

Faculty of Computers, Informatics and Microelectronics  
Technical University of Moldova

Event-Driven Programming  
Laboratory work #6

Authors:

Ganusceac Vlad

Supervisor:

Mihai Coslet

# Laboratory work #6

## 1 Purpose of the laboratory

Developing using C#. WinRT.

## 2 Laboratory Work Requirements

### Contents

---

- C#
- WinRT

### Mandatory Objectives

---

- Create an Win32 or WinRT application using C#
- Choose one of:
  - Convert a previous laboratory work to C#
  - Create a ToDo list. Should contain:
    - A list of tasks
    - A way to add new tasks from UI

### Objectives With Points

---

- Create a report (3pt)
- Use WinRT (3pt)
- Create a pull request with a meaningful fix/feature to 5th WP laboratory work of your colleagues. It should be a project in which you were not involved. (4pt)

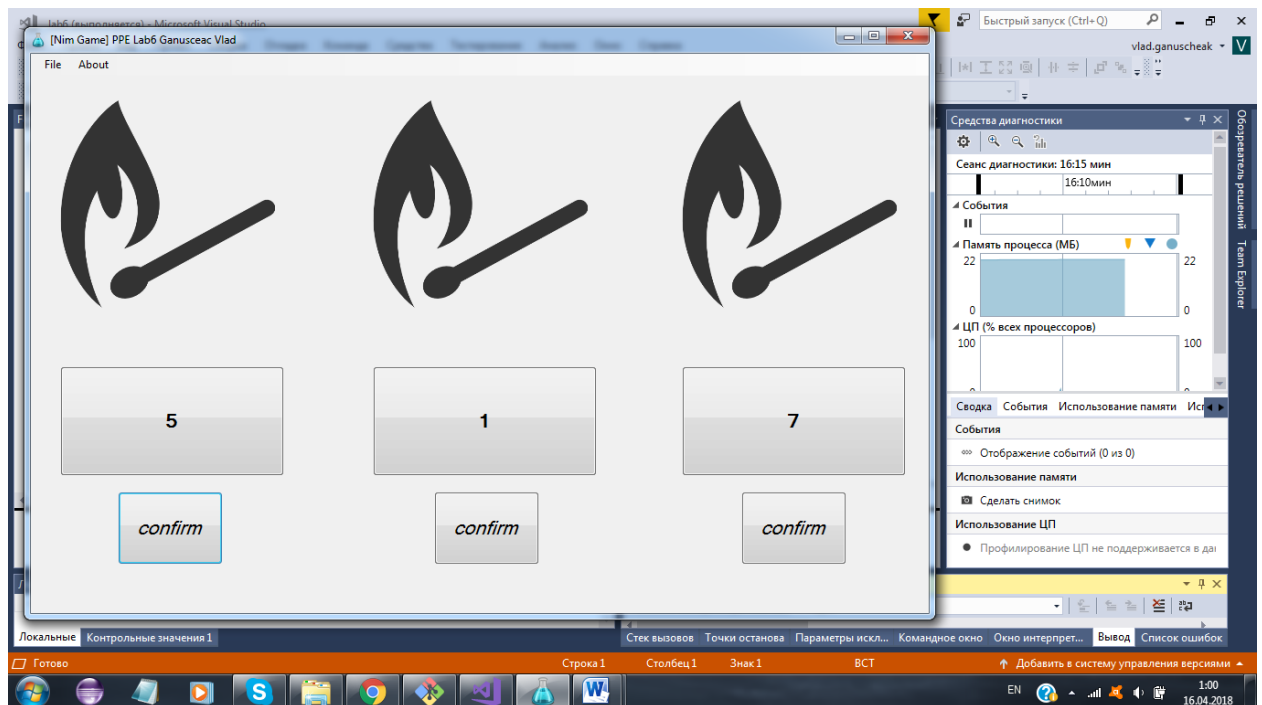
## **ToDo List:**

- 1) **Inform the player of rules of the game (Nim):**
  - The name of the game should represent it's essence (Nim)
  - In main menu should be option MENU -> ABOUT -> GAME where the laws of the game are described;
- 2) **Inform the user about author of the application: MENU -> ABOUT -> AUTHOR**
- 3) **Create several ways to end the application: clicking on the X on the right top corner, clicking on MENU -> FILE -> EXIT etc.**
- 4) **Generate randomly the number of matches in each pile:**
  - The total number of matches should be greater than 5.
  - In any pile shouldn't be more than 9 matches.
- 5) **Teacher the program in the backend to play optimally (to calculate the winning strategy)**
  - If the program isn't able to put user in losing position, it should take a random number of matches from one of piles.
  - Otherwise, to do actions which are strictly linked with winning formula.
- 6) **Do imitation of thinking process (using windows timer, for example 4 seconds).**
- 7) **Calculate different ending scenarios:**
  - Player have won the game
  - Player have lost the game

