# Rush Hour - Ernest Landrito

Generated by Doxygen 1.8.5

Mon Dec 9 2013 23:12:49

# Contents


**1 Class Index** **1**

1.1 Class List . . . . . 1

**2 Class Documentation** **3**

2.1 board Class Reference . . . . . 3

2.1.1 Constructor & Destructor Documentation . . . . . 3

2.1.1.1 board . . . . . 3

2.1.1.2 board . . . . . 4

2.1.2 Member Function Documentation . . . . . 4

2.1.2.1 canMoveHere . . . . . 4

2.1.2.2 getData . . . . . 4

2.1.2.3 getNumCars . . . . . 5

2.1.2.4 isSolved . . . . . 5

2.1.2.5 moveBackward . . . . . 6

2.1.2.6 moveForward . . . . . 6

2.1.2.7 operator!= . . . . . 6

2.1.2.8 operator< . . . . . 7

2.1.2.9 operator= . . . . . 7

2.1.2.10 operator== . . . . . 8

2.2 car Class Reference . . . . . 8

2.2.1 Member Function Documentation . . . . . 9

2.2.1.1 operator!= . . . . . 9

2.2.1.2 operator< . . . . . 9

2.2.1.3 operator== . . . . . 9

**Index** **11**

# Chapter 1

# Class Index

## 1.1 Class List

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

board . . . . . 3
car . . . . . 8

# Chapter 2

# Class Documentation

## 2.1 board Class Reference

### Public Member Functions

- board ()
- board (const board &other)
- board & operator= (const board &other)
- bool operator< (const board &other) const
- bool operator== (const board &other) const
- bool operator!= (const board &other) const
- void getData (int Cars)
- bool canMoveHere (int row, int col)
- bool isSolved () const
- bool moveForward (int carID)
- bool moveBackward (int carID)
- int getNumCars () const

### 2.1.1 Constructor & Destructor Documentation

#### 2.1.1.1 board::board ( )

**Precondition**

Unitialized board Class

**Postcondition**

numCars intialized

Algorithm:

- Assign data members values

Exceptional/Error Conditions:

- none

#### 2.1.1.2 board::board ( const board & *other* )

**Precondition**

Unitialized board Class

**Postcondition**

Grid Initialized numCars intialized

**Parameters**

| | |
|---|---|
| *other* | Other grid to be copied from |

Algorithm:

- copy data from other board

Exceptional/Error Conditions:

- none

### 2.1.2 Member Function Documentation

#### 2.1.2.1 bool board::canMoveHere ( int *row,* int *col* )

**Precondition**

Board Class

**Postcondition**

returned boolean

**Returns**

if a car can move to tht location

**Parameters**

| | |
|---|---|
| *row* | row location to check |
| *col* | column location to check |

Algorithm:

- go through each car and check their row and column
- check the row or column that the rest of the car hits

Exceptional/Error Conditions:

- none

#### 2.1.2.2 void board::getData ( int *Cars* )

**Precondition**

Board Class

**Postcondition**

data inserted into the board

**Parameters**

| | |
|---|---|
| *Cars* | number of cars to be inserted into the board |

Algorithm:

- assign the value of each car

Exceptional/Error Conditions:

- none

#### 2.1.2.3 int board::getNumCars ( ) const

**Precondition**

none

**Postcondition**

int returned

**Returns**

return number of cars in the board

Algorithm:

- return the number of cars

#### 2.1.2.4 bool board::isSolved ( ) const

**Precondition**

Board Class

**Postcondition**

returned boolean

**Returns**

if the main car is at the end of the row

Algorithm:

- check the column location of the main car

Exceptional/Error Conditions:

- none

#### 2.1.2.5 bool board::moveBackward ( int *carID* )

**Precondition**

none

**Postcondition**

Vehicle in the grid will be moved Left or down

**Returns**

return true if moved

**Parameters**

| | |
|---|---|
| *carID* | Algorithm:<br>• Check which direction to move<br>• Check if moving will go out of bounds<br>• check if can move<br>• move and return true |

#### 2.1.2.6 bool board::moveForward ( int *carID* )

**Precondition**

none

**Postcondition**

Vehicle in the grid will be moved right or Up

**Returns**

return true if moved

**Parameters**

| | |
|---|---|
| *carID* | Algorithm:<br>• Check which direction to move<br>• Check if moving will go out of bounds<br>• check if can move<br>• move and return true |

#### 2.1.2.7 bool board::operator!= ( const board & *other* ) const

**Precondition**

Board Class

**Postcondition**

boolean returned

**Returns**

Bolean if the board is not equivalent to the other

**Parameters**

| | |
|---|---|
| *other* | Other grid to be checked |

Algorithm:

- use the equivalent operator

Exceptional/Error Conditions:

- none

#### 2.1.2.8 bool board::operator< ( const board & *other* ) const

**Precondition**

Board Class

**Postcondition**

boolean returned

**Returns**

Bolean if the board is less than the other

**Parameters**

| | |
|---|---|
| *other* | Other grid to be check |

Algorithm:

- go through the grid and check if this car is less than the other car

Exceptional/Error Conditions:

- none

#### 2.1.2.9 board & board::operator= ( const board & *other* )

**Precondition**

Board Class

**Postcondition**

Grid Initialized numCars intialized

**Parameters**

| | |
|---|---|
| *other* | Other grid to be copied from |

Algorithm:

- copy data from other board

Exceptional/Error Conditions:

- none

#### 2.1.2.10 bool board::operator== ( const board & *other* ) const

**Precondition**

Board Class

**Postcondition**

boolean returned

**Returns**

Bolean if the board is equivalent to the other

**Parameters**

| | |
|---|---|
| *other* | Other grid to be checked |

Algorithm:

- go through the grid and check if each car is equivalent

Exceptional/Error Conditions:

- none

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

- rushHour.h

## 2.2 car Class Reference

### Public Member Functions

- bool operator== (const car &other) const
- bool operator!= (const car &other) const
- bool operator< (const car &other) const

### Public Attributes

- int **row**
- int **col**
- int **length**
- char **orientation**

### 2.2.1 Member Function Documentation

#### 2.2.1.1 bool car::operator!= ( const car & *other* ) const

**Precondition**

Two Car Classes

**Postcondition**

returns boolean if not equivalent

**Returns**

Returns true if one private data value of the class is not equal.

Algorithm:

- return not if equal

Exceptional/Error Conditions:

- none

#### 2.2.1.2 bool car::operator< ( const car & *other* ) const

**Precondition**

Two Car Classes

**Postcondition**

returns boolean if less than

**Returns**

True if the rhs is less than the left

Algorithm:

- check each value if one is less than starting with length

Exceptional/Error Conditions:

- none

#### 2.2.1.3 bool car::operator== ( const car & *other* ) const

**Precondition**

Two Car Classes

**Postcondition**

returns boolean if equivalent

**Returns**

Returns true if every private data value of the class is equal

Algorithm:

- return if equal

Exceptional/Error Conditions:

- none

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

- rushHour.h

# Index


board, 3
    board, 3
    canMoveHere, 4
    getData, 4
    getNumCars, 5
    isSolved, 5
    moveBackward, 5
    moveForward, 6
    operator<, 7
    operator=, 7
    operator==, 8

canMoveHere
    board, 4
car, 8
    operator<, 9
    operator==, 9

getData
    board, 4
getNumCars
    board, 5

isSolved
    board, 5

moveBackward
    board, 5
moveForward
    board, 6

operator<
    board, 7
    car, 9
operator=
    board, 7
operator==
    board, 8
    car, 9