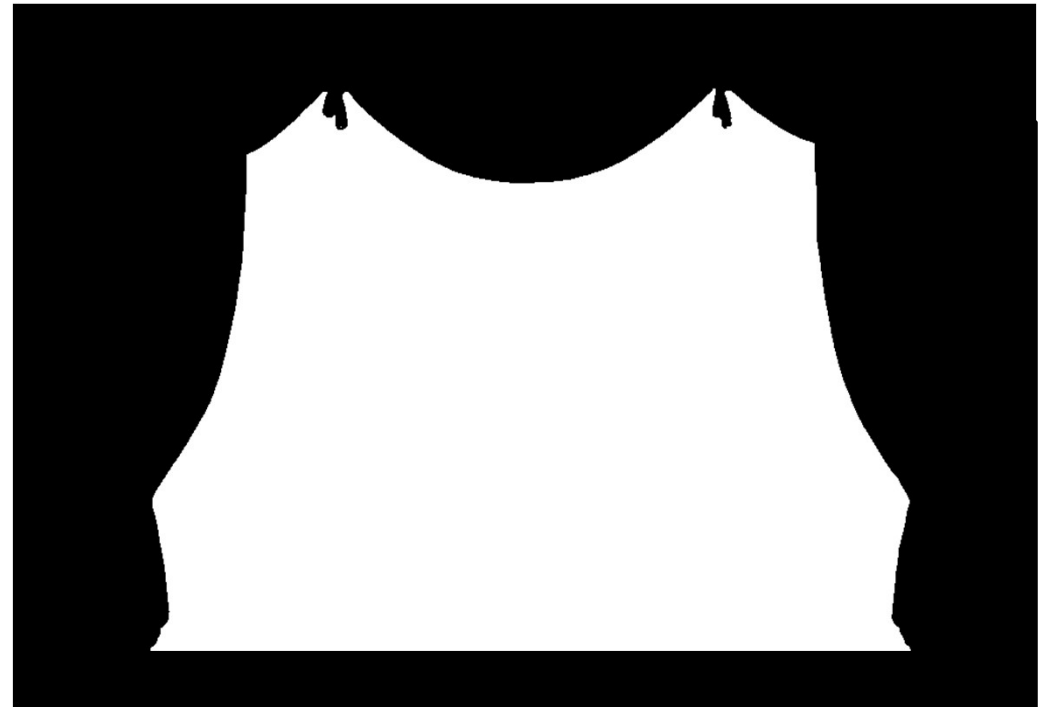


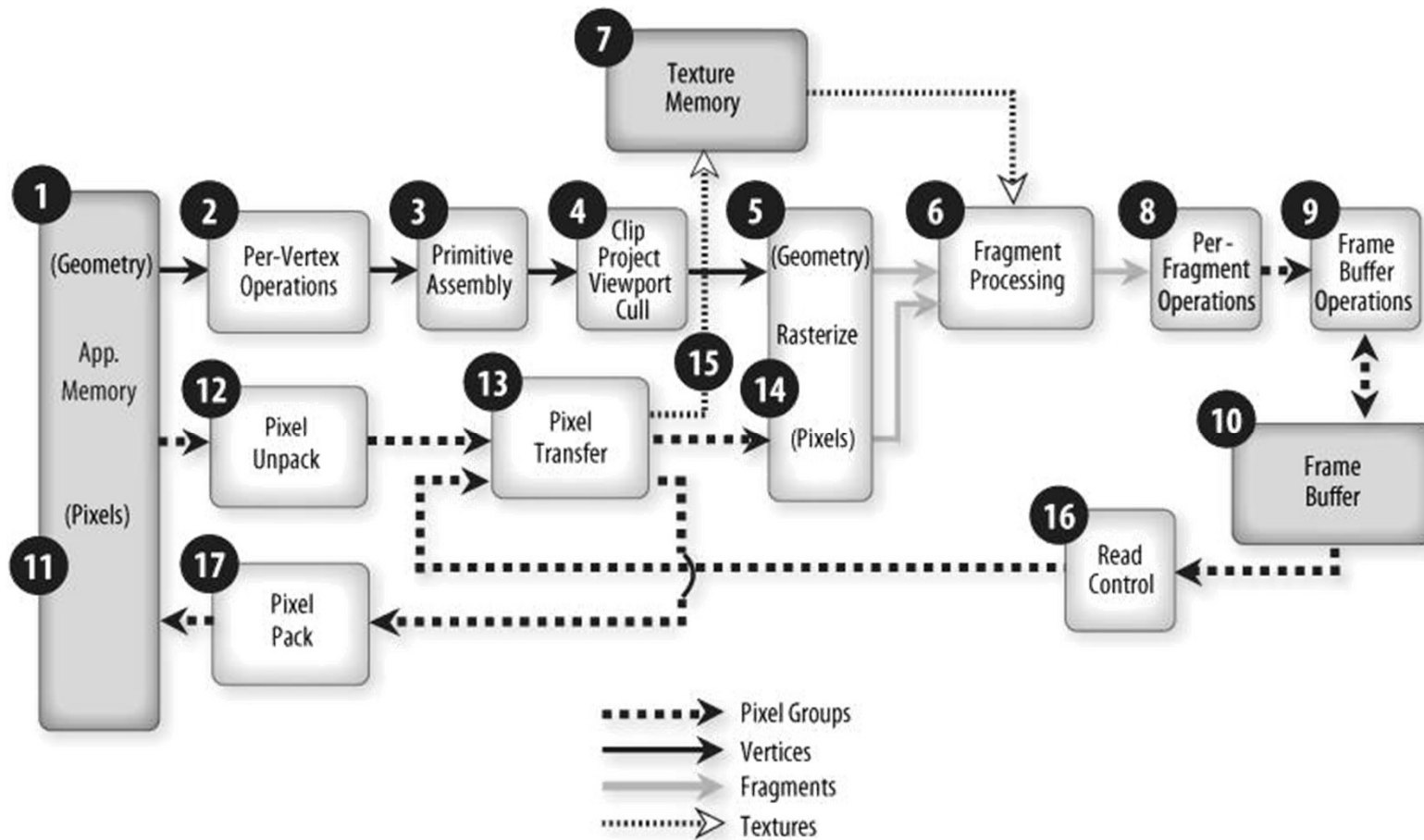
Stencil Buffer

Professors de Gràfics

Stencil buffer



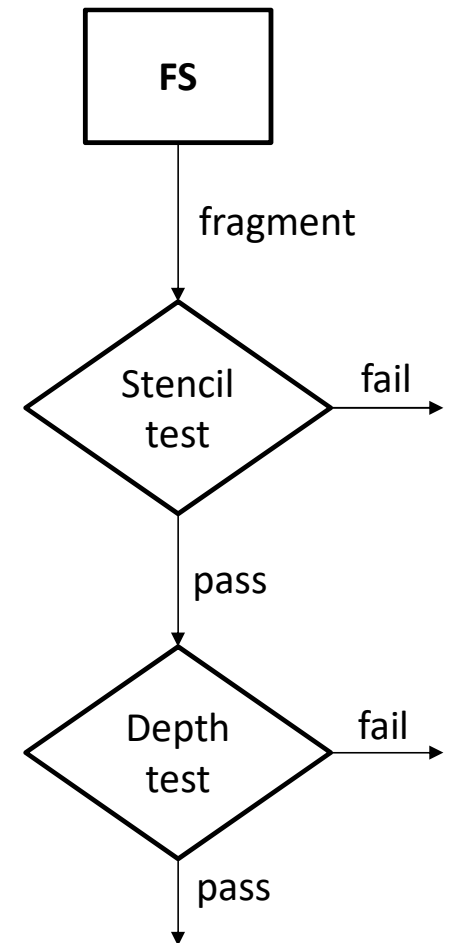
Pipeline OpenGL



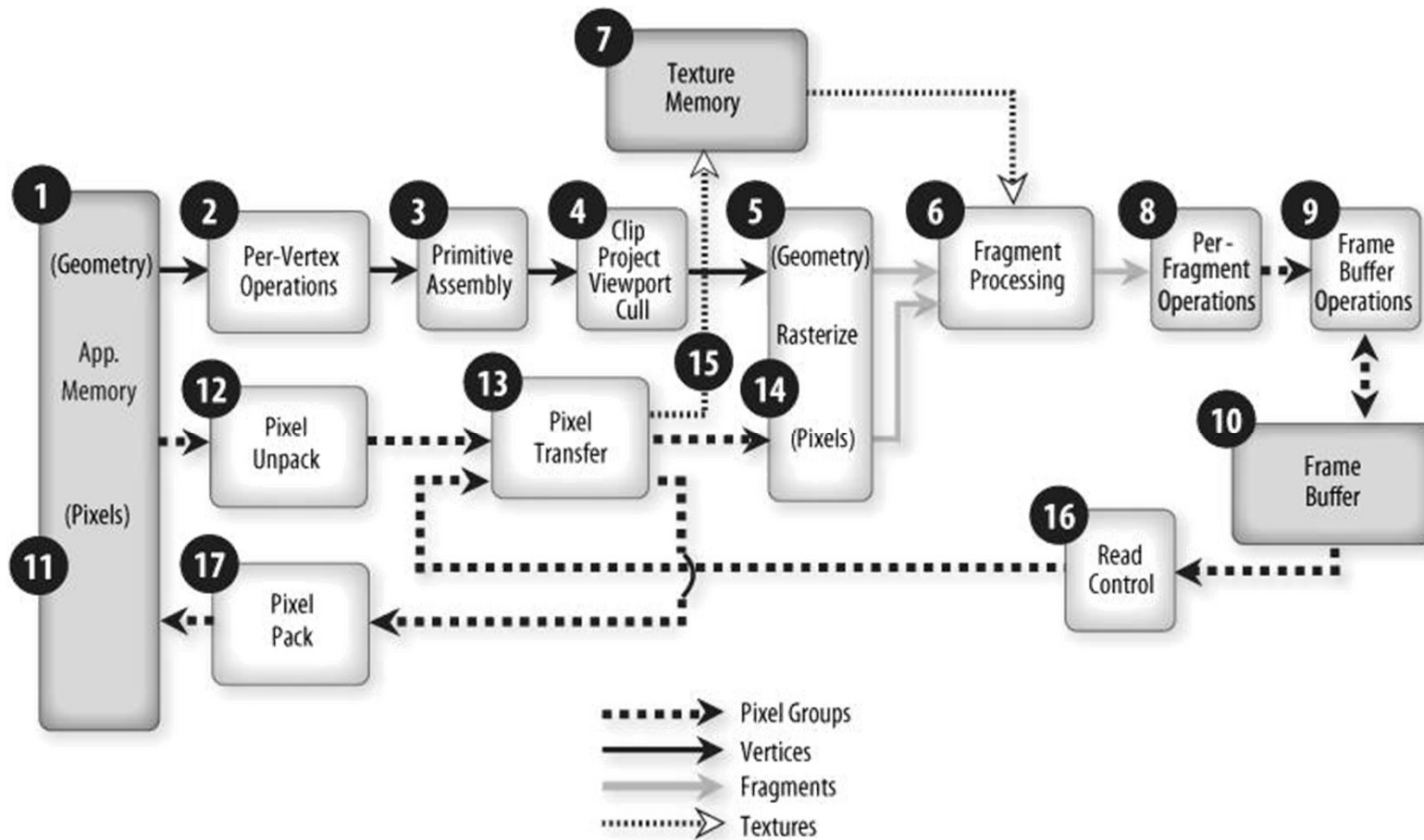
Pipeline OpenGL

8. Per-fragment operations (“raster operations”)

- Pixel ownership
- Scissor test
- Alpha test
- **Stencil test**
- **Depth test (test Z-buffer)**
- Blending
- Dithering
- Logical Ops (glLogicOp)



Pipeline OpenGL



Pipeline OpenGL

9. Frame buffer operations

- Es modifiquen els buffers que s'hagin escollit amb `glDrawBuffers`
- Es veu afectada per **`glColorMask`**, **`glDepthMask`**...

Stencil buffer

El stencil buffer guarda, per cada pixel, un enter entre $0..2^n-1$.

