ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



Xây dựng ứng dụng Paint

Môn học: Lập trình trực quan.

Lớp: IT008.J11.

Giảng viên: Huỳnh Tuấn Anh.

Danh sách thành viên:

|  |  |
| --- | --- |
| Họ & tên | MSSV |
| Hàng Đức Khuê | 15520388 |
| Trần Công Lập (Nhóm trưởng) | 15520415 |
| Trần Công Luận | 15520456 |

Mục lục

[**I.** **Thanh tiêu đề** 3](#_Toc533725116)

[I.1. Save 3](#_Toc533725117)

[I.2. Undo 4](#_Toc533725118)

[I.3. Redo 5](#_Toc533725119)

[**II.** **File Menu** 5](#_Toc533725120)

[II.1. New 5](#_Toc533725121)

[II.2. Open 6](#_Toc533725122)

[II.3. Save 8](#_Toc533725123)

[II.4. About Paint 8](#_Toc533725124)

[II.5. Close 8](#_Toc533725125)

[II.6. Recent Documents 9](#_Toc533725126)

[**III.** **Thanh công cụ** 10](#_Toc533725127)

[III.1. Image 10](#_Toc533725128)

[III.1.a. Select 10](#_Toc533725129)

[III.1.b. Resize 11](#_Toc533725130)

[III.2. Clipboard 12](#_Toc533725131)

[III.2.a. Cut 12](#_Toc533725132)

[III.2.b. Copy 12](#_Toc533725133)

[III.2.c. Paste 13](#_Toc533725134)

[III.3. Tools 13](#_Toc533725135)

[III.3.a. Pencil 13](#_Toc533725136)

[III.3.b. Eraser 13](#_Toc533725137)

[III.3.c. Colorpicker 14](#_Toc533725138)

[III.4. Fill tools 14](#_Toc533725139)

[III.4.a. Solid Color 14](#_Toc533725140)

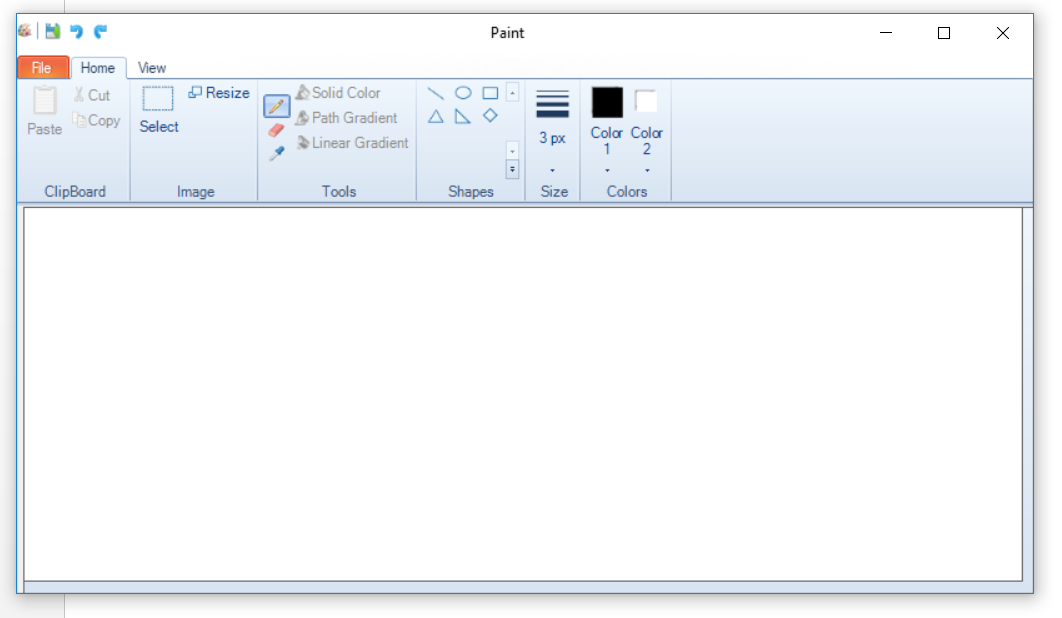
[III.4.b. Path Gradient 15](#_Toc533725141)

[III.4.c. Linear Gradient 17](#_Toc533725142)

[III.5. Shapes 18](#_Toc533725143)

[III.6. Size 20](#_Toc533725144)

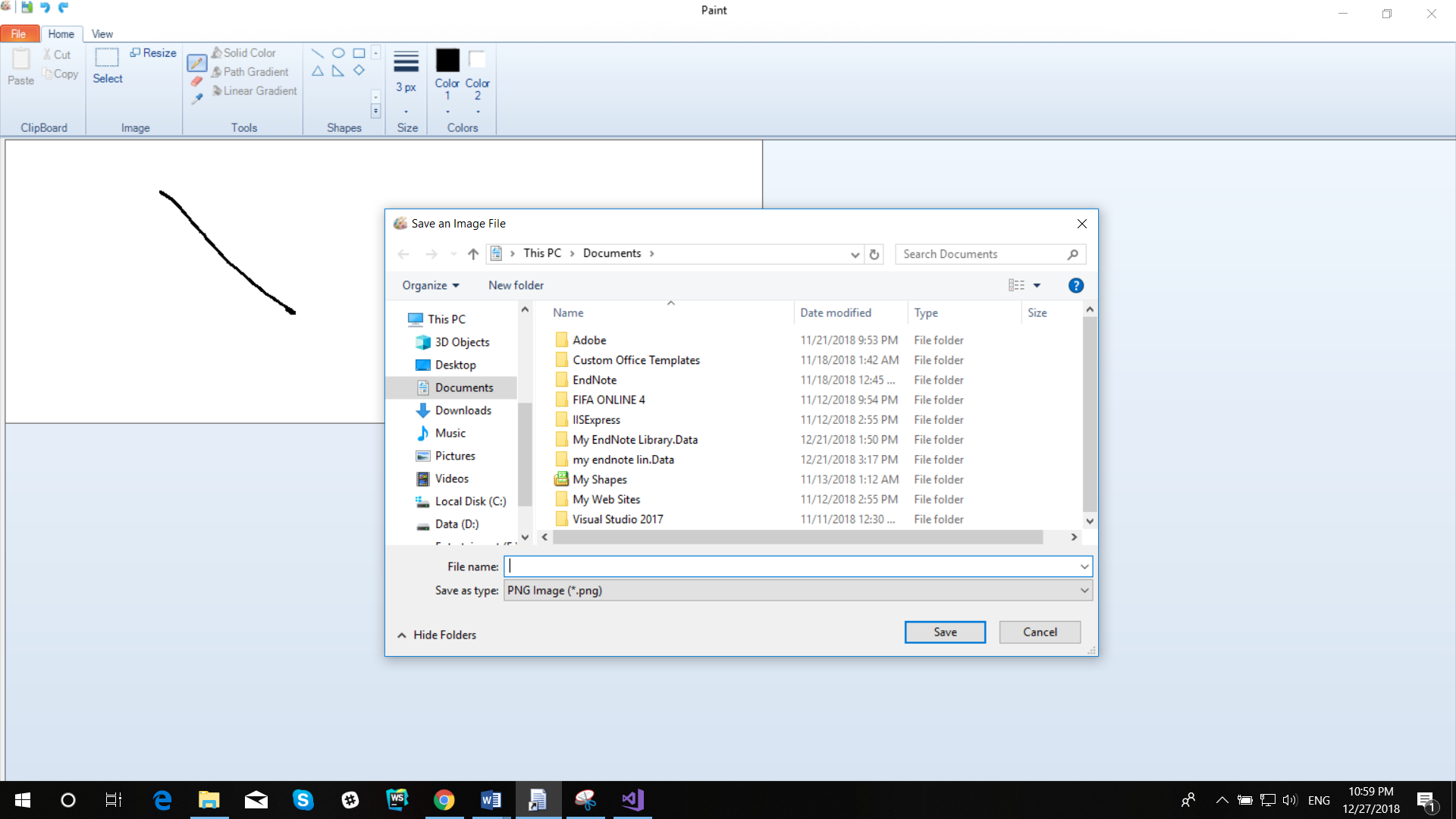
[III.7. Colors 21](#_Toc533725145)



Hình . GIao diện chính của ứng dụng.

