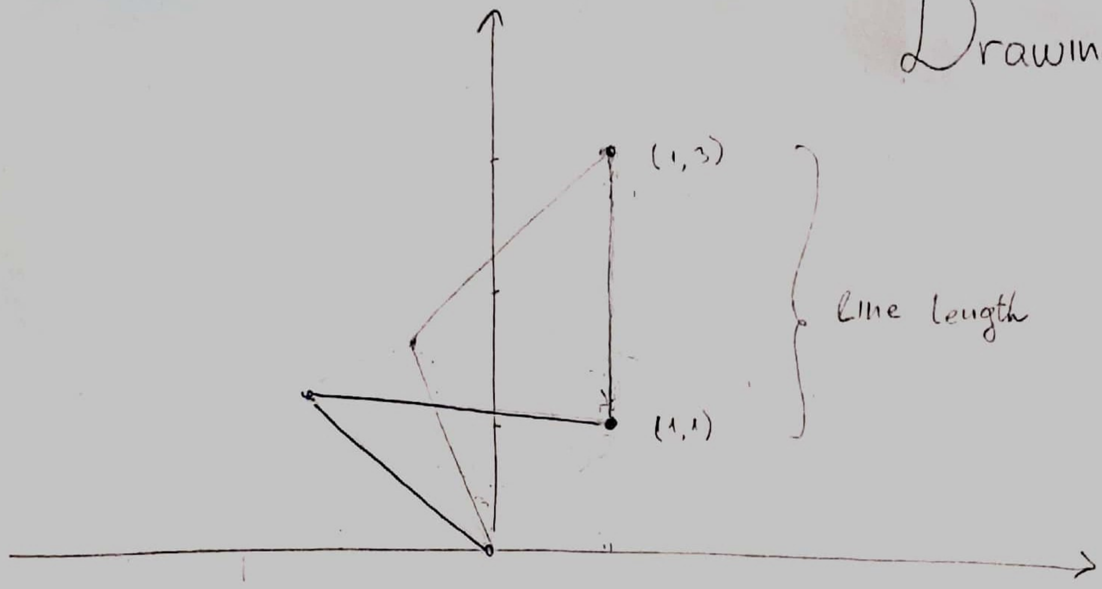
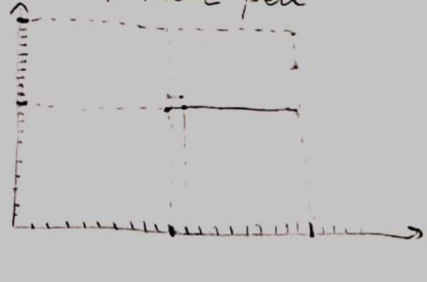
Drawing Line ①



1) move pen from (1,3) to (1,1)

2) calculate the length of the line
calculate number of steps = line length · X
↑this will divide it into X pieces . using for loop : for step in range(numberofst

```
for step in range(number of steps):
    x += step_of_x
    y += step_of_y
    ...
```



3) move pen to (1,3)
To move pen stepper motor and servo motor classes are used

move pen to (x,y) coordinate?

