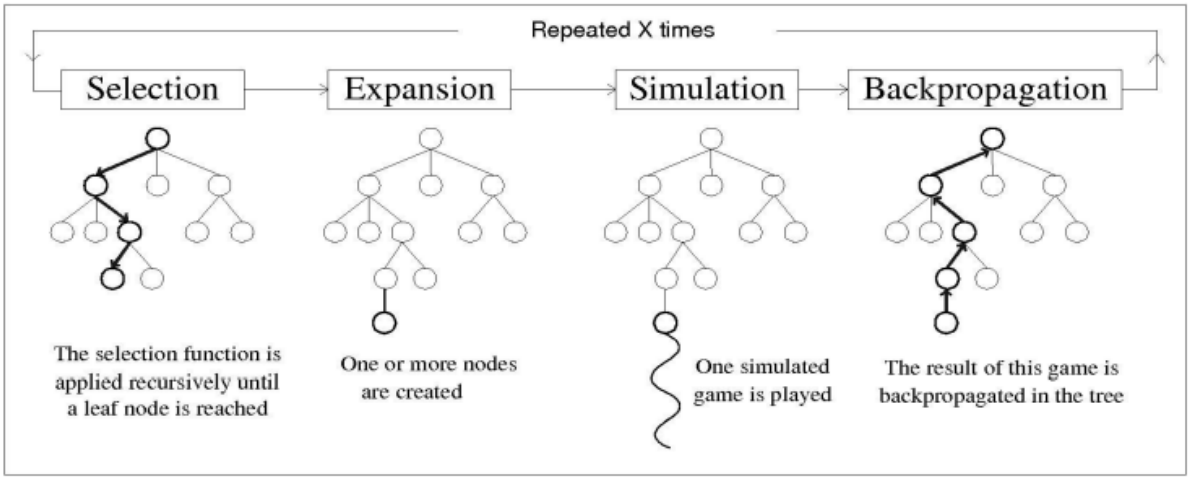
TCG-Project4-Hollow Nogo Report

Student: 翁玉芯 ID: m093875

Part I : The Used Method



First, construct a class for nodes, with its current board state, # of wins, # of visit time, child node, and parent node as node attributes.

Next, implement the basic Monte-Carlo tree search in agent.h, it consists of 4 parts:

1.Selection -> 2.Expansion -> 3.Simulation -> 4.Backpropagation

-Selection:

I use this UCB formula :

score = node.win/ node.visit + sqrt(2*log(parent.visit)/node.visit)

to calculate UCB score of the node, the selection starts from taking the root of the tree as current node, than search all the children of current node and choose one having the largest UCB score, take this node as the current node and search again, repeating this search routine until there's no child of current node.

-Expansion:

If the node has unexpanded legal move, then expand it and return the expanded child node, otherwise, return itself.

-Simulation:

Randomly simulate the node until the game is over, the return the winner of this simulation

-Backpropogation:

Based on the simulation result, we update the nodes' # of visit and # of win from the leaf.

Finally, implement the user control of searching cycle using the command:

--[player]=”mcts N=[# of searching cycles]”

Part II : Improvement

In this project, I’ve just implemented the basic version of monte-carlo tree search, and the user control of searching cycles, I would like to try implementing user control of search time per move or time management in the future.