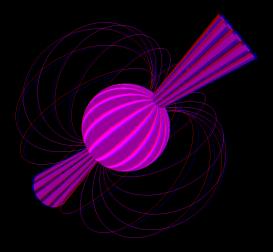


COMPUTAÇÃO GRÁFICA



Stereo



ANTÓNIO RAMIRES FERNANDES — COMPUTAÇÃO GRÁFICA



Percepção de Profundidade

• 2D

- Perspectiva (dimensão varia com a distância)
- 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 (4 buffers, um par de buffers por olho)
- Shutter Glasses (2 buffers, alternado)



Stereo - Alternativas

- Oculos Vermelho-Azul ou Vermelho-Verde
- Estratégia:
 - Inibir o desenho nos canais de cor complementares

glColorMask

- Para cada "olho" posicionar a câmara respectiva e desenhar somente no canal da côr do filtro respectivo.



Stereo - Versão Simples

• Base de trabalho:

- posição da câmara actual: pos
- vector up
- vector dir, com a direcção do olhar

Nota: os vectores devem estar normalizados

Passos:

calcular right

$$r = dir \times up$$

posição da câmara esquerda (pos1) e direita (posr)

$$posr = pos + r \times delta$$

$$posl = pos - r \times delta$$



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.
...
// draw scene
...
glClear(GL_DEPTH_BUFFER_BIT);
glColorMask(GL_FALSE, GL_TRUE, GL_TRUE, GL_TRUE);
// set camera for blue eye.
...
// draw scene
...
glColorMask(GL_TRUE, GL_TRUE, GL_TRUE, GL_TRUE);
glutSwapBuffers();
```

7



Exercício

 Implementar uma aplicação que permita a visualização stereo de um teapot branco num fundo preto, com uma câmara que se pode deslocar no espaço (similar à câmara da aula do terreno)

