

IDI – Resum primer examen

Model Geomètric

Frame Buffer: Memòria, GPU (RAM gràfica) per guardar els píxels que s'han de pintar per pantalla.

Vertex Array Object: Conjunt de VBO, representa un model que volem veure.

Vertex Buffer Object: Estructura de dades que guarda tant els vèrtex d'un model, com els colors, la normal, i altres característiques dels vèrtexs.

Visualització

Mètode `pinta_model()`: VAO → Processament de vèrtexs → Rasterització → Processament de fragments.

Processament de vèrtexs – Vertex Shader

View Matrix (Viewing Transform) → **Project Matrix** (Projection Transform) → Clipping →

→ Viewport (Device Transform)

Processament de fragments – Fragment Shader

Càlcul de color, i altres efectes.

Com pintar?

```
// Un sol cop
per cada model {
    crear i omplir VAOi
    crear i omplir VBOi
```

```
// Cada cop que es refresca
per cada objecte {
    // Calcular la TGi a aplicar
    modelTransform(TGi)
    // Indicar a OpenGL la TG.
```

Càmera

Viewport: Finestra OpenGL.

OBS: Posició del observador.

VRP: Posició a on mira la càmera.

Window: Finestra de la càmera.

up: Direcció l'eix vertical de la càmera (inclinació)

$$\text{View Matrix} = \text{VM} = \begin{pmatrix} x_{obs} \cdot x & x_{obs} \cdot y & x_{obs} \cdot z & 0 \\ y_{obs} \cdot x & y_{obs} \cdot y & y_{obs} \cdot z & 0 \\ z_{obs} \cdot x & z_{obs} \cdot y & z_{obs} \cdot z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

α_v : Angle de la càmera. $\alpha_v = \arctg\left(\frac{h_v}{2 \times zN}\right) = \arcsin\left(\frac{R}{d}\right)$

FOV: Angle d'obertura total. $FOV = 2 \times \alpha_w$

zNear: Pla de tall de la càmera per davant.

zFar: Pla de tall de la càmera per darrere.

ra_w: Relació d'aspecte del Window. $ra_w = \frac{a_w}{h_w}$

View Matrix: Matriu del observador o càmera.

Project Matrix: Matriu de la òptica de la càmera.

$$h_w = 2 \times zN \times \tan(\alpha_w); a_w = h_w \times ra_w$$

```
// View Matrix
VM = lookAt(OBS, VRP, up);
// Indicar a OpenGL la VM
viewMatrix(VM);
```

```
// Project Matrix
PM = perspective(FOV, ra, zN, zF);
// Indicar a OpenGL la PM
projectMatrix(PM);
```

Escena sense deformar

Si $ra_v < 1$, llavors: $FOV = 2 \times \alpha_v$

$$\alpha_v = \arctg\left(\frac{tg(\alpha_v)}{ra_v}\right)$$

*d > Mínim z de la Esfera Mínima Contenidora.

Zoom: Modificar FOV, mantenir ra .

Paràmetres de la càmera ortogonal

En comptes de tenir un camp de visió en forma de piràmide, és en forma de prisma.

L: Left

R: Right

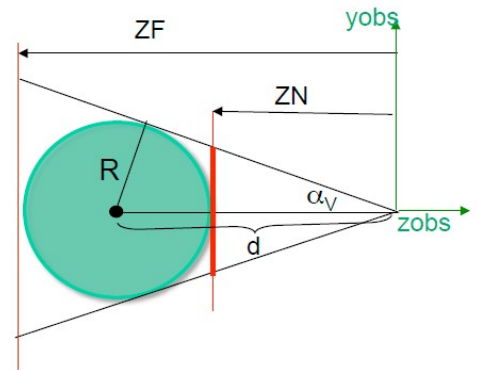
B: Bottom

T: Top **zNear, zFar, Window.**

$$h_w = t - b \rightarrow h_w = 2R; a_w = r - l$$

Rotate(angle, x, y, z);

Càmera comença mirant al 0,0,0 Z negatives, amb Zfar = 1



```
// Project Matrix
PM = ortho(l, r, b, t, zN, zF);
// Indicar a OpenGL la PM
projectMatrix(PM);
```

Realisme

Back face culling: Càlcul de les coses que es veuen i no es veuen.

Depth Buffer: Mètode EPA després de la rasterització.

Models d'il·luminació

Models locals: Només tenen en compte un objecte i el focus.

Models globals: Tenen en compte els altres objectes, obres, radiositat...

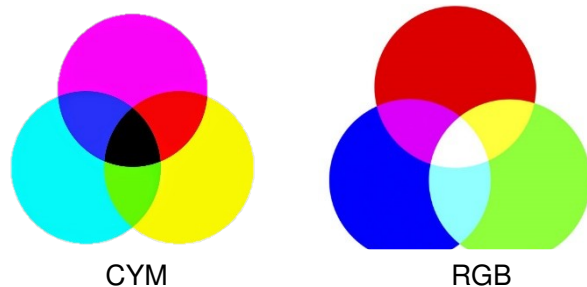
- **Model de Lambert:** Focus puntual. Objectes només tenen reflexió difusa pura. Es veuen les cares 3D.
- **Model de Phong:** Focus puntual, objectes només reflexió especular. Suavitzat d'arestes. **Canvia segons posició d'espectador.**

Elements d'il·luminació a OpenGL

- **Ka:** Coeficient de reflexió **ambient**.
- **Kd:** Coeficient de reflexió **difusa**.
- **Ks:** Coeficient de reflexió **especular**.
- **N:** Exponent de reflexió **especular**.
- **Ia:** Color llum **ambient**.
- **If:** Color focus **puntual**.

Llum en SCO: `vec4(posFocus, 1.0)`

Llum estàtica en escena: `view * vec4(posFocus, 1.0)`



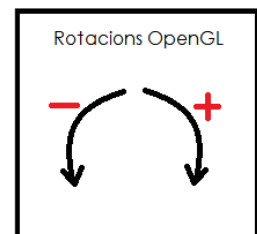
Color

Color additiu: Afegeix color R, G o B segons el desitjat.

Color substractiu: De la llum blanca, li treu R, G o B segons el desitjat.

Models de color

- **CMY(K):** **C**ian, **M**agenta, **Y**groc i **K**negre. Model substractiu, utilitzat per imprimir. $= 1-C; G = 1-M; B = 1-Y$
- **HSV:** Matís (*hue*, color), saturació (puresa), intensitat.
- **CIE:** Equació $X+Y+Z=1$, on cada variable correspon a un estímul humà.



(R)