Math for Games 2

CS 3540 – Game Programming

1

Vector Operations: Dot Product

The dot product of two vectors is the sum of the products of corresponding components, resulting in a scalar value:

$\mathbf{a} = [x_1, y_1, z_1]$

$\mathbf{b} = [x_2, y_2, z_2]$

$\mathbf{a} \cdot \mathbf{b} = x_1 x_2 + y_1 y_2 + z_1 z_2$

2

Vector3.Dot()

Returns the dot product of two vectors

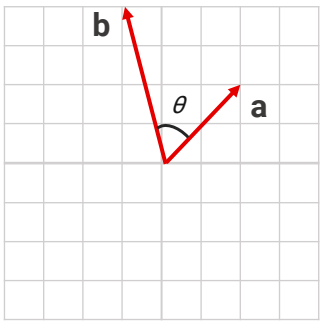
Return type is float

```
float dotProduct = Vector3.Dot(target.position, transform.position);
```

3

Dot Product and Intercepted Angle

The dot product of two vectors **a** and **b** is equal to the cosine of the angle θ between the vectors, multiplied by the lengths of the vectors



$$\mathbf{a} \cdot \mathbf{b} = ||\mathbf{a}||\ ||\mathbf{b}|| \cos\theta$$

The angle between two vectors:

$$\theta = \arccos\left(\frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||\ ||\mathbf{b}||}\right)$$

For unit vectors:

$$\hat{\mathbf{a}} \cdot \hat{\mathbf{b}} = \cos\theta$$

$$\theta = \arccos\left(\hat{\mathbf{a}} \cdot \hat{\mathbf{b}}\right)$$

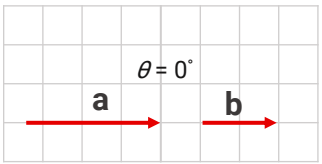
4

Dot Product and Intercepted Angle

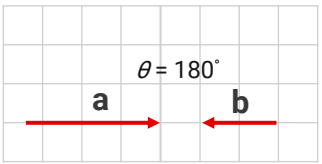
$\hat{a} \cdot \hat{b} = 1$ if the unit vectors are pointing in the same direction

$\hat{a} \cdot \hat{b} = -1$ if the unit vectors are pointing in the opposite directions

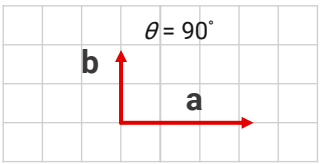
$\hat{a} \cdot \hat{b} = 0$ if the unit vectors are perpendicular to one another



$\cos 0^\circ = 1$



$\cos 180^\circ = -1$



$\cos 90^\circ = 0$

5

The Sign of Dot Product

$\mathbf{a} \cdot \mathbf{b} > 0$ when $0^\circ <= \theta < 90^\circ$

➔ **a** and **b** are pointing mostly in the same direction

$\mathbf{a} \cdot \mathbf{b} < 0$ when $90^\circ < \theta <= 180^\circ$

➔ **a** and **b** are pointing mostly in the opposite direction

$\mathbf{a} \cdot \mathbf{b} = 0$ when $\theta = 90^\circ$

➔ **a** and **b** are perpendicular

6

Vector3.Angle()

Returns the angle from one Vector3 to another Vector3

```
float angleObjects = Vector3.Angle(transform.position, target.position);
```

Also check out Vector3.SignedAngle()

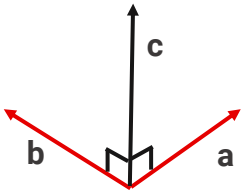
7

Vector Operations: Cross Product

The cross product of two 3D vectors, **a** and **b**, yields a 3D vector that is perpendicular to both **a** and **b**.

$\mathbf{a} = [x_1, y_1, z_1]$
$\mathbf{b} = [x_2, y_2, z_2]$

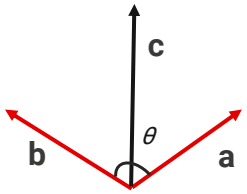
$$\rightarrow \quad \mathbf{a} \times \mathbf{b} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \times \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} y_1 z_2 - z_1 y_2 \\ z_1 x_2 - x_1 z_2 \\ x_1 y_2 - y_1 x_2 \end{pmatrix} = \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix}$$



8

Cross Product and Intercepted Angle

The length of **a** x **b** is equal to the product of the magnitudes of **a** and **b** and the sine of the angle θ between the vectors.



$$||\mathbf{a} \times \mathbf{b}|| = ||\mathbf{a}||\,||\mathbf{b}||\sin\theta$$

The angle between two vectors:

$$\theta = \arcsin\left(\frac{||\mathbf{a} \times \mathbf{b}||}{||\mathbf{a}||\,||\mathbf{b}||}\right)$$

For unit vectors:

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

$$\theta = \arcsin\left(||\hat{\mathbf{a}} \times \hat{\mathbf{b}}||\right)$$

9

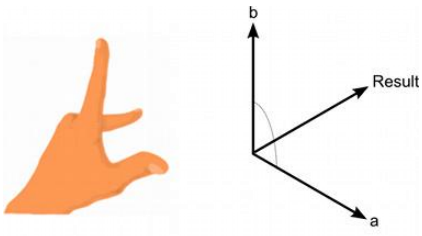
Vector3.Cross()

Returns the cross product of two vectors

Returns a Vector3 object

```
Vector3 crossProduct = Vector3.Cross(target.position, transform.position);
```

The left-hand rule applied to Cross(**a**, **b**)



10

Vector3.Lerp()

Linear Interpolation

The estimation of values of a variable between two known values based on an interpolation constant, **t**

$$(a, b, t) = a + (b-a)\, t$$

Vector3.Lerp(Vector3 a, Vector3 b, float t)

Interpolates between the vectors **a** and **b** by the interpolant **t**.

The parameter t is clamped to the range [0, 1].

11

Vector3.MoveTowards()

Moves the object toward another object

Calculates a new position based on the step or distance value provided

Returns a new position Vector3 for the calculated new position

```
transform.position = Vector3.MoveTowards(transform.position, target.position, step);
```

12

GetComponent()

The primary way of accessing components attached to a GameObject

```
void Start()
{
    renderer = GetComponent<Renderer>();
}
```

13

Readings

https://docs.unity3d.com/ScriptReference/Vector3.Dot.html

https://docs.unity3d.com/ScriptReference/Vector3.Cross.html

https://docs.unity3d.com/ScriptReference/Vector3.Angle.html

https://docs.unity3d.com/ScriptReference/Vector3.MoveTowards.html

14