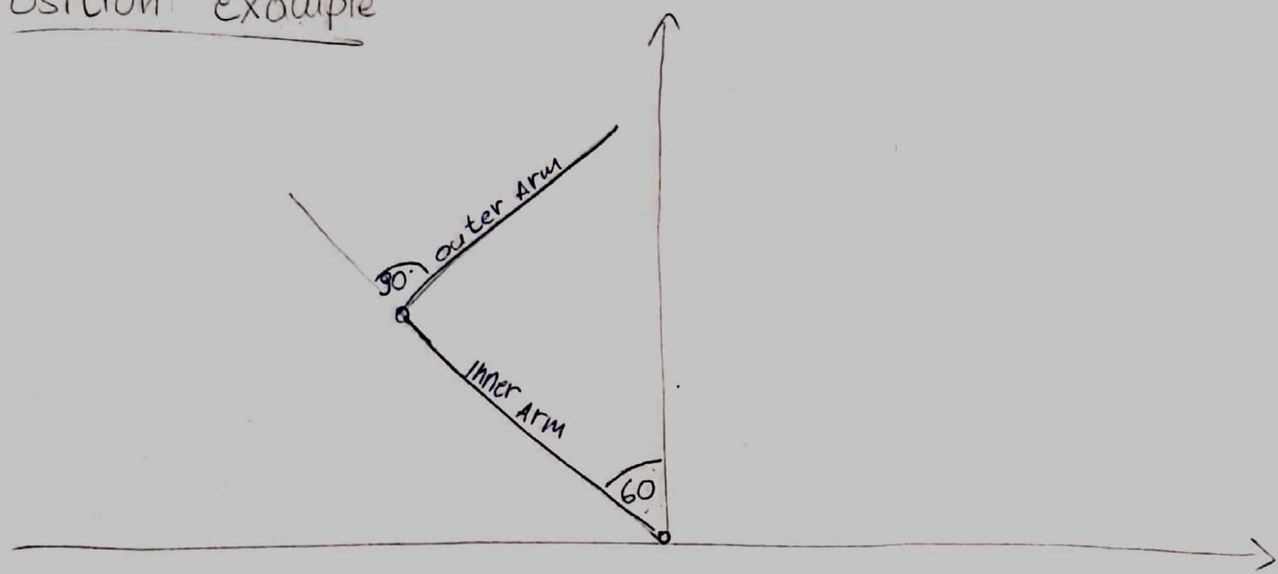
get current position (x, y)

convert xy to angles

when we get the angles we call setAngle(angle1) & setAngle(angle2)

