Class Assignment – Hare and Tortoise Part B:

1 Introduction

In this part of the assignment you will continue to develop the **Player Class** as well as adding **Controls** to the right-side panel of the Form. This part of the assignment can be completed even if you have not yet finished last step of **Part A, 9. Creating GUI Game Board**

2 Changes and additional instance variables in Player.cs

In order to add a **DataGridView** control to the form, it is necessary to implement the accessor and mutator methods for the **Player class** from **Part A** as C# Properties.

If in **Part A** you implemented the accessors and mutator methods of **Player class**, **GetName, SetName, etc** go to the end of this document and read the Appendix **Implementing C# Property** or review the first few slides of **Lecture 8** to change the code to use C# Property before continuing with the following paragraph.

Add two (2) additional instance variables to the **Player class,** an integer variable **money** and a boolean variable **hasWon.** Implement a Property for each of these variables containing both a **set** block and a **get** block.

3 Adding Controls to Panel2

The right-side panel of the HareAndTortoise Form has the following Controls, three Labels, a ComboBox, a DataGridView, a ListBox and three Buttons.

Refer to the screenshot on the last page of this document to see the relative layout of each control.

Add the Controls to the panel using the following points. **Do not add any Event handlers to any of the controls at this stage.**

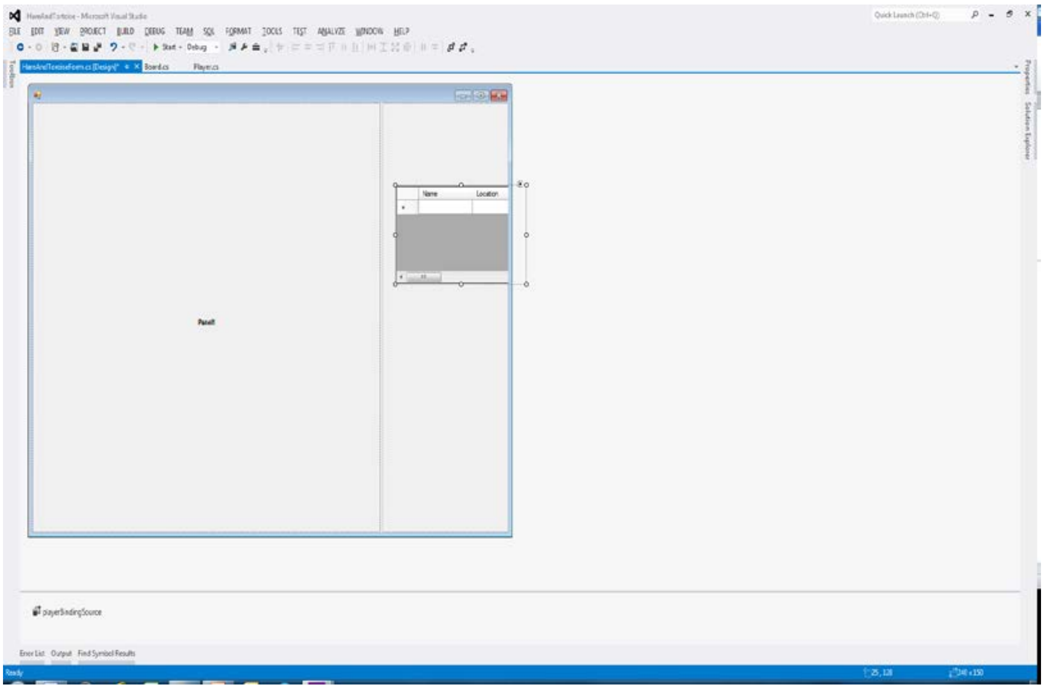
- the font used for the two labels, ***Hare and Tortoise*** and ***Players***, is 16pt, bold, italic and **Book Antiqua.** All other Text fonts are the default font provided by the Windows Forms Designer.

- the “Number of Players” **ComboBox** allows the user to select a value between 2 and 6. Use the Window Forms Designer to define the values: 2 through to 6. The size of the **ComboBox** should be sufficient to display a single digit only.

- Add a **DataGridView control** to the form. This control is located in the **Data** group of Controls. Look at the slides from lecture 10 or look at the lecture recording to see the steps required to initialise the **DataGridView**.

