

<sep13>

Erzeugt von Doxygen 1.8.5

Die Nov 5 2013 23:44:56

Inhaltsverzeichnis

1	Hierarchie-Verzeichnis	1
1.1	Klassenhierarchie	1
2	Klassen-Verzeichnis	2
2.1	Auflistung der Klassen	2
3	Klassen-Dokumentation	3
3.1	Ruleset.Card Klassenreferenz	3
3.1.1	Ausführliche Beschreibung	3
3.1.2	Dokumentation der Elementfunktionen	3
3.2	Ruleset.ClientHearts Klassenreferenz	4
3.2.1	Dokumentation der Elementfunktionen	4
3.3	Ruleset.ClientRuleset Klassenreferenz	4
3.3.1	Ausführliche Beschreibung	4
3.3.2	Dokumentation der Elementfunktionen	4
3.4	Ruleset.ClientWizard Klassenreferenz	5
3.4.1	Dokumentation der Elementfunktionen	5
3.5	Ruleset.Colour Enum-Referenz	6
3.6	Ruleset.GameClientUpdate Klassenreferenz	6
3.6.1	Ausführliche Beschreibung	6
3.6.2	Dokumentation der Elementfunktionen	6
3.7	Ruleset.GamePhase Enum-Referenz	7
3.8	Ruleset.GameState Klassenreferenz	7
3.8.1	Ausführliche Beschreibung	7
3.8.2	Dokumentation der Elementfunktionen	7
3.9	Ruleset.HeartsCard Klassenreferenz	8
3.9.1	Dokumentation der Elementfunktionen	8
3.10	Ruleset.HeartsData Klassenreferenz	9
3.10.1	Dokumentation der Elementfunktionen	9
3.11	Ruleset.HeartsID Enum-Referenz	9
3.12	Ruleset.OtherData Klassenreferenz	9
3.12.1	Dokumentation der Elementfunktionen	10
3.13	Ruleset.PlayerState Klassenreferenz	10
3.13.1	Ausführliche Beschreibung	10
3.13.2	Dokumentation der Elementfunktionen	10
3.14	Ruleset.RulesetType Enum-Referenz	11
3.15	Ruleset.ServerHearts Klassenreferenz	11
3.15.1	Ausführliche Beschreibung	11
3.15.2	Dokumentation der Elementfunktionen	11

3.16 Ruleset.ServerRuleset Klassenreferenz	12
3.16.1 Ausführliche Beschreibung	12
3.16.2 Dokumentation der Elementfunktionen	12
3.17 Ruleset.ServerWizard Klassenreferenz	13
3.17.1 Ausführliche Beschreibung	14
3.17.2 Dokumentation der Elementfunktionen	14
3.18 Ruleset.WizardCard Klassenreferenz	15
3.19 Ruleset.WizData Klassenreferenz	15
3.19.1 Dokumentation der Elementfunktionen	15
3.20 Ruleset.WizID Enum-Referenz	16

Index

17

1 Hierarchie-Verzeichnis

1.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

Ruleset.Card	3
Ruleset.HeartsCard	8
Ruleset.WizardCard	15
Ruleset.ClientRuleset	4
Ruleset.ClientHearts	4
Ruleset.ClientWizard	5
Ruleset.Colour	6
Ruleset.GameClientUpdate	6
Ruleset.GamePhase	7
Ruleset.GameState	7
Ruleset.HeartsID	9
Ruleset.OtherData	9
Ruleset.HeartsData	9
Ruleset.WizData	15
Ruleset.PlayerState	10
Ruleset.RulesetType	11
Ruleset.ServerRuleset	12
Ruleset.ServerHearts	11

