

Software Gráfico Orientado por Objectos para Aplicações Multimédia

Silvério Neves, Luís Almeida

Resumo- Neste artigo descrevemos o desenvolvimento de um ambiente integrado de alto nível para PCs com suporte de programação gráfica.

O desenvolvimento foi efectuado em ambiente Windows, em linguagem C++ e com ferramentas multimédia. O resultado deste desenvolvimento foi a criação de uma classe de objectos gráficos com características dinâmicas e estáticas e um editor gráfico com suporte de animação gráfica ("MoveIt v1.0").

Abstract- This paper describes the development of a high-level integrated environment for PCs with graphics programming support.

The development was carried out on the WindowsTM environment, with C++ language and multimedia tools. The result of this development was the creation of a graphic objects class with static and dynamic features and a graphic editor with graphic animation support ("MoveIt v1.0").

I. INTRODUÇÃO*

Nos dias de hoje, falarmos de computadores, telecomunicações ou formação faz surgir uma palavra mágica: *multimédia*. O seu significado, sendo vasto, insere-se essencialmente na conjugação de texto, som e imagem num mesmo meio de comunicação. O crescimento exponencial do uso da multimédia no nosso dia-a-dia deve-se ao aparecimento de ferramentas (melhores processadores, placas de som, placas de vídeo, redes de comunicações de alta capacidade,...) com um progressivo baixo custo para o cidadão comum.

Inserido nesta filosofia, o projecto que desenvolvemos tinha como finalidade a criação de um ambiente integrado de alto nível para PCs com suporte de programação gráfica. O seu desenvolvimento dividiu-se essencialmente em três partes :

A primeira centrava-se na elaboração de um classe de objectos gráficos com características estáticas (posição, tamanho, cor, plano,...) e dinâmicas (direcção de movimento, distância de movimento, tipo de movimento, velocidade de movimento,...).

Na segunda parte criou-se um editor gráfico com suporte de animação gráfica para a referida classe de objectos gráficos.

Para a terceira parte reservou-se a inserção de características multimédia (som, vídeo,...).

II. FERRAMENTAS UTILIZADAS

Ao nível do sistema operativo optamos pelo ambiente Windows por ser o mais utilizado nos PCs e com mais probabilidades de o continuar a ser no futuro.

Software utilizado :

Win32s - a arquitectura de 32 bits tem muitas vantagens em relação à de 16 bits: é usada nos novos processadores, os novos sistemas operativos são de raiz de 32 bits, a gestão de memória torna-se muito mais flexível, etc.

Linguagem C++ - uma linguagem extremamente flexível e apropriada para o ambiente Windows cujo modelo se baseia numa estrutura orientada por objectos [1] [2].

Borland C++ - um compilador que possuiu uma vasta livreria para o ambiente Windows (Object Windows Libray 2.0) [3] permitindo ainda compilar para Win32s.

Video for Windows Development Kit 1.1 - fornece as ferramentas necessárias para manipular ficheiros vídeo (*.AVI - Audio Video Inteleaved) [8].

Para fazer uso deste software trabalhamos num PC com processador 486DX4, 8Mbytes de RAM e placa de som Sound Blaster 16.

III.IMPLEMENTAÇÃO

A. Classe de Objectos gráficos

A classe de objectos gráficos por nós implementada assenta numa classe principal Tobject. Todos os objectos particulares são derivados desta classe como mostra a figura 1.

A class TObject possuiu as linhas de orientação para as

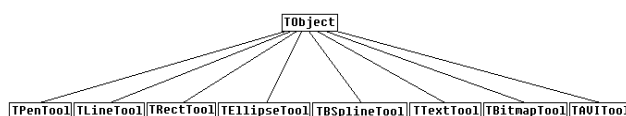


Fig. 1 - Classe de Objectos gráficos

* Trabalho realizado no âmbito da disciplina de Projecto.

classes derivadas. As suas funções membro controlam as operações do rato, o desenho do objecto, a leitura e escrita num ficheiro e a manutenção de uma tabela de movimentos e sons. Contendo informação importante relativa a cada objecto como :

- tipo do objecto (elipse, rectangulo, etc...),
- cor,
- espessura de linha,
- tamanho do objecto,
- plano,
- tabela de movimento (MotionTable array),
- etc.

