МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут телекомунікацій, радіоелектроніки та електронної техніки
Кафедра «Радіоелектронні пристрої та системи»

Звіт з лабораторної роботи № 20

«Програмування, частина 2»

Підготував:
ст. гр. АП-11
Заброварний Олег
Перевірив:
Асистент каф РЕПС
Чайковський І.Б.

Львів 2024

Тема роботи: Дослідження графічного режиму роботи мови програмування С.
Мета роботи: Дослідження основних принципів відображення графічної інформації на екрані дисплея.

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).
- Кольори задають числами, або англійськими назвами

BLACK (0)	чорний	DARKGRAY (8)	темний сірий
BLUE (1)	синій	LIGHTBLUE (9)	яскравий синій
GREEN (2)	зелений	LIGHTGREEN (10)	яскравий зелений
CYAN (3)	блакитний	LIGHTCYAN (11)	яскравий блакитний
READ (4)	червоний	LIGHTRED (12)	червоний
MAGENTA (5)	фіолетовий	LIGHTMAGENTA (13)	фіолетовий
BROWN (6)	коричневий	YELLOW (14)	жовтий
LIGHTGRAY (7)	світлий сірий	WHITE (15)	білий

Основні функції для графічних побудов:

setcolor (колір);

setbkcolor (колір) колір тла.;

putpixel (z,y,колір);

line(x1,y1,x2,y2);

lineto(x,y);

bar(x1,y1,x2,y2);

rectangle(x1,y1,x2,y2);

circle(x,y,R);

arc(x,y,початк. кут, кінцевий кут, радіус);

closegraph();

outtext(текст);

outtextxy(x,y,текст);

settextstyle(шрифт, напрям, розмір);

getmaxx() – повертає розмір екрану по горизонталі;

getmaxy() – повертає розмір екрану по вертикалі;

getcolor() – повертає значення поточного кольору;

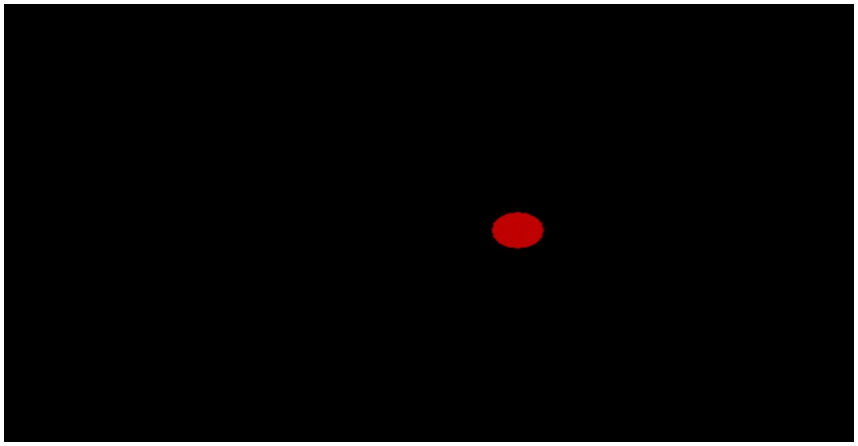
getx(), gety() – повертають координати поточного пікселя.

Завдання 1

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
main(){
   /* автоматичне визначення графічного драйвера */
```

```
    int gdriver = DETECT, gmode, errorcode;  // gdriver і gmode використовуються для автоматичного визначення графічного режиму, errorcode для перевірки помилок

    int i, size;  // і використовується як лічильник в циклі, size для зберігання розміру буфера

    void *buf;  // указівник для зберігання зображення

    /* ініціалізація графічного режиму */

    initgraph(&gdriver, &gmode, "");  // Ініціалізація графічного режиму з автоматичним визначенням драйвера і режиму

