

A Computação Gráfica no perfil de ensino em Engenharia Informática

Manuel Próspero dos Santos

Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
P-2825-144 Monte de Caparica

ps@di.fct.unl.pt

1. SUMÁRIO

A formação em Engenharia Informática, na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, tem sido suportada por um curso de Licenciatura e um outro de Mestrado. O ensino na área da Computação Gráfica está presente em ambos os cursos, tendo-se tentado seleccionar, em cada caso, as matérias que melhor se enquadram com o conjunto das restantes disciplinas. Por outro lado, as ferramentas e métodos desejáveis para o ensino e aprendizagem também se tiveram que enquadrar no aproveitamento dos recursos comuns, essencialmente computacionais, que a instituição universitária tem disponibilidade para providenciar. Esta comunicação pretende apresentar a metodologia actualmente utilizada nas diversas disciplinas de Computação Gráfica para os referidos cursos da FCT-UNL, tentando que a experiência assim adquirida no ensino desta área científica possa vir a contribuir positivamente para uma discussão mais alargada de que resultem novas ideias, quer para aplicação própria, quer para outros casos semelhantes.

1.1 Palavras-chave

Computação Gráfica, Ensino, Licenciatura, Mestrado, Laboratório.

2. INTRODUÇÃO

Em 1991, teve o Grupo Português de Computação Gráfica (GPCG) a iniciativa de promover um Workshop sobre o ensino da Computação Gráfica em Portugal [8]. Passados sete anos sobre esta realização, uma área que tem assistido a alterações tão dramáticas, quer do ponto de vista do hardware como do software, não poderia permanecer indiferente no que diz respeito à discussão dos temas e métodos mais adequados ao correspondente ensino. Este sentimento esteve bem patente durante as apresentações orais de diversas comunicações no 8.º Encontro Português de Computação Gráfica [9], muito embora o tópico específico do ensino não fosse um dos objectivos dessa conferência, também promovida pelo GPCG.

Com a presente comunicação, quer o autor dar um pequeno contributo para uma tão ampla quanto possível troca de experiências, já há muito desejada pelos responsáveis da educação neste mesmo domínio, ao nível universitário português.

3. ENQUADRAMENTO

Dada a naturalidade duma representação gráfica e o modo imediato como é interpretada por um ser humano em comparação com uma notação empregando símbolos alfanuméricos, não é de admirar que a Computação Gráfica, hoje em dia, esteja presente como ferramenta de trabalho em praticamente todas as áreas em que o computador já provou ser útil. Nos últimos anos, a Computação Gráfica tem sido considerada um dos campos da computação com crescimento mais rápido e ao qual tem sido dedicada muita atenção. É, pois, natural que haja uma preocupação acrescida na escolha das matérias e material de apoio aos respectivos cursos.

O âmbito da Computação Gráfica, no entanto, é relativamente vasto. Tendo por definição de base a criação, armazenagem e manipulação de modelos e correspondentes imagens dos objectos, tais objectivos implicam, em geral, conhecimentos básicos multi-disciplinares [2]. No mínimo, e para um ramo de Engenharia Informática, é imprescindível uma boa preparação dos alunos em algoritmia, estruturas de dados, linguagens de programação e matemática, esta última em áreas como sejam a álgebra linear, a geometria analítica, a análise numérica e a geometria no espaço. Por outro lado, e por esta mesma razão, o ensino da Computação Gráfica ao nível de licenciatura e mestrado no referido ramo de engenharia é um óptimo ambiente para, colateralmente, solidificar a preparação anterior desses alunos naquela diversidade de domínios.

4. OS PROGRAMAS

Nos actuais currículos da FCT-UNL, a Computação Gráfica está explicitamente presente com duas disciplinas na Licenciatura em Engenharia Informática (LEI), e com uma outra no Mestrado em Engenharia Informática (MEI). Do ponto de vista de domínio científico, todas essas disciplinas de Computação Gráfica se englobam no grupo de Ciência e Tecnologia da Programação. Por tal motivo, os conteúdos programáticos, logo à partida, não se orientam para aspectos como, por exemplo, a concepção, descrição e construção dos equipamentos, mas sim para os de natureza mais algorítmica, com

técnicas e métodos para modelação, interacção e *rendering* [2]. Esta opção não tem vindo a ser alterada nos últimos anos, ainda que o software, muito frequentemente, tenha uma grande dependência do tipo e arquitectura dos dispositivos físicos a que se destina. Porém, e no presente contexto, as características relevantes destes últimos são apenas abordadas ao nível funcional.

