

# CAPITULO I

## REPRESENTACIÓN VISUAL COMPUTARIZADA

1.4

# INTRODUCCIÓN A LA PROGRAMACIÓN DE GRÁFICOS GRAPHICS PROGRAMMING INTRODUCTION

# SGI and GL

- Silicon Graphics (SGI) revolutionized the graphics workstation by implementing the pipeline in hardware (1982)
- To access the system, application programmers used a library called GL
- With GL, it was relatively simple to program three dimensional interactive applications

# OpenGL

The success of GL lead to OpenGL (1992), a platform-independent API that was

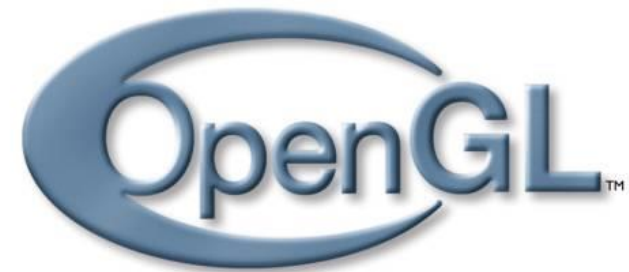
- Easy to use
- Close enough to the hardware to get excellent performance
- Focus on rendering
- Omitted windowing and input to avoid window system dependencies

# OpenGL Evolution

- Controlled by an Architectural Review Board (ARB)
  - Members include SGI, Microsoft, Nvidia, HP, 3DLabs, IBM,.....
  - Relatively stable (present version 4.6)
    - Evolution reflects new hardware capabilities
      - **3D texture mapping and texture objects**
      - **Vertex programs**
  - Allows for platform specific features through extensions

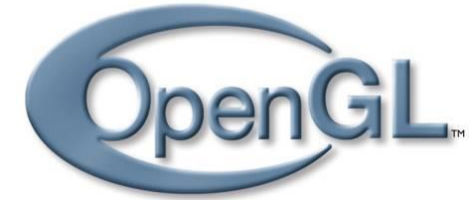
# What Is OpenGL?

- OpenGL is a computer graphics rendering *application programming interface*, or API (for short)
  - With it, you can generate high-quality color images by rendering with geometric and image primitives
  - It forms the basis of many interactive applications that include 3D graphics
  - By using OpenGL, the graphics part of your application can be
    - operating system independent
    - window system independent

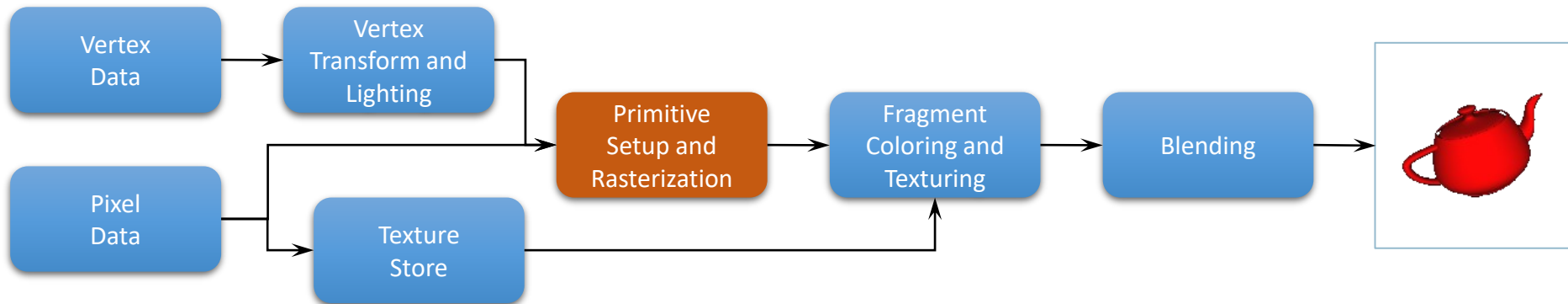


# Evolution of the OpenGL Pipeline

# In the Beginning ...



- OpenGL 1.0 was released on July 1<sup>st</sup>, 1994
- Its pipeline was entirely *fixed-function*
  - the only operations available were fixed by the implementation

