

# Relatório sobre o desenvolvimento do MODELADOR

Sílvia Delgado Olabarriga

INESC - Projecto CAD/CAM  
Av. Alves Redol, 9 - 2º  
1000 - Lisboa

Abril 1989



## Sumário

1. Introdução .....	2
2. Funções do MODELADOR .....	2
3. Protótipo .....	2
4. Interface com utilizador .....	3
4.1 Criação de malhas de polígonos .....	4
4.1.1 "Slider" .....	4
4.1.1.1 Funcionamento básico .....	4
4.1.1.2 Interface com o MODELADOR .....	4
4.1.1.3 Uso de "sliders" .....	5
4.2 Edição interactiva de malhas de polígonos .....	5
4.2.1 Edição de cor .....	5
4.2.2 Edição de geometria .....	6
4.2.3 "Pick" .....	6
4.2.3.1 Funcionamento básico .....	6
4.2.3.2 Interface com o MODELADOR .....	7
4.2.3.3 Uso de "pick" .....	7
4.2.4 Múltiplas vistas de uma malha .....	7
4.3 Implementação dos mecanismos de interacção com o utilizador .....	8
4.3.1 "Slider" .....	8
4.3.2 "Pick" .....	8
4.3.3 Manipulação directa de um vértice .....	8
5. Estágio actual da implementação do MODELADOR .....	9
5.1 Organização interna .....	10
Referências bibliográficas .....	10
ANEXO 1 - Software gráfico da IRIS .....	11
ANEXO 2 - Comentários sobre algumas publicações a respeito de interacção para modelação de objectos tri-dimensionais .....	14
ANEXO 3 - Comentários sobre as interfaces com utilizador dos programas de demonstração da IRIS .....	19
ANEXO 4 - SELIPS .....	21

## 1. Introdução

O MODELADOR é um editor gráfico para a modelação de objectos tri-dimensionais, desenvolvido para a workstation Cyber 910 (IRIS) com os seguintes objectivos:

- a) permitir a criação e edição de malhas de polígonos quaisquer armazenadas em ficheiros no formato "mesh" [Olabarriaga 89] ;
- b) definir uma interface com o utilizador que possibilite manipulação destas malhas de maneira confortável;
- c) explorar ao máximo as capacidades gráficas e interactivas da referida workstation.

Este relatório descreve o MODELADOR, sendo organizado da seguinte forma: inicialmente as funções do MODELADOR são brevemente apresentadas, sendo depois descrito o protótipo implementado para o Festival Audio-Visual Lisboa 88. Depois é apresentada uma proposta de interface com utilizador, baseada nos resultados obtidos com o protótipo e em estudo de bibliografia sobre o assunto. Por fim, o estágio actual da implementação do MODELADOR é comentado.

## 2. Funções do MODELADOR

O MODELADOR de objectos tri-dimensionais tem os seguintes comandos:

- a) criação e armazenamento em disco de malhas de polígonos no formato MESH através de interface gráfica e interactiva, incluindo especificação de geometria e cor;
- b) edição de malhas de polígonos armazenadas no referido formato, incluindo modificação da cor (de toda a malha ou de facetar individualmente) e das coordenadas de vértices manipulados directamente com o rato;
- c) visualização de malhas de polígonos a partir de diversos pontos de vista e através de variadas técnicas para apresentação de objectos tri-dimensionais em superfícies bi-dimensionais.

## 3. Protótipo

Desenvolveu-se um protótipo do MODELADOR para apresentação no Festival Audiovisual Lisboa 88, realizado no Forum Picoas entre 11 e 20 de Novembro de 1988. A finalidade deste protótipo era múltipla:

- a) apresentação de uma aplicação gráfica e interactiva desenvolvida pelo INESC na exposição paralela ao referido Festival. Era conveniente que se tivesse pelo menos um programa que utilizasse o rato, aproveitando as capacidades gráficas da workstation, já que o sistema "click" não tinha sido desenvolvido com este propósito;
- b) implementação de procedimentos básicos para tratamento de malhas de polígonos no formato do MESH, tais como: armazenamento/recuperação de ficheiros em disco, visualização em "wireframe" e preenchimento das facetar com cores aleatórias;
- c) familiarização com as capacidades gráficas da workstation Cyber 910;

d) observação do comportamento do utilizador diante de algumas técnicas de interacção (uso de menus e rato).

O programa desenvolvido tinha as seguintes características:

a) utilização de apenas uma janela controlada pelo "mex" (gestor de janelas da workstation Cyber 910), em que mostrava-se a malha de polígonos e fazia-se comunicação textual com o utilizador (mensagens curtas e leitura de nomes de ficheiros);

b) selecção de comandos através de menus controlados pelo "mex". Os menus eram accionados pelo botão direito do rato e ofereciam os seguintes comandos:

- carregamento de uma malha a partir de um ficheiro;
- apresentação da malha em "wireframe" (com depth-cueing, em uma cor);
- pintura das facetas da malha com cores aleatórias;
- rotação em tempo-real da malha nos eixos x e y consoante movimentos feitos com o rato: deslocamentos na horizontal eram traduzidos em rotações sobre o eixo y; deslocamentos na vertical eram convertidos em rotações sobre o eixo x. A visualização podia ser feita em "wireframe" ou com preenchimento das facetas;
- escolha interactiva da cor da malha através de "sliders" (ver secção 4.2.1);
- armazenamento da malha num ficheiro;
- fim da execução. O programa pedia confirmação no caso de término sem armazenamento em disco das alterações feitas no decorrer do trabalho.

O desenvolvimento deste protótipo possibilitou contacto efectivo com as funções da biblioteca gráfica da workstation, permitindo a revisão de alguns procedimentos implementados, nomeadamente: gestão de menus; desenho com dois buffers simultâneos (para animação em tempo-real da rotação); uso do rato (leitura do estado dos botões e posição); uso de texto bi-dimensional juntamente com desenho de objectos tri-dimensionais. Os comentários mais importantes a respeito deste assunto foram apresentadas na reunião do Projecto CAD/CAM do dia 14 de Dezembro de 1988, estando seu conteúdo resumido no anexo 1.

