МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Кафедра «Информационные системы»

Дисциплина «Основы алгоритмизации и программирования»

Лабораторная работа №7

Arc и др.

Polygon.

Многомодульный проект

Выполнил:

студент гр. ПИбд-12

Дозорова Алена

Ульяновск, 2021 г

Задание лабораторной работы

1. Нарисовать олимпийскую эмблему – настолько близко к оригиналу, насколько это возможно.
   * Отрисовку эмблемы олимпиады сделать в отдельной функции.
   * Создать модуль Lab7.cpp / Lab7.h.
     + Поместить в него функцию

void OlympicEmblem(HDC hdc);

* + - Создать в модуле Lab7 функцию:

void Lab7(HDC hdc, int numPicture) {

if (numPicture == 1) OlympicEmblem(hdc);

}

* + - Заставить всё работать!

1. Нарисовать будильник. Отрисовку сделать в отдельной функции.
   * Добавить в функцию Lab7() вызов функции отрисовки будильника:

void Lab7(HDC hdc, int numPicture) {

if (numPicture == 1) OlympicEmblem(hdc);

if (numPicture == 2) AlarmClock(hdc);

}

* + Заставить всё работать!

1. Нарисовать облако. Отрисовку сделать в отдельной функции
2. Сделать 3 типа звезд – с разной заливкой. Использовать Polygon.
   * Назвать функции Star1(), Star2() и Star3().
   * Используя их создать сложный красивый образ Image1
   * Образ реализовать в функции вида

void Image1(HDC hdc);

* + Добавить вызов этой функции в функцию Lab7:

void Lab7(HDC hdc, int numPicture) {

if (numPicture == 1) OlympicEmblem(hdc);

if (numPicture == 2) AlarmClock(hdc);

if (numPicture == 3) Image1(hdc);

}

* + Заставить всё работать!

1. Используя фукнции Star1(), Star2() и Star3() создать сложный красивый образ Image2
   * Образ реализовать в функции вида
2. void Image2(HDC hdc);
   * Добавить вызов этой функции в функцию Lab7:

void Lab7(HDC hdc, int numPicture) {

if (numPicture == 1) OlympicEmblem(hdc);

if (numPicture == 2) AlarmClock(hdc);

if (numPicture == 3) Image1(hdc);

if (numPicture == 4) Image2(hdc);

}

* + Заставить всё работать!

1. Переделать лабораторную работу №6 разбив её на модули. Все рисунки лабораторной работы №1 вынести в модуль Lab1. Все рисунки лабораторной работы №2 вынести в модуль Lab2. Аналогично поступить с остальными рисунками.
2. Создать ЗАВЕРШАЮЩИЙ рисунок в котором использовать по максимуму все возможности, с которыми вы успели ознакомиться за время изучения Си и WinAPI. Рисунок может содержать части созданные рекурсивными функциями, может содержать циклы, развилки. Можно использовать ваши ранее созданные рисунки.

Код программы

Laba7.cpp

|  |  |  |
| --- | --- | --- |
| case WM\_KEYDOWN: | | |
|  | | | switch (wParam) |
|  | | | { |
|  | | | case VK\_LEFT: |
|  | | | num -= 1; |
|  | | |  |
|  | | | InvalidateRect(hWnd, NULL, TRUE); |
|  | | | break; |
|  | | | case VK\_RIGHT: |
|  | | | num += 1; |
|  | | | InvalidateRect(hWnd, NULL, TRUE); |
|  | | | break; |
|  | | | } |
|  | | |  |
|  | | | break; |
| case WM\_PAINT: |
|  | { | |
|  | PAINTSTRUCT ps; | |
|  | HDC hdc = BeginPaint(hWnd, &ps); | |
|  | // TODO: Добавьте сюда любой код прорисовки, использующий HDC... | |
|  | if ((num > 0) && (num < 7)) | |
|  | { | |
|  | Lab7(hdc, num); | |
|  | } | |
|  | else | |
|  | { | |
|  | TCHAR string1[] = \_T("Вне границ"); | |
|  | TextOut(hdc, 10, 10, string1, \_tcslen(string1)); | |
|  | } | |
|  | EndPaint(hWnd, &ps); | |
|  | } | |
|  | break; | |

