

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”  
ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ ТА РАДІОЕЛЕКТРОНІКИ



**Лабораторна робота №20**

**Виконав :**

Ст. гр АП-11

Чаус Владислав

**Прийняла:**

Чайковський І. Б.

## Львів 2024

### Тема:

Дослідження графічного режиму роботи мови програмування C.

### Мета:

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.

### Завдання:

1. Нарисувати графік функції  $y = \sin(x)$  шляхом табулювання її значень на ділянці  $[0, 3\pi]$  з кроком 0.1. Графік може зображатися за допомогою пікселів або вертикальних ліній. Приклад: графік лінійно зростаючих чисел
2. Розробити простий графічний редактор, що дозволяє набирати на екрані монітора довільний текст (з можливістю динамічного вибору шрифту і розміру символів у діалоговому режимі). Для вирішення завдання використати функцію.


3. Розробити програму для виведення в графічному режимі зображення 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.
4. Розробити програму для виведення на екран у графічному режимі зображення кулі, що обертається.
5. Здійснити рух графічного об'єкта зліва направо по екрану. Для вирішення завдання використати функцію.

## Виконання роботи:

```

1 #include <stdio.h>
2 #include <math.h>
3 #include <graphics.h>
4
5 #define PI 3.14159265
6
7 int main() {
8     int gd = DETECT, gm;
9     initgraph(&gd, &gm, "");
10
11     double x, y;
12     double scale = 100.0;
13     double step = 0.1;
14     int x_pixel, y_pixel, prev_x_pixel, prev_y_pixel;
15
16     prev_x_pixel = 0;
17     prev_y_pixel = getmaxy() / 2;
18
19     for (x = 0; x <= 3 * PI; x += step) {
20         y = sin(x);
21         x_pixel = (int)(x * scale);
22         y_pixel = (int)(getmaxy() / 2 - y * scale);
23
24         line(prev_x_pixel, prev_y_pixel, x_pixel, y_pixel);
25
26         prev_x_pixel = x_pixel;
27         prev_y_pixel = y_pixel;
28     }
29
30     getch();
31     closegraph();
32     return 0;
33 }
34

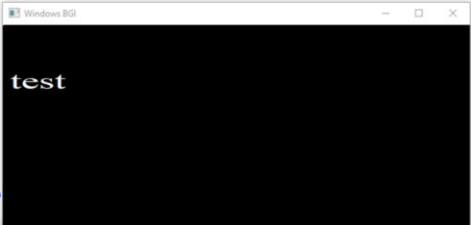
```



```

1 #include <stdio.h>
2 #include <graphics.h>
3
4 int main() {
5     int gd = DETECT, gm;
6     char text[100];
7     int x, y, fontSize, fontType;
8
9     printf("Vvedit text: ");
10    gets(text);
11
12    printf("Vvedit x ta y: ");
13    scanf("%d %d", &x, &y);
14
15    printf("Vvedit rozmir shrift (1-5): ");
16    scanf("%d", &fontSize);
17
18    printf("Oberit typ shrift (0-10):\n");
19    printf("0: DEFAULT_FONT, 1: TRIPLEX_FONT, 2: SMALL_FONT, 3: SANS_SERIF_FONT, 4: GOTHIC_FONT,\n");
20    printf("5: SCRIPT_FONT, 6: SIMPLEX_FONT, 7: TRIPLEX_SCR_FONT, 8: COMPLEX_FONT, 9: EUROPEAN_FONT, 10: BOLD_FONT\n");
21    scanf("%d", &fontType);
22
23    initgraph(&gd, &gm, "");
24    settextstyle(fontType, HORIZ_DIR, fontSize);
25    outtextxy(x, y, text);
26
27    getch();
28    closegraph();
29    return 0;
30 }
31

