

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра автоматизированных систем управления

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К КОНТРОЛЬНОЙ РАБОТЕ

по дисциплине

«КОМПЬЮТЕРНАЯ ГРАФИКА»

для студентов направления подготовки

09.03.02 Информационные системы и технологии

заочной и заочной с сокращенным сроком форм обучения

Рассмотрено на заседании кафедры

«Автоматизированные системы
управления»

Протокол № 6 от 19 января 2017 г.

Утверждено на заседании

учебно – издательского совета ДонНТУ

Протокол № 3 от 6 апреля 2017 г.

Донецк
2017

Методические указания к контрольной работе по дисциплине «Компьютерная графика» для студентов направления подготовки 09.03.02 «Информационные системы и технологии» заочной и заочной с сокращенным сроком форм обучения / Составили: Васяева Т.А., Матях И.В. – Донецк: ДНТУ, 2017. – 14 с.

Методические указания содержат краткие теоретические сведения, методические рекомендации и задания для выполнения контрольной работы по дисциплине «Компьютерная графика». Изложена методика выполнения контрольной работы, требования к содержанию отчета, список рекомендуемой литературы.

Методические указания к контрольной работе предназначены для студентов направления подготовки 09.03.02 «Информационные системы и технологии» профиль «Информационные системы и технологии в технике и бизнесе» заочной и заочной с сокращенным сроком форм обучения.

Составители:

к.т.н., доц. Васяева Т.А.

ас. Матях И.В.

Рецензент: к.т.н., доц., доц. каф. АСУ

М.В. Привалов

Рецензент: к.т.н., доц., доц. каф. АТ

В.В. Червинский

Ответственный за выпуск:

зав. каф. АСУ Привалов М.В.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
КОНТРОЛЬНАЯ РАБОТА	5
СПИСОК ЛИТЕРАТУРЫ	13

ВВЕДЕНИЕ

Компьютерная графика представляет собой одно из направлений развития систем обработки информации, связанной с изображением объектов, которое возникло в связи с необходимостью широкого использования систем компьютерной графики в виртуальной реальности, в глобальной сети Internet и системах интерактивной графики. Понятие «компьютерная графика» очень часто трактуется по-разному. Компьютерная графика – это новая отрасль знаний, которая, с одной стороны, представляет комплекс аппаратных и программных средств, используемых для формирования, преобразования и выдачи информации в визуальной форме на средства отображения ЭВМ. С другой стороны, под компьютерной графикой понимают совокупность методов и приемов для преобразования при помощи ЭВМ данных в графическое представление. Системы компьютерной графики обеспечивают пользователю широкий набор услуг и позволяют создавать целый ряд различных способов диалога, типа человек – компьютер, позволяют создавать анимационные и реалистичные изображения и совершенствуют способы ввода-вывода информации.

Изучение данной дисциплины вносит необходимый вклад в достижение ожидаемых результатов в профессиональной части программы подготовки студентов специальностей «Информационные системы и технологии».

Курс основан на изучении математических основ предметной области, значительное внимание уделяется программной реализации.

Основная задача курса – дать в доступной форме математический аппарат для создания изображений геометрических объектов на экране и их манипуляций.

Контрольная работа

Тема: алгоритмы отсечения фигур

Отсечение по полю вывода

Проблема отсечения изображения по некоторой границе, присутствует в большинстве задач компьютерной графики. Рассматривается задача применительно к отрезкам прямых и обобщается на многоугольники. Окружности и другие фигуры можно аппроксимировать отрезками (многоугольниками).

Алгоритм Козна-Сазерленда

Рассмотрим алгоритм Козна-Сазерленда для отсечения отрезков прямых. Этот алгоритм позволяет легко определять нахождение отрезка полностью внутри или полностью снаружи окна, и если так, то его можно рисовать или не рисовать, не заботясь об отсечении по границе окна.

Для работы алгоритма вся плоскость, в которой лежит окно, разбивается на девять подобластей или квадратов, как показано на рис. 1.1.

