

## 实验二 基于 SDK 编程（一）

### 一、实验目的

- 1 掌握 Windows 图形设备接口的基本概念
- 2 掌握获取设备环境的方法以及映射模式的设置
- 3 掌握绘图工具的使用和绘图函数的应用

### 二、实验要求

- 1 预习本次实验的实验要求和任务
- 2 熟悉基于 SDK 的绘图技术和方法，并独立完成本次实验

### 三、实验环境

- 1 软件：VC++ 6.0 或 Visual Studio

### 四、实验内容

- 1 创建一个图形绘制的 VC 工程，创建 cpp 文件
- 2 编写 WinMain 函数进行本工程窗口定义与设计等任务，并完成窗口函数中的基本消息处理过程
- 3 获取设备环境，设置映射模式，创建画笔，编写代码绘制风车叶片，并让风车旋转

### 五、实验过程及结果描述

- 1 依照实验一步骤创建一个基于 SDK 的工程及相应其它文件
- 2 在本工程的 cpp 文件中输入以下代码：

```
#include<windows.h>
#include<stdio.h>
#include<math.h>
#define pi 3.1415926
LRESULT CALLBACK WinProc(HWND hwnd,UINT uMsg,WPARAM wParam,LPARAM lParam );
int nNum=0,nMaxNum=20; //nMaxNum 绘图次数，nNum 当前位置
```

Winmain 函数代码如下：

```
int WINAPI WinMain(
    HINSTANCE hInstance,
    HINSTANCE hPrevInstace,
    LPSTR lpCmdLine,
```

```

        int nCmdShow
    )
{
    char lpszClassName[]="窗口";
    char lpszTitle[]="实验二";
    //设计一个窗口类
    WNDCLASS wndcls;
    wndcls.cbWndExtra=0;
    wndcls.cbClsExtra=0;
    wndcls.hbrBackground=(HBRUSH)GetStockObject(WHITE_BRUSH);
    wndcls.hCursor=LoadCursor(NULL,IDC_ARROW);
    wndcls.hIcon=LoadIcon(NULL,IDI_APPLICATION);
    wndcls.hInstance=hInstance;
    wndcls.lpfWndProc=WinProc;
    wndcls.lpszClassName=lpszClassName;
    wndcls.lpszMenuName=NULL;
    wndcls.style=CS_HREDRAW | CS_VREDRAW;
    RegisterClass(& wndcls);
    //创建窗口，定义一个变量用来保存成功创建的窗口后返回的句柄
    HWND hwnd;
    hwnd=CreateWindow(lpszClassName, lpszTitle, WS_OVERLAPPEDWINDOW,
CW_USEDEFAULT,0,600,450,NULL,NULL,hInstance,NULL);
    //显示及刷新窗口
    ShowWindow(hwnd,SW_SHOWNORMAL);
    UpdateWindow(hwnd);
    //定义消息结构体，开始消息循环
    MSG msg;
    while(GetMessage(&msg,NULL,0,0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}

```

窗口过程函数如下：

```

LRESULT CALLBACK WinProc(
    HWND hwnd,
    UINT uMsg,
    WPARAM wParam,
    LPARAM lParam
)
{
    HBRUSH hBrush;

