Nicholas DiPinto

Robert Velarde

CS 332

Dr. Siewert

10 April 2017

Assignment 6

1. For our project, we are comparing the C++ programming language to the JavaScript programming language. C++ was developed by Bjarne Stroustrup in 1979 as part of his Ph.D. thesis. After exposure to the Simula programming language, Mr. Stroustrup decided he wanted to create a C compiler that would include object-orientation. He chose to improve the C language because he was working for Bell Labs, where the C language was developed, and C was a language known for its portability and speed. The first compiler he developed to accomplish this task was called “C with Classes”, which is a self-hosting compiler. The first language manual for this language was published in 1985, called “The C++ Programming Language”. The name of the language was changed to C++ to represent the idea that this language was an incremental enhancement to the C language. C++ would later be enhanced to include support for features like regular expression support, atomics support, and a threading library (History of C++ - C++ Information, 1).

The JavaScript programming language was developed by Brandon Eich in 1995 (A Short History of JavaScript, 1). However, the language did not gain popularity until 2005, after Jesse James Garrett released a paper that describes a set of technologies implemented using JavaScript (A Short History of JavaScript, 1). This paper started a movement in large developer communities to use JavaScript as a scripting language to build web applications because data could be loaded in the background without having to reload an entire web page (A Short History of JavaScript, 1). Users of the programming language claim its benefits include a “flexible syntax, loose type system, powerful reflection mechanisms, and shorter build cycles” (Kienle, 1).

We chose the C++ programming language as our primary language because it is built upon C. Therefore, the programs written in the language are expected to be portable and easy to compile, like C. We chose JavaScript as our secondary programming language because the idea for our project is based off the efforts of the Wycliffe Associates, who are trying to build a written language translation application using JavaScript. Their choice for JavaScript, as described by one of their managers, can be attributed to its popularity, flexible language, and portability. In this project, we will be building our own simple translation machine written using both the C++ and JavaScript programming languages so that we may compare the two languages.

1. Our application will have similar algorithms written in both JavaScript and C++. The application can be broken down into three main sections: file I/O, string manipulation, and floating point arithmetic. Performance for our application will be based on these three sections. We will perform a side by side comparison of both implementations to test their performance.

The first metric is file I/O. C++ is a compiled language, while JavaScript is an interpreted language. Because of this, C++ usually runs much faster than JavaScript. Because both languages have implementations for file I/O, we believe that C++ will execute the file I/O portion of the code faster than JavaScript.

The second metric is string manipulation. JavaScript contains multiple methods for manipulating string. C++ on the other hand does not. Functions such as ‘lowercase’ are already implemented in JavaScript but must be implemented by the programmer in C++. This may result in more inefficient code in C++. Because of this, we believe that JavaScript will execute the string manipulation portion of the code faster than C++.

The third and final metric is floating point arithmetic. Fbench is a trigonometry intense floating-point benchmark. It determines the execution speed of a floating-point application and checks the accuracy of the algorithm. With C as the baseline of 1, JavaScript received a relative time of 27.6 on a Linux machine. Because of this, we believe that C++ will execute the floating-point math portion of the code faster than JavaScript.

The information supporting our hypotheses were obtained from the following sources:

<https://developer.mozilla.org/en-US/Add-ons/Code_snippets/File_I_O>

<http://www.cplusplus.com/reference/fstream/ifstream/>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String>

<http://www.cplusplus.com/reference/string/string/>

<https://www.fourmilab.ch/fbench/fbench.html>

1. Our first objective metric is cost of using both programming languages. Each language has their own unique costs that should be considered whenever choosing to use either one. For instance, using JavaScript has large costs in development and testing time. Since JavaScript is a scripting language, errors are only reported in the debugger during run time and require the use of a web browser. Therefore, errors can only be caught when they are created during program execution. This issue can cause an increase in the amount of time debugging JavaScript code.

For our next objective metric, we examined the number of lines of code it took to implement a translator program in each language. The JavaScript program was implemented in 133 lines of code, while the C++ program was implemented using approximately 189 lines of code. Therefore, JavaScript is more efficient in code size for implementing certain algorithms. Based on our experience with both programming languages, we theorize that the difference in lines of code can be attributed to JavaScript’s simpler implementation of string manipulation algorithms.

For our last objective metric, we measured the amount of time it took for each programming language to perform string manipulation, file i/o, and floating point mathematics for ten trials and measured the average time for each language to perform each of these tasks. Our results can be seen below in Table 1: Comparison Metrics:

**Table 1: Comparison Metrics**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Execution Time (milliseconds)** | | | | | |
| **Iteration #** | **String Manipulation** | | **File IO** | | **Floating Point Math** | |
|  | **C++** | **JavaScript** | **C++** | **JavaScript** | **C++** | **JavaScript** |
| **1** | 0.008 | 0.2949 | 0.62 | 0.0117 | 0.5444 | 1.6518 |
| **2** | 0.006 | 0.0581 | 0.329 | 0.1408 | 0.577 | 0.2329 |
| **3** | 0.005 | 0.0529 | 0.309 | 0.0949 | 0.546 | 0.202 |
| **4** | 0.006 | 0.0610 | 0.329 | 0.0400 | 0.52 | 0.1801 |
| **5** | 0.007 | 0.0632 | 0.528 | 0.0432 | 0.517 | 0.2368 |
| **6** | 0.006 | 0.0561 | 0.315 | 0.0419 | 0.528 | 0.1860 |
| **7** | 0.005 | 0.0449 | 0.333 | 0.0397 | 0.203 | 0.2041 |
| **8** | 0.007 | 0.1479 | 0.047 | 0.3501 | 0.464 | 3.7209 |
| **9** | 0.008 | 0.0378 | 0.314 | 0.0310 | 0.512 | 0.2880 |
| **10** | 0.005 | 0.0439 | 0.315 | 0.05102 | 0.2 | 0.1701 |
| **Total Average** | 0.006 | 0.0860 | 0.3439 | 0.084432 | 0.46114 | 0.70727 |

