My Project

Generated by Doxygen 1.8.11

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 files with brief descriptions:

Chapter 3

Class Documentation

3.1 car Class Reference

Public Member Functions

- car ()
  This is the constructor for the ADT, sets all values to NULL or 0.
- ∼car ()
  This is the constructor for the ADT.
- bool moveForward (int carVisual[ ][6])
  This is the function that moves a car ADT on the grid.
- bool moveBackward (int carVisual[ ][6])
  This is the function that moves a car ADT on the grid.
- bool didWeWin ()
  This function checks whether or not we have won the game.
- void setSize (const int newSize)
  sets the 'size' variable
- int getSize () const
  gets the 'size variable'
- void setOrientation (const char newOrientation)
  sets the 'orientation' variable
- char getOrientation () const
  gets the 'orientation' variable
- void setRow (const int newRow)
  sets the 'row' variable
- int getRow () const
  gets the 'row' variable
- void setColumn (const int newColumn)
  sets the 'column' variable
- int getColumn () const
  gets the column variable

3.1.1 Constructor & Destructor Documentation

3.1.1.1 car::car ( )

This is the constructor for the ADT, sets all values to NULL or 0.

3.1.1.2 car::∼car ( )

This is the constructor for the ADT.

3.1.2 Member Function Documentation

3.1.2.1 bool car::didWeWin ( )

This function checks whether or not we have won the game.

Returns

true or false

3.1.2.2 int car::getColumn ( ) const

gets the column variable

3.1.2.3 char car::getOrientation ( ) const

gets the 'orientation' variable

3.1.2.4 int car::getRow ( ) const

gets the 'row' variable

3.1.2.5 int car::getSize ( ) const

gets the 'size variable'

3.1.2.6 bool car::moveBackward (int *carVisual[ ][6]*)

This is the function that moves a car ADT on the grid.

Precondition

carVisual sent in, car does not exceed bounds

Postcondition

moved car or return

Returns

false or true based on whether the car can move

3.1.2.7 bool car::moveForward (int *carVisual[ ][6]*)

This is the function that moves a car ADT on the grid.

Precondition

carVisual sent in, car does not exceed bounds

Postcondition

moved car or return

Returns

false or true based on whether the car can move

3.1.2.8 void car::setColumn (const int *newColumn*)

sets the 'column' variable

Precondition

newColumn custom int input

Postcondition

column = newColumn

3.1.2.9 void car::setOrientation (const char *newOrientation*)

sets the 'orientation' variable

Precondition

newOrientation custom char input

Postcondition

orientation = newOrientation

3.1.2.10 void car::setRow (const int *newRow*)

sets the 'row' variable

Precondition

newRow custom int input

Postcondition

row = newRow

3.1.2.11 void car::setSize (const int *newSize*)

sets the 'size' variable

Precondition

newSize custom int input

Postcondition

size = newSize

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

- rushhour.cpp

Chapter 4

File Documentation

4.1 rushhour.cpp File Reference

```
#include <iostream>
```

Classes

- class car

Functions

- void solveIt (int moves, int carVisual[ ][6], int numCars, int &cap, int &best, car carsArray[ ])
  This is the recursive function to solve the number of moves for this problem.
- int main ()

4.1.1 Function Documentation

4.1.1.1 int main ( )

4.1.1.2 void solveIt (int *moves,* int *carVisual[ ][6],* int *numCars,* int & *cap,* int & *best,* car *carsArray[ ]*)

This is the recursive function to solve the number of moves for this problem.

Precondition

carVisual, moves, #cars, cap, best, and carsArray sent in

Postcondition

base case will determine stopping points

Returns

void

Index