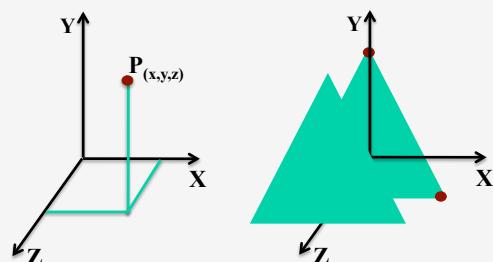
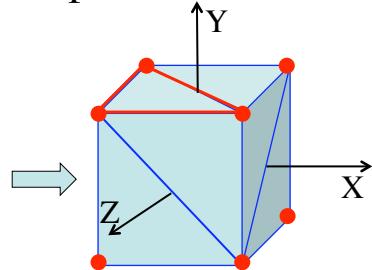
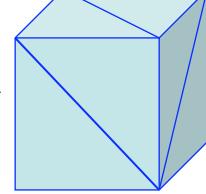
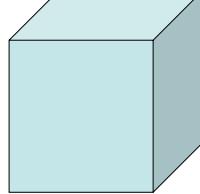


Models Geomètrics (intro)



IDI Q1 2016-2017

Model Fronteres: Exemple Cub



Per cada triangle

- Geometria
- Topologia
(implícitament)

Vèrtexs repetits ☹

Vèrtexs		
x	y	z
-1	+1	-1
-1	+1	+1
+1	+1	+1
...

Vèrtexs		
x	y	z
-1	+1	-1
-1	+1	+1
+1	-1	+1
...

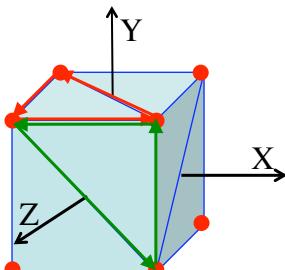
IDI Q1 2016-2017

4

Exemple: Model Fronteres		
Cares		Vèrtexs
normal	Id Vèrtexs	x y z
a_1, b_1, c_1	1,2,3	-1 1 -1
a_2, b_2, c_2	2,4,3	-1 1 1
...	...	1 1 1
		1 -1 1
		1 -1 -1
		1 1 -1
		-1 -1 -1
		-1 -1 1

