

Rotation Notes

Rotation

Transform.Rotate() - <https://docs.unity3d.com/ScriptReference/Transform.Rotate.html>

Transform.RotateAround() - <https://docs.unity3d.com/ScriptReference/Transform.RotateAround.html>

Space world and self in unity - <https://docs.unity3d.com/ScriptReference/Space.html>

Quaternions - <http://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/>

Quaternions - <http://mathworld.wolfram.com/Quaternion.html>

Angles - degrees and radians

Angles - measured in degrees, 360 degrees are full circle, 180 degrees - half a circle, 90 degrees - quarter of a circle.
Pretty intuitive to work with it, most people are thinking of rotations in degree.

Radians - are what most programming languages are using. Radian is a unit of measurement for angles defined by the ratio of the length of the arc of the circle to the radius of that circle.

1 Radian - is the angle at which that ratio equals 1.

180 degrees =

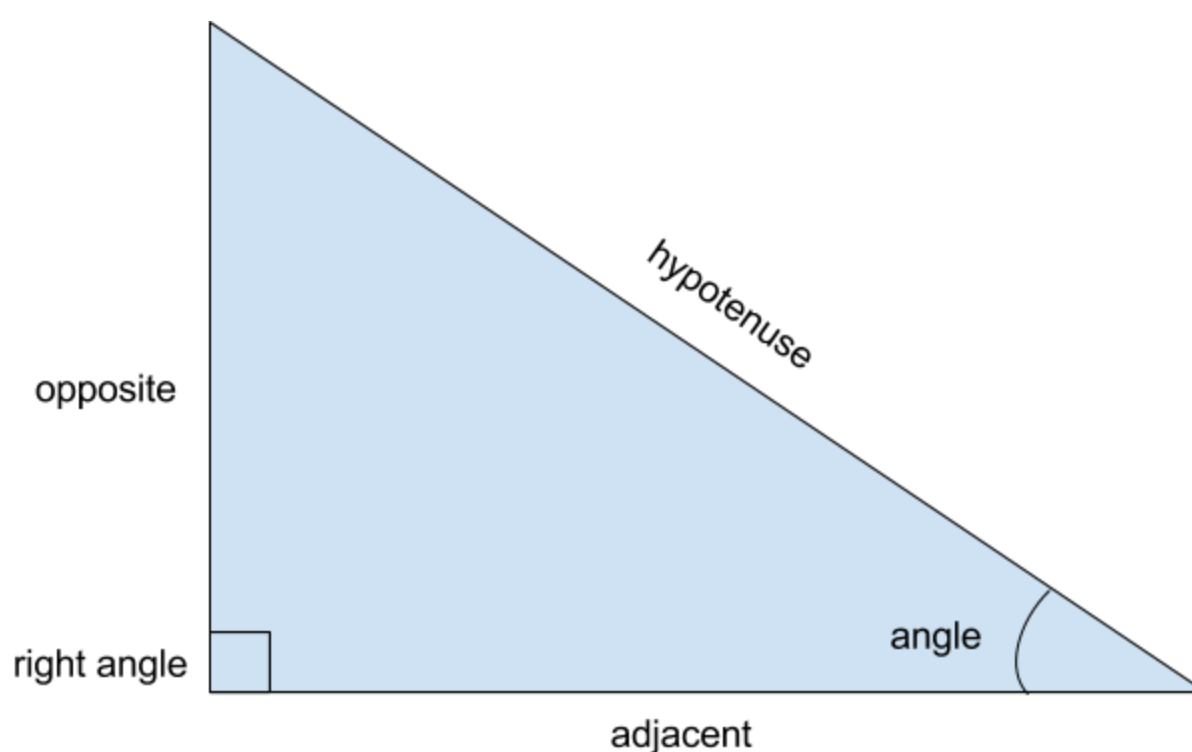
π radians;

360 degrees = $2 * \pi$ radians;

90 degrees = $\pi/2$ radians

π - mathematical constant, real number. Ratio of a circle's circumference(distance around the perimeter) to its diameter(a straight line that passes through the circle's center).

