**Ministerul Educației, Culturii și Cercetării**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Raport**

Lucrarea de laborator nr.2

Disciplina: Programarea pilotată de evenimente.

Tema: Studierea primitivelor oferite de către interfața GDI

**Efectuat**: st.gr. TI-207 Bunescu Gabriel.

**Verificat**: asist. univ. Gaidarji Alina

Chișinău 2022

**Scopul lucrării:** Studierea primitivelor oferite de către interfața GDI

**Codul sursă:**

1. #include <windows.h>
2. #include <math.h>
3. #define NUM 1000
4. #define TWOPI (2 \* 3.14159)
5. int a[3][6] = { { 123, 343, 100, 163, 232, 166 }, { 128, 15, 129, 74, 234, 255 }, { 1, 2,3,4,1,4} };
6. int circle(HWND hwnd, HDC hdc, int i, int j, int dir, RECT rect, int nr) {
7. HPEN hYellowPen; HBRUSH hRedBrush;
8. hYellowPen = CreatePen(PS\_SOLID, 1, RGB(255, 255, 0)); hRedBrush = CreateSolidBrush(RGB(255, 0, 0)); if (j + 150 > rect.bottom)
9. {
10. if (dir == 1)a[2][nr] = 2; else a[2][nr] = 3;
11. }
12. if (i + 150 > rect.right)
13. {
14. if (dir == 2)a[2][nr] = 3; else a[2][nr] = 4;
15. }
16. if (j + 50 < rect.top)
17. {
18. if (dir == 3)a[2][nr] = 4; else a[2][nr] = 1;
19. }
20. if (i + 50 < rect.left)
21. {
22. if (dir == 4)a[2][nr] = 1; else a[2][nr] = 2;
23. }
24. switch (a[2][nr])
25. {
26. case 1: {
27. a[0][nr]++; a[1][nr]++; break;
28. }
29. case 2: {
30. a[0][nr]++; a[1][nr]--; break;
31. }
32. case 3: {
33. a[0][nr]--; a[1][nr]--; break;
34. }
35. case 4: {
36. a[0][nr]--; a[1][nr]++; break;
37. } }
38. switch (nr)
39. {
40. case 0: {SelectObject(hdc, hYellowPen); SelectObject(hdc, hRedBrush); break; }
41. }
42. Rectangle(hdc, a[0][nr] + 50, a[1][nr] + 50, a[0][nr] + 150, a[1][nr] + 150);
43. DeleteObject(hYellowPen); DeleteObject(hRedBrush);
44. return dir;
45. }
46. LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);
47. int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, PSTR szCmdLine, int iCmdShow)
48. {
49. static char szAppName[] = "SineWave";
50. HWND hwnd;
51. MSG msg;
52. WNDCLASSEX wndclass;
53. wndclass.cbSize = sizeof(wndclass);
54. wndclass.style = CS\_HREDRAW | CS\_VREDRAW;
55. wndclass.lpfnWndProc = WndProc;
56. wndclass.cbClsExtra = 0;
57. wndclass.cbWndExtra = 0;
58. wndclass.hInstance = hInstance;
59. wndclass.hIcon = LoadIcon(NULL, IDI\_APPLICATION);
60. wndclass.hCursor = LoadCursor(NULL, IDC\_ARROW);
61. wndclass.hbrBackground = (HBRUSH)GetStockObject(WHITE\_BRUSH);
62. wndclass.lpszMenuName = NULL; wndclass.lpszClassName = (LPCWSTR)szAppName; wndclass.hIconSm = LoadIcon(NULL, IDI\_APPLICATION); RegisterClassEx(&wndclass);
63. hwnd = CreateWindow((LPCWSTR)szAppName, L"Interfata GDI/Bunescu Gabriel Gr.TI-207",
64. WS\_OVERLAPPEDWINDOW,
65. CW\_USEDEFAULT, CW\_USEDEFAULT,
66. CW\_USEDEFAULT, CW\_USEDEFAULT,
67. NULL, NULL, hInstance, NULL); ShowWindow(hwnd, iCmdShow); UpdateWindow(hwnd);
68. while (GetMessage(&msg, NULL, 0, 0))
69. {
70. TranslateMessage(&msg); DispatchMessage(&msg);
71. }
72. return msg.wParam;
73. }
74. LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)
75. {
76. static int cxClient, cyClient;
77. HDC hdc, hCompatibleDC;
78. int i; PAINTSTRUCT ps;
79. POINT pt[NUM];
80. PAINTSTRUCT PaintStruct;
81. HANDLE hBitmap, hOldBitmap;
82. RECT Rect;
83. BITMAP Bitmap;
84. switch (iMsg)
85. {
86. case WM\_SIZE:
87. cxClient = LOWORD(lParam);
88. cyClient = HIWORD(lParam); return 0;
89. case WM\_PAINT:
90. hdc = BeginPaint(hwnd, &ps);
91. hBitmap = LoadImage(NULL, L"asd.BMP",
92. IMAGE\_BITMAP, 700, 430, LR\_LOADFROMFILE);
93. GetObject(hBitmap, sizeof(BITMAP), &Bitmap); hCompatibleDC = CreateCompatibleDC(hdc); hOldBitmap = SelectObject(hCompatibleDC, hBitmap); GetClientRect(hwnd, &Rect);
94. StretchBlt(hdc, 0, 0, cxClient, cyClient, hCompatibleDC, 0, 0, Bitmap.bmWidth, Bitmap.bmHeight, SRCCOPY);
95. SelectObject(hCompatibleDC, hOldBitmap);
96. DeleteObject(hBitmap);
97. DeleteDC(hCompatibleDC);
98. DrawText(hdc, L"Lucrare de laborator Nr2 ", -1, &Rect,
99. DT\_SINGLELINE | DT\_CENTER | DT\_VCENTER);
100. for (i = 0; i < NUM; i++) {
101. pt[i].x = i \* cxClient / NUM;
102. pt[i].y = (int)(cyClient / 2 \*
103. (1 - sin(TWOPI \* i / NUM)));
104. }
105. Polyline(hdc, pt, NUM);
106. InvalidateRect(hwnd, NULL, TRUE);
107. circle(hwnd, hdc, a[0][0], a[1][0], a[2][0], Rect, 0); Sleep(50);
108. EndPaint(hwnd, &PaintStruct); return 0;
109. case WM\_DESTROY: PostQuitMessage(0); return 0;
110. }
111. return DefWindowProc(hwnd, iMsg, wParam, lParam);
112. }

