



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN  
MÔN: **ĐỒ HỌA MÁY TÍNH**

**BÁO CÁO ĐỒ ÁN**

## Paint 2D

*Đoàn Minh Hiếu – 1612198*

*Phan Minh Sơn - 1612888*

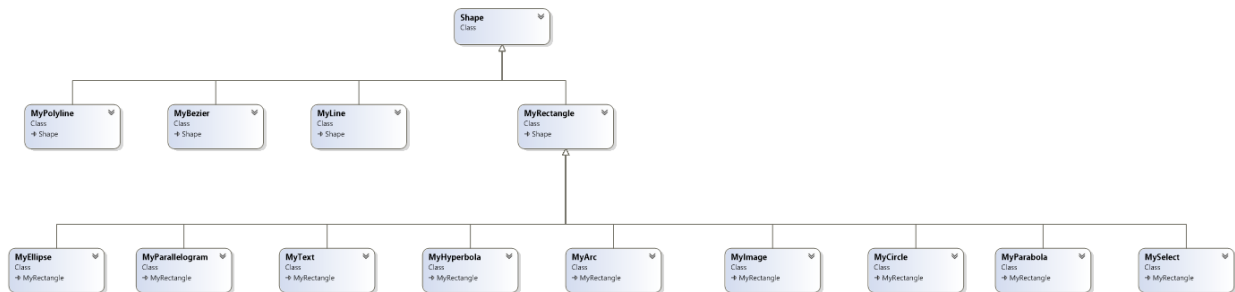
Tp Hồ Chí Minh ngày 05 tháng 11 năm 2017

## Mục Lục

I.	Cấu trúc chương trình Paint 2D .....	1
II.	Sử dụng .....	5
III.	Phân công công việc và mức độ hoàn thành .....	6
IV.	Tham khảo .....	7

## I. Cấu trúc chương trình Paint 2D

Sơ đồ UML của chương trình:



Các Class chính:

### 1. Class Shape:

#### ➤ Các thuộc tính chính:

- Điểm (PointF): startPoint, endpoint (tùy loại đối tượng mà mang vai trò khác nhau).
- Góc quay (float, độ): angle.
- Loại đối tượng (int): objType.
- GraphicsPath: gp – hình ảnh được vẽ của đối tượng.
- Danh sách Control Point (List<PointF>): listControlPoint.
- Cờ (bool):
  - isDraw – đối tượng được vẽ.
  - isFill – đối tượng được tô (chỉ các đối tượng kín mới được tô).
  - isMove – đối tượng được di chuyển.
  - isResize – đối tượng được Scale.
  - isRotate – đối tượng được quay (chỉ với một số đối tượng).
- Bút vẽ (Pen): p – Mang tính chất của nét vẽ (Size, Dash Style, Color).
- Kiểu Tô: br (HatchBrush), fillColor (Color), brStyle (int, kiểu tô đối tượng).

#### ➤ Các phương thức chính:

- Khởi tạo đối tượng: Shape ().
- Vẽ đối tượng: virtual void Draw (Graphics g).
- Quay đối tượng: virtual void rotate (Graphics g).
- Xác định Control Point: virtual void getListControlPoint ().
- Vẽ Control Point: virtual void showControlPoint (Graphics g).
- Tô màu đối tượng: virtual void fillPath (Graphics g, GraphicsPath gp).

- Gán các cờ trở lại bằng false: `virtual void release()`.
  - Tạo ra đối tượng copy: `virtual Shape Clone ()`.
  - Tạo đối tượng từ đọc file: `Shape (SerializationInfo info, StreamingContext ctxt)`.
  - Lưu đối tượng: `virtual void GetObjectData (SerializationInfo info, StreamingContext ctxt)`.
- Các sự kiện chuột:
- `virtual void mouseDown (MouseEventArgs e)`.
  - `virtual void mouseMove(MouseEventArgs e)`.
  - `virtual void mouseUp(MouseEventArgs e, object sender)`.
  - `virtual void mouseDoubleClick(int mode, MouseEventArgs e)`.

