

## Development of a 3D Chess Game for *Tango*

James Z. Liang, Marek Podgorny

### I. Introduction

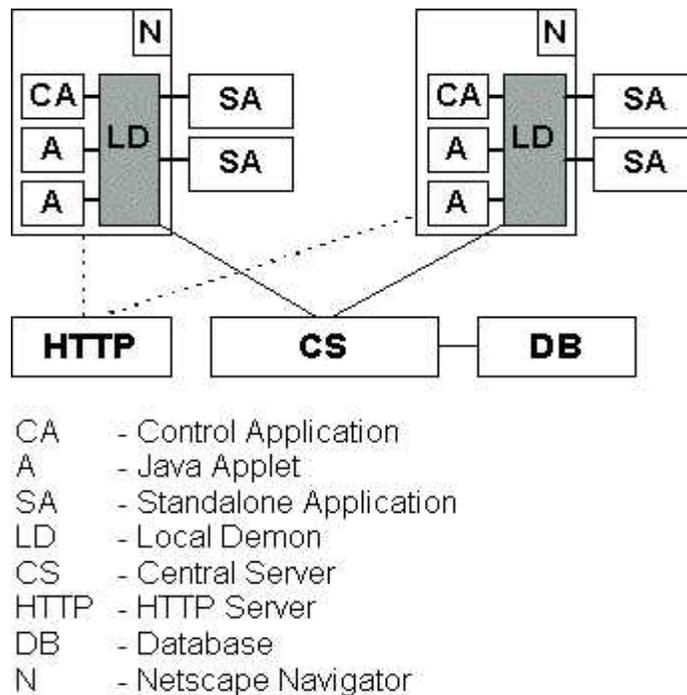
#### *Background*

The World Wide Web is a place where information can be easily accessed and exchanged by many people all over the world. In the beginning, this exchange has taken place through the use of simple web documents such as HyperText Markup Language (HTML). This language allows the publishing of information, but is as limited as other existing forms of media such as television or newspapers. It was effective for one way distribution of information, but it was stateless and did not involve any interaction between the server and the user. The introduction of the Common Gateway Interface (CGI) extended the web capabilities and allowed two-way communication with the user and the server, but there was still the problem of very little interactivity.

The solution to this problem was made possible when Sun Microsystems introduced Java. This new programming language introduced the idea of executable modules called applets that could be included with HTML, downloaded, and then executed. The web could now be used as a true interactive environment between the user and the host, but interaction between users was still very lacking. As time passed, new web capabilities such as teleconferencing and online chat were introduced, but this usage was still very limited and did not allow cooperative work. Existing programs require all users to be using the same application window and viewing the information in the same way in order to have information sharing. This was ineffective because not everyone wants to view the data in the same manner. A better system needed to be developed.

#### *Tango*

The system that developed from this need is *Tango*. Using Java and C++, a development team [1] at Syracuse University created *Tango* to be a collaborative system. This collaborative system can run multiple applications simultaneously through Netscape Navigator and allow communication between the application and the local demon called the application manager (see Figure 1). Using *Tango*, it is now possible for two people to share information in a much more effective manner.



**Figure 1**

When a user goes to the web page with the *Tango* server, it will call the *Tango* plugin to load up the control application that starts up the local demon. Java applets can be downloaded from the http server as the user calls the program and they will be run on the user's computer through Netscape Navigator. Messages can then be sent from the Java applet to the local application demon and then to the central server. Depending on the message, the central server may decide to process the command of the message, send the message to the other user or users, or both. There may also be applets that allow users to modify information that is stored in the *Tango* database. Such changes would be noticed immediately by the other users. In addition, the database is also where all the Java applets and the server information are stored.

Although the platform for collaborative information sharing on the net is now set, there are still few programs that utilize its potential. This is why I chose to write my 3D application. Since there was only one existent game for *Tango* my advisor, Marek Podgorny suggested that I write a chess game.

## II. Method

### *Working with Java*

In order complete my project, I had to become familiar with Java. Since I had never used Java before I began working on this project, I needed to learn it before I could start. My previous experience with programming was primarily with C++. Since it is so similar to Java, I was able to learn the language relatively quickly. I used online documents such as ones at Sun's homepage to get source information. With this and my reference book[2], I was able to learn

Java in just a few weeks. To help me learn the language I created simple programs such as an animation of a star and a simple tic-tac-toe game.

### *Working with VRML*

The chessboard and pieces that I used for my game were drawn by another VRML artist, but I would write the interactive part. Despite this, it was still vital for me to become familiar with VRML so that I could understand and use components such as touchsensors, timesensors, and positionInterpolators. Using Mark Pesce's book about VRML1.0[3], I began creating VRML *worlds*. First, I followed the examples in the book, but as I became more familiar with the language I created some of my own works. I started with simple objects like a small sphere, and moved slowly up to creating a desk with a chair next to it. When it came time to create more dynamic objects I switched to VRML2.0. I used ISM's Interactive VRML tutorial[4] to learn the dynamic components step by step. As I became more experienced, I created more complicated *worlds* such as one of the Earth, with the moon rotating around it, rotating around the sun. When I felt I was ready, I began looking into EAI

### *Working with EAI*

The first thing I noticed about the External Authoring Interface as opposed to everything so far was that there was very little documentation on it. It was very new and I could only find one site on the entire web that had source code containing EAI components. Once I located the example source code, EAI was surprisingly simple to get started on. It had several components that were a little confusing and it still has some bugs that make my work a lot tougher, but it was not as hard as I had feared. I started with a simple program that would create the earth when I clicked on a "create earth" button on my Java panel. I worked along slowly and eventually progressed to making a VRML chess game without *Tango* support. With that done, it was time to begin *Tango* implementation.

### *Working with Tango*

To help me learn the *Tango* API, I implemented my tic-tac-toe game for *Tango* so that I could become accustomed to its workings. The *Tango* API was not at all difficult to work with. I had to include a few extra classes in my implementation and work with the interface. The interface that I used consisted of a send method and a receive method. The send method was used to send information or requests encoded in bit format and the receive method was used to receive and decode messages. Each message contained identifying information such as its source and its type as well as data. In addition to those two methods, I also needed to modify my tic-tac-toe game to manage game technicalities such as turn taking, and selecting who gets to choose X's or O's. Other than these changes, the rest of the game such as the rule enforcement segments and the graphical interface methods remained unchanged. Once I had finished testing my *Tango* tic-tac-toe, I felt ready to begin making *Tango* Chess.

Modifying the chess game was not very different from modifying the tic-tac-toe game. There were a few extra options I added like an observer mode and the ability for a person to become the master of a chess session while it is in play, but other than that, that modifications were almost exactly the same as those that I had made earlier onto my tic-tac-toe game.



Figure 2

### III. Conclusions

#### *The Final Results*

The finished product for this project is my VRML chess game[5]. There are currently two versions of the game. One that works for *Tango* and one that does not. The non-*Tango* version allows people to play chess on one computer against an opponent sitting at the same computer. The *Tango* version allows people to logon to *Tango* and play against other people on the web. There is no limit to how many people can be in one session, but only two players may actively participate in one session at one time. If more players join the game, they will simply be observers and watch the game. They may navigate through the virtual chessboard as they wish, but only the two players may move their appropriate pieces. At any given time, one of the players may choose an opponent out of the observers to be the next player.

#### *The Research Experience*

Before I started on my work this summer, I had a set of goals that I was hoping to reach. First, I wanted to gain more knowledge about the field of computer science. I wanted to know what it was like to work outside of a classroom environment on my own project. Secondly, I wanted to get a chance to work on something new and exciting that no one else has really done before. I did not really expect to come up with revolutionary new ideas that would change the world, but I

did not want to be a mimic either. I also wanted to have something to show for my work after the project was finished. I wanted something tangible that I could have after the project was over.

I am happy to say that I have been able to reach my goals for this program. I have had a chance to experience what it is like to work on a research program. I got to learn much more about the World Wide Web in learning Java and VRML. I worked on a new and very interesting project called *Tango* and created one of the first EAI programs and the first VRML program for *Tango*.

### *Possible Future Projects*

One future project could be the modification of my chess game to have more options. These modifications could include more versatility such as having a game log or being able to save in the middle of a match and continue it on another date.

Another project that may arise might be a VRML chat program. People would be able to actually walk through the chat rooms and visualize who they are talking to. Such a program could also make it easier to screen out those you do not want to listen to. If you only hear people who are near you or facing you, it could make online chatting more organized and a lot easier to use. As more applications are developed, *Tango* will take a larger role in providing web communications and collaboration.

### References

- [1] "Tango-A Collaborative Environment for the World-Wide Web", an online paper by Lukasz Beca, Gang Cheng, Geoffrey C. Fox, Tomasz Jurga, Konrad Olszewski, Marek Podgorny, Piotr Sokolowski, Tomasz Stachowiak, and Krzysztof Walczak, <http://trurl.npac.syr.edu/tango/papers/tangowp.html>
- [2] Flanagan, David. *Java in a Nutshell*. Sebastopol, CA, 1996.
- [3] Pesce, Mark. *VRML: Browsing & Building Cyberspace*. Indianapolis, IN, 1995.
- [4] VRML-Interactive Tutorial, an online document by Interactive Systems & Multimedia. <http://sim.di.uminho.pt/vrml/>
- [5] 3D Chess with Tango support by James Liang. <http://www.npac.syr.edu/REU/reu97/www/ce97jzl/chess/>