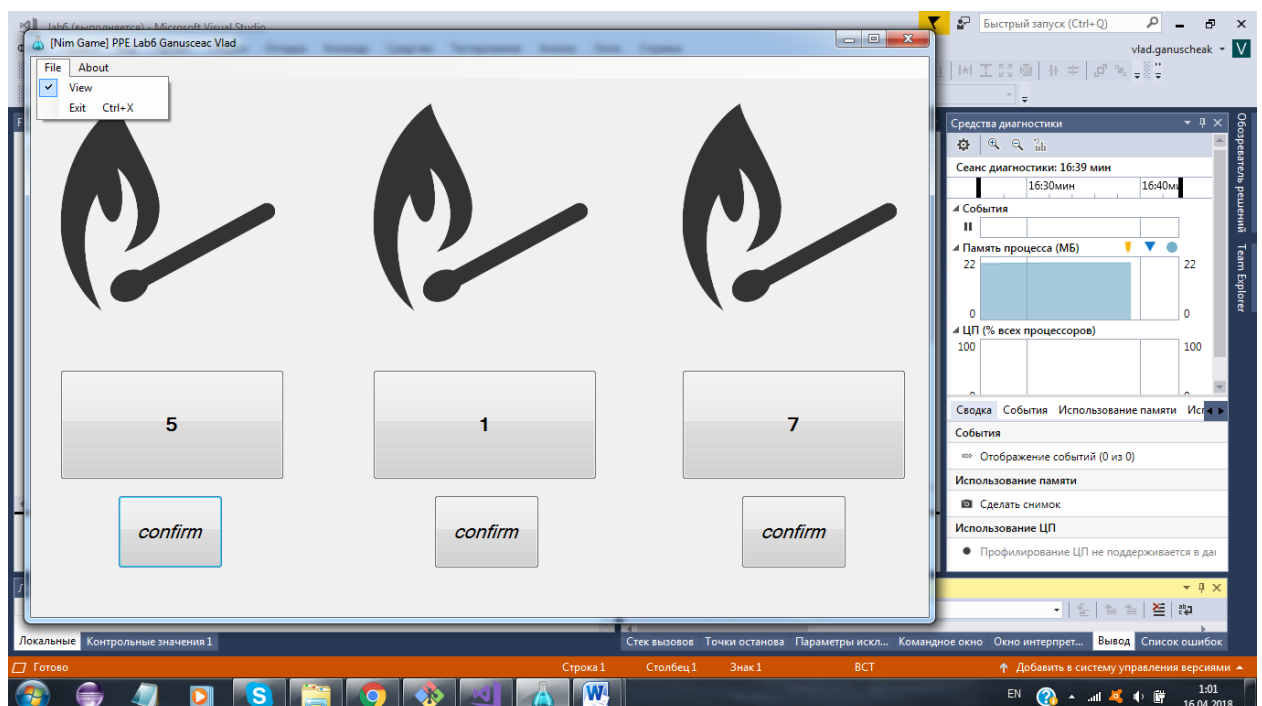
**and to inform user about it.**

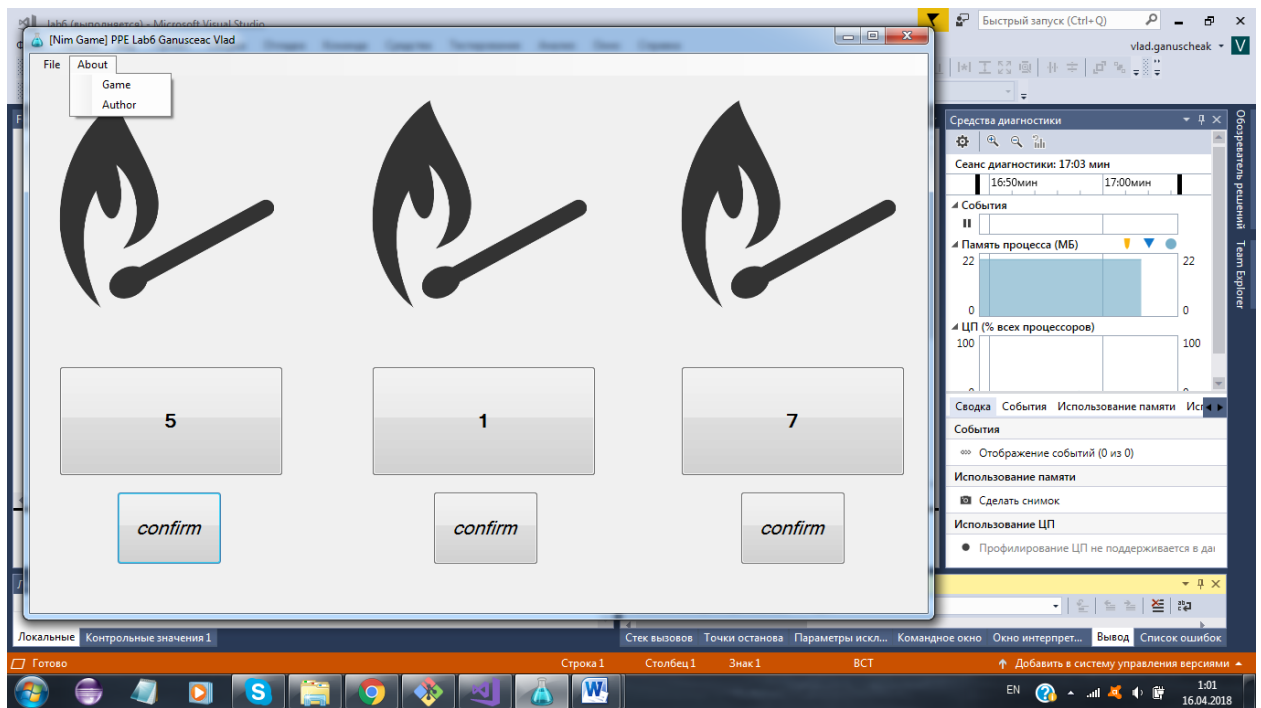
### 3 Laboratory work implementation

This is how the UI is looking in this application:

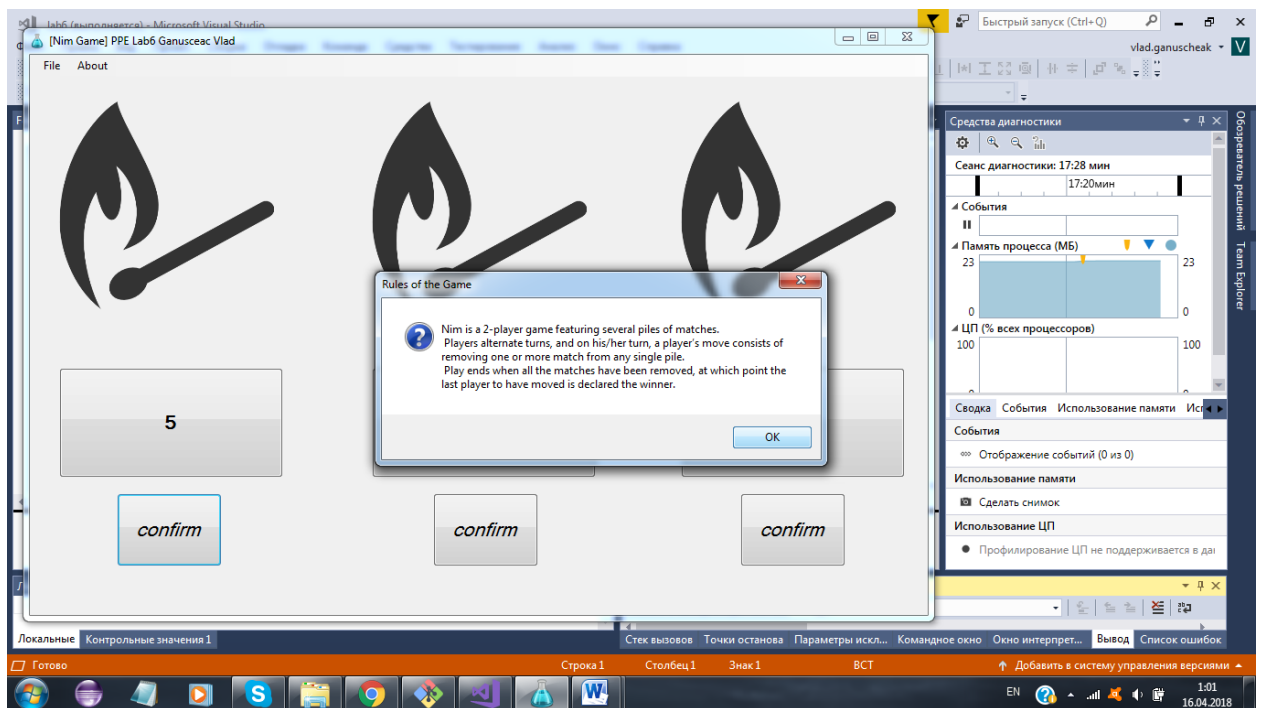


There are a list of options which the user can do using the main menu:

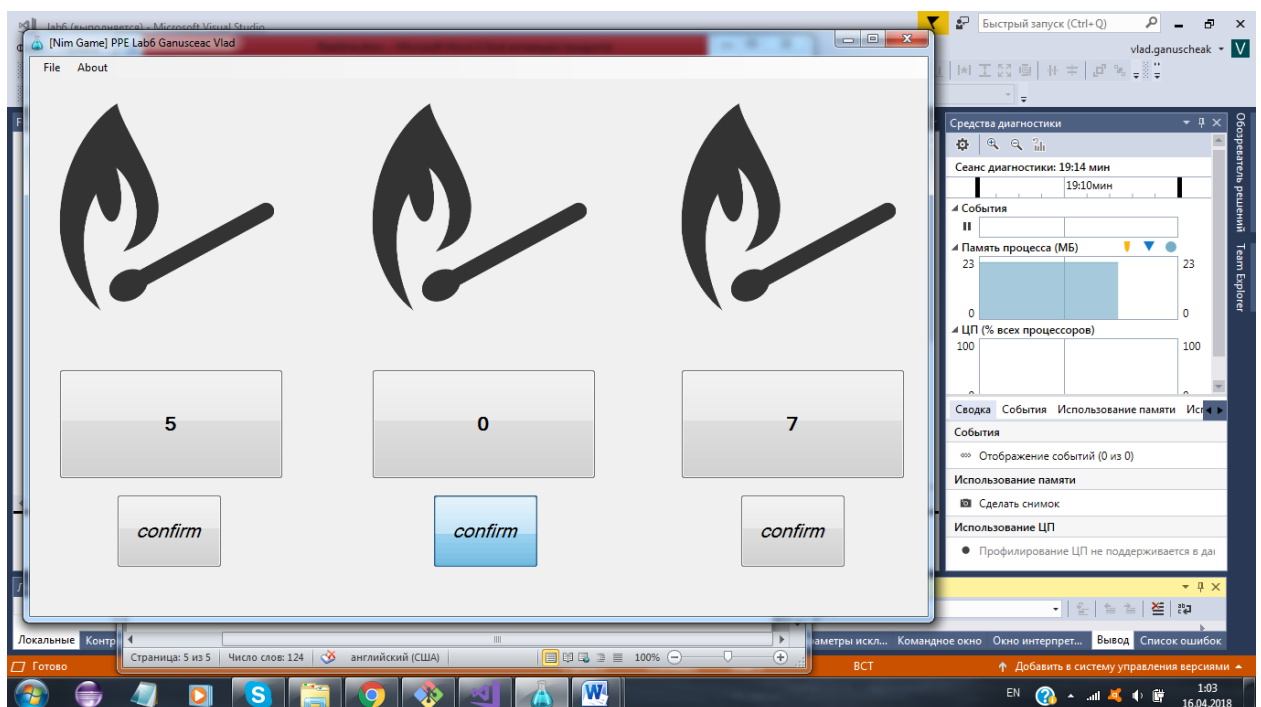
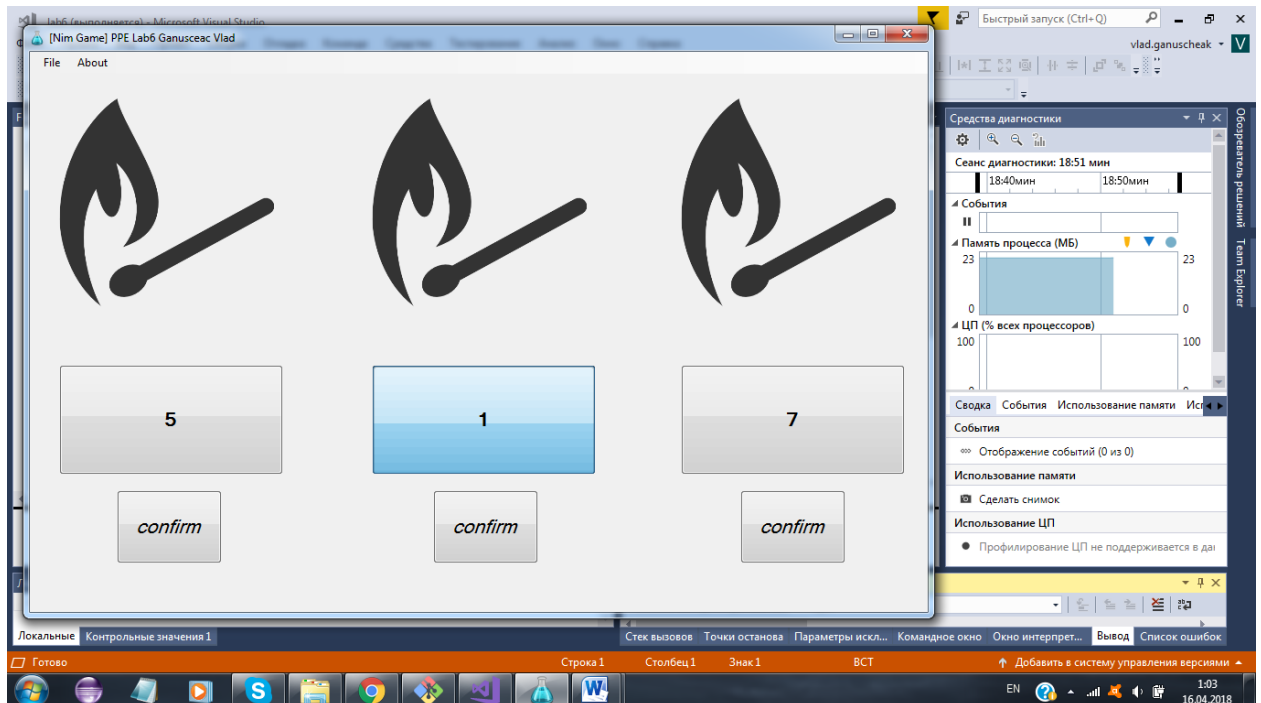


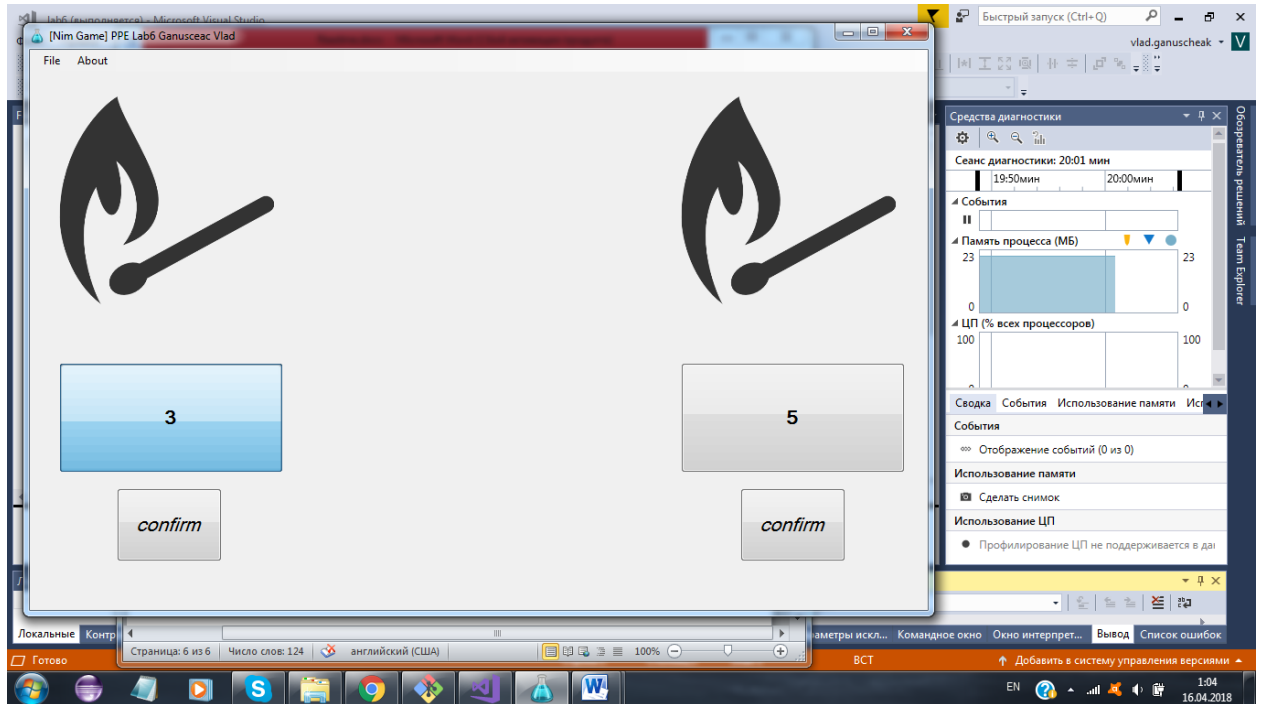
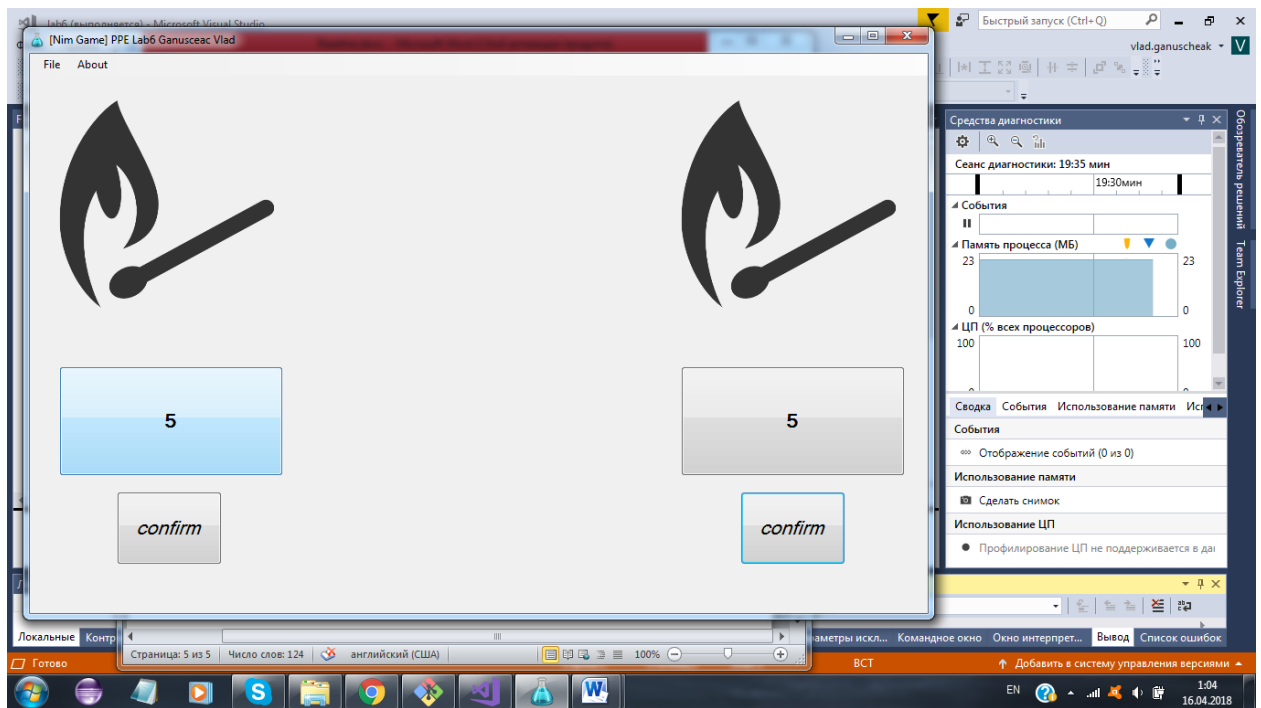


