

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**IX511001 Programming 2**

**CHECKPOINT 4- Looney Tune Horse Race**

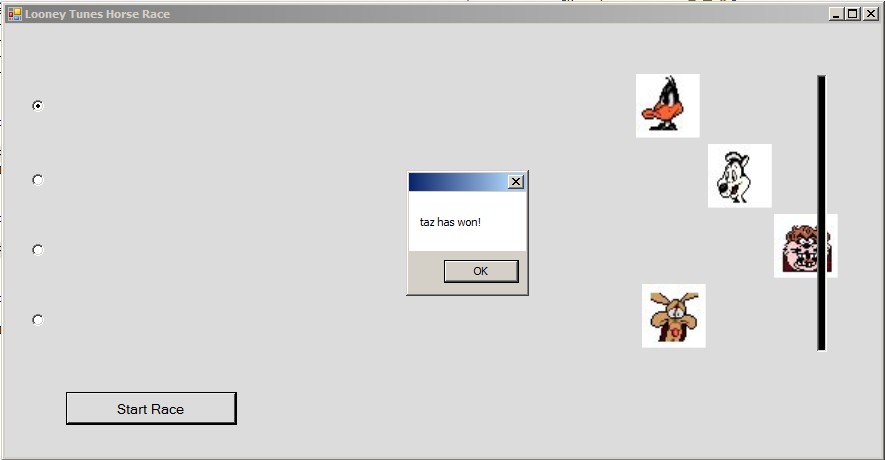
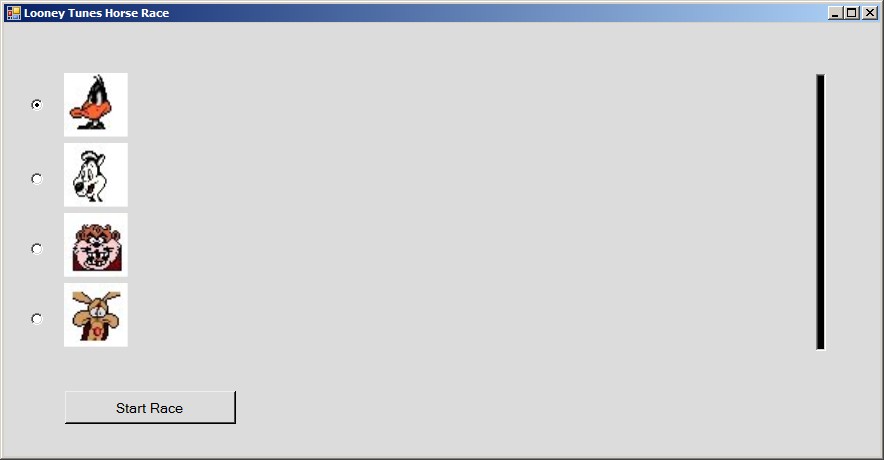
Due Week 5

Weighting/Contribution: 2.5% to final marks

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## Looney Tune Horse Race

1. In this task you are going to build a horse race game. The four “horses” are images loaded into *PictureBox* controls (use image files available on Moodle or source your own images). I have used a *Panel* control for the finish line. The winner is the first image to touch the finish line. The outcome of the race is displayed in a *Label* control. Screenshots of my implementation before and after a race are shown below:



1. **Set up the form** with four PictureBoxes, a *Start Race* Button, a Panel for the finish line and a Timer. The Radio Buttons are optional (see #5), and used for user to bet on a winner.
2. **Start by designing the classes.** You should implement two classes: Horse and Controller.

The **Horse class** needs to know its name, its PictureBox which has been loaded with its image at design time, a reference to the Random object that is created in the Form, and an integer field for the left hand side of the finish line Panel.

It needs a constructor to initialise all its field values.

It needs to know how to move itself (this should be calculated as a random value between 0 and its speed), how to reset its PictureBox back to the start position. It also needs to know and whether the right hand side of its PictureBox has passed the finish line.

Will you need to create any properties for the fields?

The **Controller class** is the game engine. It controls the race, commanding each horse in turn to move itself and querying each horse to find out if it has reached the finish line.

It should hold an array of Horses, and a Boolean field that holds whether the race is over.

Its constructor should be used to create the four Horse instances and initialise the other fields with appropriate values.

It has a method to restart the Race which tells each horse to rest itself to the starting position.

The Race method should ask each horse to move itself. It should ask if each horse has reached the finish line. If so, the race should be stopped and the winner announced.

A MessageBox.Show() is a useful method to provide feedback to the user:

MessageBox.Show(horses[i].Name + " has won!");

The CheckForWinner() method ask each horse if it has reached the finish line. Instead of just returning a Boolean to say that a winner has been found, it is preferable to return only the Horse that has won the race.

1. The **Form1 class** is the container for the controls and should be used only as the user interface.

Create an array to hold the four PictureBoxes, a Controller object and a Random object. Initialise the Timer’s enabled to false in the Form’s constructor.

The *StartRace* button will instruct the controller to restart the race and enable the timer.

On each Timer Tick, the controller object should check whether the race is over (in which case it should disable the timer) or if the race is still on (then call its Race() method).

1. **Refining your code.** Go through your code checking that all literal values are replaced with constants. Check for repetitious code, is there a need for an array? What happens if two horses win the race?
2. Add a radio button beside each image’s starting position. The user bets on one of the “horses” by clicking on the corresponding radio button. Inform the user (using *MessageBox.Show* or a Label) whether his choice won. To get really fancy, start the user with some money and allow him to state how much he bets. Keep track of his winnings. You can even offer the player odds on his bet.

**Submission:**

You must submit your **application source code** (Compress your project folder containing all of the project files with) and **Microsoft Word or pdf file** (with screenshots) through Moodle submission link.