CruceGame

v0.2.0

Generated by Doxygen 1.8.6

Sun Mar 16 2014 00:16:42

Contents

1 Class Index **1**

1.1 Class List 1

2 File Index **3**

2.1 File List 3

3 Class Documentation **5**

3.1 Card Struct Reference 5

3.1.1 Detailed Description 5

3.1.2 Member Data Documentation 5

3.1.2.1 suit 5

3.1.2.2 value 5

3.2 Deck Struct Reference 5

3.2.1 Detailed Description 6

3.2.2 Member Data Documentation 6

3.2.2.1 cards 6

3.3 Game Struct Reference 6

3.3.1 Detailed Description 6

3.3.2 Member Data Documentation 6

3.3.2.1 deck 6

3.3.2.2 numberPlayers 6

3.3.2.3 players 7

3.3.2.4 pointsNumber 7

3.3.2.5 round 7

3.3.2.6 teams 7

3.4 Hand Struct Reference 7

3.4.1 Detailed Description 7

3.4.2 Member Data Documentation 7

3.4.2.1 cards 7

3.4.2.2 players 7

3.5 Player Struct Reference 8

3.5.1 Detailed Description 8

3.5.2 Member Data Documentation . . . 8
3.5.2.1 hand . . . 8
3.5.2.2 id . . . 8
3.5.2.3 isHuman . . . 8
3.5.2.4 name . . . 8
3.5.2.5 score . . . 8
3.6 Round Struct Reference . . . 8
3.6.1 Detailed Description . . . 9
3.6.2 Member Data Documentation . . . 9
3.6.2.1 bids . . . 9
3.6.2.2 hands . . . 9
3.6.2.3 id . . . 9
3.6.2.4 players . . . 9
3.6.2.5 pointsNumber . . . 9
3.6.2.6 trump . . . 9
3.7 Team Struct Reference . . . 9
3.7.1 Detailed Description . . . 10
3.7.2 Member Data Documentation . . . 10
3.7.2.1 id . . . 10
3.7.2.2 name . . . 10
3.7.2.3 players . . . 10

4 File Documentation **11**

4.1 src/libCruceGame/deck.h File Reference . . . 11
4.1.1 Detailed Description . . . 11
4.1.2 Function Documentation . . . 12
4.1.2.1 deck_cardsNumber . . . 12
4.1.2.2 deck_compareCards . . . 13
4.1.2.3 deck_createCard . . . 13
4.1.2.4 deck_createDeck . . . 13
4.1.2.5 deck_deckShuffle . . . 13
4.1.2.6 deck_deleteCard . . . 14
4.1.2.7 deck_deleteDeck . . . 14
4.2 src/libCruceGame/game.h File Reference . . . 14
4.2.1 Detailed Description . . . 15
4.2.2 Function Documentation . . . 15
4.2.2.1 game_addPlayer . . . 15
4.2.2.2 game_addTeam . . . 15
4.2.2.3 game_checkCard . . . 15
4.2.2.4 game_createGame . . . 15

