User Interface Design

1. Introduction:

The purpose of this assignment is to investigate and evaluate few different tools for the development of user interfaces. The next four sections will briefly introduce the four tools I picked and investigate some of their advantages and disadvantages. The following section will include a table of comparative features of those four tools. The last sections will be a conclusion indicating which tools I would like to use for developing user interfaces this semester and the reasons behind it followed by a bibliography documenting the reference sources used in generating the report.

2. Java Swing:

Java is a free integrated toolkit that allows Java developers to build and test XML applications, Web services, and Web applications with the latest Web services technologies and standards implementations, and is available on http://java.sun.com/webservices/downloads/webservicespack.html

Java runs on the following platforms: Solaris, Windows, Linux.

Java supports accessibility, i.e. building the services and support into an application that enable people with disabilities to use the software. Java accessibility technologies are implemented in the Java Foundation Classes (JFC) and they comprise two separate packages: Java Accessibility API and the Java Accessibility Utilities.

Advantages of using Java as your User Interface development tool:

- Your programming language is object oriented, i.e. object-oriented concepts can be applied such as inheritance, polymorphism and abstract classes, yet it's still dead simple.
- Your development cycle is much faster because Java technology is interpreted. The compile-link-load-test-crash-debug cycle is obsolete--now you just compile and run.
- Your applications are portable across multiple platforms. Write your applications once, and you never need to port them--they will run without modification on multiple operating systems and hardware architectures.
- Your applications are robust because the Java runtime environment manages memory for you.
• Your interactive graphical applications have high performance because multiple concurrent threads of activity in your application are supported by the multithreading built into the Java programming language and runtime platform.
• Your applications are adaptable to changing environments because you can dynamically download code modules from anywhere on the network.
• Your end users can trust that your applications are secure, even though they're downloading code from all over the Internet; the Java runtime environment has built-in protection against viruses and tampering.

Disadvantages of using Java:

• Running Java-applications tend to use a lot of system-resources, even when they have to do only one small thing. Java client application consumes so many resources, that you can hardly run another application next to it. You certainly cannot run two or three big Java applications simultaneously.
• Excessive object creation can be a huge problem in Java programs. Despite continuing improvements for memory management, the object creation and garbage collection cycle is still a very expensive operation. If frequently used code creates excessive numbers of objects, performance can be slowed. This can be somehow handled by understanding how to reduce the volume of object creation in the code.

I did study Java few years ago in my undergraduate study and I programmed a game as part of my coursework. I haven’t used it though since then. I know it’s available on university computers and I’m almost sure that the instructors are familiar with it.

3. FLTK (Fast Light Toolkit):

FLTK is a C++ Graphical User Interface (GUI) toolkit for UNIX and Microsoft Windows. It was first introduced in 1998. The Fast Light Tool Kit ("FLTK", pronounced "fulltick") is a LGPL'd C++ graphical user interface toolkit for X (UNIX®), OpenGL®, and Microsoft® Windows® NT 4.0, 95, or 98.

A prerequisite for using FLTK is a foundation in C++ programming, most importantly an understanding of classes, inheritance, pointers and dynamic memory allocation.

Advantages of FLTK:

• It's Free Open Source Software under the GNU LGPL.
• Efficient and straightforward
• Uses C++
• Makes small statically linked stand alone executables that are fast
• It's cross platform with Linux/Unix, Windows and MacOSX (write once compile anywhere)
• Supports OpenGL
• Has a graphical user interface builder called FLUID
• Minimizes CPU usage (fast)
• Minimizes memory usage (light)
• Supports intelligent layout of widgets
• Support "schemes", "styles", "themes", "skinning", etc. to alter the appearance of widgets in the toolkit easily and efficiently. The purpose is to allow applications to tailor their appearance to the underlying OS or based upon personal/user preferences.
• In general terms, it’s friendly to beginners and is easily understood.

I could not find any real weaknesses or disadvantages for FLTK except for the fact that widgets are not referred by namespace but simple literal prefix “FL_”; functions and enums are though. Another weakness is the fact that FLTK’s internationalization capabilities are limited and not handled by FLTK but rather depends on OS.

