

Card-Shuffeling

Version 0.1

Erzeugt von Doxygen 1.8.6

Die Jan 12 2016 10:40:05

Inhaltsverzeichnis

1	Card-Shuffling	1
2	Klassen-Verzeichnis	1
2.1	Auflistung der Klassen	1
3	Datei-Verzeichnis	1
3.1	Auflistung der Dateien	1
4	Klassen-Dokumentation	2
4.1	CARD Strukturreferenz	2
4.1.1	Ausführliche Beschreibung	2
4.1.2	Dokumentation der Datenelemente	2
4.2	deck Klassenreferenz	2
4.2.1	Ausführliche Beschreibung	3
4.2.2	Beschreibung der Konstruktoren und Destruktoren	3
4.2.3	Dokumentation der Elementfunktionen	4
4.2.4	Dokumentation der Datenelemente	4
4.3	match Klassenreferenz	4
4.3.1	Ausführliche Beschreibung	5
4.3.2	Beschreibung der Konstruktoren und Destruktoren	6
4.3.3	Dokumentation der Elementfunktionen	6
4.3.4	Dokumentation der Datenelemente	6
4.4	player Klassenreferenz	6
4.4.1	Ausführliche Beschreibung	7
4.4.2	Beschreibung der Konstruktoren und Destruktoren	7
4.4.3	Dokumentation der Elementfunktionen	7
4.4.4	Dokumentation der Datenelemente	7
5	Datei-Dokumentation	8
5.1	main.cpp-Dateireferenz	8
5.1.1	Ausführliche Beschreibung	8
5.1.2	Dokumentation der Funktionen	8
5.2	mainpage.dox-Dateireferenz	9
5.3	match.cpp-Dateireferenz	9
5.4	match.h-Dateireferenz	9
5.4.1	Makro-Dokumentation	10
5.5	player.cpp-Dateireferenz	11
5.6	player.h-Dateireferenz	11
5.6.1	Ausführliche Beschreibung	12
5.7	set.cpp-Dateireferenz	12

5.8	set.h-Dateireferenz	12
5.8.1	Makro-Dokumentation	13
	Index	14

1 Card-Shuffeling

Das Card-Shuffeling-Tool dient zum Vorzeigen einer Doxygen-Dokumentationen. Das Programm wurde absichtlich mit einem Overhead an Klassen und Funktionen geschrieben, um das Verhalten von Doxygen zeigen zu können.

Das Programm Card-Shuffeling dient zur zufälligen Verteilung von mehreren Kartendecks (32 Karten, 4 Farben a 8 Karten) an eine eingestellte Zahl an Spielern. Es soll statistisch Auswerten, wie häufig ein "blanker Zehner" in Abhängigkeit der Anzahl der Decks und Spieler auftritt.

2 Klassen-Verzeichnis

2.1 Auflistung der Klassen

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

CARD		
CARD ist ein Struct zur Darstellung von Karten		??
deck		
Die Klasse deck repräsentiert ein gewöhnliches Kartendeck		??
match		
Die Klasse match generiert eine Rund eines Kartenspiels an welchem PLAYERS Spieler teilnehmen und SIZE_DECKS Decks verwendet werden		??
player		
Die Klasse player stellt einen Teilernehmer des Kartenspiels match dar		??

3 Datei-Verzeichnis

3.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

main.cpp		
Initialise cards matches		??
match.cpp		??
match.h		??
player.cpp		??
player.h		
Datie zur Verwaltung der Klasse(n) player		??

[set.cpp](#)

??

[set.h](#)

??

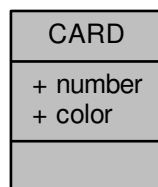
4 Klassen-Dokumentation

4.1 CARD Strukturreferenz

[CARD](#) ist ein Struct zur Darstellung von Karten.

```
#include <set.h>
```

Zusammengehörigkeiten von CARD:



Öffentliche Attribute

- int [number](#)
- int [color](#)

4.1.1 Ausführliche Beschreibung

[CARD](#) ist ein Struct zur Darstellung von Karten.

