Frequently Asked Questions

Where will I find answers to the frequently asked questions?
Frequently asked questions will be answered on this page.

SPOTS

What does it mean when SPOTS says....

Assertion failed: `card_n >= 0 && card_n < self.hand_card_count()`
This means that you're asking for a card at index `n`, where `n` is either less than 0, or (more likely) greater than the number of cards in the player's hand.

e.g. if the player has three cards:

`[RED 1 HEARTS] [RED 5 HEARTS] [RED 3 HEARTS]`, these are at index 0, 1, 2 respectively.

The `handCardCount` function will return the number of cards in the player's hand: 3 (since they have 3 cards).

But, the cards are at index 0, 1, 2, so the index passed into the `handCard` function should be strictly less than what the `handCardCount` function returns.

Assertion failed: `turn >= 0 && turn < self.num_turns()`
This is effectively the same as the above example: you're asking for information about a turn, but the turn number you're passing in isn't valid.

When calling the functions that take in a turn number or a move number, those numbers need to be strictly less than the output from the `numTurns` / `turnMoves` functions.

e.g. if we're on turn 0, numTurns will say 1; so if you're calling `turnMoves` or other functions that take in the turn to look at, it has to be looking at `numTurns-1`.

Since turns count from 0, when we have 1 turn, that's turn 0, when we have 2 turns, that's turn 0 and 1, etc.

You shouldn't call turnMoves on anything higher than `currentTurn` (or `numTurns-1`). You shouldn't call `pastMove` on anything higher than `turnMoves-1` – if the number of moves that has happened on a turn is 1, you can view turn 0, since that's the only turn that's happened.

Implementation

Can/should we use linked lists?
Yes.

Can we just use a really big array of Cards to store the deck, discard pile, or player hands?
No. There is no limit on how big a deck can be, or how many cards a player can have in their hand, beyond the memory limits of the computer running the code.

Can we make assumptions about what order the cards in a player's hand are?
No. For a full explanation, see here.

Does our ADT have to handle multiple rounds?
No. For more details, see here.

Valid Moves

Do you have a nice diagram showing what moves are valid at which point in the game?
Yes! Click this link

I have another question about valid/invalid moves
Click the link above.

Game End

Does the game end as soon as somebody plays their final card, or do they also need to call `END_TURN`?
The game ends as soon as they put their final card down; they don't have to call `END_TURN` (and can't, as the game has now ended).

When can the game end with NO_WINNER?
The game ends with no winner when a player attempts to draw a card, but there is no way to draw a card – i.e. there are no cards left in the deck, and the discard pile only has one card in it.

What are the others ways in which the game can end?
These are answered in the Game Rules.

The Deck

What size can the deck be?
It's valid for the deck to be any size larger than the minimum deck size (which is 30: 7 cards * 4 players = 28 cards, + 1 card flipped to make the discard pile, + 1 card in the deck so that a player can draw).

However, the "full deck" consists of two copies of each card, and since there are 2 of each card * 5 suits * 5 colors * 16 values = 800 cards. In tournament games we will (probably) be using this "full deck" (unless it turns out there's some reason we need to use more or fewer cards).

But in terms of writing tests, and later implementing the ADT, it's valid for the deck to be any size >= 30. (It's also valid for the deck to be > 800).

Do we need to shuffle the deck?
No, you should deal the cards in the order they were given to you when passed in to the newGame function.

What happens if somebody attempts to draw a card but there are no cards left in the deck?
The game ends, and nobody is at fault. See the main assignment page, under "A game can end in one of the following ways:"

What happens when the deck is empty?
When the deck is empty, the discard pile is "flipped over", and becomes the deck, **except** for the top card on the discard pile which stays there.

The last card to be played into the discard pile must always be the card that is on top of the discard pile. This means that when the discard pile is used to form the draw pile, the card on top remains in the discard pile.

What order are the cards in after flipping the discard pile over?
The card that was on the bottom of the discard pile (i.e. the first card that was discarded) is now the top card on the deck.

The deck / discard pile aren't shuffled at any point.

Valid / Invalid Input

Does the ADT have to deal with invalid input?
What should the ADT return if the input is invalid?
You cannot test invalid input, i.e. asking for the move from turn 100 when there haven't been 100 turns in the game thus far.

The ADT implementation can assume that the input is valid, and doesn't have to check for invalid input.

If a test gives an ADT function invalid input, that test is considered incorrect.

Saying UNO/DUO/TRIO

When should a player say UNO/DUO/TRIO?
Immediately after a player discards a card, if they now have only one card left in their hand, they must say UNO. Similarly, if the player now only has 2 cards left, they must say DUO, or for only 3 cards left, they must say TRIO.

Note: This only applies after you discard a card – if you have 1 card in your hand, and the player before you plays a `DRAW_TWO` and you draw two cards, you *don't* now have to say TRIO, even though you now have three cards in your hand.

