# Introduction to Graphics Programming and its Applications

# 繪圖程式設計與應用

## Syllabus

**Hung-Kuo Chu**

Department of Computer Science

National Tsing Hua University

CS4505

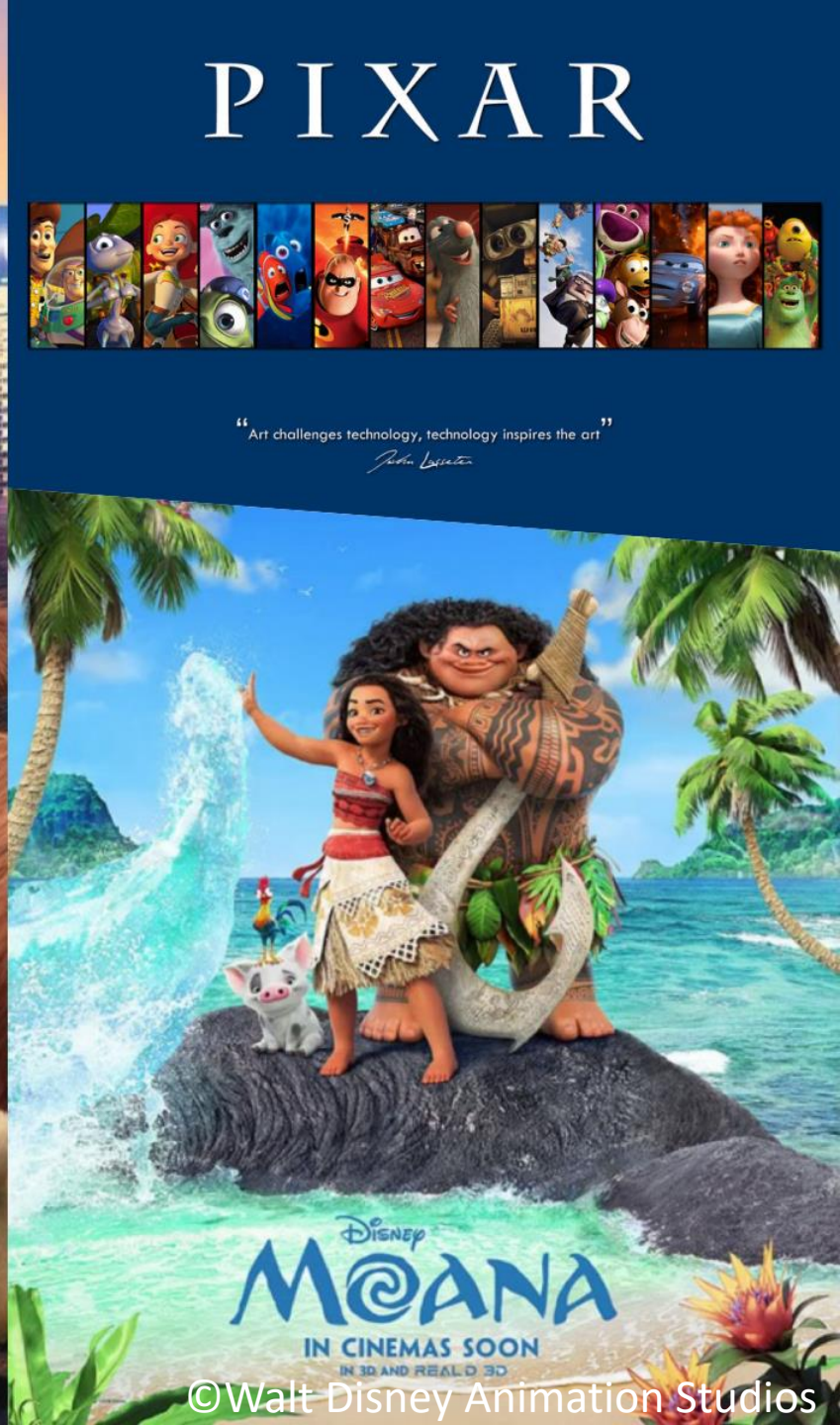
# What is Computer Graphics?