```

```

HPEN hPen;
char lpszTitle[]="My_windows";
int x,y;    //x,y 分别作为圆心的坐标
double fAngle;
switch(uMsg)
{
case WM_CREATE:
    SetTimer(hwnd,9999,100,NULL);
    break;
case WM_PAINT:
    HDC hDC;
    PAINTSTRUCT ps;
    hDC=BeginPaint(hwnd,&ps);
    SetMapMode(hDC,MM_ANISOTROPIC);
    SetWindowExtEx(hDC,400,300,NULL);    //设置窗口逻辑单位，所谓
    逻辑单位，即在比例一致情况下选择足够大的区域
    SetViewportExtEx(hDC,600,450,NULL);
    SetViewportOrgEx(hDC,300,200,NULL);
    hPen=(HPEN)GetStockObject(RGB(10,100,20));    //画外圆
    SelectObject(hDC,hPen);
    Ellipse(hDC,-100,-100,100,100);
    hBrush=CreateSolidBrush(RGB(255,0,0));
    SelectObject(hDC,hBrush);
    fAngle=2*pi/nMaxNum*nNum;//设置初始角度,1/20*pi、2/20*pi、3/20*pi
    x=(int)(50*cos(fAngle));    //初始圆心位置坐标 R(x,y)
    y=(int)(50*sin(fAngle));
    Pie(hDC,x-50,y-50,x+50,y+50,
        (int)(x+50*cos(fAngle)),(int)(y+50*sin(fAngle)),
        (int)(x+50*cos(fAngle+pi)),(int)(y+50*sin(fAngle+pi)) );
    hBrush=CreateSolidBrush(RGB(0,255,0));
    SelectObject(hDC,hBrush);
    fAngle=2*pi/nMaxNum*nNum; //设置初始角度，1/20*pi、2/20*pi、
    3/20*pi
    x=(int)(50*cos(fAngle+2*pi/3));    //初始圆心位置坐标 R(x,y)
    y=(int)(50*sin(fAngle+2*pi/3));
    Pie(hDC,x-50,y-50,x+50,y+50,    //易知
        (int)(x+50*cos(fAngle+2*pi/3)),
        (int)(y+50*sin(fAngle+2*pi/3)),
        (int)(x+50*cos(fAngle+pi+2*pi/3)),
        (int)(y+50*sin(fAngle+pi+2*pi/3)));
    hBrush=CreateSolidBrush(RGB(0,0,255));
    SelectObject(hDC,hBrush);
    fAngle=2*pi/nMaxNum*nNum;
    x=(int)(50*cos(fAngle+4*pi/3));