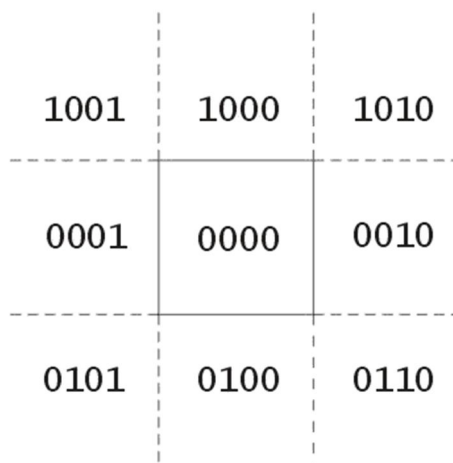


Рисунок 1.1 – Разбиение на подобласти в методе Козна-Сазерленда

Окну соответствует область обозначенная кодом 0000. Конечным точкам отрезка приписывается 4-битный код «вне/внутри», в зависимости от нахождения отрезка в соответствующей подобласти. Каждому биту присваивается значение 1 в соответствии со следующим правилом:

Бит 1 – точка находится выше окна;

Бит 2 – точка находится ниже окна;

Бит 3 – точка находится справа от окна;

Бит 4 – точка находится слева от окна;

Иначе биту присваивается нулевое значение. Значения этих битов для конечных точек отрезков легко определить по знакам соответствующих разностей (положительный результат – 0, отрицательный – 1): $(y_{\max} - y)$ – для 1-го бита, $(y - y_{\min})$ – для 2-го бита, $(x_{\max} - x)$ – для 3-го бита и $(x - x_{\min})$ – для 4-го бита.

Для каждого отрезка рассчитываются коды концов (K_1 , K_2) затем производится анализ:

- Если $K_1 \wedge K_2 \neq 0$, тогда отрезок лежит вне поля вывода – отрезок отбрасывается
- Если $K_1 = K_2 = 0$, тогда отрезок полностью лежит внутри поля вывода – отсечение не нужно, отрезок полностью прорисовывается
- Если $K_1 \wedge K_2 = 0$, отрезок может частично лежать внутри поля вывода – необходимо отсечение по полю вывода.

В этом случае применяется последовательное разделение отрезка, так что на каждом шаге конечная точка отрезка с ненулевым кодом «вне/внутри» заменяется на точку, лежащую на стороне окна или на прямой, содержащей сторону. При этом порядок перебора сторон окна не имеет значения.

Когда $K_1 \wedge K_2 = 0$ необходимо отсекать отрезок по границам поля вывода, отсечение происходит последовательно по всем сторонам рис.1.2.

На рис. 1.2 жирным выделено ребро, по которому происходит отсечение. Также надо отметить, что точки, лежащие на границе поля вывода, принадлежат полю вывода.

На каждом шаге отсечения вычисляются новые координаты одной точки, найдем формулы для вычисления новых координат.

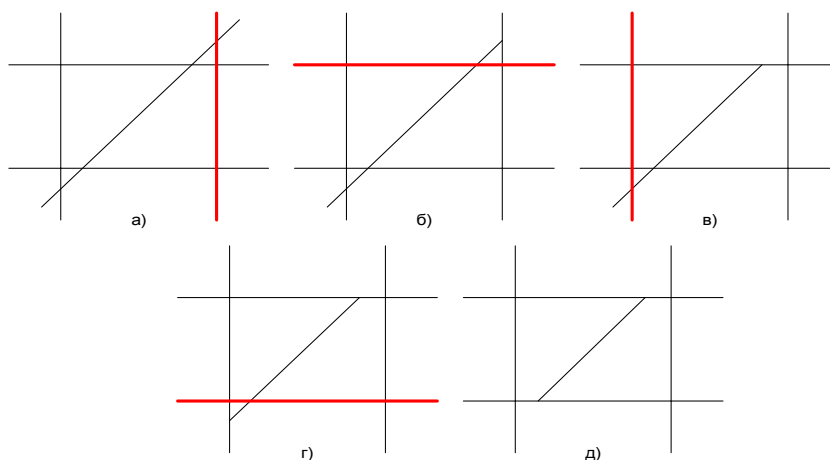


Рисунок 1.2 – «Отсечение отрезка по прямоугольной области»

Алгоритм Сазерленда-Ходгмана для отсечения многоугольников

Рассмотрим алгоритм Сазерленда-Ходгмана (Sutherland-Hodgman). В алгоритме используется стратегия «разделяй и властвуй», которая позволяет решение общей задачи свести к решению ряда простых и похожих подзадач. Примером такой подзадачи является отсечение многоугольника относительно одной отсекающей границы. Последовательное решение четырех таких задач позволяет провести отсечение относительно прямоугольной области (рис. 1.3).

