

# Vertex Arrays i Vertex Buffer Objects

C. Andujar, A. Vinacua

Març 2013

# Formes de pintar geometria

- Mode immediat (glBegin,glEnd)
- Usant vertex arrays (VAs)
- Usant Vertex buffer object (VBOs)

# Mode immediat

```
for (i=0; i<T; ++i) {  
    glBegin(GL_POLYGON);  
    glNormal3f(...);  
    glVertex3f(...);  
  
    glNormal3f(...);  
    glVertex3f(...);  
  
    glNormal3f(...);  
    glVertex3f(...);  
    glEnd();  
}
```

# Mode immediat

Client side

Vertices

x	y	z	N <sub>x</sub>	N <sub>y</sub>	N <sub>z</sub>
-1.0	-2.0	5.0	1.0	0.0	0.0
...	...	...	...	...	...
...	...	...	...	...	...

Faces

Normal	ind	ind	ind
1, 0, 0	0	4	2
...	...	...	...
...	...	...	...

T crides a glBegin()  
3\*T crides glVertex()  
3\*T crides glNormal()

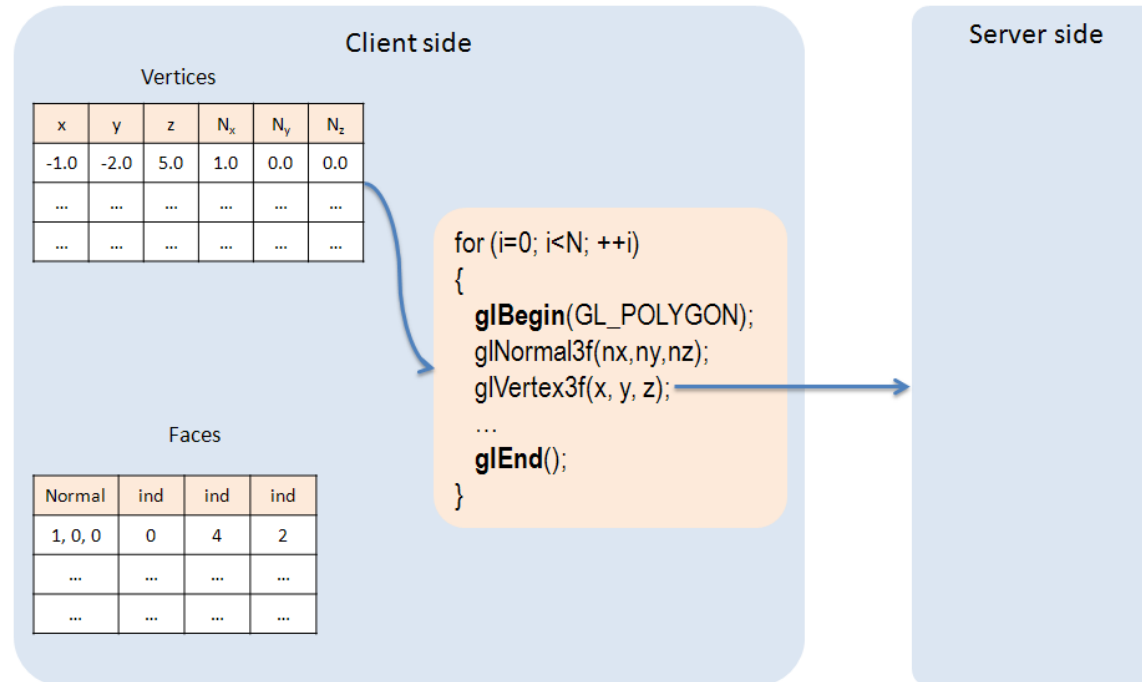
```
for (i=0; i<N; ++i)
{
    glBegin(GL_POLYGON);
    glNormal3f(nx,ny,nz);
    glVertex3f(x, y, z);
    ...
    glEnd();
}
```

**3\*2\*3\*T floats**  
(3 vèrtexs/triangle,  
2 atributs/vèrtex,  
3 float/attrib)

Server side

# Mode immediat

- Senzill, fàcil de depurar, flexible...
- Moltes crides a funcions
- Cal transferir totes les dades cada frame

