Assignment 3

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

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

Chapter 3

Class Documentation

3.1 BoardT Class Reference

BoardT abstract data type representing a gameboard of forty thieves.

`#include <GameBoard.h>`

Public Member Functions

- BoardT ()
  Empty constructor for BoardT.
- BoardT (std::vector< CardT > deck)
  Constructor for BoardT.
- bool is_valid_tab_mv (CategoryT c, nat n0, nat n1) const
  Checks if a valid move exists for moving a card from tableau to another category or tableau.
- bool is_valid_waste_mv (CategoryT c, nat n) const
  Checks if a valid move exists for moving a card from waste to another category or tableau.
- bool is_valid_deck_mv () const
  Checks if a valid move exists for moving a card from deck to waste.
- void tab_mv (CategoryT c, nat n0, nat n1)
  Moves a card from a tableau to another tableau or foundation.
- void waste_mv (CategoryT c, nat n)
  Moves a card from waste to another category or tableau.
- void deck_mv ()
  Moves a card from deck to waste.
- CardStackT get_tab (nat i) const
  Accessor - Gets the stack of cards from the tableau at index i.
- CardStackT get_foundation (nat i) const
  Accessor - Gets the stack of cards from the foundation at index i.
- CardStackT get_deck () const
  Accessor - Gets the deck.
- CardStackT get_waste () const
  Accessor - Gets the waste.
- bool valid_mv_exists () const
  Checks if a valid move exists.
- bool is_win_state () const
  Checks whether the game has been won.

3.1.1 Detailed Description

BoardT abstract data type representing a gameboard of forty thieves.

3.1.2 Member Function Documentation

3.1.2.1 get_deck()

```
CardStackT BoardT::get_deck ( ) const
```

Accessor - Gets the deck.

Returns

the stack of cards which is the deck

3.1.2.2 get_foundation()

```
CardStackT BoardT::get_foundation (
            nat i ) const
```

Accessor - Gets the stack of cards from the foundation at index i.

Parameters

i	- the index of the foundation we want to get

Returns

the stack of cards from the foundation at index i

3.1.2.3 get_tab()

```
CardStackT BoardT::get_tab (
            nat i ) const
```

Accessor - Gets the stack of cards from the tableau at index i.

Parameters

i	- the index of the tableau we want to get

Returns

a stack of cards from the tableau at index i

3.1.2.4 get_waste()

```
CardStackT BoardT::get_waste ( ) const
```

Accessor - Gets the waste.

Returns

the stack of cards which is the waste

3.1.2.5 is_valid_deck_mv()

```
bool BoardT::is_valid_deck_mv ( ) const
```

Checks if a valid move exists for moving a card from deck to waste.

Returns

boolean value representing whether a valid move from deck exists

3.1.2.6 is_valid_tab_mv()

```
bool BoardT::is_valid_tab_mv (
            CategoryT c,
            nat n0,
            nat n1 ) const
```

Checks if a valid move exists for moving a card from tableau to another category or tableau.

Parameters

c	- the category we want to move a card from tableau to
n0	- the index of the tableau we want to move from
n1	- the index of the category we want to move to

Returns

boolean value representing whether a valid move from tableau exists

3.1.2.7 is_valid_waste_mv()

```
bool BoardT::is_valid_waste_mv (
            CategoryT c,
            nat n ) const
```

Checks if a valid move exists for moving a card from waste to another category or tableau.

Parameters

c	- the category we want to move a card to
n	- the index of the category we want to move to

Returns

boolean value representing whether a valid move from waste exists

3.1.2.8 is_win_state()

```
bool BoardT::is_win_state ( ) const
```

Checks whether the game has been won.

Returns

a boolean value representing whether the game has been won

3.1.2.9 tab_mv()

```
void BoardT::tab_mv (
            CategoryT c,
            nat n0,
            nat n1 )
```

Moves a card from a tableau to another tableau or foundation.

Parameters

c	- the category we want to move a card from tableau to
n0	- the index of the tableau we want to move from
n1	- the index of the category we want to move to

3.1.2.10 valid_mv_exists()

```
bool BoardT::valid_mv_exists ( ) const
```

Checks if a valid move exists.