save current xy and current angle 1 & angle 2

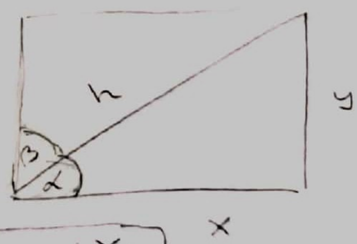
Position example

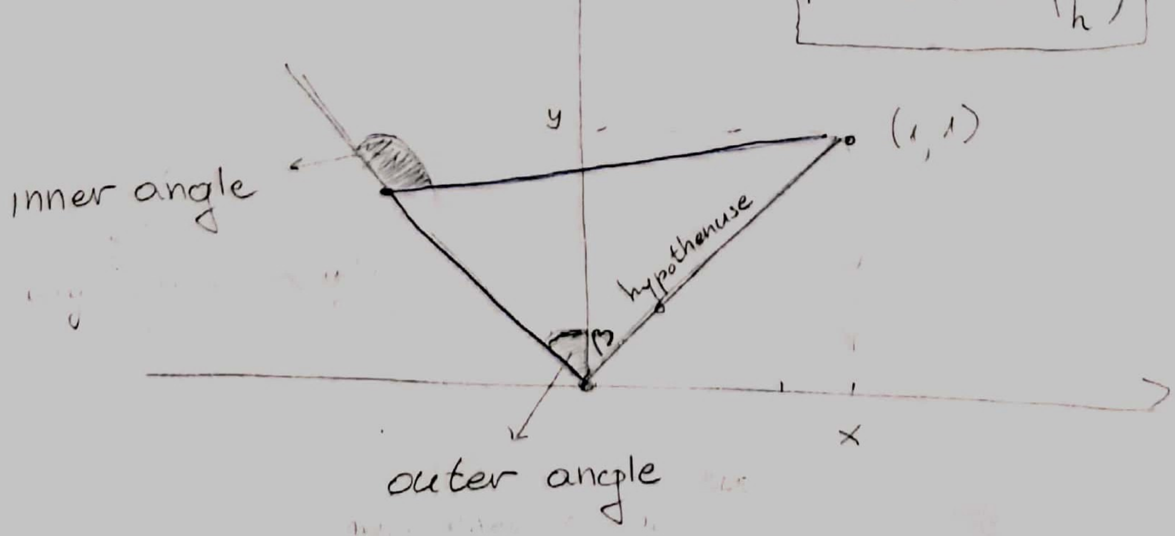


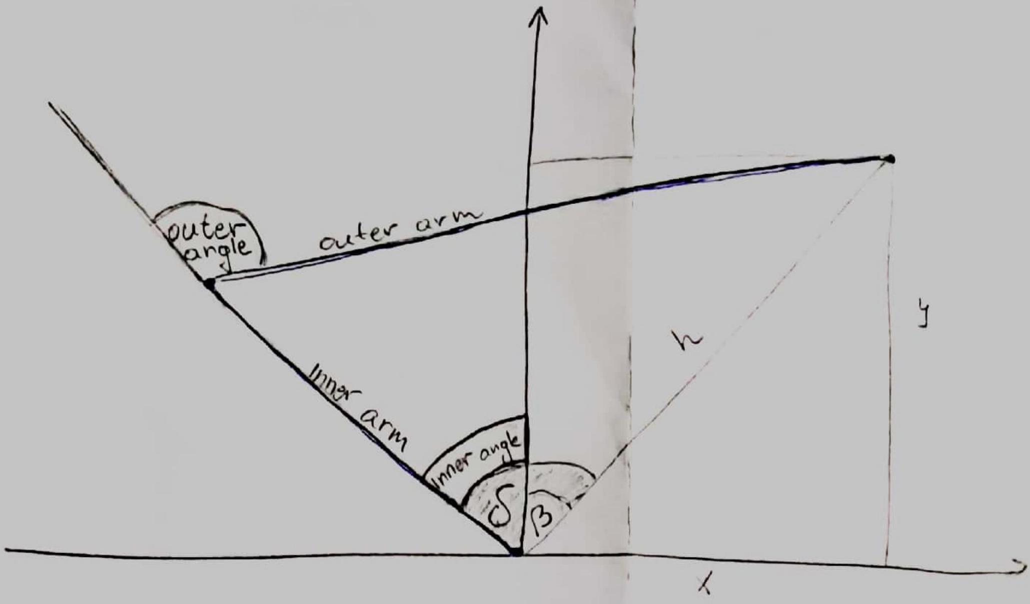
$\sin\alpha = \frac{y}{h}$

$\sin\beta = \frac{x}{h}$

$\Rightarrow \boxed{\beta = \arcsin\left(\frac{x}{h}\right)}$







inner angle = -60°
outer angle = $+120^\circ$

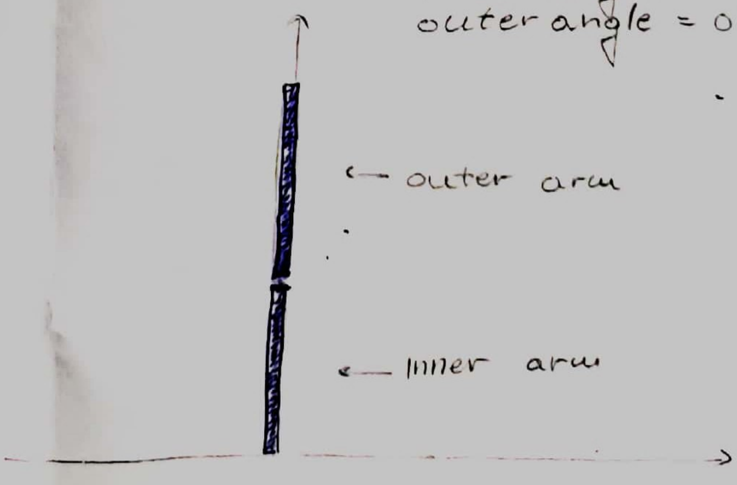
Nawings in the software

In the software solution it is very important to understand namings that are used.
The sketch above represents a possible position and nowings:

