MALMÖ HÖGSKOLA

Programming in C# and VB II

Events, Delegates and WPF

Assignment 5 – The Control Tower

Mandatory

Farid Naisan
University Lecturer
Department of Computer Science

The Control Tower

1 Objectives

Delegates are mainly used in event handling but also for method calls, i.e. when a method is to call one or more other methods. . NET uses delegates to handle events according to a publisher-subscriber model. This is a well-known methodology even for design of solutions to similar problems in object-oriented programming.

The main objectives of this assignment are:

- to implement a user-defined publisher/subscriber model,
- give you a first training in using WPF instead of Windows Form.

Note: Although the word Form may occur in this document, it is mandatory to use WPF for the GUI.

2 Description

A publisher is an object that offers services to other objects. Those objects that request to use the services of the publisher are the subscribers. Every publisher object can publish a number of events which other objects can subscribe to. An object can be both a publisher of events and a subscriber to events published by other objects. When an object subscribes to an event, the object will be notified by the publisher whenever the event is fired. The connection between a publisher and the subscribers is carried out through delegates.

In this assignment, you are to write an application called the **Control Tower**. A control tower usually controls all aircraft traffic, and determines when airplanes are to be allowed to take-off, change route and land. However, in this application we do not provide these functionalities. Instead, we let the application only prepare the planes on the Runway for take-off. The control tower should simply add a flight number, and sets the airplane permission to start.

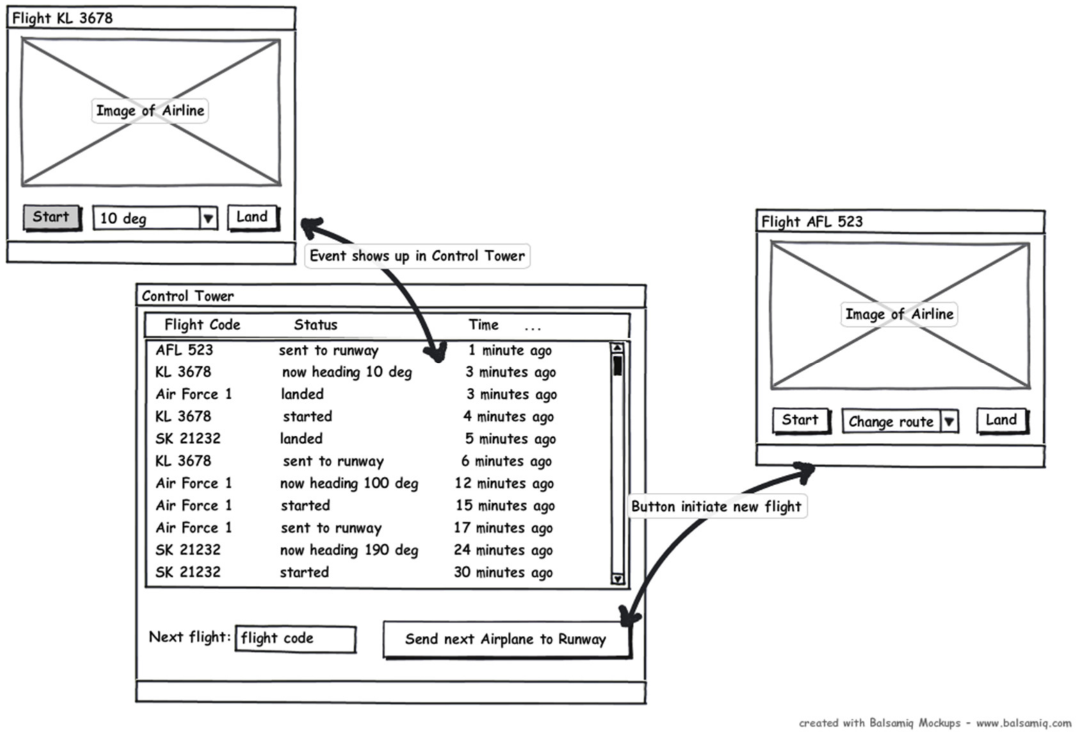
A sketch of the GUI for application is given on the next page. You may of course bring changes or use your own design. Hitting the button "Send next Airplane to Runway" starts a new Window representing a new flight.

The flight sets a logo image of the airline to a picture box on the window as illustrated in the above figure. If the logo of the airline is unknown, a default image is to appear on the window instead. The flight number is also shown on the title bar of the Window. When an airplane takes off, changes route or lands, the airplane sends notifications and info about the event. The control tower receives the notifications, and displays the flight number, status and a time stamp to its message board. The control tower has no knowledge of the airplane which fires the event. Therefore the information about the flight number must be provided by the plane sending the event together with the event notification. On the other hand, and the plane doesn't know either which objects receive the notifications. Both sides need to use a delegate to make this communication work.