4.2.2.5 game_deleteGame . . . 16
4.2.2.6 game_removePlayer . . . 16
4.2.2.7 game_removeTeam . . . 16
4.2.2.8 game_winningTeam . . . 16
4.3 src/libCruceGame/platform.h File Reference . . . 17
4.3.1 Detailed Description . . . 17
4.4 src/libCruceGame/round.h File Reference . . . 17
4.4.1 Detailed Description . . . 18
4.4.2 Function Documentation . . . 18
4.4.2.1 round_addPlayer . . . 18
4.4.2.2 round_addPlayerHand . . . 18
4.4.2.3 round_arrangePlayersHand . . . 18
4.4.2.4 round_computeScore . . . 19
4.4.2.5 round_createHand . . . 19
4.4.2.6 round_createRound . . . 19
4.4.2.7 round_deleteHand . . . 19
4.4.2.8 round_deleteRound . . . 19
4.4.2.9 round_distributeCard . . . 20
4.4.2.10 round_distributeDeck . . . 20
4.4.2.11 round_findPlayerIndexRound . . . 20
4.4.2.12 round_getBidWinner . . . 20
4.4.2.13 round_handWinner . . . 21
4.4.2.14 round_placeBid . . . 21
4.4.2.15 round_putCard . . . 21
4.4.2.16 round_removePlayer . . . 21
4.4.2.17 round_removePlayerHand . . . 22
4.5 src/libCruceGame/team.h File Reference . . . 22
4.5.1 Detailed Description . . . 23
4.5.2 Function Documentation . . . 23
4.5.2.1 team_addCard . . . 23
4.5.2.2 team_addPlayer . . . 23
4.5.2.3 team_computeScore . . . 23
4.5.2.4 team_createPlayer . . . 23
4.5.2.5 team_createTeam . . . 24
4.5.2.6 team_deletePlayer . . . 24
4.5.2.7 team_deleteTeam . . . 24
4.5.2.8 team_hasCards . . . 24
4.5.2.9 team_removePlayer . . . 25

Index **27**

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Card
: Card structure, to keep suit and value 5

Deck
: A 28 card deck used in this game 5

Game
: Game structure 6

Hand
: Hand structure 7

Player
: Player structure 8

Round
: Round structure 8

Team
: Team structure 9

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

src/libCruceGame/**constants.h** . . . . . **??**
src/libCruceGame/**cruceGame.h** . . . . . **??**
src/libCruceGame/deck.h
    Card and Deck structures, as well as helper functions 11
src/libCruceGame/**errors.h** . . . . . **??**
src/libCruceGame/game.h
    Game structures, as well as helper functions 14
src/libCruceGame/platform.h
    Contains platform specific definitions 17
src/libCruceGame/round.h
    Round and Hand structures, as well as helper functions 17
src/libCruceGame/team.h
    Player and Team structures, with helper functions 22

Chapter 3

Class Documentation

3.1 Card Struct Reference

Card structure, to keep suit and value.

`#include <deck.h>`

Public Attributes

- enum Suit suit
- int value

3.1.1 Detailed Description

Card structure, to keep suit and value.

Note that value is the game value (i.e. the points), not the face value.

3.1.2 Member Data Documentation

3.1.2.1 Card::suit

The suit of the card.

3.1.2.2 Card::value

The value of the card.

The documentation for this struct was generated from the following file:

- src/libCruceGame/deck.h

3.2 Deck Struct Reference

A 28 card deck used in this game.

`#include <deck.h>`

Public Attributes

- struct Card * cards [DECK_SIZE]

3.2.1 Detailed Description

A 28 card deck used in this game.

Before using a Deck, please use deckInit function to assign the cards.

3.2.2 Member Data Documentation

3.2.2.1 Deck::cards

Pointer to the cards of the deck.

The documentation for this struct was generated from the following file:

- src/libCruceGame/deck.h

3.3 Game Struct Reference

Game structure.

`#include <game.h>`

Public Attributes

- int numberPlayers
- int pointsNumber
- struct Round * round
- struct Player * players [MAX_GAME_PLAYERS]
- struct Team * teams [MAX_GAME_TEAMS]
- struct Deck * deck

3.3.1 Detailed Description

Game structure.

Structure used to keep information about the game data.

3.3.2 Member Data Documentation

3.3.2.1 Game::deck

Pointer to the deck of the game.

3.3.2.2 Game::numberPlayers

The number of the players that joined the game.

3.3.2.3 Game::players

Pointer to the players of the game.

3.3.2.4 Game::pointsNumber

The total amount of the points of the game.

3.3.2.5 Game::round

Pointer to the current round of the game.

3.3.2.6 Game::teams

Pointer to the teams of the game.

The documentation for this struct was generated from the following file:

- src/libCruceGame/game.h

3.4 Hand Struct Reference

Hand structure.

`#include <round.h>`

Public Attributes

- struct Card * cards [MAX_GAME_PLAYERS]
- struct Player * players [MAX_GAME_PLAYERS]

3.4.1 Detailed Description

Hand structure.

