

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

— \* —

# ĐỒ ÁN Project 1

## XÂY DỰNG CHƯƠNG TRÌNH VẼ PAINT

Sinh viên thực hiện: **Hoàng Minh Tuấn**  
Lớp CNTT2.01 -K57  
Giáo viên hướng dẫn: **Th.S. Nguyễn Đức Tiến**

HÀ NỘI 11 -2014

# PHIẾU GIAO NHIỆM VỤ MÔN HỌC

## 1. Thông tin về sinh viên

Họ và tên sinh viên: Hoàng Minh Tuấn

Điện thoại liên lạc: 098 512 6953

Lớp: CNTT 2.01 – K57

Email: tuan94bg@gmail.com

Hệ đào tạo: Đại học chính quy

## 2. Sản phẩm mục tiêu của môn học

Viết lại phần Vẽ Paint, giống như trong windows

## 3. Các tính năng

### Các tính năng bắt buộc:

- Lập trình/Tìm hiểu để hoàn thành sản phẩm: Có
- Sử dụng một kho repository để lưu trữ các phiên bản của project: Có  
<https://code.google.com/p/project-1-paint/>

### Các tính năng tùy chọn:

Tính năng	Có/Không
Sử dụng 10 đối tượng trong toolbox như button, text box, combo box, list box	Có
Xây dựng context menu (chuột phải) cho ít nhất 2 đối tượng khác nhau	Có
Xây dựng menubar với 3 cấp con bên trong và các event tương ứng	Có
Đọc thông tin từ file xml	Có
Sử dụng MS Blend để tạo giao diện, chỉ áp dụng với Project WPF	Có
Comment mã nguồn theo Doxygen, đạt tỷ lệ 40%	Có
Xem, nghe một đoạn video/hình ảnh, nhúng vào chương trình	Có
Sử dụng Word/Excel để vẽ giao diện chương trình, 5 form	Không
Kết nối database SQLServer/MySQL/PostgreSQL/MongoDB	Không
Giao diện Ribbon	Không

Tính năng khác nếu có: .....  
.....  
.....  
.....  
.....  
.....

## MỤC LỤC

<b>PHIẾU GIAO NHIỆM VỤ MÔN HỌC .....</b>	<b>2</b>
<b>MỤC LỤC .....</b>	<b>3</b>
<b>DANH MỤC HÌNH VẼ .....</b>	<b>4</b>
<b>DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ .....</b>	<b>4</b>
<b>PHẦN I: NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG .....</b>	<b>5</b>
1.1. Các sản phẩm tương tự .....	5
1.2. Các công nghệ kỹ thuật liên quan .....	5
1.2.1. Nền tảng : .....	5
1.2.2. Microsoft Blend .....	5
1.2.3. Google Code .....	6
1.2.4. Visual Studio 2013 .....	6
<b>PHẦN 2: PHÂN TÍCH THIẾT KẾ .....</b>	<b>7</b>
2.1 Cấu trúc dữ liệu .....	7
2.2 Mô hình giao diện .....	8
2.3 Các hàm chức năng .....	10
2.3.1 Chức năng vẽ: .....	10
2.3.2 Thao tác với tệp hình ảnh .....	11
2.3.3 Chức năng chọn màu sắc : .....	11
2.3.4 Chức năng thay đổi kích thước Panel Canvas .....	12
2.3.5 Chức năng chọn kích thước : .....	12
<b>PHẦN 3: SẢN PHẨM KẾT QUẢ .....</b>	<b>14</b>
3.1. Hướng dẫn cơ bản .....	14
3.1.1 Vẽ .....	14
3.1.2 Thay đổi kích thước vùng vẽ : .....	14
3.1.3 Thay đổi màu vẽ .....	15
3.1.4 Thay đổi kích thước đường thẳng ngoài danh sách .....	16
3.1.5 Xem công cụ và tọa độ chuột .....	16
3.2. Kết luận .....	16

## DANH MỤC HÌNH VẼ

Hình 1 Paint trong Window .....	5
Hình 2 Giao diện chính .....	8
Hình 3 Các phần của giao diện chính.....	9
Hình 4 Sản phẩm kết quả .....	14

