

Hands On Project 1 - Paint no terminal

Autoria do Programa: William Carrara Orlato

Contato: williamorlato@usp.br

● Introdução

O trabalho foi realizado durante o treinamento do Zenith EESC/USP do qual consta no projeto de um programa na linguagem C para a criação e modelagem de imagens em formato PGM.

Os desenhos propostos eram de: linha, círculo, disco e retângulo. Que poderiam ser desenhados por meio de comandos aplicados ao programa.

● Objetivo

O objetivo do projeto é adquirir e aplicar conhecimento sobre programação especificamente sobre linguagem C em diversas áreas desta ciência.

● Imagem

O formato PGM é um tipo de arquivo composto de 255 tons de cinzas compreendendo sua cor mínima o preto, e sua máxima o branco.

Para o manuseio desse tipo de imagem é necessário a inserção de alguns pré requisitos:

```
<magic number>  
<largura da imagem>, <altura da imagem>  
<números de tons de cinza >  
<pixel(0,0)> ... <pixel (0,n)>  
.  
.  
.
```

Onde, magic number é um identificador que no caso do PGM seria "P2", largura e altura da imagem apresenta seu número de pixels nas direções de x e y, o número de tons de cinza seria o máximo de tons trabalhados, e tudo que for colocado em baixo faz parte da matriz de pixels da imagem.

Sendo assim, para todas as imagens trabalhadas foram colocados esses elementos com pré parâmetro, antes do trabalho da matriz de pixels

● Funcionamento

O funcionamento do programa se baseia em alguns comandos que podem ser inseridos no terminal para seu funcionamento, sendo restritamente impostos da seguinte maneira:

CREATE <width> <height>

Para a inicialização de uma matriz imagem em formato PGM.

EXPORT <name>

Para salvar a imagem com o nome selecionado.

LINE <x1> <y1> <x2> <y2> <color>

Para a criação da linha que passará pelos dois pontos com a cor digitada.

CIRCLE <x> <y> <radius> <color>

Para a criação de um círculo com a origem em (x,y) e raio = radius com a cor digitada

DISK <x> <y> <radius> <color>

Para a criação de um círculo da mesma maneira que o anterior porém preenchido seu interior.

RECT <x> <y> <width> <height> <color>

Para criação de um retângulo centrado em (x,y) que possui altura = height e largura = width com a cor digitada.

Observação - Caso algum comando for digitado de forma errada o programa poderá:

1. Pedir a inserção de um novo comando - Para problemas de digitação incorreta da semântica do comando
2. Apresentar uma mensagem de "ERRO!" - Para problemas de lógica, como tentar selecionar uma área da qual não existe na imagem
3. Apresentar os dados incorretos - Para problemas de digitação errada dos moldes apresentados, como acrescentar um espaçamento onde não deveria entre os dados.

Com a digitação correta dos dados, o programa apresentará uma resposta visual sinalizando ao operador a efetivação do processo como:

```
Digite o Comando: CREATE 600 400
Criando Imagem...
Digite o Comando: _
```

Nesse exemplo “Criando Imagem...” revela que programa efetuou o que foi pedido, no caso a criação da imagem.

● O programa

O programa apresenta 7 funções mais a main (rotina de inicialização). Cada rotina pode ser encontrada no programa por uma divisão visual:

```
//=====
//                                     <nome>
//=====
```

Onde no espaço indicado consta o nome de cada uma das funções apresentadas no programa que são:

1. main
2. Comandos
3. Create
4. Export
5. Line
6. Circle
7. Disk
8. Rect

Além disso, o programa apresenta uma função Struct que possui todos os pré requisitos para uma criação de uma imagem PGM.

- **main:**

A função da main nesse programa consta especificamente para a chamada de outras funções que compõem o programa, sendo assim, ela sozinha não realiza nenhuma tarefa dos comandos digitados, apenas redireciona e salva dados do programa.

- **Comandos:**

Esta função apresenta a leitura de todos os comandos digitados, também fazendo a conversão de dados do tipo char para int. Assim possibilitando a leitura direta do comando e enviando para as próximas funções.

De exemplo o comando RECT, onde é necessária a inserção de dois pontos, a largura, a altura e a cor da imagem. Assim, a função Comandos já torna esses dados ditados em int, facilitando futuras leituras.

- **Create:**

Nesta função o programa inicializa os dados necessários para uma imagem de tipo PGM, além de acrescentar os parâmetros de altura e largura da imagem enviados pela função Comandos quando digitado o comando CREATE.

- **Export:**

Esta função do programa apresenta apenas o objetivo de salvar tudo que foi aplicado, com o nome passado pela função comando quando ditado EXPORT, criando assim um arquivo .pgm

- **Line:**

Esta função utiliza dos dados ditados no comando LINE da função Comandos: A partir deles calcula o coeficiente angular e termo independente da reta, assim imprimindo-a na matriz de pixels.

- **Circle:**

Esta função utiliza dos dados ditados no comando CIRCLE da função Comandos: Ela usa da fórmula da semicircunferência centrada na origem - $y = \sqrt{r^2 - x^2}$ para desenhar duas semicircunferências uma voltada cima e outra para baixo que posteriormente são levadas para a centralização no ponto desejados.

○ Disk

Esta função utiliza os dados ditados no comando DISK da função

Comandos: Para assim como a função Circle utilizar da fórmula da semicircunferência (visualizar no tópico anterior) para inserir duas semicircunferências na matriz de pixels, porém diferentemente de Circle, esta função acrescenta um preenchimento interno à circunferência, realizando isto com a impressão de pixels até o fim dos limites da circunferência desenhada, tanto em x quanto em y.

○ Rect

Esta função utiliza os dados ditados no comando RECT da função

Comandos: Para desenhar um retângulo na matriz de pixels, ela exerce isso utilizando a largura e altura do retângulo que são divididas por dois e acrescentadas ou diminuídas do ponto central para assim encontrar os vértices do retângulo e traçando seus lados.