In a hand, player[i] gives cards[i] and bids bid[i]. The players should be added in the order of the bids.

3.4.2 Member Data Documentation

3.4.2.1 Hand::cards

Pointer to the cards of the hand.

3.4.2.2 Hand::players

Pointer to the players of the hand.

The documentation for this struct was generated from the following file:

- src/libCruceGame/round.h

3.5 Player Struct Reference

Player structure.

`#include <team.h>`

Public Attributes

- int id
- char * name
- struct Card * hand [MAX_CARDS]
- int score
- int isHuman

3.5.1 Detailed Description

Player structure.

Structure to keep relevant informations about the players.

3.5.2 Member Data Documentation

3.5.2.1 Player::hand

Pointer to the cards of the player.

3.5.2.2 Player::id

Identifier of the player.

3.5.2.3 Player::isHuman

Flag used to indicate if the player is human or robot.

3.5.2.4 Player::name

Pointer to the name of the player.

3.5.2.5 Player::score

The amount of points earned in a hand.

The documentation for this struct was generated from the following file:

- src/libCruceGame/team.h

3.6 Round Struct Reference

Round structure.

`#include <round.h>`

Public Attributes

- int id
- enum Suit trump
- struct Hand * hands [MAX_HANDS]
- int bids [MAX_GAME_PLAYERS]
- struct Player * players [MAX_GAME_PLAYERS]
- int pointsNumber [MAX_GAME_PLAYERS]

3.6.1 Detailed Description

Round structure.

Round is a division of the game, it keeps the winning hands and computes the score until the winner of the round is found.

3.6.2 Member Data Documentation

3.6.2.1 Round::bids

The bids of the players.

3.6.2.2 Round::hands

Pointer to the hands of the round.

3.6.2.3 Round::id

Identifier of the round.

3.6.2.4 Round::players

Pointer to the players of the round.

3.6.2.5 Round::pointsNumber

The total amount of points of the round.

3.6.2.6 Round::trump

The trump of the round.

The documentation for this struct was generated from the following file:

- src/libCruceGame/round.h

3.7 Team Struct Reference

Team structure.

`#include <team.h>`

Public Attributes

- int id
- char * name
- struct Player * players [MAX_TEAM_PLAYERS]

3.7.1 Detailed Description

Team structure.

Players are grouped in teams. One team for 2-3 players, and two teams for 4 players.

3.7.2 Member Data Documentation

3.7.2.1 Team::id

The identifier of the team.

3.7.2.2 Team::name

Pointer to the name of the team.

3.7.2.3 Team::players

Pointer to the players of the team.

The documentation for this struct was generated from the following file:

- src/libCruceGame/team.h

Chapter 4

File Documentation

4.1 src/libCruceGame/deck.h File Reference

Card and Deck structures, as well as helper functions.

```
#include "platform.h"
#include "constants.h"
```

Classes

- struct Card

  Card structure, to keep suit and value.
- struct Deck

  A 28 card deck used in this game.

Functions

- EXPORT struct Card * deck_createCard (enum Suit suit, int value)

  Allocates and initializes a card.
- EXPORT int deck_deleteCard (struct Card **card)

  Frees the memory of a card and makes the pointer NULL.
- EXPORT struct Deck * deck_createDeck ()

  Allocates and initializes a deck.
- EXPORT int deck_deckShuffle (struct Deck *deck)

  Shuffles a deck.
- EXPORT int deck_deleteDeck (struct Deck **deck)

  Frees the memory of a deck and sets the pointer to NULL.
- EXPORT int deck_compareCards (const struct Card *card1, const struct Card *card2, enum Suit trump)

  Compare two cards.
- int deck_cardsNumber (struct Deck *deck)

  The function counts the cards from deck.

4.1.1 Detailed Description

Card and Deck structures, as well as helper functions.

4.1.2 Function Documentation

4.1.2.1 int deck_cardsNumber (struct Deck * *deck*)

The function counts the cards from deck.

