

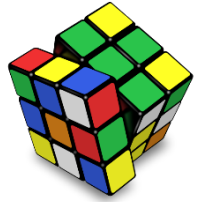
Ứng dụng MFC (Visual C++) trong mô phỏng Robot và hệ Cơ điện tử



Bài 3: Đồ họa với MFC

PHẠM MINH QUÂN

mquan.ph@gmail.com



Nội dung

1. Giới thiệu giao diện đồ họa GDI của MFC
2. Thực hành đồ họa với MFC
 - 2.1. *Vẽ lên dialog với hàm OnPaint()*
 - 2.2. *Vẽ hình động với hàm OnTimer()*
 - 2.3. *Quản lý hình vẽ bằng lớp (lập trình hướng đối tượng)*
3. Mở rộng

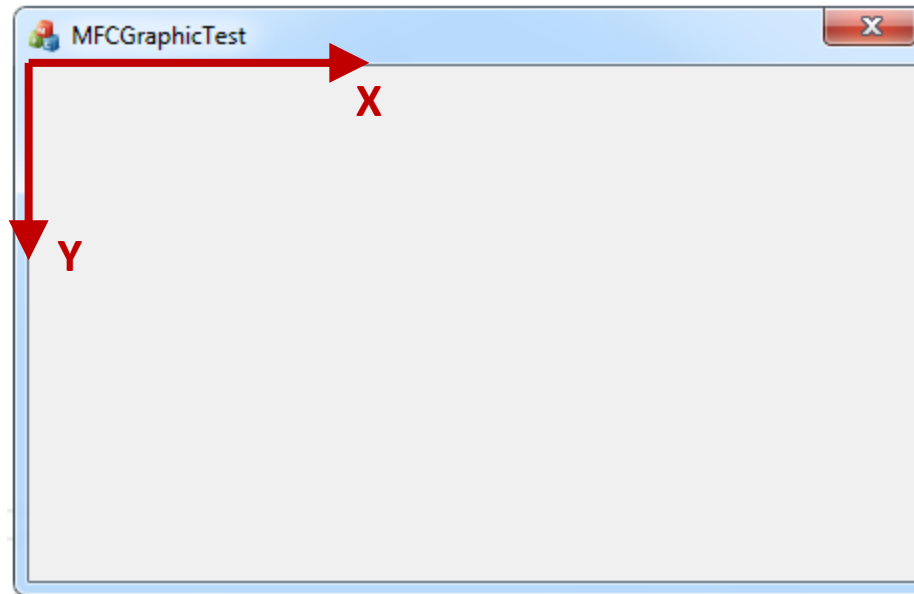
1. Giới thiệu giao diện đồ họa của MFC

- Graphics Device Interface - GDI: là giao diện thiết bị đồ họa ảo do Windows cung cấp, giúp vẽ đồ họa dễ dàng hơn, không phụ thuộc phần cứng của hệ thống.
- Device Context – DC/ Ngữ cảnh thiết bị: là cấu trúc chứa thông tin quản lý thiết bị hiển thị.
- Các công cụ:

Công cụ GDI	Lớp MFC hỗ trợ	Chức năng
Device Context	CDC	Ngữ cảnh thiết bị
Pen	CPen	Vẽ điểm hoặc đường thẳng
Brush	CBrush	Tô màu
Font	CFont	Quy định kiểu ký tự
Bitmap	CBitmap	Quản lý vùng ảnh điểm
Palette	CPalette	Quản lý bộ màu hiển thị
Region	CRgn	Quản lý vùng hiển thị có dạng bất kỳ

1. Giới thiệu giao diện đồ họa của MFC

- Hệ tọa độ trên dialog:



2. Thực hành đồ họa với MFC

2.1. Vẽ lên Dialog với hàm OnPaint()

- Tạo Project mới kiểu ứng dụng MFC là Dialog-based.
Xóa các control được tạo sẵn trên Dialog.
- Thêm đoạn code sau đây trong hàm OnPaint()

```
else  
{  
    CPaintDC dc(this);  
    CPen *pPen=new CPen(PS_SOLID,1,RGB(0,0,255));  
    dc.SelectObject(pPen);  
    dc.MoveTo(150,100);  
    dc.AngleArc(100,100,50,0,360);  
    dc.MoveTo(100,100);  
    dc.LineTo(150,100);  
    delete dc.SelectStockObject(WHITE_PEN);  
    CDialogEx::OnPaint();  
}
```

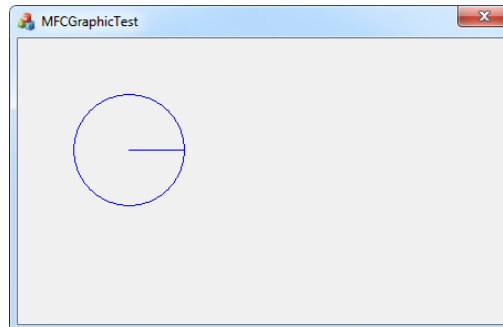
// Khai báo ngữ cảnh thiết bị (DC) → (Lưu ý: CPaintDC chỉ dùng trong hàm OnPaint)
// Khai báo con trỏ bút vẽ
// Chọn bút vẽ