    /* перевірка результату ініціалізації */

    errorcode = graphresult();  // Отримання коду результату ініціалізації

    /* якщо сталася помилка */

    if (errorcode != grOk){

        printf("Графічна помилка: %s\n", grapherrormsg(errorcode));  // Виведення повідомлення про помилку

        printf("Натисніть будь-яку клавішу для завершення:");  // Запит на натискання клавіші для завершення

        getch();  // Очікування натискання клавіші

        exit(1);  // Завершення програми з кодом 1 (помилка)

    }

    setcolor(BLACK);  // Встановлення кольору для подальших графічних операцій

    setfillstyle(1, RED);  // Встановлення стилю заливки (1 - суцільна заливка, RED - червоний колір)

    fillellipse(21, 240, 20, 20);  // Малювання заповненого еліпса з центром у точці (21, 240) і радіусами 20

    /* розрахунок розміру буфера, необхідного для збереження зображення */

    size = imagesize(1, 220, 41, 260);  // Обчислення розміру пам'яті, необхідної для збереження області зображення

    buf = malloc(size);  // Виділення пам'яті для збереження зображення
```

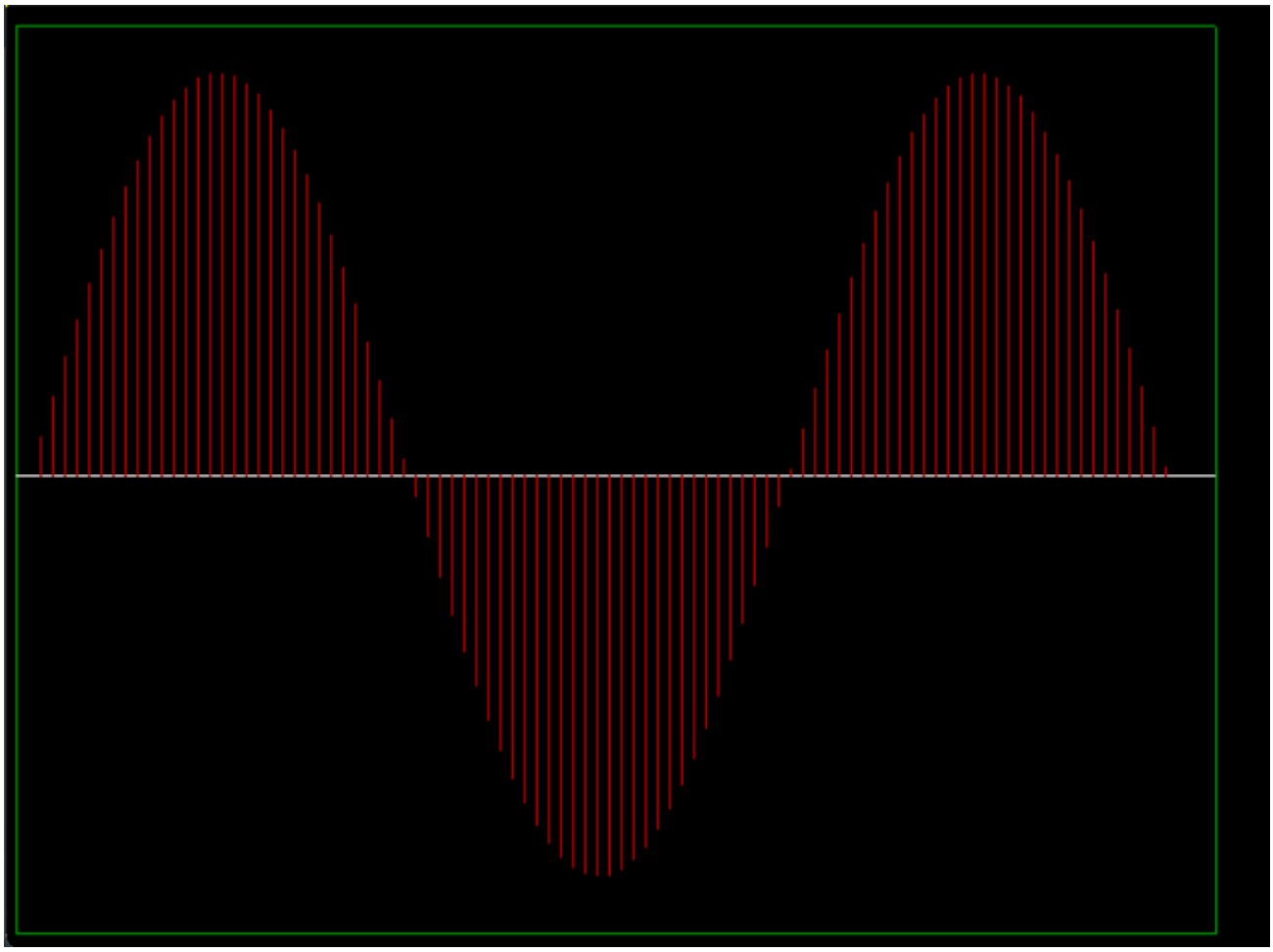
```
    if (buf == NULL) {
        printf("Помилка виділення пам'яті\n");  // Виведення повідомлення про помилку, якщо пам'ять не була виділена
        closegraph();  // Закриття графічного режиму
        exit(1);  // Завершення програми з кодом 1 (помилка)
    }
    getimage(1, 220, 41, 260, buf);  // Збереження області екрана розміром 41x260 з координатами лівого верхнього кута (1, 220) у буфер
    setfillstyle(1, BLACK);  // Встановлення стилю заливки (1 - суцільна заливка, BLACK - чорний колір)
    for(i = 1; i <= 620; i++)  // Цикл для переміщення еліпса по екрану
    {
        bar(i - 1, 220, i + 39, 260);  // Очищення попереднього положення еліпса. Задається прямокутник, який перекриває попереднє положення еліпса
        putimage(i, 220, buf, COPY_PUT);  // Відображення зображення з буфера у новій позиції. (i, 220) - нова координата лівого верхнього кута зображення
        delay(10);  // Затримка на 10 мс для створення ефекту анімації
    }
    free(buf);  // Звільнення виділеної пам'яті для буфера
    closegraph();  // Закриття графічного режиму і повернення екрану у текстовий режим
 return 0;  }
```



Завдання 2