# Game Industry

# Movie Industry

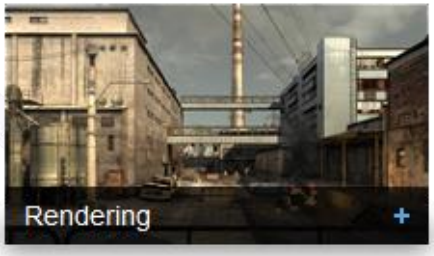

©Illumination Entertainment



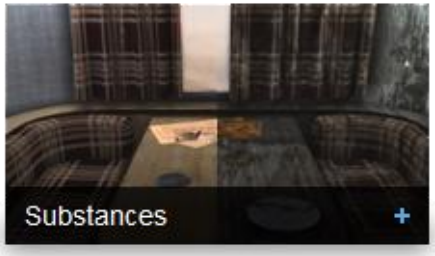

©Walt Disney Animation Studios

# Game Engines

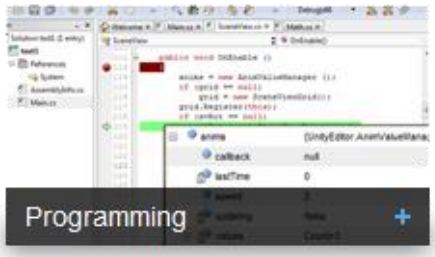










unity 5


Adam

POWERED BY

UNREAL
ENGINE

# Infiltrator

# Virtual Reality

©HTC vive

©Oculus Rift

©Samsung Gear VR

# Augmented Reality


©IKEA

©Pokemon GO

# Mixed Reality



Project Tango
Lenovo

# 3D Authoring Tools

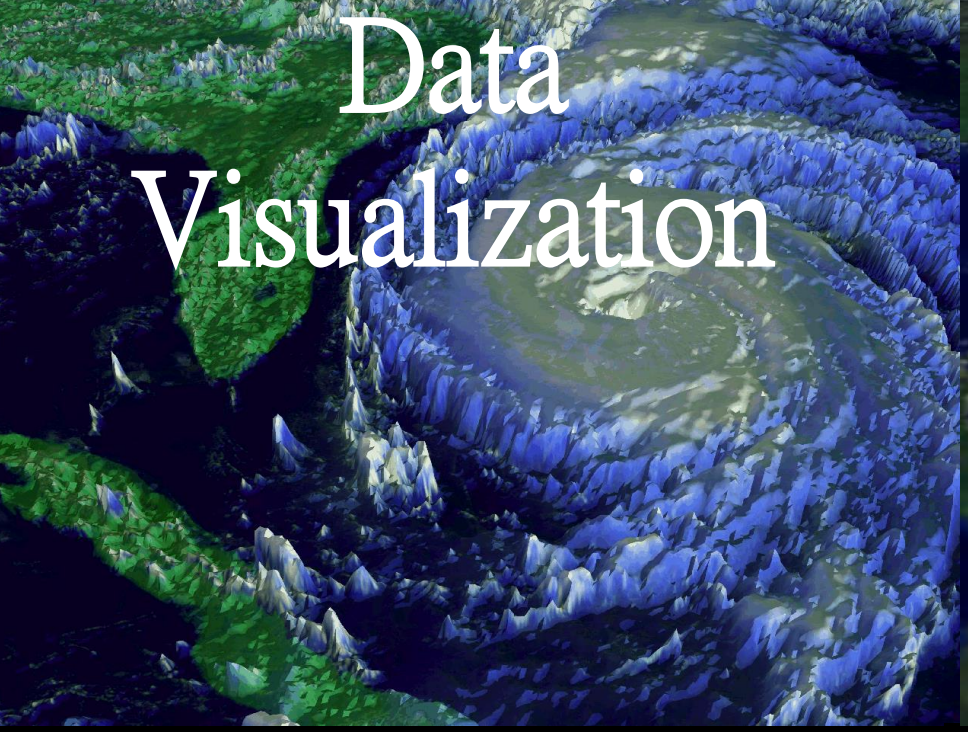

AUTO
Show Reel 2016

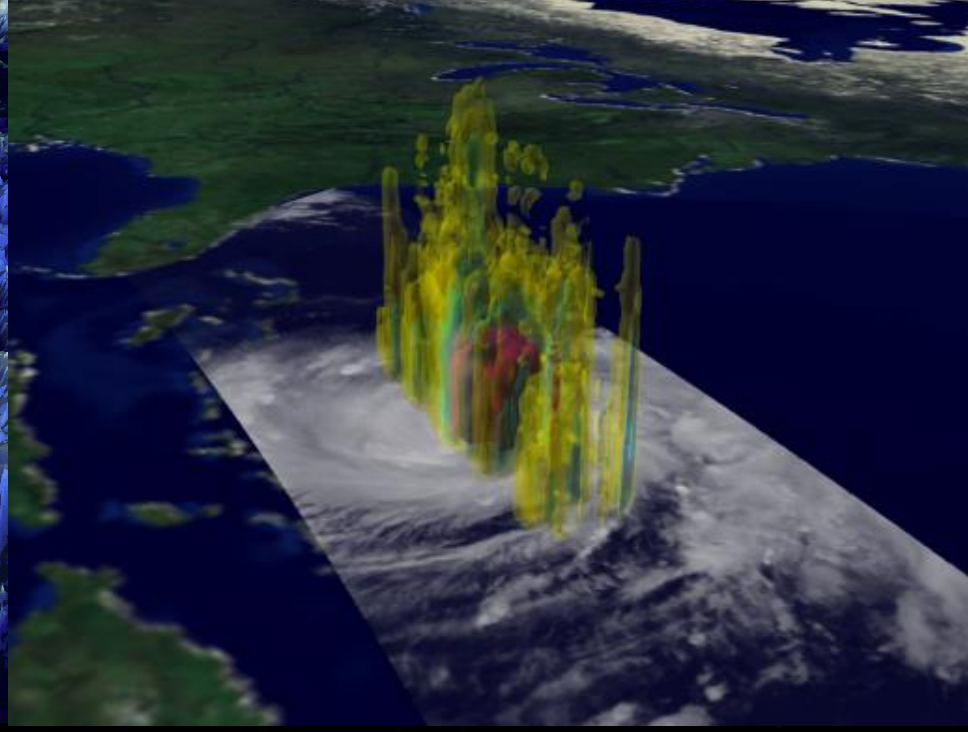

Make Ideas Real
vino
01:15
-00:15

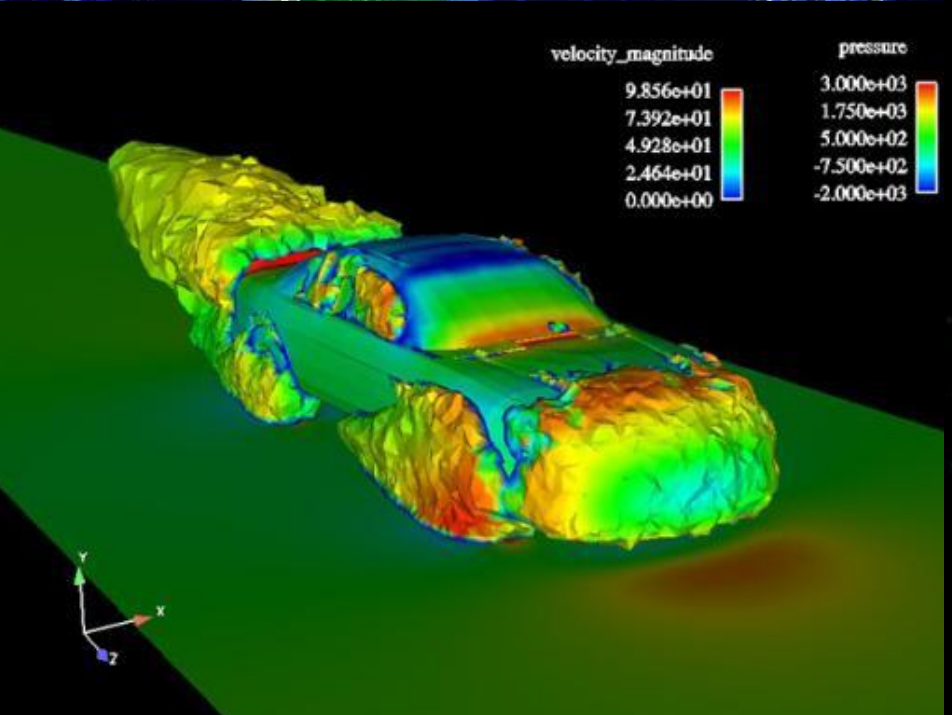

Human Computer Interaction

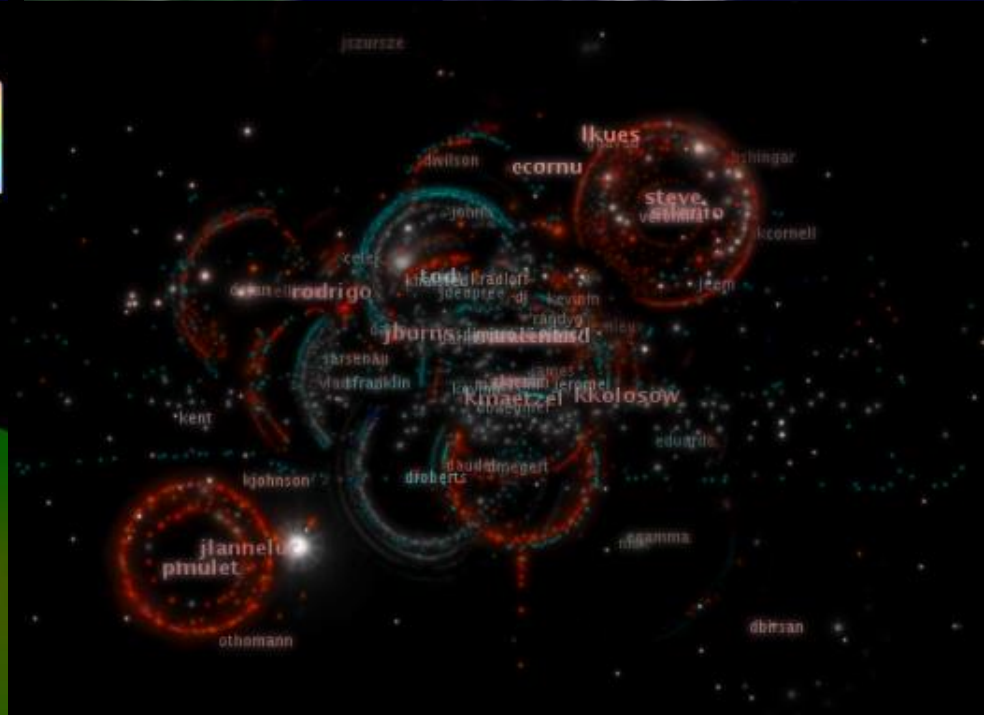
# Medical Image

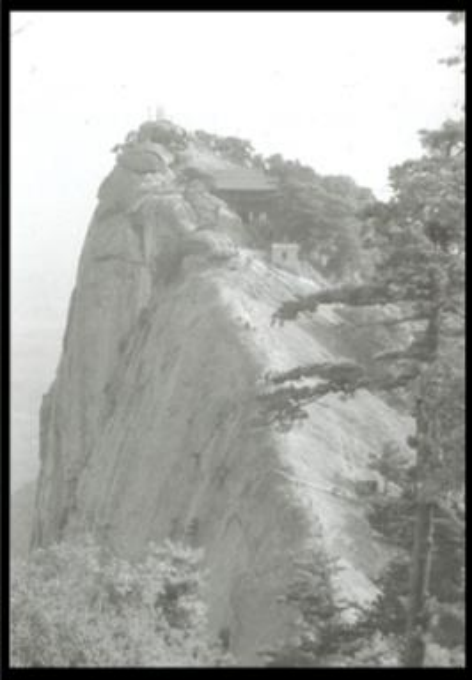
# Data Visualization









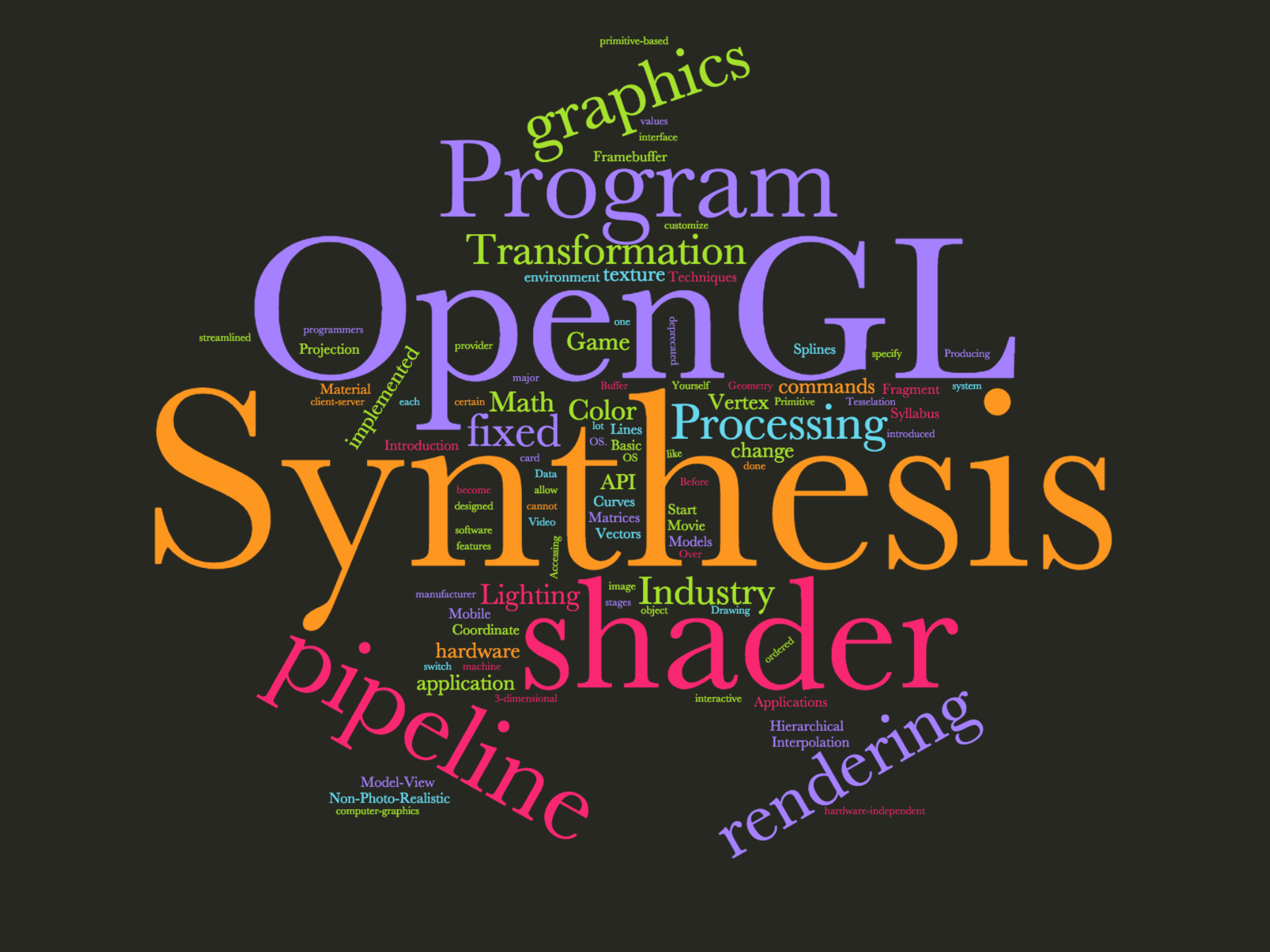



# Non-Photorealistic Rendering

So…

Computer Graphics

is all about


primitive-based
graphics
values
interface
Framebuffer
Program
customize
Transformation
environment
texture
Techniques
OpenGL
streamlined
programmers
Projection
provider
one
Game
deprecated
Splines
specify
