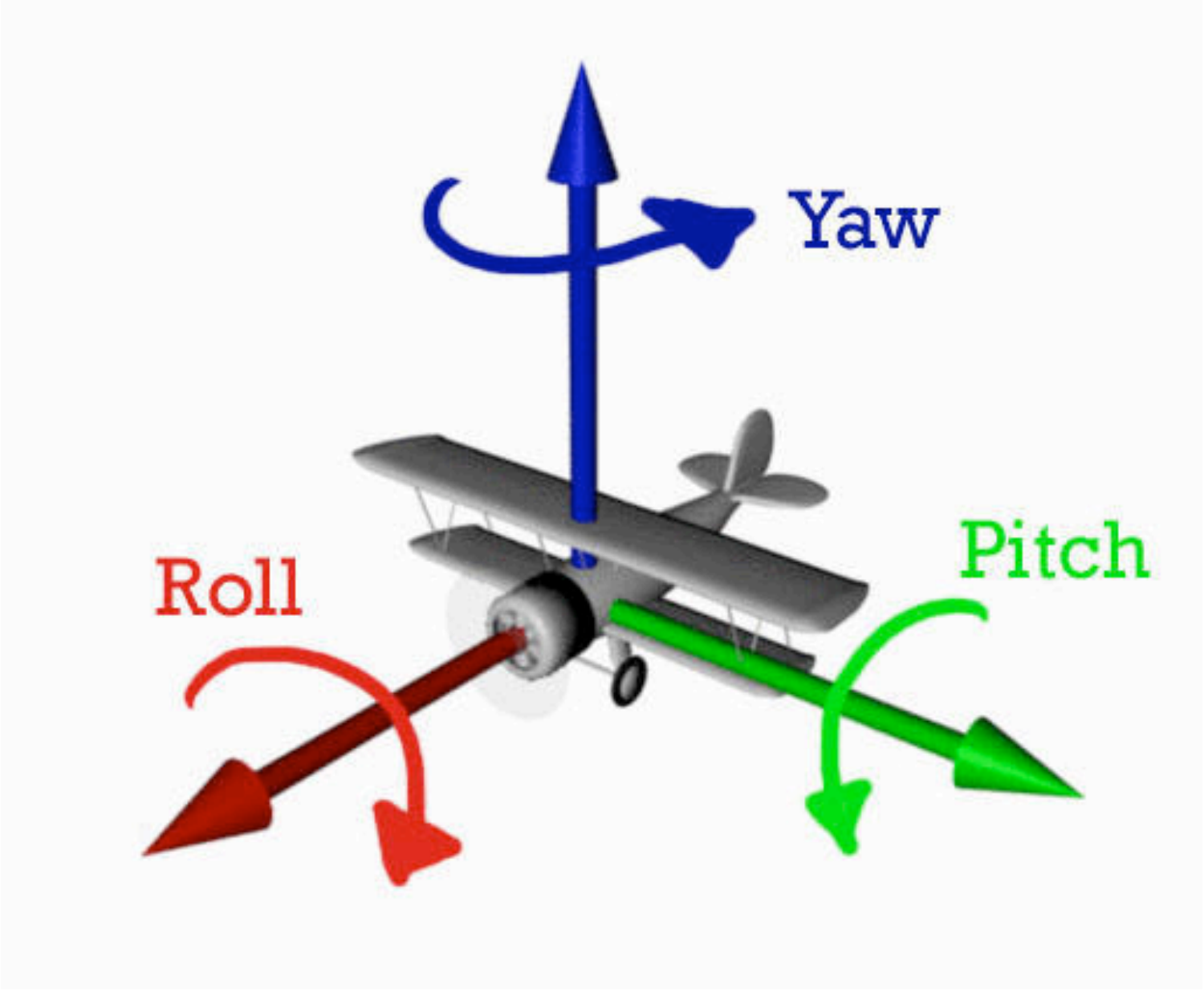