```
#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();}
```



Завдання 3

```
#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;
```

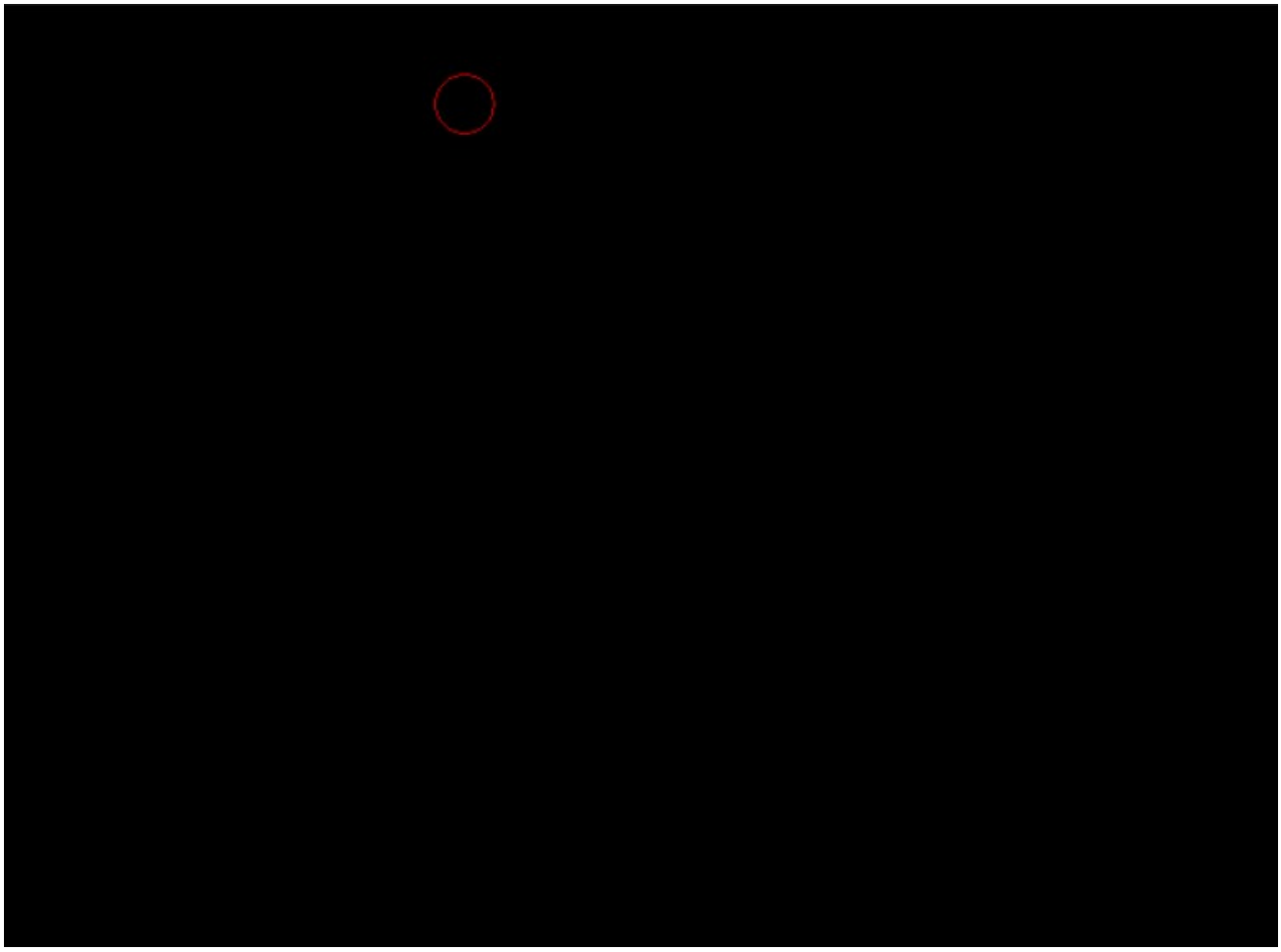
```
    printf("Введіть текст  ");
    scanf("%s",&a);
    printf("Введіть множник розміру тексту  ");
    scanf("%d",&b);
    printf("Введіть стиль тексту  ");
    scanf("%d",&c);
    int grdrv=DETECT, grmod;
    initgraph(&grdrv,&grmod, "C:\\TC\\BGI");
    setbkcolor(BLACK);
    setcolor(RED);
    settextstyle(c, 0, b);
    outtextxy(100,100, a);
    getch();
    closegraph();
}
```


```
Введіть текст  Oleh787
Введіть множник розміру тексту  10
Введіть стиль тексту  3
```

Завдання 4