The table above shows the difference in performance for string manipulation, file i/o, and floating point math for both C++ and JavaScript. On average, JavaScript outperformed C++ in file i/o, but C++ performed better on average in string manipulation and floating point math. One result to note is the outcome of our floating-point math times. Majority of the iterations of our program showed JavaScript with shorter execution times, but iterations 1 and 8 have major outliers that had a strong effect on the average score.

For our subjective metrics, we compared the readability, debugging, and extensibility of each programming language. For readability, JavaScript is superior in its syntax because it focuses on using descriptive names for each of its functions, while avoiding the use of any unintuitive operators, like C++ with the extraction (>>) and insertion (<<) operators. However, C++ is superior in debugging because execution of code in C++ requires that it be compiled first. As stated earlier, JavaScript programs are not actually compiled before they are executed. Therefore, the only way to have an error reported in the debugger is to run into the error while using the program. In terms of extensibility, both languages have a high-level of support and large number of libraries to perform additional tasks. C++ has libraries for image processing, video processing, 3D graphics, communication, concurrency, and many other types of libraries. Most of the libraries for JavaScript are focused around features commonly utilized in web application development, which is the main purpose of JavaScript. Observing the number of categories of libraries for each language reveals that C++ has more support to perform more complex tasks than JavaScript.

1. Running the JavaScript program requires opening your web browser. Any web browser will work, but the best one to use is Google Chrome because of its simple to use Debug Console. Google Chrome is free to download at <https://www.google.com/chrome/browser/desktop/index.html?brand=CHBD&gclid=Cj0KEQjwuOHHBRDmvsHs8PukyIQBEiQAlEMW0PtGSVmZSqZcVonijIZi3kvgPyKtNTUFwvQXLPZ3oQIaAuQh8P8HAQ&dclid=COHrwfCNtNMCFYVufgodQlgCpg>. After downloading Google Chrome, one can run the JavaScript program by double-clicking on the HTML file called, “translate.html”.
2. The first key difference between the two programming languages is that JavaScript is classified as a scripting language. Therefore, there is not an explicit compilation step before running code written in JavaScript. Skipping this compilation step causes errors to only be caught during run time when debugging the program. Although skipping the first compilation step could be viewed as more efficient, it can also lead to potential dangers when testing the program because the debug console will not reveal warnings or errors in the program until they are encountered during run time. Without the initial compilation step, running a program in JavaScript can prove more dangerous than running a program in C++.

A second difference that arises between programming languages and scripting languages is that scripting languages can only be ran inside of separate programs rather than independently, like programs written in C++. In the case of JavaScript, code written in JavaScript cannot be ran independently, but needs to be implemented inside of HTML files. This difference can make learning JavaScript, and scripting languages in general, more difficult to learn because someone trying to learn the language must also learn the basics of another language to implement their program. For instance, learning JavaScript required that I also learn how to create a web page in HTML that can implement the JavaScript code. C++ is easier in this respect because implementing a program in C++ requires knowing the two commands to compile and run the code inside the command line.

Lastly, C++ and JavaScript use different methods for assigning data types. In C++, when someone declares a variable they must explicitly state the type for that variable, and that variable’s type can never change. However, in JavaScript, programmers do not declare the variable’s when initializing the variable. Rather, the variable’s type is inferred during run time based on the value being passed into it. Additionally, a variable’s type in JavaScript is dynamic, which means that its type can change at any time during run time whenever a new value is assigned to the variable. The addition of this feature in JavaScript makes the programming language more flexible in terms of numeric calculations because calculating a floating-point number using integers does not require any casting.

1. Given JavaScript’s large support for strings, including regular expression support, it is an easier to use language for program’s centered around string manipulation. Additionally, support from many different APIs, like the Google API, gives JavaScript the ability to perform many different features, like the ability to create maps using GPS coordinates. However, one could argue that the number of features in JavaScript makes it an intimidating language to learn. Considering JavaScript’s built-in functions that simplifies common algorithms in C++, like breaking a large string into individual words, into one function call, JavaScript can be easier to implement in terms of number of lines of code. In terms of debugging, JavaScript is weaker because it does not catch errors unless it runs into them during run time since the code is not compiled until run time. Overall, JavaScript is a more powerful and simpler to use language if the user is an experienced programmer who can recognize the dangers of using the language. However, if the user is not experienced in programming, then they should not be using JavaScript because it is more complex to learn, and harder to completely debug.

**References**

Kienle, H. M. (2010). It's about time to take JavaScript (more) seriously.*IEEE Software, 27*(3), 60-62. doi:http://dx.doi.org.ezproxy.libproxy.db.erau.edu/10.1109/MS.2010.76

A Short History of JavaScript." *A Short History of JavaScript - Web Education Community Group*. W3, 27 June 2012. Web. 10 Apr. 2017.

*History of C++ - C++ Information*. (2017). *Cplusplus.com*. Retrieved 11 April 2017, from http://www.cplusplus.com/info/history/