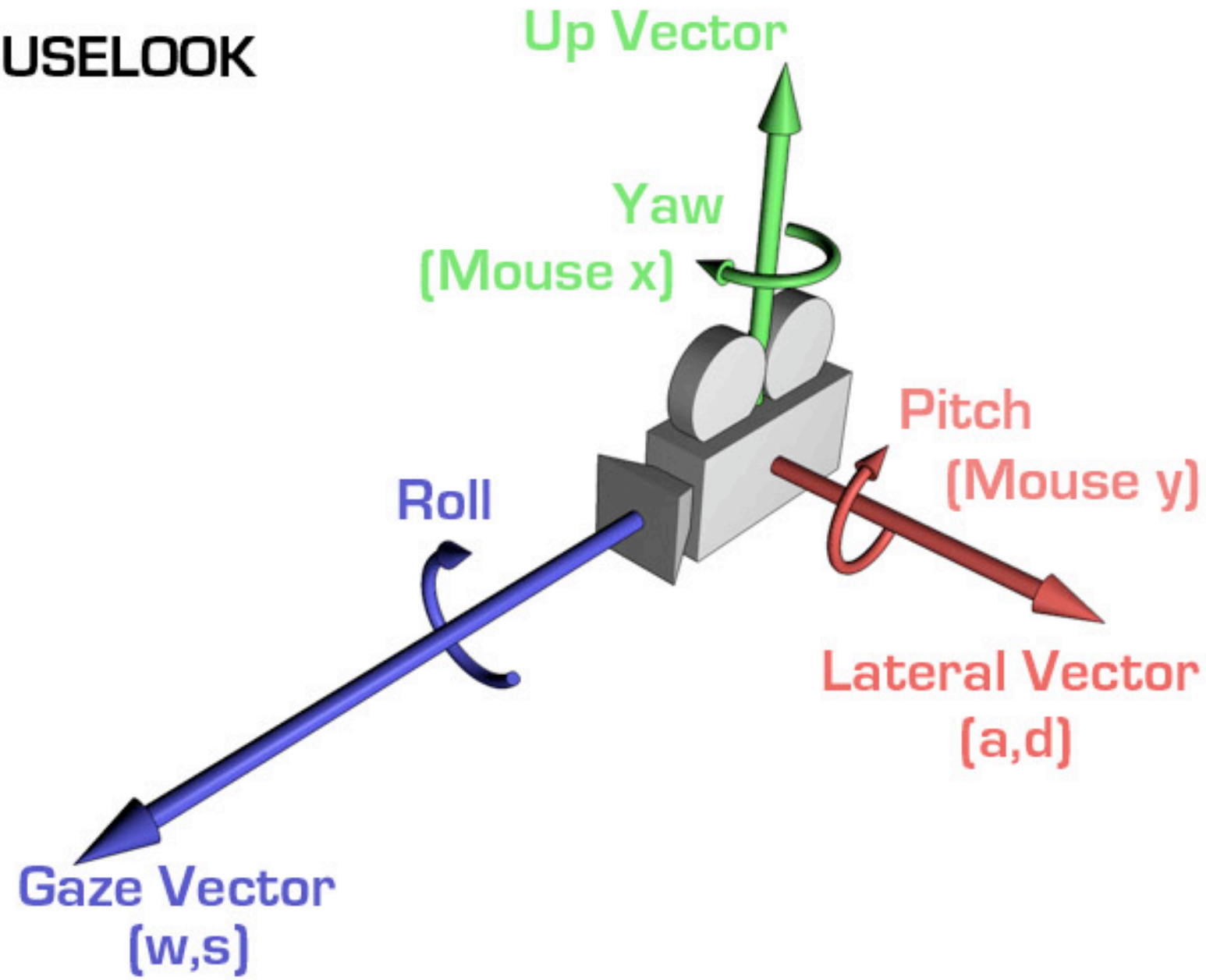
Project 2: Navigation

