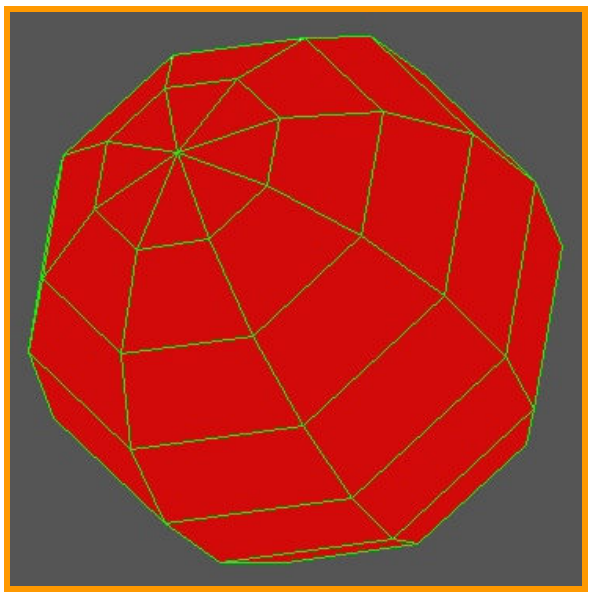
UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ – UNIOESTE
CAMPUS UNIVERSITÁRIO DE CASCAVEL
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Disciplina: Computação Gráfica.
Profº: Adair Santa Catarina.

TRABALHO FINAL DE COMPUTAÇÃO GRÁFICA

Um dos objetos primitivos essenciais da geometria são as esferas. Na computação gráfica as esferas podem ser representadas por meio de uma aproximação poligonal, onde a superfície da esfera é dividida, como o globo terrestre, em meridianos e paralelos.



A esfera acima foi dividida em 8 meridianos e 7 paralelos, ou seja, um meridiano a cada θ = 45º (360º/8) e um paralelo a cada 22,5º (180º/8). Portanto, para modelar uma esfera, deve-se definir um valor para seu raio e um número de meridianos e paralelos. O raio e o número de paralelos serão utilizados para definir um primeiro meridiano, que rotacionado ao redor do eixo vertical, formarão os demais meridianos e paralelos. A figura 1 resume a técnica de modelagem para esferas poliédricas.

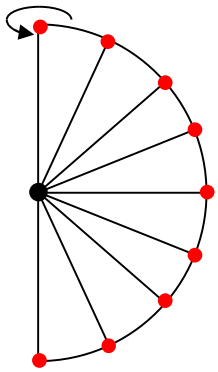


Figura 1 – Modelagem de um meridiano construção poligonal de esferas

Após modelar o primeiro meridiano, como mostrado na figura 1, este deve ser rotacionado ao redor do eixo vertical em θ^o, até completar os 360º de uma circunferência completa. Por fim, os pontos (em vermelho) devem ser corretamente ligados formando as faces que recobrem a superfície da esfera. Note que os pontos nos polos da esfera são únicos e formam, em conjunto com outros pontos, faces triangulares; as demais faces possuirão quatro lados.

Os requisitos para este trabalho são:

1) Modelar uma cena sintética inserindo várias esferas com diferentes raios, números de meridianos e paralelos. Lembre-se que estas propriedades devem ser armazenadas junto às esferas e podem ser alteradas pelo usuário.
2) Durante a execução do aplicativo o usuário poderá limpar a cena e iniciar a construção de uma nova cena, sem ter de reiniciar o software.
3) Os objetos podem ser individualizados (seleção) e sofrerem transformações geométricas de translação, rotação e escala. Como a cena modelada está em 3D, as operações acima podem ser realizadas em todas as direções (X, Y e Z). Note que a escala pode deformar a esfera, caso os fatores Sx, Sy e Sz sejam distintos.
4) Os objetos devem ser apresentados em uma única área de desenho, seja em projeção paralela axonométrica ou em projeção perspectiva. Caberá ao usuário selecionar o tipo de projeção que será empregada no processo de visualização. Para obter tais projeções será necessário informar todos os parâmetros de câmera (VRP, P, vetor view-up, tipo de projeção, distância aos planos Near, Far e plano de projeção); os limites do mundo e os limites do plano de projeção também são parâmetros definidos pelo usuário. É importante que nessa janela de projeção sejam apresentados sinalizadores mostrando a direção dos eixos principais do SRU, como mostrado no círculo verde da figura 2.

