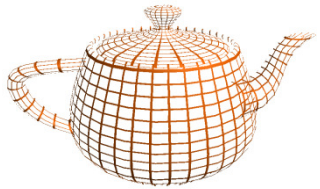
Computação Gráfica

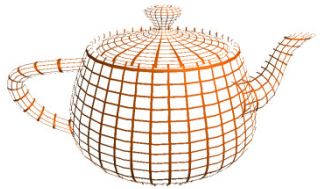
Stereo

António Ramires Fernandes - CG



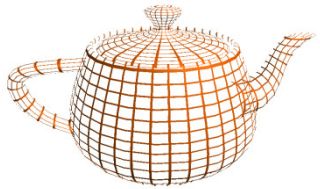
Percepção de Profundidade

- 2D
  - Perspectiva (dimensão varia com a distancia)
  - Dimensão de objectos familiares
  - Detalhe (mais próximo => mais detalhe)
  - Oclusão (um objecto ocluso está mais distante)
  - Movimento Relativo (objectos perto movem-se mais rapidamente)



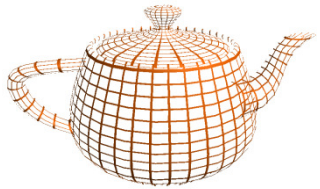
Percepção de Profundidade

- 3D
  - Disparidade binocular (diferença entre as imagens do olho direito e esquerdo)
  - Acomodação (esforço muscular para focar um determinado ponto)
  - Convergência (esforço muscular para rodar o olho para apontar para o ponto focal)



Stereo em OpenGL

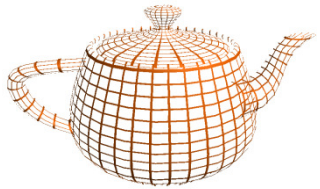
- 4 buffers
  - back left
  - back right
  - front left
  - front right
- HMD (um par por olho)
- Shutter Glasses (alternado)
- Asus e Elsa fornecem soluções quase universais



Stereo - Alternativas

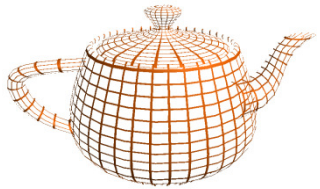
- Oculos Vermelho-Azul ou Vermelho-Verde

- Estratégia:
  - Inibir o desenho nos canais de cor complementares
    - glColorMask



Stereo - Versão Simples

- Base de trabalho:
  - posição da câmara actual - pos
  - vector up
  - vector dir, com a direcção do olhar
- Passos:
  - calcular right
    - r = dir x up
  - posição da câmara esquerda (posl) e direita (posr)
    - posl = pos + r * delta
    - posr = pos - r * delta

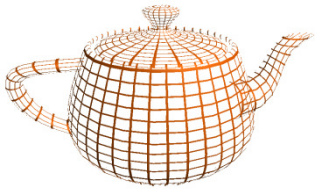


Stereo - Versão Simples

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

glColorMask(GL_TRUE, GL_FALSE, GL_FALSE, GL_TRUE);
// set camera for red eye, blue will be filtered.
...
// draw scene
...
glClear(GL_DEPTH_BUFFER_BIT);
glColorMask(GL_FALSE, GL_FALSE, GL_TRUE, GL_TRUE);
// set camera for blue eye, red will be filtered.

...
// draw scene
...
glColorMask(GL_TRUE, GL_TRUE, GL_TRUE, GL_TRUE);
glutSwapBuffers();
```



Exercício

- Implementar uma aplicação que permita a visualização stereo do cenário das aulas anteriores.