```

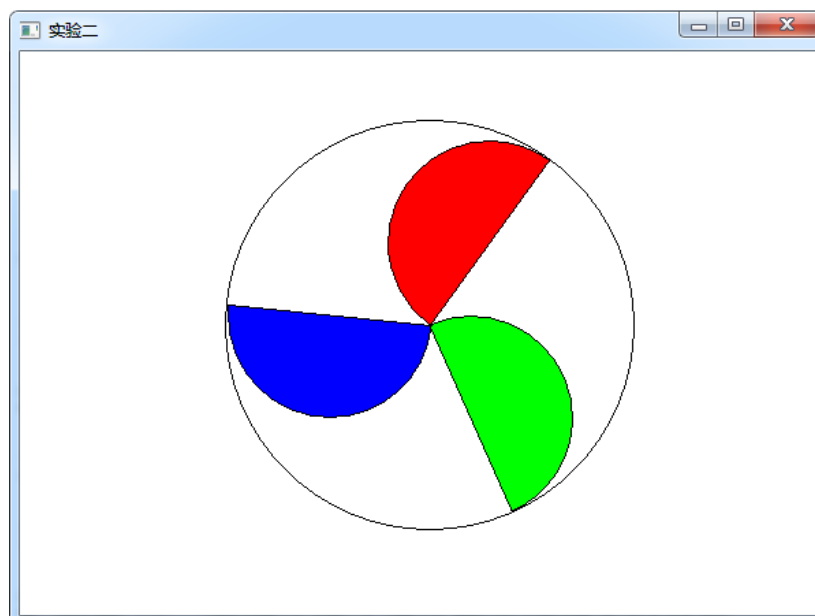
```

y=(int)(50*sin(fAngle+4*pi/3));
Pie(hDC,x-50,y-50,x+50,y+50,          //同理
    (int)(x+50*cos(fAngle+4*pi/3)),
    (int)(y+50*sin(fAngle+4*pi/3)),
    (int)(x+50*cos(fAngle+pi+4*pi/3)),
    (int)(y+50*sin(fAngle+pi+4*pi/3)) );
EndPoint(hwnd,&ps);
DeleteObject(hBrush);
DeleteObject(hPen);
return 0;
case WM_TIMER:
    if (wParam==9999)
    {
        nNum++;
        if (nNum > nMaxNum)
            nNum = 0;
        InvalidateRect(hwnd,NULL,1);    //刷新客户区
    }
    break;
case WM_DESTROY:
    PostQuitMessage(0);
    return 0;
default:
    return DefWindowProc(hwnd,uMsg,wParam,lParam);
}

return 0;
}

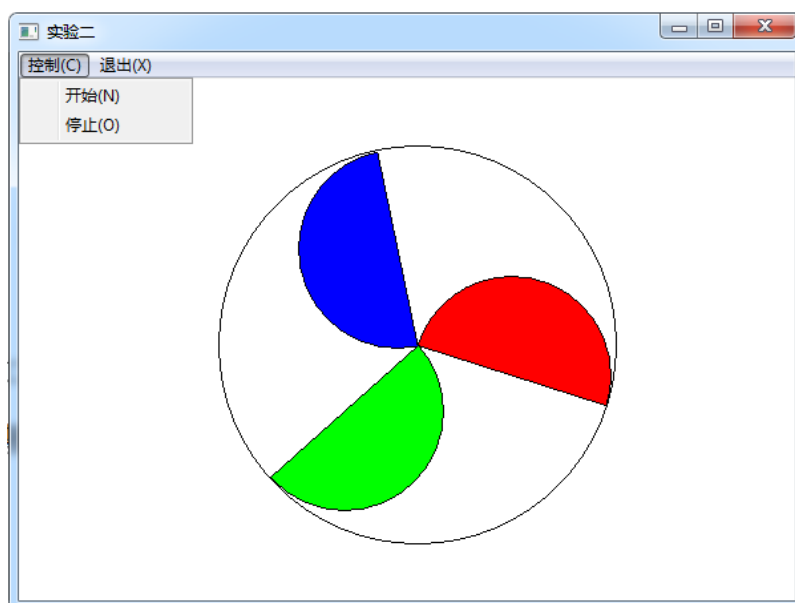
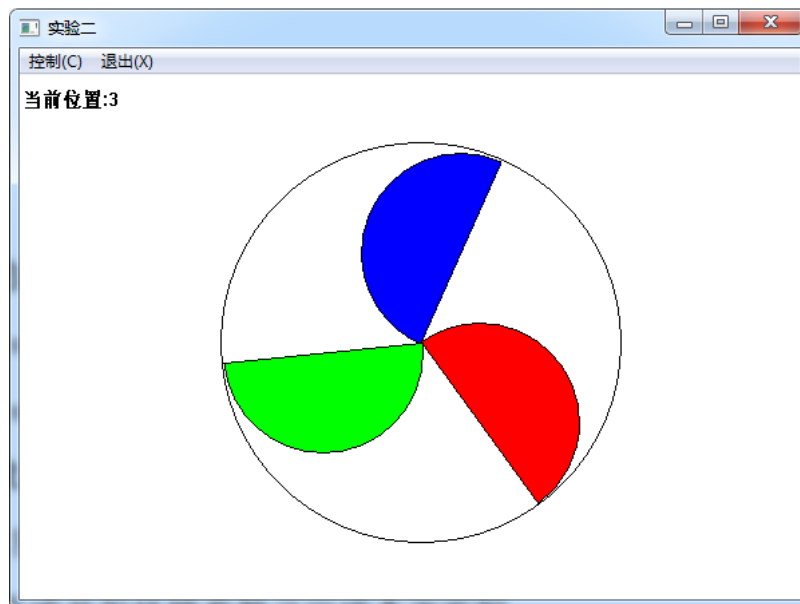
```

