

---

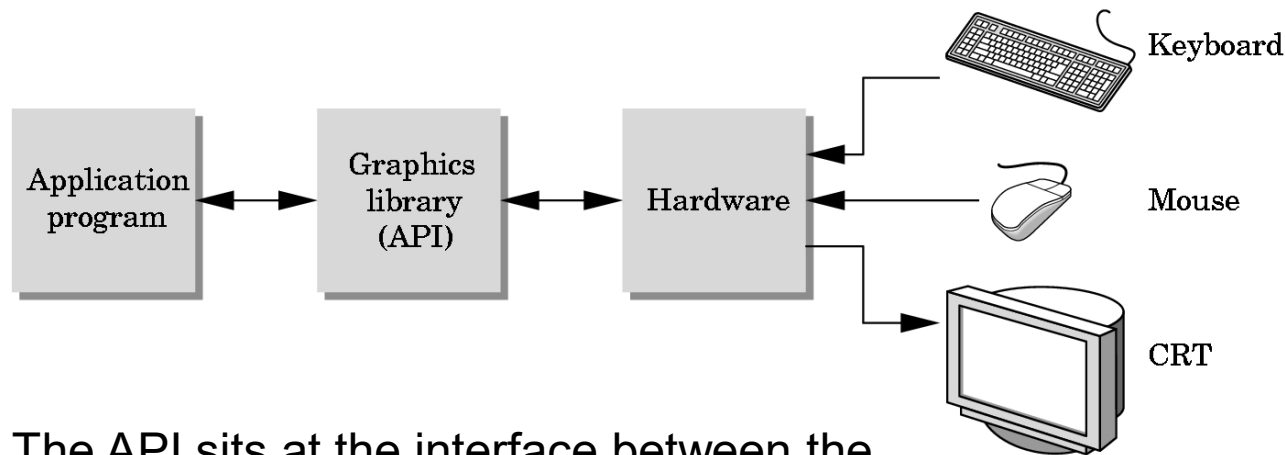
# OpenGL API

Prof. George Wolberg  
Dept. of Computer Science  
City College of New York

# The Programmer's Interface

---

- Programmer sees the graphics system through a software interface: the Application Programmer Interface (API)



The API sits at the interface between the application program and the graphics hardware

# SGI and GL

---

- Silicon Graphics (SGI) revolutionized the graphics workstation by implementing the graphics 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 led to OpenGL (1992)
- It generates 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

# OpenGL Versions

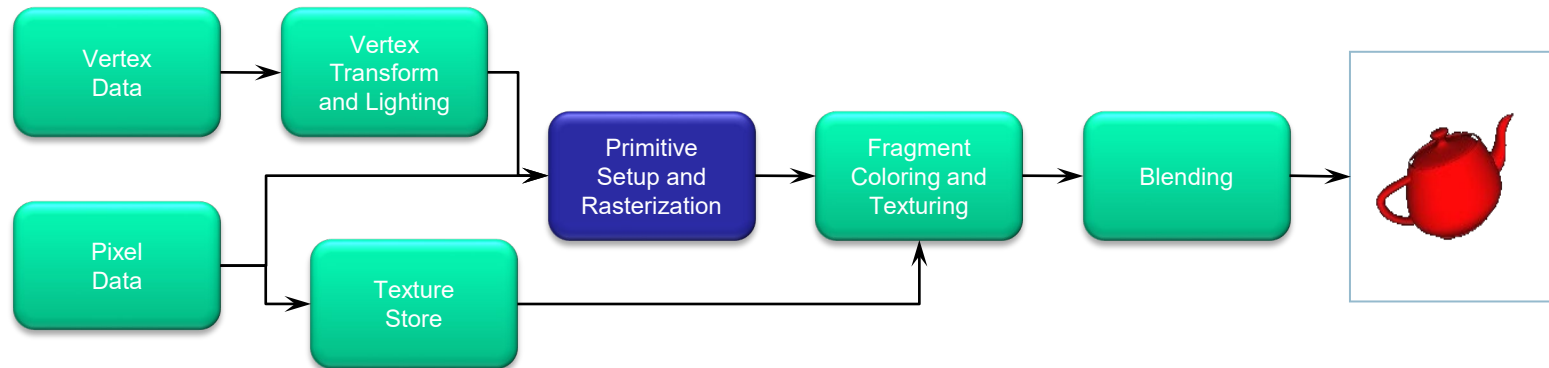
---

Version	Publication Date	Version	Publication Date
OpenGL 1.0	January 1992	OpenGL 3.0	August 2008
OpenGL 1.1	January 1997	OpenGL 3.1	March 2009
OpenGL 1.2	March 1998	OpenGL 3.3	March 2010
OpenGL 1.2.1	October 1998	OpenGL 4.0	March 2010
OpenGL 1.3	August 2001	OpenGL 4.1	July 2010
OpenGL 1.4	July 2002	OpenGL 4.2	August 2011
OpenGL 1.5	July 2003	OpenGL 4.3	August 2012
OpenGL 2.0	September 2004	OpenGL 4.4	July 2013
OpenGL 2.1	July 2006	OpenGL 4.5	August 2014
		OpenGL 4.6	July 2017