inner arm
outer arm
inner angle
outer angle

Initial Position

inner angle = 0°
outer angle = 0°



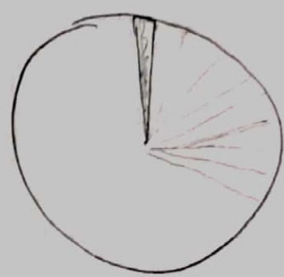
Stepper motor

code for making the full circle:

```
for i in range(50):
    forward step()
```

↑this will make 360˚

one step is $\frac{1}{50}$



$$\frac{1}{360} = \frac{x}{50}$$

$$x = \frac{5}{36}$$

$$x = 0.14$$

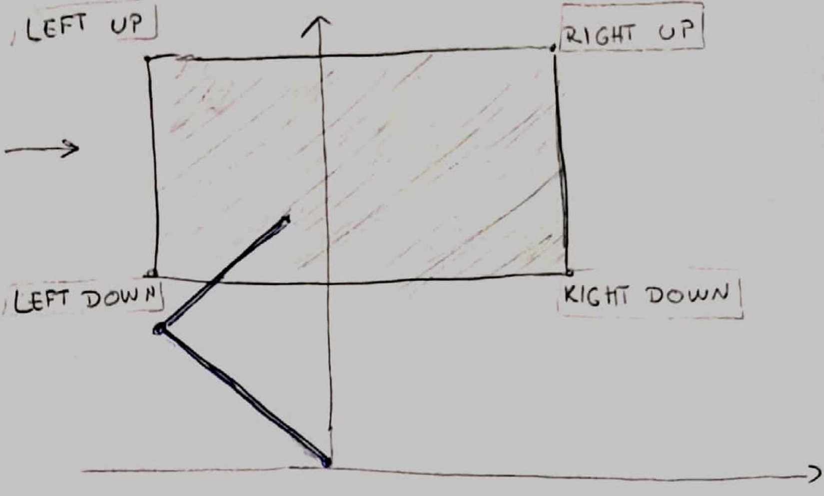
since we are gonna get angle as an input we have to transform angle into step.

This is pseudo code for moving stepper motor, with configuration of 50 steps for full circle.

Other stepper motor may have more steps in a full circle. That stepper motor would be more precise

Boundaries

Drawable area →



Drawable area must be determined in order not to break physical hardware components.

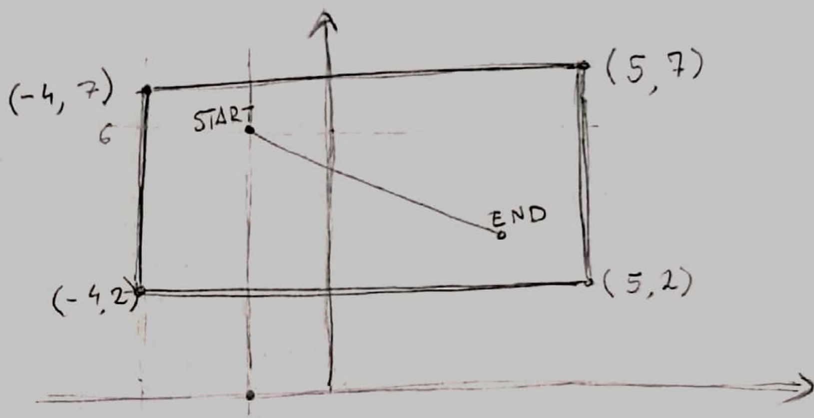
Once the hardware is made and glued all together, drawable area is determined empiracaly.

Inner and outer arm must be moved so that all 4 corners can be reached.