π - 3.14159 - 3.14



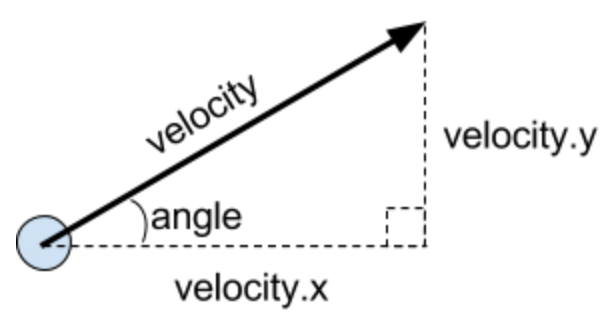
Sohcahtoa

Soh: sine = opposite/hypotenuse

Cah: cosine = adjacent/hypotenuse

Toa: tangent = opposite/adjacent

- Draw the triangle but using a vector instead. Hypotenuse: vector magnitude v, adjacent: x value of vector v, opposite: y value of vector v.



$\text{tangent}(\text{angle}) = \text{velocity.y} / \text{velocity.x}; \quad \text{angle} = ?$
 $\text{angle} = \text{arctangent}(\text{velocity.y} / \text{velocity.x})$
 $\text{angle} = \text{atan2}(\text{velocity.y} / \text{velocity.x})$ - caters for different direction that give the same result

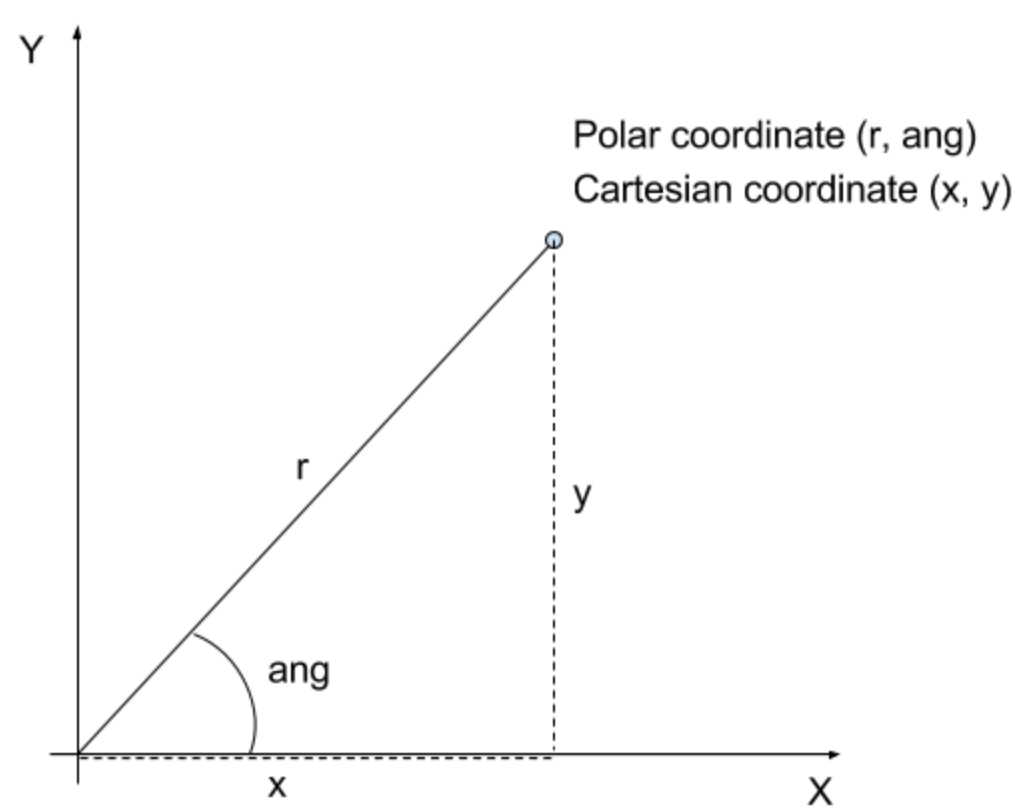
Gizmos

`OnDrawGizmos()` - <https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnDrawGizmos.html>
`OnDrawGizmosSelected()` - <https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnDrawGizmosSelected.html>

`Camera.ScreenToWorldPoint()` - <https://docs.unity3d.com/ScriptReference/Camera.ScreenToWorldPoint.html>

Polar and Cartesian Coordinates

Cartesian coordinates - describes a point in space using x and y coordinates, components of a vector
Polar coordinates - describes a point in space as an angle of rotation around the origin and a radius from the origin, the magnitude(length) and direction(angle) of a vector.



$\sin(\text{ang}) = y / r \quad y = r * \sin(\text{ang})$
 $\cos(\text{ang}) = x / r \quad x = r * \cos(\text{ang})$