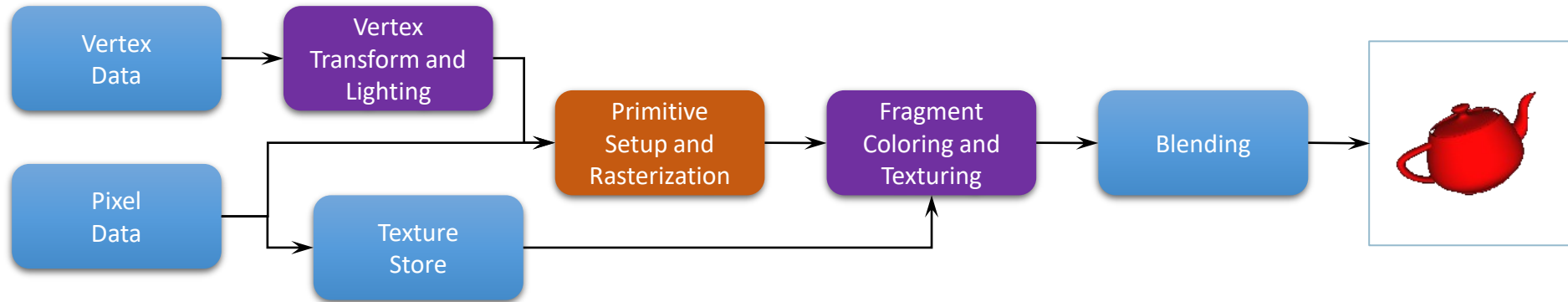


- The pipeline evolved
  - but remained based on fixed-function operation through OpenGL versions 1.1 through 2.0 (Sept. 2004)



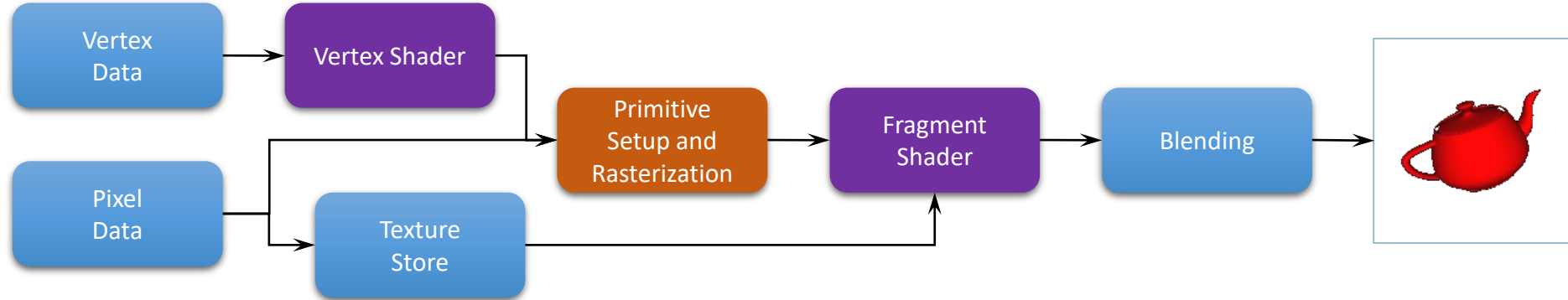
# Beginnings of The Programmable Pipeline

- OpenGL 2.0 (officially) added programmable shaders
  - *vertex shading* augmented the fixed-function transform and lighting stage
  - *fragment shading* augmented the fragment coloring stage
- However, the fixed-function pipeline was still available



# The Exclusively Programmable Pipeline

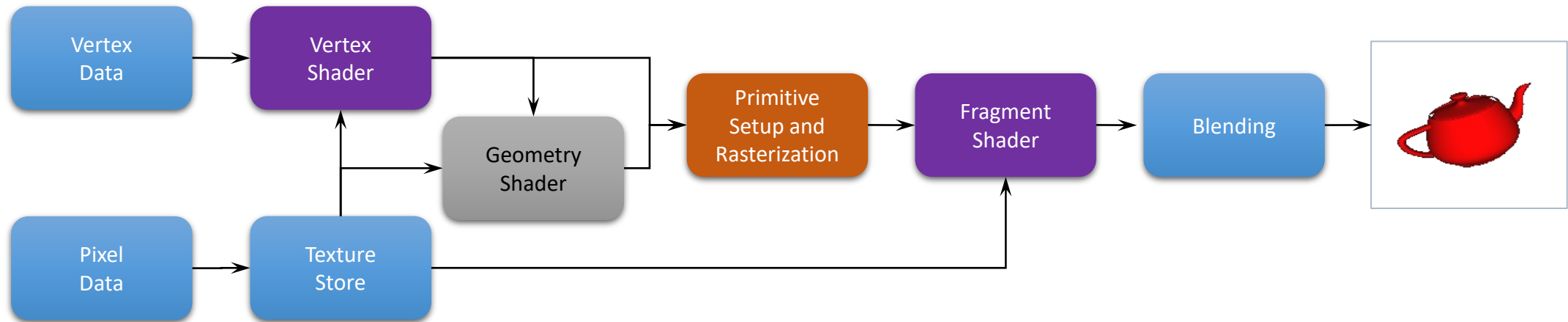
- OpenGL 3.1 removed the fixed-function pipeline
  - programs were required to use only shaders