1. **Thanh tiêu đề**
2. Save

* Ảnh mình họa: 
* Mô tả: khi click vào biểu tượng sẽ hiển thị cửa sổ cho phép người dùng lưu lại hình ảnh đã vẽ.



Hình . Giao diện lưu ảnh.

* Cách xử lý: mở SaveFileDialog, dựa vào dữ liệu dialog trả về để xử lý, thay đổi tên form và thêm file vào Recent File.
* Code xử lý:

private void saveFile()

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

saveFileDialog.Filter = "PNG Image|\*.png|JPeg Image|\*.jpg|Bitmap Image|\*.bmp";

saveFileDialog.Title = "Save an Image File";

saveFileDialog.ShowDialog();

if (saveFileDialog.FileName != "")

{

System.IO.FileStream fs = (System.IO.FileStream)saveFileDialog.OpenFile();

switch (saveFileDialog.FilterIndex)

{

case 1:

pictureBox.Image.Save(fs, System.Drawing.Imaging.ImageFormat.Png);

break;

case 2:

pictureBox.Image.Save(fs, System.Drawing.Imaging.ImageFormat.Bmp);

break;

case 3:

pictureBox.Image.Save(fs, System.Drawing.Imaging.ImageFormat.Bmp);

break;

}

changeFormText(getNameFromStream(fs));

addToRecentItem(fs);

fs.Close();

}

}

1. Undo

* Ảnh minh họa: 
* Mô tả: khi click vào biểu tượng sẽ vẽ lại hình trước đó một bước.
* Cách xử lý: lưu các hình ảnh vào một list, dựa vào đó để vẽ lại hình trước hình hiện tại.
* Code xử lý:

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

{

if (drawingList.currentIndex > 0)

{

drawing = new Bitmap(drawingList.ElementAt(--drawingList.currentIndex));

g = Graphics.FromImage(drawing);

pictureBox.Size = drawing.Size;

pictureBox.Image = drawing;

}

}

1. Redo

* Ảnh minh họa: 
* Mô tả: khi click vào biểu tượng sẽ vẽ lại hình sau đó một bước.
* Cách xử lý: lưu các hình ảnh vào một list, dựa vào đó để vẽ lại hình sau hình hiện tại.
* Code xử lý:

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

{

if (drawingList.currentIndex < drawingList.Count - 1)

{

drawing = new Bitmap(drawingList.ElementAt(++drawingList.currentIndex));

g = Graphics.FromImage(drawing);

pictureBox.Size = drawing.Size;

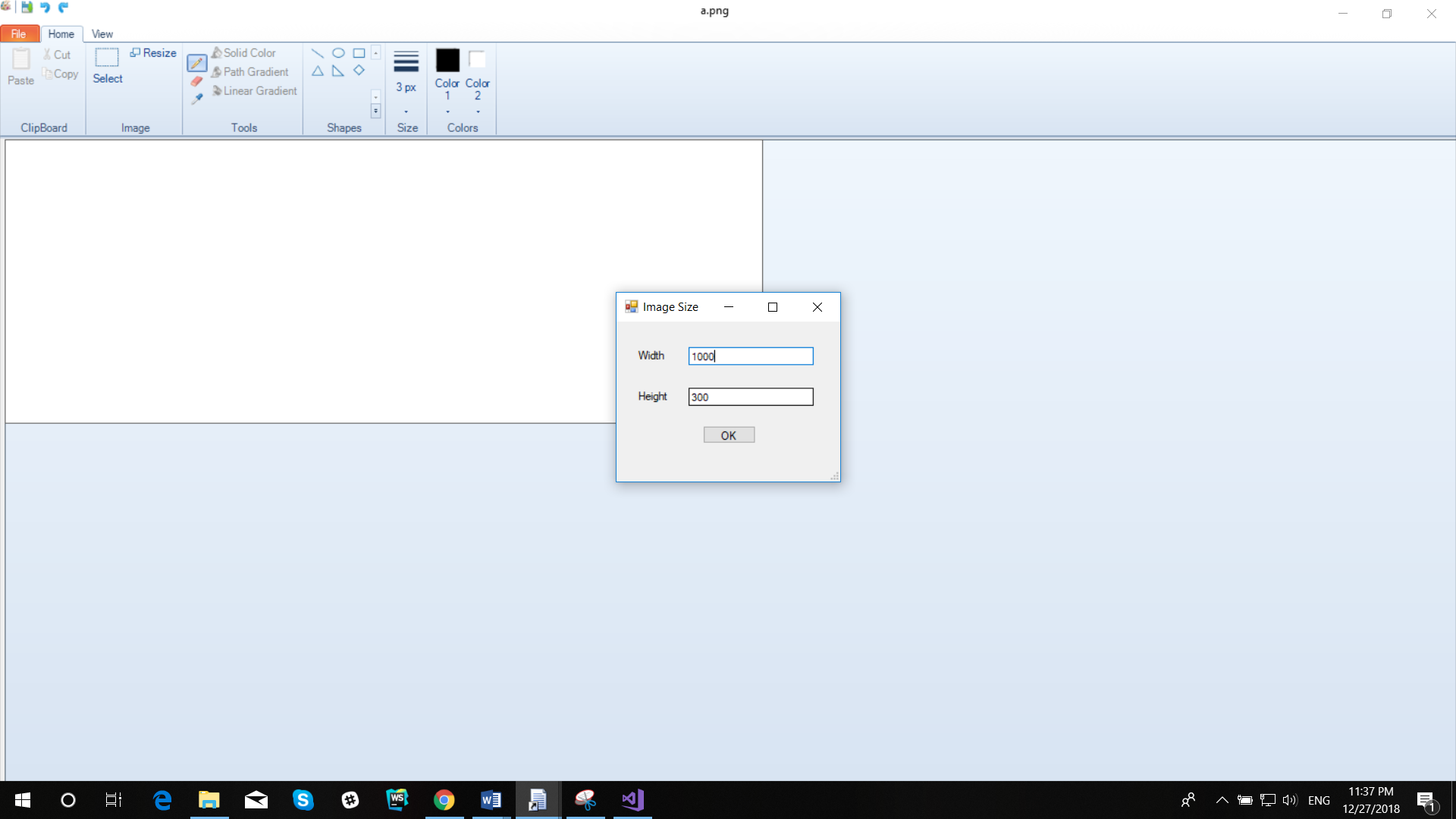
pictureBox.Image = drawing;

}

}

1. **File Menu**
2. New

* Ảnh minh họa: 
* Mô tả: khi click vào biểu tượng sẽ yêu cầu người dùng xác nhận có muốn lưu lại hình ảnh không. Nếu người dùng không hủy sẽ cho phép nhập chiều dài và chiều rộng của hình ảnh mới.



