Xi'an Jiaotong-Liverpool University
西交利物浦大学

## CPT205 Computer Graphics

# General Introduction Hardware and Software

**Lecture 01**
**2024-25**

**Yong Yue and Nan Xiang**

# Teaching Plan

| Week (c/m) | Lecture | Topic | CW | Lecturer |
|---|---|---|---|---|
| 01 (24.09.09) | Lecture 01<br>Lecture 02 | Introduction and hardware/software<br>Mathematics for computer graphics | | Yong Yue |
| 02 (24.09.16) | Lecture 03 | Geometric primitives | | Nan Xiang |
| 03 (24.09.23) | Lecture 04 | Geometric transformations | CW1 out | Nan Xiang |
| 04 (24.10.07) | Lecture 05 | Viewing and projection | | Nan Xiang |
| 05 (24.10.14) | Lecture 06 | 3D modelling | | Yong Yue |
| 06 (24.10.21) | Lecture 07 | Parametric curves and surfaces | | Yong Yue |
| 07 (24.10.28) | | Reading week | CW1 due | |
| 08 (24.11.04) | Lecture 08 | Hierarchical modelling | CW2 out | Yong Yue |
| 09 (24.11.11) | Lecture 09 | Lighting and materials | | Nan Xiang |
| 10 (24.11.18) | Lecture 10 | Texture mapping | | Nan Xiang |
| 11 (24.11.25) | Lecture 11 | Clipping | | Yong Yue |
| 12 (24.12.02) | Lecture 12 | Hidden surface removal | CW2 due | Yong Yue |
| 13 (24.12.09) | Revision | Summary and highlights of topics covered / Past exam paper | | Nan Xiang / Yong Yue |

# Delivery Schedule

## ➢ Delivery Schedule

**Lecture:** *Monday 09:00-11:00 (EB138)*

**Lab:** *D1/1: Tuesday 09:00-11:00 (SD554)*
*D1/2: Tuesday 09:00-11:00 (SD546)*
*D1/3: Tuesday 11:00-13:00 (SD554)*

*Please make sure you attend the lab session allocated to you!*

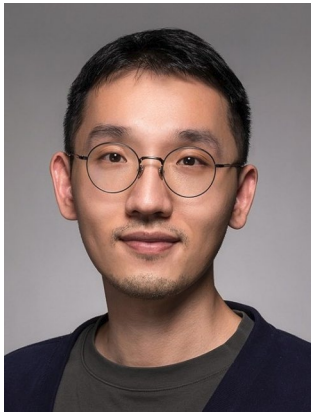
# Module Team (Lecturers)

## ➢ Module Lecturers and Contact Details


Name: *Yong Yue*
Email: *yong.yue@xjtlu.edu.cn*
Office telephone: *(0512) 8816 1503*
Room number: *SD523*
Office hours: *17:00-18:00 Monday*
*16:00-17:00 Tuesday*
Preferred means of contact: *email*



Name: *Nan Xiang*
Email: *nan.xiang@xjtlu.edu.cn*
Office telephone: *(0512) 8916 5342*
Room number: *SD441*
Office hours: *13:00-14:00 Monday*
*13:00-14:00 Thursday*
Preferred means of contact: *email*

# Module Team (TAs)

## ➢ Teaching Assistance (TAs)

Qizhong Gao (qizhong.gao22@student.xjtlu.edu.cn)
Xiaoyu Huang (xiaoyu.huang20@student.xjtlu.edu.cn)
Yijie Chu (yijie.chu16@student.xjtlu.edu.cn)
Yuhao Jin (yuhao.jin22@student.xjtlu.edu.cn)
Zihan Ye (zihan.ye22@student.xjtlu.edu.cn)
Zitian Peng (zitian.peng20@student.xjtlu.edu.cn)

Depending on the demand for support, students may be divided into groups to receive support from dedicated TAs.

# What is Computer graphics?

'Computer Graphics' is concerned with all aspects of producing pictures or images using a computer. There are three closely related meanings, each representing a different perspective on the same thing.

- the images that you see on the computer screen
- the computer code that is used to create the images
- a mathematical model of the real-world (which is sometimes called the virtual world)

When working at the most advanced levels of computer graphics, the computer graphics specialist will

- create a virtual world
- implement the virtual world in computer code
- run the code to see life-like images on the computer screen

# What are the application areas?

The ‘application areas’ are the different types of computer application that can be produced using computer graphics technology. They are also the technical areas within which a student may seek to work when he or she leaves university with a relevant degree.

Example application areas:

- Display of information
- Design
- Simulation/modelling and animation
- User interfaces
- Virtual reality

# Why learn computer graphics?

There are numerous good reasons. These include:

- the application areas are exciting;
- the subject matter is intellectually stimulating and the knowledge that you will get on the course is relatively rare;
- computer graphics is a major facet of computer science (e.g. try to think of a computer program that does not involve some kind of graphics);
- the concepts associated with computer graphics are time independent (i.e. what you learn now will still be valid in the future, unlike some programming languages which disappear with passing time);
- computer programming is the type of job in computing that is in great demand, and the computer graphics course involves a good deal of “hands on” programming.

# Basic concepts of computer graphics

- Graphics hardware and software
- Fundamental mathematics
- Objects / geometric primitives – point, curve, surface and solids
- Modelling and representation schemes 建模
- Geometric transformations
- Viewing and projections 视图和投影
- Clipping 裁剪(优化渲染)
- Removal of hidden curves and surfaces →背面剔除技术
- Lighting and materials 光照和材质(影响物体光泽之金属)
- Texture mapping 纹理(将二维图像贴到三维表面)
- Animation 动画
- Programming and applications

# This module

Introduces a wide range of topics in computer graphics and its applications, providing you with both fundamental theory and hands-on experience through lab-based practice and assessment. It follows a standard textbook with additional materials used for contemporary developments and applications.

In terms of learning outcomes, you will be able to:

- demonstrate a good understanding of topics and applications in computer graphics covered in the module;
- demonstrate an in-depth knowledge of geometric creation and transformation, projection, clipping and hidden geometry removal, lighting and materials, and texture mapping;
- apply relevant techniques / algorithms covered in the module to specific scenarios;
- write programming code in conjunction with a popular graphics platform (e.g. OpenGL).

# Delivery and assessment

**Delivery**: You will have a two-hour formal lecture followed by a two-hour lab weekly. It assumes knowledge of matrices and vectors and previous experience of computer programming in a high-level procedural language (e.g. Java or C). Sample programs will be provided during the lab sessions.

