Warsaw University of Technology

Faculty of Mathematics and Information Science

BACHELOR THESIS
COMPUTER SCIENCE

Image acquisition and editing

Program pobierania i edycji obrazów

Author:
Mikołaj Przybysz

Supervisor:
prof. nzw. dr hab. inż. Władysław Homenda

Warsaw, April 2014

…………………….………..…….. ………………….……….……….

Author Supervisor

Abstract

Along the growing supply in the market, the advertising industry is growing even faster, producing more and more promotional campaigns involving computer tools for image edition. Most of the tools available to manage raster graphics are designed for professional use and require fee for commercial usage or even for personal usage.

This Image edition software was made to provide quick and easy mean for image edition, either in terms of size manipulation, rotation manipulation or with any other task related to image retouching. Additionally, smart filters such as sepia, black and white and gray scale have been added for improved user experience.

Concluding, this paper was made to guide the reader through the planning, design and development process to show how real life applications are being produced. Additionally paper does present user manual for the software for ease of use.

Streszczenie

Wraz z rosnącą podażą na produkty codziennego użytku, branża reklamowa rozwija się w coraz to większym tempie, tworząc coraz więcej kampanii reklamowych wykorzystujących oprogramowanie do tworzenia grafiki komputerowej. Większość programów do obróbki grafiki rastrowej jest projektowanych z myślą o ich profesjonalnym przeznaczeniu stąd też wymagają one zazwyczaj licencji, aby móc z nich korzystać komercyjnie. Co więcej wiele z tych programów wymaga wykupienia licencji nawet jeżeli wykorzystuje się je w prywatnym zakresie.

Image edition software to narzędzie, które ma umożliwić zwykłym ludziom edycję swoich ulubionych zdjęć, zarówno pod kątem zmiany rozmiarów zdjęć, jak również dokonywania poprawek związanych z obróceniem zdjęcia. Ponadto, oprogramowanie to zapewnia bardzo pomocny zestaw narzędzi do rysowania prostych figur geometrycznych. Co więcej, użytkownik ma do dyspozycji paletę prostych filtrów, które każdemu zdjęciu potrafią nadać unikalny wygląd.

Podsumowując, praca ta została stworzona w celu omówienia procesu planowania, projektowania oraz wdrażania projektów, które realizowane są na co dzień przez wielkie koncerny. Dodatkowo, dla wygody użytkowników do pracy została załączona instrukcja obsługi.

TABLE OF CONTENTS

Preface

The thesis is aimed on preparing a computer program for image manipulation. The program should provide tools for image edition including creating, opening and saving images, converting between different formats, zooming in and out, undoing and redoing actions, drawing elementary geometrical figures, re-sizing and rotating about custom angle, applying basic filters (sepia, black and white with threshold setup, gray scale), selecting area of the image, cutting and coping selected area of the image to clipboard, pasting image from clipboard onto image. The program should also be easy to use therefore it should incorporate images improving user perception of different elements.

Person responsible for planning, design and development is **Mikołaj Przybysz**.

1. Basic requirements

1.1. General

This chapter provides basic overview of the project. It covers the purpose of the project and its requirements as well as description of project development methodology.

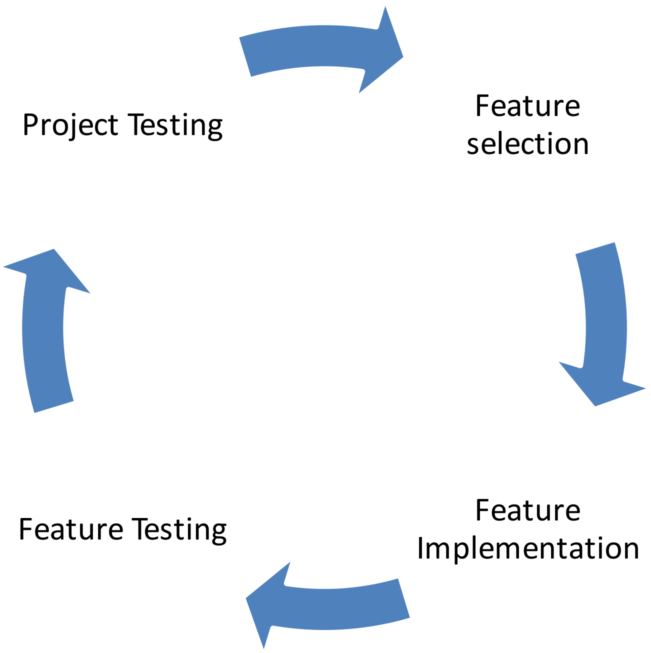
General purpose of the project is to design and implement an application allowing for image edition and manipulation. Application is designed for Windows XP or later operating system environments.

Application is to provide a user with basic means of image edition and manipulation that is:

- reading and writing basic graphic file formats
- conversion between supported graphic file formats
- mark, cut, copy, paste operations on an image
- adding to an image basic shapes (primitive)
- adding to an image basic filters (primitive)

- undo, redo, zoom capabilities of interface
- re-sizing, rotation and flipping capabilities

Chosen project development methodology was modified version of iterative model, that consisted of initial planing, iterative model shown below and deployment.



1.2. Image edition

1.2.1. Application drawing capabilities

To answer user needs built-in drawing capabilities must provide ability to draw basic shapes such as dots, lines, arcs, rectangles and ellipses. Mentioned tools should be extended with additional properties to ensure user is able to modify image in quick and easy fashion.

For drawing dots application should provide two basic tools. Those tools are pen tool which provides basic functionality for drawing one pixel thick lines and dots.
The other tool is brush capable of drawing lines and dots varying in thickness.

Rectangles and ellipses as for being closed curved shapes should be extended with the functionality for being drawn either with filled background and border or only with the filled background. Arc drawing capabilities will fill the gap between rectangle and ellipses enabling user to work even faster. All shapes should provide preview of drawn shape while it is being drawn.

Bucket tool or filling tool is a must have tool in any image edition software due to the fact it provides quick and easy way of filling area with one color that is used in almost any image development process.

All tools excluding pen tool and bucket tool should provide a way to customize line thickness. In case of closed curved shapes these means thickness of the border. Additionally, there should be a color box providing way to select fore color and back color for all the tools.

1.2.2. Image manipulation functionality

Images downloaded from cameras or scanners very often require size modification or simple rotation being applied. Additionally it happens that image requires flipping along one of the axis. Therefore application will provide functionality to rotate image about given angle (being integer value from zero to three hundred sixty) and to flip image along either axis X or axis Y. User will be able to either select angle from a user-friendly control or to input specific angle into text-box. Additionally application will provide a way to resize image, extended with optional constrain proportion feature to recalculated one of the dimensions when the other one is being edited.

Mark, cut, copy and paste are basic tools of image manipulation therefore should be also included. Mark tool enables user to mark an area of image, then cut and copy tool provide a way to copy marked area to system's clipboard, with the difference of copy from cut that the latter one will clear the selected area after its copied with the foreground color.

1.2.3. Additional effects

To enable user’s further modification, application provides images color change to either gray scale, sepia color palette or black and white where user can choose black and white threshold.

1.2.4. Interface

Interface will provide simple means of edition with one window having the whiteboard and all the other windows being separated. Beside whiteboard window there should be toolbox window with all the described tools and zoom window enabling for zoom in and out. Additionally, there should be windows opening and closing depending what tool is being selected with the configuration options related to that selected tool.

2. Description of solution

This chapter goes into the specifies of the project development methodology, implementation language and software employed for development, solutions provided in the release version and final project enviromental requirements.

2.1. Project development methodology

Decision to choose iterative development model, was based on the number of features to be developed. Project required personalization of the iterative development due to the specifics of the project and experience of the developer. This chapter provides description of the whole process.

Intial planing described basic software architecture and was preceded by analysis of open source projects available in the internet.

Iteration, consisted of:

- feature selection – where developer picked feature to be developed, from the list given in business analysis
- feature implementation – coding of the necessary classes and functions
- feature testing – ensuring that feature fulfills requirements of the business analysis and is user friendly
- project testing – ensuring that feature implementation didn't broke any already developed features

Finally deployment was concluded with public testing where internet users could test software and give feedback.

In conclusion this model provided simple yet effective methodology of software development, however as the project grow more and more testing become the most time consuming process. Solution for that problem turn out to be as simple
as introducing even further hermetization of the features and the application structure.

Software implementation have been performed using:

- Visual Studio Integrated Development Environment
- .NET 4.0 framework
- C# programming language

Supported operating systems:

- Windows 7
- Windows XP SP3

2.2. Image edition module

This chapter provides closer look at the technical aspect of the application, that means descriptions of actions performed in some cases precises algorithms incorporated into the application. As in the previous chapter it goes through drawing methods, manipulation methods, applicable effects and interface implementation in that order.

2.2.1. Application events

All the drawing tools are enclosed within three events, that is mouse click, mouse release and mouse move that take place on image within whiteboard object.

Mouse click event technically means that the drawing procedure is being initiated. At this moment mouse coordinates are being saved within whiteboard local variable. If selected tool is not cursor tool that means image changes and „image change” procedure takes place. This procedure saves unchanged version of image for „undo” action and clears cached zoom images. Addition it is checked if bitmap is pasted onto the image and if clicked coordinates are not enclosed within pasted bitmap, which disables possibility of further moving pasted bitmap. Disable takes place simply by changing the „paste” flag to false. Final step is to enable drawing, that is setting draw flag to true.

Mouse move event first action performed is updating status bar at the bottom, by changing data of X and Y coordinates of the mouse and color of the pixel that mouse pointer is hovering over. In case drawing is enabled application recalcules which coordinate should be starting coordinate and which is final coordinate due to the fact that all shapes require circumscribing rectangle which is created based on top left coordinate and bottom right coordinate. Secondly, method handling drawing is being selected for tool selected in toolbox. For brush and pencil tool drawing is being performed directly on original bitmap and on zoomed bitmap if zoom is not in neutral position (that is 100%). If selected tool points at bucket mouse move is neutral. For remaining shapes preview drawing is being performed only on the image that is shown on the whiteboard.

Mouse release event performs the same coordinates calculations as in mouse move event then based on the tool selected proper drawing is being performed. Additionally, recalculation of horizontal and vertical scroll position are being applied to the previous results of X and Y coordinates. For pencil and brush tool drawing does not require any additional calculations, however, for shapes applications must calculate the size of the brush for the zoom that is being selected.

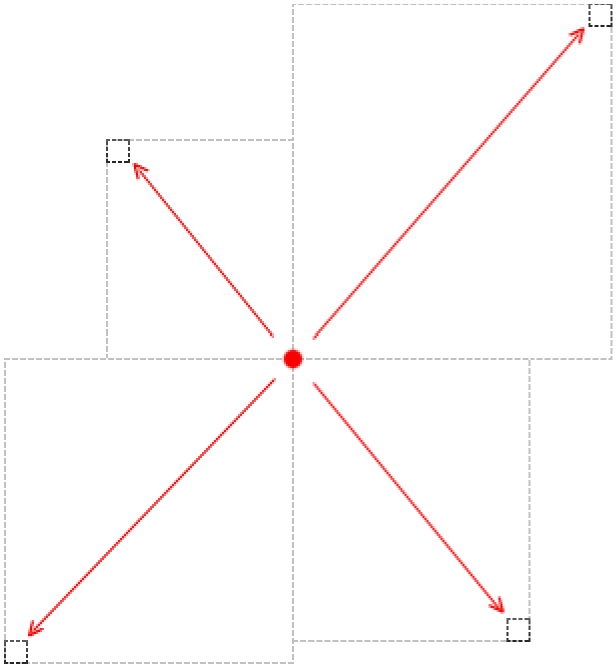


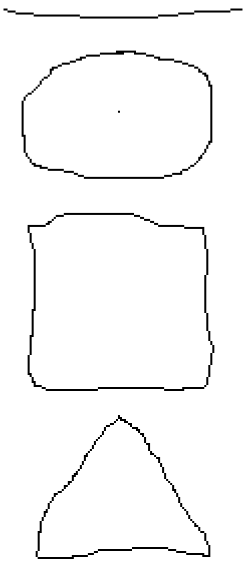
Image close checks if unsaved flag (boolean variable) is set to true, in such case „yes, no, cancel" popup window is being shown. Cancel action returns user back to application, by return from FormCloseEvent before it dispose the window. No action does noting. Yes action runs save as action.

2.2.2. Application drawing capabilities

Microsoft .NET framework provides number of methods to carry out drawing operation, therefore, after careful case study, decision was made to execute drawing procedures over the Bitmap object stored in memory and afterward present it to the user via PictureBox control.

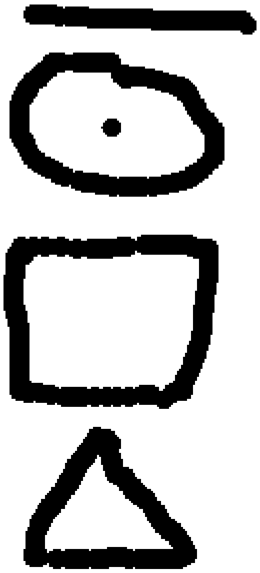
2.2.2.1. Pencil

To simulate pencil drawing, one pixel thick line is drawn from the location, where mouse have been clicked to the location where mouse have been moved and this position is saved as new starting point. This way user can continuously draw any shape. Pencil drawing uses basic line drawing and mouse move event. Dot drawing is being performed by FillRectangle function on mouse release event



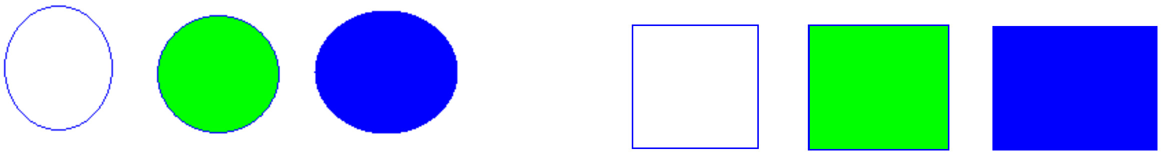
2.2.2.2. Brush

Brush drawing uses DrawEllipse for drawing on mouse move and FillEllipse and mouse release for dot drawing. To draw a dot with brush tool do mouse click on the pixel you want to put do dot on. Continues drawing is done by holding mouse button on move mouse over the pixels you to draw. Brush size can be changed via brush thickness window.

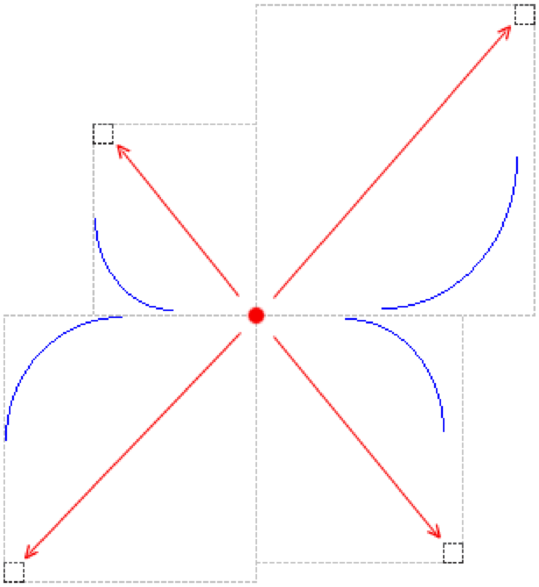


2.2.2.3. Rectangle and Ellipses

Rectangles and Ellipses are drawn similarly to line and mark tool, from the point where user clicks on the bitmap to the point where the mouse button is released. Thichness of the border can be changed via thickness window and style via shape style window.



2.2.2.4. Arc



Arc tool is very tricky due to the fact that it bends toward X axis, therefore arc drawing requires some experience with application. The blue line marks how the arc was drawn while red line marks the mouse move and its direction. Red dot is the starting point and the darker rectangle in the corner of bigger lighter rectangle is the point where the mouse have been released.

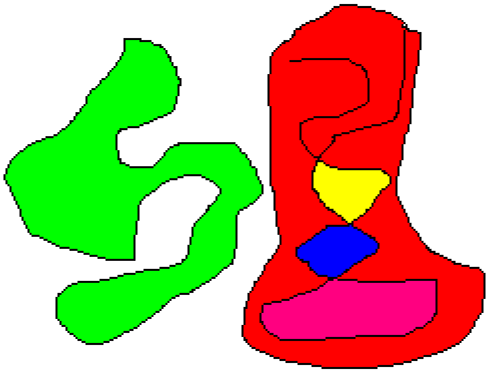
2.2.2.5. Bucket

Bucket tool fills continues region of the same color. Continues region consists of pixels on x and y axis, corner pixels are not considered as neighboring pixels.

Below filling algorithm is presented, it is custom algorithm not from .NET framework:

```
// color to be changed
Color colorTBC = wb.GetPixel(from.X, from.Y);

int maxW = wb.Width - 1;
int maxH = wb.Height - 1;

// list of colors already filled with new color
bool[,] colored = new bool[maxW + 1, maxH + 1];

// this table improves filling by factor 2
bool[,] added = new bool[maxW + 1, maxH + 1];

// list of pixels pending to be checked
List<Point> toColor = new List<Point>();

// add starting point to toColor list
toColor.Add(from);

int j = 0;
int k = 0;
while (toColor.Count > 0){

    // color current
    // get current point
    Point current = toColor[0];
    // pop it out from the toColor list
    toColor.RemoveAt(0);

    if (!colored[current.X, current.Y]){

        j++;
        // color it
        wb.SetPixel(current.X, current.Y, color);
        // add it to colored list
        colored[current.X, current.Y] = true;

        // find neighbours
        // 4-way filling adds pixels, from top, right, bottom, left
        Point[] pnts = new Point[]{
             new Point(withinInterval(current.X, 0, maxW),
                   withinInterval(current.Y -1, 0, maxH)),
             new Point(withinInterval(current.X, 0, maxW),
                   withinInterval(current.Y +1, 0, maxH)),
             new Point(withinInterval(current.X -1, 0, maxW),
                   withinInterval(current.Y, 0, maxH)),
           new Point(withinInterval(current.X +1, 0, maxW),
                   withinInterval(current.Y, 0, maxH))
           };

        // check if neighbours have colorTBC and are in image bounded region
        foreach (Point p in pnts){

            if (wb.GetPixel(p.X, p.Y) == colorTBC
                && true != colored[p.X, p.Y]
                && true != added[p.X, p.Y]
            )
            {
                k++;
                toColor.Add(p);
                added[p.X, p.Y] = true;
            }
        }
    }
}
```

Above algorithm has great performance on small regions, and gets slightly slower with larger areas. Further improvements can be done in the field of tolerance acceptance.



2.2.2.6. Line

Line is being drawn from the point where user clicks on the bitmap to the point where mouse button is released.

2.2.2.7. Cursor

This tool enables user not to make unwanted changes being made by accidental mouse click. It is also extend by two functions: image scrolling and moving pasted bitmaps.

2.2.2.7.1. Scrolling image

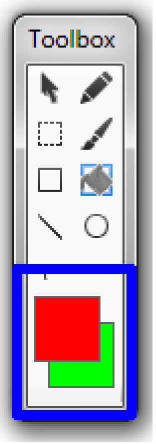
When large image (meaning one of the dimension is larger than the corresponding dimension of the PictureBox control) is loaded into application it is being shown only partially. To see remaining part of the picture PictureBox was extend with two scroll bars for each dimension. However, scroll bars are not necessary the best choice due to their lack functional optimization, that is user has to move mouse over the scroll bar, scroll it, move mouse back over the image. To optimize work with excessively large images Cursor tool includes option to scroll the image with a click over the bitmap and mouse move.

2.2.2.7.2. Moving pasted objects

When any Bitmap is being pasted from clipboard user can use Cursor tool to move pasted object over the original Bitmap to locate it in the location of user choice. User can move it until mouse click is made outside of the pasted bitmap. To achieve this result first new bitmap is being drawn in the center of the previewed bitmap. This bitmap’s location is stored in memory.

If mouse down event is done inside the pasted bitmap, on mouse move, we take original bitmap copy it to previewed bitmap and draw new bitmap in new location.

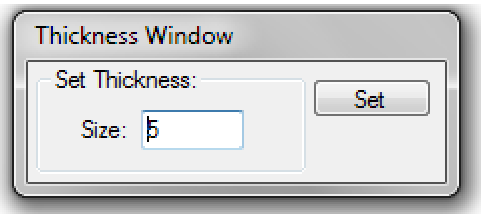
2.2.2.8. Colors



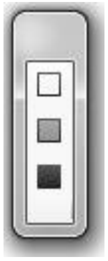
For flexibility reason application provides two colors to be selected from the palette, mainly, front color and back color. The former is used by each tool, except cursor and selection tool, as the main color of drawing. Additionally the latter provides color of background for closed curved shapes such as rectangles and ellipses. Fore color is the top color from the two colors in the blue box. Back color is the bottom color from the two colors in the blue box.

2.2.2.9. Thickness

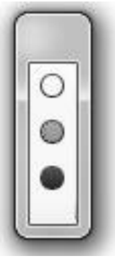
Both border of closed curved shapes and line thickness required method of customization its thickness. This have been achieved by tool thickness control, a simple window with a textbox provides full functionality required.



2.2.2.10. Shape Style



Rectangles and ellipses as for being closed curved shapes can be stylized by adding filling color and/or removed border. Filling color corresponds to back color from Color selection control.



2.2.2.10.1. ShapeStyle.FILLED

Filled style consists of border and filling.

2.2.2.10.2. ShapeStyle.BORDERLESS

Borderless style consists of filling and no border.

2.2.2.10.3. ShapeStyle.REGULAR

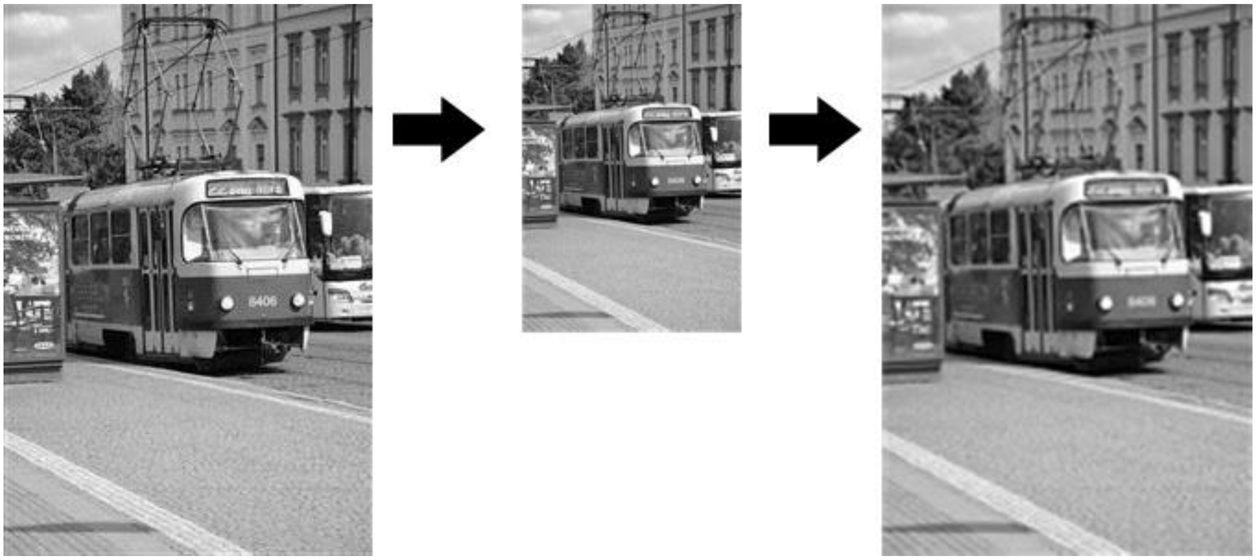
Regular style consists of border and no filling.

2.2.3. Application manipulation capabilities

Drawing tools are only part of image edition application, therefore to fulfill user needs, application was added ability to resize, rotate and flip images as well as to copy and paste parts of images.

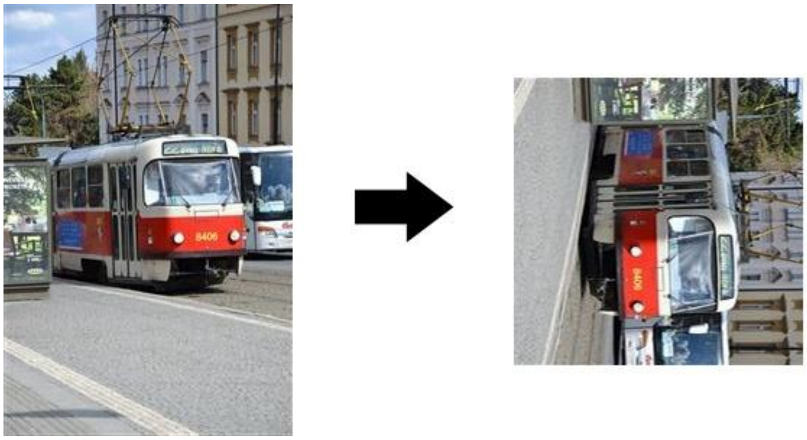
2.2.3.1. Resizing control

Resizing control role is to enable user to change final size of the image. This controller was extended with an algorithm to constrain proportions of the image when one of the dimensions is being edited. However, this algorithm can be turned off or on, depending on the user needs.



2.2.3.2. Rotation control

Beside basic rotation such as ninety degree clock-wise or counter clock-wise, Rotation control provides zero to three hundred sixty degree rotation. Additionally control provides a Clock control to ease up users work and instead of typing the angle by hand user can simple select it on the clock control.



2.2.3.3. Flip control

Flips image along either x-axis (Flip Horizontally) or y-axis (Flip Vertically).

2.2.3.4. Mark control

Selection tool enables marking rectangular area for being copied to clipboard. When making selection for dynamic selection visualization technique similar to the one used to redraw new bitmap in cursor tool at moving pasted bitmap mode.

2.2.3.5. Copy control

By selecting part of the image with a Rectangle Selection Tool user can copy this part into to the Clipboard, simply, by using Copy command from menu strip or by using CTRL+C. Afterwards part of the bitmap is being sent to the Clipboard as a Image object.

2.2.3.6. Cut control

Cut control does perform similarly to copy controller with the difference that it cuts the background of the copied regions and fills it with backcolor. This tool can be accessed from menu strip or by using CTRL+X.

2.2.3.7. Paste control

Application provides a method to paste bitmap from clipboard and to either create new image from it (that occurs when no other bitmap has been loaded) or to paste it over the already loaded bitmap.

2.2.4. Application photo effects

Effects used in the application use a ColorMatrix transformation.

Commands used:

```
using System.Drawing.Imaging;
ImageAttributes.SetColorMatrix(ColorMatrix cMatrix);
```

2.2.4.1. Grayscale effect

ColorMatrix used:

```
ColorMatrix colorMatrix = 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}
    }
);
```

2.2.4.2. Sepia effect

ColorMatrix used:

```
ColorMatrix colorMatrix = new ColorMatrix(new float[][]{
    new float[] {0.393f, 0.349f, 0.272f, 0, 0},
    new float[] {0.769f, 0.686f, 0.534f, 0, 0},
    new float[] {0.189f, 0.168f, 0.131f, 0, 0},
    new float[] {     0,      0,      0, 1, 0},
    new float[] {     0,      0,      0, 0, 1}
});
```

2.2.4.3. Black and white with threshold

ColorMatrix used:

```
ColorMatrix colorMatrix = new ColorMatrix(new float[][]{
    new float[] { 0.299f, 0.299f, 0.299f, 0, 0 },
    new float[] { 0.587f, 0.587f, 0.587f, 0, 0 },
    new float[] { 0.114f, 0.114f, 0.114f, 0, 0 },
    new float[] { 0,      0,      0,      1, 0 },
    new float[] { 0,      0,      0,      0, 1 }
});
```

Additionally to ColorMatrix this function uses

```
ImageAttributes.SetThreshold(tresholdLevel);
```

2.2.5. Application Interface

2.2.5.1. Scrolling large images

Horizontal and vertical scroll bar provides ability to move to the part of the bitmap that exceeds the size of the viewport.

2.2.5.2. Zooming in and out

Zooming control provides ability to zoom out large bitmaps to have a better preview of the whole work or simply to zoom in small pictures to see details.

2.2.5.3. Information control

Information control located in the bottom left corner shows position of the mouse over the bitmap and the color of the pixel the mouse is currently hovering over.

3. User guide

This chapter will guide the user through all the activities that the application provides. From the installation process, through drawing and painting to application of effects and interface usage.

3.1. Installation

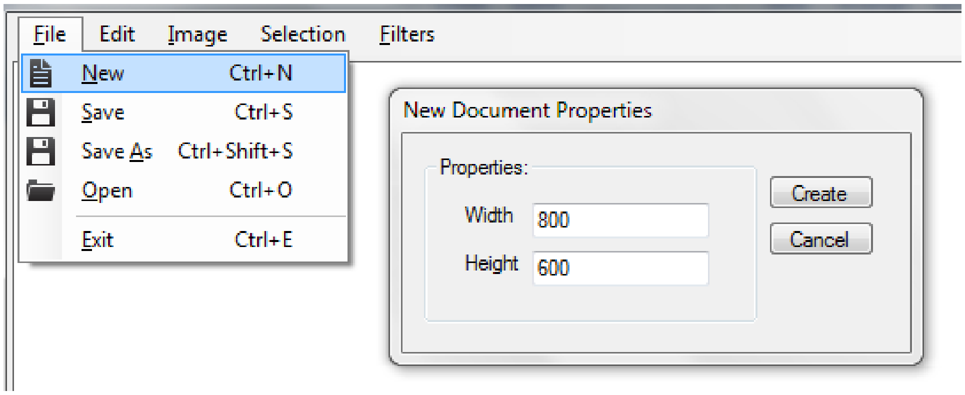
This software does not require any kind of installation. Application is made to be simple and all the files are incorporated in the exe file. Therefore user does not have to wonder about what file to choose. Simply find the exe file and run it as it is.

3.2. General

General section presents guides related to the application interface and its usage.

3.2.1. Create new image

To creating new image go to File > New or press CTRL+N, this will open new image properties window, shown below.

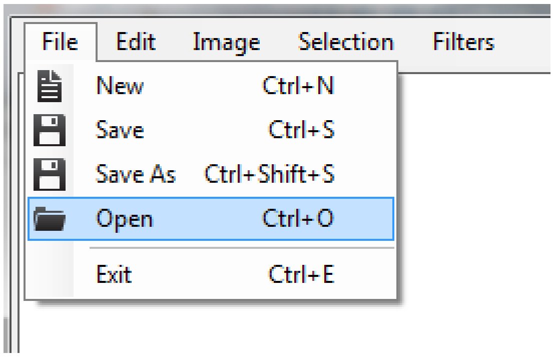


Width 800

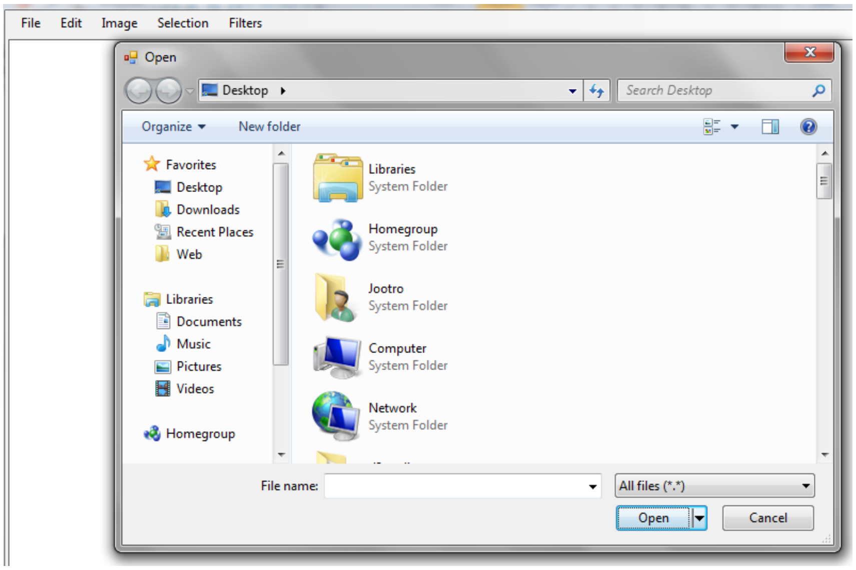
Height 600

Select dimensions of the image you want to work with and press Create button. This will create blank new bitmap with the dimensions you have typed in. Pressing Cancel button will close the New Document Properties window and take you back to the current image.

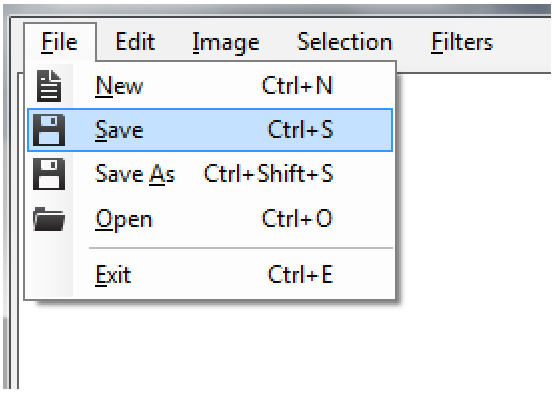
3.2.2. Opening file



To open a file go to File > Open or press CTRL+O and open file dialog will show up. Location of the Open button is shown below. Below is shown open file dialog, where in main window area shown your documents and on the left side are windows shortcuts. This is standard windows open file dialog that you encounter in other programs. To filter files only to see image files you need to roll out drop down list that is on the right side from the file name textbox in the bottom right corner of open file dialog. There you can find a filter called Image files (*.bmp, *.jpg, *.png, *.gif), selecting this option will do the job. Default option selected is All files (*.*).

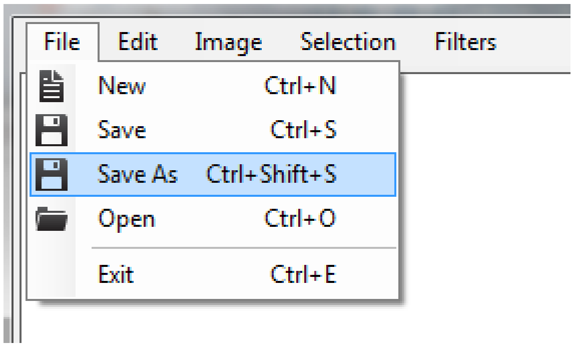


3.2.3. Quick save



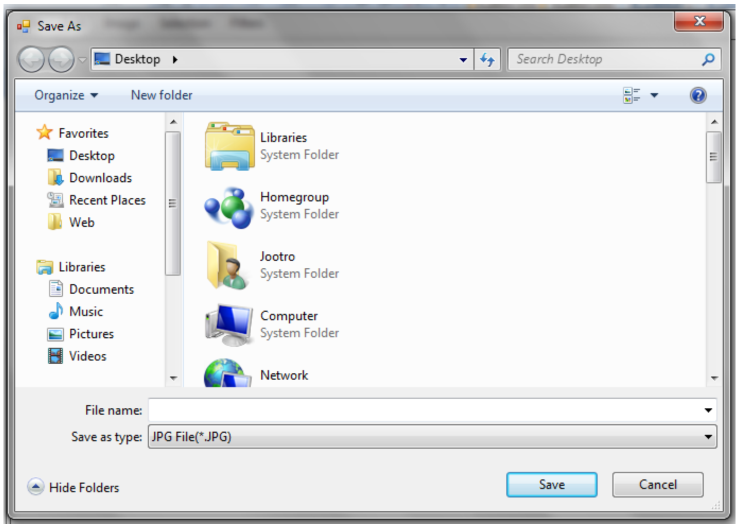
To save image in the file it was opened from simply go to File > Save or press CTRL+S. If you run quick save on new image it will run Save Image As instead as no default image exists! Location of File > Save is shown below:

3.2.4. Save image as



To save image as specific file select File > Save as option or press CTRL+SHIFT+S, it open up file dialog that.

On the bottom you can find textbox where user is suppose to type in file name under which image will be saved. Format of the file is selected from the drop down below textbox where you can choose format from bmp through jpg and gif to png.

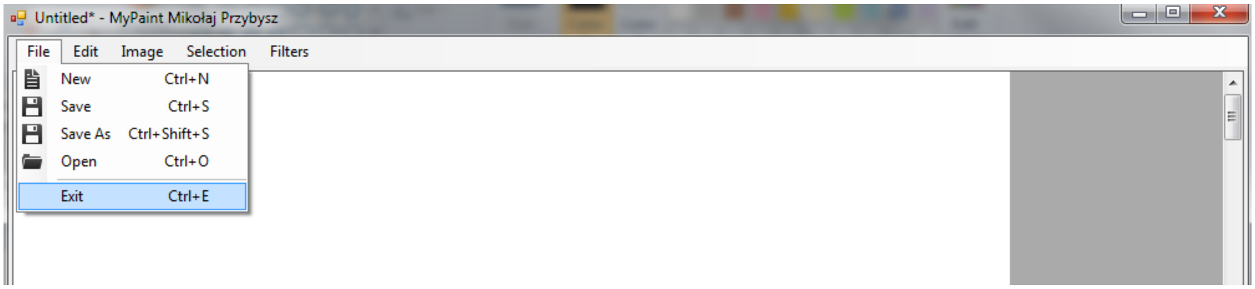


File name:

Save as type: JPG File(*.JPG)

3.2.5. Exiting

To exit the application choose File > Exit, press CTRL+E or press close window button in the top right corner marked with X.



After window close is selected depending on whether image has been changed or not application will eaither close or will ask us what to do with unsaved changes presenting us three choices in a exit dialog box.

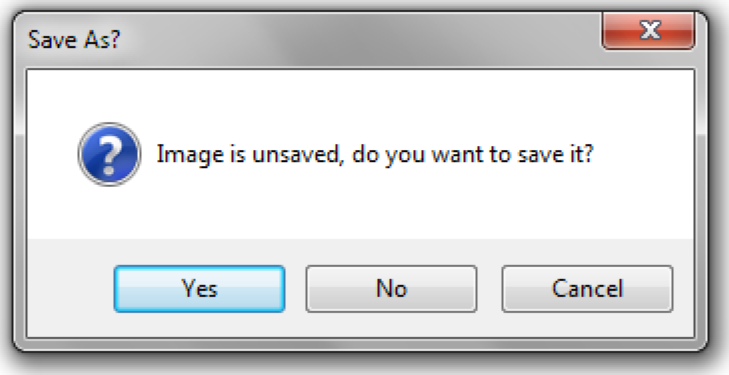
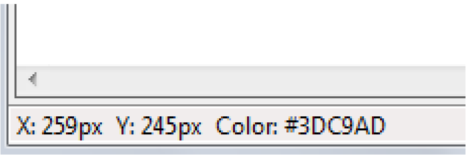

Save As?

Image is unsaved, do you want to save it?

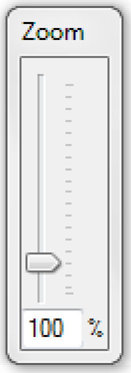
Option Yes will run quick save option, option No will discard changes and option Cancel will cancel exit process.

3.2.6. Interface

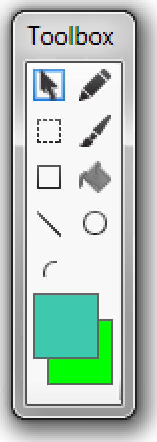

X: 259px Y: 245px Color: #3DC9AD

In the bottom left corner you can find informations related to the pixel mouse pointer is hovering currently. It presents two informations, one is pointer coordinates in the image that can help in precise drawing or selecting and pixel color information in RGB format.

Additionally application provides two functional windows, that is toolbox and zoom window. At the end of this chapter we will concentrate on zoom window and toolbox will be described further in next chapter. Zoom window enables user to zoom in and out of the image by manipulating either the scroll or by typing in zoom value in the textbox. For performance reason scroll provides quick selection of specific zoom values, that is 50%, 100%, 200% and 400%.

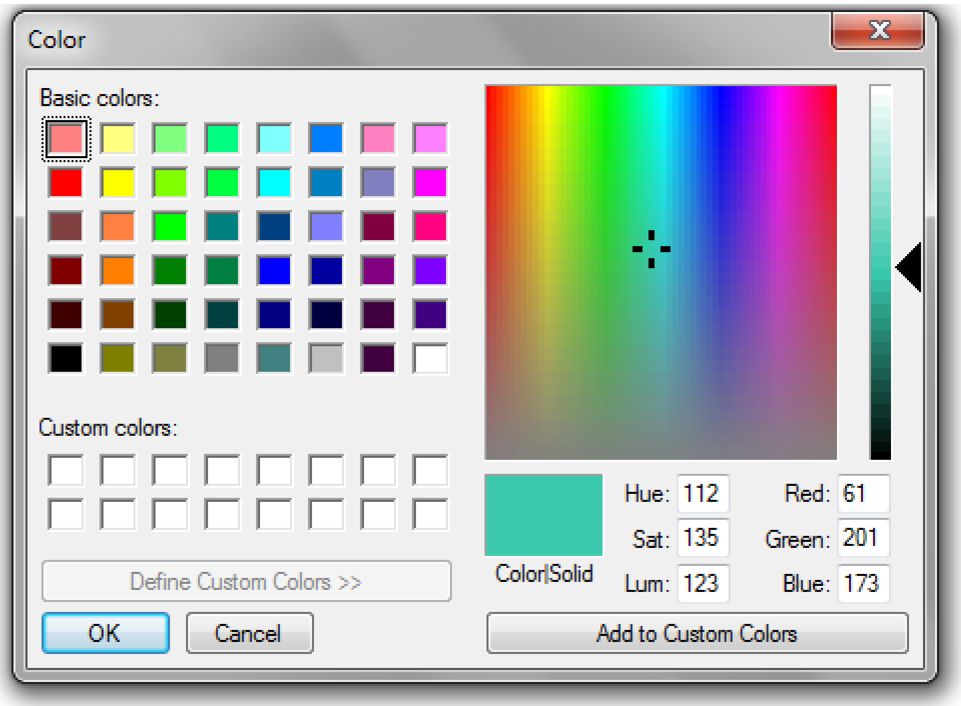


Zoom: 100 %



3.3. Painting and drawing

Painting and drawing starts at the toolbox where user have some tools to ease the job. Toolbox provides nine tools and to colors. Tools presented to user are described specifically further in the chapter. Color boxes provide front and back color and their usage is also described further in the chapter. To change color of front or back color simply click on the box and color dialog will show up where you can specify color you want to use.



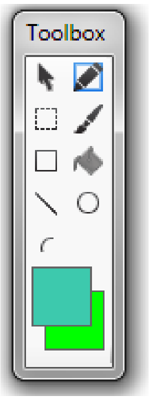
Hue: 112
Sat: 135
Lum: 123
Red: 61
Green: 201
Blue: 173

3.3.1. Pointer

The pointer tool enables to move image and auto scroll it without the need of using scrollbars and moves pasted objects into the bitmap.

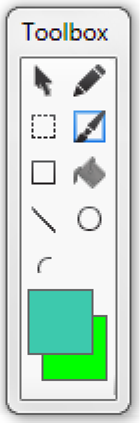
3.3.2. Pencil

Pencil tool provides capability to draw dots (one pixel point) or custom 1px thick shapes you can come up with. In order to draw dot pick pencil tool from toolbox and simply click your mouse in the place you want to put it. To draw line press left mouse button, hold it and start drawing!