3 To Do

Based on the above description, figure out which object (the plane or the control tower) should serve the publisher and which object would be a subscriber. Draw a GUI to simulate the plane and a window for the Control Tower. A suggested sketch is given below.

On startup, only the start-button should be enabled on the GUI. The Change route-ComboBox (or whatever control you find appropriate to use) and the Land-button should be disabled to increase the usability of your application. Since a flight can start only once, the start-button should be disabled once it has been pushed, and the Land-button and Change-route-control are to be enabled then. When the airplane has landed, the notification is published and the flight Window is terminated.


Flight KL 3678
Image of Airline
Start
10 deg
Land
Event shows up in Control Tower
Control Tower
Flight AFL 523
Image of Airline
Start
Change route
Land
Button initiate new flight
Next flight:
flight code
Send next Airplane to Runway
created with Balsamiq Mockups - www.balsamiq.com

Flight Code	Status	Time ...
AFL 523	sent to runway	1 minute ago
KL 3678	now heading 10 deg	3 minutes ago
Air Force 1	landed	3 minutes ago
KL 3678	started	4 minutes ago
SK 21232	landed	5 minutes ago
KL 3678	sent to runway	6 minutes ago
Air Force 1	now heading 100 deg	12 minutes ago
Air Force 1	started	15 minutes ago
Air Force 1	sent to runway	17 minutes ago
SK 21232	now heading 190 deg	24 minutes ago
SK 21232	started	30 minutes ago

The control tower needs not know which airplanes it has sent to the runway and therefore you don't have to save the flights in any collections

4 Requirements for a passing grade (G)

Create a Windows Presentation Foundation (WPF) project. The GUI for this assignment is to be done using WPF. For a passing grade the requirements below are mandatory.

4.1 Define three **EventArgs** info classes, derived from **EventArgs**, for each of the events, **Take Off**, **Change Route** and **Land**. Each class must be placed and saved in a separate file. In case the same info class can be used for more than of the events, you do not have to write one for each of the events.

4.2 Design a Window for flights, **FlightWindow**, and one handling the Control Tower, **ControlTowerWindow** and write the necessary code to make the application work as described above. Now think about which one of these classes can play the role of a publisher and which one can be a subscriber. Discuss the issue in the forum for this module and be well convinced about selecting the publisher and the subscriber classes before starting your implementation.

4.3 Define **Delegates** and notifiers, (handler methods of type Onxxx) in the publisher class.

4.4 It should be possible to start up several instances of the ***publisher*** class***.***

4.5 Any flight should be able to **start**, **change route** and **land** independent of other flights.

4.6 Find at least three images that can represent the airline owning the flight or images that have to do with the flight or the airline in question, from the Internet (or otherwise). Display the image that is related to a flight on the flight GUI using a PictureBox.

4.7 Collections are not to be used. The control tower should not have a reference to the flights other than the delegates and events.

4.8 Perform some type of a validation of the flight number inputted by the user is not an empty string.

5 Requirements for a pass with distinction Grade (VG)

The following items are required only for a higher grade, VG (in addition to the above requirements). You can skip this section, if you aim only at a passing grade G.

5.1 Use a WPF ListView for displaying the info on flights.

5.2 Make another type of object that is a subscriber and let this also be notified of some of the plane events. When a plane takes off, changes route and/or lands, both the **TowerControl** and this new object should be notified should take actions. An example could be an object that plays an applause when a plane is landing or an object that logs the landing and starting time of all the flights (but not the change route).

6 Submission

Upload your project to It's Learning as before.

7 Help

There is an example project, **BidHouse (both in C# and VB)**, in the module that should give you a good idea. Inside the code files, it is documented what the publisher and the subscriber classes should be written. Take a good time and try to understand the code. This assignment is actually easier than the mentioned example

Using delegates might seem to be difficult, and may take a while to understand when it is new to you, but once you practice with it and learn to determine the roles of the publisher and the subscribers, you will see that the mechanism makes sense.

Think of the role of the ball in a football game. When the ball is kicked, it is the ball object that gives out (publishes) the signal (event) of being kicked without knowing (or needing to know) which other objects are waiting to receive the signal. There are several objects, the referee, the players, and the goal keeper, who would like to be notified when the ball has been kicked. These are the subscribers. Now think of the two tasks: to accomplish in the code: Which code to write in the Ball class to fire the events and how you should code in each of the subscriber classes to get notified when an event is fired by the publisher object. These items are documented well in the code example, the **BidHouse**.

The Delegate type is a powerful construct in .NET and will definitely come to use in many situations. Use the forum to get ideas and more help.

Good Luck!

Farid Naisan,

Course Responsible and Instructor