```
#include <graphics.h>
#include <conio.h>
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int x = getmaxx()/2, y = getmaxy()/2;
```

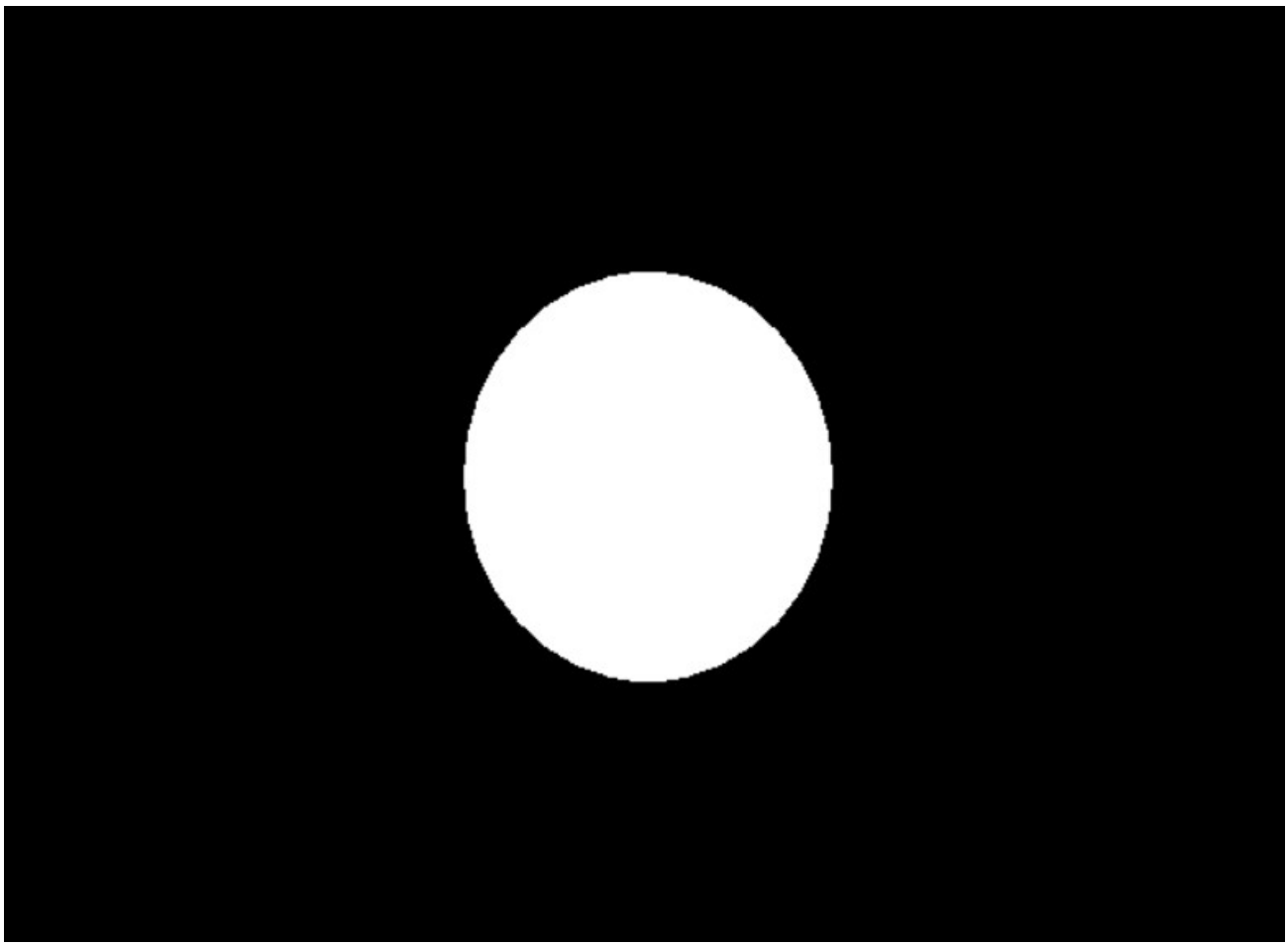
```
    int dx = 1, dy = 1, r = 15;
    while (!kbhit()) {
        cleardevice();
        int color = RED;
        circle(x, y, r);
        x += dx;
        y += dy;
        delay(2);
        if (x <= r || x >= getmaxx()-r) {
            dx = -dx;
            color = WHITE;
            setcolor(color);
        }
        if (y <= r || y >= getmaxy()-r) {
            dy = -dy;
            color= RED;
            setcolor(color);
        }
    }
    closegraph();  // Закриття графічного вікна
        return 0;
}
```



Завдання 5

