Universidad de Chile
fcfm - dcc

Control 2

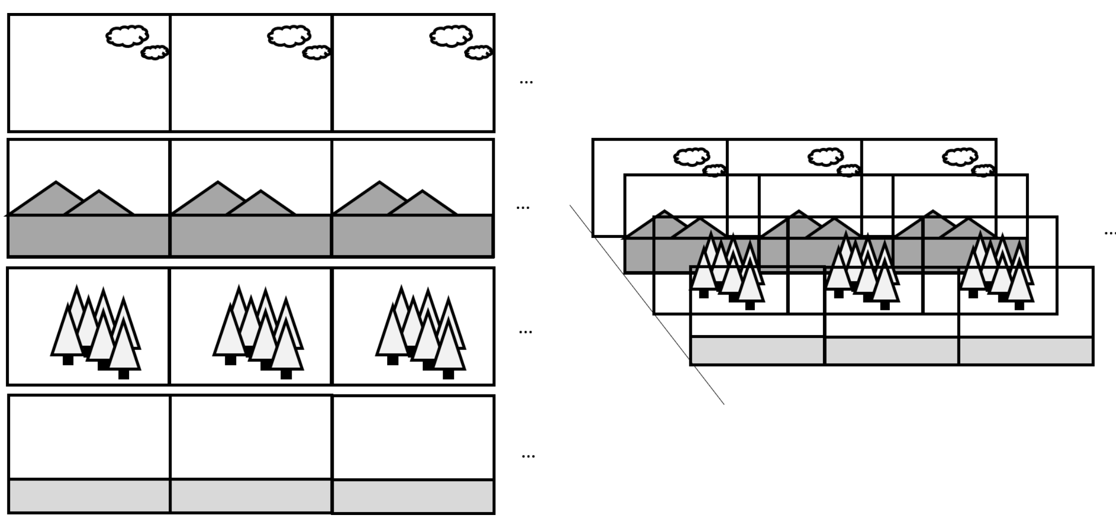
CC3501
11 de octubre de 2019

Instrucciones Generales

- Responda cada problema en hojas separadas.
- Escriba claramente su nombre y firma en cada hoja.
- Responda tan breve como sea posible.
- La evaluación termina a las 14:00 horas, no habrá tiempo adicional.
- No hay preguntas, utilice su mejor juicio para responder.
- Entre paréntesis se indica la puntuación asignada a cada problema. El control posee 6 puntos en total.

Problema 1

1. (1.5) Ahora que conocemos el pipeline de rendering en 3D y sabemos manejar texturas, implementaremos una escena 3D donde se aprecie el efecto de *parallax*, pero solo moviendo la cámara.



Considere que:

- Tenemos texturas que pueden ser repetidas para simular un fondo inmensamente grande (usamos el modo *GL_REPEAT*). Las texturas representan: nubes, montañas, bosque y un camino. Las texturas de montañas, de bosque y de camino poseen transparencias, permitiendo que se vean las texturas que se encuentran atrás de ellas.

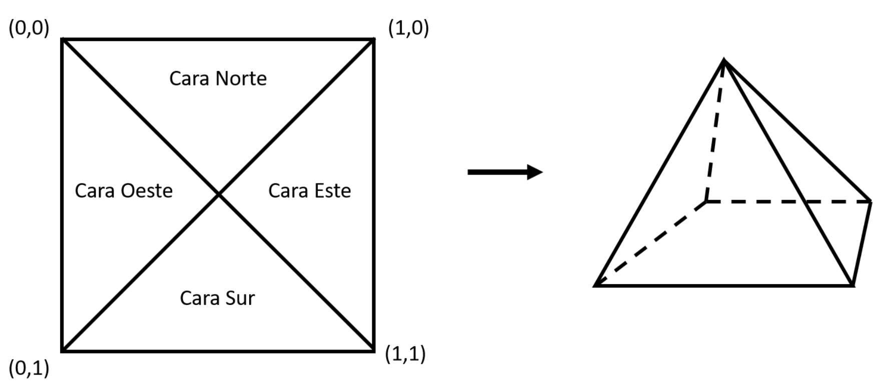
- Ubicaremos 4 rectángulos texturizados en distintos planos de un mismo eje. Cada rectángulo tiene asociada una de las texturas mencionadas.
- Su escena debe ocupar toda la ventana y permitir visualizar todas las texturas en movimiento.

Usted dispone de las siguientes funciones:

- *rectángulo(textura, veces, posición, normal, tamaño)*: dibuja un rectángulo del tamaño dado, centrado en la posición especificada y con una textura repetida varias veces horizontalmente como se indica en la figura. La orientación es especificada por la normal, esto es, el rectángulo mira en la dirección de la normal.
- $lookAt(eye, up, at)$: le permite configurar la cámara.
- *proyección(M)*: Configura M como la matriz de proyección a utilizar (consultar formulario). Debe especificar cada parámetro que se requiera para el tipo de proyección escogido.

Escriba un seudo código que modele la escena estática descrita y que permita mover la cámara hacia la derecha y hacia la izquierda al presionar las fechas del teclado.

2. (0.5) A la escena anterior queremos agregar un modelo de pirámide texturizada en 3D. Para esto, debe construir un archivo OBJ que asigne la textura siguiente a una pirámide que se encuentre inmersa en un bounding box cúbico de arista 1. No se preocupe de especificar las normales.



Problema 2

1. (0.5) Explique de qué manera podemos calcular el volumen de la pirámide de la pregunta anterior utilizando la técnica de *ray casting*.
2. (0.5) (a) Explique brevemente en qué consiste el modelo Flat, Gourad y Phong, y cuál es la principal ventaja y desventaja de usarlo.

