



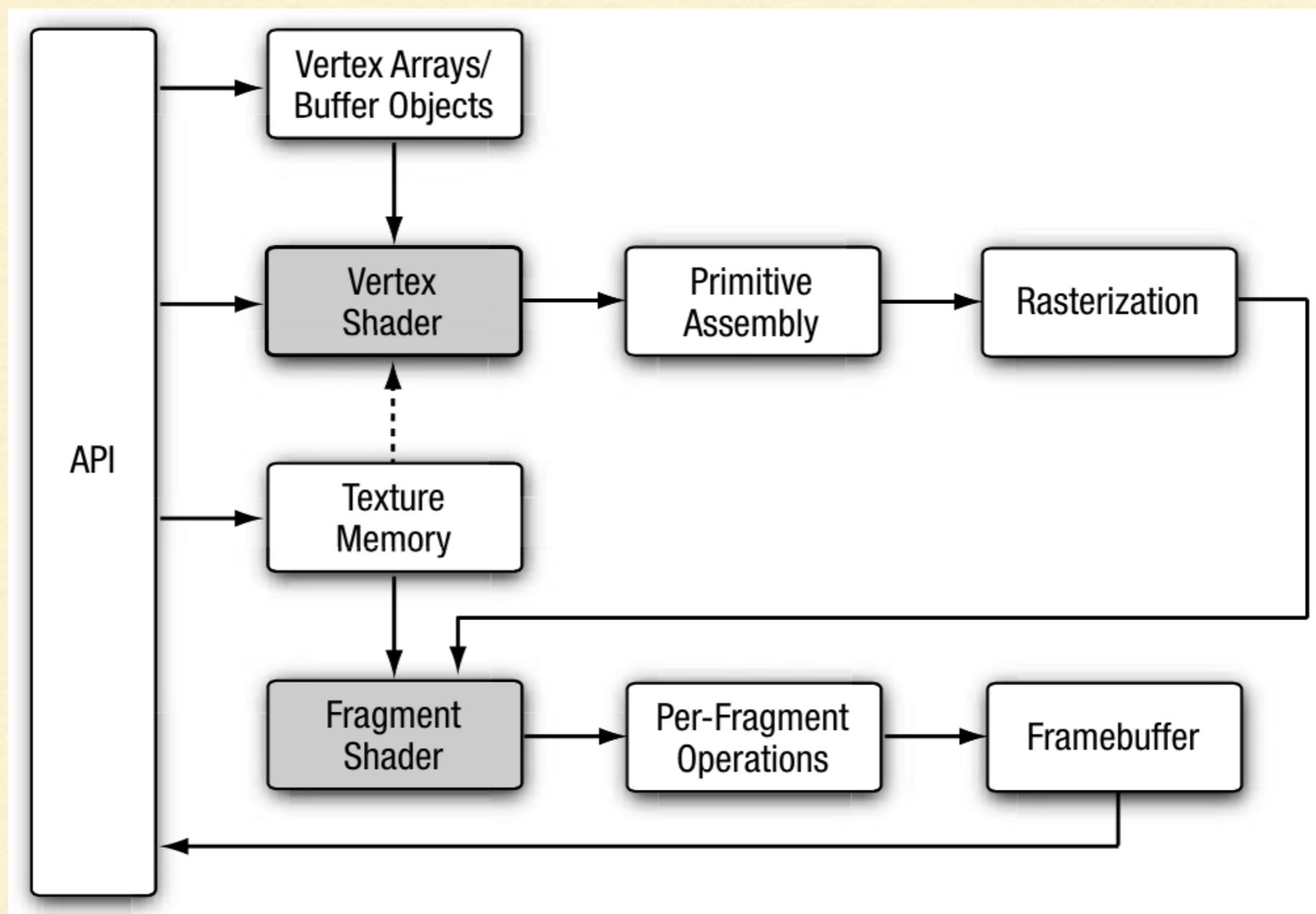
OPENGL ES 2.0

А.Г. Горских

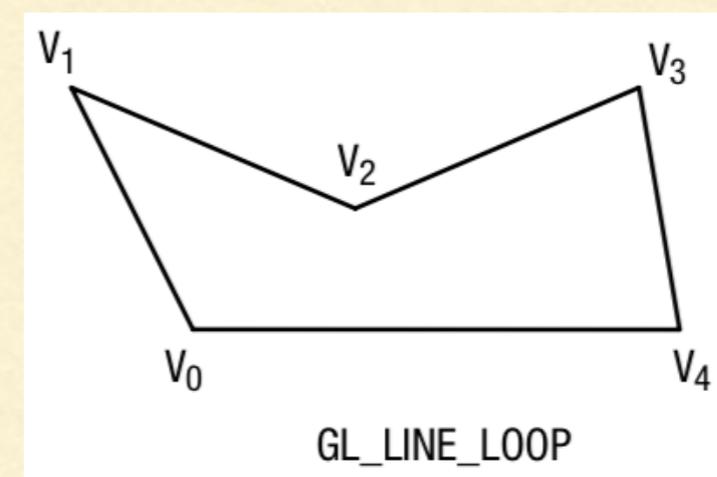
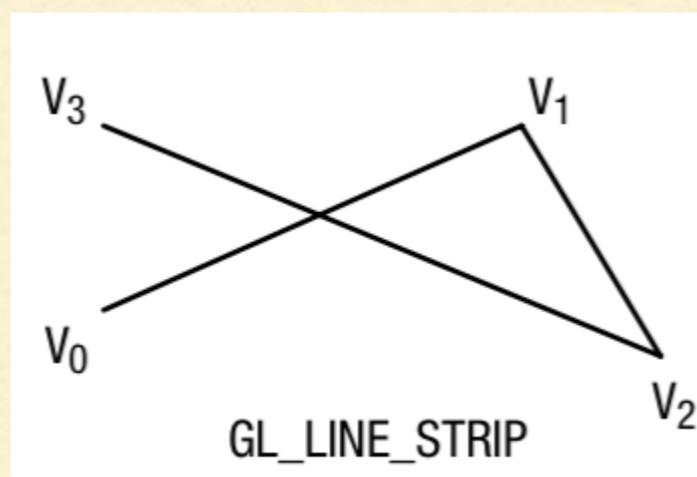
