HW 3 Report

William Chen

- Kalah game description:

  According to Wikipedia, Kalah game was developed in the United States by William Junius Champion, Jr in 1940. The game provides a Kalah board and a number of seeds, the board has 12 pits that can hold seeds and a Kalah on each side. The goal is to have more seeds than the opponent.

  Each player can choose one of their pits and take all the seeds, sowing them counterclockwise into the pits and their own Kalah, not including the opponent Kalah. If your last stone lands in your own empty Kalah, you get an extra turn. If it lands in an empty pit on your side with seeds on your opponent's side, you put the last seed and the opponent's side of seeds to your Kalah.

  The game ends when one player has no seeds on the side. The opponent gets all the remaining seeds.

- Heuristic function

  1. Simple Method

     This method evaluates the game state based on two Kalah difference.

     H(s) = Your Kalah(a_fin) – Opponent Kalah(b_fin)

     The idea behind this heuristic function is to encourage your AI to choose the way that can create the biggest positive difference.

     For example,

ex 1.	ex 2.
Your Kalah: 20 seeds.	Your Kalah: 18 seeds.
Opponent's Kalah: 15 seeds.	Opponent's Kalah: 17 seeds.

     The example 1 of (20 – 15 is 5), which is a positive number and greater than example 2 of (18 – 17 is 1) then the AI will choose the first one to move.

  2. Advance Method

     This method evaluates the game state based on two Kalah difference and the difference in the total number of seeds on your and your opponent's sides.

     H(s) = Your Kalah(a_fin) – Opponent Kalah(b_fin) + (Sum(a) – Sum(b)) * 0.3

The idea behind this heuristic function is to reinforce the simple method without considering the other pits, which may lead to a short-term focus. As for the 0.3 weight, it is being experienced to perform the best.

ex 1.	ex 2.
Your Kalah: 20 seeds.	Your Kalah: 18 seeds.
Opponent's Kalah: 15 seeds.	Opponent's Kalah: 17 seeds.
Your pits sum: 17 seeds.	Your pits sum: 27 seeds.
Opponent's pits sum: 20 seeds.	Opponent's pits sum: 10 seeds.

The example 1 of (20 – 15 + (17-20) * 0.3 is 4.1), which is smaller than example 2 of (18 – 17 + (27 - 10) * 0.3 is 6.1) then the AI will choose the second one to move.

- Program Design

  The program has four main functions.

  1. move

     The main.py will call this function for the move that AI makes. The move will call the minimax function with the parameters state, depth, alpha, and beta, is_maximizing to compute the heuristic value.
  2. minimax

     This function implements alpha-beta pruning by calling the heuristic_function function with the index position to compute a value based on the heuristic method being used. The function will be recursive until the depth is over the designated value or the game is over.
  3. kalah_move

     The function implements the next move of the Kalah game, returning the state of the next state to the minimax function. This helps the AI to determine which to move by computing up to 15 depth of further state of each pit.
  4. heuristic_function

     Compute the heuristic value based on the heuristic method and return it to the minimax function.

- Result

  1. Time for different search depths:

I had run for more than 15 times that average the search time per each depth.

Search Depth	Search time (second)
3	0.000191
5	0.000453
7	0.001387
10	0.001767
15	0.004834
20	0.043227

My device CPU speed is 3.4GHz

2. Against the local AI

I had test for each heuristic function with different max level 10 times.

Heuristic function	Against Human win rate
Simple Method (depth = 3)	30 %
Simple Method (depth = 7)	60 %
Simple Method (depth = 15)	90 %
Advance Method (depth = 3)	40 %
Advance Method (depth = 7)	80 %
Advance Method (depth = 15)	100 %

The strength of the heuristic function increases as the depth grows, allowing AI to know the further result of the move. Furthermore, advance method truly gets better performance than the simple method. The seeds remaining in the pit on two sides play a crucial role in calculating the heuristic function.