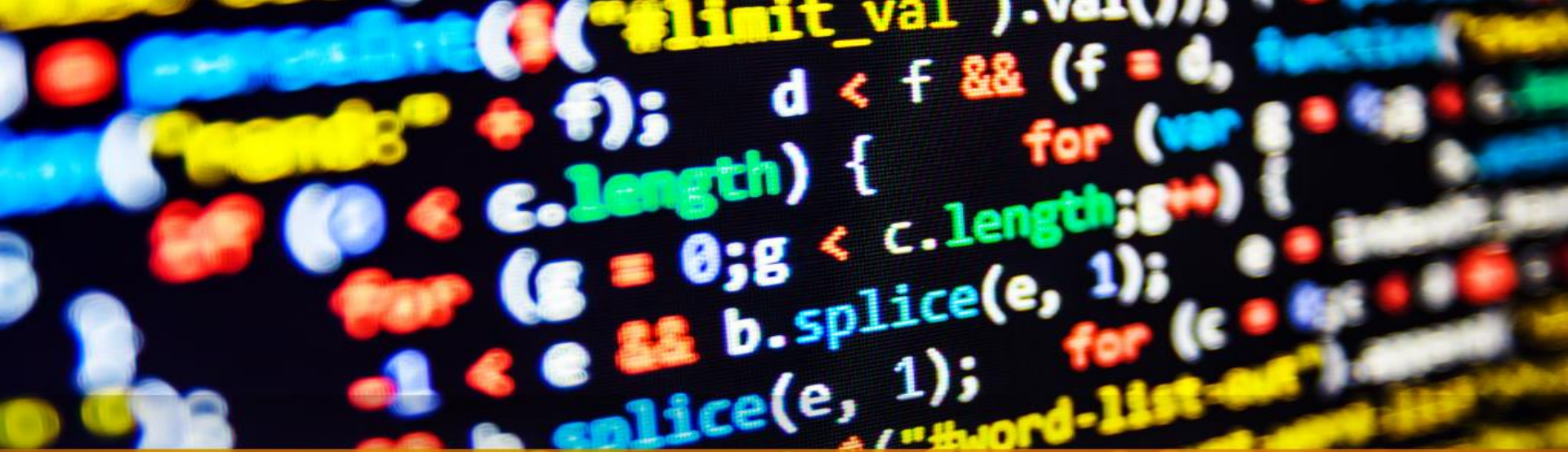
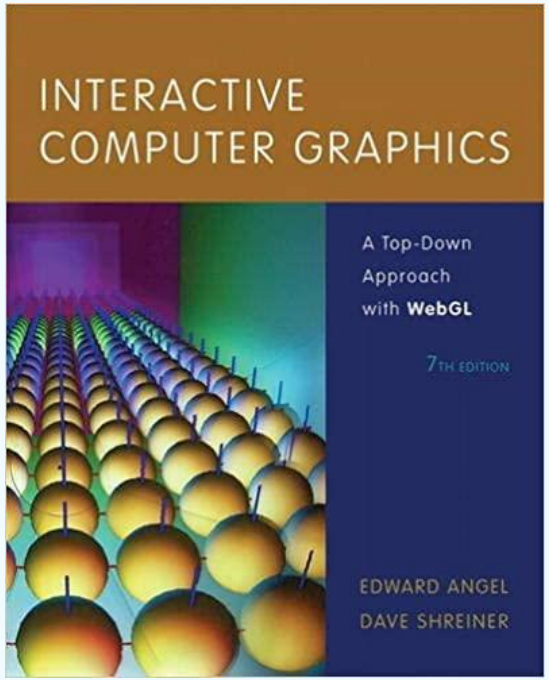


Lecture 00

About TGD2151

Prepared by Ban Kar Weng (William)

