



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ  
КВАЛИФИКАЦИИ

Кафедра «Программное обеспечение вычислительной тех-  
ники и автоматизированных систем»

## Учебно-методическое пособие по дисциплине

# «КОМПЬЮТЕРНАЯ ГРАФИКА»

Автор  
Гнедина О.А.

Ростов-на-Дону, 2018

## Аннотация

Учебно-методическое пособие предназначено для студентов очной формы обучения по направлениям 02.03.03 «Математическое обеспечение и администрирование информационных систем», 09.03.04 «Программная инженерия».

## Авторы

Ст.преподаватель  
каф. ПОВТиАС  
Гнедина О.А.



## Оглавление

<b>1. Лабораторная работа №1: КООРДИНАТНЫЕ ПРЕОБРАЗОВАНИЯ В КОНВЕЙЕРЕ OPENGL .....</b>	<b>4</b>
1.1. Задание к лабораторной работе .....	4
1.2. Контрольные вопросы.....	6
<b>2. Лабораторные работы №2-3: СОЗДАНИЕ ЛИНЕЙЧАТОЙ ПОВЕРХНОСТИ OPENGL. СОЗДАНИЕ ПОВЕРХНОСТЕЙ ВРАЩЕНИЯ ОБЪЕКТА.....</b>	<b>7</b>
2.1. Задание к лабораторной работе .....	7
2.2. Контрольные вопросы.....	7
<b>3. Лабораторная работа №4: СИСТЕМА УПРАВЛЕНИЯ ДВИЖЕНИЕМ НАБЛЮДАТЕЛЯ В OPENGL .....</b>	<b>7</b>
3.1. Задание к лабораторной работе .....	8
3.2. Контрольные вопросы.....	8
<b>4. Лабораторная работа №5: СИСТЕМА УПРАВЛЕНИЯ СВЕТОМ В OPENGL. ПОСТРОЕНИЕ ТОЧЕЧНОГО ИСТОЧНИКА СВЕТА.....</b>	<b>8</b>
4.1. Задание к лабораторной работе .....	8
4.2. Контрольные вопросы.....	8
<b>5. Лабораторная работа №6: НАЛОЖЕНИЕ ТЕКСТУР НА ТРЕХМЕРНЫЙ ОБЪЕКТ.....</b>	<b>8</b>
5.1. Задание к лабораторной работе .....	9
5.2. Контрольные вопросы.....	9
<b>6. Лабораторная работа №7: ЗЕРКАЛЬНОЕ ОТРАЖЕНИЕ ТРЕХМЕРНОГО ОБЪЕКТА .....</b>	<b>9</b>
6.1. Задание к лабораторной работе .....	9
6.2. Контрольные вопросы.....	12
<b>7. Лабораторная работа №5: ПОСТРОЕНИЕ ТЕНИ ТРЕХМЕРНОГО ОБЪЕКТА .....</b>	<b>12</b>
7.1. Задание к лабораторной работе .....	12
7.2. Контрольные вопросы.....	12

