Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

WINDOWS PROGRAMMING

Laboratory work #4

Windows Timer. Animation.

Authors:
Daniel surdu

Supervisor:
Irina Cojanu

Laboratory work #3

1 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 read-me/report the way you had implemented this

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

a) Realize the tasks from *Normal Level* without *Basic 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:

1) At the beginning you should have 3 balls of different colors of the same size
2) On interaction with each other, if they are of the same class (circle, square), they should change their color and be multiplied.
3) On interaction with the right and left wall (the margins of the window), they should be transformed into squares.
4) On interaction with the top and bottom of the window - the figures should increase their velocity.
5) Please, take into consideration that the user can increase and decrease animation speed using mouse wheel/from keyboard

– **for Bonus Point Tasks :**

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

2 Laboratory work implementation

2.1 Tasks and Points

- Increase and decrease animation speed using mouse wheel/from keyboard

I used WM_MOUSEWHEEL to increase and decrease speed of animation. In that windows message I checked if HIWORD(wParam) is bigger than 0 it means that mouse-wheel were scrolled up and if is smaller than 0 it means that mouse-wheel were scrolled down. When mouse is scrolled up I increase the acceleration of the ball object with 1 and -1 if mouse is scrolled down. Before increasing or decreasing the acceleration I check if it is not bigger than 10 or smaller than 0 so my maximum acceleration is 11 and minimum is -1.

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

To solve the flickering problem I used double-buffering. For this you need to create an off-screen DC. This is made this way: hdcMem = CreateCompatibleDC(hdc); hbmMem = CreateCompatibleBitmap(hdc, win_width, win_height);

hOld = SelectObject(hdcMem, hbmMem);

and after that you have to transfer the off-screen DC to the screen: BitBlt(hdc, 0, 0, win_width, win_height, hdcMem, 0, 0, SRCCOPY);

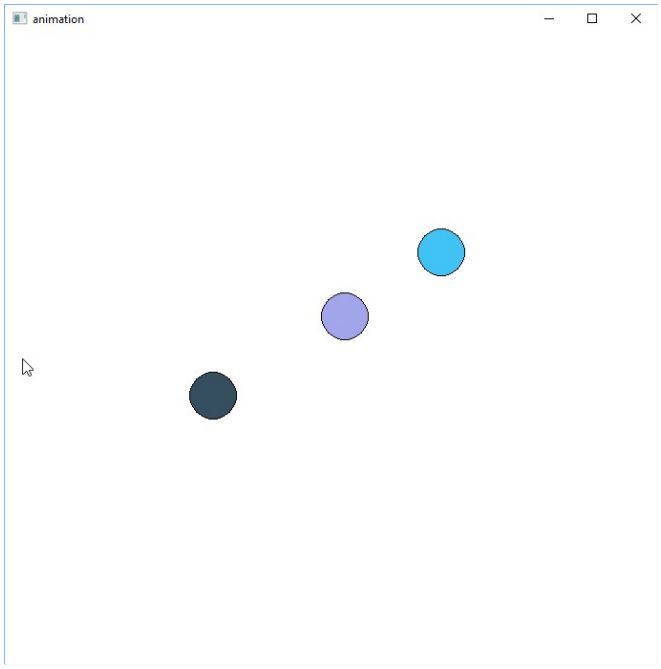
- Add 2 animated objects which will interact with each other.

I add objects using mouse, when WM_LBUTTONUP message appears I get the coordinates of the mouse at the moment and I create a ball with random color and a fixed size. When the ball touches the left or right walls of the client zone they bounce and change into the square, if they touche the top and bottom walls they also bounce but they accelerate and transform back into balls. When two balls or two squares touches, they change they color randomly and multiply (they multiply only once and their "children" can not multiply at all)

