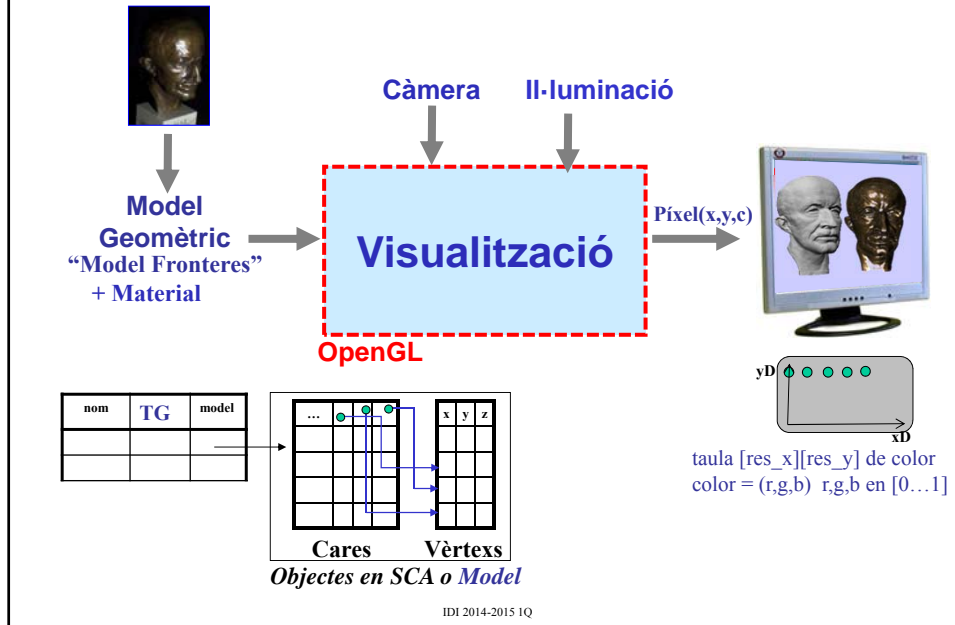
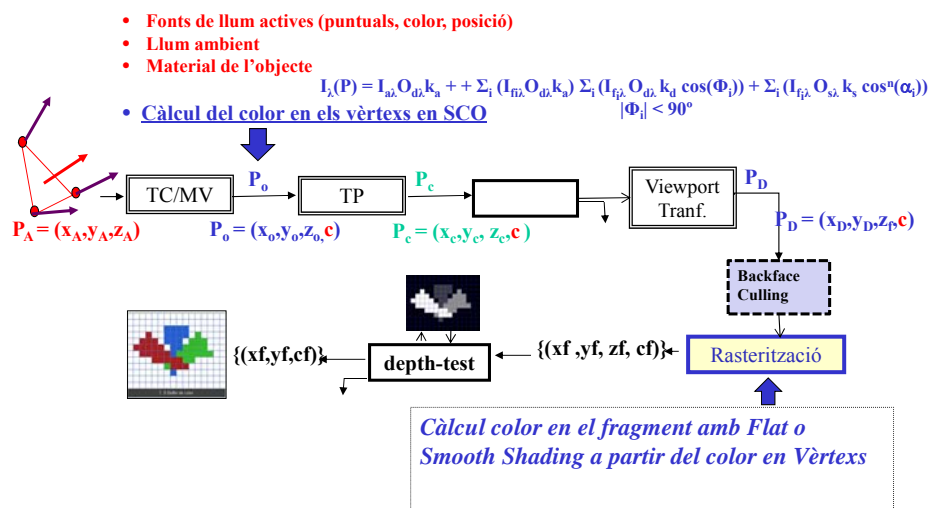


Realisme (3)



Il·luminació en OpenGL (repàs)