I personally haven’t heard of FLTK up until now but from the articles and tutorials I’ve read online it sounds like an easy to learn tool and is straightforward to program, yet still fast, light and flexible.

4. OpenGL:

OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Developers can leverage the power of OpenGL across all popular desktop and workstation platforms, ensuring wide application deployment.

Supported on all UNIX® workstations, and shipped standard with every Windows 95/98/2000/NT and MacOS PC, no other graphics API operates on a wider range of hardware platforms and software environments. OpenGL runs on every major operating system including Mac OS, OS/2, UNIX, Windows 95/98, Windows 2000, Windows NT, Linux, OPENStep, and BeOS; it also works with every major windowing system, including Win32, MacOS, Presentation Manager, and X-Window System. OpenGL is callable from Ada, C, C++, Fortran, Python, Perl and Java and offers complete independence from network protocols and topologies.
Open GL Advantages:

- Industry standard:
  OpenGL is the only truly open, vendor-neutral, multiplatform graphics standard.
- Stable:
  OpenGL implementations have been available for more than seven years on a wide variety of platforms.
- Reliable and portable:
  All OpenGL applications produce consistent visual display results on any OpenGL API-compliant hardware, regardless of operating system or windowing system.
- Evolving:
  Because of its thorough and forward-looking design, OpenGL allows new hardware innovations to be accessible through the API via the OpenGL extension mechanism.
- Scalable:
  OpenGL API-based applications can run on systems ranging from consumer electronics to PCs, workstations, and supercomputers.
- Easy to use.
- Well-documented.
- OpenGL delivers fast and complete 3D hardware acceleration: Today's applications and games manipulate massive amounts of data in real-time by using OpenGL hardware accelerated geometry, real-time lighting, clipping, transformations and rendering.
- OpenGL makes real-time 3D effects possible without compromising performance.

I haven't learned OpenGL before but I've heard a lot about it and I would be very interested to learn more about it during this semester.

5. HTML (Forms)/ Perl/ CGI/ Flash/ Dream Weaver:

5.1 HTML Forms

HTML forms are another way of interacting with users and providing them with the space to enter information to be processed. I did write HTML codes before and I'm familiar to some extent with creating these forms. However to process information we need some sort of script to do this task.

5.2 Perl

Perl is a high-level programming language. It derives from C programming language and to a lesser extent from sed, awk, the Unix shell, and at least a dozen other tools and languages. Perl's process, file, and text manipulation facilities make it particularly well-suited for tasks involving quick prototyping, system
utilities, software tools, system management tasks, database access, graphical programming, networking, and world wide web programming. These strengths make it especially popular with system administrators and CGI script authors, but mathematicians, geneticists, journalists, and even managers also use Perl.

Perl is easy to start learning. if you've ever written a C program, an awk script, a shell script, or even a BASIC program, you're already part way there. Perl's learning curve is therefore shallow (easy to learn) and long (there's a whole lot you can do if you really want).

Perl builds and runs on a bewildering number of platforms. Virtually all known and current Unix derivatives are supported (Perl's native platform), as are other systems like VMS, DOS, OS/2, Windows, QNX, BeOS, and the Amiga. There are also the beginnings of support for MPE/iX.

Perl has the following advantages over other scripting languages and tools:

- Perl is widely available for many computer platforms.
- Scripts written for one computer can usually be used with little or modification on others.
- A great number of reference sources and other support information for Perl is available in books, on the internet, and on the World Wide Web.
- Perl combines the best parts of interpreted and compiled languages. Perl scripts do not need to be manually compiled and linked before being run; instead the Perl interpreter automatically compiles a script before running it. This means that Perl scripts with large sections of iterative code are run significantly faster than shell scripts and other languages that are only interpreted.
- Perl is derived from standard Unix tools such as Unix shell programming, the C language, _sed_ , and _awk_ , making it easy to learn by anyone who is already familiar with these tools.
- Perl is not as structured as the C language, but contains the flow control and array structures that are commonly used in C programming.
- Perl's regular expressions are similar to those used in a Unix system and are powerful enough to easily handle complex text and pattern searches. Perl often can do in a few lines what it would take a complicated C language program to do.
- Perl can easily read and write binary files, treating binary data as just another string, without the programmer having to know anything about the specific platform being used.
- Perl's associative arrays are very convenient for collecting and processing data for reports.
- Perl includes several commands that simplify the generation of reports.
Perl has the following disadvantages over other scripting languages and tools:

- Because each Perl script is partially compiled at run-time, it might not be as efficient as interpreted languages for small programs or for large linear programs that do not use loops.
- Typically, Perl offers a number of ways to do the same thing, and some ways are more efficient than others. A badly written Perl script can monopolize system resources while it runs.
- Perl's relative lack of structure makes it easy to write scripts that are unintelligible to other programmers (or even the original programmer after a period of time).
- Perl's syntax for identifying variables, arrays, and file handles can be confusing, especially to a novice programmer.
- Perl has a default and implicit argument for almost every operation, such as reading from a file handle or performing character substitutions. No error message is produced if you forget to include an argument for these operations since Perl automatically uses the default argument. This can lead to problems that are difficult to debug.
- Perl is not well suited for interactive use, such as logging in to a node or querying all the data cards in a node.
- Perl scripts are not precompiled to a binary form but are executed as ASCII text files. This poses security problems if your scripts include passwords, SNMP community strings, and other privileged information.
- Perl's popularity means a great many third-party Perl libraries and scripts are available, but care must be taken when using software that comes from an unknown source. The scripts might be accidentally or maliciously destructive and should not be run until completely understood.

I did write a small perl program with the help of a friend and it seems easy to learn but I wouldn’t call myself familiar with it.

5.3 CGI

CGI ("Common Gateway Interface") is a term used to refer to programs which run on the server, to provide interactivity to Web sites. CGIs can be written in any programming language, but most are written in either Perl (a non-compiled, or "scripting" language) or C++ (a compiled, or "programming" language).

Client-side applications are a popular alternative to CGI scripts. These utilize JavaScript, Java applets, or other programs which actually run on the visitor's own computer. On the plus side, such applications can greatly reduce the amount of processing your Web server has to do, and thus, can improve the performance of
your Web site across the board. On the minus side, though, many visitors can't or won't accept or run such applications.

CGI disadvantages:

- Problems mapping top-level URLs to CGI
- Difficult for script to get original URL and/or document directory
- Use of “STDERR” not defined
- Lots of support libraries
- Excessive use can hit a performance wall
- No persistent connection

5.4 Flash

Buying the Flash program is only necessary for Flash developers. Anyone that has the free Flash plug-in /Active-X installed with their browser can view flash movies.

Flash Advantages:

- Flash uses vector graphics, which means that the graphics can be scaled to any size without losing clarity/quality.
- Flash gives the viewer a "high-tech" impression of your organization that is very impressive.
- Flash can be used to create interactive animation for entertaining "Splash" pages, ad banners and even games.
- With Flash you can deliver to your audience engaging applications and web interfaces such as training courses, tutorials and presentations.
- When compared to animated GIFs, the advantages of Flash are that: Flash movies load much faster, allow interactivity and can use more than 256 colors.
- When compared to java applets, the advantage of Flash is: Flash movies are easier to create than java applets and are more stable in web browsers than java applets.