3. (1.0) La cara norte de la pirámide de la Pregunta 1, tiene coordenadas P(0,0,0), Q(1,0,0) y R(0.5,0.5,1)), vectores normales en los vértices (0,-1,0), (1,0,0) y (0,0,1), respectivamente, coeficiente de reflexión ambiental (0.2,0.2,0.2), coeficiente de reflexión difusa (0,1,0), y coeficiente de reflexion especular (1,1,1). La escena cuenta con una fuente de luz ubicada en (0.5,-20, 0.5) que emite luz ambiental (0.1,0,1,0.1), difusa (1,1,1) y especular (1,1,1), y un observador que se encuentra en la posición (0.8,-10, 0.5). Cuál es el color para el punto S(0.5,0.25,0.5) usando el modelo de shading Flat, el modelo Gouraud y Phong? (Dejar la expresión)

Problema 3: Fractales

1. (0.5) Define una curva fractal, distinta a la de la lectura. Define el factor de escala y la característica que se repite. Dibuja la curva resultante al aplicarlo dos veces sobre un segmento de largo 1.

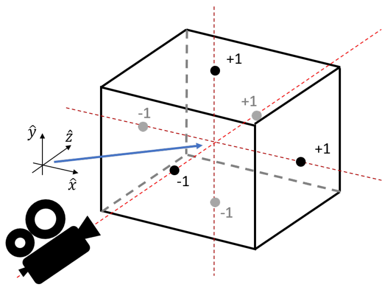
2. (1.0) Queremos dibujar un bosque en un plano. Cada árbol debe ser *único* modelado según la técnica descrita en la lectura. Para ello, implementaremos una función recursiva que en cada iteración añade dos ramitas a un trozo de tronco del árbol. Usted dispone de las siguientes funciones:

   - *línea(a,b;s)*: dibuja una línea de grosor s que conecta los puntos a y b.
   - $hoja(a; s)$: dibuja una hoja en la posición a de un tamaño s.
   - $random()$: genera un número aleatorio entre 0 y 1.

   Su misión es implementar en pseudo-código la función *árbol(x;n)* que generará un árbol en la posición x. Este árbol debe ser distinto en cada llamada y debe lograrlo iterando recursivamente n veces.

3. (0.5) En resumen: Cuáles son las dos características fundamentales de la geometría fractal? Por qué las curvas fractales no son una curva común? Qué se entiende por dimensión fractal y qué representa este número?

Formulario Control 2



Vertex	
Point	position
FaceRef	face

Face	
VertexRef	vertex[3]
FaceRef	neighbor[3]

Vertex	
Point	position
EdgeRef	edge

Face	
EdgeRef	edge

Edge	
VertexRef	vertex[2]
FaceRef	face[2]
EdgeRef	next[2]
EdgeRef	prev[2]

Vertex	
Point	position
HalfedgeRef	halfedge

Face	
HalfedgeRef	halfedge

Halfedge	
VertexRef	vertex
FaceRef	face
HalfedgeRef	next
HalfedgeRef	prev
HalfedgeRef	opposite

$$\mathcal{I} = \mathcal{K}_a\mathcal{L}_a + \frac{1}{k_c + k_l d + k_q d^2}(\mathcal{K}_d\mathcal{L}_d(l\cdot n) + \mathcal{K}_s\mathcal{L}_s(v\cdot r)^{\alpha})$$

$$d = \|Q - P\|$$

$$n_v = \frac{\sum_{\text{caras vecinas } c} n_c}{\|\sum_{\text{caras vecinas } c} n_c\|}$$

$$M_{Ortografica}(l,r,b,t,n,f) = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{Frustum}(l,r,b,t,n,f) = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & \frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$M_{Perspective}(\theta, A, n, f) = M_{Frustum}(-h_w, h_w, -h_h, h_h, n, f)$$

$$h_h = n \tan \pi \frac{\theta}{360}$$

$$h_w = h_h A$$

$$V = \begin{bmatrix} s_x & s_y & s_z & -e\cdot s \\ u'_x & u'_y & u'_z & -e\cdot u' \\ -f_x & -f_y & -f_z & e\cdot f \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$f = \frac{a-e}{\|a-e\|}$$

$$s = f \times u$$

$$u' = s \times f$$

$$f \quad v/vt/vn \quad v/vt/vn \quad v/vt/vn$$

$$V \approx \sum A_{ij}\Delta z_{ij}$$

$$V - E + F - H = 2(C - G)$$

$$d = \frac{ln(k)}{ln(n)} \qquad h = \frac{1}{n}$$

Formulario Control 1

$$S(\alpha,\beta,\gamma) = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{bmatrix}$$

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad \theta \text{ anti-horario}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$H_{xz}(s) = \begin{bmatrix} 1 & 0 & s \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(T_x, T_y, T_z) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$Q(t) = \underbrace{\begin{bmatrix} g_1 & g_2 & g_3 & g_4 \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}}_{T(t)}$$

$$\begin{bmatrix} P_1 & P_2 & T_1 & T_2 \end{bmatrix}$$

$$M_H = \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$T_i = \frac{1}{2}(P_{i+1} - P_{i-1})$$

$$C_i(t) = \begin{bmatrix} P_{i-1} & P_i & P_{i+1} & P_{i+2} \end{bmatrix} M_{CR} T(t)$$

$$M_{CR} = \frac{1}{2}\begin{bmatrix} 0 & -1 & 2 & -1 \\ 2 & 0 & -5 & 3 \\ 0 & 1 & 4 & -3 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$M_B = \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_1 = P_0 + \frac{T_0}{3}$$
$$P_2 = P_3 - \frac{T_3}{3}$$

$$t' = \frac{t - t_1}{t_2 - t_1}, \quad t \in [t_1, t_2], \quad t' \in [0, 1]$$