Final Project Write-up

1. Model structure: iterative deepening alpha-beta pruning search

It combines depth-first search's space-efficiency and breadth-first search's completeness. So we will search for the best choice by increasing depth and exit program if processing time reaches the limitation. Alpha–beta pruning will decrease the number of nodes that are evaluated by the minimax algorithm.

2. Heuristic/Evaluation function: Voronoi algorithm and flood-fill algorithm.

Tron game can be divided into two parts: trap and survive.

a. We use BFS to detect if the two players are connected.

b. In early state, two players are connected on the graph, so we try to trap the opponent by Voronoi algorithm. The Voronoi algorithm works like this: for each spot on the map, find whether we can reach it before opponent does. If we can reach the spot early, we are more likely to move to the spot. I use queue to boost Dijkstra algorithm and we will get the Manhattan distance of every spot.

c. Once we cut the opponent off, two players are no longer in the same connected component and then heuristic function switches to “survive mode”. Flood-fill algorithm evaluate how many accessible spots can be reached. Here I only implement a simple version which searches in our minimax tree to calculate the accessible spots number of certain depths.

3. Result: The player agent can beat 95% of random player on different maps.

4. Existing problems: the reason of losing game is that sometimes the AI player will go into the smaller space and get into trap since Voronoi algorithm encourages AI player to move to the spot that reach early than our opponents.

CHEN HAIBIN

2019/8/16