**Assessment**:

1) Assessment 1 (2D project – 15%): Weeks 3 to 7
2) Assessment 2 (3D project – 15%): Weeks 8 to 12
3) Assessment 3 (Final exam – 70%): Early January 2025
4) Resit Assessment (Exam – 100%): Late July / early August 2025

No resit is available for assessments 1 and 2. If a student does not obtain an overall grade of 40% or higher for Assessments 1-3, she/he will be required to take the resit exam which will have a weighting of 100% for the module (regardless of grades of assessments 1 and 2).

***Important***: *Plagiarism is a serious academic offence and will not be tolerated. Copying from other sources without appropriate acknowledgement may result in plagiarism! If in doubt, consult relevant members of academic staff*

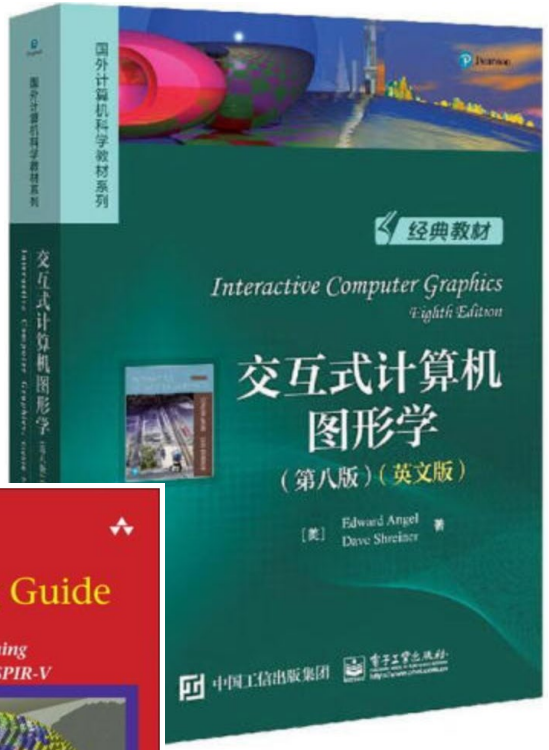
# Marking scheme of coursework

| Category | Requirements (each category builds on the requirements for the preceding category) |
|---|---|
| **First Class (≥70%)** | Overall outstanding work. Very neat program implements effectively all the graphics techniques covered.<br>Artefact produced with realistic / real-life content and visual effect.<br>Well-structured and concise written report providing all the required information. |
| **Second Upper (60 to 69%)** | Comprehensive program that utilises effectively the full range of the graphics techniques covered to date. Good commenting and layout of the program.<br>An impressive artefact produced with a good range of features achieved by calling appropriate OpenGL functions.<br>A comprehensive and clear report containing all required information within the page limit. |
| **Second Lower (50 to 59%)** | Substantial working program implements a good range of graphics techniques.<br>Nice layout and objects in the artefact.<br>Written report contains all the information of the features and functions of the program including some screenshots. |
| **Third (40 to 49%)** | Working program that generates a recognisable artefact with some objects and a limited range of the graphics techniques utilised.<br>Written report provides a good overview and describes all the basic information for the work completed. |
| **Fail (0 to 39%)** | Some code produced attempts to the use of some graphics techniques covered in the module.<br>No or very limited artefact produced.<br>Written report covers very limited number of the items required in the assignment brief, acknowledging properly sources used if any. |
| **Non-** | A mark of 0 will be awarded |

# Textbooks

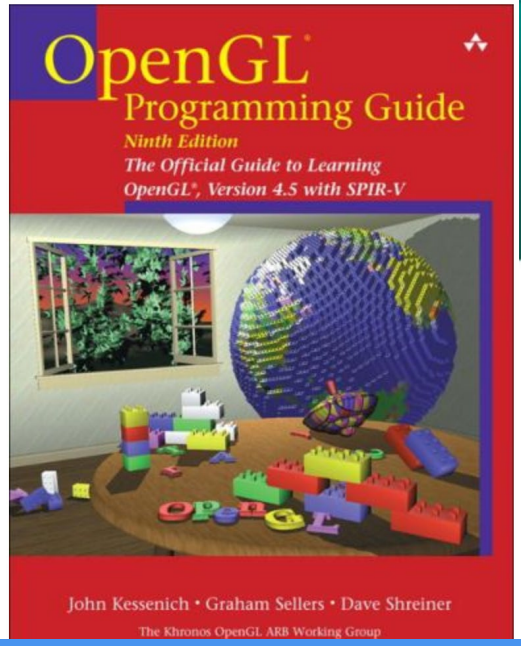
- Optional

  Edward Angel and Dave Shreiner, *Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL*, Eighth Edition, ISBN: 978-7121393983, 1 July 2017, Electronic Industry Press, China.




- Reference

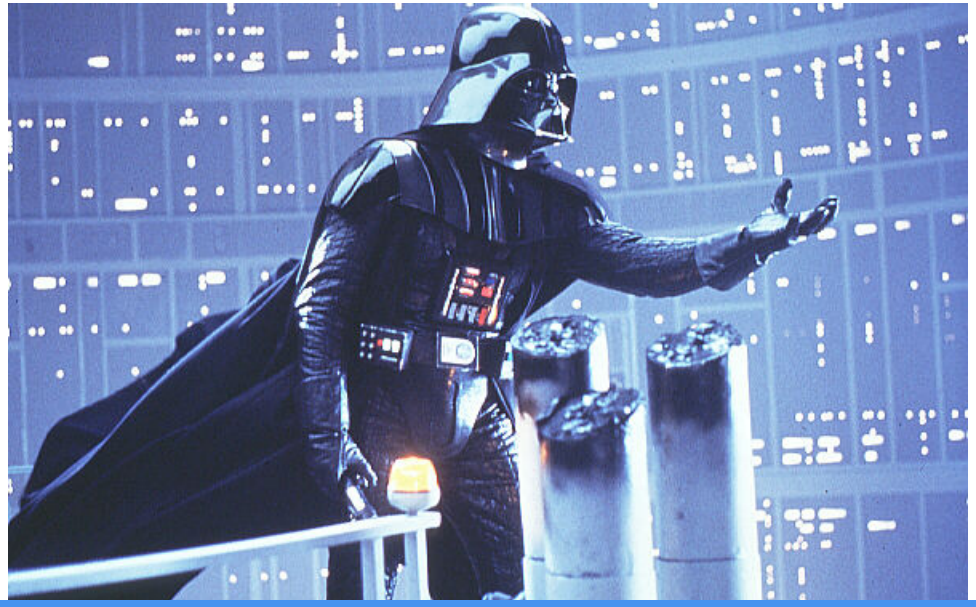
  John M. Kessenich, Graham Sellers and Dave Shreiner, *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, Versions 4.5 SPIR-V, ISBN: 978-0134495491, 8 July 2016, Addison Wesley.



