Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут телекомунікацій, радіоелектроніки та електронної техніки

кафедра «Радіоелектронні пристрої та системи»

Лабораторна робота №20

з дисципліни«Програмування частина 2»

«Дослідження графічного режиму роботи мови програмування С»

Мета роботи: Дослідження основних принципів відображення графічної інформації на екрані дисплея

Підготував:

ст. групи АП-11

Василюк Ростислав

Прийняв:

Чайковський І.Б.

Львів 2024р

Теоретичні відомості

Для оформлення діалогу користувача з комп'ютером (програмою) потрібна розвинена система функцій управління роботою екрану. Пакет функцій управління екраном ділиться на дві частини. Перша підтримує текстовий режим (text mode) роботи. У текстовому режимі екран монітора умовно розбивається на окремі ділянки, частіше всього на 25 рядків по 80 символів (знакомісць). У кожне знакомісце може бути виведений один з 256 заздалегідь заданих символів. Друга частина забезпечує роботу екрану в графічному режимі (graphics mode). Він призначений для виведення на екран графіків, діаграм, малюнків тощо. У цьому режимі екран монітора є безліччю точок, кожна з яких може бути одним із декількох кольорів. Кількість точок по горизонталі і вертикалі називається роздільною здатністю монітора в цьому режимі.

Ініціалізація графічного режиму.

До складу графічного пакету входять:

• заголовний файл graphics.h;

• бібліотечний файл graphics.lib;

• драйвери графічних пристроїв (\*.bgi);

• шрифти (\*.chr).

Керування екраном у графічному режимі здійснюється за допомогою набору функцій, прототипи яких знаходяться в заголовному файлі graphics.h. Для роботи в графічному режимі файл graphics.h має бути підключений за

допомогою директиви #include препроцесора мови C до всіх модулів, що використовують графічні підпрограми #include <graphics.h>.

Перш ніж використовувати графічні функції, необхідно перемкнути відеоадаптер у графічний режим (за замовчуванням він знаходиться в текстовому режимі). Для ініціалізації графічного режиму призначена функція initgraph().

Завдання:

1.Нарисувати графік функції y = sin(x) шляхом табулювання її значень на ділянці [0, 3π] з кроком 0.1. Графік може зображатися за допомогою пікселів або вертикальних ліній.

#include<stdio.h>

#include<math.h>

#include<graphics.h>

#include<conio.h> #include<dos.h> main(){ float a = 5; int x = 0;

int grdrv=DETECT, grmod; initgraph(&grdrv,&grmod, "C:\\TC\\BGI");

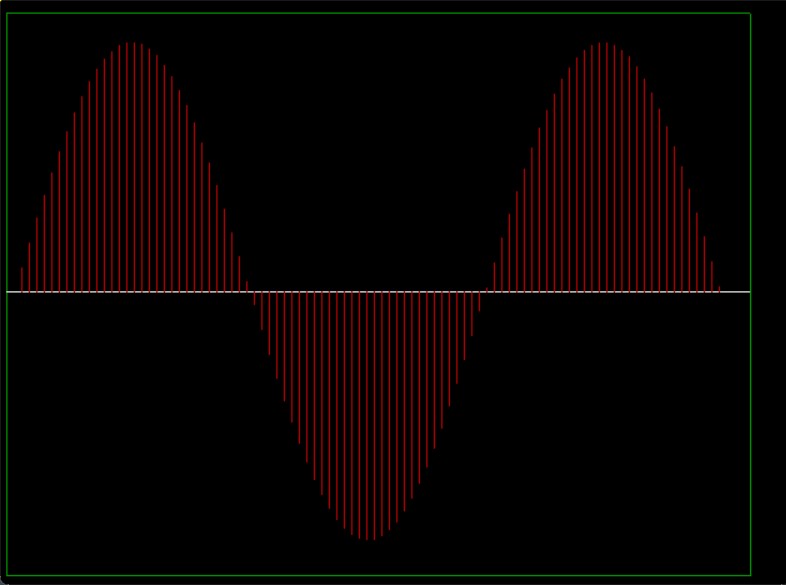
setbkcolor(BLACK); setcolor(GREEN); line(5,10,600,10); line(5,460,600,460); line(5,460,5,10); line(600,460,600,10); setcolor(WHITE); line(5,getmaxy()/2,600,getmaxy()/2); for(float i = 0; i<=3\*M\_PI;i+=0.1){

float b = sin(i); a = a+(getmaxx()/91); setcolor(RED);

line(a,getmaxy()/2,a,(getmaxy()/2)-(b\*200));

delay(10);

} getch(); closegraph();}



2.Розробити простий графічний редактор, що дозволяє набирати на екрані монітора довільний текст (з можливістю динамічного вибору шрифту ірозміру символів у діалоговому режимі). Для вирішення завдання використати функцію.