最终实验结果：转动风车截图如下：



3 在此基础上，添加如下功能：

- (1) 在窗口左上角显示当前位置，即 `nNum` 的值；
- (2) 添加菜单资源（主菜单包括控制和退出，其中控制的子菜单是开始和停止），按下“开始”时风车开始转动，按下“停止”时风车停止转动。



## 实验三 基于 SDK 编程（二）

### 一、实验目的

- 1 掌握设备驱动程序翻译的常用虚拟码
- 2 掌握两类键盘消息

### 二、实验要求

- 1 预习本次实验的实验要求和任务
- 2 熟悉基于 SDK 的键盘操作技术和方法，并独立完成本次实验

### 三、实验环境

- 1 软件：VC++ 6.0 或 Visual Studio

### 四、实验内容

- 1 创建一个键盘操作的 VC 工程，创建 cpp 文件
- 2 编写 WinMain 函数进行本工程窗口定义与设计等任务，并完成窗口函数中的基本消息处理过程
- 3 按照要求对"ctrl"键、"shift"键、下左或右箭头、Home 及 End 键、PageUp 及 PageDown 键编写相应的消息响应代码，以实现单击这些按钮时，分别可以完成不同图形的绘制以及移动等操作。

### 五、实验过程及结果描述

- 1 依照实验一步骤创建一个基于 SDK 的工程及相应其它文件
- 2 在本工程的 cpp 文件中输入以下代码：

```
#include<windows.h>
int nMode;
RECT rect;
LRESULT CALLBACK WndProc(HWND,UINT,WPARAM,LPARAM);
```

Winmain 函数代码如下：

```
int WINAPI WinMain
(
    HINSTANCE hinstaance,
    HINSTANCE pre,
    LPSTR lpszCmdLine,
    int nCmdShow
)
```

```

{
    HWND hwnd;
    MSG msg;
    WNDCLASS wndclass;
    char lpszClassName[]=" ";
    char Title[]="实验三";
    wndclass.cbClsExtra=0;
    wndclass.cbWndExtra=0;
    wndclass.hbrBackground=CreateSolidBrush(RGB(255,255,255));
    wndclass.hCursor=LoadCursor(NULL,IDC_ARROW);
    wndclass.hIcon=LoadIcon(NULL,IDI_QUESTION);
    wndclass.hInstance=hinstaance;
    wndclass.lpfnWndProc=WndProc;
    wndclass.lpszClassName=lpszClassName;
    wndclass.lpszMenuName=NULL;
    wndclass.style=0;
    RegisterClass(&wndclass);
    hwnd=CreateWindow(
        lpszClassName,
        Title,WS_OVERLAPPEDWINDOW|WS_HSCROLL|WS_VSCROLL,
        100,100,
        500,500,
        NULL,NULL,hinstaance,NULL
    );
    ShowWindow(hwnd,nCmdShow);
    UpdateWindow(hwnd);
    while(GetMessage(&msg,NULL,0,0))
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    return msg.wParam;
}

```

窗口函数代码如下：

```

LRESULT CALLBACK WndProc(HWND hwnd,UINT message,WPARAM wParam,LPARAM
lParam)
{
    HDC hdc;
    HPEN hp;
    HBRUSH hb;
    static int Circle;
    PAINTSTRUCT ps;
    switch(message)