Ruleset.ServerWizard	13
Ruleset.WizID	16

2 Klassen-Verzeichnis

2.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

Ruleset.Card	
Diese Klasse modelliert eine Spielkarte	3
Ruleset.ClientHearts	
Diese Klasse bildet das Regelwerk fr den Client bei einer Partie Hearts	4
Ruleset.ClientRuleset	
ClientRuleset ist eine abstrakte Klasse und wird zur Regelvorauswertung im Client verwendet	4
Ruleset.ClientWizard	
Diese Klasse bildet das Regelwerk fr den Client bei einer Partie Wizard	5
Ruleset.Colour	
Reprntiert die Farbe einer Karte	6
Ruleset.GameClientUpdate	
Das GameClientUpdate wird vom RuleSet ber den GameServer an den Client geschickt und enth alle derungen des GameState , die fr den Client relevant sind	6
Ruleset.GamePhase	
Die GamePhase modelliert die verschiedenen Zuste des Spiels im GameState	7
Ruleset.GameState	
Das GameState modelliert einen aktuellen Spielzustand, es wird vom GameServer instanziiert und vom RuleSet bearbeitet	7
Ruleset.HeartsCard	
Modelliert eine Heartskarte	8
Ruleset.HeartsData	
Die zuslichen Informationen eines Spielers zum Spiel Hearts	9
Ruleset.HeartsID	
Die eindeutigen IDs zu jeder Heartskarte	9
Ruleset.OtherData	
OtherData ist abstract und speichert die zuslichen Informationen eines Spielers	9
Ruleset.PlayerState	
Reprntiert den Spielzustand eines Spielers, und wird unter anderem im GameState gespeichert	10
Ruleset.RulesetType	
Die verschiedenen Regelwerke	11
Ruleset.ServerHearts	
Diese Klasse erstellt das Regelwerk zum Spiel Hearts	11

Ruleset.ServerRuleset

Das **ServerRuleset** ist eine akstrakte Klasse und fr den Ablauf und die Einhaltung der Regeln eines Spiels zustig (/L280/)

12

Ruleset.ServerWizard

Diese Klasse erstellt das Regelwerk zum Spiel Wizard

13

Ruleset.WizardCard

Modelliert eine Wizardkarte

15

Ruleset.WizData

Die zuslichen Informationen eines Spielers zum Spiel Wizard

15

Ruleset.WizID

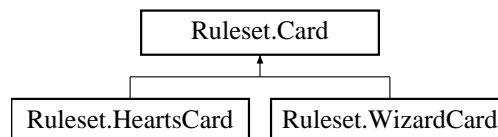
Die eindeutigen IDs zu jeder Wizardkarte

16

3 Klassen-Dokumentation

3.1 Ruleset.Card Klassenreferenz

Klassendiagramm für Ruleset.Card:



Öffentliche Methoden

- int **getValue** ()
- Colour **getColour** ()

3.1.1 Ausführliche Beschreibung

Jede Karte besitzt als Attribute einen Wert und eine Farbe.

3.1.2 Dokumentation der Elementfunktionen

3.1.2.1 Colour Ruleset.Card.getColour ()

Holt die Farbe der Karte.

Rückgabe

Die Farbe der Karte

3.1.2.2 int Ruleset.Card.getValue ()

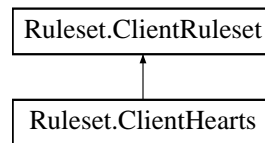
Holt den logischen Wert der Karte.

Rückgabe

Der logische Wert der Karte

3.2 Ruleset.ClientHearts Klassenreferenz

Klassendiagramm für Ruleset.ClientHearts:



Öffentliche Methoden

- boolean `isValidMove` (`Card card`)

3.2.1 Dokumentation der Elementfunktionen

3.2.1.1 boolean Ruleset.ClientHearts.isValidMove (Card card) [virtual]

erprüft ob ein gemachter Zug zu dem Spiel Hearts gtlig ist.

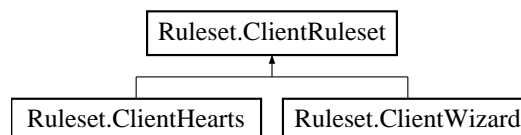
Rückgabe

isValid true falls Zug gtlig, false wenn nicht

Implementiert `Ruleset.ClientRuleSet`.

