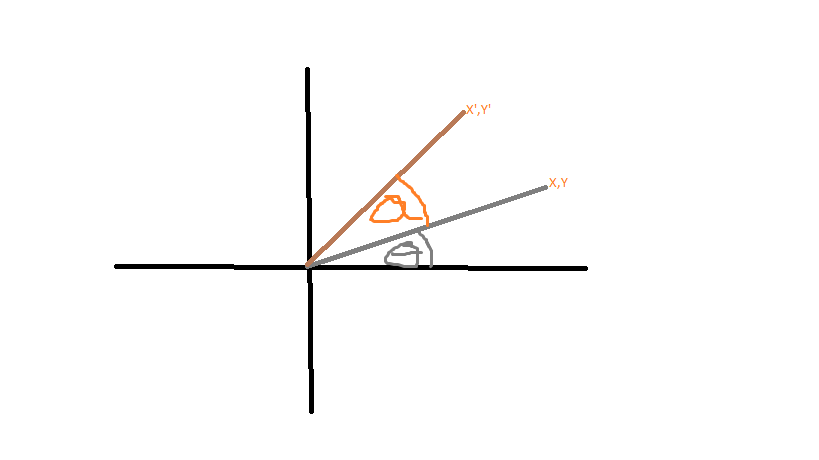
Question 2 Test

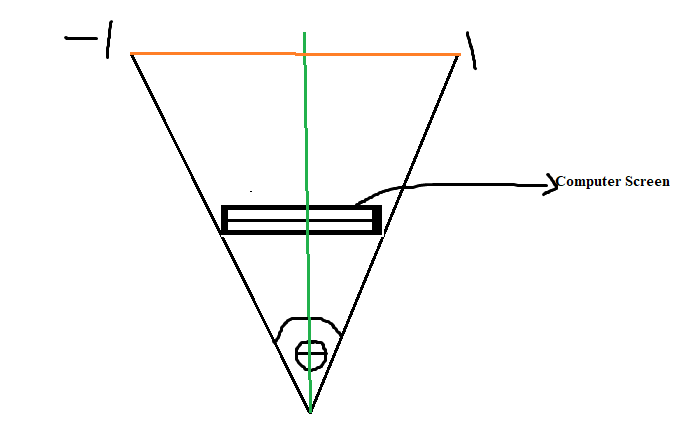
For a transformation: Rotation Matrix along z



We can represent the original co-ordinates of the system as a vector

For Translation (Moving a point from original to the other)

For the perspective Matrix



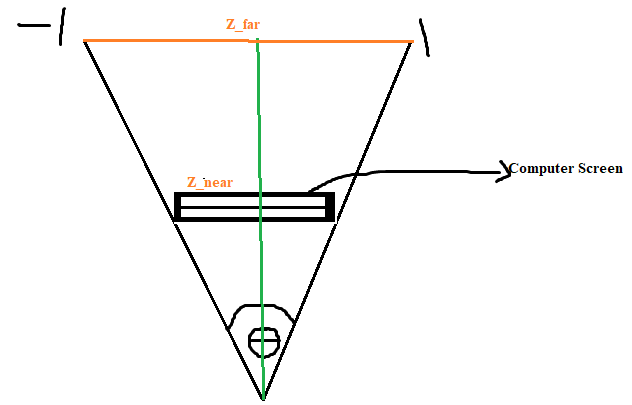
The relationship at which both x and y changes can be related and linked together using trigonometry:

Both x and y have to be scaled using this factor, One tricky part about this whole idea is the inverse relationship between the movement of the object relative to the user and its movement relative to the equation. When the FOV or Field of view increases then this implies less objects that a person is supposed to see. The lesser the field of view then the more objects the viewer is supposed to see,

With the idea of ensuring that the ratios appear as requested by a user, we firstly need to introduce the idea of Aspect ratio,

This idea depends entirely on which part of the canvas is longer, if the width is longer than the height, like the case whereby our computer screens are 1080 resolution wide and around 720 long, giving them a rectangular shape, if you draw a square on such a canvas then the co-ordinates would match a rectangle.

The z co-ordinate is responsible for the depth of the object we are drawing on the screen so it is somehow supposed to change:



Now we need deal with the relationship between the depth of the shape and the ratio between the nearer z-co-ordinate and the z\_far

But due to the scaling and the offset from the screen, we need to offset the total offset value of the depth .

Using all the information we have obtained so far, we can start deriving our Projection Matrix or Perspective Matrix