

General plan

Topic 151 – Drawing program

1. Personal info

Title: Drawing program (topic #151)

Student name: Tu Tran Le

Student number: 401311

Study program: Service Design and Engineering

Submission date: 14.04.2014

2. General description

Create a simple, graphical, mouse driven drawing program.

The target requirements are at Average level with following requirements:

- Create a new drawing
- Choose color
- Drawing a line with mouse
- Drawing empty circles, ellipses and boxes
- Clearing the drawing
- Undo
- Saving to a file
- Loading from a file
- If the drawing window becomes wholly or partially obstructed by another window, is minimized etc. The picture should be normally redrawn when that part of the image or the whole image becomes visible again.

3. A Draft of the User Interface

The user interacts with the program using mouse. The interactions are done using mouse click and mouse move, which is similar to the common built-in Paint program of the OS (e.g. Microsoft Paint). The program also supports saving/loading data using the file system (using bitmap file format). Users can start a new command by clicking on the

corresponding button on the GUI. The program can accept following commands:

- Create new drawing
- Open existing drawing
- Save the current drawing
- Select the line color
- Draw a line
- Draw circle
- Draw eclipse
- Draw box (rectangle)
- Clear the drawing
- Undo / Redo

The program layout is illustrated as follow:

Toolbar	
Contains buttons for creating/loading/saving/clearing the drawing	
Commands	Drawing area
Contains buttons for creating shapes in the drawing	Contains the actual drawing canvas
	Color picker
	Contains the color selection control

Figure 1: Program layout

4. Files and file formats

The file supports the binary Bitmap file format (with default extension .bmp). The structure of the file format is as follow (quoted from Wikipedia):

Name	Optional	Size	Purpose
Bitmap File Header	No	14 Bytes	To store general information about the Bitmap Image File
DIB Header	No	Fixed-size (however 7 different versions exist)	To store detailed information about the bitmap image and define the pixel format
Extra bit masks	Yes	3 or 4 DWORDs ^[6] (12 or 16 Bytes)	To define the pixel format
Color Table	Semi-optional	Variable-size	To define colors used by the bitmap image data (Pixel Array)
Gap1	Yes	Variable-size	Structure alignment
Pixel Array	No	Variable-size	To define the actual values of the pixels
Gap2	Yes	Variable-size	Structure alignment
ICC Color Profile	Yes	Variable-size	To define the color profile for color management

Figure 2. Bitmap file format.

5. System testing plan

The system can be tested so that it can fulfill all the requirements. As it is difficult to automate the testing of user interactions and the screen output, thus the testing can be done by actually using the program and partially by unit tests. The basic operations to be tested:

- File manipulation (opening/saving)
- Modify & clear the drawing (involves creating shapes and undoing)

In principle, the program should not crash in any case. If there is an error, the program should show an error message with sufficient details (or with instructions how to

proceed if possible). The target is to verify that all basic features are functioning properly. Some of the preliminary use cases are as follow:

- Create a new drawing
 - If no drawing is opened: Create a new empty drawing
 - If drawing is being modified: Asks if the users want to save the existing one. Afterwards, create a new empty drawing
- Open existing drawing for modification
 - If the file can be opened: Show the file content in the drawing canvas area
 - If the file is too large/corrupted: Inform the users. One way to determine if the drawing is too large is to use a file size limit, or compare the file size with the amount of free system memory
- Save the current drawing
 - If the file cannot be saved: Inform the users
- Select the line color: All new shapes created afterwards should have this line color
- Draw a shape on blank or existing drawing (line, circle, eclipse, box)
- Clear the drawing: The program should ask for confirmation first
- Exit the program:
 - If the drawing is being modified: Ask if the users want to save the drawing
 - If there is no opened drawing / empty drawing canvas: Exit without confirmation
- Undo / Redo: The command stack should work repeat the corresponding commands in a logical manner

The order of testing is to test all common use cases first. Afterwards, the program can be tested for unhandled exceptions by trying out erroneous and extreme cases.