

INFO0004-2: Belote Referee

Sami BEN MARIEM, Cyril SOLDANI, Pr Laurent MATHY

April 20, 2020

1 Introduction

In this assignment, you will design and implement a referee for the game of Belote.

The referee that you must implement will receive as inputs the chosen trump suit, the team which is taking the contract, and the sequence of cards played during each trick.

After each trick, your program should display the current scores of both teams and the player who won the previous trick and is to play next.

Once all 8 tricks have been played, your program should display the final scores for both teams.

In addition to keeping track of the scores, the referee should also report any detected transgression of Belote's rules (see section 2).

2 Rules

Belote is a 32-cards, *trick-taking* game which is one of the most popular card games played in France. Four players form two teams of two partners which are never next to each others.

2.1 Dealing & bidding phase

Any Belote game starts with the dealing and negotiation phases where each player is given 8 cards and the contract is negotiated. A team will take the contract and the *trump suit* will be decided.

The organisation of this part of the game will not be further detailed since it is not part of the project. Indeed, the *trump suit*, cards played by the players as well as the team who is bidding will be given to you as input.

It is worth mentioning that the team taking the contract should have at least as many points as the opponents to win any points at all. Otherwise, the opposing team will get all the points.

2.2 Main phase

Once the dealing and bidding phase is over, the teams will compete for winning tricks.

A player starts each tricks by playing a card which fix the suit of the trick. Then, the players must respect the following rules:

1. If he can, a player must play a card of the suit of the trick.

2. If he cannot, there are two cases:
 - (a) **His partner is master** (*i.e.* is currently winning the trick) in which case he can play any card he wants.
 - (b) **His partner is not master** in which case he must play a trump card (*cut*) if possible.
3. If a player **has to** play a trump card, he must play a higher trump card than the ones already on the board if he can.

The first player of the game is the one to the left of the dealer, which we will call *player 1*. After the first trick, the first player of each trick is the one who won the previous trick.

The trick is won by the player who played the highest-rank trump card or, if no trump card was played, by the highest-rank card that respects the suit of the trick. The ranks and scores of each card are represented in table 1.

2.3 Scores

After each trick

Each won trick earns the team that wins it a number of points, obtained by summing the trick's card points as given by the following table.

Plain suit rank			A	10	K	Q	J	9	8	7
Score	20	14	11	10	4	3	2	0	0	0
Trump suit rank	J	9	A	10	K	Q			8	7

Table 1: Belote card ranks and scores.

Additionally, if one player had the king and queen of trumps and just played the second of those cards, he makes a **belote** and his team immediately wins 20 bonus points. The playing order of the king and queen is irrelevant, and the *belote* points are won even if the player lost one or both of those cards.

Finally, if it is the last trick, the team that wins it earns the **10 de der** bonus, worth 10 points.

Final game score

After having accounted for all 8 tricks, the scores of both teams should sum to 162 points (182 points if a *belote* occurred).

If a team won all the tricks, we say there is **capot**, and its score rises up to 252 (272 if that team also had a *belote*).

If the team taking the contract has less points than the opposing team, it fails to make its contract and is **inside** (“être dedans”). In this case, the inside team does not get any points (except for any *belote*) and the opposing team will receive all 162 points (or 252 points, if *capot*), plus any *belote*.

Note that to simplify matters, we don't apply the *hanging* rule when both teams have the same number of points. We consider the contract fulfilled in this case.

3 Interface

Your referee will be used through the following function:

```
bool game(std::istream& in, std::ostream& out, std::ostream& err);
```

where

in is an input stream on which game information and played cards will be written;

out is an output stream where scores and players who won the tricks must be printed;

err is an output stream where any error should be reported.

This function should return `true` when the game is over (*i.e.* all 8 tricks have been played, and final scores have been printed), and `false` as soon as a rule transgression has been detected.

This function declaration must be contained in a header file called `belote.hh`. This file may also contain other lines of code, *but only what is required* to support the use of this interface.

All the code required for the implementation of this interface must be in a file called `belote.cc`.

You can consult the Standard C++ Library Reference for further documentation about the STL data structures and algorithms, at <http://cppreference.com/>.

3.1 Input format

The input information will be given on the input stream as follows. The first line will contain the trump suit (one of `s`, `h`, `d`, `c` for spades, hearts, diamonds and clubs, respectively), followed by a space and then the team taking the contract (1 for the team of players 1 and 3, or 2 for the team of players 2 and 4).