# Evolution of the OpenGL Pipeline

---

- OpenGL 1.0 was released on July 1<sup>st</sup>, 1992
- Its pipeline was entirely *fixed-function*
  - the only operations available were fixed by the implementation

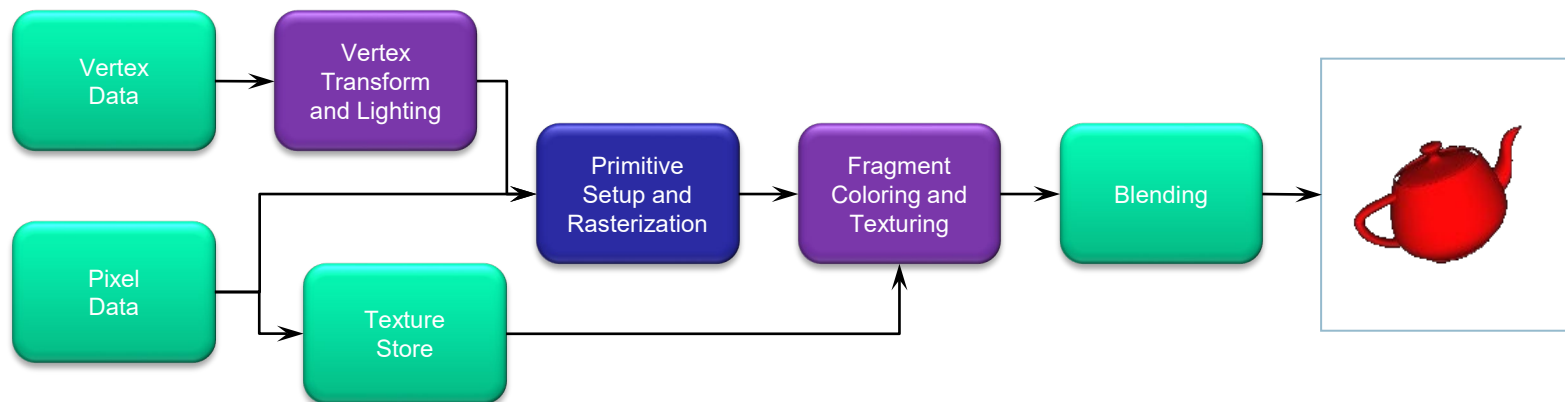


- The pipeline evolved
  - but remained based on fixed-function operation through OpenGL versions 1.1 through 2.0 (Sept. 2004)

# 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



# An Evolutionary Change

---

- OpenGL 3.0 (2008) introduced the *deprecation model*
  - the method used to remove features from OpenGL
- Pipeline remained the same until OpenGL 3.1 (2009)
  - Introduced change in how OpenGL contexts are used

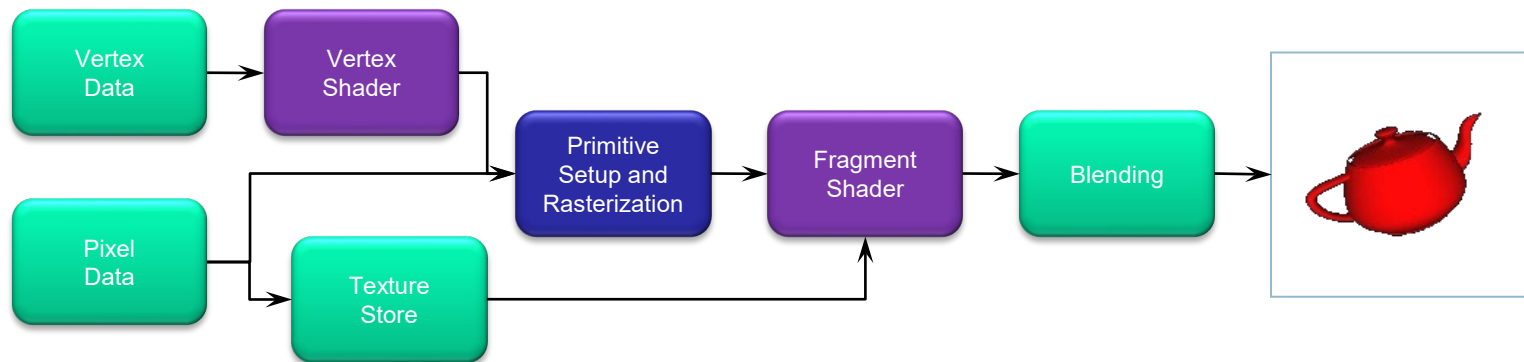
Context Type	Description
Full	Includes all features (including those marked deprecated) available in the current version of OpenGL
Forward Compatible	Includes all non-deprecated features (i.e., creates a context that would be similar to the next version of OpenGL)



# The Exclusively Programmable Pipeline

---

- OpenGL 3.1 removed the fixed-function pipeline
  - programs were required to use only shaders
  - no default shaders
  - app must provide both a vertex and a fragment shader