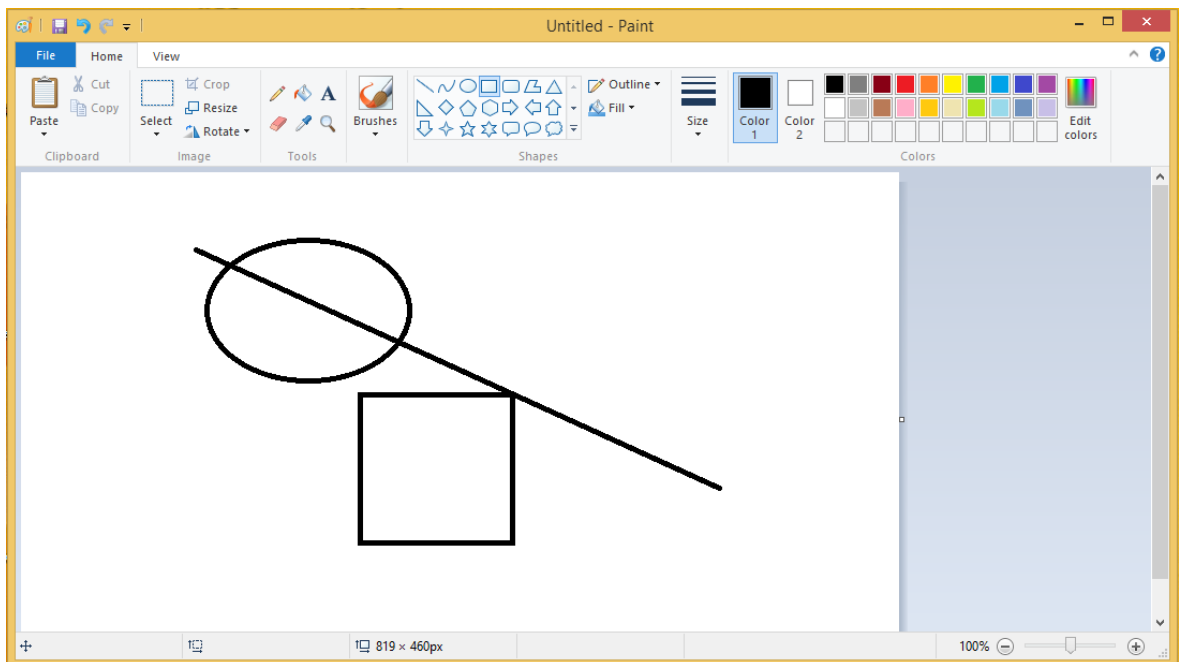
## DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

Số thứ tự	Từ viết tắt	Ý nghĩa
1	IDE	Integrated Development Environment
2	WPF	Windows Presentation Foundation
3	MS	Microsoft
4		
5		

# PHẦN I: NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN ỨNG DỤNG

## 1.1. Các sản phẩm tương tự

- Phần mềm vẽ Paint trong MS Window



Hình 1 Paint trong Window

## 1.2. Các công nghệ kỹ thuật liên quan

### 1.2.1. Nền tảng :

- Ứng dụng này được xây dựng bằng ngôn ngữ lập trình C#
- Là một ứng dụng Windows Presentation Foundation
- Sử dụng các thư viện đồ họa 2D đã được xây dựng

### 1.2.2. Microsoft Blend

Là một công cụ mở rộng trong Visual Studio để lập trình ứng dụng Windows. MS Blend là công cụ thiết kế giao diện người dùng cho cả ứng dụng app và web. Blend có giao diện tương tác kéo thả, giúp tạo giao diện ứng dụng nhanh chóng, đẹp mắt. Toàn bộ giao diện được lưu trữ trong một file XAML (dạng XML) và có thể mở được bằng



Visual Studio trong việc tạo ra các ứng dụng Windows Presentation Foundation và Silverlight.

Trong môn học này, MS Blend được dùng để tạo giao diện ứng dụng.

### 1.2.3. Google Code

Google Code <http://code.google.com>, là một trang web của Google trong đó tập trung các nhà phát triển các dự án phần mềm mã nguồn mở được hỗ trợ bởi google.

Trong đó có rất nhiều mã nguồn phần mềm và danh sách các dịch vụ có hỗ trợ các đặc tả API công cộng để phục vụ cho việc phát triển các phần mềm hỗ trợ khác.

Trong môn học này, Google Code được dùng để lưu trữ mã nguồn và báo cáo môn học.



### 1.2.4. Visual Studio 2013

Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.



Trong môn học này, Visual Studio 2013 được dùng là công cụ chính để biên tập mã nguồn.

## PHẦN 2: PHÂN TÍCH THIẾT KẾ

### 2.1 Cấu trúc dữ liệu

- Status.cs :

```
public class Status
```

+ Thuộc tính :

```
private String ToolBar;  
private Color FillColor;  
private Color BoderColor;  
private int Size;
```

- MainWindow.xaml : File chứa giao diện chính  
MainWindow.xaml.cs : Chứa code Behind của giao diện chính

```
public partial class MainWindow : Window
```

Thuộc tính :

```
public Status status = new Status();  
public Line line = new Line();  
public Polyline polyl = new Polyline();  
public Ellipse ellip = new Ellipse();  
public Rectangle rectangle = new Rectangle();  
public Point point_start = new Point();  
public Point pointdrag = new Point();  
public Boolean ClickDown = false;  
public Boolean Dragging = new Boolean();  
public TextBox text;  
public BitmapSource source;  
public Canvas savedCanvas = null;  
public Image img = new Image();
```