4.1 Na Licenciatura

A existência de dois módulos distintos e a diferentes níveis tem-se mostrado de toda a conveniência para o ensino da Computação Gráfica numa licenciatura em Engenharia Informática.

Um primeiro módulo terá que ter sempre, como é óbvio, um programa de carácter mais introdutório. Na LEI, ele corresponde actualmente à disciplina obrigatória de Computação Gráfica I, com a duração de um semestre e cujo número de alunos inscritos tem obrigado à existência de quatro a seis turnos de aulas práticas. Dos tópicos abrangidos pretende-se uma relativa diversificação, mas, sempre que possível, com uma índole mais formativa que informativa.

Quanto a um segundo módulo a nível superior para formação em Computação Gráfica, dever-se-á ser mais específico, com uma escolha de tópicos que apontem na direcção duma certa área de especialização. É o que acontece com a disciplina de Computação Gráfica II da LEI, de natureza facultativa mas igualmente semestral.

No plano curricular de um curso de licenciatura em Engenharia prevê-se a execução de um Projecto individual no último ano, que é uma excelente oportunidade para o aluno, com base nos conhecimentos até então adquiridos, poder conceber, decidir e implementar soluções para a resolução de problemas muito concretos. Naturalmente que a Computação Gráfica também poderá aqui entrar, em maior ou menor grau, consoante a natureza do problema de cada Projecto e as opções tomadas para a sua execução. Mas, devido a esta particularidade de não generalização e ao facto de estar mais relacionado com a aplicação de conhecimentos do que a métodos de ensino, a sua discussão sai fora do âmbito pretendido para a presente comunicação.

4.1.1 Computação Gráfica I

O principal objectivo da disciplina de Computação Gráfica I é dar a conhecer, quer as bases matemáticas, quer os principais conceitos e algoritmos usados para a programação de aplicações gráficas interactivas, com a correspondente componente prática de laboratório.

1. Definição, história e aplicações da Computação Gráfica
2. Principais algoritmos para conversão de primitivas gráficas e tratamento dos respectivos atributos em tecnologia raster
3. Recorte de linhas e de polígonos
4. As primitivas geométricas disponibilizadas por alguns sistemas gráficos

5. Mecanismos e funções de controlo importantes em PHIGS e em OpenGL™
6. Sistemas de coordenadas e transformações de enquadramento
7. Transformações geométricas elementares, em 2D e 3D, e sua composição
8. Utilitários para a especificação funcional de transformações geométricas
9. Projecções geométricas planas (paralela e perspectiva)
10. Métodos interactivos para o design de curvas 2D e 3D, incluindo as técnicas para representação e avaliação
11. Métodos interactivos para o *design* de superfícies, incluindo as técnicas para representação e avaliação
12. *Pipelines* de visionamento para 2D
13. Conceitos de modelação hierárquica e edição, com a correspondente aplicação em PHIGS e em OpenGL™.
14. Dispositivos lógicos de entrada e técnicas de interacção
15. Técnicas de implementação dos dispositivos lógicos Locator e Pick

É na disciplina de Computação Gráfica I que os alunos se familiarizarão com a existência de normas ISO e ANSI ao nível API (*Application Programmer Interface*). Muito embora possam ser apontadas grandes desvantagens ao uso generalizado de normas, como sejam

- a não adaptação a casos especiais
 - a rápida desactualização em relação ao estado da arte
- a realidade é que elas têm também um papel importante na educação, garantindo
- boa estruturação dos conceitos
 - independência dos fabricantes
 - segurança, qualidade e consistência na utilização.

Para além destes aspectos, tais normas têm implementações concretas no mercado ao nível de empresas bastante importantes no ramo informático, sendo opções reais na vida profissional de um engenheiro.

A formação do aluno em Computação Gráfica I tem por base a programação sobre a plataforma Windows [6] e o uso de subconjuntos dos sistemas gráficos PHIGS e OpenGL™ [4], sendo GKS apenas referido nos aspectos de evolução histórica e como introdução simples aos conceitos mais complexos.

No entanto, as referências à normalização no que diz respeito às matérias relacionadas com o espaço 3D, pela sua maior complexidade matemática que obrigaria a reduzir o número de tópicos abordados na disciplina, são ser deixadas para o módulo seguinte de Computação Gráfica.

