AI – ASSIGNMENT 1

TEAM: BOT KYLE

MEMBERS : Kunal Vaswani , Shubhankar Bhagwat

SEARCH:

1. Monte Carlo:

- Each path to next node is assigned some probability, which changes as the game progresses.

-Monte Carlo would consider both exploitation and exploration factors to achieve a winning node.

-The Monte Carlo method, uses randomness for deterministic problems and thus is beneficial compared search algorithms.
It’s disadvantage is that it might take time to actually find the best possible path and we have a 24 second limit.
We may use pseudo random version with the help of our heuristic.

2. Minimax and Alpha-Beta Pruning:

-Minimax is very easy to implement thus we may prefer it over other search algos.

-We could make a variant such that difference between max and second max could be considered to use Minimax or go with Monte Carlo.

-Major problem is time limit. To improve it Alpha-Beta Pruning guidance by our heuristic and depth limited search may be used.

HEURISTIC:

To explain our heuristic we will have to define few terms which will be used in the formulation.

BW => Move resulting in winning a Big Board.
BoW => Move intended to stop opponent from winning a Big Board.
SW => Move resulting in winning a Small Board.
SoW => Move intended to stop opponent from winning a Small Board.
DiW => Move to prevent opponent from coming to the same Small Board.
BP => Move resulting in best progress towards big board win
BoP => Move to block best progress towards big board win of opponent.
SP => Move resulting in best progress towards small board win.
SoP => Move to block best progress towards small board win of opponent.
Sco => Move intended to maximize Score value of a smallboard win.

First conditions which governs the heuristic is that whether agent is in open move state or non-open move state.

Case 1: Open Move State
a)Firstly Choose which Smallboard to to play on using Following Formula

$$H(step1)=U1*BW+ U2*BoW + U3*BP + U4*BoP + U5*Sco$$

Ui are to assign utility and help heuristic judge the value of a node.
U1 and U2 are exponentially big as they are immediate priorities.
U3 > U4 > U5 so as to assert weight of a choice.Same goes for Ui of “b” step written below.

Small Board with highest value of H(step1) is expanded first,next we choos which cell of the chosen small board to choose and expand state.

b)To choose the cell use the following formula =>

$$H(step2)= U1*SW+ U2*SoW + U3*SP + U4*SoP$$

Case 2: Non-Open Move state
a)Firstly Choose which of the 2 Smallboards to to play on using Following Formula

$$H(step1)=U1*BW+ U2*BoW + U3*BP + U4*BoP$$

Ui are to assign utility and help heuristic judge the value of a node.
U1 and U2 are exponentially big as they are immediate priorities.
U3 > U4 > U5 so as to assert weight of a choice.Same goes for Ui of “b” step written below.

Small Board with highest value of H(step1) is expanded first,next we choos which cell of the chosen small board to choose and expand state.

b)To choose the cell use the following formula =>

$$H(step2)= U1*SW+ U2*SoW + U3*SP + U4*SoP$$

Examples-

For BW: one move away from winning the board.
For BoW: one move away for opponent to win the board.
Similarly for rest.
For BoP: say opponent has won the top right and top middle small boards then we can be sure that next move will be in the top left thus we’ll try to block it’s progress in top left small board.

Reason for competence of heuristic is that our heuristic design is very flexible. The cases spanned by terms like BoP and SoP can be changed easily as the definition is wide enough to include them.