3.3 Ruleset.ClientRuleSet Klassenreferenz

Klassendiagramm für Ruleset.ClientRuleSet:



Öffentliche Methoden

- void `resolveMessage` (`MsgClientUpdate clientUpdate`)
- void `resolveMessage` (`MsgCardRequest msgCardRequest`)
- void `resolveMessage` (`MsgMultipleCardsRequest msgMultiCardsRequest`)
- void `resolveMessage` (`MsgNumberRequest msgNumber`)
- void `resolveMessage` (`MsgSelectionRequest msgSelection`)

3.3.1 Ausführliche Beschreibung

Dazu benutzt es die `isValidMove()` Methode. Des Weiteren kann es vom ClientModel erhaltene RulesetMessages mit der `resolveMessage()` Methode behandeln.

3.3.2 Dokumentation der Elementfunktionen

3.3.2.1 void Ruleset.ClientRuleSet.resolveMessage (MsgClientUpdate clientUpdate)

Verarbeitet die RulesetMessage dass der Server ein Spielupdate an den Client schickt.

Parameter

<i>clientUpdate</i>	Die Nachricht vom Server
---------------------	--------------------------

3.3.2.2 void Ruleset.ClientRuleset.resolveMessage (MsgCardRequest msgCardRequest)

Verarbeitet die RulesetMessage dass der Server von dem Spieler verlangt eine Karte zu spielen.

Parameter

<i>msgCard-Request</i>	Die Nachricht vom Server
------------------------	--------------------------

3.3.2.3 void Ruleset.ClientRuleset.resolveMessage (MsgMultipleCardsRequest msgMultiCardsRequest)

Verarbeitet die RulesetMessage dass der Server von dem Spieler verlangt mehrere Karten anzugeben.

Parameter

<i>msgMultiCards-Request</i>	Die Nachricht vom Server
------------------------------	--------------------------

3.3.2.4 void Ruleset.ClientRuleset.resolveMessage (MsgNumberRequest msgNumber)

Verarbeitet die RulesetMessage dass der Server von dem Spieler verlangt eine Stichanzahl anzugeben.

Parameter

<i>msgNumber</i>	Die Nachricht vom Server
------------------	--------------------------

3.3.2.5 void Ruleset.ClientRuleset.resolveMessage (MsgSelectionRequest msgSelection)

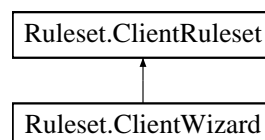
Verarbeitet die RulesetMessage dass der Server von dem Spieler verlangt eine Farbe auszuwählen.

Parameter

<i>msgSelection</i>	Die Nachricht vom Server
---------------------	--------------------------

3.4 Ruleset.ClientWizard Klassenreferenz

Klassendiagramm für Ruleset.ClientWizard:



Öffentliche Methoden

- boolean isValidMove (Card card)

3.4.1 Dokumentation der Elementfunktionen

3.4.1.1 boolean Ruleset.ClientWizard.isValidMove (Card card) [virtual]

Prft ob ein gemachter Zug zum Spiel Wizard gltig ist.

Rückgabe

isValid true falls Zug gltig, false wenn nicht

Implementiert [Ruleset.ClientRuleset](#).

3.5 Ruleset.Colour Enum-Referenz**3.6 Ruleset.GameClientUpdate Klassenreferenz****Öffentliche Methoden**

- Set< [Card](#) > [getOwnHand](#) ()
- Set< [Card](#) > [getPlayedCards](#) ()
- [OtherData](#) [getOwnData](#) ()
- Set< [OtherData](#) > [getOtherPlayerData](#) ()

3.6.1 Ausführliche Beschreibung

Das wn seine Spielhand, der Ablagestapel sowie die Otherdata von allen Spielern. Bei Wizard enth es auch die momentane Trumpfkarte.

3.6.2 Dokumentation der Elementfunktionen**3.6.2.1 Set<OtherData> Ruleset.GameClientUpdate.getOtherPlayerData ()**

Holt die Spieldaten der anderen Spieler.

Rückgabe

otherPlayerData Die Spieldaten der anderen Spieler

3.6.2.2 OtherData Ruleset.GameClientUpdate.getOwnData ()

Holt die zuslichen Spieldaten des Client.

