

Computação Gráfica e Interfaces

2017-2018
Fernando Birra

Picking

2017-2018

Fernando Birra

Objetivos

- Como identificar os objetos no ecrã?
- Estudo resumido de 3 métodos:
 - *selection buffer*
 - Frame-buffer invisível com pseudo-cor
 - caixas envolventes

Problema

- Dado um ponto no canvas, como mapeamos esse ponto de volta para um objeto?

Dificuldades:

- Mais do que uma resposta (não unicidade)
- A natureza do pipeline (dos objetos para o ecrã)
- É necessário ter em atenção (tolerar) a dificuldade em “apontar” com precisão uma posição usando um dispositivo de input

Selection Buffer

- A cada primitiva (ou grupo) é atribuído um **id** pela aplicação, indicando a que objeto pertence.
- à medida que a imagem é produzida (a representação da cena), os **ids** das primitivas que são varridas perto da posição indicada são colocados numa lista de resultados
- Examina-se a lista de resultados no final
- Suportado em versões antigas do OpenGL (pipeline fixo), mas atualmente deprecado.

Selection Buffer

- Implementa-se criando uma pequena janela que corresponde a uma área envolvendo a posição indicada (pelo rato, possivelmente)
- Pode-se determinar se uma primitiva é ou não gerada nessa região
- Requer um passo adicional de desenho (embora não se veja o resultado). Desenha-se *off-screen* num buffer extra.
- É necessário integrar com a profundidade (visibilidade)

Picking com cores

- Pode usar-se `gl.readPixels` para obter a cor numa dada localização da janela
- A ideia é a de usar a cor para identificar o objeto. Mas:
 - Múltiplos objetos podem ter a mesma cor
 - Um objeto sombreado apresentará variadas cores
- Solução: atribuir uma cor única a cada objeto (serve de id) e desenhar num buffer off-screen (tal como com selection buffer)
 - usar `gl.readPixels` nesse buffer nas coordenadas indicadas
 - Usar uma tabela (dicionário) para passar da cor (id) ao objeto

Picking com Caixas Envolventes

- Ambos os métodos anteriores requerem um passo de *rendering* adicional de cada vez que se pretende fazer *pick*
- Uma alternativa a esse processo é a de se usarem caixas envolventes (alinhadas com os eixos)
- Mapear a localização do ponteiro para o objeto faz-se identificando em primeiro lugar qual a caixa envolvente a que a localização indicada pertence.
- Em caso de múltiplos resultados, deve dar-se prioridade aos objetos mais pequenos