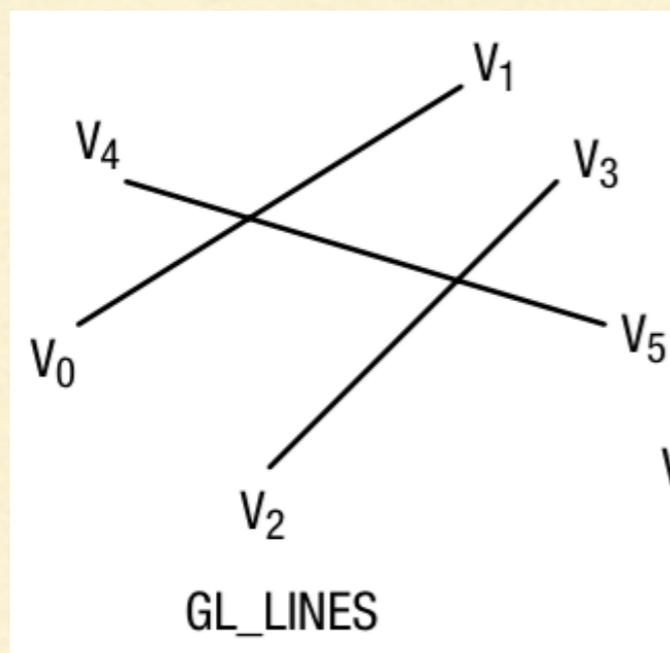
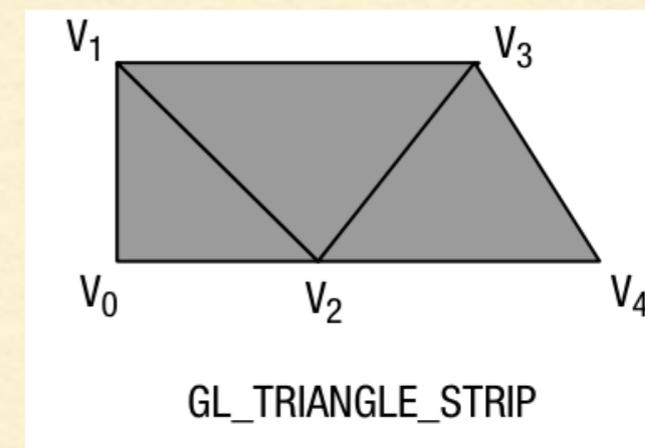
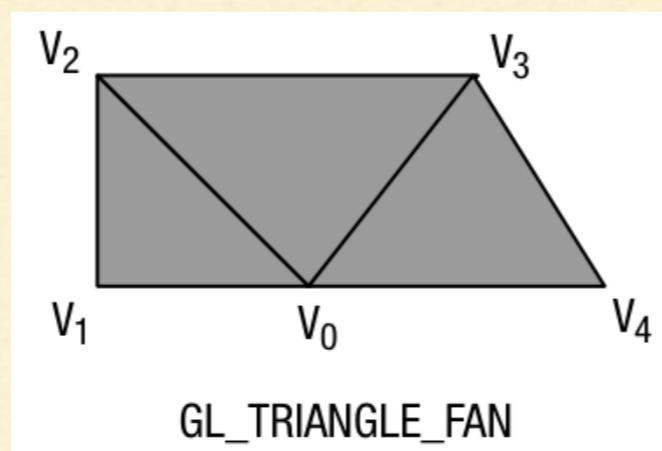
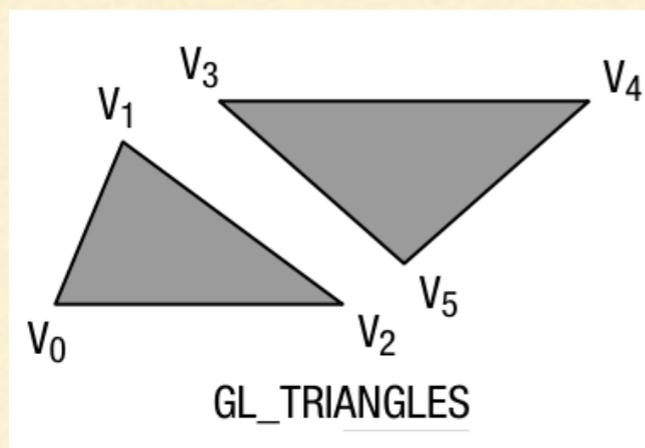
OPENGL

