Assignment 2: Comparing C# to C.

Gabriel Stroe

Comparative Programming Languages

# Introduction

Programming languages have come a long way since the development of C. The biggest and probably most impactful development since C language is the development of the object oriented programming. Programming languages such as C++, Java, python and C# take advantage of this new style of programming. Creating objects called classes that can be passed around, manipulated and have functions that the said objects perform themselves really allows the programmer to take advantage of these features. Things such as the modern GUI, external resources as objects, or even the abstraction of real world objects is what becomes available with object oriented programming. The following will describe two programs written in C# that show these features and are programs that would be much more difficult to create in C. Of course as with almost all languages, given enough time and resources just about everything is possible in C, but the features given in language like C# allows for a much faster and less complicated development.

# Program One Outline: Shapes

The first program is called Shapes. This is a relatively simple text based program with a simple idea. This program allows you to enter one of 3 different types of shapes: ambiguous, rectangle, and triangle. These shapes that are created get added to a list of shapes that the user can print and see. Each type of shape has some specific features associated with them. Ambiguous shapes all have 5 or more sides to them, have and area, and a perimeter. There is no correlation between the sides, the perimeter and the area as the shape is ambiguous. Rectangles are shapes that have an x length, a y length, 4 sides, an area, and a perimeter. Creating the shape will calculate the area and perimeter based on the length of the sides. Also when editing any value of a rectangle, the object changes the values accordingly to match. For simplicity it is assumed that we an area or perimeter is changed for a rectangle the object is assumed to go square and the sides are changed accordingly. The last available shape type is a triangle. Similar to a square it has a set amount of sides, and 3 side lengths. It also still has an area and perimeter since all shapes have this. Also like the rectangle changes one feature of this shape will change the others to match accordingly. Changing the area or the perimeter of a triangle will scale the sides to be an equilateral triangle.

# Program One Comparison: Shapes

There a few features that exist in C# but not C that I wanted to bring out for this program. First, C has no form of overloading functions (Miller). Overloading functions in this application is mostly used for the creation of the shapes. The constructors for every shape can all accept just a name or even one size for this. Also, since C only has structs and no form of classes, these object that are created have no methods (Walls). In order to have the shape have a print method one would need to have a generic print method that would be the same for all structs. With classes in C#, we can make the printAll() method be unique to every object. In addition, because we can use inherited classes in C# we can use the general shape skeleton for all shapes and only add a few features that are need for the specific shape like the sides of a rectangle for instance. Another key feature that was used in this shape program is virtual methods that can be overridden (Walls). This in the shapes program is used to added different type of constructors to the shapes. This way the programmer may make the shape without adding the lengths or area and just use the name or vice versa. This overriding method also works with the setValues set of functions like setArea(double Area) or setPerimeter(double Perimeter). Overriding these inherited functions allow our rectangle and triangle shapes to calculate their other values. For example, the setArea method for an ambiguous shape just sets area of said shape, while in a rectangle it also computes the equivalent side lengths and perimeter. To do something similar in C one would need to have a completely function name and constantly be check which type of shape is being looked at write now. The last tool used that doesn’t exist in c is a List. For this application a List<Shape> is used to store the list of created shapes and easily add, edit and remove the shapes. In C this can be still done, but by a linked list which are significantly harder to use (Allian). Whenever an item is removed the links would have to be recreated and the loop would always need to be traversed from the beginning (Allian). So using a List in C# is just significantly simpler.

# Program Two Outline: System Monitor

The second program is called System monitor. This is a GUI based application that will show the user in the system tray if the hard drive is being written to. To do this we use what is called a ManagementClass in the .net platform. This class allows us to use the Windows Management Interface (WMI) to get system performance metrics which then can be used to monitor the amount of data being written on the hard drive. The icon in the system tray will show red if there is data being written. Otherwise it will show black. In addition, right clicking this brings up additional menu that will all the user to pull for CPU performance. This also uses the WMI to receive system information and grabs the CPU Clock and Voltage data when the user presses the load current button.

# Program Two Comparison: System Monitor

Program two has a few new features that would be fairly difficult to program in C. The lack of classes alone makes it difficult to create a GUI in C. One can program a GUI in C but it requires the use of a 3rd party library that generally is much more difficult to learn and use (Grey). In C# the .net standard libraries provide all the necessary classes and types to make this type of GUI. In addition, with C there is not any sort of event handling. Instead one must be constantly checking to see if the flag can now be set to true (Kaur). With C# we can just use an event handler that will run in that routine as soon as the event takes place. This in the system monitor program is how the button is programmed to work. The mouse click “event” on the button cause the program to run the loadButton\_Click() method, which will get the CPU Clock and Voltage. The last feature of this application that would be much more difficult to program in C is the use of the WMI. Thanks to C#’s bases on the .net platform which is developed by Microsoft the Management library can be easily imported and used. To do this same thing in C one would have to use the COM API for WMI (COM API). This isn’t an extremely difficult endeavor, but it is more challenging than simple using the Management Library. To use the management library, one must also use a windows provided tooled called wbemtest.exe to get the name of the peripheral they wanted to read data about.

# Conclusion

So in conclusion there most of the benefits of C# over C, come from C#’s classes and .net framework. The utility that comes from classes allows the programmer to do much more and much easier. Creating artificial class-style objects in see would be significantly more work. Also having the .net platform so well integrated with the Windows environment really was useful for the WMI system performance monitoring as well as some of the GUI features (setting an application as only visible in the system tray). It’s encouraging to see how much easier programming applications have become to create since the release of C. Perhaps soon a new language will be created that makes C#’s tools look as crude as C today.

Works Cited

Allian, Alex. "Lesson 15: Singly Linked Lists in C." Cprogramming.com. Cprogramming.com, n.d. Web. 8 Apr. 2017.

"COM API for WMI." COM API for WMI (Windows). Microsoft, n.d. Web. 8 Apr. 2017.

Gray, Cody. "Re: How Can I Do GUI Programming in C." Blog comment. Stack Overflow. N.p., 27 Mar. 2011. Web. 9 Apr. 2017.

Kaur, Priyadeep. "Interrupts short and Simple." Embedded. N.p., 12 Oct. 2012. Web. 9 Apr. 2017.

Miller, Mark. "Re: What Is the Difference between C, C++ and C#?" Blog comment. Quora. N.p., 11 Apr. 2016. Web. 5 Apr. 2017.

Walls, Colin. "Struct vs Class in C++." The Colin Walls Blog RSS. Mentor Graphics, 2 June 2014. Web. 07 Apr. 2017.