

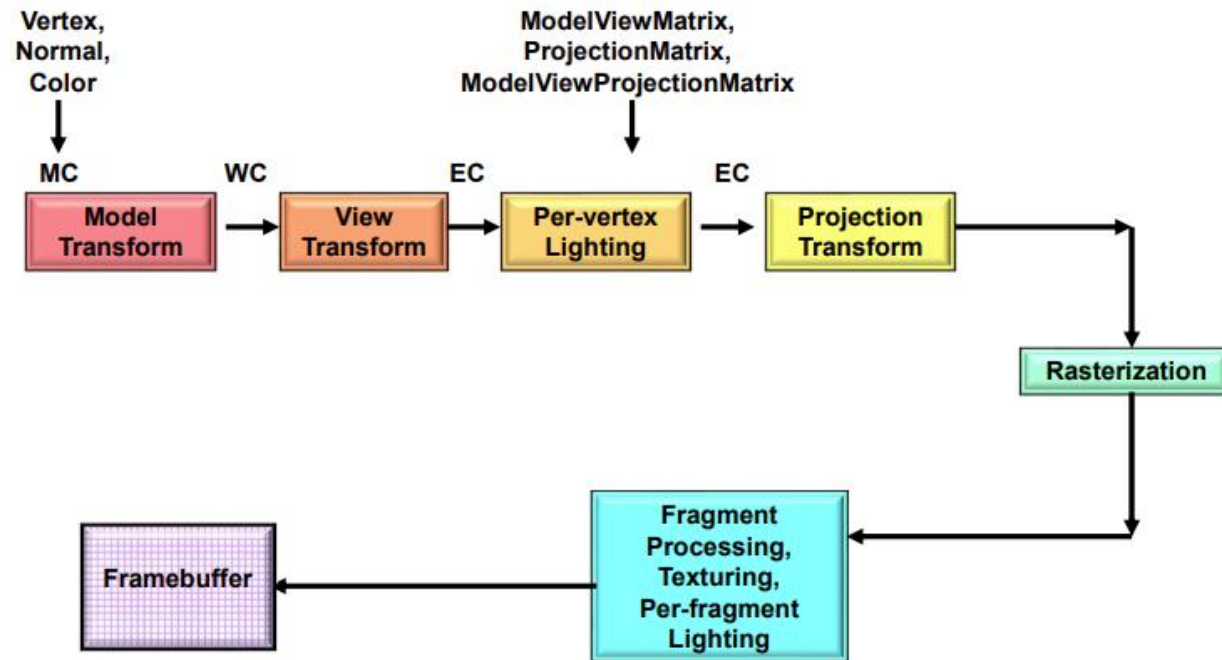
Computação Gráfica

Shaders e GLSL

