

RAPORT

Lucrare de laborator nr. 2

Tema: Interfața GDI

A efectuat:
A verificat:

student a gr. TI-102 ***Sili Gheorghe***
lector universitar ***N. Sava***

Scopul lucrării: De studiat primitivele oferite de interfața GDI

Sarcina lucrării: Scrieți un program care afișează în zona de client un desen animat, utilizând toate primitivele GDI.

Codul sursă:

```
#include <windows.h>
#include <windowsx.h>
#include <math.h>

LRESULT CALLBACK WndProc(HWND,UINT,WPARAM,LPARAM);

/* GLOBALS */
int timp=-300, dtime = 0, position;

POINT coords[4], pt_arr[8];
HBRUSH hRedBrush, hBlueBrush, hGreenBrush, hRandBrush, hSurBrush, hYellowBrush;
/*****/

void AnimateCar(HDC);
void Semafor(HDC);
void Intersectie(HDC);
void Copyright(HDC);

int WINAPI WinMain(HINSTANCE hInst,HINSTANCE hPrevInst,PSTR szCmdLine,int iCmdShow)
{
    char szClassName[]="Lab2_AAA";
    char szWindowName[]="Lab. nr.2 | Sili Gheorghe (TI-102)";
    HWND hwnd;
    MSG msg;
    WNDCLASS wc;

    memset(&wc,0,sizeof(wc));
    wc.style= CS_HREDRAW | CS_VREDRAW;
    wc.lpfnWndProc= WndProc;
    wc.hInstance= hInst;
    wc.hbrBackground= CreateSolidBrush(RGB(230, 230, 230));
    wc.lpszClassName= szClassName;
    RegisterClass(&wc);

    hwnd=CreateWindow(
        szClassName,
        szWindowName,
        WS_OVERLAPPEDWINDOW,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        1000,
        700,
        NULL,
        NULL,
        hInst,
        NULL
    );
    ShowWindow(hwnd,SW_SHOWNORMAL);

    while(GetMessage(&msg,NULL,0,0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return 0;
}

LRESULT CALLBACK WndProc(HWND hwnd,UINT msg,WPARAM wParam,LPARAM lParam)
```

```

{

switch(msg){
case WM_CREATE:
    hRedBrush = CreateSolidBrush(RGB(255, 0, 0));
    hBlueBrush= CreateSolidBrush(RGB(0, 0, 255));
    hGreenBrush = CreateSolidBrush(RGB(0, 255, 0));
    hRandBrush = CreateSolidBrush(RGB(152, 45, 10));
    hSurBrush = CreateSolidBrush(RGB(206, 206, 206));
    hYellowBrush = CreateSolidBrush(RGB(240, 255, 13));

    SetTimer(hwnd, 0, 1, NULL);
    break;

case WM_TIMER:
    timp++;
    dtime++;
    InvalidateRect(hwnd, NULL, TRUE);
    break;

case WM_PAINT:
    HDC hdc;
    PAINTSTRUCT ps;

    hdc = BeginPaint(hwnd, &ps);

    AnimateCar(hdc);
    Intersectie(hdc);
    Semafor(hdc);
    Copyright(hdc);

    EndPaint(hwnd, &ps);
    break;

case WM_DESTROY:
    PostQuitMessage(0);
    return 0;
}
return(DefWindowProc(hwnd,msg,wParam,lParam));
}

void Copyright(HDC hdc)
{
    TextOut (hdc, 100, 20, "Efectuat: Sili Gheorghe | TI-102", 33) ;
    TextOut (hdc, 640, 20, "Controlat: N.Sava", 17) ;
}

void AnimateCar(HDC hdc)
{
    SelectBrush(hdc,hBlueBrush);

    // drumul este intre bottom=700, right=1000, i_bottom=350, i_top=250, i_left=450, i_right=550

    if(dtime < 100) position = -150; else if(dtime < 520) position++;
    else if(dtime > 1200) position++;

    Rectangle(hdc, position, 270, position+150, 330);
    SetTextColor(hdc, RGB(0, 0, 0));
    TextOut (hdc, position+20, 290, "Laboratorul nr.2", 16) ;

    return;
}

void Semafor(HDC hdc)
{
    SelectBrush(hdc,hSurBrush);

