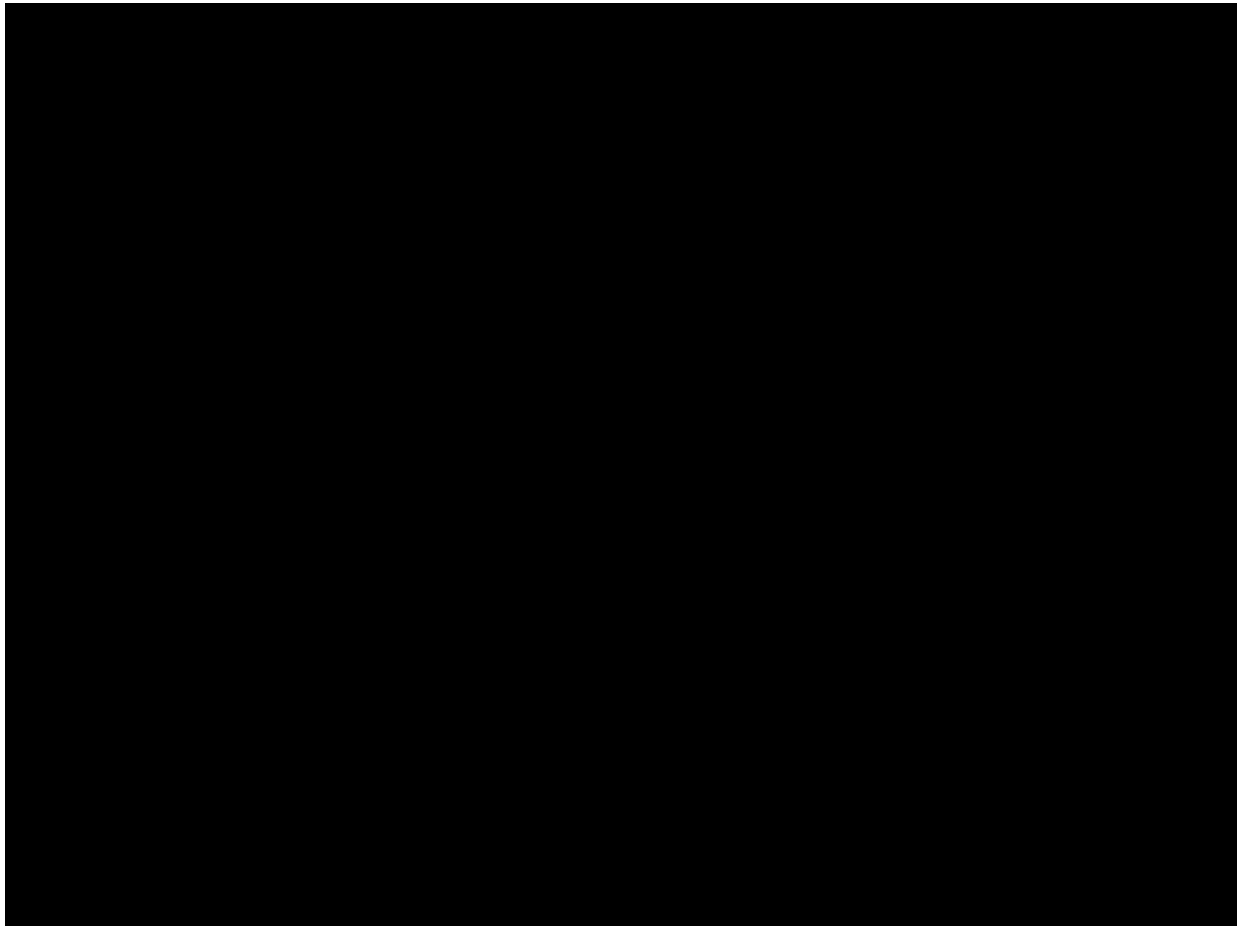
[van Dam]

Computer Graphics: 1960 – 1970

Wireframe graphics

- Draw only lines !



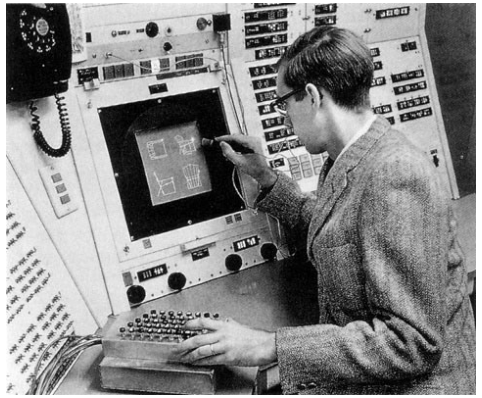


https://youtu.be/Uw2X8J53k2E?si=lAPds-h5lwt9bsC6

Computer Graphics: 1960 – 1970

Ivan Sutherland’s Sketchpad

- PhD thesis at MIT (1963)
- Man-machine interaction
- **Processing loop**
  - Display something
  - Wait for user input
  - Generate new display



[http://history-computer.com]

https://youtu.be/J6UAYZxFwLc?si=ihLOTnVFcMIINCwE

Computer Graphics: 1970 – 1980

First graphics standards

Workstations and PCs

WIMP GUI + WYSI**A**WYG

- Desktop metaphor
- Selection and direct manipulation

Raster graphics

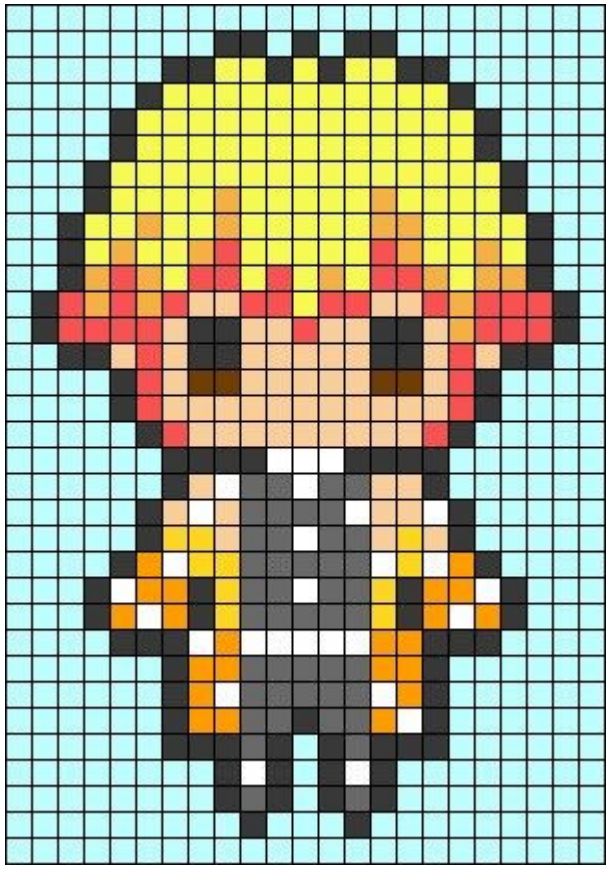
- Allows drawing polygons

Raster graphics

Image produced as an array (the **raster**) of picture elements (**pixels**) in the **frame buffer**

Raster Graphics

Building graphics
as arrays of pixels

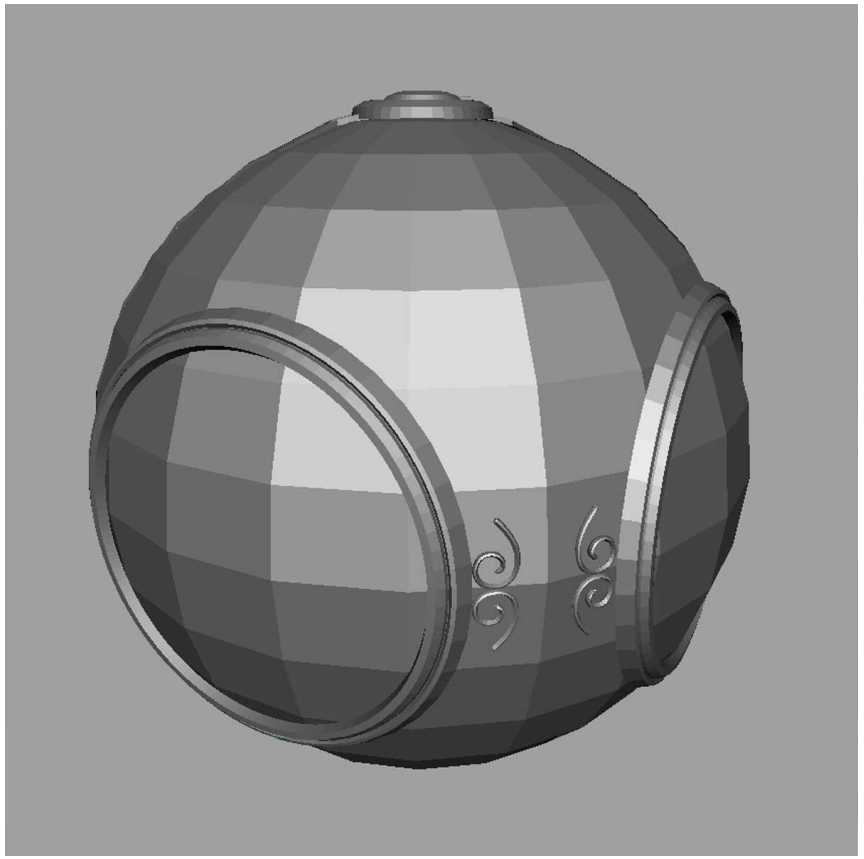


Raster graphics

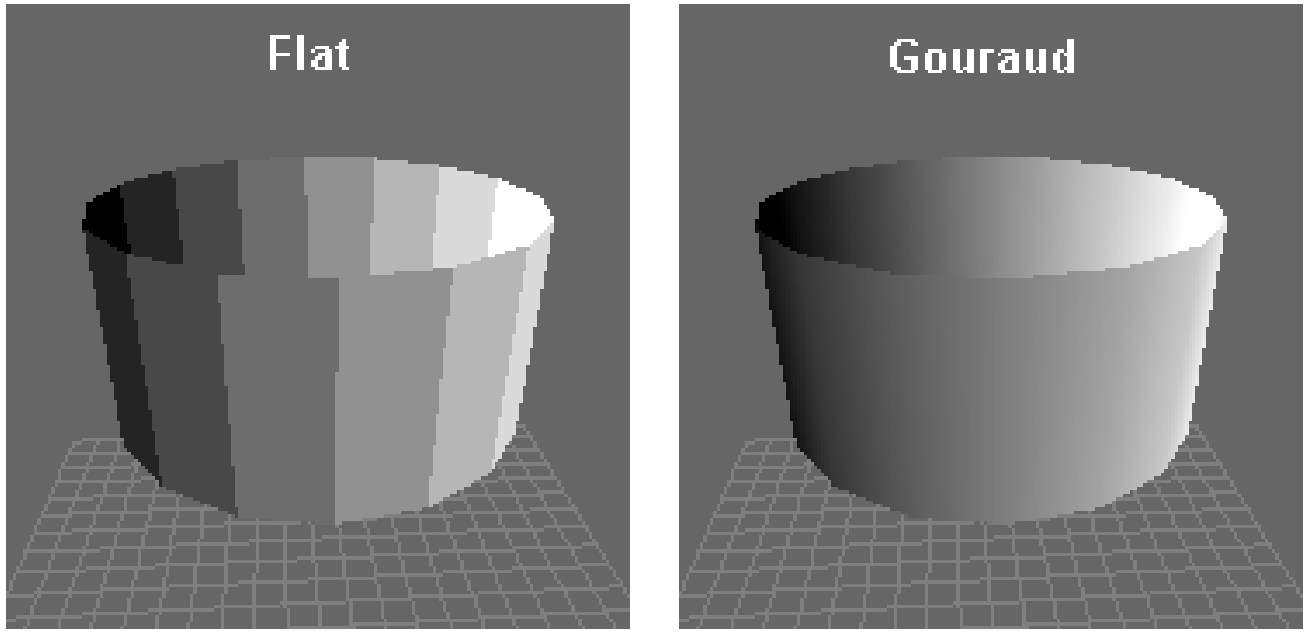
Drawing **polygons**

Illumination models

Shading methods

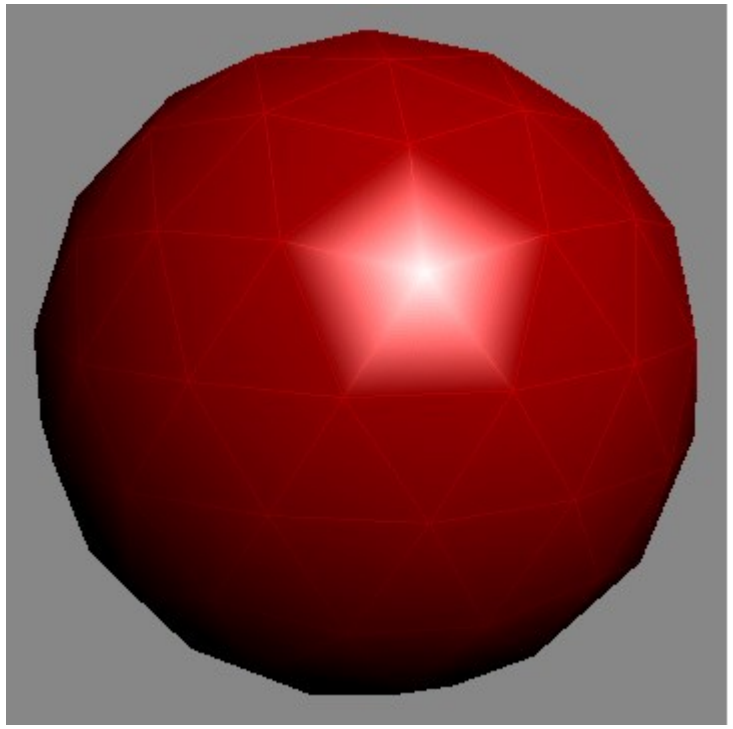


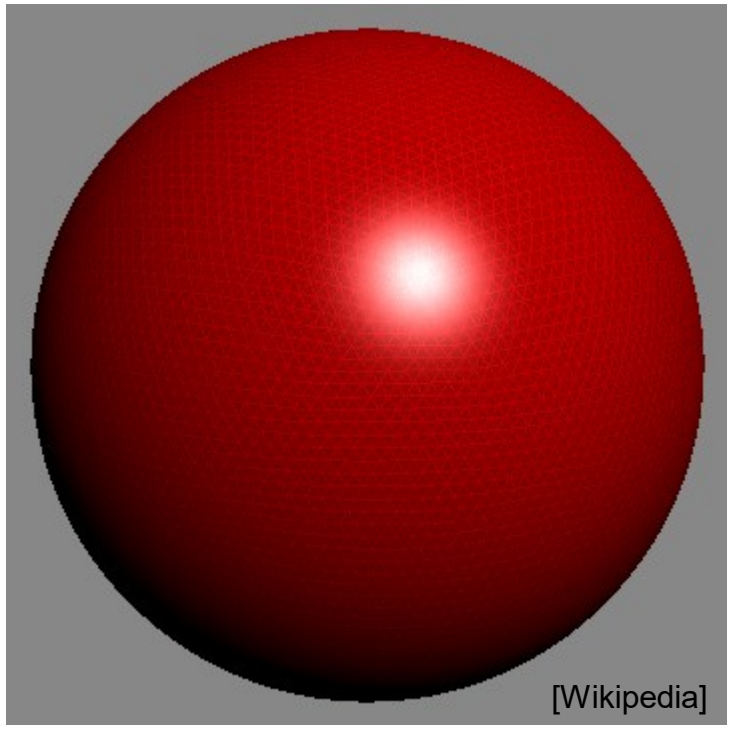
Gouraud shading – 1971



[Wikipedia]