- OpenGL — спецификация, определяющая независимый от языка программирования платформонезависимый программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику.

АРХИТЕКТУРА OPENGL ES 2.0



ПРИМИТИВЫ OpenGL ES 2.0



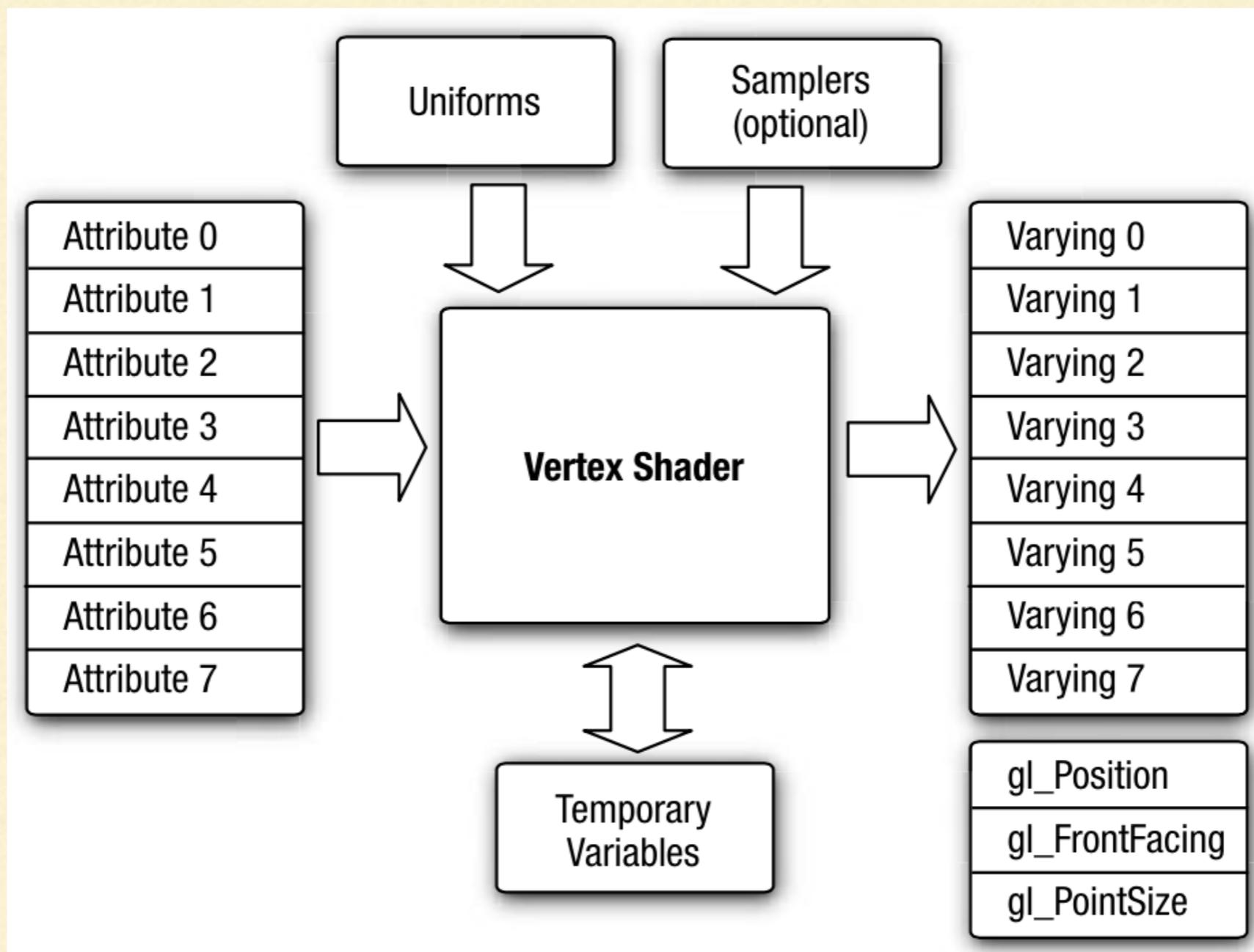
ШЕЙДЕРЫ

- Шейдер – это программа для одной из ступеней графического конвейера, используемая в трёхмерной графике для определения окончательных параметров объекта или изображения
- GLSL – язык программирования шейдеров

РАЗНОВИДНОСТИ ШЕЙДЕРОВ

- Вершинные шейдеры
- Пиксельные шейдеры
- Геометрические шейдеры
- Шейдеры тесселяции
- Вычислительные шейдеры
- UberShader
- ...