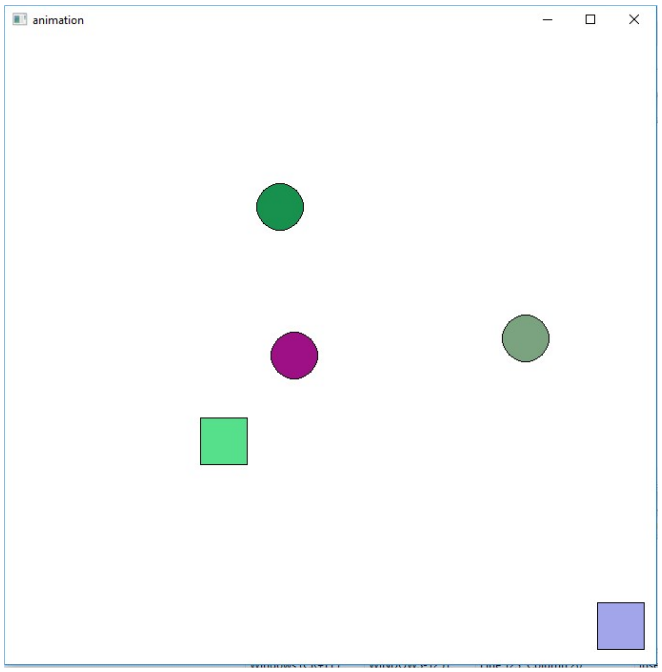
2.2 Laboratory work analysis

https://github.com/UnnemotionalHyena/WP

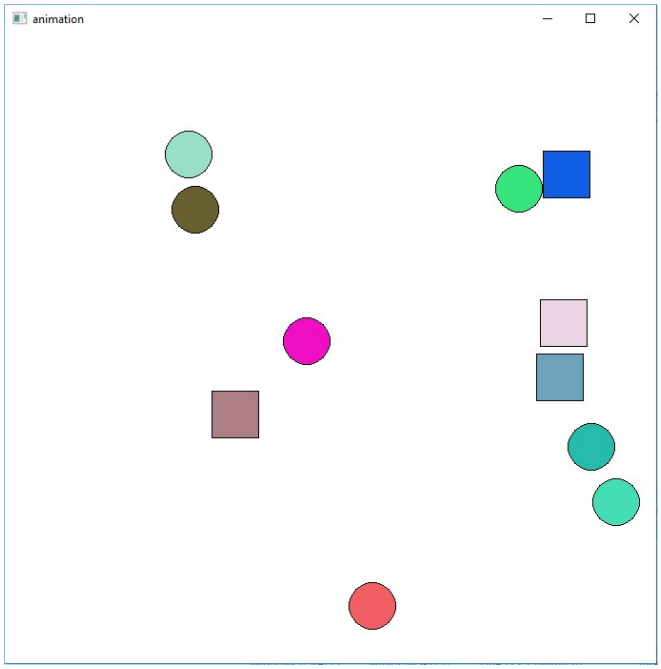
2.3 Prove your work with screens



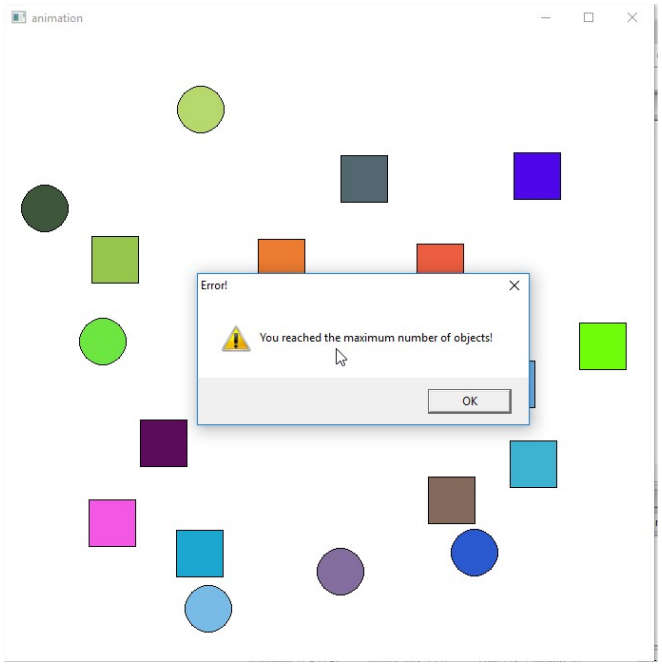
The basic window with three balls created using mouse



Balls change into squares and multiply



More objects multiplied and transformed into squares or back into balls



You reached the maximum number of balls

Conclusions

During this laboratory work, I learned about how to work with timer and how to create animations in win32 api. Animation in win32 have such problems as flickering when objects move, this is why you have to use double buffering but this means that your program will work slower. Win32 is not created to make animation so it's far from perfect, are a lot better tools to do that, basic C with game loops libraries is a lot better for animation than win32 and there you'll not have suck problems as flickering or a lot other problems that appeared during that laboratory work. I do not recommend anyone to do animations in win32 api.

References

1 Charles Petzold, *Programming Windows, 5th Edition*, 1998