

Universidade Federal de Minas Gerais

# Documentação do Trabalho Prático: PhotoCommand

Lucas Augusto F. De Oliveira  
Rafael Wemerson S. Porto

Turma: M  
Disciplina: AEDS1

Belo Horizonte/2014

## Sumário

O objetivo do trabalho é criar um editor gráfico de modo texto que gera figuras do tipo bitmap de no máximo 250 linhas e 250 colunas. O editor a ser criado é um editor que funciona a partir da linha de comando. Com certos comandos pré-estabelecidos ele edita um bitmap conceitual, pois para esse problema utilizaremos uma ideia de bitmap. O programa deve ignorar comandos inválidos.

# Descrição dos algoritmos

## Estrutura de Dados

Para implementação do trabalho foi criado uma estrutura contendo uma matriz com as configurações máximas, o nome do arquivo e os limites de uma matriz específica fornecida pelo o usuário.

## Leitura de Comandos

Para se fazer a leitura e análise dos comandos digitados, primeiramente consideramos que se o comando é válido consequentemente a quantidade de parâmetros e os tipos de parâmetros para seu respectivo comando são válidos. Na linha de comando, para ser analisado o comando é lido em partes, o primeiro carácter é recebido e nele é testado um switch que analisa os comandos válidos se não é válido toda a linha é ignorada e se é válido a leitura dos parâmetros ocorre e estes são passados para as devidas funções de acordo com o comando.

## Funções e Procedimentos

### **zero(bitmap bp, char dados[MAX\_LIN][MAX\_COL]):**

Recebe a estrutura do bitmap e a matriz de dados da estrutura de bitmap e "colore" todos os termos da mesma com a "cor" O.

### **colorir(bitmap bp, char dados[MAX\_LIN][MAX\_COL], int x, int y, char cor):**

Recebe a estrutura do bitmap e a matriz de dados da estrutura de bitmap, as coordenadas do ponto e a cor escolhida pelo usuário. Essa função é uma das principais funções do programa, devido ao fato de ela ser chamada em outras três funções e basicamente por ser a fundamental função na edição de uma imagem.

### **colorir\_linha\_vertical(bitmap bp, char dados[MAX\_LIN][MAX\_COL], int x, int y1, int y2, char c) :**

Recebe a coordenada x para indicar qual coluna deve ser colorida e as coordenadas y1 e y2 para indicar onde deve começar e onde deve terminar de colorir. Dentro dessa função, a função colorir é chamada para colorir cada "pixel".

### **colorir\_linha\_horizontal(bitmap bp, char dados[MAX\_LIN][MAX\_COL], int x1, int x2, int y, char c):**

Recebe a coordenada y para indicar qual linha deve ser colorida e as coordenadas x1 e x2 para indicar onde deve começar e onde deve terminar de colorir. Dentro dessa função, a função colorir é chamada para colorir cada "pixel".

### **desenhar\_retangulo(bitmap bp, char dados[MAX\_LIN][MAX\_COL], int x1, int x2, int y1, int y2, char c)**

Recebe as coordenadas x1, y1, x2, y2 para indicar quais são os vértices do retângulo, onde o retângulo começa e onde ele termina. Essa função utiliza a função colorir\_linha\_horizontal para desenhar o retângulo preenchido.

### **desenhar\_bitmap(bitmap bitmapsSalvos[MAX\_BITMAPS], int numBitmaps):**

Recebe um vetor de estruturas do tipo bitmap e o número total de bitmaps armazenados no sistema e com esses dados imprime todos os bitmaps no formato nome do bitmap e dados do bitmap.

## **Decisões de implementação**

Uma das decisões tomadas mais importantes foi a de mostrar/imprimir os bitmaps gerados pelos comandos apenas quando um programa recebe o comando de fechamento. A partir dessa decisão o nosso programa armazena através de um vetor de bitmaps as estruturas do bitmaps que foram editados e salvos podem assim imprimir todos no final.

Isso também cria certas limitações como na quantidade de bitmaps que podem ser editados em uma determinada "sessão" do programa.

## Testes

### Teste Padrão

I 5 6

C 2 3 A

S one.bmp

G 2 3 J

R 1 1 5 6 J

V 2 3 4 W

H 3 4 2 Z

S two.bmp

F

### Teste Padrão Com Comando Inválido Seguido de Válido

I 5 6

C 2 3 A

G C 2 3 J

S one.bmp

G C 2 3 J

R 1 1 5 6 J

V 2 3 4 W

H 3 4 2 Z

S two.bmp

F

### Teste Padrão com Coordenadas Invertidos no Retângulo

I 5 6

C 2 3 A

G C 2 3 J

S one.bmp

G 2 3 J

R 5 6 1 1 J

V 2 3 4 W

H 3 4 2 Z

S two.bmp

F