O significado que se atribui a “utilizador dum sistema gráfico” pode diferir substancialmente, dependendo do ponto de vista. Com efeito, um programador de aplicações utiliza um sistema gráfico escrevendo código, o que está muito longe de ser o procedimento do utilizador final da aplicação. Vem esta observação com o propósito de chamar a atenção para a existência de diferentes níveis de abordagem nos tópicos acima apresentados: o nível funcional, mais elevado, o nível procedimental, intermédio, e o nível dos dispositivos físicos, considerado mais básico. Como consequência, o aluno não só ficará com um conhecimento que lhe permitirá agir na esfera do programador de aplicações (API), como ter acesso às técnicas indispensáveis à construção do próprio sistema gráfico, acrescentando-se ainda uma capacidade de interferência na construção e/ou manutenção dos controladores dos dispositivos gráficos e noções mais exactas sobre a problemática das interfaces gráficas.

4.1.2 Computação Gráfica II

Em Computação Gráfica II, de carácter opcional, não tem havido necessidade de subdivisão do único turno de laboratório. Com esta disciplina pretende-se ampliar os conhecimentos anteriores dos alunos com as técnicas para a produção de imagens de síntese realistas, mais concretamente nos aspectos que dizem respeito ao visionamento e desenho de cenas tridimensionais. A exploração dos gráficos 3D para plataformas de uso em larga escala e a crescente aplicação em diversas áreas da actividade humana dão ao tema, como se sabe, uma grande actualidade [1].

O programa que se escolheu para esta disciplina apoia-se fortemente nos conhecimentos adquiridos pelos alunos no decurso de Computação Gráfica I, resultando numa maior especialização em assuntos de um domínio mais específico:

1. Introdução ao problema da síntese de imagem fotograficamente realista
2. Integração das projecções geométricas planas
3. Volumes de visão para as projecções paralela e perspectiva arbitrárias
4. Paradigma da máquina fotográfica
5. Transformação normalizante e transformação de perspectiva
6. Recorte de cenas 3D em coordenadas homogéneas
7. Técnicas e algoritmos para remoção de linhas e/ou superfícies ocultas
8. Cor acromática e técnicas de meios-tons
9. Princípios de Colorimetria
10. Modelos de Cor
11. Regras para uso e harmonia das cores
12. Noções de Fotometria
13. Modelos de Iluminação

14. Efeitos especiais
15. Métodos de sombreado de superfícies em malha de polígonos
16. Texturas 2D por mapeamento
17. Ray-Tracing
18. Radiosidade
19. Técnicas de *anti-aliasing* por pós-processamento

4.2 Em Pós-Graduação

A maior dificuldade para a concepção de um programa de Computação Gráfica num curso de pós-graduação em Engenharia Informática é, talvez, a da grande heterogeneidade que é muito provável existir na formação de base dos alunos envolvidos. Uma possibilidade de resolução deste problema estaria na oferta de cursos com perfis variados, perante os quais os alunos fariam as escolhas mais adequadas. Mas a realidade é bem diferente, pois não existem normalmente recursos suficientes para tal diversidade. Além disso, e do ponto de vista da legislação actual, nem sequer pode ser autorizado um mestrado com o elevado número de disciplinas que esta metodologia implicaria.

A estratégia de elaboração dos programas deverá então ser feita de molde a introduzir tópicos que, pela sua novidade e pertinência, interessem a qualquer aluno em geral, mas que não venha a penalizar excessivamente os que não possuam tão boa formação curricular anterior. O sucesso deste compromisso é muito difícil de garantir à partida, pelo que se sugere a adopção dum programa inicial que possa ir sofrendo adaptações diversas ao longo do próprio semestre lectivo.

Um programa aceitável para o MEI deverá, pois, permitir que os alunos menos experientes alcancem rapidamente um nível de preparação suficiente para a compreensão dos principais temas a tratar na disciplina. Neste contexto, o programa actual baseia-se nos seguintes tópicos:

1. Enquadramento da Computação Gráfica
2. Principais métodos de representação gráfica
3. Elementos de geometria computacional
4. Modelação geométrica de suporte à imagem de síntese
5. Modelos e esquemas de representação de sólidos, com particular referência a B-rep, Octree, BSP e CSG
6. Conceitos, paradigmas e métodos de programação usando PHIGS/PHIGS+, OpenGL™, Java e VRML
7. Modelos de referência e normas API
8. Métodos de visualização de dados, incluindo a geração de iso-superfícies, sistemas de partículas e técnicas de visualização em volume.

Como naturalmente se nota, os primeiros assuntos destinam-se à uniformização de conhecimentos dos diversos alunos. Porém, é deixado espaço suficiente para

a introdução de temas actuais da especialidade. Por isso, os tópicos acima enumerados servem o propósito de enquadrar as diversas matérias a serem trabalhadas durante o período lectivo e não devem entender-se como uma mera compilação de assuntos de rigidez absoluta.

