

RP3 PROJEKT

Picture Tool

Renato Babojelić, Krešimir Sinković

15. veljače 2017.

1 Uvod

Alat Picture Tool je razvijen kao završni projekt na kolegiju Računarski praktikum 3. Radi se o editoru rasterske grafike, svojevrsnom klonu popularnog Windows editora Paint. Program podržava stvaranje novih ili otvaranje već postojećih grafičkih datoteka, crtanje linija, pravaca, te kružnica, elipsi i pravokutnika. Dijelove slike je moguće označiti te izrezati, kopirati i zalijepiti na drugo mjesto, podržana je i operacija *crop*, odnosno izrezivanje slike na dio koji je selektiran. Nadalje, implementirane su operacije rotiranja, te osne simetrije s obzirom na horizontalnu ili verikalnu os. Na sliku je moguće primijeniti i neki filter, ugrađeni su standardni *grayscale*, *negative* i *sepia* efekti. U program je ugrađena i *undo-redo* funkcionalnost, tako da je svaku od navedenih operacija moguće poništiti ili ponovno primijeniti. U PictureTool ugrađen je i alat za izradu mozaika i panorama.

Projekt je napisan u programskom jeziku C#, za prikaz grafičkog sučelja koriste se *Windows Forms*, dok su prikaz grafike korištene klase iz *System.Drawing* imenika .NET frameworka.

2 Kako koristiti Picture Tool

Pokretanjem programa otvara se glavni prozor. Iz padajućeg izbornika odabiremo stvaranje nove datoteke ili otvaranje već postojeće. Prilikom stvaranja nove datoteke odabiremo visinu i širinu slike u pikselima.

Sada je moguće crtati i manipulirati slikom s nekim od sljedećih alata.

2.1 Alatna traka



Stvara novu sliku bijele pozadine. Otvara dijaloški okvir u kojem se unosi visina i širina slike.



Otvara dijaloški okvir u kojem se odabire slika koju želimo otvoriti.



Sprema sve promjene na slici u postojeću datoteku.



Sprema sve promjene na slici u novu datoteku. Otvara se dijaloški okvir za odabir nove datoteke.



Izrezuje označeni dio slike i sprema ga u *Clipboard*.



Kopira označeni dio slike i sprema ga u *Clipboard*.



Ljepi sadržaj *Clipboarda* na sliku. Zaljepljeni dio je moguće pomicati mišem na željenu lokaciju.



Izrezuje sve osim označenog dijela slike i smanjuje sliku na dimenzije označenog dijela.



Poništava zadnju promjenu na slici.



Vraća zadnju poništenu promjenu na slici.



Rotira sliku za 90° u pozitivnom smjeru.



Rotira sliku za 90° u negativnom smjeru.



Zrcali sliku s obzirom na vertikalnu os.



Zrcali sliku s obzirom na horizontalnu os.

2.2 Traka s alatima za crtanje



Alat za selektiranje. Na dio slike označem ovim pravokutnikom je moguće primijeniti operacije *Cut*, *Copy* i *Crop*.



Alat za prostoručno crtanje linija. Moguće je birati boju, debljinu i vrstu linije.



Alat za crtanje ravnih linija. Moguće je birati boju, debljinu i vrstu linije.



Alat za crtanje kružnica. Moguće je birati boju, debljinu i vrstu linije te ispunu.



Alat za crtanje elipsi. Moguće je birati boju, debljinu i vrstu linije te ispunu.



Alat za crtanje pravokutnika. Moguće je birati boju, debljinu i vrstu linije te ispunu.

Za svaki od ovih alata boju biramo klikom na ikonice pored natpisa *Primary* odnosno *Secondary*. Boja *Primary* se koristi ako crtamo klikom na lijevu tipku miša, dok klikom na desnu tipku koristimo *Secondary* boju. U padajućem izborniku *Width* biramo debljinu linije. Izbornikom *Fill* biramo ispunu (bez, puna, ili crtkana) za kružnicu, elipsu i pravokutnik, dok izbornikom *Line* odabiremo vrstu linije (puna, ili neka varijanta crtkane).

2.3 Izbornik Tools

Uz već navedene alate, u izborniku *Tools* nalaze se jos i filteri:

- *Grayscale* – pretvara sliku u crno-bijelu;
- *Negative* – pretvara sliku u njen negativ;
- *Sepia* – primjenjuje istoimeni filter na sliku.

Klikom na *Mosaic...* otvara se novi prozor u kojem je moguće izraditi mozaik. Veličinu gotovog mozaika biramo kroz izbornik *File/New...*. Slike biramo klikom na izbornik *File/Add...* te se one dodaju u lijevi dio prozora, nakon čega ih je moguće uključiti ili isključiti iz mozaika klikom na kvačicu pored imena. Kroz izbornik *Edit/Pattern...* odabiremo koliko će slika biti horizontalno, odnosno vertikalno. Na posljetku sliku spremamo odabirom *File/Save...*

3 Implementacija

3.1 Alati

Za iscrtavanje objekata mišem na slici brinu se sljedeće funkcije.

```
private void drawSelectionRectangle(PaintEventArgs e)
{
    selectedArea = new Rectangle(
        Math.Min(currentStroke.First().X,
            currentStroke.Last().X),
        Math.Min(currentStroke.First().Y,
            currentStroke.Last().Y),
        Math.Abs(currentStroke.First().X -
            currentStroke.Last().X),
```

```

        Math.Abs(currentStroke.First().Y -
            currentStroke.Last().Y));
var p = new Pen(Color.DarkGray, 1);
float[] dashes = { 5, 2 };
p.DashPattern = dashes;
e.Graphics.DrawRectangle(p, selectedArea);
}
private void drawRectangle(PaintEventArgs e)
{
    var rec = new Rectangle(
        Math.Min(currentStroke.First().X,
            currentStroke.Last().X),
        Math.Min(currentStroke.First().Y,
            currentStroke.Last().Y),
        Math.Abs(currentStroke.First().X -
            currentStroke.Last().X),
        Math.Abs(currentStroke.First().Y -
            currentStroke.Last().Y));
    var p = new Pen(drawingColor, width);
    p.DashStyle = line;
    e.Graphics.DrawRectangle(p, rec);
    e.Graphics.FillRectangle(selectedBrush, rec);
}
private void drawLine(PaintEventArgs e)
{
    var p = new Pen(drawingColor, width);
    p.DashStyle = line;
    e.Graphics.DrawLine(
        p,
        currentStroke.First(),
        currentStroke.Last());
}
private void drawPencil(PaintEventArgs e)
{
    var p = new Pen(drawingColor, width);
    p.DashStyle = line;
    e.Graphics.DrawLines(
        p,
        currentStroke.ToArray());
}
private void drawCircle(PaintEventArgs e)
{
    int a = Math.Min(
        Math.Abs(currentStroke.First().X -

```

```

        currentStroke.Last().X),
        Math.Abs(currentStroke.First().Y -
            currentStroke.Last().Y));
var rec = new Rectangle(
    Math.Min(currentStroke.First().X,
        currentStroke.Last().X),
    Math.Min(currentStroke.First().Y,
        currentStroke.Last().Y),
    a, a);
var p = new Pen(drawingColor, width);
p.DashStyle = line;
e.Graphics.DrawEllipse(
    p,
    rec);
e.Graphics.FillEllipse(selectedBrush, rec);
}
private void drawEllipse(PaintEventArgs e)
{
    var rec = new Rectangle(
        Math.Min(currentStroke.First().X,
            currentStroke.Last().X),
        Math.Min(currentStroke.First().Y,
            currentStroke.Last().Y),
        Math.Abs(currentStroke.First().X -
            currentStroke.Last().X),
        Math.Abs(currentStroke.First().Y -
            currentStroke.Last().Y));
    var p = new Pen(drawingColor, width);
    p.DashStyle = line;
    e.Graphics.DrawEllipse(
        p,
        rec);
    e.Graphics.FillEllipse(selectedBrush, rec);
}

```

Iscrtavanje kroz navedene funkcije odvija se unutar funkcije pictureBox1-Paint koja se izvršava prilikom *Paint eventa* kontrole **picturebox**. Dok ostale mouse funkcije služe za praćenje klikova mišem i bilježe koordinate preko kojih se onda iscrtavaju likovi.

```

private void pictureBox1_Paint(object sender, PaintEventArgs
    e)
{
    if (currentStroke.Count > 1)
    {

```

```

        if (!pastedArea.IsEmpty)
        {
            drawPastedRect(e);
            return;
        }
        switch (selectedTool)
        {
            case Tools.RectangleSelect:
                drawSelectionRectangle(e);
                break;
            case Tools.Rectangle:
                drawRectangle(e);
                goto default;
            case Tools.Line:
                drawLine(e);
                goto default;
            case Tools.Pencil:
                drawPencil(e);
                goto default;
            case Tools.Circle:
                drawCircle(e);
                goto default;
            case Tools.Ellipse:
                drawEllipse(e);
                goto default;
            default:
                isSaved = false;
                break;
        }
    }
}

private void pictureBox1_MouseDown(object sender,
    MouseEventArgs e)
{
    if (!pastedArea.IsEmpty)
    {
        if (pastedArea.Contains(e.Location))
            dragging = true;
    }
    currentStroke.Clear();
    selectedArea = new Rectangle();
    mouseDown = true;
    mouseButtonPressed = e.Button;
    currentStroke.Add(e.Location);
}