How can you call somebody out for not saying UNO/DUO/TRIO?
If the player before you didn't say UNO/DUO/TRIO when they were meant to, you can "call them out" on it – i.e. say that they made a mistake, and that they should be penalised.

To do this, your very first move of your turn must be SAY_UNO / SAY_DUO / SAY_TRIO, before you make any other moves.

What happens if you say UNO/DUO/TRIO when you shouldn't be saying it?
If you say UNO/DUO/TRIO at the start of your turn, and the other person was correct (i.e. either they did say UNO/DUO/TRIO, or they didn't need to), you must draw 4 cards as a penalty.

> **This must be handled by the ADT**, i.e. as soon as the player says UNO/DUO/TRIO invalidly, the ADT must give that player the top 4 cards from the deck into their hand.

It is otherwise an invalid move to say UNO/DUO/TRIO, unless you have just played a card and now have 1/2/3 cards left in your hand, or you are calling somebody out for not saying UNO/DUO/TRIO.

What happens if you call somebody out for not saying UNO/DUO/TRIO?
If you successfully call somebody out for not saying UNO/DUO/TRIO, they must draw two cards as a penalty.

> **This must be handled by the ADT**, i.e. as soon as you call the player out, the ADT must give that player the top 2 cards from the deck into their hand.

How does saying UNO/DUO/TRIO interact with a `CONTINUE` ?
If a player uses a continue, it is possible that they have to say UNO, DUO, *and* TRIO on their turn, and if they fail any one of these the next player may call them out on it.

If you play a card, and then immediately after playing that card you now have (3/2/1) cards in your hand, you **must** say (TRIO/DUO/UNO). If you played e.g. a `CONTINUE` you can now play another card, so if you now have (3/2/1) cards in your hand you must now say (TRIO/DUO/UNO) etc.

An example:

Move 0: play a `CONTINUE` , you now have 3 cards left in your hand.

Move 1: play `SAY_TRIO` (if you don't do this as your move immediately after the previous move, you can be called out for it).

Move 2: play another card, you now have 2 cards left in your hand.

Move 3: play `SAY_DUO`.

Move 4: play `END_TURN`.

Special Cards

What happens if the first card in the discard pile at the start of the game is a special card?
The card will have the effect that it usually had:

- If a `DRAW_TWO` is the initial discard pile card, the first player must draw two cards (or play a `DRAW_TWO` of their own).
- If the first card is `ADVANCE`, the first player is skipped and it becomes the second player's turn.
- If the first card is `BACKWARDS`, play starts in the opposite direction (i.e. the first player starts, but then player 3 follows, rather than player 1)
- If the first card is `CONTINUE`, or `DECLARE`, nothing special happens – the cards are treated like any other card (you can match on suit, color, or value).

ZERO

How does a ZERO card work?
The ZERO card is like a "wild card" in other card games – whenever it's valid to play a card, the ZERO is a valid card to play.

For example, if the top card on the deck was a BLUE 3 of HEARTS, a ZERO of *any* color or suit is a valid match (rather than needing to be a BLUE card or a HEARTS card).

If it's not currently valid to play a card, then it's not valid to play a ZERO, e.g. if a DRAW_TWO has been played and you still need to draw more cards, you can't play a ZERO.

DRAW_TWO

With a `DRAW_TWO` card, what does it mean that cards can be stacked on top of each other?
If a player plays a `DRAW_TWO` card, the next player must either draw two cards and end their turn, unless they also have a `DRAW_TWO` card in their hand.

If they have a `DRAW_TWO` card in their hand, they can choose to play it, and end their turn. The following player would then have to draw four cards, or play a `DRAW_TWO` themselves, forcing the next player to draw six cards, etc.

> *Our four players are Alice, Bob, Jack, and Mallory.*
>
> *Let's say the discard pile starts with a 4, RED, HEARTS card.*
>
> *Alice starts. She can play any card that either has the value 4, the color RED, or the suit HEARTS. Her hand will have 7 cards, let's say one of them is a BLUE 6 of HEARTS. She plays this card (it matches because they're both hearts).*
>
> *Now it's Bob's turn. He has to play a card that matches BLUE, 6, or HEARTS. Let's say he plays a BLUE 2 of DIAMONDS.*
>
> *Now it's Jack's turn. Because Bob played a 2 (or a `DRAW_TWO`), Jack has to either draw 2 cards, or play another `DRAW_TWO` card. Let's say that Jack plays a GREEN 2 of QUESTIONS.*
>
> *It's now Mallory's turn. Because two `DRAW_TWO` cards have been played in succession, she now has to either draw 4 cards, or play another `DRAW_TWO`. If she had a `DRAW_TWO` to play, Alice would then have to either play another `DRAW_TWO`, or draw six cards.*
>
> *But to keep the example short, let's say Mallory doesn't have any cards with value 2. She now has to draw 4 cards: 2 for Bob's `DRAW_TWO` he played, and 2 for Jack's `DRAW_TWO` he stacked on top of it.*

CONTINUE

After a player plays a `CONTINUE`, what can their additional card be?
The second card that they play must be a valid card, i.e. it must have the same value, suit, or color as the `CONTINUE`.