Parameters

deck	Pointer to the deck from which it counts.

Returns

The cards number from deck.

4.1.2.2 EXPORT int deck_compareCards (const struct Card * *card1,* const struct Card * *card2,* enum Suit *trump*)

Compare two cards.

Parameters

card1	The first card.
card2	The second card.
trump	The trump of the round.

Returns

0 If the cards are equal. 1 If the first card is winning. 2 If the second card is winning. Error code otherwise.

4.1.2.3 EXPORT struct Card* deck_createCard (enum Suit *suit,* int *value*)

Allocates and initializes a card.

Parameters

suit	The suit of the new card.
value	The value of the new card.

Returns

Pointer to the new card on success or NULL on failure.

4.1.2.4 EXPORT struct Deck* deck_createDeck ( )

Allocates and initializes a deck.

Returns

Pointer to the new deck on success or NULL on failure.

This function initializes a deck by iterating over all values and suits available. The deck will be always the same.

4.1.2.5 EXPORT int deck_deckShuffle (struct Deck * *deck*)

Shuffles a deck.

Parameters

deck	The deck to be shuffled.

Returns

NO_ERROR on success, error code otherwise.

This function randomly shuffles the deck. It uses rand function from stdlib with time as seed. The shuffle is performed by random swaps. The number of swaps is also random, but it is at least SWAP_MIN and smaller then SWAP_MAX.

4.1.2.6 EXPORT int deck_deleteCard (struct Card ** *card*)

Frees the memory of a card and makes the pointer NULL.

Parameters

card	Pointer to the pointer to be freed.

Returns

NO_ERROR on success, error code otherwise.

4.1.2.7 EXPORT int deck_deleteDeck (struct Deck ** *deck*)

Frees the memory of a deck and sets the pointer to NULL.

Parameters

deck	Pointer to the pointer to be freed.

Returns

NO_ERROR on success, error code otherwise.

4.2 src/libCruceGame/game.h File Reference

Game structures, as well as helper functions.

```
#include "platform.h"
#include "constants.h"
#include "team.h"
#include "deck.h"
#include "round.h"
```

Classes

- struct Game

  Game structure.

Functions

- EXPORT struct Game * game_createGame (int numberPoints)

  Allocates memory for and initializes a game.
- EXPORT int game_deleteGame (struct Game **game)

  Frees the memory of a game and makes the pointer NULL.
- EXPORT int game_addPlayer (struct Player *player, struct Game *game)

  Adds a player to a game.
- EXPORT int game_removePlayer (struct Player *player, struct Game *game)

  Removes a player from a game.
- EXPORT int game_addTeam (struct Team *team, struct Game *game)

  Adds a team to a game.
- EXPORT int game_removeTeam (struct Team *team, struct Game *game)

  Removes a team from a game.

- EXPORT struct Team * game_winningTeam (struct Game *game)
  Searches the winning team of a game.
- EXPORT int game_checkCard (struct Player *player, struct Game *game, struct Hand *hand, int idCard)
  Function checks if the player can put a card down.

4.2.1 Detailed Description

Game structures, as well as helper functions.

4.2.2 Function Documentation

4.2.2.1 EXPORT int game_addPlayer (struct Player * *player,* struct Game * *game*)

Adds a player to a game.

Parameters

player	The player to be added.
game	The game where the player is to be added.

Returns

NO_ERROR on success, error code otherwise.

4.2.2.2 EXPORT int game_addTeam (struct Team * *team,* struct Game * *game*)

Adds a team to a game.

Parameters

team	The team to be added.
game	The game where the team is to be added to.

Returns

NO_ERROR on success, error code otherwise.

4.2.2.3 EXPORT int game_checkCard (struct Player * *player,* struct Game * *game,* struct Hand * *hand,* int *idCard*)

Function checks if the player can put a card down.

Parameters

player	The player who wants to put the card down.
game	The game where the player is located.
hand	The hand in which should put the card.
idCard	The id of the card.

Returns

1 if the player may to put the card down 0 if the player can't to put the card down other value on failure.