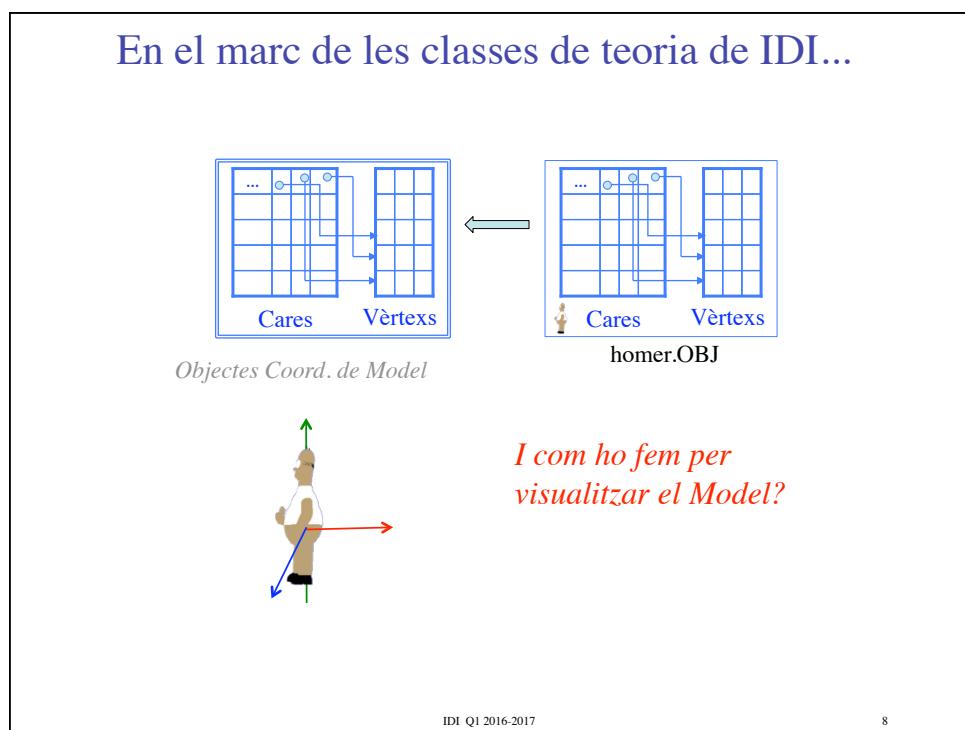
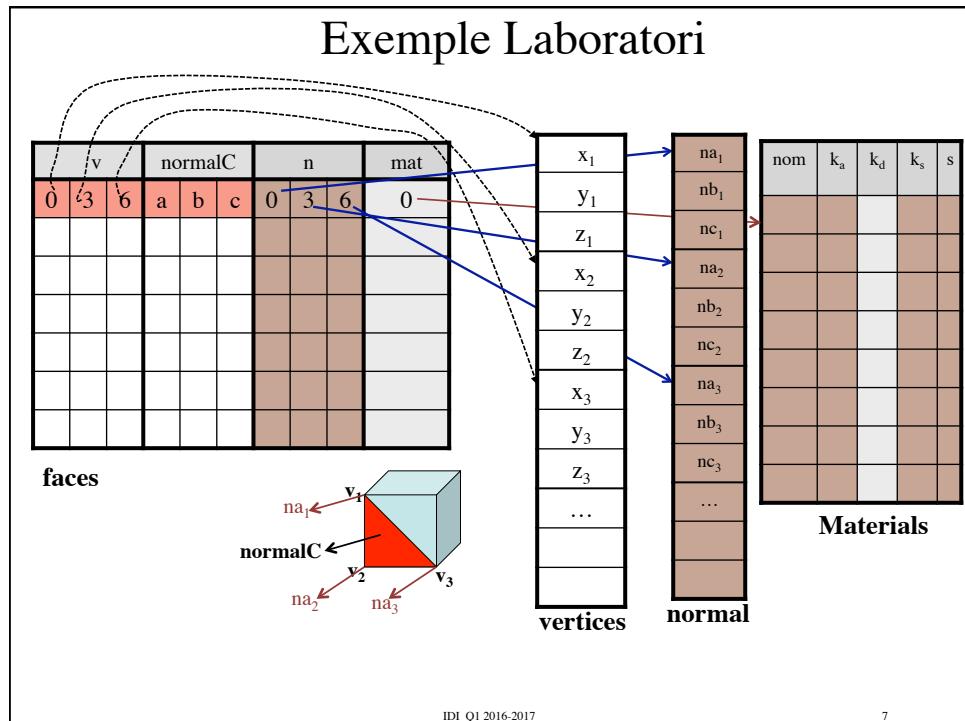
IDI Q1 2016-2017