## 2. Class MyLine:

Line được xác định chủ yếu bởi 2 điểm `startPoint` và `endPoint` là 2 điểm mút của đoạn thẳng.

Control Point bao gồm 2 vị trí tại 2 đầu mút.

Các chức năng di chuyển hay thay đổi kích thước được xử lý trong các sự kiện chuột.

Di chuyển: di chuyển thực hiện bằng cách di chuyển cả `startPoint`, `endPoint`.

Biến hình: được thực hiện khi bắt được sự kiện nằm bên trong control Point. Scale làm thay đổi 1 trong 2 vị trí điểm mút.

## 3. Class MyBezier:

Bezier được xác định bằng 4 điểm trong (`Point []`) `boundPoint`. Tương ứng vị trí của 4 điểm đó là 4 control Point.

Các chức năng vi di chuyển, biến hình tương tự như MyLine.

## 4. Class MyPolyline:

MyPolyline được xác định bằng danh sách các đường Myline nối tiếp nhau (điểm endpoint của `listline[i]` là `startPoint` của `listline[i+1]`) (`List<Myline>`) `listline`. Danh sách Control Point là vị trí các điểm `startPoint` của các đường Myline và endpoint của đường MyLine cuối cùng.

Chức năng di chuyển của MyPolyline chính là việc move đồng thời tất cả các đường trong `listline`.

Chức năng Scale tương tự như MyLine.

## 5. Class MyRectangle:

Class Rectangle có thêm 1 thuộc tính (`GraphicsPath`) `mRec` là bounding box hình chữ nhật của MyRectangle, trong đối tượng này, `boundingBox` trùng với đối tượng được lưu trong (`GraphicsPath`) `gp`. `startPoint` chính là điểm trên bên trái và `endPoint` là điểm dưới bên phải của bounding box (Lưu ý: 2 điểm này là 2 điểm ở vị trí đã nêu bounding box khi xoay sang trạng thái nằm ngang).

Danh sách Control Point gồm 9 điểm, trong đó có 4 điểm là đỉnh của hình chữ nhật, 4 điểm là trung điểm của các cạnh và 1 điểm là điểm xoay.

Chức năng di chuyển là di chuyển cả mRec và gp cùng một khoảng dX và dY.

Chức năng biến dạng được thực hiện trong hàm Scale như sau: Xét trường hợp 4 đỉnh, chọn 1 đỉnh (PointF) CurP để Scale thì đỉnh đối diện của nó (PointF) ConstP được cố định. Đầu tiên, di chuyển hình về gốc tạo độ sao cho ConstP == O (0,0). Sau đó thực hiện phép xoay góc -angle tại O để chuyển hình thành trạng thái nằm ngang. Rồi tính tỷ lệ Scale để scale hình, sau đó xoay một góc angle tại O rồi dịch lại vị trí ban đầu (ConstP mới bằng ConstP cũ). Việc scale tại 4 trung điểm của cạnh được đưa về 4 đỉnh để thực hiện.

Các Class Còn lại là kế thừa lại Class MyRectangle nên các chức năng di chuyển và Scale cũng được kế thừa.

#### **6. Class MyEllipse:**

Ellipse được xác định bằng hình chữ nhật bounding box.

#### **7. Class MyCircle:**

Giống như MyEllipse nhưng với điều kiện là bounding box là hình vuông.

#### **8. Class MyParallelogram:**

Hình bình hành được xác định bởi bounding box hình chữ nhật và (float)subtract là độ lệch của độ dài cạnh nằm ngang so với chiều rộng hình chữ nhật, trong đó subtract được cố định subtract=width/4 (width: chiều rộng bounding box).

#### **9. Class MyParabola:**

Parabol được xác định bởi bounding box hình chữ nhật, parabol đi qua 2 đỉnh phía trên của bounding box và nhận trung điểm của cạnh ngang dưới làm tâm. Phương trình parabol là  $y = ax^2$  (luôn vị tự điểm tâm trở thành O (0,0)). Class lưu lại (float)f với  $a=1/(2f)$  và hình ảnh parabol được dựng bằng tập các đoạn thẳng tạo từ các điểm nó đi qua trong giới hạn bounding box. Và các điểm đó được tìm bằng phương pháp DDA.