```





```

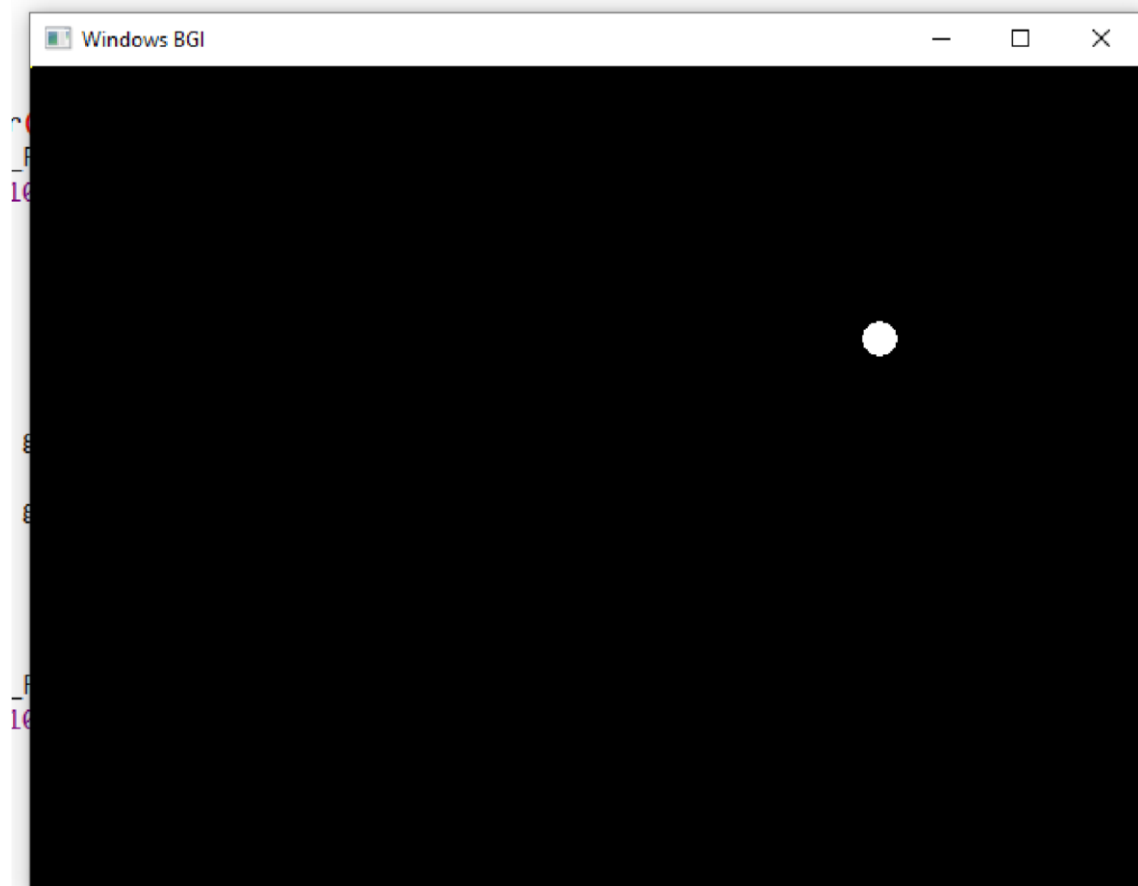
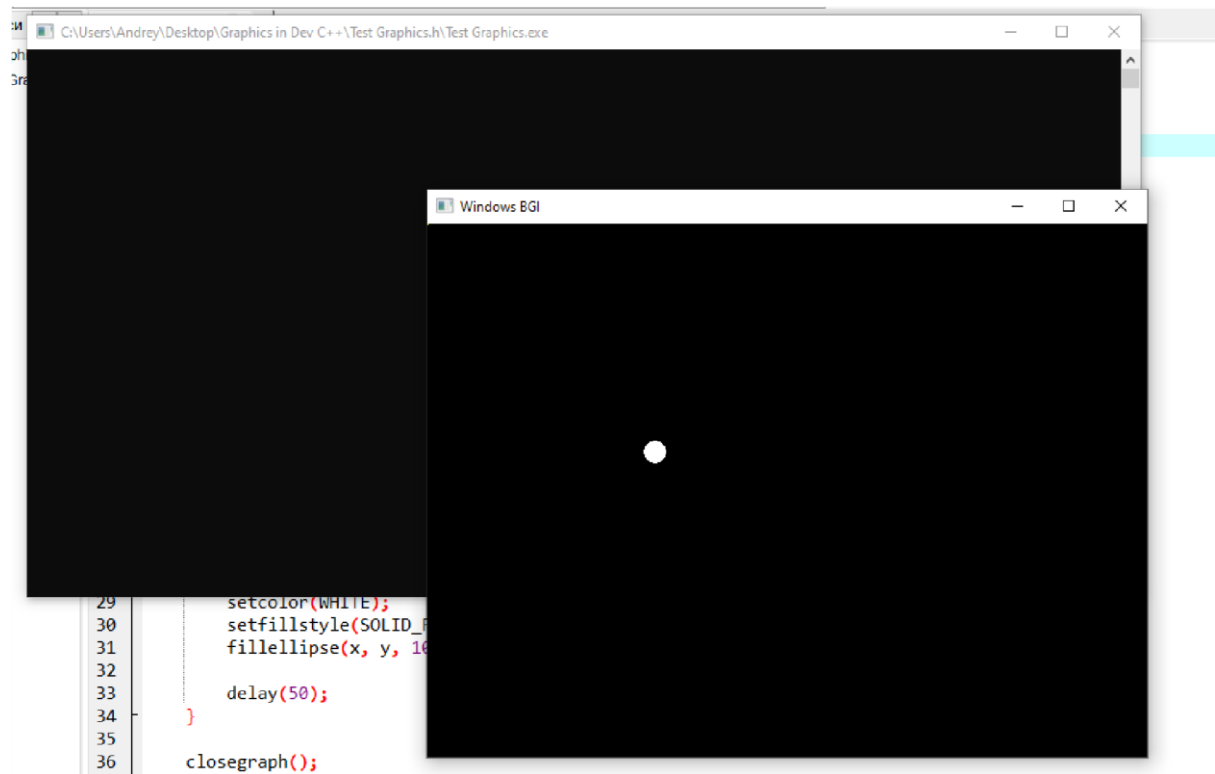
1  #include <stdio.h>
2  #include <graphics.h>
3
4  int main() {
5      int gd = DETECT, gm;
6      char text[100];
7      int x, y, fontSize, fontType;
8
9      printf("Vvedit text: ");
10     gets(text);
11
12     printf("Vvedit x ta y: ");
13     scanf("%d %d", &x, &y);
14
15     printf("Vvedit rozmir shrift (1-5): ");
16     scanf("%d", &fontSize);
17
18     printf("Oberit typ shrift (0-10):\n");
19     printf("0: DEFAULT_FONT, 1: TRIPLEX_FONT, 2: SMALL_FONT, 3: SANS_SERIF_FONT, 4: GOTHIC_FONT,\n");
20     printf("5: SCRIPT_FONT, 6: SIMPLEX_FONT, 7: TRIPLEX_SCR_FONT, 8: COMPLEX_FONT, 9: EUROPEAN_FONT, 10: BOLD_FONT\n");
21     scanf("%d", &fontType);
22
23     initgraph(&gd, &gm, "");
24     settextstyle(fontType, HORIZ_DIR, fontSize);
25     outtextxy(x, y, text);
26
27     getch();
28     closegraph();
29     return 0;
30 }
31

