Cards

Description

Class designed to ease the creation and implementation of traditional card games. The card pool is organized into an array of arrays called fields. By default a 52 element array of cards is created and added to fields[0].

A field is any distinct pool of cards in a game, for example Poker may use fields[0] as the deck but add fields[1] as the player's hand but a game with no deck, like War, may use fields[0] for whatever they so choose.

Each card is represented by a dictionary with 3 elements: SUIT, VALUE and FACE-UP:

int SUIT: Represented as an int between 0 and 3 inclusive (0: Spades, 1: Clubs, 2: Diamonds, 3: Hearts)

int VALUE: valid inputs range between 1 and 13, number cards correspond directly, 1 = Ace, 11 – 13 are Jack, Queen and King respectively

This class is meant to be game-agnostic and thereby, does not provide any game-specific functions such as determining a context-appropriate value for an Ace in Blackjack. The goal is simply to create cards and track their corresponding field, however it may be used

Class Variables

const int DECKSIZE	Standard size of a deck according to the class. Default is 52
Array fields []	Array of Arrays, each element is an Array of

	dictionaries
Array deck []	Initial array of dictionaries (cards) assigned to fields[0] at _init()
int deckIndex	Index of whichever field is considered the deck. Altering will change the behavior of cardDraw(). Default is 0

Member Fucntions

void	_init()
void	addField(int n = 1)
void	addDeck(Array currField, int n = 1)
Array	getField(int pos)
void	flipCard(card card)
void	setRevealed(card card, bool faceup)
void	createCard(Array field, int suit, int value)
void	destroyCard(Array field, int index)
void	drawCard(Array handField)
void	passCard(card card, Array currField, Array newField)
card	getCard(int field, int index)
int/char	getSuit(card currCard, bool mask = true)
int/char	getValue(card currCard, bool mask = true)

String	getDescStr(card currCard, maskV = true, maskS = true)
int	getPoolSize()
int	getFieldCount()
int	getFieldSize(int pos = 0)
bool	fieldExists(int pos)
void	shuffleField(int pos = 0)
void	shuffleAll()
void	sortField(int pos)
void	sortAll()
void	printSummary()

Member Function Description

_init()

Creates a 52 element Array of card dictionaries and appends it to the 0 index of the fields Array

void addField(int n = 1)

Adds *n* new Arrays to the *fields* Array

void addDeck(Array currField, int n = 1)

Adds *n* new 52-card decks to the *currField* Array

Array getField(int pos)

Returns the Array at fields[*pos*]

void flipCard(card card)

Toggles *card's* FACE-UP value

void setRevealed(card card, bool faceup)

Manually sets *card's* FACE-UP value to specified value

void createCard(Array field, int suit, int value)

Creates a card using the passed *suit* and *value* parameters and adds it to the specified *field's* Array

void destroyCard(Array field, card index)

Removes card at *index* in the passed *field* Array from the game completely

void drawCard(Array handField)

Moves the front card from fields[deck_index] (default: 0) to passed *handField* Array

void passCard(card card, Array currField, Array newField)

Moves *card* from *currField* to *newField*

card getCard(int field, int index)

Returns card from specified *index* in fields[*field*]. TODO: Make this work with Array field rather than int field

int/char getSuit(card currCard, mask = true)

Returns int associated with *currCard.SUIT* or the corresponding char if *mask* is set to true

int/char getValue(card currCard, mask = true)

Returns int associated with *currCard.VALUE* or the corresponding char if *mask* is set to true and value is a face card

String getDescStr(card currCard, maskV = true, maskS = true)

Concatenate getValue and getSuit into a string and return the result

int getPoolSize()

Return the total number of cards in the game among all fields

int getFieldCount()

Return the amount of elements in fields

int getFieldSize(int pos = 0)

Return the size of the field in fields[pos]

bool fieldExists(int pos = 0)

Returns true if a field exists at index *pos* in fields

void shuffleField(int pos = 0)

Randomly rearranges the cards located in fields[*pos*]

void shuffleAll()

Shuffles each index in fields

void sortField(int pos)

Sorts the cards in fields[*pos*] based on their value (Ace - King)

void sortAll()

Sorts each index in fields

void printSummary()

Prints a summary of the game state including, how many fields there are and how many cards each field contains. Example:

```
There are 2 fields

Field 0 has 50 elements

Field 1 has 2 elements
```

static void cardSort(a, b)

Static function to make sortField(int) and sortAll() work properly. Held in a separate class. TODO: Make functioning cardSortSuit(a,b)