Em cada classe derivada definem-se propriedades próprias dos diferentes objectos gráficos, reescrevendo se necessário funções da classe Tobject.

A tabela de movimento é um array de classes ObjMotion ordenado pela varivel de tempo de cada elemento. A classe ObjMotion possuiu as seguintes variáveis : *Time*, *Position*, *Motion*, *RefPoint* e *Wave*.

Definindo a que tempo *Time* o objecto se situa na posição *Position*, o tipo de movimento *Motion* que efectua para chegar à posição *Position* e o som que deve ser tocado no tempo *Time*.

B. Estructura de ligação Tstate

Para fornecer uma eficaz ligação entre a nossa aplicação e a classe de Objectos gráficos implementou-se a estrutura Tstate. Esta estrutura possuiu as seguintes variáveis :

- tipo de objecto seleccionado,
- variável de cada tipo de Objecto gráfico,
- plano disponível para desenhar,
- espessura da caneta seleccionada,
- cor seleccionada,
- tipo de letra seleccionada,
- posição do rato,
- tempo actual da cena,
- se algum objecto está seleccionado,
- tipo de movimento seleccionado,
- palette da aplicação,

C. Duplo Buffer

Essencialmente há três técnicas de animação, a aplicação lógica XOR, a troca de página video e o duplo buffer [9].

A primeira consite na escrita directa de um sprite (conjunto de bits que descrevem um objecto) na memória video através da operação lógica XOR. Esta técnica tem um grande inconveniente, só funciona com fundos sólidos.

A troca de página video consiste em dividir a memória video em diferentes páginas (caso o modo gráfico o permita) e activar só uma delas. Numa página não activa escreve-se a cena para o instante de tempo que se segue,

actualizando-se o ecrã pela activação dessa mesma página.

O buffer duplo é conceptualmente similar à anterior, sómente que se usa um buffer em vez de uma página video. Esta técnica tem um senão, é a cópia da memória em massa para o ecrã.

A nossa escolha recaiu no buffer duplo porque as duas primeiras técnicas lidam directamente com o hardware do adaptador de video. O ambiente Windows é independente do hardware (uso de drivers), tornando-se inconcebível a aplicação lógica XOR ou a troca de página video como técnica de animação.

D. Palette

A gestão da palette no ambiente Windows é efectuada através de uma lookup table indexada [9]. Isto é, temos uma palette de hardware e uma palette lógica. A palette de hardware possuiu o numero de cores que a memória do adaptador video permitir, mas a palette lógica possuiu um conjunto de cores indexadas com numero superior ou igual à palette de hardware. As cores disponiveis para o ecrã são as presentes na palette lógica. Como ilustra a figura 2, esta gestão de cores pode tornar-se morosa.

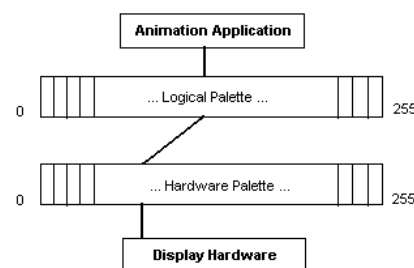


Fig. 2 - Gestão da palette

Para uma melhor performance da animação e uma uniformização de cores durante a mesma é conveniente criar uma palette identidade. Este procedimento em termos práticos “cola” a palette lógica à palette de hardware.

E. Timer

No coração do velho IBM PC, e ainda na maior parte dos PCs de hoje, há um timer de hardware que origina uma interrupção em cada 55 milissegundos. Significando pulsos de relógio 18,21 vezes por segundo [10].

Para o ambiente Windows, este timer é uma limitação para aplicações multimédia, visto que, as mensagens do Windows não são atendidas da fila de espera mais rápido do que 18,21 vezes por segundo.

Na nossa aplicação usamos um timer de 10 vezes por segundo. Esta opção aliada a uma boa estrutura da nossa aplicação mostrou-se boa, contando que durante a animação praticamente a única mensagem necessária é a do timer.

