

ЛАБОРАТОРНАЯ РАБОТА №7

Смешивание и прозрачность.

Цель работы: добавит смешивание и прозрачность к прошлой работе.
Выполнить зачетное задание.

Задание: написать код для создания массива фигур по варианту с применением градации по прозрачности.

Смешивание и прозрачность

В OpenGL смешивание и прозрачность достигаются за счет использования функций смешивания. Смешивание - это процесс объединения цветов двух или более объектов для создания нового цвета. Уравнение смешивания учитывает исходный цвет (цвет рисуемого объекта) и целевой цвет (цвет, уже находящийся в буфере кадров в этом местоположении пикселя) для вычисления конечного цвета.

Прозрачность, с другой стороны, относится к степени, в которой объект позволяет свету проходить через него. В OpenGL прозрачность достигается путем установки альфа-значения цвета объекта. Альфа-значение определяет непрозрачность объекта, при этом 0 означает полную прозрачность, а 1 - полную непрозрачность.

`glBlendFunc()` - это функция OpenGL, которая устанавливает функцию смешивания, используемую графическим конвейером для объединения значений исходного и конечного пикселей во время рендеринга.

Функция принимает два параметра, `src` и `dst`, которые определяют исходный и конечный коэффициенты, используемые в уравнении смешивания. Возможными значениями этих параметров являются:

- `GL_ZERO`: коэффициент равен нулю.
- `GL_ONE`: коэффициент равен единице.
- `GL_SRC_COLOR`: фактором является исходный цвет.
- `GL_ONE_MINUS_SRC_COLOR`: коэффициент равен единице минус исходный цвет.
- `GL_DST_COLOR`: фактором является цвет назначения.
- `GL_ONE_MINUS_DST_COLOR`: коэффициент равен единице минус цвет назначения.
- `GL_SRC_ALPHA`: коэффициент - это исходное альфа-значение.

- GL_ONE_MINUS_SRC_ALPHA: коэффициент равен единице минус исходное альфа-значение.
- GL_DST_ALPHA: коэффициент - это конечное альфа-значение.
- GL_ONE_MINUS_DST_ALPHA: коэффициент равен единице минус конечное альфа-значение.
- GL_CONSTANT_COLOR: коэффициент - это постоянное значение цвета.
- GL_ONE_MINUS_CONSTANT_COLOR: коэффициент равен единице минус постоянное значение цвета.
- GL_CONSTANT_ALPHA: коэффициент представляет собой постоянное альфа-значение.
- GL_ONE_MINUS_CONSTANT_ALPHA: коэффициент равен единице минус постоянное альфа-значение.

Уравнение смешивания, используемое OpenGL, таково:

$$\text{Pixel_Color} = (\text{src} * \text{src_factor}) + (\text{dst} * \text{dst_factor})$$

где src - цвет исходного пикселя, dst - цвет конечного пикселя, а src_factor и dst_factor - это коэффициенты, указанные glBlendFunc().

Выбирая различные комбинации коэффициентов src и dst, разработчики могут добиться различных эффектов смешивания, таких как прозрачность, аддитивное смешивание и многое другое.

Пример:

```
...
GLfloat mat_diffuse[] = {1.0, 1.0, 1.0, 0.5}; // ставим альфа-канал 0.5
GLfloat mat_ambient[] = {1.0, 1.0, 1.0, 0.5};
...
glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse);
glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
...
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
...
```

В этом примере мы установили, как рассеянный, так и окружающий цвета материала на белый с альфа-значением 0,5 (прозрачность 50%). Затем мы включаем смешивание с помощью glEnable(GL_BLEND) и настраиваем функцию смешивания на использование исходного альфа-значения

(GL_SRC_ALPHA), чтобы определить, какая часть исходного цвета должна смешиваться с цветом назначения (GL_ONE_MINUS_SRC_ALPHA). Это позволит правильно рисовать объекты из прозрачных материалов поверх других объектов в сцене.