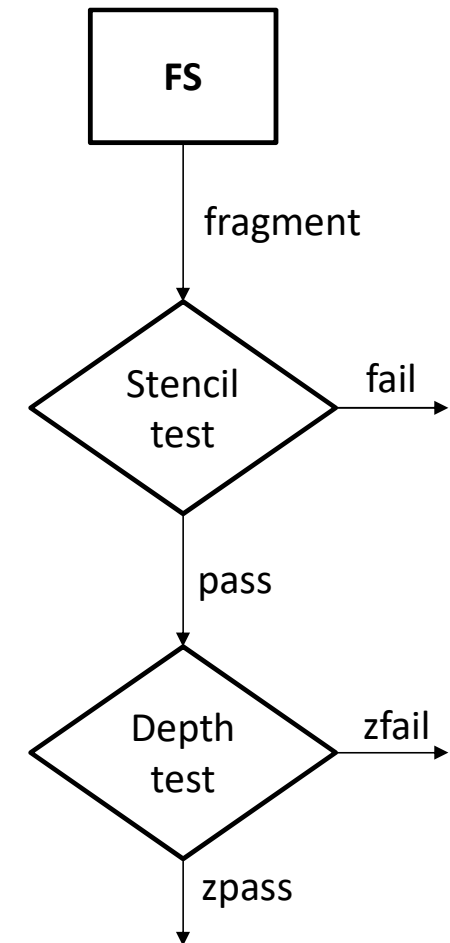
- Demanar una finestra OpenGL amb stencil:
 - `QOpenGLformat f;`
 - `f.setStencil(true);`
 - `QOpenGLformat::setDefaultFormat(f);`
- Obtenir el núm. de bits del stencil:
 - `glGetIntegerv(GL_STENCIL_BITS, &nbits);`
- Esborrar stencil (no li afecta `glStencilFunc()`, sí `glStencilMask`):
 - `glClearStencil(0);`
 - `glClear(GL_STENCIL_BUFFER_BIT);`

Stencil buffer

- Establir el test de comparació:
 - glEnable(GL_STENCIL_TEST);
 - **glStencilFunc**(comparació, valorRef, mask)
 - Comparació pot ser: GL_NEVER, GL_ALWAYS, GL_LESS...
 - Ex: GL_LESS: $(\text{valorRef} \& \text{mask}) < (\text{valorStencil} \& \text{mask})$
- Operacions a fer a stencil buffer segons el resultat del test:
 - **glStencilOp**(fail, zfail, zpass)
 - fail -> op. a fer quan el fragment no passa el test de stencil
 - Zfail -> op. a fer quan passa stencil, pero no passa z-buffer
 - Zpass -> op. a fer quan passa stencil i passa z-buffer
 - Cadascú dels paràmetres anteriors pot ser:
 - GL_KEEP, GL_ZERO, GL_INCR, GL_DECR, GL_INVERT
 - GL_REPLACE (usa valor refèrència)