RAFAEL TEIXEIRA SOUSA

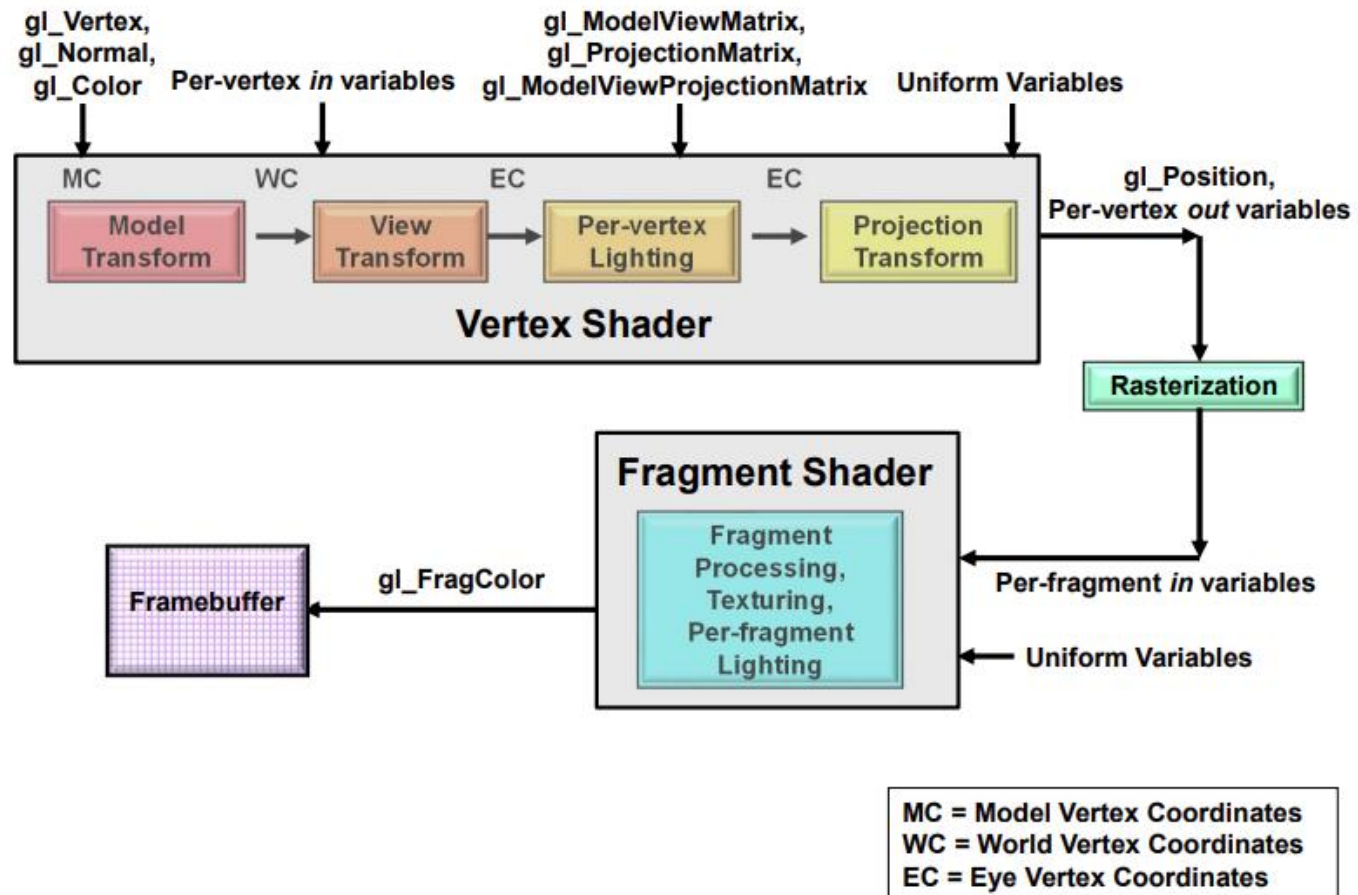


Pipeline tradicional de iluminação