4.2.2.4 EXPORT struct Game* game_createGame (int *numberPoints*)

Allocates memory for and initializes a game.

Parameters

numberPoints	The number of points required for winning the game.

Returns

Pointer to the new game on success or NULL on failure.

4.2.2.5 EXPORT int game_deleteGame (struct Game ** *game*)

Frees the memory of a game and makes the pointer NULL.

Parameters

game	Pointer to the game to be deleted.

Returns

NO_ERROR on success, error code otherwise.

4.2.2.6 EXPORT int game_removePlayer (struct Player * *player,* struct Game * *game*)

Removes a player from a game.

Parameters

player	The player to be removed.
game	The game from where the player is to be removed.

Returns

NO_ERROR on success, error code otherwise.

4.2.2.7 EXPORT int game_removeTeam (struct Team * *team,* struct Game * *game*)

Removes a team from a game.

Parameters

team	The team to be removed.
game	The game from where the team is to be removed.

Returns

NO_ERROR on success, error code otherwise.

4.2.2.8 EXPORT struct Team* game_winningTeam (struct Game * *game*)

Searches the winning team of a game.

Parameters

game	The game in which the winning team is to be search.

Returns

Pointer to the winner team on success or NULL on failure.

4.3 src/libCruceGame/platform.h File Reference

Contains platform specific definitions.

4.3.1 Detailed Description

Contains platform specific definitions.

4.4 src/libCruceGame/round.h File Reference

Round and Hand structures, as well as helper functions.

```
#include "platform.h"
#include "deck.h"
#include "team.h"
#include "constants.h"
#include "errors.h"
```

Classes

- struct Hand

  Hand structure.
- struct Round

  Round structure.

Functions

- EXPORT struct Player * round_getBidWinner (const struct Round *round)

  Finds the winner of a bid in a round.
- EXPORT int round_placeBid (const struct Player *player, int bid, struct Round *round)

  Places the bid of a player.
- EXPORT int round_addPlayer (struct Player *player, struct Round *round)

  Add a player to a round.
- EXPORT int round_findPlayerIndexRound (const struct Player *player, const struct Round *round)

  Helper to find player in a round.
- EXPORT int round_addPlayerHand (struct Player *player, struct Hand *hand)

  Adds a player to a hand.
- EXPORT int round_putCard (struct Player *player, int cardId, struct Hand *hand)

  Places a card from a player to a hand.
- EXPORT int round_computeScore (const struct Hand *hand)

  Computes the score of a hand (in game points).
- EXPORT struct Round * round_createRound ()

  Allocates memory for and initializes a round.

- EXPORT int round_deleteRound (struct Round **round)
  Frees the memory of a round. Makes pointer NULL.
- EXPORT struct Hand * round_createHand ()
  Allocates memory for and initializes a hand.
- EXPORT int round_deleteHand (struct Hand **hand)
  Frees the memory of a hand. Makes pointer NULL.
- EXPORT int round_removePlayer (struct Player *player, struct Round *round)
  Removes a player from a round.
- EXPORT int round_removePlayerHand (struct Player *player, struct Hand *hand)
  Removes a player from a hand.
- EXPORT struct Player * round_handWinner (const struct Hand *hand, enum Suit trump, struct Round *round)
  Determines the winner of a hand.
- EXPORT int round_distributeCard (struct Deck *deck, const struct Round *round)
  Distributes one card to every player.
- EXPORT int round_distributeDeck (struct Deck *deck, const struct Round *round)
  Distributes cards to players.
- EXPORT int round_arrangePlayersHand (struct Round *round, int i)
  The function arranges the players in a hand.

4.4.1 Detailed Description

Round and Hand structures, as well as helper functions.

4.4.2 Function Documentation

4.4.2.1 EXPORT int round_addPlayer (struct Player * *player,* struct Round * *round*)

Add a player to a round.

Parameters

player	Pointer to the player to be added.
round	Pointer to the round where to add player.

Returns

NO_ERROR on success, error code otherwise.

