Exercise 2 – Memory Game for Console

Objective

- Using Classes for implementing Object Oriented Programing concepts
- Using Constructors, Enums, Properties, Access modifiers, Modifiers
- Working with Arrays / Collections / Data Structures
- Use of the String Class
- Referencing External DII (Assembly)

Prior Knowledge

- Acquaintance with Microsoft Visual Studio .NET
- Basic C# Syntax knowledge
- Working with Arrays / Collections / Data Structures
- Working with Classes (Access modifiers, Constructors, Properties)
- Use of the String Class
- Referencing External DII (Assembly)

Pre – Preparing

- Microsoft Visual Studio installed
- Ex02.ConsoleUtils.dll file (part of the download package – Moodle)

The Exercise

You must implement the Memory Game for Console.

The Program:

The program will allow two human players to play against eachother by turns, or for a human player to play against the computer.

The program will display an empty board of rows X columns (even number of cells):

```
    A   B   C   D   E   F
  =========================
1 |   |   |   |   |   |   |
  =========================
2 |   |   |   |   |   |   |
  =========================
3 |   |   |   |   |   |   |
  =========================
4 |   |   |   |   |   |   |
  =========================
```

In each step, the user will be asked to choose which cell to expose (each cell contains a letter which is not displayed). For example, this is what will be displayed to the user after choosing to expose cell E3 that contain the letter 'T':

```
    A   B   C   D   E   F
  =========================
1 |   |   |   |   |   |   |
  =========================
2 |   |   |   |   |   |   |
  =========================
3 |   |   |   |   | T |   |
  =========================
4 |   |   |   |   |   |   |
  =========================
```

At this point, the user will be asked to choose another cell to expose, in which he hopes to find the other 'T' (the board contains pairs of identical letters).

The flow:

1. The user will be asked to enter his name (no spaces, max 20 chars).
2. The user will be asked to choose weather to play against a computer upponent or another human upponent. In case of another human uponent, the user will be asked to enter the upponents name (no spaces, max 20 chars).
3. The user will be asked to choose board size (minimum 4x4, maximum 6x6). The board must contain even number of cells (5x5 is illegal)
4. The game starts with an initialize board (6x4 example):

```
   A   B   C   D   E   F
 =========================
1|   |   |   |   |   |   |
 =========================
2|   |   |   |   |   |   |
 =========================
3|   |   |   |   |   |   |
 =========================
4|   |   |   |   |   |   |
 =========================

Dani's turn:
```

5. At each stage, the player will be asked to enter his 'choice'.
   If the choice is 'illegal', the use will be prompted to enter a valid move, and so on until he chooses a legal choice (a legal choice is an empty cell in the board. Any other input is invalid and a proper message should be displayed to the user.
6. After a valid input was entered by the user, the screen will be cleared and the board will be re-drawn with the selected cell 'exposed':

```
   A   B   C   D   E   F
 =========================
1|   |   |   |   |   |   |
 =========================
2|   |   |   |   |   |   |
 =========================
3|   |   |   |   | T |   |
 =========================
4|   |   |   |   |   |   |
 =========================
```

7. The player will be asked to enter his next 'choice' in which he wishes to find a pair.
8. If the player revelaed a matching pair, the pair will remain exposed till the end of the game, the player will gain a point, and the turn will remain his to trye to find another turn (back to stage 5).
9. If the player revelaed an un-matching pair, the pair will remain exposed for 2 seconds, after which the cells will become 'hidden' again and the turn will move to the opponent (back to stage 5).
10. When the last pair is exposed (all pairs are exposed) is 'Game Over' and the winner (who ever got most points) will be declared including the score.
11. The user will be ask to decide if he wants to play another round or to quit.
12. Starting stage 5, the user can quit the app at any point by typing 'Q' as input.

General Instructions

- You may use the Next method of class Random for randomization (to set up the board and to produce computer-based 'choices')
- You may use the Sleep method of class System.Threading.Thread in order to implement the 2 seconds wait.
- You must validate each input and prompt the user in case of any invalid input. You must differentiate between **syntax-invalid** inputs and **logical-invalid** inputs:
    - Syntax-invalid: inputing a number instead of a letter
    - Logical-invalid: inputing an out of range letter (i.e. Y)
- **Before printing each stage of the game board, you must clear the screen. For this, use the file Ex02.ConsoleUtils.dll, which contains the service class Ex02.ConsoleUtilts.Screen and use the static method Clear().**
  To use this dll:

- Right click in the Solution Explorer window, above the project's Refrences.
- Pick Add Reference and then choose the dll using Browse.
- Now, you are able to access the Ex02.ConsoleUtils class (which is defined in the dll) and use the methods of that class as you'd use any other normal library methods.
  Important – Do not attach this file in your submission – GMAIL will reject your work
  (The grader has the file in the same location)

- **Architecture and Software Engineering:**
    - You must use Object Oriented architecture!
    - You must demonstrate correct use of C# 2.0 and .Net capabilities
    - **You must apply correct separation and segregation between the Classes who manage the logics and data of the game and the Classes who are responsible for the UI and User Interactions.**
      In other words: Seperation between the implementation of the User Interface (UI) and the implementation of the system's logic.
      You must keep in mind that these implementations will serve you when you will want to develop the game with an alternative UI in the future, and the aspiration is to have as many reusable parts (unchanged) as possible.
- Implemeting AI-based computer 'choices' (instead of just random choices) will gain up to 6 bonus points.
- No need for XML documentation.
- 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. **Pay extra attention to the standards of Class Field Names and Function Parameters**. Points will be deducted to whom ever does not comply with these standards.
- 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 Jue 10th 2024, 22:00.

Good Luck ☺