# What's next?

**To do well on the course:**

- try to enjoy yourself
- attend all lecture and practical sessions
- work consistently



# Hardware and Software

- Graphics Hardware
  - Input, Processing and Output Devices
  - Framebuffers 帧缓冲器
  - Pixels and Screen Resolution

- Graphics Software
  - Techniques (Algorithms, Procedures)
  - Programming Library / API (OpenGL, JOGL and so on)
  - Not our focus: High level Interactive Systems (Maya, Studio Max, Unity, AutoCAD and so on)

# Think about

- Basic
  - How will you define graphics hardware?
  - Does graphics hardware involve input, processing and output?
  - When you buy a computer how will you specify the graphics requirement?
- Intermediate
  - What is the graphics card on your home desktop computer?
  - How many lines/sec or triangles/sec can you draw on your computer?
  - Screen resolution: How many pixels do you see on your screen?
- Advanced
  - How do you represent a scene (e.g. a house) graphically in your computer?
  - How do you display the house on the computer screen (framebuffer)?
  - How does the graphics hardware transform your scene to the display?

# Graphics devices

- Input Devices
- Processing Devices
- Output Devices

How do we link the three?

# Output devices

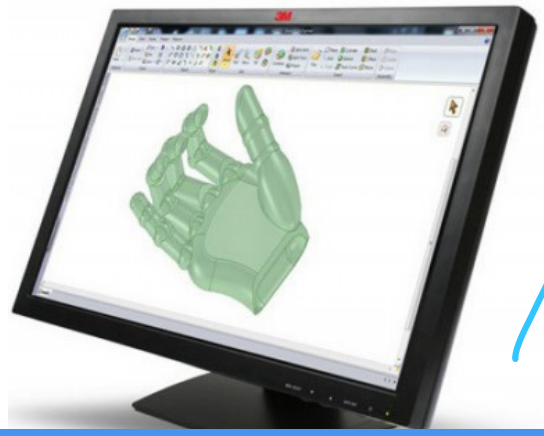
Cathode Ray Tube 阴极射线管显示器

CRT

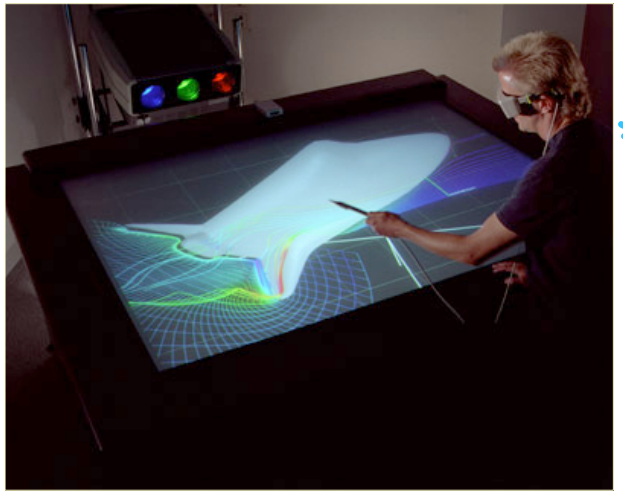
Laptop LCD

Liquid Crystal Display

液晶显示器



LCD



显示3D模型

投影桌面

# Graphic display in virtual reality

头戴式显示器

- Head-Mounted Displays (HMDs)
  - The display and a position tracker are attached to the user's head

跟踪显示器

- Head-Tracked Displays (HTDs)
  - Display is stationary, tracker tracks the user's head relative to the display.
  - Example: CAVE, Workbench, Stereo monitor

