

# Mestrado em Computação Gráfica e Ambientes Virtuais

## H2: Programação e Interacção

Avaliação por trabalho individual

Ano Lectivo 2005/06

### Introdução

O trabalho que a seguir se apresenta constitui a avaliação da disciplina H2: Tópicos de Programação e Interacção do Mestrado em Computação Gráfica e Ambientes Virtuais. O objectivo principal do trabalho é consolidar os conhecimentos adquiridos nas aulas e que visaram fornecer mecanismos de homogeneização nas linguagens de programação que vão ser necessárias no decurso do mestrado.

Com vista ao reforço das noções de programação em C++, o presente trabalho opta, a exemplo dos anos anteriores, por uma problemática retirada da área da computação gráfica, por forma a que esforços efectuados no âmbito da resolução do trabalho possam eventualmente ser reutilizados noutras disciplinas.

Também como motivação extra para a utilização do paradigma da programação por objectos e da consequente reutilização de componentes de software, neste trabalho é fornecido um módulo integralmente desenvolvido que deverá ser integrado e encapsulado, por forma a criar uma estrutura mais complexa.

O tempo para resolução do trabalho é de 3 semanas, estando a sua apresentação oral marcada para o fim mês de Dezembro de 2005. Os alunos devem enviar email para [anr@di.uminho.pt](mailto:anr@di.uminho.pt) por forma a sugerir alternativas para o dia da apresentação.

Durante a apresentação os alunos devem entregar um *pequeno relatório* com a descrição do trabalho realizado, principais decisões tomadas - ao nível das estruturas de dados - e anexar o código fonte.

### Objectivos

O trabalho proposto para esta disciplina tira partido do investimento feito ao longo das aulas em perceber e resolver o trabalho do ano anterior, reutilizando conceitos nessa altura explorados, como é o conceito e definição de **bounding box** e **bounding sphere**.

É objectivo do trabalho que se possam efectuar as seguintes operações:

1. Utilizar o *loader* de objectos fornecido por forma a ler ficheiros OBJ, que contém informação sobre vértices de objectos (entre outras coisas). O código fornecido tem funções que permitem carregar a informação existente no ficheiro para uma estrutura que descreve um modelo e que fica preenchida com os dados existentes num ficheiro OBJ. O método principal a utilizar para carregar a informação de um ficheiro para as estruturas de dados é `glmReadOBJ`, em que como parâmetro aceita o nome do ficheiro;
2. Modificar o *loader* de objectos por forma a encapsular o código numa classe que permita carregar informação de mais do que um ficheiro. O loader fornecido cria uma estrutura para cada ficheiro lido. É necessário criar uma classe que permita invocar várias vezes a função de carregamento para memória e guarde os vários modelos lidos numa estrutura de dados (array ou vector);
3. Criar uma hierarquia de classes que permita ter várias classes que sobre uma núvem de pontos, definam uma estrutura de fronteira (bounding) que englobe os pontos existentes no espaço;
4. Criar 2 tipos de estruturas de bounding - uma **bounding sphere** e **axis-aligned bounding box** (um cubo alinhado com os eixos). Para a bounding sphere é necessário saber qual é o ponto central e o raio, enquanto que para a bounding box o programa deve dar indicação dos quatro vértices;
5. Identificar se 2 estruturas (cada uma para um ficheiro OBJ) se intersectam;

## Entrega do trabalho

Por forma a entregar o trabalho deve criar um programa (um ficheiro.cpp) que contenha um método `main()` no qual declara as variáveis que forem necessárias. Na entrega do trabalho poderemos alterar o programa para acrescentar mais informação (por exemplo: mais ficheiros OBJ)

## Anexos

Na página da disciplina encontra o código do *loader* de ficheiros OBJ e um ficheiro exemplo. Pode encontrar mais exemplos, entre os muito recursos Web existentes, em:

<http://www.csit.fsu.edu/~burkardt/data/obj/obj.html>