PC-2: Rush Hour - Ernest Landrito

Generated by Doxygen 1.8.5

Tue Oct 15 2013 14:33:15

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 traffic Class Reference 5

3.1.1 Constructor & Destructor Documentation 5

3.1.1.1 traffic 5

3.1.2 Member Function Documentation 6

3.1.2.1 canMoveBack 6

3.1.2.2 canMoveForward 6

3.1.2.3 getData 6

3.1.2.4 moveBackward 7

3.1.2.5 moveForward 7

3.1.2.6 printMoves 8

3.1.2.7 recSolve 8

3.1.2.8 setDefault 8

3.2 vehicle Class Reference 9

3.2.1 Constructor & Destructor Documentation 9

3.2.1.1 vehicle 9

3.2.1.2 vehicle 9

3.2.2 Member Data Documentation 10

3.2.2.1 id 10

3.2.2.2 justMovedBackward 10

3.2.2.3 justMovedForward 10

3.2.2.4 leftX 10

3.2.2.5 orientation 10

3.2.2.6 size 10

3.2.2.7 topY 10

4 File Documentation **11**

4.1 rushHour.cpp File Reference 11

4.1.1 Function Documentation 11

4.1.1.1 main 11

4.2 rushHour.h File Reference 11

4.2.1 Variable Documentation 11

4.2.1.1 EMPTY_VAL 11

Index **12**

Chapter 1

Class Index

1.1 Class List

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

traffic 5
vehicle 9

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

rushHour.cpp 11
rushHour.h 11

Chapter 3

Class Documentation

3.1 traffic Class Reference

`#include <rushHour.h>`

Public Member Functions

- traffic ()
- void getData (int numCars)
- bool canMoveForward (int xPos, int yPos) const
- bool canMoveBack (int xPos, int yPos) const
- void recSolve (int movesSoFar)
- void moveBackward (int xPos, int yPos)
- void moveForward (int xPos, int yPos)
- void printMoves () const
- void setDefault (vehicle &other)

3.1.1 Constructor & Destructor Documentation

3.1.1.1 traffic::traffic ( )

Precondition

Unitialized Traffic Class

Postcondition

Grid Initialized numCars and numMoves intialized

Algorithm:

- Assign data members values

Exceptional/Error Conditions:

- none

3.1.2 Member Function Documentation

3.1.2.1 bool traffic::canMoveBack (int *xPos,* int *yPos*) const

Precondition

none

Postcondition

none

Parameters

xPos	The x position of the vehicle to be checked
yPos	The y position of the vehicle to be checked

Returns

Returns if the car can move left or up

Algorithm:

- Check if moving will move it out of bounds
- check if moving will cause a collision

3.1.2.2 bool traffic::canMoveForward (int *xPos,* int *yPos*) const

Precondition

none

Postcondition

none

Parameters

xPos	The x position of the vehicle to be checked
yPos	The y position of the vehicle to be checked

Returns

Returns if the car can move right or down

Algorithm:

- Check if moving will move it out of bounds
- check if moving will cause a collision

3.1.2.3 void traffic::getData (int *numCars*)

Precondition

Initialized Grid

Postcondition

Grid filled with vehicle data

Parameters

numCars	This is how many cars to put into the grid

Algorithm:

- Get number of cars
- Insert cars into the Grid based on the input values

3.1.2.4 void traffic::moveBackward (int *xPos*, int *yPos*)

Precondition

none

Postcondition

Vehicle in the grid will be moved Left or Up

Parameters

xPos	This is the x Position of the vehicle head to be moved
yPos	This is the y Position of the vehicle head to be moved

Algorithm:

- Change the head coordinates
- Assign new head coordinate the vehicle
- Set the tail of the vehicle to default values

3.1.2.5 void traffic::moveForward (int *xPos*, int *yPos*)

Precondition

none

Postcondition

Vehicle in the grid will be moved right or down

Parameters

xPos	This is the x Position of the vehicle head to be moved
yPos	This is the y Position of the vehicle head to be moved

Algorithm:

- Change the head coordinates
- Assign new head coordinate the vehicle
- Set the head of the vehicle to default values

3.1.2.6 void traffic::printMoves ( ) const

Precondition

none

Postcondition

printed moves

Algorithm:

- output numMoves

3.1.2.7 void traffic::recSolve (int *movesSoFar*)

Precondition

none

Postcondition

numMoves will now be the smallest moves possible

Parameters

movesSoFar	This is how many moves have been taken

Algorithm:

- Check if used too many moves
- Check if finished the puzzle
- If not, find any cars that can move
- Move the cars that can and repeat

3.1.2.8 void traffic::setDefault (vehicle & *other*)

Precondition

none

Postcondition

vehicle values changed to default

Parameters

other	This is the vehicle to be changed

Algorithm:

- assign vehicle data members default values

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

- rushHour.h

3.2 vehicle Class Reference

`#include <rushHour.h>`

Public Member Functions

- vehicle ()
- vehicle (const vehicle &other)

Public Attributes

- int id
- int size
- char orientation
- int leftX
- int topY
- bool justMovedForward
- bool justMovedBackward

3.2.1 Constructor & Destructor Documentation

3.2.1.1 vehicle::vehicle ( )

Precondition

unitialized Vehicle class

Postcondition

initialized data members

Algorithm:

- initialize data members

3.2.1.2 vehicle::vehicle (const vehicle & *other*)

Precondition

unitialized Vehicle class

Postcondition

copied vehicle class

Algorithm:

- initialize data members using others

3.2.2 Member Data Documentation

3.2.2.1 int vehicle::id

3.2.2.2 bool vehicle::justMovedBackward

3.2.2.3 bool vehicle::justMovedForward

3.2.2.4 int vehicle::leftX

3.2.2.5 char vehicle::orientation

3.2.2.6 int vehicle::size

3.2.2.7 int vehicle::topY

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

- rushHour.h

Chapter 4

File Documentation

4.1 rushHour.cpp File Reference

```
#include <iostream>
#include "rushHour.h"
```

Functions

- int main ()

4.1.1 Function Documentation

4.1.1.1 int main ( )

4.2 rushHour.h File Reference

```
#include <iostream>
```

Classes

- class vehicle
- class traffic

Variables

- const int EMPTY_VAL = -1

4.2.1 Variable Documentation

4.2.1.1 const int EMPTY_VAL = -1

Index

canMoveBack
    traffic, 6
canMoveForward
    traffic, 6

EMPTY_VAL
    rushHour.h, 11

getData
    traffic, 6

id
    vehicle, 10

justMovedBackward
    vehicle, 10
justMovedForward
    vehicle, 10

leftX
    vehicle, 10

main
    rushHour.cpp, 11
moveBackward
    traffic, 7
moveForward
    traffic, 7

orientation
    vehicle, 10

printMoves
    traffic, 7

recSolve
    traffic, 8
rushHour.cpp, 11
    main, 11
rushHour.h, 11
    EMPTY_VAL, 11

setDefault
    traffic, 8
size
    vehicle, 10

topY
    vehicle, 10
traffic, 5
    canMoveBack, 6
    canMoveForward, 6
    getData, 6
    moveBackward, 7
    moveForward, 7
    printMoves, 7
    recSolve, 8
    setDefault, 8
    traffic, 5

vehicle, 9
    id, 10
    justMovedBackward, 10
    justMovedForward, 10
    leftX, 10
    orientation, 10
    size, 10
    topY, 10
    vehicle, 9