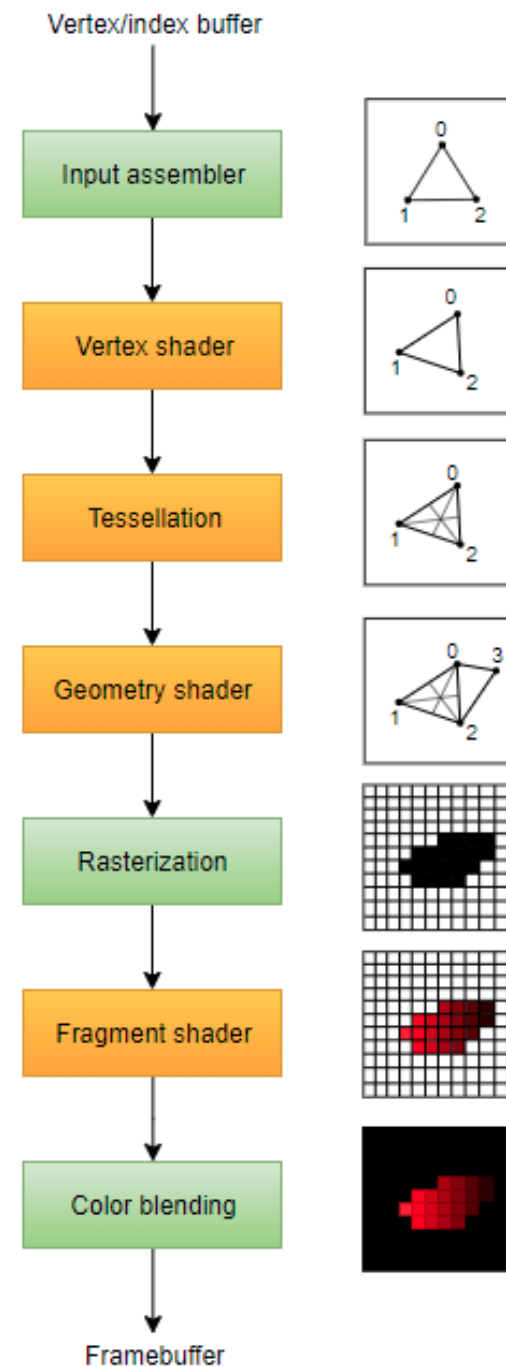
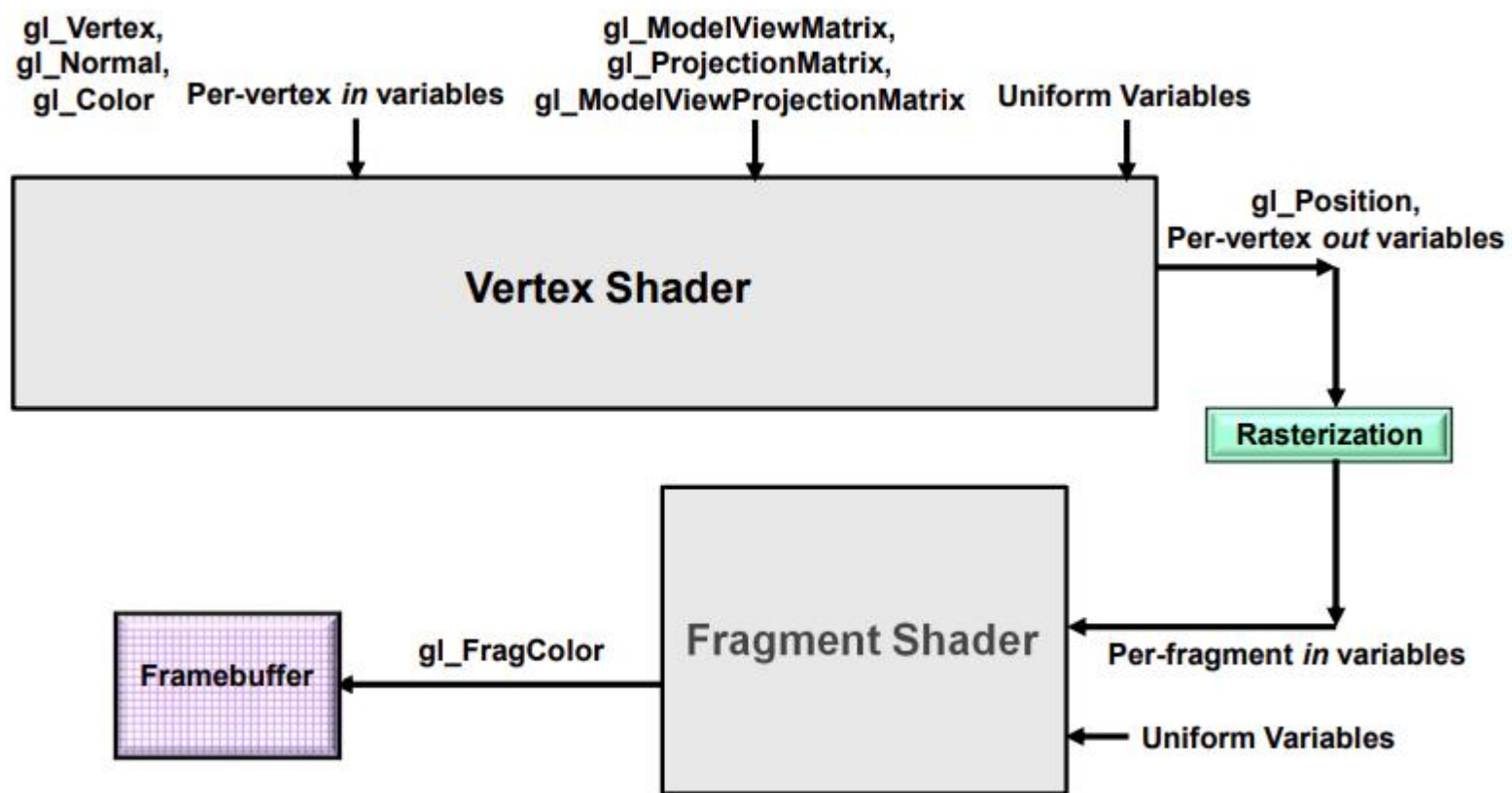