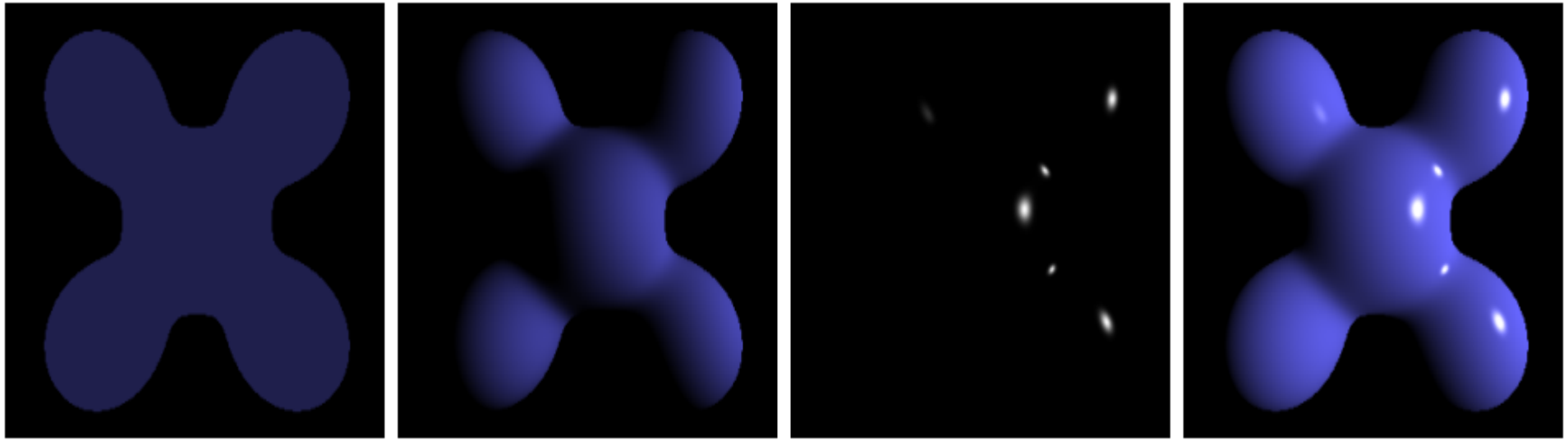
Gouraud shading



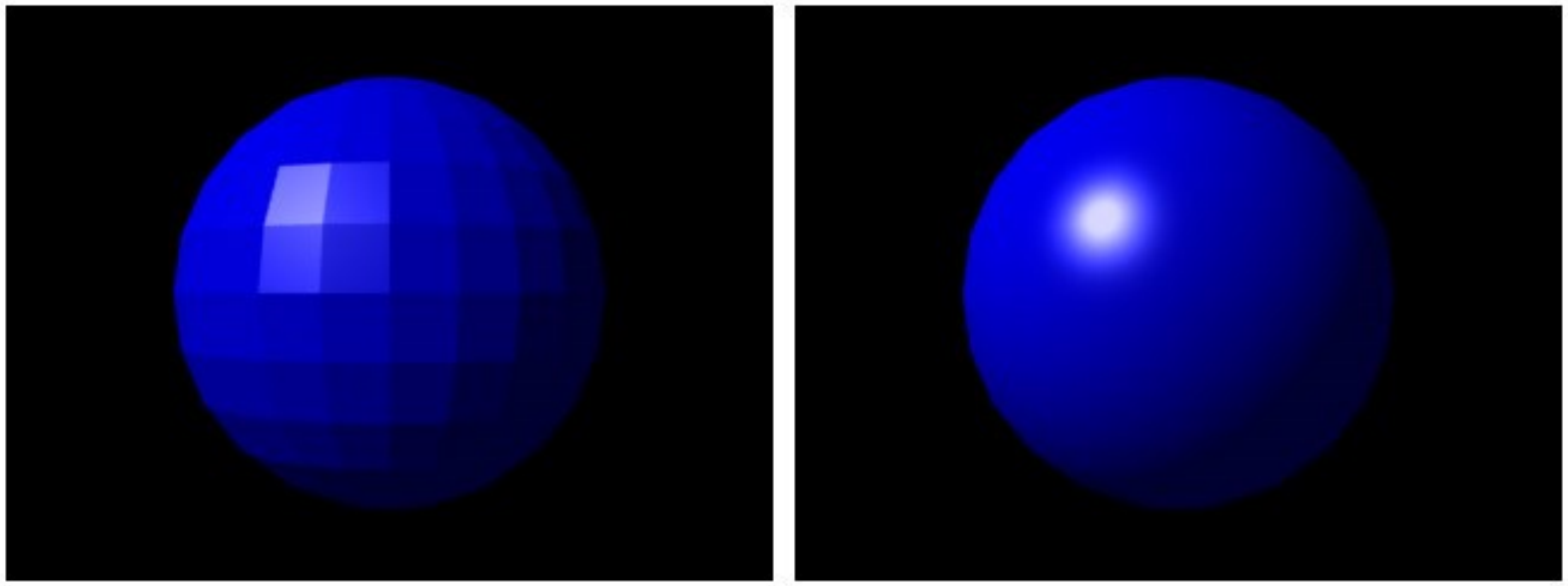


[Wikipedia]

Phong reflection model – 1973



Phong shading – 1973



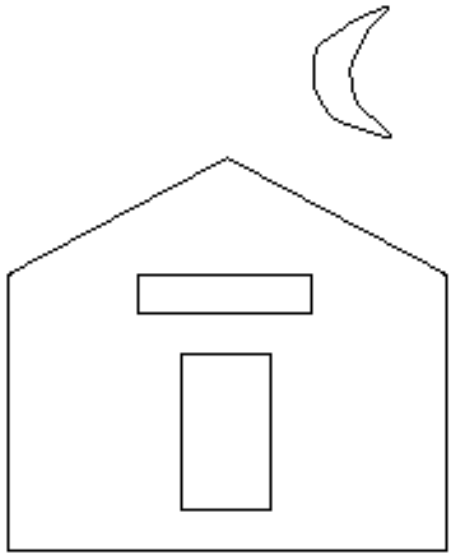
FLAT SHADING

PHONG SHADING

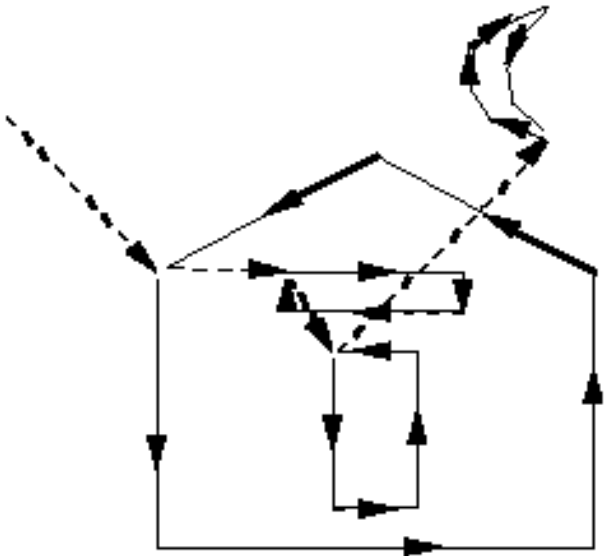
Vector graphics vs Raster graphics

Vector graphics is driven by display commands

- move(x,y); line(x,y); …
- Survives as **SVG** –
  Scalable Vector Graphics



Ideal Drawing



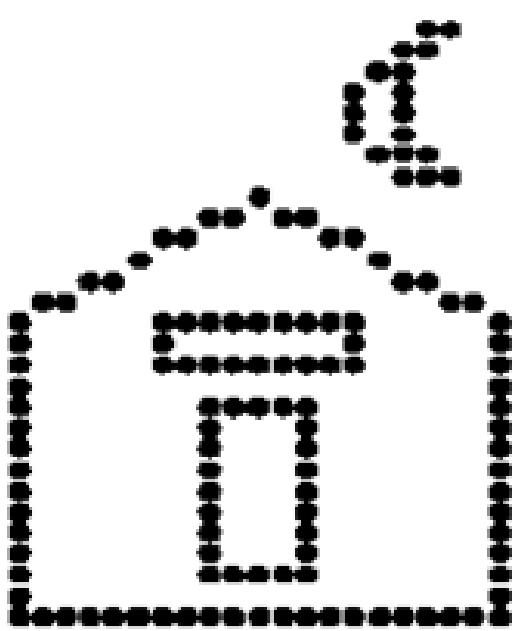
Vector Drawing

[van Dam]

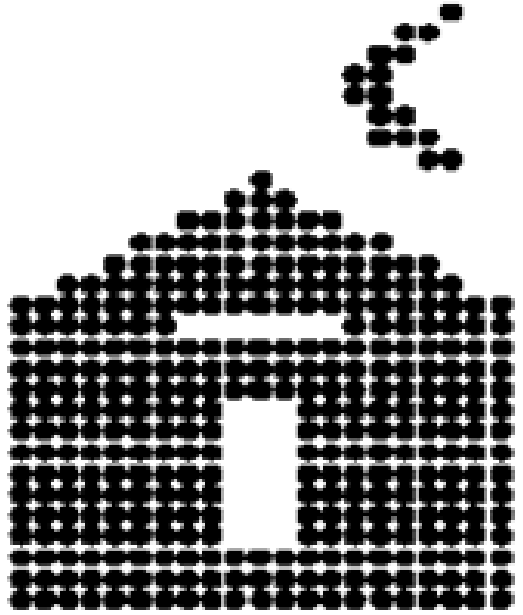
Vector graphics vs Raster graphics

Raster graphics is used in TV displays and laser printers

- Lowest level of representation
- No semantics
- BUT aliasing errors



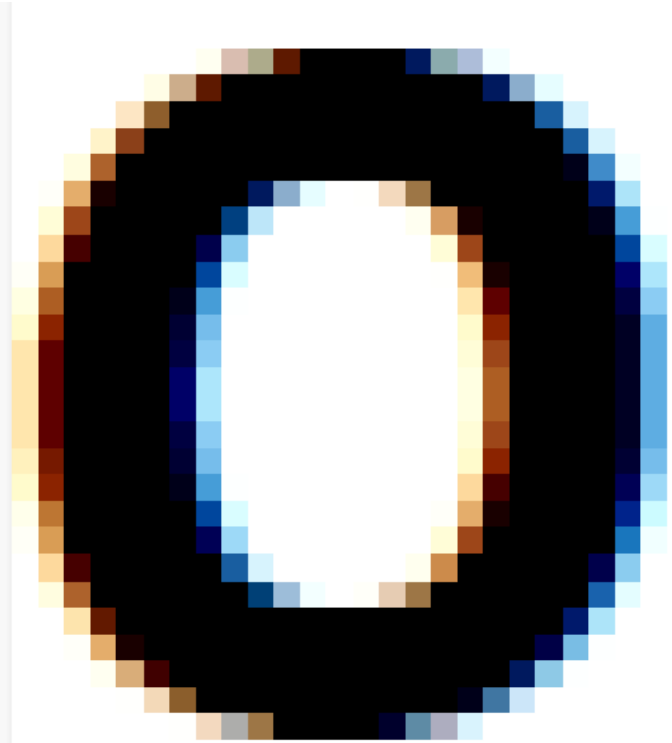
Outline

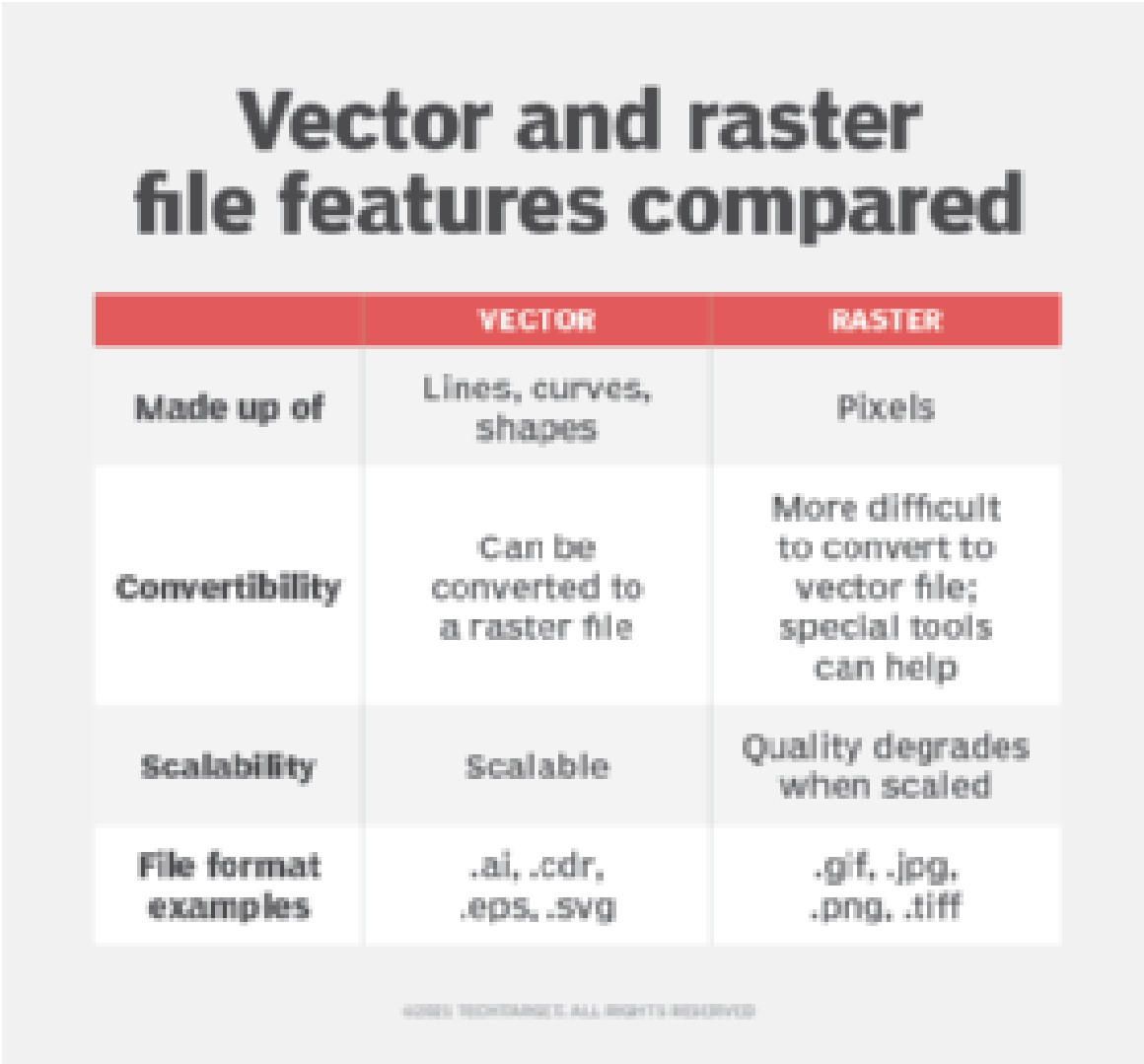


Filled

[van Dam]