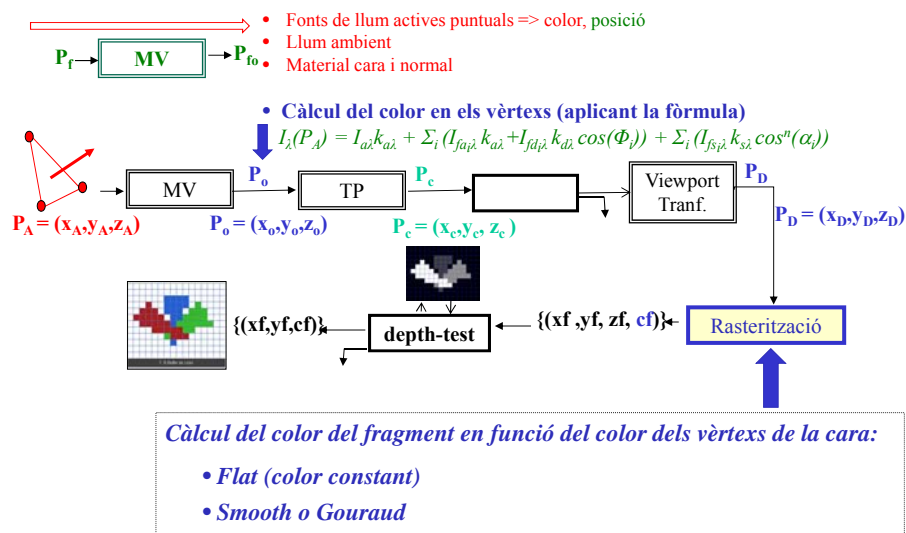


Realisme en OpenGL

- Resum del procés (instruccions en manual):
 - Definir fonts de llum: color, posició, tipus
 - Definir propietats de material: constants k_λ
 - Especificar model de colorat; flat/smooth
 - Activar il·luminació (lighting) i focus
 - Activar test de profunditat (Z-Buffer, culling)
 - Pintar triangle:
 - especificar Normal per Vèrtex o per Cara
 - glVertex3fv(...)

IDI 2014-2015 1Q

Shading (colorat) de polígons (3)



IDI 2014-2015 1Q


Il·luminació en OpenGL

IMPORTANT:

- OpenGL realitza els càlculs de la il·luminació per vèrtex en *coordenades d'observador*
- La matriu Modelview activa en el moment de definir la posició del focus de llum determina la seva posició en el moment de fer els càlculs:
 - *Modelview amb la TG de la càmera*
Llum en posició fixa respecte l'escena
 - *Modelview identitat*
Llum en posició fixa respecte l'observador/càmera
 - *Modelview amb $TG_camera * TG_objecte$*
Llum en posició fixa respecte d'un objecte

IDI 2014-2015 1Q

```
1. setProjection(); //inicialitza la gluPerspective
2. glMatrixMode(GL_MODELVIEW);
3. glLoadIdentity();
4. glTranslatef(0,0,-dist);
5. glRotatef(-angleZ,0,0,1);
6. glRotatef(angleX,1,0,0);
7. glRotatef(-angleY,0,1,0);
8. glTranslatef(-VRP.x,-VRP.y,-VRP.z);
9. Pinta_Cub();
```



gluLookAt()

IDI 2014-2015 1Q

Exercici 33:

El següent codi permet pintar un cub de costat 6 centrat en el punt (0,3,0). El volem il·luminar amb dues llums. La llum 0 s'ha de moure amb la càmera. La llum 1 estarà sempre ubicada sobre el cub en la recta que passa pel seu centre en la direcció de l'eix Y de l'aplicació i a distància 3 del centre de la seva cara superior. Contesta raonadament les respostes.

- Indica les crides a `glLight` necessàries per a posicionar ambdues llums i el lloc del codi en que les posaries.

Observació: els colors de les llums ja estan inicialitzats, només has de fer la crida per a indicar la posició.

- Imagineu que volem moure el cub en direcció de l'eix X amb la llum 1 a sobre. Indica i justifica les modificacions requerides del codi.

```
1. setProjection(); //inicialitza la gluPerspective
2. glMatrixMode(GL_MODELVIEW);
3. glLoadIdentity();
4. glTranslatef(0,0,-dist);
5. glRotatef(-angleZ,0,0,1);
6. glRotatef(angleX,1,0,0);
7. glRotatef(-angleY,0,1,0);
8. glTranslatef(-VRP.x,-VRP.y,-VRP.z);
9. Pinta_Cub();
```

IDI 2014-2015 1Q

Exercici 18:

Donat el tros de codi OpenGL següent:

```
float posllum1[4] = {0, 10, -10, 1};
float posllum2[4] = {0, 10, 0, 1};
glMatrixMode (GL_MODELVIEW);
glLoadIdentity ();
glTranslate (0, 0, -10);
glRotatef (-90, 0, 1, 0);
glLightfv (GL_LIGHT1, GL_POSITION, posllum1);
glPushMatrix ();
glLoadIdentity ();
glLightfv (GL_LIGHT2, GL_POSITION, posllum2);
glPopMatrix ();
pintaescena ();
```

Indica quines són les posicions de l'observador, del llum `GL_LIGHT1` i del llum `GL_LIGHT2` (en el sistema de coordenades de l'escena) quan es pinta l'escena. Justifica la resposta.

