

Shaoting Zhang
Office Hours: Wed 5-7pm
Office: Hill-418
Webpage: <http://paul.rutgers.edu/~shaoting/fall08.html>

1 Review: math, java, etc.

dot product

$$a \cdot b = a^T b$$

and

$$a \cdot b = |a||b| \cos \theta$$

where

$|a|$ and $|b|$ denote the length of a and b

θ is the angle between them.

$$\text{Thus } \cos \theta = \frac{a \cdot b}{|a||b|}$$

java.vecmath.Vector3d: double dot(Vector3d v1)

returns the dot product of this vector and vector v1.

cross product

$$a \times b = a_2 b_3 - a_3 b_2, a_3 b_1 - a_1 b_3, a_1 b_2 - a_2 b_1$$

$$a \times b = |a||b| \sin \theta \hat{n}$$

java.vecmath.Vector3d: void cross(Vector3d v1, Vector3d v2)

Sets this vector to the vector cross product of vectors v1 and v2

Pitfall in Java: call by value or call by reference

Java is a call-by-value language, but since most Java expressions are references to anonymous objects, it frequently displays call-by-reference semantics without the need for any explicit reference syntax.

If a function receives an object, it receives the REFERENCE to this object. This means that the function can modify this object, what will be seen by its caller.

Pitfall in Java: Type casting

Wrong:

```
double ratio;
```

```
int height = 150;
```

```
int width = 250;
```

```
ratio = height / width;
```

Correct:

```
ratio = (double)height / (double)width;
```

Pitfall in OpenGL: Improperly scaling normals for lighting

We can use `glNormal3f()` to specify normal vectors that indicate the orientation of the surface at each vertex. For OpenGL's lighting equations to operate properly, the assumption OpenGL makes by default is that the normals passed to it are vectors of length 1.0.

Pitfall in OpenGL: Remember your matrix mode

OpenGL has a number of 4 by 4 matrices that control the transformation of vertices, normals, and texture coordinates. The core OpenGL standard specifies the modelview matrix, the projection matrix, and the texture matrix. A common pitfall is forgetting the current setting of the matrix mode and performing operations on the wrong matrix stack. If later you assume the matrix mode is set to a particular state, you both fail to update the matrix you intended and screw up whatever the actual current matrix is.

Check out the course webpage for more information, like how to configure and use JOGL, how to submit assignments and check grades, OpenGL manual, etc.