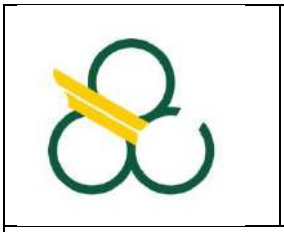


ESZI017 – Fundamentos de Processamento Gráfico

Laboratório 03: **Modelagem e Transformações**

1. Introdução

O objetivo desta aula é praticar e aprofundar os fundamentos de modelamento de objetos e de transformações, utilizando a linguagem C++ e a biblioteca **OpenGL**.

Este roteiro deverá ser executado no sistema **Ubuntu-linux** instalado no laboratório didático. Instruções são apresentadas e exemplos que o aluno deve executar no laboratório, e verificar os resultados obtidos. Ao final do roteiro são propostos alguns exercicios para entregar que o aluno deverá realizar individualmente.

2. Exemplos Básicos

Esta primeira parte de aula visa executar alguns projetos, e verificar os conceitos sobre: modelamento, transformações (translação, rotação, e escalonamento), aspectos de programação de animação e interatividade (pelo teclado e mouse), e ainda grafos de cena (através do uso do *Stack*).

Acesse o MOODLE, e faça o *download* do arquivo "lab03_modelagem_code.zip", e descompacte cada programa fornecido numa sub-pasta "lab03".

Os programas deverão ser elaborados no editor Geany, e compilados com o comando "gcc", conforme as instruções da aula anterior.

1º. Projeto: "model.c"

- Execute o programa
- Verifique no código a implementação dos seguintes conceitos:
  - . Desenho de um objeto (triangulo)
  - . Desenho do mesmo objeto Transladado
  - . Desenho do mesmo objeto Escalonado
  - . Desenho do mesmo objeto Rotacionado
  - . Matriz Indentidade. Por que ela foi carregada?
  - . Tecla para saída da execução do programa.
- Responda: quais os valores dos argumentos das transformações realizadas?
  - Translação: T(*dx, dy*)
  - Escalonamento: S(*sx, sy*)
  - Rotação: R(θ)

2º. Projeto: "double.c"

- Execute o programa
- Verifique no código a implementação dos seguintes conceitos:
  - . Buffer Duplo
  - . Desenho de um objeto
  - . Desenho do mesmo objeto Rotacionado
  - . Guarda da Matriz na Pilha. Por que isso deve ser feito?
  - . Uso do mouse para parar e iniciar a animação.

- Responda:
    . Qual o valor do angulo de Rotação: R(θ) em cada apresentação?
    . Por que a animação (mudança do desenho na tela) ocorre no programa?

3º. Projeto: "rotacao.c"
- Execute o programa
- Verifique no código a implementação dos seguintes conceitos:
    . Buffer Duplo
    . Uso das teclas para mudança de cor do desenho.
    . Por que não fez a guarda da Matriz na Pilha?
- Responda:
    . Qual o valor do angulo de Rotação: R(θ) em cada apresentação?

4º. Projeto: "planet.c"
- Execute o programa
- Verifique no código a implementação dos seguintes conceitos:
    . Desenho de objetos do GLUT. Qual é o Sol e qual é o planeta?
    . Uso das teclas para mudança de posição do planeta.
- Responda:
    . Quais as transformações utilizadas para os dois tipos de movimento do planeta?
    . Qual o valor do angulo de Rotação: R(θ) em cada apresentação?