Producing
implemented
Material
client-server
each
certain
major
Buffer
Yourself
Geometry
commands
Fragment
system
Math
Color
Vertex
Primitive
Tesselation
Syllabus
fixed
lot
Lines
Processing
introduced
Introduction
OS.
Basic
OS
like
change
done
card
Data
API
Before
become
allow
designed
cannot
Curves
Start
Video
Matrices
Movie
software
Vectors
Models
features
Accessing
Over
manufacturer
Lighting
image
Industry
stages
object
Drawing
Mobile
Coordinate
shader
hardware
ordered
switch
machine
application
3-dimensional
interactive
Applications
pipeline
Hierarchical
Interpolation
rendering
Model-View
Non-Photo-Realistic
computer-graphics
hardware-independent

# Synthesis / Rendering

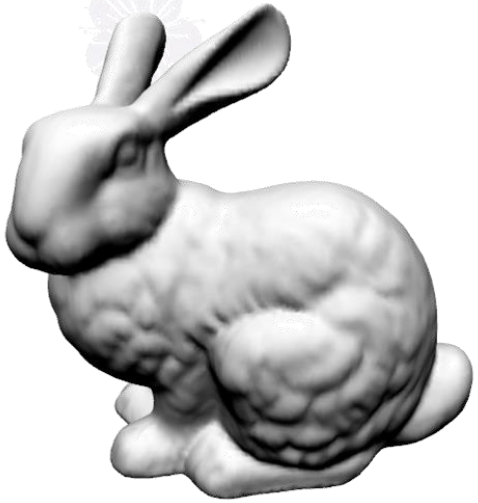
Data

You

?



Hardwares



Image

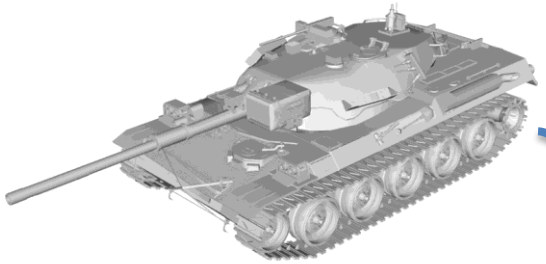
Introducing…


# Open Graphics Library

# What is OpenGL?

- OpenGL is an *application programming interface (API*)
- Accessing features in graphics hardware
- Over 500 commands to specify image, texture, object and shader programs
- Producing interactive 3-dimensional computer-graphics applications
- Latest version: OpenGL 4.5