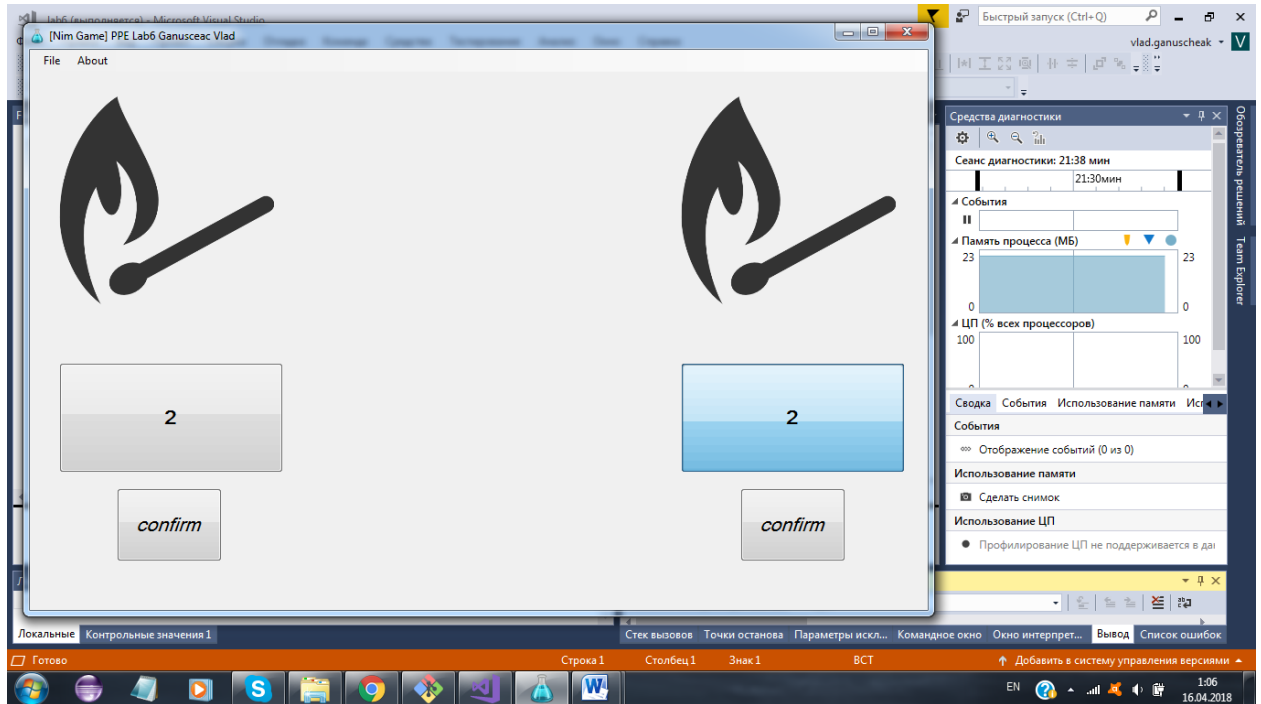
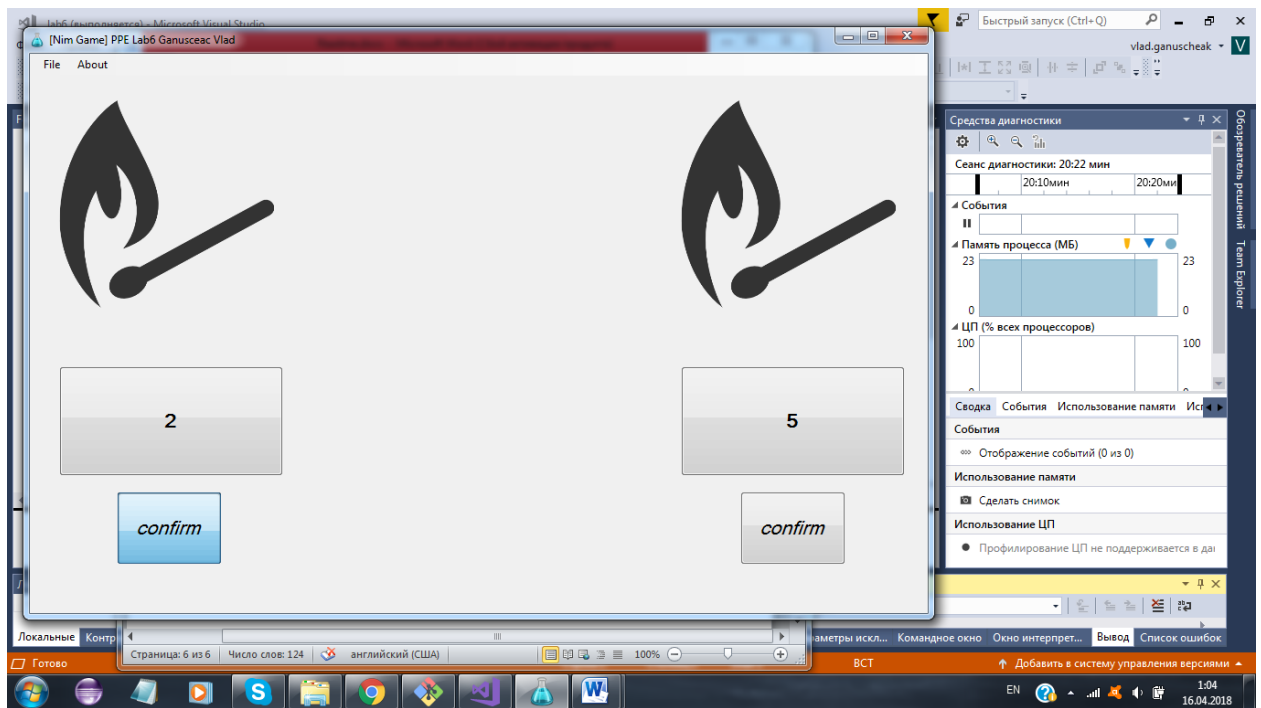
The user is able to see the rules of the game Nim:



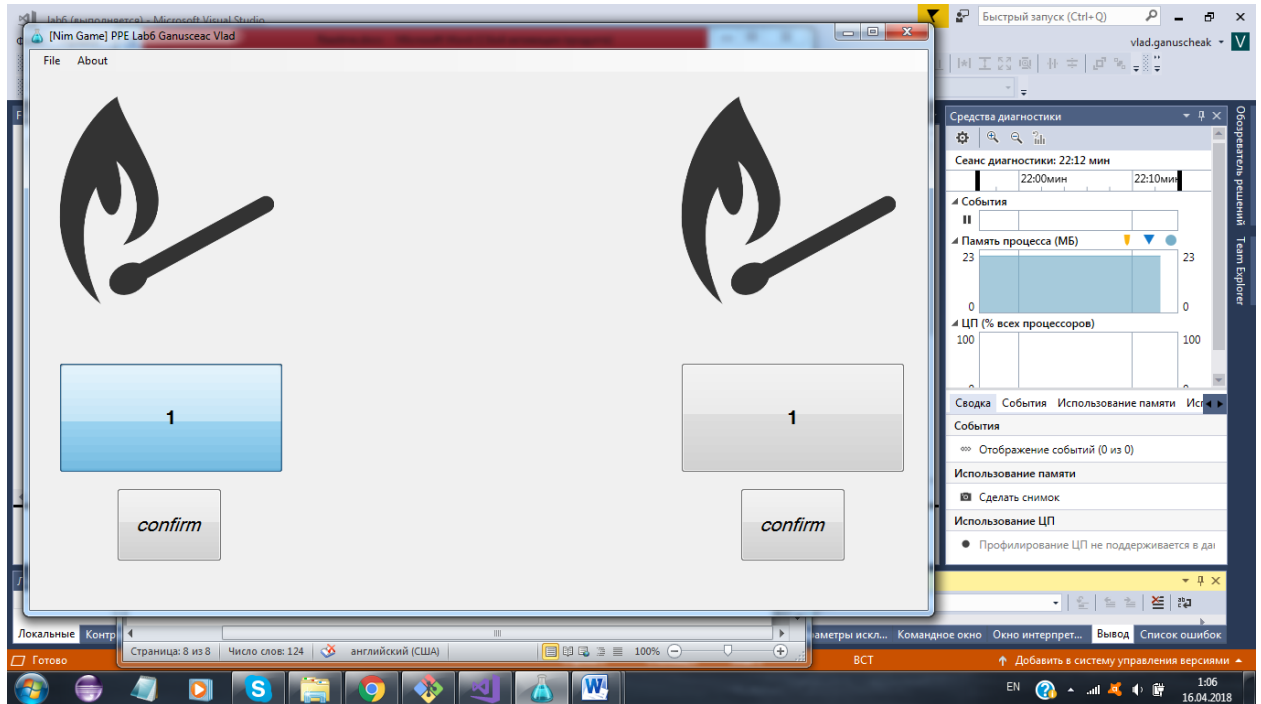
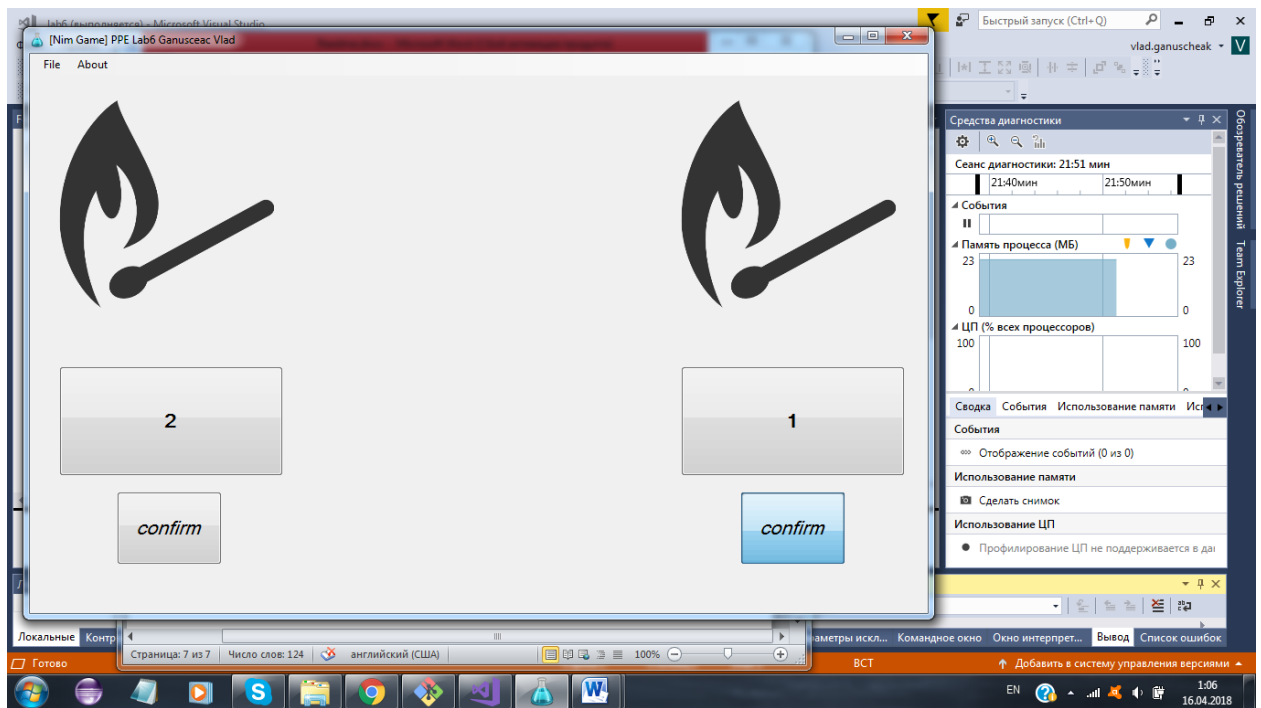
One of scenarios of this game. The user got the set of 5, 1 and 7 matches (3 piles). He/she is able to take from 1 to `NUMBER_OF_MATCHES_IN_A_PILE` matches. In other words, the player should click the number of the matches in a pile which will remain after his/her turn.