5

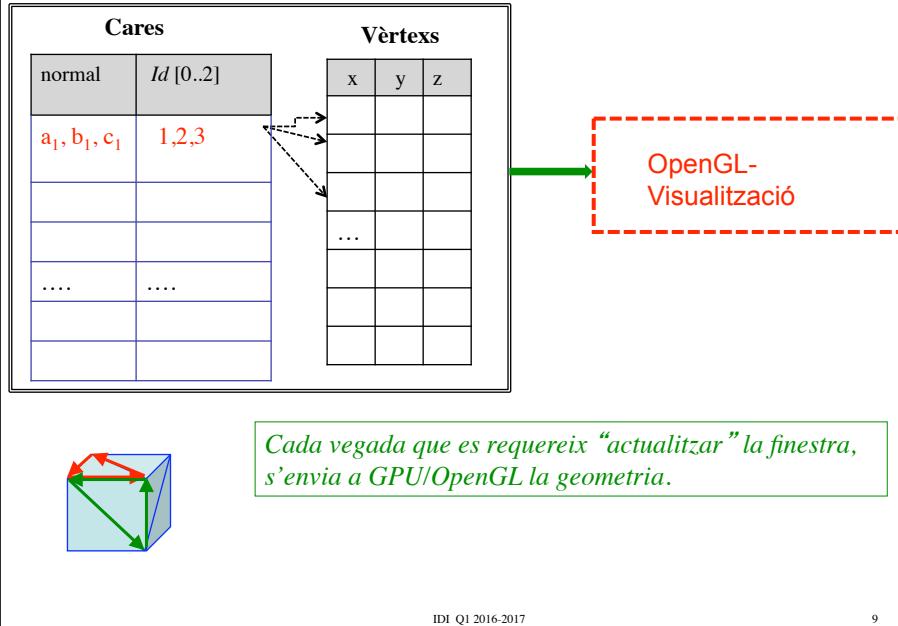
Model Fronteres Vàlid		
<ul style="list-style-type: none"> - Cares “orientades”. - Ordenació vèrtexs coherent amb l’orientació de les cares. - Cada aresta separa 2 cares. 		
		

IDI Q1 2016-2017

6



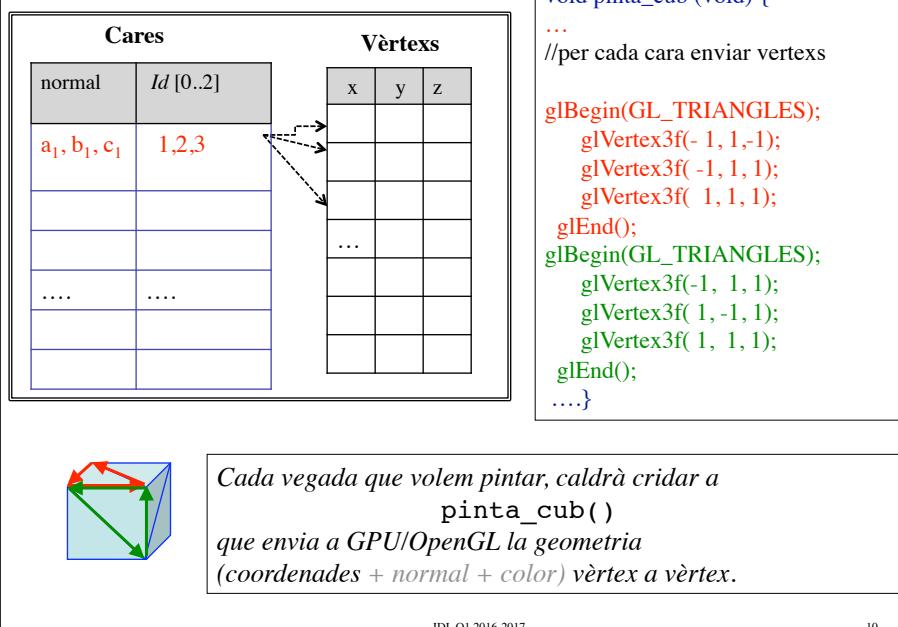
Pintar en OpenGL: Mode immediat



IDI Q1 2016-2017

9

Pintar en OpenGL: Mode Immediat



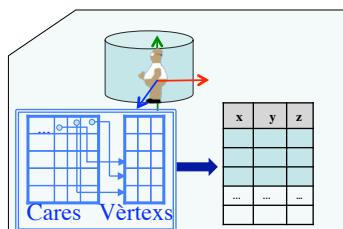
IDI Q1 2016-2017

10

Pintar en OpenGL 3.3: “core” mode

Un sol cop cal enviar/passar el *model/geometria* a la GPU/OpenGL com una llista amb les coordenades dels vèrtexs de cada triangle o com *triangle strips*.