IDI 2014-2015 1Q

Exercici 57:

Es vol pintar una escena amb un focus situat a la posició (10, 20, 0) en coordenades de l'aplicació, indica en quina posició del codi següent posaries la instrucció:

```
glLightfv (GL_LIGHT0, GL_POSITION, posfocus)
```

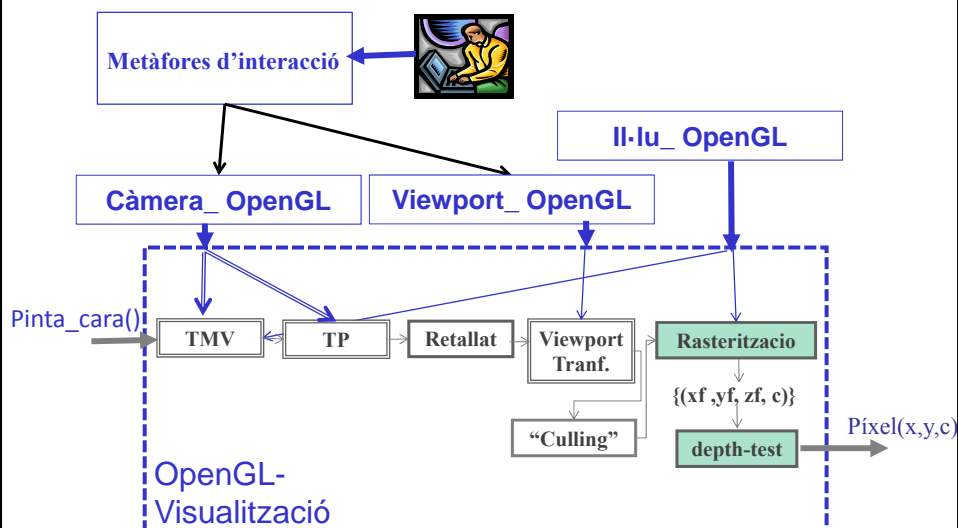
i amb quins valors s'ha d'inicialitzar la posició del focus (posfocus)

```
1. glMatrixMode (GL_PROJECTION);
2. glLoadIdentity ();
3. gluPerspective (alfa, ra, znear, zfar);
4. glMatrixMode (GL_MODELVIEW);
5. glLoadIdentity ();
6. gluLookAt (0, 20, 0, 0, 0, 0, 0, 0, 1);
7. drawScene ();
```

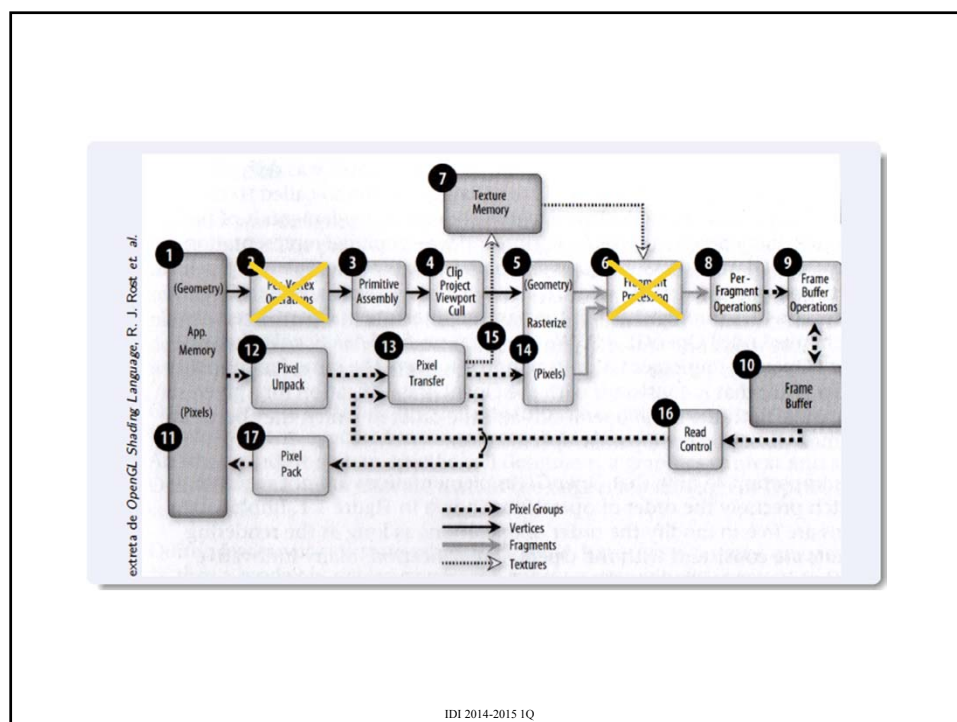
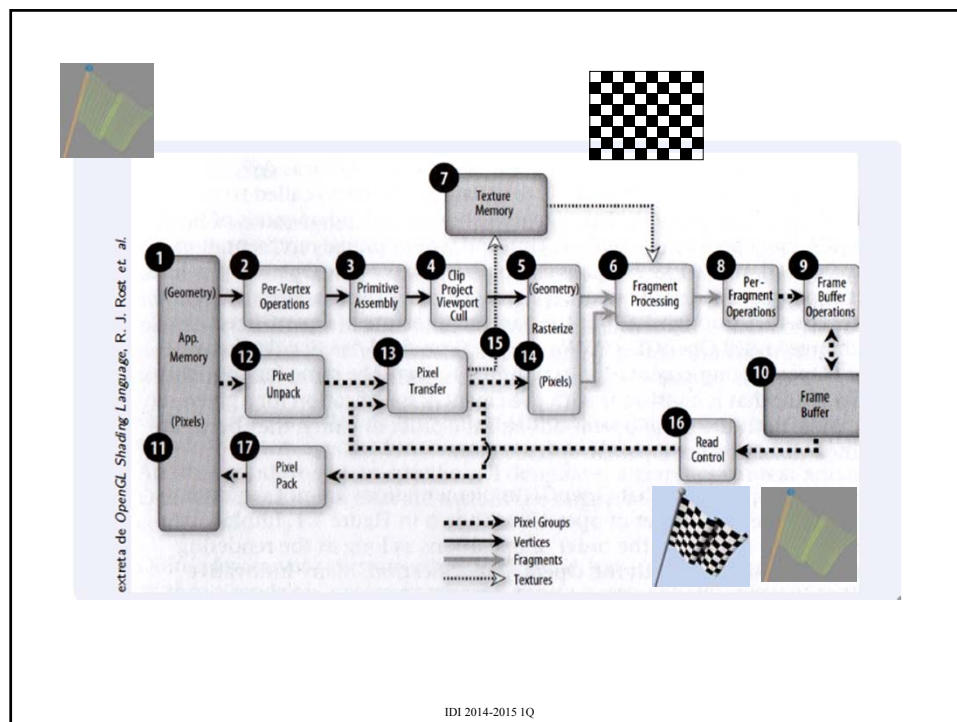
- a) Entre les instruccions 5 i 6, i posfocus=(-10,0,0,1)
- b) Entre les instruccions 6 i 7, i posfocus=(-10,-20,0,1)
- c) Després de la instrucció 7, i posfocus=(10,20,0,1)
- d) Entre les instruccions 6 i 7, i posfocus=(-10,0,0,1)

