Isolation Game

Udacity Artificial intelligence nanodegree

Abdelrhman-Yasser

25th January, 2019

Introduction

In this report we will evaluate implement a game agent that plays isolation game and analyse performance of this agents against several agents.

Problem : Isolation game

1. Initially two players are at random place on 9*11 board.
2. Players take turns to move from their place using chess horse movement.
3. Each time player move to a position on board this position become closed and no one can move to it again .
4. Player who have no move loses the game and other win.

Search techniques used

1. Minimax with alpha-beta pruning
2. Minimax with alpha-beta pruning and dynamic programming

Heuristics

I. zero_sum_h:

- This heuristic is based on idea that if player has more available moves than other player he has upper hand in this situation.

$$H = own\ moves - 3 * enemy\ moves$$

II. keep_near_center_h:

- This heuristic modifies a corner case in previous heuristic where sometimes previous heuristic gives a zero estimation when own moves equals 3 * enemy moves, in that case we will favourite going near center as long as the game is in the begging.

$$H = -manhattan(player, center) / \text{turns played}$$

III.keep_near_center_h2:

- This heuristic modifies a corner case in previous heuristic where sometimes previous heuristic gives a zero estimation when we are near end of game where number of turns played increase, so in that situation we will favourite state with less obstacles around player.

$$H = -percentage\ of\ obstacles\ in\ 5 * 5\ grid\ around\ player$$

IV.keep_near_center_h3:

- This heuristic modifies a corner case in previous heuristic where i can be in position with no obstacles around me, so in that case will favourite going near enemy.

$$H = -manhattan(player, enemy)$$

Evaluation Performance (% win over 200 matches)

- Playing against random agent

heuristic / search	Alpha Beta	Memorization
default	98.1%	**99.2%**
zero_sum_h	94%	95.2%
keep_near_center_h	95.8%	97.8%
keep_near_center_h2	94.5%	96.5%
keep_near_center_h3	96.2%	97.0%

- Playing against greedy agent

heuristic / search	Alpha Beta	Memorization
default	**93.2%**	92.0%
zero_sum_h	89.2%	86.2%
keep_near_center_h	89.5%	89.5%
keep_near_center_h2	87.5%	86.0%
keep_near_center_h3	89.5%	90.5%

- Playing against minimax agent

heuristic / search	Alpha Beta	Memorization
default	75%	78.5%
zero_sum_h	78.0%	70%
keep_near_center_h	76.0%	**79.8%**
keep_near_center_h2	64.8%	65.2%
keep_near_center_h3	75.5%	75.0%

Conclusion

- Default heuristic seems to have better performance against simple agents as greedy and random agents, but my agents have better performance on playing against smart agents as minimax

- Time don't really matter with my agents as i am putting limit on time of taking an action to be nearly less than 1 sec.