Dabei gibt Number die Nummer (7, 8, 9, 10, K, O, U, A) an und color steht für die ID einer Farbe.

Definiert in Zeile 19 der Datei set.h.

4.1.2 Dokumentation der Datenelemente

4.1.2.1 int CARD::color

Definiert in Zeile 21 der Datei set.h.

Wird benutzt von `deck::deck()`.

4.1.2.2 int CARD::number

Definiert in Zeile 20 der Datei set.h.

Wird benutzt von `deck::deck()`.

Die Dokumentation für diese Struktur wurde erzeugt aufgrund der Datei:

- [set.h](#)

4.2 deck Klassenreferenz

Die Klasse deck repräsentiert ein gewöhnliches Kartendeck.

```
#include <set.h>
```

Zusammengehörigkeiten von deck:

deck
- cards
+ deck() + getCards() + isEmpty() + getRandomCard() - remove()

Öffentliche Methoden

- `deck ()`
deck::deck Füllt ein Deck mit `COLORS*NUMBERS` Karten auf.
- `std::vector< CARD > getCards (void)`
- `int isEmpty ()`
- `CARD getRandomCard ()`
deck::getRandomCard zieht eine zufällige Karte auf dem Kartendeck und entfernt diese.

Private Methoden

- `void remove (std::vector< CARD > &vec, size_t pos)`
deck::remove Entfernt aus dem Vector vec das Item an der Stelle pos.

Private Attribute

- `std::vector< CARD > cards`

4.2.1 Ausführliche Beschreibung

Die Klasse deck repräsentiert ein gewöhnliches Kartendeck.

Hier dient sie um größere Kartendecks innerhalb eines Matches zusammen zu stellen.

Definiert in Zeile 29 der Datei set.h.

4.2.2 Beschreibung der Konstruktoren und Destruktoren

4.2.2.1 deck::deck ()

deck::deck Füllt ein Deck mit `COLORS*NUMBERS` Karten auf.

Definiert in Zeile 6 der Datei set.cpp.

Benutzt cards, CARD::color, COLORS, CARD::number und NUMBERS.

4.2.3 Dokumentation der Elementfunktionen

4.2.3.1 `std::vector<CARD> deck::getCards (void) [inline]`

Definiert in Zeile 33 der Datei set.h.

Benutzt cards.

4.2.3.2 `CARD deck::getRandomCard ()`

[deck::getRandomCard](#) zieht eine zufällige Karte auf dem Kartendeck und entfernt diese.

Rückgabe

gezogene Karte.

Definiert in Zeile 22 der Datei set.cpp.

Benutzt cards.

4.2.3.3 `int deck::isEmpty () [inline]`

Definiert in Zeile 34 der Datei set.h.

Benutzt cards.

4.2.3.4 `void deck::remove (std::vector< CARD > & vec, size_t pos) [private]`

[deck::remove](#) Entfernt aus dem Vector vec das Item an der Stelle pos.

Parameter

<code>vec</code>	vector aus welchem ein Item entfernt werden soll.
<code>pos</code>	Position von welchem das Item entfernt werden soll.

Definiert in Zeile 37 der Datei set.cpp.

4.2.4 Dokumentation der Datenelemente

4.2.4.1 `std::vector<CARD> deck::cards [private]`

Definiert in Zeile 39 der Datei set.h.

Wird benutzt von `deck()`, `getCards()`, `getRandomCard()` und `isEmpty()`.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

