Gavin Bolin

CS470

Adversarial Search 1

1) The evaluation function is simply a function that checks for the number of connections for each player in the board. The check function checks for connections in each row, column, diagonal forward, and diagonal backward connections for each player piece. First it checks for connections of 2, then 3, and then 4. Each return of the count function returns the value of connections of that number (2,3,4) multiplied by 10 the power of the length of connections (10^2, 10^3, 10^4). Obviously if 4 is found then there is a win or a loss depending on which player has a 4 in a row in the certain iteration but also drastically affects the value that is returned by that sub. The entire board is checked with a double for loop for the specific players pieces.
2) For depth limit of 1 and 2 it was almost instantaneous for the beginning of the game. However, when depth was increased to 3 and on it took longer as the sheer number of branches would increase exponentially. Using depth of 4 would result in each choice taking around 5 seconds and around the same for a depth of 5. Some branches would be pruned but there is a lot more to compute in further-depth searches. The time was not that much more but was noticeably longer than the shallower iterations.
3) When playing against the ai, if I go first then no matter which depth level used, the ai would struggle to easily win. However, when going second and allowing the ai to go first it seems to have a greater edge when the depth level is higher, mostly after depth level 4. This shows that there is a heavy weight on going first in the game of connect 4.
4) The following graph shows the win rate of the random agent compared to the ab ai in 10 games. Each was run twice to get an average. It seems like further depth gives more of an advantage, but with a game with only 7 options at one point its hard to always ensure victory.

Random / ab d=1	Random / ab d=2	Random / ab d=3	Random / ab d=4	Random / ab d=5
5% win rate	10% win rate	5% win rate	5% win rate	0% win rate

5) When playing two agents that are different than each other the one that has a deeper depth will win most of the time however there was an instance with one ai with depth 2 that won over an ai with depth of 3. Similarly a depth of 3 and 4 won equally as well showing that two ai using similar depth searches did not have much of an advantage over each other. But when there was a bigger difference in depth the ai with the higher depth would win more often.
6) Initially I thought that there was more of an advantage to going first as there were games that the random player won even though it was playing against an ai. In the case of the different depth ai playing against each other, the lower depth level ai would sometimes win even if it was more than 1 depth level lower than the opponent. But when run the opposite, with the higher depth level would almost always have the advantage even if it wasn’t that much higher depth. An experiment I did was to put two ai against each other, one was depth of 3 and the other of 4. I tried putting them against each other with 10 games and seeing which would win when switching who goes first. But in this case they both won equally both times regardless of which

was run first. This would mean that the advantage of going first is not as influential as I thought, however is not negligeable.

7) This assignment was challenging when trying to make sure that the alpha beta algorithm was operating the correct way and making the evaluation function was confusing but just because of bug checking and making the best heuristic to evaluate the state of the board. If I were to do it again I would spend more time making a better evaluation function to future check for the other player and the potential scores they could make rather than just checking for your own score.