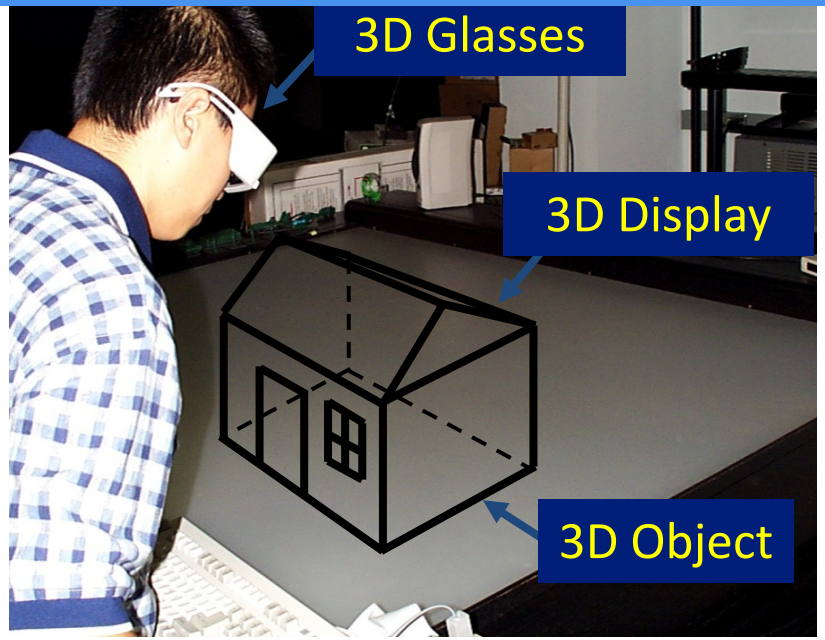
有间大小的投影系统，可穿过观察墙型，地面感受虚拟场景

CCave Automatic Virtual Environment

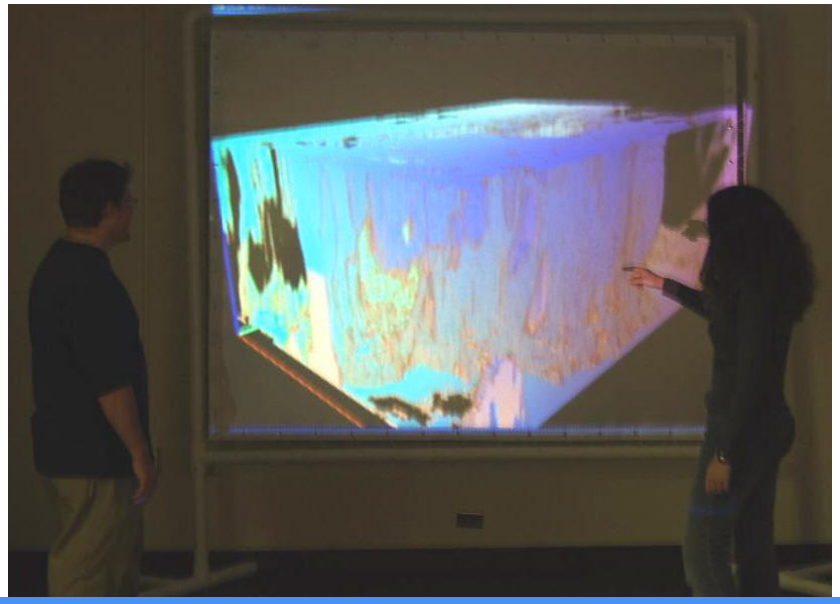
立体的显示系统三维可视化



提供立体感显示的实现，改善教学







# Graphic processing unit (GPU)

- Graphics Cards / GPUs sit inside

# Input devices

- Enables graphical interaction

# Input devices

When queried, locator devices return a position and/or orientation

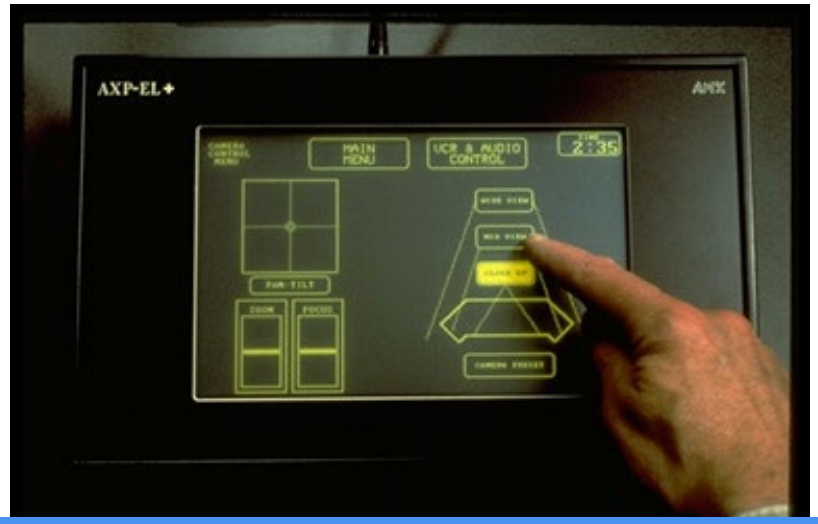
- Tablets
- Virtual Reality Trackers
  - Data Gloves
  - Digitisers

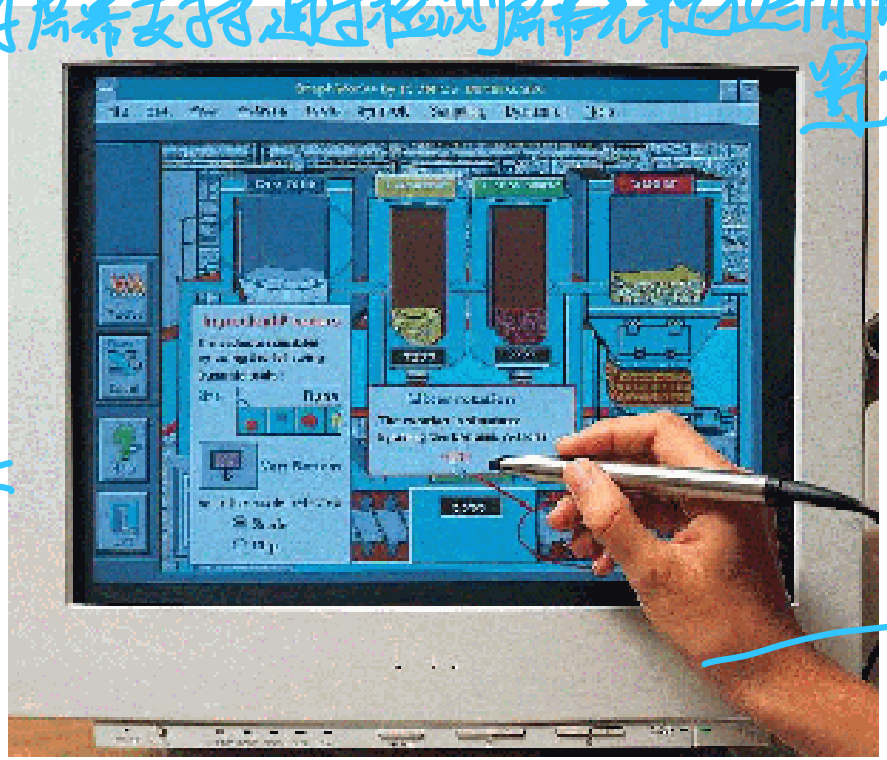
# Input devices

适用于：

- Light Pens 光笔（需要走门的屏幕支持通过检测屏幕光来定位指针位置）
- Voice Systems
- Touch Panels 触摸屏
- Camera/Vision based

Which is best? 取决于应用场景



# Input devices for games

索尼通过物理按键，加上运动传感器，PlayStation Eye 摄像头来精确输入


I MOVE YOU!

VS.


Wii

Wii MOVE YOU!

VS.

微软开发体感控制器，无需手持设备，支持手势，能够识别和面部识别，不光线不足时动作捕捉受限


KINECT TO ME!