1.ª Reunião

O 



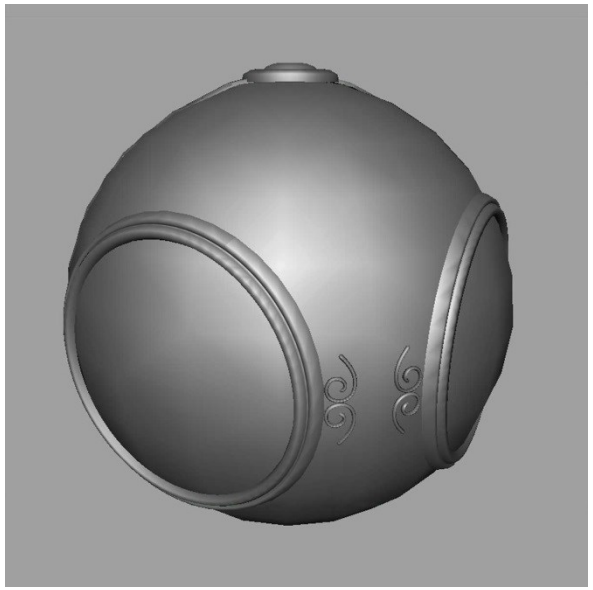
Vector and raster file features compared

	VECTOR	RASTER
Made up of	Lines, curves, shapes	Pixels
Convertibility	Can be converted to a raster file	More difficult to convert to vector file; special tools can help
Scalability	Scalable	Quality degrades when scaled
File format examples	.ai, .cdr, .eps, .svg	.gif, .jpg, .png, .tiff

...and as you can see, this table was saved using raster graphics...

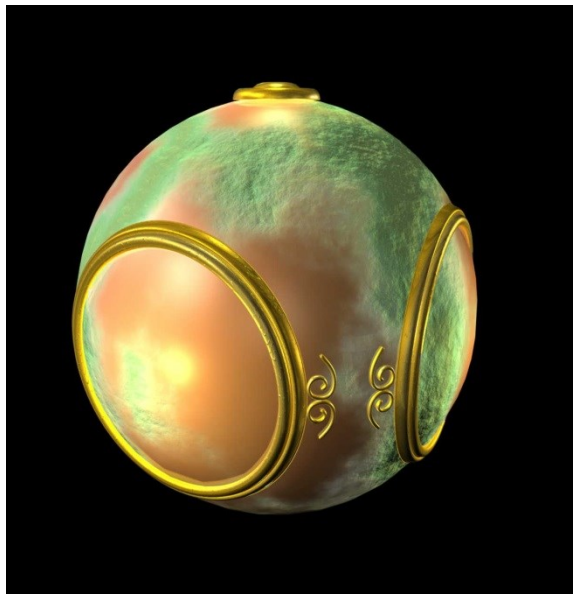
Computer Graphics: 1980 – 1990

The quest for **realism**



Smooth shading

Environment mapping



Bump mapping

Computer Graphics: 1980 – 1990

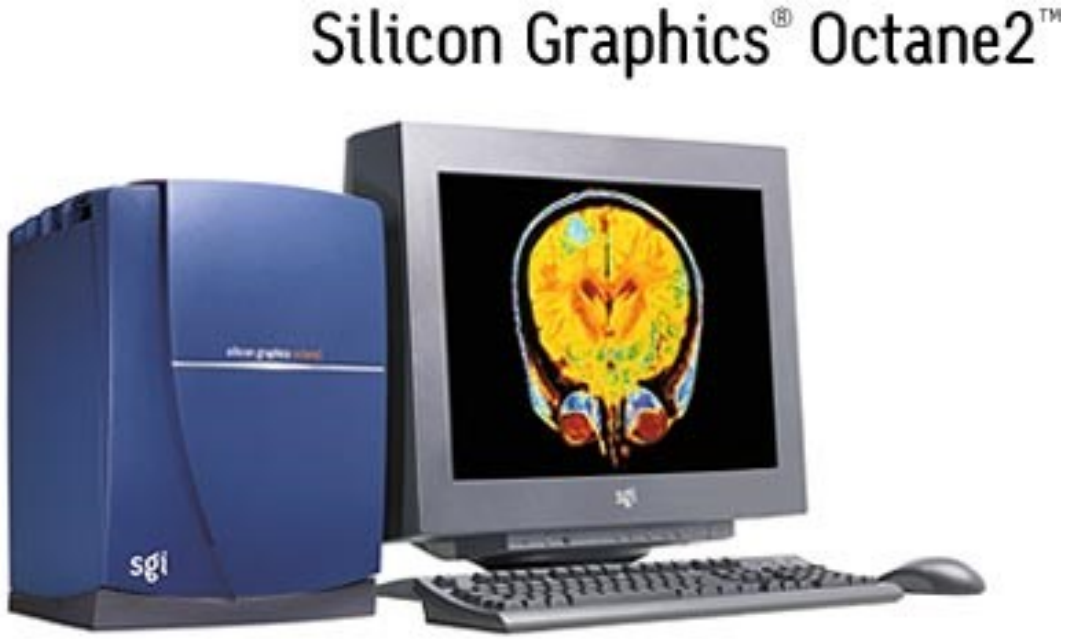
Special purpose **hardware**

- Graphics workstations

Industry-based **standards**

- PHIGS
- RenderMan

Human-Computer Interaction



Computer Graphics: 1990 – 2000

OpenGL API

First successful computer-generated feature- length animation film: **Toy Story**

New hardware capabilities

SGI Octane at the core….

Computer Graphics: 2000 – …

Photorealism

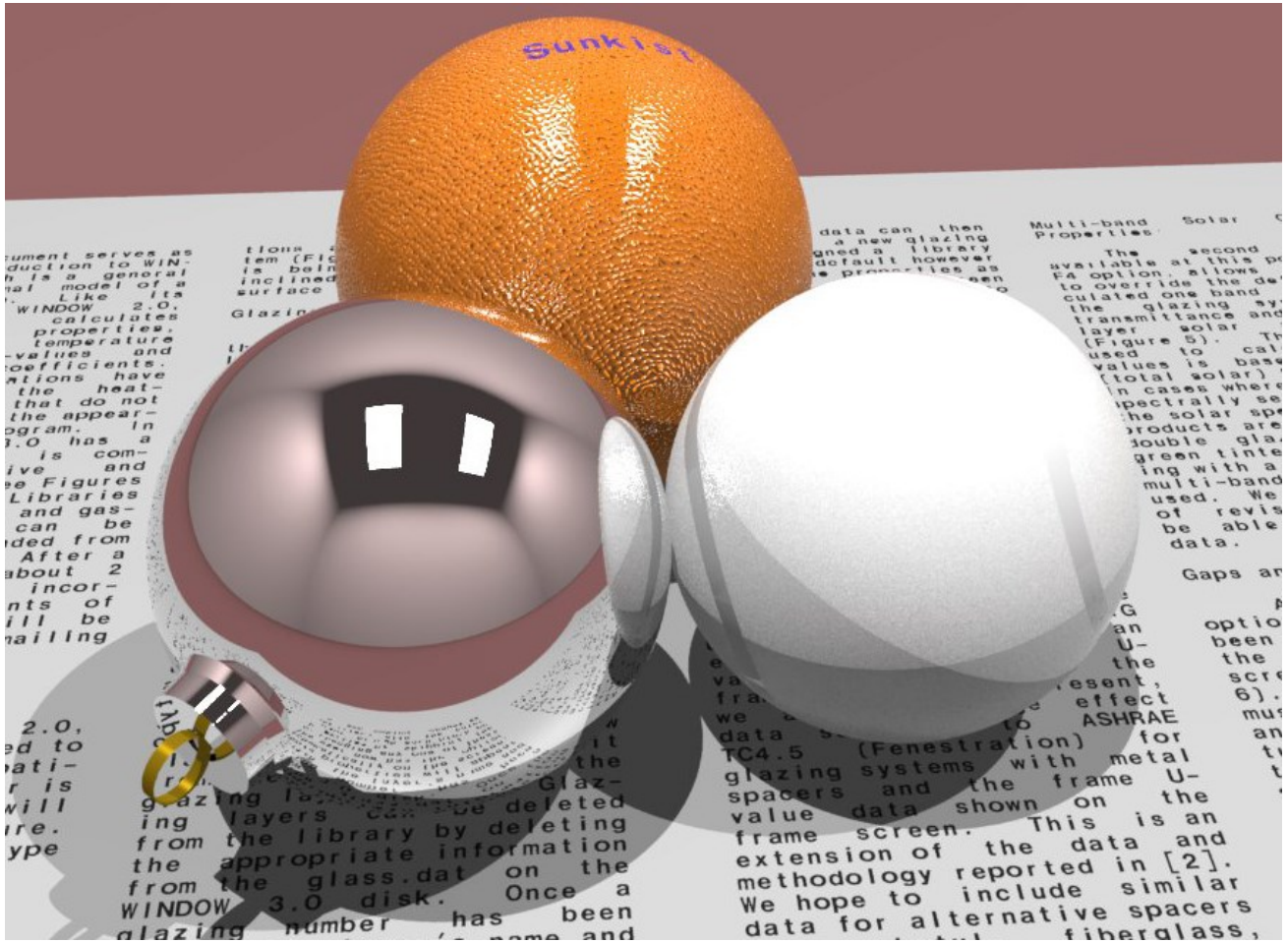
Graphics cards for PCs dominate the market