- PickColor.xaml : File chứa giao diện Dialog PickColor  
PickColor.xaml.cs : File chứa Code Behind của Dialog PickColor

```
public partial class PickColor : Window
```

Thuộc tính :

```
public Boolean result;  
public Color color;
```

- Resize.xaml : File chứa giao diện của dialog Resize  
Resize.xaml.cs : File chứa Code Behind của dialog Resize

```
public partial class Resize : Window
```

Thuộc tính :

```
public double width;  
public double height;
```

```
public String Mode="Pixel";  
public Boolean result = false;
```

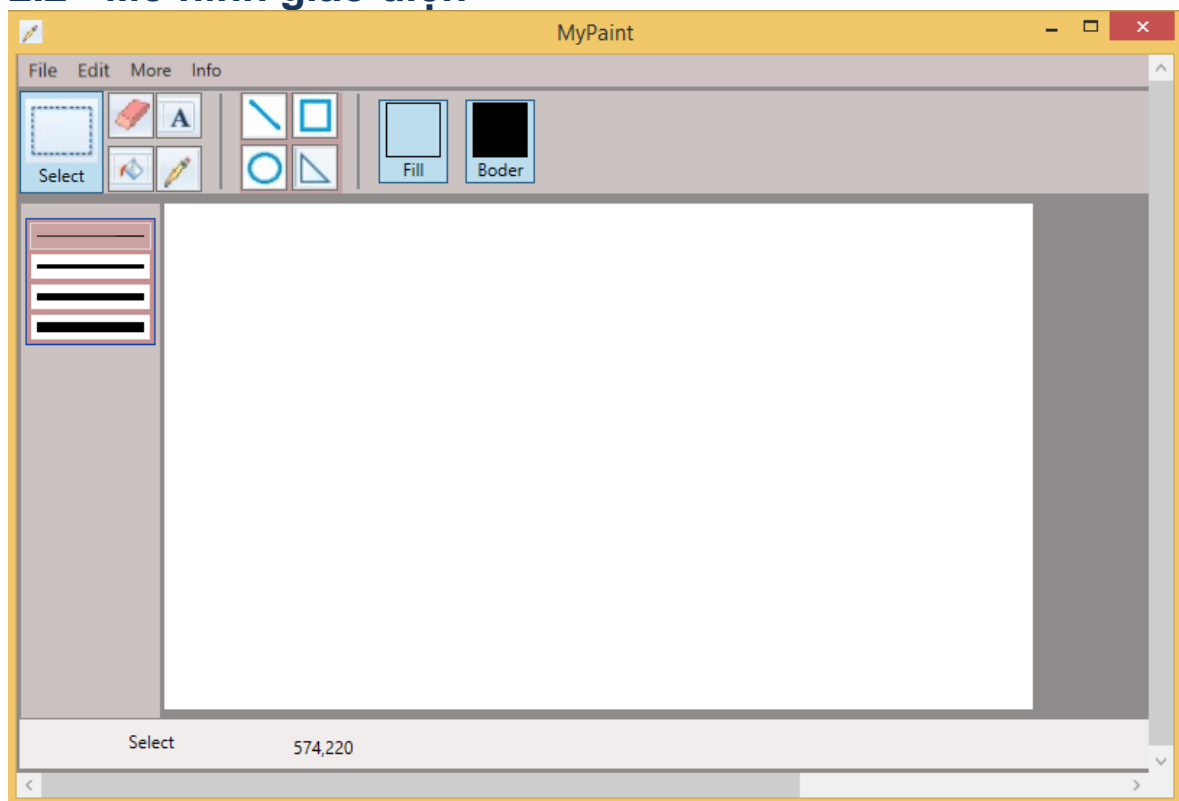
- MoreSize.xaml : File chứa giao diện dialog chọn kích thước đường thẳng  
MoreSize.xaml : File chứa code Behind của dialog chọn kích thước đường thẳng

```
public partial class MoreSize : Window
```

Thuộc tính :

```
public bool result = false;  
public bool change = false;  
public int int_size=2;
```