ВЕРШИННЫЕ ШЕЙДЕРЫ



ПРИМЕР ВЕРШИННОГО ШЕЙДЕРА

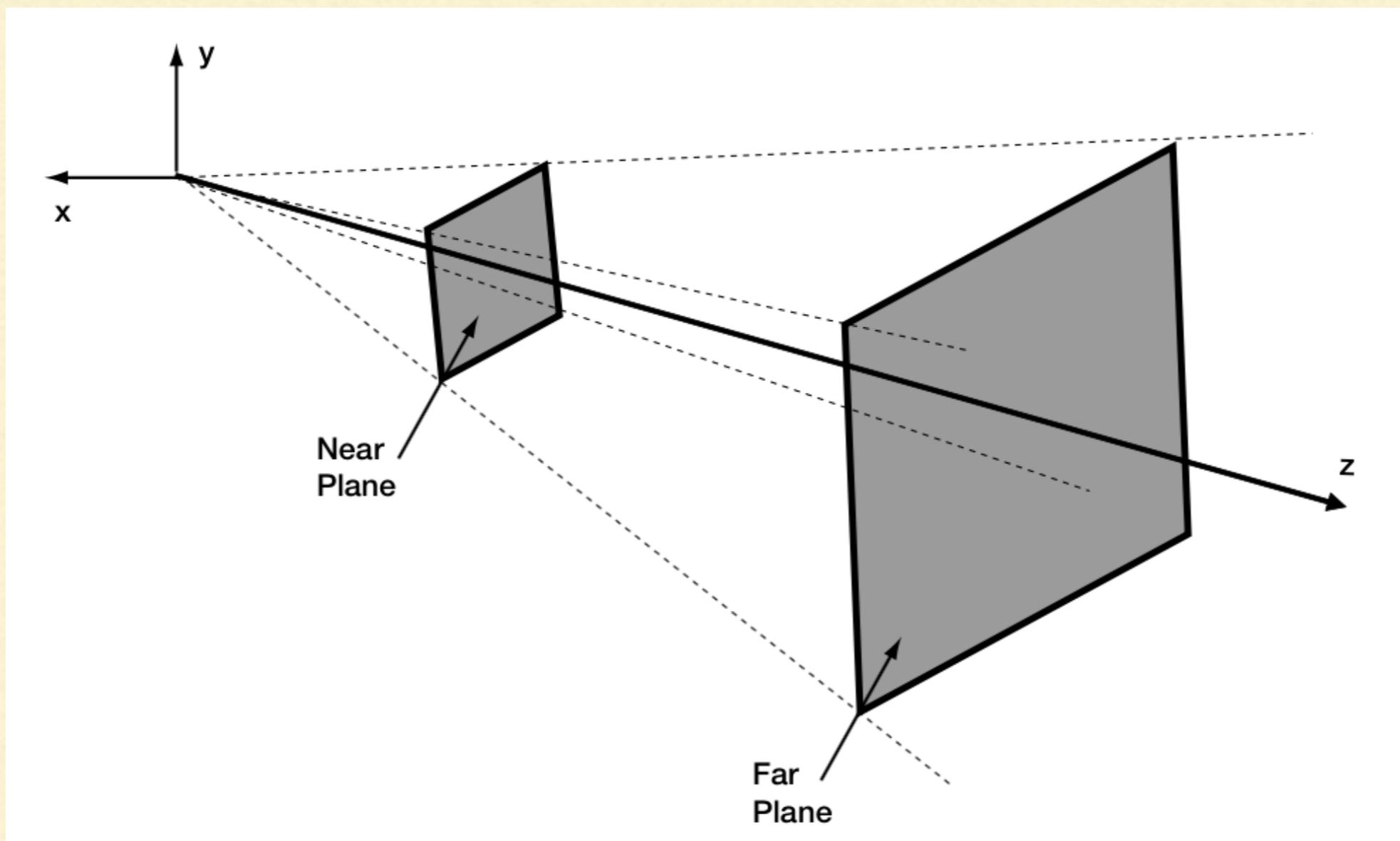
```
// uniforms used by the vertex shader
uniform mat4 u_mvp_matrix; // matrix to convert P from
                            // model space to clip space.

// attributes input to the vertex shader