```

```

// Top-left
Rectangle(hdc, 400, 120, 440, 240);
if(timp >= 300 && timp <= 600 || (timp >= 650 && timp <= 690) || (timp >= 730 && timp <= 770) || (timp >= 810 &&
timp <= 850)) SelectBrush(hdc,hGreenBrush);
Ellipse(hdc, 400, 120, 440, 160);
SelectBrush(hdc,hSurBrush);
if(timp >= 200 && timp <= 300) SelectBrush(hdc,hYellowBrush);
Ellipse(hdc, 400, 160, 440, 200);
SelectBrush(hdc,hSurBrush);
if(timp <=300) SelectBrush(hdc,hRedBrush);
Ellipse(hdc, 400, 200, 440, 240);
SelectBrush(hdc,hSurBrush);

// Bottom-right
Rectangle(hdc, 560, 360, 600, 480);
if(timp <=300) SelectBrush(hdc,hRedBrush);
Ellipse(hdc, 560, 360, 600, 400);
SelectBrush(hdc,hSurBrush);
if(timp >= 200 && timp <= 300) SelectBrush(hdc,hYellowBrush);
Ellipse(hdc, 560, 400, 600, 440);
SelectBrush(hdc,hSurBrush);
if(timp >= 300 && timp <= 600 || (timp >= 650 && timp <= 690) || (timp >= 730 && timp <= 770) || (timp >= 810 &&
timp <= 850)) SelectBrush(hdc,hGreenBrush);
Ellipse(hdc, 560, 440, 600, 480);
SelectBrush(hdc,hSurBrush);

// top-right
Rectangle(hdc, 560, 200, 680, 240);
if(timp >= 300 && timp <= 850) SelectBrush(hdc,hRedBrush);
Ellipse(hdc, 560, 200, 600, 240);
SelectBrush(hdc,hSurBrush);
if(timp >= 750 && timp <= 850) SelectBrush(hdc,hYellowBrush);
Ellipse(hdc, 600, 200, 640, 240);
SelectBrush(hdc,hSurBrush);
if(timp <= 50 || (timp >= 100 && timp <= 140) || (timp >= 180 && timp <= 220) || (timp >= 260 && timp <= 300))
SelectBrush(hdc,hGreenBrush);
Ellipse(hdc, 640, 200, 680, 240);
SelectBrush(hdc,hSurBrush);

// bottom-left
Rectangle(hdc, 320, 360, 440, 400);
if(timp <= 50 || (timp >= 100 && timp <= 140) || (timp >= 180 && timp <= 220) || (timp >= 260 && timp <= 300))
SelectBrush(hdc,hGreenBrush);
Ellipse(hdc, 320, 360, 360, 400);
SelectBrush(hdc,hSurBrush);
if(timp >= 750 && timp <= 850) SelectBrush(hdc,hYellowBrush);
Ellipse(hdc, 360, 360, 400, 400);
SelectBrush(hdc,hSurBrush);
if(timp >= 300 && timp <= 850) SelectBrush(hdc,hRedBrush);
Ellipse(hdc, 400, 360, 440, 400);
SelectBrush(hdc,hSurBrush);

if(timp > 850) timp = -300;
if(dtime > 2300) dtime = 0;
}

void Intersectie(HDC hdc)
{
    SelectBrush(hdc,hRandBrush);

    // stinga
    MoveToEx (hdc, 0, 250, NULL) ;
    LineTo(hdc, 450, 250);
    MoveToEx (hdc, 0, 350, NULL) ;
    LineTo(hdc, 450, 350);

```

```
// dreapta
MoveToEx (hdc, 550, 250, NULL) ;
LineTo(hdc, 1000, 250);
MoveToEx (hdc, 550, 350, NULL) ;
LineTo(hdc, 1000, 350);
```

```
// top
MoveToEx (hdc, 450, 0, NULL) ;
LineTo(hdc, 450, 250);
MoveToEx (hdc, 550, 0, NULL) ;
LineTo(hdc, 550, 250);
```

```
// bottom
MoveToEx (hdc, 450, 350, NULL) ;
LineTo(hdc, 450, 700);
MoveToEx (hdc, 550, 350, NULL) ;
LineTo(hdc, 550, 700);
```

```
int bottom=700, right=1000, i_bottom=350, i_top=250, i_left=450, i_right=550, margin=0, depl = 0, margin1=0, depl1 =
```

0;

```
SelectBrush(hdc,hGreenBrush);
```

```
// Desenam iarba (top-left)
for(int i=20; i < 700; i+=20)
{
    if(i>i_top) { margin = i_top; depl += 20; } else { margin = i; depl = 0; }
    MoveToEx (hdc, depl, margin, NULL) ;
    if(i>i_left) { margin1 = i_left; depl1 += 20; } else { margin1 = i; depl1 = 0; }
    LineTo(hdc, margin1, depl1);
}
```

```
// Desenam iarba (top-right)
for(int i=0; i < 700; i+=20)
{
    if(i>i_top) { margin = i_top; depl += 20; } else { margin = i; depl = i_right; }
    MoveToEx (hdc, depl, margin, NULL) ;
    LineTo(hdc, i+i_right, 0);
}
```

```
// Desenam iarba (bottom-left)
for(int i=20; i < 800; i+=20)
{
    MoveToEx (hdc, 0, i+i_bottom, NULL) ;
    if(i>i_left) { margin1 = i_left; depl1 += 20; } else { margin1 = i; depl1 = i_bottom; }
    LineTo(hdc, margin1, depl1);
}
```

```
// Desenam iarba (bottom-right)
for(int i=0; i < 800; i+=20)
{
    MoveToEx (hdc, i_right, i+i_bottom, NULL) ;
    LineTo(hdc, i+i_right, i_bottom);
}
```

```
}
```

Rezultatele execuției programului:

Figura 1. Automobilul în apropiere de intersecție

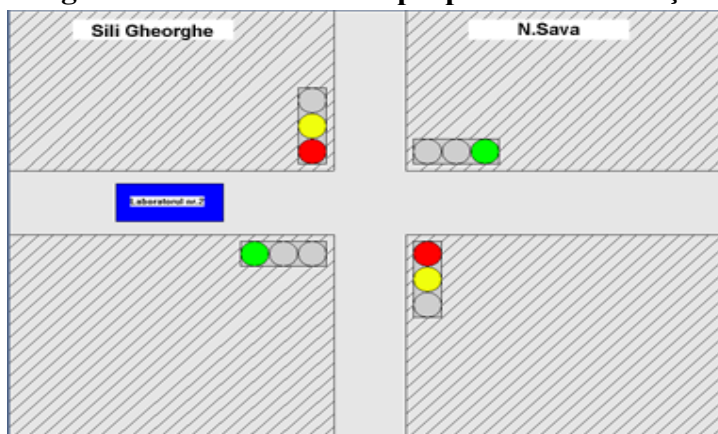


Figura 2. Automobilul este în stare de așteptare a culorii VERDE a semaforului

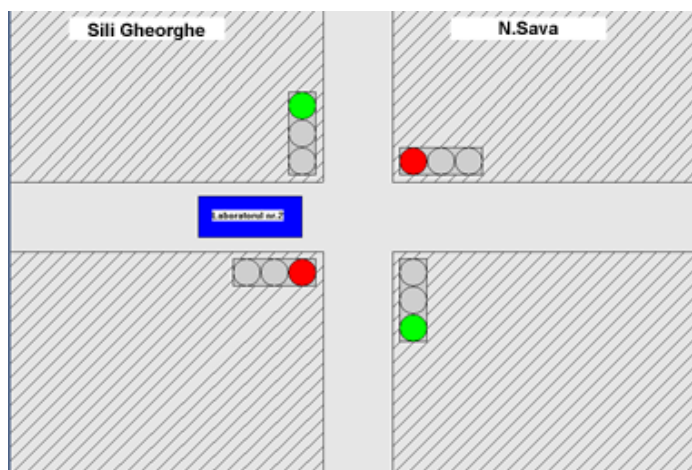
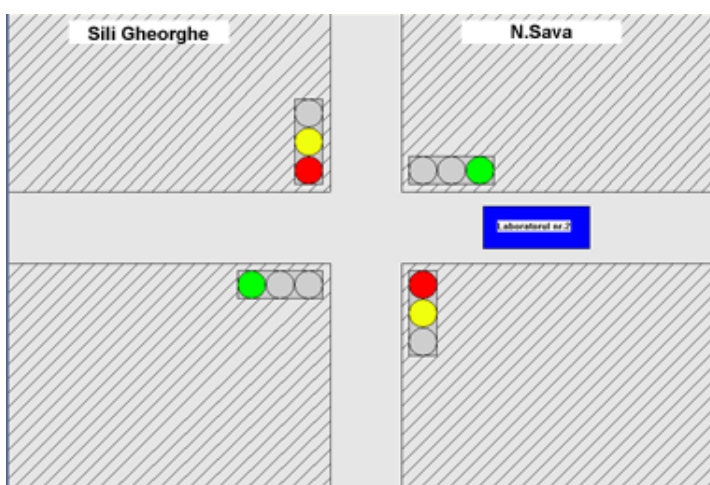


Figura 3. Automobilul traversează intersecția



Răspuns la întrebări:

1. Descrieți principalele primitive ale interfeței grafice.

Unele dintre primitivele interfeței grafice sunt: Linii, curbe, Suprafețe pline, Imagini bitmap, Text. Liniile reprezintă baza oricărui sistem de desenare vectorial. GDI permite folosirea liniilor drepte, a dreptunghiurilor, a elipselor, a arcelor ș.a. . În zonele de client este posibil de folosit diferite figure atașându-le prin coordonate având ca axa de coordonate zona de client unde punctul din stînga sus este punctual 0, 0.

2. Enumerați metodele de obținere a variabilei handle a dispozitivului de context.

Sistemul de operare Windows vă pune la dispoziție mai multe metode pentru obținerea variabilei handle a contextului de dispozitiv. Dacă obținerea variabilei handle a contextului de dispozitiv. Dacă obțineți o variabilă handle a contextului de dispozitiv în timpul prelucrării unui mesaj, ar trebui să ștergeți această variabilă înainte de ieșirea din procedura de fereastră. După ce este ștearsă, variabila handle nu mai poate fi folosită (nu mai este validă=.

Cea mai cunoscută metodă de obținere și de ștergere a variabilei handle a contextului de dispozitiv implică folosirea funcțiilor BeginPaint și EndPaint în timpul prelucrării mesajului WM_PAINT.

Concluzii: În urma efectuării laboratorului nr.2 am avut de lucru cu interfața GDI. Am aplicat în laborator conform cerințelor animație, adică obiecte geometrice în mișcare.