# What is OpenGL?



3D Model

Texture

Light =(1.5, 0.6, 0.2)
Material=……

OpenGL®

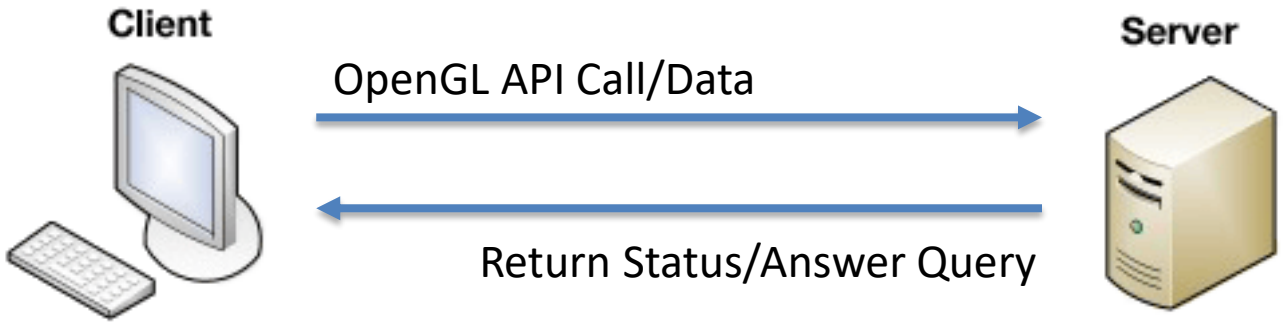
Your OpenGL application

Rendered image

# What is OpenGL?

- OpenGL is designed as:
  - A streamlined, hardware-independent API
  - A primitive-based rendering pipeline
  - A client-server system
- OpenGL is implemented by:
  - Video card manufacturer for each hardware/OS
    - Nvidia, AMD
  - OS provider (software rendering)
  - Yourself! (***Computer Graphics CS 5500***)

# What is OpenGL?




Your Application Code

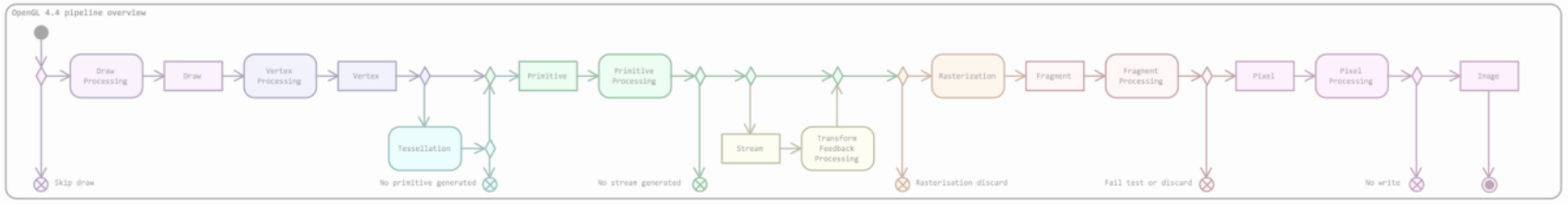
Rendering Algorithm
Load Data from Hard Disk
…

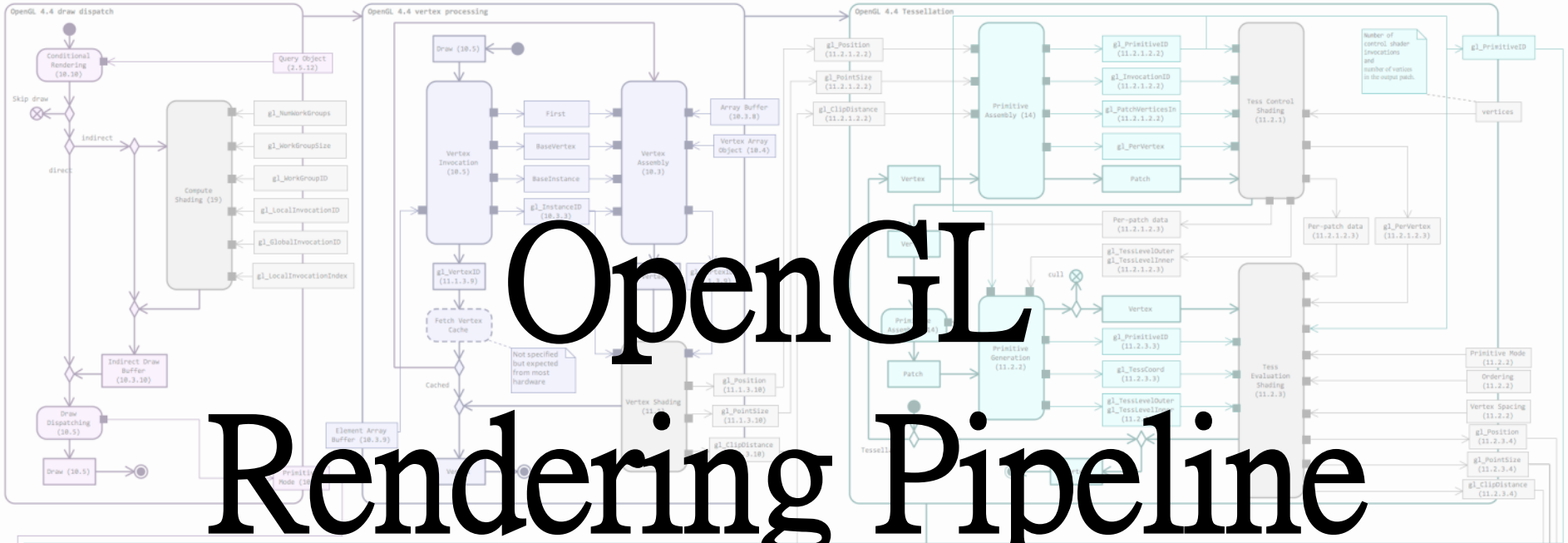
OpenGL Implementation
(AMD/NVIDIA Driver)

OpenGL States:
DEPTH_TEST=ON
Shader ID=1
…

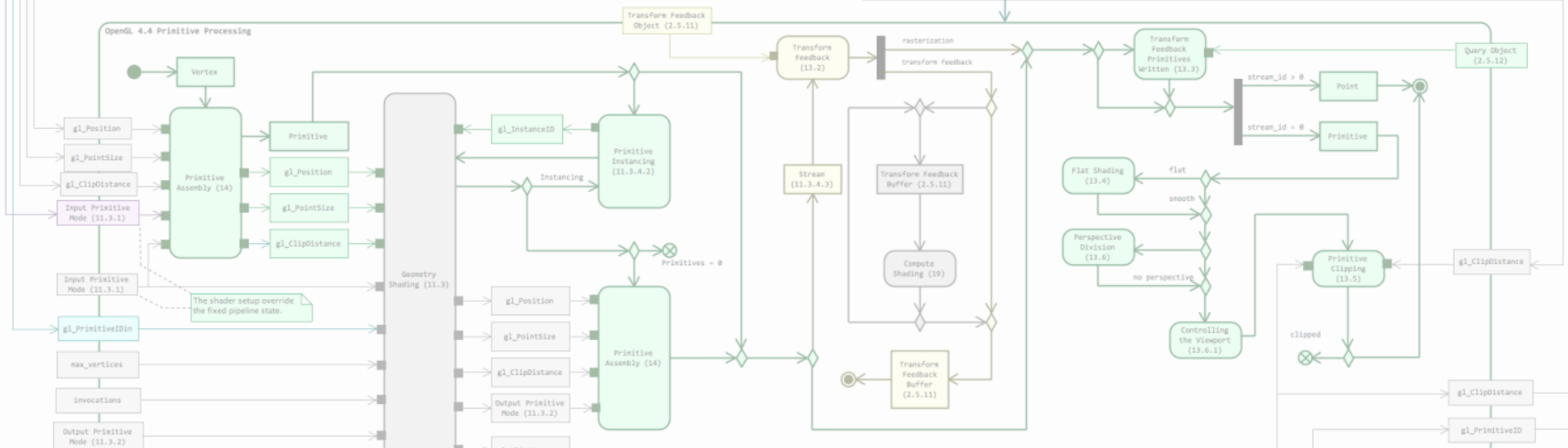
# Vulkan

- A.K.A. OpenGL Next
- Idea similar to AMD Mantle and Direct3D 12
- A Fully cross-platform API
  - Windows, Linux, Android…
  - Google has acquired the mobile group of LunarG, the develop team of the Vulkan driver



