

Principles of Computer Game Design and Implementation

Lecture 10

Quiz

We already learned

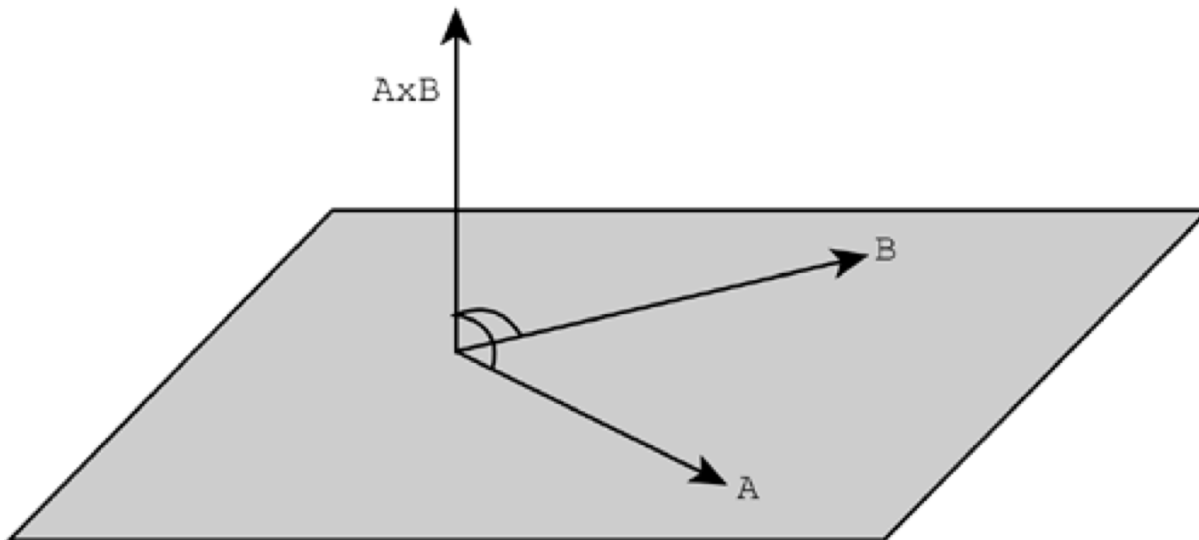
- Translation
- Movement
- Rotation
- Dot product

Outline for today

- Cross product
- Explanation of first assignment

The Cross Product

- The cross product of two vectors is a vector
 - Only applies in three dimensions
 - *The cross product is perpendicular to both vectors*
 - The cross product between two parallel vectors is the zero vector $(0, 0, 0)$



The Cross Product

- The cross product between \mathbf{V} and \mathbf{W} is

$$\mathbf{V} = \begin{pmatrix} x_v & y_v & z_v \end{pmatrix} \quad \mathbf{W} = \begin{pmatrix} x_w & y_w & z_w \end{pmatrix}$$

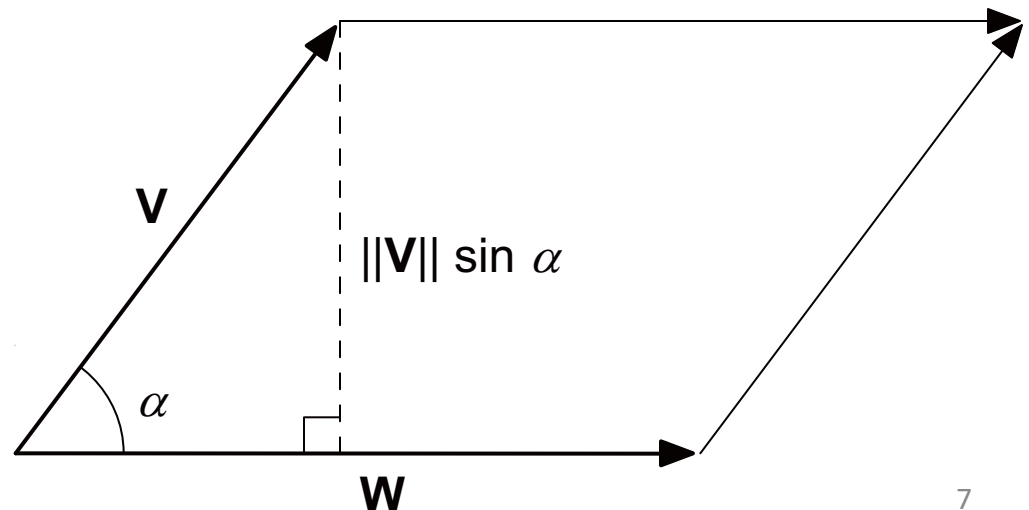
$$\mathbf{V} \times \mathbf{W} = \begin{pmatrix} y_v z_w - z_v y_w, & z_v x_w - x_v z_w, & x_v y_w - y_v x_w \end{pmatrix}$$

