

2º Encontro Português de Computação Gráfica

Outubro de 1989

Porto

Um Gestor Simples de Janelas

Paulo Ferreira, INESC Norte

Mário Seixas, FMP/INESC Norte

F. Nunes Ferreira, DEEC FEUP / INESC Norte

INTRODUÇÃO

Durante a realização de um sistema de aquisição de imagens para Medicina Nuclear, fomos confrontados com a necessidade de realizar uma "user-interface" intuitiva e de fácil compreensão. Para isso, procurámos um gestor de écran, que permitisse a visualização de janelas e uma fácil utilização das rotinas gráficas entretanto desenvolvidas.

A simplicidade era a característica mais desejada do gestor de écran, uma vez que a sua implementação e também a sua utilização, deveriam de ser feitas em linguagem "assembly". Por outro lado, para além de não se fazer sentir a falta de um sistema particularmente sofisticado para a aplicação em vista, a capacidade de processamento disponível era de tal modo reduzida que impedia na prática a implementação de gestores mais "pesados".

Os ambientes mais vulgarizados de utilização de janelas (Microsoft Windows, Macintosh Toolbox e GEM) permitem realizar uma "user-interface" bastante atraente e com uma grande simplicidade de utilização. No entanto, esta simplicidade de utilização para o utilizador-final acarreta uma grande complexidade de programação, normalmente escondida através de "toolboxes" de ajuda à programação.

A realização de um gestor de janelas dessa complexidade é dificultada, não só pela sua envergadura, mas também pelo facto de ser quase obrigatório fazê-lo acompanhar por um "toolbox" de programação, sob pena da sua utilização se tornar muito difícil.

A nossa escolha recaiu sobre um sistema muito simples, baseado em stacks, mas permitindo um aspecto gráfico razoável e com uma utilização muito fácil [1].

O nosso trabalho tomou por base uma placa gráfica desenvolvida localmente [2], cujas características são sucintamente descritas. Imediatamente após esta descrição, faremos uma apresentação do funcionamento das janelas adoptadas e de vários aspectos relacionados com a implementação do gestor de écran respectivo.

1- SUPORTE HARDWARE DE VISUALIZAÇÃO

O suporte hardware de visualização utilizado foi uma placa gráfica compatível com o bus XT/AT e é constituído por um processador gráfico TMS 34010, memória gráfica, paleta de cores e monitor RGB [2].

A memória gráfica encontra-se estruturada em quatro blocos independentes: dois "buffers" de visualização para imagens (B1 e B2) e dois "buffers" para representação de gráficos em "overlay" (O1 e O2).

Desta forma é possível ter no ecrã uma imagem relativa a um exame de Medicina Nuclear e, sobre ela, mas sem a destruir, definir zonas de interesse, escrever legendas, desenhar cursores, grelhas e menus.

A visualização pode ser feita com qualquer das seguintes combinações: B1, B2, B1+O1, B2+O2, B1+O2, B2+O1. A mudança de uma destas combinações para outra processa-se sem qualquer perturbação no ecrã (mudança durante o retorno vertical).

A resolução é de 512*512 pixels, sendo em B1 e B2 de 8 bits/pixel e em O1 e O2 de 4 bits/pixel.

A paleta de cores é programável e permite em cada momento 256+15 cores (256 para as imagens e 15 para os "overlays") à escolha entre 16.4 milhões.

2- FUNCIONAMENTO DAS JANELAS

O funcionamento básico de um sistema de janelas consiste na definição de um "pseudo-ecrã" sobre o verdadeiro ecrã e, a partir desse momento, toda a visualização de texto e/ou de gráficos é feita dentro dele. Uma janela ao abrir deve esconder tudo o que estiver por baixo e, ao fechar, deve fazer aparecer tudo o que escondeu, para que o sistema dê uma ilusão de maior espaço de visualização. Para que assim aconteça, duas estratégias de funcionamento são imediatas. Na primeira, a responsabilidade de redesenhar o ecrã, cada vez que algum pormenor muda, cabe inteiramente ao

programador, o que acaba por oferecer a facilidade de alterar o tamanho e o estado das janelas (de activas para inactivas e vice-versa).

A segunda estratégia, por nós escolhida, muito mais simples de implementar do que a anterior, oferece um grau de flexibilidade muito menor na utilização das janelas.

Agora, quando uma janela é aberta, a área do ecrã escondida por esta é salva em memória para ser repostada quando a janela for fechada.