Hình . Giao diện khi tạo hình ảnh mới.

* Cách xử lý: khi click vào biểu tượng sẽ hiển thị MessageBox yêu cầu người dùng xác nhận có muốn lưu lại hình ảnh không:
* Nếu người dùng Cancel sẽ bỏ qua không thực hiện tiếp.
* Nếu người dùng chọn Yes sẽ lưu lại hình ảnh và cho phép người dùng nhập chiều dài và chiều rộng của hình ảnh mới.
* Nếu người dùng chọn No sẽ cho phép người dùng nhập chiều dài và chiều rộng của hình ảnh mới và không lưu lại hình ảnh cũ.
* Code xử lý:

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

{

FormClosingEventArgs ex = new FormClosingEventArgs(CloseReason.UserClosing, false);

closeUseDialog(ex);

if (ex.Cancel == false)

{

ImageSizeForm imageSizeForm = new ImageSizeForm(pictureBox.Width, pictureBox.Height);

imageSizeForm.StartPosition = FormStartPosition.CenterScreen;

if (imageSizeForm.ShowDialog(this) == DialogResult.OK)

{

pictureBox.Width = imageSizeForm.Width;

pictureBox.Height = imageSizeForm.Height;

}

drawing = new Bitmap(pictureBox.Width, pictureBox.Height);

g = Graphics.FromImage(drawing);

g.Clear(Color.White);

drawingList.AddBaseCapacity(drawing);

pictureBox.Image = drawing;

}

}

private void closeUseDialog(FormClosingEventArgs e)

{

DialogResult result = MessageBox.Show("Would you like to save your changes?", "Save",

MessageBoxButtons.YesNoCancel,

MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

saveFile();

}

else if (result == DialogResult.Cancel)

{

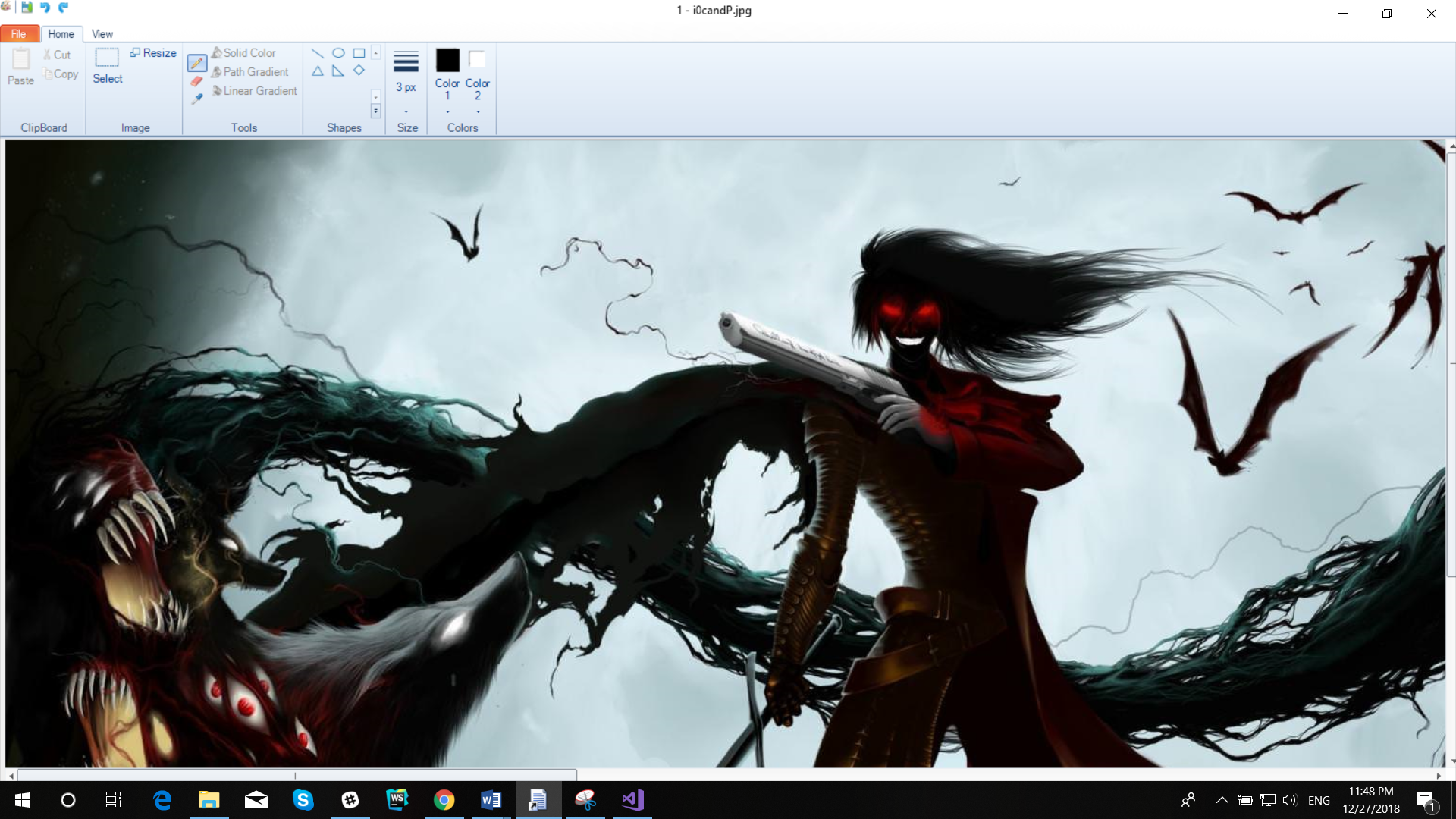
e.Cancel = true;

}

}

1. Open

* Ảnh minh họa: 
* Mô tả: khi click vào biểu tượng sẽ mở dialog cho phép người dùng chọn hình ảnh để mở ra ở Paint.



Hình . Giao diện khi mở ảnh.

* Cách xử lý: mở OpenFileDialog, dựa vào dữ liệu dialog trả về để xử lý
* Code xử lý:

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

{

System.IO.Stream myStream;

OpenFileDialog openFileDialog1 = new OpenFileDialog();

openFileDialog1.InitialDirectory = "D:\\";

openFileDialog1.Filter = "PNG files (\*.png)|\*.png|All files (\*.\*)|\*.\*";

openFileDialog1.FilterIndex = 2;

openFileDialog1.RestoreDirectory = true;

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

try

{

if ((myStream = openFileDialog1.OpenFile()) != null)

{

using (myStream)

{

drawing = new Bitmap(myStream);

drawingList.AddBaseCapacity(drawing);

pictureBox.Size = drawing.Size;

pictureBox.SizeMode = PictureBoxSizeMode.AutoSize;

pictureBox.Image = drawing;

drawing = new Bitmap(drawingList.Last());

g = Graphics.FromImage(drawing);

changeFormText(getNameFromStream(myStream));

addToRecentItem(myStream);

}

}

}

catch (Exception ex)

{

MessageBox.Show("Error: Could not read file from disk. Original error: " + ex.Message);

}

}

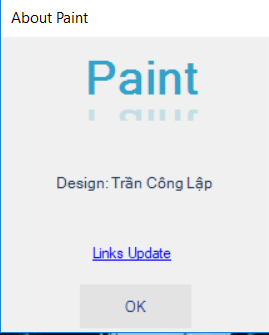
}

1. Save

* Ảnh mình họa: 
* Mô tả, cách xử lý và code xử lý giống như Save trên Thanh tiêu đề

1. About Paint

* Ảnh mình họa: 
* Mô tả: khi click vào biểu tượng sẽ hiển thị dialog thông tin ứng dụng.



