

Transformações Geométricas e Animação

SCC0250 - Computação Gráfica

Prof. Rosane Minghim

[https://sites.google.com/site/computacaograficaicmc2017t2/
rminghim@icmc.usp.br](https://sites.google.com/site/computacaograficaicmc2017t2/rminghim@icmc.usp.br)

P.A.E. Nicolas Roque nrsantos@usp.br

Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)

23 de março de 2017



Transformações Geométricas

- Inicialmente deve-se transladar o objeto para a origem.
- Aplica-se a transformação.
- Por fim, translada-se o objeto para sua posição inicial.
- **Na OpenGL a sequência das transformações segue a ordem invertida.**
 - Exemplo: Suponha que deseja-se rotacionar e depois modificar a escala de um objeto. Na OpenGL a ordem seria: escala e depois a rotação.

Matriz de Transformação

- Todos os comandos de transformação são compostos em uma matriz de transformação.
- Cada novo comando é acumulado, alterando a configuração da matriz.
- Ao especificar um novo vértice, a sua posição é calculada aplicando-se a matriz de transformação corrente às suas coordenadas.
- A matriz de transformação é inicializada com a matriz identidade.
 - Na OpenGL isso é feito pela função `glLoadIdentity();`

Transformações em OpenGL

- Translação
 - `glTranslatef(x, y, z)`
 - `x`, `y` e `z` representam a quantidade a ser translada nos respectivos eixos.
 - Parâmetros são do tipo `GLfloat`.
 - Em 2D: $z = 0$.

Transformações em OpenGL

- Rotação

- `glRotatef(angulo, x, y, z)`
- Parâmetro *ângulo*: especifica, em graus, o ângulo de rotação.
- x, y e z: indicam o eixo a ser realizada a rotação.
 - Exemplo: Rotação de 30° em torno do eixo y seria `glRotatef(30.0f, 0.0f, 1.0f, 0.0f)`
- Parâmetros são do tipo `GLfloat`.
- Em 2D: $x = 0$, $y = 0$ e $z = 1$.

Transformações em OpenGL

- Escala
 - `glScalef(x, y, z)`
 - `x`, `y` e `z` representam o fator de escala nos respectivos eixos.
 - Exemplo: Escala de 2 nos eixos `x` e `y` seria `glScalef(2.0f, 2.0f, 0.0f)`
 - Parâmetros são do tipo `GLfloat`.
 - Em 2D: `z = 0`.

Transformações Hierárquicas

- São transformações diferentes aplicadas em objetos que seguem uma hierarquia.
- Por exemplo, podemos aplicar diferentes transformações sobre partes do corpo humano.
 - Rotacionar o corpo, transladar o braço, entre outras.
- As transformações de mais baixa hierarquia (braços) acumulam as transformações da hierarquia mais alta (corpo).

Transformações Hierárquicas

- É possível salvar a matriz de transformação corrente em uma pilha para aplicar transformações hierárquicas em diversos objetos.
- Cada objeto transformado utiliza sua própria versão da matriz de transformação, que pode ser empilhada ou desempilhada conforme a necessidade.

Transformações Hierárquicas

- `glPushMatrix()`
 - Empilha a matriz de transformação atual.
 - Não altera a matriz atual.
- `glPopMatrix()`
 - Retira a matriz do topo da pilha.
 - A matriz atual é substituída pela matriz desempilhada.

Animação

- Animação tradicional envolve uma sequência de imagens em alta velocidade.
- Velocidade de exibição (frame rate) varia de acordo a mídia utilizada.
- Importante garantir que a aparência da imagem não mude muito rapidamente, pois pode gerar temporal aliasing.
 - Exemplo: <http://www.nvidia.com.br/object/txaa-anti-aliasing-technology-br.html>

Animação - OpenGL

- Redesenhar a cena continuamente.
- Cada quadro deve possuir uma ligeira diferença em relação aos outros, criando a ilusão de movimento.
 - Movimento dos objetos.
 - Cor.
 - Forma.
- Em 3D, além das mudanças acima, pode-se mudar a posição do observador.

Animação - OpenGL

- É necessário forçar uma atualização contínua da tela.
- Deve-se evitar que a imagem fique "piscando" com o contínuo redesenho.
 - Utiliza-se 2 buffers de exibição.
 - `glutInitDisplayMode(GLUT_DOUBLE | ...)`
 - Um dos buffers será preenchido enquanto o outro é exibido.
 - Troca de buffers é feita por `glutSwapBuffers()`

Animação - OpenGL

- A GLUT oferece uma função callback que é invocada em um tempo predeterminado.
- `glutTimerFunc(int msec, void* timer(int value), int value)`
 - `int msec`: Tempo em milisegundos
 - `void* timer(int value)`: Função de animação
 - `int value`: Valor passado para a função timer.
 - Necessita ser invocada várias vezes para não parar (recursão).

Bibliografia

- **Básica:**

- Hearn, D. Baker, M. P. Computer Graphics with OpenGL, Prentice Hall, 2004. **(livro texto)**
- Angel, E. Interactive computer graphics: a top-down approach with OpenGL, Addison Wesley, 2000.
- Foley, J. et. al - Introduction to Computer Graphics, Addison-Wesley, 1993.
- Kessenich, J., Sellers, G., Shreiner, D. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V - Ninth Edition.

Bibliografia

- **Complementar:**

- Computer Graphics Comes of Age: An Interview with Andries van Dam. CACM, vol. 27, no. 7. 1982
- The RenderMan – And the Oscar Goes to... IEEE Spectrum, vol. 38, no. 4, abril de 2001.