Flash Disadvantages

- There are a variety of reasons why Flash content may be inaccessible. First, Flash files require a plug-in called the Flash Player. Many older browsers do not support this plug-in so individuals using those browsers will not be able to view Flash content.
- The "Back" button does not work. If you navigate within a Flash object, the standard backtracking method takes you out of the multimedia object and not, as expected, to the previous state.
- Link colors don't work. Given this, you cannot easily see where you've been and which links you've yet to visit. This lack of orientation creates navigational confusion.
- The "Make text bigger/smaller" button does not work.
• Flash reduces accessibility for users with disabilities. (Although the latest version "Flash MX" supposedly addresses many of these issues.)
• The "Find in page" feature does not work. In general, Flash integrates poorly with search.
• Internationalization and localization is complicated. Text that moves is harder to read for users who lack fluency in the language.
• Many viewers get annoyed when they have to wait for the Flash presentations to load and will end up going elsewhere instead of waiting.
• Not free and rather expensive ($400)
• Flash is not very search engine friendly. Therefore, a Flash-based web site may not rank well in the search engine results
• It is more expensive and time consuming to update and maintain a Flash web site

5.5 Dream Weaver

Dreamweaver MX 2004 delivers powerful, standards-based controls to ensure high quality design. A design environment built around Cascading Style Sheets (CSS) enables faster and more efficient development of clean-coded, professional sites.

Macromedia Dreamweaver MX 2004 is $399 and the upgrade is $199.

Dream Weaver Advantages:

• Use a world-class design and code editor in one tool.
• Build sophisticated, standards-based sites with rich CSS support.
• Improve end-user experience with a dynamic cross-browser validation feature that automatically checks tags and CSS rules for compatibility across all the leading browsers.
• Save time with a built-in graphics editor. Crop, resize, and make minor edits without leaving Dreamweaver by using built-in Macromedia Fireworks technology.
• Create better user interfaces without constant browser checks with a Live Data View that allows applications developers to see live server-side data while in design mode.
• Accessibility Support Accessibility reference material is integrated into the Dreamweaver MX 2004 Reference panel. The Reference panel gives context and background on best practices in designing for accessibility according to U.S. law and international standards.
### 6. Comparison

<table>
<thead>
<tr>
<th></th>
<th>Java Swing</th>
<th>FLTK</th>
<th>HTML/Perl/CGI/Flash/DreamWeaver</th>
<th>OpenGL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Do I know how to use it?</strong></td>
<td>Yes</td>
<td>NO</td>
<td>SOME</td>
<td>NO</td>
</tr>
<tr>
<td><strong>Is it a new learning tool for me?</strong></td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td><strong>Is it available on university computers?</strong></td>
<td>YES</td>
<td>?</td>
<td>SOME</td>
<td>YES</td>
</tr>
<tr>
<td><strong>Do I have it on my computer?</strong></td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td><strong>Support handicapped users?</strong></td>
<td>YES</td>
<td>?</td>
<td>SOME</td>
<td>?</td>
</tr>
<tr>
<td><strong>How easy is it to use? On a scale of 1 (easy) to 5 (hard)</strong></td>
<td>3</td>
<td>Maybe 2</td>
<td>2</td>
<td>Maybe 4</td>
</tr>
<tr>
<td><strong>Preference?</strong></td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
6. **Conclusion**

After investigating and evaluating each of these 4 tools, I’m hesitant between OpenGL and FLTK for developing user interfaces although more leaning toward OpenGL. That’s because I’ve heard a lot about OpenGL online, friends and through job search but never had the opportunity to actually study it or even take a close look at it. All of these tools have advantages and disadvantages but I guess as a personal preference to learn something new I’m more interested in OpenGL.

7. **References**

- [www.Fltk.net](http://www.Fltk.net)
- [http://freshmeat.net/articles/view/928/](http://freshmeat.net/articles/view/928/)
- [http://opengl.org/resources/faq/getting_started.html](http://opengl.org/resources/faq/getting_started.html)
- [http://java.sun.com/docs/white/langenv/Intro.doc.html#318](http://java.sun.com/docs/white/langenv/Intro.doc.html#318)
- [www.perl.com](http://www.perl.com)
- [http://www.echoecho.com/flashbasics.htm](http://www.echoecho.com/flashbasics.htm)
- [http://website-design.anandsoft.com/flash-web-design.html](http://website-design.anandsoft.com/flash-web-design.html)
- [http://www.kestrel-designs.com/text/articles/flash_good_bad.html](http://www.kestrel-designs.com/text/articles/flash_good_bad.html)