James Magnarelli, Matthew Strax-Haber, Brad Fournier

Professor Gillian Smith

CS 4100/5100 – Foundations of Artificial Intelligence

November 19, 2012

Final Project Midpoint Report

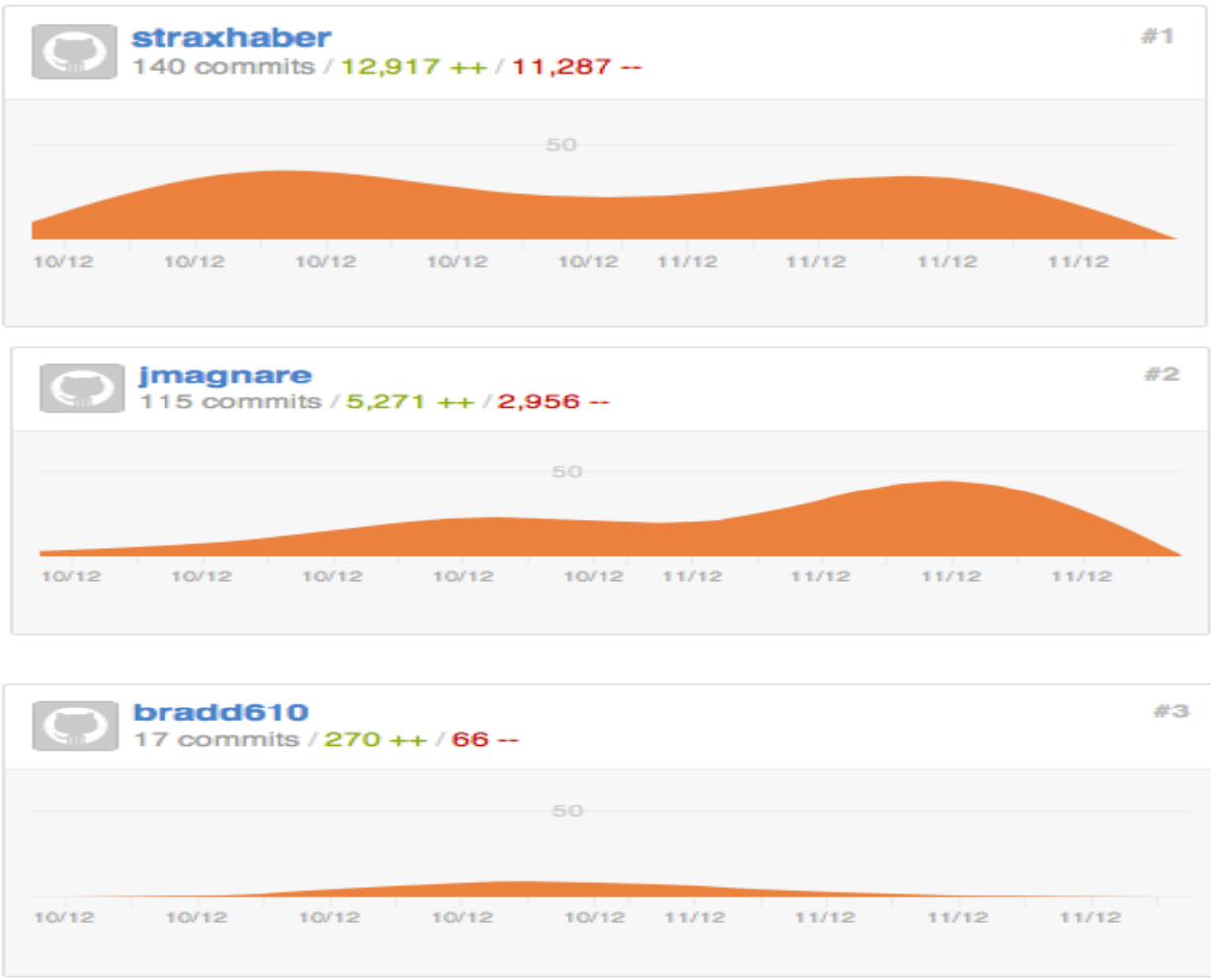
Current State

Our group has made significant progress toward completion of our final project. We have undergone many long nights of coding, and our code repository sports 4,000+ lines of thoroughly commented, documented, tightly written, and tested code written over the course of 325 commits. A few graphs of our progress can be found on the next few pages.

