

Rotation in Unity3D is *hard*

a.k.a. "handedness in 2D->3D sucks"

Why?

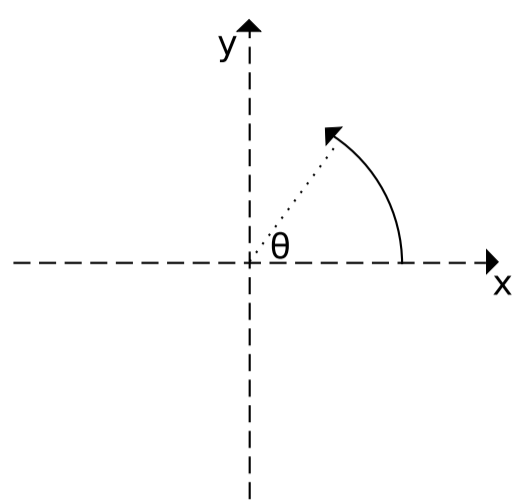
I have a Maths-based degree, and yet I regularly forget subtleties of Unity's rotation.

The Unity docs gloss over all this, don't give the needed info, and IMHO are actively misleading for some of this (c.f. TRS).

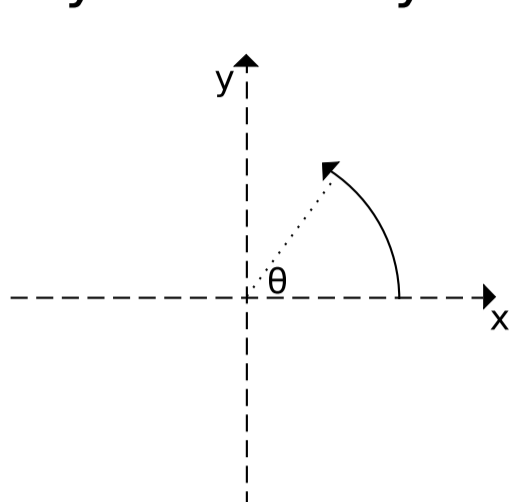
So that I never forget again, here's a cheat-sheet.

Basics: Positive Rotation

Maths:

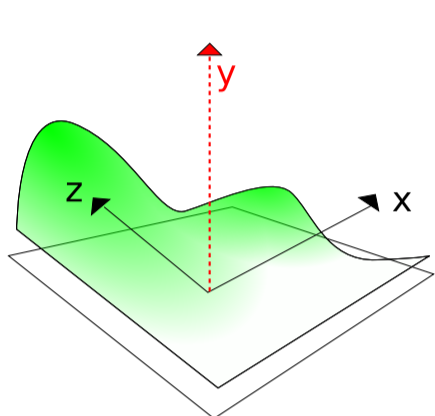


Unity2D + Unity GUI:



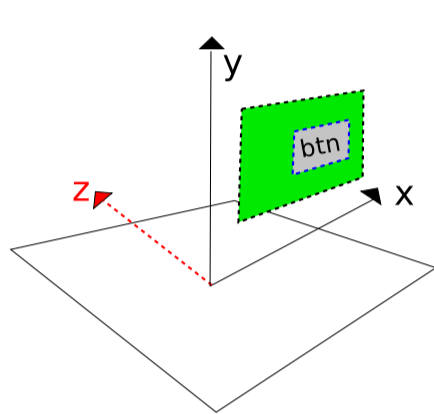
Is 3rd dimension ("up") Y .. or Z?

Unity 3D: Terrains = Y

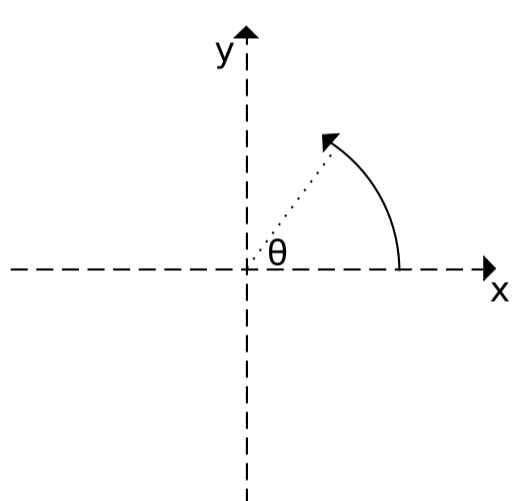


Unity 3D: GUI = Z

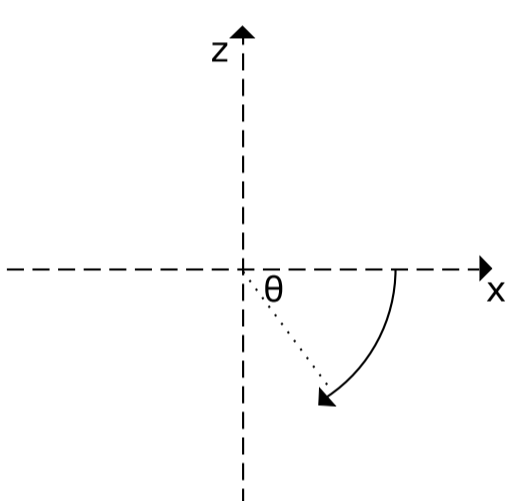
Unity 3D: Platformer = Z



Unity3D:



but:



Effect:

Making a dungeon? A map?

+VE rotations are CLOCKWISE

Making a GUI? A sidescrolling SHMUP? A platformer?

+VE rotations are ANTI-CLOCKWISE

Matrices

Maths: (can't argue with this; it's fact)

"Scale, then Rotate, then Translate"

Written:

Translate(Rotate(Scale(x)))

C#:

"Without brackets, Left first, Right second"

Written:

((Scale(x)) Rotate) Translate

=

Scale * Rotate * Translate * x

(x at end? confusing? Yes)

Unity Docs:

"Translate Rotate, Scale"

Unity Code:

"Scale, Roate, Translate"

Errors etc

@t_machine_org