МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра вычислительной техники

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Организация ЭВМ и систем»

Тема: Исследование видеосистемы (графический режим)

Студентка гр. 3353 _____ Шинкарь К. Д.

Преподаватель _____ Гречухин М.Н.

Санкт-Петербург
2022

Цель работы: изучение работы с видеосистемой в графическом режиме, вывод графика заданной функции с масштабированием и разметкой осей.

Задание: Разработать программу для вывода на экран графика заданной функции. В моем случае (1 вариант) – $Sin^2(x/2)+Sqrt(x)$. Период: от $3\pi/2$ до 15π. Произвести разметку осей и проставить истинные значения точек. Найти максимальное значение функции на заданном интервале и вывести в отдельное окно на экране.

Краткие сведения о видеосистемах ПЭВМ, графическом режиме их работы и функциях обслуживания графического режима.

Использование графики в языке C++ - это многошаговый процесс. Прежде всего необходимо определить тип видеоадаптера. Затем устанавливается подходящий режим его работы и выполняется инициализация графической системы в выбранном режиме. После этого становятся доступными для использования функции графической библиотеки graphics.h для построения основных графических примитивов: отрезков прямых линий, окружностей, эллипсов, прямоугольников, секторов, дуг и т.д., появляется возможность вывода текста с использованием различных шрифтов.

Использование библиотеки графики намного сокращает объем программирования для вывода основных графических примитивов. C++ "маскирует" многие технические детали управления оборудованием, о которых пользователь должен быть осведомлен при работе с видеоадаптером через порты или BIOS. Платой за эти удобства является значительное увеличение размера EXE-файлов.

Использование графической библиотеки C++ требует знакомства с моделью графической системы, применяемой компилятором для представления графической системы компьютера. Весь код библиотеки графики разбивается на две части: немобильную, которая зависит от типа видеоадаптера и мобильную.

Немобильная часть представляет собой так называемый BGI-драйвер (BGI - Borland Graphics Interface). Драйвер является обработчиком прерывания 10h, который должен дополнить системный обработчик до того, как будут использоваться мобильные функции. Перед завершением программы таблица векторов прерывания восстанавливается. Основные функции, выполняемые BGI-драйвером, сводятся к установке и обновлению ряда внешних переменных, которые могут изменяться как функциями системного обработчика прерывания 10h (например, при переключении видеорежима,

изменении регистров палитры и т.п.), так и мобильными функциями библиотеки графики.

C++ включает целую коллекцию драйверов для каждого из типов адаптеров, хранимых обычно в отдельном поддиректории. Определение и установка графического режима.

Целая группа функций – getgraphmode(), getmaxmode(), getmodename() , getmoderange() - упрощает работу по определению текущего установленного режима.

Две функции позволяют определить ширину и высоту экрана в пикселах для текущего видеорежима: getmaxx() и getmaxy().

Функция restorecrtmode() возвращает видеоадаптер в текстовый режим.

int getgraphmode (void) возвращает текущий графический режим (возвращаемое число соответствует номеру режима).

void setgraphmode(int mode) устанавливает видеосистему в режим, заданный значением переменной mode, и сбрасывает значения внутренних переменных системы графики в их значения по умолчанию.

void restorecrtmode(void) возвращает видеоадаптер в режим, в котором он был до выполнения инициализации системы графики (текстовый).

Функция putpixel(int x, int y, int pixelcolor) Определяет, лежит ли пиксел с координатами (x, y) в текущем графическом окне, и, если лежит, выводит на экран пиксел, код цвета которого равен pixelcolor. В противном случае цвет пиксела не изменяется. Используя функцию putpixel(), можно "стереть" пиксел, если вывести его с кодом цвета фона.

void line(int x1, int y1, int x2, int y2) выводит отрезок прямой линии между двумя явно специфицированными точками (x1, y1) и (x2, y2), используя текущие цвет, стиль, толщину и режим вывода линии. Координаты (x1, y1) и (x2, y2) задаются относительно левого верхнего угла текущего графического окна. Функция не изменяет текущую позицию. void setcolor (int color) Устанавливает цвет, используемый функциями графического вывода в значение, заданное аргументом color. До того момента, пока цвет не

установлен, используется максимальный (из палитры) номер цвета. В случае, если color задает недопустимый номер цвета для текущей палитры, текущий цвет остается неизменным.

void floodfill (int x, int y, int border) Заполняет текущим стилем область экрана, ограниченную непрерывной линией с цветом border, начиная с точки с координатами (x, y). Функция заполняет область либо внутри замкнутой линии, либо вне ее. Это зависит от положения начальной точки: если она лежит внутри области, заполняется внутренняя область; если точка лежит вне замкнутой области, заполняется внешняя область; если точка лежит точно на линии цвета border, заполнение не производится. Заполнение начинается с начальной точки и продолжается во всех направлениях, пока не встретится пиксел с цветом border. Цвет border должен отличаться от цвета заполнения, в противном случае будет заполнен весь экран.

Цвет и маска заполнения могут быть заданы с помощью функций setfillpattern() и setfillstyle().

void setfillstyle(int pattern, int color) выбирает один из предопределенных стилей заполнения. Значение pattern идентифицирует стиль. Возможные значения для pattern приведены в табл. 3.7. Аргумент color задает цвет, используемый для пикселов по заданному шаблону. Данная функция не предназначена для установки определенной пользователем маски заполнения. Для этого используется функция setfillpattern().

void rectangle(int left, int top, int right, int bottom) Выводит контур прямоугольника, заданного координатами левого верхнего (left, top) и правого нижнего (right, bottom) углов. Координаты углов задаются относительно координат левого верхнего угла текущего графического окна. Контур выводится линией текущего цвета и стиля. Цвет контура может быть установлен функцией setcolor(). Стиль линии может быть выбран или задан функцией setlinestyle().