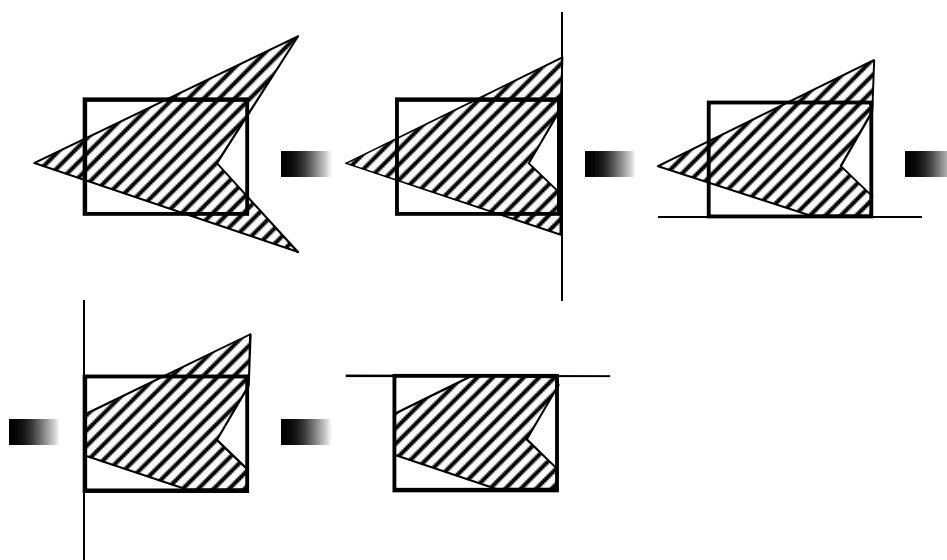


Рисунок 1.3 –Последовательное отсечение многоугольника

На вход алгоритма поступает последовательность вершин многоугольника V_1, V_2, \dots, V_n . Ребра многоугольника проходят от V_i к V_{i+1} , от V_n к V_1 . С

помощью алгоритма производится отсечение относительно ребра и выводится другая последовательность вершин, описывающая усеченный многоугольник.

Алгоритм «обходит» вокруг многоугольника от V_n к V_1 и обратно к V_n , проверяя на каждом шаге соотношение между последовательными вершинами и отсекающей границей. Необходимо проанализировать четыре случая (рис. 1.4).

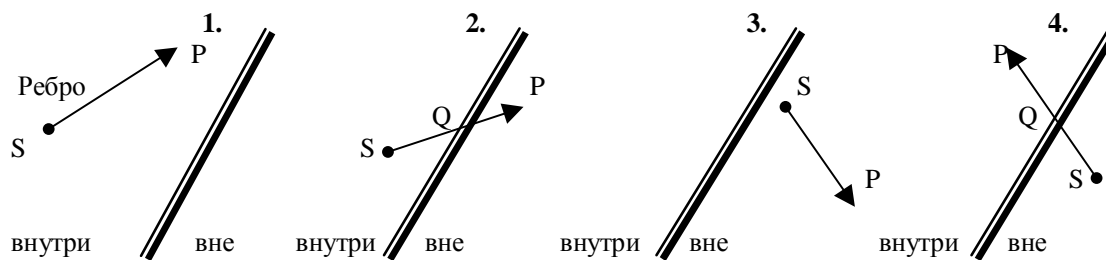


Рисунок 1.4 – Случаи, возникающие при отсечении многоугольников
Может также возникнуть вариант, когда ребро совпадает с границей.