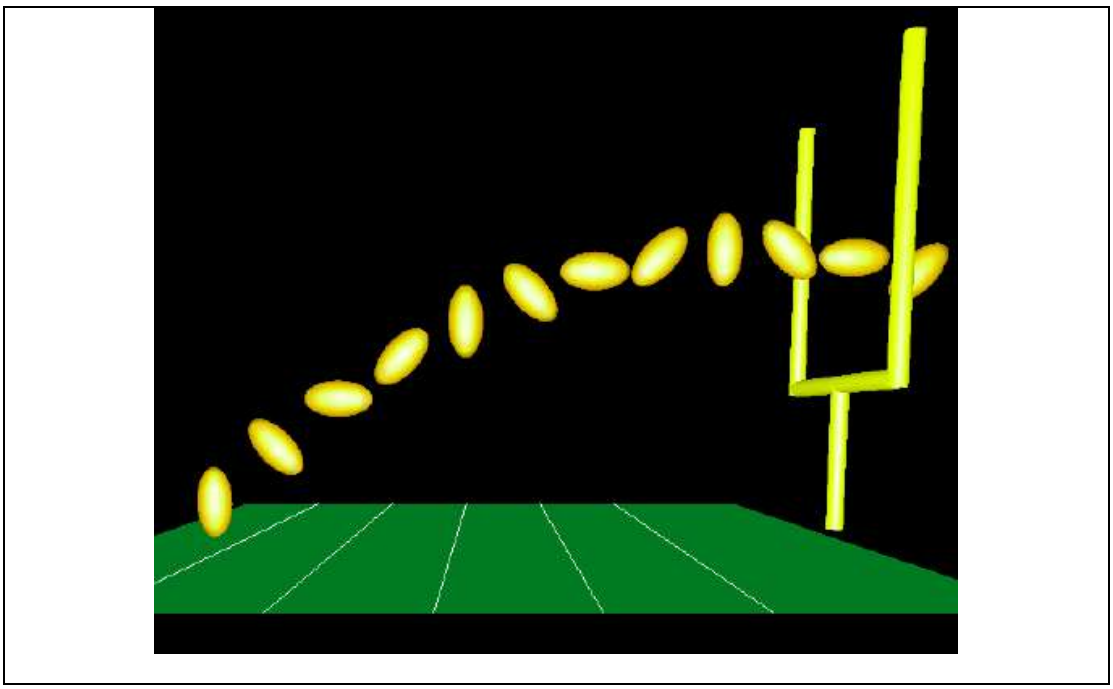
5º. Projeto: "robot.c"
- Execute o programa
- Verifique no código a implementação dos seguintes conceitos:
    . Desenho de objetos do GLUT. (Ombro e braço)
    . Modelamento composto de cena.
    . Uso das teclas para mudança de posição de forma independente.
    . Uso da pilha para guarda das matrizes de transformação.
- Responda:
    . Qual a dimensão inicial de cada um dos objetos da cena?
    . Qual a finalidade de cada caso de uso da pilha (*push* e *pop*) no método "display"?
    . Desenhe o grafo de cena num papel.

3. Programas a elaborar na aula

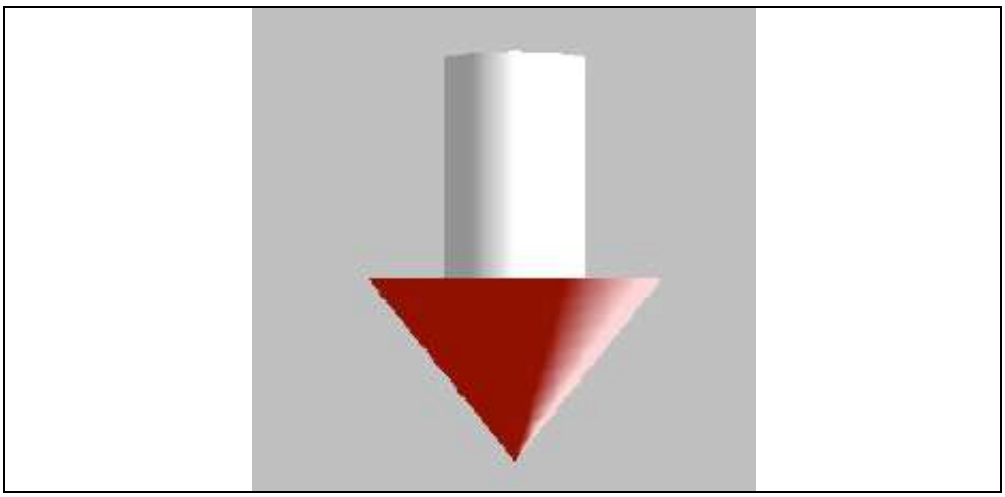
Nesta segunda parte o objetivo é praticar a elaboração de programas e aplicar os conceitos de modelamento de objetos, de transformações (translação, rotação, e escalonamento), de programação de animação e interatividade (pelo teclado e mouse), e ainda grafos de cena (através do uso do *Stack*). Solicita-se as imagens renderizadas sejam salvas no formato jpeg ou similar.

Importante: *em todos os arquivos de código faça um cabeçalho (com comentários) colocando o seu nome completo, RA, data do programa, nome do programa, e exemplo de chamada do programa no prompt do Linux. Além disso, no TÍTULO das janelas OpenGL criadas pelo programa coloque o título do programa e o seu nome e sobrenome.*