Do có thể thay đổi kích thước và xoay nên parabol có thể thể hiện được mọi dạng phương trình parabol.

#### **10. Class MyHyperbola:**

Hyperbol cũng được xác định bởi bounding box hình chữ nhật, chọn tâm của hình chữ nhật là tâm của hyperbol và 2 đường chéo là tiệm cận, với (float) a=width/10 và (float) b=height/10. Phương trình hyperbol là  $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$  (luôn vị tự tâm về O (0,0)). Hình ảnh parabol được dựng bằng tập các đoạn thẳng tạo từ các điểm nó đi qua trong giới hạn bounding box. Và các điểm đó được tìm bằng phương pháp DDA.

### 11. Class MyArc:

Vẽ 1 nửa cung Ellipse.

### 12. Class MyText:

Vẽ chữ được gõ vào. Đối tượng có sử dụng (TextBox) tx để hỗ trợ việc gõ chữ.

### 13. Class MySelect:

Dùng để chọn một phần của bản vẽ để copy.

### 14. Class MyImage:

Dùng để load bitmap đã copy lên.

### 15. Class Paint2D:

Đây là class Form để hiện giao diện.

#### ➤ Các thuộc tính chính:

- Đối tượng đang được lựa chọn (Shape): curObj.
- Image được dùng để hiện thị (Bitmap): PrimaryBMP, load BMP.
- Danh sách các đối tượng (List<Shape>): listObj.
- Loại đối tượng (int): objType.
- Loại chức năng thực hiện (int): mode (1: Vẽ ,2: Chọn,3: Di chuyển,4: Biến hình).
- Đối tượng lưu trữ Copy (Shape): copyShape.
- Binary Formatter (BinaryFormatter): formatter.

#### ➤ Các phương thức chính:

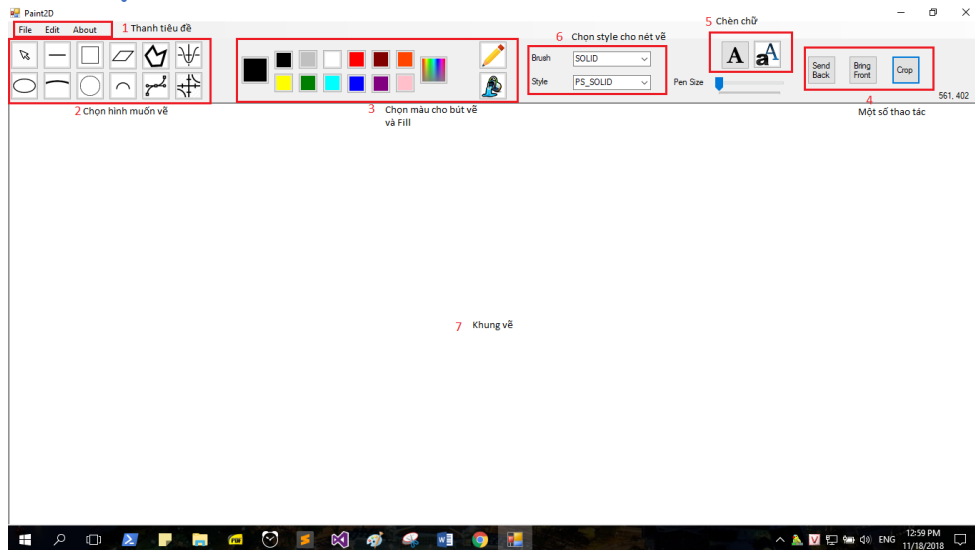
- Khởi tạo chương trình: Paint2D ().
- Vẽ các đối tượng: void picBox\_paint (object sender, PaintEventArgs e).
- Khởi tạo đối tượng: void initObject (int objType).
- Các chức năng khác phục vụ ...

