Introducing a Controller Class

Look at the code for your *Slot Machine* application that you built in **05-multiple-classes.** Notice that we created a *Spinner* class to look after the spinning functionality. However, when you look at the *Form 1* class, there is a lot of code that is concerned with the game play and it is not concerned with the interaction with the user. We need to separate these two very different aspects of our code, by introducing a new class called the *Controller* class. (Some people would call this a *Game Engine* class, either term is acceptable).

To the user, today's *Slot Machine* will look and operate in exactly the same way as it did in **05-multiple-classes**. The changes we will make are to do with how we organise our code.

Implementation Suggestions:

1. **Save** the *SlotMachine* folder and name it *SlotMachinewithController.*

2. **Open** your new *SlotMachinewithController* application.

   The *Spinner* class will remain as before. The *Spinner* class correctly looks after just the spinning of each of the three spinners**.**

3. **Make a Controller class.**
   It will need two ***fields,*** an array to hold the three spinner objects and an integer for the winnings.

   In the ***constructor*** create the three spinner objects and set the winnings to the starting amount of $100.

   There should be three ***methods:***
   CheckEnoughMoney*()* which checks whether there is at least $10 in the winnings, and returns either true or false**.**
   RunGame() which iterates through the spinners array, spinning each spinner 20 times. Remember to add the *DoEvents* and *Sleep* statements here.
   CheckForWinner() which checks to see if the three image numbers match. If so, winnings is incremented, and true or false is returned from the method.

   Create a ***property*** for the *winnings* field, so that it can be accessed from Form 1.

   Make **constants** for the amount you start to play with ($100), the cost of each round ($10), the amount you can win ($50), the sleep interval (100 ms), the number of spins (20) and the number of spinners (3).

4. **Refactoring the Form1 class.**
   In the *Form1* class, create an **array** of *PictureBox* objects. This will make it easier to pass the three *PictureBoxes* across to the *Controller* class.

   Create a ***random* object** that is also passed to Controller class.

Create a ***controller* object** that takes a reference to the *random* object and the *pictureBoxes* array in its constructor.

In the ***button1_Click()* event,** ask the controller whether the user has enough money to play, and if so tell the controller to run the game. Ask the controller if there is winner. At each step, give appropriate feedback to the user.

5. **Comment your code.**
   This is important - so that others can understand what you have written and also to remind yourself next week of the new ideas that have been introduced today.