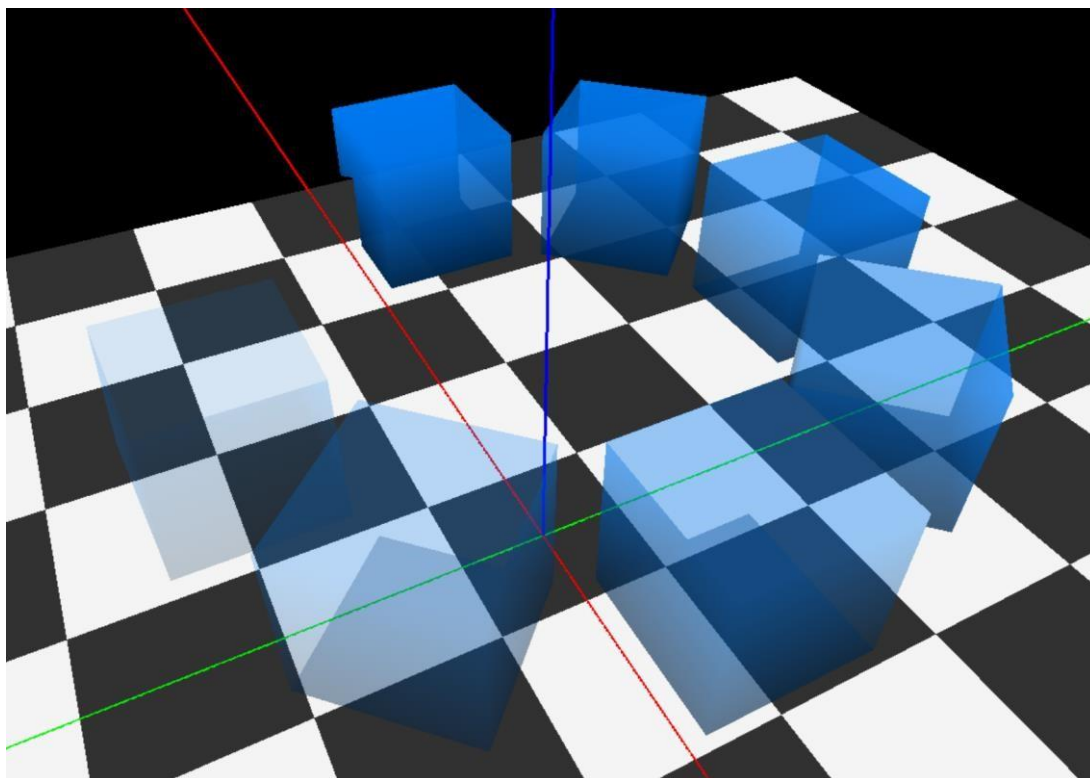


Рисунок 1 Пример массива из кубов с различной прозрачностью (от 0 до 0.9)

Задание

На основе 5 и 6 лабораторных работ написать код для массива фигур (см. Рисунок 1) с определённым количеством вершин в сечениях.

Напоминание: для нормальной работы освещения необходима создание фигур с нормальными.

Сечение каждой призмы является вписанный в окружность многоугольник с указанным кол-вом сторон.

Для задания нормалей используйте значение вершин. В прошлой работе с освещением при использовании нормалей к плоскости и получили жесткие ребра для фигуры (Рисунок 2 Hard Shading). Если использовать вершины как нормали получается сглаженная фигура (Рисунок 2 Smooth Shading).

