

# IDI : RESUM PRIMER PARCIAL :

**MODEL GEOMÈTRIC** : MODEL DE FRONTERES (APROXIMACIÓ PER CÀRRES PLANES (TRIANGLES)). (OM GUARDEM ELS TRIANGLES? ES GUARDEN PER CÀRRES QUE ES RELACIONEN AMB ELS SEUS TRES VÈRTEXS.

PER VISUALITZAR MODELS 3D NECESITEM:

- Vertex Array Object: conjunt de VBO's, representa el model.
- Vertex Buffer Object: EMmagatzema tant els vèrtexs, com els colors, la normal i altres.

**VISUALITZACIÓ** : MÈTODE `pinta_model()`: VAO → PROCESSAMENT VÈRTEXS → RASTERITZACIÓ → PROCESSA FRAGMENTS

- PROCESSAMENT VÈRTEXS: VERTEX SHADER, VIEW MATRIX → PROJECT MATRIX → CLIPPING → VIEWPORT
- PROCESSAMENT FRAGMENTS: FRAGMENT SHADER: càlcul de color i altres.

Per poder definir una càmera necessitem:

- OBS: Punt on està la cam, NO PARALLELA A DIRECCIÓ CAM.
- VECTOR UP: VERTICAL DE LA CAM (inclinació).
- VRP: Punt on mira la cam.

Per tant, ViewMatrix = VM =

```
// VIEW MATRIX
VM = lookAt(OBS, VRP, UP);
// INDICAR A OPENGL LA VM
glViewMatrix (VM);
```

```
s.x s.y s.z 0
w.x w.y w.z 0
F.x F.y F.z 0
0 0 0 1
```

com l'apliquem?  
// VERTEX SHADER

TRANSFORMACIÓ GEOMÈTRICA

Project Matrix

$gl\_position = PM * VM * TG * vec4(vertex, 1.);$

```
(r-a)^2 0 0 0
0 d^2 0 0
0 0 c d
0 0 -1 0
```

**ÒPTICA PERSPECTIVA**:

- $\alpha_v$ : angle de la cam.  $\alpha_v = \arctan\left(\frac{h_v}{2 \cdot z_N}\right) = \arcsin\left(\frac{R}{d}\right)$
- FOV: angle obertura:  $FOV = 2 \cdot \alpha_{uv}$
- zNear: distància entre OBS i window.
- zFar: límit de renderitzat.
- r\_w: relació aspecte window,  $r_w = \frac{a_w}{h_w}$

$$h_w = 2 \cdot z_N \cdot \tan(\alpha_w)$$

Per tant, Project Matrix = PM =

on:  $-a = \tan(FOV/2)$

$$c = (z_F + z_N) / (z_N - z_F)$$

$$d = 2 \cdot z_N \cdot z_F / (z_N - z_F)$$

```
// Project Matrix
PM = perspective(FOV, r_w, z_N, z_F);
// INDICAR A OPENGL LA PM
glProjectMatrix (PM);
```

**ÒPTICA ORTOGONAL**:

- L: Left  $hw = t - b = 2 \cdot R$
- R: Right  $aw = r - l$
- B: Bottom  $r_w = aw / hw$
- T: Top

Per tant, Project Matrix = PM =

```
PM = ortho(l, r, b, t, z_N, z_F);
ProjectMatrix (PM);
```

```
a 0 0 0
0 b 0 0
0 0 c d
0 0 0 1
```

on:  $-a = 2 / (r - l)$   
 $-b = 2 / (t - b)$   
 $-c = 2 / (z_F - z_N)$   
 $-d = (z_N + z_F) / (z_F - z_N)$   
 $-e = 2 / (t - b)$

Si modifiquem el viewport es produeixen deformacions que s'han d'evitar.

- La relació del viewport i la del window han de coincidir, per tant, hem de modificar el window.

a) Si  $r_w < 1$  ALTESSES  $FOV = 2 \cdot \alpha_v$

b) Si  $r_w > 1$  ALCS HORES  $r_w = r_v$

$$\alpha_v = \arctan\left(\frac{t_s(h_v)}{r_w}\right)$$

com podem aconseguir zoom depenent de l'òptica que usem: modificant el window de la càmera.

- PERSPECTIVA: modificar el FOV, mantenint la  $r_w$ .
- ORTOGONAL: modificar el window, mantenint la  $r_w$ .

