U10787

—

LAB 3 - Part A: Algorithms and GUI

Author	**Dr. Alexios Louridas**
Last Update	**2023-02-13**
Number of pages	**6**

1 Introduction

This lab to enhance your GUI capabilities. Although we are using windows forms which is an old technology there are some great benefits:

- WinForms is the best technology for making a quick prototype of an application.
- WinForms are used heavily to create a simple testing applications in certain industries cause it is not about beauty but about functionality with the smallest development time.
- The learning curve is a lot less steep than it is for WPF or UWP.
- Learning how WindForms work is a great paving stone for easier learning a professional GUI building environment.
- The UI editor is easy with little complexity compared to XAML
- If the main focus of a project is on algorithmic code but still need some UI enviromet then WinForms is actually a better choice. WinForms allows you to focus on your business logic, and spend less time developing your UI.

2 Task - Input Creation

You are going to build a graphical tool that creates a bar graph based on the data a user provides.

Your first task is to create an empty .NET Framework project and name it **GraphMaker**. Within the form create a `GroupBox`. Within the `GroupBox` make sure you have the following:

- `MultiLine TextBox` for inputting the data.
- `MultiLine TextBox` for category names.
- `TextBox` for inputting the graph title.
- `TextBox` for inputting the X axis name variable.
- `TextBox` for inputting the Y axis name variable.
- `Button` for creating the graph.

For each of the GUI elements you have created make sure that you have provided a good name using a good naming convention, and that everything is aligned appropriately. A good UI should:

> **Consistency and Standards**
> Ensure that both the graphic and textual elements have consistent appearance and meaning across pages/platforms. e.g. an icon should always have the same purpose, avoiding using common icons from other platforms in new unexpected ways. Stick to community standards wherever possible, e.g. why use “promote” or “+1” when you could just use “like”?

Figure 1 shows a possible look of your application at the end of this Task. Please do not concern yourself if your application does not look like Figure 1.

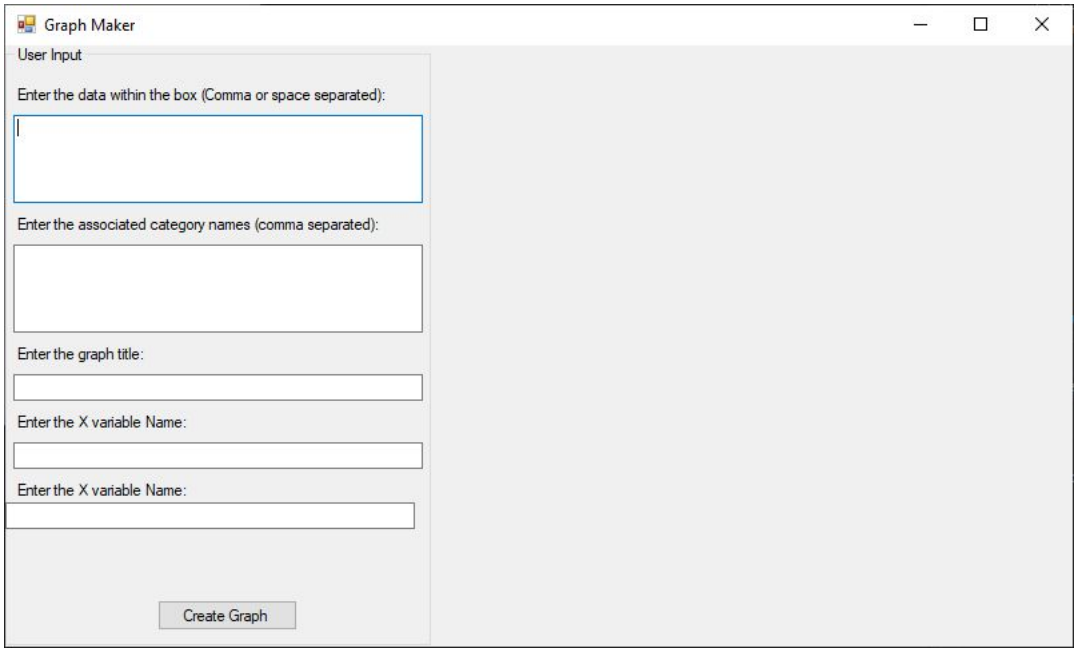

Figure 1: Application at end of Task 1.

3 Task - Reading Data

Continuing from your previous Task it is time to write code to collect the user inputs and verifying them. Since all our inputs are `TextBoxes` that means that they are all strings. Write a verification method for each input that takes a `string` as a parameter and return `bool`.

Each method when called should verify and save the corresponding input to an appropriate variable or variables. These methods should return `true` if the user inputs were acceptable and `false` if the inputs entered were not in the appropriate format.