5. ENSINO EM LABORATÓRIO

Através dos imprescindíveis trabalhos de laboratório pretende-se dar uma sensibilização aos alunos para as diferentes técnicas que lhes terão sido apresentadas, na parte teórica, como solução de problemas muito concretos. Algumas dessas técnicas encontram-se já hoje implementadas no próprio hardware, mas continua a haver duas justificações fundamentais para o facto de virem a ser enunciadas em trabalhos práticos. Em primeiro lugar, pela sua vertente didáctica, podendo o aluno reflectir nas diversas implicações inerentes à própria estratégia de implementação. Por outro lado, uma determinada técnica poderá vir a ser utilizada num outro contexto, havendo nesse caso necessidade que o aluno a programe com as devidas adaptações.

5.1 Trabalhos Práticos Introdutórios

Dá-se conta, seguidamente, da configuração de trabalhos que os alunos resolverão em grupo e durante as aulas práticas do primeiro módulo de Computação Gráfica da licenciatura, servindo-se, para tal, de computadores pessoais e de C/C++ como linguagem de programação.

5.1.1 Programação em Windows

Com o intuito de introduzir os conceitos básicos ao nível API do Windows 95, em especial a forma de detectar e gerir a ocorrência de eventos, começa-se por mostrar aos alunos um programa já completo, com o qual é possível a um utilizador final especificar segmentos de recta pela técnica interactiva de *rubber-banding*. Cada grupo é então incentivado a experimentar alterações no código, registando os seus efeitos durante a execução do programa.

5.1.2 Preenchimento de polígonos

De entre os algoritmos de conversão para raster das primitivas gráficas, escolheu-se o do preenchimento de polígonos. Este requer, da parte do aluno, bons conhecimentos sobre estruturas de dados e programação. A especificação de cada polígono é feita por *rubber-banding*, por forma a aproveitar-se o código do programa anterior, já exercitado, de introdução ao Windows.

É de salientar que, muito embora a API utilizada disponibilize formas mais eficientes de obter o mesmo resultado, o programa construído por cada um dos grupos de alunos pode ter um tipo de utilização futura impossível de conseguir-se de outra maneira, como se verá mais adiante no caso de Computação Gráfica II. Por outro lado, este algoritmo é paradigmático em relação à técnica de varrimento presente em muitos outros, como sejam os de Watkins e de Z-buffer.

5.1.3 Transformações de enquadramento

Enuncia-se um problema que envolve múltiplas transformações de enquadramento Janela-Visor, convertendo-se as coordenadas do mundo real (coordenadas universais) para as do dispositivo. A solução terá que ser resolvida pelos alunos sem recurso às funções API específicas do ambiente de programação. Dessa forma, pretende-se que a matemática do problema seja bem compreendida pelos alunos, tendo como base a composição de operadores lineares construídos a partir das transformações geométricas elementares.

5.1.4 Projecções geométricas planas

Este trabalho é uma continuação do anterior, pois a área de desenho irá corresponder a quatro quadrantes, considerados visores e destinados a conter as imagens de determinados objectos tridimensionais após projecção no plano (Fig 1).

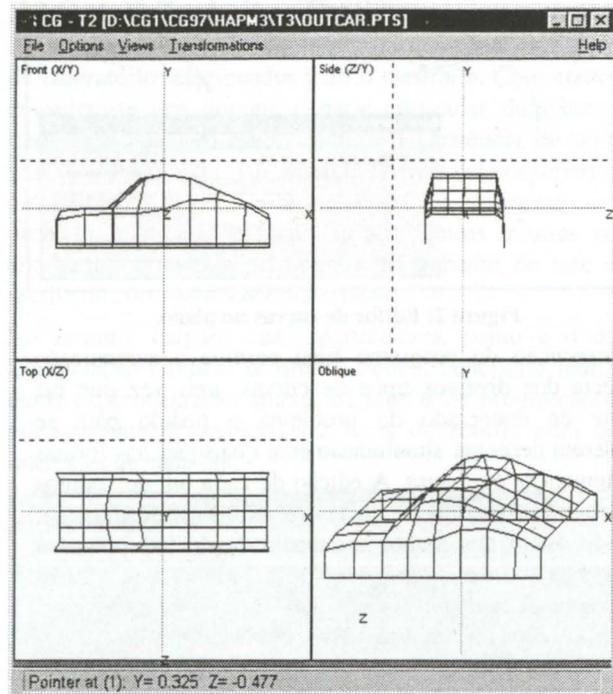


Figura 1: Projecções geométricas.