Quanto à avaliação da interface com o utilizador, concluiu-se que o uso de menus e a técnica usada para selecção de cor apresentaram bons resultados. O mesmo não pode ser dito, entretanto, da interface para rotacionar a malha; impunha-se a necessidade de definir uma técnica para visualização a partir de pontos de vista variáveis que oferecesse maior controle ao utilizador (secção 4.2.4).

#### 4. Interface com o utilizador

A interface do MODELADOR com o utilizador foi definida com base em dois estudos:

a) da bibliografia existente sobre interacção homem-máquina para modelação de objectos 3D. Os comentários mais importantes sobre os artigos consultados estão no anexo 2;

b) dos programas de demonstração da workstation Cyber 910, que apresentam interfaces com utilizador bastante variadas. O relato sobre o que foi observado é apresentado no anexo 3.

Utilizando os resultados de ambos os estudos, assim como as conclusões obtidas através da avaliação do protótipo (ver secção 3), foi definida uma proposta de interface, que será descrita a seguir.

De maneira geral, foram mantidas as técnicas para interacção com o utilizador em relação ao que havia sido implementado no protótipo: menus controlados pelo mex,

comunicação textual feita através de curtas mensagens apresentadas na parte inferior da janela, "slider" para escolha da cor e rato para manipulação directa da malha de polígonos. Para a implementação dos comandos de criação e edição (que não foram incluídos no protótipo), foram definidos mecanismos especiais para interacção, que serão descritos nas secções a seguir.

## 4.1 Criação de malhas de polígonos

A função que permite a criação de malhas de polígonos pode ser implementada através de diversos procedimentos: parametrização de objectos primitivos (tais como cubo, esfera, etc); entrada livre de coordenadas de vértices e informações topológicas por meio de texto ou desenhos feitos com o rato; deslocamento de uma curva 2D sobre um caminho 3D; utilização das equações de superfícies; etc.

Como a natureza dos dados de entrada varia muito conforme o procedimento empregado na criação destas malhas, é difícil especificar uma interface única que seja simultaneamente genérica e confortável para o utilizador. Por outro lado, a integração destes procedimentos ao MODELADOR passa necessariamente pela homogeneização das respectivas interfaces, já que é altamente desejável que a forma de comunicação entre o utilizador e os diversos componentes do MODELADOR seja regular. A fim de normalizar esta interface, propõe-se que todos os procedimentos utilizem a mesma técnica de interacção, designada por "slider".

### 4.1.1 "Slider"

#### 4.1.1.1 Funcionamento básico

Um "slider" é semelhante a um "valuator" da norma GKS: uma barra horizontal com valores crescentes da direita para a esquerda com uma marca na posição correspondente ao valor currentemente seleccionado (ver figura 1).

Para escolher um valor, o utilizador desloca o cursor do rato sobre a barra na posição correspondente ao valor desejado e pressiona o botão direito. Imediatamente a marca é deslocada para aquela posição e o valor correspondente escrito à direita da barra.

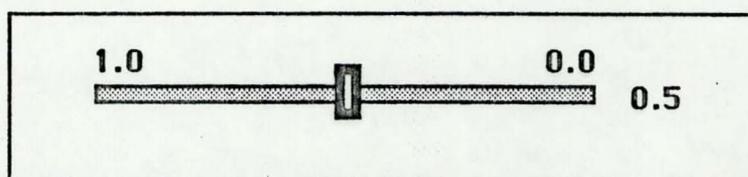


Figura 1 - Slider

#### 4.1.1.2 Interface com o MODELADOR

A interface com o MODELADOR é feita por meio de três funções:

a) `cria_slider`: fixa a posição, o tamanho, os valores das extremidades da barra, a cor para preenchimento do seu interior, etc. A rotina desenha o "slider" com as características indicadas, inicializa variáveis internas de controle e posiciona a marca no valor inicial.;

b) `lê_slider`: lê as coordenadas do cursor e, se este está sobre a barra, desloca para lá a marca e escreve o novo valor. Retorna ao programa principal o valor correspondente à posição actual da marca sobre a barra;

c) `remove_slider`: apaga o "slider" do ecran, recompõe a imagem de fundo (se necessário) e destrói as estruturas de dados internas de controle para ele alocadas.

#### 4.1.1.3 Uso de "sliders"

Esta ferramenta pode ser usada para obtenção dos valores de parâmetros durante a geração de uma malha de polígonos através de um algoritmo qualquer, com realimentação em tempo-real da forma 3D produzida à medida que o valor é alterado. Para tanto, o MODELADOR deve ficar lendo o valor indicado pelo utilizador através do rato, recalculando a malha e redesenhando-a continuamente, até que seja detectada a condição de término ou cancelamento da operação.

## 4.2 Edição Interactiva de malhas de polígonos

O comando de edição de uma malha de polígonos permite basicamente a modificação de sua geometria e das propriedades visuais das facetas, não possibilitando a alteração da topologia (tabela de facetas).

As propriedades visuais são definidas através da cor (no sistema RGB) e de coeficientes usados para cálculo da quantidade de luz reflectida e absorvida pelo material de que o objecto é feito. Para modificar a cor, o utilizador deve indicar as novas componentes RGB, o que é feito por meio da interface descrita na secção 4.2.1. A modificação dos coeficientes será feita utilizando o mesmo tipo de interface, ou seja, através de "sliders" (ver secção 4.1.1). Como a cor das facetas pode ser indicada individualmente, existe a possibilidade de efectuar a modificação sobre uma faceta seleccionada através do rato, usando a técnica descrita na secção 4.2.3.

A alteração da geometria dos vértices da malha de polígonos pode ser feita de diversas formas, porém em contactos com potenciais utilizadores do MODELADOR, determinou-se ser importante haver um mecanismo para manipulação directa dos vértices, permitindo deformação da malha original. A interface com utilizador proposta para este comando é descrita na secção 4.2.2.

### 4.2.1 Edição de cor

O MODELADOR possui um módulo desenvolvido pelo aluno Carlos Reino (IST) que permite a selecção interactiva de uma cor através da indicação do valor de suas componentes no sistema RGB (Red Green Blue) ou no sistema HSV (Hue Saturation Value).