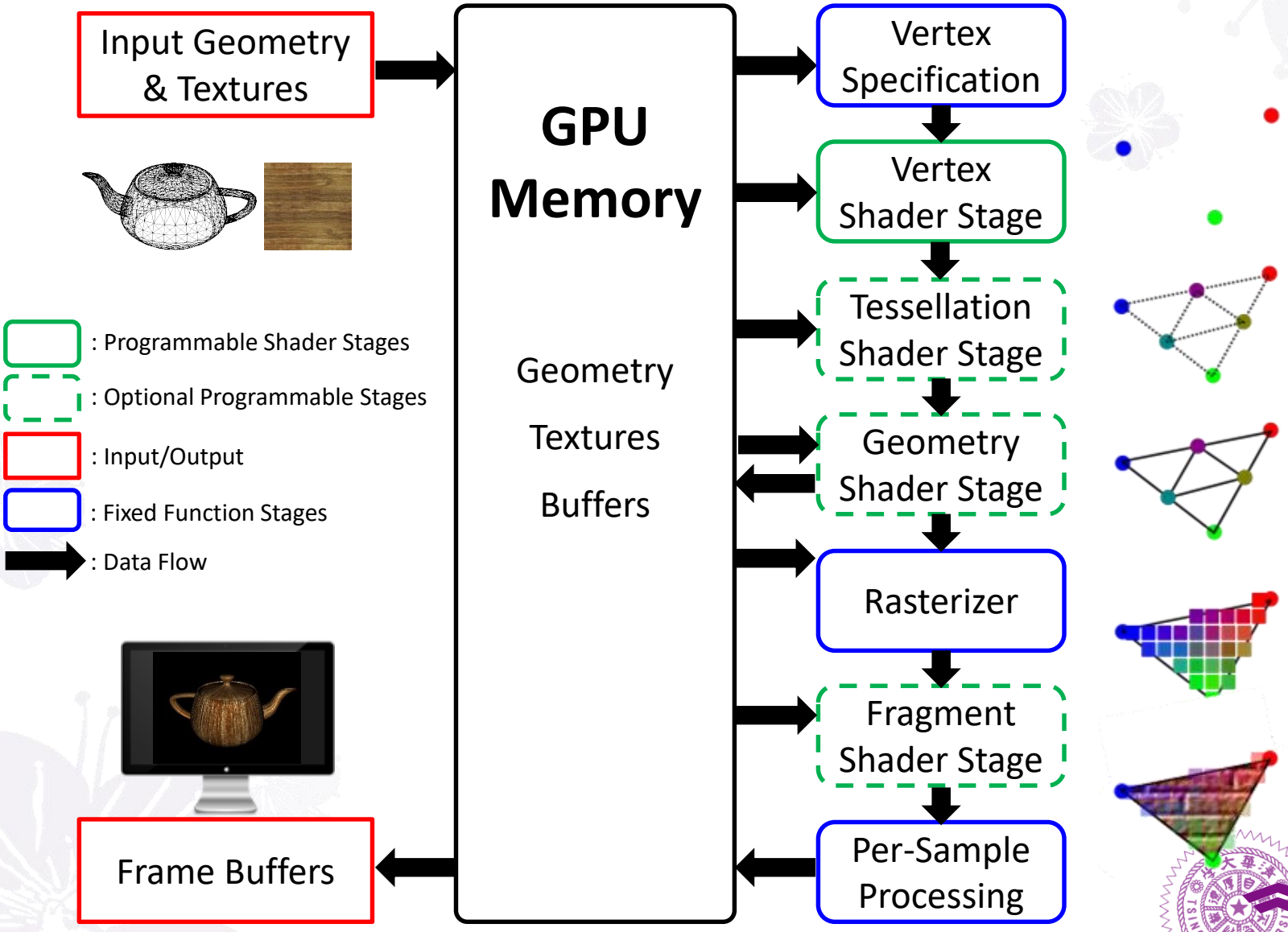

# OpenGL Rendering Pipeline



# Rendering Pipeline

- Pipeline: consists of multiple stages. Data flows in, being processed in each stages, then flows out
- Stages: each stage represents an unique function to process the input data
  - **Fixed function** stages: limited customization capability, typically exposes states for configuration
  - **Programmable shader** stages: allow custom shader programs to be executed within, providing broader capability of customization

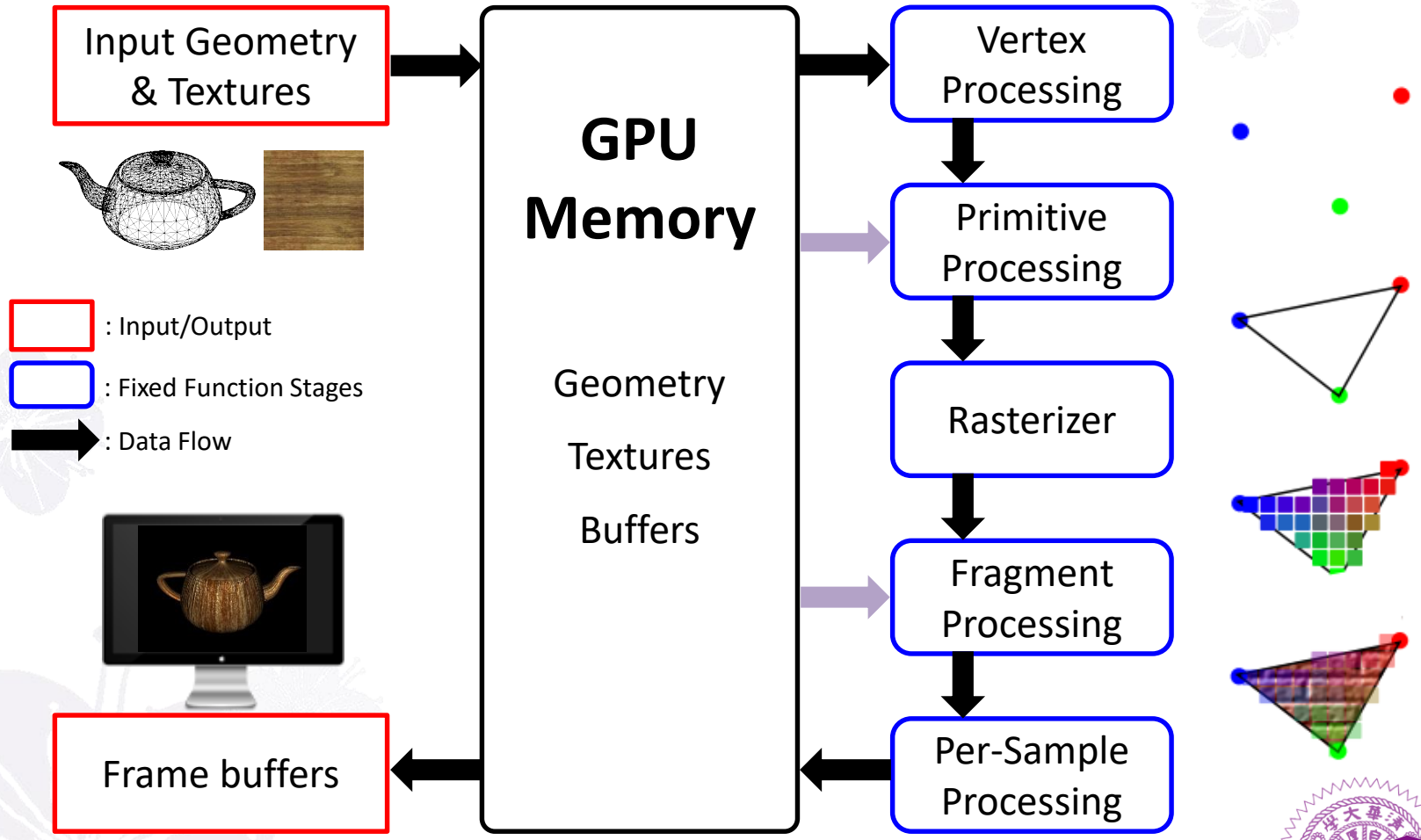
# The OpenGL Rendering Pipeline



# Fixed Function Pipeline (Legacy)

- Before OpenGL 3.0, OpenGL rendering is done in a fixed function pipeline
- Fixed pipeline is like an machine with a lot of switches/values to configure
- One cannot change how the function is implemented as well as the order of execution

