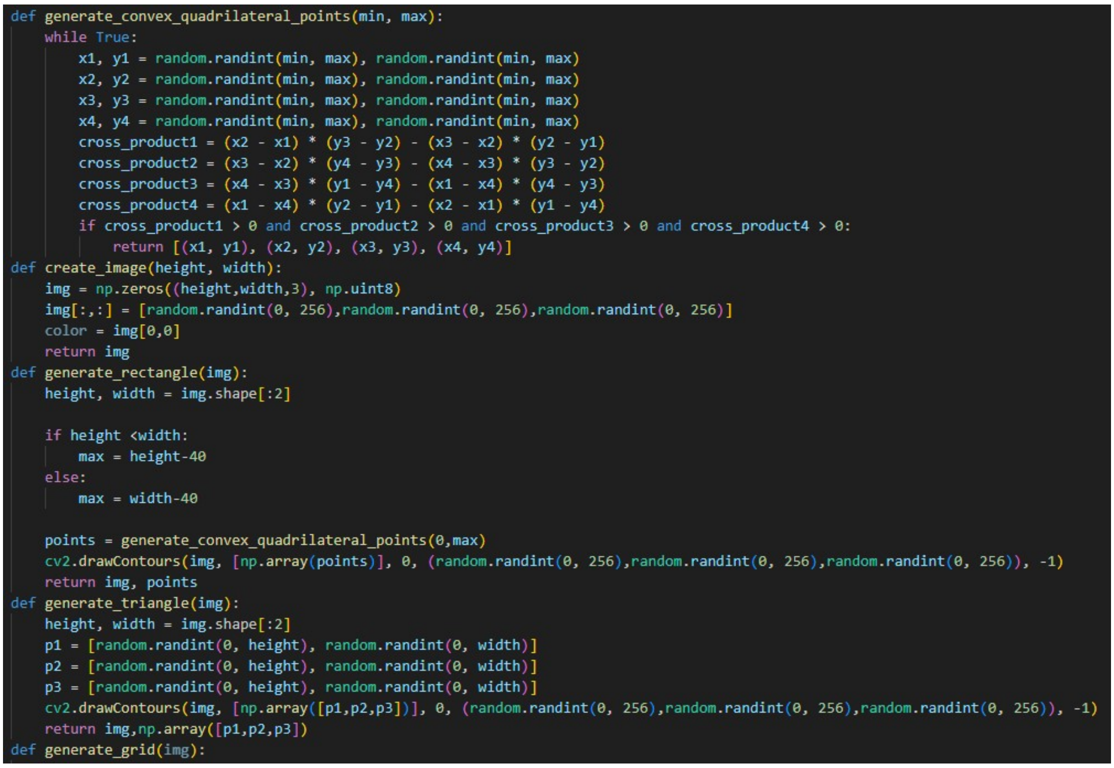
Poročilo - Vaja 1

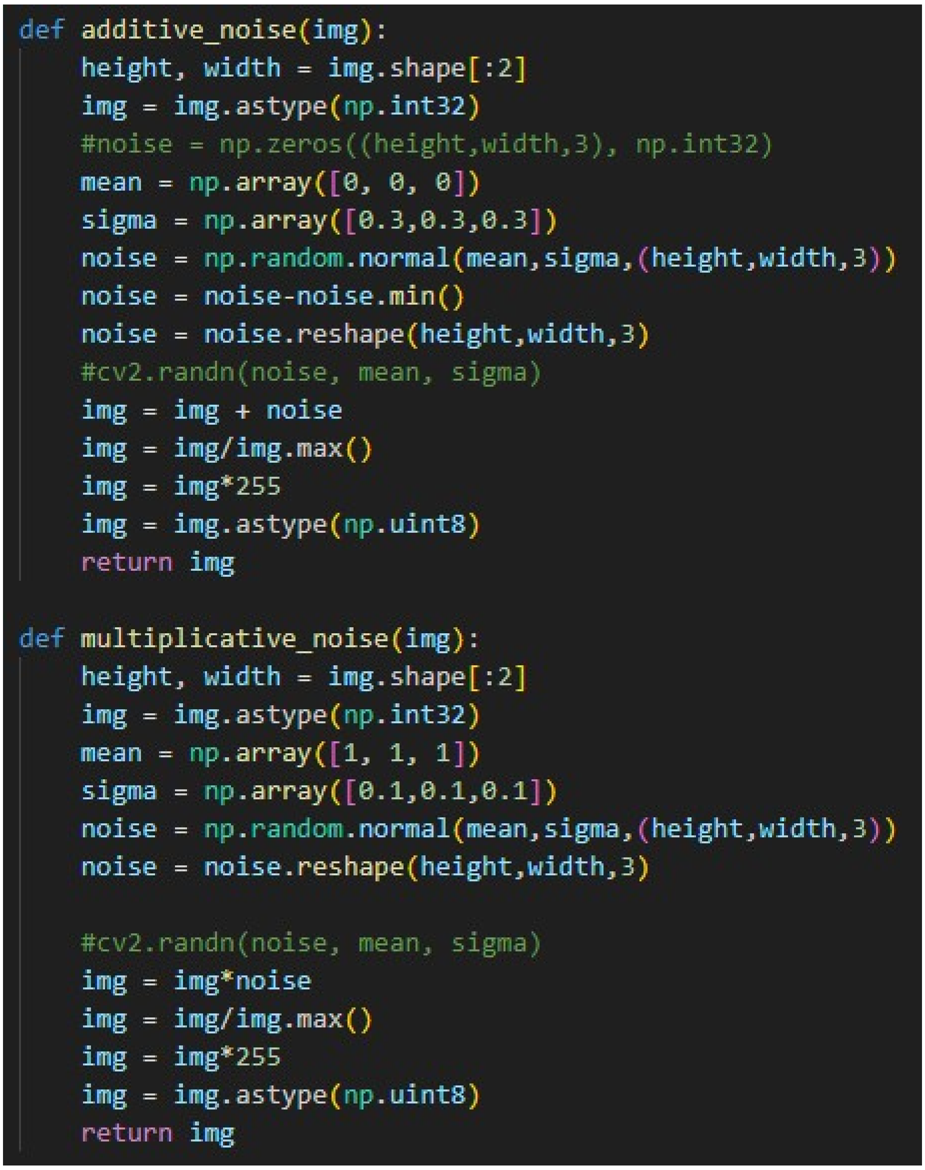
Tukaj smo implementirali genarator različnih likov ter šuma nad proizvedenimi slikami.

Naša koda najprej tvori naključne like na sliki - Elipsa, Štirikotnik, Trikotnik, Zvezda