O valor das componentes da cor é seleccionado com o rato através de "sliders" (ver secção 4.1.1). Há três "sliders":

a) no sistema RGB, os "sliders" são preenchidos com tonalidades da respectiva componente (em vermelho, verde ou azul). O valor "0" corresponde ao tom mais escuro, enquanto que "1" corresponde à cor pura.

b) no sistema HSV, um dos "sliders" contém um arco-íris (Hue), outra é toda branca (Saturation) e a terceira é toda preta (Value).

A troca de sistema de especificação de cor é feita pelo menu accionado através do botão direito do rato. A escolha da cor é encerrada quando o utilizador selecciona a opção "fim" deste menu. A malha de polígonos é então desenhada na nova cor, porém o ficheiro permanece inalterado até a execução do comando de armazenamento em disco.

#### 4.2.2 Edição de geometria

A modificação das coordenadas dos vértices de uma malha de polígonos é feita através do comando de edição de forma e tem as seguintes características:

a) o utilizador selecciona com o rato o vértice ou faceta que quer alterar. No caso de escolher uma faceta, a transformação será aplicada simultaneamente a todos os vértices da mesma. O mecanismo usado para selecção de um elemento é o "pick", descrito na secção 4.2.3;

b) indica através dos botões do rato o tipo de movimentação, e o cursor troca de formato para indicar o comando activado; enquanto os botões permanecerem pressionados, o vértice é deslocado conforme os movimentos do rato ("rubber-banding"), assim como todas as arestas incidentes a ele. A transformação aplicada aos vértices depende dos botões premidos:

- botão esquerdo = o vértice desloca-se para cima, baixo, à esquerda ou direita sobre um plano paralelo ao plano em que é feita a projecção para visualização 3D. Este plano é determinado pela posição e orientação da câmara;

- botão do meio + esquerdo = o vértice é movimentado segundo a orientação determinada pela normal ao plano da projecção (para dentro ou para fora do objecto);

- três botões simultaneamente = retorna o vértice à posição inicial ("undo" das modificações feitas enquanto algum dos botões estava pressionado).

c) quando a posição desejada foi atingida, o utilizador solta o botão e as novas coordenadas são armazenadas na estrutura de dados da malha;

d) o botão direito é sempre utilizado para activação do menu, que neste caso oferece as seguintes opções:

- troca o tipo de apontamento: em dado instante, o utilizador pode trabalhar (apontar e editar) com um único vértice ou com todos os vértices que definem o contorno da faceta apontada pelo rato;

- troca o ponto de vista, usando os procedimentos para controle de câmara descritos na secção 4.2.4;

- fim da edição de vértices.

#### 4.2.3 "Pick"

##### 4.2.3.1 Funcionamento básico

"Pick" é um mecanismo de interacção que permite ao utilizador seleccionar uma faceta ou vértice de um objecto tri-dimensional através do rato. Para seleccionar um elemento da malha de polígonos, o utilizador movimenta o rato, cuja posição é indicada no ecrã como uma marca, designada por "cursor". O deslocamento do cursor é monitorado, sendo desenhados em cor diferenciada os elementos da malha que estão sob ele em determinado instante. A selecção de determinado elemento dá-se quando algum botão do rato é pressionado.

A posição do rato é utilizada para calcular as coordenadas tri-dimensionais do ponto do objecto que o utilizador deseja seleccionar. A obtenção das coordenadas 3D de um ponto a partir da posição 2D do rato é feita da seguinte maneira: ligando-se o ponto correspondente à posição da câmara (olho) ao ponto correspondente à posição do rato sobre o plano da

projectão, define-se uma recta; a intersecção desta recta com uma faceta ou sua passagem nas proximidades de algum vértice do objecto determina o elemento seleccionado.

É possível, no entanto, que mais de um elemento seja classificado como "próximo" à referida recta em dado instante. Assim sendo, é necessário definir uma regra que estabeleça prioridades entre os elementos: será seleccionado aquele que estiver mais próximo do utilizador (calcula-se a distância entre o observador e o elemento).

#### 4.2.3.2 Interface com o MODELADOR

No MODELADOR, tal mecanismo é acedido por meio de uma função que acompanha os movimentos do rato, salientando em cor diferente o elemento seleccionado em dado instante. A função retorna apenas quando algum botão foi premido, retornando ao MODELADOR apenas o identificador do elemento (vértice ou faceta) seleccionado.

Para que o elemento sob o cursor seja desenhado em cor diferenciada, é necessário que esta função conheça a geometria do mesmo, o que pode ser feito através de estruturas de dados ou procedimentos recebidos como parâmetro.

#### 4.2.3.3 Uso de "pick"

O mecanismo de "pick" é usado para indicar os elementos que devem sofrer determinada alteração, como mudança de cor de uma faceta ou deformação através da alteração das coordenadas de um conjunto de vértices. O MODELADOR deve desenhá-la malha de polígonos e especificar a transformação de projectão e depois chamar a função de "pick".

#### 4.2.4 Múltiplas vistas de uma malha

O utilizador pode trabalhar simultaneamente com várias vistas do mesmo objecto, podendo criar novas janelas (no máximo 4) e controlar a câmara usada para visualização em cada uma delas através da combinação de botões e movimentos do rato:

- botão esquerdo = o observador desloca-se para cima, baixo, à esquerda ou direita sobre um plano paralelo ao plano da projectão definido pela transformação de visualização daquela janela;

- botão do meio = o ponto de interesse é deslocado sobre um plano paralelo ao plano da projectão naquela janela;

- botão do meio + esquerdo = a imagem é ampliada ou reduzida por manipulação do ângulo de visão (lente);

- três botões simultaneamente = inicializa posição da câmara.

Quando o observador ou o ponto de interesse são modificados, a orientação do plano de projectão é alterada e, conseqüentemente, o plano em que as transformações dos vértices da malha serão feitas.

Sempre que um ou mais botões são pressionados, o formato do cursor que informa a posição corrente do rato é alterado para lembrar o tipo de alteração que será aplicada ao ponto de vista. Além disto, os eixos do objecto são desenhados em cor diferenciada como referencial da posição original.

