Using Open Source Software to Build a C++ Teaching Aide

Elizabeth Heithoff

University of Iowa, Iowa City, Iowa
BethersUI@aol.com

Introduction

Computer programmers are essential to the technological advances of a company. However, they require system administrators who are knowledgeable about the systems on which the programs run. Introductory computer science courses primarily focus on programming. In order to gain experience with commercial hardware and software, it is necessary to perform an independent honors project or work in a business environment. This honors project will provide practical experience in hardware setup and configuration. It will also provide exposure to infrastructure-related software applications such as Apache, MySQL, and the Linux OS. Lastly, it will provide an opportunity to develop a complete web application that will directly use the infrastructure. While providing many additional educational experiences to supplement course work, the project also has practical importance. It will be used for the instruction of students in the Computer Science department. The application will support the teaching of C++ to Computer Science 3 students by creating a challenging and exciting project as the final assignment in that class. The remainder of this proposal will examine motivation for the project, the project architecture and the associated details.

Motivation

In an ideal situation, programmers within a company that create and maintain applications would completely understand the platform on which the programs run. However, in many large corporations, there are two completely separate sides of the IT department; the developer side and the architecture side. One of the major problems that employees on the architecture side encounter is that programmers do not fully understand the web architecture. This can result in many problems related to security, efficiency, and reliability. This project allows the ability to learn how the separated tasks of the IT department can be brought together.

Currently, students of Computer Science 3 write several projects in C++. This web application will help to motivate the students to create more efficient projects and give them something to look forward to once their project is completed. It will also give them a chance to use their project above and beyond what was previously expected of

them. The students will be able to share the results of their code with other students. This will give them a chance to see what areas they lack in as a programmer, and will provide an opportunity for student-to-student constructive criticism. This project will provide a valuable teaching tool for the students of Computer Science 3.

Project Summary

There are two goals associated with this project. The first is to gain knowledge and experience with the infrastructure associated with web development. The second is to develop a web application which will allow students of Computer Science 3 to upload their code for *Critical Mass* and play their classmates on the Internet.

Project Details

There will be three components of this project. Project ideas are generated when a user makes a request. In this situation, there was a need for a web application to be used as a teaching tool. So the first component is the application, design, and development. It is this component that will be visible to the students. The students will be using the application for a game they will create called *Critical Mass*. *Critical Mass* is a variation of *Minesweeper* and is played by placing bombs on a game board, attempting to blow up all of your opponent's pieces. The students and the professor of this course need a web application that will allow the students to log on to the site. Then the students must be able to upload their code to be saved into a database. They also must be able to challenge other student's code, in a tournament environment. The application will also have other common options such as changing their password.

Once the student's requests have been evaluated, it is necessary to get a view of what the application will actually look like. When a student clicks on a link to this application, the first screen they will see is a login screen. Once properly logged in, the student will be presented with a variety of choices, including uploading their code, playing a game, changing their password, and checking their grades. They will also be able to see directly where they stand compared to other students in the class as far as who has won the most games. They will be able to challenge students within five rankings of themselves. In order to challenge other students, they will click on the student's name. This will trigger a game to be played. This game can be played in many variations, including computer versus computer, computer versus human, and human versus human. The application will initially allow a student to challenge another person's program. At this point, the game will be computer versus computer. In the future, it would be valuable to add the possibility of allowing the person making the challenge to physically play the other person's program (human versus computer). It would also be beneficial to allow students who are on the web site at the same time to play each other (human versus human). Once the game is finished, the student can request to see what went on during the game. The game will then be played back for them with a similar GUI as in their real program.

Once a problem is identified, and specifications are made about what must be done in order to suit the user's needs, it is necessary to identify what tools are required to support the application. The second component to this project is the hardware and software infrastructure that supports the application. This involves all the components of the application that need to be created, and also how they work together. This is illustrated in Figure 1. The student login information will be stored in a MySQL database. Also stored in the database will be information including their number of winnings and their program. When a student requests to challenge another student, many actions will occur. The Java applet will trigger another program that we will call the game controller. The game controller will interact with the database to retrieve code from both students participating in the game, compile it, and run it. After the game is played, the game controller will return a string to the database containing all the moves that were made throughout the game and the outcome of the game. This way, once the game is finished, the student can opt to play back the game. All basic web pages will be done in HTML, and the student can view game results using a Java applet. This Java applet may later be extended to allow the user to view games in progress, and allow human-vs.-computer, and human-vs.-human games to be played.

While developing this application, one could expect to come across many challenges. The first is to find out how to communicate between a program written in C++ and a MySQL database. Research will be necessary in order to find out which library makes this possible, and then learn the syntax which will be used to actually manipulate the database. Another possible challenge is to allow students to play each other manually. This is a challenge because it is difficult to link together HTML pages, Java applets, C++, and Java all into one coherent system.