Rückgabe

ownData Die Spieldaten des Clients

3.6.2.3 Set<Card> Ruleset.GameClientUpdate.getOwnHand ()

Holt die Karten die der Client auf der Hand hat.

Rückgabe

ownHand Die Hand des Clients

3.6.2.4 Set<Card> Ruleset.GameClientUpdate.getPlayedCards ()

Holt die gespielten Karten auf dem Ablagestapel.

Rückgabe

playedCards Die gespielten Karten

3.7 Ruleset.GamePhase Enum-Referenz

3.8 Ruleset.GameState Klassenreferenz

Öffentliche Methoden

- void [setCurrentPlayer](#) ([PlayerState](#) player)
- [PlayerState](#) [getCurrentPlayer](#) ()
- Set< [Card](#) > [getCardsLeftInDeck](#) ()
- void [getPlayedCards](#) ()
- [PlayerState](#) [getPlayer](#) (String name)
- void [setTrumpCard](#) ([Card](#) trumpCard)
- [Card](#) [getTrumpCard](#) ()
- void [playCard](#) ([Card](#) card)

3.8.1 Ausführliche Beschreibung

Es enth die einzelnen PlayerStates, sowie Informationen zum Ablage-, Aufnahmestapel, Rundenanzahl, den momentan aktiven Spieler sowie [GamePhase](#).

3.8.2 Dokumentation der Elementfunktionen

3.8.2.1 Set<Card> Ruleset.GameState.getCardsLeftInDeck ()

Holt die Karten die noch im Aufnahmestapel sind.

Rückgabe

cardsLeftInDeck Holt die Karten die noch im Aufnahmestapel sind

3.8.2.2 PlayerState Ruleset.GameState.getCurrentPlayer ()

Holt den Spieler der momentan am Zug ist.

Rückgabe

currentPlayer Der Spielzustand des Spielers der grad am Zug ist

3.8.2.3 void Ruleset.GameState.getPlayedCards ()

Holt die gespielten Karten im Ablagestapel.

Rückgabe

playedCards Die gespielten Karten

3.8.2.4 PlayerState Ruleset.GameState.getPlayer (String name)

Holt einen bestimmten Spieler.

Parameter

<i>name</i>	Der Name des Spielers
-------------	-----------------------

Rückgabe

player Der Spielzustand des Spielers

3.8.2.5 Card Ruleset.GameState.getTrumpCard ()

Holt die momentane Trumpfkarte im Spiel.

Rückgabe

trumpCard Die momentane Trumpfkarte

3.8.2.6 void Ruleset.GameState.playCard (Card card)

Entfernt eine Karte aus der Hand des currentPlayer und legt sie auf dem Ablagestapel.

Parameter

<i>card</i>	Die gespielte Karte
-------------	---------------------

3.8.2.7 void Ruleset.GameState.setCurrentPlayer (PlayerState player)

Setzt einen neuen Spieler als currentPlayer.

Parameter

<i>player</i>	Der neue currentPlayer
---------------	------------------------

3.8.2.8 void Ruleset.GameState.setTrumpCard (Card trumpCard)

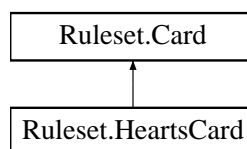
Setzt die Trumpfkarte.

Parameter

<i>trumpCard</i>	Die Trumpfkarte
------------------	-----------------

3.9 Ruleset.HeartsCard Klassenreferenz

Klassendiagramm für Ruleset.HeartsCard:

**Öffentliche Methoden**

- int [getPoints](#) ()
- [HeartsID](#) [getID](#) ()

3.9.1 Dokumentation der Elementfunktionen

3.9.1.1 HeartsID Ruleset.HeartsCard.getID ()

Holt die ID der Karte.

Rückgabe

points Die ID der Karte

3.9.1.2 int Ruleset.HeartsCard.getPoints ()

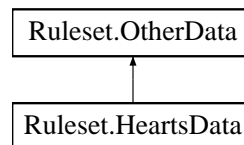
Holt die Punkte der Karte.