IV. MOVEIt v1.0

Como resultado do nosso desenvolvimento surgiu a aplicação multimédia “MoveIt v1.0”. É uma aplicação MDI (Multiple Document Interface), em que cada janela MDI Child representa uma cena de animação. A janela principal possuiu vários elementos de interacção com o utilizador: menu, control bar, status bar, tools windows, animate window e time bar como poderemos observar na figura 3.

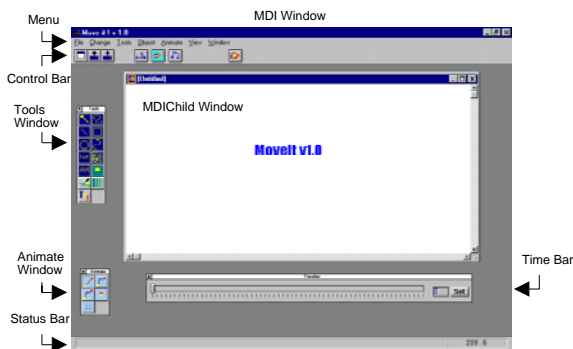


Fig. 3 - Janela principal da aplicação MoveIt v1.0

A. Tools Window

A Tools Window permite-nos lidar com os objectos gráficos por nós criados. A figura 4 mostra-nos esta janela.

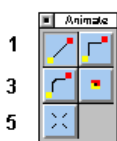
- 1 - Modo de selecção.
- 2 - Desenho livre.
- 3 - Linha.
- 4 - Rectângulo.
- 5 - Elipse.
- 6 - Curva B-Spline.
- 7 - Texto.
- 8 - Bitmap.
- 9 - AVI.
- 10 - Côr de fundo.
- 11 - Côr da linha.
- 12 - Espessura da linha.
- 13 - Tipo de letra.



Fig. 4 - Tools Window

B. Animate Window

Para atribuição do tipo de movimento ao objecto gráfico usamos a Animate Window como mostra a figura 5.



- 1 - Movimento directo.
- 2 - Movimento com quebra.
- 3 - Movimento curvo.
- 4 - Parado.
- 5 - Desaparecer.

Fig. 5 - Animate Window

C. Control Bar

O Control Bar como mostra a figura 6 possuiu 7 botões com as seguintes funções :

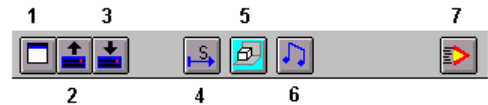


Fig. 6 - Control Bar

- 1 - Criar um novo ficheiro MoveIt.
- 2 - Abrir um ficheiro MoveIt.
- 3 - Gravar a cena corrente.
- 4 - Define-se o tempo da cena.
- 5 - Permite-nos escolher o fundo da cena. Pode ser uma côr ou um bitmap.
- 6 - Selecciona um ficheiro de som Wave para tocar em associação com o objecto seleccionado.
- 7 - Este importante comando executa a animação que está criada na cena corrente.

D. Menu

Todas os comandos do menu existem em ícones na Tools Window, Animate Window e Control Bar menos os do grupo Object (figura 7).

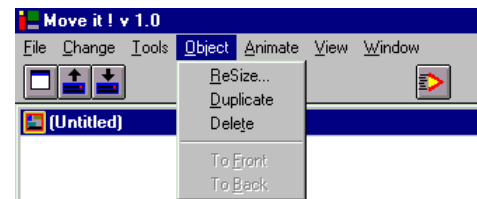


Fig. 7 - Object pull-down menu do menu principal

Este grupo do menu possibilita-nos redimensionar o tamanho do objecto, duplicá-lo ou apagá-lo. Estas acções sobre os objectos só têm efeito quando seleccionados.

E. Time Bar

A figura 8 mostra-nos o Time Bar.



Fig. 8 - Time Bar

O Time Bar é um elemento fundamental de interacção para a concepção da cena de animação. Ao deslocarmos o cursor ao longo do Time Bar estamos a situar a cena temporalmente, permitindo inserir objectos gráficos no tempo desejado como visualizar a posição dos objectos ao longo do tempo. Esta funcionalidade do Time Bar dá-nos a possibilidades de conceber uma animação com precisão em “Draw Time” de um segundo.

