Soluciones al Recuperatorio 1 de 2024

Ej1: renderizado y escena

- ¿Qué es renderizar? Tomar la descripción de la escena y generar una imagen raster de cómo se vería eso.
- ¿Qué incluye la escena? La descripción de la escena debe incluir a los objetos (forma y material), las luces y la cámara. La forma de los objetos se aproxima con primitivas (triángulos), donde los vértices llevan propiedades adicionales (normal, coord. de textura, color, etc). Por cada luz mínimamente se necesita su posición o dirección y color. Para la cámara, además de la posición y orientación, se debe definir el tipo de proyección.
- ¿Cómo se define cada cosa? los colores (como el de la luz, o los de los materiales) como RGB o RGBA, los puntos (vértice, pos de la cámara o de la luz) y vectores (normales, dirección de luces en el infinito, etc) como XYZ o XYZW. Respecto a las demás propiedades de la cámara, se podría decir que esta info se da como dos matrices de 4x4 (view y projection), o pensar en los argumentos de las funciones que las arman (looktAt y perspective u ortho).
- Se podrían agregar las texturas (imagenes raster) y las coordenadas por vértice para mapearlas (par de reales st); aunque no habíamos visto texturas para el 1er parcial.

Ej2: CMYK

- ¿por qué? El ojo tiene receptores especializados para más o menos rojo, verde y azul (RBG). Como la hoja refleja todo, si queremos poder absorber R, G y/o B a discreción, usamos una tinta para cada uno. C absorbe R, M absorbe G, e Y absorbe B. El K reemplaza a un mínimo común a los 3 (CMY) y es para ahorrar tinta y para que los negros/grises sean de mejor calidad (la mezcla de tintas no es tan exacta en el papel).

- ¿cómo se pasa a CMYK?
    - C'=1-R, M'=1-G, Y'=1-B
    - K=min(C',M',Y')
    - C=C'-K, M=M'-K, Y=Y'-K

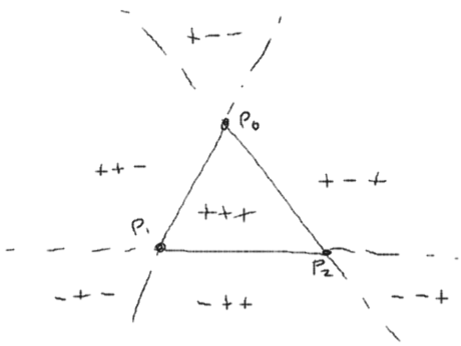
Ej3: Contigüidad

- El requisito de contigüidad pide que al pintar dos píxeles "consecutivos" de la recta o curva, no dejemos un "hueco", no saltemos más de una fila y/o una columna. Formalmente: $max(\Delta x, \Delta y) <= 1$
- En DDA calculamos el Δt para que esto no ocurra: según la tendencia (horizontal o vertical), planteamos el avance de *t* en función de una de las coordenadas (la fórmula salía de aplicar Taylor alrededor de *t* y despejar Δt, pero no hacía falta explicar todo).
- En subdivisión, como pintamos segmentos (convertimos la curva en una secuencia de segmentos conectados) y no pixeles sueltos, no puede haber discontigüidad (le pateamos el problema al algoritmo de segmentos, que será un DDA o Bresenham).

Ej4: Phong

- Ecuación: $I_a K_a + I_d K_d(\mathbf{n}\cdot\mathbf{l}) + I_s K_s(\mathbf{r}\cdot\mathbf{v})^q$
- En el ejemplo:
  - $I_a = \{0.2, 0.2, 0.2\}$, $I_d = I_s = \{1,1,1\}$
  - $K_a = K_d = \{1,0,0\}$, $K_s = \{0.5, 0.5, 0.5\}$, $q = 2$
  - $\mathbf{n} = \{0,0,1\}$,
  - $\hat{\mathbf{l}} = \{2,0,6\} - \{5,0,2\} = \{-3,0,4\}$,
    - normalizado: $\mathbf{l} = \frac{\hat{\mathbf{l}}}{|\hat{\mathbf{l}}|} = \frac{\{-3,0,4\}}{\sqrt{-3^2+0^2+4^2}} = \{-3/5, 0, 4/5\}$
    - reflejado: $\mathbf{r} = \{3/5, 0, 4/5\}$
  - $\hat{\mathbf{v}} = \{5, \sqrt{12}, 4\} - \{5,0,2\} = \{0, \sqrt{12}, 2\}$
    - normalizado: $\mathbf{v} = \frac{\hat{\mathbf{v}}}{|\hat{\mathbf{v}}|} = \frac{\{0,\sqrt{12},2\}}{\sqrt{0^2+12+2^2}} = \{0, \sqrt{12}/4, 2/4\}$
  - ambiente: $\{0.2, 0.2, 0.2\} * \{1,0,0\} = \{0.2, 0, 0\}$
  - difusa: $\{1,1,1\} * \{1,0,0\} * 0.8 = \{0.8, 0, 0\}$
    - $\mathbf{n}\cdot\mathbf{l} = \{0,0,1\}\cdot\{-3/5,0,4/5\} = 0+0+4/5 = 0.8$
  - especular: $\{1,1,1\} * \{0.5,0.5,0.5\} * 0.4^2 = \{0.08, 0.08, 0.08\}$
    - $\mathbf{r}\cdot\mathbf{v} = \{3/5,0,4/5\}\cdot\{0,\sqrt{12},2/4\} = 0+0+2/5 = 0.4$
  - suma: $\{0.2,0,0\} + \{0.8,0,0\} + \{0.08,0.08,0.08\} = \{1.08, 0.08, 0.08\}$
    - pero nunca un canal podría ser más de 1, así que hay que clampearlo a $\{1, 0.08, 0.08\}$