O ecrã físico pode ser dividido em vários ecrãs virtuais, "janelas-mãe", podendo posteriormente cada uma delas dar origem a uma família de janelas.

As janelas-mãe podem existir em qualquer número desde que não se sobreponham. Por sua vez em cada uma delas, pode abrir-se um número ilimitado de janelas, sobrepostas ou não, desde que a última janela aberta em cada família seja a única activa e que as janelas sejam fechadas pela ordem inversa à da abertura, nunca mudando de tamanho desde a sua abertura até ao seu fecho.

4- IMPLEMENTAÇÃO DO GESTOR DE JANELAS

O funcionamento do gestor de janelas obriga à definição das seguintes constantes:

Xmin,Ymin	(Canto Superior Esquerdo do Ecrã)
Xmax,Ymax	(Canto Inferior Direito do Ecrã)
TipoTitulo	(Tipo a utilizar nos Títulos)
MargemTitulo	(Largura da Margem Superior da Janela)
MargemNormal	(Largura das Outras Margens)

Cada janela-mãe é associada a uma estrutura de dados, que contém entre outros elementos, a posição do cursor, os limites físicos da janela que está activa, o tipo do texto e as cores principais a utilizar, entre outras variáveis.

Desta estrutura de dados salientam-se as seguintes variáveis:

Cursor	(Posição do Cursor)
EffXmin, EffYmin	(Canto Superior Esquerdo do ViewPort)
EffXmax, EffYmax	(Canto Inferior Direito do ViewPort)
Largura, Altura	(Dimensões da Janela)
Côr0, Côr1, Côr2	(3 Cores diferentes para Tinta)
CôrPapel	(Côr do Background)
TipoActiv	(Tipo a Utilizar para Fazer Letras)

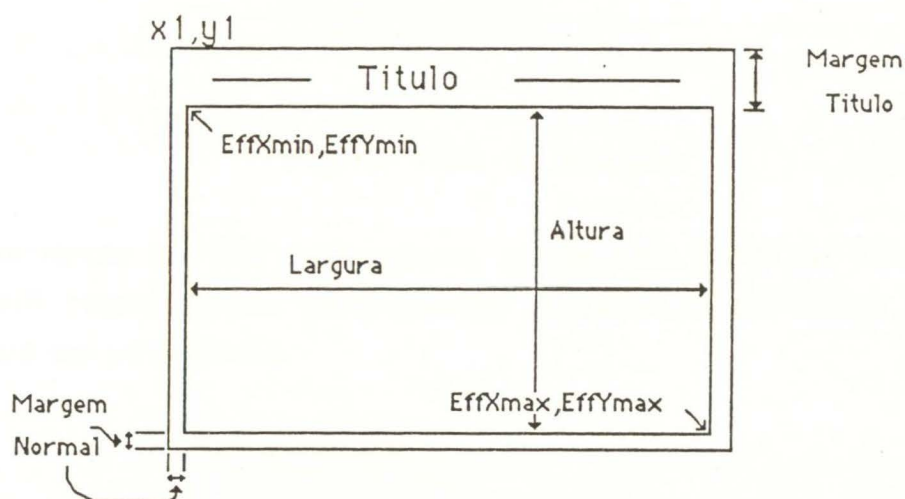


Fig. 1- Aspecto de uma janela

Como se pode ver na figura 1, uma parte da área ocupada pela janela é reservada para uma moldura que contém o título na parte superior.

As rotinas gráficas e as de texto, têm como parâmetro principal o número da janela-mãe em que se deseja desenhar. Dispondo da informação guardada na respectiva estrutura de dados, as rotinas de visualização utilizam e respeitam essa informação para o seu funcionamento.

Cada janela-mãe tem associada a si duas pilhas. A primeira, que guarda as áreas de ecrã que vão ficando debaixo das sucessivas janelas, apresenta-se dividida em partes diferentes, uma vez que as janelas têm normalmente tamanhos diferentes (Fig. 2).

