Principles of Computer Game Design and Implementation

Lecture 9

We already knew

- Translation
- movement
- Rotation

Outline for Today

- Dot Product and its application

- Question: how to convert a world coordinate to a local coordinate

Credits

- J.M. van Verth, L.M. Bishop "Essential Mathematics for Games & Interactive Applications: A Programmer's Guide". Morgan Kaufman Publishers, 2008.
- + Slides

Coordinate Systems

- Until now, we only considered the “screen” coordinates (e.g. 800x600) and “world” coordinates
- On the other hand, we’ve seen that translation and rotation are independent
- Well, how independent are they?

Planets Example Revisited (1)

```
protected void simpleInitGame() {
…
  moon.setLocalTranslation(40, 0, 0);
…
  pivotNode.attachChild(moon);
…
}
```

Planet Example Revisited (2)

```
public void simpleUpdate(float tpf) {
   quat.fromAngleAxis(tpf, axis);
   pivotNode.rotate(quat);
   moon.move(tpf,0,0);
}
```

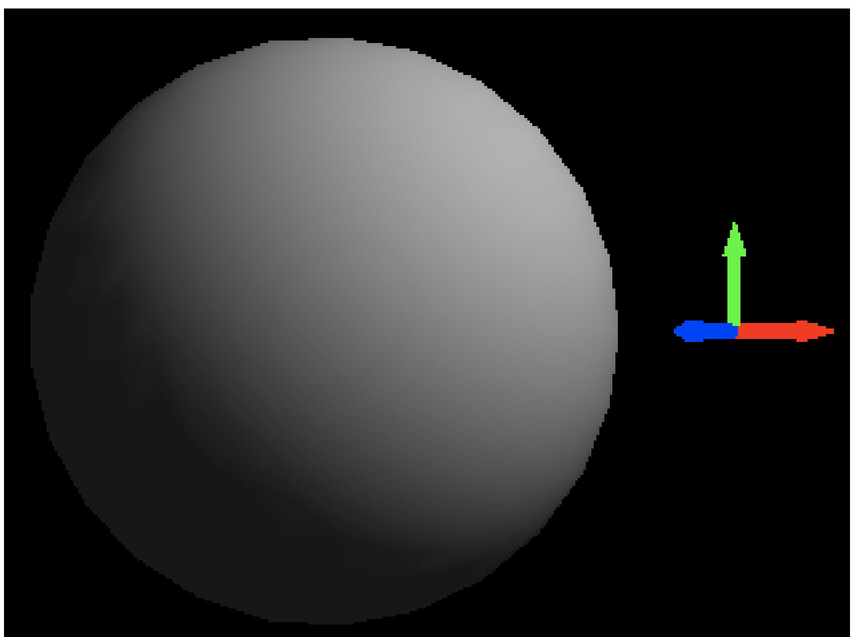
New simpleUpdate()

```
protected void simpleUpdate() {
   if (tpf < 1) {
     angle = angle + (tpf * 1);
     if (angle > 2*FastMath.PI) {
       angle -= 2*FastMath.PI;
     }
   }
   rotQuat.fromAngleAxis(angle, axis);
   pivotNode.setLocalRotation(rotQuat);

 moon.setLocalTranslation(moon.getLocalTransla
 tion().add(tpf*5,0,0));
 }
```

To See It Better

Replace sphere with AxisRods
or Box



Local Coordinate System

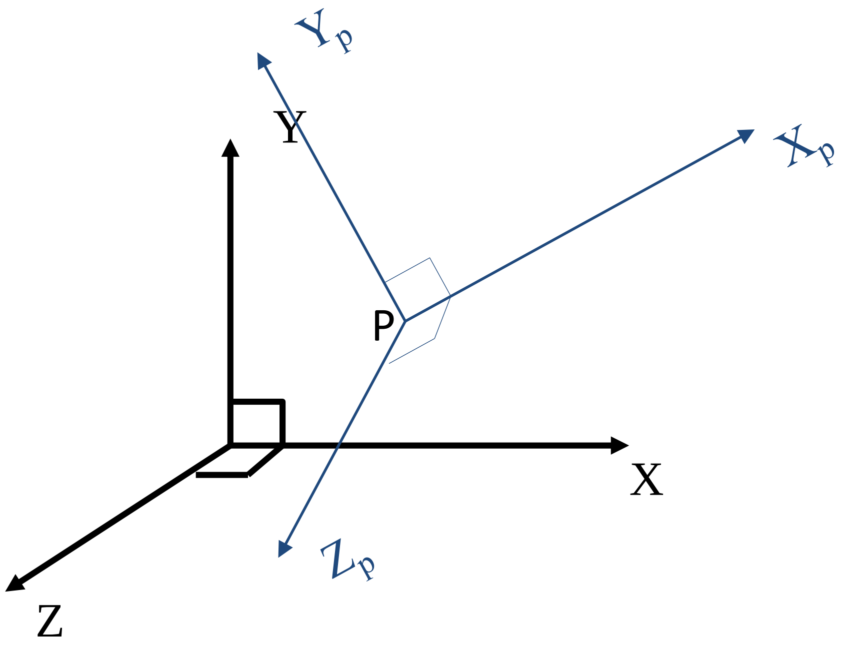
- Every object has it's own "local" coordinate system, which can be placed into the world coordinate system
  - Screen coordinates and world coordinates are local coordinates