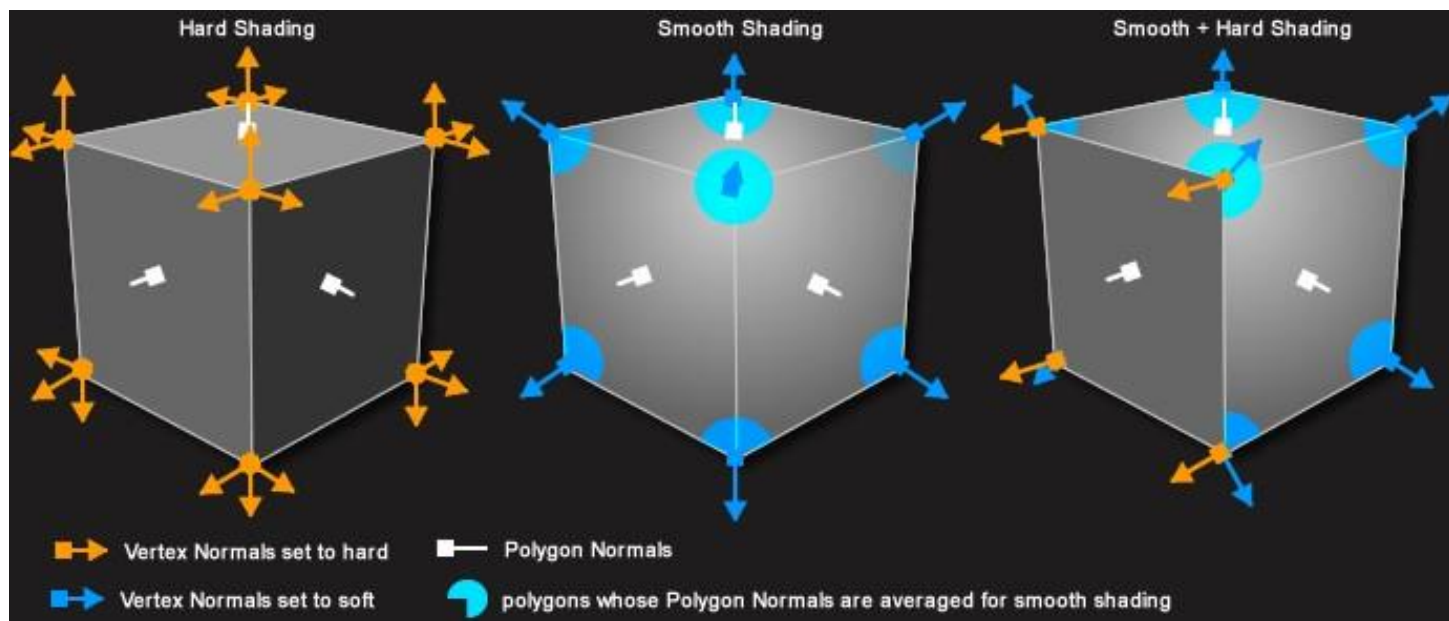


Рисунок 2 Варианты задания нормалей

Варианты на задание

Таблица №1 Тип фигуры

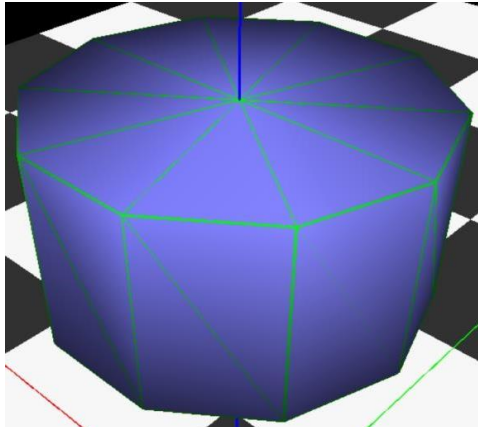
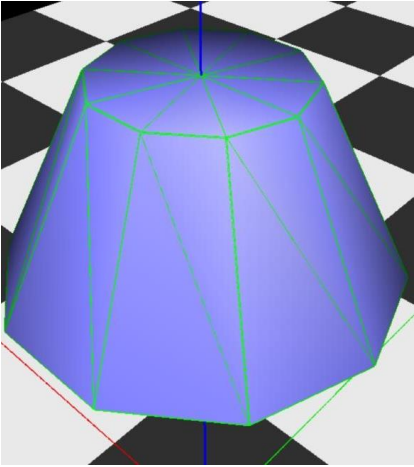
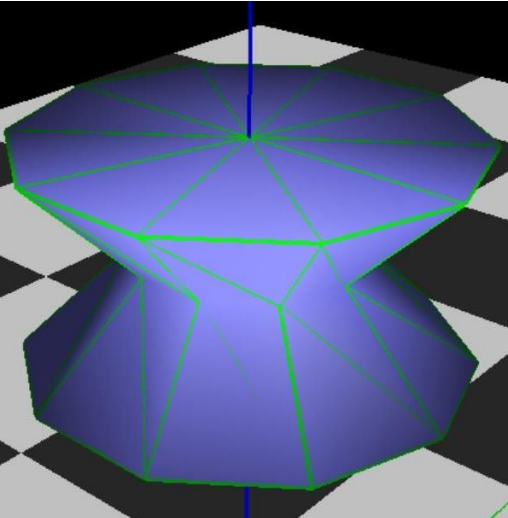
Тип I	Тип II	Тип III
		

Таблица №2 Варианты на задание

Вариант	Количество вершин	Тип фигуры
1	23	3
2	9	2
3	22	2
4	13	2
5	9	1
6	8	1
7	6	1
8	11	2
9	8	2
10	9	3
11	11	1
12	27	2
13	18	2
14	7	1
15	18	1
16	6	2
17	6	3
18	13	1
19	19	3
20	15	3
21	10	1
22	15	1
23	8	3
24	20	2
25	7	2
26	7	3
27	21	3
28	15	2
29	18	3
30	10	2