В случае 1: Добавить в список рисуемых вершин P;

2: Добавить в список Q;

3: Ничего не добавляется;

4: Добавить в список P,Q;

Алгоритм Вейлера – Азертонна

Как обрабатываемый (объект), так и отсекающий (отсекатель) многоугольники описываются в алгоритме циклическими списками их вершин. Внешняя граница каждого из этих многоугольников обходится по часовой стрелке, а внутренние границы или отверстия – против часовой стрелки. Это условие означает, что при обходе вершин многоугольника в порядке их следования, в соответствующем списке внутренняя его область будет расположена справа от границы. Границы обрабатываемого (объекта) и отсекающего (отсекателя) многоугольников могут пересекаться или не пересекаться между собой. Если они пересекаются, то точки пересечения образуют пары. Одно пересечение из пары возникает, когда ребро обрабатываемого многоугольни-

ка входит внутрь отсекающего многоугольника, а другое – когда оно выходит оттуда. Основная идея заключается в том, что алгоритм начинается с точки пересечения входного типа, затем он прослеживает внешнюю границу по часовой стрелке до тех пор, пока не обнаруживается еще одно ее пересечение с отсекающим многоугольником. В точке последнего пересечения производится поворот направо и далее прослеживается внешняя граница отсекателя по часовой стрелке до тех пор, пока не обнаруживается ее пересечение с обрабатываемым многоугольником. И вновь, в точке последнего пересечения производится поворот направо и далее прослеживается граница обрабатываемого многоугольника. Этот процесс продолжается до тех пор, пока не встретится начальная вершина. Внутренние границы обрабатываемого многоугольника обходятся против часовой стрелки.

Пример. Отсечение с помощью алгоритма Вейлера — Азертон.

Случай простого многоугольника. Рассмотрим многоугольник, показанный на рис. 1.5, который отсекается по квадрату. На рисунке изображены также точки пересечения этих многоугольников, помеченные через I_t . Ниже приводятся списки вершин обрабатываемого и отсекающего многоугольников. В список входов занесены вершины I_2, I_4, I_6 и I_8 , а в список выходов — вершины I_1, I_3, I_5, I_7 .

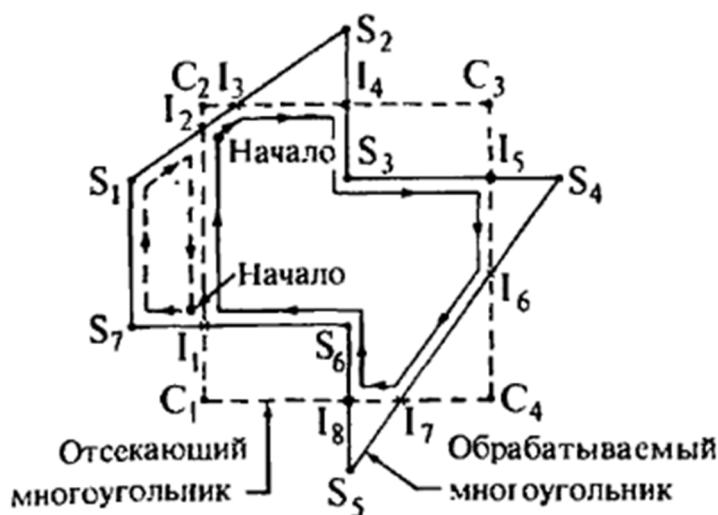


Рисунок 1.5 – Алгоритм Вейлера -Азертон

Для формирования внутреннего многоугольника возьмем первую точку пересечения из списка входов – I_2 . Описанный алгоритм дает результаты, показанные сплошной линией со стрелками на рис. 1.5, а также в таблице 1.1. Результирующий внутренний многоугольник таков:

$$I_2 I_3 I_4 S_3 I_5 I_6 I_7 I_8 S_6 I_1 I_2.$$

Таблица 1.1 - Пример отсечения

Вершины обрабатываемого многоугольника	Вершины отсекающего многоугольника	Вершины обрабатываемого многоугольника	Вершины отсекающего многоугольника
<p>Начало</p> <p>S_1</p> <p>I_2</p> <p>I_3</p> <p>S_2</p> <p>I_4</p> <p>S_3</p> <p>I_5</p> <p>S_4</p> <p>I_6</p> <p>I_7</p> <p>S_5</p> <p>I_8</p> <p>S_6</p> <p>I_1</p> <p>S_7</p> <p>S_1</p>	<p>C_1</p> <p>I_1</p> <p>I_2 Конец</p> <p>C_2</p> <p>I_3</p> <p>I_4</p> <p>C_3</p> <p>I_5</p> <p>I_6</p> <p>C_4</p> <p>I_7</p> <p>I_8</p> <p>C_1</p>	<p>S_1</p> <p>I_2</p> <p>I_3</p> <p>S_2</p> <p>I_4</p> <p>S_3</p> <p>I_5</p> <p>S_4</p> <p>I_6</p> <p>I_7</p> <p>S_5</p> <p>I_8</p> <p>S_6</p> <p>I_1 Начало</p> <p>S_7</p> <p>S_1</p>	<p>C_1</p> <p>I_1 Конец</p> <p>I_2</p> <p>C_2</p> <p>I_3</p> <p>I_4</p> <p>C_3</p> <p>I_5</p> <p>I_6</p> <p>C_4</p> <p>I_7</p> <p>I_8</p> <p>C_1</p>
Внутренний многоугольник		Внешний многоугольник	

Если начинать с других точек пересечения из списка входов, т. е. с точек I_4 , I_6 и I_8 , то получится тот же самый результат.

Для формирования внешних многоугольников возьмем первую точку пересечения из списка выходов – I_1 . Описанный выше алгоритм дает результаты, показанные штриховой линией со стрелками на рис.1.5, и в таблице 1.1. Заметим, что вершины из списка вершин отсекаателя проходятся в обратном порядке от I_1 к I_2 . Получающиеся внешний многоугольник таков:

$$I_1 S_7 S_1 I_2 I_1.$$

Если же начинать построение с других точек пересечения – I_3 , I_5 или I_7 – из списка выходов, то получатся другие внешние многоугольники, соответственно:

$$I_3 S_2 I_4 I_3, I_5 S_4 I_6 I_5 \text{ или } I_7 S_5 I_8 I_7.$$

Задание

Написать и отладить программу на языке высокого уровня с использованием функциональных возможностей OpenGL для реализации отсечения фигур по полю вывода. Для этого:

1. Нарисовать поле вывода в центре экрана.
2. Нарисовать свою фигуру (табл.1.2) в произвольном месте. Координаты фигуры в таблице 1.2 задают относительные пропорции, а не абсолютные величины. Если одна часть фигуры внутри поля вывода, а другая – вне ее, то закрасить ту часть фигуры, которая внутри. Часть (фигура) вне поля вывода рисуется только контуром.
3. Обеспечить перемещение фигуры стрелками или другими клавишами (без изменения (движения) поля вывода) соблюдая правило: внутри поля вывода – закрашенная фигура, вне – только контур.

Таблица 1.2 - Варианты заданий

№	Фигура		Цвет контура	Цвет заливки
1	A(10,40), C(30,25), E(45,35).	B(15,15), D(50,18),	красный	синий
2	A(20,40), C(45,25), E(55,35).	B(25,15), D(60,18),	синий	красный
3	A(10,50), C(30,35), E(45,45).	B(15,25), D(50,28),	зеленый	белый
4	A(10,40), C(30,5), E(45,35).	B(15,15), D(50,18),	серый	желтый

№	Фигура	Цвет контура	Цвет заливки
5	A(5,40), B(10,15), C(15,10), D(45,18), E(40,35).	красный	белый
6	A(20,49), B(25,5), C(40,35), D(60,40), E(70,65).	желтый	зеленый
7	A(15,40), B(20,15), C(35,25), D(55,18), E(50,35).	белый	Красный
8	A(10,45), B(15,20), C(30,30), D(50,25), E(45,40).	синий	желтый
9	A(15,45), B(20,20), C(35,30), D(55,25), E(50,40).	серый	белый
10	A(12,40), B(18,15), C(35,25), D(54,18), E(47,35).	зеленый	синий

Содержание отчета

1. Титульный лист.
2. Задание.
3. Описание используемых методов.
4. Текст программы.