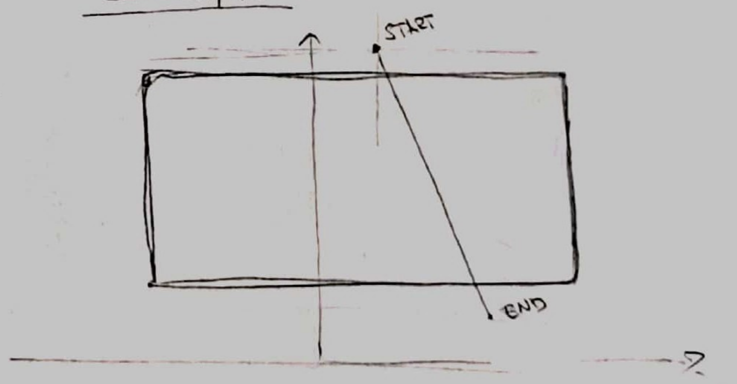
Those 4 corners should form a rectangle.

The drawable area is the frame for a picture which will be drawn inside the frame.

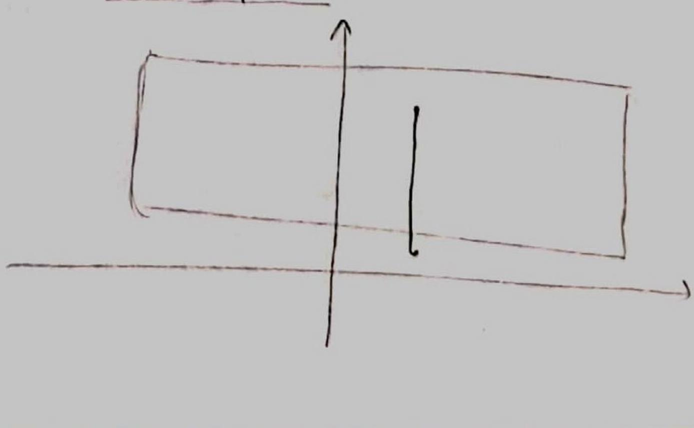
Example 1



example 2

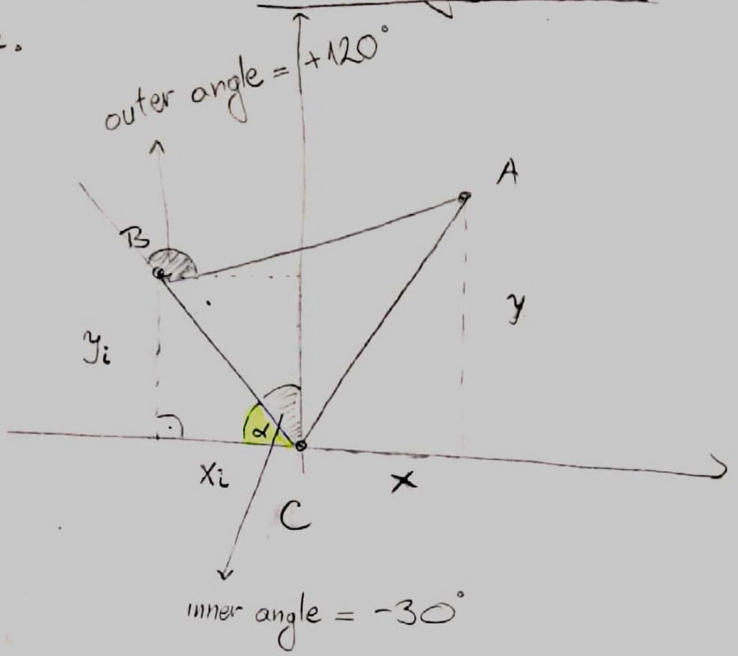


example 3



Plotting simulation

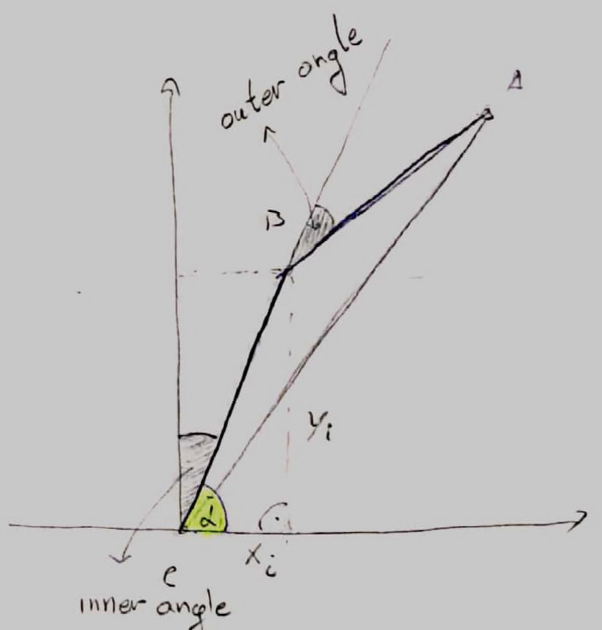
If rpi and harware components aren't conected, simulation of movement is made. It is good for testing before testing on actual harduare.



First of all $\boxed{\alpha}$ must be found.