MC = Model Vertex Coordinates
WC = World Vertex Coordinates
EC = Eye Vertex Coordinates

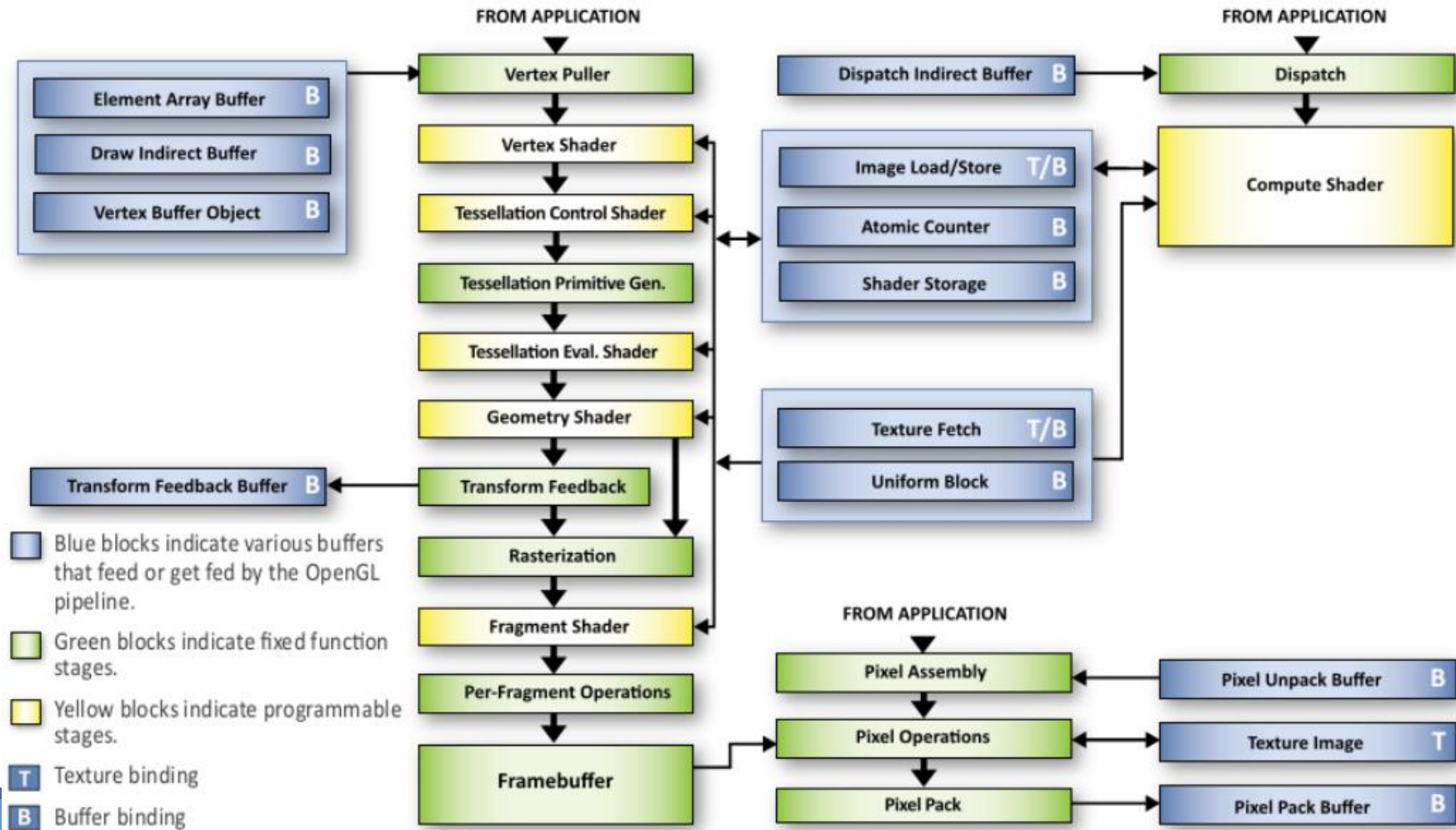
Pipeline com shaders



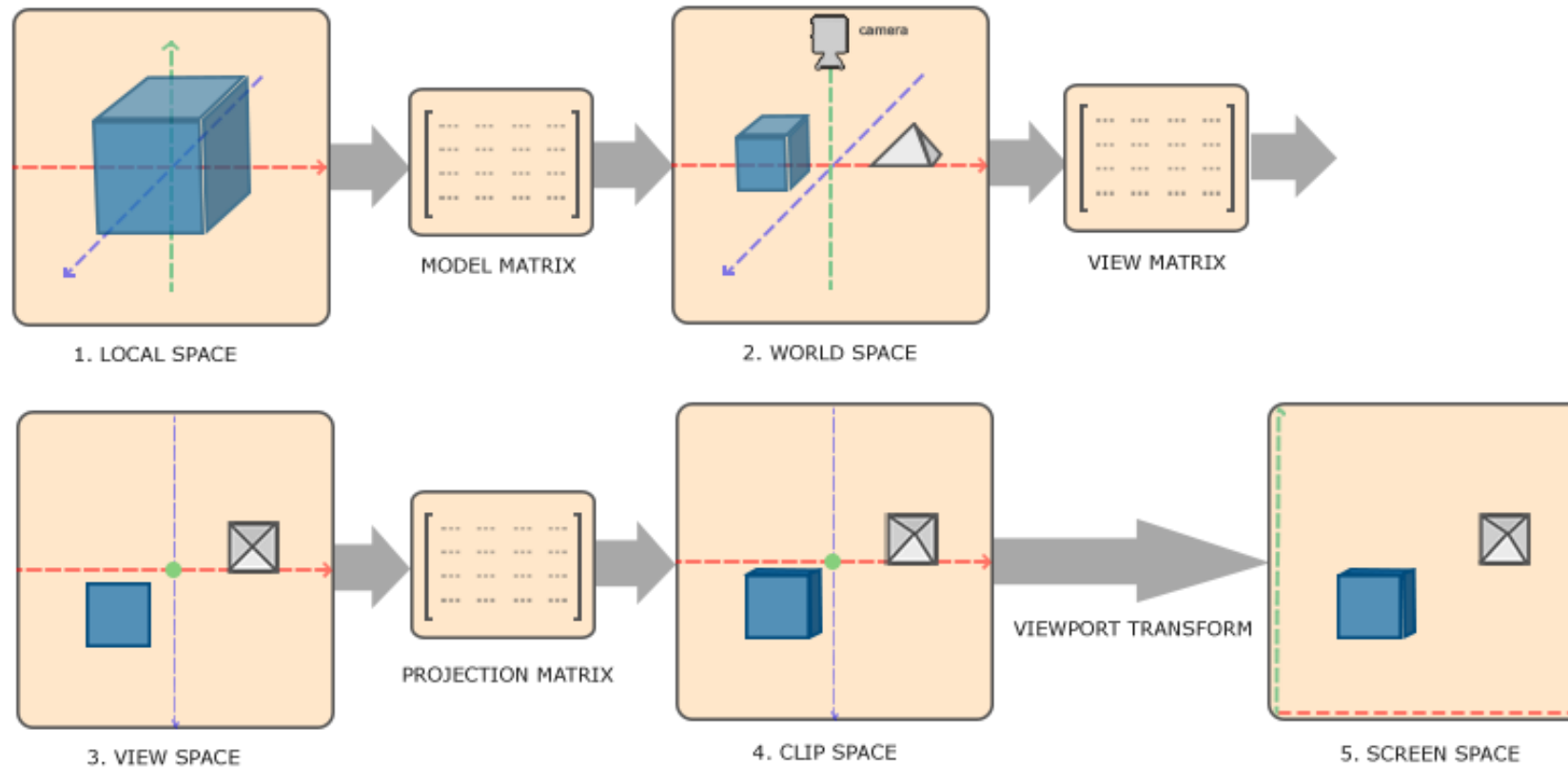
Pipeline com shaders



Pipeline moderno



Espaços



GLSL – OpenGL Shading Language

Baseada em C

Funciona em múltiplos hardwares

Várias versões (atual 4.60)