Lastly, once it has been decided what languages the program will use, it is necessary to choose the appropriate platform on which to run the program. The first step is to acquire and assemble the parts necessary for a high-performance machine. The infrastructure configuration will consist of many parts. In order to write the programs, a basic editor will be used that comes with Linux (gedit). Apache is an open source program which will be used as a web server for this project. The database will be MySQL. Finally, the operating system on which the whole project will be run is Linux. Linux is also open source software.

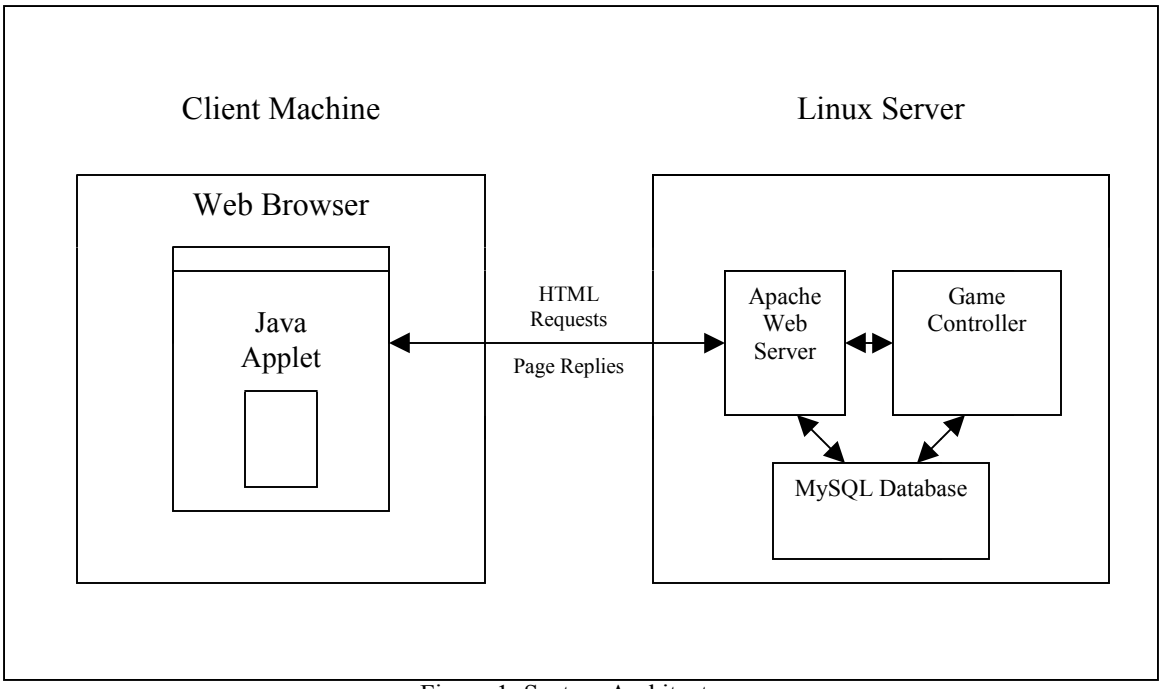

Figure 1: System Architecture

Project timeline

- ✓ To be finished in two weeks: Acquire and assemble a computer without an operating system, partition the hard drive, and install both Microsoft Windows XP and Red Hat Linux version 7.3.
- ✓ To be finished in three weeks: Install Apache and write the basic HTML pages that will be displayed in the application.
- ✓ To be finished in one month: Create a MySQL database for storing student application information. Enable the login screen to work with the database.
- ✓ To be finished in the remainder of the term: Write the application to support the games.
- ✓ To be finished during the term of use: Write a thesis on my project and maintain the website throughout the semester that it will be in use. During this time, the project may expand to have human versus computer or human versus human capabilities, or changes may be made as necessary. Further, a survey of the effectiveness of the application as a teaching tool will be performed.

Conclusion

In order for a programmer to be successful, it is useful to fully understand the architecture on which programs run. This project will provide the opportunity to completely create a program, from building the computer to installing the software that it will run on. Most importantly, this project will provide a teaching aide for 22C:30 which may be used for years to come. It will help to teach students the effectiveness and efficiency of their programming, in an interesting way.

References

[1] Peter Wainwright. Professional Apache 2.0. Wrox Press Ltd, Birmingham, UK, May 2002.

[2] Ben Laurie & Peter Laurie. Apache – The Definitive Guide. O'Reilly, Sebastopol, CA, February 1999.

[3] James Goodwill. Apache Jakarta – Tomcat. Apress, New York, NY, 2002.

[4] Christopher Negus. Red Hat Linux Bible. IDG Books Worldwide, Foster City, CA, January 1999.