The following verification should be considered:

Input	Required?	Verification
Data	Mandatory	Integers only and comma or space separated
Categories	Optional	Strings separated by commas only
Graph Title	Optional	A single string, containing at least a single character
X Axis Name	Optional	A single string, containing at least a single character
Y Axis Name	Optional	A single string, containing at least a single character

Table 1: Input verification table

Now call all verification methods when pressing the `Button` to create the graph. Have a `MessageBox` appear if the Mandatory inputs fail verification, stating that the inputs entered are not correct. If any of the optional inputs fail their verification make certain that the variables storing these are empty.

You can download a solution up to this point from BB. The solution will appear some time after the lab has been given.

4 Task - Table Output

This task is outputting the data in a table format. Add a `dataGridView` to your form and name it appropriately. It is now up to you to research how this control works in order to make it have:

- Have two columns, X-Axis, Y-Axis.
- First row to have the titles of X-Axis and Y-Axis entered by the user or "X-Axis" and "Y-Axis" if not entered.
- All other rows should have the X-Axis filled by the categories entered by the user. If a category is not present the cell should be empty.
- The Y-Axis cells should contain all the values of data entered by the user.

Some output examples can be seen in the following figures:

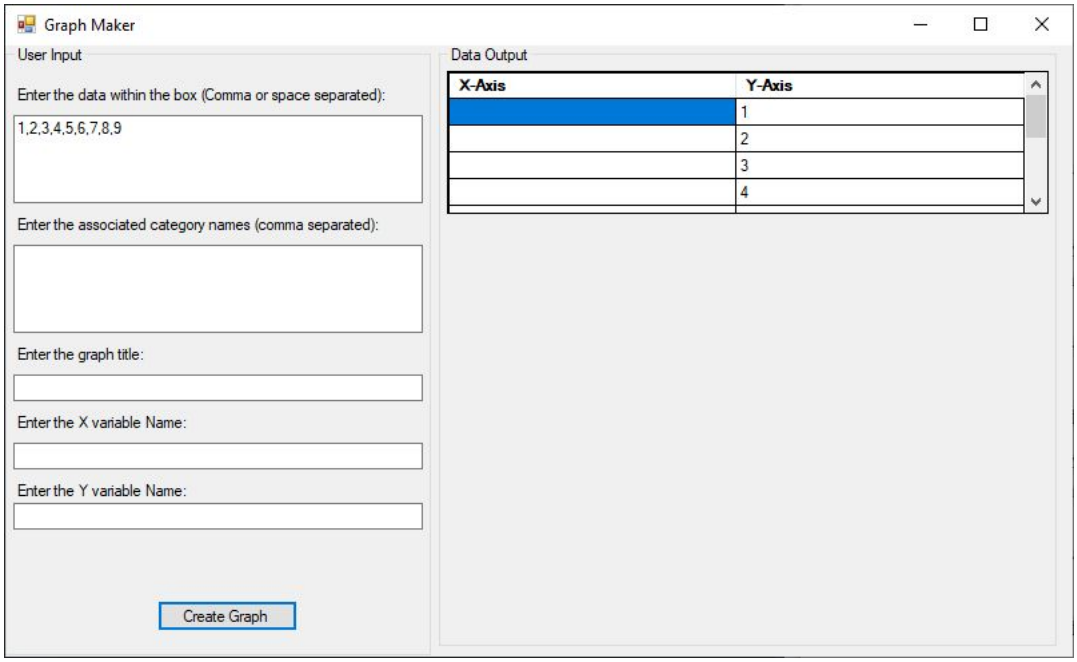

Figure 2: 1st Example of DataGridView

Example 2:

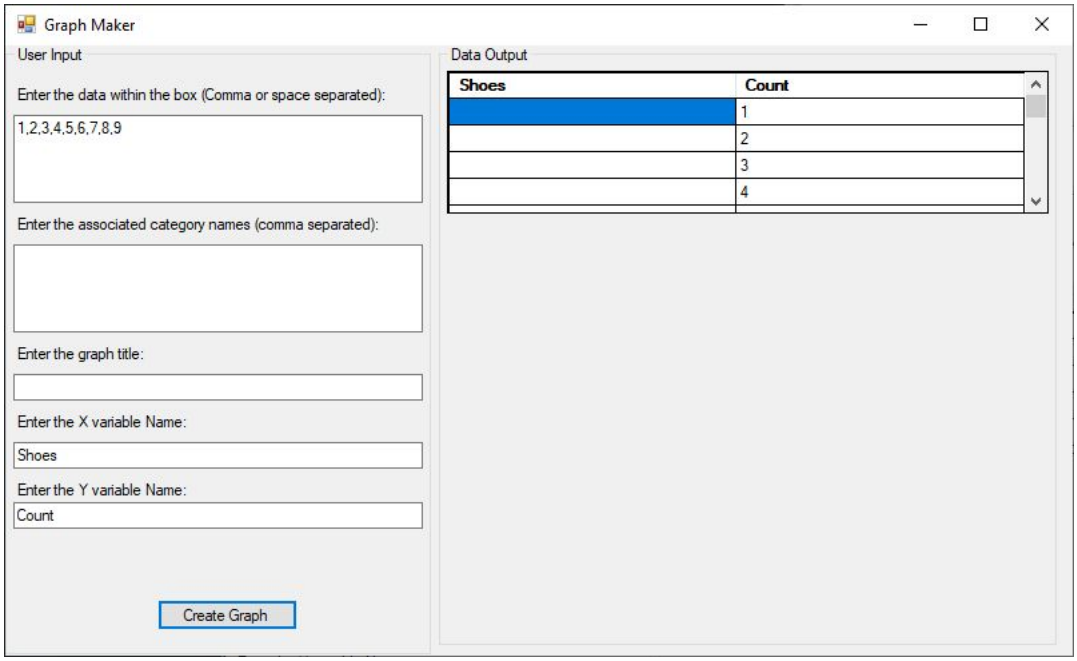

Figure 3: 2nd Example of DataGridView

Example 3:

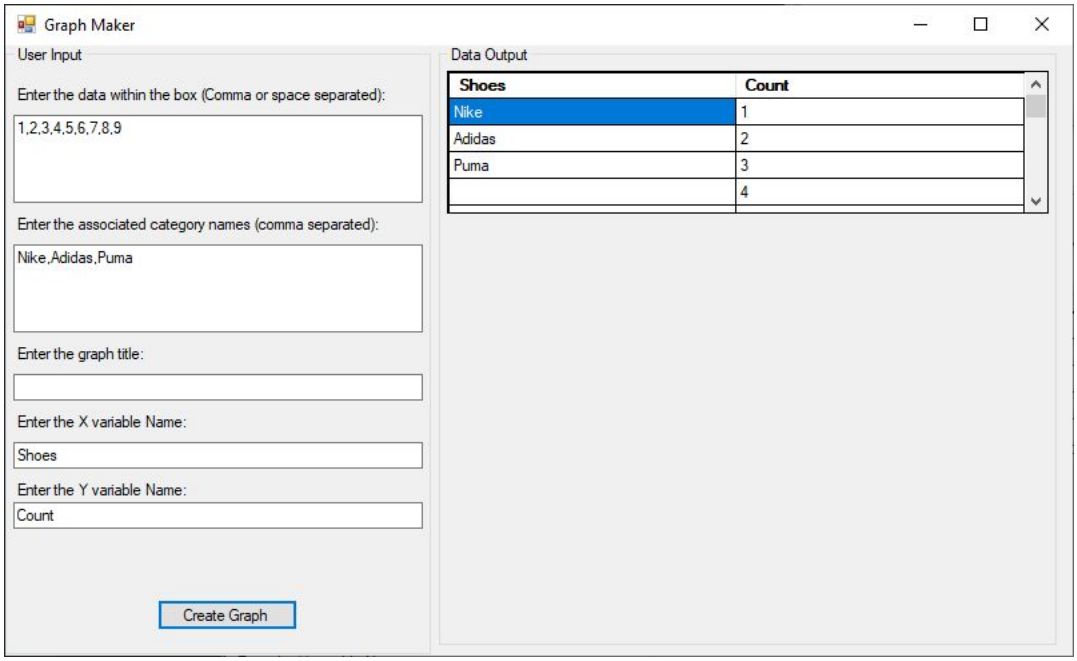

Figure 4: 3rd Example of DataGridView

You can download a solution up to this point from BB. The solution will appear some time after the lab has been given.

5 Task - Bar Graph

Now that the data appear in your table it is time to create the bar graph. First of all you need to understand some of the coordinates of GDI+ for all all form controls. The origin (0,0) is the upper left corner. You can find the relavent information on how to start with drawing here:

Graphics in Windows

Using the link provided, the help from your tutors, add a `PictureBox` in your form design. Add a `Label` on the top left corner of your `PictureBox` and a `Label` on the bottom right. These should be used to add the labels selected by the user for the y and x axis respectively. To draw in the `PictureBox` you would need to get the Graphics object of an image and then draw the image on the `PictureBox`.

