

Relatório da Execução do Projeto 1 de Fundamentos:  
Antonio Galvão Martins Neto - 2557908  
BSI - 16/05/2023

Parte 1 - Entender as lógicas, fazer a Função 1 e mudar o endereço no main\_codifica:

Ao ler o documento de instruções a primeira dificuldade que tive foi de entender a lógica da compressão dos números hexadecimais. Depois que eu percebi o que tinha que fazer para comprimir os bits tive dificuldade para entender a lógica da descompressão da imagem, após ficar a par das lógicas o processo de compreensão do problema ficou um pouco mais fácil.

Depois disso, ao tentar implementar a minha lógica ao código estava ocorrendo um problema na compilação porque eu não tinha percebido que o parâmetro da função `enviaByteRBD` era um `unsigned char`, depois de declarar as variáveis corretamente e tentar rodar o programa ocorreu um problema (esse definitivamente pior que o anterior) em que a pasta dentro dos meus arquivos não estava sendo encontrada pelo programa e aparecia a seguinte mensagem: "Erro abrindo imagem ..". Onde depois dos dois pontos aparecia o endereço colocado dentro do programa, depois de mandar mensagem pro professor Pigatto e conversar com alguns amigos que tinham passado por esse mesmo problema percebi que eu tinha que mudar algumas coisas dentro das minhas pastas salvas.

Após fazer isso e tentar rodar o programa, ocorreu novamente o mesmo erro. No fim, decidi colocar o endereço de memória completo (desde C: Documentos) e ao fazer isso deu certo e meu programa rodou e resultou num arquivo `rbd`.

Parte 2 - Fazer a Função 2 e mudar o endereço no main\_decodifica:

Depois de fazer a função 1, montei a função 2 que estava bem mais fácil em comparação a 1. A lógica era um pouco parecida com a da função 1, mas mudava algumas coisas. Por exemplo, ele move pra esquerda antes de mover pra direita e que eu precisaria de uma variável auxiliar que seria utilizada pra ter o seu valor alterado sempre e que depois das operações serem feitas ela seria enviada como parâmetro da função `enviaPixel` e com isso a imagem seria descomprimida.

Não demorei tanto tempo na função 2, logo no terceiro teste o programa já estava rodando. E pra ver se estava funcionando direito eu testei com os arquivos de teste do professor, após isso eu só mudei uma coisa no meu programa que foi o laço de repetição. Eu estava usando o "do while" e mudei pra "while" e eu chamava a função antes do laço, coisa que eu não fazia na estrutura anterior. Fiz essa mudança pra evitar erros na hora de rodar o programa, se logo no primeiro byte a função retornasse `0xFFFFFFFF` (que é o valor de parada do laço) ele rodaria do mesmo jeito, porque o laço "do while" não checa o valor de parada na primeira vez que você roda o programa e nesse caso daria erro na hora de continuar.

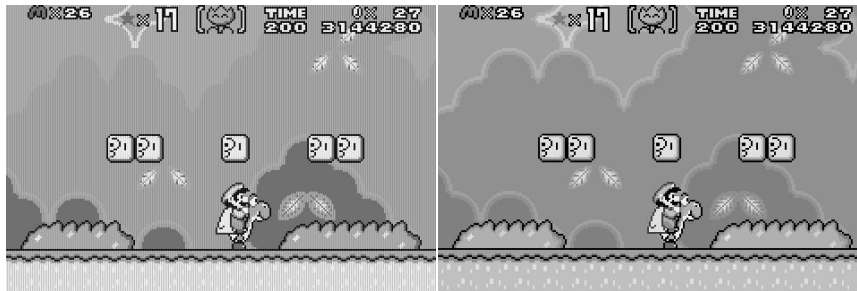
Parte 3 - Checar se a Função 1 estava funcionando e arrumá-la:

Depois de testar a Função 2 com o arquivo RBD do professor, fui testar com o meu arquivo e vi que estava dando erro e apresentava uma imagem que não tinha nenhuma relação com a imagem original e aí eu comecei a segunda jornada para ver o que tinha de errado com a minha função.

Após abrir a função já percebi o que tinha dado errado, era que eu não tinha feito-a de forma automatizada, eu fiz o programa de forma que ele só funcionasse perfeitamente se `n_bits` fosse 2. Quando percebi, pensei em um jeito de automatizar o processo de compressão, então criei uma constante chamada "BITS" que teria o valor igual a 8, porque eu vi que esse valor repetia diversas vezes no programa, tanto dentro do laço de repetição (para você comprimir um byte, era preciso rodar dentro do laço while um laço de repetição que acontecia `BITS/n_bits` vezes), quanto na hora de atribuir as variáveis (a quantidade de vezes que o valor que estava atribuído à função

pegaProximoPixel iria ser jogado para direita com >> e a quantidade de vezes que ele seria jogado para esquerda com <<) eram iguais a BITS - n\_bits).

Quando fiz isso a Função 1 ficou automatizada, mas depois de testar percebi que tinham ficado pequenas linhas cinzas na imagem que era gerada e essa era a única diferença da imagem que eu tinha gerado e a imagem gerada pelo professor. Fiquei muito tempo pensando como que eu tirava essas linhas cinzas que cortavam a imagem e esse foi o maior desafio que tive no projeto inteiro, porque não era um erro na lógica do programa, ela estava certa, meu erro estava na declaração de uma variável. Como a variável que seria utilizada como parâmetro da função enviaByteRBD era unsigned char, eu pensei que era para declarar ela sendo igual a '0' (com aspas simples) e depois de mudar várias vezes a função tentando achar o erro, eu mudei a declaração dessa variável e igualei somente a 0 (sem aspas simples).



Acima, foto mais à esquerda era o meu teste, foto à direita o teste do professor. Percebe-se que a diferença são as pequenas linhas cinzas “cortando” a imagem do meu teste.

#### Parte 4 - Desafio e problemas com as declarações de variáveis

Na aula do dia 11/05/2023, destinada a tirar dúvidas sobre o projeto, eu fiz o desafio. Percebi ao ler as palavras “geração aleatória de números” que era para uma função rand. Depois que pensei e implementei a minha lógica, vi que tinha dado errado porque a imagem estava ficando totalmente “despixelada”, depois de uns dias pensando no que poderia estar errado percebi que meu erro estava na declaração das variáveis. Depois de mudar unsigned int para unsigned char meu programa rodou e a imagem estava bem melhor aparente do que a imagem da primeira vez que tentei fazer o desafio.

Respostas dos questionamentos sobre o desafio:

- a) O preenchimento com zeros escurece a imagem e deixa alguns detalhes de fora.
- b) O preenchimento de intensidade média deixa a imagem mais escura ainda e há uma perda grande de detalhes.
- c) O preenchimento com valores aleatórios altera a pigmentação da imagem e deixa ela um pouco “borrada”.
- d) O preenchimento de intensidade média com valores aleatórios faz com que a imagem fique mais pigmentada e muito “borrada” ao ponto de não entender direito o conteúdo da imagem.