O botão direito acciona um menu que permite a especificação do tipo de vista e de projecção que deve ser apresentada na janela em questão:

- frontal ou traseira: projecção ortográfica que possibilita a visão da parte frontal ou traseira do objecto, conforme o observador esteja na parte positiva ou negativa do eixo z;

- superior ou inferior: projecção ortográfica que possibilita a visão da parte superior ou inferior do objecto, conforme o observador esteja na parte positiva ou negativa do eixo y;

- lateral: projecção ortográfica que possibilita a visão de um dos lados do objecto, conforme o observador esteja na parte positiva ou negativa do eixo x;

Os dois tipos de projecção oferecidos ao utilizador são complementares: enquanto as ortográficas permitem grande precisão durante a movimentação dos vértices, as perspectivas fornecem vistas mais realistas do objecto modelado.

### 4.3 Implementação dos mecanismos para interacção com o utilizador

#### 4.3.1 "Slider"

O "slider" pode ser implementado quase que directamente com as funções gráficas da Cyber 910:

- leitura da posição e do estado dos botões do rato (funções "getvaluator" e "getbutton");

- sobreposição de imagens sem destruição do fundo, para deslocamento da marca sobre a barra (função "writemask").

#### 4.3.2 "Pick"

O mecanismo para "pick" também pode ser implementado quase directamente com as funções gráficas da workstation em questão:

- identificação de elementos pictóricos localizados sob o cursor (função "pick");

- diferenciação do elemento sob o cursor sem destruição do fundo (função "writemask").

#### 4.3.3 Manipulação directa de um vértice

a) o movimento de um ponto sobre um plano paralelo ao plano de projecção é calculado com base em duas posições do rato. As coordenadas 2D são transformadas para 3D através da aplicação da transformação inversa à visualização, o que é feito directamente por uma função da Cyber 910 ("mapw"). Calcula-se então a diferença entre as duas posições (nos três eixos) e aplica-se às coordenadas que devem ser modificadas (de um vértice, do ponto de interesse ou da posição do observador). Os movimentos do rato são traduzidos da seguinte forma (ver figura 2):

- direita = deslocamento positivo em x;
- esquerda = deslocamento negativo em x;

- frente = deslocamento positivo em y;
- trás = deslocamento negativo em y;

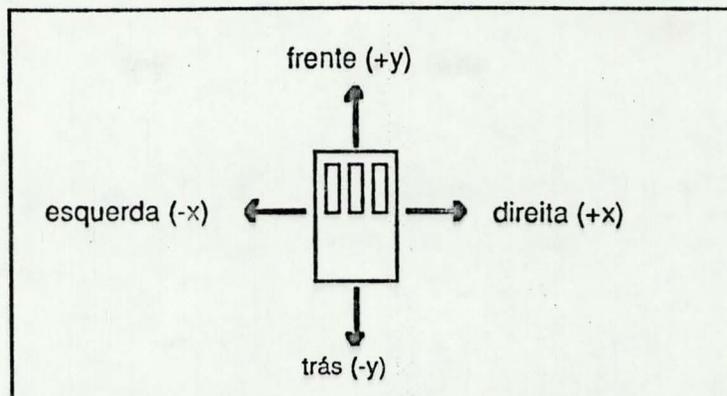


Figura 2 - Movimentação do rato para variação em dois eixos

b) a obtenção de um único valor a partir de determinado movimento do rato é feita com base em duas posições. A distância entre elas é medida ( em coordenadas de ecrã) e um factor aplicado para adaptação a valores aceitáveis para o parâmetro a ser alterado. O valor resultante é negativo quando a coordenada y da posição final é menor do que a inicial (ver figura 3).

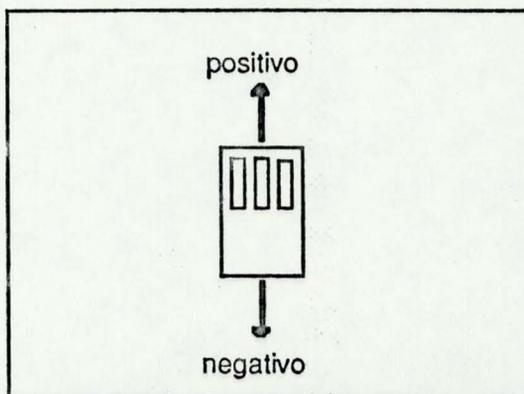


Figura 3 - Movimentação do rato para variação de um valor

c) a implementação de múltiplas vistas requer procedimentos especiais para ~~activação da~~ janela que está sob o cursor em dado momento. É necessário que o programa determine a janela onde está o cursor e depois active-a, utilizando as funções "winat" e "winattach". Além disto, o uso de "doublebuffer" com múltiplas janelas exige cuidados especiais na programação (ver anexo 1). **IMPORTANTE:** Sabe-se que o gestor de janelas da IRIS 4D trata estes pormenores de maneira diferente da IRIS 3000!

## 5. Estágio actual da implementação do MODELADOR

Apenas alguns comandos foram implementados até o presente momento:

- a) leitura e escrita de malhas em disco;
- b) visualização de malhas em "wireframe", com ou sem depth-cueing;

c) criação de malhas de polígonos baseadas em super-elipsóides. O utilizador pode informar os seguintes parâmetros:

- nome do objecto (o sufixo ".mod" é anexado automaticamente);
- dois valores para controle da forma (plano xy ou yz);
- escala nos eixos x, y e z;
- número de polígonos no plano xy (determina a variação angular usada para geração dos vértices da malha);

d) modificação da cor de toda a malha de polígonos;

e) criação de múltiplas janelas com controle individual da câmara, porém apenas com projecção perspectiva;

f) edição de geometria (em implementação).

### 5.1 Organização Interna

O MODELADOR está organizado nos seguintes ficheiros da directoria ~silvia/mod:

- a) mod.h: contém definições de tipos e constantes usadas por todos os módulos;
- b) modglobals.h: contém declarações das variáveis globais a todos os módulos;
- c) modtexto2D.h: contém definições necessárias para utilização das rotinas do módulo modtexto2D.c;
- d) modmouse3D.h: contém definições necessárias para utilização das rotinas do módulo modmouse3D.c;
- e) modgraf.h: contém macros e constantes para binding de acesso às rotinas gráficas da IRIS. Obs: as rotinas de gestão de janelas não estão incluídas neste binding;
- f) modprinc.c: contém procedimentos para menus e gestão das estruturas de dados das malhas de polígonos;
- g) modtexto2D.c: contém rotinas utilitárias para interacção textual com utilizador em 2D (mensagens, leitura de strings, etc.);
- h) modmouse3D.c: contém rotinas para acesso ao rato em 2D (posição e botões) e transformações para 3D;
- i) modgraf.c: contém binding para acesso às funções gráficas da workstation;
- j) modcamera.c: procedimentos para manipulação das vistas;
- k) moddraw.c: procedimentos para desenho da malha de polígonos em wireframe (com ou sem depth-cueing).
- l) modselips.c e modselips.h: procedimentos e constantes para comando de criação de malhas a partir das equações das superfícies super-elipsóides.