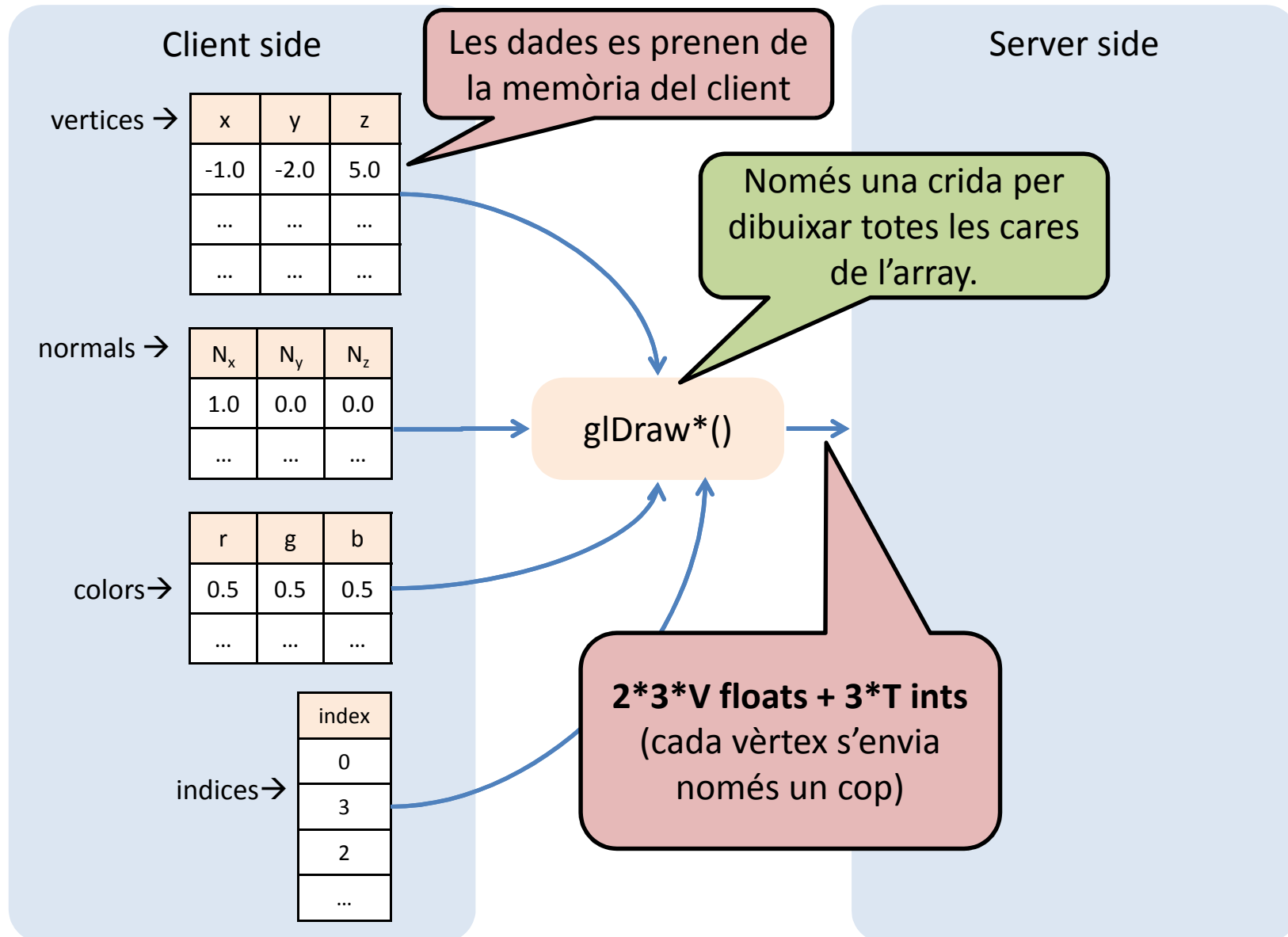


# Vertex Arrays

Objectius:

- Reduir crides a OpenGL
- Enviar (i processar) un cop cada vèrtex

# Vertex Arrays



# Vertex Arrays

`glDrawElements(GL_TRIANGLES, 36, GL_UNSIGNED_INT, indices)`

❶

❷

❸

❹

- ❶ És la primitiva: GL\_TRIANGLES, GL\_QUADS ...
- ❷ És el número d'índexos a l'array (ex. 12 triangles  $\rightarrow 12 \cdot 3 = 36$ )
- ❸ És el tipus dels índexs (normalment GL\_UNSIGNED\_INT)
- ❹ És l'apuntador a l'array amb els índexs (que haurem definit previament)

Quins atributs (normal, color, coords textura...) s'usaran?  
Com s'especifiquen els apuntadors a aquests atributs?



