“Five in a Row” Class Project

CS43600 Principles of Computer Networking

Myson Burch

Nick Baker

[mcburch@iupui.edu](mailto:mcburch@iupui.edu)

[nvbaker@iupui.edu](mailto:nvbaker@iupui.edu)

Myson Burch University ID: 0003198940

Nick Baker University ID: 0003246911

28 November 2016

The goal of this project is to create a peer-to-peer networking application that two peers can use to play Gomoku or the “Five in a Row” game. The game needs to use HTTP protocol to implement the game in real-time and also Wireshark needs to be used to keep record of the network protocol activities for both players during the game.

We used Node.js in our implementation of the game. Node.js is an open source, cross-platform Javascript development environment with many tools and libraries that make this a feasible approach to making this network application. Node.js is event-driven which seems natural to use in a click-based game where there are many input/output operations with real-time communication between the peers. Node.js is compatible across OS X, Linux, Windows and other operating systems. Node.js is also lightweight and efficient which makes it convenient to code in especially with so many open source libraries we have access to. The library that we are interested in is the websocket library. WebSockets are a bi-directional and persistent connection from a web browser to a server that we will use to implement the network side of the application between the peers. This allows us to communicate with any number of open connections at a time since the websocket initiates a single running server, but for our purposes we will only need two connections. We are using GitHub to host and manage our project. Git is very useful in terms of version control, feature requests and error tracking. Using this application to host our project allows us to work more seamlessly as a group and is also convenient.

We decided to design the game using a simple game engine developed by Mr. Andy Harris at Indiana University-Purdue University of Indianapolis. As said previously, the peer-to-peer aspect will be handled using Node.js and the node.js websocket library.

Wireshark was used in our project to analyze the network protocols in our application and what exactly was going on between the peers.

A common constraint on any project is time. We did not have as much time as we would have wanted to rigorously outline our design and implementation. Because of this, we tried to make the simplest implementation we could think of so we could get the game working locally on our machines then work on the network elements of the project afterwards. Another roadblock that we had to overcome was the actual design of the game. Obviously once a development stack is picked and certain design choices are made, certain issues are automatically introduced in terms of performance, compatibility, and overall feasibility of implementation of the task. The algorithm for checking wins can possibly be improved versus looping through an array to check for the winning conditions. Lastly, applying the websocket library to make the game a true network application between two peers was a constraint on the development. Since our group had not dealt with network applications before, this was completely new territory for us.

Despite any roadblocks we had, we were able to successfully implement the network application and enjoyed the entire process. There is much room for improvement for our short-term project management as well as project design and we will take lessons learned developing this project into other projects, collaborations, and assignments we may be a part of in the future.