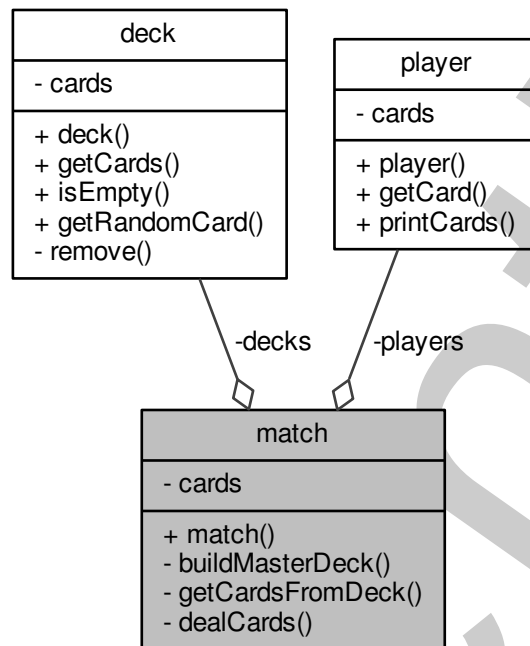
- [set.h](#)
- [set.cpp](#)

4.3 match Klassenreferenz

Die Klasse [match](#) generiert eine Rund eines Kartenspiels an welchem [PLAYERS](#) Spieler teilnehmen und [SIZE_DECKS](#) Decks verwendet werden.

```
#include <match.h>
```

Zusammengehörigkeiten von match:



Öffentliche Methoden

- `match ()`

Private Methoden

- `void buildMasterDeck ()`
- `void getCardsFromDeck (deck tDeck)`
- `void dealCards ()`

Private Attribute

- `std::vector< CARD > cards`
- `deck decks [SIZE_DECKS]`
- `player players [PLAYERS]`

4.3.1 Ausführliche Beschreibung

Die Klasse `match` generiert eine Runde eines Kartenspiels an welchem `PLAYERS` Spieler teilnehmen und `SIZE_DECKS` Decks verwendet werden.

Dabei werden an alle Teilnehmer gleich viele Karten aus den zuvor erstellten Decks zufällig verteilt.

Definiert in Zeile 18 der Datei `match.h`.

4.3.2 Beschreibung der Konstruktoren und Destruktoren

4.3.2.1 `match::match ()`

4.3.3 Dokumentation der Elementfunktionen

4.3.3.1 `void match::buildMasterDeck () [private]`

4.3.3.2 `void match::dealCards () [private]`

4.3.3.3 `void match::getCardsFromDeck (deck tDeck) [private]`

4.3.4 Dokumentation der Datenelemente

4.3.4.1 `std::vector<CARD> match::cards [private]`

Definiert in Zeile 27 der Datei `match.h`.

4.3.4.2 `deck match::decks[SIZE_DECKS] [private]`

Definiert in Zeile 29 der Datei `match.h`.

4.3.4.3 `player match::players[PLAYERS] [private]`

Definiert in Zeile 30 der Datei `match.h`.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [match.h](#)

4.4 `player` Klassenreferenz

Die Klasse `player` stellt einen Teilernehmer des Kartenspiels `match` dar.

```
#include <player.h>
```

Zusammengehörigkeiten von `player`:

player
- cards
+ player() + getCard() + printCards()

Öffentliche Methoden

- `player ()`
 `player::player`
- `void getCard (CARD card)`

- [player::getCard](#) deals a card to the player
- void [printCards](#) ()
 - [player::printCards](#) listet alle Handkarten des Spielers auf.

Private Attribute

- std::vector< [CARD](#) > [cards](#)

4.4.1 Ausführliche Beschreibung

Die Klasse [player](#) stellt einen Teilnehmer des Kartenspiels [match](#) dar.

Der Spieler kann Karten annehmen und seine Handkarten printen.

Definiert in Zeile 21 der Datei player.h.

4.4.2 Beschreibung der Konstruktoren und Destruktoren

4.4.2.1 [player::player](#) ()

[player::player](#)

Definiert in Zeile 6 der Datei player.cpp.

4.4.3 Dokumentation der Elementfunktionen

4.4.3.1 void [player::getCard](#) ([CARD](#) [card](#))

[player::getCard](#) deals a card to the player

Parameter

card	which is dealt to the player
----------------------	------------------------------

Definiert in Zeile 14 der Datei player.cpp.

Benutzt cards.

4.4.3.2 void [player::printCards](#) ()

[player::printCards](#) listet alle Handkarten des Spielers auf.

Definiert in Zeile 22 der Datei player.cpp.

