Faculty of Computers, Informatics and Microelectronics

Technical University of Moldova

Event-Driven Programming

Laboratory work #4

Authors:

Ganusceac Vlad

Supervisor:

Mihai Coslet

Laboratory work #4

1 Purpose of the laboratory

Windows Timer. Animation.

2 Laboratory Work Requirements

Contents:

- Windows timer

Mandatory Objectives:

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

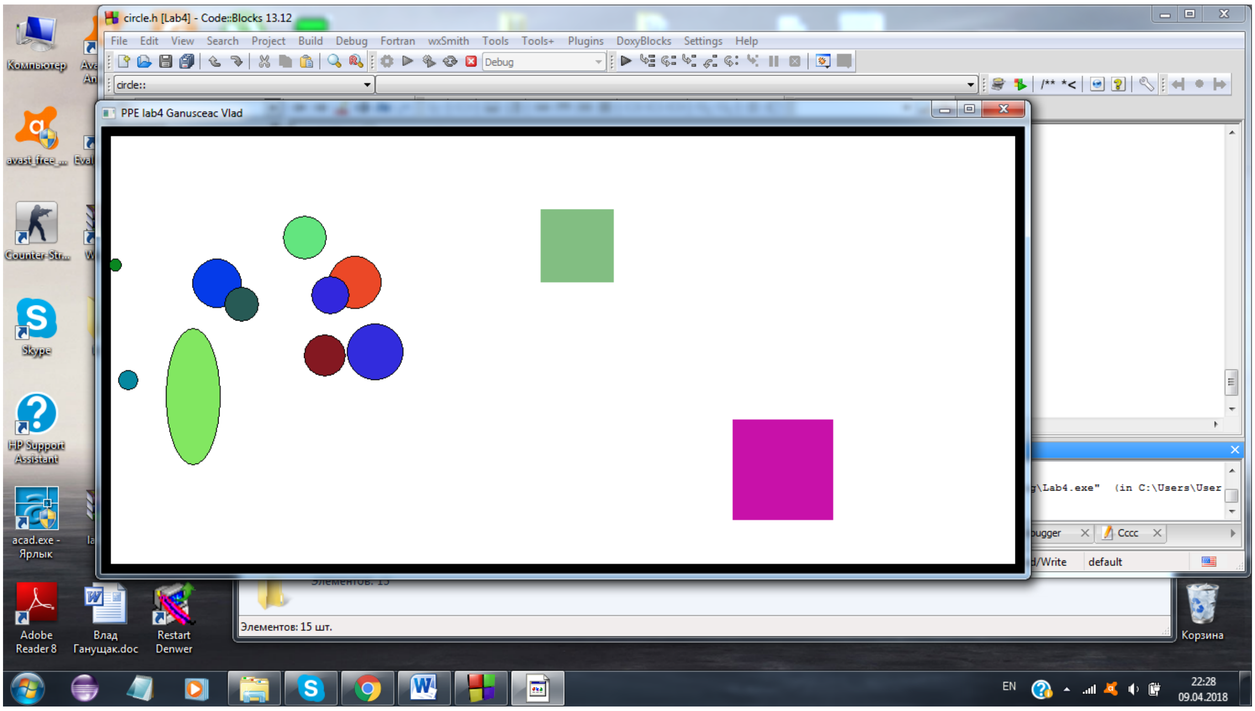
Objectives With Points:

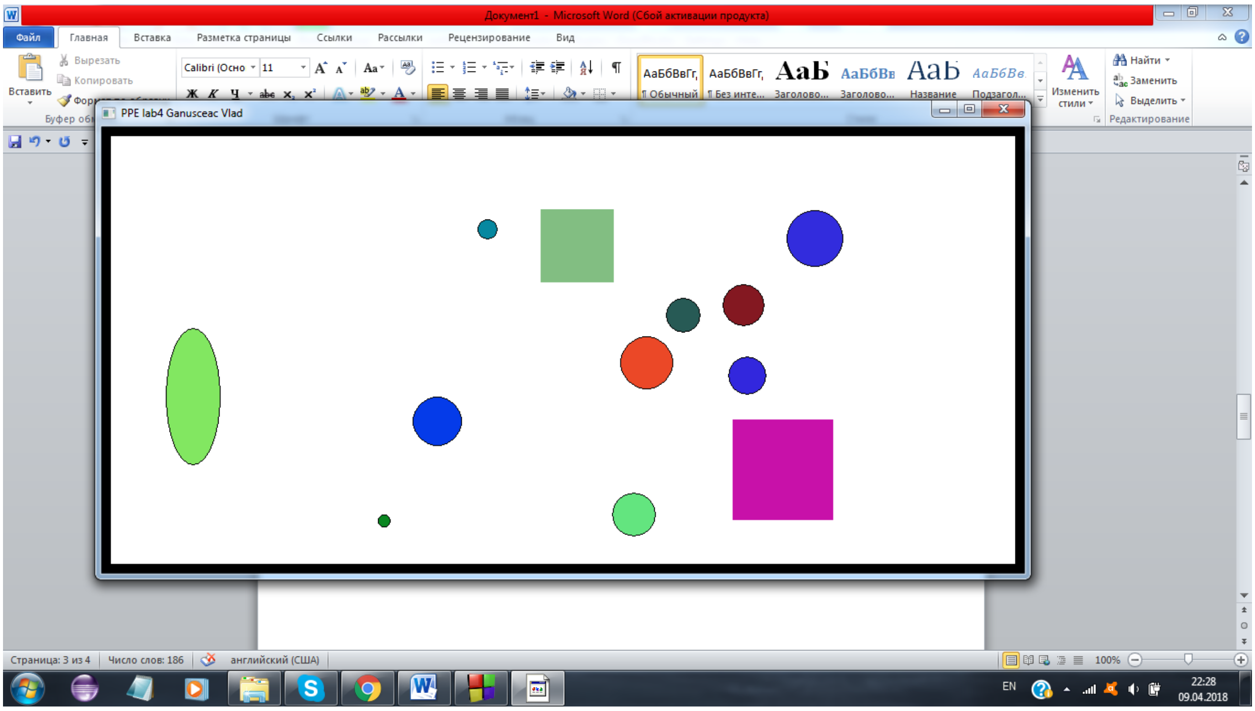
- Increase and decrease animation speed using mouse wheel (2 pt)
- Solve flickering problem (2 pt) *please describe in your readme/report how you did it*
- Add animated objects which interact with each other (2-6 pt), ex.:
  - Few balls which have different velocity and moving angles. In order to get max points, add balls with mouse, make balls to change color on interaction and any other things that will show your engineering spirit
  - Any other interesting and reach in animation application
- Animate a Nyan Cat that leaves a rainbow tail (Math.floor(+35% for task with interacting objects))

3 Laboratory work implementation

The objects are devided into two big types: static and dynamic figures. The static figures are rectangles, squares, ellipses. They are the bareers for dynamic objects.

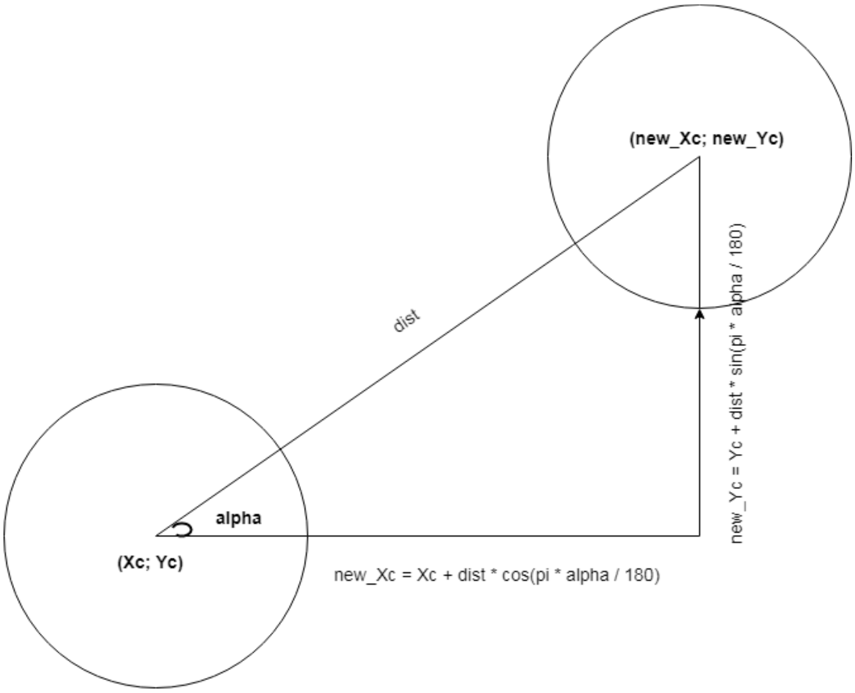
The dynamic objects are squares. They have different size and color. Interacting with static objects, dynamic objects change their angle of direction.