Previamente preparado está já um conjunto de ficheiros, contendo cada um deles a representação geométrica dum objecto simples definido num espaço tridimensional normalizado. É divulgada a sintaxe necessária à interpretação dos dados, bem como as regras semânticas associadas à topologia do modelo. Trata-se duma representação por linhas, que se justifica plenamente, quer pela simplicidade de implementação do algoritmo de visualização, quer pelo restrito conteúdo programático da disciplina neste domínio específico.

No ecrã, três dos quadrantes constituem o destino para igual número de vistas do objecto escolhido quando submetido a projecção ortogonal múltipla: alçado principal, planta e alçado lateral esquerdo. O quarto irá conter a imagem do objecto após uma projecção oblíqua, axonométrica ou perspectiva, como é habitual encontrar-

se no desenho técnico que apoia outros ramos da engenharia.

5.1.5 Curvas de formas livres

Pretende-se, neste caso, a construção de um muito elementar editor de curvas no plano. Com a finalidade dos alunos experimentarem a diversidade de comportamentos devida às diferentes formulações matemáticas descritas nas aulas teóricas, exige-se que a resolução permita o traçado de curvas cúbicas de Bézier, de Catmull-Rom, Beta-spline e B-spline (Fig. 2). Para diminuir o esforço de programação, o número de pontos de controlo das curvas é fixado à partida.

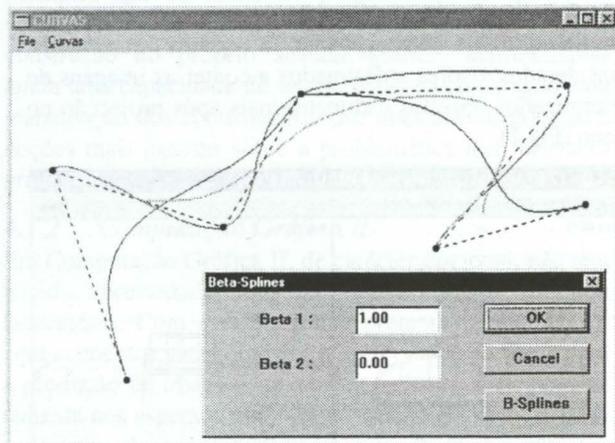


Figura 2: Editor de curvas no plano.

A execução do programa final permite a comparação directa dos diversos tipos de curvas, uma vez que faz parte do enunciado do problema o pedido para se poderem desenhar simultaneamente quaisquer das formas matemáticas em causa. A edição de cada um dos pontos de controlo também faz parte dos requisitos do trabalho, dando assim aos alunos a capacidade de interpretar experimentalmente as conclusões que lhes foram previamente demonstradas nas aulas teóricas por manipulações matemáticas das fórmulas utilizadas.

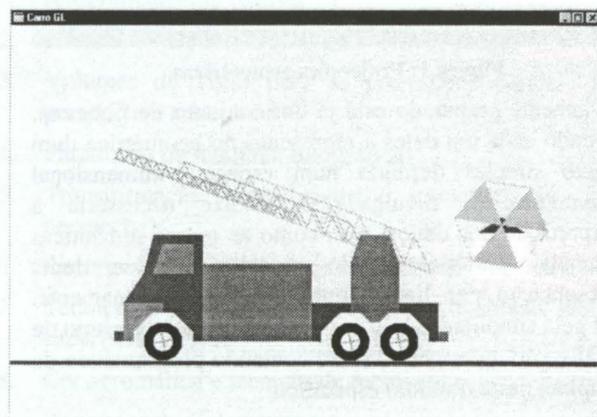


Figura 3: Modelação hierárquica em 2D.

5.1.6 Transformações geométricas

A programação num sistema gráfico independente da plataforma utilizada é iniciada por este trabalho em que, para além de se pretender ensinar as bases de OpenGL™,

são ao mesmo tempo exercitadas as transformações geométricas 2D e a modelação hierárquica.

No exemplo mostrado na Figura 3, o enunciado do problema era o da construção de um carro de bombeiros. Embora com a restrição ao espaço 2D, a composição hierárquica dos diversos componentes deveria permitir que se pudesse controlar um certo número de funções do veículo como, por exemplo, andar para a frente ou para trás, elevar/baixar as escadas, ou alterar o comprimento das mesmas.

5.1 Trabalhos Práticos Avançados

Como se referiu anteriormente, o principal objectivo da disciplina de Computação Gráfica II é o da construção de imagens com realismo. Ora neste caso, contrariamente ao que acontecia com a disciplina precedente, coloca-se imediatamente o problema do tempo necessário à realização dum conjunto satisfatório de trabalhos de laboratório nessas matérias, cujas complexidade e interdependência são relativamente elevadas. Com efeito, torna-se muito difícil, senão impossível, querer que o aluno aplique isoladamente cada uma das técnicas mais significativas, de acordo com a escolha dos docentes responsáveis. Em favor duma maior integração, daqui resulta uma diminuição natural do número de trabalhos laboratoriais.