- Nvidia
- AMD (ATI)
- Intel Arc

Game boxes / players determine the market

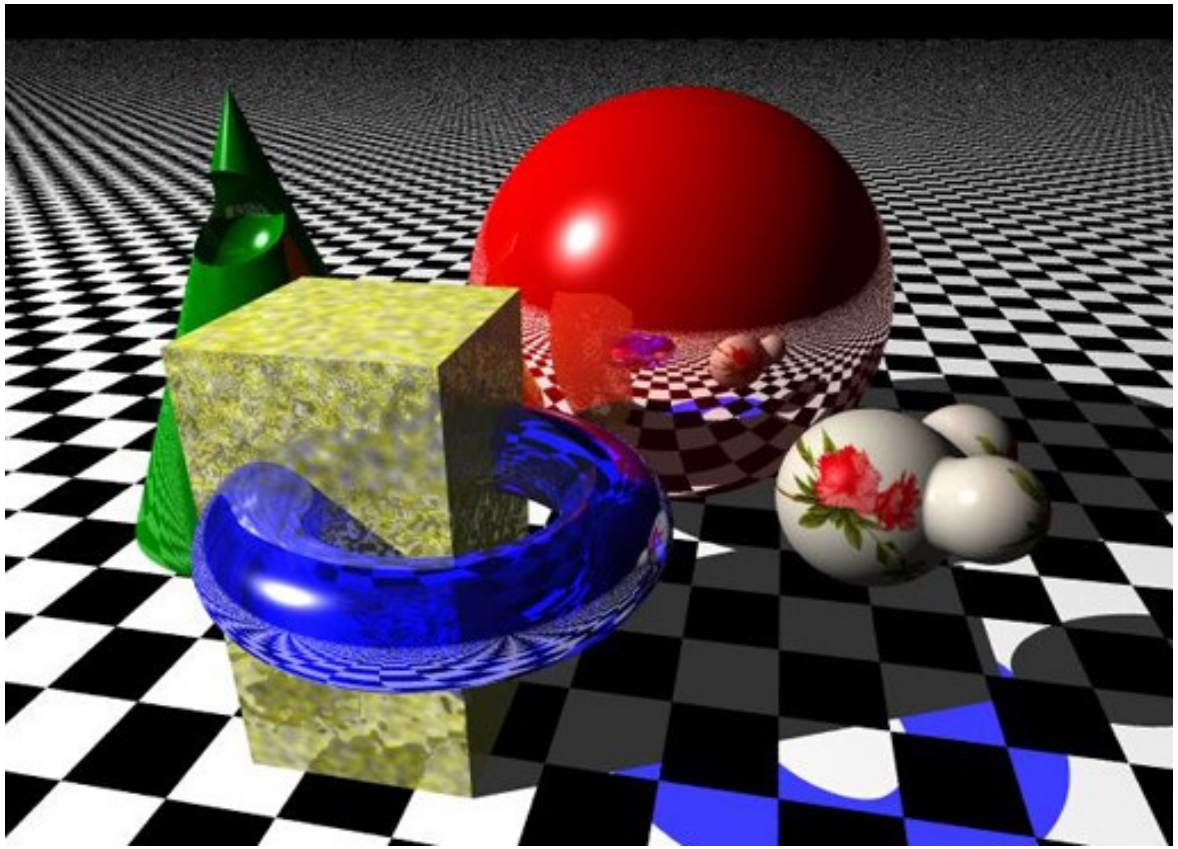
CG is routine in the film industry

Ray-tracing



http://radsite.lbl.gov/radiance/book/img/plate10.jpg

Ray-Tracing



http://www.tjhsst.edu/~dhyatt/superap/samplex.jpg

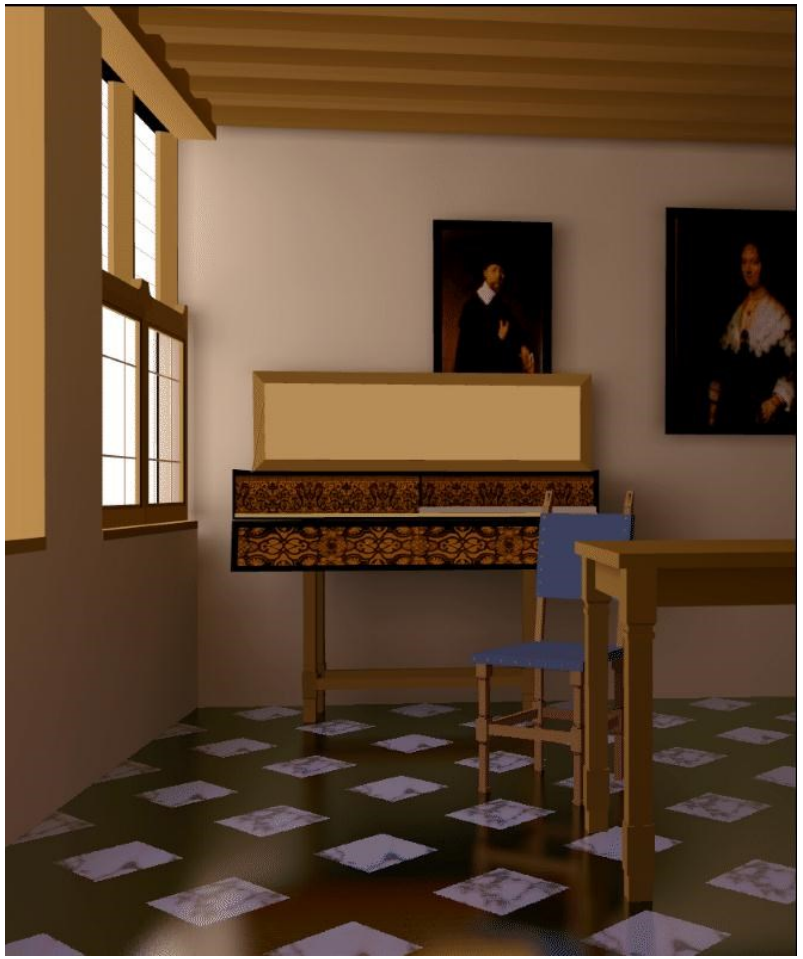

THE WITCHER WILD HUNT
THE WITCHER WILD HUNT
COMPLETE EDITION
Ray tracing

Ray tracing

CYBERPUNK 2077

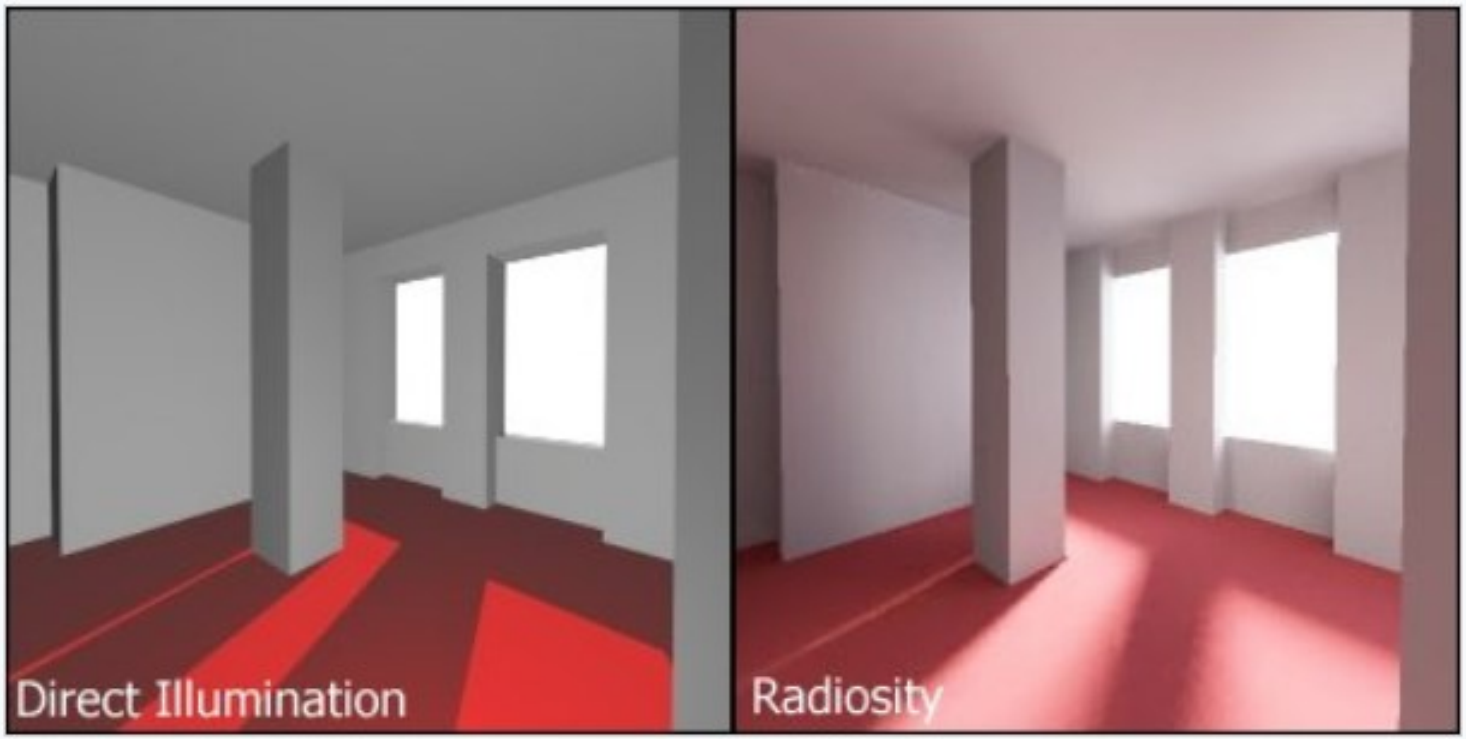
NVIDIA GEFORCE RTX

“Vermeer’s Studio”



Wallace & Cohen, 1987: Radiosity and Ray-Tracing

Radiosity



Difference between standard direct illumination without shadow umbra, and radiosity with shadow umbra

Raytracing + Radiosity

Radiosity

Without radiosity

With radiosity

[Burdea]

Textures simulating raytracing

Increased realism. 11 light sources + 25 texture maps

https://schoolofmusic.ucla.edu/

Current Panorama

Augmented Reality used in Hollywood film making

Virtual Reality is becoming massive !

- Crash test dummies
- New car design / development
- Entertainment

Pixar!

- RenderMan is free

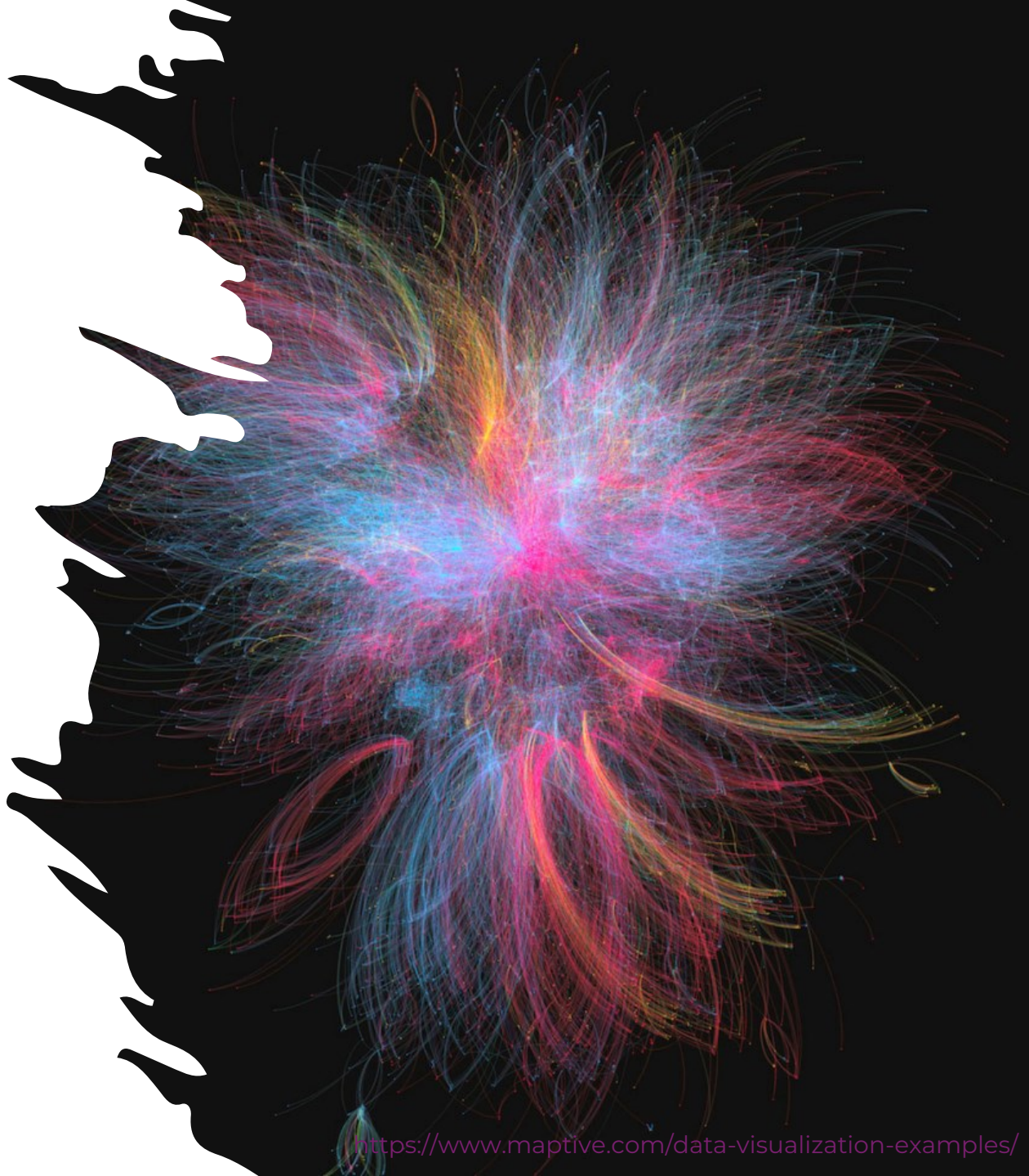
Current Panorama

Science and Maths profiting from advances

- Modeling / Simulation / Animation

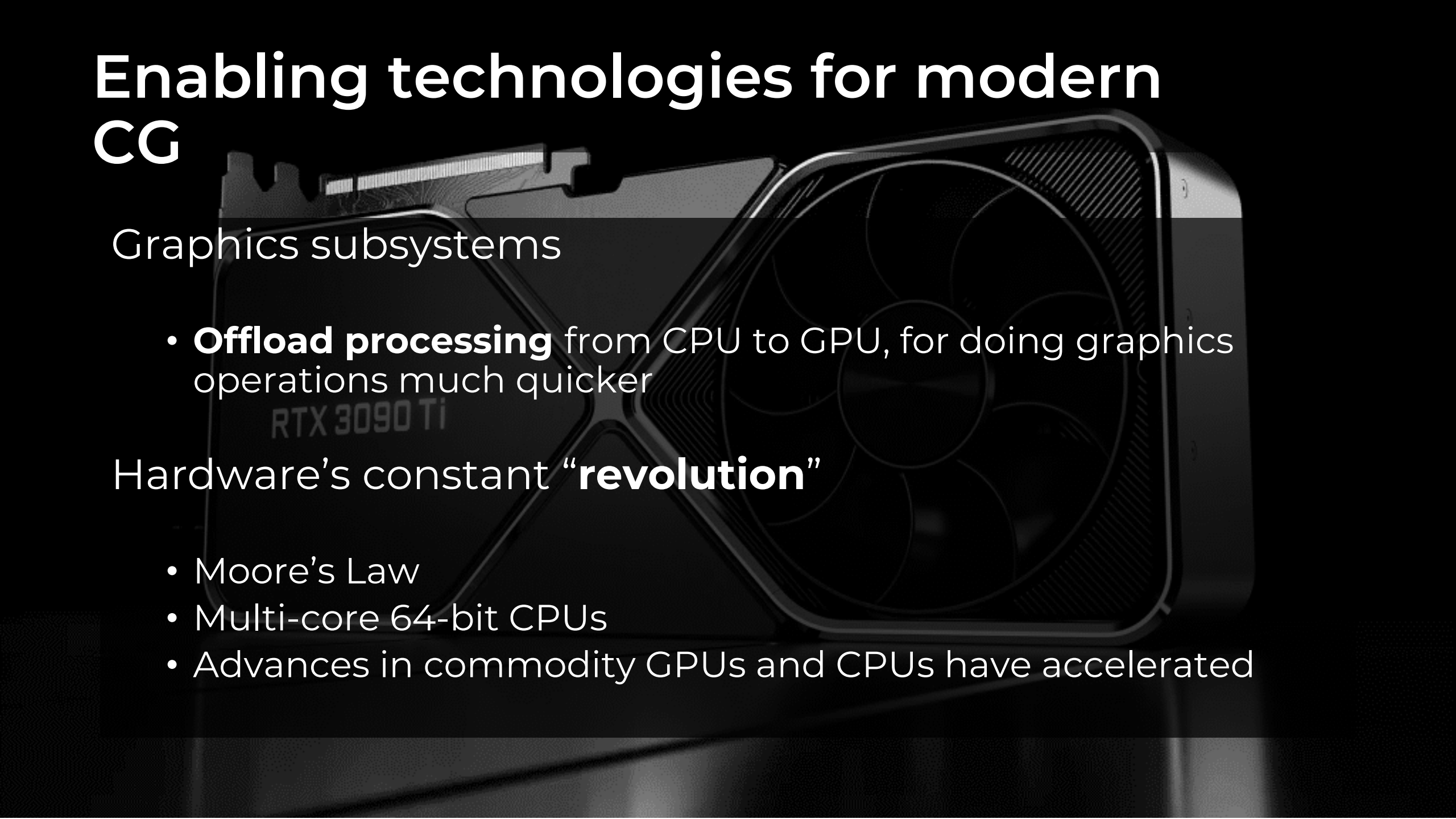
Gamification

- Unreal Engine / Unity / Cryengine



https://www.maptive.com/data-visualization-examples/

Enabling technologies for modern CG



Graphics subsystems

- **Offload processing** from CPU to GPU, for doing graphics operations much quicker

Hardware's constant "**revolution**"

- Moore's Law
- Multi-core 64-bit CPUs
- Advances in commodity GPUs and CPUs have accelerated

Interactive vs batch CG

CG Main Tasks

Modeling

- Construct individual models / objects
- Assemble them into a 2D or 3D scene

Animation

- Static vs. dynamic scenes
- Movement and / or deformation

Rendering

- Generate final images
- Where is the observer?
- How is he / she looking at the scene?