circle.h [Lab4] - Code::Blocks 13.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
PPE lab4 Ganusceac Vlad
Элементов: 15 шт.
22:28
09.04.2018

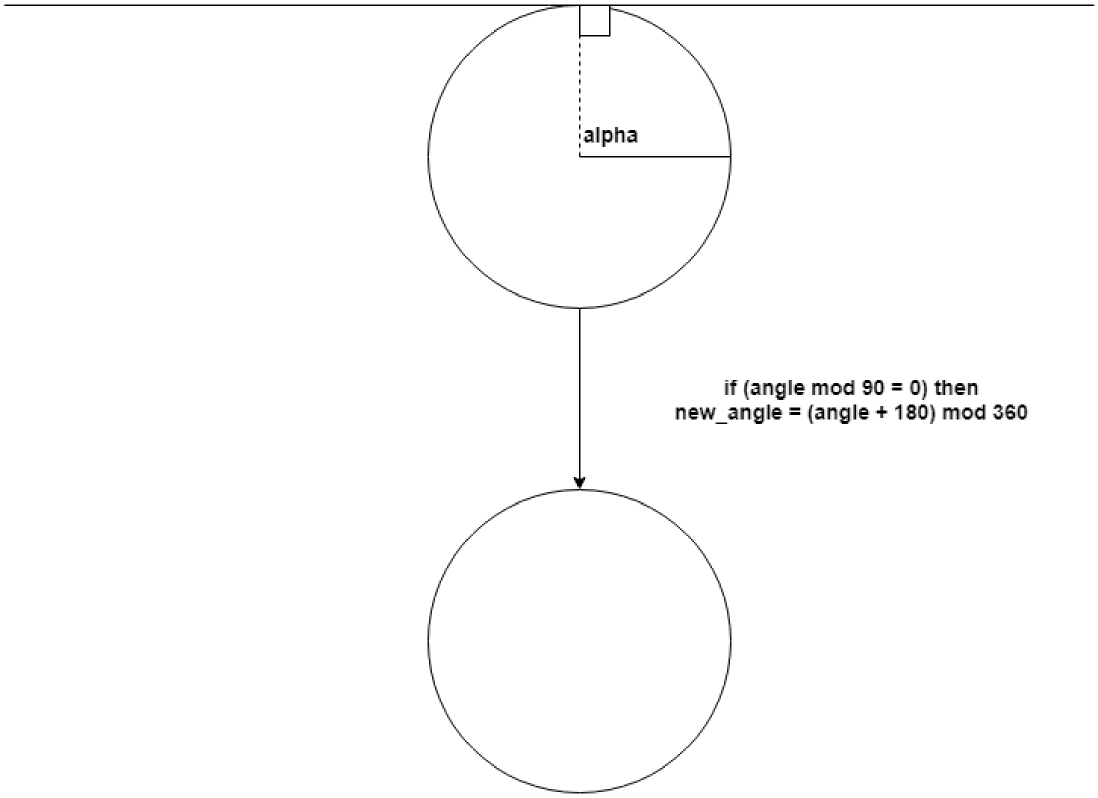

Документ1 - Microsoft Word (Сбой активации продукта)
Файл Главная Вставка Разметка страницы Ссылки Рассылки Рецензирование Вид
PPE lab4 Ganusceac Vlad
Страница: 3 из 4 Число слов: 186 английский (США)
22:28
09.04.2018