Rückgabe

points Die Punkte der Karte

3.10 Ruleset.HeartsData Klassenreferenz

Klassendiagramm für Ruleset.HeartsData:



Öffentliche Methoden

- int [getCompletePoints](#) ()
- int [getCurrentPoints](#) ()

3.10.1 Dokumentation der Elementfunktionen

3.10.1.1 int Ruleset.HeartsData.getCompletePoints ()

Holt den gesamten Punktestand eines Spielers.

Rückgabe

Der Gesamtpunktstand eines Spielers

3.10.1.2 int Ruleset.HeartsData.getCurrentPoints ()

Holt den momentanen Punktestand eines Spielers.

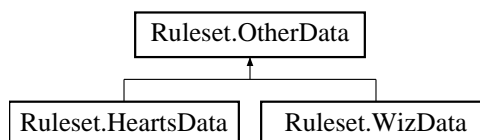
Rückgabe

Der momentane Punktestand eines Spielers

3.11 Ruleset.HeartsID Enum-Referenz

3.12 Ruleset.OtherData Klassenreferenz

Klassendiagramm für Ruleset.OtherData:



Öffentliche Methoden

- String [getname](#) ()

3.12.1 Dokumentation der Elementfunktionen

3.12.1.1 String Ruleset.OtherData.getname ()

Holt den Namen des Spielers.

Rückgabe

name Der Name des Spielers

3.13 Ruleset.PlayerState Klassenreferenz

Öffentliche Methoden

- String [getName](#) ()
- Set< [Card](#) > [getHand](#) ()
- [OtherData](#) [getOtherData](#) ()
- void [addCard](#) ([Card](#) card)
- void [removeCard](#) ([Card](#) card)

3.13.1 Ausführliche Beschreibung

Sie enth den Namen des Spielers, seine Handkarten und [OtherData](#).

3.13.2 Dokumentation der Elementfunktionen

3.13.2.1 void Ruleset.PlayerState.addCard ([Card](#) card)

Gibt dem Spieler eine Karte.

Parameter

card	Die Karte die dem Spieler gegeben wird
----------------------	--

3.13.2.2 Set<[Card](#)> Ruleset.PlayerState.getHand ()

Holt die Kartenhand des Spielers.

Rückgabe

ownHand Die Kartenhand des Spielers

3.13.2.3 String Ruleset.PlayerState.getName ()

Holt den namen eines Spielers.

Rückgabe

name Der Name des Spielers

3.13.2.4 OtherData Ruleset.PlayerState.getOtherData ()

Holt die zuslichen Informationen des Spielers.

Rückgabe

ownHand Die zuslichen Informationen des Spielers

3.13.2.5 void Ruleset.PlayerState.removeCard (Card card)

Entfernt eine Karte aus der Hand des Spielers.

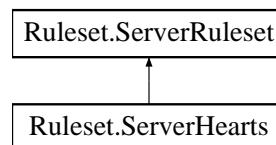
Parameter

<i>card</i>	
-------------	--

3.14 Ruleset.RulesetType Enum-Referenz

3.15 Ruleset.ServerHearts Klassenreferenz

Klassendiagramm für Ruleset.ServerHearts:



Öffentliche Methoden

- boolean [isValidMove](#) (Card card)
- MsgMultipleCardsRequest [sendMultiCardRequest](#) (String name)

3.15.1 Ausführliche Beschreibung

Sie enth zudem weitere Methoden, welche fr das Spiel Hearts spezifisch bentigt werden, wie die Regelung zum Tausch von Karten und die Berechnung der Stichpunkten.

3.15.2 Dokumentation der Elementfunktionen

3.15.2.1 boolean Ruleset.ServerHearts.isValidMove (Card card) [virtual]

Prft ob ein gemachter Zug in einem Hearts Spiel gltig ist.

Rückgabe

isValid true falls Zug gltig, false wenn nicht

Implementiert [Ruleset.ServerRuleset](#).

3.15.2.2 MsgMultipleCardsRequest Ruleset.ServerHearts.sendMultiCardRequest (String name)

Schickt einem Spieler die Aufforderung Karten zu geben.