Per tant, caldrà crear una estructura auxiliar/temporal amb aquesta informació a partir de la nostra estructura de dades.



Aplicació. Model Geomètric

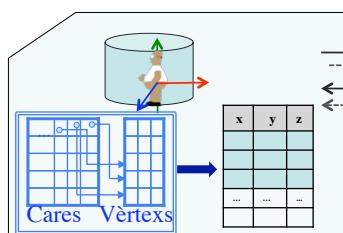


IDI Q1 2016-2017

12

Pintar en OpenGL 3.3: “core” mode

1. Crear en GPU/OpenGL un VAO que encapsularà dades del model. Crear VBO que guardarà les coordenades dels vèrtexs (potser cal altres per normal, color,...)



Aplicació. Model Geomètric

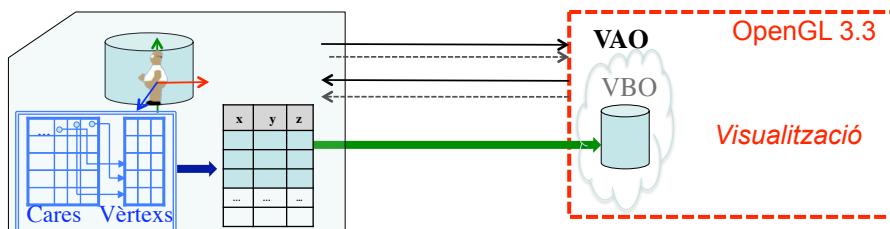


IDI Q1 2016-2017

13

Pintar en OpenGL 3.3: “core” mode

1. Crear en GPU/OpenGL un *VAO* que encapsularà dades del model.
Crear *VBO* que guardarà les coordenades dels vèrtexs (potser cal altres per normal, color,...)
2. Guardar llista de vèrtexs (amb repetició) o *strips* en el seu *VBO*
–i si cal, color i normal en els seus *VBO*-



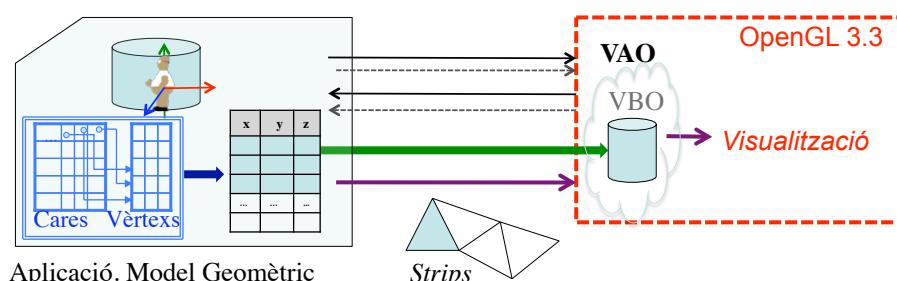
Aplicació. Model Geomètric

IDI Q1 2016-2017

14

Pintar en OpenGL 3.3: “core” mode

1. Crear en GPU/OpenGL un *VAO* que encapsularà dades del model.
Crear *VBO* que guardarà les coordenades dels vèrtexs (potser cal altres per normal, color,...)
2. Guardar llista de vèrtexs (amb repetició) o *strips* en el seu *VBO*
–i si cal, color i normal en els seus *VBO*-
3. Cada cop que es requereix pintar, indicar el *VAO* a pintar i dir que es pinta: *glDrawArrays(...)*. Acció **pinta_model()** a teoria.



Aplicació. Model Geomètric

IDI Q1 2016-2017

15

Classe 1: contigut

- Introducció a la Informàtica Gràfica
- Elements d'un sistema gràfic interactiu
 - Hardware de sortida
 - Models geomètrics (1ra part)