4.4.2.2 EXPORT int round_addPlayerHand (struct Player * *player,* struct Hand * *hand*)

Adds a player to a hand.

Parameters

player	Pointer to the player to be added.
hand	Pointer to the hand where the player is to be added.

Returns

NO_ERROR on success, error code otherwise.

4.4.2.3 EXPORT int round_arrangePlayersHand (struct Round * *round,* int *i*)

The function arranges the players in a hand.

Parameters

round	Pointer to the round from which arranges it the players.
i	The position from where begin arranging.

Returns

NO_ERROR or 0 on success, other value on failure.

4.4.2.4 EXPORT int round_computeScore (const struct Hand * *hand*)

Computes the score of a hand (in game points).

Parameters

hand	Pointer to the hand for which the score is computed.

Returns

Integer representing the score or negative error code on failure.

4.4.2.5 EXPORT struct Hand* round_createHand ( )

Allocates memory for and initializes a hand.

Returns

Pointer to the new hand on success or NULL on failure.

4.4.2.6 EXPORT struct Round* round_createRound ( )

Allocates memory for and initializes a round.

Returns

Pointer to the new round on success or NULL on failure.

4.4.2.7 EXPORT int round_deleteHand (struct Hand ** *hand*)

Frees the memory of a hand. Makes pointer NULL.

Parameters

hand	Pointer to pointer to the hand to be deleted.

Returns

NO_ERROR on success, error code otherwise.

4.4.2.8 EXPORT int round_deleteRound (struct Round ** *round*)

Frees the memory of a round. Makes pointer NULL.

Parameters

round	Pointer to pointer to the round to be deleted.

Returns

NO_ERROR on success, error code otherwise.

4.4.2.9 EXPORT int round_distributeCard (struct Deck * *deck*, const struct Round * *round*)

Distributes one card to every player.

Parameters

deck	Pointer to the deck from where cards are distributed.
round	Pointer to the round containing the players that receive the cards.

Returns

NO_ERROR on success, error code otherwise.

4.4.2.10 EXPORT int round_distributeDeck (struct Deck * *deck*, const struct Round * *round*)

Distributes cards to players.

Parameters

deck	Pointer to the deck from where cards are distributed.
round	Pointer to the round that deck is distributed to.

Returns

NO_ERROR on success, error code otherwise.

4.4.2.11 EXPORT int round_findPlayerIndexRound (const struct Player * *player*, const struct Round * *round*)

Helper to find player in a round.

Parameters

player	Player to find.
round	Round to search for player.

Returns

Id of the player if found, negative value otherwise.

4.4.2.12 EXPORT struct Player* round_getBidWinner (const struct Round * *round*)

Finds the winner of a bid in a round.

Parameters

round	Pointer to the round where to find the bid winner.

Returns

Pointer to the bid winner player on success or NULL on failure.

4.4.2.13 EXPORT struct Player* round_handWinner (const struct Hand * *hand,* enum Suit *trump,* struct Round * *round*)

Determines the winner of a hand.

Parameters

hand	Pointer to the hand.
trump	The trump of round.
round	Pointer to the round containing the hand that the player won.

Returns

Pointer to the winning player or NULL on failure.

4.4.2.14 EXPORT int round_placeBid (const struct Player * *player,* int *bid,* struct Round * *round*)

Places the bid of a player.

Parameters

player	Pointer to the player who places the bid.
bid	The value of the bid.
round	Pointer to the round where to place the bid.

Returns

NO_ERROR on success, error code otherwise.

4.4.2.15 EXPORT int round_putCard (struct Player * *player,* int *cardId,* struct Hand * *hand*)

Places a card from a player to a hand.

Parameters

player	Pointer to the player who places the card.
cardId	Id of the card placed by the player (id from Player.cards).
hand	Pointer to the hand in which the card is placed.

Returns

NO_ERROR on success, error code otherwise.

4.4.2.16 EXPORT int round_removePlayer (struct Player * *player,* struct Round * *round*)

Removes a player from a round.