Parameter

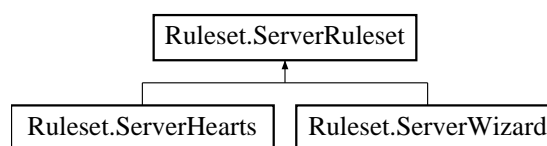
<i>name</i>	Name des Spielers
-------------	-------------------

Rückgabe

new MsgMultipleCardsRequest() Eine Nachricht an den Spieler

3.16 Ruleset.ServerRuleset Klassenreferenz

Klassendiagramm für Ruleset.ServerRuleset:



Öffentliche Methoden

- [PlayerState](#) [getCurrentPlayer](#) ()
- [PlayerState](#) [getPlayerState](#) (String name)
- void [resolveMessage](#) (MsgCard msgCard, String name)
- void [resolveMessage](#) (MsgMultiCards msgMultiCard, String name)
- void [resolveMessage](#) (MsgNumber msgNumber, String name)
- void [resolveMessage](#) (MsgSelection msgSelection, String name)
- MsgCardRequest [sendCardRequest](#) (String name)

3.16.1 Ausführliche Beschreibung

Das [ServerRuleset](#) wird im GameServer instanziiert und verwaltet die Zustände des GameStates im Server. Mit der Methode [isValidMove\(\)](#) wird eine Eingabe eines Clients auf Regelkonformität überprüft und dann im GameServer das [GameState](#) verortet. [resolveMessage\(\)](#) kann eine GameServerinstanz eine RulesetMessage vom Player an das Ruleset weiterleiten.

3.16.2 Dokumentation der Elementfunktionen

3.16.2.1 PlayerState Ruleset.ServerRuleset.getCurrentPlayer ()

Holt den Spieler der gerade am Zug ist.

Rückgabe

currentPlayer Der Spielzustand des Spielers der grad am Zug ist

3.16.2.2 PlayerState Ruleset.ServerRuleset.getPlayerState (String name)

Holt den Spielerzustand eines Spielers.

Parameter

<i>name</i>	Der Name des Spielers
-------------	-----------------------

Rückgabe

playerState Spielzustand eines Spielers

3.16.2.3 void Ruleset.ServerRuleset.resolveMessage (MsgCard msgCard, String name)

Verarbeitet die RulesetMessage dass eine Karte vom Spieler gespielt.

Parameter

<i>msgCard</i>	Die Nachricht vom Client welche Karte gespielt wurde
<i>name</i>	Der Name des Spielers

3.16.2.4 void Ruleset.ServerRuleset.resolveMessage (MsgMultiCards msgMultiCard, String name)

Verarbeitet die RulesetMessage dass mehrere Karten von einem Spieler bergeben wurden.

Parameter

<i>msgMultiCard</i>	Die Nachricht vom Client
<i>name</i>	Der Name des Spielers

3.16.2.5 void Ruleset.ServerRuleset.resolveMessage (MsgNumber msgNumber, String name)

Verarbeitet die RulesetMessage dass ein Spieler eine Stichangabe gemacht hat.

Parameter

<i>msgNumber</i>	Die Nachricht vom Client
<i>name</i>	Der Name des Spielers

3.16.2.6 void Ruleset.ServerRuleset.resolveMessage (MsgSelection msgSelection, String name)

Verarbeitet die RulesetMessage dass ein Spieler eine Farbe ausgewt hat.

Parameter

<i>msgSelection</i>	Die Nachricht vom Client
<i>name</i>	

3.16.2.7 MsgCardRequest Ruleset.ServerRuleset.sendCardRequest (String name)

Schickt einem Spieler die Aufforderung eine Karte zu spielen.

Parameter

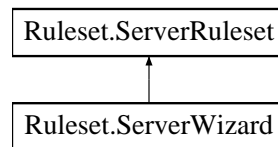
<i>name</i>	Name des Spielers
-------------	-------------------

Rückgabe

new MsgCardRequest() Eine Nachricht an den Spieler

3.17 Ruleset.ServerWizard Klassenreferenz

Klassendiagramm für Ruleset.ServerWizard:



Öffentliche Methoden

- boolean `isValidMove` (`Card card`)
- `MsgNumberRequest` `sendNumberRequest` (`String name`)
- `MsgSelectionRequest` `sendSelectionRequest` (`String name`)

3.17.1 Ausführliche Beschreibung

Sie enth zudem weitere Methoden, welche fr das Spiel Wizard spezifisch bentigt werden, wie das Bestimmen einer Trumpffarbe und die Bestimmung der Rundenanzahl.