# Fixed Function Pipeline (Legacy)



# Fixed Function Pipeline: Metaphor

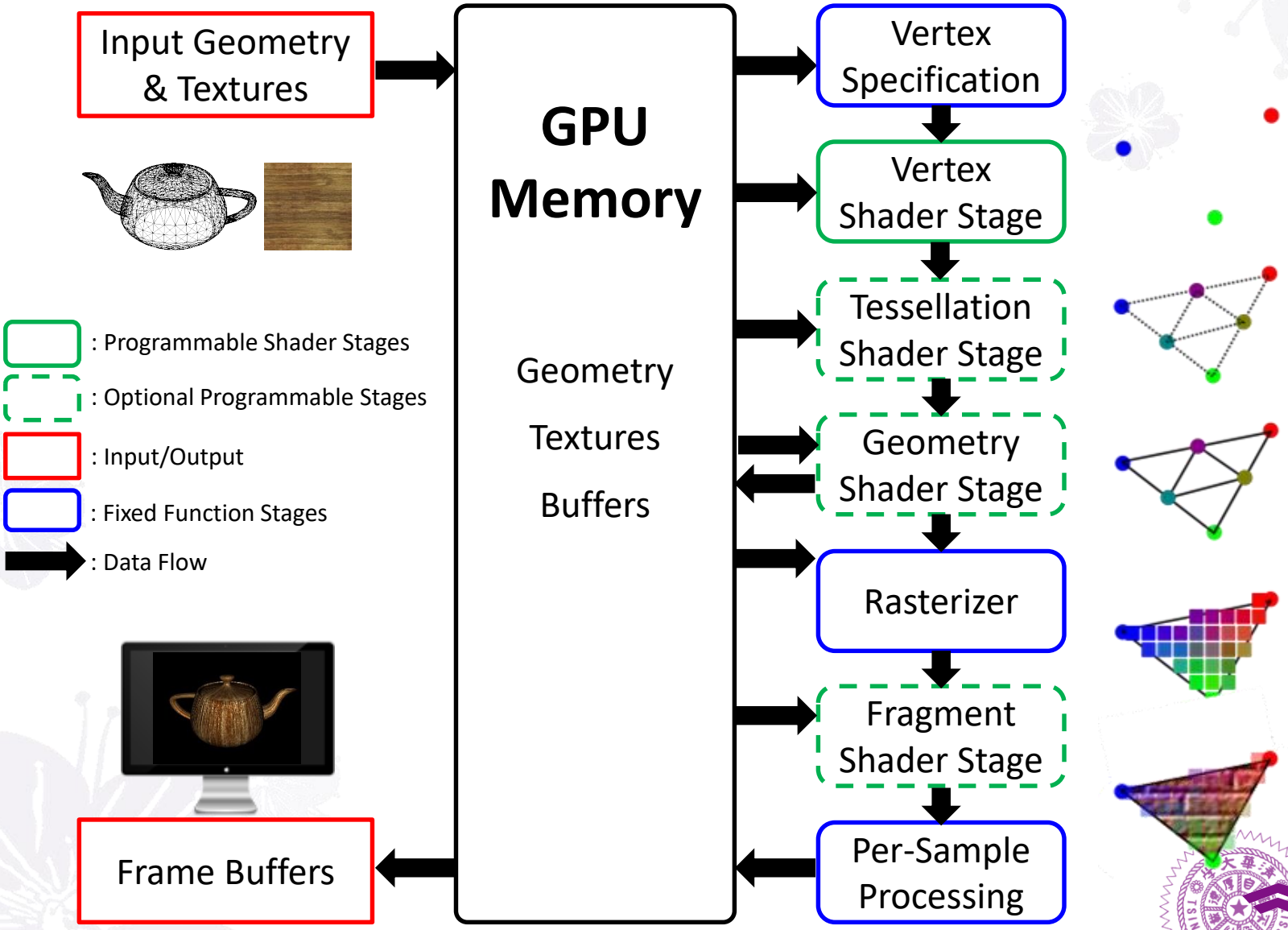
OpenGL Fixed Function Pipeline


How do I press these buttons to get desired effect?

# Programmable Pipeline

- Shader programs are introduced in OpenGL 2.0, and included in the core profile in OpenGL 3.0
- Fixed function pipeline is deprecated since OpenGL 3.0
- Shader programs, written in OpenGL Shading Language(GLSL), allow the programmers to customize certain stages in the OpenGL rendering pipeline

# The OpenGL Rendering Pipeline



# Programmable Pipeline: Metaphor

Shader Programs

```
uniform float t;     // Time (passed in from the application)
attribute vec4 vel; // particle velocity

const vec4 g = vec4( 0.0, -9.80, 0.0 );

void main()
{
    vec4 position = gl_Vertex;
    position += t*vel + t*t*g;

    gl_Position = gl_ModelViewProjectionMatrix * position;
}
```

Let's write a program
to create the effect!

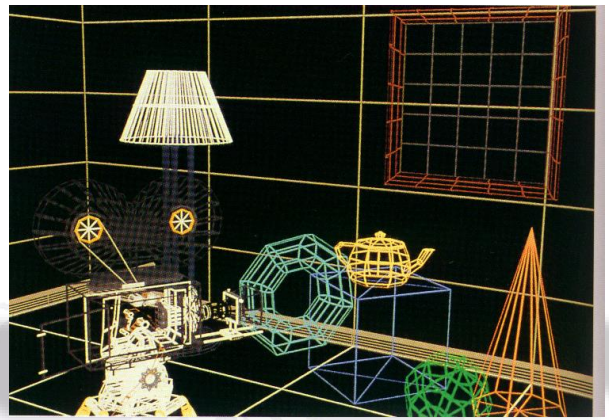
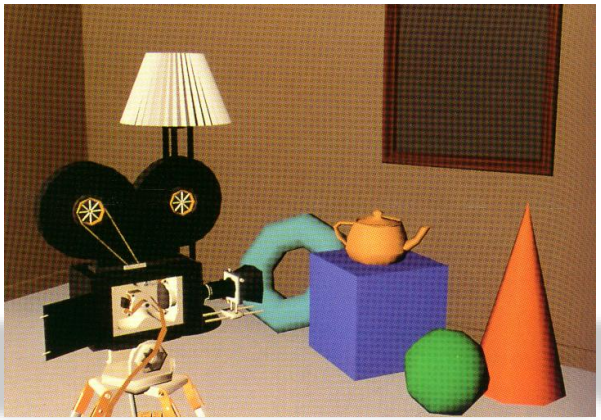
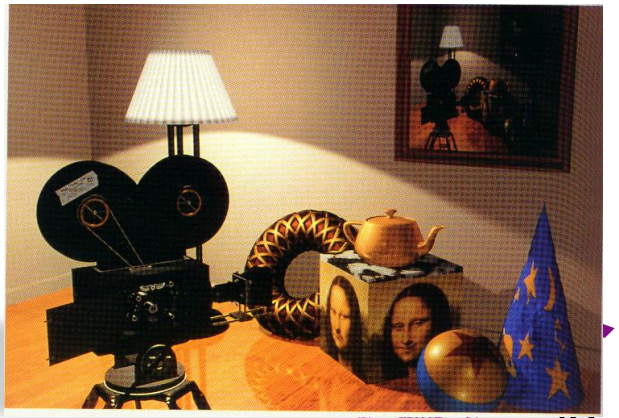
# Programmable Pipeline V.S. Fixed Function Pipeline

| | Programmable Pipeline | Fixed Function Pipeline |
|---|---|---|
| Flexibility | +implement various algorithms in shader programs | -limited customization capability |
| For Simple Application | -must configure the whole pipeline | +performs simple task with less configurations |
| For Complex Application | +can achieve various effects | -most advanced effects are impossible |
| Learning Curve | -one must have full knowledge of the pipeline and GLSL before writing an application | +can create simple applications without much knowledge |
| Deploy | -should consider all graphics driver environment | +works in most graphics driver environment |

