Meteor Challenge

Bernardo do Nascimento Nunes

19 de abril de 2024

1 Introdução

Para resolver o desafio utilizei a linguagem Python(v3.8.10) e as bibliotecas numpy e pillow. Essas escolhas se deram pelo fato de eu já ter realizado um trabalho na disciplina de Estrutura de Dados que consistia em transformar uma imagem colorida em uma imagem preto em branco. Nesse aspecto, o trabalho em questão foi produzido em C, mas eu sabendo que envolveria manipulação de matrizes e arrays muito grandes, optei para esse desafio o uso de Python, que possui a biblioteca numpy que facilitaria a resolução do problema. Ademais, a biblioteca pillow surgiu como solução para transformação de imagens em matrizes após eu realizar pesquisas na internet, uma vez que nunca realizei análise de imagens em Python anteriormente.

Além disso, o código possui funções que, mesmo não fazendo parte da solução, me auxiliaram na compreensão dos problemas, que decidi deixar no código para fins de construção do raciocínio. Por fim, realizei o código em português uma vez que se trata de um projeto individual, que não necessita da colaboração de outros possíveis membros, logo, dispensando a necessidade da universalidade de comunicação que o inglês pode proporcionar.

2 Resolução dos desafios

2.1 Count the number of Stars and the number of Meteors

As duas primeiras tarefas são bem parecidas. Após ter transformado minha imagem em uma matriz, apenas fiz a contagem de vezes que cada um dos pontos, os vermelhos (pure red) e os brancos (pure white), aparecem.

Para isso, utilizei uma combinação de duas funções do numpy que formaram a linha np.sum(np.all(matriz == cor, axis=-1)), que basicamente faz a soma da quantidade de vezes que um determinado elemento aparece na matriz. Bem simples e eficaz, talvez o único ponto diferente possa ter sido a criação de uma função para isso, uma vez que quis tornar o código mais modular e utilizar o benefício de conseguir nomear os métodos mais explicitamente de acordo com seu papel, já que contar_cores_especificas é mais auto descriptivo que np.sum(np.all(matriz == cor, axis=-1)).

2.2 Meteors falling on the Water

Para essa parte eu apenas verifiquei se na coluna do meteoro havia um pixel azul (pure blue), uma vez que não tinha pixels azuis abaixo dos pretos, o que tornou a solução bem mais fácil. Portanto, apenas realizei manipulações utilizando as funções do Numpy.

2.3 Hidden Phrase

O desafio bônus eu não consegui concluir. Contudo, tentei diversas coisas e acho válido citar.

Inicialmente, pensei que poderia haver pixels com tons levemente distintos da cor do céu, então criei uma nova imagem com a cor do céu totalmente diferente, para destacar os outros pixels, sem sucesso.

Após isso, pensei na possibilidade de gerar envoltórias convexas com um determinado número de pontos no céu para ver se de alguma forma era formado caracteres. Entretanto, a soma dos pontos do céu não era um número divisível por 175, o que tornaria arbitrária minha tentativa de solução, já que

eu precisaria definir um número aleatório de pontos para gerar a envoltória de cada caracter. Perceba que: as primeiras propostas se tratavam de situações mais visuais e que eu senti não serem o caminho da solução.

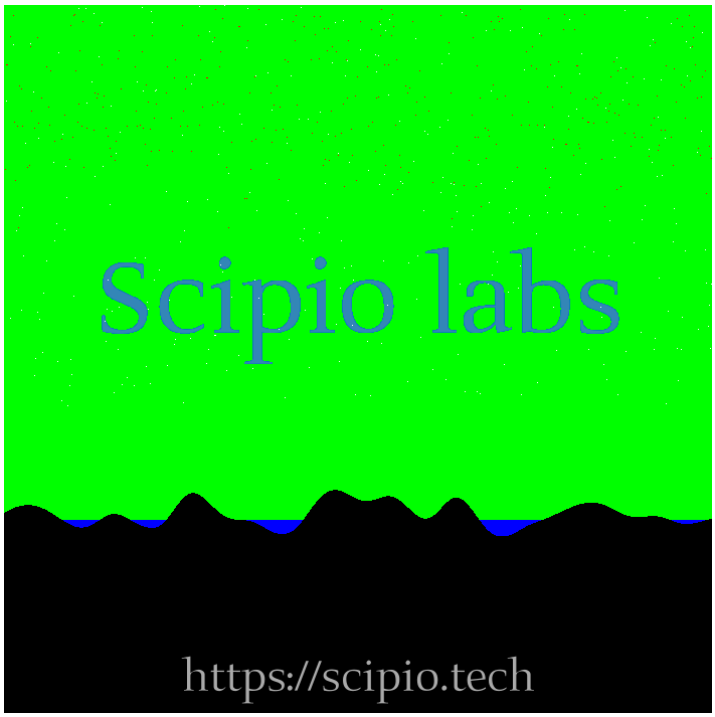

Figura 1: Exemplo de imagem com cores alteradas.

Assim, após perceber que possivelmente a mensagem estaria de alguma forma oculta, visualmente inalcançável mesmo com manipulações, parti para soluções que envolviam conversão da representação visual dos pontos para letras, uma vez que pelas dicas foi dado a entender que se usaria as funções já implementadas. Assim, a contagem/identificação dos pontos vermelhos e brancos na imagem foi o foco do código para solução das outras tarefas do desafio, o que era um forte indicativo da minha premissa.

Nesse sentido, pensei bastante acerca de transformações binárias para caracteres ASCII. Contudo, para fazer essa análise precisava de haver 175 "algos", uma vez que é o número de caracteres. Esses "algos" poderiam ser número de linhas, colunas, estrelas, entre outros. Fazendo análise desses possíveis elementos não achei nada que tivesse o número 175 relacionado. Portanto, mesmo acreditando que a solução esteja atrelada aos pixels e conversão de binários para ASCII não consegui implementar nenhuma solução que gerasse uma mensagem coerente.

3 Execução

Para executar o código Python é essencial ter um ambiente Python com as bibliotecas NumPy e Pillow (PIL) instaladas, que são utilizadas para manipulação de matrizes e imagens, respectivamente. Ademais, caso for puxado do repositório apenas o arquivo .py, uma imagem chamada 'meteor_img.png' deve estar acessível no caminho especificado ou ajustado no código.

Assim, após a execução é esperado uma saída no formato abaixo:

```
bernardo@bernardo-H510M-H:/Area de Trabalho/GitHub/meteor$ python3 meteor.py
Estrelas: 315
Meteoros: 328
Numero de meteoros na agua: 105
```

4 Resultados obtidos

Por fim, os resultados finais do desafio estão dispostos na Tabela 1.

Elemento	Quantidade
Estrelas	315
Meteoros	328
Meteoros na água	105

Tabela 1: Resultados dos experimentos.