https://www.atlassian.com/blog/trello/7-ways-to-track-tasks-in-trello

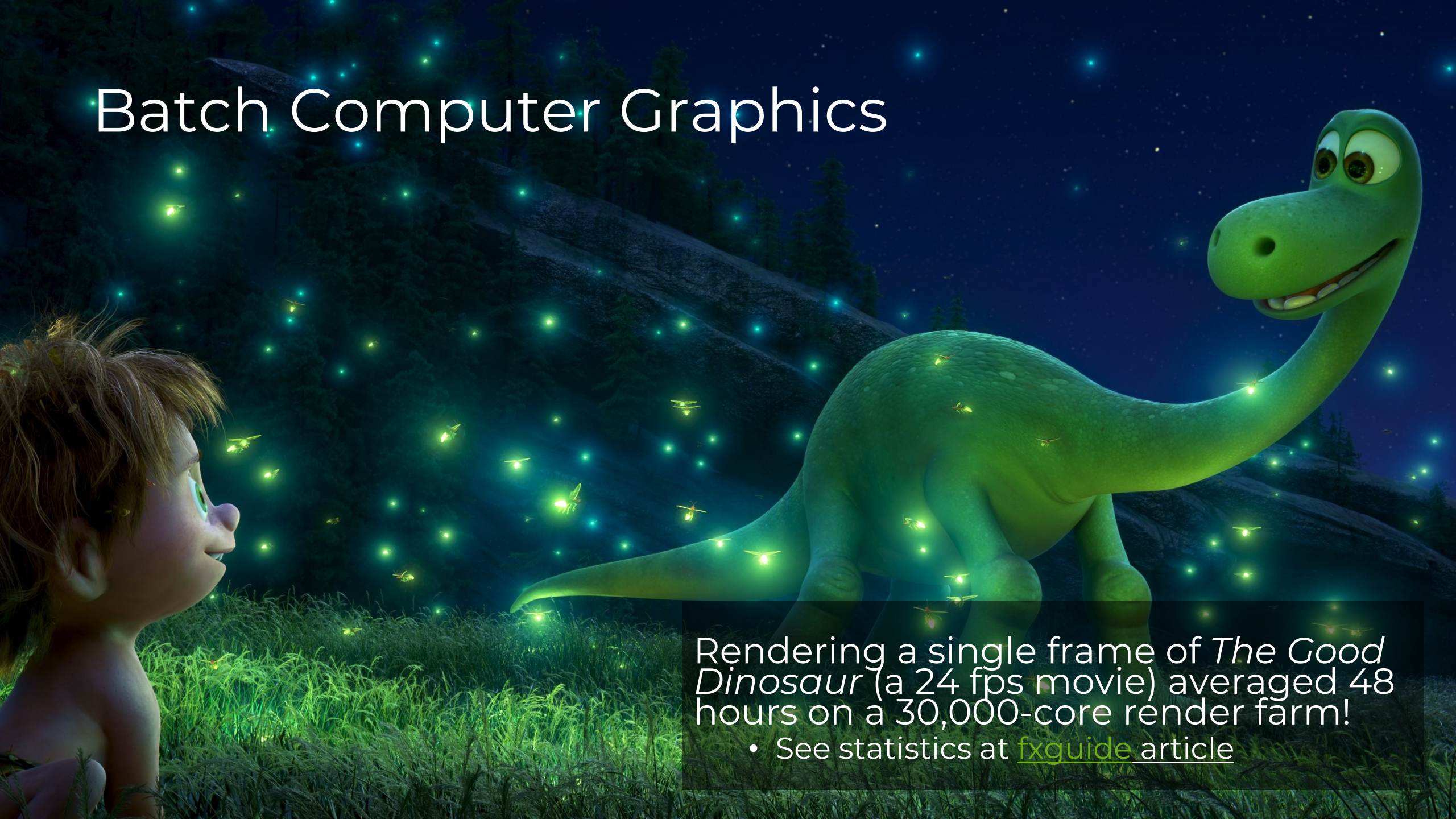
Batch Computer Graphics

Non-interactive, off-line rendering

Final production-quality video and film

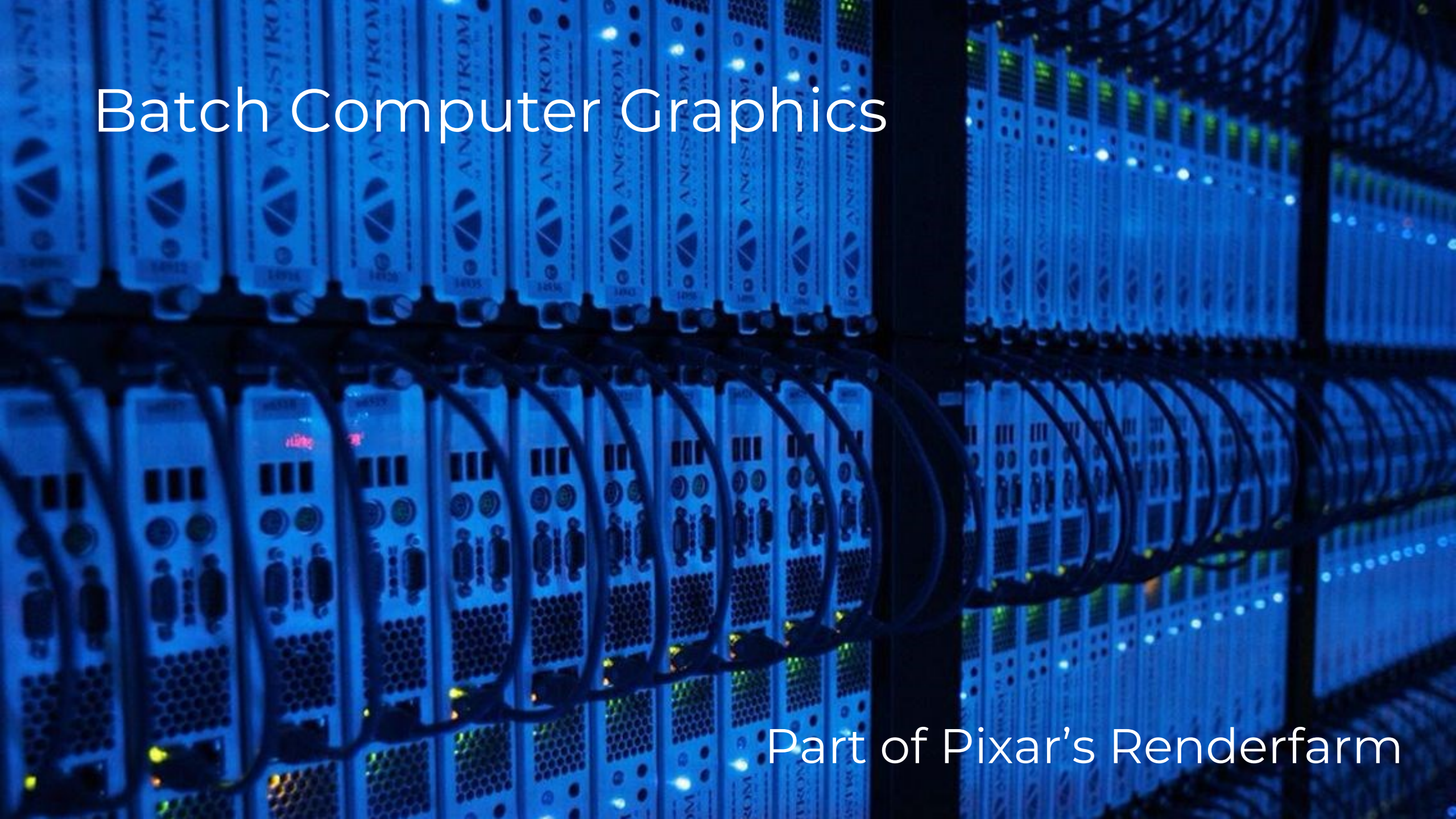
- Animation / Special effects – FX

Batch Computer Graphics



Rendering a single frame of *The Good Dinosaur* (a 24 fps movie) averaged 48 hours on a 30,000-core render farm!

- See statistics at fxguide article


Batch Computer Graphics
Part of Pixar’s Renderfarm

Interactive Computer Graphics

User controls content, structure, and appearance of objects and their displayed images, via rapid visual feedback

- Also called **real-time computer graphics** or, in certain contexts, real-time rendering

Remember Sutherland's Sketchpad (1963)

- Monitor + light pen + function-key panels
- Bimanual operation

Basic Graphics System

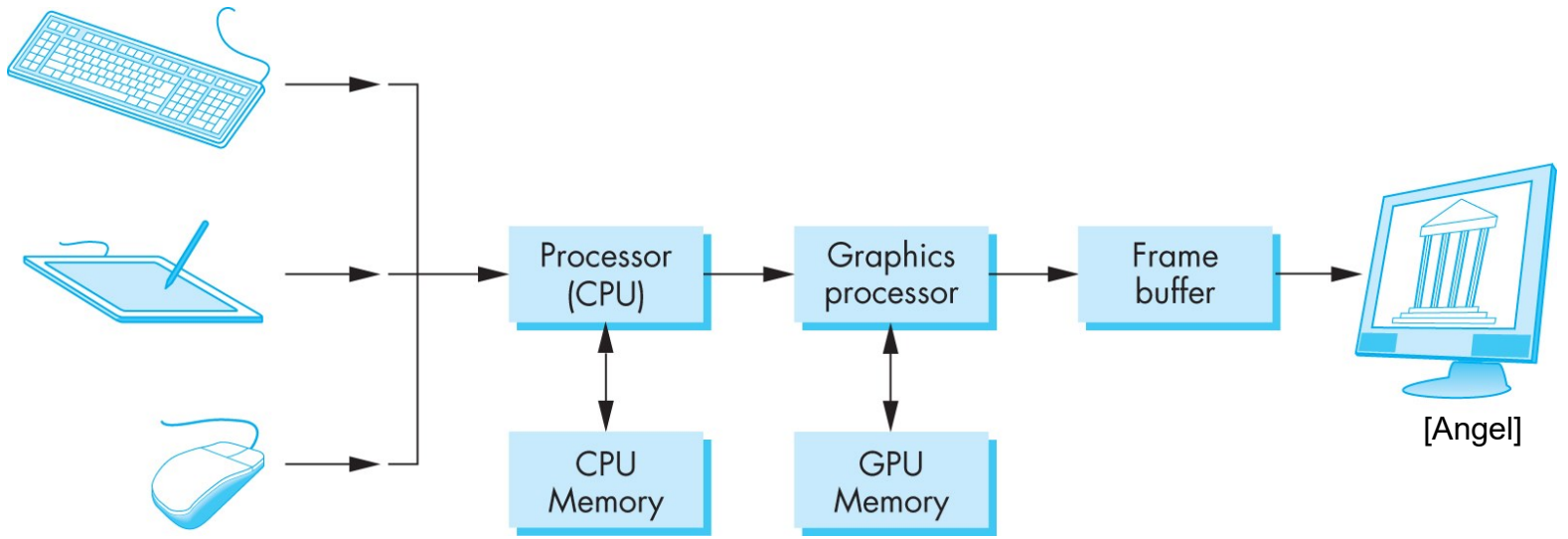


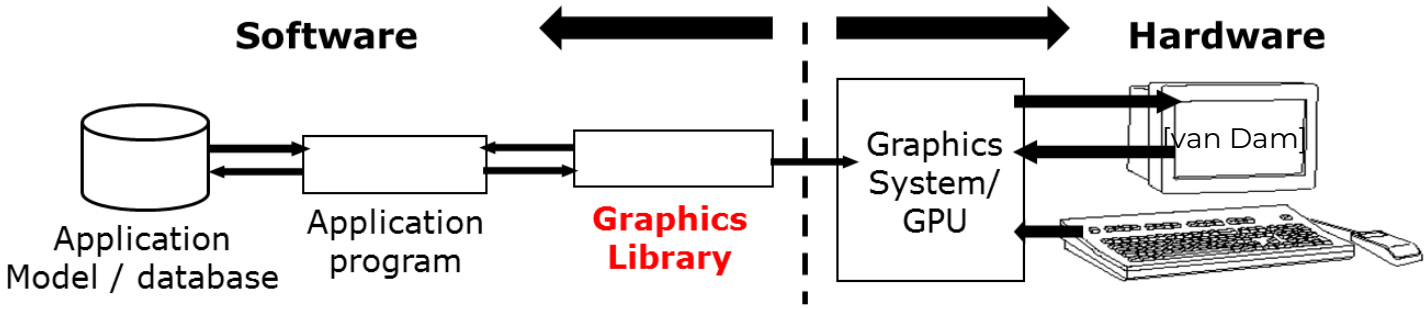
Image formed in the frame buffer… actually, the back buffer, but we will get into that later

Interactive Computer Graphics

Graphics library / package is intermediary between application and display hardware

Application program maps / renders objects / models to images by calling on the graphics library

User interaction allows image and / or model modification



Computer graphics APIs

Samsung Developers (w/ Vulkan)