**Rezultatul execuției:**

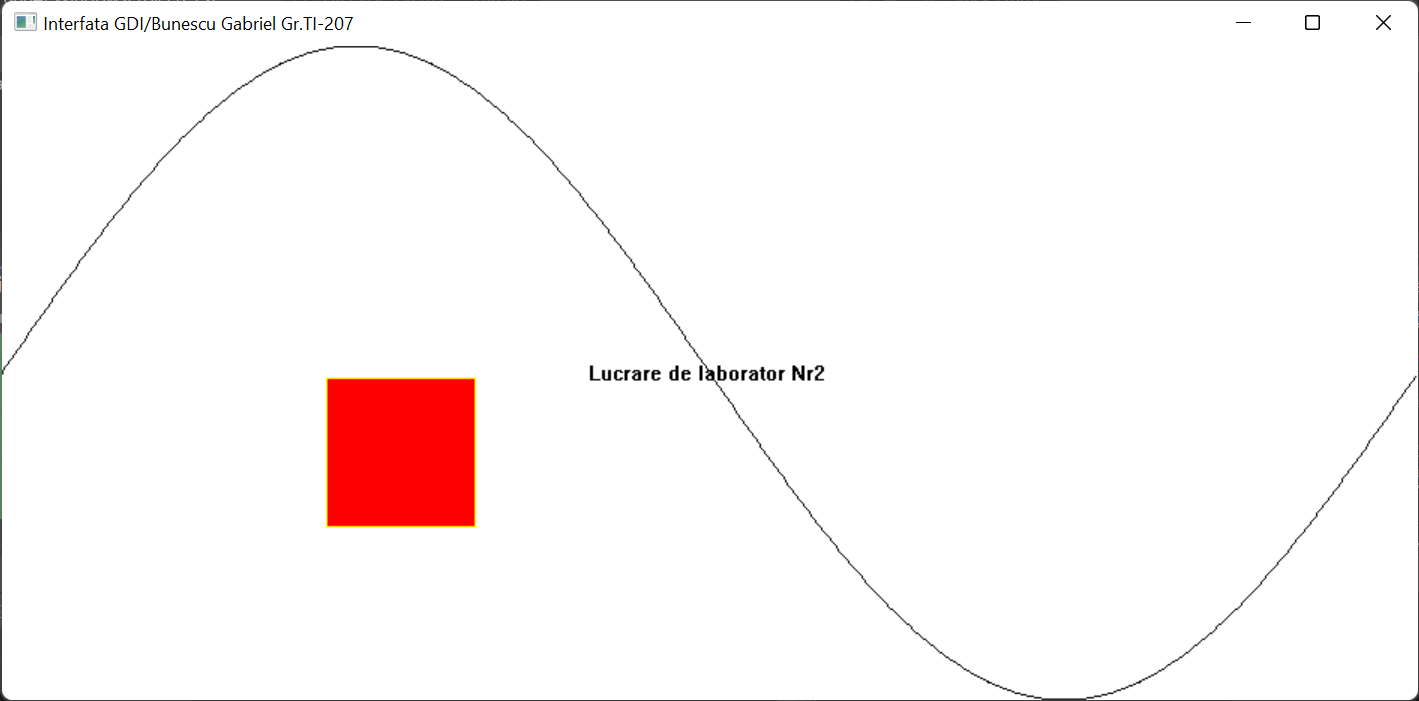


Figura 1. Rezultatul execuției

**Concluzie:**

Pe parcursul elaborării acestei lucrări de laborator eu m-am făcut cunoscut cu Aplicații Windows, am studiat despre primitivele oferite de interfața GDI, pe parcurs am obținut cunoștințe în crearea obiectelor și deplasarea acestor în windows.