Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

WINDOWS PROGRAMMING

Laboratory work #4

Windows Timer. Animation.

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

-Create an animation based on Windows timer which involves at least 5 different drawn objects Actually the animation can involve up to 60 objects, created with a mouse press. More than 60 objects generally can be created. It only depends on the size of array of objects we declare.

-Increase and decrease animation speed using mouse wheel/from keebord
In order to do, the logic was introduced under the WM MOUSEWHEEL message. While the wheel was moved, the variable timerSpeed was increased, or vice versa - decreased, depending on wheel moving direction.

-Solve flicking problem describe in your readme/report the way you had implemented this
The primary cause of flickering is erasing the background, then immediately drawing over it. And because we draw the whole client area of our window in WMPAINT, I add a handler that returns the true value to indicate that the background has been erased.

-Add 2 animated objects which will interact with each other. Balls that have different velocity and moving angles.
At interraction the objects change their color.

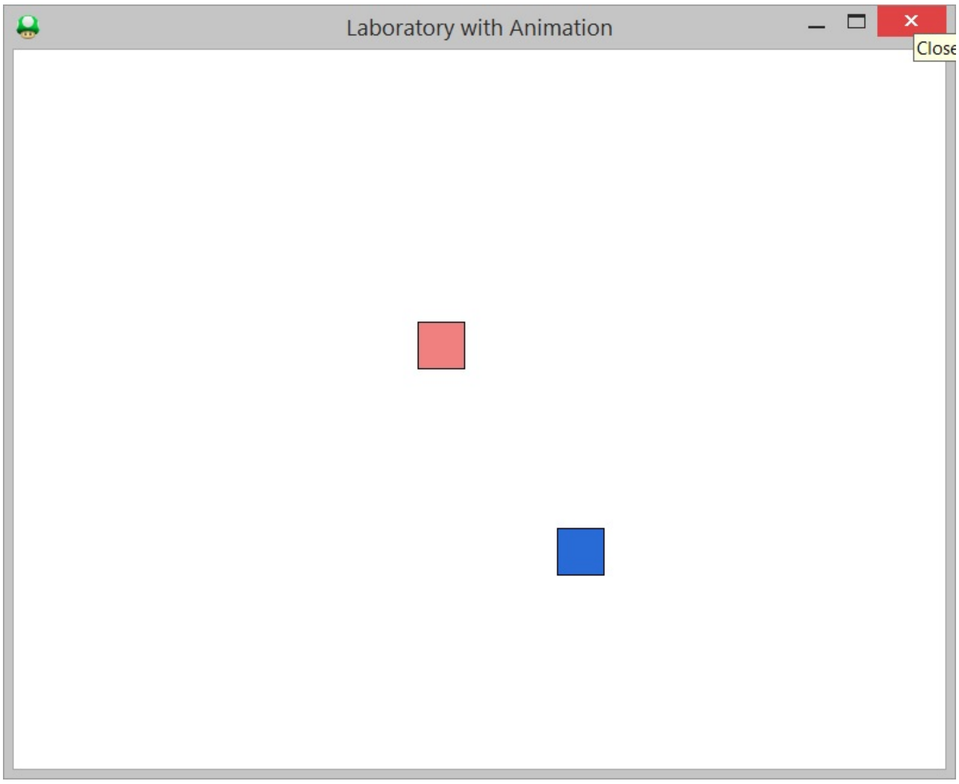
-Bonus point
When the button was pressed, a new object was created. It was done using the message WM LBUTTONDOWN. Using "new", a new instance of the object class was created.Also added a icon just for fun.