  Be sure to connect to the **Player class** as the **Data Source.**

The **DataGridView control** will extend beyond the size of the form similar to the screenshot below.



Do not increase the size of the **form** or the size of the **DataGridView control**.

Initially the **DataGridView** will have a column heading for every instance variable of **Player.cs**. The column heading will be in the order that the corresponding Property is defined in your **Player.cs**

In the Properties of the **DataGridView**, change the Property **RowHeadersVisible** to **False**.

Remove the columns labelled **Location** and **PlayerTokenColour.**

Reorder the remaining Columns to be in the order, from left to right, **PlayerTokenImage, Name, Money, HasWon**

Change the (name) property of the following Columns:

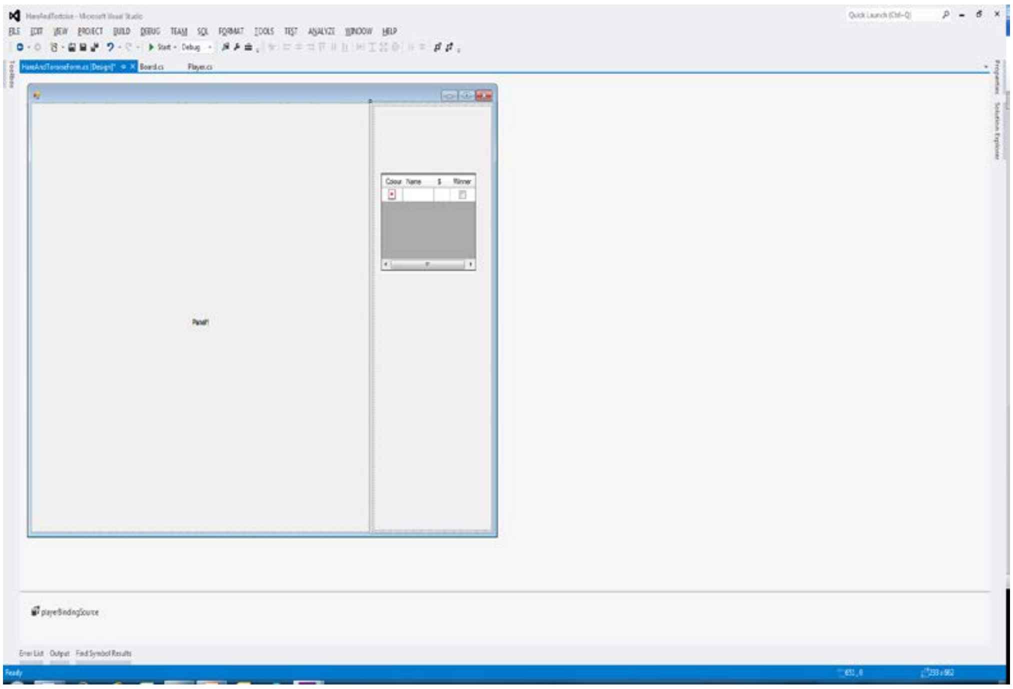
- **PlayerTokenImage** to **Colour**
- **Money** to **$**
- **HasWon** to **Winner**

Change the **Width** property of each of the Columns to the value below

- **Colour : 40**
- **Name : 60**
- **$: 30**

- **Winner : 50**

The **DataGridView** should now fit easily within the right panel.


FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL FORMAT TOOLS TEST ANALYZE WINDOW HELP
Start Debug
Quick Launch (Ctrl+Q)
Board.cs
Player.cs
Colour Name $ Winner
Panel1
playerBindingSource
Error List Output Find Symbol Results
Ready

- Now add a **ListBox** below the **DataGridView.** The **ListBox** will be used for debugging during the development phase of the assignment. How to use the **ListBox** will be explained later. For this part of the assignment you need to set the following properties of the **ListBox**:

  Size = 200, 290
  Anchor = Top, Bottom, Left, Right

  Do not worry if the Designer alters the Size values slightly from the values given above.

  the text, **listBox1**, in the top left hand corner is not visible at run-time. See the final screenshot.