### Referências bibliográficas

[Olabarriaga 89] Olabarriaga S. Proposta para descrição de malhas de polígonos (MESH). INESC, Janeiro 1989.

## ANEXO 1

## Software gráfico da IRIS (Cyber 910)

Sílvia Delgado Olabariaga

INESC - Projecto CAD/CAM  
Av. Alves Redol, 9 - 2<sup>o</sup>  
1000 - Lisboa

Dezembro 1988

Tópicos da apresentação sobre o uso do software gráfico da workstation Cyber 910 (IRIS) feita ao Projecto CAD/CAM em 14 de dezembro de 1988.

- include files e compilação:

```
#include <device.h>
#include <gl.h>
```

usar "-Zg" na ligação

- rotinas de saída:

- pontos, linhas, círculos, áreas, texto em diversos estilos
- o nome indica o tipo dos parâmetros:
  - x,y,z -> float (nome puro), int (sufixo "i"), short (sufixo "s")
  - x,y (z=0) -> float (sufixo "2"), int (sufixo "2i"), short (sufixo "2s")
- sempre passam pelo pipeline de visualização 3D

- saída 3D:

- pilha de matrizes (apenas a do topo é usada na transformação)
- etapas para especificação da visualização:
  - perspective*, *ortho* ou *ortho2* (inicializa a matriz)
  - lookat* ou *polarview* (ponto de vista)
  - rotate*, *translate* e *scale* (transformações sobre o objecto)
- é possível especificar a *viewport*

- cores:

- 12 bitplanes (as cores entre 0 e 7 são reservadas)
- 8 bits para cada componente r, g, b
- *color*
- para fazer sobreposição de imagens sem destruí-las, usar *writemask*
- modos:
  - singlebuffer*
  - doublebuffer*, *swapbuffers*
- é preciso executar *gconfig* quando se troca de modo
- *greset* inicializa a pallet.

- texto
  - rotina *cmov* para determinação da posição
  - *charstr*: escreve um string sem ajuste ao tamanho da viewport
  - o texto é sobreposto ao fundo, sem apagar os pontos que não pertencem ao corpo da letra
  - cuidado, pois a posição do texto também sofre transformação de visualização!
- entrada:
  - buttons (MIDDLE, LEFT, RIGHT, on/off)
  - valuator (MOUSEX, MOUSEY) -> em coordenadas de ecran
  - keyboard (KEYBD)
  - podem ser queued ou unqueued
  - pode-se controlar o formato do cursor
  - não há entrada de string (tem que ser feita byte a byte)
  - *blanktime (nframes)* : permite o controle do número de frames que o monitor irá mostrar antes de apagar-se por não ter recebido entrada alguma. O ecran sempre é ligado quando esta função é executada.
- visualização em 3D:
  - backface
  - *depthcue, shaderange e setdepth* : para ter bom resultado, os planos de clipping (near e far da função *perspective*) devem estar justos ao objecto (ver exemplos do manual para valores que devem ser usados).
  - shading de Gouraud: especifica-se a cor dos vértices de um polígono e a interpolação é feita pela própria função de preenchimento de área. Obs: as cores são especificadas em termos de índices na pallet.
  - Z-buffer: não funciona na nossa IRIS, pois precisa de mais bitplanes.
- objectos:
  - *makeobj, closeobj, callobj*
  - fica muito mais rápido, mas não é necessário usá-los.
- mapeamento 2D->3D:
  - faz a transformação inversa à de visualização
  - *mapw, mapw2*
  - tem que haver um "object" de referência
- "pick":
  - permite identificar elementos (linhas, pontos) com um nome e determinar se o cursor está perto deles em dado instante;
  - uso:
    - desenhar objecto segundo procedimento normal
    - entrar no modo "pick" (*pick*)
    - inicializar a matriz com transformação de visualização
    - redesenhar o objecto, dando nomes às partes que podem ser apontadas  
(*loadname, pushname, popname*)
    - sair do modo "pick" (*endpick*)-> a função da IRIS retorna num buffer o identificador dos elementos seleccionados
  - clear não deve ser feito em modo "pick", senão todo o ecran é seleccionado;
  - não é necessário usar os mesmos procedimentos para seleccionar e para desenhar o objecto (pode-se desenhar as arestas e seleccionar apenas os vértices, por exemplo);
  - a área de influência dos elementos para verificação da proximidade é determinada pelo tamanho de uma matriz usada como máscara de comparação (*picksiz*)
  - *select* é semelhante ao *pick*, porém permite a identificação de uma região do espaço.

## - mex

- gestão de janelas e menus
- janelas:
  - winopen, winclose*
  - winattach, winpop, winpush, winset, winat*
  - reshapeviewport, getsize, getorigin*
  - cuidados:
    - a) máximo 10 janelas simultâneas
    - b) *winclose* de todas as janelas pendura a IRIS
    - c) REDRAW da janela deve ser feito pela aplicação
    - d) gestão de múltiplas janelas com doublebuffer é chato, pois o ecran todo é trocado quando se faz *swapbuffers* (deve-se duplicar a imagem das janelas que permanecem estáticas nos dois buffers)
- menus:
  - botão direito acciona automaticamente o menu do mex (*attach, reshape, etc.*)
  - > para evitar isto, usar este botão no modo *queued*
  - menus definidos pelo programa: *defpup, dopup*
  - cuidado com os valores retornados por *dopup* quando há submenus ou funções chamadas directamente pelo menu -> pode haver conflito, pois as opções sempre são numeradas a partir de 1, não interessando o nível do menu na hierarquia.
  - reserva 2 bitplanes (em *singlebuffer*) ou 4 (em *doublebuffer*) para armazenamento das janelas, o que reduz significativamente o número de cores simultâneas disponíveis para a aplicação (máximo 16 cores com 2 buffers).
  - uso dos bitplanes reservados ao mex:
    - pupmode, pupcolor, endpupmode*: habilitam o uso dos bitplanes mais significativos para sobreposição de imagens sem destruição do fundo.
    - atenção: "mex -d" inicializa o mex para execução em *doublebuffer*. Se a aplicação usar *doublebuffer* sem ter invocado o mex desta maneira, ficam guardadas janelas de execuções antigas que reaparecem "magicamente" em momentos indesejáveis. No manual diz que se deve usar <Ctrl-C> para apagá-las, mas nem sempre isto funciona ...

