Assignment 2: Report

Strategy

The strategy used can be summarized to perform the following:

- Remember what card has been discarded by both players each round
- Draw from discard if it forms more melds/potential melds in hand
- Store the cards which opposing player rejected and accepted
- Find the best meld possible with the current hand
- Determine the best card to discard based on the information stored in memory (Further explained below)
- Call Knock or Gin whenever possible (Reasoning explained under **Play**)
- Decide when to discard cards in a potential meld
  (Potential melds will be explained under **Interesting Parts**)

Explanation

The strategy was devised as such to prevents discarding cards which benefit the opponent and retrieve/keep cards only if it benefits the player. The aggressive nature of this strategy is what allows it to thrive as the decision making will be at its peak mid game where the information stored will allow more rewarding selections.

How does the memory work?

The memory represents an array which provides crucial information for the AI each round and is structured as followed:

[Cards that were discarded, Cards opponent rejects, Cards opponent accepted]

(e.g. "[[SA,S2,S3],[HA,H2],[C3,C4]]")

- Usage of discard pile tracking
  - Discarded cards can be viewed as cards which will never reappear in stock or top of discard. So, the only way to keep track of these cards is through storing in the memory. This is important as the 3rd memory relies on this storage to figure what card the opponent drew from the discard pile (if he chose discard)
- Usage of tracking cards which the opponent rejected
  - Cards discarded by the opponent are cards they do not require. The card in the top of the discard pile which were ignored by the opponent are also cards they do not require. The AI keeps track of these cards as they are the best cards to discard each turn since they will not benefit the opponent. Furthermore, cards of the same rank and adjacent cards with the same suit are also good choices for discard.

- Usage of tracking cards which the opponents accepted
  - If the card picks a card from the discard pile, these cards are what the opponent wants and can also be referred as cards known in the opponent's hand. So, it is highly likely that related cards will form meld/potential melds in the opponent's hand. Keeping related cards in your hand benefits greatly as it avoids the opponent from forming melds and also allows you form melds in the opponent's hand if they perform a knock.

Parsing

Parsing was a concept learn in week 11 and the AI's parsing process utilizes code which were done during Tutorial 11 (such as between). The main functions of the code transfer a string as a form of memory, but strings have their limitations which is why parsing is required to ensure these strings can be converted to relevant data. Several parsers were designed to produces the structure of the memory.

Parser combinators

The memory structure is essentially a list containing 3 list of cards with varying size. So, several parsers were required because each component within the string represents data such as a Card. In order to transform the input string, the sub-strings also require parsers as well which is why Parser combinators are essential for the memory string to be converted to the appropriated data structure.

Example:

"[[C3],[S2,DA],[]]" → [[Card Club Three], [Card Spade Two, Card Diamond Ace],[]]

These parsers include:

1. parseSuit
2. parseRank
3. parseCard (utilizes parser 1 and 2)
4. parseArray (utilizes parser 3 and betweenSepbyComma from tutorial 11)
5. parseMemory (utilizes parser 4 and betweenSepbyComma from tutorial11)

(parseResult will have its store data extracted via pattern matching)

Conversion to string

After the string was parsed, the memory that is being passed is still a string so the output of the parsed must be able to revert back to its string representation. There are several methods which does exactly and are capable of handling an array of cards for any given size.

Melds

There were several changes in code which evolves the computation involving melds. Initially the way the best meld combination was found was through brute force but was later changed as a better alternative which requires less computation was found.

Finding melds

There are only two forms of meld, sets and straights.

- Straights
  - Straights have a max length of 5 and minimum length of 3. A straight must have all cards be of the same suit and of sequential rank. The function matchMeld with the nextCard argument will check if a list of cards forms a valid straight.
  - matchMeld will perform a check on each card via nextCard which is a predicate that returns True if the current card and the next have the same suit and are of sequential ranks.
- Sets
  - Sets have a max length of 4 and minimum length of 3. A set must have all cards be of the same rank. The function matchMeld with the sameRank argument will check if a list of cards forms a valid set.
  - matchMeld will perform a check on each card via sameRank which is a predicate that returns True if the current card and the next have the same rank.

As explained above, the AI contain methods allowing it to determine when a list of cards is a valid meld. Furthermore, a function toMeld will be able to determine what meld a list of cards is with these components.

Best combination of melds

As stated in the beginning, the AI currently avoids performing any brute force approach as the time complexity is horrendous O(N!), especially given that the hand is of a fixed size 10 (occasionally 11 if any computation with the picked card in the start of each turn was included). Instead of approaching all possible combination of cards and melds.

From the code you will see a large amount code dedicated to finding the combination of melds, this approach may be larger in size but has a much better time complexity O(N). This is because of the properties of Gin Rummy where the hand size is always either 10 or 11.