Текст программы

```
#include <conio.h>
#include <graphics.h>
#include <math.h>
#include <stdio.h>
#include <string.h>

int Xmax, Ymax, X0, X1, Y0, Y1;
float Rmax = 0.0;

// Function to draw the graph of f(x) = sin^2(x/2) + sqrt(x)
void drawf(int pixelToX, int pixelToY) {
   int startXcoord = X0 + (int)(1.5 * pixelToX); // Start point at 1.5*pi
   int endXcoord = X0 + (int)(15 * pixelToX);    // End point at 15*pi
   int N = endXcoord - startXcoord;              // Number of pixels for the graph
   double dx2 = (15 * M_PI - 1.5 * M_PI) / N;    // Step change in x
   double x = 1.5 * M_PI;                        // Initial value of x (1.5*pi)
   int i;
   float R;

   // Loop to plot the graph
   for (i = 0; x <= 15 * M_PI; x += dx2, i++) {
      // Calculate the function value
      R = (pow(sin(x / 2), 2.0) + sqrt(x));

      // Update the maximum value of the function
      if (Rmax < R) Rmax = R;

      // Draw the pixel on the graph
      // Y0 - 225 adjusts the vertical shift of the graph
      putpixel(startXcoord + i, Y0 - 225 - (int)(R * pixelToY), WHITE);
   }
}
```

```
int main() {
    char cRmax[20];
    char str[50];
    int driver, mode, graph_error;
    int zeroX;
    int pixelToX;
    int pixelToY;
    int i;
    int j;
    char kPi[10];
    char kPj[10];

    clrscr();
    driver = DETECT;
    initgraph(&driver, &mode, "C://Turboc3//BGI");
    graph_error = graphresult();

    if (graph_error != grOk) {
        cprintf("ERROR WITH GRAPH %d", graph_error);
        getchar();
        return 255;
    }
    // Get maximum window dimensions
    Xmax = getmaxx();
    Ymax = getmaxy();

    // Set the starting and ending coordinates for the graph
    X0 = 20;
    Y0 = Ymax - 19;
    X1 = Xmax - 19;
    Y1 = 10;
```

```
// Set line style and color
setlinestyle(0, 1, 3);
setcolor(WHITE);

// Draw the graph window border
rectangle(2, 0, Xmax - 2, Ymax - 30);

// Fill the graph area with black
setfillstyle(SOLID_FILL, BLACK);
floodfill(20, 20, WHITE);

// Reset line style
setlinestyle(0, 1, 1);

// Set the zero axis for X
zeroX = Y0 - 225;

// Draw the coordinate axes
line(X0, Y0 - 15, X0, Y1 - 20);  // Y-axis
line(X0, zeroX, X1, zeroX);      // X-axis

// Add text labels
outtextxy(X0 + 40, Y1 + 40, "sin^2(x/2)+sqrt(x)"); // Updated function
outtextxy(X1 + 3, zeroX + 5, "x");
outtextxy(X0 + 3, Y1 + 3, "y");
settextstyle(SMALL_FONT, HORIZ_DIR, 4);

// Define scales for the axes
pixelToX = (X1 - X0) / 16;  // X-axis scale to 16*pi
pixelToY = 15;

// Call the function to draw the graph
drawf(pixelToX, pixelToY);
```

```
    // Draw ticks and labels on the X-axis up to 16*pi
    for (i = 1; i <= 16; ++i) {  // Ticks up to 16*pi
        line(X0 + pixelToX * i, zeroX, X0 + pixelToX * i, zeroX - 5);
        sprintf(kPi, "%dPI", i);
        outtextxy(X0 + pixelToX * i - 10, zeroX + 5, kPi);
    }
    // Draw ticks and labels on the Y-axis
    for (j = 1; j <= 10; ++j) {  // Fixed loop condition
        line(X0, zeroX - pixelToY * j, X0 + 10, zeroX - pixelToY * j);
        sprintf(kPj, "%d", j);
        if (j != 10) {
            outtextxy(X0 - 15, zeroX - pixelToY * j - 5, kPj);
        } else {
            outtextxy(X0 - 20, zeroX - pixelToY * j - 5, kPj);
        }
    }
    // Display the maximum function value
    sprintf(cRmax, "%.2f", Rmax); // Format with two decimal places
    strcpy(str, "MAX(f(x)) = ");
    strcat(str, cRmax);

    setlinestyle(0, 1, 3);
    setcolor(RED);
    rectangle(X0 + 25, Ymax - 25, X0 + 200, Ymax);
    setfillstyle(SOLID_FILL, LIGHTCYAN);
    floodfill(X0 + 30, Ymax - 10, WHITE); // Fill color
    setcolor(WHITE);
    outtextxy(X0 + 30, Ymax - 20, str);
    getchar();
    closegraph();
    return 0;
}
```

Вывод

В ходе выполнения лабораторной работы была разработана программа, которая выводит на экран график заданной функции. Также была произведена разметка осей и проставлены истинные значения точек. Максимальное значение функции на заданном интервале — 7,86. Оно выводится в отдельное окно на экране.