- Additionally, almost all data is GPU-resident
  - all vertex data sent using buffer objects

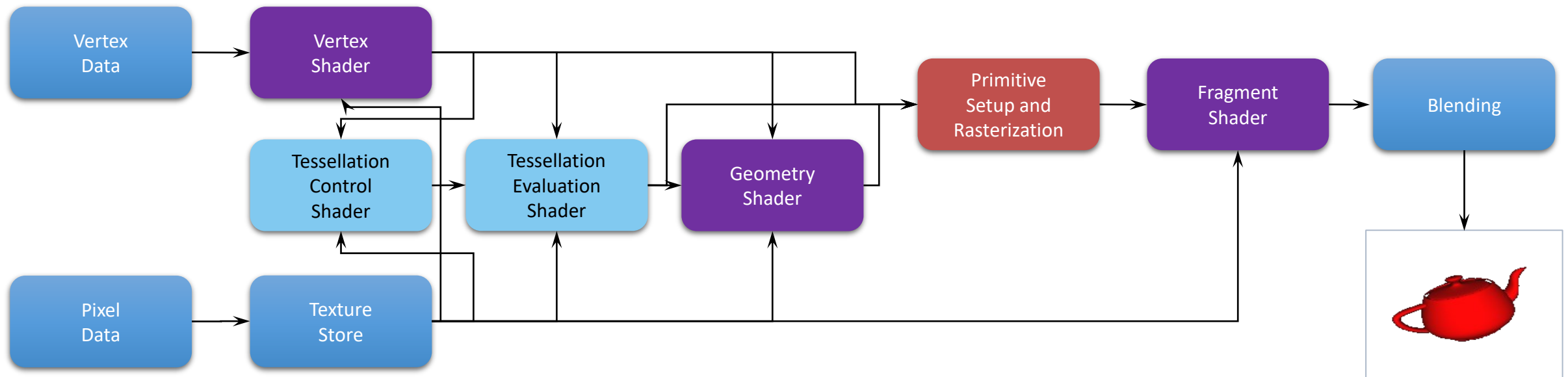
# More Programmability

- OpenGL 3.2 (released August 3<sup>rd</sup>, 2009) added an additional shading stage – geometry shaders
  - modify geometric primitives within the graphics pipeline

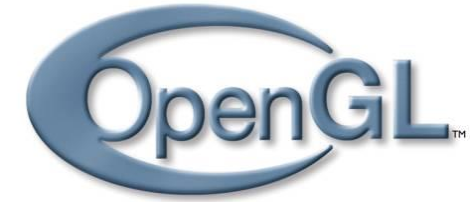


# The Latest Pipelines

- OpenGL 4.1 (released July 25<sup>th</sup>, 2010) included additional shading stages – *tessellation-control* and *tessellation-evaluation* shaders
- Latest version is 4.6

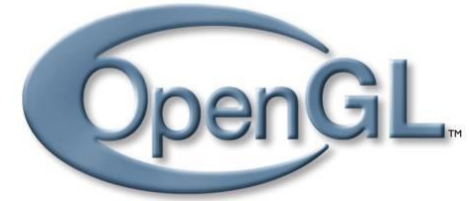


# OpenGL ES and WebGL



- OpenGL ES 3.2
  - Designed for embedded and hand-held devices such as cell phones
  - Based on OpenGL 4.3
  - Shader based
- WebGL 2.0
  - JavaScript implementation of ES 3.0
  - Runs on most recent browsers

# OpenGL Libraries



- OpenGL core library
  - OpenGL32 on Windows
  - GL on most unix/linux systems (libGL.a)

