Tên: Huỳnh Thanh Quy - 21110622

**Đồ án giữa kì môn Lập trình window**

**Paint application**

**1/ Giới thiệu chung về paint app:**

App paint tạo ra thực hiện các chức năng chính như sau:

-Chức năng cơ bản:

+ Vẽ và fill được các shape như: hình chữ nhật, ellipse, đa giác, đường cong, đường thẳng.

+ cho người dùng chọn màu, size nét vẽ

+ chọn và kéo thả các shape

+ phóng to thu nhỏ kích thước của shape

-Chức năng nâng cao:

+ Xóa được các shape đã chọn (có thể chọn nhiều shape nhờ vào Ctr+click chuột)

+ Group và ungroup được các shape.

**A/ Giao diện chính:**

**A screenshot of a computer

Description automatically generated**

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated

**B/ giải thích sơ về cấu trúc code:**

A screenshot of a computer

Description automatically generated

**Folder : Class\_Shape**

- chứa các class vẽ hình ảnh

- shapeObject class chính

+ chứa group

+ ungroup

+ 8 hình vuông để resize

**2/ Chức năng:**

A picture containing font, white, graphics, logo

Description automatically generated A screenshot of a cell phone

Description automatically generated with low confidence

Vd: Tạo đối Elipse

resetVar();

objectCurr = new clsEllipse();

private void pnlMain\_MouseDown(object sender, MouseEventArgs e){

if (objectCurr != null)

{

this.isPress = true;

objectCurr = Init\_Object(objectCurr);

objectCurr.lstPoints.Add(e.Location);

this.lstObject.Add(objectCurr);

return;

}

}

private void pnlMain\_MouseMove(object sender, MouseEventArgs e)

{

if (this.isPress)

{

var obj = this.lstObject[this.lstObject.Count - 1].lstPoints;

if (obj.Count < 2)

obj.Add(e.Location);

else obj[obj.Count - 1] = e.Location;

this.pnlMain.Refresh();

return;

}

}

private void pnlMain\_MouseUp(object sender, MouseEventArgs e)

{

if (this.isPress)

{

this.isPress = false;

this.lstObject[this.lstObject.Count - 1].lstPoints[1] = e.Location;

this.lstObject[this.lstObject.Count - 1].updatePoints();

objectCurr = null;

this.pnlMain.Refresh();

}

}

Mô tả: sử đụng mouse down để bắt đầu vẽ, mouse move để hiện đối tượng đang vẽ, mouse up hoàn thành đổi tượng vẽ

A picture containing frame

Description automatically generated A black rectangle with a white background

Description automatically generated with low confidence

Vd: Vẽ polygon

private void btnPolygon\_Click(object sender, EventArgs e)

{

resetVar();

isPolygon = true; // Bật vẽ hình

objectCurr = new clsPolygon();

}

private void pnlMain\_MouseDown(object sender, MouseEventArgs e)

{

if (isPress)

{

objectCurr.lstPoints.Add(e.Location);

if (e.Button == MouseButtons.Right)

{

objectCurr = null;

isPress = false;

}

return;

}

if (objectCurr != null)

{

this.isPress = true;

objectCurr = Init\_Object(objectCurr);

objectCurr.lstPoints.Add(e.Location);

this.lstObject.Add(objectCurr);

return;

}

}

Mô tả: sử dụng trái để vẽ hình, chuột phải để kết thúc hình ,

A picture containing rectangle, screenshot, line, design

Description automatically generated A red triangle on a white background

Description automatically generated with medium confidence

private void btnFill\_Click(object sender, EventArgs e)

{

isFill = !isFill;

btnFill.FlatStyle = FlatStyle.Standard;

btnFill.FlatAppearance.BorderColor = SystemColors.ControlDarkDark;

if (isFill)

{

btnFill.FlatStyle = FlatStyle.Flat;

btnFill.FlatAppearance.BorderColor = Color.Orange;

}

}

Mô tả: bật Fill để vẽ hình ảnh lấp đầy.

A picture containing colorfulness, screenshot, design

Description automatically generated

Mô tả : Chọn mùa sắc, màu hiện tại ở Color.

