Rush Hour v. STL

AUTHOR

Version 2.00

12/05/2014

Table of Contents

Rush Hour

The goal of Rush Hour is to unstick a car from a traffic game. The puzzle involves a 6x6 grid of squares. Vehicles are scattered over the grid at integer locations. The goal is to move your car out of the 6x6 grid and escape the traffic jam.

Author:

Saharath Kleips

Version:

2.00

Vehicles are always 1 square wide, but cars are always 2 squares long and trucks are 3 squares long. Vehicles are orientated either horizontally or vectically relative to the grid. Vehicles cannot move through each other, cannot turn, and cannot move off the edge of the grid. They may move forwards and backwards with respect to their orientation only if they are not blocked by another vehicle. Only one vehicle may move at a time and may only move one square at a time. Move vehicles until your car arrives at the rightmost edge of the grid, where it has escaped the traffic jam.

This program will solve any solvable Rush Hour puzzle and output the minimum possible number of (1 square at a time) moves.

Class Index

Class List

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

File Index

File List

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

Class Documentation

Board Struct Reference

Public Member Functions

Board ()

Default constructor for class Board.

bool isSolved ()

Determines whether the state of the grid is solved by checking the last column for any 0s.

bool moveForward (char)

Moves a vehicle forward on the Board corresponding to its ID.

bool moveBackward (char)

Moves a vehicle backwards on the Board corresponding to its ID.

bool canForward (char)

Checks if a vehicle can move backwards on the Board corresponding to its ID.

bool canBackward (char)

Checks if a vehicle can move backwards on the Board corresponding to its ID.

void printState ()

Prints a formatted representation of the state of the Board.

Public Attributes

int minSolution

The lowest possible number of moves to solve the puzzle.

int totalVehicles

The total amount of vehicles in this instance of the puzzle.

Vehicle state [6][6]

The current state of the board and all of its vehicles.

Detailed Description

Definition at line 50 of file RushHour.cpp.

Constructor & Destructor Documentation

Board::Board ()

Default constructor for class Board.

Definition at line 195 of file RushHour.cpp.

Member Function Documentation

bool Board::canBackward (char *cID*)

Checks if a vehicle can move backwards on the Board corresponding to its ID.

Parameters:

cID	is the ID of the vehicle on the Board.

True if the vehicle can move. False if the vehicle can not move.

Definition at line 458 of file RushHour.cpp.

bool Board::canForward (char *cID*)

Checks if a vehicle can move backwards on the Board corresponding to its ID.

Parameters:

Returns:

cID	is the ID of the vehicle on the Board.

True if the vehicle can move. False if the vehicle can not move.

Definition at line 411 of file RushHour.cpp.

bool Board::isSolved ()

Determines whether the state of the grid is solved by checking the last column for any 0s.

Returns:

True if the grid is solved, False if it is not.

Definition at line 506 of file RushHour.cpp.

bool Board::moveBackward (char *cID*)

Moves a vehicle backwards on the Board corresponding to its ID.

Postcondition:

The Board will be updated with the new positions of the vehicles.

Parameters:

Returns:

cID	is the ID of the vehicle on the Board.

True if the vehicle was moved. False if the vehicle was not moved.

Definition at line 346 of file RushHour.cpp.

bool Board::moveForward (char *cID*)

Moves a vehicle forward on the Board corresponding to its ID.

Postcondition:
The Board will be updated with the new positions of the vehicles.

Parameters:

Returns:

cID	is the ID of the vehicle on the Board.

True if the vehicle was moved. False if the vehicle was not moved.

Definition at line 284 of file RushHour.cpp.

void Board::printState ()

Prints a formatted representation of the state of the Board.

Parameters:

Returns:

state	is the current state of the puzzle to print out.

Member Data Documentation

int Board::minSolution

The lowest possible number of moves to solve the puzzle.

Definition at line 53 of file RushHour.cpp.