Tenta-se, então, que as diversas técnicas sejam englobadas por apenas um ou dois trabalhos, por ordem tal que a implementação de umas possa servir de base às seguintes. Os trabalhos deverão, pois, ser sequencializados no tempo por forma a irem avançando à medida que os fundamentos necessários vão sendo tratados nas aulas teóricas.

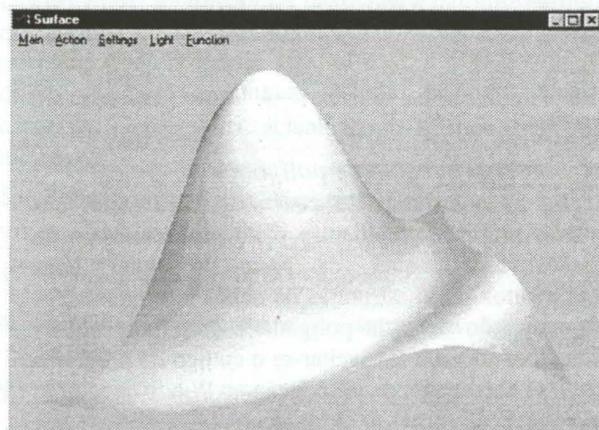


Figura 4: Criação e visualização de superfícies.

Seguindo a direcção apontada pela generalidade do software gráfico que se encontra actualmente no mercado, a imagem duma cena tridimensional será obtida por *rendering* de polígonos. Torna-se, portanto, imprescindível que o aluno use uma ferramenta para preenchimento de polígonos. No entanto, as implementações encontradas no software comercial, apesar de até serem frequentemente suportadas por hardware dedicado, não são suficientemente versáteis para permitir o controlo dos atributos de cada pixel. Ora este controlo é indis-

pensável à independência dos programas em relação às técnicas de iluminação e sombreado a usar posteriormente. Por tal motivo, embora privilegiando a versatilidade em relação à eficiência, o aluno deverá reutilizar aqui a implementação desenvolvida no laboratório de Computação Gráfica I.

Os aspectos da modelação geométrica não são aprofundados nesta disciplina, pelo que as representações geométricas a usar nas aulas práticas são normalmente muito simples. As opções a levar em conta têm sido as seguintes:

- Cálculo analítico, com base na avaliação de expressões matemáticas.
- Geração procedimental de objectos parametrizados.
- Representações geométricas criadas por uma qualquer aplicação adequada e lidas de ficheiros exteriores onde tenham sido armazenadas.

A Figura 4 ilustra uma situação que se enquadra no primeiro caso, visto tratar-se de funções dadas na forma explícita, enquanto que no trabalho relativo à Figura 5 se usou a segunda opção.

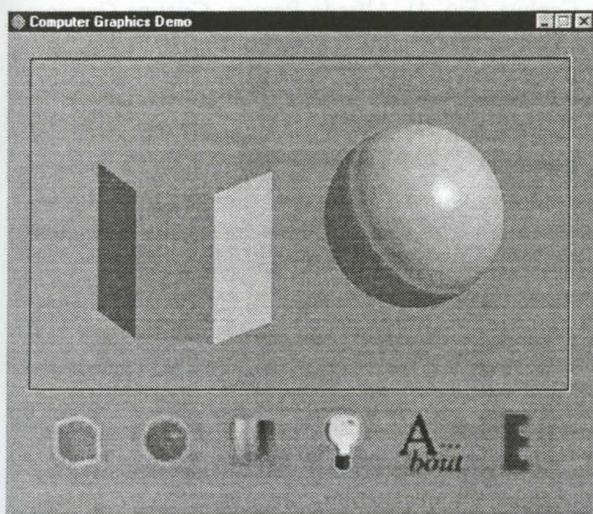


Figura 5: Modelos de iluminação local e sombreado.

A simulação de superfícies curvas é uma das últimas técnicas que o aluno terá que implementar no seu trabalho. Com esse objectivo, deverá usar os dois métodos de sombreado por interpolação que constam do programa da disciplina: o de Gouraud e o de Phong. Através duma interface adequada, estes métodos podem ser activados ou desactivados pelo utilizador final do programa. Além disso, os valores de todos os parâmetros da equação do modelo de iluminação escolhido podem também ser modificados interactivamente. Do ponto de vista didáctico, torna-se extraordinariamente vantajoso para o aluno o facto de poder interpretar visualmente o modelo experimentando qualquer combinação de valores que, por outro lado, também servirão para validar a própria implementação.