A screenshot of a computer

Description automatically generated with low confidence A picture containing screenshot, circle, design

Description automatically generated

private void btnZoomin\_Click(object sender, EventArgs e)

{

pnlMain.Scale(new SizeF(2.0f, 2.0f));

foreach (var obj in lstObject)

{

zoom(obj, 2.0f);

if (obj.getGroup().Count !=0) // nối tượng đã được group

{

obj.lstPoints[0] = new PointF(obj.lstPoints[0].X \* 2, obj.lstPoints[0].Y \* 2);

obj.lstPoints[1] = new PointF(obj.lstPoints[1].X \* 2, obj.lstPoints[1].Y \* 2);

}

}

pnlMain.Refresh();

}

Mô tả: khi nhấn vào zoom đối tượng vẽ được tăng/ giảm kích thước , chiều rộng, panel được mở rộng.

A screenshot of a computer

Description automatically generated with medium confidence

private void pnlMain\_MouseClick(object sender, MouseEventArgs e)

{

if (Control.ModifierKeys == Keys.Control && e.Button == MouseButtons.Left)

{

clsDrawObject cls = checkCursorCurr(e.Location);

if (cls != null)

{

isObjSelected = true;

if (!lstSelected.Contains(cls))

this.lstSelected.Add(cls);

this.pnlMain.Refresh();

}

}

}

Mô tả : đùng ctrl +click để chọn đối tượng, nhiều đối tượng. (xác định đối tượng được chọn bằng con trỏ chuột ).

A picture containing line, design

Description automatically generated

Mô tả: vẽ ra một hình vuông các đối tượng nằm trong hình vuông sẽ được chọn.(sử dụng vẽ hình vuông nhưng không lưu).

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated with medium confidence A black and red lines on a white background

Description automatically generated with low confidence

Mô tả: chọn dash Style của pen , chọn style của brush

A picture containing screenshot, rectangle, electric blue, design

Description automatically generated A picture containing sketch, linedrawing, design

Description automatically generated

private void pnlMain\_MouseDown(object sender, MouseEventArgs e)

{} Tạo đối tượng

private void pnlMain\_MouseMove(object sender, MouseEventArgs e)

{

if (isPen && isPress)

{

this.lstObject[this.lstObject.Count - 1].lstPoints.Add(e.Location);

pnlMain.Refresh();

return;

}

}// them đối tượng

private void pnlMain\_MouseUp(object sender, MouseEventArgs e)

{

if (isPen)

{

this.lstObject[this.lstObject.Count - 1].lstPoints.Add(e.Location);

isPen = false;

objectCurr = null;

isPress = false;

return;

}}

// kết thúc vẽ pen

Mô tả : bút vẽ được tạo bằng các lines với nhau

A screenshot of a computer

Description automatically generated with low confidence A red and blue rectangle with blue lines

Description automatically generated with low confidence

private void btnGroup\_Click(object sender, EventArgs e)

{

if (lstSelected.Count > 1)

{

clsDrawObject objGroup = new clsDrawObject();

objGroup.CreateGroupObject(lstSelected);

float x\_min = lstSelected.Min(x => x.lstPoints.Min(p => p.X));

float y\_min = lstSelected.Min(y => y.lstPoints.Min(p => p.Y));

float x\_max = lstSelected.Max(x => x.lstPoints.Max(p => p.X));

float y\_max = lstSelected.Max(y => y.lstPoints.Max(p => p.Y));

lstSelected.ForEach(obj =>

{

lstObject.Remove(obj);

});

objGroup.lstPoints.Add(new PointF(x\_min, y\_min));

objGroup.lstPoints.Add(new PointF(x\_max, y\_max));

lstObject.Add(objGroup);

lstSelected.Clear();

lstSelected.Add(objGroup);

this.pnlMain.Refresh();

}

public void CreateGroupObject(List<clsDrawObject> lstObjSelected)

{

this.lstGroups.AddRange(lstObjSelected);

}

// tạo ra một đối tượng mới, them các đối tượng gộp nhóm vào lstGroups của clsDrawObjet

//xóa các đội tượng được gộp nhóm khỏi lstObject

Mô tả: group các đối tượng dược chọn ,

A picture containing rectangle, line, screenshot, square

Description automatically generatedA red and blue rectangle with black dots

Description automatically generated with low confidence

private void btnUngroup\_Click(object sender, EventArgs e)

{

if (lstSelected.Count > 0)

{

List<clsDrawObject> lstObjGroup = new List<clsDrawObject>();

clsDrawObject objGroup = new clsDrawObject();

lstObjGroup = objGroup.UngroupObjects(lstSelected);

if (lstObjGroup.Count == 0) return;

lstSelected.ForEach(obj =>

{

lstObject.Remove(obj);