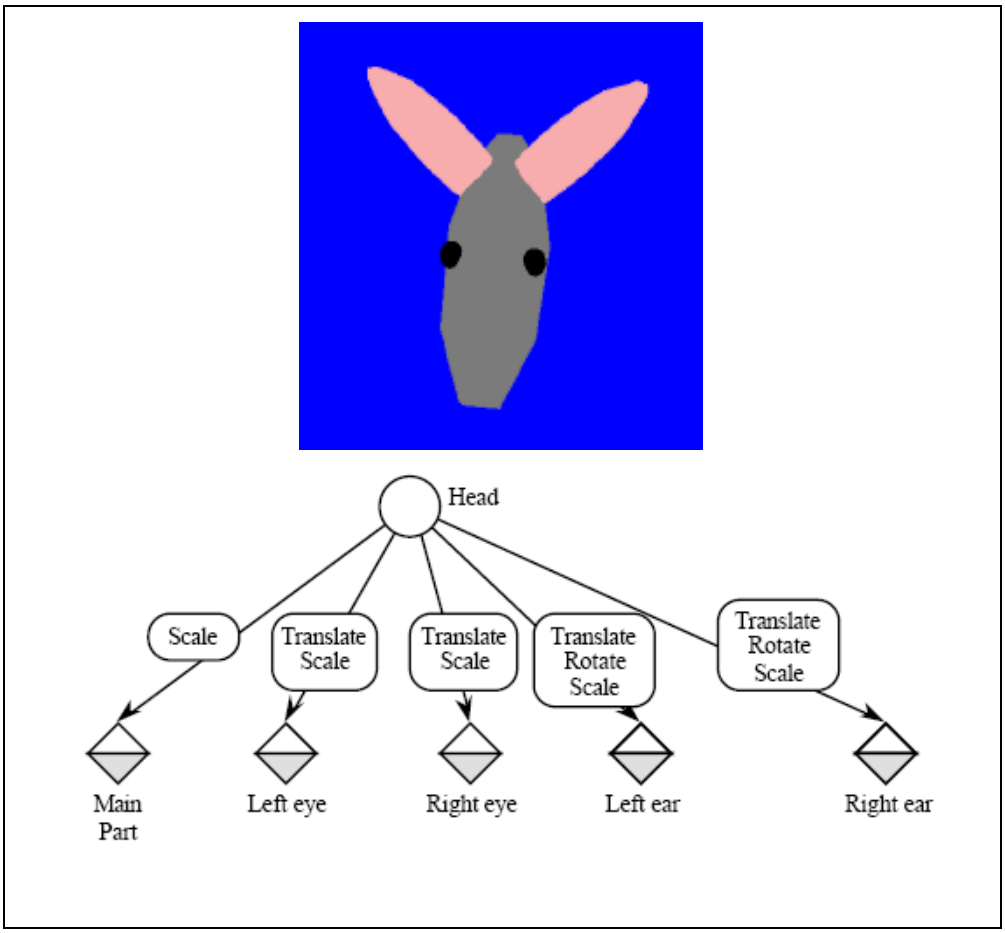
1. Elabore um programa para gerar uma única imagem que represente o movimento da bola de rugby, similar que está mostrado na figura abaixo.



2. Elabore um programa para gerar uma seta similar que está mostrado na figura abaixo, através de objetos do GLUT, e transformações adequadas.



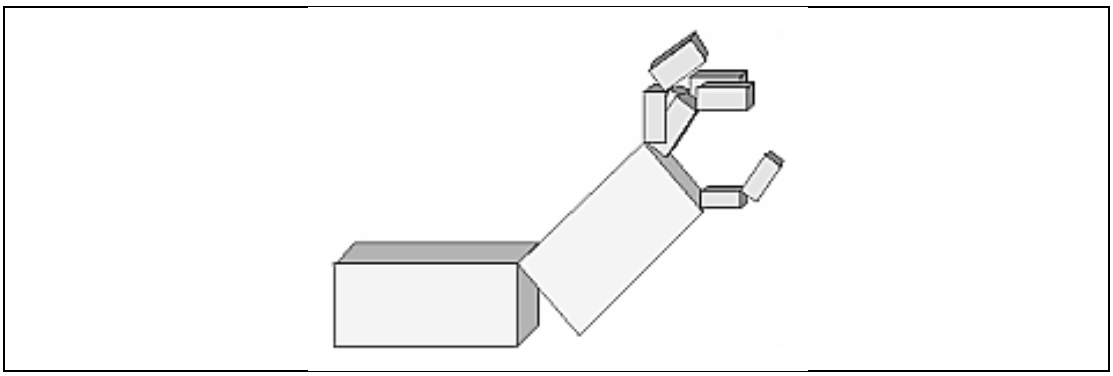
3. Exercício de modelamento composto e de grafo de cena. Elabore um programa para gerar desenho simples da cabeça um animal similar ao que está mostrado na figura abaixo.
A cabeça principal é um elipsoide grande, os olhos são duas pequenas esferas, e as orelhas são dois elipsoides de tamanho medio.
Para montar o desenho, utilize o conceito de grafo de cena, mostrado na figura abaixo, através dos comandos push e pop na pilha da matriz de transformações do OpenGL.