Lab7.h

|  |
| --- |
| #pragma once |
|  | #define \_CRT\_SECURE\_NO\_WARNINGS |
|  | #include "stdio.h" |
|  | #define WIN32\_LEAN\_AND\_MEAN |
|  | #include <stdlib.h> |
|  |  |
|  | #include "framework.h" |
|  |  |
|  | void OlympicEmblem(HDC hdc); |
|  | void AlarmClock(HDC hdc); |
|  | void Cloud(HDC hdc); |
|  | void Star1(HDC hdc, int cx, int cy, int size); |
|  | void Star2(HDC hdc, int cx, int cy, int size); |
|  | void Star3(HDC hdc, int cx, int cy, int size); |
|  | void Image1(HDC hdc); |
|  | void Image2(HDC hdc); |
|  | void my\_picture(HDC hdc); |
|  | void ship(int x, int y, HDC hdc); |
|  | void Lab7(HDC hdc, int numPicture); |

Lab7.cpp

|  |
| --- |
| #pragma once |
|  | #define \_CRT\_SECURE\_NO\_WARNINGS |
|  | #include "stdio.h" |
|  | #define WIN32\_LEAN\_AND\_MEAN |
|  | #include <stdlib.h> |
|  | #include "framework.h" |
|  |  |
|  | // Ôàéëû çàãîëîâêîâ ñðåäû âûïîëíåíèÿ C |
|  | #include <stdlib.h> |
|  | #include <malloc.h> |
|  | #include <memory.h> |
|  | #include <tchar.h> |
|  | #include <cmath> |
|  |  |
|  |  |
|  |  |
|  | void OlympicEmblem(HDC hdc) |
|  | { |
|  | // Êèñòü - ïðîçðà÷íàÿ |
|  | SelectObject(hdc, GetStockObject(NULL\_BRUSH)); |
|  |  |
|  | // âåðõíèé ðÿä êîëåö |
|  | HPEN hPen = CreatePen(PS\_SOLID, 5, RGB(0, 0, 255)); |
|  | SelectObject(hdc, hPen); |
|  | Ellipse(hdc, 20, 40, 60, 80); |
|  |  |
|  | hPen = CreatePen(PS\_SOLID, 5, RGB(0, 0, 0)); |
|  | SelectObject(hdc, hPen); |
|  | Ellipse(hdc, 70, 40, 110, 80); |
|  |  |
|  | hPen = CreatePen(PS\_SOLID, 5, RGB(255, 0, 0)); |
|  | SelectObject(hdc, hPen); |
|  | Ellipse(hdc, 120, 40, 160, 80); |
|  |  |
|  | // íèæíèé ðÿä êîëåö |
|  | hPen = CreatePen(PS\_SOLID, 5, RGB(255, 255, 0)); |
|  | SelectObject(hdc, hPen); |
|  | Ellipse(hdc, 45, 60, 85, 100); |
|  |  |
|  | hPen = CreatePen(PS\_SOLID, 5, RGB(0, 255, 0)); |
|  | SelectObject(hdc, hPen); |
|  | Ellipse(hdc, 95, 60, 135, 100); |
|  |  |
|  | } |
|  |  |
|  | void AlarmClock(HDC hdc) |
|  | { |
|  | HPEN hPen = CreatePen(PS\_SOLID, 5, RGB(0, 255, 75)); |
|  | SelectObject(hdc, hPen); |
|  | Ellipse(hdc, 180, 40, 260, 120); |
|  | Chord(hdc, 180, 30, 215, 60, 215, 25, 175, 55); |
|  | Chord(hdc, 225, 30, 260, 60, 265, 55, 225, 25); |
|  | } |
|  |  |
|  | void Cloud(HDC hdc) |
|  | { |
|  | HBRUSH hBrush = CreateSolidBrush(RGB(226, 226, 241)); |
|  | SelectObject(hdc, hBrush); |
|  | HPEN hPen = CreatePen(PS\_ALTERNATE, 2, RGB(226, 226, 241)); |
|  | SelectObject(hdc, hPen); |
|  | Ellipse(hdc, 200, 150, 300, 200); |
|  | Ellipse(hdc, 270, 155, 350, 210); |
|  | Ellipse(hdc, 350, 60, 480, 200); |
|  | Ellipse(hdc, 210, 75, 512, 190); |
|  | } |
|  | void Star1(HDC hdc, int cx, int cy, int size) |
|  | { |
|  | POINT p[9] = |
|  | { |
|  | cx, cy - size, |
|  | cx + size / 4, cy - size / 4, |
|  | cx + size, cy, |
|  | cx + size / 4, cy + size / 4, |
|  | cx, cy + size, |
|  | cx - size / 4, cy + size / 4, |
|  | cx - size, cy, |
|  | cx - size/4, cy-size/4, |
|  | cx, cy - size |