Advantages

- Objects can be manipulated “locally”
- Physical interaction is easier to describe
- Lighting of 3D objects is easier to compute

- However, a *transformation* from one coordinate system to another is required

What Are Local Coordinates



Three vectors $\mathbf{X}_p$ $\mathbf{Y}_p$ $\mathbf{Z}_p$ *define* local coordinates at P

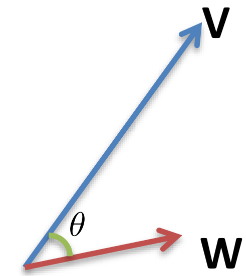
So, we need a way to transform between coordinates

Dot and Cross Products

- Given two vectors, **V** and **W**, there are two product operators
  - **V•W** – a “dot” product
    - A number
    - Used to **project**
  - **V × W** – a “cross” product
    - A vector
    - Used to find normals

Dot Product

- Coordinate-independent definition
  - Given **V** and **W**



$$\mathbf{V} \cdot \mathbf{W} = \|\mathbf{V}\| \cdot \|\mathbf{W}\| \cdot \cos\theta$$

where $\|\,\|$ is the length and
θ is the angle between the vectors

Dot Product

- In coordinates

$$\begin{aligned} \mathbf{V} &= (x_v, y_v, z_v) \\ \mathbf{W} &= (x_w, y_w, z_w) \\ \mathbf{V} \cdot \mathbf{W} &= x_v \cdot x_w + y_v \cdot y_w + z_v \cdot z_w \end{aligned}$$

Uses: Vector Length

- Since $\cos(0) = 1$

$$\mathbf{V} \cdot \mathbf{V} = \|\mathbf{V}\| \cdot \|\mathbf{V}\| \cdot \cos 0 = \|\mathbf{V}\|^2$$

- Hence,

$$\|\mathbf{V}\| = \sqrt{\mathbf{V} \cdot \mathbf{V}}$$

Uses: Measuring Angles

- On the one hand,

$$\mathbf{V} \cdot \mathbf{W} = \|\mathbf{V}\| \cdot \|\mathbf{W}\| \cdot \cos\theta$$

- On the other,

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

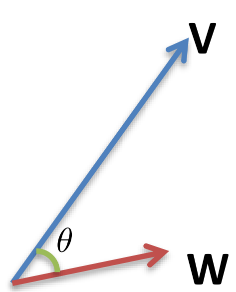
- So,

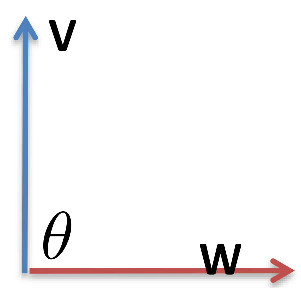
$$\theta = \cos^{-1}\left(\frac{\mathbf{V} \cdot \mathbf{W}}{\|\mathbf{V}\| \cdot \|\mathbf{W}\|}\right)$$

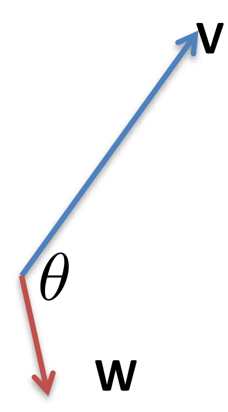
Can be computed from vector coordinates

Uses: Classifying Angle

- Since $\|\mathbf{V}\|$ and $\|\mathbf{W}\|$ are non-negative
  - If $\mathbf{V} \cdot \mathbf{W} > 0$ then angle < 90°
  - If $\mathbf{V} \cdot \mathbf{W} = 0$ then angle = 90°
  - If $\mathbf{V} \cdot \mathbf{W} < 0$ then angle > 90°







Application: Collision Response

- **V** and **W** are speed vectors of two balls
- Three options:

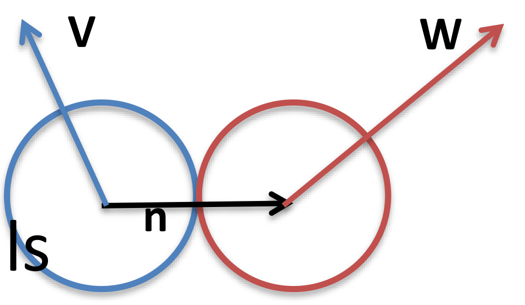
$$(\mathbf{V} - \mathbf{W}) \cdot \mathbf{n} < 0 \qquad \text{Separating}$$

$$(\mathbf{V} - \mathbf{W}) \cdot \mathbf{n} = 0 \qquad \text{Resting}$$

$$(\mathbf{V} - \mathbf{W}) \cdot \mathbf{n} > 0 \qquad \text{Colliding}$$

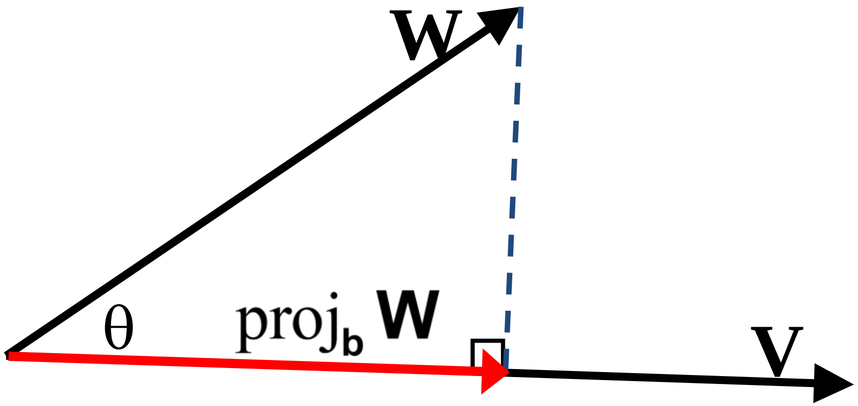
V – **W** is the *relative velocity*

n is the vector between centres of balls



Uses: Projection

- Suppose want to *project* **W** onto **V**

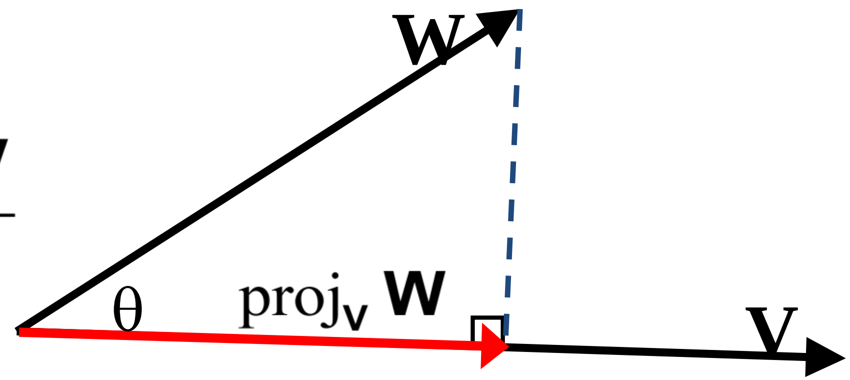


- Is part of **W** pointing along **V**
- Represented as

Uses: Projection

- From trig

$$\left\|\text{proj}_{\mathbf{V}}\,\mathbf{W}\right\| = \left\|\mathbf{W}\right\|\cos\theta = \frac{\mathbf{W}\bullet\mathbf{V}}{\left\|\mathbf{V}\right\|}$$



- Now multiply by normalized $\mathbf{V}$, so

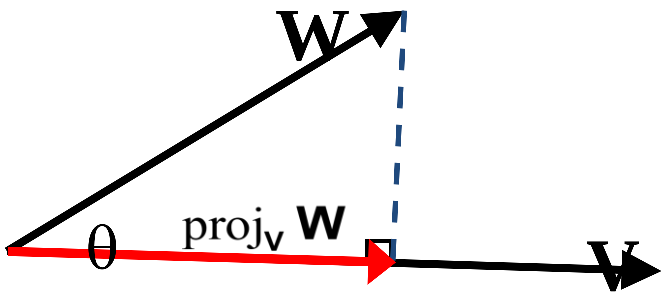
$$\text{proj}_{\mathbf{V}}\,\mathbf{W} = \left\|\text{proj}_{\mathbf{V}}\,\mathbf{W}\right\| \cdot \mathbf{U}_{\mathbf{V}} = \frac{\mathbf{W}\bullet\mathbf{V}}{\left\|\mathbf{V}\right\|} \cdot \frac{\mathbf{V}}{\left\|\mathbf{V}\right\|}$$

$$= \frac{\mathbf{W}\bullet\mathbf{V}}{\mathbf{V}\bullet\mathbf{V}}\,\mathbf{V}$$

Uses: Projection

- So,

$$\text{proj}_{\mathbf{V}}\,\mathbf{W} = \frac{\mathbf{W} \bullet \mathbf{V}}{\mathbf{V} \bullet \mathbf{V}}\mathbf{V}$$