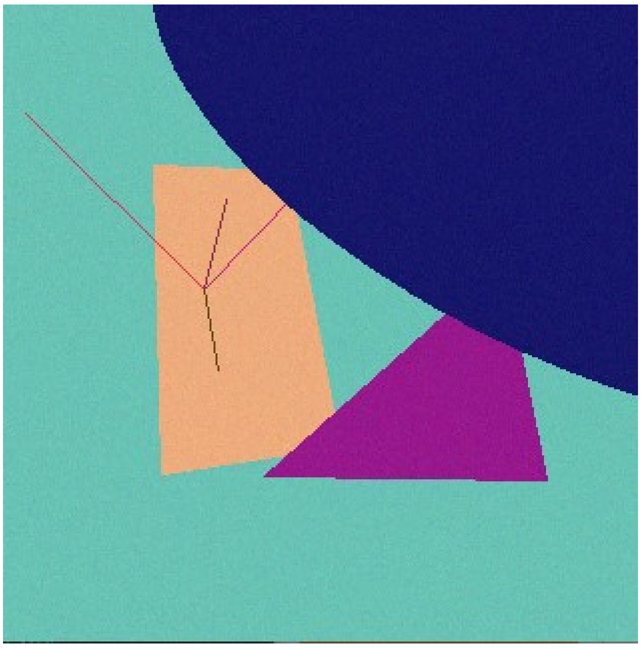
```python
def generate_convex_quadrilateral_points(min, max):
    while True:
        x1, y1 = random.randint(min, max), random.randint(min, max)
        x2, y2 = random.randint(min, max), random.randint(min, max)
        x3, y3 = random.randint(min, max), random.randint(min, max)
        x4, y4 = random.randint(min, max), random.randint(min, max)
        cross_product1 = (x2 - x1) * (y3 - y2) - (x3 - x2) * (y2 - y1)
        cross_product2 = (x3 - x2) * (y4 - y3) - (x4 - x3) * (y3 - y2)
        cross_product3 = (x4 - x3) * (y1 - y4) - (x1 - x4) * (y4 - y3)
        cross_product4 = (x1 - x4) * (y2 - y1) - (x2 - x1) * (y1 - y4)
        if cross_product1 > 0 and cross_product2 > 0 and cross_product3 > 0 and cross_product4 > 0:
            return [(x1, y1), (x2, y2), (x3, y3), (x4, y4)]
def create_image(height, width):
    img = np.zeros((height,width,3), np.uint8)
    img[:,:] = [random.randint(0, 256),random.randint(0, 256),random.randint(0, 256)]
    color = img[0,0]
    return img
def generate_rectangle(img):
    height, width = img.shape[:2]

    if height <width:
        max = height-40
    else:
        max = width-40

    points = generate_convex_quadrilateral_points(0,max)
    cv2.drawContours(img, [np.array(points)], 0, (random.randint(0, 256),random.randint(0, 256),random.randint(0, 256)), -1)
    return img, points
def generate_triangle(img):
    height, width = img.shape[:2]
    p1 = [random.randint(0, height), random.randint(0, width)]
    p2 = [random.randint(0, height), random.randint(0, width)]
    p3 = [random.randint(0, height), random.randint(0, width)]
    cv2.drawContours(img, [np.array([p1,p2,p3])], 0, (random.randint(0, 256),random.randint(0, 256),random.randint(0, 256)), -1)
    return img,np.array([p1,p2,p3])
def generate_grid(img):
```

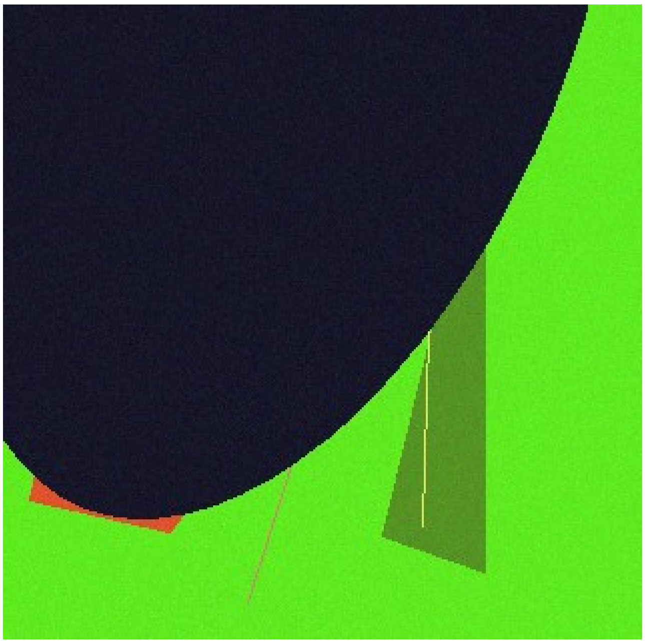
Nato na proizvedeni sliki doda dve vrsti šuma - Aditivni ter Pegast multiplikativni.