Scenarios and functions

In short, there are only 5 scenarios which we should consider whenever we are finding the best combinations of melds:

a) Prioritize finding straights before sets
b) Prioritize finding sets before straights

c) Prioritize finding straights before sets
   (The first/second straight found must be <=4 in size)
d) Prioritize finding straights of size 3 before sets
   (The first/second straight found must be <=3 in size)
e) Prioritize finding sets of size 3 before straights
   (The first/second set found must be <=4 in size)

The functions which perform the above are findMelds which works on a list of cards left to right and finds for melds based on its corresponding scenario. It utilizes the code matchMeld and filter functions (functions which filter the remaining cards to retrieve all cards with the same rank or same suit as the current).

The bestMeld function retrieve the result of all findMelds functions and selects the best out of all (The one with the least number of deadwood).

This approach works because the max number of melds is only 3, the time complexity is O(N) as each computation for finding the best melds only requires computing the above which varies based on size. The approach works for any hand size of less than 12 but computes much more efficiently compared to the brute force approach as it requires searching all possible combinations.

Pick

The way the AI selects what cards to pick can be described as such:

- If the top card in the discard pile can form a meld in the current hand, draw from discard
- If the top card in the discard pile can form a potential meld in the current hand, draw from the discard.

Updating memory

It is important to point out that the pickCard function is also the one responsible for updating the memory as it is the only function that has access to all the required information relevant to the memory structure.

Potential melds

The next best alternative to a card that forms a meld is a card that forms potential meld, potential melds is essentially two cards which have the same rank or have the same suit and are of sequential rank. These cards have high potential to form meld as the game proceeds as only one other card will be required to complete a meld with these cards.

twoMeld is a function which determines if any cards have this sort of relations.

Play

The most important aspect of the game as it involves manipulating what options the opponent will have for drawing and decides when the game ends. This is also where the player's memory is fully utilized.

Gin, Knock or Drop

As stated in the summary of the strategy. The AI will always Gin or Knock whenever possible because this strategy will not benefit as the game drags longer so being the "First to strike" will always be better.

extractN and extractK are just used to search for the cards to discards.

How to check for Knock

To ensure a knock is possible, the total deadwood value should not exceed 10. This done via the checkKnock method and utilizes fromEnum to get the deadwood value of each individual card

(If fromEnum produces a value exceeding 10 it means the card one of the ranks Jack, Queen or King so the function ensures to set the value output to 10 when this occurs)

Discard

There was a sufficient amount of explanation on how the discard is selected based on the memory so this part will only emphasize on certain aspects. Only the deadwood cards will be under the option of cards for discarding. The function which selects the best card to drop is called bestDrop. The priority of what to cards to keep are as such:

- Cards which are part of a potential meld under certain conditions
- Cards which the opponent wants
- Cards which has no relation between the player and the opponent
- Cards which the opponent might
- Cards which the opponent completely rejects

A card that the opponent has discarded is highly likely to be a card that the opponent will never benefit them.

Interesting parts

Discarding cards from potential melds

The program initially used the discard pile to determine when a potential meld should be discarded but was changed to only discard from potential melds if all deadwood cards are part of a potential meld.

- Initial concept
  If all cards that completes the potential melds are all in the discard, then the potential melds will never be completed and render useless on the player's hand. So, the AI would had checked for such case and add the potential

melds as options for discard. However, this had a severe drawback as if the player fails to complete these potential melds the player has a higher likelihood to lose to Knocks. This is because the player only considers what's in the stock but not the opponent's hand so if the opponent holds the cards which completes the potential melds the Player may never discard these cards which will place the Player in a bad position.

- Current strategy
  The Player discards any potential melds if all deadwood cards are part of a potential meld. This is a better option as the AI's core strategy is to discard cards which the opponent never benefits from. So, this ensures the best card to discard from the potential melds will always be selected.

Tournament results

Overall, the AI has a very high win rate in the Tournament and has the potential to beat all bots as shown below:

Welcome wtah0001

Upload a new player here.

View upload status.

Your current results are:

Id	Rank	Elo
495	38	1319.8834

View on the ladder.

Download logs here.

You can click on the number of hands played to see the game.

Time	Score	Hands played
Nov 8 19:21:11	130	2
Nov 8 18:59:57	110	6
Nov 8 18:56:07	115	2
Nov 8 18:19:26	105	1
Nov 8 18:14:32	104	3
Nov 8 18:14:13	140	4
Nov 8 18:13:59	143	2
Nov 8 18:13:49	147	2
Nov 8 18:13:34	104	8
Nov 8 18:13:19	130	6

View latest: 10, 100, all.

Based on all the results from upload till the latest time (shown in the image above) the AI has a win rate of approximately 87%.