## ANEXO 2

## Comentários sobre algumas publicações a respeito de interacção para modelação de objectos tri-dimensionais

Sílvia Delgado Olabariaga

INESC - Projecto CAD/CAM  
Av. Alves Redol, 9 - 2º  
1000 - Lisboa

Novembro 1988

**Badler N. Manoochehri K. & Baraff D. *Multi-dimensional Input Techniques and Articulated Figure Positioning by Multiple Constraints. Workshop on Interactive 3D Graphics, ACM-SIGGRAPH, 1986.***

Comentam experiências feitas usando dispositivos de entrada 3D, dizendo que não são eficazes, pois o ecran continua a ser 2D: isto exige que o utilizador faça mentalmente projecções para acertar exactamente o ponto desejado, pois não há um objecto real manipular --> falta feedback espacial! Outra observação interessante diz respeito à dificuldade em controlar muitos graus de liberdade simultaneamente, o que os levou a desabilitar alguns deles temporariamente (permitem movimentação em no máximo dois eixos ao mesmo tempo).

**Baecker R. *Towards an effective characterization of graphical interaction. In: Readings in Human-Computer Interaction - A multidisciplinary approach. Morgan Kaufmann, 1987.***

Defende a posição de que independência de dispositivo de entrada e interfaces com bom desempenho são incompatíveis pelos seguintes motivos:

- a separação dos procedimentos de entrada e de saída existente nos pacotes gráficos com dispositivos virtuais dificulta a programação de interfaces de alto grau de interactividade;

- dispositivos virtuais restringem a interface a um conjunto mínimo de operações que a tornam muito limitada. Além disto, afirma que a interface com o utilizador deve ser refeita sempre que o dispositivo de interacção for alterado, a fim de aproveitar ao máximo as capacidades do mesmo.

Salienta características importantes de uma interface homem-máquina:

- feedback imediato
- flexibilidade
- generalidade
- potência
- interpretação adequada de dados incorretos ou incompletos consoante o contexto

Diz ainda que os jogos devem ser usados como base para discussões, já que normalmente apresentam interfaces homem-máquina bastante eficazes. Os jogos devem ser de fácil operação e suficientemente desafiadores para manter o utilizador interessado... Comenta também as características de interfaces de acordo com o hardware disponível para entrada e saída gráfica, potência computacional e tempo de resposta exigido.

**Bier E. *Skitters and Jacks: Interactive 3D positioning tools.* Workshop on Interactive 3D Graphics, ACM-SIGGRAPH, 1986.**

Propõe um método para interacção com utilizador para composição de cenas (escolha da posição e orientação dos objectos no mundo real).

Apresenta inicialmente alguns critérios para medida de eficácia de interfaces com utilizador, classificando os tipos actualmente existentes:

- interactividade (manipulação directa, uso de rato, etc.)
- facilidade e liberdade para especificação de âncoras para a transformação a ser aplicada ao objecto (eixo para rotação, direcção de translação, etc.). É importante haver mecanismos que permitam uso do sistema de coordenadas do mundo real, do objecto, arestas, facetas, etc. como referenciais;
- liberdade e facilidade para especificação de parâmetros para a transformação (ângulo de rotação, unidades de deslocamento, etc.) É importante possibilitar a especificação de valores absolutos, relativos ao estado actual ou ainda valores relativos a outros objectos;
- suavidade com que as transformações são mostradas ao utilizador à medida que são aplicadas ao objecto.

Depois comenta que há basicamente duas abordagens para desenvolvimento de interfaces com bom desempenho :

- menus hierárquicos com apontamento simples para seleção dos parâmetros e âncoras. Sobre esta abordagem, apresenta os seguintes comentários :
  - é preciso seleccionar a âncora e os parâmetros pelo menu (lento)
  - gasta-se muito tempo percorrendo a árvore de comandos
  - os menus apresentam muitas opções (-> muitos comandos)
  - os parâmetros não podem ser reusados em vários comandos
- "jacks" (sistema de coordenadas definidos pelo utilizador para orientação em um ou mais comandos). Podem ser usados como parâmetro ou âncora, tornando a interface homogênea. São definidos por meio de um rato 3D ("skitter"), que pode ser deslocado sobre o objecto (facetas, vértices, arestas) ou livremente no mundo real. Há um comando específico para criar "jacks" num "skitter" ("MakeJackAtSkitter"), podendo estes receberem identificadores e serem atrelados a um objecto da cena.

Interessante em tudo isto :

- movimentação do rato 3D: pode ser feita sobre a superfície de um objecto, sobre uma aresta, sobre as facetas posteriores, etc. Usam o conceito de "gravidade" para possibilitar o movimento consoante diferentes tipos de elementos simultaneamente. Por exemplo, andar sobre as facetas anteriores de um objecto, porém com forte gravidade nos vértices, de forma que o cursor é puxado para o vértice quando dele se aproxima.

- diz nas conclusões que "simplicidade" e "generalidade" são objectivos incompatíveis numa única interface.

**Bruderlin B. *Constructing Three-dimensional Geometric Objects Defined by Constraints.* Workshop on Interactive 3D Graphics, ACM-SIGGRAPH, 1986.**

Entre outros assuntos, apresenta uma calculadora de sólidos, para CSG. Os comandos são escolhidos em um menu (move, scale, rotate, etc.) e os parâmetros indicados via movimentação de um "slider" com o rato. Para compor o objecto, aparece uma calculadora com as operações de união (+), subtração (-), etc.