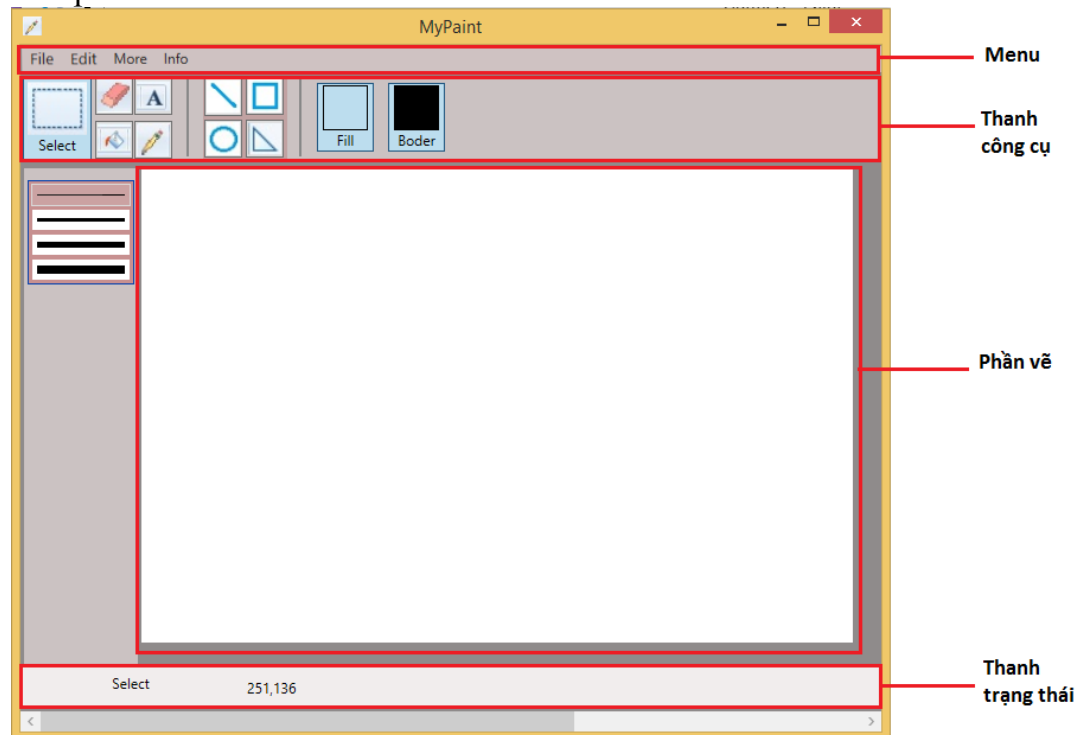
## 2.2 Mô hình giao diện



Hình 2 Giao diện chính



Các phần :



Hình 3 Các phần của giao diện chính

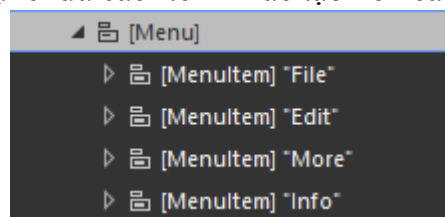
Trong đó gồm có :

1.2.1. Phần Menu và các cấp tương ứng :

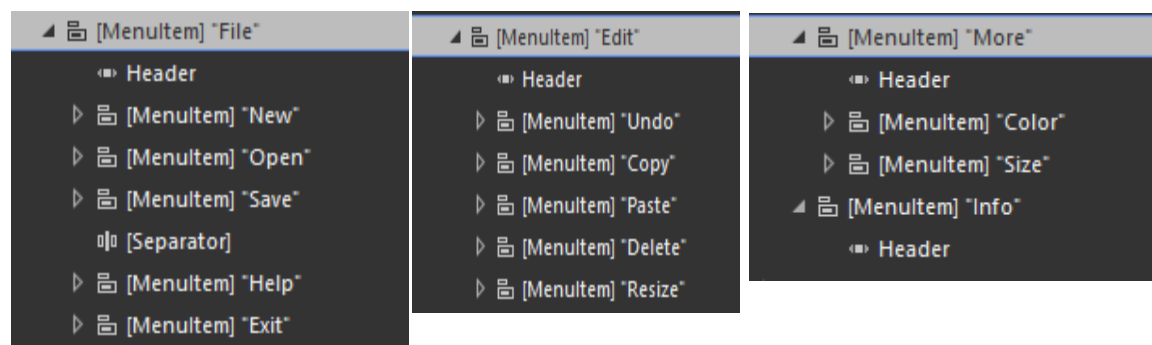
Menu nằm ở phía trên của sản phẩm, bao gồm các Item:



Trong các Item đó lại chứa các Item khác tạo nên các cấp khác nhau :



Các cấp thấp hơn:



### 1.2.2. Phần thanh công cụ



- **ToolBar** là một **Grid** chứa các **RadioButton** cho phép chọn công cụ để vẽ, màu sắc
- Gồm các công cụ có thể lựa chọn để sử dụng vẽ trong phần mềm.

Bao gồm :

- + Công cụ select
- + Tẩy
- + Thêm textbox
- + Bút
- + Vẽ hình ( đường thẳng, hình chữ nhật, elip)
- + Chọn màu

Ngoài ra còn có ListBox chọn kích thước đường thẳng ở phía bên trái giao diện.

### 1.2.3. Phần vẽ :

**Painting Area** là một panel **Canvas** được chứa trong một **Scrollviewer** có màu nền trắng cho phép vẽ các đối tượng đồ họa 2D lên đó.

