

Лекция 7. Тесты глубины и трафарета

Илья Макаров

ИТМО

26 октября 2022

Санкт-Петербург

Буфер глубины

Хранит определенную информацию для каждого фрагмента.

- Обычно, имеет размеры совпадающие с размерами буфера цвета.
- Буфер глубины для FBO по умолчанию создается оконной системой и хранит значения в виде 16, 24 или 32 битных числе.
- В большинстве систем по умолчанию создается буфер с точностью 24 бита.

Буфер глубины

При включенном тесте глубины OpenGL

```
glEnable(GL_DEPTH_TEST)
```

- производит проверку глубины каждого обрабатываемого фрагмента относительно данных, хранимых в буфере;
- при прохождении теста содержимое буфера будет обновлено значением глубины обрабатываемого фрагмента;
- при провале теста – хранимое значение останется прежним, а фрагмент отбрасывается.

Ранний тест глубины

Тест глубины производится в экранном пространстве после выполнения фрагментного шейдера. В GLSL:

- **gl_FragCoord** в коде фрагментного шейдера хранит соответствующие координаты;
- **(x, y)** - положение фрагмента;
- **z** - значение для сравнения глубины.

В современных видеокартах используется трюк - **ранний тест глубины**.

Настройка теста глубины

Очистка буфера для отрисовки очередного кадра:

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
```

Маска, позволяет производить сам тест, но не перезаписывать значение:

```
glDepthMask(GL_FALSE)
```

Функция, позволяет настроить поведения для сравнения:

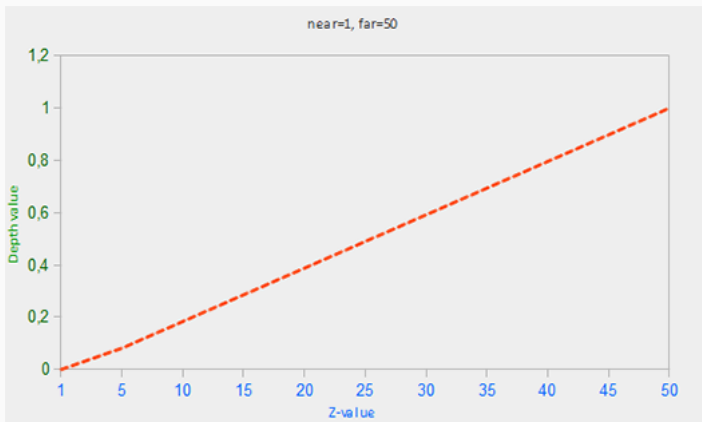
```
glDepthFunc(GL_LESS)
```

Функции для сравнения

Функция	Описание
GL_ALWAYS	Фрагмент всегда проходит тест глубины.
GL_NEVER	Фрагмент никогда не проходит тест глубины.
GL_LESS	Фрагмент проходит тест, если его значение глубины меньше хранимого в буфере.
GL_EQUAL	Фрагмент проходит тест, если его значение глубины равно хранимому в буфере.
GL_LEQUAL	Фрагмент проходит тест, если его значение глубины меньше либо равно хранимому в буфере.
GL_GREATER	Фрагмент проходит тест, если его значение глубины больше хранимого в буфере.
GL_NOTEQUAL	Фрагмент проходит тест, если его значение глубины отличается от хранимого в буфере.
GL_GEQUAL	Фрагмент проходит тест, если его значение глубины больше либо равно хранимому в буфере.

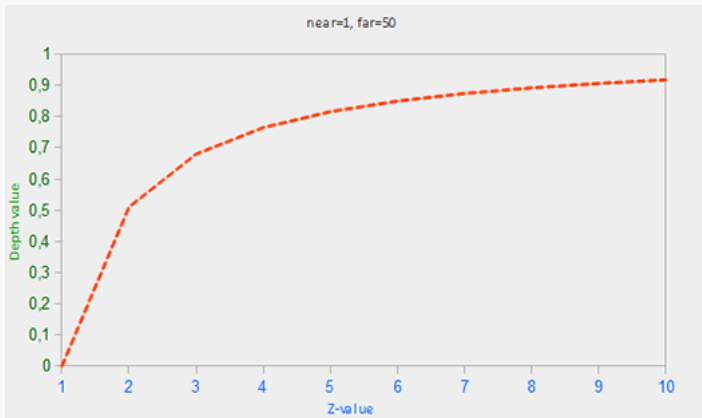
Z-Fight

Линейная интерполяция: $depth(z) = \frac{z - z_{Near}}{z_{Far} - z_{Near}}$



Z-Fight

Для обратной величины: $depth(z) = \frac{z^{-1} - zNear^{-1}}{zFar^{-1} - zNear^{-1}}$



Z-Fight

Для преодоления проблем точности:

- не стоит располагать объекты слишком близко друг к другу (с риском наложения треугольников);
- задавать ближнюю плоскости отсечения как можно дальше;
- использовать формат буфера глубины с большей точностью.

В современных видеокартах используется трюк - **ранний тест глубины**.

Визуализация буфера глубины

Простейший шейдер:

```
void main() {  
    out_color = vec4(vec3(gl_FragCoord.z), 1.0);  
}
```

Для линеаризации значений, нужно обратить формулы.