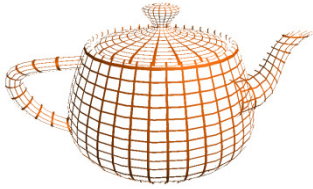




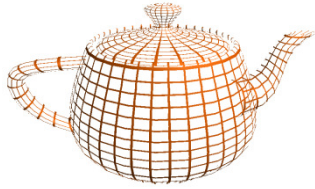
Computação Gráfica

Vertex Buffer Objects



VBOs

- Passo 1 - Criação dos VBOs
 - alocar arrays para vértices, normais e índices.
 - Activar Buffers
 - `glEnableClientState(GL_VERTEX_ARRAY);`
 - `glEnableClientState(GL_NORMAL_ARRAY);`

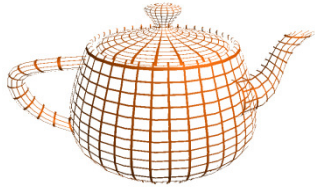


VBOs

• Passo 2: Gerar VBOs

- GLuint buffers[n];
- float *vertexB, *normalB; // arrays de vértices, normais, etc...
- ...
- glGenBuffers(n, buffers);
- glBindBuffer(GL_ARRAY_BUFFER, buffers[0]);
- glBufferData(GL_ARRAY_BUFFER, arraySize, vertexB, GL_STATIC_DRAW);
- glBindBuffer(GL_ARRAY_BUFFER, buffers[1]);
- glBufferData(GL_ARRAY_BUFFER, arraySize, normalB, GL_STATIC_DRAW);

em bytes

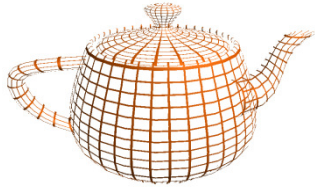


VBOs

- Passo 3a: Desenhar com VBOs
 - Indicar para cada buffer qual a sua utilização

```
glBindBuffer(GL_ARRAY_BUFFER, buffers[0]);  
glVertexPointer(3, GL_FLOAT, 0, 0);
```

```
glBindBuffer(GL_ARRAY_BUFFER, buffers[1]);  
glNormalPointer(GL_FLOAT, 0, 0);
```



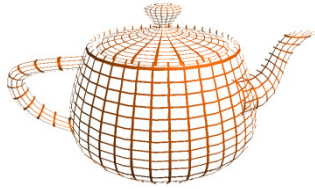
VBOs

- Passo 3b : Desenhar com VBOs
 - Desenho recorrendo a uma lista de índices

```
GLuint *pointer;
```

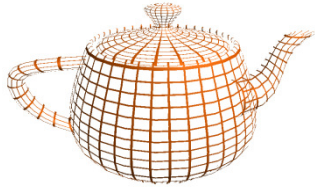
```
...
```

```
glDrawElements(GL_TRIANGLE_STRIP, count, GL_UNSIGNED_INT, pointer);
```



Exercício

- Defina Vertex Buffer Objects para o terreno do cenário virtual
- Na inicialização fazer:
 - Crie os arrays com a dimensão adequada para o terreno em causa
 - cada vértice e cada normal necessitam de 3 floats
 - existem tantos vértices quantos pixels da imagem
 - Preencha os arrays de vértices e normais com os valores que já usava para desenhar o terreno anteriormente
 - Crie um array de índices apropriado para strips
 - Active os VBOs
 - Crie os VBOs



Exercício

- Na fase de render fazer:
 - preparar os VBOs
 - Desenhar com `glDrawElements`
 - uma invocação para cada strip