Project Report

Junda Lou

CS420 p3

The most important part in the minimax algorithm is the evaluation function, which is also the hardest part to implement. To make the function complete, I detect every possible situations.

1. 4-connect, with spaces to extend -> 5000 points
2. 3-connect, with spaces to extend -> 1000 points
3. 2-connect, with 2 spaces to extend -> 500 points
4. 2-connect, with 1 space to extend -> 300 points
5. 1 piece, with 3 spaces to extend -> 100 points
5. 1 piece, with 2 spaces to extend -> 50 points
5. 1 piece, with 1 space to extend -> 10 points

If it is opponent's turn, the points will be negative.
If there are, say 3-connect, but either it at the side wall, or shut down by the opponent, there is no positive advantage points for it, so for opponent's turn.

Implementing the Alpha-Beta Pruning, I combine the Min-value and Max-value function in a single big ab_prune function, which divides them using a if statement to make it easier. There are also 2 special parameters involving in the function call, which are time limit and depth. As required in the project spec, 5 pile depth is a minimum, and the time limit of generating next move is 30 seconds. This is implemented using clock provided in <time.h>.