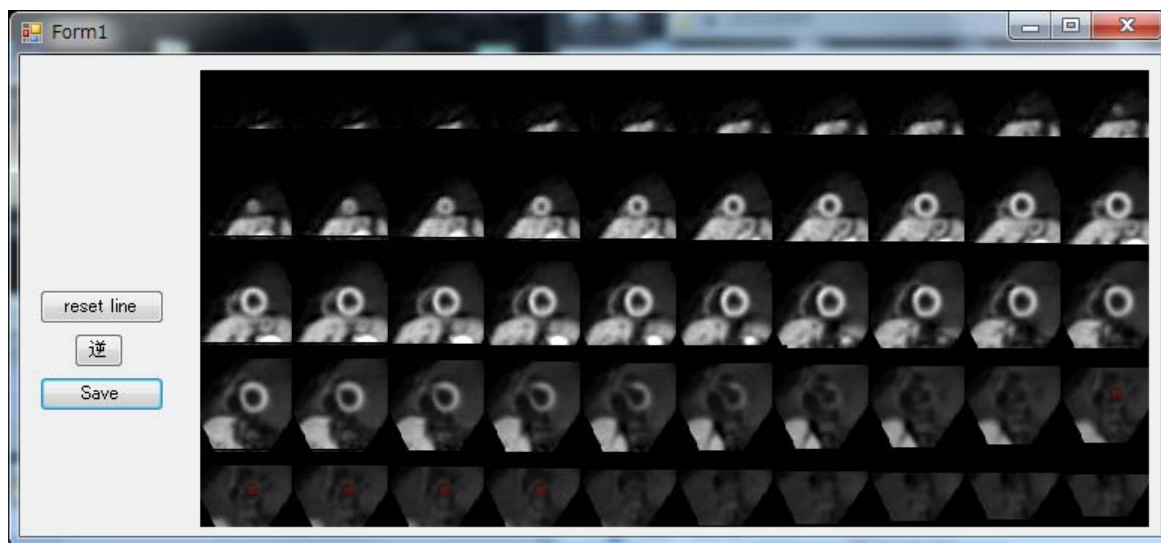


久々 Sample

今年の末くらいにココを消してもらおうと思っているので
Sample になりそうな Program を作成したら
特に推敲しないで随時アップしていこうと思う。

盗めるところをコピーで盗むのが賢いやり方。

今回作ったプログラム



トリミングツール。
jpegやbitmap画像を
D&Dで読み込んで、
クリックで囲われた
領域のみを採用し
jpegで保存する。

プログラムの流れ

- ① D&D(drag & drop)で画像を読み込んで表示
- ② クリックで領域設定
- ③ 保存

① D&D(drag & drop)で画像を読み込んで表示

```
private void Form1_DragEnter(object sender, DragEventArgs e)
{
    if (e.Data.GetDataPresent(DataFormats.FileDrop)) e.Effect = DragDropEffects.Copy;
    else e.Effect = DragDropEffects.None;
}

private void Form1_DragDrop(object sender, DragEventArgs e)
{
    try
    {
        fileName = (string[])e.Data.GetData(DataFormats.FileDrop, false);
        FileStream sr = new FileStream(fileName[0], FileMode.Open, FileAccess.Read);
        Bitmap bmp = (Bitmap)System.Drawing.Bitmap.FromStream(sr);
        sr.Close();
        pictureBox1.Image = new Bitmap(bmp);
        pictureBox1.Width = bmp.Size.Width;
        pictureBox1.Height = bmp.Size.Height;
        this.SetBounds(0, 0, 400 - 256 + pictureBox1.Width, 310 - 256 + pictureBox1.Height, BoundsSpecified.Size);
        img = new int[bmp.Size.Width*3, bmp.Size.Height];
        |
        mx = new int[100]; my = new int[100]; mflg = 0;
        for (int j = 0; j < bmp.Size.Height; j++)
        {
            for (int i = 0; i < bmp.Size.Width; i++)
            {
                img[i * 3    , j] = bmp.GetPixel(i, j).R;
                img[i * 3 + 1, j] = bmp.GetPixel(i, j).G;
                img[i * 3 + 2, j] = bmp.GetPixel(i, j).B;
            }
        }
        bmp.Dispose();
    }
    catch { }
}
```