3.17.2 Dokumentation der Elementfunktionen

3.17.2.1 boolean `Ruleset.ServerWizard.isValidMove (Card card)` `[virtual]`

Prft ob ein gemachter Zug in einem Wizard Spiel gltig ist.

Rückgabe

`isValid` true falls Zug gltig, false wenn nicht

Implementiert `Ruleset.ServerRuleset`.

3.17.2.2 `MsgNumberRequest Ruleset.ServerWizard.sendNumberRequest (String name)`

Schickt einem Spieler die Aufforderung eine Stichanzahl anzugeben.

Parameter

<i>name</i>	Name des Spielers
-------------	-------------------

Rückgabe

`new MsgNumberRequest()` Eine Nachricht an den Spieler

3.17.2.3 `MsgSelectionRequest Ruleset.ServerWizard.sendSelectionRequest (String name)`

Schickt einem Spieler die Aufforderung eine Farbe zu wen.

Parameter

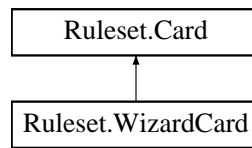
<i>name</i>	Name des Spielers
-------------	-------------------

Rückgabe

`new MsgSelectionRequest()` Eine Nachricht an den Spieler

3.18 Ruleset.WizardCard Klassenreferenz

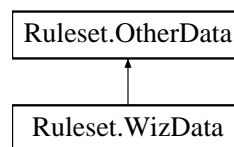
Klassendiagramm für Ruleset.WizardCard:



Weitere Geerbte Elemente

3.19 Ruleset.WizData Klassenreferenz

Klassendiagramm für Ruleset.WizData:



Öffentliche Methoden

- int [getAnnouncedTricks](#) ()
- int [getAchievedTricks](#) ()
- int [getPoints](#) ()
- void [announceTricks](#) (int annouceTricks)
- void [setPoints](#) (int points)

3.19.1 Dokumentation der Elementfunktionen

3.19.1.1 void Ruleset.WizData.announceTricks (int *annouceTricks*)

Beim Spielstart werden die vorausgesagten Stiche des Spieler gespeichert und die gemachten Stiche zurückgesetzt.

Parameter

<i>annouceTricks</i>	Die vorausgesagten Stiche des Spielers
----------------------	--

3.19.1.2 int Ruleset.WizData.getAchievedTricks ()

Holt die erreichten Stiche des Spielers.

Rückgabe

achievedTricks Die gemachten Stiche eines Spielers

3.19.1.3 int Ruleset.WizData.getAnnouncedTricks ()

Holt die angesagten Stiche des Spielers.

Rückgabe

announcedTricks Die angesagten Stiche

3.19.1.4 `int Ruleset.WizData.getPoints ()`

Holt den Punktestand des Spielers.

Rückgabe

points Der Punktestand des Spielers

3.19.1.5 `void Ruleset.WizData.setPoints (int points)`

Setzt den Punktestand eines Spielers.

Parameter

<i>points</i>	Der Punktestand eines Spielers
---------------	--------------------------------

