1. (i)Finite State Machine

(ii) FSMs are implemented using enums and switches

Enum States {standing, searching, chasing}

Enum Transitions {seeSomething, hearSomething,none}

There are 3 phase – searching phase(determines current transition)

* Thinking phase(determines current state)
* Acting phase (performs actions based on current state)

Switch(currentState){

Case States.standing:

Switch(currentTransition){

Case Transitions.seeSomething:

Return states.chasing

Case Transitions.hearSomething

Return states.searching

Default

Return currentState

}

//cases for remaining states

}

(iii)hierarchal FSM – FSM within an FSM. Ie patrolling transitioning to attacking, attacking is its own FSM

Markov – adds randomness

1. (i)Search tree – a tree that shows all possible moves and outcomes for a game

Utility function – adds numeric values to the end points of a game. i.e. win=1 draw=0 loss = -1

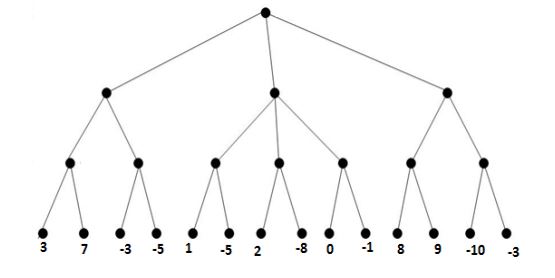
Min Max algorithm – applies numeric value to the nodes(individual moves)

(ii)to many possible moves, become to complicated

(iii)Evaluation Function searches all nodes of a tree to to the ply depth to determine which move will give the best profit at that point

(iv)pruning and symatry

(v)



-3

0

-3

7

-3

1

2

0

9

-3