Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

WINDOWS PROGRAMMING

Laboratory work #3

Basics of Working with the Mouse. GDI Primitives. Bezier Curve.

Authors:
Mereuta Alex

Supervisor:
Irina Cojanu

Laboratory work #4

1 Purpose of the laboratory

Gain knowledge about basics of event-driven programming, Windows Timer. Animation.

2 Laboratory Work Requirements

– **Basic Level (grade 5 - 6) you should be able to:**

a) Create an animation based on Windows timer which involves at least 5 different drawn objects

– **Normal Level (grade 7 - 8) you should be able to:**

a) Realize the tasks from *Basic Level.*

b) Increase and decrease animation speed using mouse wheel/from keyboard

c) Solve flicking problem describe in your readme/report the way you had implemented this

– **Advanced Level (grade 9 - 10) you should be able to:**

a) Realize the tasks from *Normal Level.*

b) Add 2 animated objects which will interact with each other. Balls that have different velocity and moving angles. They should behave based on following rules:
-At the begining you should have 3 bals of different colours of the same size
-On interaction with each other, if they are of the same class (circle, square), they shuld change their color and be multiplied.
-On interaction with the right and left wall (the margins of the window), they should be transformed into squares.
-On interaction with the top and bottom of the window - the figures should increase their velocity.
-Please, take into consideration that the user can increase and decrease animation speed using mouse wheel/from keyboard

c) For the task above, add balls with mouse.

3 Laboratory work implementation

3.1 Tasks and Points

-Draw few lines of different colors and weights Lines are drawn with the MoveToEx() function which moves to the specified point and LineTo() - draws a line from specified position to another one.You can choose whatever color and weight you want your line to be from the menu.

-Draw a Bezier(one or more) curve Bezier Curve is determined by 4 points(x,y). I've specified a POINT array with custom points. Therefore curve is drawn with the PolyBezier() function.

-Draw few plane objects of different colors, weights, filled and not Drawn objects : Ellipse with Ellipse() function, rectangle with Rectangle() function. Weight is determined by the value of weight variable. For filled option there is a checkbox, if it is checked - the object is drawn filled, otherwise - not.Therefore you can draw whatever objects you want and I even made shortcuts for one of them.

-Draw 2 different objects using mouse Objects that are drawn with mouse, are managed in the WMLBUTTONDOWN , WMLBUTTONUP , WMMOUSEMOVE and WMRBUTTONDOWN. Objects that are drawn with mouse - Pen, Line, Polyline, Ellipse, Rectangle.

-Draw a custom bitmap image (1 pt) I actually made a bitmap image in paint from more elements, aka mended more tools in one picture.The bitmap was attached to the application with the help of LoadImage() , GetObject() and BitBlt() functions.

-Add a switch (button, select list...) that will change mouse ability to draw objects There is a group-box of radiocheck buttons for choosing desired color, made with WSGROUP style. Also there are several push buttons that allows the user to choose the desired tool to draw (paint).

-Hook keyboard input. Add 2 different keyboard combinations that will change mouse ability to draw objects (ex. on Ctrl+C will draw circles, on Alt+R will continue to draw circles but of read color) I made some shortcuts to be used from the keyboard.One is for choosing to draw rectangles by pressing CTRL + E and another option is clearing the screen from the keyboard by pressing CTRL + C.

-Added 1 button for clearing the drawing area The clear button which also has a shortcut

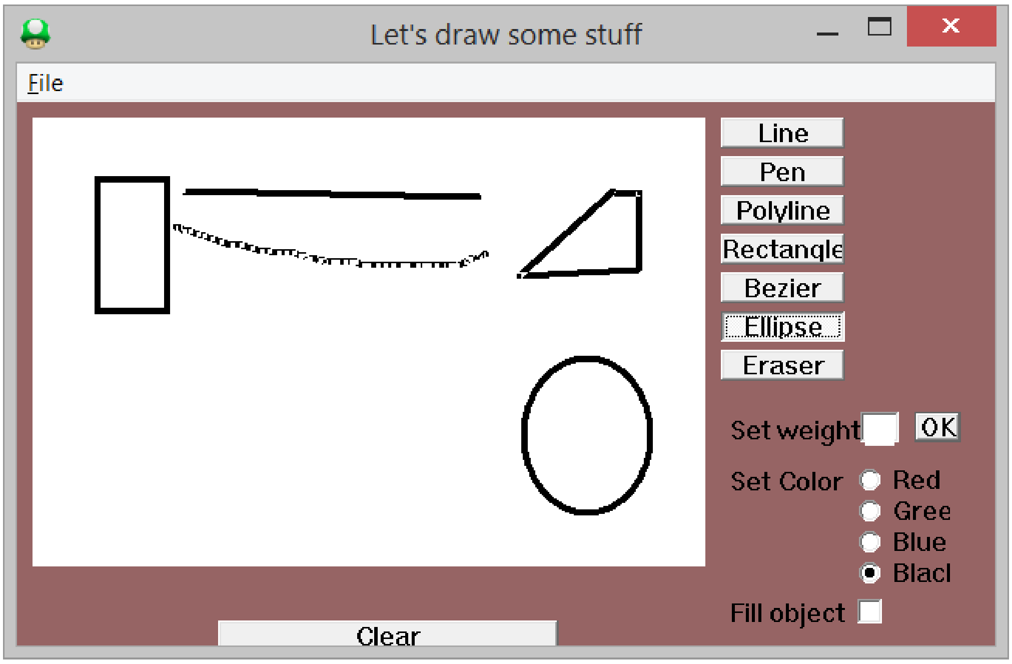
3.2 Laboratory work analysis

Add link to your repository. Create a README.md file for each laboratory work you submit. `https://github.com/TUM-FAF/FAF-141-Mereuta-Alex`

I already explained what my laboratory works contains in the paragraph above.

3.3 Prove your work with screens

Here we have a line, the pen, a polyline, a rectangle and an eclipse.