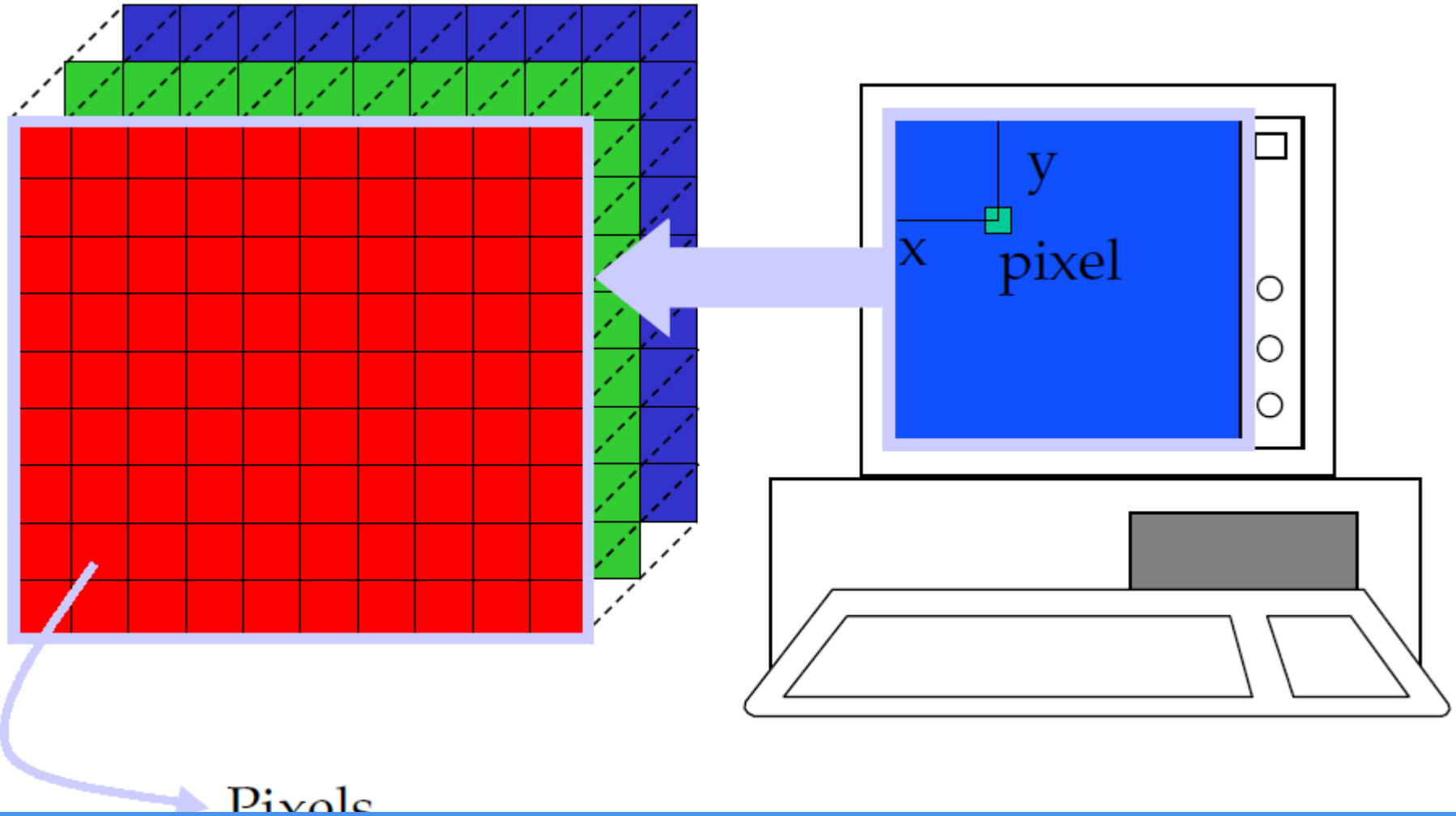
任天堂以体感控制，传统按钮配合加速度计红外传感器捕捉输入，支持各种配件（动向盘，枪托）但是动作捕捉精度低

# Framebuffer 帧缓冲器

**Framebuffer** – A block of memory, dedicated to graphics output, that holds the content of what will be displayed.

**Pixel** - an element of the framebuffer.

# Framebuffer



# Framebuffer

pixels = 1920 × 1080

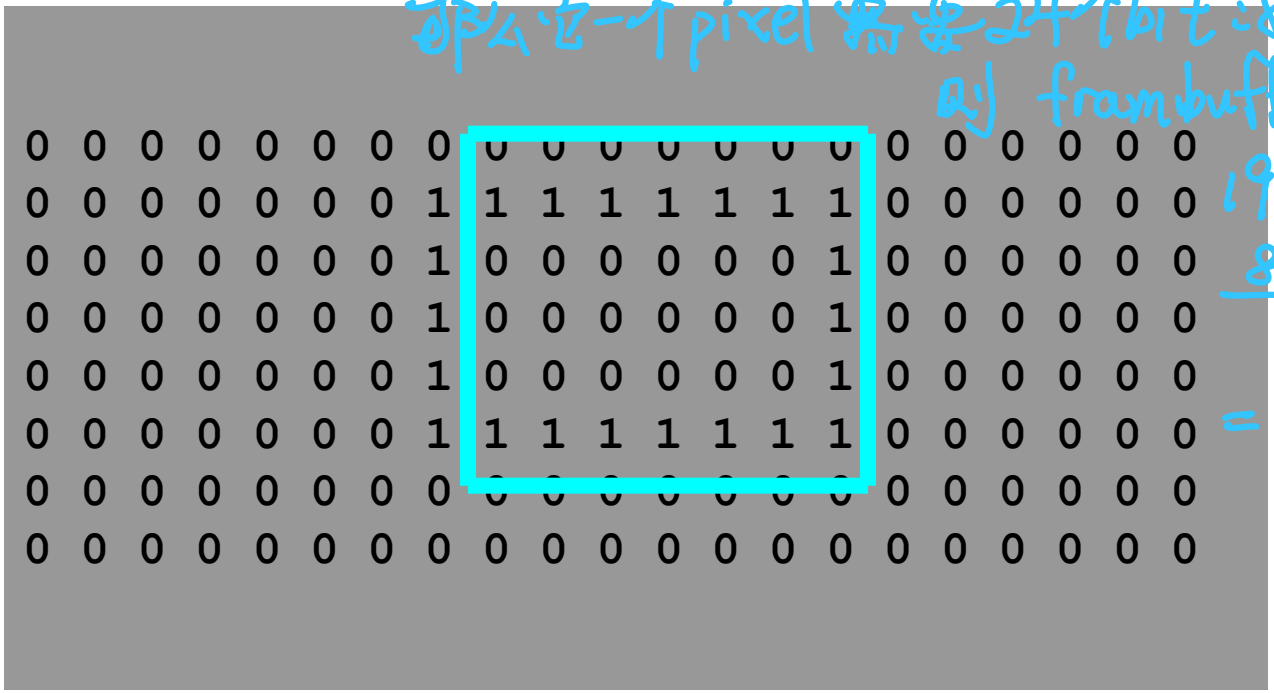
对于常见 1920 × 1080 显示器：若它的色深是 8 bit

那么它一个 pixel 需要 24 个 bit：8 红，8 绿，8 蓝

则 framebuffer =

$$\frac{1920 \times 1080 \times (8+8+8)}{8}$$

$$= 1920 \times 1080 \times 3 \text{ bytes}$$




**Questions:** How many pixels are there?

What is the largest image you can display?

How big is the framebuffer?

How much memory do we need to allocate for the framebuffer?

# Framebuffer in memory

- If we want a framebuffer of 640 pixels by 480 pixles, we should allocate:

  framebuffer = 640*480 bits

- How many bits should we allocate?

  Q: What do more bits get you?

  A: More values to be stored at each pixel.

- Then, why would you want to store something other than a 1 or 0?