The $\boxed{x_i}$ and $\boxed{y_i}$

When $\boxed{\alpha}$, $\boxed{x_i}$ an $\boxed{y_i}$ are found, then we know $B(x_i, y_i)$.
Now we know 3 edges A, B & C so we can plot arms in plotting simulation.

$$\sin\alpha = \frac{y_i}{\text{inner arm}}, \qquad \cos\alpha = \frac{x_i}{\text{inner arm}}$$



$$\alpha = \operatorname{asin}\left(\frac{y_i}{\text{inner arm}}\right).$$

$$x_i^2 + y_i^2 = \text{inner\_arm}^2$$

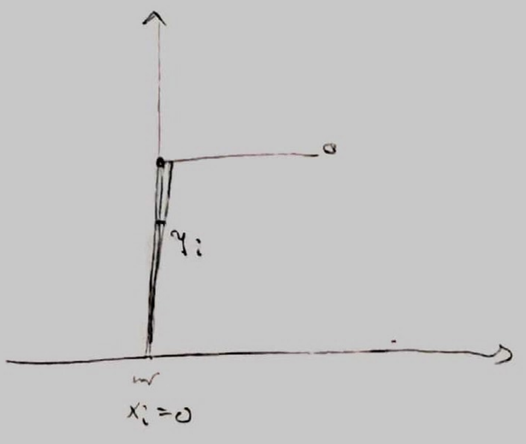
$$y_i = \sin(\alpha) \cdot \text{inner arm}$$