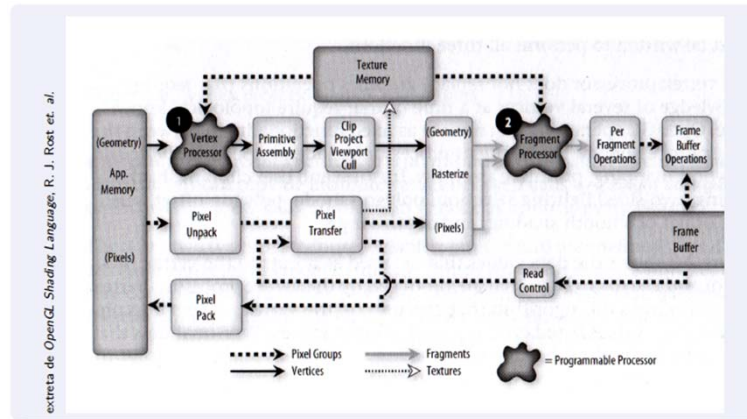
IDI 2014-2015 1Q

Visualització OpenGL: Pipeline bàsic

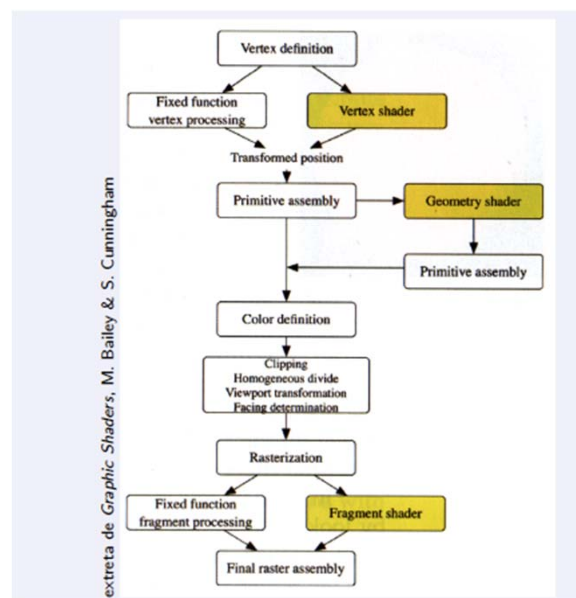


IDI 2014-2015 1Q





IDI 2014-2015 1Q



IDI 2014-2015 1Q