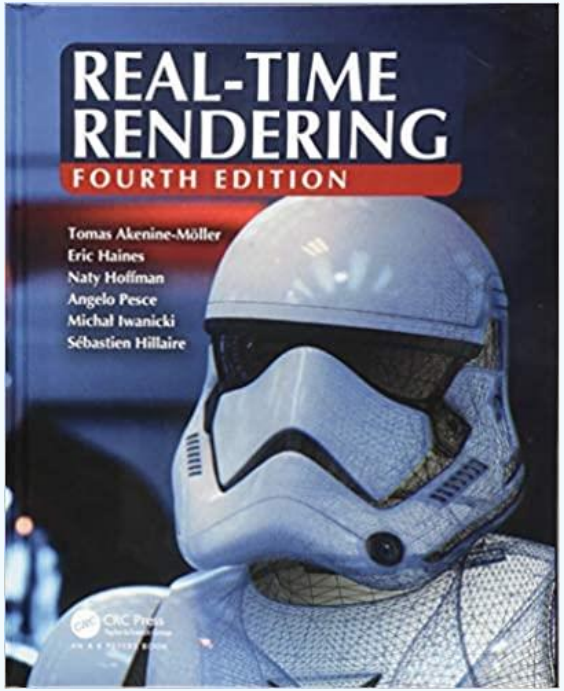
Textbook | Reference (Lecture)



Interactive Computer Graphics: A Top-Down Approach with WebGL (7th Ed)

Edward Angel, Dave Shreiner

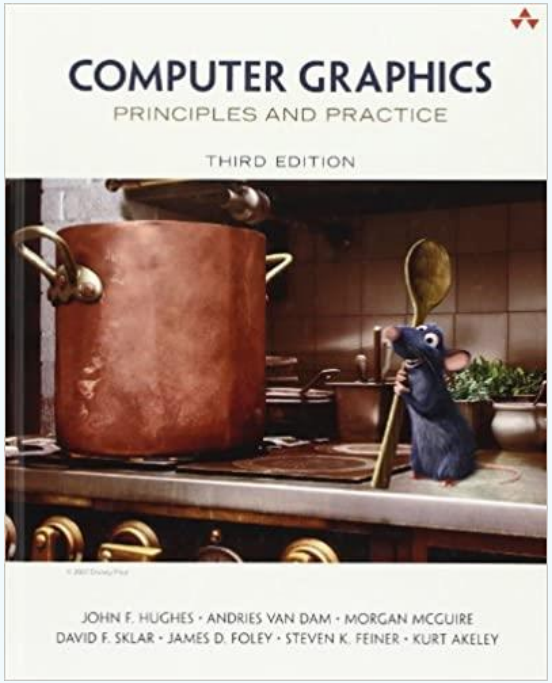
Textbook | Reference (Lecture)



Real-time Rendering (4th Edition)

Tomas Akenine-Moller, Eric Haines, Naty Hoffman

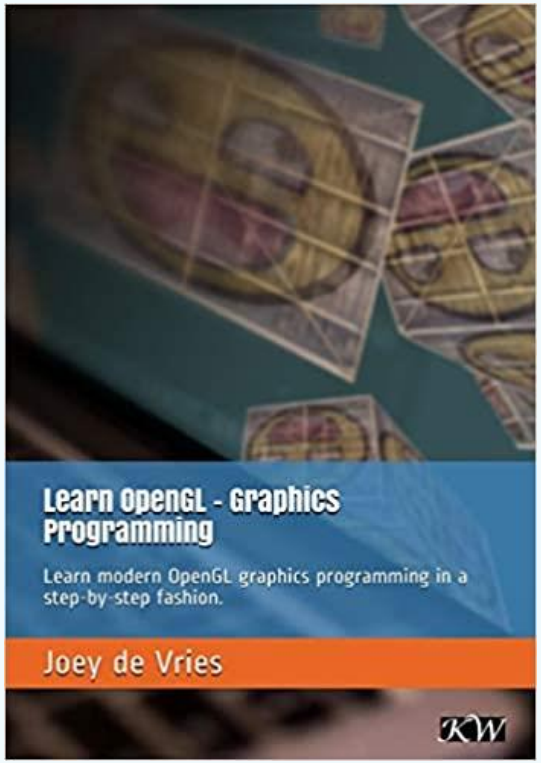
Textbook | Reference (Lecture)



Computer Graphics: Principles and Practice (3rd Edition)

John Hughes, Andries van Dam, Morgan McGuire, David Sklar, James Foley, Steven Feiner, Kurt Akeley

Textbook | Reference (Lab)