```

```

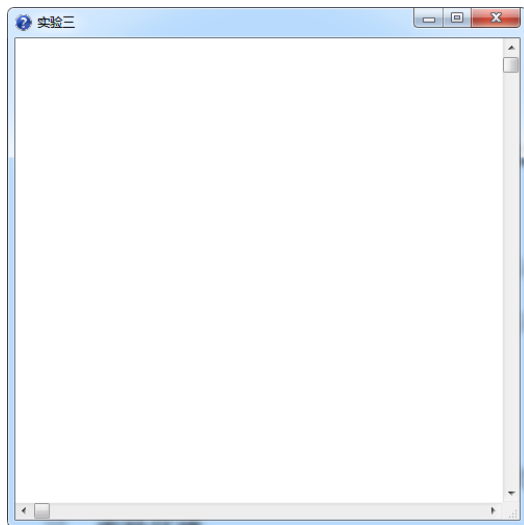
{
case WM_KEYDOWN:
    if(wParam==VK_CONTROL)
    {
        Circle=1;
        rect.left=0;
        rect.right=50;
        rect.top=0;
        rect.bottom=50;
        InvalidateRect(hwnd,NULL,1);
    }
    else if(wParam==VK_HOME)
    {
        rect.left=rect.left-10;
        rect.right=rect.right-10;
        if(Circle==1)
        {
            InvalidateRect(hwnd,NULL,1);
        }
    }
    else if(wParam==VK_END)
    {
        rect.left=rect.left+10;
        rect.right=rect.right+10;
        if(Circle==1)
        {
            InvalidateRect(hwnd,NULL,1);
        }
    }
    break;
case WM_PAINT:
    hdc=BeginPaint(hwnd,&ps);
    SetMapMode(hdc,MM_ISOTROPIC);
    SetWindowExtEx(hdc,1,1,NULL);
    SetViewportExtEx(hdc,1,1,NULL);
    hp=CreatePen(PS_SOLID,0,RGB(0,0,255));
    hb=CreateSolidBrush(RGB(0,255,0));
    SelectObject(hdc,hb);
    SelectObject(hdc,hp);
    if(Circle==1)
    {
        Ellipse(hdc,rect.left,rect.top,rect.right,rect.bottom);
    }
    EndPaint(hwnd,&ps);

```

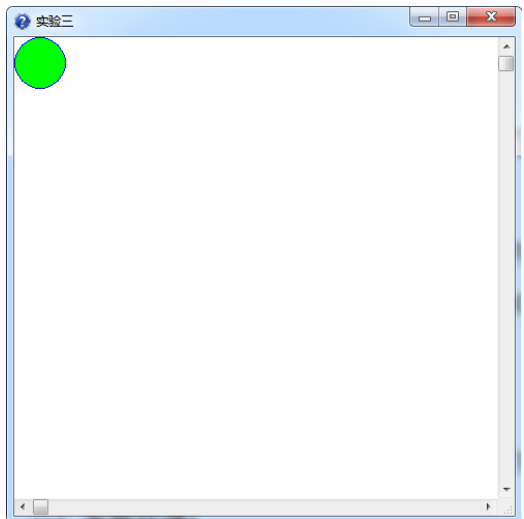


```
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    default:
        return(DefWindowProc(hwnd,message,wParam,lParam));
    }
    return 0;
}
```

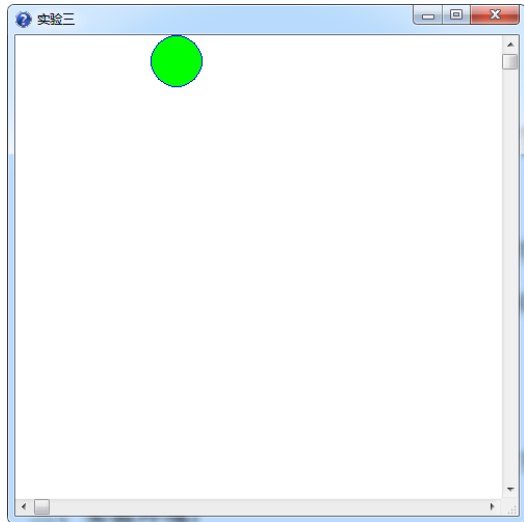
最终实验结果截图如下：



按下键盘上的"ctrl"键时，进行椭圆绘制：



按下 Home 或 End 键时，矩形或椭圆向左或向右移动：



- 3 在此基础上，添加如下功能：
- (1) 按下"shift"键时，进行矩形绘制；
  - (2) 按下左或右箭头，矩形或椭圆高度或长度加 10；
  - (3) 按下 PageUp 或 PageDown 时，矩形或椭圆向上或向下移动。