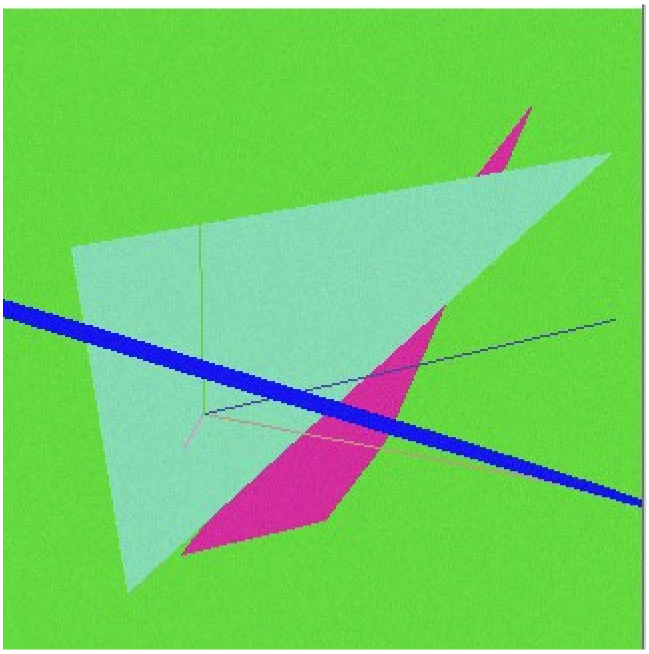


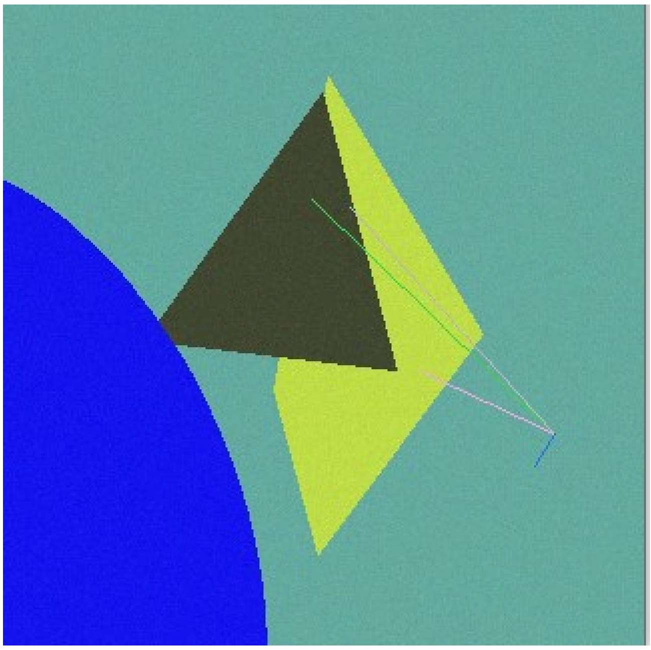
```
def additive_noise(img):
    height, width = img.shape[:2]
    img = img.astype(np.int32)
    #noise = np.zeros((height,width,3), np.int32)
    mean = np.array([0, 0, 0])
    sigma = np.array([0.3,0.3,0.3])
    noise = np.random.normal(mean,sigma,(height,width,3))
    noise = noise-noise.min()
    noise = noise.reshape(height,width,3)
    #cv2.randn(noise, mean, sigma)
    img = img + noise
    img = img/img.max()
    img = img*255
    img = img.astype(np.uint8)
    return img

def multiplicative_noise(img):
    height, width = img.shape[:2]
    img = img.astype(np.int32)
    mean = np.array([1, 1, 1])
    sigma = np.array([0.1,0.1,0.1])
    noise = np.random.normal(mean,sigma,(height,width,3))
    noise = noise.reshape(height,width,3)

    #cv2.randn(noise, mean, sigma)
    img = img*noise
    img = img/img.max()
    img = img*255
    img = img.astype(np.uint8)
    return img
```

Rezultati aditivnega šuma:









Rezultati multiplikativnega šuma: