Richard Young
CS165A

MP2 Report

Architecture
There are a few classes in the project and are grouped into two sections: Player files and Game files. In the first grouping are Player, Human, and AI java files. Human and AI are subclasses of the parent Player class, which allows me to separate the technical algorithms required for the AI subclass from the more basic Human subclass that only really relies upon taking input and modifying the game board. In the other grouping are the Game and Board java files. Game is entirely dedicated to the manipulation of inputs, and hosting an environment for all of the other files to work together. It includes the declaration of the other classes and the loop it all runs in. The Board class deals with the manipulation of the game board, and it's method of printing that's required in order for a human to be able to play.

Search
The program employs a very basic minimax algorithm that includes alpha-beta pruning and a depth of about 9. The algorithm iterates through a list of possible moves up to 9 moves ahead, or less if the possible amount of moves is less, sorted by their position recursively taking the minimum cost choice each iteration.

Challenges
Implementing the algorithm in and of itself was not too hard, but having the algorithm work with the game was a challenge that I tried to solve by moving all of the unnecessary algorithm functions to the AI class. It was initially part of the Player.java file. In addition, working the actual arrays and setting a given position as taken proved to be a bit of a challenge as well as interpreting and displaying the inputs and outputs in a clean, easy to read format.

Weaknesses
This code is hardly optimized, and probably runs very poorly given a sufficiently large data set. In addition, there are definitely situations where the algorithm clearly misses something that I can take advantage of when playing. Making the minimax algorithm more rigorous, and therefore more intelligent would greatly aid the program, but given the time constraints it works well enough.