SOURCE CODE

```
# 2d transformation

import pygame
import sys
import math

def translation(x,y,tx,ty):
    return (x + tx, y + ty)

def scaling(x,y,sx,sy):
    return (x * sx, y * sy)

def rotation(x,y,angle):
    radian = math.radians(angle)
    x_new = x * math.cos(radian) - y * math.sin(radian)
    y_new = x * math.sin(radian) + y * math.cos(radian)
    return (x_new, y_new)


pygame.init()
WIDTH = 1000
HEIGHT = 1000

screen = pygame.display.set_mode((WIDTH, HEIGHT))

pygame.display.set_caption("2d Transformation")

WHITE = (255, 255, 255)
BLACK = (0, 0, 0)


def main():
    screen.fill(BLACK)
    x1 = 100
    y1 = 100
    x2 = 700
    y2 = 200

    pygame.draw.line(screen, WHITE, (x1,y1), (x2,y2), 1)

    #after translation
    tx1,ty1 = translation(x1,y1,50,50)
    tx2,ty2 = translation(x2,y2,50,50)
    pygame.draw.line(screen, (0,0,255), (tx1,ty1), (tx2,ty2), 1)

    #after scaling
    sx1, sy1 = scaling(x1, y1, 2, 2)
    sx2, sy2 = scaling(x2, y2, 2, 2)
    pygame.draw.line(screen, (255,0,0), (sx1, sy1), (sx2, sy2), 1)

    #after rotation
    angle = 45  #
    rx1, ry1 = rotation(x1, y1, angle)
    rx2, ry2 = rotation(x2, y2, angle)
    pygame.draw.line(screen, (0,255,0), (rx1, ry1), (rx2, ry2), 1)


    pygame.display.flip()
    pygame.time.delay(100)


if __name__ == "__main__":
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
        main()
        pygame.time.delay(100)
```

OUTPUT

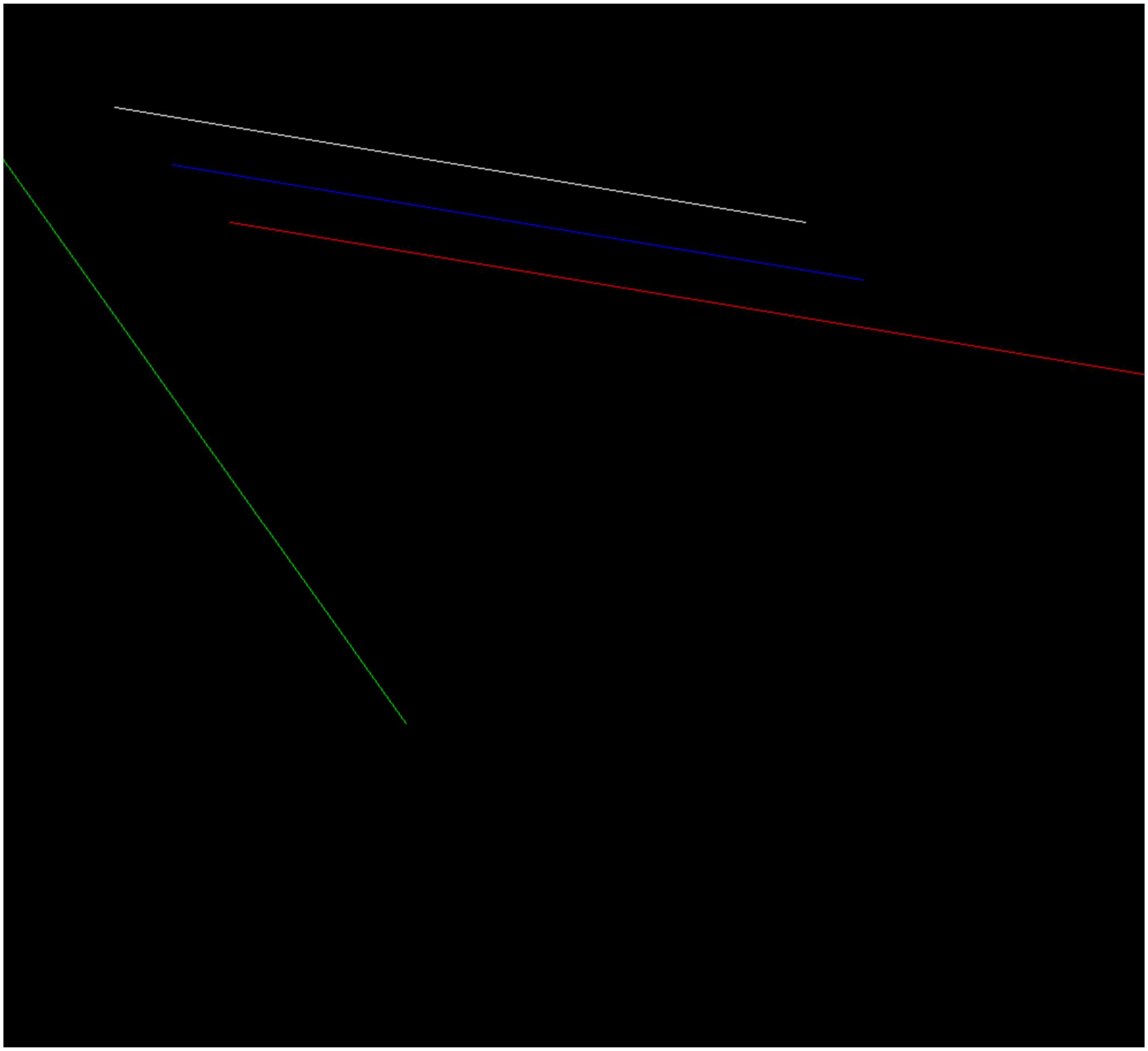

Fig. 6.1: Output of 2d Transformation