

Trabalho III

1 Especificação do Problema

Em computação gráfica, superfícies são normalmente representadas por meio de um conjunto de polígonos, denominado malhas poligonais. Devido a sua simplicidade, triângulos são primitivas gráficas comuns utilizadas em malhas poligonais.

O objetivo deste trabalho é aplicar textura a objetos representados por malhas triangulares. O programa deve utilizar as bibliotecas OpenGL e GLUT, as quais deverão ser incorporadas em um programa escrito em linguagem de programação C, C++ ou Java.

Um menu deve permitir a seleção dos possíveis padrões de textura. Alguns exemplos de mapeamento de textura para o objeto chaleira (*teapot*) são ilustrados na figura 1.

Seu programa deve inicialmente carregar um modelo no formato (simplificado) Wavefront OBJ, o qual contém informações sobre os vértices e as faces das malhas triangulares. Uma estrutura de dados deve ser criada para armazenar a malha, mantendo-se a sua geometria e topologia.

A taxa de renderização dos objetos, dada em quadros por segundo (*fps*, *frames per second*), pode ser medida por meio do seguinte trecho de código:

```
int start, end;

start = glutGet(GLUT_ELAPSED_TIME);
...
end = glutGet(GLUT_ELAPSED_TIME);

printf("Taxa: %.3g fps (%d ms)\n", 1000.0/(end-start), end-start);
```

2 Formato OBJ

O formato OBJ foi desenvolvido pela Wavefront para especificar objetos representados por malhas poligonais. O formato apresenta inicialmente uma lista de vértices seguida de uma lista de faces.

O caracter *v* especifica um vértice, enquanto o caracter *f* especifica uma face. Um exemplo de representação de um cubo é mostrado a seguir. Como todas as faces são triangulares, elas são formadas pelas coordenadas de três vértices. A face 1, por exemplo, é formada pelos vértices 1, 2 e 3 da lista de vértices.

```
v -0.5 0.5 0.5
v -0.5 -0.5 0.5
v 0.5 -0.5 0.5
v 0.5 0.5 0.5
v -0.5 0.5 -0.5
v -0.5 -0.5 -0.5
v 0.5 -0.5 -0.5
v 0.5 0.5 -0.5
f 1 2 3
f 8 7 6
f 4 3 7
f 5 1 4
```



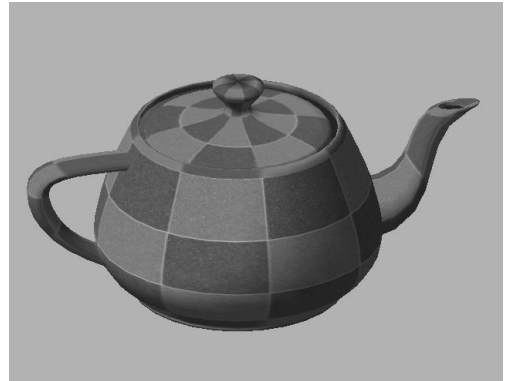
(a)



(b)



(c)



(d)



(e)



(f)

Figura 1: Exemplos de mapeamento de textura em objeto.

```
f 5 6 2
f 2 6 7
f 1 3 4
f 8 6 5
f 4 7 8
f 5 4 8
f 5 2 1
f 2 7 3
```

3 Exemplos de Modelos

Alguns modelos no formato OBJ estão disponíveis em:

http://www.ic.unicamp.br/~helio/modelos_obj/

4 Padrões de Textura

Alguns padrões de textura no formato RGB estão disponíveis em:

http://www.ic.unicamp.br/~helio/padroes_textura/

5 Documentação das Bibliotecas

Alguns manuais das bibliotecas OpenGL e GLUT podem ser encontrados nos endereços:

<http://www.opengl.org>

<http://www.glprogramming.com/red>

<http://www.sgi.com/products/software/opengl>

<http://www.lighthouse3d.com/opengl/glut/>

6 Exemplos de Códigos OpenGL

Alguns códigos escritos em OpenGL estão disponíveis em:

http://www.ic.unicamp.br/~helio/codigos_opengl/

7 Especificação da Entrega

- A entrega do trabalho deve conter os seguintes itens:

- código fonte: o arquivo final deve estar no formato *zip* ou no formato *tar* compactado com *gzip*, contendo todos os programas necessários para sua execução. Favor incluir um Makefile para facilitar a compilação dos programas.
- relatório impresso: deve conter uma descrição dos algoritmos e estruturas de dados, considerações adotadas na solução do problema, testes executados, eventuais limitações ou situações especiais não tratadas pelo programa.

- Data de entrega: 22 de outubro de 2009

8 Observações Gerais

- Os programas serão compilados com em ambiente Linux. Os formatos de entrada e saída dos dados devem ser rigorosamente respeitados pelo programa, conforme definidos anteriormente. Trabalhos entregues com atraso terão 10% da nota descontada por dia de atraso. Não serão aceitos trabalhos após 5 dias da data de entrega.
- Os seguintes aspectos serão considerados na avaliação: funcionamento da implementação, clareza do código, qualidade do relatório técnico.