$$x_i = \cos(\alpha) \cdot \text{inner arm}$$

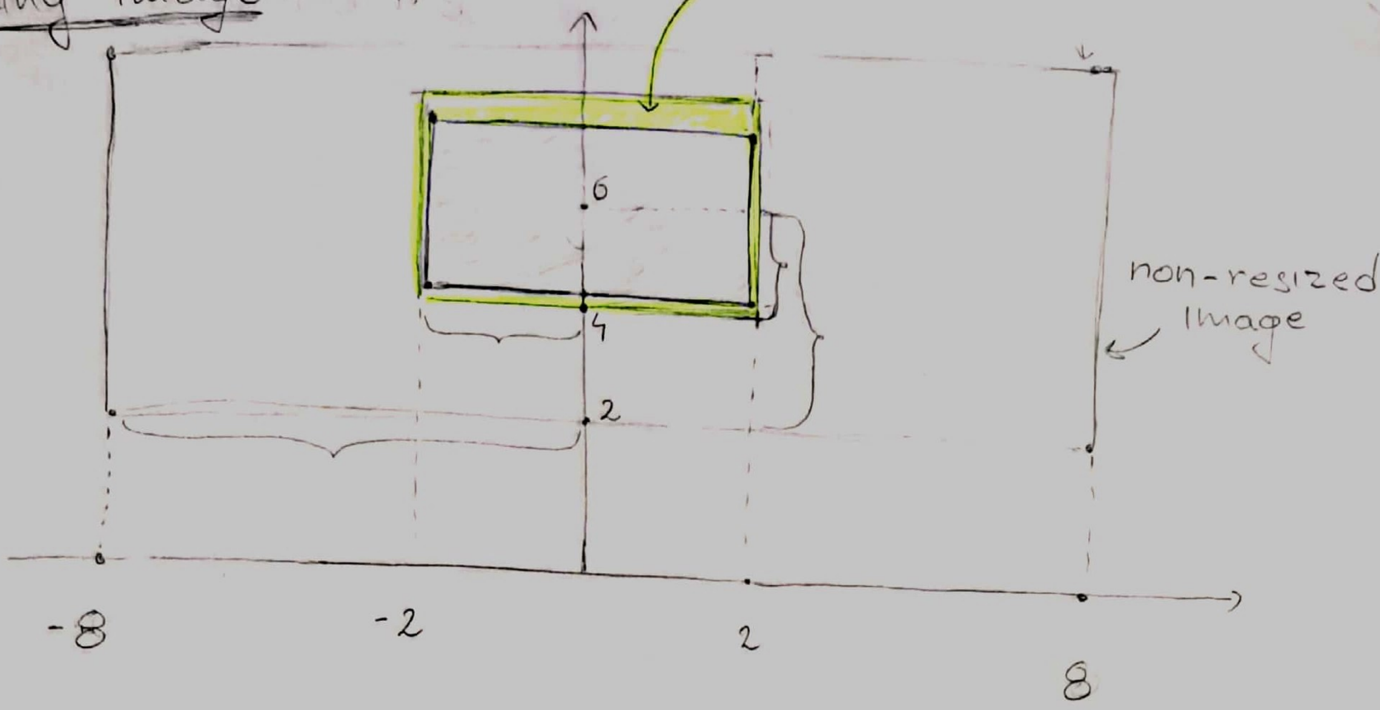
$\boxed{\text{In python}}$

$$y_i = \sin(\text{radians}(\alpha)) \cdot \text{inner arm}$$

$$x_i = \cos(\text{radians}(\alpha)) \cdot \text{inner arm}$$



Resizing image



1.

scaling factor $= \frac{-8}{-2} = 4$

scaling factor $= \frac{6-2}{4-2} = 1.5$

great scaling factor should be used ⇒ x scaling factor

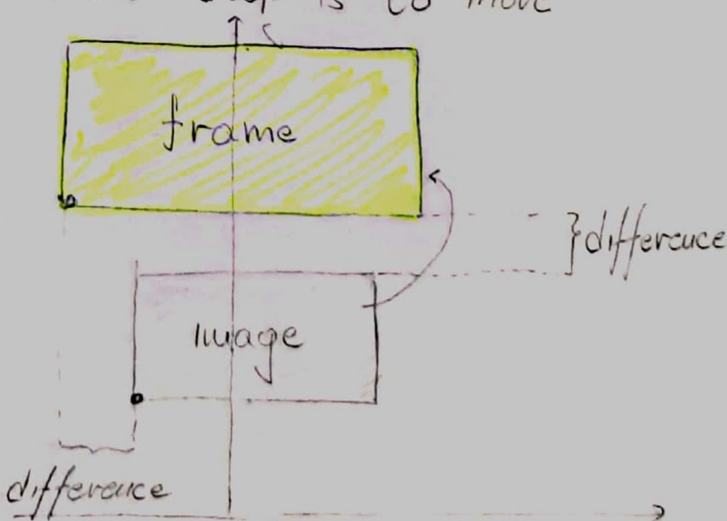
2.

each (x, y) scale by x scaling factor (greater scaling factor)

3.

NOW we Have our image scaled. Next step is to move THE image into frame.

foreach (x, y) add differences on x-axis and y-axis



4

image is in the frame now →