Below is an example of drawing a red and white strip flag using filled rectangles:

```
PictureBox pictureBox1 = new PictureBox();
public void CreateBitmapAtRuntime()
{
    pictureBox1.Size = new Size(210, 110);
    this.Controls.Add(pictureBox1);

    Bitmap flag = new Bitmap(200, 100);
    Graphics flagGraphics = Graphics.FromImage(flag);
    int red = 0;
    int white = 11;
    while (white <= 100) {
        flagGraphics.FillRectangle(Brushes.Red, 0, red, 200,10);
        flagGraphics.FillRectangle(Brushes.White, 0, white, 200, 10);
        red += 20;
        white += 20;
    }
    pictureBox1.Image = flag;
}
```

Using all the above to help you create a bar graph from the numbers supplied by the user. The bar graphs should be vertical. An example of how the bar graph should look is shown in Figure 5.

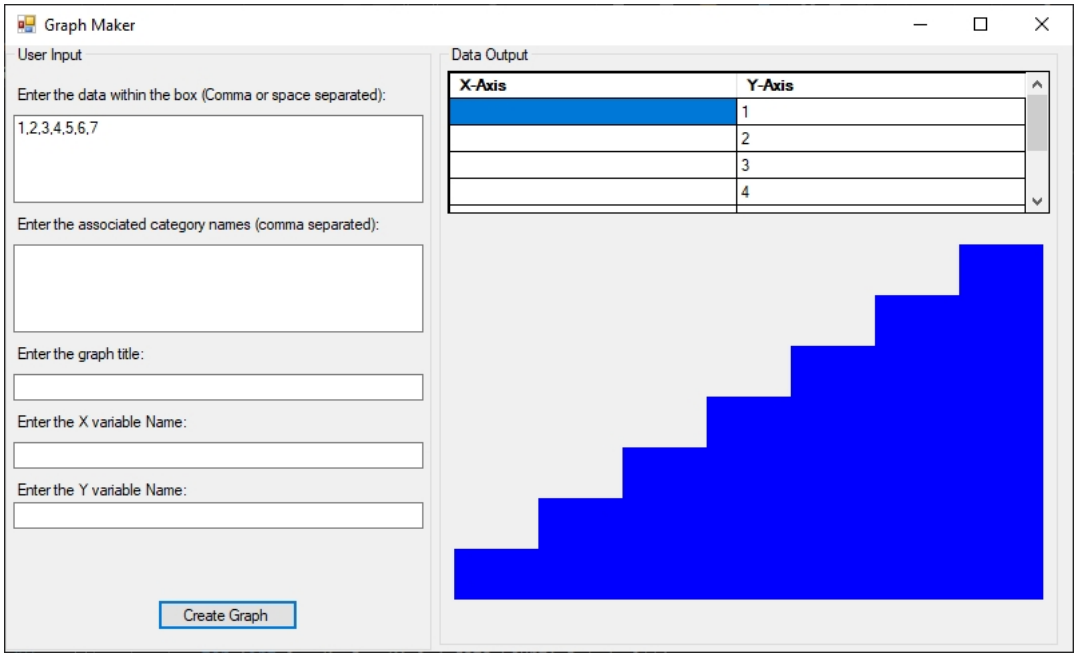

Figure 5: Example of bar graph

6 Task - Nicer Graph

For this task we enhance the bar chart by adding some axis and making sure there is some space between the bars. So make changes so your project in order to look like the Figure 6.

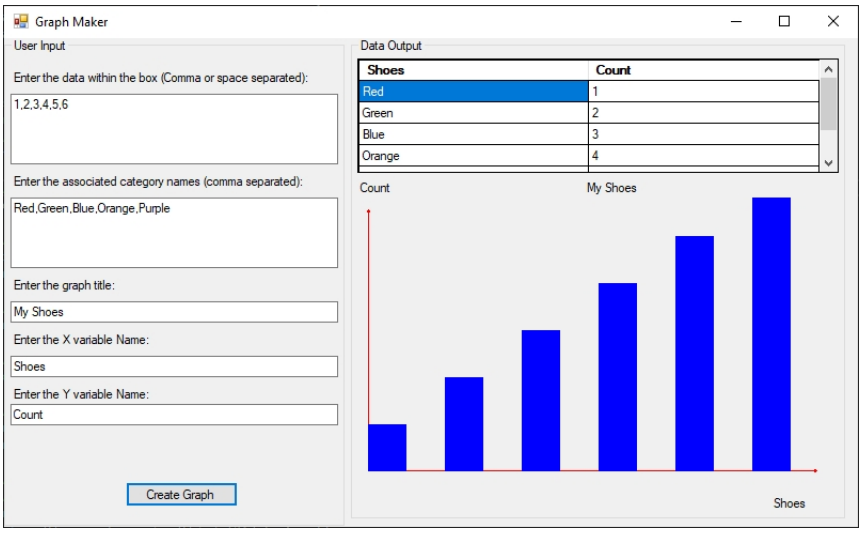

Figure 6: Example of final graph application.