# Framebuffer bit depths

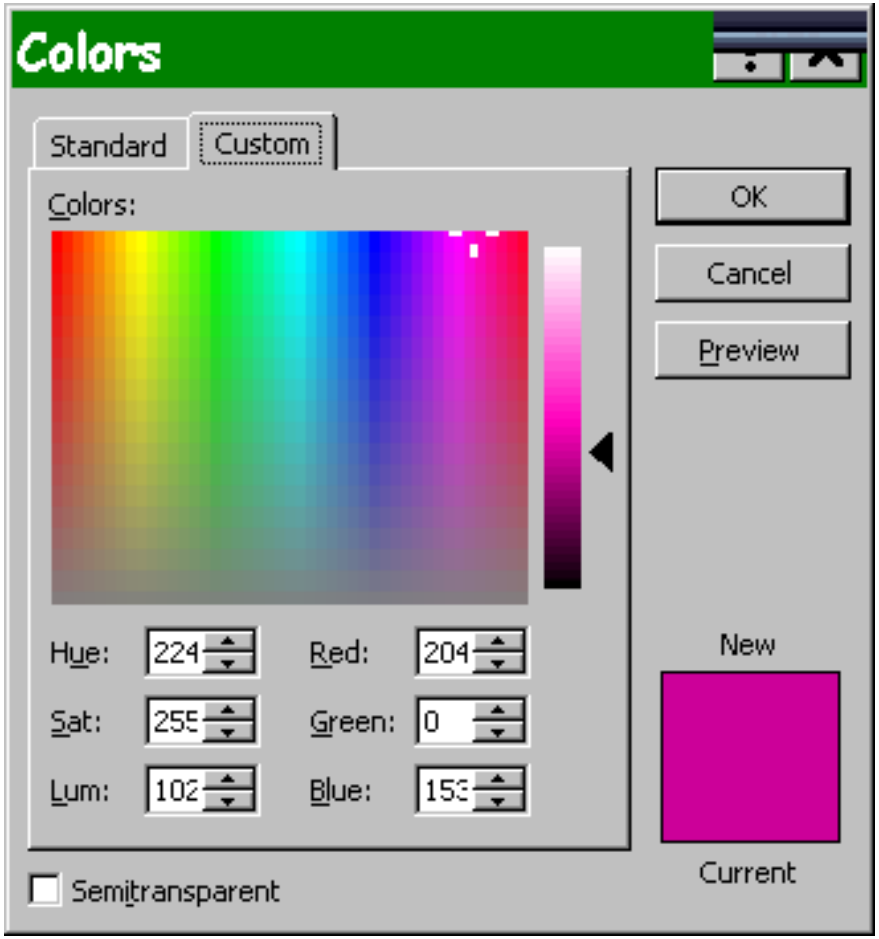
- How many colours does 1 bit get you?
- How many colours do 8 bits get you?
  - Monochrome systems use this (green/grey scale)
- What bit depth would you want for your framebuffer?

**bit depth -** number of bits allocated per pixel in a buffer.

# Framebuffer bit depths

- Remember, we are asking “how much memory we allocate to store the colour at each pixel?”
- Common answer:
  - 32 bits RGBA

这个被称为“真彩色”，8红，8绿，8蓝，8透明度。



# Framebuffer bit depths

- 32 bits per pixel (true colour)
  - 8 bits for red, green, blue and alpha
  - potential for 256 reds, greens and blues
  - total colours: 16,777,216 (more than the eye can distinguish) $256 \times 256 \times 256$
- Let's look at Display Control Panel

# Data type refresher 复习

bit - a 0 or 1. Can represent 2 unique values

byte - 8 bits or 256 values

word - 32 bits or 4,294,967,296 values

int - 32 bits

float - 32 bits

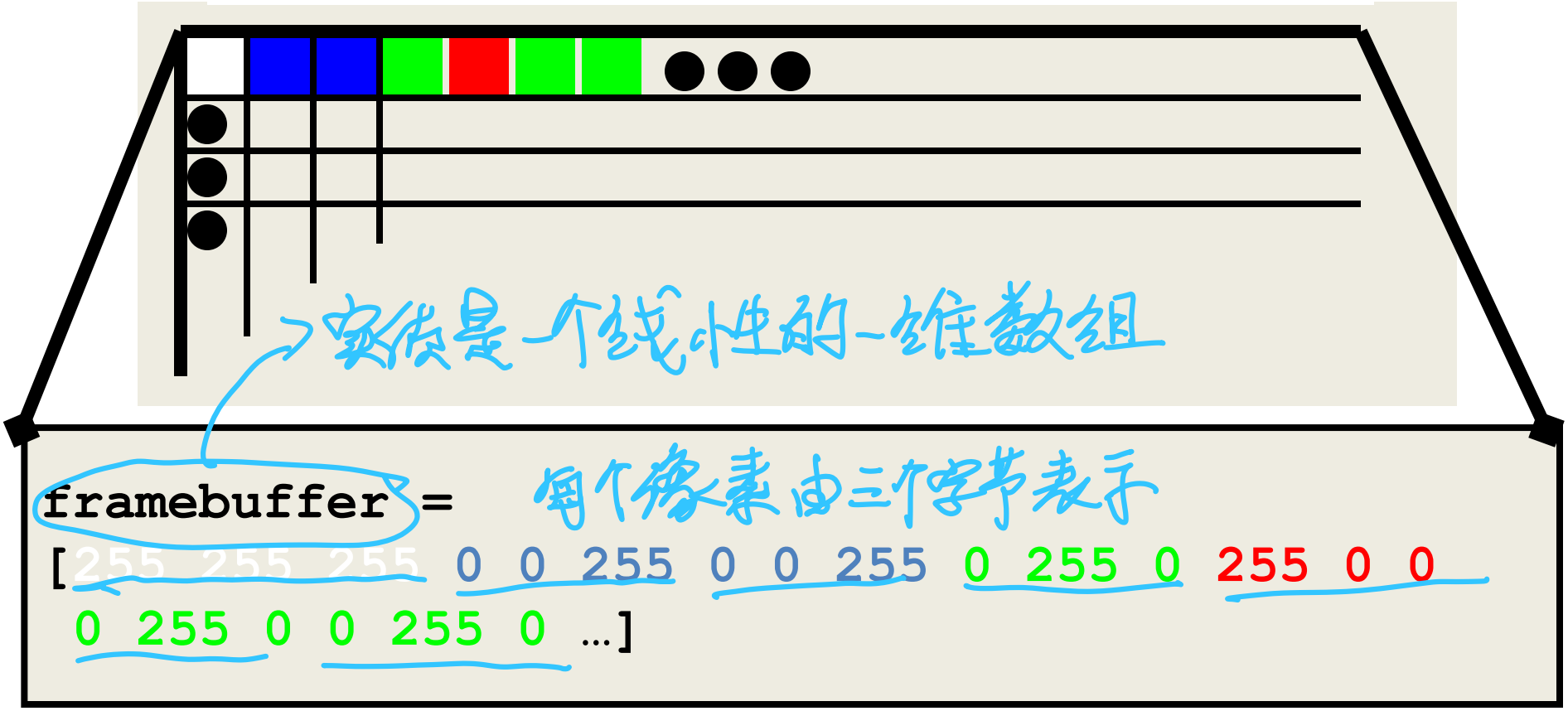
double - 64 bits

unsigned byte - 8 bits

无符号字节

# Framebuffer bit depths

unsigned byte framebuffer [640*480*3]; B



# Graphic card memory

- How much memory is on our graphic card?
  - 640 * 480 * 32 bits = 1,228,800 bytes
  - 1024 * 768 * 32 bits = 3,145,728 bytes
  - 1600 * 1200 * 32 bits = 7,680,000 bytes

- How much memory is on your graphics card?

