

## EECE 478: Computer Graphics

Graphics Systems

---

---

---

---

---

---

---

---

## OpenGL API

- Programmer's interface
  - in C/C++
  - function calls for sending commands to rendering pipeline
  - *calls match mathematics, geometry and rendering hardware well*
  - 3D model usable for 2D

---

---

---

---

---

---

---

---

## 3D Drawing Model

- Functions for
  - Objects
  - Viewer
  - Light sources
  - Material properties
- Perfect match for Synthetic Camera

---

---

---

---

---

---

---

---

## Objects: Polygon

- Object faces are polygons
- Prescribed by list of vertices

```
glBegin (GL_POLYGON);  
  glVertex3f (0.0, 0.0, 0.0);  
  glVertex3f (0.0, 1.0, 0.0);  
  glVertex3f (0.0, 0.0, 1.0);  
glEnd ();
```

---

---

---

---

---

---

---

---

## Objects: Faces

- glBegin/glEnd pair defines surface
- GL\_POLYGON: Any number of vertices
  - assumes convex
- Other primitives for optimization
  - GL\_TRIANGLES, GL\_QUADS
  - GL\_TRIANGLE\_STRIP
- Other primitives for points, lines
  - GL\_LINE\_STRIP, GL\_POINTS

---

---

---

---

---

---

---

---

## Viewers: Camera Parameters

- Camera position
  - center of projection (COP)
- Orientation
  - direction it is pointing
  - which way is up
- Focal length
  - distance of projection plane from COP
- Film plane
  - size of projection surface

---

---

---

---

---

---

---

---

## Viewer: Camera Parameters

- External parameters:
  - define camera wrt. outside world
  - COP and orientation
- Internal parameters:
  - don't change when just moving camera
  - focal length and field of view

---

---

---

---

---

---

---

---

## Viewer: Camera API

- External parameters  $\Rightarrow$  *view* matrix
- Internal parameters  $\Rightarrow$  *projection* matrix
- OpenGL has functions to set these:
  - set matrix directly
  - define COP, orientation, focal length and FOV
  - *look at* a point from a COP

---

---

---

---

---

---

---

---

## Light Sources

- Location
  - point light sources
- Intensity
- Color
- Directionality
  - omnidirectional
  - spotlight

---

---

---

---

---

---

---

---

## Material Properties

- Ambient reflection
- Diffuse reflection
- Specular reflection
- Emissivity
- Smoothness

---

---

---

---

---

---

---

---

## Graphics Architecture

Model  $\Rightarrow$  code  $\Rightarrow$  API  $\Rightarrow$  graphics pipeline

- Data flow architecture
- Pipeline is connected set of stages
- Each stage performs simple transform on inputs to create output
- Graphics pipeline takes vertex geometry and produces frame buffer pixels

---

---

---

---

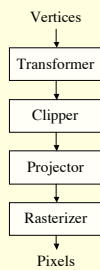
---

---

---

---

## Graphics Pipeline



---

---

---

---

---

---

---

---

## Transformer

Vertices  $\Rightarrow$  Vertices

- Change of coordinate systems
  - model coordinates to world coordinates
  - world coordinates to camera coordinates
- Simple matrix multiplications
  - purely linear
  - very parallelizable

---

---

---

---

---

---

---

---

## Clipper

Vertices  $\Rightarrow$  fewer Vertices

- Need not process that which will not be seen
- Different kinds of clippers
  - clipping rectangle of projection plane
  - windows

---

---

---

---

---

---

---

---

## Projector

3D Vertices  $\Rightarrow$  2D Vertices

- Near final stage
- Perspective or orthogonal
- Linear transform + division

---

---

---

---

---

---

---

---

## Rasterizer

2D vertices  $\Rightarrow$  pixels

- End of pipeline
- Primitives for drawing 2D points, lines and triangles in color

---

---

---

---

---

---

---

---