Benutzt cards.

4.4.4 Dokumentation der Datenelemente

4.4.4.1 std::vector<[CARD](#)> [player::cards](#) [private]

Definiert in Zeile 29 der Datei player.h.

Wird benutzt von [getCard\(\)](#) und [printCards\(\)](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Dateien:

- [player.h](#)
- [player.cpp](#)

5 Datei-Dokumentation

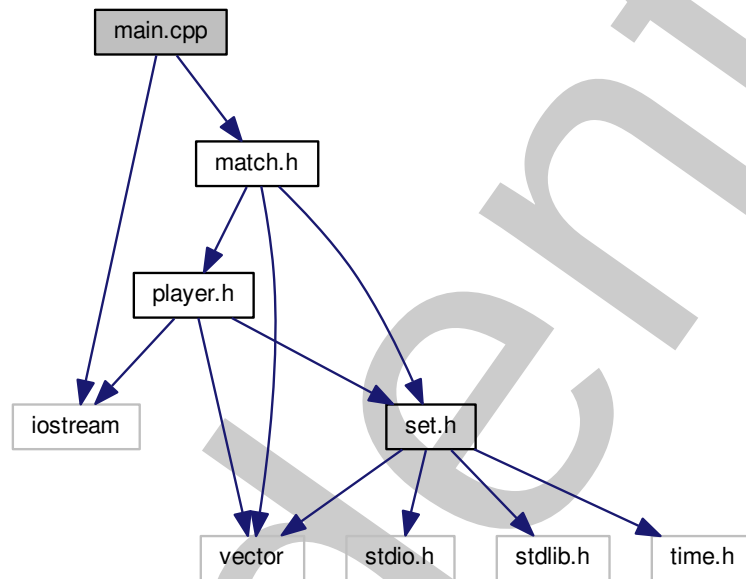
5.1 main.cpp-Dateireferenz

Initialise cards matches.

```
#include <iostream>
```

```
#include "match.h"
```

Include-Abhängigkeitsdiagramm für main.cpp:



Funktionen

- int [main](#) ()

5.1.1 Ausführliche Beschreibung

Initialise cards matches.

Autor

Andreas Conrads (aco@cadsoft.de)

Datum

Jan, 2016 Initialise and execute #GAMES card matches.

Definiert in Datei [main.cpp](#).

5.1.2 Dokumentation der Funktionen

5.1.2.1 int main ()

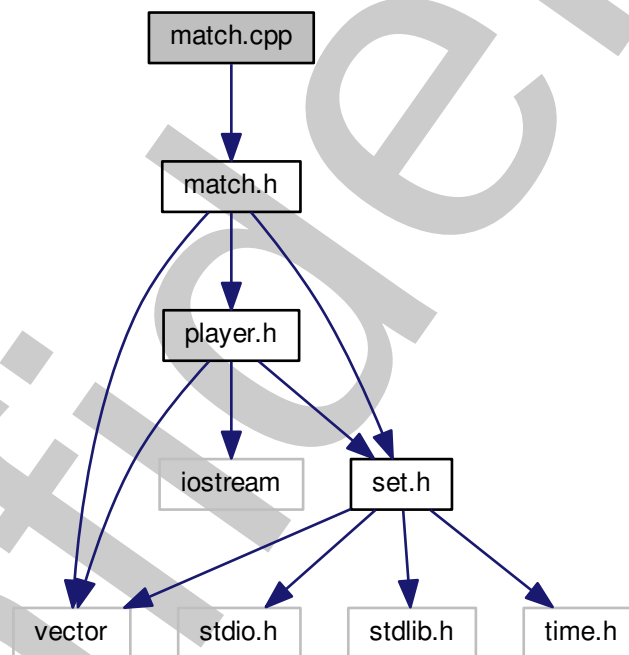
Definiert in Zeile 15 der Datei main.cpp.