3.3.3. Brush

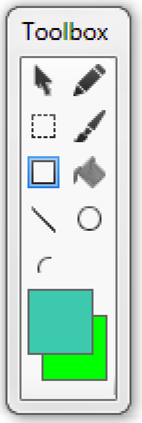
Brush tool similarly to pencil tool enables to draw lines and dots, however, provides ability to manipulate their thickness via thickness window. To draw dot click on the pixel you want to color and ellipse with radius of selected thickness divided by 2. To draw a line hold mouse button pressed and mouse mouse.

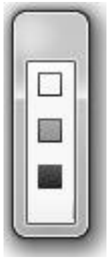


3.3.4. Rectangle

Rectangle tool is shipped with style window and thickness window. Style window provides three styles:

- Border only – fore color is the color of the border
- Filling with border – fore color represents border and back color represents filling
- Filling only – fore color represents filling color

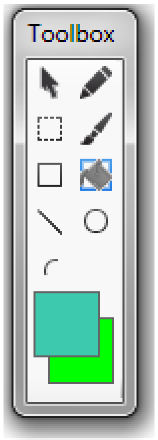




Thickness window sets border thickness. Description of this window can be found at the beginnig of this chapter.

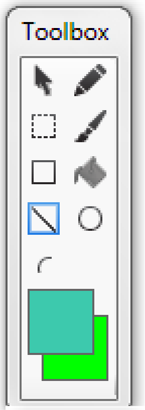
To draw rectangle select point which you want to start you rectangle at and release mouse button at the point opposite via diagonal.

3.3.5. Bucket



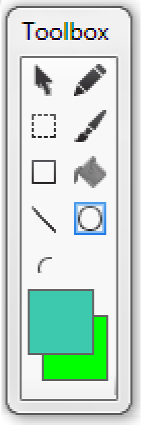
The bucket tool continuously fills adjoining pixels of the same color with the fore color. Simply pick a pixel in area you want to color and click on it with your mouse.

3.3.6. Line



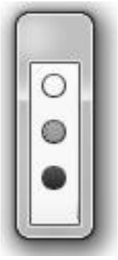
The line tool draws a fore color line with the thickness set in thickness window. To draw the line press you mouse button at the place you want to start and release it.

3.3.7. Ellipse



Ellipse tool is shipped with style window and thickness window. Style window provides three styles:

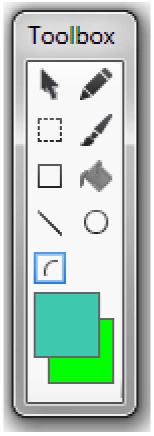
- Border only – fore color is the color of the border
- Filling with border – fore color represents border and back color represents filling
- Filling only – fore color represents filling color



Thickness window sets border thickness. Description of this window can be found at the beging of this chapter.

To draw ellipse select point which you want to start you rectangle at and release mouse button at the point opposite via diagonal.

3.3.8. Arc



The arc tool draws arc bended toward the X axis. To draw the line press you mouse button at the place you want to start and release it.

3.4. Manipulating

3.4.1. Mark

The mark tool lets you select rectangular area you want to copy or cut to clipboard.

3.4.2. Cut

The cut tool require selection with mark tool. If you have already select area you wish to cut or copy, then simply select Selection / Cut from top menu or press CTRL+X. Cut command will copy selected bitmap to clipboard and in its place it will put rectangle filled with solid back color and turn off selection.

3.4.3. Copy

The copy tool require selection with mark tool. If you have already select area you wish to cut or copy, then simply select Selection / Copy from top menu or press CTRL+C. Copy command will copy selected bitmap to clipboard and turn off selection.

3.4.4. Paste

The paste command will paste bitmap from the clipboard into to the center of currently open bitmap. User can move around pasted bitmap until mouse pointer does not click outside of the pasted bitmap. This command can be run from top menu strip by selecting Selection / Paste or pressing CTRL+V.

3.4.5. Resize

The resize tool does let the user to change the dimensions of the image. Dimensions can be constrained or not depending on the checkbox in resize window. In case checkbox is checked proportions of the image will be constrained, otherwise user can type custom width and/or height. Resize tool can be accessed from Image / Resize in menu strip.

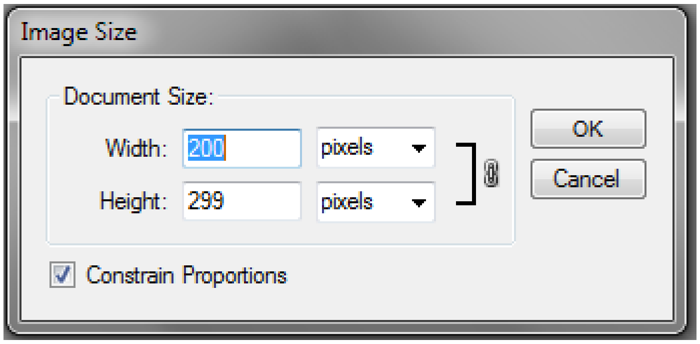


Image Size

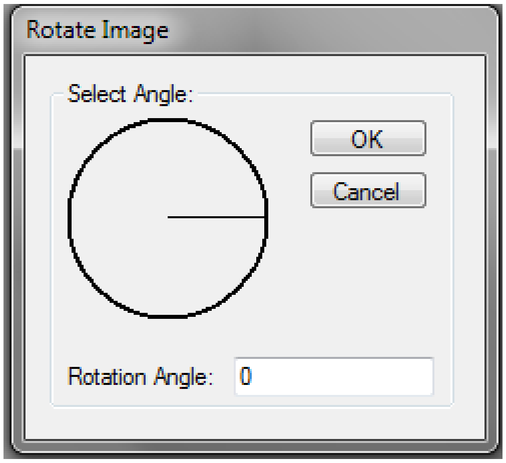
Document Size:

Width: 200 pixels

Height: 299 pixels

☑ Constrain Proportions

3.4.6. Rotate

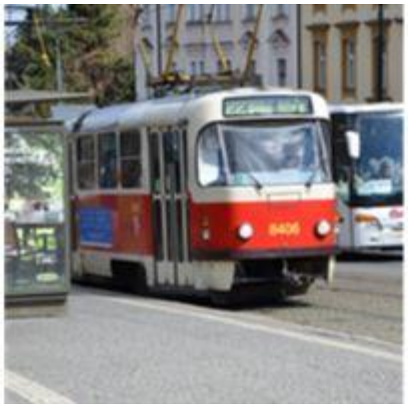


Rotate Image

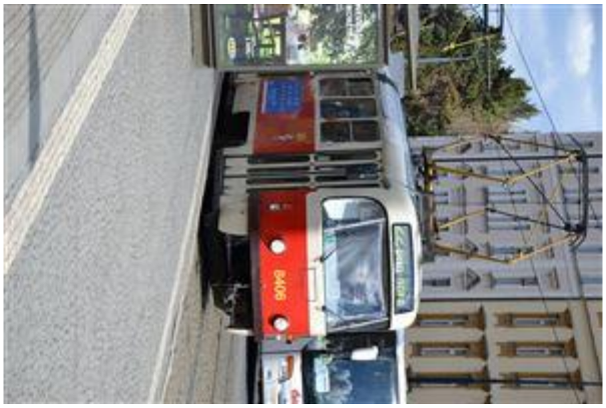
Select Angle:

Rotation Angle: 0

The rotate tool does let the user to pick specific angle or type it manually and rotate the image clock wise around the center point of the image. To enter rotate tool go to Image / Adjustments / Rotate.



