Assignment 2
Group 16
Tanmay Ghode (cs17m055)
Athul Vincent(cs17m054)

Algorithm 1 Algorithm for AlphaBeta

AlphaBeta (j, α, β)
  /* To return the minimax value of a node j */
  /* Initially $\alpha = -LARGE$, and $\beta = +LARGE$ */
if $Terminal(j)$
 then return $e(j)$
 else if j is a MAX node
  then for $i \leftarrow 1$ to b
   do $\alpha \leftarrow Max(\alpha, AlphaBeta(j_i, \alpha, \beta))$
    if $\alpha \geq \beta$ **then return** β
    if $i = b$ **then return** α
  else /* j is MIN */
   for $i \leftarrow 1$ to b
    do $\beta \leftarrow Min(\beta, AlphaBeta(j_i, \alpha, \beta))$
     if $\alpha \geq \beta$ **then return** α
     if $i = b$ **then return** β

AlphaBeta Algorithmt like works just like Minmax Algorithm. If has α and β which are bounds and searches till when we are in this bound. It is important that better moves are generated earlier. AlphaBeta does this and hence removes the over computation of minmax.

We have set the plies for searchinbg the game tree. 7 plies are used considering the timeout condition for the seepest search. We will use the encountered condition if we get the good result earlier.

Heuristic Function
Coin Parity : Calculates the difference between min and max player.

CP = 100 ∗(Max Player coins - Min Player Coins)

(Max Player Coins + Min Player Coins)

Mobility : Calculates the relative difference between opponent and his own possible moves. We intend to have the more Mobility as it will be helpful in improving our performance.
if (Max Player Moves + Min Player Moves != 0)
{
Mobility= 100 ∗(Max Player Moves - Min Player Moves)

(Max Player Moves + Min Player Moves)
}
else
{
Mobility = 0
}
Combination of the following heuristic functions used is in ratio 1:7 respectively.