No tipo de trabalhos exemplificados pelas Figuras 4 e 5, todos eles programados directamente em Windows, o tratamento das projecções é muito superficial. De que

maneira se deveria aprofundar esta matéria se é grande a limitação temporal imposta curricularmente? Com efeito, o esforço para construir, de raiz, um *pipeline* completo que permitisse projecções arbitrárias, com operações de recorte e algoritmos para remoção de partes ocultas de aplicação geral, embora não sendo impossível, implica um acréscimo em disponibilidade de tempo que não é viável. A alternativa que ultimamente tem sido levada a cabo em Computação Gráfica II é a do uso de OpenGL™. Tem-se revelado uma boa escolha, permitindo uma rápida aprendizagem dos principais conceitos e técnicas no tratamento duma geometria 3D, tanto mais que os alunos já se encontram familiarizados, nesta altura, com uma utilização em 2D (Fig. 3). Pode-se, então, enunciar um trabalho suplementar nesta área, incluindo também nos seus objectivos uma introdução simples à animação em computador [10].

5.3 Outros Trabalhos Práticos

As secções anteriores não referiram quaisquer trabalhos de laboratório relacionados com o mestrado. Com efeito, considera-se que durante a parte curricular dum curso deste tipo não tem muito sentido a existência de uma grande componente laboratorial. A verdadeira oportunidade para investigar e resolver integralmente um problema complexo é facultada aos futuros mestres no ano seguinte, com a preparação do trabalho de tese e escrita da correspondente dissertação.

No entanto, alguns casos particulares, como é o da Computação Gráfica, os alunos podem beneficiar muito duma experimentação através de tarefas de programação relativamente diversificadas e mais de acordo com os interesses de cada um.

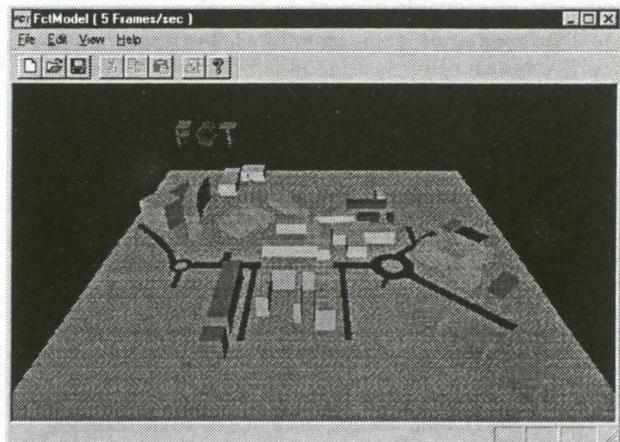


Figura 6: Modelação 3D e controlo do visionamento em tempo real.

Recomenda-se, assim, que os alunos de mestrado venham a programar pequenas aplicações onde sintam o nível de complexidade de diversos problemas analisados durante as aulas. Pode, inclusivamente, ser esta uma actividade imprescindível para o aluno que pretenda realizar posterior trabalho de investigação nesta área, como seja o da preparação da dissertação final.

As propostas actuais para este tipo de trabalhos passam pelo uso de OpenGL™, Java e/ou VRML [7]. A Figura 6

mostra, a propósito, um exemplo programado em OpenGL™.

6. CONCLUSÕES

A evolução dos métodos de ensino, duma maneira geral, resulta fundamentalmente duma análise das experiências passadas. Mas os sistemas são muito complexos e os responsáveis pela educação nem sempre, conscientemente ou não, estão de posse de todos os parâmetros relevantes. É claro que se devem considerar sempre como potencialmente úteis para aplicação a um determinado caso a tratar as metodologias que tenham dado bons resultados em situações semelhantes. Mas pode existir alguma pequena diferença, aparentemente insignificante, que venha a implicar uma completa inadequação ao caso concreto. Por vezes o ensino e/ou a aprendizagem podem ser substancialmente melhorados tendo apenas em conta certos pormenores. Por exemplo, notou-se que a motivação dos alunos no laboratório de Computação Gráfica I aumentou substancialmente quando se passou a dar, como primeiro trabalho prático, uma tarefa pré-programada de que, em poucos minutos, resultava uma imagem no ecrã do computador [3]. Por outro lado, a experiência de ensino a nível universitário tem mostrado que as indesejáveis consequências da aglomeração de trabalhos das várias disciplinas no final de um semestre podem ser drasticamente reduzidas se se estipular que a apresentação dos programas construídos pelos alunos se vá fazendo também ao longo das aulas, com datas marcadas e inadiáveis aquando da entrega do enunciado do problema a resolver.