### 1.2.4. Thanh trạng thái :

Hiển thị tọa độ chuột cũng như trạng thái hiện thời của phần mềm

-

## 2.3 Các hàm chức năng

### 2.3.1 Chức năng vẽ:

Sau khi chọn công cụ vẽ, trạng thái của phần mềm sẽ được thiết lập vào một biến status thuộc kiểu dữ liệu Status. Tùy thuộc vào trạng thái, mà các hàm xử lý sự kiện chuột sẽ thực hiện vẽ hình 2D bằng chuột (thứ tự `MouseLeftButtonDown => MouseMove=> Mouse LeftButtonUp`) trên **Painting Area**:

```
public void Mycanvas_Mouse_Down(object sender, MouseEventArgs e)
```

- + Hàm này để xử lý sự kiện chuột `MouseLeftButtonDown` trên phần **Painting Area**
  - + Tham số vào là đối tượng `sender` và sự kiện chuột `e` trong đó với tham số `e` sẽ cho biết điểm khi mà chuột `MouseDown` trên vùng **Painting Area**
  - + Hàm này được gọi ra khi Click chuột vào **Painting Area** cụ thể là **Panel Canvas**
- Hàm xử lý sự kiện `MouseMove` :

```
public void Mycanvas_Mouse_Move(object sender, MouseEventArgs e)
```

- + Hàm này xử lý sự kiện chuột `MouseMove` trên vùng **Painting Area**

- + Với tham số truyền vào là đối tượng sender và MouseEventArgs e trong đó e là tham số truyền vào có thể cho biết tọa độ điểm khi mà di chuyển chuột trên Window
- + Hàm này được gọi ra khi chuột di chuyển trên cửa sổ chương trình tuy nhiên đa phần chỉ được thực hiện khi đã MouseLeftButtonDown trên panel Canvas

```
public void Mycanvas_Mouse_Up(object sender, MouseEventArgs e)
```

- + Hàm này xử lý sự kiện chuột MouseLeft Up trên vùng Painting Area
- + Với tham số truyền vào là đối tượng sender và MouseEventArgs e trong đó e là tham số truyền vào có thể cho biết tọa độ điểm khi mà MouseLeftButtonUp trên Painting Area(Canvas)
- + Hàm này được gọi ra khi MouseLeftButtonUp trên Window MyPaint

### 2.3.2 Thao tác với tệp hình ảnh

- Có thể mở file Image hiển thị trên Panel Canvas hoặc từ hình vẽ 2D trên Panel Canvas xuất ra File dạng Image.
- Ví dụ :  
Khi Click vào Menu “File > Open > Open Image” thì hàm Open\_Click được gọi ra:

```
public void Open_Click(object sender, RoutedEventArgs e)
```

```
OpenFileDialog dlg = new OpenFileDialog();  
dlg.Filter = "All Image File | *.*";  
dlg.Multiselect = false;
```

```
ImageSource img = new BitmapImage(new Uri(dlg.FileName));  
Image bitmap = new Image { Source = img };
```

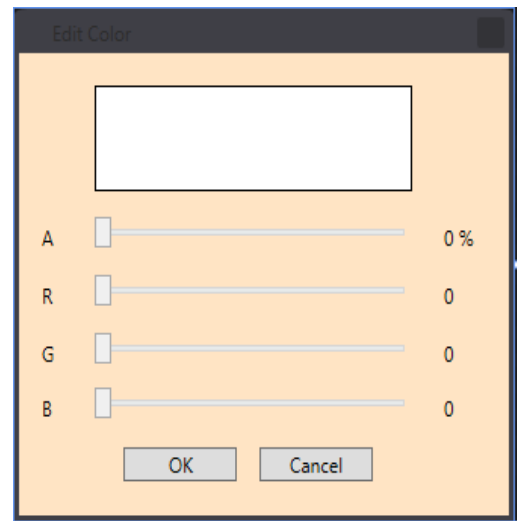
- ⇒ Hàm này sẽ mở một OpenFileDialog ra cho phép chọn các file Image sau đó hiển thị trên Panel Canvas thuộc Painting Area

### 2.3.3 Chức năng chọn màu sắc :

- Ngay khi **Click** vào Menu: More>Color>... hoặc Button Fill hay Boder thì Dialog **PickColor** này sẽ hiện ra.