Then, each trick will be input on a separate line, as a sequence of four cards (separated by spaces), where a card is composed of a *rank* (one of `A`, `K`, `Q`, `J`, `T`, `9`, `8`, `7`) immediately followed by a *suit* (as above). The order of the cards in the trick is the order of play, *not* the order of players. *E.g.* if player 3 won the last trick and is thus the first player to play, the trick

```
7d Ad Td 9s
```

means that player 3 plays the 7, player 4 the ace, player 1 the 10, and player 2 the 9. By definition, player 1 is the first player in the first trick.

To simplify this assignment, you can assume that the input is well-formed. You can also assume that no card will be played twice, and that only valid cards will be played.

3.2 Output format

After each trick (including the 8th one), your function will print a line with the scores of both teams followed by the player who has won the trick, separated by spaces. *E.g.* if at the end of a trick team 1 has 74 points, team 2 has 42 points, and player 2 won the trick, your program will print

```
74 42 2
```

At the end of the game, after the 8th trick, your program will additionally print a line with the final scores of both teams, separated by spaces, *e.g.*

```
162 20
```

3.3 Error format

When you detect a rule of Belote has been transgressed, you should report an error on the error stream, and immediately exit returning `false`.

The format of this error message is left to your appreciation. Try to make it as descriptive as possible, so that the players can easily understand which rule has been broken, and why.

3.4 Example game

To illustrate the different input/output formats, here is an example game. On the input stream, your program first reads

```
s 2
Qh 8h Ah 9h
```

The trump suit is thus *spades*, and team 2 (*i.e.* players 2 and 4) is the one making the contract. Player 1 plays a queen of hearts, player 2 a 8 of hearts, *etc.* The trick is won by player 3, which played the strongest card. Since the queen is worth 3 points, the ace 11 points, and the other two cards are worthless, the trick value is $11 + 3 = 14$. Your program should then output

```
14 0 3
```

The next trick now available on the input stream is

```
Kh Th 7h Qs
```

Since player 3 won the previous trick, he plays first and is thus the one playing the king of hearts. Player 2, who cuts with the queen of spades (the trump suit), wins the trick, which is worth $4 + 10 + 3 = 17$ points. Your program should then output

```
14 17 2
```

The next trick available on the input stream is the line

```
Jh 9s 8s Js
```

Since player 2 won the previous trick, he is the one playing the jack of hearts. However, in trick 2, he played the queen of spades when the played suit was hearts. He should thus have no more hearts cards. The rule that says you have to play the suit of the first played card if you can has been transgressed. Your program should report an error on the error stream, *e.g.*

```
Error: player 2 played a jack of hearts, but he should not have any hearts left
       since he played a queen of spades over hearts in trick 2.
```

Since an error was detected, your program should exit with `false` without further processing.

4 Remarks

4.1 No new classes or structures

In this project, you **must not** define any new `class` or `struct`. You should rely only on existing C++ and STL-defined data types, although you can declare aliases with `typedefs`.

4.2 Respect the interface

Submission must scrupulously **follow the interface** defined above. You can of course define auxiliary functions as you see fit, but these should be properly hidden from the users of the interface.

4.3 Readability

Your code must be easy to read and understand:

- Make the organisation of your code as obvious as possible. Remember you can create as many auxiliary functions as you see fit.
- Use descriptive names for functions and global variables (if you *need* some).
- Complement your self-documenting code with comments, where appropriate.
- Choose a coding convention, and stick to it. *Consistency* is key.

4.4 Robustness

Your code must be robust. The `const` keyword must be used where appropriate, memory must be managed appropriately, the program must run to completion **without crash**.

However, to simplify this assignment, you can assume that the input format will be respected, and that only valid Belote cards will be played, and only once.

4.5 Warnings

Your code **must** compile without error or warning with `g++ -std=c++14 -Wall` on `ms8??` machines. However, we advise you to check your code also with `g++ -std=c++14 -Wall -Wextra` or, even better, `clang++ -Weverything -Wno-c++98-compat`.

4.6 Error messages

Try to make your error messages as descriptive as possible to help players quickly identify which rule was broken, and why.

4.7 Evaluation

Your code will be evaluated based on all the above criteria. Failure to comply with any of the points mentioned in **bold face** can (and most often will) result in a zero mark!

5 Submission

Projects should be submitted through the submission platform before **Thursday March the 19th, 23:59 CET**. Any late submission will receive a penalty of $2^n - 1$ points (over 20), where n is the number of started days after the deadline.

You will submit a `s<ID>.tar.xz` archive, containing a `s<ID>` folder containing `belote.{cc,hh}`, where `s<ID>` is your ULiège student ID.

The submission platform will do basic checks on your submission, and you can submit multiple times, so check your submission early!