#include<stdio.h>

#include<graphics.h>

#include<conio.h> #include<windows.h> main(){

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

char a[50];

int b=1;

int c = 0;

printf("Введіть текст ");

scanf("%s",&a);

printf("Введіть множник розміру тексту ");

scanf("%d",&b);

printf("Введіть стиль тексту ");

scanf("%d",&c);

int grdrv=DETECT, grmod;

initgraph(&grdrv,&grmod, "C:\\TC\\BGI");

setbkcolor(BLACK);

setcolor(RED); settextstyle(c, 0, b); outtextxy(100,100, a); getch(); closegraph(); }



3. Розробити програму для виведення в графічному режимі зображения об'єкта (кулі), що рухається по діагоналі екрану.

Для імітації руху зображення об'єкта на екрані необхідно виконати такий алгоритм: 1. Нарисувати об’єкт у заданій точці. 2. Витерти об’єкт, замалювавши його кольором тла. 3. Змінити координати об’єкта. 4. Перейти до пункту 1.

#include <graphics.h> #include <conio.h> int main() {

int gd = DETECT, gm; // Змінні для графічного режиму та графічного режиму

initgraph(&gd, &gm, ""); // Ініціалізація графічного режиму int x = getmaxx()/2, y = getmaxy()/2; // Початкові координати круга int dx = 1, dy = 1, r = 15; // Кроки переміщення по осях x та y

// Безкінечний цикл, поки не буде натиснута клавіша

while (!kbhit()) {

cleardevice(); // Очищення вікна

int color = RED;

circle(x, y, r); // Малювання круга з центром у (x, y) та радіусом 15 x += dx; // Зміна координати x y += dy; // Зміна координати y

delay(2); // Затримка для плавного анімованого ефекту

// Перевірка зіткнення з межами вікна по осі x

if (x <= r || x >= getmaxx()-r) {

dx = -dx; // Зміна напрямку руху по осі x color = WHITE; setcolor(color);

}

// Перевірка зіткнення з межами вікна по осі y

if (y <= r || y >= getmaxy()-r) {

dy = -dy; // Зміна напрямку руху по осі y color= RED; setcolor(color);

}

}

closegraph(); // Закриття графічного вікна

return 0;



4. Розробити програму для виведення на екран у графічному режимі зображения кулі, що обертається.

#include <graphics.h> #include <conio.h> main(){ int xr=100; int gd = DETECT, gm; initgraph(&gd, &gm, ""); setcolor(WHITE); while (!kbhit()) { for (xr = 100; xr >= 0; xr-=2) { cleardevice();

fillellipse(getmaxx()/2, getmaxy()/2, xr, 100); delay(20);

}

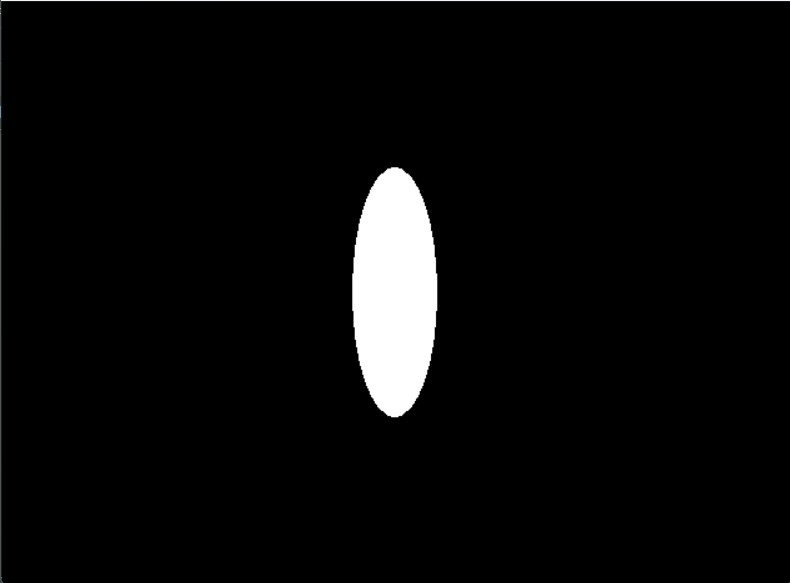
for (xr = 0; xr <= 100; xr+=2) { cleardevice();

fillellipse(getmaxx()/2, getmaxy()/2, xr, 100); delay(20);

}

}

closegraph(); }



5. Здійснити рух графічного об'єкта зліва направо по екрану. Для вирішення завдання використати функцію.

