* Meteor Challenge (Part 1)

Tasks:

Count the number of Stars
Count the number of Meteors
If the Meteors are falling perpendicularly to the Ground (Water level), count how many will fall on the Water
(optional) Find the phrase that is hidden in the dots in the sky.
HINT 1: 175 Characters
HINT 2: Most of the last tasks' code can be reused for this one

Please, send us the result and code you used to solve the tasks above. Explain how you achieved the results in each question. Good work!!

Subject: [CHALLENGE] [METEOR] Marcos Vinicius Vargas Mello

Answers:

Number of Stars	315
Number of Meteors	328
Meteors falling on the Water	105
(optional) Hidden Phrase	"It's not about how hard you hit. It's about how hard you can get hit and keep moving forward. How much you can take and keep moving forward" - Rocky Balboa/Sylvester Stallone

Pixel Ref:

(pure white) Stars
(pure red) Meteors
(pure blue) Water
(pure black) Ground

1. Estratégia de Implementação da Solução

No início do desafio, optei por utilizar a linguagem Python devido à ampla variedade de bibliotecas que ela oferece, as quais poderiam facilitar a implementação, como a PIL (Python Imaging Library - utilizada por meio do fork Pillow) ou o OpenCV. Após uma análise das funcionalidades, foi decidido utilizar a PIL, pois atende plenamente às necessidades do projeto, uma vez que não se faz necessário o uso dos recursos adicionais que o OpenCV fornece - como análise de imagens em tempo real.

Além disso, a fim de otimizar a classificação de meteoros caindo na água no terceiro desafio, foi definida uma ordem específica para a leitura da imagem. Essa, inicia-se no canto inferior direito da imagem, percorrendo primeiramente as linhas de baixo para cima (da altura - 1 até 0) e, dentro de cada linha, escaneando os elementos da esquerda para a direita (de 0 até largura - 1). Por conta dessa abordagem, podemos identificar todas as posições necessárias para a solução do desafio em tempo de complexidade O(altura * largura).

2. Implementação

    a. Contagem das Águas:

Inicialmente, identificam-se todas as posições de água ao verificar a cor dos pixels correspondentes - onde RGB = (0, 0, 255) representa água. Essa informação é armazenada em um array is_water, que contém 1 para posições com água e 0 para posições sem água. Esse, será utilizado na contagem de meteoros caindo na água, pois por meio do acesso às posições deste array será possível definir se um meteoro cairá ou não em uma posição com água.

    b. Contagem de Meteoros e Estrelas:

Após a identificação das áreas de água, ocorre a contagem de meteoros e estrelas com base na cor de seus pixels - RGB = (255, 0, 0) para meteoros e RGB = (255, 255, 255) para estrelas. A presença de meteoros e estrelas em cada posição do eixo x (largura) é também registrada em arrays em seus respectivos arrays, sendo eles is_meteor e is_star.

        i. Contagem de Meteoros Caindo na Água:

Durante a contagem de meteoros, é verificado também se cada meteoro cai sobre uma área de água. Esse contagem é realizada somando o valor de is_water na posição x que está atualmente sendo verificada (1 ou 0) ao valor total registrado em meteors_into_water.

    c. Decodificação da Frase Oculta:

Para identificar a frase escondida, inicialmente testei esteganografia para verificar se havia uma mensagem oculta na imagem por meio dessa técnica, mas sem sucesso. Após isso, com base na dica sobre o uso de códigos das outras etapas, montei as arrays is_meteor e is_star com base na is_water para tentar chegar em um código binário - que dá-se por conta da natureza das mesmas. A mensagem foi então extraída e decodificada, convertendo cada byte em um caractere por meio do ultimo loop for.