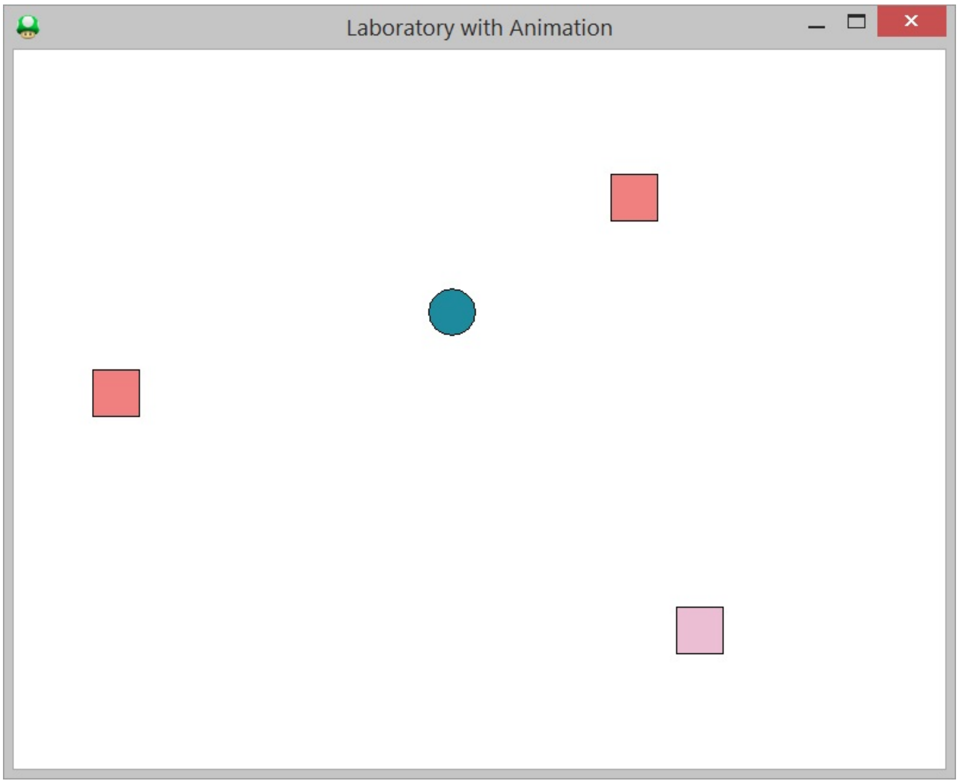
3.2 Laboratory work analysis

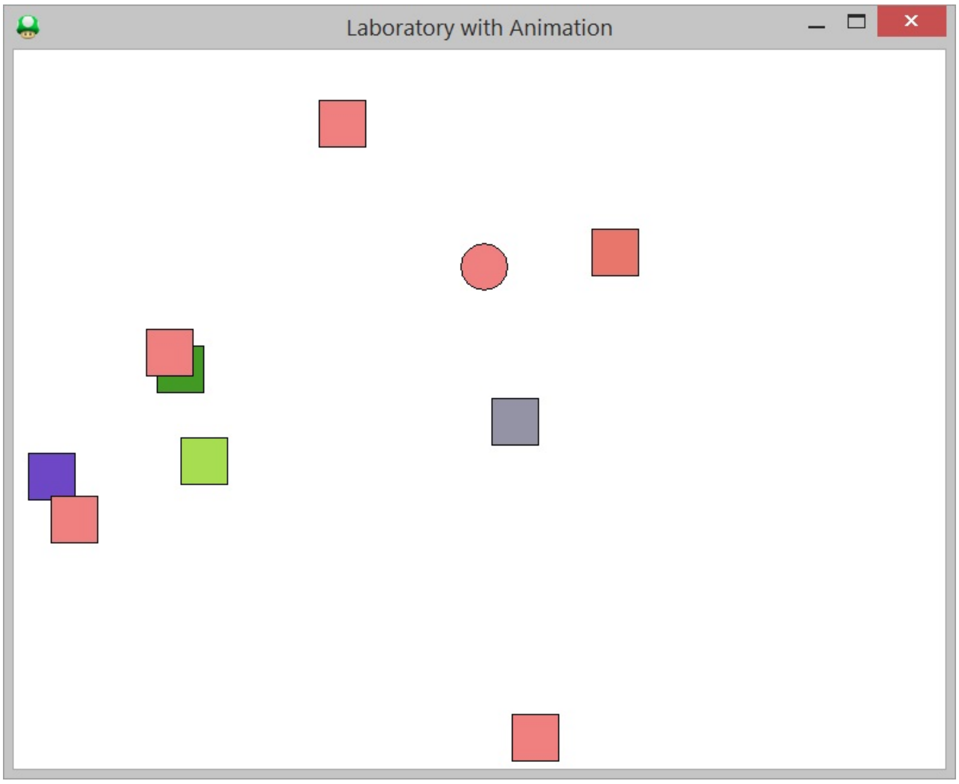
Add link to your repository. Create a README.md file for each laboratory work you submit. `https://github.com/TUM-FAF/FAF-141-Mereuta-Alex`

I already explained what my laboratory works contains in the paragraph above. See more details in the README.md

3.3 Prove your work with screens






Laboratory with Animation

Conclusions

I have created a Windows application that can create an animation based on Windows timer which involves more drawn objects. You can increase and decrease animation speed using mouse wheel/from keyboard. Also, you can add objects (balls) with mouse. These balls change their color at the interaction with each other and they became squares on interaction with the right and left wall. To solve the flicking problem I used double-buffering technique.

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