CMSC 491G/691G

Computer Graphics for Games

Marc Olano

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 visibilty change
  - Idle, Timer
- Networking
  - Message received/sent, Synchronize

Input Devices

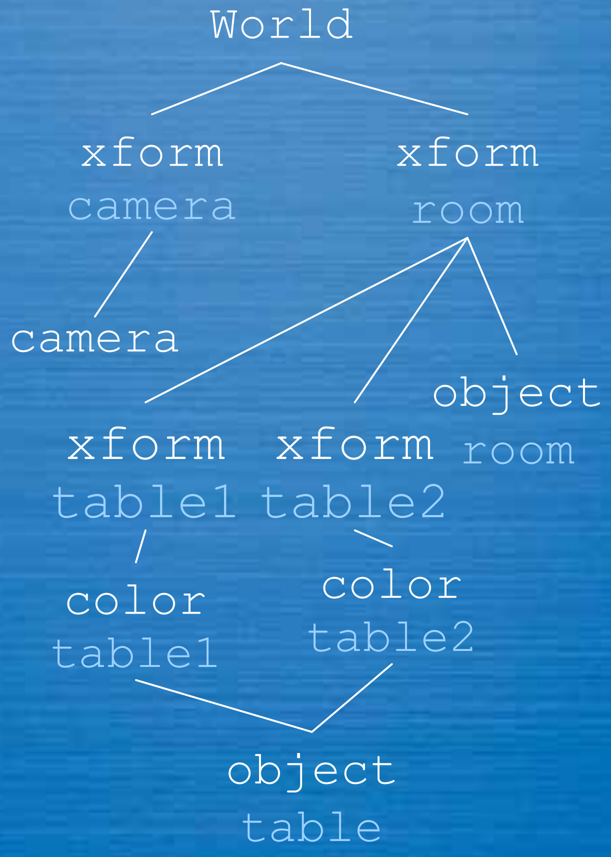
- Keyboard, Button devices
- Mouse, Joystick, Wiimote
- Touch screen
- Magnetic / Ultrasonic Tracker
- Video

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



World

xform
camera

xform
room

camera

object
room

xform
table1

xform
table2

color
table1

color
table2

object
table

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