V. SHOWIt v1.0

Desenvolveu-se outra aplicação denominada ShowIt v1.0. Embora no MoveIt possamos correr a animação da cena da janela MDI Child activa, esta aplicação permite-nos executá-la separadamente.

A figura 9 mostra-nos a interface gráfica desta aplicação.

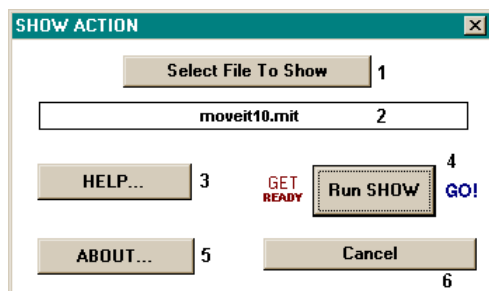


Fig. 9 - Diálogo inicial com o utilizador da aplicação ShowIt v1.0

- 1- Seleccionar a animação pretendida.
- 2- Mostra o ficheiro seleccionado.
- 3- Ajuda da aplicação.
- 4- Corre a animação.
- 5- Janela About.
- 6- Cancelar a selecção.

Durante a animação com a ajuda de algumas teclas podemos gerar algumas acções. A tabela 1 mostra as teclas especiais e as suas respectivas acções.

Tabela 1

Tecla	Acção
P	Pausa na animação
G	Eliminação da acção Pausa
B	Ir para o início da animação
E	Ir para o fim da animação
N	Seleccionar nova animação
Esc	Sair da aplicação

VI. FUTUROS MELHORAMENTOS

A aplicação MoveIt pode ser melhorada num futuro próximo em alguns aspectos :

- Implementação de botões interactivos.
- Elaboração de um ficheiro projecto, que possa englobar várias cenas de animação.
- Novos tipos de movimento.
- Sistema de "help on-line".

- Impressão.
- Introdução de objectos gráficos tri-dimensionais.

Numa perspectiva mais vasta, podemos inserir a classe de objectos gráficos numa Dynamic Link Library para uso público.

VII. CONCLUSÕES

No fim do desenvolvimento deste projecto ficamos ainda mais cientes das potencialidades que a multimédia nos oferece. Especialmente no ambiente Windows, poderemos afirmar que com uma linguagem tão versátil como o C++ e com as ferramentas de software e hardware que estão ao nosso dispor nos dias de hoje, podemos desenvolver um excelente ambiente multimédia sobre ele.

Em relação à aplicação desenvolvida, "MoveIt v1.0", sendo um bom meio de efectuar apresentações a sua área de aplicação torna-se bastante vasta. A implementação dos botões interactivos referidos nos futuros melhoramentos alargam ainda mais essa área permitindo, por exemplo, situar-se na área da educação servindo de editor de tutoriais e de enciclopédias interactivas.

VIII. AGRADECIMENTOS

Agradecemos o Prof. Dr. V. Sklyarov a orientação do projecto.

REFERÊNCIAS

- [1] Borland Internacional, "Borland C++ for Windows - Programmer's Guide", 1993
- [2] Borland Internacional, "Borland Object Windows for C++ - Programmer's Guide", 1993
- [3] Borland Internacional, "Borland Object Windows for C++ - Reference Guide", 1993
- [4] Anthony Porter, "C++ Programming for Windows", Osborne, McGraw-Hill, 1993
- [5] Namir Clement Shammis, "What every Borland C++ 4 programmer should know", Sams Publishing, 1994
- [6] Ori Gurawich and Nathan Gurewich, "Borland C++ Multimedia Programming", SYBEX, 1994
- [7] Michael J. Young, "Windows Animation Programming with C++", AP Professional, 1994
- [8] Paul DiLascia, "Exploit Video for Windows 1.1", Microsoft Systems Journal, Janeiro 1995
- [9] James Finnegan, "Implementing Games for Windows", Microsoft Systems Journal, Janeiro 1995
- [10] Rick Grehan, "The Software Stopwatch", BYTE - The Magazine of Technology Integration, Abril 1995