#### ➤ Các sự kiện chuột:

- void onMouseDown (object sender, MouseEventArgs e).
- void onMouseUp (object sender, MouseEventArgs e).
- void onMouseMove (object sender, MouseEventArgs e).
- void onMouseDoubleClick (object sender, MouseEventArgs e).

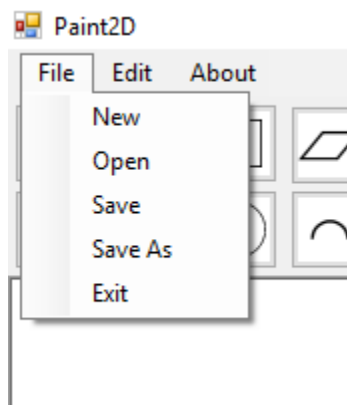
## II. Sử dụng

### 1. Giao diện



Ban đầu, chương trình sẽ ở chế độ chọn, click vào các hình cần vẽ để vẽ. Để chuyển về chế độ chọn ta click vào icon con trỏ bên góc trái, lúc này ta có thể thực hiện các thao tác trên các đối tượng đã vẽ như: move, scale, đổi màu, fill, copy, paste, ...

### 2. Một số thao tác



Chọn Save để thực hiện lưu: gồm 2 chế độ: lưu nhị phân và lưu bitmap

Chọn Open để load hình đã lưu (2 chế độ tương ứng)

Chọn Exit để thoát khỏi chương trình

Chọn New để tạo mới

### III. Phân công công việc và mức độ hoàn thành

Công việc	Mức độ hoàn thành	Người thực hiện
<b>Yêu cầu 1: Vẽ các đối tượng đồ họa 2D</b>		
- Đường thẳng, hình chữ nhật, hình bình hành, đa giác, đường gấp khúc	100%	Đoàn Minh Hiếu
- Đường tròn, cung tròn, Ellipse, cung ellipse, Parabola, Hyperbola	100%	Phan Minh Sơn
- Kí tự	100%	Đoàn Minh Hiếu
- Đường cong Bezier	100%	Phan Minh Sơn
- Xóa đi một phần bản vẽ	50% (Chỉ có xóa cả đối tượng)	Đoàn Minh Hiếu
<b>Yêu cầu 2: Cho phép người dùng nạp bản vẽ xuống đĩa, nạp bản vẽ có sẵn từ đĩa lên</b>	100%	Đoàn Minh Hiếu
<b>Yêu cầu 3: Cho phép chọn và thay đổi các thuộc tính của đối tượng</b>		
- Thay đổi các thuộc tính đường biên	100%	Phan Minh Sơn
- Thay đổi các thuộc tính tô màu	100%	Phan Minh Sơn
- Chọn font chữ	100%	Phan Minh Sơn



<b>Yêu cầu 4: Cho phép thực hiện các phép biến hình trên các đối tượng</b>		
- Phép quay	95% (Text chưa có phép quay như MS PowerPoint)	Phan Minh Sơn
- Tịnh tiến	100%	Đoàn Minh Hiếu
- Tỷ lệ	95% (Hình bình hành, cung Ellipse chưa có điều chỉnh như MS PowerPoint,)	Phan Minh Sơn
- Biến dạng	0%	
<b>Yêu cầu 5: Các chức năng nâng cao</b>		
- Thay đổi thứ tự trên-dưới giữa các đối tượng.	100%	Đoàn Minh Hiếu
- Save và Load Image (BMP, GIF, JPE, PNG, TIFF).	100%	Đoàn Minh Hiếu
- Copy,Cut,Paste.	50% (Chỉ có Copy dành cho đơn đối tượng và 1 phần ảnh bitmap (Mức độ 1))	Phan Minh Sơn

#### IV. Tham khảo

1. Sách Eric White - Pro .NET 2.0 Graphics Programming (2005)
2. Custom-Serialization-Example  
<https://www.codeproject.com/Articles/22787/Custom-Serialization-Example>
3. Slide bài giảng môn Đồ họa máy tính, Đại học KHTN Tp HCM