```
#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();
}
```

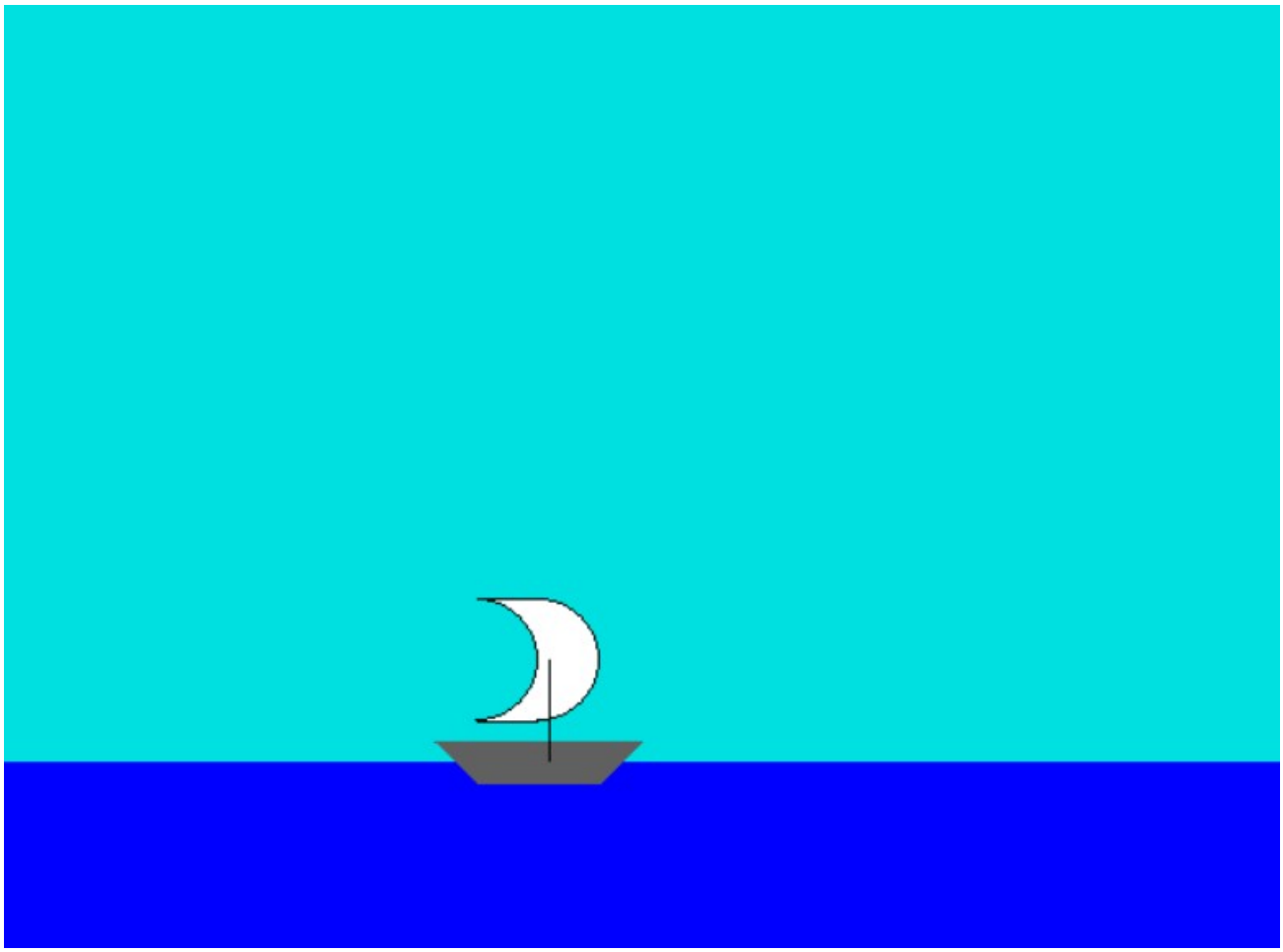


Завдання 6

```
#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();
}
```



Висновок: На даній лабораторній роботі я ознайомився з дослідженням графічного режиму роботи мови програмування С.