|  | }; |
|  |  |
|  | HBRUSH hBrush = CreateSolidBrush(RGB(255, 255, 0)); |
|  | SelectObject(hdc, hBrush); |
|  |  |
|  | Polygon(hdc, p, 9); |
|  |  |
|  | DeleteObject(hBrush); |
|  | } |
|  |  |
|  | void Star2(HDC hdc, int cx, int cy, int size) |
|  | { |
|  | POINT p[9] = |
|  | { |
|  | cx, cy - size, |
|  | cx + size / 4, cy - size / 4, |
|  | cx + size, cy, |
|  | cx + size / 4, cy + size / 4, |
|  | cx, cy + size, |
|  | cx - size / 4, cy + size / 4, |
|  | cx - size, cy, |
|  | cx - size / 4, cy - size / 4, |
|  | cx, cy - size |
|  | }; |
|  |  |
|  | HBRUSH hBrush = CreateSolidBrush(RGB(245, 139, 10)); |
|  | SelectObject(hdc, hBrush); |
|  |  |
|  | Polygon(hdc, p, 9); |
|  |  |
|  | DeleteObject(hBrush); |
|  | } |
|  | void Star3(HDC hdc, int cx, int cy, int size) |
|  | { |
|  | POINT p[9] = |
|  | { |
|  | cx, cy - size, |
|  | cx + size / 4, cy - size / 4, |
|  | cx + size, cy, |
|  | cx + size / 4, cy + size / 4, |
|  | cx, cy + size, |
|  | cx - size / 4, cy + size / 4, |
|  | cx - size, cy, |
|  | cx - size / 4, cy - size / 4, |
|  | cx, cy - size |
|  | }; |
|  |  |
|  | HBRUSH hBrush = CreateSolidBrush(RGB(255, 10, 10)); |
|  | SelectObject(hdc, hBrush); |
|  |  |
|  | Polygon(hdc, p, 9); |
|  |  |
|  | DeleteObject(hBrush); |
|  | } |
|  | void Image1(HDC hdc) |
|  | { |
|  | int size = 15; |
|  | int i = 0; |
|  | int x = 500; |
|  | int y = 20; |
|  | while (i < 5) |
|  | { |
|  | int j = 0; |
|  | while (j <= i) |
|  | { |
|  | Star1(hdc, x - j \* 20, y, size); |
|  | Star2(hdc, x + j \* 20, y, size); |
|  | j += 1; |
|  | } |
|  | y += 20; |
|  | i++; |
|  | } |
|  |  |
|  | y = 400; |
|  | i = 0; |
|  | while (i < 5) |
|  | { |
|  | int j = 0; |
|  | while (j <= i) |
|  | { |
|  | Star3(hdc, x - j \* 20, y, size); |
|  | Star2(hdc, x + j \* 20, y, size); |
|  | j += 1; |
|  | } |
|  | y -= 20; |
|  | i++; |
|  | } |
|  | y = 200; |
|  | x = 300; |
|  | i = 0; |
|  | while (i < 5) |
|  | { |
|  | int j = 0; |
|  | while (j <= i) |
|  | { |
|  | Star2(hdc, x , y - j \* 20, size); |
|  | Star3(hdc, x , y + j \* 20, size); |
|  | j += 1; |
|  | } |
|  | x += 20; |
|  | i++; |
|  | } |
|  |  |
|  | x = 700; |
|  | i = 0; |
|  | while (i < 5) |
|  | { |
|  | int j = 0; |
|  | while (j <= i) |
|  | { |
|  | Star1(hdc, x, y - j \* 20, size); |
|  | Star3(hdc, x, y + j \* 20, size); |
|  | j += 1; |
|  | } |
|  | x -= 20; |
|  | i++; |
|  | } |
|  |  |
|  | x = 455; |
|  | y = 170; |
|  | i = 0; |
|  | while (i < 5) |
|  | { |
|  | int j = 0; |
|  | while (j < 5) |
|  | { |
|  | if (i % 3 == 0) |
|  | { |
|  | Star1(hdc, x+j\*20, y+i\*20, size); |
|  | } |
|  | if (i % 3 == 1) |
|  | { |
|  | Star2(hdc, x + j \* 20, y + i \* 20, size); |
|  | } |
|  | if (i % 3 == 2) |
|  | { |
|  | Star3(hdc, x + j \* 20, y + i \* 20, size); |
|  | } |
|  | j++; |
|  | } |
|  | i++; |
|  | } |
|  | } |
|  |  |
|  | void Image2(HDC hdc) |
|  | { |
|  | HWND hwnd = GetConsoleWindow(); |
|  | for (int j = 0; j < 3; j++) |