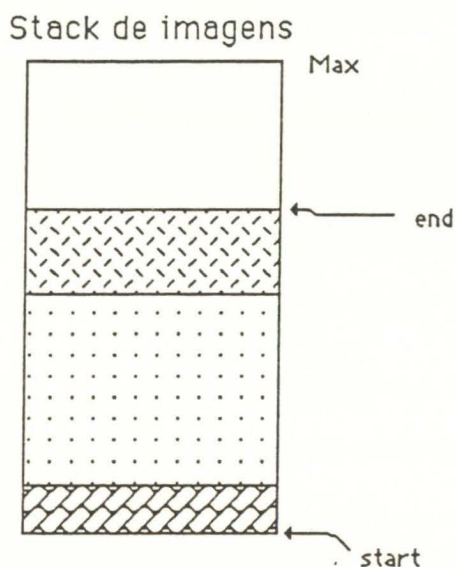


Fig. 2- Pilha de Imagens

Complementando a Pilha de Imagens, existe uma Pilha de Valores (Fig. 3), dividida em partes iguais, para guardar a estrutura de dados activa, antes da abertura de uma janela.

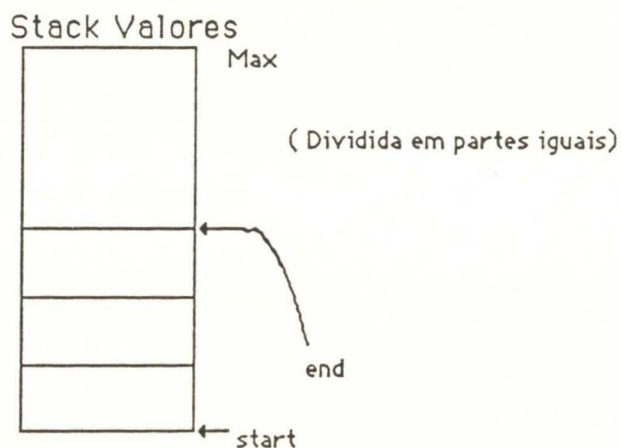


Fig. 3- Pilha de Valores

Na nossa implementação existem as seguintes rotinas globais:

LimparÉcrã(X1,Y1,X2,Y2,Côr Papel)

= desenhar rectângulo com Côr Papel entre X1+MolduraNormal, Y1+MolduraTitulo, X2-MoldNorm, Y2-MoldNorm.

FazMoldura(X1,Y1,X2,Y2,Titulo,CôrPapel,CôrTinta)

= desenhar moldura entre X1,Y1,X2,Y2 e X1+MolduraNormal, Y1+MolduraTitulo, X2-MoldNorm, Y2-MoldNorm e escrever título com TipoTitulo.

Guardar_Variáveis

= guardar variáveis activas (estrutura de dados) na pilha de valores.

Guardar_Stack_de_Imagens(X1,Y1,X2,Y2)

= guardar na pilha de imagens rectângulo compreendido entre X1,Y1 e X2,Y2.

ScrollUp(n)

= fazer "scroll" de n pixels da janela activa, limpando a última linha desta.

As rotinas de visualização de texto, que suporta tipos proporcionais, embora não suporte "kerning", sugere ainda algumas indicações adicionais. A variável Cursor é sempre somada a (EffXmin, EffYmin) para se obter o ponto real do ecrã onde o cursor se encontra, tendo como limites máximos (Largura, Altura).

O algoritmo utilizado para actualizar a posição do cursor durante a visualização de texto é o seguinte:

```
--Se Cursor.X+Largura_Letra > Largura
  então
  {
  Cursor.X=0
  se Cursor.Y+2*Altura_Tipo > Altura
    então
      ScrollUp( Altura do Tipo )
    senão
      Cursor.Y+=Altura do Tipo
  }
```

O pseudo-código para abrir uma janela é, por exemplo, o seguinte:

```
procedure AbrirJanela( X1,Y1,Larg,Alt,Erro,
    Côr do PapelJ, 3 Côres Tinta,Tipo da Janela
    Côr Tinta da Moldura,Côr do Papel da Moldura,Titulo)
```

parâmetros

X1	=coordenada X (pixel) do canto superior esquerdo
Y1	=coordenada Y (pixel) do canto superior esquerdo
Larg	=largura (em pixels) da janela
Alt	=altura (in pixels) da janela
Erro	=devolve o tipo de êrro se algum tiver ocorrido
Côr do PapelJ	=côr do background
Cores Tinta	=côr1, côr2, e côr3 para texto, desenhos e outros
Tipo Janela	=Tipo a utilizar na janela
Côr Tinta da Moldura	
Côr do Papel da Moldura	
Titulo	=string a imprimir na moldura