Hình . Thông tin ứng dụng.

* Cách xử lý: mở InfoForm.
* Code xử lý:

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

{

InfoForm infoForm = new InfoForm();

infoForm.StartPosition = FormStartPosition.CenterScreen;

infoForm.Show(this);

}

1. Close

* Ảnh minh họa: 
* Mô tả: khi click vào biểu tượng sẽ yêu cầu người dùng xác nhận có muốn lưu lại hình ảnh không. Sau đó xử lý và đóng ứng dụng.
* Cách xử lý: Cách xử lý: khi click vào biểu tượng sẽ hiển thị MessageBox yêu cầu người dùng xác nhận có muốn lưu lại hình ảnh không:
* Nếu người dùng chọn Cancel trở lại giao diện chính.
* Nếu người dùng chọn Yes sẽ lưu lại hình ảnh và đóng form.
* Nếu người dùng chọn No sẽ đóng Form.
* Code xử lý:

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

{

FormClosingEventArgs ex = new FormClosingEventArgs(CloseReason.UserClosing, false);

closeUseDialog(ex);

if (ex.Cancel == false)

{

exit = true;

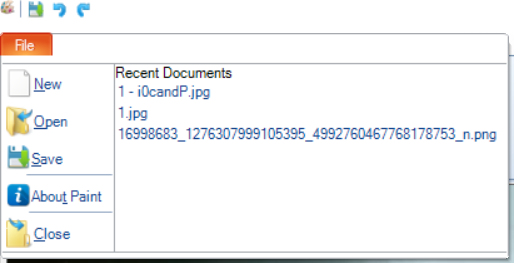
this.Close();

}

}

1. Recent Documents

* Ảnh mình họa:



Hình . Các hình ảnh gần đây.

* Mô tả: hiển thị các hình ảnh được lưu hoặc mở gần đây.
* Cách xử lý: cho phép lưu 3 hình ảnh gần đây, mỗi lần có hình ảnh mới sẽ thêm vào đầu danh sách và xóa hình ảnh cuối cùng.
* Code xử lý:

private void addToRecentItem(Stream stream)

{

FileStream fs = stream as FileStream;

if (recentItem2.Text != "")

{

recentItem3.Text = recentItem2.Text;

recentItem3.Tooltip = recentItem2.Tooltip;

recentItem3.Enabled = true;

}

if(recentItem1.Text!="")

{

recentItem2.Text = recentItem1.Text;

recentItem2.Tooltip = recentItem1.Tooltip;

recentItem2.Enabled = true;

}

recentItem1.Tooltip = fs.Name;

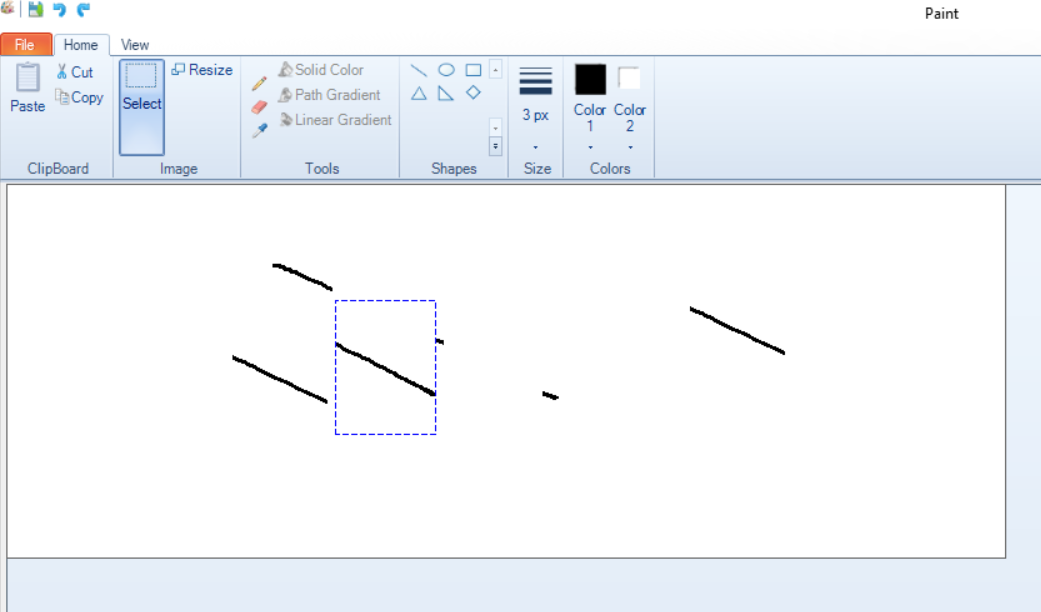
recentItem1.Text = getNameFromStream(stream);

recentItem1.Enabled = true;

}

1. **Thanh công cụ**
2. Image
3. Select

* Ảnh mình họa: 
* Mô tả: chọn một vùng để cut, copy hoặc dragdrop.



Hình . Chọn một vùng hình ảnh.

* Cách xử lý: lưu một Rectangle chứa toạ độ vùng chọn và 1 Bitmap chứa hình ảnh nằm trong vùng chọn đó. Nếu đã chọn được vùng chọn thì chuyển sang chế độ dragdrop.
* Code xử lý:

if (dragNdrop == false)

{

selectedArea = Rectangle1(x, y, x1, y1);

saveSelectedArea();

}

else

{

selectedArea.X += (x1 - x);

selectedArea.Y += (y1 - y);

dragNdrop = false;

drawingList.AddBaseCapacity(drawing);

drawing = new Bitmap(drawingList.Last());

g = Graphics.FromImage(drawing);