// Vẽ hình

// Xóa bút vẽ



Kết quả chạy:



2. Thực hành đồ họa với MFC

2.2. Vẽ hình động với hàm *OnTimer()*

- Sử dụng Class Wizard để thêm hàm `OnTimer()` vào chương trình (như hướng dẫn trong Bài 1).
- Thêm một biến để lưu giữ thông tin chuyển động của hình. Khởi tạo giá trị ban đầu cho biến này.

File ...Dlg.h

```
float phi;
```

File ...Dlg.cpp

```
CMFCGraphicTestDlg::CMFCGraphicTestDlg(CWnd* pParent /*=NULL*/)
: CDialogEx(CMFCGraphicTestDlg::IDD, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    phi = 0;
}
```

- Thêm khai báo thư viện “math.h” (dùng cho các tính toán sin, cos...)

```
#include "stdafx.h"
#include "MFCGraphicTest.h"
#include "MFCGraphicTestDlg.h"
#include "afxdialogex.h"
#include "math.h"
```

2. Thực hành đồ họa với MFC

2.2. Vẽ hình động với hàm OnTimer()

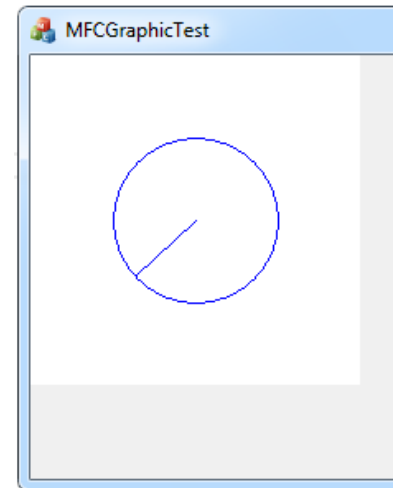
- Thêm câu lệnh sau vào hàm OnInitDialog() (thiết lập timer ngay từ khi mở hộp thoại)

```
// TODO: Add extra initialization here
SetTimer(1,100,NULL);
return TRUE; // return TRUE unless you set the focus to a control
}
```

- Viết đoạn code sau đây trong hàm OnTimer()

```
void CMFCGraphicTestDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    CClientDC dc(this); // Khai báo ngữ cảnh thiết bị (DC) (CClientDC dùng ngoài hàm OnPaint)
    dc.SetBkMode(TRANSPARENT);
    CPen *pPen=new CPen(PS_SOLID,1,RGB(0,0,255)); // Khai báo con trỏ bút vẽ
    CBrush *pBrush=new CBrush(RGB(255,255,255)); // Khai báo con trỏ chổi vẽ
    dc.SelectObject(pPen); // Chọn bút vẽ
    dc.SelectObject(pBrush); // Chọn chổi vẽ
    CRect rect(0,0,200,200);
    dc.FillRect(&rect,pBrush); // Tô màu hình chữ nhật nền
    dc.MoveTo(150,100);
    dc.AngleArc(100,100,50,0,360);
    dc.MoveTo(100,100);
    dc.LineTo(100+50*cos(phi*3.14/180),100+50*sin(phi*3.14/180)); // Vẽ hình
    phi++;
    if (phi>360) phi=0;
    delete dc.SelectStockObject(WHITE_PEN); // Xóa bút vẽ
    delete dc.SelectStockObject(WHITE_BRUSH); // Xóa chổi vẽ
    CDialogEx::OnTimer(nIDEvent);
}
```

➡ Kết quả chạy:
Bánh xe quay đều



2. Thực hành đồ họa với MFC

2.3. Quản lý hình vẽ bằng lớp (lập trình hướng đối tượng)

➤ Khai báo và định nghĩa các thuộc tính và phương thức của lớp mới như sau.