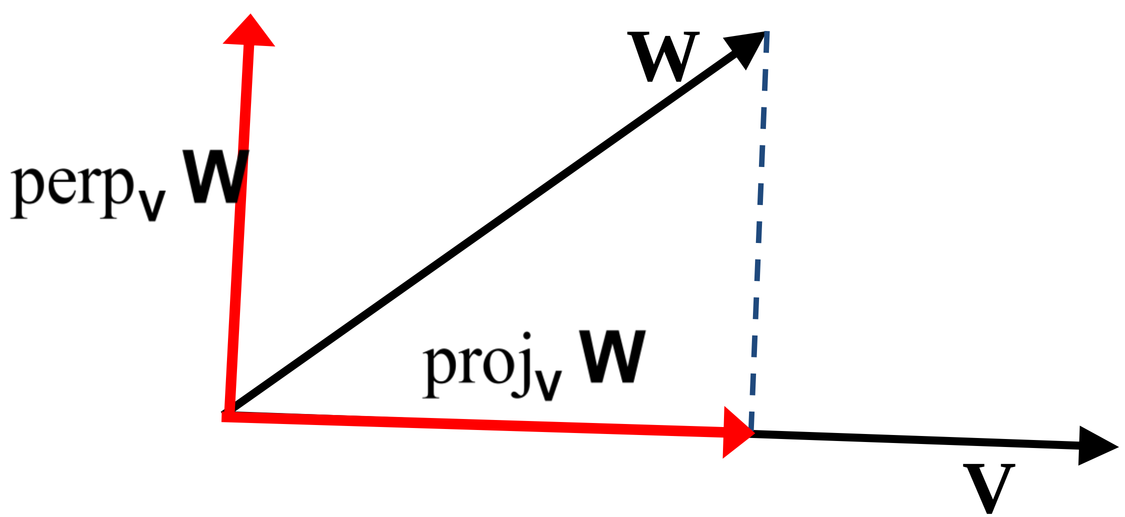
- If **V** is already normalized (often the case), then becomes

$$\text{proj}_{\mathbf{U}}\,\mathbf{W} = (\mathbf{W} \bullet \mathbf{U})\mathbf{U}$$

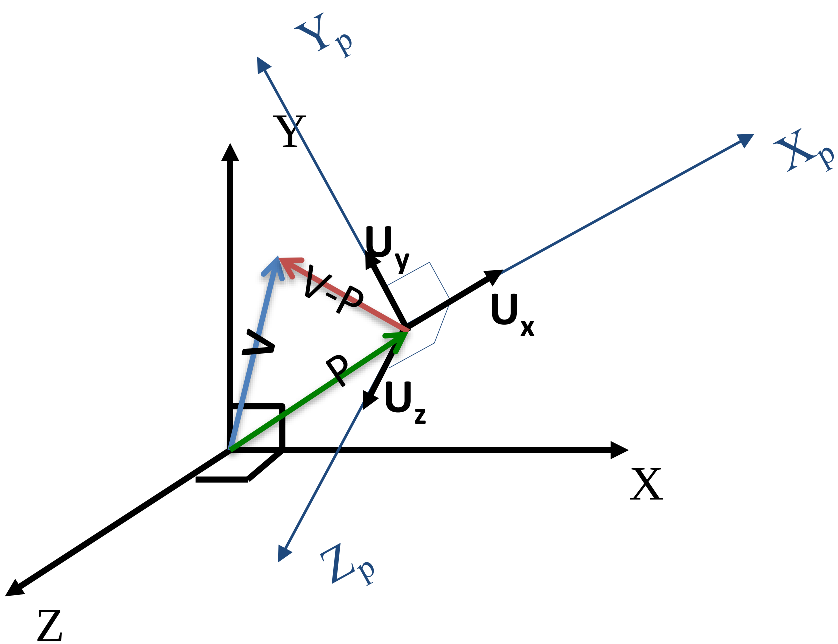
Uses: Projection

- Can use this to break $\mathbf{W}$ into components parallel and perpendicular to $\mathbf{V}$

$$\mathrm{perp}_{\mathbf{V}}\,\mathbf{W} = \mathbf{W} - \mathrm{proj}_{\mathbf{V}}\,\mathbf{W}$$



Uses: World Coordinates to Local Coordinates



$$X_p = ((\mathbf{V} - \mathbf{P}) \bullet \mathbf{U_x})$$

$$Y_p = ((\mathbf{V} - \mathbf{P}) \bullet \mathbf{U_y})$$

$$Z_p = ((\mathbf{V} - \mathbf{P}) \bullet \mathbf{U_z})$$

Need to know unit vectors $\mathbf{U_z}$, $\mathbf{U_y}$, $\mathbf{U_z}$