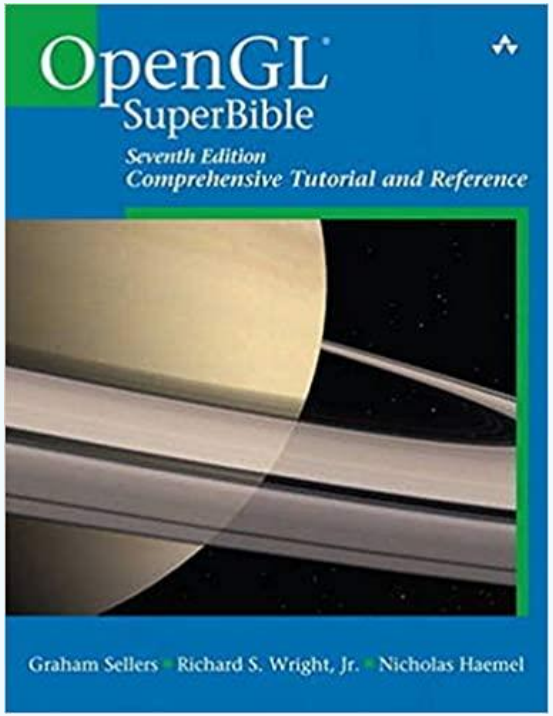


Learn OpenGL – Graphics Programming

Joey de Vries

- Download the e-book here.
- LearnOpenGL.com

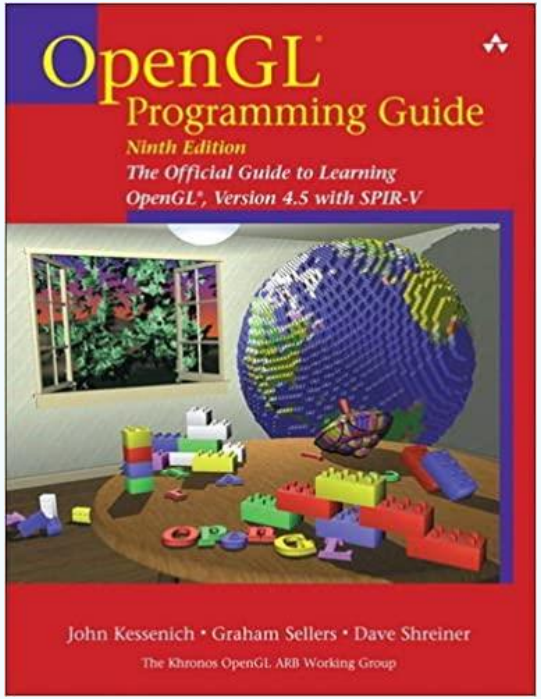
Textbook | Reference (Lab)



OpenGL Superbible: Comprehensive Tutorial and Reference (7th Edition)

Graham Sellers, Richard Wright Jr., Nicholas Haemel

Textbook | Reference (Lab)



OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V (9th Edition)

John Kessenich, Graham Sellers, Dave Shreiner

Coverage (Part 1)

Lec01 – Introduction to Computer Graphics Lec02 – Mathematical Fundamentals
Lec03 – Modelling and Transforms (Part 1) Lec04 – Modelling and Transforms (Part 2)
Lec05 – Viewing Lec06 – Visible Surface Determination
Lec07 – Lighting, Shading, and Texture Mapping (Part 1) Lec08 – Lighting, Shading, and Texture Mapping (Part 2)

Coverage (Part 2)

Lec09 – Lighting, Shading, and Texture Mapping (Part 3) Lec10 – Special Effects
Lec11 – Curves and Surfaces Lec12 – Advanced Topics
Lec13 – * Project Development and Submission Lec14 – * Project Presentation

Coursework Distribution

Assessment Component	Percentage
Quiz	10%
Assignment	30%
Test	20%
Project (Final Assessment)	40%

Course Policies

1. Collaboration Policy (Honour Code)

- **Acknowledge** the people you ascertain help from (other students, external parties)
- **Acknowledge** material taken from elsewhere
- **Acknowledge** source of code used

No acknowledgement or referencing is same as saying “I did everything myself” when you did not.