DECLARE

What cards are valid to play after a `DECLARE`?
A card with the same value or suit but not the same color can be played. It's treated as if the `DECLARE` that was played is any other card, except that its effective color (in terms of what move is legal next time) becomes the color that was declared, not the color on the card.

Does a ZERO card of any color bypass this rule?
Yes.

Specific ADT Functions

numCards

Will numCards always give 800?
And thus, numOfSuit = 800/5 = 160?
numOfColor = 800/5 = 160?

numOfValue = 800/16 = 50?
No, you can't assume fixed values. numCards won't always be 800. It will be however many cards are in the deck that's provided to the game. So the numbers specified above (numCards giving 800, numOfColor giving 160, etc) won't be right.

newGame

What is deckSize in the newGame function?
In the `newGame` function:

```
Game newGame(int deckSize, value values[], color colors[], suit suits[]);
```

`deckSize` is the size of the deck, i.e. the number of cards (and is what numCards should return).

What input does the newGame function take?
When you call `newGame`, you give it 3 arrays, each that are `deckSize` big. The deck must contain at least 30 cards.

So if `deckSize` was 50, you'd give it an array of 50 values, another array of 50 colors, and another array of 50 suits.

The order that these values are in each array gives the cards in the deck, i.e. the first card in the deck has the value `values[0]`, the color `colors[0]`, and the suit `suits[0]`.

In the newGame function, can we assume that the arrays of values, colors and suits we are passed has already been 'shuffled'?
Yes. The order that the cards are passed in is the order that they should be in the deck, you don't need to (and shouldn't) change the ordering.

Is index 0 of the values/colors/suits arrays on the top or bottom of the deck?
Index 0 is the first card on the deck (so it will be dealt immediately to the first player).

If the card values you were passed in were like {RED, BLUE, GREEN, YELLOW, PURPLE} {HEARTS, DIAMONDS, HEARTS, DIAMONDS, HEARTS} {3, 4, 5, 6, 7}, the dealing would go like this:

```
Player 1 gets a RED 3 of HEARTS

Player 2 gets a BLUE 4 of DIAMONDS

Player 3 gets a GREEN 5 of HEARTS

Player 4 gets a YELLOW 6 of DIAMONDS

Player 1 gets a PURPLE 7 of HEARTS

(etc for the rest of the cards)
```

playMove

What values should the unused fields of playerMove have?
If the nextColor field and/or card members of playerMove are not used (because, for example, the action is `DRAW_CARD`), no value should be assigned to them.

playerPoints

Do the points have to be calculated at any point during the game, or just at the end?
`playerPoints` could be called at any point during the game, and must return the sum of the values of cards in each player's hand.

Definitions

round
one full playthrough of the game, from when the game first begins (the cards are dealt, etc) through to when somebody puts their final card down (or the game ends in another way, i.e. through an invalid move or running out of cards)

turn
all of the actions that one player does, starting immediately after the player before them has finished making moves, and ending immediately before the player after them makes any moves. e.g. if Alice, Bob, Jack, and Mallory are playing, Bob's "turn" would come after Alice has finished all of her moves, and Bob's "turn" would end once he's finished playing all of his moves, at which point it would become Jack's "turn".

move
one individual action played by a player, during their turn.
The possible moves are defined in the playerMove struct, and are currently (as of 2017-10-02):

- DRAW_CARD
  - Draw a single card from the deck.
- PLAY_CARD
  - Play a single card onto the discard pile.
- SAY_UNO

    - Say the word “UNO”.
- SAY_DUO
    - Say the word “DUO”.
- SAY_TRIO,
    - Say the word “TRIO”.
- END_TURN
    - End the player’s turn.

  It’s often valid to make several moves, such as drawing multiple cards (e.g. if the player before you played a `DRAW_TWO`), or playing multiple cards (e.g. if the first card you play is a `CONTINUE`).

discard
play
place a card from your hand onto the discard pile. (note that the two terms are equivalent)

ChangeLog

- **v0.0.4** (2017-10-25): Added information about common SPOTS errors; ZERO card.
- **v0.0.3** (2017-10-21): Added information about the game ending, and what happens when the deck is empty.
- **v0.0.2** (2017-10-13): Added information about valid moves, interaction between various special cards.
- **v0.0.1** (2017-10-02): Added information about deck size, valid/invalid input, saying uno/duo/trio, `DRAW_TWO`, `CONTINUE`, numCards, newGame; definitions of `round`, `turn`, `move`, `play`, `discard`.
- **v0.0.0** (2017-09-18): No questions answered yet