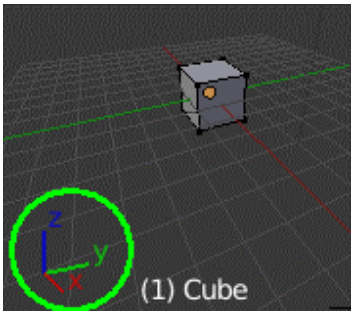


Figura 2 – Eixos principais do SRU exibidos na vista em perspectiva.

5) Os objetos da cena sintética podem ser renderizados em sombreamento constante, sombreamento Gouraud ou sombreamento Phong simplificado. Caberá ao usuário selecionar uma das opções de renderização da cena.
6) O sistema deverá permitir a edição, em tempo de execução, de todos os parâmetros da cena (câmera, tipo de projeção, propriedades das fontes luminosas – ambiente e pontual, dos parâmetros dos materiais aplicados a cada um dos objetos – Ka, Kd, Ks e n). Há cor na cena sintética, o que significa que os parâmetros das fontes luminosas e dos materiais vinculados aos objetos devem ser definidos em tuplas em consonância com o sistema de cores RGB. A criação da

cena, em um primeiro momento, pode ser realizada com parâmetros pré-definidos, mas depois podem ser editados pelo usuário, conforme sua necessidade.

7) O sistema deve ser estável. Ou seja, deve suportar as interações com o usuário sem travar, abortar operações ou ser encerrado devido a erros em tempo de execução.

Data de entrega: 25/07/2022 (iniciem a codificação o mais rapidamente possível, pois os anos letivos estão encurtados devido à recuperação de calendários em função da pandemia).

Avaliação:

Para que o trabalho possa ser corrigido é essencial que os requisitos 1, 4 e 5 (ao menos sombreamento constante) sejam implementados. Trabalhos que não implementem estes 3 requisitos terão nota zero.

Critérios:

Requisito 1) 10 pontos;
Requisito 2) 5 pontos;
Requisito 3) 15 pontos;
Requisito 4) 20 pontos;
Requisito 5) 30 pontos;
Requisito 6) 15 pontos;
Requisito 7) 5 pontos.

O trabalho poderá ser desenvolvido em **grupo com até 3 integrantes**. Trabalho desenvolvido individualmente terá sua nota final multiplicada pelo fator 1,2. Trabalho desenvolvido em dupla terá sua nota final multiplicada pelo fator 1,1.

Os códigos fonte dos trabalhos desenvolvidos devem ser agrupados em um único arquivo .zip que será entregue através da plataforma Teams. Todo e qualquer módulo adicional utilizado no desenvolvimento do software também deverá ser entregue no mesmo arquivo, bem como um tutorial que explique como o programa deve ser "compilado" e/ou executado.

A linguagem/ambiente para desenvolvimento do trabalho é de livre escolha. São linguagens recomendadas:

1) JavaScript + bibliotecas (p5.js) + framework que facilite o desenvolvimento de aplicações com o objeto Canvas do HTML5;
2) Python + interface gráfica (Tkinter, PySimpleGUI, Kivy);
3) Java + (Swing ou JavaFX);
4) Borland C++ Builder (procure-me).

Quaisquer dúvidas que persistirem poderão e deverão ser esclarecidas através de consultas ao professor da disciplina.

Como exemplo de trabalho a ser desenvolvido, baixe a aplicação que está disponível em: https://www.inf.unioeste.br/~adair/CG/Software/CG3DStudio.jar

Maximize a janela da aplicação e a tela de projeção em perspectiva para experenciar um modelador 3D simplificado.