Returns

a boolean value representing whether a move exists

3.1.2.11 waste_mv()

```
void BoardT::waste_mv (
            CategoryT c,
            nat n )
```

Moves a card from waste to another category or tableau.

Parameters

c	- the category we want to move a card to
n	- the index of the category we want to move to

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

- include/GameBoard.h

3.2 CardT Struct Reference

Describes a card.

```
#include <CardTypes.h>
```

Public Attributes

- SuitT **s**
- RankT **r**

3.2.1 Detailed Description

Describes a card.

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

- include/CardTypes.h

3.3 Stack< T > Class Template Reference

Stack abstract data type.

`#include <Stack.h>`

Public Member Functions

- Stack ()
  Empty constructor for a stack.
- Stack (std::vector< T > s)
  Empty constructor for a stack.
- Stack push (T e)
  Mutator - push element into stack.
- Stack pop ()
  Mutator - pop element off stack.
- T top () const
  Accessor - gets top of stack.
- nat size () const
  Gets size of stack.
- std::vector< T > toSeq () const
  Accessor - get stack in form of sequence.

3.3.1 Detailed Description

template<class T>
class Stack< T >

Stack abstract data type.

3.3.2 Member Function Documentation

3.3.2.1 pop()

```
template<class T>
Stack Stack< T >::pop ( )
```

Mutator - pop element off stack.

Returns

Ouput a new stack

3.3.2.2 push()

```
template<class T>
Stack Stack< T >::push (
            T e )
```

Mutator - push element into stack.

Parameters

e	- Element to be pushed of type T

Returns

Ouput a new stack

3.3.2.3 size()

```
template<class T>
nat Stack< T >::size ( ) const
```

Gets size of stack.

Returns

Ouput size of stack

3.3.2.4 top()

```
template<class T>
T Stack< T >::top ( ) const
```

Accessor - gets top of stack.

Returns

Ouput element at top of stack

3.3.2.5 toSeq()

```
template<class T>
std::vector<T> Stack< T >::toSeq ( ) const
```

Accessor - get stack in form of sequence.

Returns

Stack in form of sequence

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

- include/Stack.h

Chapter 4

File Documentation

4.1 include/CardStack.h File Reference

```
#include "CardTypes.h"
#include "Stack.h"
```

Typedefs

- typedef Stack< CardT > CardStackT

  Describes a stack of cards.

4.1.1 Detailed Description

Author

Leon So | macid: sol4

Date

2019-03-26

4.2 include/CardTypes.h File Reference

Classes

- struct CardT

  Describes a card.

Macros

- #define ACE 1

  RankT for an Ace.
- #define JACK 11

  RankT for an Jack.
- #define QUEEN 12

  RankT for a Queen.
- #define KING 13

  RankT for a King.
- #define TOTAL_CARDS 104

  Total number of cards.

Typedefs

- typedef unsigned int nat

  Describes a natural number.
- typedef unsigned short int RankT

  Describes the rank of a card.

Enumerations

- enum CategoryT { **Tableau**, **Foundation**, **Deck**, **Waste** }

  Describes the category of a stack of cards.
- enum SuitT { **Heart**, **Diamond**, **Club**, **Spade** }

  Describes the suit of a card.

4.2.1 Detailed Description

Author

Leon So | macid: sol4

Date

2019-03-26

4.3 include/GameBoard.h File Reference

```
#include "CardTypes.h"
#include "CardStack.h"
#include <functional>
```

Classes

- class BoardT

  BoardT abstract data type representing a gameboard of forty thieves.

Typedefs

- typedef std::vector< CardStackT > SeqCrdStckT

4.3.1 Detailed Description

Author

Leon So | macid: sol4

Date

2019-03-26

4.3.2 Typedef Documentation

4.3.2.1 SeqCrdStckT

```
typedef std::vector<CardStackT> SeqCrdStckT
```

Describes a sequence of card stacks

4.4 include/Stack.h File Reference

```
#include <vector>
```

Classes

- class Stack< T >

  Stack abstract data type.

Typedefs

- typedef unsigned int nat

  Describes a natural number.

4.4.1 Detailed Description

Author

Leon So | macid: sol4

Date

2019-03-26

Index