- five ways to navigate
  - Absolute Rotate/Translate Keyboard
  - Absolute Lookat Keyboard
    - move wrt global coordinate system
  - Relative Rolling Ball Mouse
    - spin around with mouse, as discussed in class
  - Relative Flying
  - Relative Mouselook
    - use both mouse and keyboard, move wrt camera
- template: colored ground plane

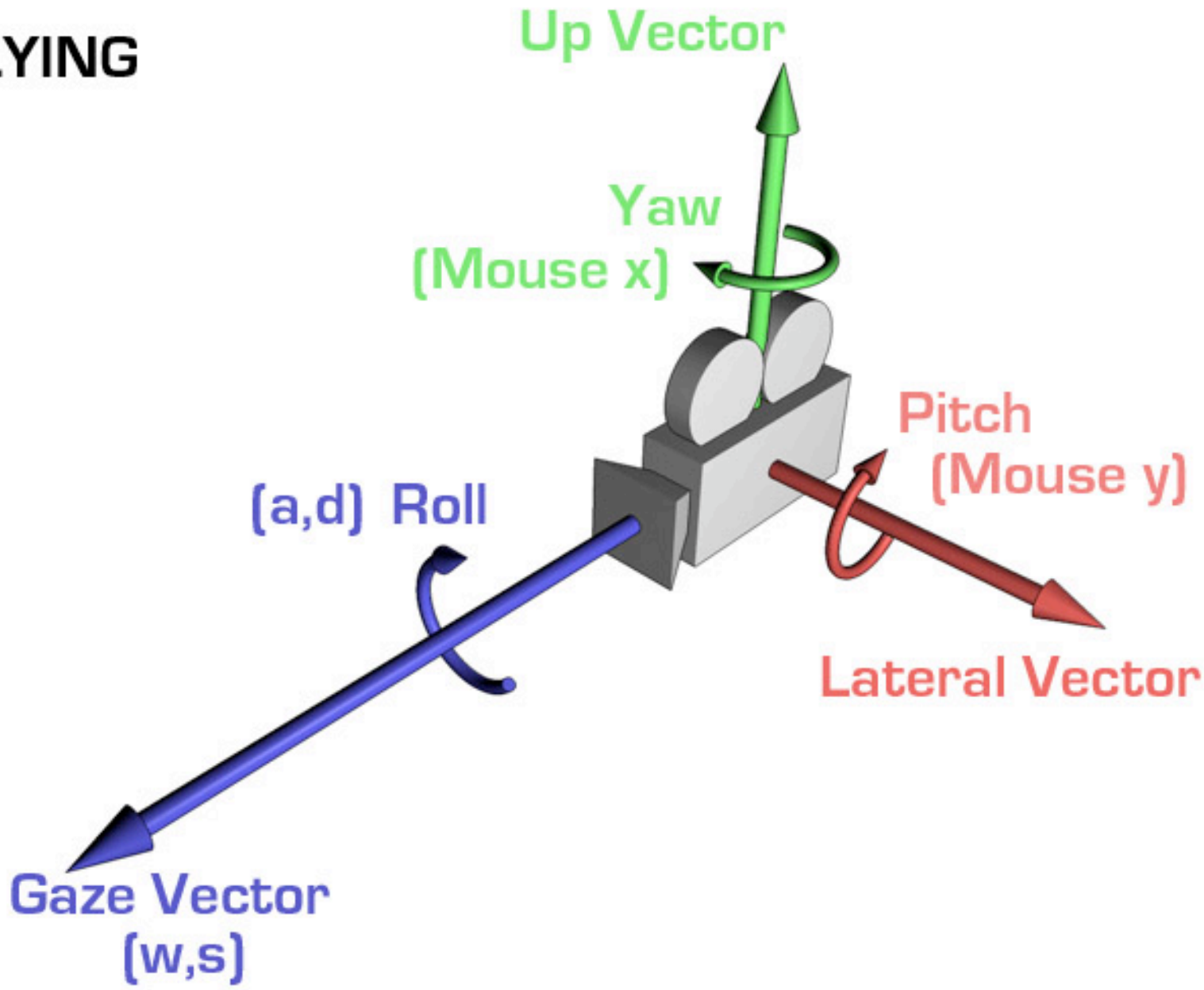
Roll/Pitch/Yaw



MOUSELOOK


Up Vector
Yaw
(Mouse x)
Pitch
(Mouse y)
Roll
Lateral Vector
(a,d)
Gaze Vector
(w,s)

FLYING


Up Vector
Yaw
(Mouse x)
Pitch
(Mouse y)
(a,d) Roll
Lateral Vector
Gaze Vector
(w,s)

Demo

Hints: Viewing

- don’t forget to flip y coordinate from mouse
  - window system origin upper left
  - OpenGL origin lower left

- all viewing transformations belong in modelview matrix, not projection matrix

Hint: Incremental Relative Motion

- motion is wrt current camera coords
  - maintaining cumulative angles wrt world coords would be difficult
  - computation in coord system used to draw previous frame (what you see!) is simple
    - at time k, want p' = $I_k I_{k-1} \ldots I_5 I_4 I_3 I_2 I_1 Cp$
    - thus you want to premultiply: p'=ICp
    - but postmultiplying by new matrix gives p'=CIp
  - OpenGL modelview matrix has the info! sneaky trick:
    - dump out modelview matrix with **`glGetDoublev()`**
    - wipe the stack with **`glIdentity()`**
    - apply incremental update matrix
    - apply current camera coord matrix
  - be careful to leave the modelview matrix unchanged after your display call (using push/pop)

Caution: OpenGL Matrix Storage

- OpenGL internal matrix storage is columnwise, not rowwise

```
a   e   i   m
b   f   j   n
c   g   k   o
d   h   l   p
```

  - opposite of standard C/C++/Java convention
  - possibly confusing if you look at the matrix from glGetDoublev()!