

**Data de entrega: 10 de jun.**

Projetos de Implementação:

**1. Superfície B-Spline Racional (NURBS) (Diego):**

O objetivo é a conversão de uma superfície para malha de triângulos que aproximam a superfície. A entrada inclui um arquivo-texto com os seguintes parâmetros:

$$n_u n_v k_u k_v k_u^r k_v^r$$

$$u_1 r_{u_1}$$

$$u_2 r_{u_2}$$

$$\vdots$$

$$u_{k_u} r_{u_{k_u}}$$

$$v_1 r_{v_1}$$

$$v_2 r_{v_2}$$

$$\vdots$$

$$u_{k_v} r_{v_{k_v}}$$

$$x_{00} y_{00} z_{00} w_{00}$$

$$x_{01} y_{01} z_{01} w_{01}$$

$$\vdots \quad \vdots$$

$$x_{0(c_v-1)} y_{0(c_v-1)} z_{0(c_v-1)} w_{0(c_v-1)}$$

$$x_{10} y_{10} z_{10} w_{10}$$

$$\vdots \quad \vdots$$

$$\vdots \quad \vdots$$

$$x_{(c_u-1)(c_v-1)} y_{(c_u-1)(c_v-1)} z_{(c_u-1)(c_v-1)} w_{(c_u-1)(c_v-1)}$$

Onde:

$n_u$  é o grau da superfície na direção  $u$ ;

$n_v$  é o grau da superfície na direção  $v$ ;

$k_u$  é o número de nós sem repetições na direção  $u$ ;

$k_v$  é o número de nós sem repetições na direção  $v$ ;

$k_u^r$  é o número de nós incluindo repetições na direção  $u$ ;

$k_v^r$  é o número de nós incluindo repetições na direção  $v$ ;

$u_i$  é o  $i$ -ésimo nó na direção  $u$ ;

$v_i$  é o  $i$ -ésimo nó na direção  $v$ ;

$r_{u_i}$  é o número de repetições do nó  $u_i$ ;

$r_{v_i}$  é o número de repetições do nó  $v_i$ ;

$x_{ij} y_{ij} z_{ij}$  são as coordenadas mundiais do ponto de controle de índices  $ij$ ;

$w_{ij}$  é o peso associado ao ponto de controle de índices  $ij$ ;

$c_u$  é o número de pontos de controle na direção  $u$ ;

$c_v$  é o número de pontos de controle na direção  $v$ .

Note que o total de pontos de controle é  $c_u * c_v$ , onde  $c_u = k_u^r - 1 - n_u$  e  $c_v = k_v^r - 1 - n_v$ .

Note também que  $\sum_{i=1}^{k_u} r_{u_i} = k_u^r$  e  $\sum_{i=1}^{k_v} r_{v_i} = k_v^r$ . Lembre-se que os intervalos de domínio

(intervalos úteis) começam em  $u_{n_u-1}$  e terminam em  $u_{k_u^r-n_u}$  na direção  $u$ , e começam em  $v_{n_v-1}$  e terminam em  $v_{k_v^r-n_v}$  na direção  $v$ . A entrada também deverá incluir os parâmetros de câmera (por arquivo ou na interface):

$C = (x_c, y_c, z_c)$ : o foco da câmera (ponto);

$N = (x_N, y_N, z_N)$ : a mira da câmera (vetor);

$V = (x_V, y_V, z_V)$ : o vetor que aponta para cima ("up vector");

$d$ : distância que o plano de projeção se encontra de  $C$ ;

$h_x h_y$ : tamanho da área útil de visualização dentro do plano de projeção.

A entrada também inclui a granularidade da malha de triângulos em cada direção (número de pontos avaliados na superfície em cada direção). A saída é uma visualização dos triângulos com respeito à câmera escolhida (na forma de um aramado), e um arquivo com a triangulação da superfície. A triangulação deve ter por base as curvas isoparamétricas. O arquivo deve ter o formato:

$p_u \ p_v$

$x_{00} y_{00} z_{00}$

$x_{01} y_{01} z_{01}$

$\vdots \quad \vdots$

$x_{0(p_v-1)} y_{0(p_v-1)} z_{0(p_v-1)}$

$x_{10} y_{10} z_{10}$

$\vdots \quad \vdots$

$\vdots \quad \vdots$

$$X_{(p_u-1)(p_v-1)} Y_{(p_u-1)(p_v-1)} Z_{(p_u-1)(p_v-1)}$$

Onde:

$p_u$  é o número de pontos avaliados na superfície na direção  $u$ ;

$p_v$  é o número de pontos avaliados na superfície na direção  $v$ ;

$x_{ij} y_{ij} z_{ij}$  são as coordenadas mundiais do ponto da superfície de índices  $ij$ ;

Note que o número total de triângulos da malha será  $2 * (p_u - 1) * (p_v - 1)$ .