```

```

        pictureBox1.Invalidate();
    }
    private void pictureBox1_MouseMove(object sender,
        MouseEventArgs e)
    {
        if (mouseDown)
        {
            currentStroke.Add(e.Location);
            pictureBox1.Invalidate();
        }
    }
    private void pictureBox1_MouseUp(object sender,
        MouseEventArgs e)
    {
        mouseDown = false;
        if(selectedTool != Tools.RectangleSelect || dragging)
        {
            history.Save();
            pictureBox1.DrawToBitmap(
                (Bitmap)pictureBox1.Image,
                new Rectangle(
                    0, 0, pictureBox1.Image.Width,
                    pictureBox1.Image.Height));
            currentStroke.Clear();
            isSaved = false;
        }
        dragging = false;
        pastedArea = new Rectangle();
    }
}

```

3.2 Copy - Paste

Sljedeće funkcije implementiraju funkcionalnosti izrezivanja, kopiranja i ljepljenja označenog dijela slike.

```

private void CopyToClipboard(Rectangle src_rect)
{
    Bitmap bm = new Bitmap(src_rect.Width, src_rect.Height);

    using (Graphics gr = Graphics.FromImage(bm))
    {
        Rectangle dest_rect = new Rectangle(0, 0,
            src_rect.Width, src_rect.Height);
    }
}

```

```

        gr.DrawImage(pictureBox1.Image, dest_rect, src_rect,
            GraphicsUnit.Pixel);
    }
    Clipboard.SetImage(bm);
}
private void copyStripButton_Click(object sender, EventArgs e)
{
    if (!selectedArea.IsEmpty)
        CopyToClipboard(selectedArea);
}
private void cutStripButton_Click(object sender, EventArgs e)
{
    if (!selectedArea.IsEmpty)
    {
        CopyToClipboard(selectedArea);
        history.Save();
        using (Graphics gr =
            Graphics.FromImage(pictureBox1.Image))
        {
            using (SolidBrush br = new
                SolidBrush(Color.White))
            {
                gr.FillRectangle(br, selectedArea);
            }
        }
        pictureBox1.Image = new Bitmap(pictureBox1.Image);
    }
}
private void pasteStripButton_Click(object sender, EventArgs
e)
{
    if (!Clipboard.ContainsImage()) return;

    Image clipboard_img = Clipboard.GetImage();
    pastedArea = new Rectangle(0, 0, clipboard_img.Width,
        clipboard_img.Height);
    var gr = pictureBox1.CreateGraphics();
    gr.DrawImage(clipboard_img, new Point(0, 0));
}

```


3.3 Filteri

Za definiranje filtera koji ćemo primjeniti na sliku koristi se `ColorMatrix` klasa. Primjenu možemo vidjeti u sljedećoj funkciji.

```
private void grayscaleToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    colorm = new ColorMatrix(
        new float[][]
        {
            new float[] { .3f, .3f, .3f, 0, 0 },
            new float[] { .59f, .59f, .59f, 0, 0 },
            new float[] { .11f, .11f, .11f, 0, 0 },
            new float[] { 0, 0, 0, 1, 0 },
            new float[] { 0, 0, 0, 0, 1 }
        });
    ImageAttributes imageAttributes = new ImageAttributes();
    imageAttributes.SetColorMatrix(
        colorm, ColorMatrixFlag.Default,
        ColorAdjustType.Bitmap);

    var bmp = new Bitmap(pictureBox1.Width,
        pictureBox1.Height);
    var gr = Graphics.FromImage(bmp);
    var rect = new Rectangle(0, 0, pictureBox1.Width,
        pictureBox1.Height);
    gr.DrawImage(
        pictureBox1.Image,
        new Rectangle(0, 0, pictureBox1.Width,
            pictureBox1.Height),
        0, 0,
        pictureBox1.Width,
        pictureBox1.Height,
        GraphicsUnit.Pixel,
        imageAttributes);
    history.Save();
    pictureBox1.Image = bmp;
}
```

3.4 Undo - Redo

Ova funkcionalnost implementirana je u klasi `UndoRedo`. Koriste se dva stoga `Undo` i `Redo`. U trenutku kada korisnik mijenja sliku, trenutna se slika spremi na `Undo` stog, a na njenu se novu kopiju primjene promjene koje je inicirao korisnik. Ukoliko korisnik pozove *Undo* akciju trenutna se slika sprema na `Redo` stog. Ukoliko se trenutna slika mijenja `Redo` stog se prazni.

```
public static UndoRedo getInstance(PictureBox pbox)
{
    INSTANCE.pbox = pbox;
    return INSTANCE;
}
public void UndoAction()
{
    if (IsUndoable)
    {
        RedoStack.Push(pbox.Image);
        if (pbox.Size != UndoStack.Peek().Size)
            pbox.Size = UndoStack.Peek().Size;
        pbox.Image = UndoStack.Pop();
    }
}
public void RedoActon()
{
    if (IsRedoable)
    {
        UndoStack.Push(pbox.Image);
        if(pbox.Size != RedoStack.Peek().Size)
            pbox.Size = RedoStack.Peek().Size;
        pbox.Image = RedoStack.Pop();
    }
}
public void Save()
{
    Image img = new Bitmap(pbox.Image);
    UndoStack.Push(img);
    RedoStack.Clear();
}
```