The Cross Product

- The cross product satisfies the trigonometric relationship

$$\|\mathbf{V} \times \mathbf{W}\| = \|\mathbf{V}\| \|\mathbf{W}\| \sin \alpha$$

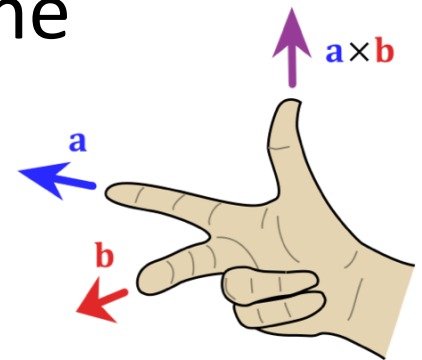
- This is the area of the parallelogram formed by \mathbf{V} and \mathbf{W}



The Cross Product

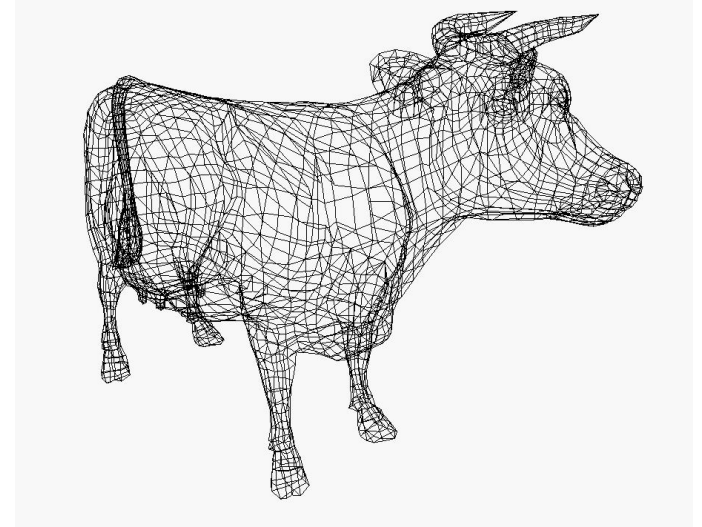
- Cross products obey the right hand rule
 - If first vector points along right index finger, and second vector points along middle finger,
 - Then cross product points out of right thumb
- Reversing order of vectors negates the cross product:

$$\mathbf{W} \times \mathbf{V} = -\mathbf{V} \times \mathbf{W}$$



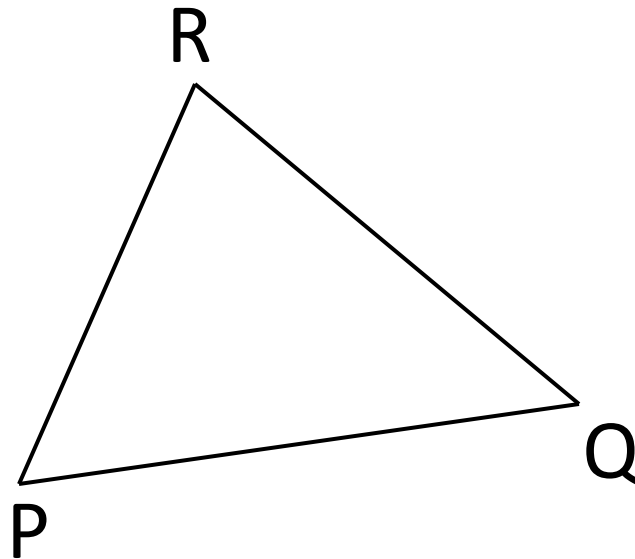
Uses: Face Normal

- Complex 3D models are build from *polygons* mostly *triangles*
- When determining the luminance of a triangle, we need to know the angle between the plain in which it lays and the light beam.