3.20 Ruleset.WizID Enum-Referenz

Index

addCard
 Ruleset::PlayerState, 10
announceTricks
 Ruleset::WizData, 15

getAchievedTricks
 Ruleset::WizData, 15
getAnnouncedTricks
 Ruleset::WizData, 15
getCardsLeftInDeck
 Ruleset::GameState, 7
getColour
 Ruleset::Card, 3
getCompletePoints
 Ruleset::HeartsData, 9
getCurrentPlayer
 Ruleset::GameState, 7
 Ruleset::ServerRuleset, 12
getCurrentPoints
 Ruleset::HeartsData, 9
getHand
 Ruleset::PlayerState, 10
getID
 Ruleset::HeartsCard, 8
getName
 Ruleset::PlayerState, 10
getOtherData
 Ruleset::PlayerState, 11
getOtherPlayerData
 Ruleset::GameClientUpdate, 6
getOwnData
 Ruleset::GameClientUpdate, 6
getOwnHand
 Ruleset::GameClientUpdate, 6
getPlayedCards
 Ruleset::GameClientUpdate, 6
 Ruleset::GameState, 7
getPlayer
 Ruleset::GameState, 7
getPlayerState
 Ruleset::ServerRuleset, 12
getPoints
 Ruleset::HeartsCard, 9
 Ruleset::WizData, 15
getTrumpCard
 Ruleset::GameState, 8
getValue
 Ruleset::Card, 3
getname
 Ruleset::OtherData, 10

isValidMove
 Ruleset::ClientHearts, 4
 Ruleset::ClientWizard, 5
 Ruleset::ServerHearts, 11
 Ruleset::ServerWizard, 14

playCard
 Ruleset::GameState, 8

removeCard
 Ruleset::PlayerState, 11
resolveMessage
 Ruleset::ClientRuleset, 4, 5
 Ruleset::ServerRuleset, 13
Ruleset.Card, 3
Ruleset.ClientHearts, 4
Ruleset.ClientRuleset, 4
Ruleset.ClientWizard, 5
Ruleset.Colour, 6
Ruleset.GameClientUpdate, 6
Ruleset.GamePhase, 7
Ruleset.GameState, 7
Ruleset.HeartsCard, 8
Ruleset.HeartsData, 9
Ruleset.HeartsID, 9
Ruleset.OtherData, 9
Ruleset.PlayerState, 10
Ruleset.RulesetType, 11
Ruleset.ServerHearts, 11
Ruleset.ServerRuleset, 12
Ruleset.ServerWizard, 13
Ruleset.WizData, 15
Ruleset.WizID, 16
Ruleset.WizardCard, 15
Ruleset::Card
 getColour, 3
 getValue, 3
Ruleset::ClientHearts
 isValidMove, 4
Ruleset::ClientRuleset
 resolveMessage, 4, 5
Ruleset::ClientWizard
 isValidMove, 5
Ruleset::GameClientUpdate
 getOtherPlayerData, 6
 getOwnData, 6
 getOwnHand, 6
 getPlayedCards, 6
Ruleset::GameState
 getCardsLeftInDeck, 7
 getCurrentPlayer, 7
 getPlayedCards, 7
 getPlayer, 7
 getTrumpCard, 8
 playCard, 8
 setCurrentPlayer, 8
 setTrumpCard, 8
Ruleset::HeartsCard
 getID, 8
 getPoints, 9
Ruleset::HeartsData

- getCompletePoints, 9
- getCurrentPoints, 9
- Ruleset::OtherData
 - getname, 10
- Ruleset::PlayerState
 - addCard, 10
 - getHand, 10
 - getName, 10
 - getOtherData, 11
 - removeCard, 11
- Ruleset::ServerHearts
 - isValidMove, 11
 - sendMultiCardRequest, 11
- Ruleset::ServerRuleset
 - getCurrentPlayer, 12
 - getPlayerState, 12
 - resolveMessage, 13
 - sendCardRequest, 13
- Ruleset::ServerWizard
 - isValidMove, 14
 - sendNumberRequest, 14
 - sendSelectionRequest, 14
- Ruleset::WizData
 - announceTricks, 15
 - getAchievedTricks, 15
 - getAnnouncedTricks, 15
 - getPoints, 15
 - setPoints, 16
- sendCardRequest
 - Ruleset::ServerRuleset, 13
- sendMultiCardRequest
 - Ruleset::ServerHearts, 11
- sendNumberRequest
 - Ruleset::ServerWizard, 14
- sendSelectionRequest
 - Ruleset::ServerWizard, 14
- setCurrentPlayer
 - Ruleset::GameState, 8
- setPoints
 - Ruleset::WizData, 16
- setTrumpCard
 - Ruleset::GameState, 8