

Hands-on Project 1 - Paint no Terminal

Autor, João Matheus Siqueira Souza

Departamento de Sistemas Embarcados, Zenith EESC/USP

Objetivo

Familiarizar os membros com o desenvolvimento de projetos de software reais, modelagem do problema de forma computacional, criação de bibliotecas e boas práticas de programação.

Imagens PBM|PGM|PPM

O formato PBM (Portable Bitmap) engloba três tipos de imagens: preto e branco, escalas de cinza, e colorido. Esse formato tem uma estrutura comum que se aplica aos tipos de imagem, respectivamente, PBM (Portable BitMap), PGM (Portable GrayMap) e PPM (Portable PixMap). Esse formato foi criado inicialmente pra permitir o envio de imagens via correio eletrônico, que não permitia o envio de arquivos anexados até então.

Um arquivo de imagens PGM tem a forma dada pelos seguintes campos:

```
<magic number>
<altura da imagem "m" pixels> <largura da imagem "n" pixels>
<número tons de cinza>
<valor do pixel (0,0)> ... <valor do pixel (0,n)>
.
.
.
<valor do pixel (m,0)> ... <valor do pixel (m,n)>
```

O Identificador do tipo de formato (designado por “magic number”), é determinado de acordo com a tabela 1, enquanto os outros campos dependerão da imagem em si.

Tipo	ASCII	Binário
PBM	P1	P4
PGM	P2	P5
PPM	P3	P6

Table 1:Relação entre Formato e Identificador

Para mais informações sobre o formato PGM, deixamos [aqui](#) o link para a documentação.

Proposta

A ideia com esse projeto é criar uma espécie de Paint que utilizará comandos no terminal para criar uma imagem em escalas de cinza. Os comandos mínimos para o funcionamento do sistema são:

- **CREATE <width> <heigth>**: Esse comando deve criar uma estrutura que permita a manipulação de uma imagem PGM de dimensões width x heigth;
- **EXPORT <name>**: Esse comando deve salvar a imagem no diretório presente com o nome 'name';
- **LINE <x1> <y1> <x2> <y2> <color>**: Esse comando deve criar uma linha, de cor dada por 'color', que se inicia no pixel de coordenadas (x1, y1) e vai até o pixel de coordenadas (x2, y2);
- **CIRCLE <x> <y> <radius> <color>**: Esse comando cria um círculo de raio dado por 'radius' centrado no pixel de coordenadas (x,y) de cor dada por 'color';
- **DISK <x> <y> <radius> <color>**: Esse comando cria um disco de raio dado por 'radius' centrado no pixel de coordenadas (x,y) de cor dada por 'color';
- **RECT <x> <y> <width> <heigth> <color>**: Esse comando cria um retângulo, centrado no pixel de coordenadas (x,y) .

Um exemplo de entrada para o software é:

```
» CREATE 600 400
» RECT 300 200 600 400 0
» DISK 300 200 300 200
» CIRCLE 300 200 200 120
» CIRCLE 300 200 200 60
» CIRCLE 300 200 200 30
» LINE 0 200 600 200 255
» LINE 300 0 300 400 255
» EXPORT my_draw.pgm
```

Essa entrada deve fazer o software criar uma imagem de 600 x 400, criar um retângulo sobre toda sua extensão, pintando-o de preto. Logo após, criar um disco no centro da imagem com cor cinza 200, criar três círculos concêntricos de cores cinza 120, 60 e 30 e criar linhas perpendiculares brancas no centro da imagem. Por fim, deve-se salvar a imagem criada com o nome "my_draw.pgm".

Dicas e Orientações

Não se assuste. Parece bastante coisa e realmente é. Se achar que não sabe por onde começar, acredite, é normal. Tente desenhar o funcionamento do código utilizando blocos. Pense na entrada do usuário como um bloco, o processamento dos dados como outro, o armazenamento como outro e assim vai. Quando terminar pense se já consegue implementar o código. Se ainda não conseguir, tente quebrar esses blocos em sub-blocos, até que a implementação fique mais clara em sua mente. Desenvolva o código de forma modular, primeiro crie tudo que precisa para a entrada do usuário, teste, aprimore. Funcionando? Ótimo, agora passe para outra etapa. Pensar em todos os aspectos de um software de uma vez só é bem complicado e pode atrapalhar mais que ajudar. **Deixe seu código organizado.** Não fique fazendo remendos, gambiarras aqui e ali. Pode funcionar na hora, mas quando tiver que tudo funcionar, *não vai*. Sempre desenvolva o código pensando que uma outra pessoa que nunca o viu vai conseguir entender e trabalhar sobre ele (essa é a alma da programação). Esse problema proposto terá inúmeros tratamentos especiais a serem realizados, não dê principal atenção a eles, primeiro ache uma solução para o problema, depois trate os problemas adicionais. *Por exemplo*, você precisa traçar um círculo de raio 90 pixels. Ótimo, descubra como traçar um círculo de raio 90 pixels, depois se preocupe no que fazer caso o usuário coloque esse círculo em uma região que irá cortar uma parte do círculo. Enfim, boa sorte!

Informações de Contato

- Git: @jmssouza
- E-mail: joamatheus@usp.br