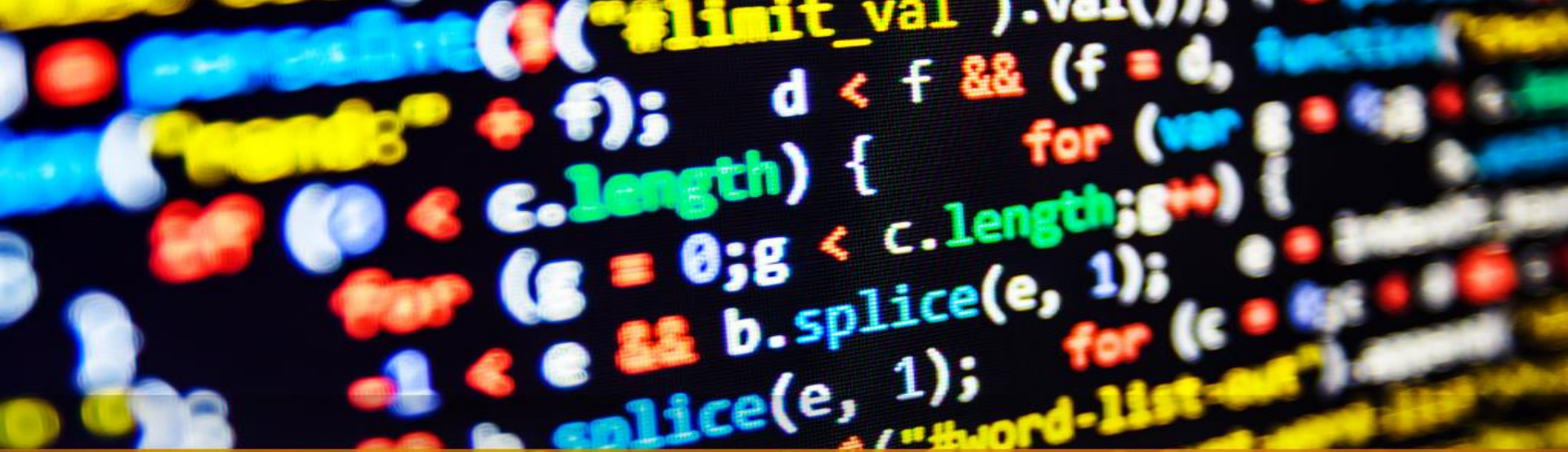
Course Policies

2. Late Policy

- Late submissions will be penalized **20% (of total marks of that coursework) per day late**.
- No submissions allowed after the **hard deadline** of that coursework.

3. Missed Presentations/Demos

- No replacements (marks considered deducted) unless for very valid reasons.



Lecture 01

Introduction to Computer Graphics

Prepared by Ban Kar Weng (William)

Overview of Computer Graphics

What is Computer Graphics?

- **Science** and **art** of communicating visually via a computer's display and its interaction devices.
- **Cross-disciplinary field**
  - Physics
  - Mathematics
  - Human perception
  - Human-computer interaction
  - Engineering
  - Graphics design
  - Art

Graphics Application

Games

All image are taken from the Internet. To take the address, right-click on the image and click the “Copy Hyperlink” context menu item.

Film and Entertainment

Animated Movies

(1)

1994

(2)

2019

Visual Effects in Movies

(3)

(1) https://www.youtube.com/watch?v=nbY_aP-alkw | (2) https://www.youtube.com/watch?v=0MxulhivCvI | (3) https://www.youtube.com/watch?v=6Aeko7mYAV4

Virtual Reality & Augmented Reality

Interactive experience of real-world environment enhanced by computer-generated perceptual information.

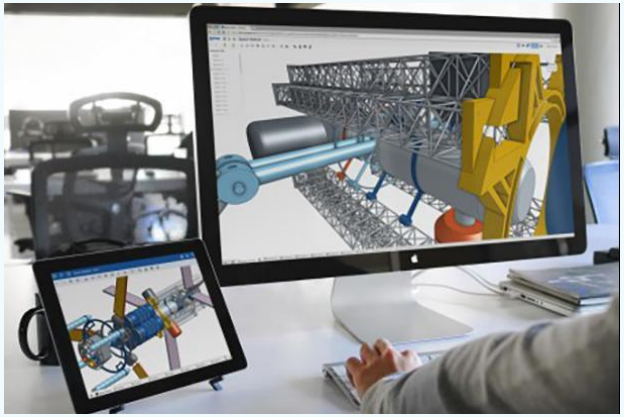
All image are taken from the Internet. To take the address, right-click on the image and click the “Copy Hyperlink” context menu item.