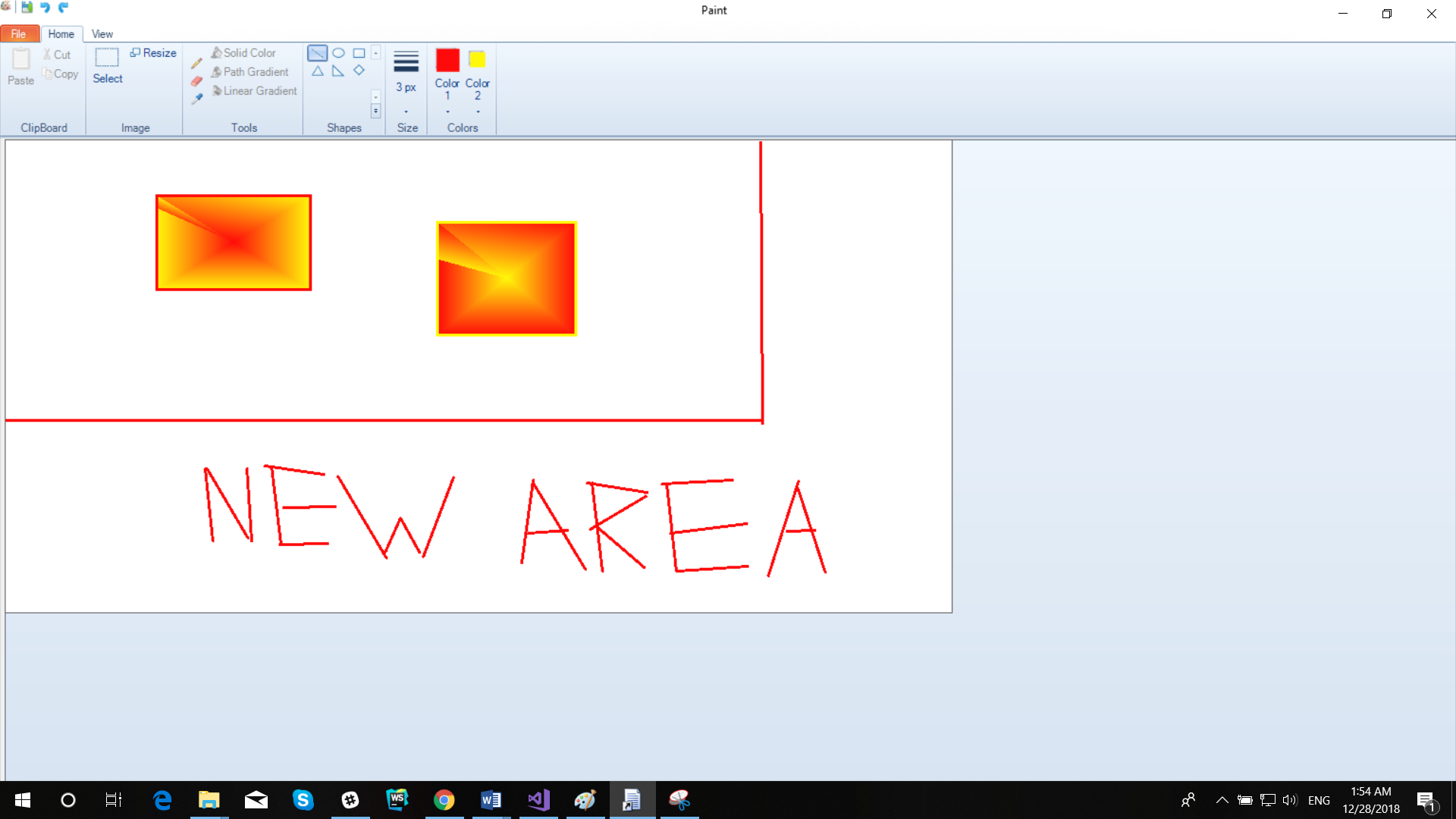
dashDraw(selectedArea);

pictureBox.Image = drawing;

}

1. Resize

* Ảnh mình họa: 
* Mô tả: điều chỉnh kích thước hình ảnh vẫn giữ lại phần ảnh đã vẽ.



Hình . Điều chỉnh kích thước ảnh.

* Cách xử lý: mở ImageSizeForm, nhận giá trị và thay đổi kích thước hình ảnh (vẫn giữ lại những nét vẽ đã có sẵn).
* Code xử lý:

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

{

ImageSizeForm imageSizeForm = new ImageSizeForm(pictureBox.Width, pictureBox.Height);

imageSizeForm.StartPosition = FormStartPosition.CenterScreen;

if (imageSizeForm.ShowDialog(this) == DialogResult.OK)

{

Bitmap drawingClone = (Bitmap)drawing.Clone();

drawing = new Bitmap(imageSizeForm.Width, imageSizeForm.Height);

g = Graphics.FromImage(drawing);

g.Clear(Color.White);

using (Graphics g = Graphics.FromImage((Image)drawing))

{

g.InterpolationMode = InterpolationMode.HighQualityBicubic;

g.DrawImageUnscaled(drawingClone, 0, 0, imageSizeForm.Width, imageSizeForm.Height);

}

drawingList.AddBaseCapacity(drawing);

pictureBox.Width = imageSizeForm.Width;

pictureBox.Height = imageSizeForm.Height;

pictureBox.Image = drawing;

}

}

1. Clipboard
2. Cut

* Ảnh minh họa: 
* Mô tả: xóa phần được chọn và sẵn sàng để dán vào nơi khác.
* Cách xử lý: xoá hình vẽ trong select hiện tại, lưu vào clipboard hình vẽ đó và Enable button Paste.
* Code xử lý:

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

{

Bitmap nullImage = new Bitmap(selectImage);

Graphics nullGraphics = Graphics.FromImage(nullImage);

nullGraphics.Clear(Color.White);

drawing = new Bitmap(drawingList.ElementAt(drawingList.currentIndex));

g = Graphics.FromImage(drawing);

g.DrawImageUnscaled(nullImage, selectedArea);

Console.WriteLine(drawingList.currentIndex);

pictureBox.Image = drawing;

drawingList.AddBaseCapacity(drawing);

pasteItem.Enabled = true;

}

1. Copy

* Ảnh minh họa: 
* Mô tả: sao chép phần được chọn và sẵn sàng để dán vào nơi khác
* Cách xử lý: lưu vào clipboard hình vẽ trong select hiện tại (đã lưu lúc select) và Enable button Paste.
* Code xử lý:

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

{

pasteItem.Enabled = true;

}

1. Paste

* Ảnh mình họa: 
* Mô tả: vẽ hình đã được cut hoặc copy trước đó vào góc trái hình ảnh và cho phép dragdrop.
* Cách xử lý: vẽ hình trong clipboard ra vị trí 0,0 và cho phép dragdrop.
* Code xử lý:

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

{

g.DrawImageUnscaled(selectImage, 0, 0);

drawingList.AddBaseCapacity(drawing);

drawing = new Bitmap(drawingList.Last());

g = Graphics.FromImage(drawing);

selectedArea = Rectangle1(0, 0, selectImage.Width, selectImage.Height);

dashDraw(selectedArea);

pictureBox.Image = drawing;

}

1. Tools
2. Pencil

* Ảnh minh họa: 
* Mô tả: chọn bút để vẽ.
* Cách xử lý: bắt sự kiện mousemove và vẽ.
* Code xử lý:

case item.Pen:

{

pointList.Add(new Point(x1, y1));

try

{

g.DrawLines(pen, pointList.ToArray());

}

catch (Exception ex)

{}

break;

}

1. Eraser

* Ảnh minh họa: 
* Mô tả: xóa những phần không cần thiết của hình ảnh.
* Cách xử lý: vẽ các hình chữ nhật màu trắng.
* Code xử lý:

case item.Eraser:

{

SolidBrush brush = new SolidBrush(Color.White);

rectangleList.Add(new Rectangle(new Point(x1, y1), new Size(currentWidth, currentWidth)));

g.FillRectangles(brush, rectangleList.ToArray());

break;