Ej5: Coord. baricéntricas

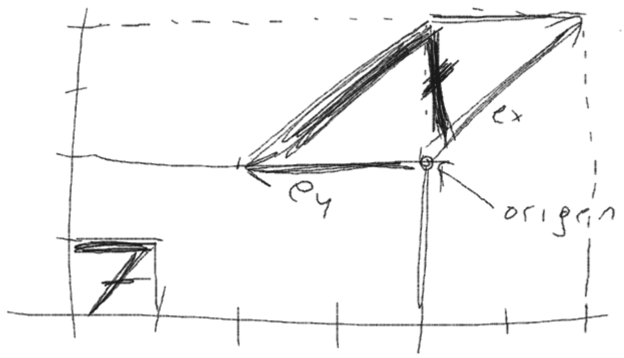


- Para calcularlas hay que plantear los "vectores area" (es decir, las áreas calculadas como producto cruz) para el punto móvil y proyectarlos sobre el vector área total (un vector no tiene signo, el signo sale de la proyección, del coseno del prod. punto). Si solo interesa el signo, no hace falta dividir por al area total al cuadrado para que sumen 1.
  - Las 3 negativas no pueden ser en ningún lado (si los prod. vectoriales están bien ordenados)
- Para la parte de rasterización, habría que calcularle los signos a los 4 puntos que definen el rectángulo.
  - Si los 4 están dentro (todos los alfas positivos), se pinta todo el rectángulo sin más análisis.

- Si los 4 están fuera "para el mismo lado" (lo que sería que los 4 tienen un mismo alfa negativo, o un mismo alfa mayor a 1) se descarta completo sin más análisis.
- Si no cae en ninguno de los dos casos anteriores, hay que subdividir o analizar pixel por pixel.

Ej6: Transformación del 7

- Es afín. No es menos que afín (no es lineal) porque hay una translación/respecto a la identidad hay cambios en la última columna. No es más que afín (no es proyectiva) porque respecto a la identidad no se altera la última fila/no giramos el plano ideal.
- El nuevo origen estará en {4,2} (última col de la matriz), y los nuevos vectores base son {2,2} y {-2,0} (primeras 2 cols de la matriz), con eso puedo dibujar el 7. Para que sea más simple, trazo paralelas a los ejes mapeando primero el cuadrado de 1x1, y adentro dibujo el 7 respetando proporciones y paralelismos:



- Si el último coeficiente pasa a ser 2, todos los puntos transformados tendrán w=2, por lo que al hacer la división perspectiva sus coordenadas se reducirán a la mitad... es como escalar todo a la mitad (todo implica los vectores base y la traslación del origen también, ,el nuevo 7 será más chico y estará más cerca del origen inicial).

Ej7: Billboard

- Para que el cuad esté de frente a la cámara su normal debe estar alineada (misma dirección, sentido opuesto) con el vector *e-t*, entonces el nuevo eje *z* será *t-e* normalizado.
- Para que gire de forma que los laterales del quad quede alineados (parelelos) a los de la imagen, el nuevo eje *y* debe ir en dirección del *u*, será el *u* normalizado (si asumimos que *u* es perpendicular a *e-t*, si no habría que corregirlo con un par e productos cruz).
- Para que los bordes inferior y superior del quad queden alineados (paralelos) a los de la imagen, el nuevo eje *x* debe ir hacia la derecha del plano que definen *e*, *t* y *u*, así que el nuevo eje *x* será *(e-t)* x *u* normalizado.
- Con esos ejes, giramos el quad manteniendo el tamaño.. En realidad para que el tamaño sea *w* x *h* hay que multiplicar por *W* y por *h* a elos nuevo ejes *x* e *y*.
- Si usamos P como traslación no va a quedar centrado así que habría que moverlo *w*/2 en *x* y *h/2* en *y* (esto se hace con los nuevos ejes *x* e *y*).

$$e_z = \frac{e - t}{|e - t|}$$

$$e_y = v/|v|$$

$$e_x = e_z \times e_y$$

$$O = P - \frac{w}{2} e_x - \frac{h}{2} e_y$$

$$\begin{array}{cccc} w e_x & h e_y & e_z & O \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array}$$
$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Alternativamente podríamos hacer una primera matriz para escalarlo con *w* y *h*, y centrarlo en el origen, y luego (premultiplicarle) una como la de la figura pero sin escalar los ejes con *w* y *h*, y con *P* directamente como nuevo origen.