- Dialog **PickColor** chứa 1 Grid Panel chứa tất cả các Control bên trong, bao gồm:

- + **Rectangle** nơi hiển thị màu sắc được chọn, khởi đầu là màu sắc đang dùng.
- + 4 **Slider** tương ứng với giá trị của 4 thuộc tính **ARGB** của **Color**, có thể thay đổi được. Ngay khi thay đổi 1 trong các giá trị này thì **Rectangle** trên sẽ hiển thị màu sắc hiện tại.
- + **Button OK** để cập nhật việc thay đổi màu, **Cancel** để hủy bỏ việc thay đổi màu sắc
- + Cập nhật màu sắc sẽ tùy thuộc vào Menu được chọn mà thay đổi **FillColor** hay **BoderColor**



### 2.3.4 Chức năng thay đổi kích thước Panel Canvas

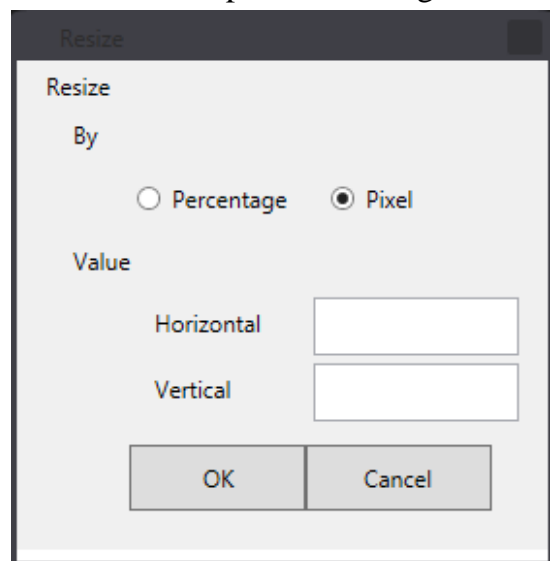
- Khi Click vào Menu : Edit > Resize thực hiện gọi hàm :

```
public void Resize_Click(object menuResize, RoutedEventArgs e)
```

thì một dialog Resize sẽ hiện ra cho phép nhập các giá trị để thay đổi kích thước phần Painting Area cụ thể là thay đổi kích thước Panel Canvas.

- Dialog này phần chính là một StackPanel chứa các thành phần bên trong;

- + Một StackPanel với nội dung theo chiều ngang chứa 2 RadioButton cùng một Group để chọn cách thức thay đổi kích thước theo phần trăm hay theo pixel.
- + 2 StackPanel khác chứa label và TextBox tương ứng với chiều ngang và chiều dọc, cho phép nhập giá trị số để thay đổi kích thước.
- + StackPanel chứa 2 button OK và Cancel cho phép đóng dialog này và cập nhật nếu click vào “OK”, hủy nếu “Cancel”.



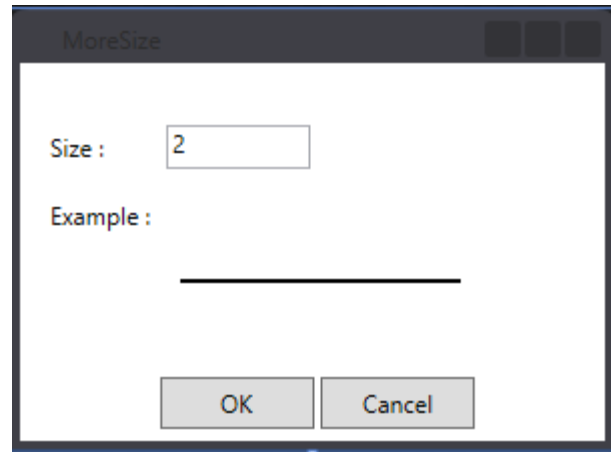
### 2.3.5 Chức năng chọn kích thước :

- Click phải chuột vào ListBox listSize chọn Menu ‘More’ hoặc kích vào Menu More>Size chương trình sẽ thực hiện

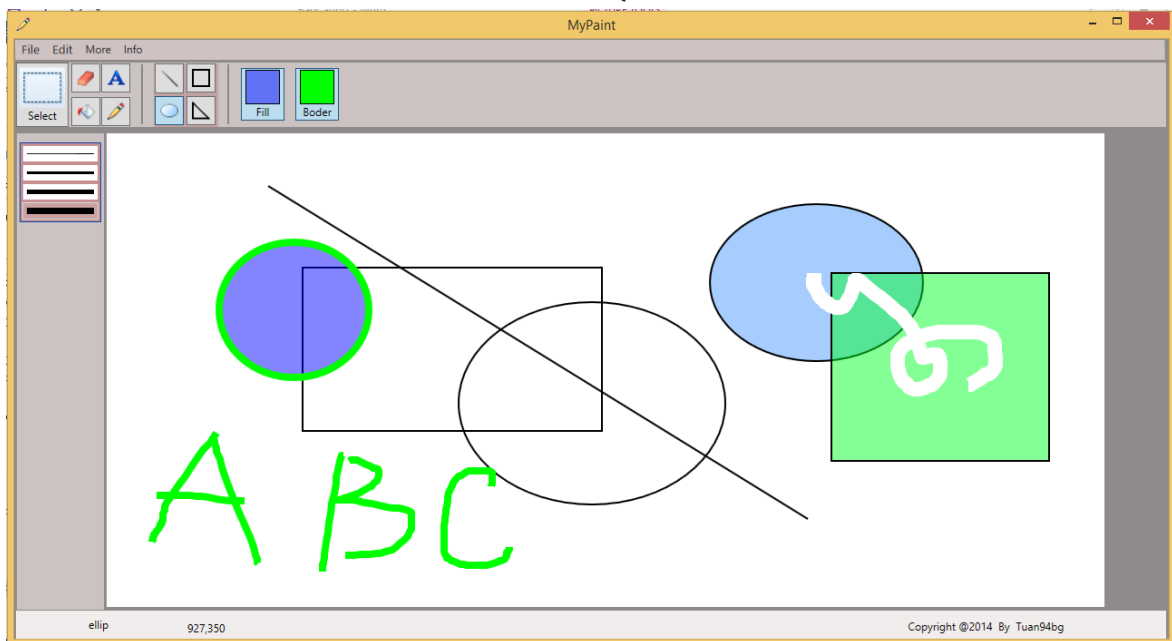
```
public void More_Size_Click(object sender, RoutedEventArgs e)
```