Computer Graphics APIs

Create 2D / 3D scenes from simple primitives

OpenGL

- Rendering
- No modeling or interaction facilities

RenderMan, DirectX, Windows Presentation Foundation (WPF), HTML5 + WebGL, …


Graphics Libraries / APIs

Primitives : characters, points, lines, triangles…

Attributes : color, line /polygon style, …

Transformations : rotation, scaling, …

Light sources

Viewing…

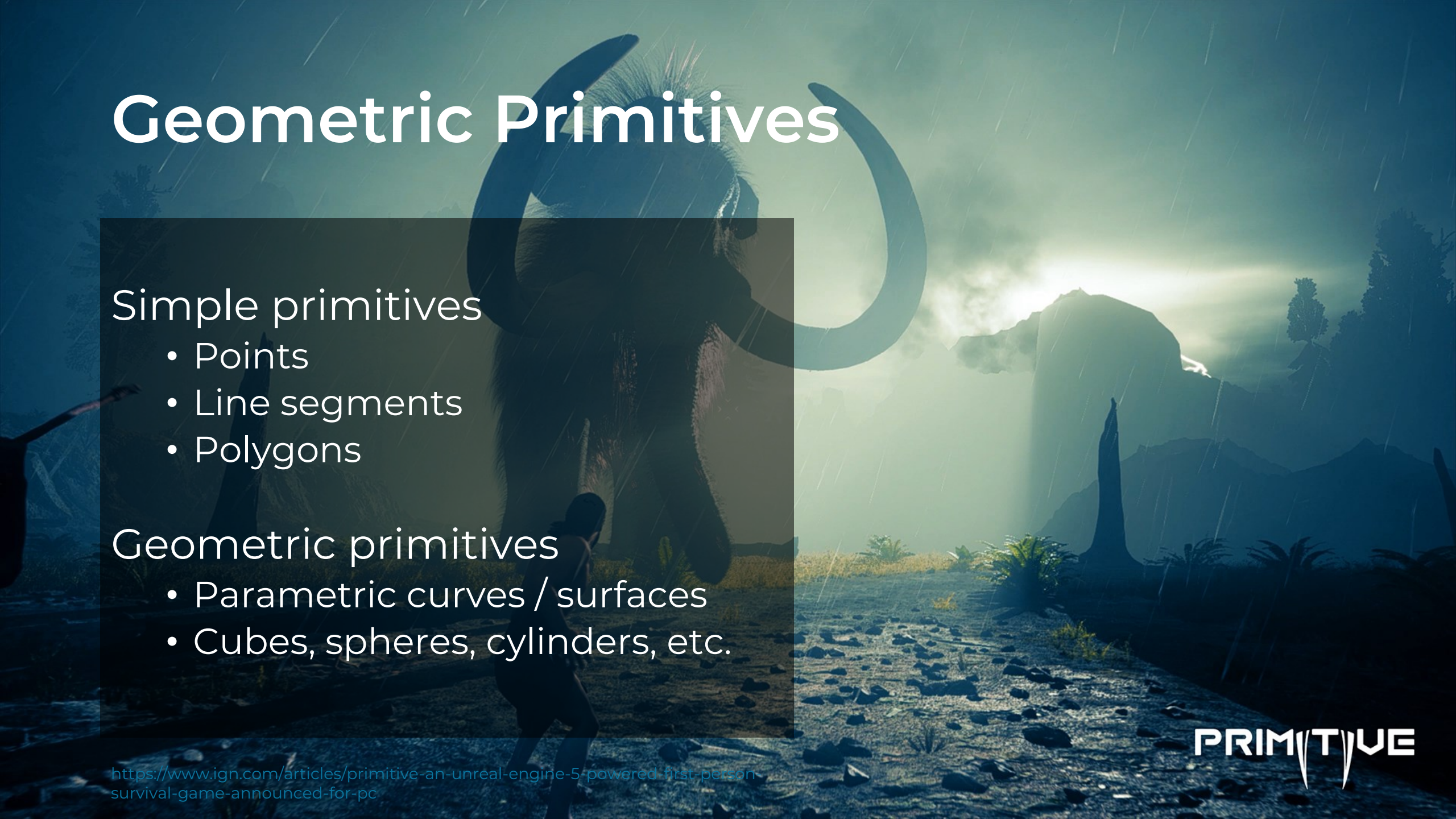
Geometric Primitives

Simple primitives

- Points
- Line segments
- Polygons

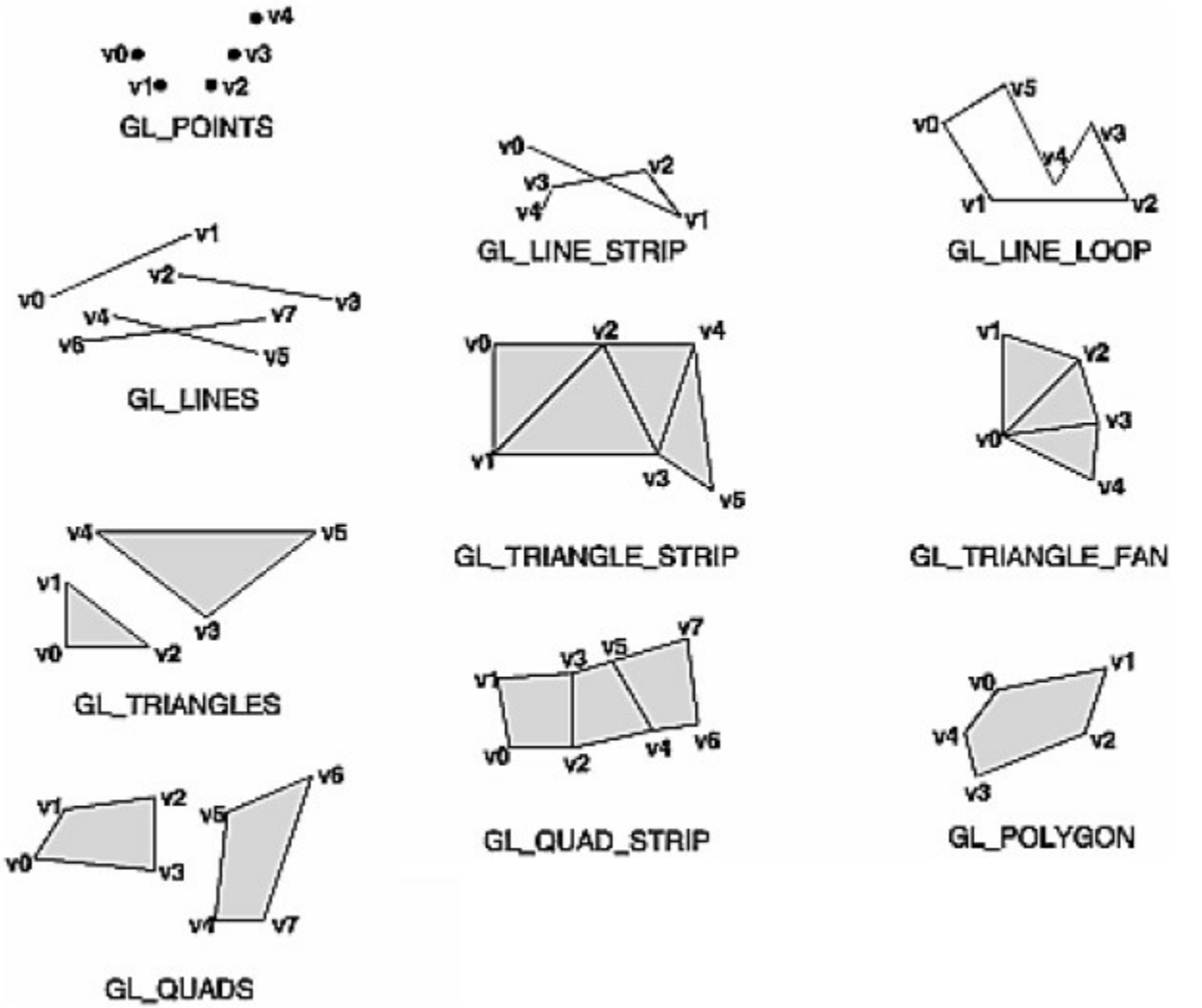
Geometric primitives

- Parametric curves / surfaces
- Cubes, spheres, cylinders, etc.

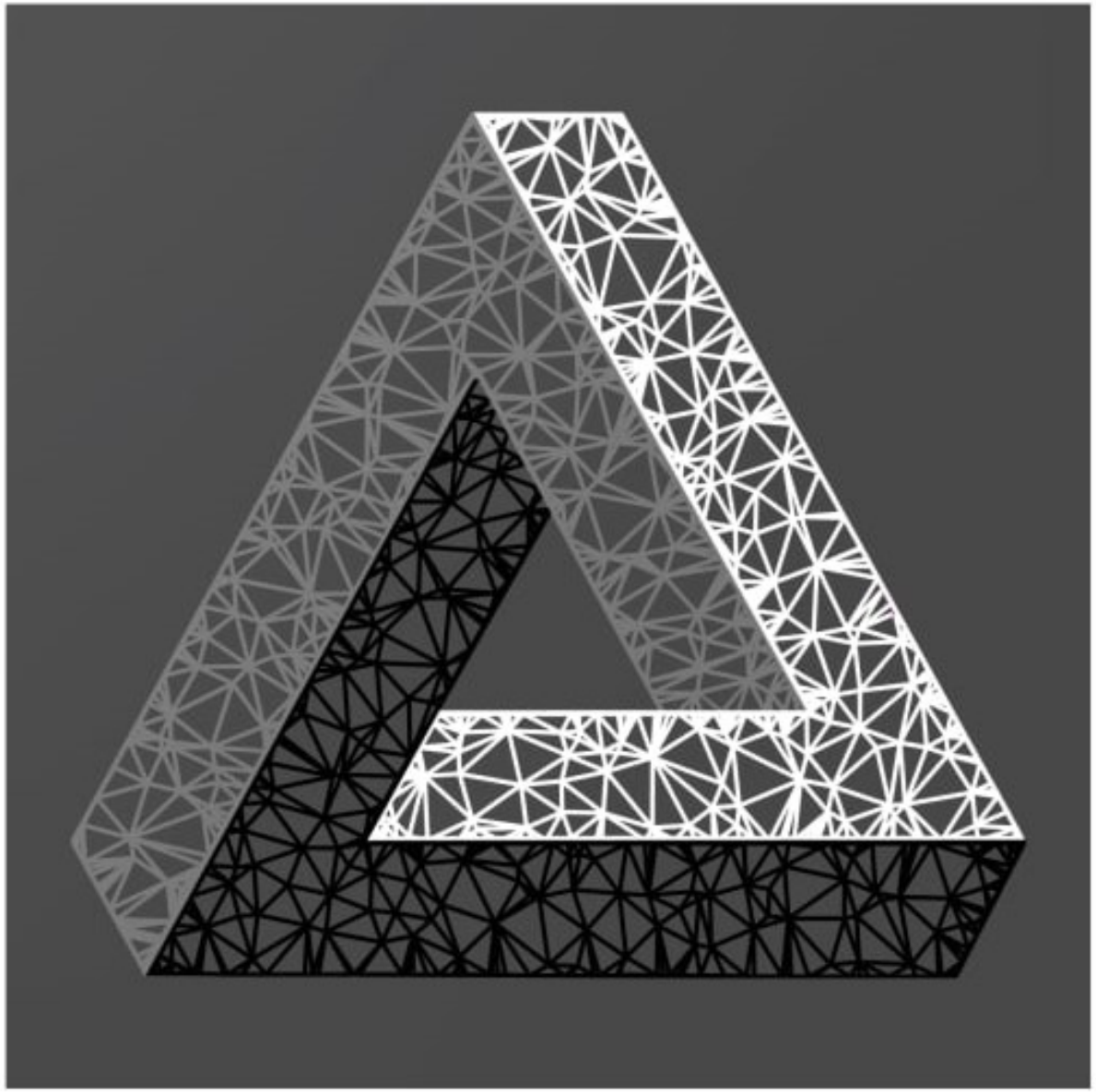


https://www.ign.com/articles/primitive-an-unreal-engine-5-powered-first-person-survival-game-announced-for-pc

OpenGL Primitives



With so many shapes, why is this fixation for triangles in Computer Graphics?



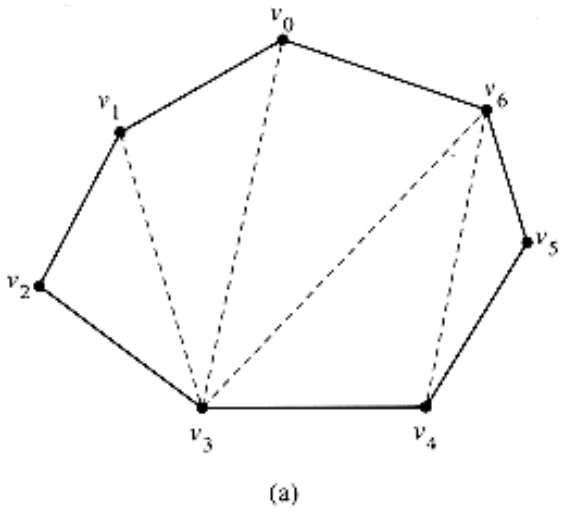
Why triangles?

Triangles are:

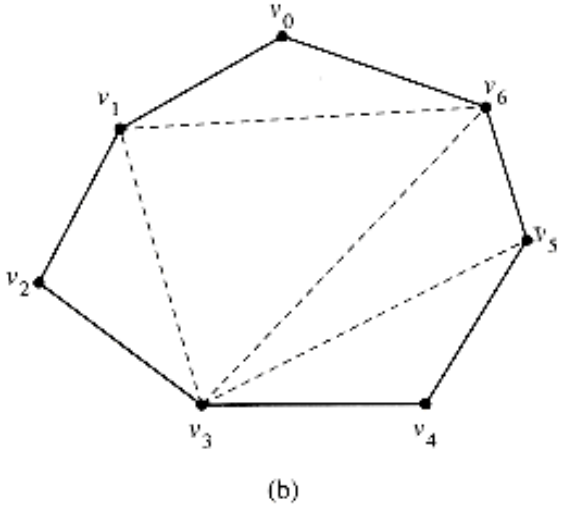
- Simple
  - Edges cannot cross
- Convex
  - All points on a line segment between two points on a triangle also belong to the triangle
- Flat
  - All triangle points belong to the same plane

Triangulations

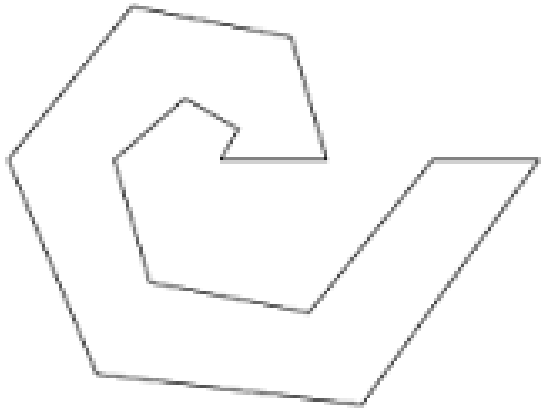
Application program must tesselate a polygon into triangles

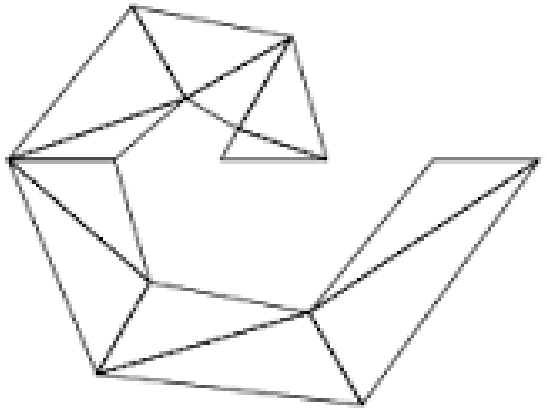


(a)



(b)





[ustc.edu.cn] [Mathworld]

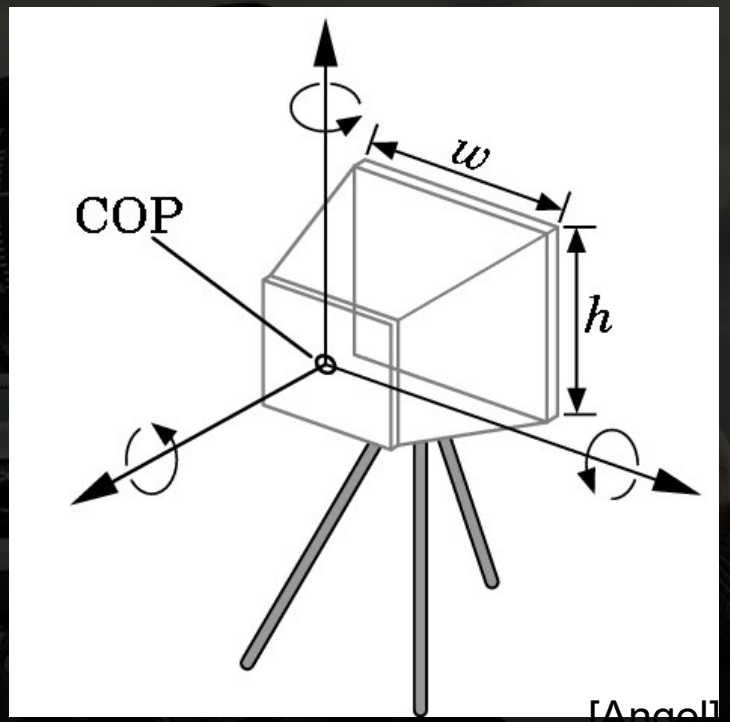
Attributes

Determine the appearance of primitives

- **Color**
- Size and width
- Stipple pattern
- Polygon mode
  - Display as **filled**: solid color or stipple pattern
  - Display **edges**
  - Display **vertices**

