

Sometimes its tough being the best-Write up

You have been give a image. In the description you have been given that the image was created by ken. If searched online u can see that Ken perlin was the developer of the perlin noise algorithm. So this is a perlin noise image. The description also specifies TomIacko who found out the seed of the home screen display of mind craft and the see was 2151901553968352745 in the description its given that ken used only first 4 most significant bits to create the image. It is also given that he is a python dev and likes noise. If searched online then you can find that noise is a lib in python that has a function pnoise2 to create a perlin noise image. If the description is read carefully, you will find the parameter values needed to be passed to the pnoise2 function. So all you have to do now is write a code using pnoise2 in python to create a perlin noise image with the given parameters and seed. If this image is compared to the image provided then you wont be able to notice any difference, but however few pixels have been changed. And these changed pixels spell out the flag. So you can write a code in python to compare there two images and create a new image which is completely black. And then change the pixels which differ in the 2 images you compared and colour these pixels white in the black image so you can see the pixels that spell out the flag.

The code you need to write is:

```
import noise
from PIL import Image
import numpy as np

# Set the seed for the Perlin noise generator
seed = 2151

# Set the size of the image
width = 1920
height = 1080

# Set the scale of the Perlin noise
scale = 100.0

# Create an empty NumPy array for the image
img_array = np.zeros((height, width))

# Loop through each pixel in the image and generate Perlin noise
for y in range(height):
    for x in range(width):
        img_array[y, x] = noise.pnoise2(x/scale, y/scale, octaves=6, persistence=0.5, lacunarity=2.0,
repeatx=1024, repeaty=1024, base=seed)
```

```

# Create a PIL image from the NumPy array
img = Image.fromarray(np.uint8(img_array * 255))

img.save("original_perlin_noise.png")

# Load the modified image
mod_img = Image.open('Kens_Terrain.png')

# Create a new black image for the changed pixels
new_img = Image.new('RGB', (width, height), color='black')

# Get the pixel access objects for the images
orig_pixels = img.load()
mod_pixels = mod_img.load()
new_pixels = new_img.load()

# Loop through the pixels and identify changed pixels
for x in range(width):
    for y in range(height):
        orig_pixel = orig_pixels[x, y]
        mod_pixel = mod_pixels[x, y]
        if orig_pixel != mod_pixel:
            # Change the pixel in the new image to white
            new_pixels[x, y] = (255, 255, 255)

# Save the new image with the changed pixels in white
new_img.save('changed_pixels_flag.png')

```

where original image is the perlin noise image you create, Kens_terrain is the image provided to you and changed_pixels_flag is the image you create for the flag.

The changed_pixels_flag image is something like this:

(zoom in to see the flag)



{T3chn0blad3_N3v3r_Di3s}

Thus you can see the flag: {T3chn0blad3_N3v3r_Di3s}