Example: Normal of a Triangle

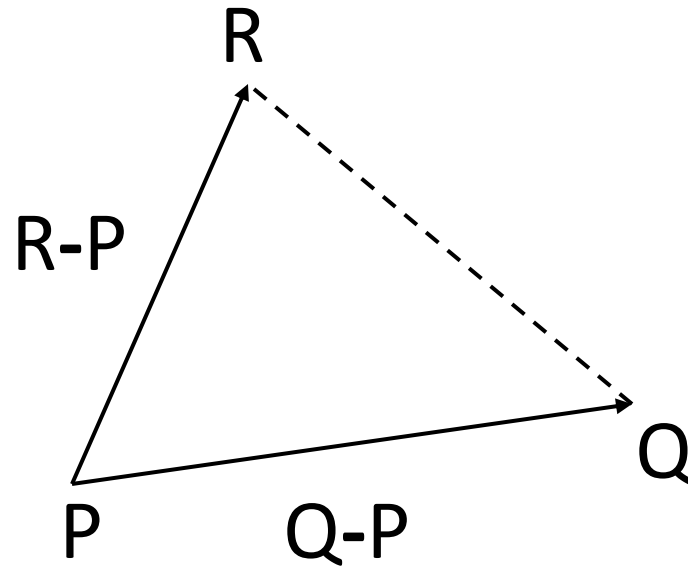
- Find the unit length normal of the triangle defined by 3D points P, Q, and R



Example: Normal of a Triangle

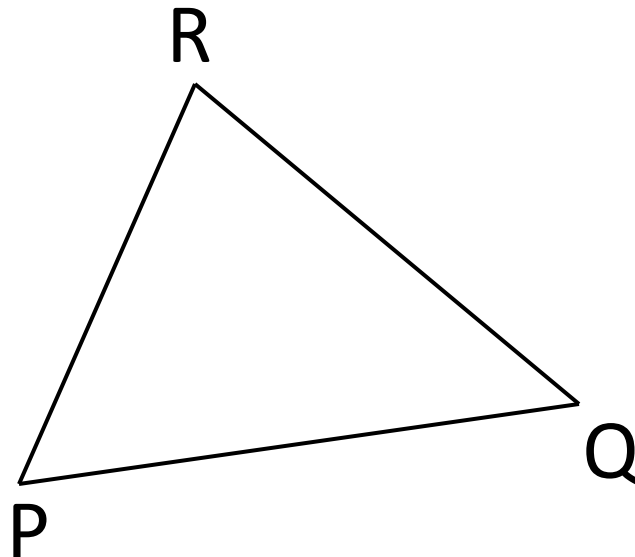
$$\mathbf{n}^* = (R - P) \times (Q - P)$$