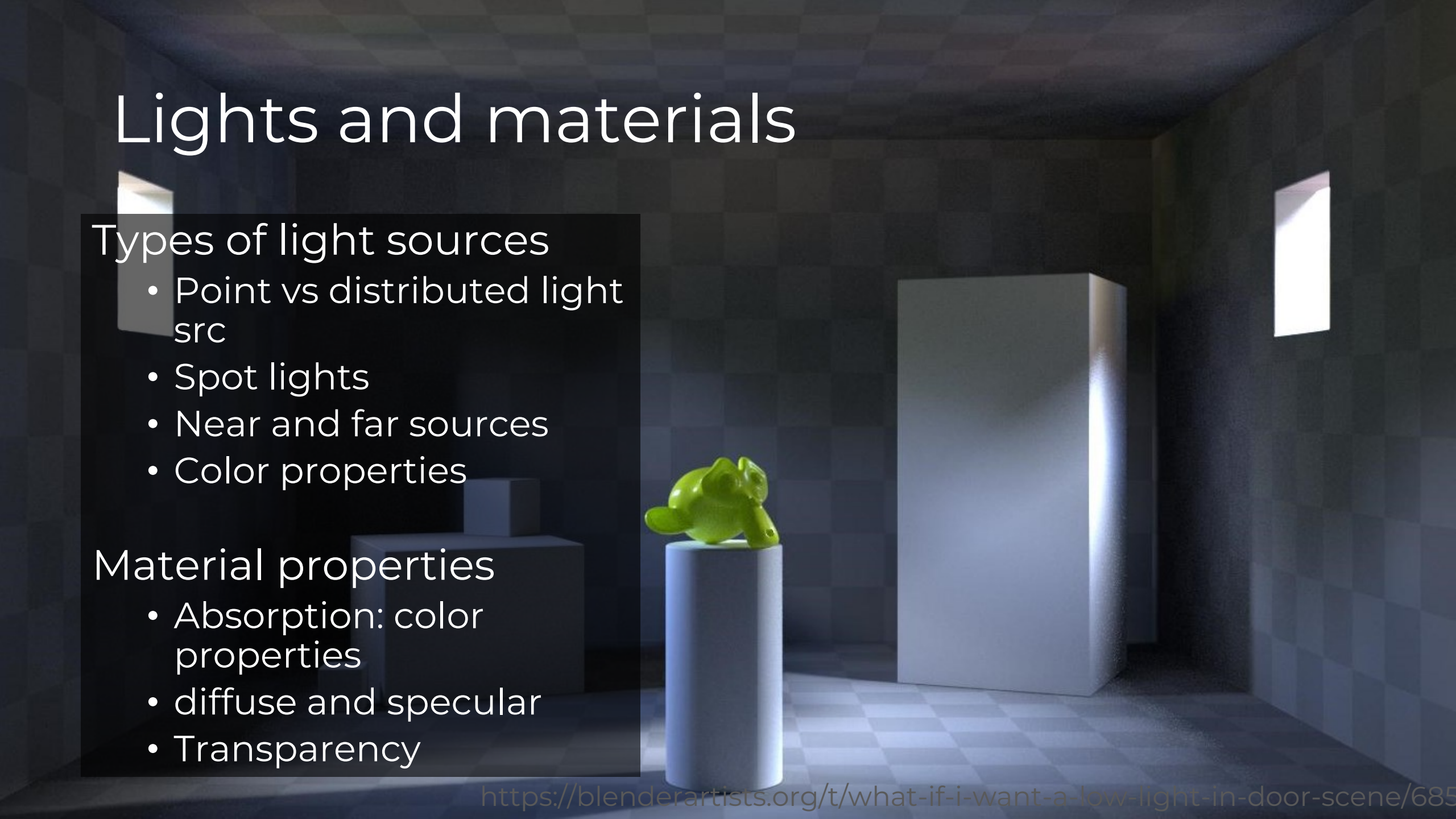
Camera specification

- Six degrees of freedom
  - Position of lens center
- Lens
- Film size
- Orientation of film plane



[Angel]

Lights and materials



Types of light sources

- Point vs distributed light src
- Spot lights
- Near and far sources
- Color properties

Material properties

- Absorption: color properties
- diffuse and specular
- Transparency

https://blenderartists.org/t/what-if-i-want-a-low-light-in-door-scene/685

OpenGL

Multi-platform API for rendering 2D and 3D computer graphics

Interaction with the GPU to achieve hardware-accelerated rendering

Application areas

- CAD
- Virtual reality
- Scientific and Information Visualization

Why OpenGL ?

Industry **standard** for real-time **2D / 3D CG**

Available on most platforms

- Desktop / laptop operating systems
- Mobile devices – OpenGL ES
- Browsers – **WebGL**

Immediate-mode graphics API

- Applications instruct OpenGL to **draw** the **primitives**

OpenGL (old and new)

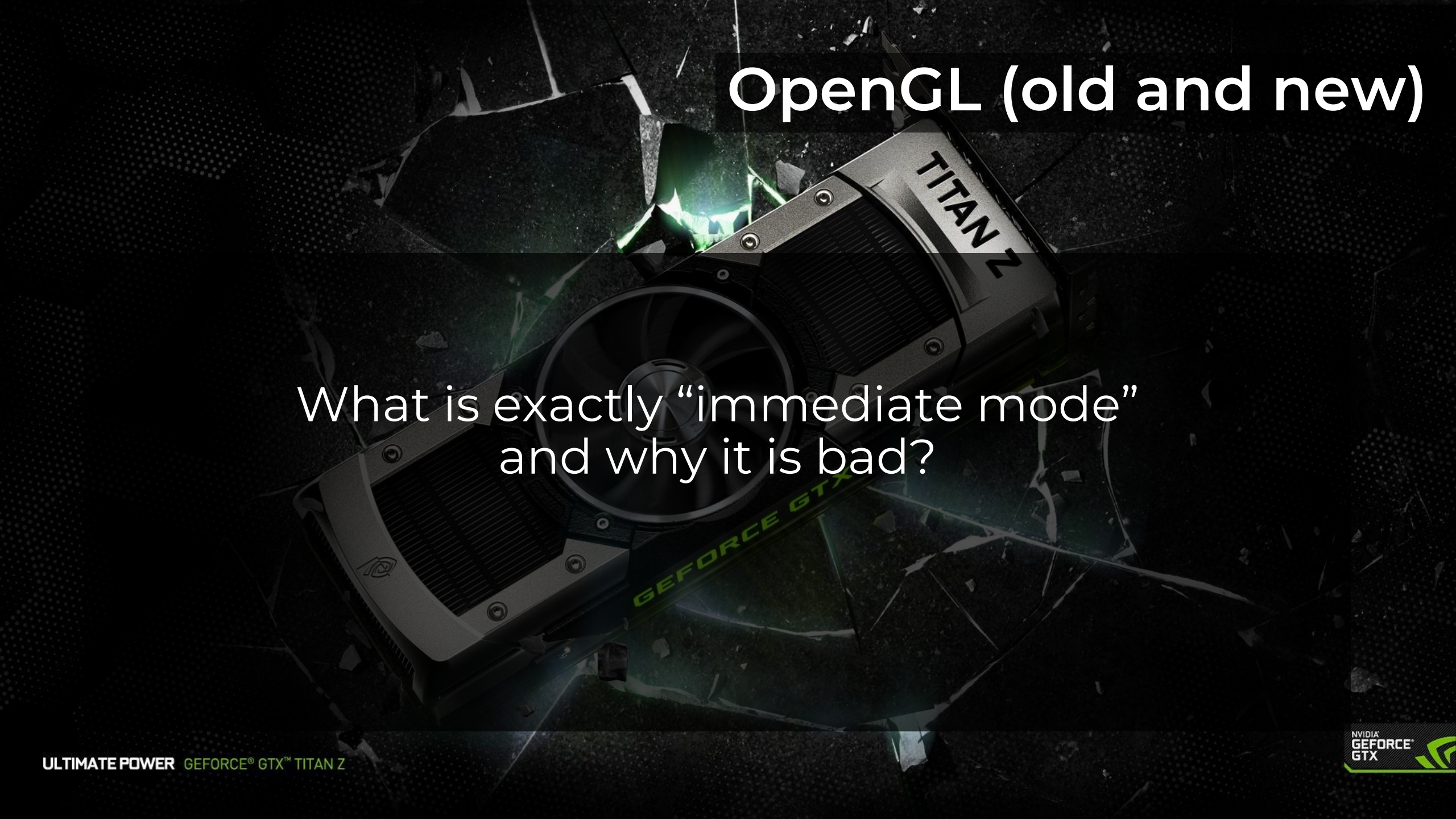
Older API (OpenGL 1.0) provides features for rapid prototyping

Works in “immediate mode”

- glBegin… glEnd…

It is **deprecated**!

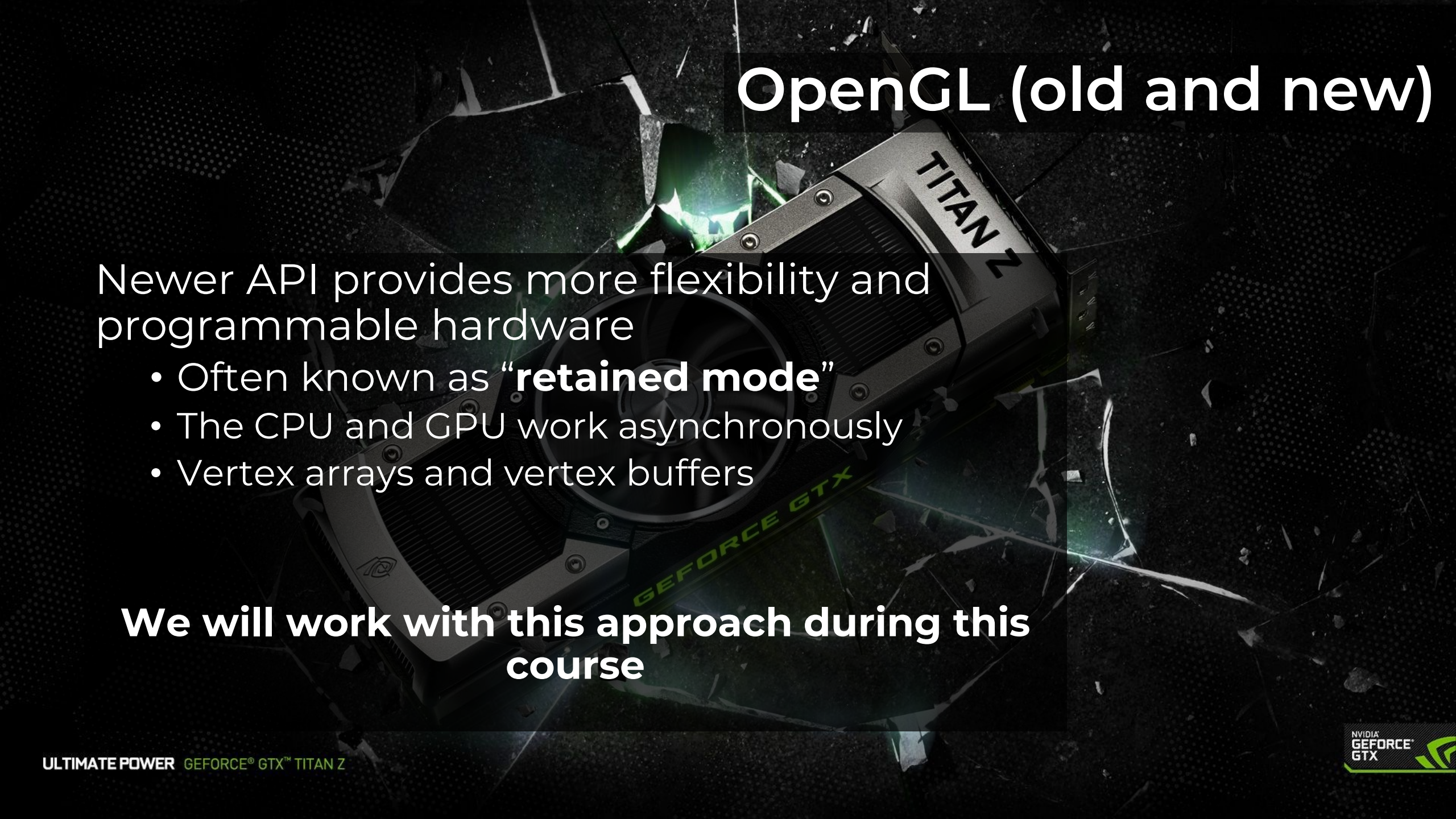
OpenGL (old and new)



What is exactly “immediate mode”
and why it is bad?