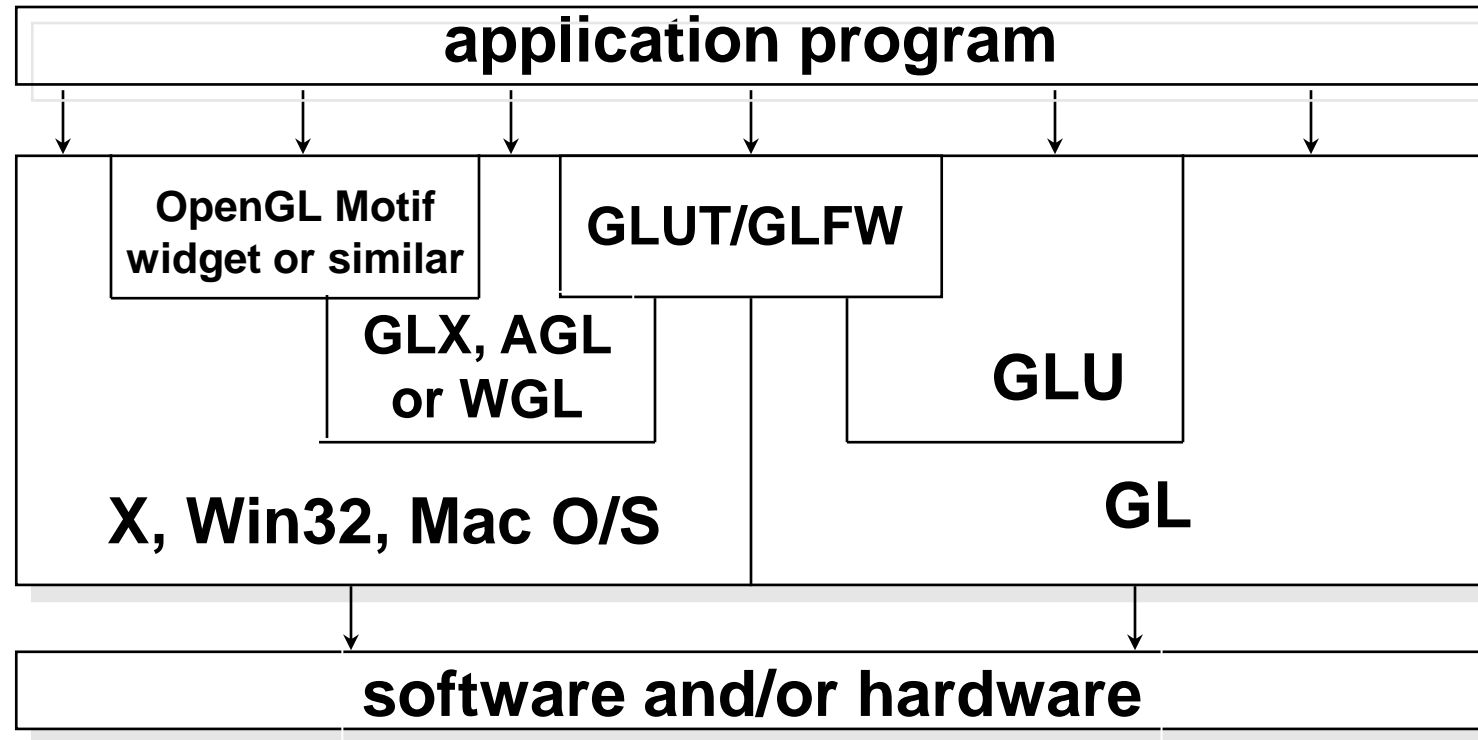
Related toolkits and APIs:

[https://www.khronos.org/opengl/wiki/Related\\_toolkits\\_and\\_APIs#Beginner\\_frameworks](https://www.khronos.org/opengl/wiki/Related_toolkits_and_APIs#Beginner_frameworks)

# GLFW

- GLFW is a free, Open Source, multi-platform library for OpenGL, OpenGL ES and Vulkan application development.
- It provides a simple, platform-independent API for creating windows, contexts and surfaces, reading input, handling events, etc.
- <https://www.glfw.org/docs/latest/index.html>

# Software Organization





# Software Org. – Window & Graphics Systems

- **Window system** controls/manages window creation, appearance, messages (event queue), etc.
  - Has its own API for programming, with and without accessing OpenGL
- **Graphics system** controls graphics display hardware
  - Here, OpenGL – GL and its utilities, GLU

