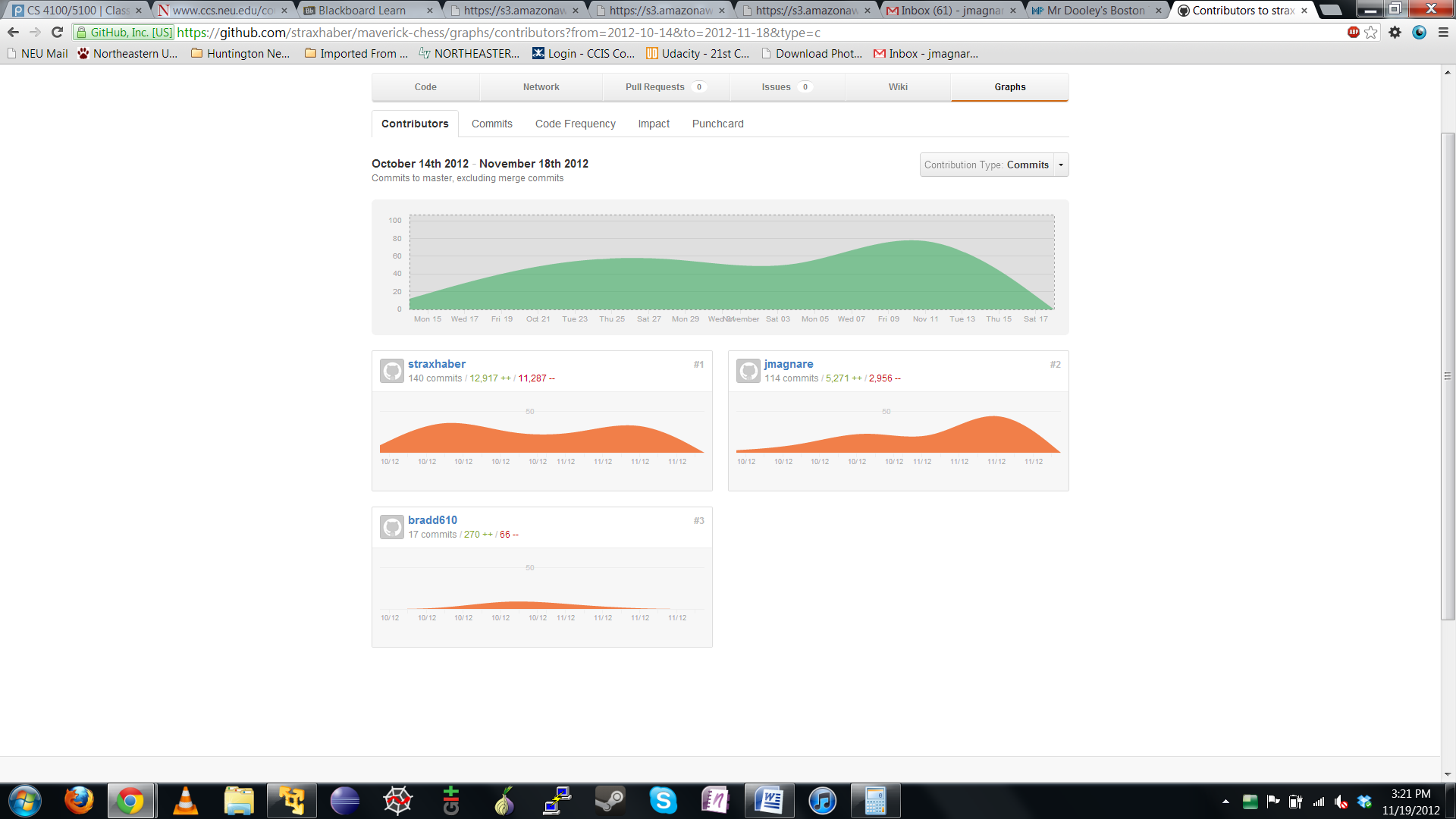
James Magnarelli, Matthew Strax-Haber, Brad Fournier

Professor Gillian Smith

CS 4100/5100

November 19, 2012

Final Project Midpoint Report

Our group has made significant progress toward completion of our final project. We have put in a number of long nights of coding, and our code repository sports 4,000+ lines of code written over the course of 325 commits. 

Commit history for our project, as available to project collaborators at https://github.com/straxhaber/maverick-chess/graphs

We have a functional, network-based chess tournament system with server-side rules enforcement and checkmate detection. We have also implemented a client library for the construction of AI players and a textual interface, the latter of which is complete. Though most elements of our project were developed collaboratively, a rough breakdown of individual responsibility follows:

* Matthew was chiefly responsible for the client/server interaction, the common AI/client code, and some significant stylistic refactoring, among other things
* James developed much of the server code, including the code for checkmate detection, legal move calculation, and board representation, among other things
* Brad worked on move enumeration and overall code style

The AI player is a work in progress, but is capable of enumerating legal moves and selecting them randomly. The only large, unimplemented piece remaining is the AI's quiescent search for possible moves, which is our current focus as a group.

Moving forward, we intend to devote some time to researching how others have implemented quiescent search for chess. We will then sit down to whiteboard a sensible approach, and make a first effort at implementing the search. We will have to conduct extensive testing to develop sensible heuristic weights for board likability evaluation, but this might be possible through the use of a genetic algorithm. Whether we take this approach to optimization of the weights will likely depend on time constraints. Overall, we feel that we are in a very good position to finish achieve our goals well before the end of the semester.

All of us on the team have learned a great deal so far. We have become intimately familiar with Google's Python style guide, we have worked with the Twisted server framework, and we have been forced to consider optimization of heuristic functions for rapid, repeated use. Tackling quiescent search will, most likely, be the project's most cerebral challenge. However, we are eager to tackle it and to utilize the tools we have built to construct a competent chess player.