

CMSC 435/634

Interactive Rendering
Marc Olano

UMBC
AN HONORS
UNIVERSITY
IN MARYLAND

Classifying Graphics by Rendering Speed

- How long to generate an image
- Image *frames* may be precomputed and displayed at a different rate later

UMBC
AN HONORS
UNIVERSITY
IN MARYLAND

Realistic

- Take as long as necessary
- Often only rendering one frame



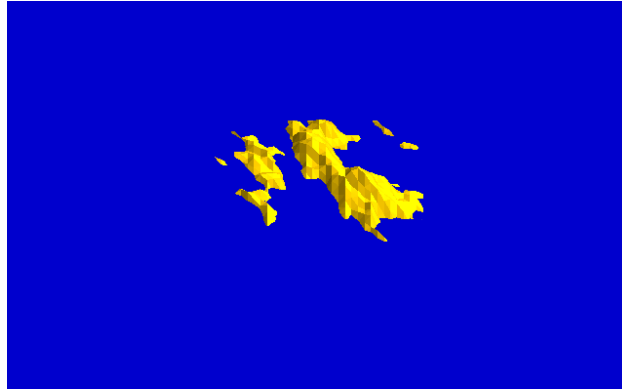
Production/Movies

- Limited by production schedule
- Display 24-60 frames per second



Interactive

- Application & rendering at display speed



Frame Rate

- New image how often?
 - Slow: seconds per frame
 - Fast: frames per second
- Some rates:
 - 1-2 sec/frame: barely able to interact
 - 12 fps: start to fuse motion
 - 24 fps: film
 - 30-60 fps: TV
 - up to ~100 fps: computer display

Rendering vs. Display Frame Rate

- *Latency*: time between action and result of that action
- Render slower than display rate
 - Prediction or long latency
 - Double buffer; double / triple frame
 - Ghosting
 - Pipeline
- Render much faster than display
 - Latency < frame time
 - Common in games

Anatomy of an Interactive Application

```
setup
do {
  render frame
  read input
  update state
} until done
```

Event Loop

- More sophisticated application

setup

```
do {
```

```
    wait for event
```

```
    while (events in queue)
```

```
        process event
```

```
} until done
```

Why Events?

- Receive and queue events asynchronously
- Event callback
 - User code called to respond to event
- Event response can queue new events
 - Including a *Render event*: not every event requires re-rendering
- Similar model used by windowing systems

Some Common Events

- From GLUT
 - Display
 - Key press
 - Mouse button, mouse motion
 - Menu, mouse enter/leave window
 - Reshape window, window visibility change
 - Idle

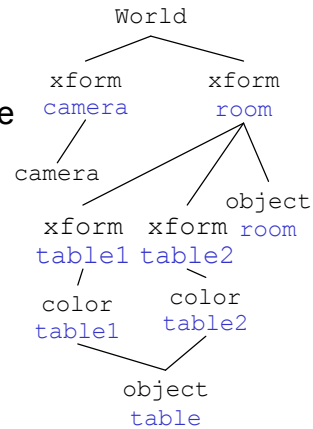
What About Rendering?

- Same models, rerendered every frame with minor changes
- Persistent data structure for scene
 - Other events modify data structure
 - Display event renders as it exists now

Scene Graph

- Tree / DAG representation of scene

- Interior nodes
 - transforms, appearance
- Leaf nodes
 - geometry

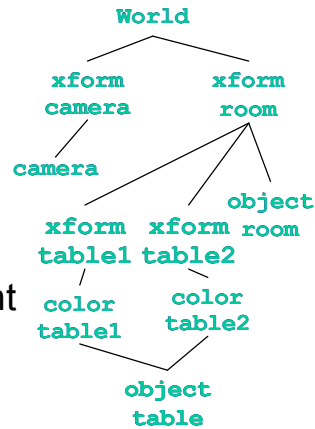


Scene Graph Details

- Each node has
 - Node type
 - Children
 - Auxiliary node-type specific data
- Find locations by name or pointer
 - Update data
 - *Rewrite* sections of graph

Scene Graph Rendering

- Traverse Graph
- Display actions
 - 1-handful GL calls
- Options
 - action does traversal
 - pre & post actions
 - SG state management



Scene Graph Traversal

- Not just for rendering
- Save
 - Rather than render, print to file
- Optimize
 - Sort by state changes
 - Sort by distance for transparency
- Produce intermediate *display list*

Scene Graph Layout

- All about the coordinate spaces
 - World
 - Room
 - UserN
 - Leg, Arm, Head, ...
 - Lights
 - ...