

Curso de CG 2014/2- Unidade 1- Modelagem de objetos
Exercício 1 (individual) - Entrega: quarta 13/08

Escolha um objeto entre os que possivelmente compõem uma **clínica médica de exames mastológicos** para você modelar como primeiro trabalho do curso. Não precisa criá-lo ainda, mas apenas defini-lo. A escolha é individual, se houver 2 objetos iguais o professor se reserva o direito definir outro, ao aluno, para substituí-lo.

Entregue respondidas **a mão**, isto é manuscrita (não será aceito qualquer forma de impressão) as respostas às perguntas que seguem:

1. O **Blender** é baseado no **OpenGL** e independente do Sistema Operacional. Qual o **tipo** de sistema de coordenadas usado no **Blender**, no **OpenGL** e na maioria dos sistemas genéricos para CG?
2. Como transformar o **tipo** de sistema de coordenadas usado no **Blender** em outro **tipo** mais útil em determinada aplicação como os **polares, cilíndricos e esféricos**?
3. Um sistema de coordenadas cartesiano é composto de 3 planos e 3 eixos ortogonais. O Blender usa letras maiúsculas para representar cada um destes eixos. Como eles podem ser chamados? O que é um sistema cartesiano **positivo** ou com os **eixos orientados pela regra da mão direita**?
4. O que **eixos orientados pela regra da mão direita** têm a ver com o **produto vetorial** da álgebra linear?
5. Mesmo usando sempre apenas sistemas de **coordenadas cartesianos**, no **Blender** existem diversos sistemas de **coordenadas** ativos ao mesmo tempo em uma cena (**desde a tela de abertura inicial**). Quantos sistemas de **coordenadas cartesianos** você consegue identificar?
6. A maior parte das aplicações da CG visa criações de objetos do nosso mundo, ou objetos do mundo 3D que podem existir, os chamados **Sólidos Realizáveis**. Estes são descritos sempre a partir de coleção de pontos com 3 coordenadas cada um. O que significa cada uma das coordenadas?
7. As coordenadas de um ponto só fazem sentido em relação a um sistema de eixos de coordenadas perfeitamente caracterizado: *i.e.* Centralizado em um ponto bem definido (chamado **origem** do sistema de coordenadas). **Explique** porque também é importante identificar a unidade usada e a direção considerada positiva em cada eixo.
8. Na maior parte das aplicações da CG, mesmo quando os objetos parecem ter superfícies não planas, eles serão de fato representados por **faces (F)** planas, limitadas por **arestas (A)** retas. Cada **aresta** ou lado das faces são limitados por pontos chamados **vértices (V)**. Para simplificar ainda mais a programação as faces dos objetos **não** são polígonos quaisquer e **sim**, geralmente, apenas **triângulos** ou **quadriláteros**. Para que um objeto poliédrico de **genus zero** seja realizável é condição necessária que seu número de **vértices (V)**, **arestas (A)** e **faces (F)** satisfaça a chamada **equação de Euler** (atenção: como esse nome é suíço-alemão você deve ler “**oiler**”) : $V-A+F=2$. Desenhe um sólido qualquer, mas **diferente** de um cubo. Descreva as coordenadas dos seus vértices em relação a um sistema de

coordenadas que você definir de modo que seu objeto seja perfeitamente desenhado. Verifique se ele satisfaz a equação de Euler.

9. Como queremos que em CG um objeto pareça o mais real possível, precisamos lhe dar aspecto de ser um objeto real (**renderizá-lo**), isso é feito incluindo nas suas faces a aparência de serem constituídas de um material com cor, texturas, brilho e outras características de uma superfície real. Para isso ele deve ser composto de faces. Como é importante identificar a **direção perpendicular** de cada face, ou seja, sua **normal**, cada fase deve ser identificada por uma lista de vértices em uma ordenação orientada seguindo a regra da **mão direita**. Assim precisamos ter um **estrutura de dados** para um objeto que seja composto por pelo menos 2 listas: (1) Uma lista de faces, onde cada face é identificada pelo nome dos seus vértices ordenados seqüencialmente de modo que sua normal aponte para a direção externa do objeto; e (2) um lista de vértices onde cada um seja identificado pelas suas coordenadas. Essa estrutura de dados para representação de objetos em CG é chamada de “**Baseada em Vértices**”. Represente essas listas para o objeto que você desenhou na questão anterior.

10. Existem outras estruturas de dados possíveis em CG. Por exemplo: uma estrutura de dados onde cada aresta seja também identificada é muitas vezes mais útil. Com ela é mais fácil representar um objeto apenas como **wire frame** e selecionar seus limites para modificar suas formas. Assim surge outra estrutura de dados baseada em 3 listas: (1) Uma lista de faces, onde cada face é identificada pelo nome das arestas ordenadas seqüencialmente de modo que façam um **loop** no sentido anti-horário; (2) um lista de arestas onde cada uma seja identificada pelos seus vértices na direção adequada para que a normal da face aponte para a direção externa do objeto; e (3) a lista de coordenadas dos vértices (como na questão anterior). Essa estrutura de dados para representação de objetos em CG é chamada de “**Baseada em lados**” (ou em “**arestas**”). Represente essas listas para o objeto que você desenhou em 9.

11. Voltando ao Blender: Indique a versão que você baixou e está usando. Descreva algumas formas de modelagem que esteja presente nesta versão para o usuário criar objetos.

12. Um conceito muito importante em CG é o da **bounding box** (BB) de um objeto. De uma forma bem intuitiva seria a **caixa mínima** (Box) que você usaria para poder guardar seu objeto. Alguns softwares confundem BB com a chamada de **Axial aligned bounding box** (AABB), onde a caixa seria paralela aos eixos usado no programa para descrever os objetos (que às vezes são chamados de sistemas de coordenadas do mundo, do universo, globais, etc.). Responda: Qual destas é usada no seu Blender como BB? Em relação a que sistema de eixos ele descreve os objetos?

13. Vamos continuar fixando esses 2 conceitos fundamentais: sistemas de coordenadas e coordenadas. Qual a diferença entre as operações de **panned** (panorâmica), **zoomed**, **translação do objeto**, e **mudança de escala nas 3 direções**?

14. Como você pode dar o mesmo efeito visual do **Zoom in** e **zoom out** através da **mudança de escala** do objeto? As coordenadas do objeto são alteradas em qual dos casos?

15. Como você pode dar o mesmo efeito visual do **panned left** (ou anti clock wise) e **panned right** (ou clock wise) através da **translação do objeto**? As coordenadas do objeto são alteradas em qual dos casos?