Parameters

player	Pointer to the player to be removed.
round	Pointer to the round from which the player is removed.

Returns

NO_ERROR on success, error code otherwise.

4.4.2.17 EXPORT int round_removePlayerHand (struct Player * *player*, struct Hand * *hand*)

Removes a player from a hand.

Parameters

player	Pointer to the player to be removed.
hand	Pointer to the hand from where the player is removed.

Returns

NO_ERROR on success, error code otherwise.

4.5 src/libCruceGame/team.h File Reference

Player and Team structures, with helper functions.

```
#include "deck.h"
```

Classes

- struct Player

  Player structure.
- struct Team

  Team structure.

Functions

- EXPORT struct Player * team_createPlayer (const char *name, int isHuman)

  Creates a player.
- EXPORT struct Team * team_createTeam (const char *name)

  Creates a team.
- EXPORT int team_addPlayer (struct Team *team, struct Player *player)

  Adds a player to a team.
- EXPORT int team_removePlayer (struct Team *team, const struct Player *player)

  Removes a player from a team.
- EXPORT int team_deleteTeam (struct Team **team)

  Deletes a team and sets the pointer to NULL.
- EXPORT int team_deletePlayer (struct Player **player)

  Deletes a player and sets the pointer to NULL.
- EXPORT int team_computeScore (const struct Team *team)

  Calculates the score of a team.
- EXPORT int team_addCard (struct Player *player, struct Card *card)

Passes a card to a player. The function doesn't check if the card has valid value and valid suit.

- EXPORT int team_hasCards (struct Player *player)

  Checks if a player has any card.

4.5.1 Detailed Description

Player and Team structures, with helper functions.

4.5.2 Function Documentation

4.5.2.1 EXPORT int team_addCard (struct Player * *player,* struct Card * *card*)

Passes a card to a player. The function doesn't check if the card has valid value and valid suit.

Parameters

player	The player who receives the card.
card	The card to be received.

Returns

NO_ERROR on success, error code otherwise.

4.5.2.2 EXPORT int team_addPlayer (struct Team * *team,* struct Player * *player*)

Adds a player to a team.

Parameters

team	The team to which the player is added.
player	The player to be added to the team.

Returns

NO_ERROR on success, error code otherwise.

4.5.2.3 EXPORT int team_computeScore (const struct Team * *team*)

Calculates the score of a team.

Parameters

team	The team for which the score is to be be calculated

Returns

Integer representing the score or negative error code on failure.

4.5.2.4 EXPORT struct Player* team_createPlayer (const char * *name,* int *isHuman*)

Creates a player.

Parameters

name	The name of the new player.
isHuman	Player type.

Returns

Pointer to the created player. Needs to be freed.

4.5.2.5 EXPORT struct Team* team_createTeam (const char * *name*)

Creates a team.

Parameters

name	The name of the new team.

Returns

Pointer to the created team. Needs to be freed.

4.5.2.6 EXPORT int team_deletePlayer (struct Player ** *player*)

Deletes a player and sets the pointer to NULL.

Parameters

player	The player to be deleted.

Returns

NO_ERROR on success, error code otherwise.

4.5.2.7 EXPORT int team_deleteTeam (struct Team ** *team*)

Deletes a team and sets the pointer to NULL.

Parameters

team	The team to be deleted.

Returns

NO_ERROR on success, error code otherwise.

4.5.2.8 EXPORT int team_hasCards (struct Player * *player*)

Checks if a player has any card.

Parameters

player	Pointer to the player that is ckecked

Returns

1 in case of succes, 0 otherwise

4.5.2.9 EXPORT int team_removePlayer (struct Team * *team,* const struct Player * *player*)

Removes a player from a team.

Parameters

team	The team from where the player is removed.
player	The player that will be removed.

Returns

NO_ERROR on success, error code otherwise.

