# Quaternions in the Roblox engine:

Quaternions are four dimensional in nature, and useful when doing rotations.

Single Axis Rotations:
To simulate a 2D rotation with two numbers instead of an angle, the rotation is easily represented by coordinates on the edge of a circle, with 45deg being equivalent to (0.707, 0.707)

Quaternions around a single axis are similar. If these numbers were plugged into the W and X components of a quaternion, the result would be a 90deg rotation around the X axis

Experimenting on other single axis rotations gives light to a pattern, taking any point of the edge of the circle and finding the angle, when plugging the point into the W and X components of a quaternion, a rotation of twice the amount of the initial angle is done. A full rotation of a circle would therefore be 720deg
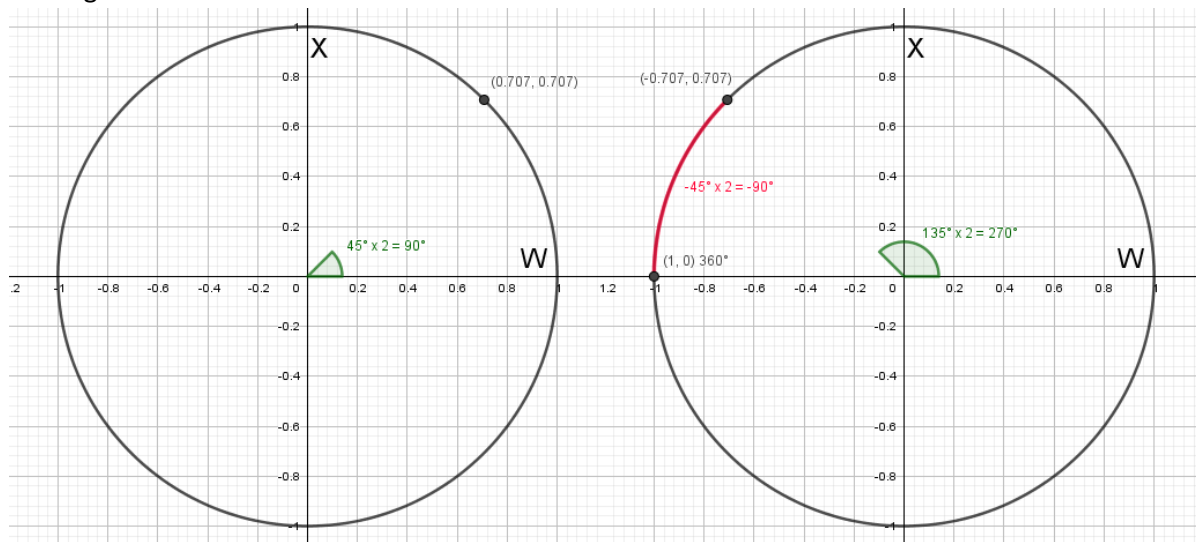

Flipping the W component in the 90deg rotation example (above) gives a total rotation of 270deg on the X axis which is the same as -90 on the X axis, giving the same magnitude of rotation but in the opposite direction.

Flipping all of the components gives a 450deg rotation, which shows that the end rotation of something may be represented by different numbers, which are going to be referred to as 'quaternion twins'

'quaternion twins', every quaternion has a matching pair with flipped components that represents the same end rotation. For any SLERP (spherical linear interpolation), there are two ways to reach the destination, clockwise and counterclockwise

When CFrames LEPR, they use the 'twin' with the shortest rotational path. Figuring out which is closer is done by comparing the dot product of the quaternion you're going from to the one you're going to, ensuring the two are on the same half of the 4D unit sphere (therefore closest)
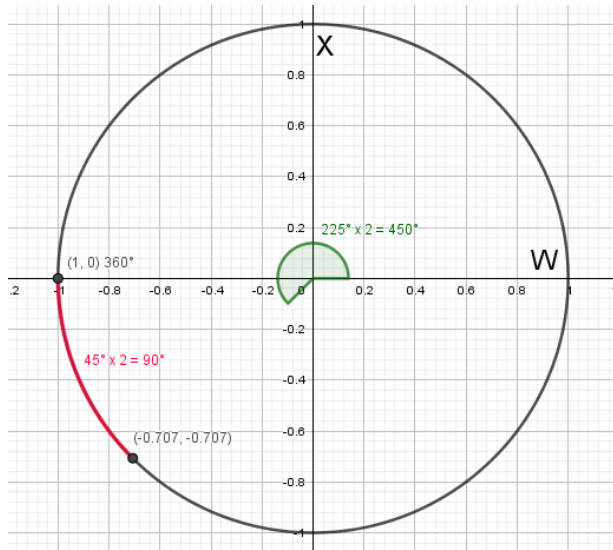
Two Axis Rotations:

To represent a rotation on two axes, three numbers are needed. These will be labeled as W, X and Y components.

The point (0, 0, 1) is equivalent to 180deg around the Y axis and (0, 1, 0) is equivalent to 190deg around the X axis

It can be seen why a quaternion doubles the angle represented by the point, assuming that X and Y did represent 90 degrees on their respective axis, it would mean that the points representing 180 degrees around both X and Y would be the same point (-1, 0, 0)
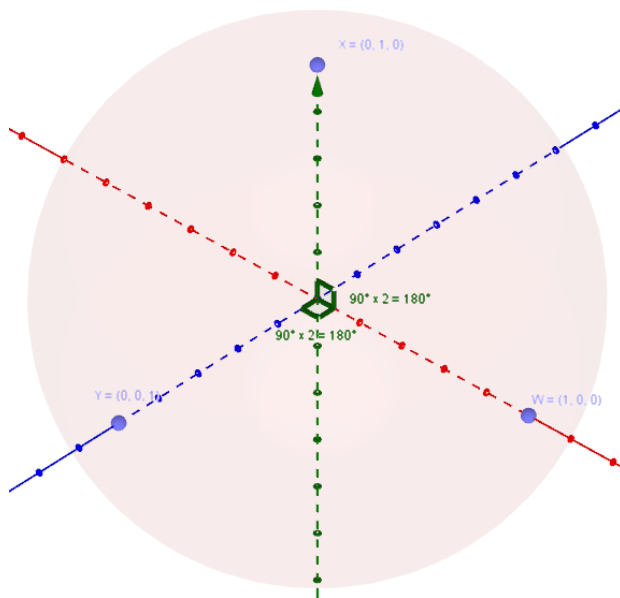
All Axes Rotations:

Blending rotations on the surface of a circular object is how full 4D quaternions work, but it cannot be visualized in the full 3-axes form because it would require a 4D unit sphere. In roblox, the CFrame constructor normalizes the points, allowing the user to not have to calculate averages that lay on the surface of the 4D sphere and figure out what unit blend would be on the surface of the 4D sphere.

IE:

W = (1, 0, 0, 0) -> Unrotated
X = (0, 1, 0, 0) -> 180 degrees around X-axis
Y = (0, 0, 1, 0) -> 180 degrees around Y-axis
Z = (0, 0, 0, 1) -> 180 degrees around Z-axis

Visualization of blend between 180 degrees on Y and unrotated:

W	X	Y	Z

CFrame.new(0, 0, 0, 0, 1, 0, 1)

W	X	Y	Z

CFrame.new(0, 0, 0, 0, 1, 0, 1)