$$\mathbf{n} = \frac{\mathbf{n}^*}{|\mathbf{n}^*|}$$



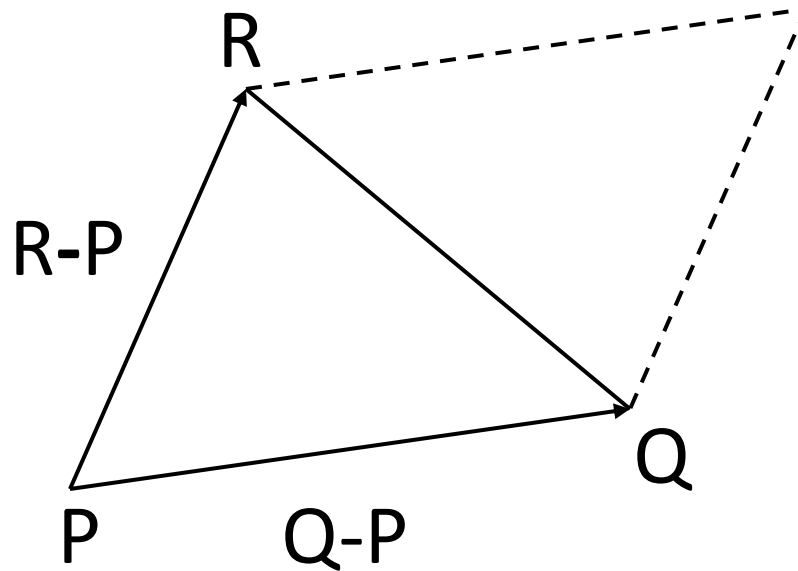
Example: Area of a Triangle

- Find the area of the triangle defined by 3D points P, Q, and R



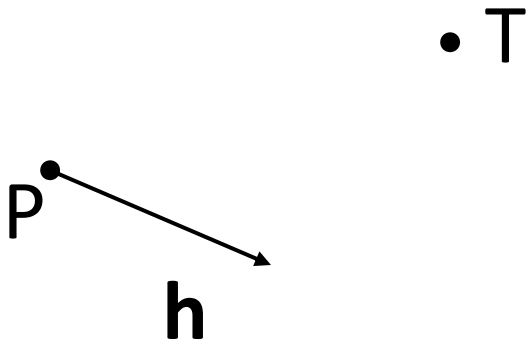
Example: Area of a Triangle

$$area = \frac{1}{2} |(Q - P) \times (R - P)|$$



Example: Alignment to Target

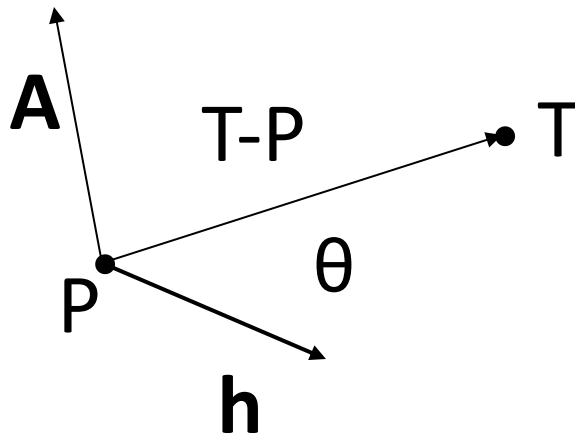
- An object is at position P with a unit length heading of \mathbf{h} . We want to rotate it so that the heading is facing some target T . Find a unit axis \mathbf{A} and an angle θ to rotate around.



Example: Alignment to Target

$$\mathbf{a} = \frac{\mathbf{h} \times (T - P)}{|\mathbf{h} \times (T - P)|}$$

$$\theta = \cos^{-1} \left(\frac{\mathbf{h} \cdot (T - P)}{|(T - P)|} \right)$$



jME Example

```
Vector3f u = new Vector3f(x, y, z).normalize();
```

```
Arrow yArrow = new Arrow(Vector3f.UNIT_Y);
```

```
gyArrow = new Geometry("Y", yArrow);
```

```
...
```

```
rootNode.attachChild(gyArrow);
```

```
Vector3f axis = Vector3f.UNIT_Y.cross(u);
```

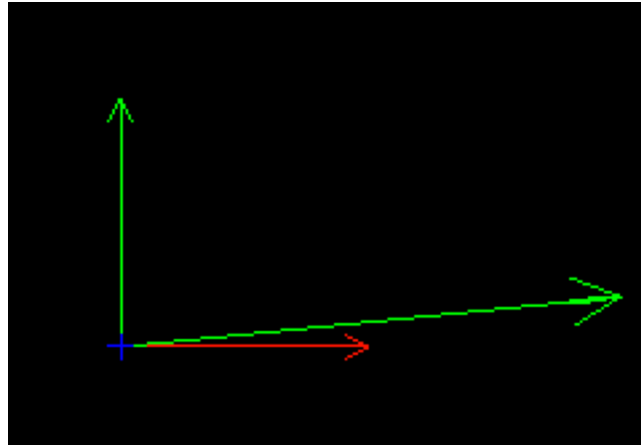
```
float angle = FastMath.acos(Vector3f.UNIT_Y.dot(u));
```

```
Quaternion q = new Quaternion();
```

```
q.fromAngleAxis(angle, axis);
```

```
gyArrow.setLocalRotation(q);
```


It Works



(I've added the AxisRods to the picture to show the reference point)

Gradual Rotation: simpleUpdate

```
float curT = t.getTimeInSeconds();
if (curT < (startT + timeoutR))
{
    float currentAngle =
        (startAngle + ((curT-startT)/
                       timeoutR)*targetAngle);
    Quaternion q = new Quaternion();
    currentAngle += tpf;
    q.fromAngleAxis(currentAngle, axisV);
    gyArrow.setLocalRotation(q);
}
```

Conclusion

- Dot and cross products are both used in 3D graphics
- Dot product is a number
- Cross product is a vector

$$\mathbf{V} = \begin{pmatrix} x_v & y_v & z_v \end{pmatrix} \quad \mathbf{W} = \begin{pmatrix} x_w & y_w & z_w \end{pmatrix}$$

$$\mathbf{V} \cdot \mathbf{W} = \begin{pmatrix} x_v x_w + y_v y_w + z_v z_w \end{pmatrix}$$

$$\mathbf{V} \times \mathbf{W} = \begin{pmatrix} y_v z_w - z_v y_w, & z_v x_w - x_v z_w, & x_v y_w - y_v x_w \end{pmatrix}$$