Index

bids
    Round, 9

Card, 5
    suit, 5
    value, 5
cards
    Deck, 6
    Hand, 7

Deck, 5
    cards, 6
deck
    Game, 6
deck.h
    deck_cardsNumber, 12
    deck_compareCards, 13
    deck_createCard, 13
    deck_createDeck, 13
    deck_deckShuffle, 13
    deck_deleteCard, 13
    deck_deleteDeck, 14
deck_cardsNumber
    deck.h, 12
deck_compareCards
    deck.h, 13
deck_createCard
    deck.h, 13
deck_createDeck
    deck.h, 13
deck_deckShuffle
    deck.h, 13
deck_deleteCard
    deck.h, 13
deck_deleteDeck
    deck.h, 14

Game, 6
    deck, 6
    numberPlayers, 6
    players, 6
    pointsNumber, 7
    round, 7
    teams, 7
game.h
    game_addPlayer, 15
    game_addTeam, 15
    game_checkCard, 15
    game_createGame, 15
    game_deleteGame, 16
    game_removePlayer, 16
    game_removeTeam, 16
    game_winningTeam, 16
game_addPlayer
    game.h, 15
game_addTeam
    game.h, 15
game_checkCard
    game.h, 15
game_createGame
    game.h, 15
game_deleteGame
    game.h, 16
game_removePlayer
    game.h, 16
game_removeTeam
    game.h, 16
game_winningTeam
    game.h, 16

Hand, 7
    cards, 7
    players, 7
hand
    Player, 8
hands
    Round, 9

id
    Player, 8
    Round, 9
    Team, 10
isHuman
    Player, 8

name
    Player, 8
    Team, 10
numberPlayers
    Game, 6

Player, 8
    hand, 8
    id, 8
    isHuman, 8
    name, 8
    score, 8
players
    Game, 6
    Hand, 7

Round, 9
Team, 10
pointsNumber
Game, 7
Round, 9

Round, 8
bids, 9
hands, 9
id, 9
players, 9
pointsNumber, 9
trump, 9
round
Game, 7
round.h
round_addPlayer, 18
round_addPlayerHand, 18
round_arrangePlayersHand, 18
round_computeScore, 19
round_createHand, 19
round_createRound, 19
round_deleteHand, 19
round_deleteRound, 19
round_distributeCard, 20
round_distributeDeck, 20
round_findPlayerIndexRound, 20
round_getBidWinner, 20
round_handWinner, 21
round_placeBid, 21
round_putCard, 21
round_removePlayer, 21
round_removePlayerHand, 22
round_addPlayer
round.h, 18
round_addPlayerHand
round.h, 18
round_arrangePlayersHand
round.h, 18
round_computeScore
round.h, 19
round_createHand
round.h, 19
round_createRound
round.h, 19
round_deleteHand
round.h, 19
round_deleteRound
round.h, 19
round_distributeCard
round.h, 20
round_distributeDeck
round.h, 20
round_findPlayerIndexRound
round.h, 20
round_getBidWinner
round.h, 20
round_handWinner
round.h, 21
round_placeBid
round.h, 21
round_putCard
round.h, 21
round_removePlayer
round.h, 21
round_removePlayerHand
round.h, 22

score
Player, 8
src/libCruceGame/deck.h, 11
src/libCruceGame/game.h, 14
src/libCruceGame/platform.h, 17
src/libCruceGame/round.h, 17
src/libCruceGame/team.h, 22
suit
Card, 5

Team, 9
id, 10
name, 10
players, 10
team.h
team_addCard, 23
team_addPlayer, 23
team_computeScore, 23
team_createPlayer, 23
team_createTeam, 24
team_deletePlayer, 24
team_deleteTeam, 24
team_hasCards, 24
team_removePlayer, 24
team_addCard
team.h, 23
team_addPlayer
team.h, 23
team_computeScore
team.h, 23
team_createPlayer
team.h, 23
team_createTeam
team.h, 24
team_deletePlayer
team.h, 24
team_deleteTeam
team.h, 24
team_hasCards
team.h, 24
team_removePlayer
team.h, 24
teams
Game, 7
trump
Round, 9

value
Card, 5