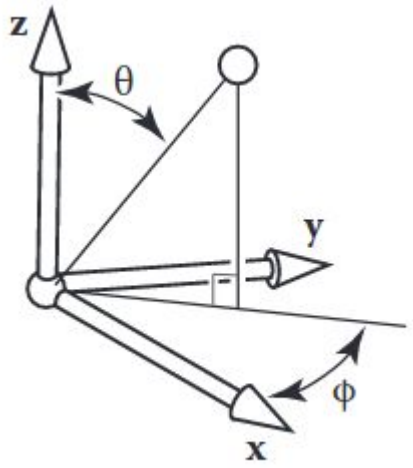
MAC0420 Introdução à Computação Gráfica

Mundo Esférico

por Julio Kenji Ueda 9298281

Ambas a soluções levam em consideração o seguinte modelo de coordenadas:



Solução do Modelo 1

(Geração de Esfera utilizando Coordenadas Polares)

A classe **RenderWidget** foi modificada com a opção Sphere onde o usuário pode escolher entre 6 configurações de esferas com diferentes parâmetros (raio e números de amostras na vertical e horizonal).

A classe **Sphere** (subclasse de **Actor**) foi criada com parâmetros de entrada **r**, **v** e **h** para modelar uma esfera de raio **r** com **v** amostras igualmente espaçadas do ângulo θ e **h** amostras igualmente espaçadas do ângulo ϕ.

- A geração da geometria da esfera é baseado no modelo do **Cone** e ocorre em três etapas:
    - O topo da esfera é modelado com **h** triângulos dispostos em forma circular e renderizado com modo **GL_TRIANGLE_FAN**.
    - O corpo da espera é modelado por **h** quadriláteros dispostos em forma circular na horizontal em **v**-2 na vertical. Cada quadrilátero é formado por 2 triângulos e renderizado com modo **GL_TRIANGLES**
    - A base da esfera é modelado como o topo da esfera.
- Os vetores normais são as posições dos vértices normalizados

- Os parâmetros u e v das coordenadas de textura são calculados como:
  $u = \frac{\varphi}{2\pi},\ v = 1 - \frac{\theta}{\pi}$

Solução do Modelo 2

(Geração de Esfera por Subdivisão de Icosaedro no Tessellation Shader)

A classe **RenderWidget** foi modificada com a opção **Tessellated Sphere** onde o usuário pode escolher entre 6 esferas com diferentes parâmetros (raio e nível de tesselação).

A classe **Sphere** recebe como parâmetros **r** e **s** para gerar uma malha esférica de **raio r** a partir da tesselação do icosaedro em **s níveis** (interno e externo). Algumas observações:

- Os shaders utilizados são: **Vertex**, **Tessellation Control**, **Tessellation Evaluation** e **Fragment**.
- Os valores do raio e o nível de tesselação são passados ao pipeline como valores uniformes.
- A posição final dos vértices, os vetores normais e as coordenadas de textura são calculados no **Tessellation Evaluation Shader**.
- Os parâmetros **u** e **v** das coordenadas de textura são calculados como:
  $u = \frac{\varphi}{2\pi} = \frac{arctg\left(\frac{y}{x}\right)}{2\pi},\ v = 1 - \frac{\theta}{\pi} = 1 - \frac{arc\cos(z)}{\pi}$
- Os vetores normais da esfera são as posições dos vértices normalizados.
- Modo de renderização utiliza **GL_PATCHES** devido ao uso da Tesselação.