**TRANSFORMACIONS GEOMÈTRIQUES**: N'hi ha de diversos tipus:

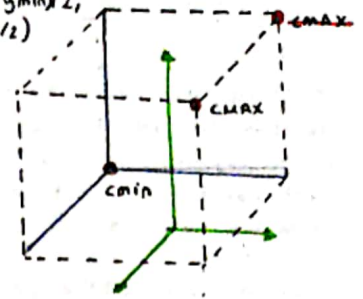
- TRASLACIÓ:  $T(t_x, t_y, t_z)$ . DESPLAÇA Els VÈRTEXS ALLS QUE INDICA EL VECTOR.
- ROTACIÓ:  $R_e(\alpha)$ .  $e \in \{x, y, z\}$ . REGLA À DRETA PER SABER SI LA ROTACIÓ HA DE SER POSITIVA O NEGATIVA.
- ESCALAT:  $S(s_x, s_y, s_z)$ . s'indica el factor d'escalat de cada eix, no la mida final.

PM: SCA → SCO, TG: SCM → SCA, VM: SCO → SCC  
 VM: SMM → SCA → SCobs → SCclip → SCnorm → SCdevice

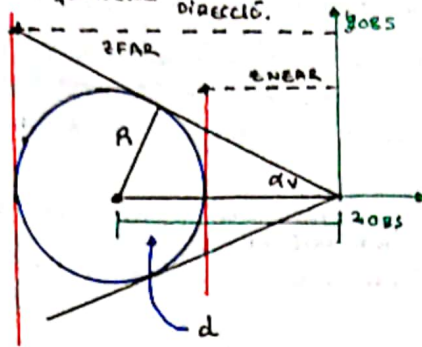
**CÀMERA EN 3A PERSONA:** Implica que es veu tota l'escena sense retallar les passos:

1. Trobar el centre de l'escena i situar el VRP allà.  $\text{centre} = ((x_{\max} + x_{\min})/2, (y_{\max} + y_{\min})/2, (z_{\max} + z_{\min})/2)$
2. CAPÇA MÍNIMA CONTENIDORA DE L'ESCENA.  $\text{cmin} = (x_{\min}, y_{\min}, z_{\min})$   $\text{cmax} = (x_{\max}, y_{\max}, z_{\max})$
3. OBS HA D'ESTAR FORA CAPÇA MÍNIMA CONT.

PERTANT, OBS FORA ESFERA ENCLUBANT DE LA CAPÇA  $\Rightarrow$  distància "d" de l'OBS a VRP superior a R.  
 $R = \text{dist}((x_{\min}, y_{\min}, z_{\min}), (x_{\max}, y_{\max}, z_{\max})) / 2$   
 $\text{dist} = \sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2 + (z_{\max} - z_{\min})^2}$   
 $\text{OBS} = \text{VRP} + d \cdot V \rightarrow V$  NORMALITAT qualsevol direcció.



4.  $z_{\text{NEAR}} = d - R$ ,  $z_{\text{FAR}} = d + R$
5.  $\alpha_V = 2 \arcsin(R/d)$ ,  $\text{FOV} = 2 \cdot \alpha_V$
6.  $\text{FW} = 2w / h_w = 1$



- RGB  $\rightarrow$  CMY:  $(1, .8, 0) \rightarrow (.0, .2, 1)$   
 - CMY  $\rightarrow$  RGB:  $(.4, .2, 1) \rightarrow (.6, .8, 0)$

**ANGLES D'EULER:** PERMETEN CALCULAR LA VIEWMATRIX DIRECTAMENT. COM SI GIREM L'ESFERA PER A QUE LA SEUA POSICIÓ RESPECTE LA CàMERA PER DEFECTE SIGUI LA MATEIXA. ADOPTAR L'ESFERA I POSICIONAR LA. NO ÉS NECESSARI MODIFICAR L'ÒPTICA. COM CALCULEM ES ANGLES?

1. T(-VRP): MOURE L'ESCENA A L'ORIGEN.
2.  $R_y(-\psi)$ : ROTACIÓ CILY.
3.  $R_x(\theta)$ : ROTACIÓ CILX.
4. T(0,0,d): MOURE L'ESCENA A DISTÀNCIA d EN +Z.

$$VM = T(0,0,-d) \cdot R_x(\theta) \cdot R_y(-\psi) \cdot T(-VRP)$$

**INTRODUCCIÓ A HCI:** HCI ÉS UN CAMP QUE TRACTA L'ESTUDI DE COM ELS HUMANS INTERACTUEN AMB LES MÀQUINES. UNA APP HA DE CUMPLIR ELS SEUS REQUISITS I POVEIR FÀCIL ACCÉS AIS SEUS FEATURES.