Scientific Visualization

- Process of **representing raw, scientific data** as images
- Helps **improve scientists' interpretations** of large datasets
- **Gain insights** that may be overlooked by statistical methods alone.

https://www.youtube.com/watch?v=BeQtJAqs5wc&t=12s

CAD



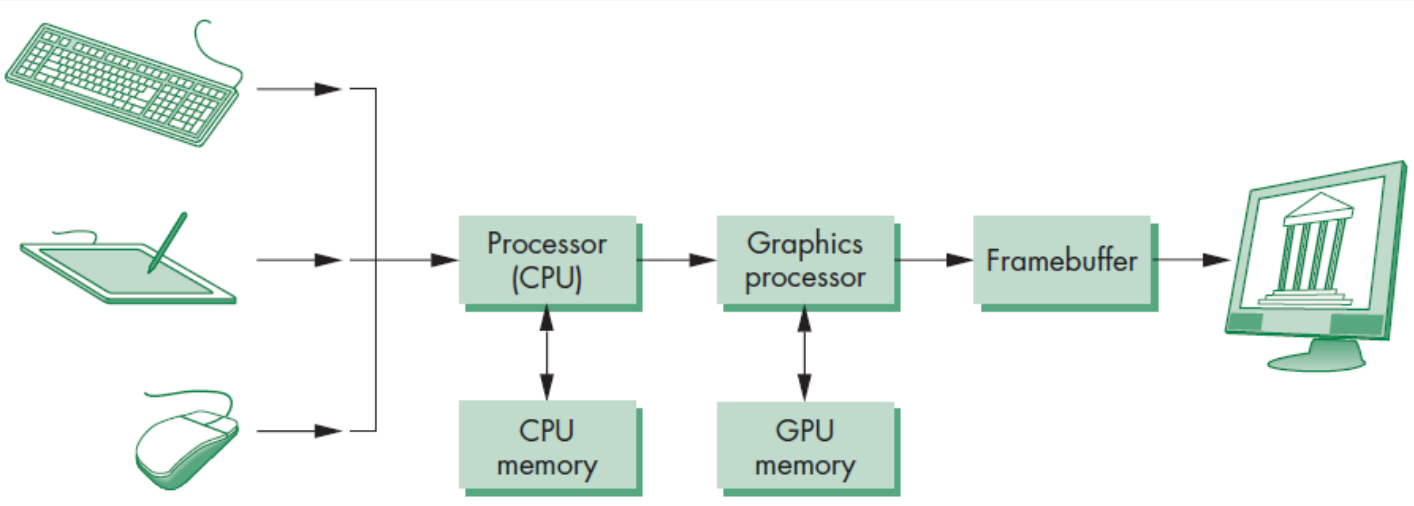
Computer-aided design (CAD)

- Increase designer’s productivity and improve quality of design.
- Automotive, shipbuilding, aerospace, architectural design

All image are taken from the Internet. To take the address, right-click on the image and click the “Copy Hyperlink” context menu item.

Graphics System

Graphics System



Six Major Elements of a Graphics System

1. Input Devices
2. Central Processing Unit
3. Graphics Processing Unit
4. Memory
5. Framebuffer
6. Output Devices

Image taken from Figure 1.1 of Interactive Computer Graphics – A Top Down Approach using WebGL (7th Edition)