OpenGL ES

- Versão mobile

WebGL

- Versão web

Alternativas

DirectX High-Level Shader Language (HLSL)

- PC e Xbox

PlayStation Shader Language (PSSL)

- PS4

GLSL variáveis

attribute:

- Variáveis por vértices passadas para o vertex shader

uniform:

- Constantes globais passadas para todos os shaders

in/out:

- Variáveis passadas entre shaders

GLSL tipos

int, ivec2, ivec3, ivec4

float, vec2, vec3, vec4

mat2, mat3, mat4

bool bvec2, bvec3, bvec4

- Vetores são acessados com [index], .rgba, .xyzw, ou .stpq
- Operações vetoriais são SIMD:
 vec4 a, b, c;
 a = b + c;
- Controle de fluxo igual a C

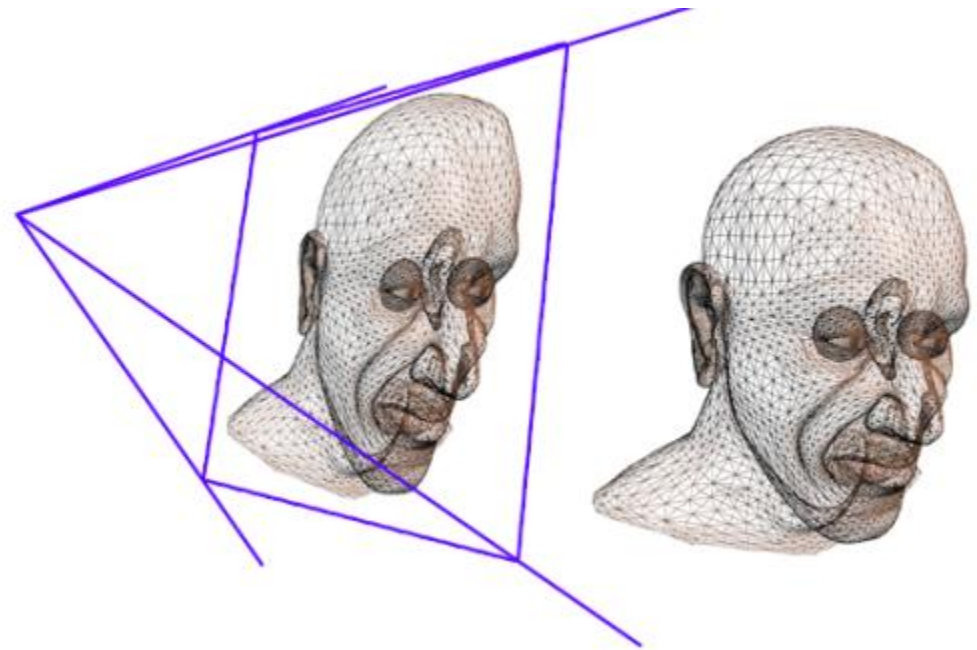
Diferenças de C

- Sem mudanças de tipos, em troca se usa construtores: “int i = int(x);”
- Sem ponteiros
- Sem strings

Vertex Shaders

Vertex shaders substituem as operações de:

- Transformações de vértices
- Transformações de normais
- Normalização de normais
- Computação de iluminação



Variáveis embutidas no Vertex shader

vec4 gl_Vertex

vec3 gl_Normal

vec4 gl_Color

vec4 gl_MultiTexCoord0

mat4 gl_ModelViewMatrix

mat4 gl_ProjectionMatrix

mat4 gl_ModelViewProjectionMatrix

mat3 gl_NormalMatrix

vec4 gl_Position (posição de saída)