- Add the three buttons labelled **Roll Dice**, **Reset** and **Exit**. The buttons need to have their **Anchor** property set to **Bottom, Right**

If you have place all of the required controls onto the form as per the preceding instructions, in **Design View** your form should look like the first screenshot in **Appendix B**.

If you have completed Part A and B and execute your program your runtime form should look like the second screenshot of **Appendix B**.

Though all care has been taken in the production of this specification, there may be a need to notify by class email any alterations/clarifications to this specification. **Check your QUT email daily, even if you have redirecting this to your own email check QUT directly in case your email provider blocks the QUT group email.**

Part C will be released before Friday 2nd October.

Appendix A: Implementing C# Property

C# Property is an alternative implementation of the traditional accessor and mutator methods of earlier Object Oriented Languages.

If in the **Player Class** you implemented the methods **GetName** and **SetName** as follows:

```
private string name;


public void SetName(string aName) {
    name = aName;
}

public string GetName() {
    return name;
}
```

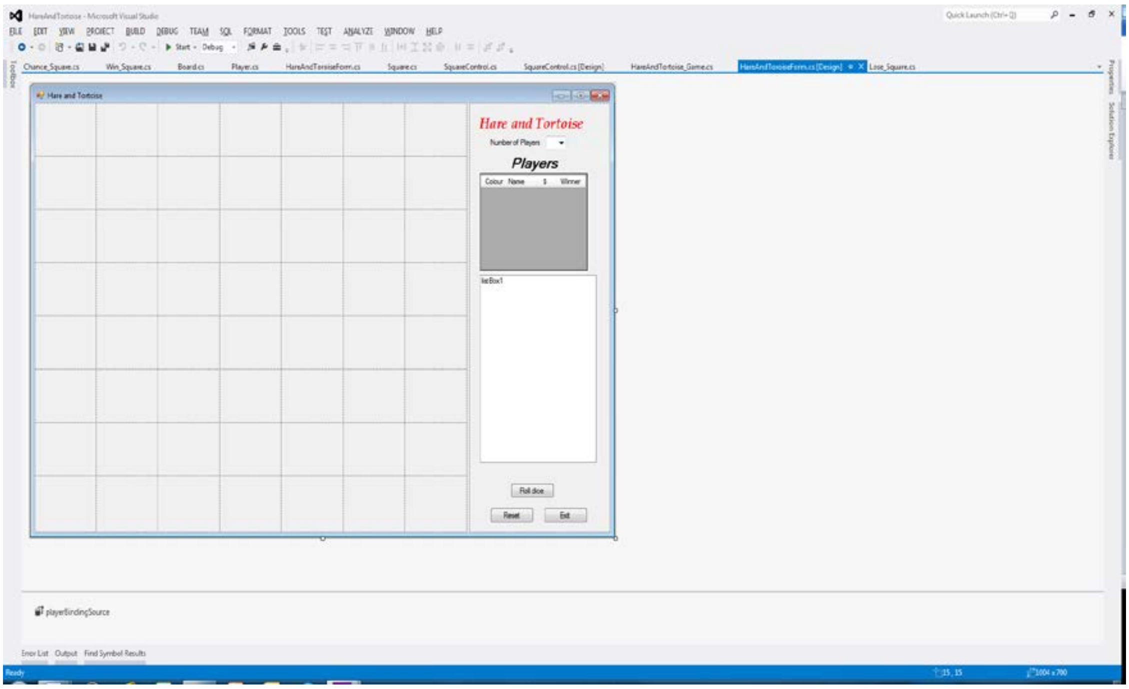
The two methods can be replace by a single C# Property declaration

```
private string name;


public string Name {

    set {
         name = value;
    }

    get {
        return name;
    }
} //end Name Property
```