attribute vec4 a_position; // input position value
attribute vec4 a_color;    // input vertex color

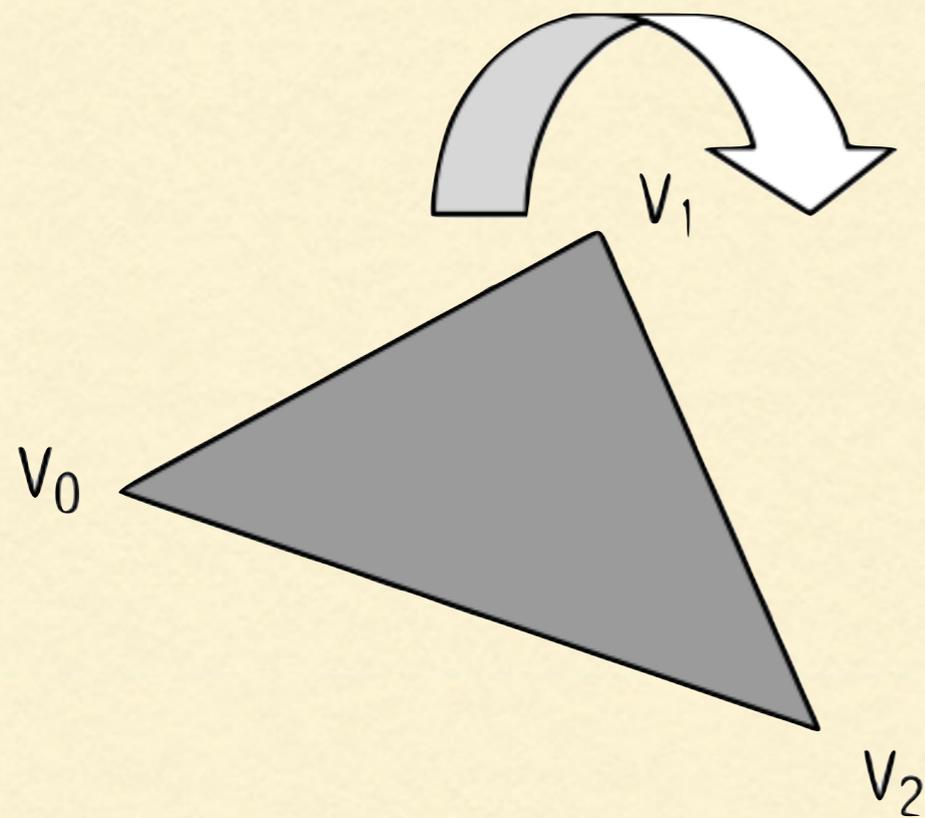
// varying variables – input to the fragment shader
varying vec4 v_color;     // output vertex color

void main()
{
    v_color = a_color;
    gl_Position = u_mvp_matrix * a_position;
}
```