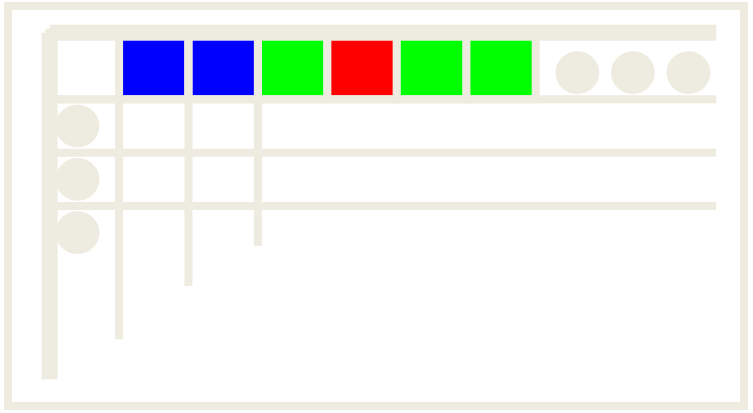
2560 x 1600 x 32 bits

# Framebuffer -> monitor

帧缓冲区数据会被转换成模拟信号传送给显示器

- The values in the framebuffer are converted from a digital (1s and 0s representation, the bits) to an analog signal that goes out to the monitor.
- This is done automatically (not controlled by your code), and the conversion can be done while writing to the framebuffer.

→数据转换可以在数据写入缓冲区时同时完成



# Image quality issues

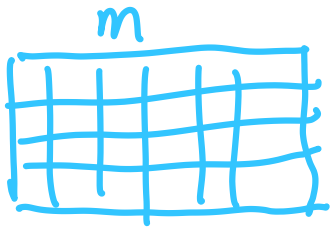
影响图像质量因素

- Screen resolution 屏幕分辨率
- Colour 颜色(色准)
- Refresh rate 刷新率
- Brightness 亮度
- Contrast 对比度
- Sensitivity of display to viewing angle (可视角度)

# Pixels

- Pixel - The most basic addressable image element in a screen 屏幕上最基本可寻址图像单元
  - CRT - Colour triad (RGB phosphor dots)
  - LCD - Single colour element (液晶显示器，每个像素代表一个子图)
- Screen Resolution - measure of number of pixels on a screen (m by n)
  - m - Horizontal screen resolution
  - n - Vertical screen resolution

阴极射线管显示器




m x n 分辨率表示

# Video formats

以下均为计算器显示标准

- NTSC - 525x480, 30f/s, interlaced
- PAL - 625x480, 25f/s, interlaced
- VGA - 640x480, 60f/s, non-interlaced
- SVGA - 800x600, 60f/s non-interlaced
- RGB - 3 independent video signals and synchronization signal, vary in resolution and refresh rate
- Interlaced - scan every other line at a time, or scan odd and even lines alternatively; the even scan lines are drawn and then the odd scan lines are drawn on the screen to make up one video frame.

隔行扫描

非隔行扫描

3个独立不见监视信号+1个同步信号

分辨率和刷新率会不相同

交替扫描

先绘制偶数线 后 奇数线组成视频帧

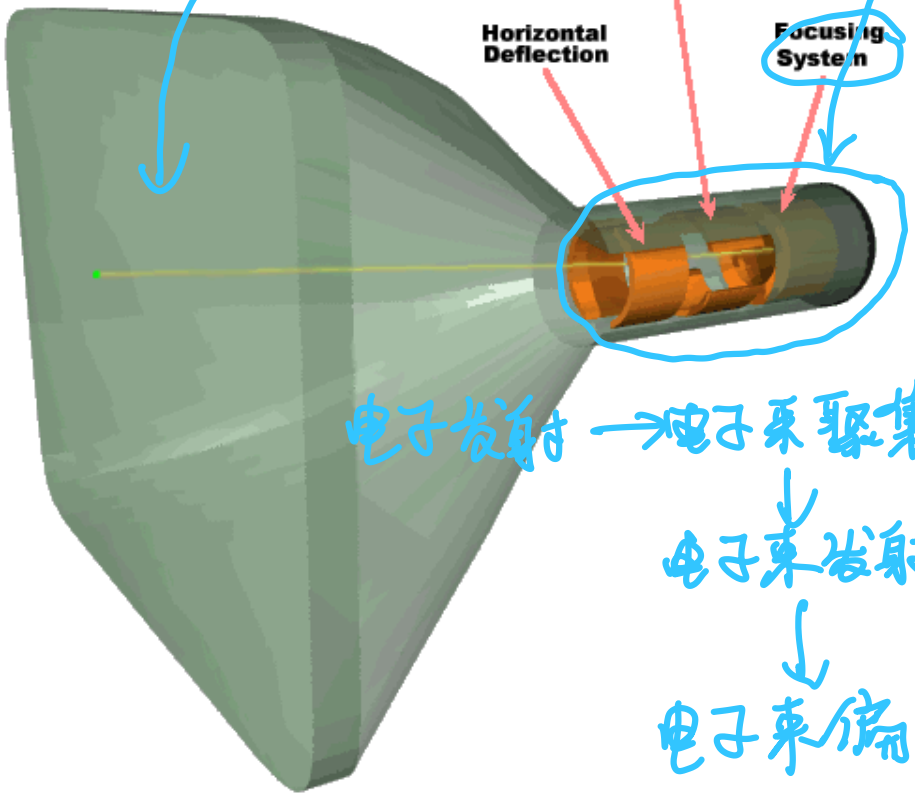
# Raster display 光栅显示

- Cathode Ray Tubes (CRTs), most “tube” monitors you might see. Used to be very common, but big and bulky. (电子管，过去大，笨重)
- Liquid Crystal Displays (LCDs), there are two types: transmissive (laptops, those snazzy new flat panel monitors) and reflective (wrist watches).