## 1. ЛАБОРАТОРНАЯ РАБОТА №1: КООРДИНАТНЫЕ ПРЕОБРАЗОВАНИЯ В КОНВЕЙЕРЕ OPENGL

### 1.1. Задание к лабораторной работе

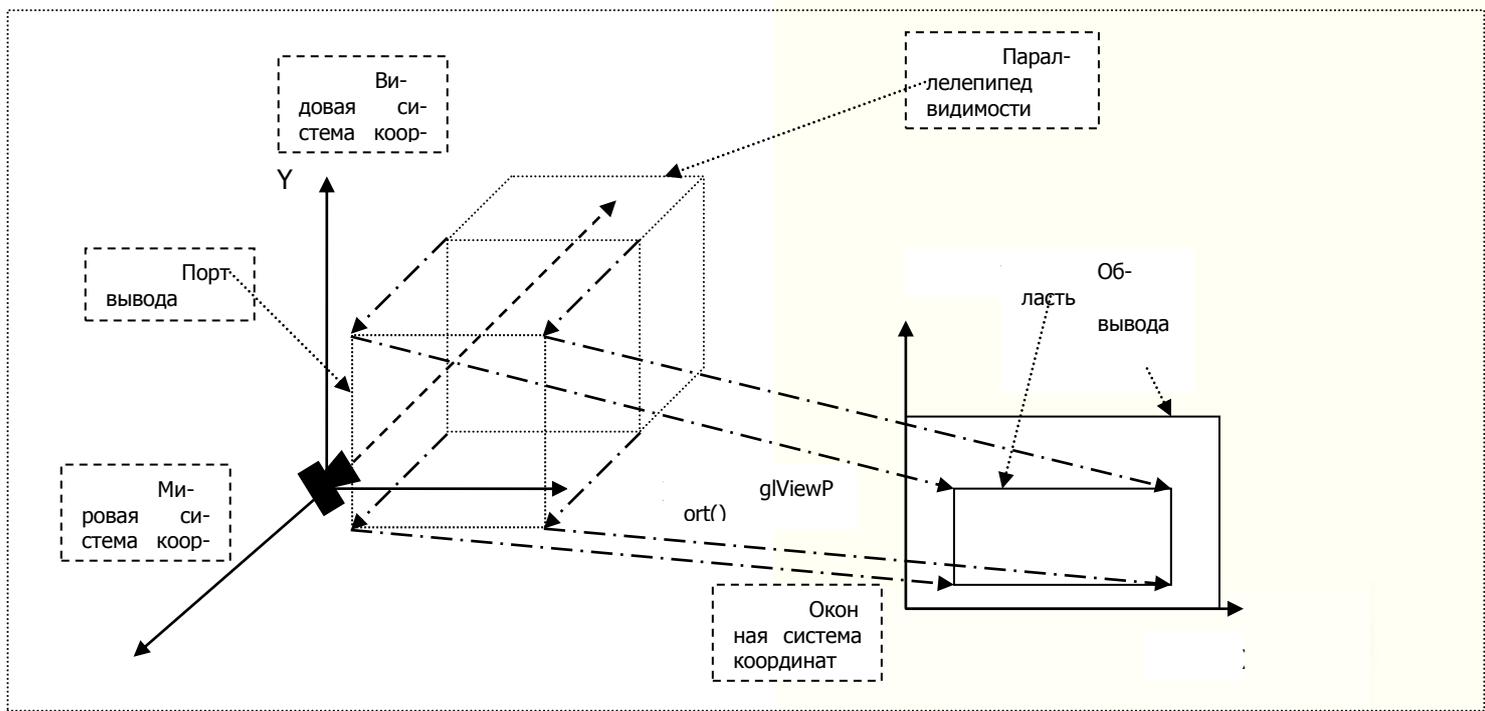
Создать на экране окно. В окне должны летать шарики. Шарики должны сталкиваться между собой и отражаться от стенок окна. При изменении размеров окна шарики должны оставаться внутри окна, по-прежнему должны отражаться от стенок окна и не должны искажаться, т.е. должны оставаться шариками, а не превращаться в эллипсоиды. Радиусы шариков также не должны меняться

В OpenGL есть три системы координат: мировая (трехмерная правая), видовая (трехмерная левая) и оконная (двумерная).

Камера (или глаз наблюдателя) всегда находится в начале видовой системы координат и взгляд направлен вдоль оси  $Z'$  этой системы координат.

1. Преобразования при работе с `glOrtho()`. При работе `glOrtho` предполагается, что начало видовой ( $X'Y'Z'$ ) и мировой ( $XYZ$ ) систем совпадают, а оси  $Z'$  и  $Z$  направлены в разные стороны. Остальные оси совпадают. Команда (ее параметры) задает положение параллелепипеда видимости в мировой системе координат и проецирует его на плоскость ( $X''Y''$ ) видовой системы координат в порт вывода, который проецируется командой `glViewport()` в оконную систему координат ( $X'''Y'''$ ) в область вывода на экране.

### Компьютерная графика



Параметры `glViewport()` задают положение области вывода в оконной системе координат, а сама команда проецирует порт вывода на область вывода.

Таким образом на экране вы видите изображение только тех объектов, которые находятся внутри параллелепипеда видимости в мировой системе координат.

Параллелепипед видимости и область вывода Вы должны задать при инициализации OpenGL.

Боковые плоскости параллелепипеда видимости должны отражать Ваш шар, а область вывода занимать все окно. Но тогда, при изменении размеров окна, чтобы шар на экране отражался от стенок окна нужно пропорционально изменить размеры порта вывода, т.е. изменить размеры параллелепипеда видимости и, следовательно, координаты отражающих плоскостей, поскольку шар по-прежнему должен отражаться от стенок пар-да видимости. Это делается в обработчике события изменения размеров окна. Чтобы иметь возможность вычислять коэффициенты пропорциональности нужно в глобальных переменных запоминать размеры окна и использовать их вместе с новыми размерами окна.

$$A = \text{НовыйРазмер} / \text{СтарыйРазмер}.$$

Вместе с пересчетом размеров пар-да видимости Вы должны пересчитать и координаты отражающих плоскостей. Если мы не будем их пересчитывать, и не будем менять размеры порта вывода, то начнет деформироваться шар в случае, если новое окно будет не подобно старому.