A vastidão da área actualmente abrangida pela Computação Gráfica não permite que esta seja razoavelmente coberta por um reduzido número de disciplinas semestrais em qualquer curso de nível superior. Observa-se, assim, que as diversas instituições que têm a seu cargo o ensino da Computação Gráfica de alguma forma elegeram certos temas mais orientados para o tipo de formação por elas pretendido e neles se especializaram. Esta é, contudo, uma tarefa difícil para os docentes responsáveis, e deveria implicar, naturalmente, uma análise de cada edição decorrida por ano lectivo para, tendo como referência o estado da arte, as exigências do mercado e a configuração própria do curso, permitir a definição de novas estratégias programáticas a aplicar no futuro.

A referência [5] diz respeito ao projecto HyperGraph, para ensino da Computação Gráfica. A título comparativo, e retirando completamente desse currículo os aspectos mais relacionados com o hardware ou com algum produto comercial de software, os restantes temas nele agrangidos são quase todos, em maior ou menor grau de profundidade, referenciados em pelo menos um dos três módulos semestrais de Computação Gráfica da FCT-UNL.

Mas existe uma diversidade de cursos universitários que apresentam vertentes da Computação Gráfica muito diferentes, como acontece na Universidade de Stanford [11]. De facto, dos nove cursos aí propostos, apenas dois

ou três se poderiam directamente equivaler aos da FCT-UNL. Tal não significa, no entanto, que os restantes não tivessem interesse para a formação em Engenharia Informática. Porém, a não ser que se pudessem criar mais disciplinas opcionais nos actuais cursos de licenciatura e mestrado, no sentido de um complemento configurável pelo aluno ao conhecimento adquirido em disciplinas de carácter mais nuclear, os principais objectivos de formação dos alunos teriam certamente que ser alterados para que umas matérias viessem então a substituir outras.

7. AGRADECIMENTOS

O universo dos trabalhos laboratoriais efectuados pelos alunos de Engenharia Informática da FCT-UNL no âmbito das disciplinas referenciadas nas secções anteriores tem mostrado, felizmente, uma qualidade considerável. É disso exemplo o facto de todas as figuras utilizadas ao longo do texto desta comunicação corresponderem a trabalhos cuja selecção não envolveu nenhum esforço especial da parte do autor desta comunicação. A autoria das figuras é atribuída aos alunos Hugo Abreu e Paulo Matos (Fig. 1), Telmo Cruz e Paulo Santos (Fig. 2), Sérgio Estêvão e João Feliciano (Fig. 3), João Canhita (Fig. 4), Alfredo Pereira, Cristina Garcia e Pedro Valente (Fig. 5), e Rui Mendes (Fig. 6).

8. REFERÊNCIAS

- [1] Donna Coco. 3D Stars on TV. *Computer Graphics World*. 21(4), 40-51 (April 1998).
- [2] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes. *Computer Graphics — Principles and Practice*. Second Edition in C. Addison-Wesley (1996).
- [3] F.W. Jansen, P.R. Nieuwenhuizen. Computer Graphics Education at Delft University of Technology. *Proceedings of the Eurographics Workshop on Graphics and Visualization Education*, Oslo (1994).
- [4] J. Neider, T. Davis, M. Woo (OpenGL ARB). *OpenGL Programming Guide*. Second Edition, Version 1.1. Addison-Wesley (1997).
- [5] G. Scott Owen (project director). HyperGraph — Teaching Computer Graphics: A project of the ACM SIGGRAPH Education Committee, the Hypermedia and Visualization Laboratory, Georgia State University, and the National Science Foundation. <http://www.education.siggraph.org>
- [6] C. Petzold. *Programming Windows 95*. Microsoft Press (1996).
- [7] B. Roehl, J. Couch, C. Reed-Ballreich, T. Rohaly, G. Brown. *Late Night VRML 2.0 with Java*. Ziff-Davis Press (1997).
- [8] H.P. Santo (Ed.). Actas do *Workshop: Computação Gráfica no Ensino*, Grupo Português de Computação Gráfica — EUROGRAPHICS, Lisboa (1991).
- [9] A.A. de Sousa, J.C. Teixeira (Ed.). Actas do *Oitavo Encontro Português de Computação Gráfica*, Grupo

Português de Computação Gráfica —
EUROGRAPHICS, Coimbra (1998).

[10]Nadia Thalmann, Daniel Thalmann (Ed.). *Interactive Computer Animation*. Prentice-Hall Professional Technical Reference (1996).

[11]SCGL, Courses in Graphics, Stanford University.
<http://graphics.stanford.edu/courses/>