# Vertex Arrays

`glVertexPointer(3, GL_FLOAT, 0, (GLvoid*)vertices)`

❶

❷

❸

❹

- ❶ **size**: Número de coordenades per vèrtex (2,3,4)
- ❷ **type**: Tipus de cada coordenada: GL\_FLOAT o GL\_DOUBLE.
- ❸ **stride**: separació (en bytes) entre l'inici de les coordenades de dos vèrtexs consecutius. Si cada atribut està en un array independent, és 0.
- ❹ **pointer**: Apuntador al array de vèrtexs.

# Vertex Arrays

- La resta d'atributs es defineixen de forma anàloga:

**glVertexPointer**(GLint size, GLenum type, GLsizei stride, const GLvoid\* vertices)

**glNormalPointer**(GLenum type, GLsizei stride, const GLvoid\* normals)

**glColorPointer**(GLint size, GLenum type, GLsizei stride, const GLvoid\* colors)

**glTexCoordPointer**(GLint size, GLenum type, GLsizei stride, const GLvoid\* texcoords)

- **glDrawElements** només farà servir els atributs activats:

**glEnableClientState**(GL\_VERTEX\_ARRAY);

**glEnableClientState**(GL\_NORMAL\_ARRAY);

**glEnableClientState**(GL\_TEXTURE\_COORD\_ARRAY);

**glEnableClientState**(GL\_COLOR\_ARRAY);

# Vertex Arrays - resum

**glVertexPointer**(..., const GLvoid\* vertices)

**glNormalPointer**(..., const GLvoid\* normals)

...

**glEnableClientState**(GL\_VERTEX\_ARRAY);

**glEnableClientState**(GL\_NORMAL\_ARRAY);

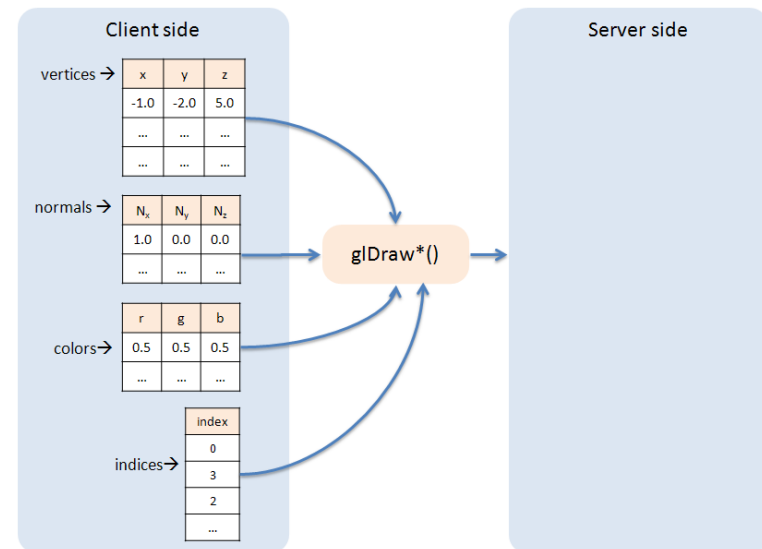
...

**glDrawElements**(GL\_TRIANGLES, 36, GL\_UNSIGNED\_INT, indices)

**glDisableClientState**(GL\_VERTEX\_ARRAY);

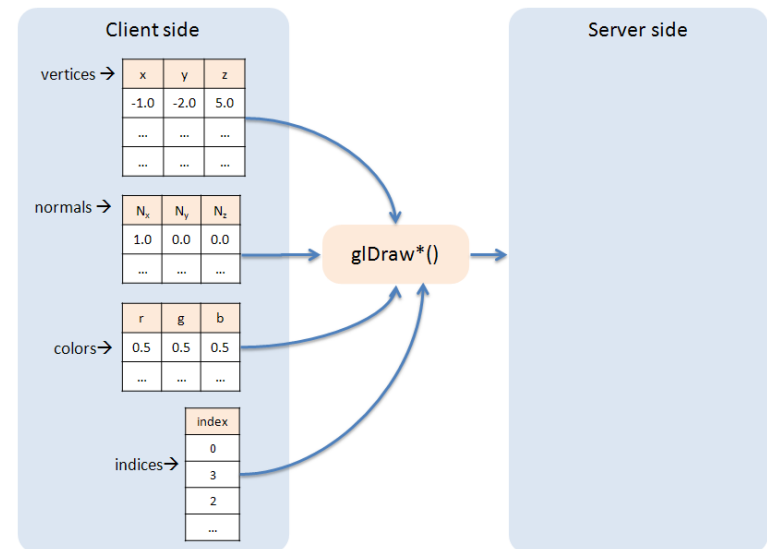
**glDisableClientState**(GL\_NORMAL\_ARRAY);

...



