Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

Event-Driven Programming

Laboratory work #1

Window. Window handling. Basic window's form elements

Authors:
Dumitru Chetrusca

Supervisor:
Mihai Coșleț

Laboratory work #1

1 Purpose of the laboratory

Gain knowledge about basics of event-driven programming, understanding of window's class and basic possibilities of Win32 API. Also she will try to understand and process OS messages.

2 Laboratory Work Requirements

- **Mandatory Objectives**
  - Choose a *Programming Style Guideline* that you'll follow
  - Create a **Windows application**
  - Add 2 `buttons` to window: one with default styles, one with custom styles (size, background, text color, font family, font size)
  - Add 2 `text` elements to window: one with default styles, one with custom styles (size, background, text color, font family, font size) *[one of them should be something funny]*
  - On windows resize, one of the `text`s should "reflow" and be in window's center (vertically and horizontally)
- **Objectives With Points:**
  - `(1pt)` Add 2 text inputs to window: one with default styles, one with custom styles (size, background, text color, font family, font size)
  - `(1pt)` Make elements to fit window on resize *(hint: you can limit minimal window width and height)*
  - `(0-2pt)`Make elements to interact or change other elements (`1pt` each different interactions) *(ex. on button click, change text element color or position)*
  - `(1pt)` Change behavior of different window actions (at least 3). For ex.: on clicking close button, move window to a random location on display's working space
  - `(1pt)` Write your own PSG (you can take existent one and modify it) and argue why it is better (for you)

3 Laboratory work implementation

3.1 Tasks and Points

I have implemented all the tasks in this laboratory work.I have made the app in CodeBlocks IDE in C++ language using the template for a Win32 GUI application. The program is ussual with the 2 main methods : WinMain and WindowProcedure.The WinMain method is almost as the default one except I have changed the background color,name and position of window with the following snippets :

wincl.hbrBackground = CreateSolidBrush(RGB(0,0,255));

With this snippet I have setted the background color with blue.

WindowProcedure is the function called at every message here I am doing the main work. Initially I declare some variables which I will use. I won't enumerate all because are many. In some words there are variables for fonts, for handles of controls, for brushes etc. The variables are named in such way to understand the purpose. Then with the main switch statement I begin to handle the messages.First message is WMCREATE. In this case i create the required controls,buttons and text fields. This are created with CreateWindow() or CreateWindowEx() for extended styles. Then in case of WMPaint I have prepared the background and the used fonts. Also I have sent messages to the controls about the used font. E.g. :

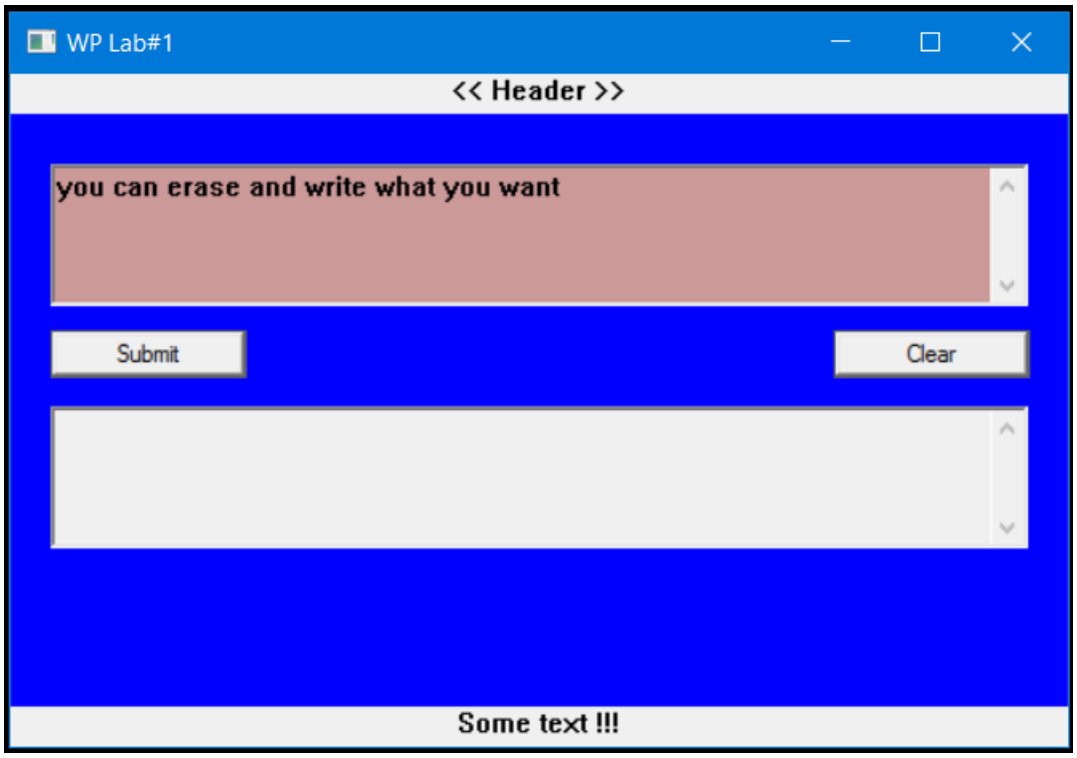
$SendMessage(buttonSubmit, WM_SETFONT, (WPARAM)defaultFont, MAKELPARAM(FAL$

Also for the styling purpose I have used the messages WMCTLCOLOREDIT and WMCTLCOLORSTATIC which are used to change styles of edit and static text boxes.More precisely in those the brush is prepared and is sent to the control. Also there is specified the text color.The positions of controls are specified in WMSIZE. It is more comfortable this way because here we can make them change acording to the resizing.The button Submit is used to add the text form upper edit field to the other one and the clear button clears the second edit field.And finnally to change the bahaviour of window actions we treat the WM SYSCOMMAND.

3.2 Laboratory work analysis

Event-Driven Programming Laboratories:
https://github.com/DEMENCI/WPLabs

3.3 Prove your work with screens



Conclusion

While I have been doing this laboratory work was certainly interesting,there were complications the most memorable of which was the inability to change the button color in a more "user friendly" manner . The work with different controls made me realize more indepth at how windows programs actually work and it is a thing I very much enjoyed .Anyway, I think that it is very nice to learn different technologies not only for the purpose of using them directly, but for creating some concepts from the architectural point of view. Overall I haven't encountered big problems and anything was quite clear explained on MSDN.

References

1 Very official Document, *official page*, `www.google.com`

2 Timo Ojala, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*, 2002