Formのイベントハンドラ(雷マーク)で上の二つを追加
また、プロパティのallowdropをtrueにしておく

① D&D(drag & drop)で画像を読み込んで表示

DragEnterでファイルのドロップを許可している。

これをかかないとドロップを受け付けない(はず)。

よくわからないがとりあえず書いておく。

DragDropでドロップされたファイルのパス(場所)を読んでいる。

FileStreamで画像を読み、pictureBox1に表示。

WidthやHeightは読み込んだ画像のもの。

それに合わせてpictureBoxやForm(SetBounds)の大きさを調整している。

① D&D(drag & drop)で画像を読み込んで表示

mx, my, mflgはクリックを受け付ける準備。

最後に、やはり画像情報は配列でほしいのでbitmapを配列に入れる。
普通の画像はカラーであるから、RGBそれぞれを別個に格納していく。

disposeは「もういらぬ」の意味。特に影響ないので気にしなくてよい。

変数をどこで宣言しているのかは各自確認してください。

② クリックで領域設定

```
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    if (img != null)
    {
        if ((Control.MouseButtons & MouseButtons.Left) == MouseButtons.Left)
        {
            mx[mflg] = e.X; my[mflg] = e.Y;
            var g = pictureBox1.CreateGraphics();
            g.DrawLine(Pens.Yellow, e.X - 2, e.Y, e.X + 2, e.Y);
            g.DrawLine(Pens.Yellow, e.X, e.Y - 2, e.X, e.Y + 2);
            g.Dispose();
            mflg++;
        }
        if ((Control.MouseButtons & MouseButtons.Right) == MouseButtons.Right)
        {
            mx[mflg] = e.X; my[mflg] = e.Y;
            mflg++;
            mx[mflg] = mx[0]; my[mflg] = my[0];
            line = new int[pictureBox1.Width, pictureBox1.Height];
            for (int i = 0; i < mflg; i++)
            {
                D.Line(mx[i], my[i], mx[i + 1], my[i + 1], pictureBox1, ref line);
            }
            for (int j = 1; j < pictureBox1.Height - 1; j++)
            {
                for (int i = 1; i < pictureBox1.Width - 1; i++)
                {
                    if (line[i, j] != 1)
                    {
                        label = new int[pictureBox1.Width, pictureBox1.Height];
                        label[i, j] = 1; break;
                    }
                }
                if (label != null) break;
            }
            while (true)
            {
                int counter = 0;
                for (int j = 1; j < pictureBox1.Height - 1; j++)
                {
                    for (int i = 1; i < pictureBox1.Width - 1; i++)
                    {
                        if (label[i, j] == 1)
                        {
                            if (line[i + 1, j] == 0 && label[i + 1, j] == 0) { label[i + 1, j] = 1; counter++; }
                            if (line[i - 1, j] == 0 && label[i - 1, j] == 0) { label[i - 1, j] = 1; counter++; }
                            if (line[i, j + 1] == 0 && label[i, j + 1] == 0) { label[i, j + 1] = 1; counter++; }
                            if (line[i, j - 1] == 0 && label[i, j - 1] == 0) { label[i, j - 1] = 1; counter++; }
                        }
                    }
                }
                if (counter == 0) break;
            }
            D.Disp(img, label, pictureBox1);
        }
    }
    else
    {
        MessageBox.Show("画像未読\nD&Dする");
    }
}
```

文字にするとややこしいことになりそうなので細かい部分は割愛します。

ポイントはif文。

ラベリングは勉強済み。

② クリックで領域設定

最初のif+elseで、画像がD&Dされてるかを確認している。

している場合はクリックを受け付け、していない場合は受け付けない。

elseに入った場合、メッセージボックスを表示させる。

また、右クリックと左クリックを区別するためのif文が中にある。

右クリックされたときに、領域設定を終了する作りになっている。

あとは解読してください。「reset」、「逆」ボタンもいじったりコード見たりすればわかってもらえるはず。

③ 保存

```
private void button1_Click(object sender, EventArgs e)//save
{
    if (img != null)
    {
        string[] name = fileName[0].Split('.');
        pictureBox1.Image.Save(name[0] + "_trim.jpg", ImageFormat.Jpeg);
        MessageBox.Show("同フォルダに保存しました");
    }
    else
    {
        MessageBox.Show("画像未読%nD&Dすればよい");
    }
}
```

Splitは文字列を指定した文字で分割するもの。

ここでは'.'を指定しているので「image.jpg」→「image」「jpg」のように分割する。

おわり