}

1. Colorpicker

* Ảnh minh họa: 
* Mô tả: chọn màu khi click vào hình ảnh.
* Cách xử lý: chọn màu ở điểm click chuột.
* Code xử lý:

if(currentItem==item.ColorPicker)

{

Color colorpicker = drawing.GetPixel(e.X, e.Y);

if (e.Button == System.Windows.Forms.MouseButtons.Left)

colorPickerDropDown1.SelectedColor = currentColor = colorpicker;

else

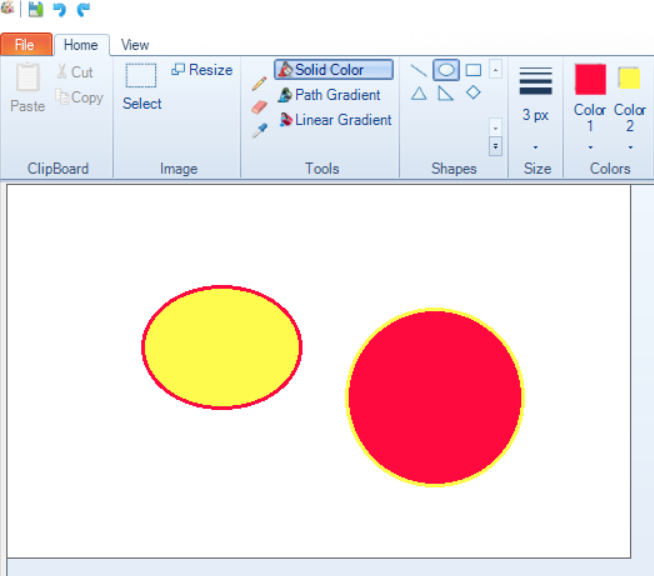
colorPickerDropDown1.SelectedColor = currentColor2 = colorpicker;

goto jm;

}

1. Fill tools
2. Solid Color

* Ảnh minh họa: 
* Mô tả: tô màu theo kiểu Solid Color, tô ngược lại nếu dùng chuột phải.



Hình . Tô màu theo kiểu Solid Color.

* Cách xử lý: tạo brush và vẽ.
* Code xử lý:

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

{

Brush brush;

uncheckedAllFill();

if (currentFillItem == fillType.Solid)

{

currentFillItem = fillType.NoFill;

brush = new SolidBrush(Color.White);

}

else

{

fillSolidItem.Checked = true;

currentFillItem = fillType.Solid;

Color primaryColor, subColor;

if (leftMouseButton == true)

{

primaryColor = currentColor2;

subColor = currentColor;

}

else

{

primaryColor = currentColor;

subColor = currentColor2;

}

brush = new SolidBrush(primaryColor);

}

if (\_x == 0 && \_x1 == 0 && \_y == 0 && \_y1 == 0)

return;

brushShape(brush, \_x, \_y, \_x1, \_y1);

pictureBox.Image = drawing;

drawingList.AddBaseCapacity(drawing);

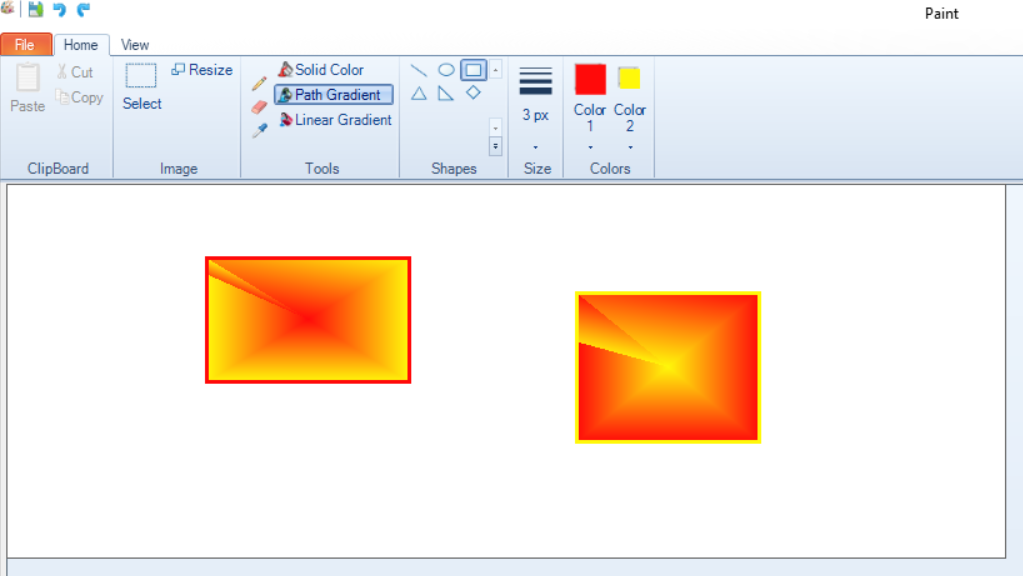
drawing = new Bitmap(drawingList.Last());

g = Graphics.FromImage(drawing);

}

1. Path Gradient

* Ảnh minh họa: 
* Mô tả: tô màu theo kiểu Path Gradient, tô ngược lại nếu dùng chuột phải.



Hình . Tô màu theo kiểu Path Gradient.

* Cách xử lý: tạo brush và vẽ.
* Code xử lý:

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

{

Brush brush;

uncheckedAllFill();

if (currentFillItem == fillType.PathGradient)

{

currentFillItem = fillType.NoFill;

brush = new SolidBrush(Color.White);

}

else

{

fillPathItem.Checked = true;

currentFillItem = fillType.PathGradient;

Color primaryColor, subColor;

if (leftMouseButton == true)

{

primaryColor = currentColor2;

subColor = currentColor;

}

else

{

primaryColor = currentColor;

subColor = currentColor2;

}

Point[] pList = new Point[5];

pList[0] = new Point(x, y);

pList[1] = new Point(x, y1);

pList[2] = new Point(x1, y1);

pList[3] = new Point(x1, y);

PathGradientBrush pthGrBrush = new PathGradientBrush(pList);

pthGrBrush.CenterPoint = new PointF((x + x1) / 2, (y + y1) / 2);

pthGrBrush.CenterColor = subColor;

Color[] colors = { primaryColor };

pthGrBrush.SurroundColors = colors;

brush = pthGrBrush;

}

if (\_x == 0 && \_x1 == 0 && \_y == 0 && \_y1 == 0)

return;

brushShape(brush, \_x, \_y, \_x1, \_y1);

pictureBox.Image = drawing;

drawingList.AddBaseCapacity(drawing);