ORIGINAL
REMASTER
Quake II used
immediate mode

OpenGL (old and new)

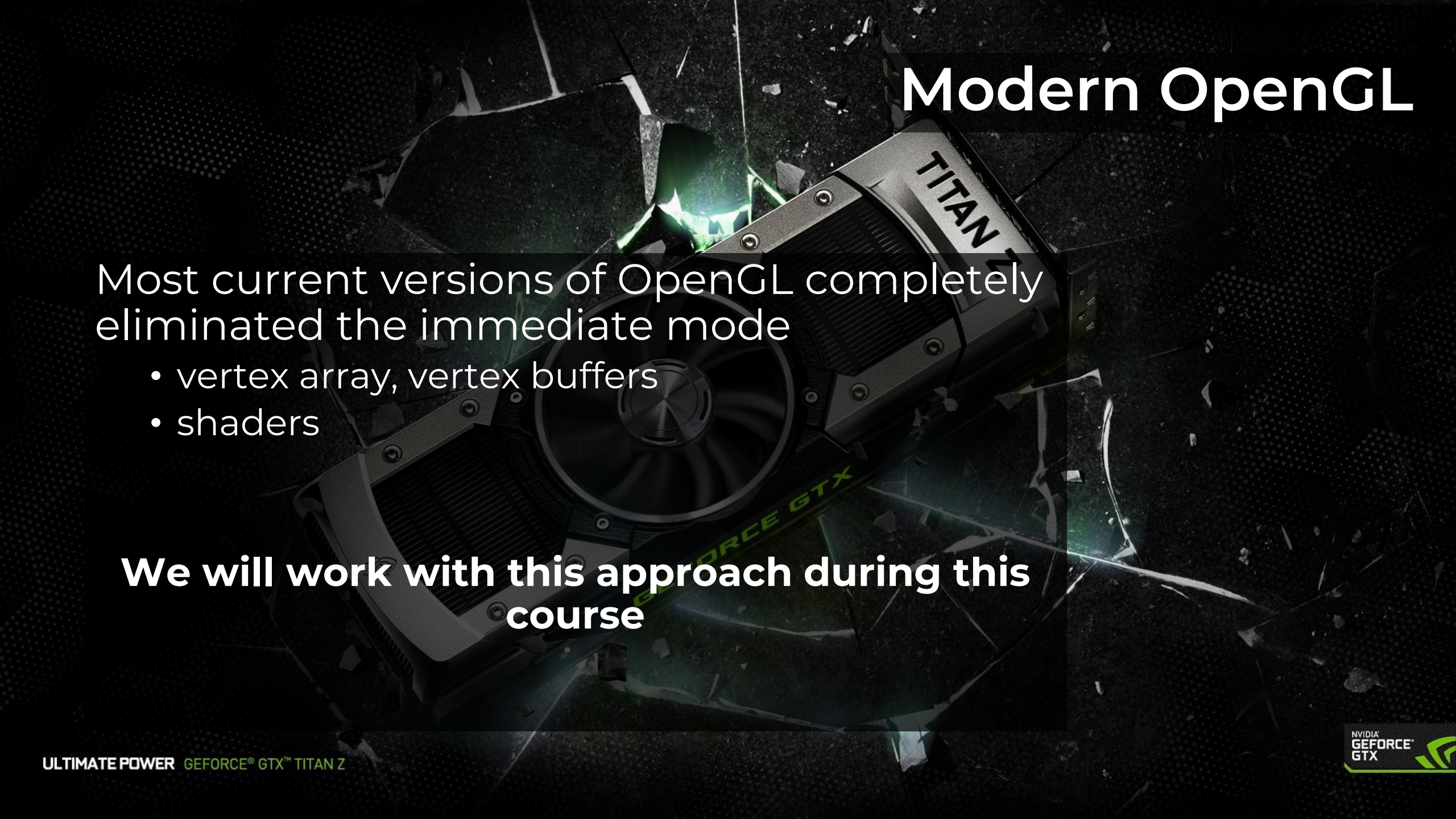


Newer API provides more flexibility and programmable hardware

- Often known as “**retained mode**”
- The CPU and GPU work asynchronously
- Vertex arrays and vertex buffers

We will work with this approach during this course

Modern OpenGL



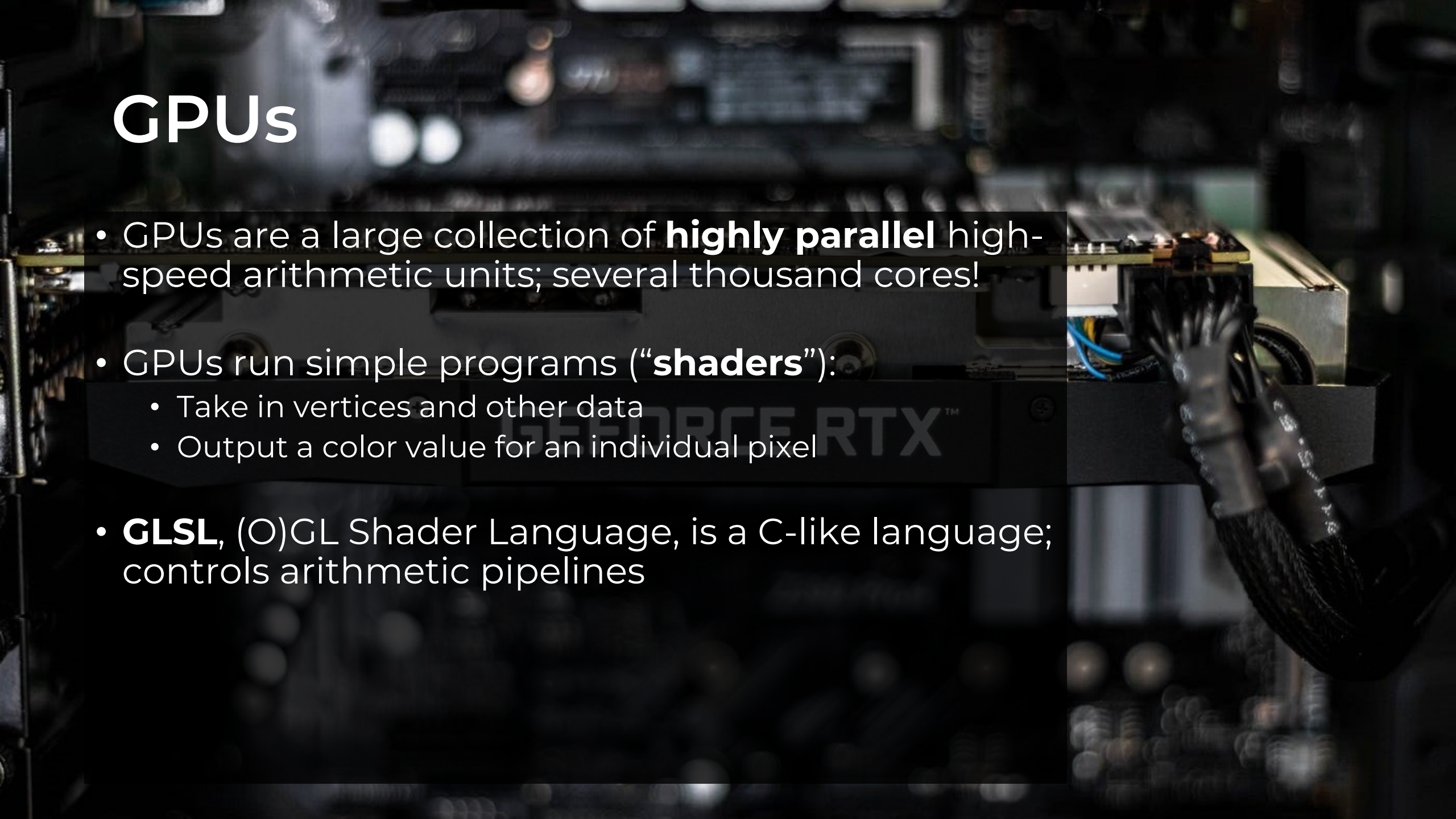
Most current versions of OpenGL completely eliminated the immediate mode

- vertex array, vertex buffers
- shaders

We will work with this approach during this course

GPUs

- GPUs are a large collection of **highly parallel** high-speed arithmetic units; several thousand cores!
- GPUs run simple programs (“**shaders**”):
  - Take in vertices and other data
  - Output a color value for an individual pixel
- **GLSL**, (O)GL Shader Language, is a C-like language; controls arithmetic pipelines

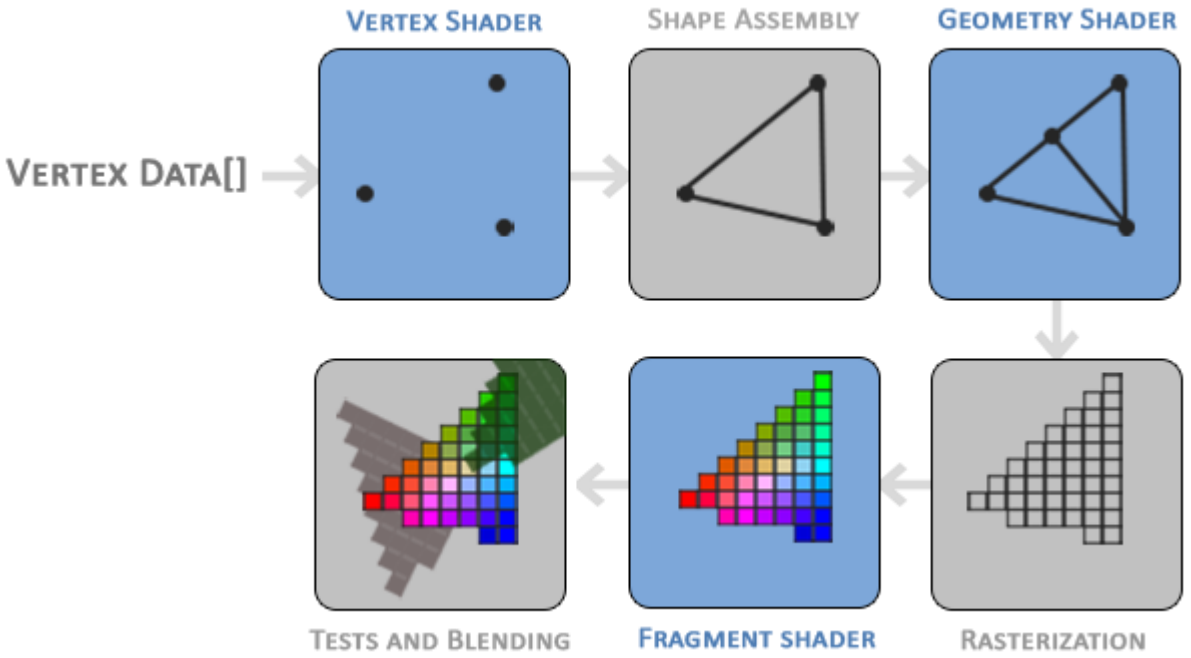


Computer Graphics Pipeline

Everything starts with vertex data and into the **vertex shader**

Shapes are assembled

A geometry shader deals with tesselation

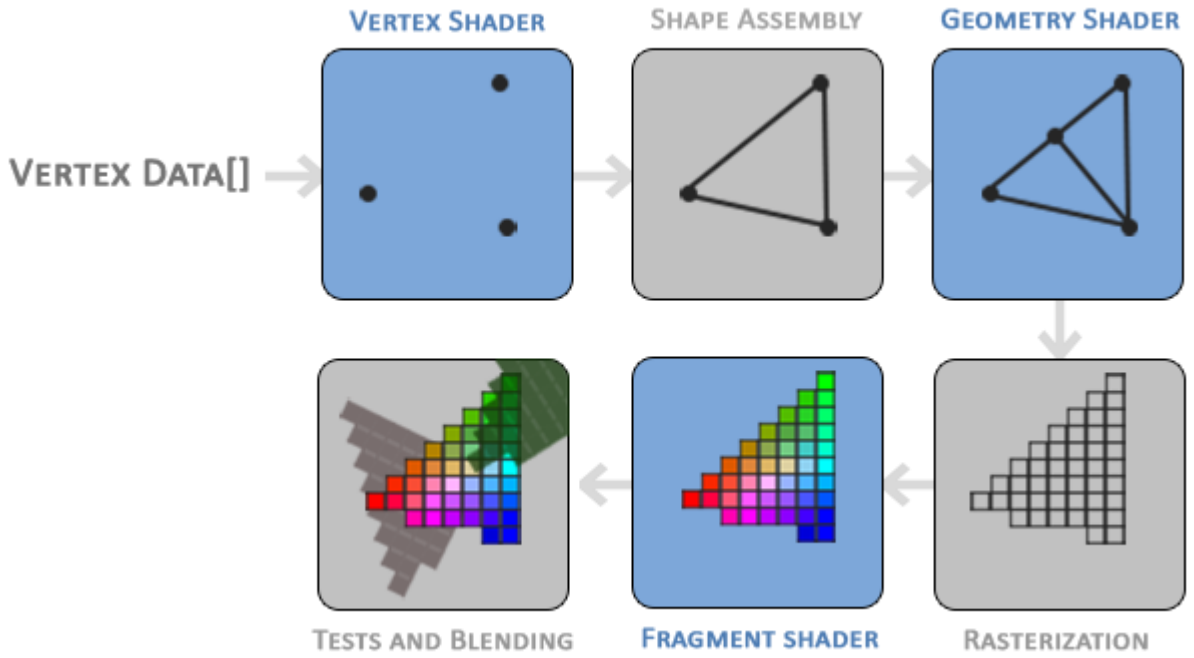


Computer Graphics Pipeline

Rasterization is performed (which pixels do I need to fill?)

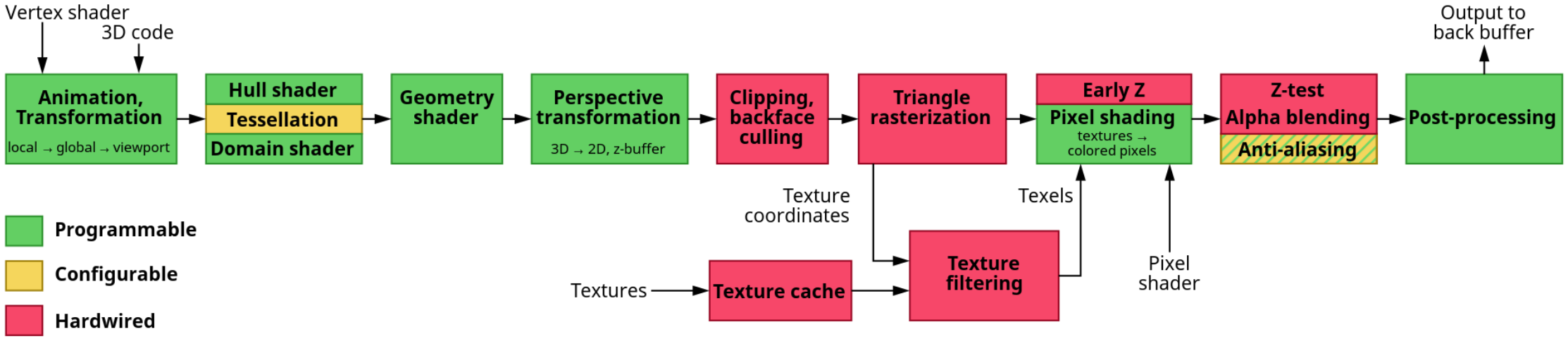
Colour is attributed in the **fragment shader**

Occlusion/blending are performed



learnopengl.com

Computer Graphics Pipeline



It is a little bit more complicated, but we will not go into all the detail

Why Learn 2D first ?

- A good stepping stone towards 3D
- Many issues easier to understand in 2D
- No need to simulate cameras / light sources / light interaction with objects ...
- 2D is still the most common use of CG

Bibliography

For a bit more detail:

- K. Akeley, J. D. Foley, et al., “Computer Graphics: Principles and Practice”, Addison-Wesley, 2013 https://learning.oreilly.com/library/view/hughes-computer-graphics-3-e/9780133373721/

- “Hello Triangle” in LearningOpenGL.com, (this reference provides OpenGL examples in C, but the explanation of the graphics pipeline is good and the code is not that different from how to do it in Python https://learnopengl.com/Getting-started/Hello-Triangle

- S. Marschner, P. Shrley, “The Graphics Pipeline”, chapter 8 of “Fundamentals of Computer Graphics”, 4th ed, A K Peters, 2018 https://learning.oreilly.com/library/view/fundamentals-of-computer/9781482229417/K22616_C008.xhtml#sec8-1-1

And now...

Hands on!

https://img.freepik.com/free-photo/construction-worker_329181-2850.jpg?w=2