Steven Myers

December 5, 2016

CSCI-A290 – Windows Programming in C#

Assignment MINI 3

**Topic 1 – Learning C#’s “syntactic sugar” and operators**

Sources: <http://stackoverflow.com/questions/3622160/c-sharp-passing-function-as-argument>, <https://msdn.microsoft.com/en-us/library/bb549151(v=vs.110).aspx>, <https://msdn.microsoft.com/en-us/library/aa288459(v=vs.71).aspx>

Summary:

C# has a delegate type in order to help programmers encapsulate functions. It is “similar to a function pointer in C or C++.” It allows a program to reference a method “without knowing at compile time which method will be invoked.” Delegates can serve a somewhat similar purpose to that of interfaces in Java, where methods themselves are abstract and the code bodies are to be filled in for any object implementing the interface. We can make a delegate called “processBooks” and we can fill in the code body of this delegate or reference other methods that can perform some action called ‘processBooks.’ It could be that the delegate method could use multiple methods based on the input.

I personally used the built-in Func Delegate in my RPNCalculator application. Using the func delegate, I created an anonymous delegate type – a lot like Java’s lambda expressions. By doing so, I could fill in parameters to functions that accepted a delegate type, and applies arguments to that function all the while keeping the function both dynamic, short, and efficient. I think this is important so that you can begin to minimize the number of lines of code are in a program and simplify code blocks while still performing the task you need to do. It also allows for a more abstract, modular programming style to form.

Example:

void static calculateSomething(Func<integer, integer, integer>) {…}

…

calculateSomething((x,y) => x / y)

**Topic 2 – HTMLAgilityPack (Web Scraping in C#)**

Sources: <http://www.4guysfromrolla.com/articles/011211-1.aspx>, <https://msdn.microsoft.com/en-us/library/system.net.httpwebrequest.aspx>,

<http://www.mikesdotnetting.com/article/273/using-the-htmlagilitypack-to-parse-html-in-asp-net>

Summary:

The HTMLAgilityPack (HAP) is a tool written in C# to parse html documents. Without HAP, it can be difficult and cumbersome to parse HTML documents. The built-in HttpWebRequest class in C# can be used to grab HTML documents and convert it into a string representation; however, you are still left with the dirty working of parsing the HTML with regular expressions or other means of string parsing.

The HAP aims at doing a lot of this work for you by giving you a set of tools to easily scan through the DOM (document object model) of an HTML document. It makes it quite simple to iterate and parse through an HTML document and grab the attributes you need via XPath expressions. You can also recur over all of the nodes using some of the HAP functions.

The links I have provided essentially cover how simple it is to use HAP to parse HTML. Here is a short snippet from *mikesdotnetting.com*’s example:

var html = new HtmlDocument();

html.LoadHtml(new WebClient().DownloadString("http://forums.asp.net/members/Mikesdotnetting.aspx"));

var root = html.DocumentNode;

var p = root.Descendants()

.Where(n => n.GetAttributeValue("class", "").Equals("module-profile-recognition"))

.Single()

.Descendants("p")

// do stuff with p

I think that, in our digital age, if we are going to do any kind of software development, it is important to understand how to interact with HTML with native applications.

**Topic 3 – MSAGL (Microsoft Automatic Graph Layout)**

Sources: <https://github.com/Microsoft/automatic-graph-layout>, <https://www.microsoft.com/en-us/research/project/microsoft-automatic-graph-layout/>,

<http://mathinsight.org/definition/directed_graph>

Summary:

I recently started working with and learning about a library/framework called Microsoft Automatic Graph Layout ( MSAGL). It converts a graph into a visual representation and has some built-in functionality to manipulate and explore the graph.

Before describing too much about MSAGL, I will briefly summarize an article I used to learn about what directed graphs are. Directed graphs are a “set of objects (called nodes) that are connected together” where each node has an edge or path leading from one node to another in a direction, hence the name “directed graphs.” These directed graphs are important for storing all kinds of real-world representations of problems such as a map containing cities and roads, or a social network where people are the nodes and relationships are the edges/paths.

With MSAGL, we have a way of not only storing these real-world ideas into a directed graph format, but displaying them visually so that we can analyze the connections in a visual format. MSAGL uses a variety of algorithms to automatically place nodes such that they are aesthetically pleasing and easy to read. I find this technology very important because it allows us to make high-level assumptions and decisions based on visual results created from graphs.