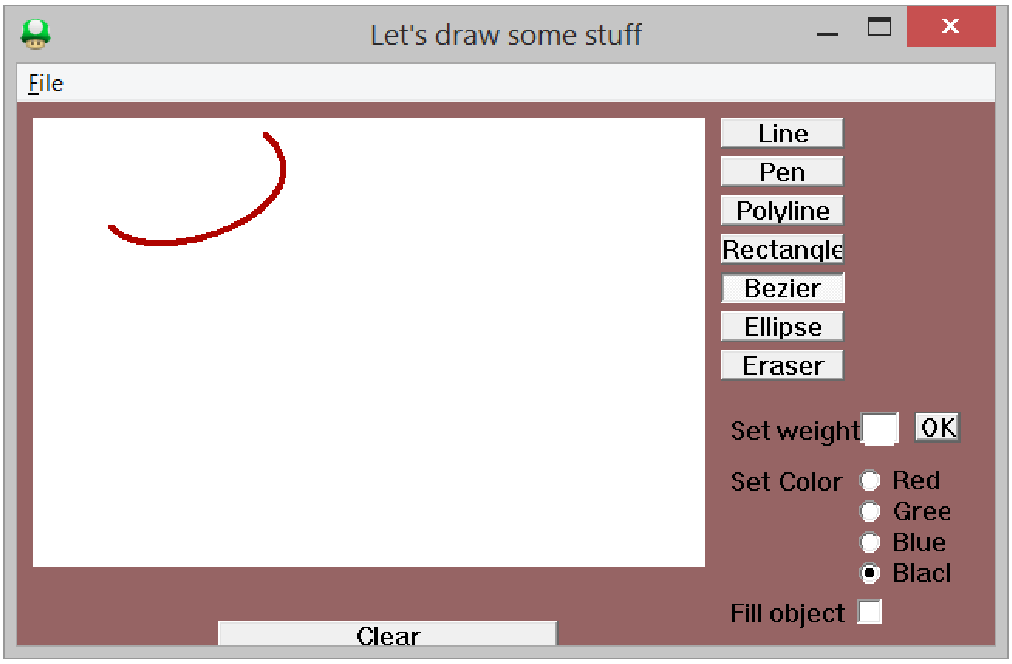


Set weight

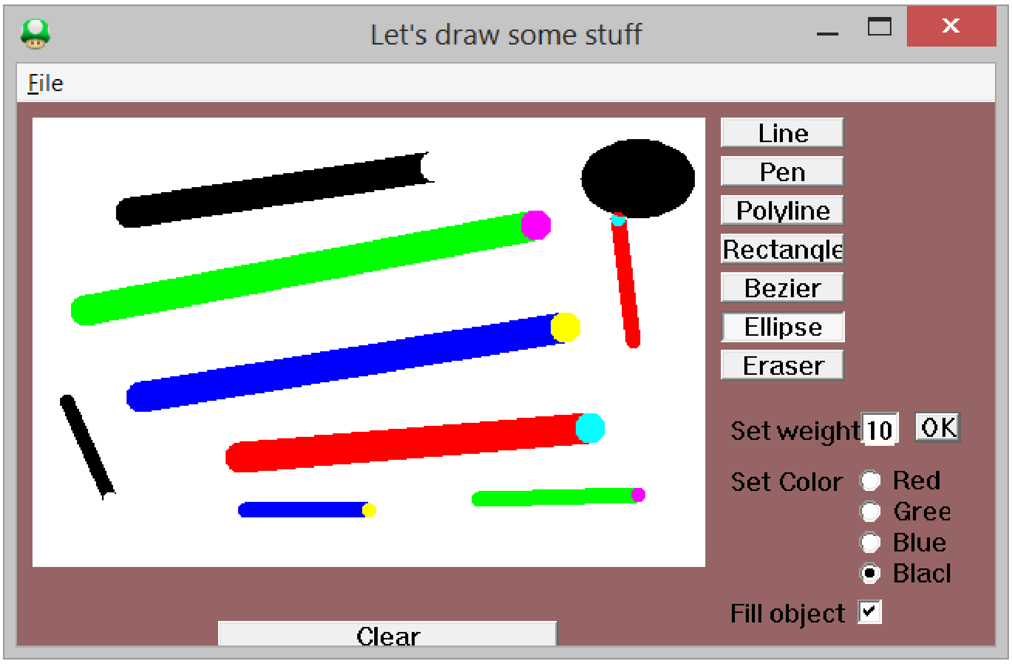
Set Color ○ Red
○ Gree
○ Blue
● Blacl

Fill object □

Here we used the clear function and drew a Bezier line.



Here I draw lines in different colors and different sizes, I also used the ”fill” option for that eclipse.



Set weight 10

Set Color ○ Red ○ Gree ○ Blue ● Blacl

Fill object ☑

Conclusions

This laboratory work was hard but with a lot of fun. I developed skills in working with GDI primitives. It was shown that we can draw different objects like (lines, ellipses) with different width and colors.I had a lot of problems with drawing the bezier, but after some research and help I finally managed to do it.In this laboratory work I familiarized myself with all the drawing possibilities the Windows API gives us, and with the right way of working with the mouse, and letting the user control the application features using mouse clicks. It was challenging to complete the tasks, I really enjoyed performing them. For me it's a new experience and it improve my skills as programmer. I think there are a little bit space for improvements, and I'll make them as soon as possible. This laboratory work broadened and deepened my understanding of Event-Driven Programming, message-sending, and Win32 API in particular.

References

1 Microsoft Windows, *official page*, https://msdn.microsoft.com/en-us/library/bb384843.aspx

2 C++ website, http://www.cprogramming.com/tutorial/opengl_first_windows_app.html

3 LateX basics, https://www.sharelatex.com