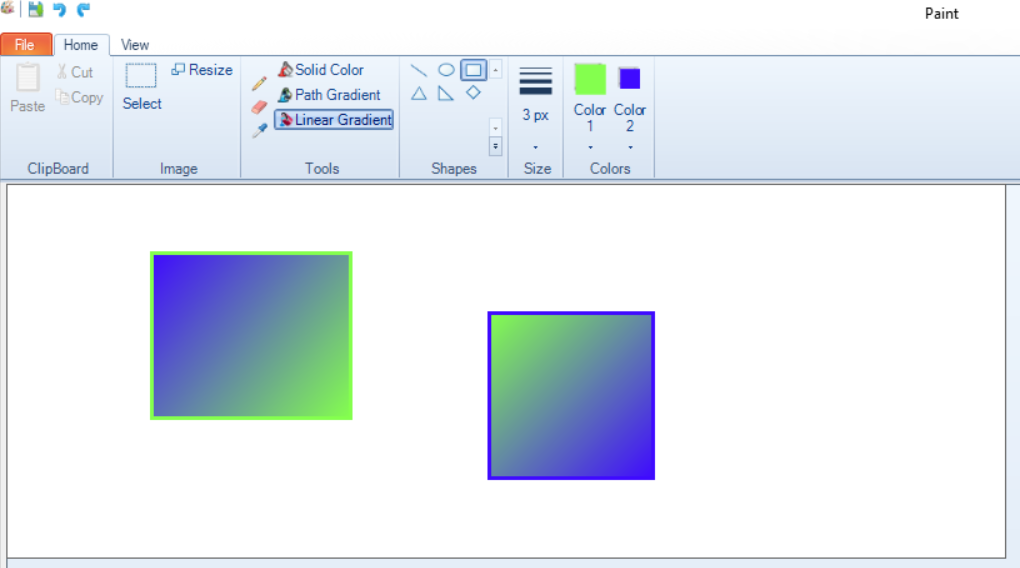
drawing = new Bitmap(drawingList.Last());

g = Graphics.FromImage(drawing);

}

1. Linear Gradient

* Ảnh minh họa: 
* Mô tả: tô màu theo kiểu Linear Gradient, tô ngược lại nếu dùng chuột phải.



Hình . Tô màu theo kiểu Linear Gradient.

* Cách xử lý: Tạo brush và vẽ.
* Code xử lý:

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

{

Brush brush;

uncheckedAllFill();

if (currentFillItem == fillType.LinearGradient)

{

currentFillItem = fillType.NoFill;

brush = new SolidBrush(Color.White);

}

else

{

fillLinearItem.Checked = true;

currentFillItem = fillType.LinearGradient;

Color primaryColor, subColor;

if (leftMouseButton == true)

{

primaryColor = currentColor2;

subColor = currentColor;

}

else

{

primaryColor = currentColor;

subColor = currentColor2;

}

brush = new LinearGradientBrush(new Point(x, y),new Point(x1, y1),currentColor, currentColor2);

}

if (\_x == 0 && \_x1 == 0 && \_y == 0 && \_y1 == 0)

return;

brushShape(brush, \_x, \_y, \_x1, \_y1);

pictureBox.Image = drawing;

drawingList.AddBaseCapacity(drawing);

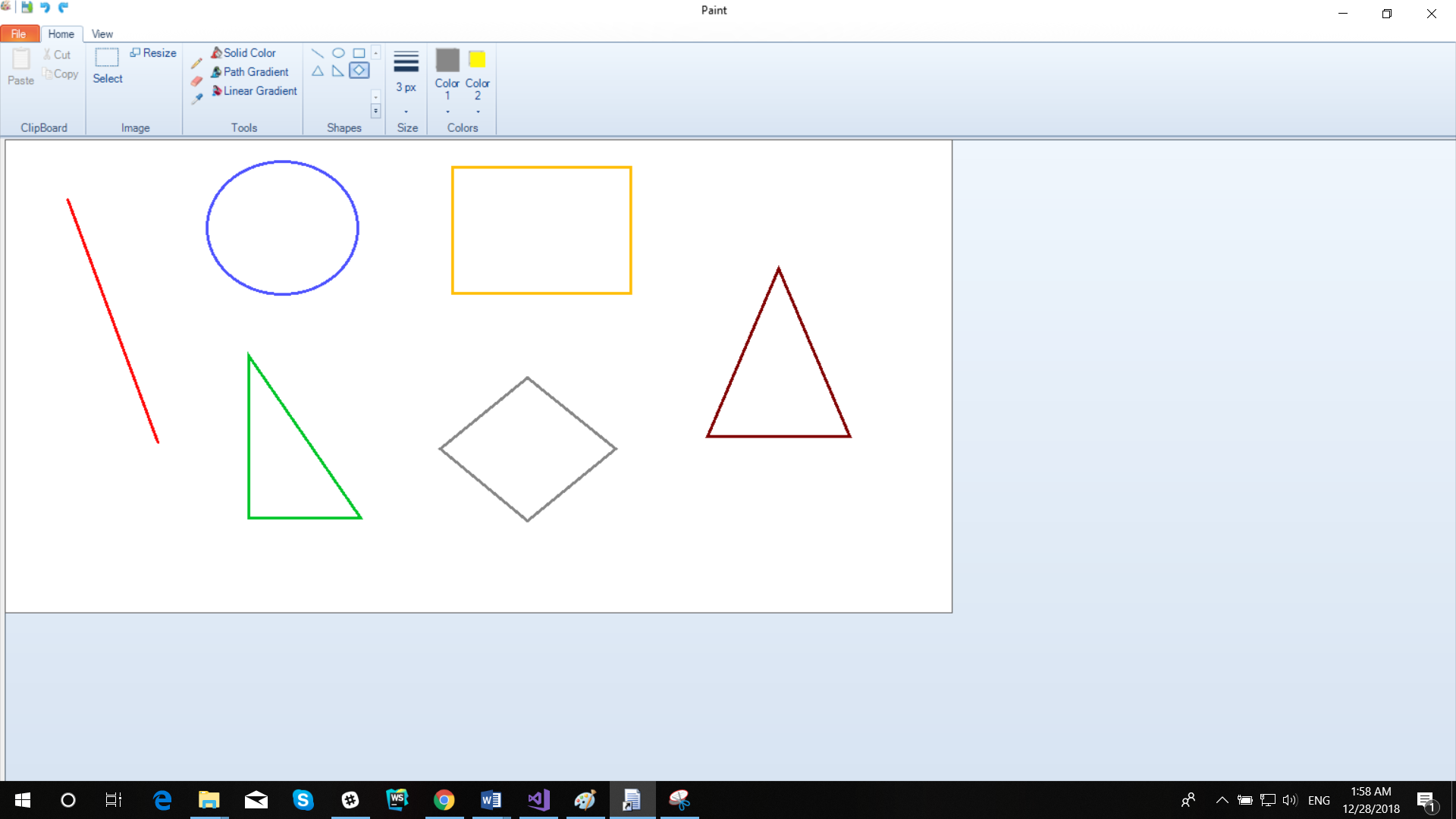
drawing = new Bitmap(drawingList.Last());

g = Graphics.FromImage(drawing);

}

1. Shapes

* Ảnh minh họa: 
* Mô tả: vẽ các hình tương ứng.



Hình . Các shape trong ứng dụng.