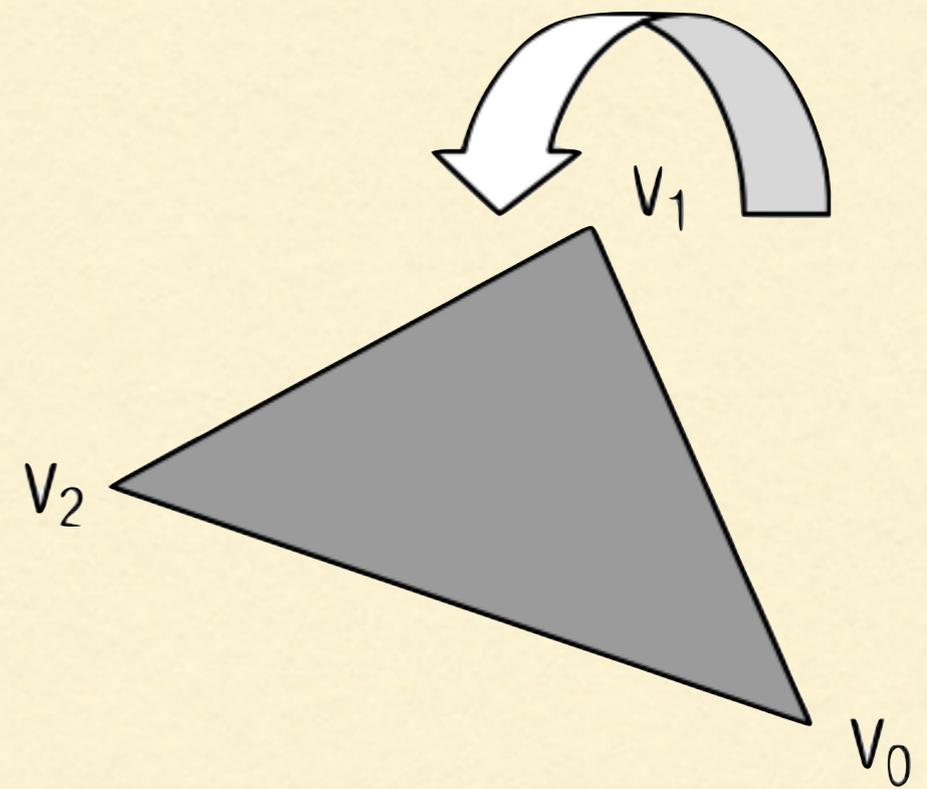
ОТСЕЧЕНИЕ



ОТБРАСЫВАНИЕ ТРЕУГОЛЬНИКОВ



Clockwise (CW)
Orientation

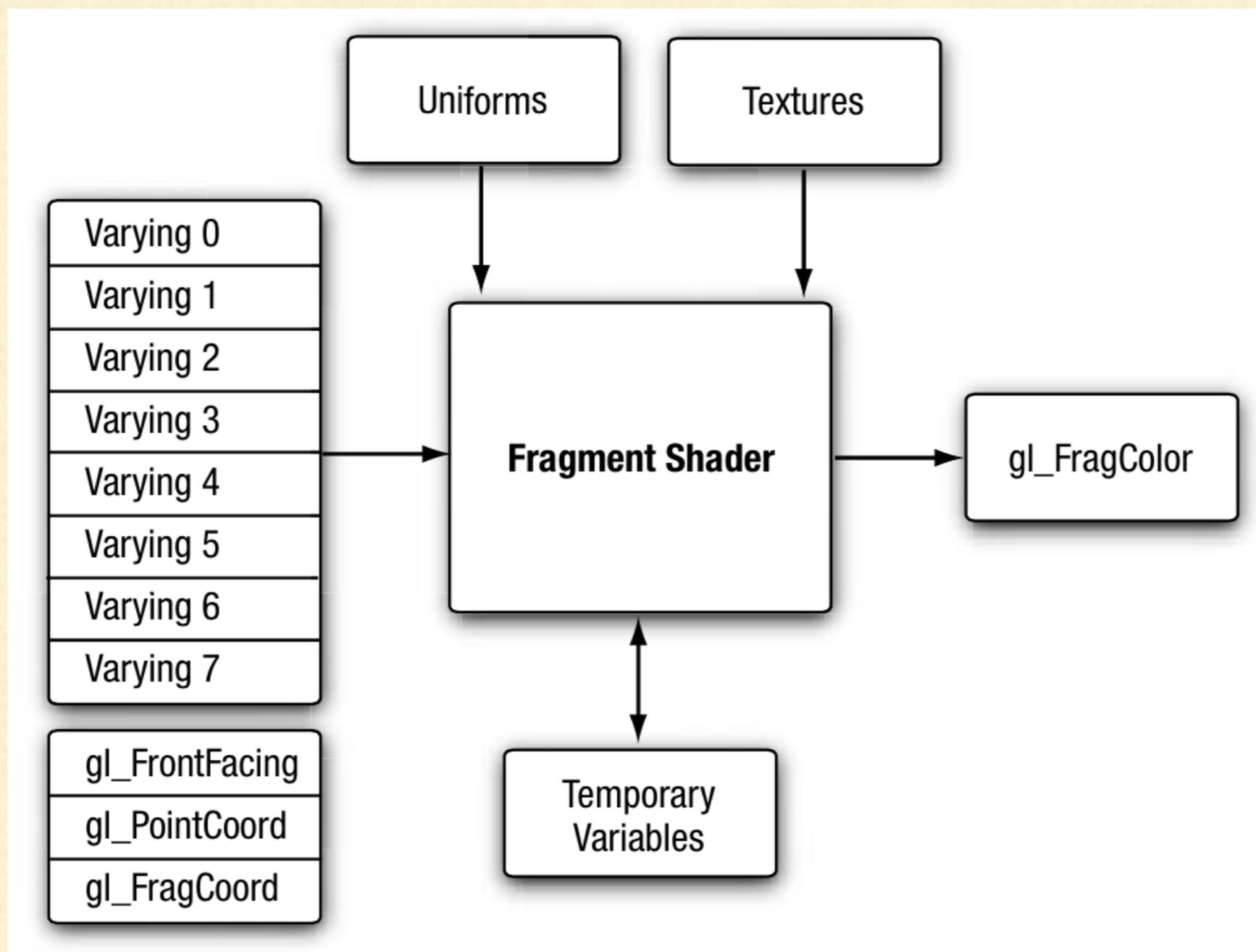


Counter-Clockwise (CCW)
Orientation

РАСТЕРИЗАЦИЯ

- Растеризация спрайтов
- Растеризация линий
- Растеризация треугольников
- Алгоритм Брезенхэма

ПИКСЕЛЬНЫЕ ШЕЙДЕРЫ



ПРИМЕР ПИКсельНОГО ШЕЙДЕРА

```
varying vec3 normal;
varying vec3 vertex_to_light_vector;

void main()
{
    // Defining The Material Colors
    const vec4 AmbientColor = vec4(0.1, 0.0, 0.0, 1.0);
    const vec4 DiffuseColor = vec4(1.0, 0.0, 0.0, 1.0);

    // Scaling The Input Vector To Length 1
    vec3 normalized_normal = normalize(normal);
    vec3 normalized_vertex_to_light_vector = normalize(vertex_to_light_vector);

    // Calculating The Diffuse Term And Clamping It To [0;1]
    float DiffuseTerm = clamp(dot(normalized_normal, normalized_vertex_to_light_vector), 0.0, 1.0);

    // Calculating The Final Color
    gl_FragColor = AmbientColor + DiffuseColor * DiffuseTerm;
}
```

ФРЕЙМБУФЕР

- Фреймбуфер – реальное или виртуальное электронное устройство, или область памяти для кратковременного хранения одного или нескольких кадров в цифровом виде перед его отправкой на устройство видеовывода
- Блиттинг – быстрое копирование и наложения фрагментов изображений в памяти

ССЫЛКИ

- <http://opengles-book.com/es2/>
- <http://habrahabr.ru/post/248153/>
- <http://habrahabr.ru/post/248159/>
- <http://habrahabr.ru/post/248179/>
- https://www.khronos.org/opengles/2_X/