**Nielson G. & Olsen D. *Direct Manipulation Techniques for 3D Objects using 2D locator devices.* Workshop on Interactive 3D Graphics, ACM-SIGGRAPH, 1986.**

Comenta que técnicas existentes para interação 3D a partir de dispositivos 2D apresentam deficiências:

- são indirectas por natureza: o utilizador manipula um objecto 2D que por sua vez é mapeado para o objecto 3D real (ex: deslocamento horizontal do rato para indicar rotação);
- feedback é bi-dimensional (vistas do objecto 3D).

Afirma que o ideal é manipular directamente a projecção perspectiva do objecto real, nos três eixos!

Descreve, então o "triad mouse" (rato com posição 3D) e aponta que há 2 problemas básicos a resolver na sua implementação:

- feedback da posição 3D: deve-se ter um referencial espacial (no caso, um cubo correspondente ao mundo real) e ter um símbolo 3D para o cursor (no caso, 3 eixos ou um cubo pequeno).
- mapeamento do movimento bi-dimensional para 3D: utiliza a projecção dos eixos do rato (3D) e o vector de deslocamento do rato (2D). Verifica o eixo cuja projecção faz o menor ângulo com o vector de deslocamento do rato e aplica a transformação neste eixo. As rotações, translações e escalas são especificadas através de 3 ou 2 pontos, que podem ser posicionados segundo uma aresta, faceta ou outro referencial qualquer do sistema.

Observações:

- reconhecem que esta técnica está intimamente ligada às características do objecto, porém afirmam ser uma boa abordagem inicial para o problema;
- dizem que o desempenho é razoável num IBM-PC com co-processador aritmético.

**Parke F. *Interactive tools to Support Animation Tasks.* Workshop on Interactive 3D Graphics, ACM-SIGGRAPH, 1986.**

Comenta interacção em sistemas de animação 2D e 3D, enfatizando o BBOP (NYIT, 1983):

- "joyswitch": usado para escolha da junta a animar;
- "joystick": com movimento em 3 eixos, é usado em translações e rotações;
- mesa digitalizadora é usada para entrada de outros comandos simples.

Afirma que mesmo as workstations mais potentes têm pouco suporte para modelação e animação de objectos com superfícies deformáveis.

**Phillips C. & Badler N. *Jack: A toolkit for manipulating articulated figures.* Symposium on User Interface Software, ACM-SIGGRAPH, 1988.**

Diz que as técnicas para posicionamento e manipulação de figuras tri-dimensionais articuladas são difíceis de usar e propõe um pacote consistente e flexível para programação de interfaces com o utilizador (basicamente manipulação interactiva de transformações especificadas via rato).

Características desejáveis de uma interface:

- alteração do ponto de vista deve ser simples e fácil, assim como pegar no objecto e virá-lo para observá-lo a partir de outros ângulos;
- feedback em tempo-real das transformações, para agilizar o processo de tentativa-e-erro;

- deve ter comportamento facilmente previsível consoante a entrada fornecida pelo utilizador (concordância de sistemas de coordenadas, por exemplo);
- movimento globais (operações muito frequentes) devem estar "à mão". Como exemplo cita o caso de translações e rotações sem ter que especificar sempre o eixo através de menus.

#### Entrada de dados 3D:

- a partir de dispositivos 3D: não há boas referências (desconforto para o utilizador e dificuldade de controlar simultaneamente muitos graus de liberdade);
- a partir de dispositivos 2D: menciona "triad mouse" e "skitters and jacks".

#### Passa, então, a descrever o pacote "Jack" :

- permite a visualização e manipulação de objectos tri-dimensionais articulados em animação e simulação
- implementado sobre o window manager da IRIS (mex)
- estrutura dos objectos é chamada de "peabody": segmentos rígidos conectados por "juntas", cada uma com seu próprio sistema de coordenadas. Estes segmentos armazenam a massa, momento de inércia e a superfície envolvente de uma parte do objecto.

#### A interface com o utilizador tem as seguintes características interessantes:

- desenha uma grelha no chão, assim como as projecções ortogonais do objecto nas paredes do mundo real, em cor mais escura;
- os elementos desenhados são seleccionáveis pelo utilizador (apenas vértices, arestas, juntas, grelha, etc);
- à base de menus, porém a maioria dos comandos podem ser seleccionados por uma tecla;
- quase todas as operações podem ser feitas com o rato;
- selecção de comandos é feita inicialmente no nível mais alto (rotação), depois os operandos (objecto) e por último os parâmetros (ângulo);
- comandos para controle do ponto de vista, accionados via botões do rato:
  - sweep (movimenta o observador)
  - pan (movimenta o ponto de interesse)
  - zoom (movimenta o observador em direcção ao ponto de interesse)
- translação e rotação são definidas em função da "mouse line", determinada pela posição do rato e pela transformação inversa de visualização. Cada botão habilita o movimento num eixo (no máximo dois simultâneos), sendo o formato do cursor alterado para lembrar o tipo de movimento seleccionado.

#### **Forrest A.R. *User Interfaces for Three-dimensional Geometric Modelling.* Workshop on Interactive 3D Graphics, ACM-SIGGRAPH, 1986.**

Comenta opções e escolhas razoáveis para interfaces com o utilizador para modelação de sólidos em função do equipamento disponível e tipo de aplicação. Algumas opiniões interessantes:

- o melhor seria ter saída 3D, porém os dispositivos ainda são muito desconfortáveis;
- trabalhar com várias vistas ortogonais é mais preciso do que com perspectiva, que só deve ser usada quando os objectos forem familiares;
- wireframe é rápido, simples e contém toda a informação necessária, mas pode ser ambíguo;
- sombreamento realista nem sempre é necessário, pois a informação pode eventualmente ser melhor transmitida através de textura;
- diz que é difícil um ser humano trabalhar sem restrições de ferramental, e portanto pode-se pensar em utilizar artifícios para melhorar a interface de uma maneira geral, mesmo que possam vir a ser limitantes;

- o utilizador pode cooperar com o sistema, em vez de ter tudo automático, pois assim gera-se modelos correctos mais rapidamente;

- diz que há pouco esforço para melhorar a visualização em wireframe de sólidos descritos por CSG (soluções em hardware).