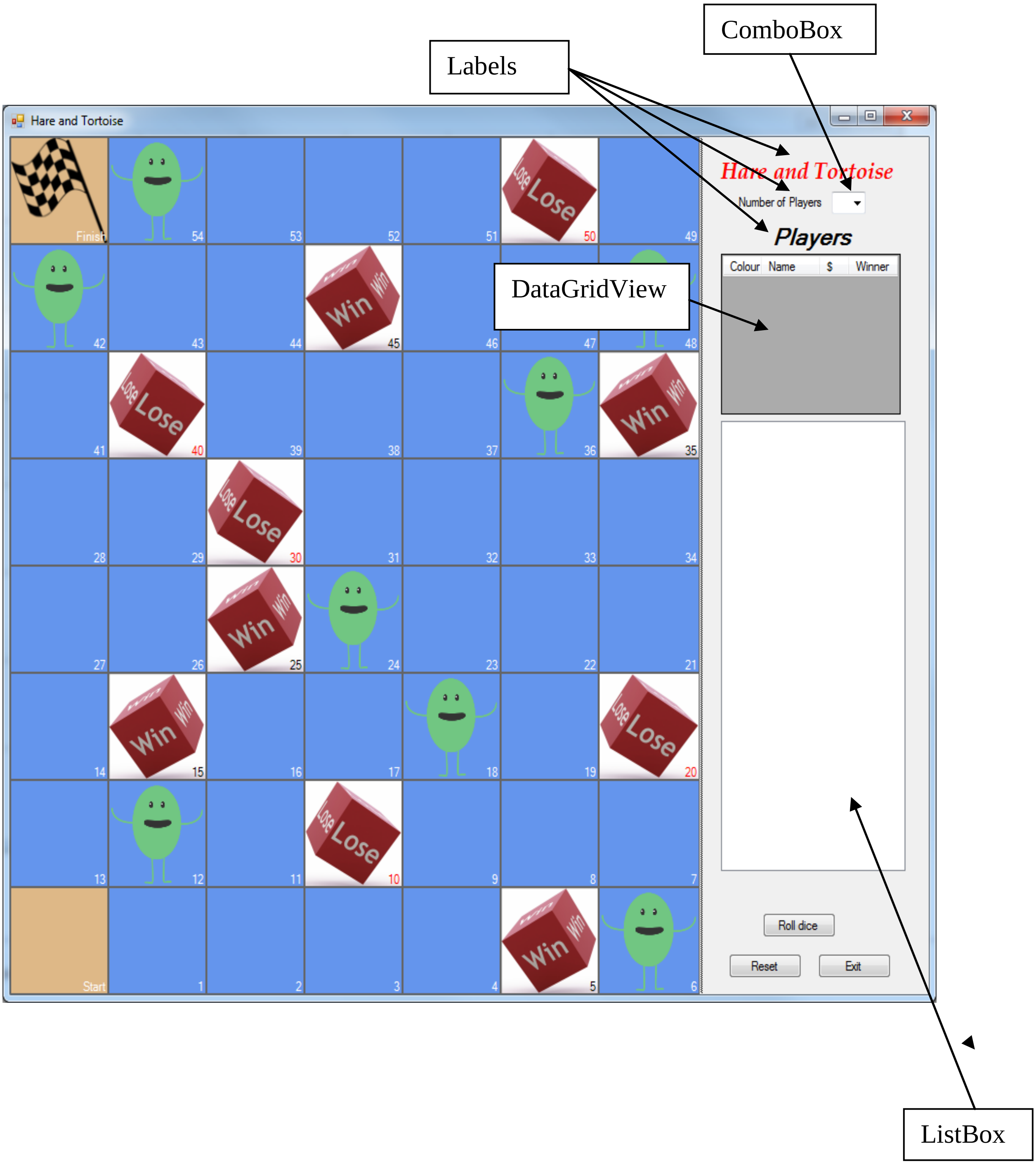
The “name” of the Property is the same as the instance variable but starts with an upper case letter as in the example above, **Name**.

Appendix B: Hare and Tortoise Form

Design View of HareAndTortoise_Form



The Form being executed having compelted Parts A and B.


ComboBox
Labels
Hare and Tortoise
Finish
54
53
52
51
Lose
50
49
42
43
44
Win
45
46
47
48
41
40
39
38
37
36
35
28
29
30
31
32
33
34
27
26
25
24
23
22
21
14
15
16
17
18
19
20
13
12
11
10
9
8
7
Start
1
2
3
4
5
6
Hare and Tortoise
Number of Players
Players
DataGridView
Colour
Name
$
Winner
Roll dice
Reset
Exit
ListBox