Rotations

Thursday, December 31, 2020 3:48 PM

Quaternions: Simply put, how Unity handles rotations.
Euler Angles: Amount of degrees an object is rotating along a particular axis.
Always handle rotations through quaternions!!

Quaternion.Identity: Identity rotation. No rotation. Set x, y, z rotation to 0.
Quaternion.Euler: Return a rotation that rotates z around the z axis, x degrees along the x axis, and y degrees along the y axis.
Quaternion.LookAtRotation: Set the rotation to aim towards a specified target.

```
[SerializeField] Transform aimTarget;
// direction = destination - source
Vector3 directionToFace = aimTarget.position - transform.position;
// Draw out a ray towards the specified target.
Debug.DrawRay(transform.position, directionToFace, Color.green);
transform.rotation = Quaternion.LookAtRotation(directionToFace);
```

Quaternion.Slerp
Linear interpolation but spherically. Helps to smooth out rotations.
Quarternion targetRotation = Quaternion.LookAtRotation(directionToFace);
transform.rotation = Quaternion.Slerp(transform.rotation, targetRotation, Time.deltaTime);

From rotation
To rotation
Time it takes to complete rotation