Por fim, descreve o sistema "CUBELAND", que permite modelação de objectos 3D através de paralelepípedos posicionados e escalados através de uma linguagem textual.

## ANEXO 3

## Comentários sobre as interfaces com utilizador dos programas de demonstração da IRIS

Sílvia Delgado Olabbarriaga

INESC - Projecto CAD/CAM  
Av. Alves Redol, 9 - 2ª  
1000 - Lisboa

Novembro 1988

**arch**

Permite movimentação do observador por uma cidade através do rato:

- botão direito = deslocamento à direita;
- botão esquerdo = deslocamento à esquerda;
- botões direito ou esquerdo + botão do meio = para cima e para baixo;
- botões direito e esquerdo simultaneamente = para trás.

**curve**

Permite a alteração dos pontos de controle de uma curva. Salienta a área de influência do ponto seleccionado em determinado instante pelo rato (desenha em outra cor as linhas que seriam alteradas em caso de reposicionamento do ponto).

**jet**

Permite observação de um avião a partir de vários ângulos:

- mostra os eixos do mundo real no centro do objecto;
- botão direito = rotação em z;
- botão esquerdo = rotação em x;
- botão do meio = rotação em y;
- botões direito e do meio simultaneamente = "zoom-in";
- botões direito e esquerdo simultaneamente = "zoom-out";
- botões esquerdo e do meio simultaneamente = translação no plano xy;
- três botões simultaneamente = fim

**robot**

Permite a movimentação de partes do braço de um robô por movimentos do rato combinados aos botões. Mostra na linha inferior do ecrã uma mensagem indicando a parte do robô que será movimentada de acordo com o botão que for pressionado.

**colored**

Seleccção interactiva de cor pelo sistema RGB, através de técnica semelhante à utilizada no MODELADOR (já havia sido definida e implementada à altura deste estudo), porém com uma característica interessante: a cor interna das três barras é modificada para indicar ao utilizador exactamente a cor resultante se o valor correspondente àquela posição for escolhido.

**poputils**

Permite aplicação de transformações a um cubo. Mostra os eixos de referência para as transformações e um "slider" na parte inferior do ecrã. A transformação é seleccionada via menu, e o respectivo parâmetro indicado pela posição do rato sobre uma barra com valores desenhada na parte inferior do ecrã. A transformação só ocorre se o rato estiver sobre esta barra. Nas facetas do cubo é desenhado o nome do eixo correspondente à rotação que fará modificar sua orientação.

**sdemo**

Permite a alteração dos pontos de controle de uma superfície que descreve um objecto semelhante a um carro:

- há uma grelha de referência nas paredes do mundo real;
- os pontos são projectados nas paredes em cor diferente;
- o formato do cursor varia conforme a função seleccionada em dado instante;
- usa o botão do rato no modo "queued", o que traz transtornos quando a operação a ser realizada é lenta, pois permanecem na fila eventos que só são processados quando já estão obsoletos;
- permite a utilização de planos de corte para eliminar partes do objecto e assim reduzir a complexidade do desenho;
- o eixo de rotação é escolhido através de um icon contendo os três eixos e setas circulares indicando o sentido de rotação;
- o vértice apontado pelo rato em dado instante é salientado tanto no próprio objecto (perspectiva) como nas projecções nas paredes.

**ANEXO 4****SELIPS**

Sílvia Delgado Olabariaga

INESC - Projecto CAD/CAM  
Av. Alves Redol, 9 - 2º  
1000 - Lisboa

Novembro 1988

Selips é um programa que gera malhas de polígonos no formato do "mesh" que aproximam superfícies super-elipsóides. Os parâmetros são informados pelo utilizador via interface textual:

- expoente para determinar a forma do sólido no plano yz;
- expoente para determinar a forma no plano xy;
- escala diferenciada nos três eixos;
- número de polígonos (n) que a malha deve ter nos cortes paralelos ao plano xy (no plano zy serão gerados n/2 polígonos);
- factor de escala a ser aplicado a todos os vértices;
- posição do objecto no mundo real;
- ângulo de rotação nos três eixos.

A malha é gerada segundo algoritmo descrito em "SELIPS - Um programa para auxiliar o ensino de super-elipsóides", de Sílvia Delgado Olabariaga, CPGCC-UFRGS, 1986.

# Adendo ao relatório sobre o MODELADOR Situação actual

Sílvia Delgado Olabariaga

INESC - Projecto CAD/CAM  
Av. Alves Redol, 9 - 2º  
1000 - Lisboa

Novembro 1989

## 1. Comandos Adicionais

O comando introduzido no modelador permite a manipulação directa das coordenadas dos vértices, através do mecanismo descrito no relatório anterior.

## 2. Novos Módulos

Os seguintes módulos foram introduzidos no MODELADOR:

- **modshape.c:** contém procedimentos para interacção com o utilizador para manipulação directa dos vértices de uma malha de polígonos. Inclui funções para eco da posição do rato sobre a malha e modificação das coordenadas dos vértices segundo os movimentos do mesmo.
- **modvistas.c:** contém procedimentos para inicialização e modificação de uma vista, utilizando funções genéricas para acesso a janelas. Chama-se de "vista" ao conjunto de uma janela sobre o ecrã com determinados parâmetros para visualização (câmara e tipo de projecção)
- **modvistas.h:** define as funções e estruturas de dados para acesso a uma vista.
- **modwindows.c:** contém rotinas genéricas para manipulação de janelas (criação, activação, etc.). Faz a interface com o gestor de janelas da Iris.
- **modwindows.h:** contém definição de estruturas de dados e constantes para acesso a janelas do módulo modwindows.c.

### **3. Avaliação**

Há pelo menos dois comentários a registrar:

a) a manipulação interactiva da malha oferece pouca possibilidade de controle sobre a modificação dos valores das coordenadas, pois o rato é deslocado sobre um plano em projecção perspectiva. É indispensável que, nestes casos, a vista seja feita em projecção paralela, a exemplo do que acontece nos modeladores do sistema de animação "explore", da TDI.

b) a estrutura interna do modelador não permite que seu transporte para outros equipamentos gráficos seja feito com facilidade. É necessário isolar os procedimentos para interacção com o utilizador (rato, janelas, menus, etc.), tornando o programa menos dependente da biblioteca de funções gráficas da iris.