透射型

反射型

# Cathode ray tubes (CRTs)



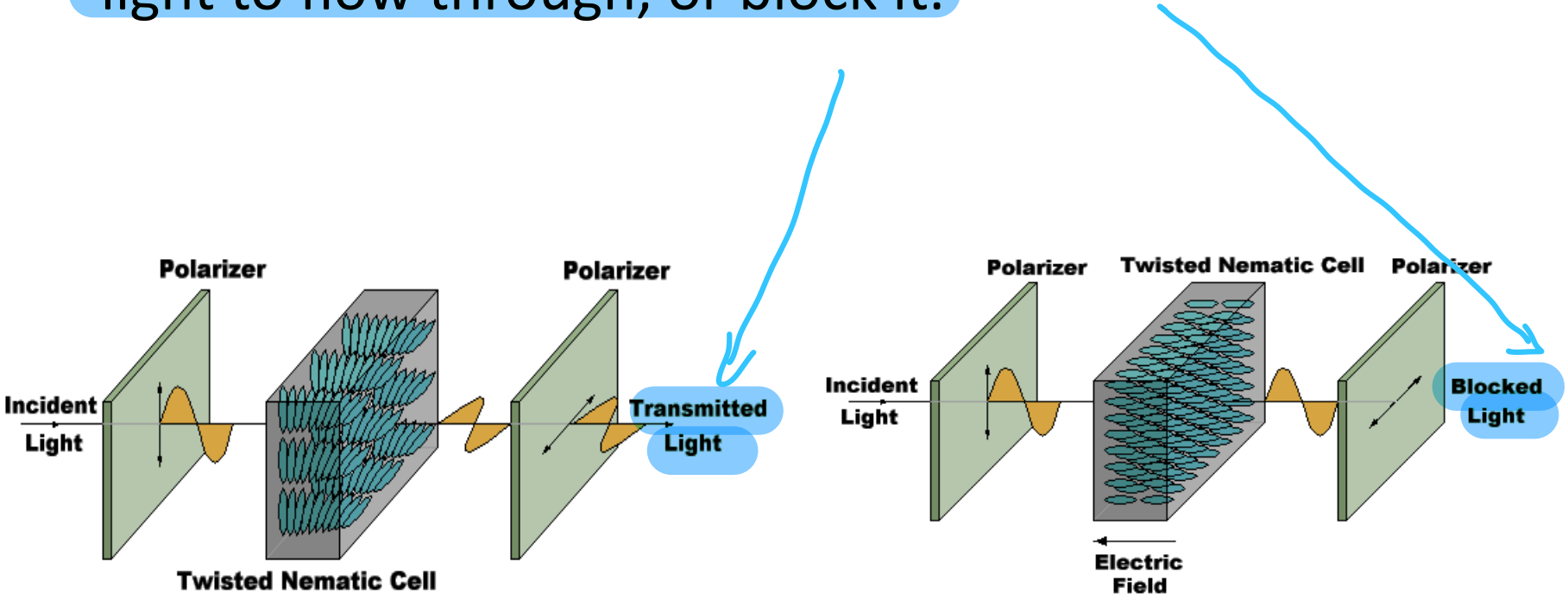

- Heating element on the yolk
- Phosphor coated screen
- Electrons are boiled off the filament and drawn to the focusing system.
- The electrons are focused into a beam and “shot” down the cylinder.
- The deflection plates “aim” the electrons to a specific position on the screen.

# Cathode ray tubes (CRTs)

- Strong electrical fields and high voltage 强电场 高电压
- Very good resolution
- Heavy, not flat

# Liquid crystal displays (LCDs)

- Also divided into pixels, but without an electron gun firing at a screen; LCDs have cells that either allow light to flow through, or block it.



# Liquid crystal displays (LCDs)

- Liquid crystal displays use small flat chips which change their transparency properties when a voltage is applied.
- LCD elements are arranged in an n x m array called the LCD matrix.
- The level of voltage controls grey levels. 电压级来控制灰度.
- LCDs elements do not emit light; use backlights behind the LCD matrix. LCD中液晶分子本身不能发光需要背后有光源来提供光线这个光源叫做背光源
- Colour is obtained by placing filters in front of each LCD element.
- Image quality dependent on viewing angle. 图像质量取决于视角

# Advantages of LCDs

- Flat 平
- Light weight
- Low power consumption 低功耗

# What is graphics software?

- Graphics drivers 图形驱动
- Graphics libraries 图形库
- Graphics editors 图形编辑器
- Geometric modellers 几何建模器
- VR modellers VR模型
- Games 游戏
- Scientific visualisation packages 科学可视化包
- …

# Graphics software

- How to talk to the hardware? 工具包
  Algorithms, Procedures, Toolkits and Packages
  (Low Level ⟷ High Level)
- Programming API (helps to program, for our labs) 编程API
  - OpenGL (our focus)
  - JOGL (Open GL for Java)
  - OpenCV
  - DirectX
  - …
- Special purpose software (not our focus) 专用软件
  - Excel, Matlab, …
  - AutoCAD, Studio Max, Unity, Maya, …
  - Medical visualisation, modelling, …

# OpenGL


- First introduced in 1992.
- The OpenGL graphics system is a software interface to graphics hardware (GL stands for Graphics Library).
- For interactive programs that produce colour images of moving three-dimensional objects.
- It consists of over 200 distinct commands that you can use to specify the objects and operations needed to produce interactive three-dimensional applications.


# OpenGL


- OpenGL is designed as a streamlined, hardware-independent interface to be implemented on many different hardware platforms.
- Similarly, OpenGL does not provide high-level commands for describing models of three-dimensional objects.
- With OpenGL, you must build up your desired model from a small set of geometric primitives - points, lines, and polygons.
- With OpenGL, you can control computer-graphics technology to produce realistic pictures or ones that depart from reality in imaginative ways.
- OpenGL has become the industry standard for graphics applications and games.

# Summary

- These are interesting times for computer graphics.
  - Commodity graphics cards are highly capable.
  - New algorithms, long-offline algorithms are becoming possible.
  - Hard to keep up, even for “experts”.
- What’s pushing the technology curve?
  - Games
  - Movies
  - Interactive graphics applications

# Topics covered today and …

- Computer Graphics and its main topics
- Graphics Hardware
  - Input, Processing and Output Devices
  - Framebuffers
  - Pixels and Screen Resolution
- Graphics Software
  - Techniques (Algorithms, Procedures)
  - Programming Library / API (OpenGL, JOGL, OpenCV, DirectX)
  - Not our focus: High level Interactive Systems (Maya, Studio Max, Unity, AutoCAD, and so on)
- Think about …
  - What are possible bottlenecks in system performance of a graphics system?
  - How can you achieve realism in a graphics system?
  - How do we combine the software and hardware to maximise performance?