

Primitivas de desenho CSG

**Alunos: Caio Silva Martins de Oliveira,
José Eliton de Albuquerque Filho e
Victor Hugo de Oliveira da Silva**

Instituição: Instituto Militar de Engenharia - IME

Orientador: Thomas Lewiner

Introdução

A Geometria Sólida Construtiva (CSG – Constructive Solid Geometry, em inglês) é a geometria de construção de sólidos a partir de primitivas e operações booleanas. As operações booleanas básicas são união, diferença e intersecção. A partir dessas operações e de primitivas, como cubo, esfera, entre outros sólidos, podemos construir sólidos complexos. É importante ressaltar que a ordem da aplicação dessas operações interfere no resultado final do sólido. A partir de primitivas construídas implicitamente usando OpenGL, aplicam-se as operações booleanas e obtêm-se combinações de figuras.

Objetivos

Estudar uma metodologia para a combinação de figuras através de uma árvore de operações, ou seja, determinação da figura final através de primitivas iniciais (figuras iniciais, como círculo, quadrado) e de operações booleanas. Desenvolvimento de uma linha de código em C++.

Metodologia

Segundo Azevedo (2003), em muitas situações, as expressões matemáticas implícitas dos sólidos modelados podem ser armazenadas e usadas para a modelagem. Podendo, mesmo depois, serem combinadas com outras formas de representação como, por exemplo, para gerar novos objetos por combinações, por união de objetos [...] Essa forma tira proveito do fato de que muitas formas Euclidianas têm equações perfeitamente conhecidas e simples. Assim, se os sólidos forem esferas, elipsóides, cilindros elípticos e diversos outros, usar as equações implícitas dessas formas é uma maneira simples de armazená-los, operá-los e mesmo renderizá-los eficientemente.

Como já foi dito, as três operações básicas na Geometria Sólida Construtiva (CSG) são a união, a intersecção e a diferença. Dados dois sólidos A e B representados respectivamente por duas funções implícitas f e g, com a seguinte convenção: $f(x,y,z)=0$ e $g(x,y,z)=f$ formam o sólido, $f(x,y,z)<0$ e $g(x,y,z)<0$ dentro do sólido e $f(x,y,z)>0$ e $g(x,y,z)>0$ fora do sólido.

Assim, a união dos sólidos A e B será dada por:

$$A \cup B = \min(f(x, y, z), g(x, y, z))$$

A intersecção entre A e B será dada por:

$$A \cap B = \max(f(x, y, z), g(x, y, z))$$

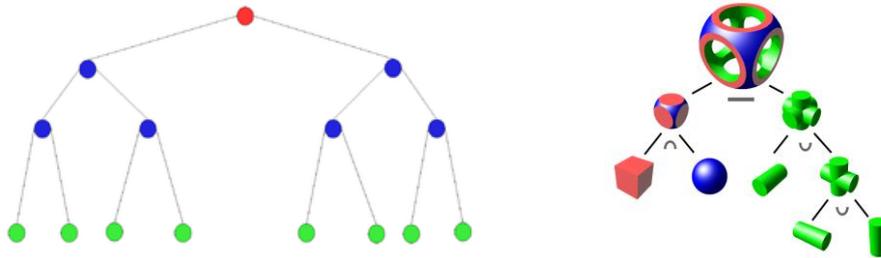
A diferença entre A e B será dada por:

$$A - B = \max(f(x, y, z), -g(x, y, z))$$

Assim, a partir de primitivas, dadas suas funções implícitas, podemos usar a CSG para construir objetos complexos a partir dessas operações básicas.

O desenvolvimento da linha de código para tal foi feito utilizando linguagem de programação C++. Com base nessas primitivas foram utilizadas árvores binárias de operações para o armazenamento de informações no programa. Cada nó de uma árvore contém um componente, que pode ser uma operação booleana, uma primitiva ou pode estar vazio.

A árvore é armazenada no programa através de um vetor ordenado do topo (raiz da árvore) para a base. Por exemplo, se queremos fazer a operação $(A \cap B) \cup C$ o vetor seria $|\cup|\cap|C|A|B|VAZIO|VAZIO$. Na linha de código, temos uma função que analisa o vetor termo a termo e, analisando o elemento n em relação ao $2n+1$ e ao elemento $2n+2$ (os elementos logo abaixo de n na árvore), aplica as operações de união, intersecção e diferença entre as primitivas já armazenadas nesse vetor. Dessa forma, conseguimos obter diferentes combinações de operações e figuras diferentes.



Figuras 1 e 2– Exemplo de árvore binária e sua aplicação no programa

Há também uma função responsável pela composição de árvores, em que dadas duas árvores a função compõe as duas pela adição de vetores, que basicamente adiciona os elementos da segunda árvore no final do vetor da primeira árvore.

Conclusões

O estudo inicial de funções implícitas permitiu um maior entendimento da combinação de primitivas, já que a combinação destas foi definida por funções implícitas. Isso permitiu a aplicação das operações booleanas.

A linha de código desenvolvida poderá ser utilizada, por exemplo, como base para a criação de um aplicativo de desenho de figuras para iPhone.

Referências

AZEVEDO, Eduardo; CONCI, Aura. **Computação Gráfica: Geração de Imagens**. 7.ed. Rio de Janeiro: Campus, 2003.

HIJAZI, Younis ; et al.

CSG Operations of Arbitrary Primitives with Interval

Arithmetic and Real-Time Ray Tracing. Utah: SCI Institute, University of Utah, 2008.