Report

- Search strategy: Iterative deepening minimax with alpha-beta pruning
  1. Set time limit to 15 sec for a single move.
  2. Run iterative deepening minimax with alpha-beta pruning
     until TIMEOUT. (i.e. if TIMEOUT then break and return the best move of the deepest depth)
  3. Heuristic evaluation function
     For cut-off states, evaluate a heuristic.
     $h(state) = 30\times\Delta13 + 9\times\Delta8 + 5\times\Delta5 + 3\times\Delta3 + 3\times\Delta2$,
     where
     $\Delta13$ = (number of 13 in AI_cards) − (number of 13 in User_cards) +
     0.8 × ((number of 13 on board for AI) − (number of 13 on board for User))
     $\Delta8$ = (number of 8 in AI_cards) − (number of 8 in User_cards) +
     1 × ((number of 8 on board for AI) − (number of 8 on board for User))
     $\Delta5$ = (number of 5 in AI_cards) − (number of 5 in User_cards) +
     1 × ((number of 5 on board for AI) − (number of 5 on board for User))
     $\Delta3$ = (number of 3 in AI_cards) − (number of 3 in User_cards) +
     1 × ((number of 3 on board for AI) − (number of 3 on board for User))
     $\Delta2$ = (number of 2 in AI_cards) − (number of 2 in User_cards) +
     1 × ((number of 2 on board for AI) − (number of 2 on board for User))
  4. Utility function
     For terminal states, check winner and return utility.
     (1) If AI wins, then return 1000+score(AI)−score(User).
     (2) If User wins, then return −1000−(score(AI)−score(User)).
     (3) If tie is happened, then return 0. (score and the max of card of AI and User are the same)