Input Devices

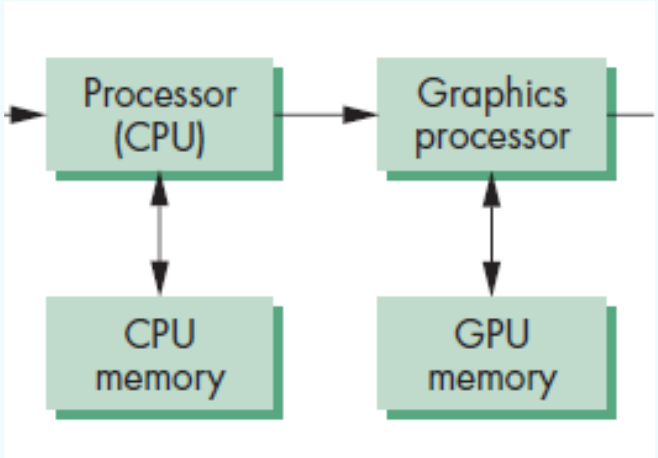
Common input devices

- Mouse, joystick, data tablet.
- Often called **pointing devices**, as they allow a user to indicate a particular location on the display.
- Equipped with button(s) to provide signals to the processor.

Multidimensional input devices

- Many degrees of freedom.
- E.g. Spaceball.

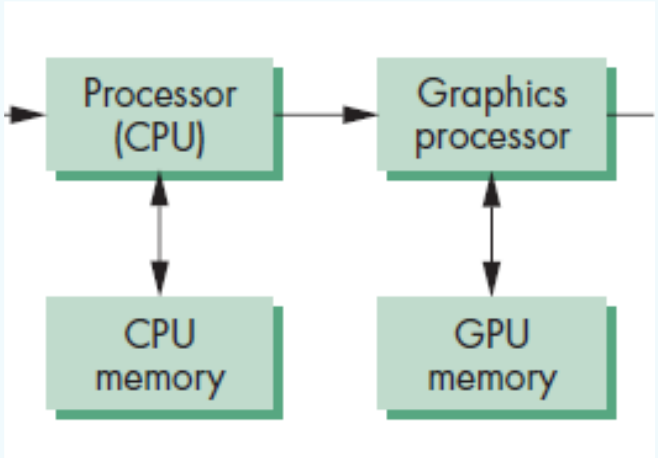
CPU and GPU



CPU

- In early systems, CPU perform both **normal processing** and **graphical processing**.
- **Rasterization** or **Scan conversion** = given geometries (e.g. lines, polygons, etc), determine which pixels the geometries appear in the framebuffer.
- Framebuffer was part of RAM.

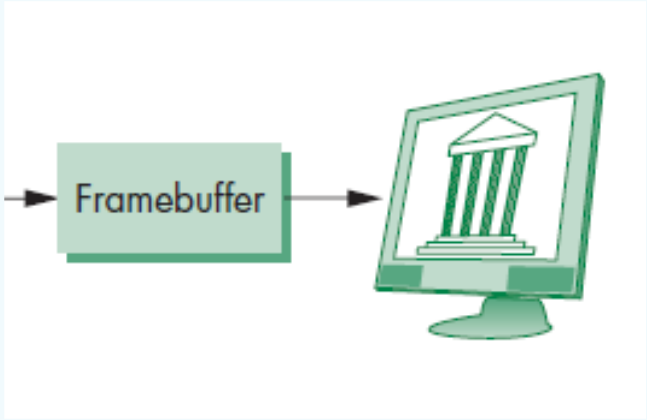
CPU and GPU



GPU

- Nowadays, graphical processing now performed by GPU.
- GPU can be located on the motherboard (integrated graphics) or the graphics card (dedicated graphics).
- Framebuffer is on the same circuit board as GPU.
- High degree of parallelism compared to CPU.

Framebuffer



Framebuffer

- Part of the memory containing all the pixels in a complete video frame.
- Comprises multiple buffers (e.g. colour buffer, depth buffer)