# COURSE OUTLINES

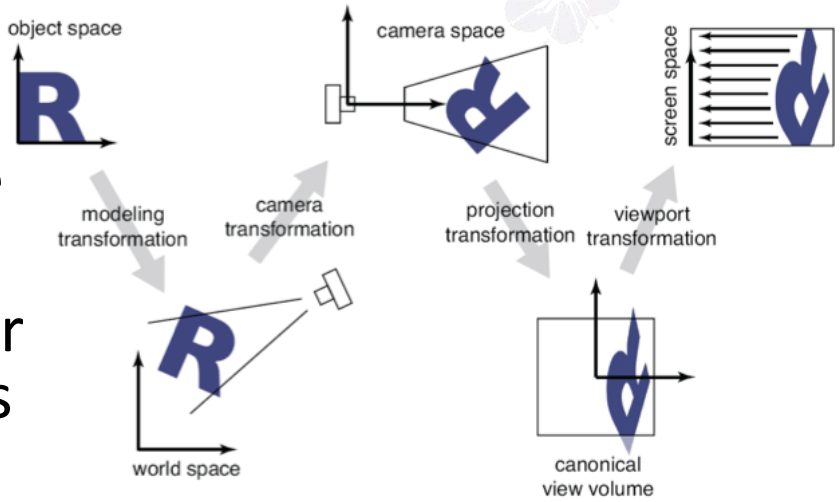
# Fundamental of Computer Graphics

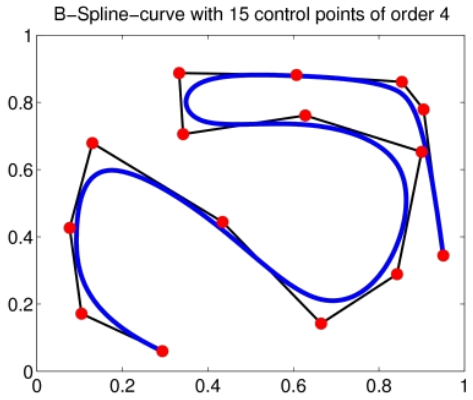
- 3D geometry representation
- 3D projection and transformation
- Color, material and lighting
- Texture mapping
- Recommend ***Computer Graphics (CS 5500)***

# Shader Programming (OpenGL)

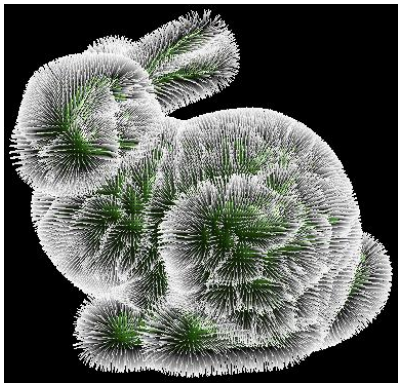
- Rendering Pipeline
  - Configuring the pipeline
  - OpenGL Shading Language (GLSL) basics
  - Writing shader program for each programmable stages
- Math
  - Vectors and matrices
  - Coordinate spaces and transformations
  - Higher-order lines, curves, splines

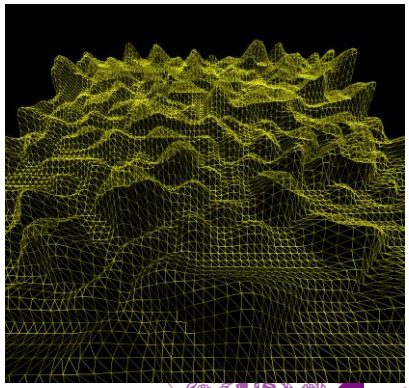


# Shader Programming (OpenGL)

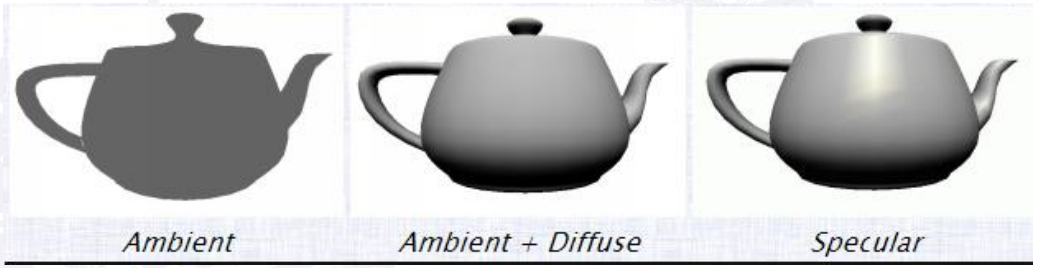
- Data
  - Buffer
  - Texture
- Vertex Processing
  - Vertex shader
  - Drawing commands
- Primitive Processing
  - Geometry shader
  - Tessellation shader





# Shader Programming (OpenGL)

- Fragment processing
  - Color
  - Frame buffer processing
- Rendering techniques
  - Lighting models
  - Non-photorealistic rendering


# Schedule (Temporary)

| Date | Course | Quiz | Homework |
|---|---|---|---|
| 2/13 | Syllabus | | |
| 2/20 | Shader Pipeline | | |
| 2/27 | **2 / 28** 補假 | Quiz1 | |
| 3/06 | Projection and Transformation | | |
| 3/13 | Buffer, Texture and Uniform | | |
| 3/20 | Vertex Processing and Drawing Commands | | HW1 |
| 3/27 | Vertex Processing and Drawing Commands | Quiz2 | |
| 4/03 | 清明連假 | | HW1 Deadline |
| 4/10 | Fragment Processing and Frame Buffer | Quiz3 | |
| 4/17 | Fragment Processing and Frame Buffer | Quiz4 | |

# Schedule (Temporary)

| Date | Course | Quiz | Homework |
|---|---|---|---|
| 4/24 | Mesh Rendering | | HW2 |
| 5/01 | Tessellation Shader | Quiz5 | |
| 5/08 | Geometry Shader | | HW2 Deadline |
| 5/15 | Skinned Animation | | HW3 |
| 5/22 | Shadowing, Normal Mapping and SSAO | | |
| 5/29 | 端午節放假 | | HW3 Deadline |
| 6/05 | Compute Shader (Water, Cloth Simulation) | | |
| 6/12 | TBA | | |
| 6/19 | 暑假開始 **(Final demo)** | | Final Deadline |

# Prerequisites

- **C/C++** programming skill.
- Fundamental **linear algebra** knowledge.
- **Hard/smart working**.
- Enjoying the **programming**.
- Enthusiasm for **computer graphics**.