Vehicle Board::state[6][6]

The current state of the board and all of its vehicles.

Definition at line 57 of file RushHour.cpp.

int Board::totalVehicles

The total amount of vehicles in this instance of the puzzle.

Definition at line 55 of file RushHour.cpp.

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

1. RushHour.cpp

Vehicle Struct Reference

Public Attributes

char id

The character representation of this Vehicle.

char orientation

The orientation, either horizontal or vertical, of this Vehicle.

int length

The length of this Vehicle.

Detailed Description

Definition at line 40 of file RushHour.cpp.

Member Data Documentation

char Vehicle::id

The character representation of this Vehicle.

Definition at line 43 of file RushHour.cpp.

int Vehicle::length

The length of this Vehicle.

Definition at line 47 of file RushHour.cpp.

char Vehicle::orientation

The orientation, either horizontal or vertical, of this Vehicle.

Definition at line 45 of file RushHour.cpp.

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

2 RushHour.cpp

File Documentation

RushHour.cpp File Reference

```
#include <queue>
#include <map>
#include <iostream>
```

Classes

struct Vehicle
struct Board

Functions

Board loadPuzzle ()

Loads Board with a Rush Hour puzzle scenario from the console.

int solve (Board &board)

Solves the current state of the Board by implementing a breath-first search with previous state comparisons.

string boardToString (Board b)

Converts a Board to a String of IDs followed by orientations without character breaks for new rows or columns.

int main ()

Main function that controls user input, console output, and program loops.

Board stringToBoard (string s)

Converts a String to a Board with full IDs and Orientations, but does not contain lengths.

Variables

const bool _DEBUG_ = false

Debug console output and function tracing.

const bool _FULL_OUTPUT_ = false

Extended console output for puzzle diagram, moves used, etc.

Board **stringToBoard** (string)

Detailed Description

Definition in file RushHour.cpp.

Function Documentation

string boardToString (Board *b*)

Converts a Board to a String of IDs followed by orientations without character breaks for new rows or columns.

Parameters:

Definition at line 563 of file RushHour.cpp.

b	is the Board to convert to a String.

The converted Board.

Definition at line 522 of file RushHour.cpp.

Board loadPuzzle ()

Loads Board with a Rush Hour puzzle scenario from the console.

The first integer indicates the number of vehicles (n) in the scenario between 0 <= n <= 10. The next n lines represent 1 Vehicle where each line consists of a space separated list. Each list contains a number (2 or 3) indicating length, a letter (H or V) indicating orientation, and a row number (0-5) of the up most square if orientated vertically or the column number (0-5) of the left most square if orientated horizontally. The first Vehicle is treated as the escape Vehicle and must be orientated horizontally.

Precondition:

User input of a rush hour problem must be valid.

Postcondition:

Board is solvable and meets the criteria of the above description.

Returns:

Board is the Rush Hour scenario specified by the user.

The number of vehicles in the puzzle.

The length of the Vehicle.

The orientation of the Vehicle.

The starting row of the Vehicle.

The starting column of the Vehicle.

Temporary variable used to grab user input.

Definition at line 112 of file RushHour.cpp.

int solve (Board & *board*)

Solves the current state of the Board by implementing a breath-first search with previous state comparisons.

Postcondition:

minSolutions will be updated with the minimum amount of moves it takes to solve the puzzle.

Definition at line 214 of file RushHour.cpp.

Board stringToBoard (string *s*)

Converts a String to a Board with full IDs and Orientations, but does not contain lengths.

Parameters:

Returns:

s	is the String to convert to a Board.

The converted String.

Definition at line 543 of file RushHour.cpp.

Variable Documentation

const bool _DEBUG_ = false

Debug console output and function tracing.

Definition at line 33 of file RushHour.cpp.

const bool _FULL_OUTPUT_ = false

Extended console output for puzzle diagram, moves used, etc.

Definition at line 35 of file RushHour.cpp.

Index