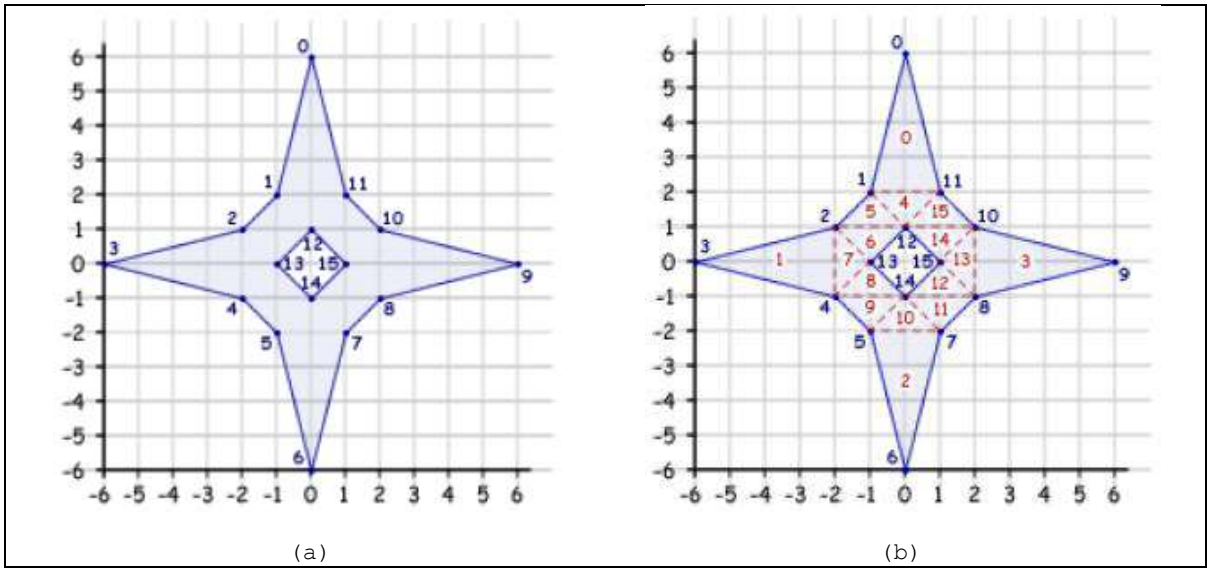
4. Exercícios para Entregar

A solução deverá ser em projeto C++ e OpenGL, e a entrega através do TIDIA.
Importante: em todos os arquivos de código faça um cabeçalho (com comentários) colocando o seu nome completo. Além disso, no título das janelas OpenGL criadas pelo programa coloque o título do programa e o seu nome e sobrenome.

1) Modifique o exemplo do 5º. Projeto: "robot.c" para adicionar alguns dedos no pulso do braço conforme mostrado na figura abaixo. Utilize glPushMatrix() e glPopMatrix() para salvar e restaurar a posição e orientação do sistema de coordenadas do pulso. Nesse caso, necessita-se salvar a matriz corrente antes de posicionar cada dedo e restaurar a matriz corrente após cada dedo ter sido desenhado.



2) Composição de poligono: antes de descrever a função para controlar o display, deve-se projetar o objeto a desenhar. Isto pode ser conseguido com relativa facilidade, desenhando o objeto pretendido numa folha de papel quadriculado. A Figura (a) mostra o objeto escolhido para este exemplo, com os vértices devidamente numerados de 0 a 15:
Como o OpenGL apenas consegue desenhar formas mais simples, é necessário dividir o objeto em triângulos e mandar desenha-los um por um. Esta tarefa pode parecer complicada, mas com alguma organização acaba por se tornar bastante simples. A Figura (b) mostra o objeto dividido em triângulos. Tal como anteriormente, estes foram devidamente numerados de 0 a 15, desta vez a vermelho para se poderem distinguir dos vértices:



Pede-se:

i. Escreva uma função para desenhar este objeto por meio da composição de triângulos, de forma otimizada.
ii. Caso tenha feito o item (i), elabore o programa completo, de forma que o objeto tenha rotação controlado por mouse e mudança de cores através de teclado, baseando-se nos exemplos desta aula.

6. Referências

[1] OpenGL Architecture Review Board, Dave Shreiner, Mason Woo, Jackie Neider, Tom Davis. **OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 2.1**. 6th Ed. Addison-Wesley, 2008.

7. Relatório

Envie o relatório na forma de arquivo eletrônico **HTML**, e os **programas C++**, com cabeçalho (comentário) incluindo seu nome completo, RA, data do programa, nome do programa, e exemplo de chamada do programa no prompt do linux.
Elaborar um relatório contendo:
- Todos os procedimentos detalhados executados no laboratório.
- Respostas das perguntas
- Imagens obtidas. Tabelas de análises comparativas de resultados.
- Análise e conclusões.
- Exercícios para entregar