#include<graphics.h>

#include<conio.h> #include<math.h> main(){

int grdrv=DETECT, grmod,size; initgraph(&grdrv,&grmod, "C:\\TC\\BGI"); setbkcolor(CYAN); int x=0,y=getmaxy()\*4/5-10;

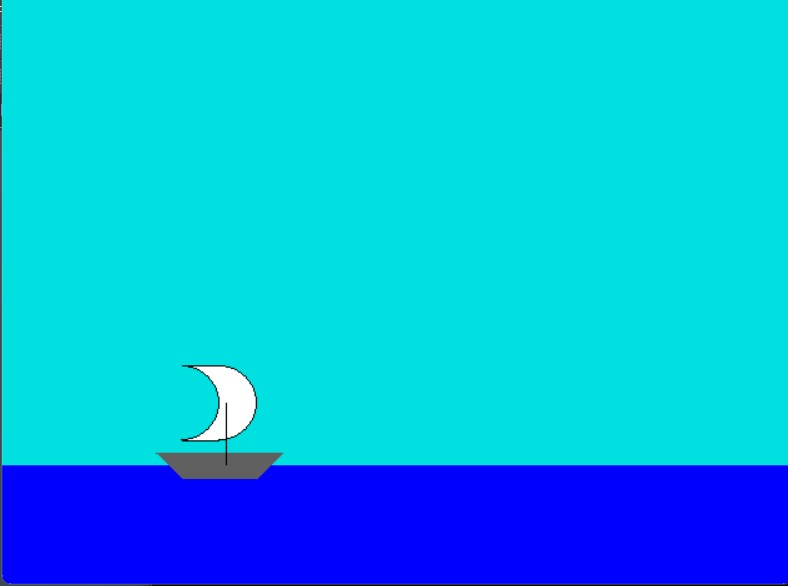
void \*buf; cleardevice(); setcolor(BLUE); setfillstyle(SOLID\_FILL, BLUE); bar(0,getmaxy()\*4/5,getmaxx()+1,getmaxy()+1); setcolor(DARKGRAY); setfillstyle(SOLID\_FILL, DARKGRAY);

int base[] = {x, y, x + 100, y, x + 80, y+20, x + 20, y+20}; fillpoly(4, base); setcolor(BLACK); setfillstyle(SOLID\_FILL, WHITE); arc(50,getmaxy()\*4/5-50,-90,90,30); arc(20,getmaxy()\*4/5-50,-90,90,30); line(20,getmaxy()\*4/5-80,50,getmaxy()\*4/5-80); line(20,getmaxy()\*4/5-20,50,getmaxy()\*4/5-20); line(55,getmaxy()\*4/5-50,55,getmaxy()\*4/5); floodfill(60,getmaxy()\*4/5-50 , BLACK); buf=malloc(size);

getimage(0,getmaxy()\*4/5-90,110,getmaxy()\*4/5+20, buf);

setfillstyle(1, CYAN); for(int i=1;i<=getmaxx();i++){ bar(i - 1,getmaxy()\*4/5-90,110,getmaxy()\*4/5); putimage(i, getmaxy()\*4/5-90, buf, COPY\_PUT);

delay(10);} closegraph();}



Контрольні запитання.

1.Текстовий і графічний режими роботи в мові програмування С.

У мові програмування C існують два основні режими роботи: текстовий та графічний.

Текстовий режим:

Це стандартний режим вводу/виводу в C, який використовує ASCII-символи.

Функції для роботи в текстовому режимі включають printf(), scanf(), puts(), gets() тощо.

Текстовий режим дозволяє виводити текст, числа та інші дані на консоль або в текстовий файл.

Графічний режим:

Графічний режим дозволяє малювати точки, лінії, фігури та інші графічні об'єкти на екрані.

Для переходу в графічний режим використовуються спеціальні функції, такі як initgraph() та closegraph().

Графічний режим дає більше можливостей для створення інтерактивних програм та ігор.

2.Керування кольором і вибір палітри в мові С.

У графічному режимі мови C можна керувати кольорами та вибирати палітру:

Функція setcolor() встановлює колір для малювання графічних об'єктів.

Функція setbkcolor() встановлює колір фону.

Функція getmaxcolor() повертає максимальне число кольорів, доступних на даній системі.

Функція getpalette() повертає поточну палітру кольорів.

Функція setpalette() дозволяє змінювати палітру кольорів.

3.Основні функції для графічного режиму роботи в мові С.

Деякі з основних функцій для графічного режиму в C:

initgraph() ініціалізує графічний режим.

closegraph() завершує роботу в графічному режимі.

putpixel() малює один піксель на екрані.

line() малює лінію між двома точками.

rectangle() малює прямокутник.

circle() малює коло.

arc() малює дугу.

fillpoly() заповнює многокутник.

4.Принципи роботи з частинами графічного екрану в мові С.

Принципи роботи з частинами графічного екрану в мові C

У графічному режимі мови C можна працювати з окремими частинами екрану:

Функція getviewsettings() повертає поточні налаштування віртуального екрану.

Функція setviewport() встановлює віртуальне "вікно" на екрані, в якому будуть малюватися графічні об'єкти.

Функція clearviewport() очищає поточне "вікно" на екрані.

Функція moveto() переміщує "перо" малювання в задану позицію.

Функція lineto() малює лінію від поточної позиції "пера" до нової.