## 1.2. Контрольные вопросы

1. Что такое полигон?
2. Что такое полигональная модель и геометрический примитив?
3. Отличие камеры и наблюдателя.
4. Принцип столкновения сферических объектов.
5. Что такое порт вывода на экран.
6. Что такое параллелепипед видимости? Понятие сцены и размещения объектов на сцене

## **2. ЛАБОРАТОРНЫЕ РАБОТЫ №2-3: СОЗДАНИЕ ЛИНЕЙЧАТОЙ ПОВЕРХНОСТИ OPENGL. СОЗДАНИЕ ПОВЕРХНОСТЕЙ ВРАЩЕНИЯ ОБЪЕКТА**

### **2.1. Задание к лабораторной работе**

Создать объект, замкнутая поверхность которого состоит из плоских, линейчатых поверхностей и поверхностей вращения. Поставить точечный источник света. Написать программу управления движением объекта с помощью клавиатуры. Объект должен перемещаться в трех направлениях и вращаться относительно трех осей объектной системы координат.

Построение линейчатой поверхности проводить следующим образом:

1. Необходимо построить две пространственные кривые на параллельных гранях параллелепипеда. Кривые не должны быть параллельны!
2. На кривых построить одинаковое количество узлов
3. Каждая пара узлов на двух кривых попарно будут образовывать неплоский пространственный четырехугольник, поэтому его необходимо разбить диагональю на два треугольника – полигона.
4. Перехват событий клавиатуры осуществить с помощью функций библиотеки GLUT. Управление объектом осуществить с помощью функций `glTranslate()` и `glRotate()`
5. При движении объекта источник света не должен двигаться.

### **2.2. Контрольные вопросы**

1. Что такое линейчатая поверхность?
2. Виды освещения.
3. Что такое кривые при построении объектов?
4. Принцип создания поверхности вращения.

## **3. ЛАБОРАТОРНАЯ РАБОТА №4: СИСТЕМА УПРАВЛЕНИЯ ДВИЖЕНИЕМ НАБЛЮДАТЕЛЯ В OPENGL**

### 3.1. Задание к лабораторной работе

Необходимо написать систему управления движением наблюдателя. Наблюдатель – удаленный. С помощью клавиатуры изменять полярный и аксиальный углы сферических координат, задающих положение наблюдателя в объектной системе координат. Сцена – деталь из лаб.раб. 2-3.

### 3.2. Контрольные вопросы

1. Принцип работы алгоритма наблюдателя и камеры?
2. Что такое Полярный и аксиальный угол?
3. Что такое объектная система координат?
4. Система перемещения от наблюдателя и к наблюдателю.

## 4. ЛАБОРАТОРНАЯ РАБОТА №5: СИСТЕМА УПРАВЛЕНИЯ СВЕТОМ В OPENGL. ПОСТРОЕНИЕ ТОЧЕЧНОГО ИСТОЧНИКА СВЕТА

### 4.1. Задание к лабораторной работе

Для управления светом реализовать два варианта:

- независимое управление положением удаленного источника света с помощью клавиатуры. Изменяются полярный и аксиальный углы сферических координат, задающих положение источника в объектной системе координат.

- источник перемещается вместе с наблюдателем (лаб.раб. 4) и направление света направлено в центр объектной системы координат. С помощью клавиатуры изменяются параметры освещения источника.

### 4.2. Контрольные вопросы

1. Что такое точечный источник света?
2. Алгоритмы управления светом?
3. Рассеянный свет и фрактальный источник света.

## 5. ЛАБОРАТОРНАЯ РАБОТА №6: НАЛОЖЕНИЕ

## ТЕКСТУР НА ТРЕХМЕРНЫЙ ОБЪЕКТ

### 5.1. Задание к лабораторной работе

На грани детали из лаб. раб. 2-3 наложить текстуры. При чем должны быть продемонстрированы оба метода наложения - натягивание текстуры и «замаскивание» грани текстурой.

1. При наложении текстуры на линейчатую поверхность необходимо расположить параллелепипед так, чтобы его основание – грань, противоположная криволинейной грани, лежала в плоскости  $XOZ$  мировой системы координат.

2. Установить режим линейной генерации текстурных координат.

3. В качестве плоскостей отсчета текстурных координат выбрать плоскости  $XOY$  и  $YOZ$ .

При наложении текстуры на сферическую поверхность использовать режим сферической генерации текстурных координат.

### 5.2. Контрольные вопросы

1. Что такое текстура?
2. Что такое текстурное наложение?
3. Отличие методов наложения текстур.
4. Что такое режим линейной генерации текстурных координат.

## 6. ЛАБОРАТОРНАЯ РАБОТА №7: ЗЕРКАЛЬНОЕ ОТРАЖЕНИЕ ТРЕХМЕРНОГО ОБЪЕКТА

### 6.1. Задание к лабораторной работе

Сцена представляет объект, созданный в лаб.раб. 2 – 6 и прямоугольного зеркала. Объекты, которые расположены со стороны зеркальной поверхности зеркала, должны отражаться в зеркале. У зеркала только одна зеркальная поверхность. Объекты за зеркалом, должны зеркалом заслоняться. Сцена должна вращаться, управление вращением - с помощью клавиатуры.