thì Dialog MoreSize hiện ra cho phép chọn kích thước đường thẳng.

- Dialog MoreSize bao gồm một TextBox để nhập giá trị số kích thước và một đường thẳng có kích thước thay đổi bằng với kích thước nhập vào.
- 2 button confirm OK, Cancel .



## PHẦN 3: SẢN PHẨM KẾT QUẢ



Hình 4 Sản phẩm kết quả

### 3.1. Hướng dẫn cơ bản

#### 3.1.1 Vẽ

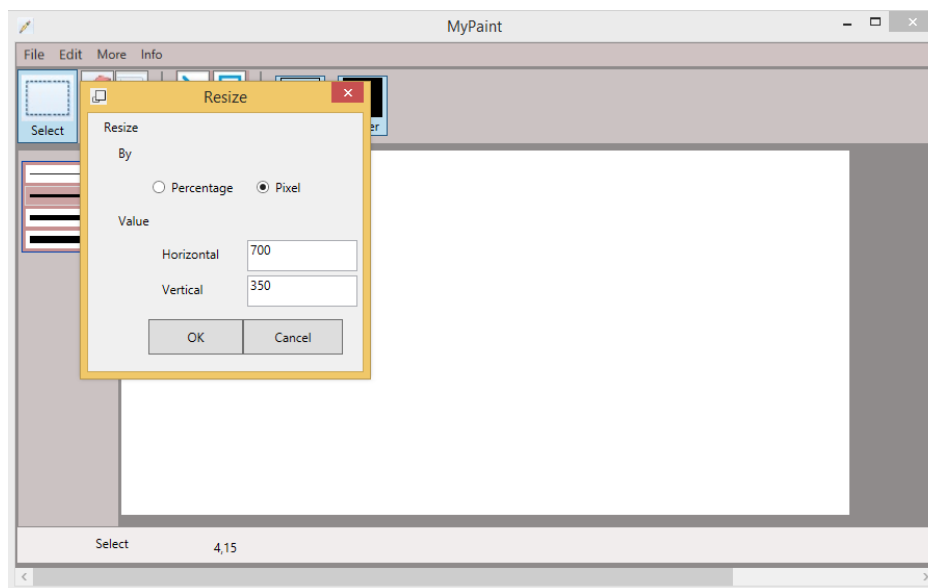
- Chọn một trong số các công cụ :



- Sử dụng chuột để vẽ:
  - + Đối với đường thẳng : Click chuột trái để xác định điểm đầu của đoạn thẳng và giữ chuột, di chuyển trên Painting Area , buông chuột để xác định điểm cuối và cố định đường thẳng
  - + Hình chữ nhật Click giữ và kéo chuột trái trên Panel Canvas để xác định 2 điểm Trái-Trên và Phải-Dưới của hình chữ nhật, buông chuột trái để cố định hình
  - + Hình elip: tương tự như hình chữ nhật, do hình elip được xác định nằm trong một hình chữ nhật