File Wheel.cpp

```
#include "stdafx.h"
#include "Wheel.h"
#include "math.h"
```

```
Wheel::Wheel(int x, int y, int r, bool dir,
             COLORREF C, float ph, float Spd)
```

```
{
    center_x=x;
    center_y=y;
    radius = r;
    crColor = C;
    DirCCW = dir;
    phi = ph;
    RotSpd = Spd;
}
```

```
Wheel::~Wheel(void)
{
}
```

```
void Wheel::SetProperties(int x, int y, int r, bool dir,
                        COLORREF C, float ph, float Spd)
```

```
{
    center_x=x;
    center_y=y;
    radius = r;
    crColor = C;
    DirCCW = dir;
    phi = ph;
    RotSpd = Spd;
}
```

```
void Wheels::DrawWheel(CDC*pDC)
```

```
{
    pDC->SetBkMode(TRANSPARENT);
    CPen *pPen=new CPen(PS_SOLID,1,crColor);
    CBrush *pBrush=new CBrush(RGB(255,255,255));
    pDC->SelectObject(pPen);
    pDC->SelectObject(pBrush);
    CRect rect(center_x-radius,center_y-radius,center_x+radius,center_y+radius);
    pDC->FillRect(&rect,pBrush);
    pDC->MoveTo(center_x+radius,center_y);
    pDC->AngleArc(center_x,center_y,radius,0,360);
    pDC->MoveTo(center_x,center_y);
    pDC->LineTo(center_x+radius*cos(phi*3.14/180),center_y+radius*sin(phi*3.14/180));
    if (DirCCW)phi += RotSpd; else phi -= RotSpd;
    if (phi>360) phi=0; else if (phi<0) phi =360;
    delete pDC->SelectStockObject(WHITE_PEN);
    delete pDC->SelectStockObject(WHITE_BRUSH);
}
```

2. Thực hành đồ họa với MFC

2.3. Quản lý hình vẽ bằng lớp (lập trình hướng đối tượng)

- Khai báo đối tượng của lớp mới trong lớp C...Dlg

File ...Dlg.h

```
// MFCGraphicTestDlg.h : header file
//
#include "Wheel.h"
...
Wheel circle1, circle2;
```

- Thiết lập các giá trị thuộc tính mong muốn cho đối tượng

File ...Dlg.cpp (hàm *OnInitDialog()*)

```
// TODO: Add extra initialization here
circle1.SetProperties(100,100,50,TRUE,RGB(0,0,255),0,3);
circle2.SetProperties(300,150,100,FALSE,RGB(255,0,0),90,1);
SetTimer(1,100,NULL);
```

2. Thực hành đồ họa với MFC

2.3. Quản lý hình vẽ bằng lớp (lập trình hướng đối tượng)

- Thay đổi hàm OnTimer()

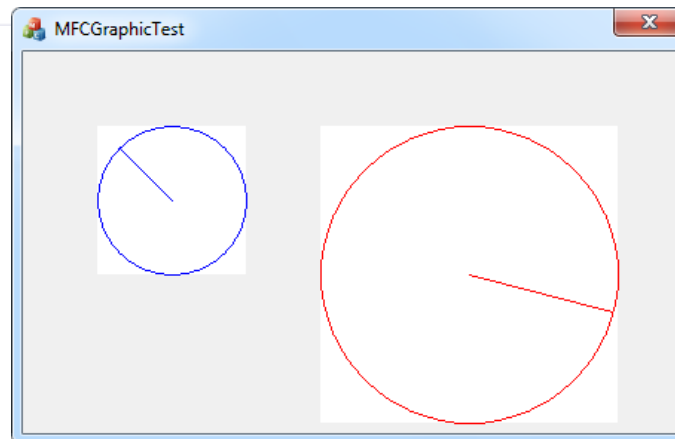
File ...Dlg.cpp

```
void CMFCGraphicTestDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: Add your message handler code here and/or call default
    CClientDC dc(this);
    circle1.DrawWheel(&dc);
    circle2.DrawWheel(&dc);

    CDialogEx::OnTimer(nIDEvent);
}
```



*Kết quả chạy:
Hai bánh xe quay với các
thông số khác nhau*



* Tương tự có thể thêm bao nhiêu đối tượng tùy thích, với giá trị thuộc tính tùy ý

3. Mở rộng

Tra cứu các lớp, hàm đồ họa của MFC trong thư viện MSDN

<https://msdn.microsoft.com/en-us/library/fxhhde73.aspx>

CDC Class

Visual Studio 2015 | [Other Versions](#) ▼

Defines a class of device-context objects.

Syntax

```
class CDC : public CObject
```

Members

Public Constructors

Name	Description
CDC::CDC	Constructs a CDC object.

Public Methods

Name	Description
CDC::AbortDoc	Terminates the current print job, erasing everything the application has written to the device since the last call of the StartDoc member function.
CDC::AbortPath	Closes and discards any paths in the device context.
CDC::AddMetaFileComment	Copies the comment from a buffer into a specified enhanced-format metafile.
CDC::AlphaBlend	Displays bitmaps that have transparent or semitransparent pixels.
CDC::AngleArc	Draws a line segment and an arc, and moves the current position to the ending point of the arc.
CDC::Arc	Draws an elliptical arc.
CDC::ArcTo	Draws an elliptical arc. This function is similar to Arc , except that the current position is updated.
CDC::Attach	Attaches a Windows device context to this CDC object.
CDC::BeginPath	Opens a path bracket in the device context.
CDC::BitBlt	Copies a bitmap from a specified device context.
CDC::Chord	Draws a chord (a closed figure bounded by the intersection of an ellipse and a line segment).
CDC::CloseFigure	Closes an open figure in a path.

Hết Bài 3