Как задать трафарет

Допустим нам нужно на экране выделить область, в которую выводится изображение, не совпадающее с тем, что находится

ся в области видимости наблюдателя.

Пусть

- Обл.1 – область экрана, куда должна выводиться область видимости. Установим в ней значение трафарета 0x01
- Обл.2 – область экрана, куда выводится дополнительное изображение. Пусть эта область определяется проекцией на экран некоторого объекта сцены. Установим в ней значение трафарета 0x02

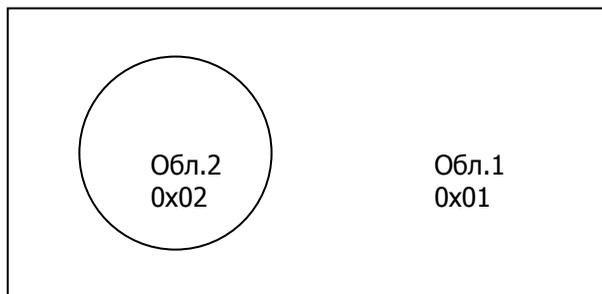


Рисунок 2 – Трафарет

1. Инициализируем буфер трафарета
  - 1.1. `glClearStencil(0x01);` //значение которым инициализируется буфер трафарета при очистке
  - 1.2. `glStencilMask(0xFF);` //маска буфера – определяет битовые плоскости буфера, которые могут модифицироваться
  - 1.3. `glClear(GL_STENCIL_BUFFER_BIT);` //значение буфера в Обл.1
  - 1.4. `glEnable(GL_STENCIL_TEST);` //разрешаем тест трафарета
2. Рисуем видимую сцену без объекта, определяющего область дополнительного изображения
  - 2.1. `DrawScene()`
3. Определяем значение трафарета в Обл.2
  - 3.1. `glStencilFunc(GL_EQUAL, 0x02, 0x01);` // т.е. при выводе изображения тест трафарета будет завершаться //отрицательно, поскольку  $0x02 \& 0x01 \neq 0x01$
  - 3.2. `glStencilOp(GL_REPLACE, GL_KEEP, GL_KEEP);` //буфер

Компьютерная графика

трафарета изменит значение на 0x02, когда не выполнится тест трафарета

- 3.3. Рисуем объект, изображением которого определяет Обл.2 (например, круг - лупа)

3.3.1. glBegin( ... );

// рисуем объект

// в области экрана, где рисуется объект, буфер трафарета устанавливается = 0x02

glEnd();

4. Разрешаем вывод изображения только в область со значением буфера трафарета = 0x02

4.1. glStencilFunc(GL\_EQUAL, 2, 2);

4.2. glStencilOp(GL\_KEEP, GL\_KEEP, GL\_KEEP);

5. Рисуем дополнительное изображение которое должно быть в круге (в лупе)

6. // Блокируем тест трафарета  
glDisable(GL\_STENCIL\_TEST);

Как формировать дополнительное изображение  
Зеркальное изображение (изображение в зеркале)  
Зеркало делит сцену на два полупространства:

1. Полупространство перед зеркалом (предметы находящиеся там отображаются в зеркале)
2. Полупространство за зеркалом (предметы находящиеся там не отображаются в зеркале)

При формировании зеркального изображения нужно:

1. Заполнить буфер трафарета так, чтобы в области зеркала значение буфера было равно 2, а во всей остальной части экрана значение буфера трафарета = 1
2. Разрешить вывод изображения в область значений буфера трафарета = 1
3. Вывести всю сцену
4. Разрешить вывод изображения в область значений буфера трафарета = 2
5. Объекты, находящиеся в полупространстве перед зеркалом отразить относительно плоскости зеркала (поместить их за зеркало) и построить их изображение.

### **6.2. Контрольные вопросы**

1. Что такое зеркальное отображение?
2. Принцип построения прямоугольного зеркала.
3. Что такое буфер трафарета?
4. Принцип наложения трафарета?

## **7. ЛАБОРАТОРНАЯ РАБОТА №5: ПОСТРОЕНИЕ ТЕНИ ТРЕХМЕРНОГО ОБЪЕКТА**

### **7.1. Задание к лабораторной работе**

Построить тень от источника света и детали, созданным в лаб.раб. 2 – 6. В качестве "земли" использовать одну из координатных плоскостей объектной системы координат.

### **7.2. Контрольные вопросы**

1. Принцип построения матрицы тени?
2. Построение тени от точечного источника света?
3. Построение тени от рассеянного источника света?
4. Что такое зернистая тень?
5. Что такое полная тень и полутень?