

Para responder os desafios foram utilizados a linguagem Python com auxílio de bibliotecas como matplotlib, OpenCV e PIL.

1- O primeiro passo se resolve ao percorrer cada pixel da imagem verificando o seu valor BGR (pois o OpenCV trabalha com BGR) e ao encontrar um pixel branco um valor é adicionado ao contador de estrelas.

2- O segundo passo se resolve de forma semelhante ao primeiro, trocando apenas a o valor BGR buscado e o contador que será incrementado será o contador de meteoros.

3- Para resolver o terceiro desafio eu procurei a camada equivalente ao nível do mar e removi as montanhas da imagem. Ao descobrir que o pixel 512 (de altura) era o equivalente ao nível do mar, percorri todos os pontos vermelhos e verifiquei quais deles possuíam um ponto azul no pixel (512, x), sendo x a posição horizontal do pixel vermelho comparado. Dessa forma foi possível saber se o meteoro vai cair na água ou na terra.

4- Para o quarto desafio eu procurei e tentei várias abordagens:

- Tentei organizar os pixels todos na mesma horizontal para observar se aparecia algo que me ajudasse. Em seguida tentei substituir vermelho/branco por 0/1 e 1/0 para testar em um tradutor de código binário.
- Também repeti a tentativa acima, porém utilizando código morse no local de código binário, de forma que vermelho/branco valeriam -/. e ./-
- Simulei a queda dos meteoros removendo os meteoros da imagem. Inicialmente testei apenas os que estivessem sobre a água e depois tentei sem nenhum meteoro. Em seguida apliquei novamente as técnicas de código binário e código morse.
- Apliquei esses processos na imagem original, sem alinhar os pontos.

Mesmo após várias tentativas utilizando abordagens diferentes, não consegui encontrar a frase escondida. Acredito que cheguei perto pois ao decodificar os valores binários obtive 176 caracteres, muito próximo dos 177 que estavam descritos no desafio. Talvez com mais algumas tentativas eu conseguisse encontrar a frase.