# Class Information

- Classroom
  - Lecture: 台達103
  - Quiz: 資電326, 328
- Class time (星期一):
  - Lecture: 15:30-18:20
  - Quiz: 17:30-18:20
  - LAB: 18:30-19:20
- Course webpage:
  - ILMS (http://lms.nthu.edu.tw/)

# Instructor & TAs

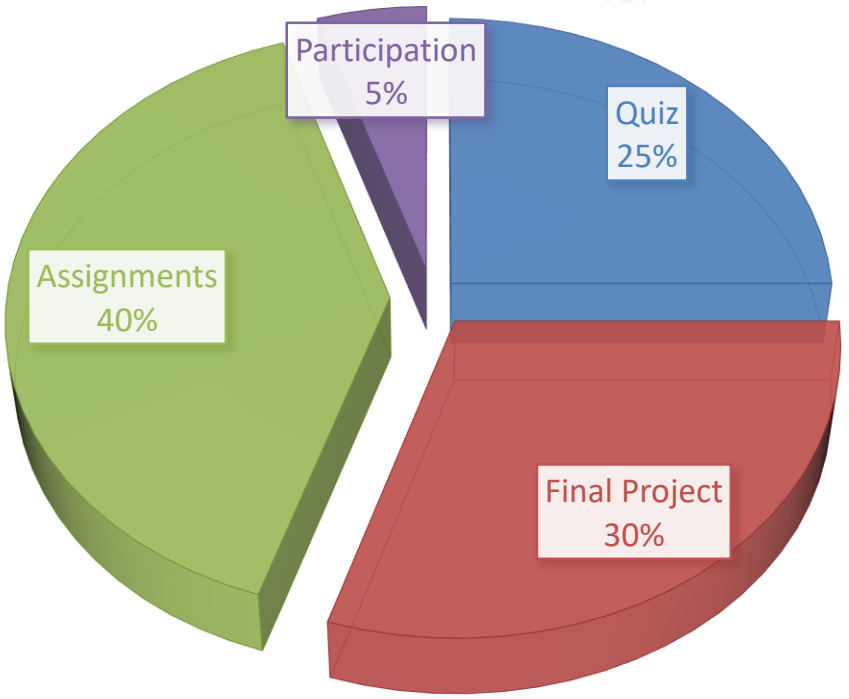
| Name | 朱宏國 | TAs |
|---|---|---|
| Office | 台達館641室 | 資電館839/840室 |
| TEL | 03-5731215 | 03-5733525, 03-5733531 |
| E-Mail | hkchu@cs.nthu.edu.tw | 林佑恩，張宏瑞，林育睿，楊尚達<br>**Contact TAs via ILMS** |
| Office Hours | Anytime (email me) | |

# Workload ?

Image courtesy of Stuart Miles / FreeDigitalPhotos.net

# Workload & Evaluation

- Final project (30%)
  - TBA
- Assignment (40%)
  - Homework: 3
- Quiz (25%)
  - In-class: 5
- Participation (5%)

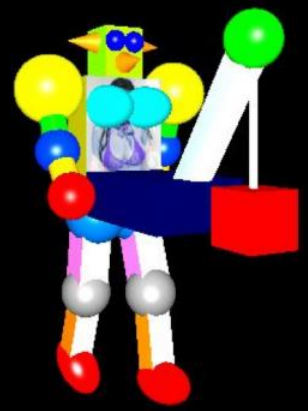


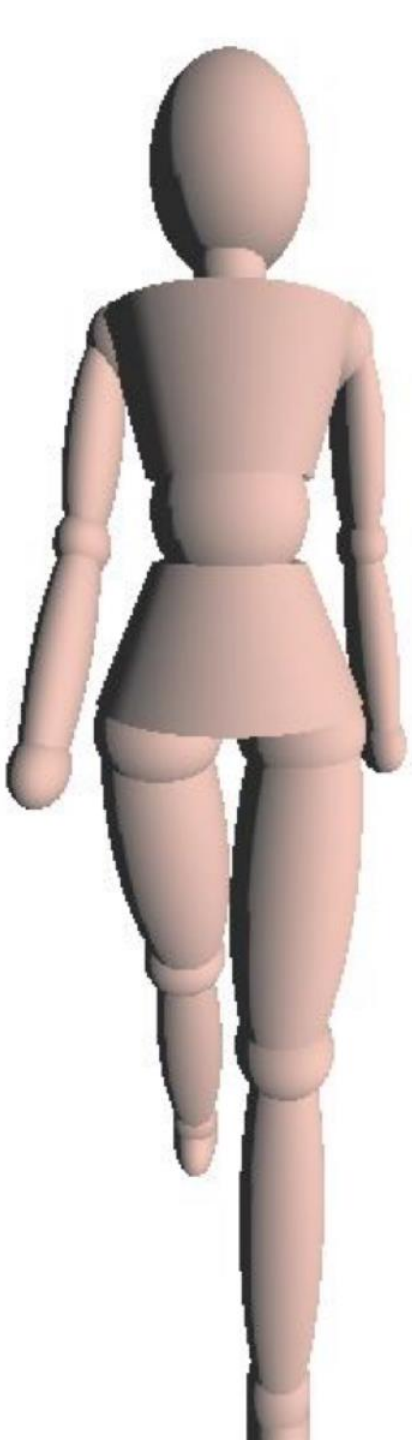
# Quiz Rules

- Time period: 5:30 P.M. – 6:20 P.M.
- Class room: 326 & 328
- Demo screen output to TAs before 6:20 P.M.
- Communication with classmates is forbidden.

# ASSIGNMENT AND FINAL PREVIEWS

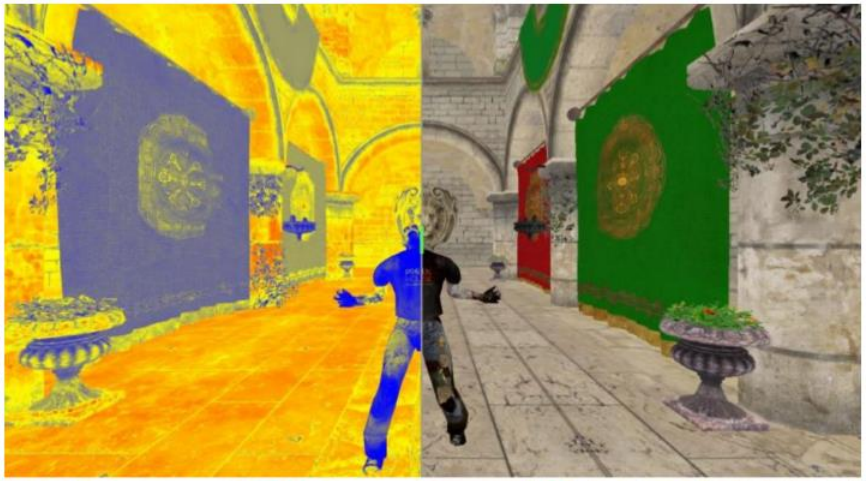
# Assignment: Animated robot

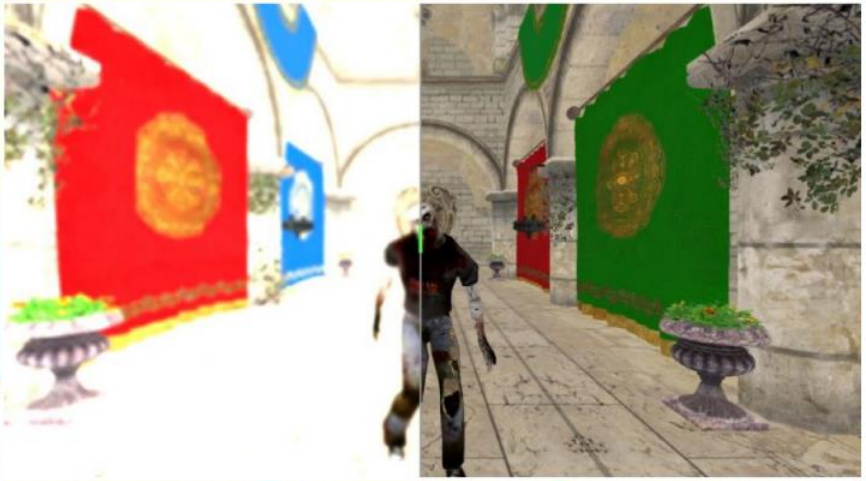




# Assignment:
# Create/Load a complex scene

# Assignment: Post processing






2016 Final Project: Theme park

# Final Project: TBA

Any suggestion is welcomed

# Minimal Requirement for Video Cards

NVDIA

https://developer.nvidia.com/opengl-driver

AMD

http://www.geeks3d.com/20130724/amd-catalyst-opengl-4-3-graphics-driver-7-new-opengl-extensions/


thank you!
Question?