Resolution

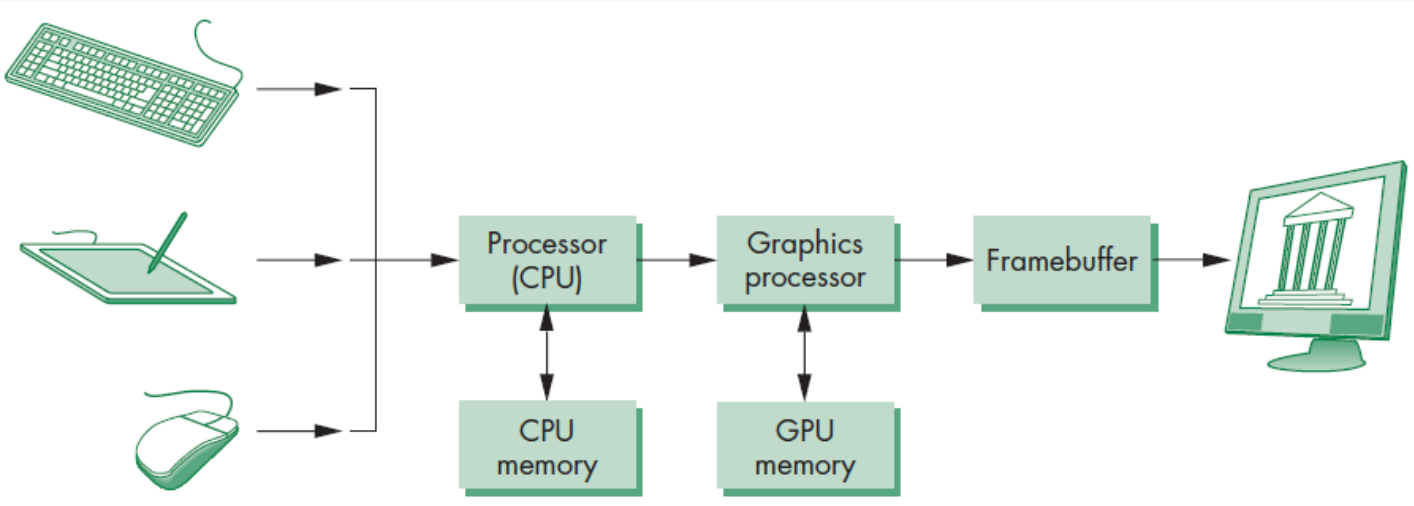
- The number of pixels in the framebuffer.

Depth

- Number of bits used for each pixel.
- **Full-colour system** : 24 bits per pixel.
- **High dynamic range system** : Use 12 or more bits in each pixel.