5.2 mainpage.dox-Dateireferenz

5.3 match.cpp-Dateireferenz

```
#include "match.h"
```

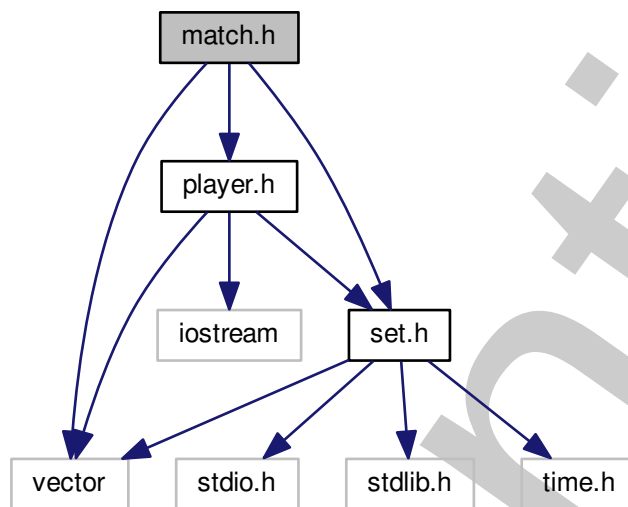
Include-Abhängigkeitsdiagramm für match.cpp:



5.4 match.h-Dateireferenz

```
#include <vector>
#include "set.h"
#include "player.h"
```

Include-Abhängigkeitsdiagramm für match.h:



Klassen

- class [match](#)

Die Klasse [match](#) generiert eine Rund eines Kartenspiels an welchem [PLAYERS](#) Spieler teilnehmen und [SIZE_DECKS](#) Decks verwendet werden.

Makrodefinitionen

- #define [SIZE_DECKS](#) 2
- #define [PLAYERS](#) 8

5.4.1 Makro-Dokumentation

5.4.1.1 #define PLAYERS 8

Definiert in Zeile 5 der Datei match.h.

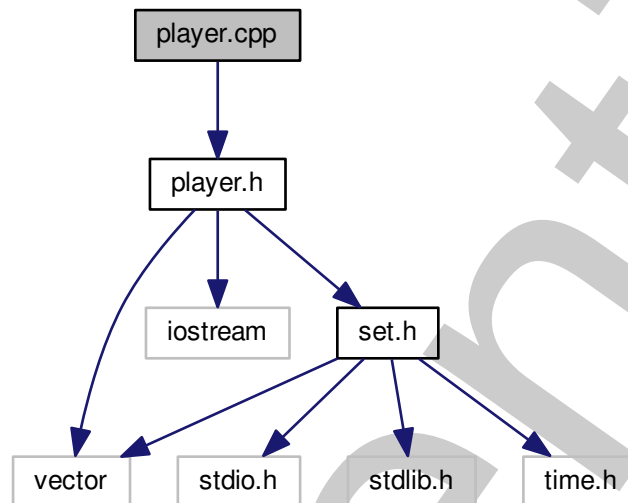
5.4.1.2 #define SIZE_DECKS 2

Definiert in Zeile 4 der Datei match.h.

5.5 player.cpp-Dateireferenz

```
#include "player.h"
```

Include-Abhängigkeitsdiagramm für player.cpp:

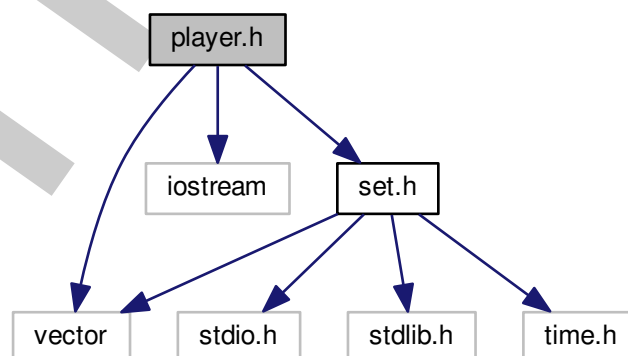


5.6 player.h-Dateireferenz

Datte zur Verwaltung der Klasse(n) [player](#).

```
#include <vector>
#include <iostream>
#include "set.h"
```

Include-Abhängigkeitsdiagramm für player.h:



Klassen

- class `player`

Die Klasse `player` stellt einen Teilnehmer des Kartenspiels `match` dar.

