

# OpenGL (2º Parte)

---

Professora: Mercedes Gonzales Márquez

# Listas

---

- Em OpenGL os dados podem ser conservados em uma lista da exposição para uso corrente ou serem usados mais tarde. (A alternativa além de manter dados em uma lista de exposição é processar os dados imediatamente – também conhecido como modo imediato.)
- Quando uma lista da exposição é executada, os dados retidos são enviados da lista apenas como se fossem enviados pela aplicação no modo imediato.

# OpenGL - Listas

---

- As listas de exibição aperfeiçoam o desempenho de aplicações OpenGL. Porém uma vez que uma lista de exibição é criada, não pode ser modificada.
- O modo no qual são otimizados os comandos em uma lista de exibição pode variar de implementação para implementação. Por exemplo, um comando tão simples quanto `glRotate * ()` poderia mostrar uma significativa melhoria de performance se estivesse em uma lista de exibição, uma vez que os cálculos para produzir a matriz de rotação não são triviais.
- As listas de exibição são indicadas para: Operações complexas com matrizes, luzes, propriedades de materiais e modelos de iluminação complexos, texturas.

# OpenGL - Listas

---

- `GLuint glGenLists(GLsizei range);`

Este comando aloca um conjunto de números contíguos, previamente não alocados por um índice da lista de exibição. O valor inteiro retornado é o índice que marca o início de um bloco contíguo de um display de visualização. Os índices retornados são marcados como vazios e usados então por chamadas subsequentes do comando `glGenLists()`.

# OpenGL - Listas

---

- `void glNewList (GLuint lista, GLenum modo);`

Especifica o início de uma lista de exibição. As rotinas OpenGL chamadas subsequentemente ( enquanto o comando `glEndList()` não for executado) são armazenados na lista de visualização, exceto uma lista restrita de rotinas OpenGL que não podem ser armazenados na lista . O parâmetro `lista` é um inteiro positivo maior que zero, que identifica unicamente a lista de visualização . Os possíveis valores para `modo` são `GL_COMPILE` e `GL_COMPILE_AND_EXECUTE`. O parâmetro `GL_COMPILE` deve ser usado se deseja-se que os comandos da lista de exibição não sejam executados imediatamente após serem colocados na lista, neste caso deve ser utilizado o parâmetro `GL_COMPILE_AND_EXECUTE`.

# OpenGL - Listas

---

- `void glEndList (void);`

Marca o final de uma lista de exibição. Após a lista ter sido criada a mesma poderá ser executada através do comando `glCallList()`.

- `void glCallList (GLuint list);`

Esta rotina executa a lista de exibição especificada pelo parâmetro. Os comandos na lista de execução são então executados na ordem que foram salvos.

# OpenGL - Listas

---

- `void glEndList (void);`

Marca o final de uma lista de exibição. Após a lista ter sido criada a mesma poderá ser executada através do comando `glCallList()`.

- `void glCallList (GLuint list);`

Esta rotina executa a lista de exibição especificada pelo parâmetro. Os comandos na lista de execução são então executados na ordem que foram salvos.

Veja Experimento 3.6.

Exercício : Faça um anel de círculos concêntricos de múltiplas cores no plano xy usando listas. Use o fator de escala  $u$ , assim `glScale (u,u,1.0);`

# OpenGL - Fontes

---

- Veja o experimento 3.8.



# OpenGL - Mouse

---

- Botões do mouse podem ser programados para responder a cliques e ao movimento do mesmo.
- Veja o experimento 3.9 e 3.10.
- Veja um programa em C na página da disciplina.
- Exercício: Escreva um programa para desenhar um círculo depois de dois cliques do mouse. O primeiro clique será para o ponto central e o segundo clique para definirmos o raio.
- Exercício: Melhore o programa anterior para permitir que usuário veja o círculo mudando à medida que ele arrasta o segundo ponto.