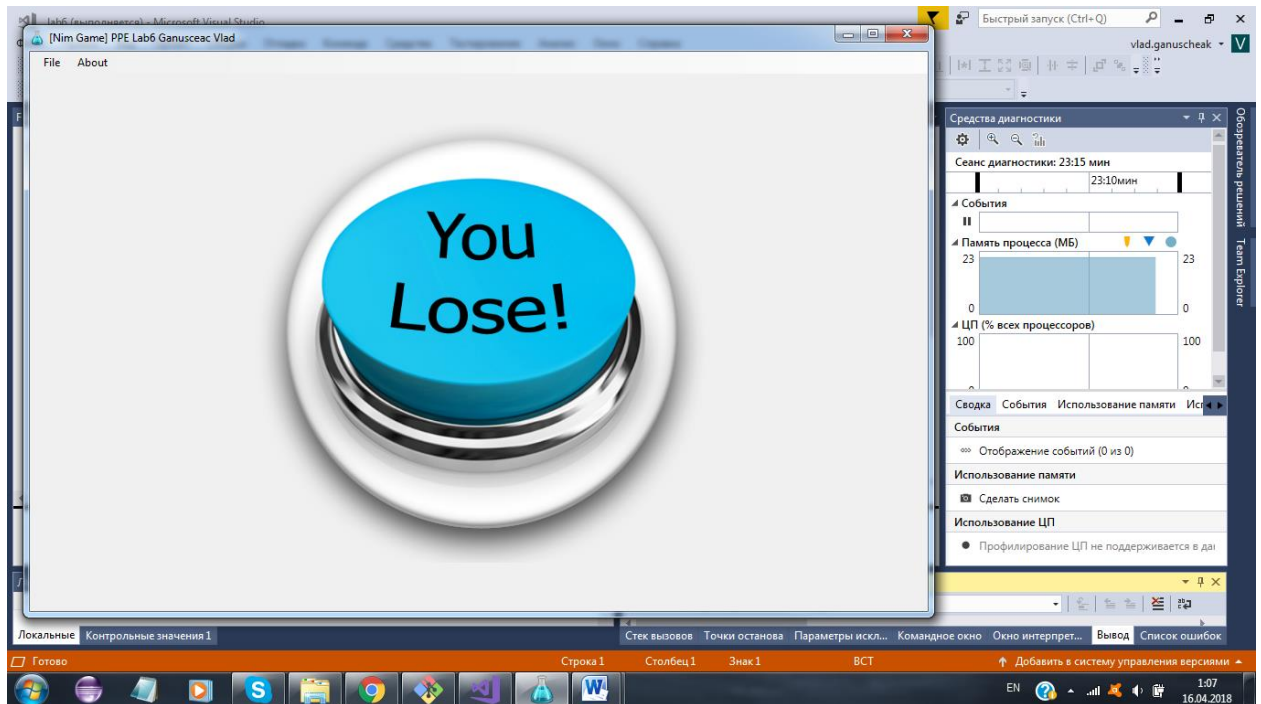
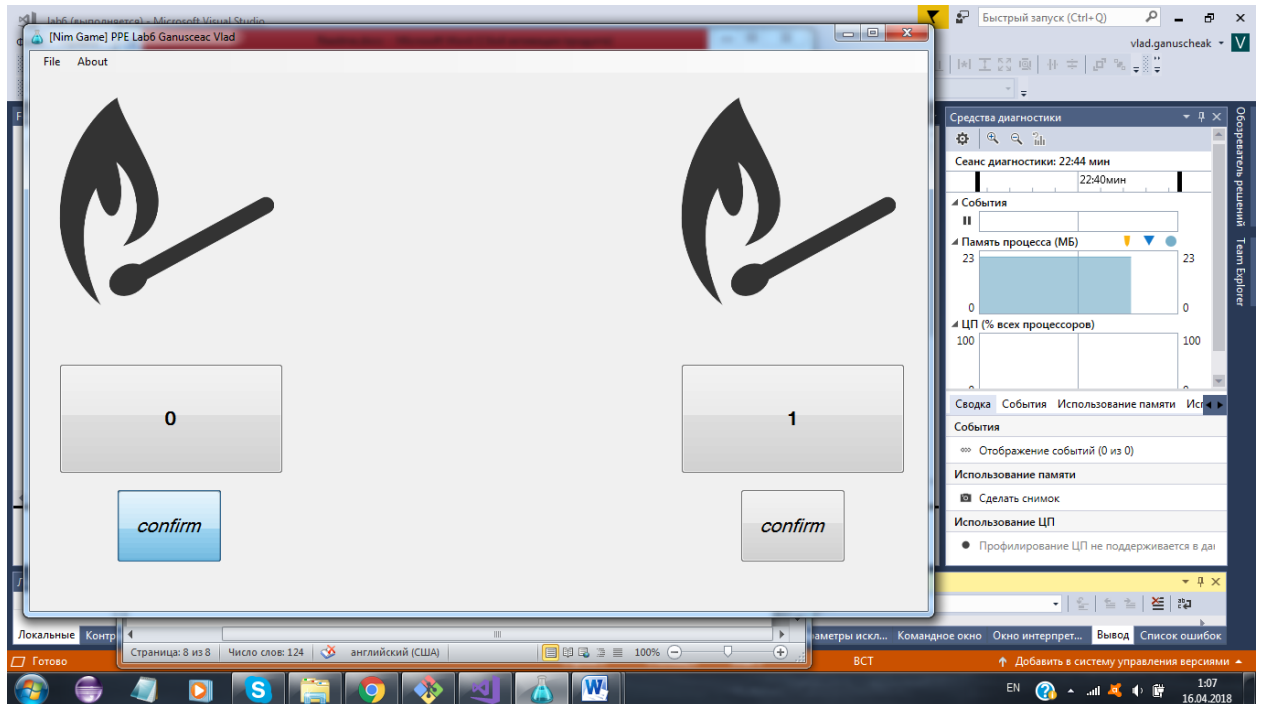








Now it is absolutely clear that the user will lose, because the computer will take the last match!



**Conclusion:**

In this laboratory work we wrote a program in c# using WinRT. We have implemented all points from ToDo List: from creating the main menu with all its options to establishing the winning strategy to computer in this game. The user will have idea that the computer is thinking while it is doing its moves. Different objects interact with each other during the multiple events (for example, user's inputs). The application is useful, because its implementation is linked with such domain of the special mathematics as Game Theory.

**References:**

A set of 4 articles for creating a ToDo list in WPF:

<https://www.aspfree.com/c/a/windows-scripting/wpf-through-an-example-introduction/>

<https://www.aspfree.com/c/a/windows-scripting/adding-controls-to-an-application-with-wpf/>

<https://www.aspfree.com/c/a/windows-scripting/data-converstion-and-task-addition-with-wpf/>

<https://www.aspfree.com/c/a/windows-scripting/completing-a-wpf-to-do-list-application/>

A ToDo app example:

<https://github.com/Mellen/To-Do-List>