EGG202

Lab 7: Memory Game

Description

For this lab, you will develop a windows form application that implements the memory game. The way this game works is you have a set of cards flipped on its back, you turn a card over, and then turn another card over, if they match you remove them from from the game, otherwise you turn them back over and try again, so the idea is you have to try and remember where the cards were to try to win the game (you want to use as few moves as possible, but as long as you win the game, you're fine). I have given you a set of cards that you can use to display the game, you will need to have a set of `PictureBox` objects on your form to display the cards and implement a click event for them.

Form Objects

The following objects will be on the form

- A start new game button, when clicked shuffles the cards and displays them flipped backwards
- A checkbox, that allows you to run in debug mode (debug mode will have the cards in designated locations), otherwise they will be random
- 20 ImageBoxes (for each card and they all will call the same click event)

Recommended Variables

You can have the following variables to help you implement the game

- `PictureBox[,] cards = new PictureBox[4, 5]` - an array that contains a reference to each image box object on the form, you can cycle through this array to set the cards visible, disabled, etc, each `PictureBox` object on the form should be named using the following naming convention `pBoxij`, where `i` would be a number between 1 and 4, and `j` would be a number between 1 and 5, this allows you to easily determine which `PictureBox` object `card[, j]` is referring to
- `bool[,] cardTurned = new bool[4, 5]` - denotes if a card is turned over showing its face or its back is showing, thus if `cardTurned[i, j]` contains `true`, then you want `cards[i, j].image` would be set to `Properties.Resources.CardName` (based on the id stored in `cardId[i, j]`), and if `cardTurned[, j]` contains `false`, then `card[i, j].image` would be set to `Properties.Resources.redBack`
- `int[,] cardId = new int[4, 5]` - contains a card id for a card at position `[, j]`, you have cards 2S, 3S, 4S, 5S, 6S,10S, JS, QS, KS, AS so each card has an id from 1 to 10 for each card, so for example if `cardId[i, j]` contains 1 then `cards[i, j].image` would be set to `Properties.Resources._2S` which is the image for 2S (2 of spades)
- `int matches` - amount of matches so far, once this amount is 10, then the player wins

- `bool turn` - denotes whether you drew the first card or you're drawing the second card, so if `turn` contains `true` then the player is selecting the first card in the turn, and if `turn` contains `false`, then the player is in the process of selecting the second card, `turn` alternates accordingly
- `int lastTurnedX, lastTurnedY` - stores the `i` and `j` indices of the card that is currently flipped (showing its face), obviously we only care what's stored in these variables when `turn` contains `false`

Event Function Details

You will need to implement the following event functions

- `private void btnNewGame_Click(object sender, EventArgs e)` - initializes a new game, checks if the check box object is selected or not, if selected place the cards in designated areas (non random), else randomize them, set the `card[, j]` with a number from 1 to 10, this designates a card in a spot on the board
- `private async void Clicked_On_Card(object sender, EventArgs)` - turns the card over (reveals the card based on the id stored in `cardId[i, j]`), and sets `cardTurned[i, j]` to `true`, and then
  - if `turn` contains `true`, change it to `false`, and store the coordinates of this image into `lastTurnedX` and `lastTurnedY`
  - `turn` is set to `false`, then change it to `true`, pause the program for two seconds using

    ```
    await Task.Delay(2000)
    ```

    and disable all the PictureBoxes (so nothing can be activated during this pause), then check the `cardId` at index `lastTurnedX, lastTurnedY]` if it matches `cardId[i, j]` then set both PictureBox object's visible fields to `false`, else turn them both over (set both of their image fields to `Properties.Resource.redBack`) and set `cardTurned[i, j]` and `cardTurned[lastTurnedX, lastTurnedY]` to `false`
- `private void Form1_Load(object sender, EventArgs e)` - sets all the `PictureBox` objects into an `PictureBox` array, so each `PictureBox` can be accessed through the array, set default values to any variables
- You may have any additional functions in the program as needed to help with this

Submission

Compress your project files into a zip file and upload to the canvas site by the deadline