При защите контрольной работы тестирование программы **обязательно!**

СПИСОК ЛИТЕРАТУРЫ

1. Блинова Т.А. Компьютерная графика: учебник для вузов / Т. А. Блинова, В. Н. Порев ; Т.А. Блинова, В.Н. Порев; под ред. В.Н. Порева. - К. : Юниор; СПб. : Корона принт, 2006. - 520с. : ил. - (Учебное пособие). - ISBN 966-7323-48-X.
2. Роджерс Д.Ф. Алгоритмические основы машиной графики / Д. Ф. Роджерс; Д.Ф. Роджерс; пер. с англ.: С.А. Вичеса и др.; под ред.: Ю.М. Баяковского, В.А. Галактионова. - М. : Мир, 1989. - 503с. : ил. - Перевод изд.: Procedural elements for computer graphics/D. F. Rogers.
3. Дейтел, Х.М. Как программировать на С++ [Электронный ресурс] / Х. М. Дейтел, Дейтел П.Дж. ; Х.М. Дейтел, П.Дж. Дейтел ; пер. с англ. под ред. В.В. Тимофеева. - 5-е изд. - 19 Мб. - М. : Бином-Пресс, 2008. - 1 файл. - Перевод изд.: С++ How to Program/Н.М. Deitel, Р.Ј. Deitel.
4. OpenGL. Руководство по программированию / М. Ву [и др.] ; М.Ву, Т. Девис, Дж. Нейдер, Д. Шрайнер ; пер. с англ.: Е. Васильев, Е. Эрман. - 4-е изд. - СПб. : Питер, 2006. - 624с.: ил. - (Библиотека программиста). - Перевод изд.: OpenGL/ М. Woo, Т. Davis, J. Neider, D. Shreiner. - ISBN 5-94723-827-6.
5. Шрайнер Д. OpenGL. Официальный справочник : перевод с английского / Д. Шрайнер; Д. Шрайнер ; под ред. Д. Шрейнера. - СПб.: ДиаСофтЮП, 2002. - 512с. - Перевод изд.: OpenGL Reference Manual, Third Edition/ ed. D. Shreiner. - ISBN 5-93772-048-2.
6. Тихомиров, Ю.В. OpenGL. Программирование трехмерной графики / Ю. В. Тихомиров ; Ю.В. Тихомиров. - 2-е изд. - СПб. : БХВ-Петербург, 2002. - 304с.: ил. - (Мастер программ). - ISBN 5-94157-174-7.
7. Боресков, А.В. Компьютерная графика: первое знакомство / А. В. Боресков, Г. Е. Шикин, Г. Е. Шикина ; А.В. Боресков, Е.В. Шикин, Г.Е. Шикина ; под ред. Е.В. Шикина . - М. : Финансы и статистика, 1996. - 176с. : ил. - (Диалог с компьютером).
8. Гайван А.В. Компьютерная графика и численные методы / А. В. Гайван ; А.В. Гайван. - М. : ВА Принт, 1994. - 114с. : ил.

9. Залогова Л.А. Компьютерная графика : практикум / Л. А. Залогова ; Л.А. Залогова ; науч. ред. С.В. Русаков. - 2-е изд. - М. : БИНОМ. Лаборатория знаний, 2007. - 245с.: ил. - (Элективный курс. Информатика). - ISBN 978-5-94774-656-3.

10. Корриган Д. Компьютерная графика : секреты и решения / Д. Корриган ; пер.с англ. Д.А. Куликова. - М. : Энтроп, 1995. - 352с. : ил. - Перевод изд.: Computer graphics/J. Corrigan.

11. Коцюбинский А.О. Компьютерная графика : практическое пособие / А. О. Коцюбинский, С. В. Грошев ; А.О.Коцюбинский, С.В.Грошев. - М. : ТЕХНОЛОДЖИ-3000, 2001. - 752с. : ил. - ISBN 5-94472-011-8.

12. Дегтярев, В.М. Компьютерная геометрия и графика: учебник для вузов / В. М. Дегтярев ; В.М. Дегтярев. - М. : ИЦ "Академия", 2010. - 192с. - (Высшее профессиональное образование. Информатика и вычислительная техника). - ISBN 978-5-7695-5888-7.