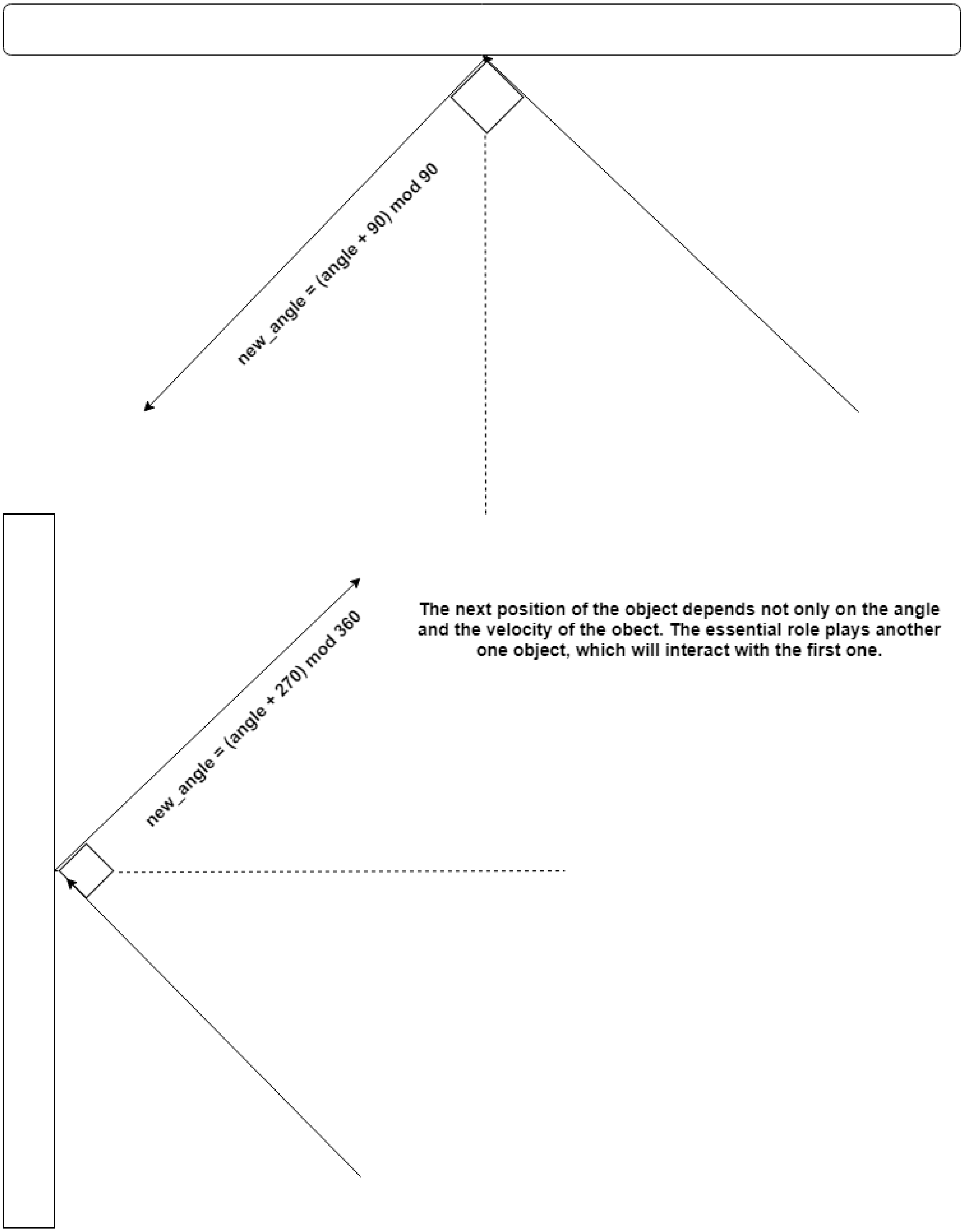
In order to implement all this it is necessary to do the trigonometric calculations for each case!

This is how the new position of the dynamic object should be counted:


(new_Xc; new_Yc)
dist
new_Yc = Yc + dist * sin(pi * alpha / 180)
alpha
(Xc; Yc)
new_Xc = Xc + dist * cos(pi * alpha / 180)

Angle, velocity and bareers also does matter in calculations:




new_angle = (angle + 90) mod 90
new_angle = (angle + 270) mod 360

The next position of the object depends not only on the angle and the velocity of the obect. The essential role plays another one object, which will interact with the first one.

Conclusion:

In this laboratory work we used Windows Timer in order to do animation. The animation is based on trigonometrics, geometry and arithmetics. All calculations are calculated in the equal periods of types (0.3 s), because we don’t want to see many flickers (also we use double buffering in WM_TIMER) during the execution of the application! The user may increase or decrease the velocity of the circle objects with help of mousewheel.

References:

Programming Windows by Charlez Petzold, 5th edition: (Section I, Chapters 5, 7, 8)

http://www.frolov-lib.ru/books/bsp/v14/ch2_3.htm (Графический интерфейс GDI в Microsoft Windows)

http://radiofront.narod.ru/htm/prog/htm/winda/api/paint.html#10 (Как рисовать в Windows)
https://msdn.microsoft.com/en-us/library/ms969905 (how to avoid flickers)