5.6.1 Ausführliche Beschreibung

Datien zur Verwaltung der Klasse(n) `player`.

Autor

Andreas Conrads

Datum

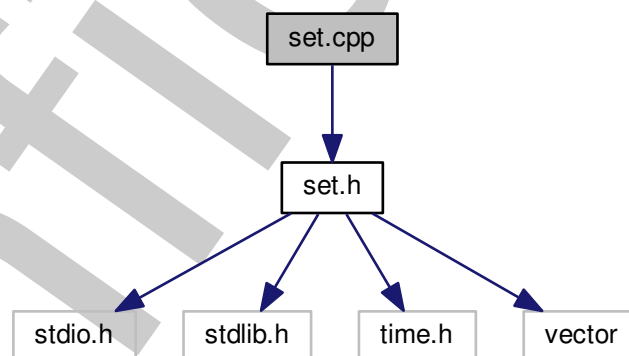
2016/01/08

Definiert in Datei `player.h`.

5.7 set.cpp-Dateireferenz

```
#include "set.h"
```

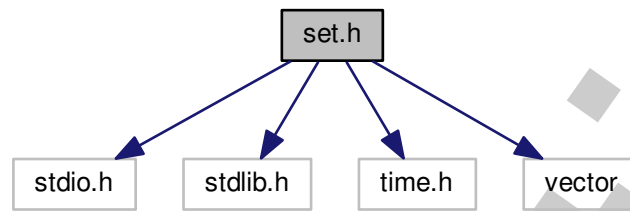
Include-Abhängigkeitsdiagramm für `set.cpp`:



5.8 set.h-Dateireferenz

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <vector>
```

Include-Abhängigkeitsdiagramm für set.h:



Klassen

- struct [CARD](#)
CARD ist ein Struct zur Darstellung von Karten.
- class [deck](#)
Die Klasse deck repräsentiert ein gewöhnliches Kartendeck.

Makrodefinitionen

- #define [SET_SIZE](#) 32
- #define [NUMBERS](#) 7
- #define [COLORS](#) 4

5.8.1 Makro-Dokumentation

5.8.1.1 #define COLORS 4

Definiert in Zeile 6 der Datei set.h.

Wird benutzt von `deck::deck()`.

5.8.1.2 #define NUMBERS 7

Definiert in Zeile 5 der Datei set.h.

Wird benutzt von `deck::deck()`.

5.8.1.3 #define SET_SIZE 32

Definiert in Zeile 4 der Datei set.h.

Index

buildMasterDeck
 match, 6

CARD, 2
 color, 2
 number, 2

COLORS
 set.h, 13

cards
 deck, 4
 match, 6
 player, 7

color
 CARD, 2

dealCards
 match, 6

deck, 2
 cards, 4
 deck, 3
 getCards, 4
 getRandomCard, 4
 isEmpty, 4
 remove, 4

decks
 match, 6

getCard
 player, 7

getCards
 deck, 4

getCardsFromDeck
 match, 6

getRandomCard
 deck, 4

isEmpty
 deck, 4

main
 main.cpp, 8

main.cpp, 8
 main, 8

mainpage.dox, 9
match, 4

 buildMasterDeck, 6
 cards, 6
 dealCards, 6
 decks, 6
 getCardsFromDeck, 6
 match, 6
 players, 6

match.cpp, 9

match.h, 9
 PLAYERS, 10
 SIZE_DECKS, 10

NUMBERS
 set.h, 13
number
 CARD, 2

PLAYERS
 match.h, 10

player, 6
 cards, 7
 getCard, 7
 player, 7
 printCards, 7

player.cpp, 11

player.h, 11

players
 match, 6
printCards
 player, 7

remove
 deck, 4

SET_SIZE
 set.h, 13

SIZE_DECKS
 match.h, 10

set.cpp, 12

set.h, 12

 COLORS, 13

 NUMBERS, 13

 SET_SIZE, 13