variáveis locais

Tamanho	=tamanho (em bits) da janela a abrir
X2	=coordenada X (pixel) do canto inferior direito
Y2	=coordenada Y (pixel) do canto inferior direito

```
//      inicializar variáveis
--- Erro = Falso
--- Tamanho = 8 * Larg * Alt           // 1 pixel = 8 bits !!!
--- X2 = X1 + Larg -1
--- Y2 = Y1 + Alt -1

//      verificar se há erros
--- se Tamanho > ( Max Stack Janelas - End Stack Janelas )
      ou
      ( n/ há espaço na stack de valores )
{
  Erro = ' Muito Grande '
  fim_de( AbrirJanela )
}      // stacks de valores e/ou de imagens cheias
```



```

--- se ( não(Xmin < X1 < Xmax )) ou ( não(Ymin < Y1 < Ymax)) então
  {
    Erro = ' Coordenadas não são válidas '
    fim_de( AbrirJanela )
  }
--- se ( não(Xmin < X2 < Xmax )) ou ( não(Ymin < Y2 < Ymax)) então
  {
    Erro = ' Janela não cabe no écran '
    fim_de( AbrirJanela )
  }

// guardar variáveis e imagem
--- Guardar variáveis activas na stack de valores
  Cursor
  EffXmin, EffYmin
  EffXmax, EffYmax
  Largura, Altura
  Tinta 1
  Tinta 2
  Tinta 3
  Côr do Papel
  Tipo Activo
--- guardar área por baixo na stack de imagens
  Guardar_Stack_de_Imagens(X1, Y1, X2, Y2)

// inicialização da janela
--- FazMoldura(X1, Y1, X2, Y2, Titulo, Côres papel e tinta da moldura)
--- LimparÉcrã( X1, Y1, X2, Y2 , Côr do Papel)
--- Tinta 1,2,3 = Côres 1,2,3
--- Tipo Activo = Tipo da janela
--- Côr Papel = Côr PapelJ
--- EffXmin= X1+Margem Normal
--- EffYmin= Y1+Margem Titulo
--- EffXmax= X2-Margem Normal
--- EffYmax= Y2-Margem Normal
--- Cursor=(0,0)

```

Por outro lado o pseudo-código para fechar uma janela é o seguinte:

Procedure FecharJanela(Erro)

parâmetros

Erro = devolve o código do erro se houver algum

variáveis locais

X1 = coordenada X (pixel) do canto superior esquerdo

Y1 = coordenada Y (pixel) do canto superior esquerdo

Larg = largura (em pixels) da janela

Alt = altura (em pixels) da janela

X2 = coordenada (pixel) do canto inferior direito

Y2 = coordenada (pixel) do canto inferior direito

// verificar se há alguma janela para fechar

--- Erro = falso

--- se start_stack_valores = end_stack_valores então

{

Erro = ' Não há janela(s) para fechar'

fim_de(FecharJanela)

}

// copiar imagem para o ecrã

Recuperar_Stack_Imagens

// recuperar parâmetros antigos do ecrã

Recuperar_Stack_Valores

Cursor

EffXmin, EffYmin

EffXmax, EffYmax

Largura, Altura

Tinta 1

Tinta 2

Tinta 3

Côr do Papel

Tipo Activo

5- CONCLUSÃO

Além do gestor de janelas foram desenvolvidas várias rotinas gráficas [3,4,5,8], que permitem desenhar texto, pontos, rectas, elipses e quartos de elipses. As experiências já realizadas são encorajadoras e confirmam o acerto da nossa decisão ao escolhermos um gestor simples para a visualização de janelas.

Presentemente as rotinas encontram-se numa fase de modificação, dada a recente disponibilidade de um compilador de C razoável para o TMS34010. Por este motivo, a passagem de parâmetros está a ser alterada, tendo em vista a compatibilidade das nossas rotinas "assembly" com a linguagem C.

BIBLIOGRAFIA

1- A Simple Windowing System

Bruce Webster

BYTE, March-April 1986

2- Hardware Gráfico para Medicina Nuclear

F. Nunes Ferreira, Mário Seixas, Domingos B. Cruz, J. Moreira da Silva,
Paulo A. Ferreira

ENDIEL 89

3- An Efficient Ellipse Drawing Algorithm

Jerry R. Van Aken

IEEE CG&A September 1984

4- Better Bit-Mapped Lines

Jerry R. Van Aken and Carrel Killebrew Jr.

BYTE March 1988

5- **TMS34010 User's Guide**

Texas Instruments 1988

6- **TMS34010 Assembly Language Tools User's Guide**

Texas Instruments 1987

7- **TMS34010 C Compiler Reference Guide**

Texas Instruments 1988

8- **Procedural Elements for Computer Graphics**

David F. Rogers

McGraw Hill 1985