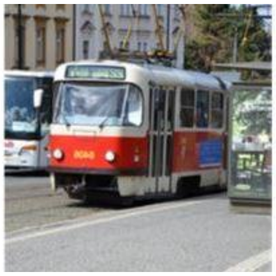
Rotated 270 degree clock wise



Original

3.4.7. Flip via X axis

This tool flips the whole image via the X axis.

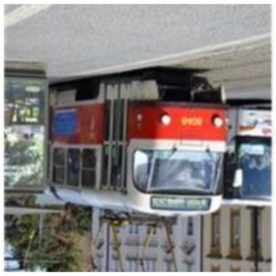


Flip via X axis

Original

3.4.8. Flip via Y axis

This tool flips the whole image via the Y axis.



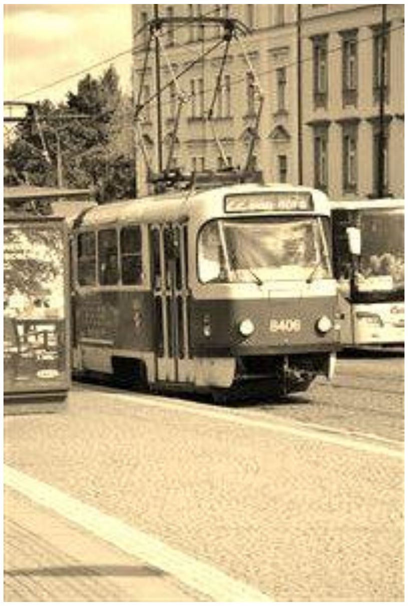
Flip via Y axis

Original

3.5. Effects

3.5.1. Sepia

This effect converts all colors to sepia scale.

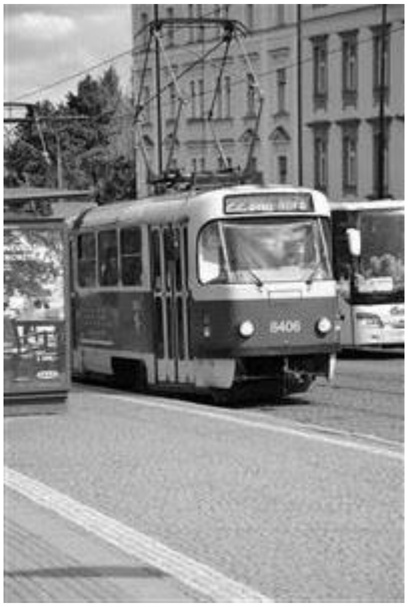


Sepia

Original

3.5.2. Monochromatic

This effect converts all colors to gray scale.

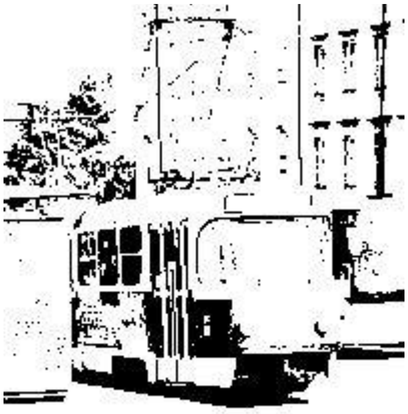


Monochromatic

Original

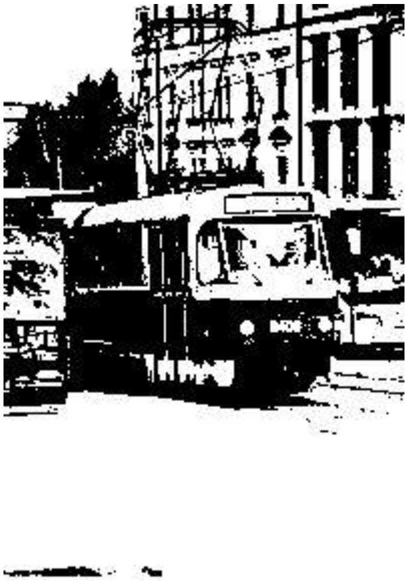
3.5.3. Black and white with threshold.

This effect converst all to colors to either black or white depending on the threshold user chooses.



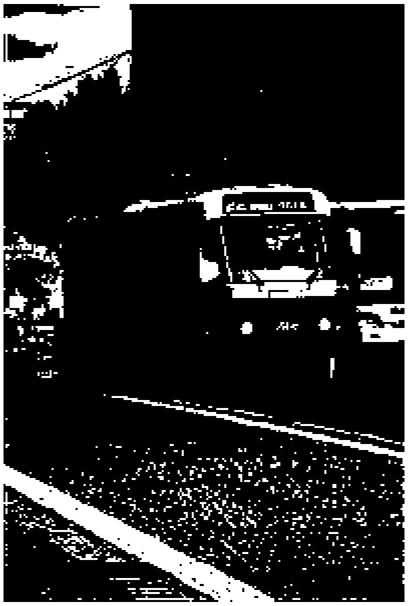
Threshold 25

Original



Threshold 50

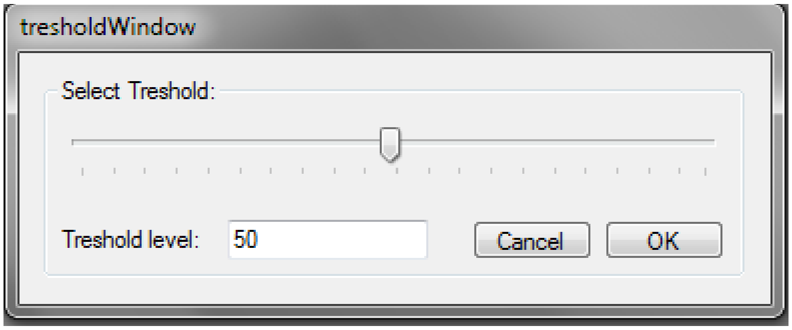
Original



Threshold 75

Original

The level of the threshold can be specified in the threshold window:



tresholdWindow

Select Treshold:

Treshold level: 50

4. Conclusion

In conclusion, all the aspects specified in business analysis have been planned within technical analysis and implemented at one of the iterations steps. Nevertheless, after final application testing it was found that it possess great potential for further improvement in the field of new features and implemented tools usability. New tools such as selection manipulation or different brush styles would enhance graphics development speed. On the other hand improvement of the implement tools usability such as zoom would extend the editorial usability scope of the application. Solution that would minimize list of potential critical opinions on the final application is definitely incorporating application beta testing with real life users and business and technical analysis reviewing.

5. Bibliography

[1] Booch Grady, *The infied modeling language user guide*, Addison Wesley, 2005

[2] Albahari Ben, *C# 4.0 in a nutshell*, O`Reilly Media, March 2002

[3] Stephens Rod, *C# Graphics Programming*, John Wiley & Sons, Nov 17, 2010

[4] Erich Gamma, Ralph Johnson, *Design Patterns: Elements of Resuable Object-Oriented Software*, Addison Wesley, 1994

Warszawa, dnia ....

Oświadczenie

Oświadczam, że pracę inżynierską pod tytułem; „Image acquistion and editing” , której promotorem jest prof. nzw. dr hab. inż. Władysław Homenda wykonałem samodzielnie, co poświadczam własnoręcznym podpisem.

……………………………………