```

```

5  int main() {
6      int gd = DETECT, gm;
7      initgraph(&gd, &gm, "");
8
9      int x = 100, y = 100;
10     int dx = 1, dy = 1;
11
12     while (!kbhit()) {
13
14         setcolor(getbkcolor());
15         setfillstyle(SOLID_FILL, getbkcolor());
16         fillellipse(x, y, 10, 10);
17
18
19         x += dx;
20         y += dy;
21
22
23         if (x <= 0 || x >= getmaxx())
24             dx = -dx;
25         if (y <= 0 || y >= getmaxy())
26             dy = -dy;
27
28
29         setcolor(WHITE);
30         setfillstyle(SOLID_FILL, WHITE);
31         fillellipse(x, y, 10, 10);
32
33         delay(50);
34     }
35
36     closegraph();
37     return 0;
38 }
39

```



## Контрольні запитання:

### 1. Текстовий і графічний режими роботи в мові програмування C.

У мові програмування C текстовий і графічний режими роботи можна використовувати за допомогою бібліотеки, такої як `'ncurses'` для текстового інтерфейсу або `'SDL'` для графічного інтерфейсу. Текстовий режим дає можливість працювати з консольним виведенням і вводом, в той час як графічний режим відкриває можливості для створення віконних додатків з використанням графіки.

### 2. Керування кольором і вибір палітри в мові C.

У мові C керування кольором та вибір палітри в графічному режимі зазвичай здійснюється за допомогою функцій, наданих графічними бібліотеками, наприклад, функцій з бібліотеки `'SDL'`. Ці функції дозволяють встановлювати колір для кожного пікселя або області екрану, а також працювати з палітрою, змінюючи колірну схему екрану.

### 3. Основні функції для графічного режиму роботи в мові C.

Основні функції для графічного режиму роботи в мові C можуть відрізнятися залежно від використовуваної бібліотеки. Наприклад, якщо використовується бібліотека `'SDL'`, деякі з основних функцій можуть включати `'SDL_Init()'` для ініціалізації SDL, `'SDL_CreateWindow()'` для створення вікна, `'SDL_CreateRenderer()'` для створення рендерера тощо.

### 4. Принципи роботи з частинами графічного екрану в мові C.

У мові C, для роботи з частинами графічного екрану, ви зазвичай використовуєте функції, що дозволяють малювати або маніпулювати пікселями або графічними об'єктами на екрані. Ці функції можуть включати встановлення кольору, розміру, позиції елементів, зміну їхнього стану і так далі. Наприклад, у бібліотеці `'SDL'` для малювання на екрані використовуються функції, такі як `'SDL_RenderDrawPoint()'`,

``SDL_RenderDrawLine()`, `SDL_RenderFillRect()`, які дозволяють малювати пікселі, лінії, прямокутники відповідно.`