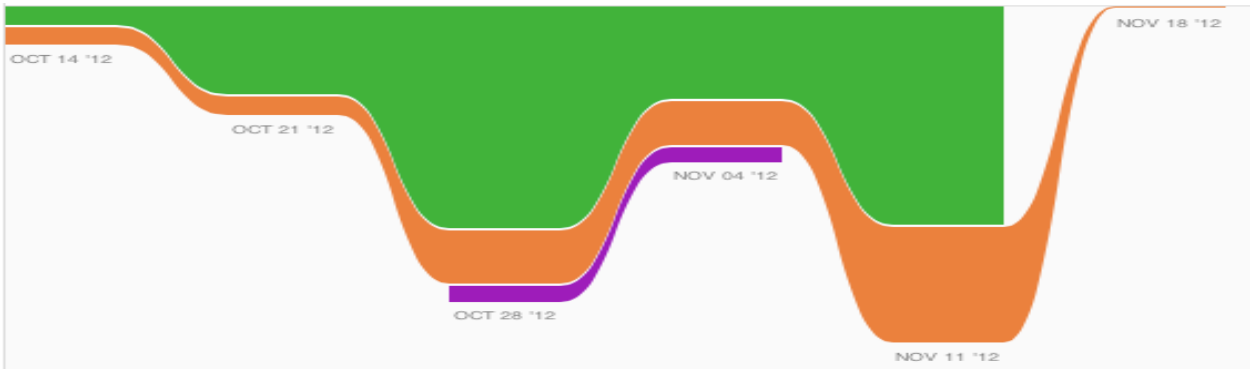
We have a functional, network-based chess tournament system with server-side rules enforcement and game management. We have also implemented a client library for the construction of AI players and a textual interface, both of which are complete. The AI, which is not complete, is built separately from the library it uses for communicating with the chess server, allowing for re-use of the libraries and server architecture without sharing AI techniques. Though most elements of our project were developed collaboratively, a rough breakdown of individual contributions follows:

- Matthew was chiefly responsible for architecting the system, client/server network code, AI/client libraries, move enumeration, and some significant stylistic refactoring, among other things
- James developed much of the server-side representation and manipulations, including the code for checkmate detection, legal move calculation, and board representation, among other things. James also helped with refactoring.
- Brad worked on move enumeration and overall code style

Contribution Statistics (from https://github.com/straxhaber/maverick-chess/graphs)



Commit history for our project (# of commits)



Graph of 'impact' on project (measured as lines added + lines deleted for all non-merge commits during each week period) — green is Matt, orange is James, and purple is Brad

Future Work

The AI is a work in progress, but we have developed a proof-of-concept AI that selects moves randomly from a list of enumerated moves. We seem to have a small and elusive bug in either the server-side legal move checker or the client-side move enumerator, but the 'random' AI is mostly functional. We have also developed a number of heuristics for analyzing board states, and these are mostly done except for fine-tuning the various heuristics' weights. 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 already have come up with a number of ideas for how to do this, but referencing best practices may aid in our development. We will then sit down to whiteboard a sensible approach combining our ideas and those we find, and make a first effort at implementing the search. We will also 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 achieve most of our goals well before the end of the semester.

All of us on the team have learned a great deal about both AI and software development so far. We have researched relevant AI concepts, such as heuristics for the game of Chess, game tree search (quiescence, iterative deepening, A*, etc.), and representation techniques for large game state trees. In addition to our research on relevant AI concepts, we have become intimately familiar with Google's Python style guide, worked with the Twisted server framework, and

considered the 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 semi-competent chess player that demonstrates many of the concepts we have learned in this class.