|  | { |
|  | int x = 0; |
|  | for (float i = 0; i < 3.14 \* 100; i += 1) |
|  | { |
|  | Star1(hdc, x, 75 + 25 \* cos(i)+50\*j, 15); |
|  | x += 20; |
|  | } |
|  | x = 0; |
|  | for (float i = 0; i < 3.14 \* 100; i += 1) |
|  | { |
|  | Star2(hdc, x, 125 + 25 \* cos(i)+50 \* j, 15); |
|  | x += 20; |
|  | } |
|  | x = 0; |
|  | for (float i = 0; i < 3.14 \* 100; i += 1) |
|  | { |
|  | Star3(hdc, x, 160 + 25 \* cos(i)+ 50 \* j, 15); |
|  | x += 20; |
|  | } |
|  | } |
|  |  |
|  | ReleaseDC(hwnd, hdc); |
|  | } |
|  | void rec\_ellipce( HDC hdc,int cx, int cy, int size) |
|  | { |
|  | HBRUSH hBrush = CreateSolidBrush(RGB(100, 0, 255)); |
|  | SelectObject(hdc, hBrush); |
|  | Ellipse(hdc, cx - 3\*size, cy - size, cx+2\*size, cy); |
|  | if (size < 5) |
|  | { |
|  | return; |
|  | } |
|  | rec\_ellipce(hdc, cx, cy, size-10); |
|  | } |
|  |  |
|  | void ship(int x, int y, HDC hdc) |
|  | { |
|  | HBRUSH hBrush = CreateSolidBrush(RGB(255, 255, 255)); |
|  | SelectObject(hdc, hBrush); |
|  | POINT p[18] ={ |
|  | x - 40, y - 30, |
|  | x - 40, y - 70, |
|  | x - 60, y - 70, |
|  | x - 60, y - 30, |
|  | x - 70, y - 30, |
|  | x - 70, y - 10, |
|  | x - 90, y - 10, |
|  | x - 100, y - 20, |
|  | x - 150, y - 20, |
|  | x - 100, y+ 30, |
|  | x + 70, y + 30, |
|  | x + 90, y - 10, |
|  | x + 90, y - 20, |
|  | x + 30, y - 20, |
|  | x + 10, y - 10, |
|  | x + 20, y - 10, |
|  | x - 30, y - 30, |
|  | x - 40, y - 40 |
|  | }; |
|  |  |
|  | Polygon(hdc, p, 18); |
|  | } |
|  | void my\_picture(HDC hdc) |
|  | { |
|  | HBRUSH hBrush = CreateSolidBrush(RGB(0, 0, 50)); |
|  | SelectObject(hdc, hBrush); |
|  | Rectangle(hdc, 0, 0, 1000, 1000); |
|  | HWND hwnd = GetConsoleWindow(); |
|  | for (int j = 0; j < 6; j++) |
|  | { |
|  | int x = 0; |
|  | for (float i = 0; i < 3.14 \* 100; i += 1) |
|  | { |
|  | Star1(hdc, x, 75 + 25 \* cos(i) + 100 \* j, 5); |
|  | x += 20; |
|  | } |
|  |  |
|  | } |
|  | DeleteObject(hBrush); |
|  |  |
|  |  |
|  | hBrush = CreateSolidBrush(RGB(100, 0, 255)); |
|  | SelectObject(hdc, hBrush); |
|  |  |
|  | Rectangle(hdc, 0, 250, 1000, 1000); |
|  | rec\_ellipce(hdc, 260, 345, 30); |
|  |  |
|  | rec\_ellipce(hdc, 0, 280, 10); |
|  |  |
|  | rec\_ellipce(hdc, 1000, 400, 20); |
|  |  |
|  | rec\_ellipce(hdc, 600, 340, 18); |
|  |  |
|  | rec\_ellipce(hdc, 325, 280, 35); |
|  |  |
|  | rec\_ellipce(hdc, 400, 310, 18); |
|  | ship(600, 300, hdc); |
|  | DeleteObject(hBrush); |
|  | ReleaseDC(hwnd, hdc); |
|  | } |
|  | void Lab7(HDC hdc, int numPicture) { |
|  | if (numPicture == 1) OlympicEmblem(hdc); |
|  | else if (numPicture == 2) AlarmClock(hdc); |
|  | else if (numPicture == 3) Cloud(hdc); |
|  | else if (numPicture == 4) Image1(hdc); |
|  | else if (numPicture == 5) Image2(hdc); |
|  | else if (numPicture == 6) my\_picture(hdc); |
|  | } |

Заключение

Я освоила и применила на практике создание многомодульных проектов, создала итоговый для раздела графики рисунок, используя полученные знания.