});

lstSelected.Clear();

lstSelected.AddRange(lstObjGroup);

lstObject.AddRange(lstObjGroup);

this.pnlMain.Refresh();

}

}

public List<clsDrawObject> UngroupObjects(List<clsDrawObject> lstObjSelected)

{

List<clsDrawObject> lstObj = new List<clsDrawObject>();

lstObjSelected.ForEach(x =>

{

if (x.getGroup().Count == 0)

lstObj.Add(x);

else lstObj.AddRange(x.lstGroups);

});

return lstObj;

}

// lấy lại các đối tượng lstGoups của đối tượng được chọn, xóa đối tượng được chọn //khỏi lstObject, các đối tượng được lấy ra từ lstGroups add lại vào lstObject

Mô tả: tách các đối tượng được group

A picture containing symbol, design

Description automatically generated

private void btnDelete\_Click(object sender, EventArgs e)

{

foreach (var x in lstSelected)

{

lstObject.Remove(x);

}

lstSelected.Clear();

this.pnlMain.Refresh();

}

// Xóa các đối tượng được chọn khỏi lstObject

Mô tả: Xóa các đối tượng được chọn.

A picture containing symbol, electric blue, design

Description automatically generated

private void btnClear\_Click(object sender, EventArgs e)

{

this.lstObject.Clear();

this.lstSelected.Clear();

this.pnlMain.Refresh();

}

//Xóa tất cả lstObject, ..

Mô tả: xóa tất cả đối tượng trên panel vẽ

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

private clsDrawObject moveObject(clsDrawObject clsObj, PointF start, PointF end)

{

this.pointStart = end;

float x = end.X - start.X;

float y = end.Y - start.Y;

moveCoordAllOfOgroup(clsObj, x, y);

clsObj.calculateResizePoints();

return clsObj;

}

//duy chuyển đối tượng

private void moveCoordAllOfOgroup(clsDrawObject clsObj, float x, float y)

{

for (int i = 0; i < clsObj.lstPoints.Count; i++)

{

PointF p = clsObj.lstPoints[i];

clsObj.lstPoints[i] = new PointF(p.X + x, p.Y + y);

}

clsObj.getGroup().ForEach(obj =>

{

moveCoordAllOfOgroup(obj, x, y);

});

}

// duy chuyển nhiều đối tượng cùng lúc trong trường đã groups

Mô tả : đề chuột vào đối tượng để di chuyển đối tượng từ ảnh 1 sang ảnh 2

A screenshot of a computer

Description automatically generated with medium confidence A screenshot of a computer

Description automatically generated with medium confidence

private void resizeObject(clsDrawObject clsObj, int ind, float x, float y)

{

switch (ind)

{

case 0:

clsObj.lstPoints[0] = new PointF(clsObj.lstPoints[0].X + x, clsObj.lstPoints[0].Y + y);

break;

case 4:

clsObj.lstPoints[1] = new PointF(clsObj.lstPoints[1].X + x, clsObj.lstPoints[1].Y + y);

break;

case 2:

clsObj.lstPoints[0] = new PointF(clsObj.lstPoints[0].X, clsObj.lstPoints[0].Y + y);

clsObj.lstPoints[1] = new PointF(clsObj.lstPoints[1].X + x, clsObj.lstPoints[1].Y);

break;

case 6:

clsObj.lstPoints[0] = new PointF(clsObj.lstPoints[0].X + x, clsObj.lstPoints[0].Y);

clsObj.lstPoints[1] = new PointF(clsObj.lstPoints[1].X, clsObj.lstPoints[1].Y + y);

break;

case 1:

clsObj.lstPoints[0] = new PointF(clsObj.lstPoints[0].X, clsObj.lstPoints[0].Y + y);

break;

case 5:

clsObj.lstPoints[1] = new PointF(clsObj.lstPoints[1].X, clsObj.lstPoints[1].Y + y);

break;

case 7:

clsObj.lstPoints[0] = new PointF(clsObj.lstPoints[0].X + x, clsObj.lstPoints[0].Y);

break;

case 3:

clsObj.lstPoints[1] = new PointF(clsObj.lstPoints[1].X + x, clsObj.lstPoints[1].Y);

break;

default: break;

}

clsObj.calculateResizePoints();

clsObj.getGroup().ForEach(obj =>

{

resizeObject(obj, ind, x, y);

});

}

// Xác định vị trí cần thay đổi

Mô tả : Chọn vào đội , click vào 8 hình vuông để thay đổi kích thước của đối tượng