# Vertex Arrays - resum

- Una única crida a funció
- Els vèrtexs s'envien (i processen) un cop (\*)
- Menys flexible que el mode immediat
- Encara cal transferir moltes dades cada frame



# Vertex buffer object

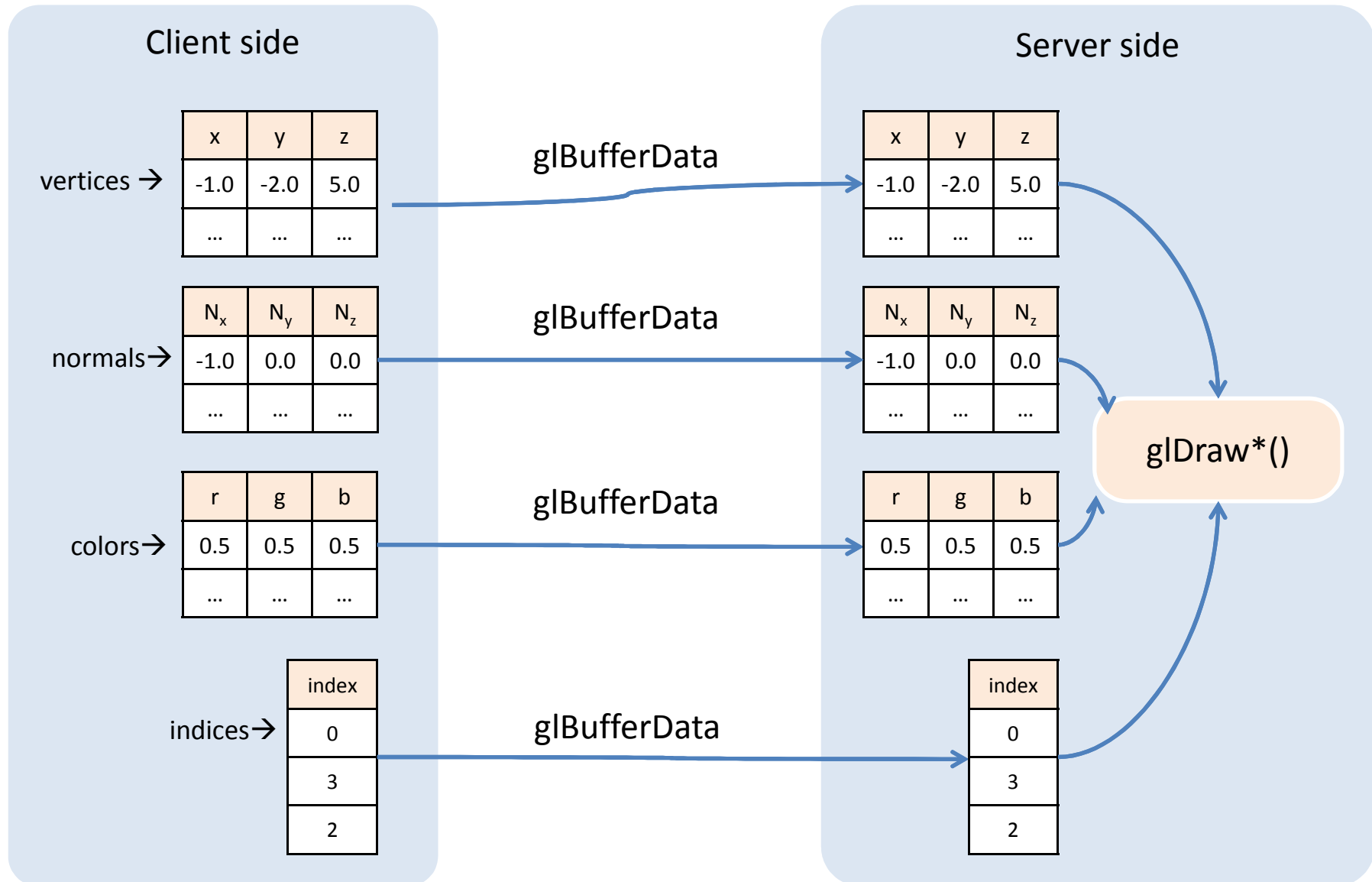
Objectiu:

- Evitar transferir les dades cada frame

Idea:

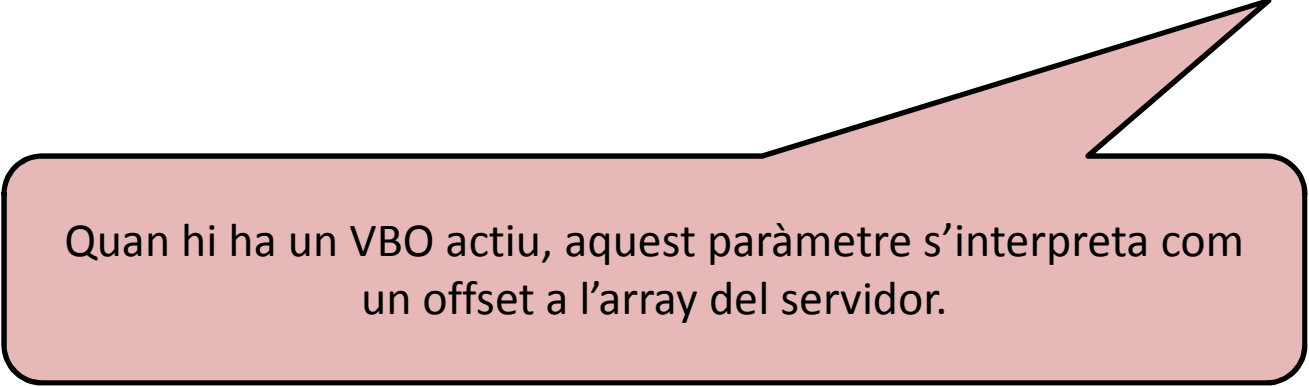
- Emmagatzemar les dades del VA al servidor!

# Vertex buffer object



# Vertex buffer object

```
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, id);  
glDrawElements(GL_TRIANGLES, 36, GL_UNSIGNED_INT, (Glvoid *) 0 )
```



Quan hi ha un VBO actiu, aquest paràmetre s'interpreta com un offset a l'array del servidor.

# Vertex buffer object

Per cada array d'atributs:

GLuint verticesID;

**glGenBuffers**(1, &verticesID);

**glBindBuffer**(GL\_ARRAY\_BUFFER, verticesID);

**glBufferData**(GL\_ARRAY\_BUFFER, sizeof(**vertices**), **vertices**,  
GL\_STATIC\_DRAW);

Per l'array d'índexs:

GLuint indicesID;

**glGenBuffers**(1, &indicesID);

**glBindBuffer**(GL\_ELEMENT\_ARRAY\_BUFFER, indicesID);

**glBufferData**(GL\_ELEMENT\_ARRAY\_BUFFER, sizeof(**indices**),  
**indices**, GL\_STATIC\_DRAW);



# Vertex buffer object - resum

```
glGenBuffers(1, &verticesID);  
glBindBuffer(GL_ARRAY_BUFFER, verticesID);  
glBufferData(GL_ARRAY_BUFFER, ...);  
glGenBuffers(1, &indicesID);  
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, indicesID);  
glBufferData(GL_ELEMENT_ARRAY_BUFFER, ...);
```

Per cada array d'atributs

Per l'array índices

Un cop

```
glBindBuffer(GL_ARRAY_BUFFER, id)  
glVertexPointer(3, GL_FLOAT, 0, (GLvoid*) 0);  
glEnableClientState(GL_VERTEX_ARRAY);  
  
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, id)  
glDrawElements(GL_TRIANGLES, 36, ...)  
  
glDisableClientState(GL_VERTEX_ARRAY);  
glBindBuffer(GL_ARRAY_BUFFER, 0);  
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, 0);
```

Per cada array d'atributs

Rendering

Per cada atribut

Cada frame

# Vertex Buffer Objects - resum

- Una única crida a funció
- Els vèrtexs s'envien (i processen) un cop (\*)
- Les dades es transfereixen al servidor
- Menys flexible que el mode immediat