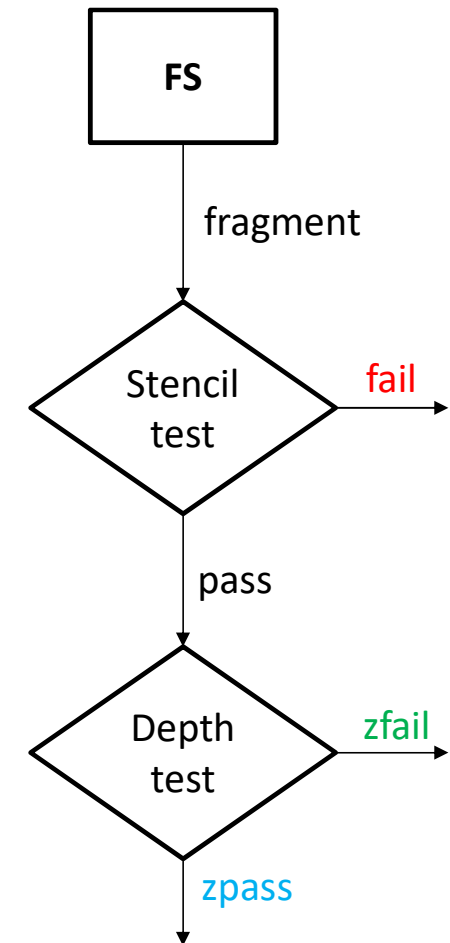
Stencil buffer

- Establir el test de comparació:
 - `glEnable(GL_STENCIL_TEST);`
 - **`glStencilFunc`**(comparació, valorRef, mask)
 - Comparació pot ser: `GL_NEVER`, `GL_ALWAYS`, `GL_LESS`...
 - Ex: `GL_LESS`: $(\text{valorRef} \& \text{mask}) < (\text{valorStencil} \& \text{mask})$
- Operacions a fer a stencil buffer segons el resultat del test:
 - **`glStencilOp`**(fail, zfail, zpass)
 - fail -> op. a fer quan el fragment no passa el test de stencil
 - Zfail -> op. a fer quan passa stencil, pero no passa z-buffer
 - Zpass -> op. a fer quan passa stencil i passa z-buffer
 - Cadascú dels paràmetres anteriors pot ser:
 - `GL_KEEP`, `GL_ZERO`, `GL_INCR`, `GL_DECR`, `GL_INVERT`
 - `GL_REPLACE` (usa valor refèrència)



Stencil buffer

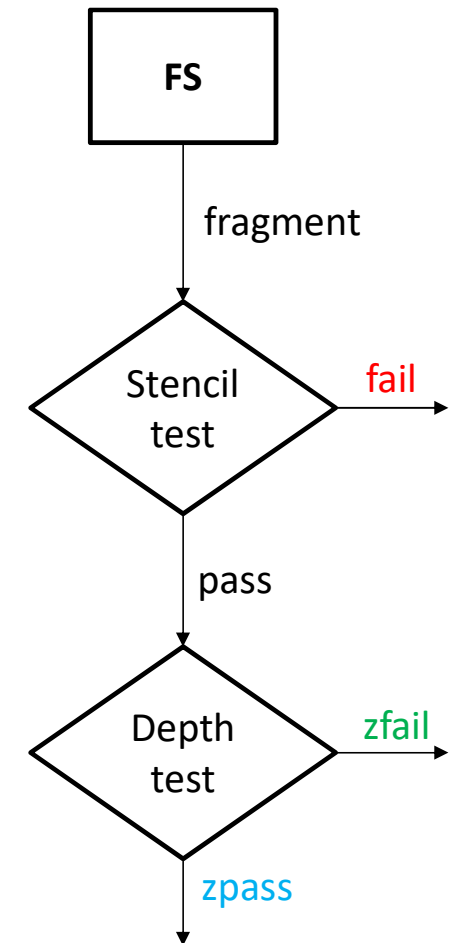
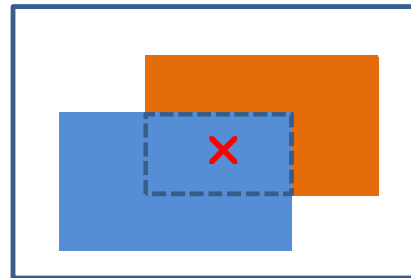
```
glEnable(GL_STENCIL_TEST);  
glStencilFunc(GL_ALWAYS, 1, 255);  
glStencilOp(GL_KEEP, GL_KEEP, GL_REPLACE);  
drawQuad()
```



Stencil buffer

```
glEnable(GL_STENCIL_TEST);  
glStencilFunc(GL_ALWAYS, 1, 255);  
glStencilOp(GL_KEEP, GL_KEEP, GL_REPLACE);  
drawQuad();
```

```
glStencilFunc(GL_EQUAL, 0, 255);  
glStencilOp(GL_KEEP, GL_KEEP, GL_KEEP);  
drawQuad();
```



Stencil buffer

```
glEnable(GL_STENCIL_TEST);  
glStencilFunc(GL_ALWAYS, 1, 255);  
glStencilOp(GL_KEEP, GL_KEEP, GL_REPLACE);  
drawQuad();
```

```
glStencilFunc(GL_EQUAL, 0, 255);  
glStencilOp(GL_KEEP, GL_KEEP, GL_KEEP);  
drawQuad();
```

