Exercise 5 – American Checkers for Windows

Objective

- Developing a basic yet complex Windows desktop application using .NET WinForms
- Working with controls, forms, and events

Prior Knowledge

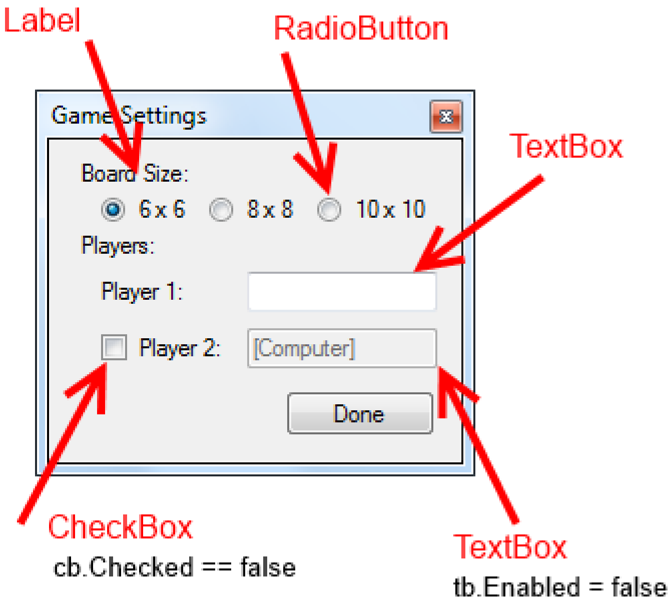
- Using .NET WinForms to develop multi-form Windows desktop application
- Working with basic controls and their events

The Exercise

Implementing the game American Checkers again, but this time with a Window GUI!

The Program:

The first form will let the user configure the game:



Game Settings

Board Size:

● 6 x 6 ○ 8 x 8 ○ 10 x 10

Players:

Player 1:

□ Player 2: [Computer]

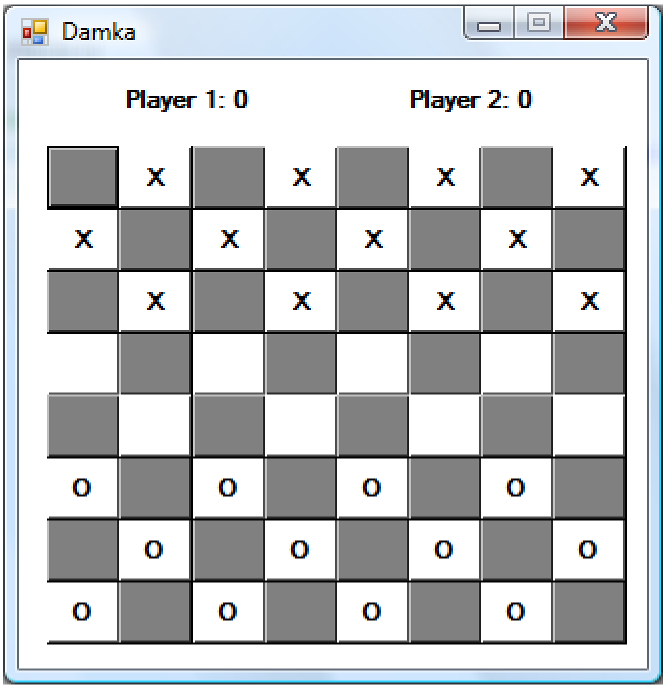
Done

Checking he "Player 2" checkbox will make the second TextBox enabled, and thuse letting the user to set the name of the second player.

Clicking "Done" will close the "Game Properties" form, and the main Damka form will be displayed, according to the configuration set in the properties form.

Note:

Not like in this sample screenshot, the 'X's will be placed in the lower side of the board and the 'O's in the upper side, like in Assignment 2.

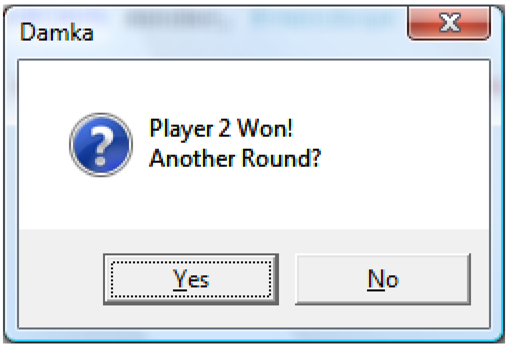


Player 1: 0 Player 2: 0

▨	X	▨	X	▨	X	▨	X
X	▨	X	▨	X	▨	X	▨
▨	X	▨	X	▨	X	▨	X
	▨		▨		▨		▨
▨		▨		▨		▨	
O	▨	O	▨	O	▨	O	▨
▨	O	▨	O	▨	O	▨	O
O	▨	O	▨	O	▨	O	▨

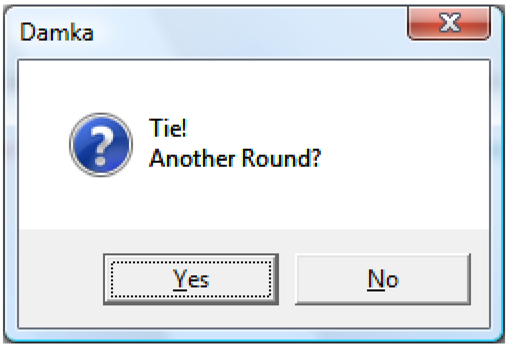
- The controls on the top that display the score are of type Lable.
- The buttons matrix is built according to the settings in the properties form
- The size of the form will be set according to the board size.
- The gray buttons, which represents blank spots in the board to which you can not move, will be disabled (`button.Enabled = false;`)
- **The first click on button with a checker, will change its BackColor to LightBlue**
- A second click on that button (a click on a LighBlue button) will de-select it and will reset its color to White.
- When there is a 'selected' (LightBlue) button, a click on an empty white button in the board, will:
  - Move the checker to the desired empty spot, if it is a legal move
  - Display a Message Box (MessageBox.Show("…")) with a relevant "Illegal Move" message.

- When one of the player wins, a message will be displayed:



Player 2 Won!
Another Round?

- When a Tie is achived, a message will be displayed:



Tie!
Another Round?

- Clicking "No" will end the execution of the application.
- Clicking "Yes" will reset the board, udate the score, and a new round will start.

Note:

- The click events raised by the buttons pass a 'sender' object as the first parameter, which references the button which raised the event. You're advised to use this parameter to avoid code duplication.
- You must start with an Empty Project just like demonstrated in the lecture.
- A bonus of up to 5 points will be given for implementing events raised from the Logic Layer and handled in the UI layer for updating the display (For example: when a certain cell in the board changes its content, it will notify about that. The relevant UI element will be the Listener of this notification and will update its display content accordingly).
- A bonus of up to 5 points will be given for implementing a more "rich" UI in terms of graphics / animations / movements. Please describe your unique implementation if relevant, in the 'comments' section of your submission email.

General Instructions

- You may use in the course's facebook group in order to ask questions regarding this assignemt.
- You must comply with the coding standards, as stated in the relevant document, found on the course website. Points will be deducted to whom ever does not comply with these standards.
- Send your submission to the email address as described in the "Ex_Submit_Instructions_DN_IDC_22B.pdf" document, which can be found in the course's site. Points will be deducted for not following the instructions carefully.
- Avoid cheating (Do not use other students assignments as a basis for yours. Refrain from copying the work of fellow students from your group or previous semesters. Cheaters will be caught and punished. Work independently!)
- Submission is due to June 21st 2022, 22:00.

Good Luck ☺