Fragment Shaders

Fragment shaders substituem as operações de:

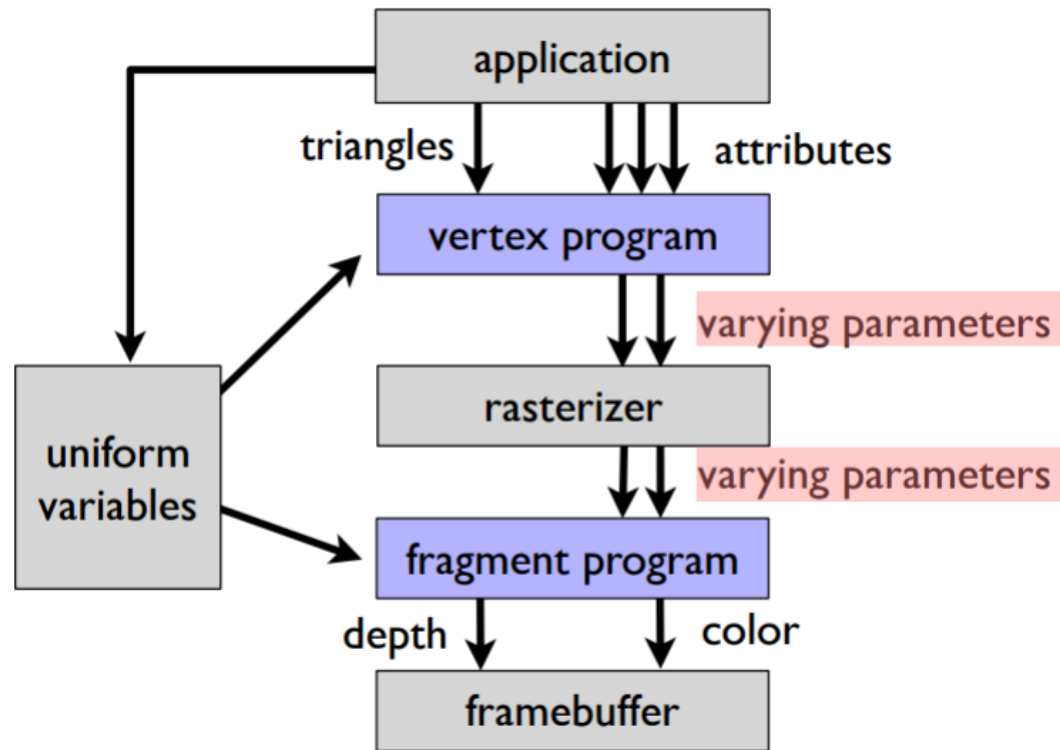
- Definição de cor a cada fragmento (pixel)
- Textura
- Blending

Variável embutida:

in vec4 gl_FragCoord (posição na tela)

out vec4 gl_FragColor (cor de saída)

Variáveis no pipeline



Exemplo

Vertex shader:

```
#version 330 compability

void main(){
    gl_Position = gl_ModelViewProjection * gl_Vertex;
}
```

Fragment shader:

```
#version 330 compability

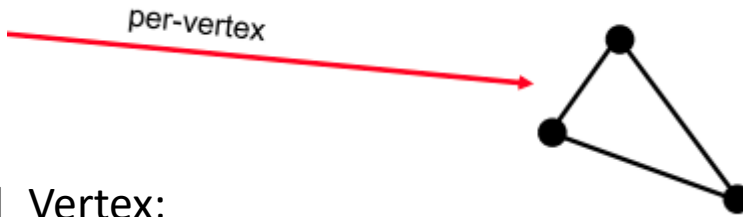
void main(){
    gl_FragColor = vec4(1., 0. ,0. , 1.);
}
```


Exemplo

Vertex shader:

```
#version 330 compability
out vec3 vColor;

void main(){
    vec4 pos = gl_Vertex;
    vColor = pos.xyz;
    gl_Position = gl_ModelViewProjection * gl_Vertex;
}
```



Fragment shader:

```
#version 330 compability
in vec3 vColor;

void main(){
    gl_FragColor = vec4(vColor , 1.);
}
```

