

Lab 05

Generated by Doxygen 1.8.11

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	rushhour Class Reference	5
3.1.1	Constructor & Destructor Documentation	5
3.1.1.1	rushhour()	5
3.1.2	Member Function Documentation	6
3.1.2.1	buildBoard()	6
3.1.2.2	didWeWin()	6
3.1.2.3	moveBackward(int)	7
3.1.2.4	moveForward(int)	7
3.1.2.5	printBoard()	7
3.1.2.6	printResults(int)	8
3.1.2.7	solveIt(int)	8
3.2	vehicles Struct Reference	9
3.2.1	Member Data Documentation	9
3.2.1.1	column	9
3.2.1.2	orientation	9
3.2.1.3	row	9
3.2.1.4	size	9
4	File Documentation	11
4.1	rushhour.cpp File Reference	11
4.1.1	Detailed Description	11
4.1.2	Function Documentation	11
4.1.2.1	main()	11
4.2	rushhour.h File Reference	12
4.2.1	Detailed Description	12
	Index	13

Chapter 1

Class Index

1.1 Class List

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

rushhour	5
vehicles	9

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

rushhour.cpp	A simulation of the puzzle game, Rush hour	11
rushhour.h	A simulation of the puzzle game, Rush hour	12

Chapter 3

Class Documentation

3.1 rushhour Class Reference

```
#include <rushhour.h>
```

Public Member Functions

- `rushhour ()`
default constructor for rushhour data type
- `int buildBoard ()`
a setup function for a user defined rush hour board
- `bool didWeWin ()`
a boolean function that checks to see if vehicle zero meets win conditions
- `bool moveForward (int)`
a boolean function that returns true if the vehicle can move forward, or false if it cannot
- `bool moveBackward (int)`
a boolean function that returns true if the vehicle can move backward, or false if it cannot
- `void solvelt (int)`
a void function that uses recursion to solve the amount of moves necessary to solve the board
- `void printResults (int)`
a void function that prints out the solution if isSolved is true, and an error if it is false
- `void printBoard ()`
a void function that prints the board; used only for testing

3.1.1 Constructor & Destructor Documentation

3.1.1.1 `rushhour::rushhour ()`

default constructor for rushhour data type

Precondition

none

Postcondition

creates an empty rushhour

Parameters

<i>none</i>	
-------------	--

Returns

none

3.1.2 Member Function Documentation

3.1.2.1 int rushhour::buildBoard ()

a setup function for a user defined rush hour board

Precondition

none

Postcondition

initializes rushhour data type with user defined board

Parameters

<i>none</i>	
-------------	--

Returns

none

3.1.2.2 bool rushhour::didWeWin ()

a boolean function that checks to see if vehicle zero meets win conditions

Precondition

none

Postcondition

returns true or false depending on condition

Parameters

<i>none</i>	
-------------	--

Returns

true or false

3.1.2.3 bool rushhour::moveBackward (int *carNumber*)

a boolean function that returns true if the vehicle can move backward, or false if it cannot

Precondition

none

Postcondition

moves passed in vehicle number, one space backward, either vertical or horizontal

Parameters

<i>int</i>	carNumber, used to specify which vehicle to move
------------	--

Returns

true or false

3.1.2.4 bool rushhour::moveForward (int *carNumber*)

a boolean function that returns true if the vehicle can move forward, or false if it cannot

Precondition

none

Postcondition

moves passed in vehicle number, one space forward, either vertical or horizontal

Parameters

<i>int</i>	carNumber, used to specify which vehicle to move
------------	--

Returns

true or false

3.1.2.5 void rushhour::printBoard ()

a void function that prints the board; used only for testing

Precondition

none

Postcondition

none

Parameters

<i>none</i>	
-------------	--

Returns

none

3.1.2.6 void rushhour::printResults (int numVal)

a void function that prints out the solution if isSolved is true, and an error if it is false

Precondition

none

Postcondition

none

Parameters

<i>int</i>	numVal, used to print out the amount of moves necessary to complete the board
------------	---

Returns

none

3.1.2.7 void rushhour::solveIt (int numVal)

a void function that uses recursion to solve the amount of moves necessary to solve the board

Precondition

[didWeWin\(\)](#), [moveForward\(\)](#), [moveBackward\(\)](#)

Postcondition

updates the amount of moves necessary to solve the board, as well as the cap after a solution is found

Parameters

<i>int</i>	numVal, used to specify the amount of moves to solve the board
------------	--

Returns

none

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

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

3.2 vehicles Struct Reference

```
#include <rushhour.h>
```

Public Attributes

- int [size](#)
- int [row](#)
- int [column](#)
- char [orientation](#)

3.2.1 Member Data Documentation

3.2.1.1 int vehicles::column

3.2.1.2 char vehicles::orientation

3.2.1.3 int vehicles::row

3.2.1.4 int vehicles::size

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

- [rushhour.h](#)

Chapter 4

File Documentation

4.1 rushhour.cpp File Reference

A simulation of the puzzle game, Rush hour.

```
#include "rushhour.h"
```

Functions

- int `main` ()
the main driver that builds and runs the rushhour game

4.1.1 Detailed Description

A simulation of the puzzle game, Rush hour.

Author

Christopher Eichstedt

4.1.2 Function Documentation

4.1.2.1 int main ()

the main driver that builds and runs the rushhour game

Precondition

`rushhour()`, `solveIt()`, `buildBoard()`, `printResults()`

Postcondition

builds and solves the board for the allotted input, continues until 0 is given for numOfCars

Parameters

<i>none</i>	
-------------	--

Returns

returns 0 to end main driver

4.2 rushhour.h File Reference

A simulation of the puzzle game, Rush hour.

```
#include <iostream>
```

Classes

- struct [vehicles](#)
- class [rushhour](#)

4.2.1 Detailed Description

A simulation of the puzzle game, Rush hour.

Author

Christopher Eichstedt

Index

- buildBoard
 - [rushhour, 6](#)
- column
 - [vehicles, 9](#)
- didWeWin
 - [rushhour, 6](#)
- main
 - [rushhour.cpp, 11](#)
- moveBackward
 - [rushhour, 7](#)
- moveForward
 - [rushhour, 7](#)
- orientation
 - [vehicles, 9](#)
- printBoard
 - [rushhour, 7](#)
- printResults
 - [rushhour, 8](#)
- row
 - [vehicles, 9](#)
- [rushhour, 5](#)
 - [buildBoard, 6](#)
 - [didWeWin, 6](#)
 - [moveBackward, 7](#)
 - [moveForward, 7](#)
 - [printBoard, 7](#)
 - [printResults, 8](#)
 - [rushhour, 5](#)
 - [solvelt, 8](#)
- [rushhour.cpp, 11](#)
 - [main, 11](#)
- [rushhour.h, 12](#)
- size
 - [vehicles, 9](#)
- solvelt
 - [rushhour, 8](#)
- vehicles, [9](#)
 - [column, 9](#)
 - [orientation, 9](#)
 - [row, 9](#)
 - [size, 9](#)