- EFICÀCIA: CAPACITAT DE CONSEGUIR UN OBJECTIU DE MANERA CORRECTA I COMPLETA.
- EFICIÈNCIA: RELACIÓ ENTRE RECURSOS I LA EXACTITUD I LA COMPLETSSA DELS OBJECTIUS ACONSEGUITS.
- SATISFACCIÓ: COMODITAT I ACEPTACIÓ D'UN SISTEMA PELS USERS.
- User experience: ÉS INHERENT A COM SE SENT L'USUARI I QUANT RECORDA DE L'ÚS D'UN DISPOSITIU.

**PRINCIPIS DE DISSENY:**

1. ESTÈTICA: UNA ESTÈTICA AGRAÐABLE NO S'HA D'INTERFERIR A LA USABILITAT.
2. ANTICIPACIÓ: LLIURAR AL USER INFO. I TOOLS NECESSÀRIES PER CADA PAS DEL PROCÉ. SI UNA FEATURE NO ESTROBA, MÉS S'USARÀ. SI L'EXPERIÈNCIA ÉS FRUSTANT ES POT PERDRE A L'USUARI.
3. AUTONOMIA: L'USUARI HA DE PODER PRENDRE LES SEVES PROPIES DECISIONS. L'USUARI HA DE TENIR INFO. DE L'ESTAT I PROCESSOS ACTUALS. MOSTRAR DE MANERA PRECISA TEMPS DESTANT.
4. COLOR: TENIR EN COMPTA POSSIBLES DISCAPACITATS VISUALS, NO COLORES QUE CAUIN CONFUSIÓ.
5. CONSISTÈNCIA:
  - a) PLATAFORMA: MANTENIR LOOK GENERAL A TRAVÉS D'ES PRODUCTES ENTRE PLATAFORMES.
  - b) SUITE DE PRODUCTES: SI TENIM DIVERSES APPS, BONA PRÀXIS TENIR LOOK I COMPORTAMENT SIMILAR.
  - c) IN-APP: A TOTA L'APLICACIÓ MANTENIR UN LOOK / COMPORTAMENT.
  - d) ESTRUCTURES VISIBLES: ICONS, SÍMBOLS I POSICIONAMENT SIMILAR.
  - e) ESTRUCTURES NO VISIBLES: ÚS D'AQUESTES ESTRUCTURES SIMILAR A TOTA L'APP.
  - f) COMPORTAMENT DE L'USUARI: NO CANVIA EL SIGNIFICAT D'UNA ACCIÓ HABITUAL PER A L'USUARI.
  - g) TIPSUS DE CONSISTÈNCIA: INCONSISTÈNCIA INDUÏDA (FER OBJ DIFF. SI ACTUEN DIFF.), CONTINUITAT INDUÏDA (MANTENIR CONTINUITAT TOT I RELACES QUE INTRODUEIXIN CANVIS) I CONSISTÈNCIA AMB USER EXPECT (SI EN USER CREU QUE ALGO ES FÀCIL D'USAR, NO CANVI).
6. VÍDEOS DE DEFECTE: FÀCILS D'ESCRIBRE, NO SEMPRE NECESSÀRI SINÓ APORTA MÉS VERITATJA AL USER.
7. DESCOBRIR: SI NO ES POT TROBAR NO EXISTEIX, CONTROLS VISIBLES, BASTANT COMPLEXITAT POT INCREMENTAR LA, TUTORIALS... (PERQUÍ).
8. EFICIÈNCIA: EVALUAR PRODUCTIVITAT USUARI, MISSATGES D'ERROR CLARS, MANTENIR L'USUARI OCUPAT, PERDRE EFICI. VENE CANVIA.
9. INTERFACES EXPLORABLES: L'USUARI HA DE PODER EXPLORAR L'INTERFACE PER TANT, FER ACCIONS REVERSIBLES.
10. FITT'S LAW: "THE TIME TO ACQUITE A TARGET IS A FUNCTION OF THE DISTANCE TO AND THE SIZE OF THE TARGET". BIG OBJECTS ARE EASY TO FIND.
11. OBJECTES INTERFICIL HUMANS: METÀFORES, SLIDERS QUE ES DESLITZEN, BOTONS QUE CÀPIELEN...
12. INFORMING USERS: INFORMAR A L'USUARI SI HI HA ALGÚN TIPUS D'ERRORE.
13. USABILITAT: ESTOILLIR METÀFORES QUE PERMETIN A L'USUARI CAPTAR A L'INSTANT ELS DETALLS MÉS FINS DEL MODEL CONCEPTUAL. ASSIGURAR-SE QUE L'USUARI NO PERD LA FINESSA (SAVE BUTTON). TEXT LLEGIBLE ALT CONTRAST FONS.