Graphics API

Recall that a Graphics System is ……

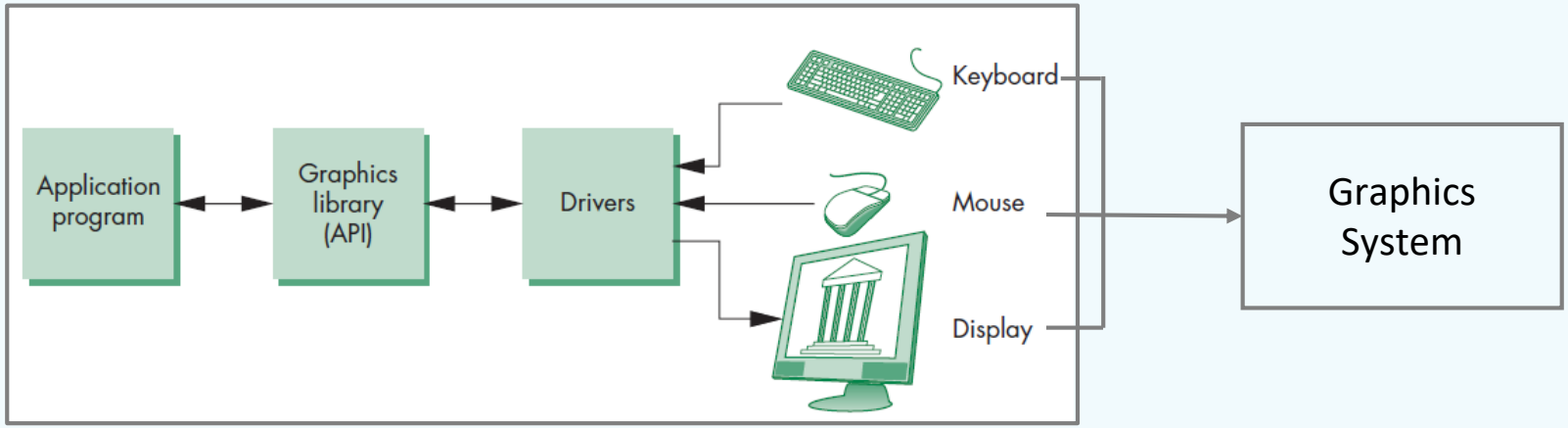


Six Major Elements of a Graphics System

1. Input Devices
2. Central Processing Unit
3. Graphics Processing Unit
4. Memory
5. Framebuffer
6. Output Devices

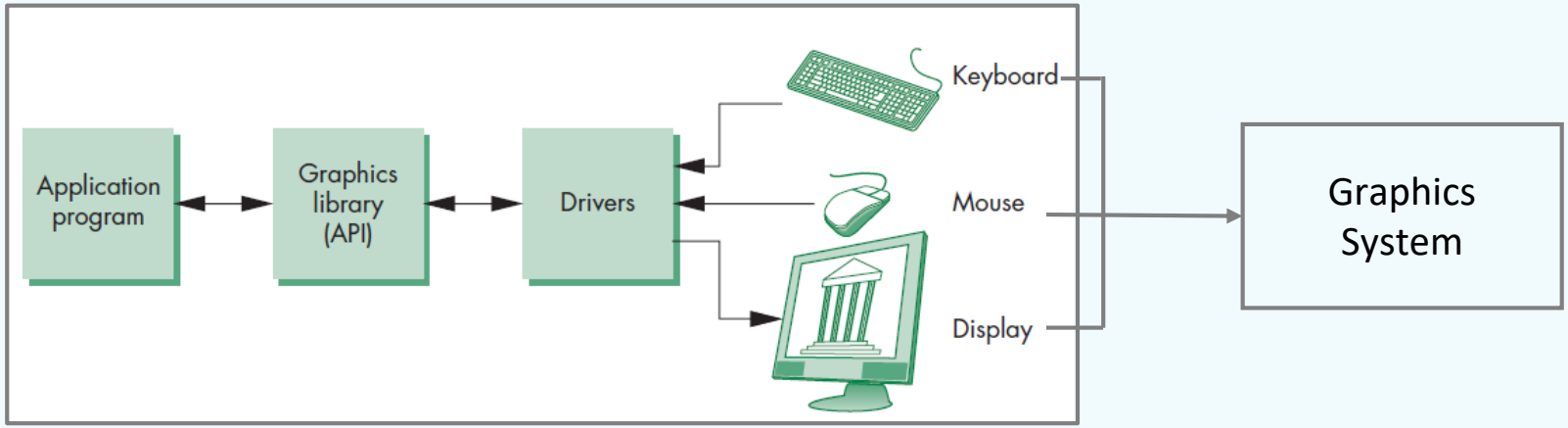
Image taken from Figure 1.1 of Interactive Computer Graphics – A Top Down Approach using WebGL (7th Edition)

Graphics API



- API = Application Programming Interface
- **API** specifies **a set of functions** that allows an application program to interact with a graphics system.
- Hides the actual implementation details of hardware and software.

Graphics API



- **Drivers** interpret the output data of graphics API and convert these data to a form that is understood by the particular hardware.
- Using the API, applications can be used with different hardware and software platforms.

Examples of 3D Graphics API

Cross Platform, High Level

OpenGL

- Cross-language, cross-platform API for rendering 2D and 3D graphics.

OpenGL ES

- OpenGL for Embedded and Mobile System.
- Suitable for low-power devices.

WebGL

- OpenGL for the Web
- Based on OpenGL ES, exposed to ECMAScript via the HTML5 Canvas

Cross Platform, Low Level

~

+

Vulkan

- Cross-platform 3D graphics + computing API.
- Targets high-performance real-time 3D graphics applications (e.g. video games, interactive media)
- “Next generation OpenGL Initiative”

Advantages over OpenGL

- Single API for both desktop and mobile devices.
- Designed to better utilize multi-core CPU.
- Unified management of compute kernel and graphics shaders.

More graphics API standards by Khronos Group.

Vendor Specific, Low & High Level

DirectX

- A collection of APIs for handling multimedia-related tasks.
- Works only on Microsoft platforms (i.e. Windows)

Direct3D (version 11 and below)

- High level 3D API.
- Similar to OpenGL

Direct3D (version 12)

- Low level 3D API
- Similar to Vulkan

Q & A

Acknowledgement

- This presentation has been designed using resources from PoweredTemplate.com