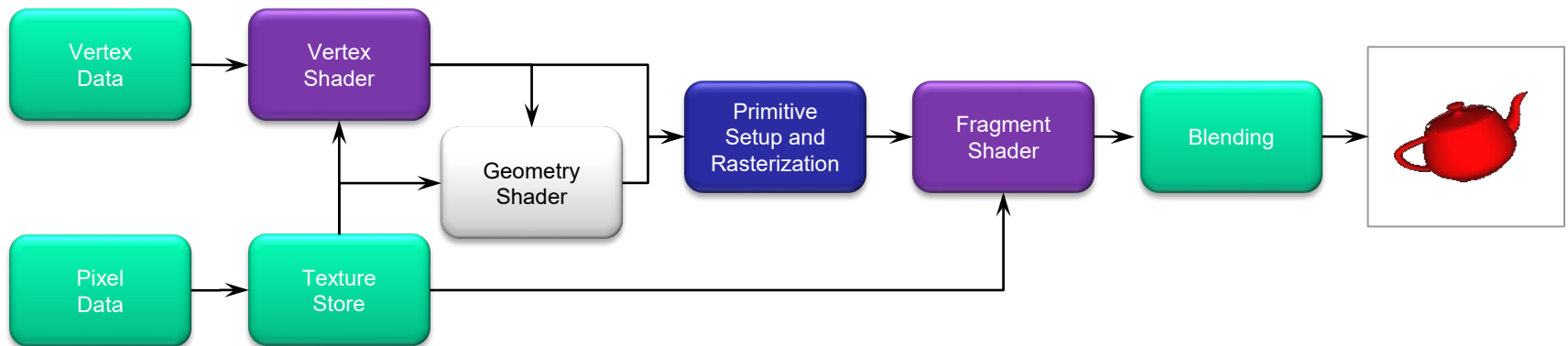


- Additionally, almost all data is GPU-resident
  - all vertex data sent using buffer objects

# 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



# More Evolution: Context Profiles

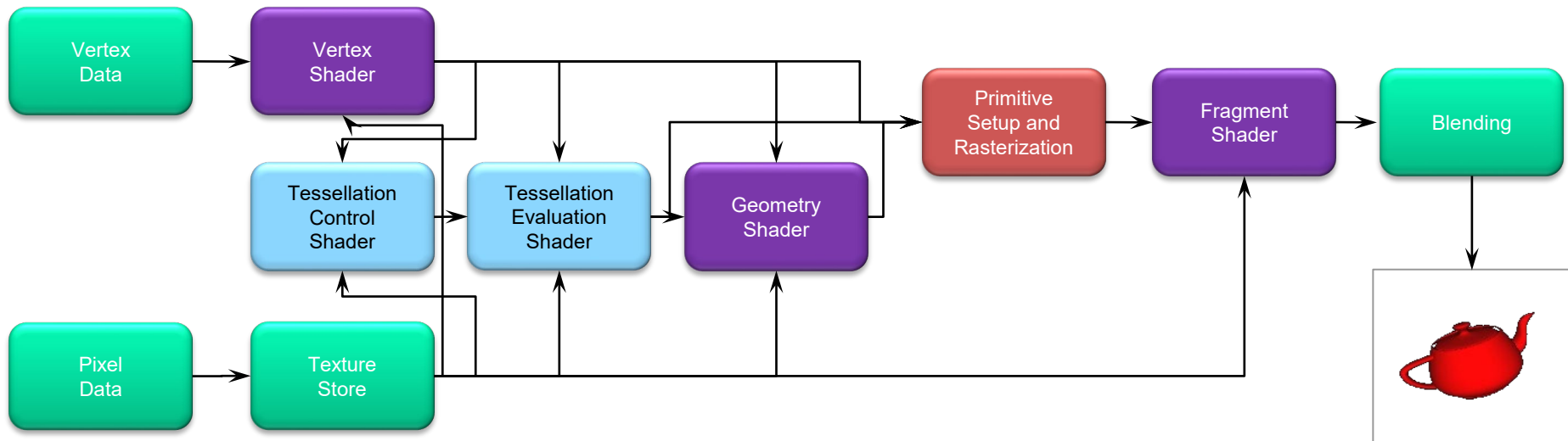
---

- OpenGL 3.2 also introduced *context profiles*
  - profiles control which features are exposed
  - currently two types of profiles: *core* and *compatible*
  - *Core* profile deprecates legacy features (trim version)
  - *Compatibility* profile maintains backwards compatibility with all versions back to version 1.0.

Context Type	Profile	
Full	core	All features of the current release
	compatible	All features ever in OpenGL
Forward Compatible	Core	All non-deprecated features
	Compatible	Not supported

# 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 (July 2017)



# Other Shader-Based Versions

---

- OpenGL ES

- Designed for embedded and hand-held devices such as cell phones and tablets
- Version 1.0 (2003) based on OpenGL 2.1
- Version 2.0 (2007) based on OpenGL 3.1
- Version 3.0 (2012) based on OpenGL 4.3
- Version 3.2 (August 2015)

- WebGL

- Javascript implementation of ES 2.0
- Runs on most recent browsers