#### 3.1.2 Thay đổi kích thước vùng vẽ :

- Click vào Menu :  
Edit > Resize để mở cửa sổ thay đổi kích thước



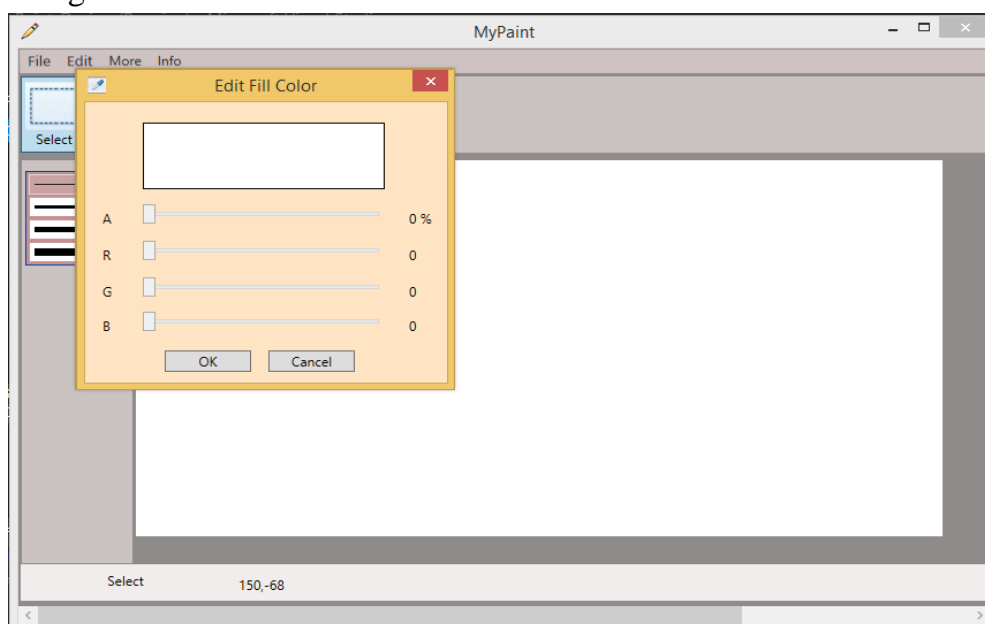
- Trong cửa sổ Resize có 2 cách thay đổi :
  - + Percentage : thay đổi theo % với giá trị ban đầu là 100%
  - + Pixel : Thay đổi theo giá trị chiều dài chiều rộng, giá trị ban đầu là kích thước hiện tại
- Phần Horizontal : nhập giá trị chiều ngang muốn thay đổi
- Phần Vertical : nhập giá trị chiều dọc muốn thay đổi
- Sau đó click vào Ok để thay đổi.

### 3.1.3 Thay đổi màu vẽ

- Click vào Menu More > Color>Fill hoặc Click vào  để chọn Fill Color cho các hình vẽ

Hoặc cũng có thể

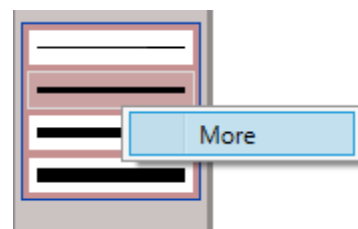
Vào Menu More > Color>Boder hay Click lên  để chọn màu cho đường thẳng và Boder cho các hình vẽ



- Chọn màu : Thay đổi các giá trị của những Slider để thay đổi màu với cơ chế hệ màu cộng:
  - + A : thông số Alpha thể hiện độ trong suốt
  - + R : Red - Giá trị thông số cho màu đỏ
  - + G : Green – Giá trị thông số cho màu xanh lá cây
  - + B : Blue – Giá trị thông số màu xanh dương
- Chọn OK để chọn, màu sẽ thay đổi ở Button được chọn ở bước 1.

### 3.1.4 Thay đổi kích thước đường thẳng ngoài danh sách

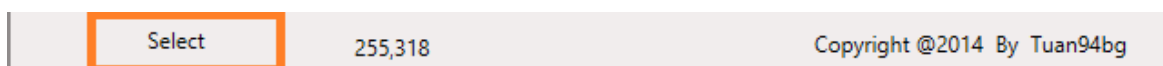
- Chọn More>Size hoặc click phải chuột vào danh sách chọn kích thước bên trái rồi chọn More ( như hình):



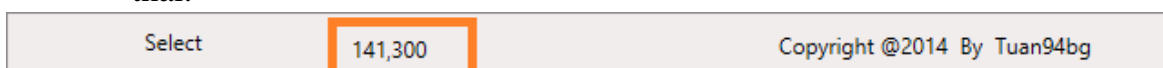
- Cửa sổ chọn kích thước xuất hiện, Nhập số trong phần Size, đường thẳng với kích thước đã nhập xuất hiện ở phía dưới.
- Chọn OK sau đó vẽ với kích thước mới đó.

### 3.1.5 Xem công cụ và tọa độ chuột

Công cụ đang được chọn có thể xem được bằng cách xem button nào trên thanh công cụ đang được chọn tuy nhiên có thể xem trên thanh trạng thái:



Tọa độ chuột tương đối với Panel Canvas được hiển thị trên thanh trạng thái:



## 3.2. Kết luận

- Chương trình là sản phẩm kết quả của quá trình tìm hiểu, thiết kế, xây dựng của cá nhân.
- Về mặt chức năng chính của một phần mềm vẽ có thể :
  - + Vẽ đường hình học ( đường thẳng, hình chữ nhật. hình tròn)
  - + Vẽ tự do như bằng bút
  - + Vẽ các hình có màu sắc
  - + Tẩy, xóa
  - + Lưu, mở file ảnh
  - + Paste ảnh từ Clipboard
  - + Chọn, di chuyển giống trong MS Paint