* Cách xử lý: dùng các hàm DrawLine, DrawEllipse, DrawRectangle,…
* Code xử lý:

case item.Line:

{

g.DrawLine(pen, new Point(x, y), new Point(x1, y1));

break;

}

case item.Oval:

{

g.DrawEllipse(pen, x, y, x1 - x, y1 - y);

break;

}

case item.Rectangle:

{

g.DrawRectangle(pen,Rectangle1(x,y,x1,y1));

break;

}

case item.TriCan:

{

Point[] pList = new Point[3];

pList[0]=new Point((x+x1)/2,y);

pList[1]=new Point(x,y1);

pList[2]=new Point(x1,y1);

g.DrawPolygon(pen,pList);

break;

}

case item.TriVuong:

{

Point[] pList = new Point[3];

pList[0] = new Point(x,y);

pList[1] = new Point(x, y1);

pList[2] = new Point(x1, y1);

g.DrawPolygon(pen, pList);

break;

}

case item.Diamond:

{

Point[] pList = new Point[4];

pList[0] = new Point((x + x1) / 2, y);

pList[1] = new Point(x, (y + y1) / 2);

pList[2] = new Point((x + x1) / 2, y1);

pList[3] = new Point(x1, (y + y1) / 2);

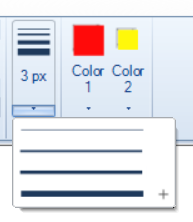
g.DrawPolygon(pen, pList);

break;

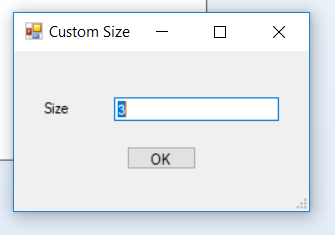
}

1. Size

* Ảnh minh họa:



* Mô tả: chọn hoặc nhập kích thước cho nét vẽ.



Hình . Điều chỉnh kích thước cho nét vẽ.

* Cách xử lý: set giá trị hoặc mở OneTextBoxForm khi click vào item tương ứng.
* Code xử lý:

private void sizeItem1\_Click(object sender, EventArgs e)

{

currentWidth = 1;

sizeItem.Text = "1 px";

}

private void sizeItem2\_Click(object sender, EventArgs e)

{

currentWidth = 3;

sizeItem.Text = "3 px";

}

private void sizeItem3\_Click(object sender, EventArgs e)

{

currentWidth = 5;

sizeItem.Text = "5 px";

}

private void sizeItem4\_Click(object sender, EventArgs e)

{

currentWidth = 8;

sizeItem.Text = "8 px";

}

private void sizeItem5\_Click(object sender, EventArgs e)

{

OneTextBoxForm oneTextBoxForm = new OneTextBoxForm("Custom Size", "Size", currentWidth);

oneTextBoxForm.StartPosition = FormStartPosition.CenterScreen;

if (oneTextBoxForm.ShowDialog(this) == DialogResult.OK)

{

currentWidth = oneTextBoxForm.Value;

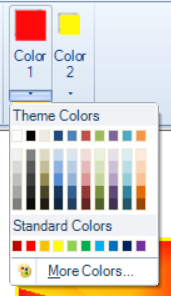
sizeItem.Text = oneTextBoxForm.Value + " px";

}

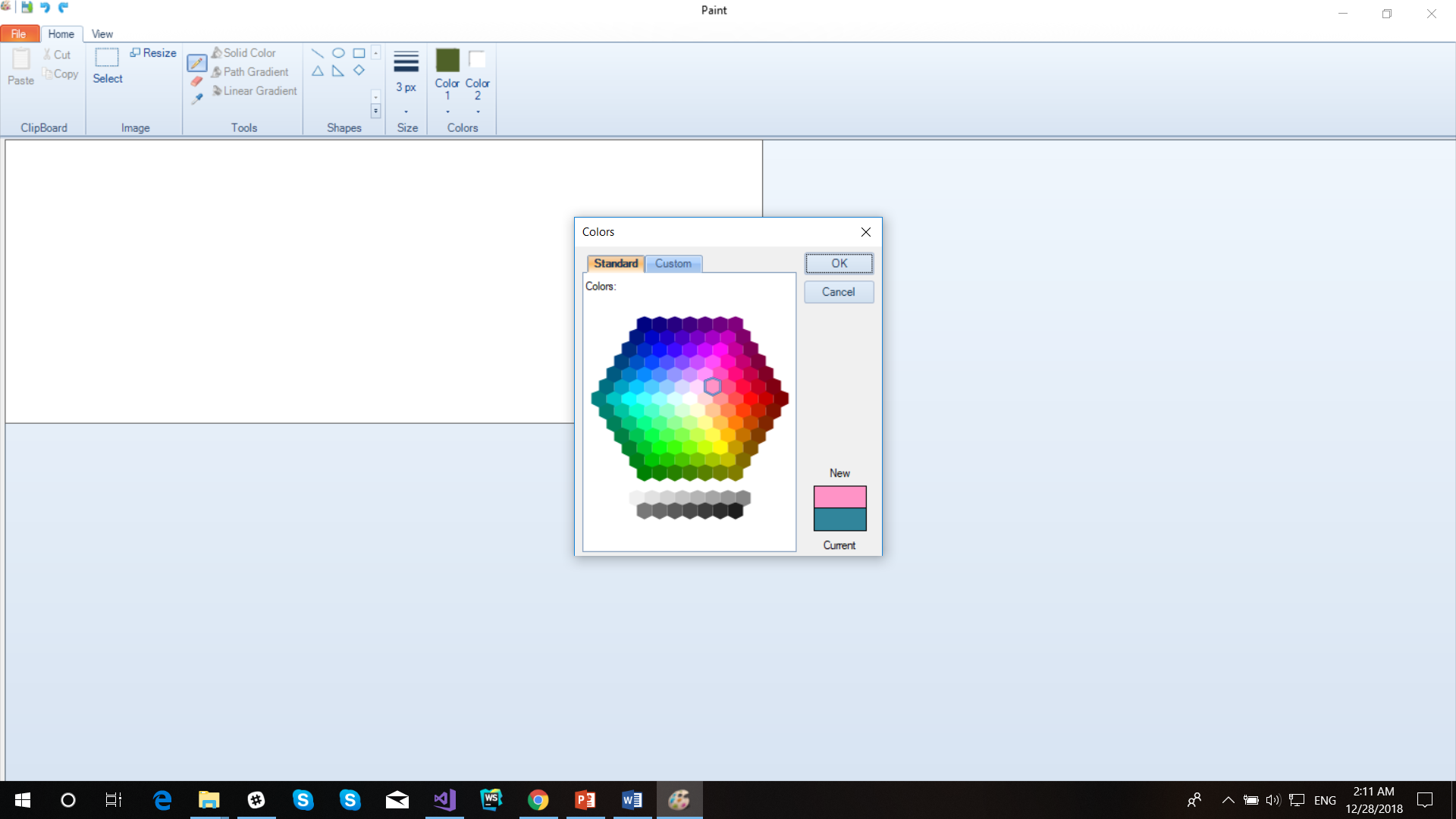
}

1. Colors

* Ảnh minh họa:



* Mô tả: chọn màu cho Color 1 và Color 2.



Hình . Chọn màu.

* Cách xử lý: mở DisplayMoreColorsDialog và xử lý khi nhận kết quả.
* Code xử lý:

private void colorPickerDropDown1\_Click(object sender, System.EventArgs e)

{

colorPickerDropDown1.DisplayMoreColorsDialog();

}

private void colorPickerDropDown2\_Click(object sender, System.EventArgs e)

{

colorPickerDropDown2.DisplayMoreColorsDialog();

}

private void colorPickerDropDown1\_SelectedColorChanged(object sender, EventArgs e)

{

currentColor = colorPickerDropDown1.SelectedColor;

}

private void colorPickerDropDown2\_SelectedColorChanged(object sender, EventArgs e)

{

currentColor2 = colorPickerDropDown2.SelectedColor;

}