## **2. Visualizador por varredura de triângulos (Marcos):**

O objetivo do projeto é receber uma lista de triângulos a partir de um arquivo-texto como descrito no item anterior, juntamente com os parâmetros de câmera também descritos no item anterior e visualizar os triângulos projetados com os interiores pintados segundo o algoritmo de varredura de triângulos, com coerência geométrica. A cor de cada triângulo é cinza, cujo tom depende da altura do baricentro: quanto maior, mais claro. Para cada pixel no interior, determinado pela varredura, deve-se calcular as coordenadas baricêntricas do pixel com relação aos vértices projetados do triângulo e multiplicar os coeficientes pelos respectivos vértices originais não projetados (3D, em coordenadas de vista). Assim toma-se o  $z$  do ponto 3D encontrado para se utilizar no algoritmo  $z$ -buffer, i.e., para só pintar o pixel caso o  $z$  correspondente do  $z$ -buffer seja maior (ou seja, se o ponto que foi projetado naquele mesmo pixel (numa passada de um triângulo anterior) estiver mais distante).

## **3. Iluminador com aplicação de textura (Rodrigo):**

O objetivo do projeto é receber: (a). Um arquivo de imagem que é a Textura (T); (b). Um arquivo do tipo RGB contendo o Normal Map (NM); (c). Um arquivo do tipo RGB contendo o Light Map (LM); e (d). Um arquivo do tipo RGB contendo o Eye Map (EM). A textura é a única que pode ter dimensões distintas dos outros arquivos. Nos arquivos de normal map, light map e eye map, vale a seguinte correspondência:  $R \rightarrow x, G \rightarrow y, B \rightarrow z$  e, em termos de valores, cada componente é de um byte, onde:  $0 \rightarrow (-1)$ ,  $128 \rightarrow 0$  e  $255 \rightarrow 1$  (vetores unitários). Este projeto, pensado na integração com os itens anteriores, deverá percorrer cada pixel de uma janela e, se a janela tiver dimensões  $W \times H$  e o pixel atual tiver índices  $(i, j)$

então a sua posição relativa será  $(\frac{i}{W}, \frac{j}{H})$ . Com essa posição relativa pode-se

acessar o pixel correspondente nos arquivos de mapping, ao se multiplicar pelas dimensões desses arquivos e arredondar-se para os inteiros mais próximos, dando a linha e a coluna do pixel:  $(I, J)$ . Para calcular a cor do pixel, utiliza-se a equação de Phong, exibida abaixo. Para isso, o usuário deve entrar com: atributos do objeto ( $k_a, k_d$  e  $k_s$ , pontos flutuantes entre 0 e 1,  $k_s$  ponto flutuante positivo), atributos da cena ( $I_a, I_L$ , triplas de inteiros entre 0 e 255). A Equação de Phong para este caso é:

$$C[I, J] = I_a k_a + I_L * \{T[I, J] \cdot k_d \cdot \langle NM[I, J], LM[I, J] \rangle\} + I_L \cdot k_s \cdot \langle R[I, J], EM[I, J] \rangle^\eta$$

Onde  $C[I, J]$  é a cor com a qual o pixel  $(I, J)$  deve ser pintado e:

$R[I, J] = 2 \cdot \langle LM[I, J], NM[I, J] \rangle \cdot NM[I, J] - LM[I, J]$ , o vetor de reflexão especular. Se  $\langle NM[I, J], EM[I, J] \rangle < 0$ , então  $NM[I, J] \leftarrow -NM[I, J]$ . Se, após isso, tivermos  $\langle NM[I, J], LM[I, J] \rangle < 0$ , então as componentes difusa e especular são anuladas. Se  $\langle NM[I, J], LM[I, J] \rangle > 0$  e tivermos  $\langle R[I, J], EM[I, J] \rangle < 0$ , então somente a componente especular é anulada.