

# CPSC 314 Computer Graphics

Dinesh K. Pai

A first look at OpenGL, and Geometry

Many slides courtesy of Min Hyuk Kim, KAIST and Steven Gortler, Harvard

## Preliminaries

---

- Connect, Piazza ok?
- Pickup prerequisite letters

## Today

---

- What are WebGL/OpenGL?
- Three.js?

## What is OpenGL/WebGL?

---

- OpenGL = Open Graphics Library
  - An open industry-standard API for hardware accelerated graphics drawing
  - Implemented by graphics-card vendors
  - Maintained by the Khronos group
- OpenGL ES = Embedded Systems version of OpenGL with reduced functions
- WebGL 1.0 is based on OpenGL ES 2.0, accessible from JavaScript
- Same underlying graphics architecture

4

## What is OpenGL?

---

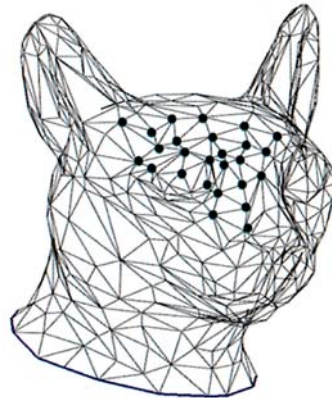
- Pros & Cons:
  - + Full specification freely available
  - + Everyone can use it
  - + Can use it anywhere (Windows, Linux, Mac, BSD, Mobile phones, Web-pages, ...)
  - + Long-term maintenance for older applications
  - + New functionality usually available earlier through Extensions
  - - Inclusion of Extensions to core may take longer
  - ? Game-Industry

5

## OpenGL Pipeline

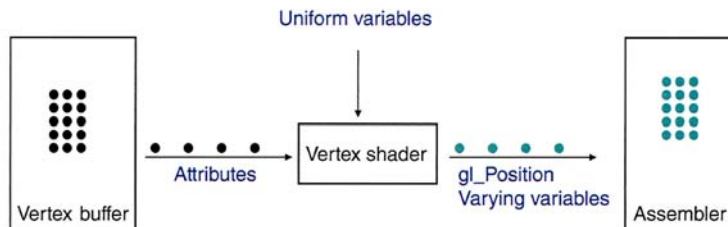
---

- Reference:  
Textbook Chapter 1
- Shapes are “discretized”  
into primitives:  
triangles, line segments, ...
- We’ll focus on triangles most of the time



6

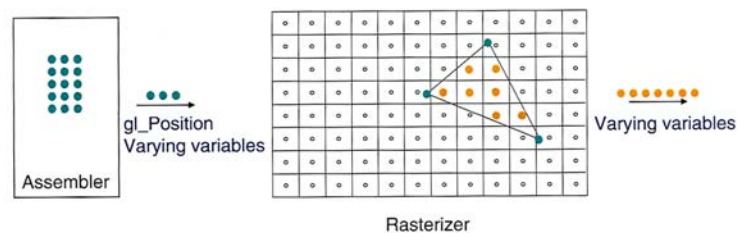
## OpenGL Pipeline: Vertex Shader



- Vertices are stored in a vertex buffer.
- When a draw call is issued, each of the vertices passes through the vertex shader
- On input to the vertex shader, each vertex (black) has associated attributes.
- On output, each vertex (cyan) has a value for `gl_Position` and for its varying variables.

7

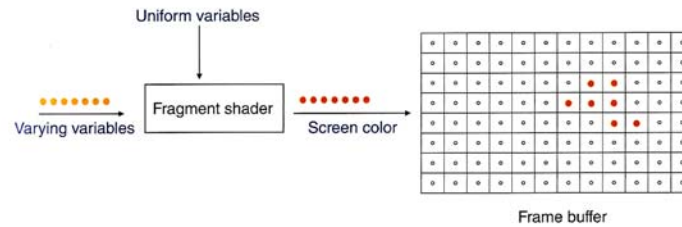
## OpenGL Pipeline: Rasterization



- The data in `gl_Position` are used to place the three vertices of the triangle on a virtual screen.
- The rasterizer figures out which pixels (orange) are inside the triangle and interpolates the varying variables from the vertices to each of these pixels.

8

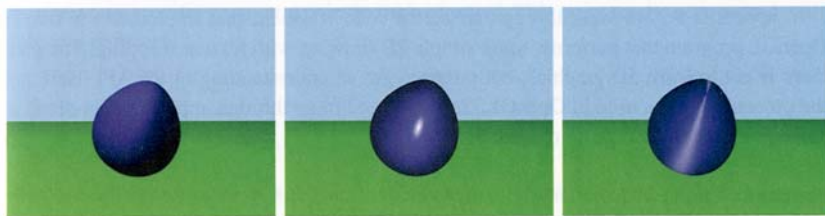
## OpenGL Pipeline: Fragment Shader



- Each pixel (orange) is passed through the fragment shader, which computes the final color of the pixel (pink).
- The pixel is then placed in the framebuffer for display.

9

## OpenGL Pipeline: Fragment Shader

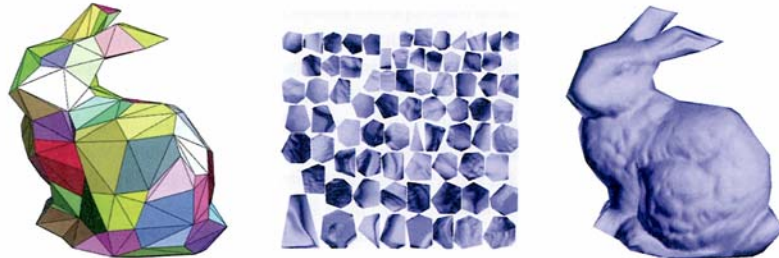


- By changing the fragment shader, we can simulate light reflecting off of different kinds of **materials**.

10

## Texture Mapping

---



- A simple geometric object described by a small number of triangles.
- Details stored in an auxiliary image called a texture.
- Parts of the texture are glued onto each triangle giving a more complicated appearance.

11

## A brief look at Three.js

---

- A high level library that can use WebGL for rendering
  - Can also use the basic HTML5 canvas for simple things
- Setup is much easier compared to WebGL
- Implements “scene” and “mesh” abstractions
- Mesh  $\cong$  geometry + material properties
  - Warning: this usage of “mesh” is non-standard
- Scene contains a hierarchy of mesh objects
- Render a scene using a Camera

## Demo

<http://mrdoob.com/projects/htmleditor/>

## Summary

---

- What is OpenGL/WebGL?
  - A software interface that allows a programmer to communicate with the graphics hardware
  - A programming interface for rendering 2D and 3D graphics
  - A cross-language multi-platform API for computer graphics
- What is Three.js
  - A high level JavaScript library that provides easy setup and access to WebGL