

Package ‘BMLGrid’

May 11, 2015

Type Package

Title BMLGrid: package to simulate the Biham-Middleton-Levine Traffic Model

Version 1.1

Date 2015-05-08

Author Wenhao Wu <wnhwu@ucdavis.edu>

Maintainer Wenhao Wu <wnhwu@ucdavis.edu>

Description This version is meant to provide a comparison between the performance of BML simulation with R vectorized operation and c++ for loop.

URL https://bitbucket.org/shasqua/stat242_2015_assignment3/

License GPL (>=3)

LinkingTo Rcpp

Imports animation, Rcpp

Suggests testthat

NeedsCompilation yes

R topics documented:

| | |
|-----------------|---|
| BMLGrid-package | 2 |
| cidx_right | 2 |
| cidx_up | 3 |
| createBMLGrid | 3 |
| crunBMLGrid1 | 4 |
| crunBMLGrid2 | 4 |
| plot.BMLGrid | 5 |
| runBMLGrid | 5 |
| summary.BMLGrid | 6 |

| | |
|--------------|----------|
| Index | 7 |
|--------------|----------|

| | |
|-----------------|---|
| BMLGrid-package | <i>BMLGrid: a package to simulate the Biham-Middleton-Levine Traffic Model.</i> |
|-----------------|---|

Description

The BMLGrid provides a constructor function `createBMLGrid` for the S3 class BMLGrid, two S3 methods `plot.BMLGrid` and `summary.BMLGrid`. The workhorse function that simulate the moving process of BML model from a given initial step throughout a given number of steps are `runBMLGrid`, `crunBMLGrid1` (with key routines replaced by C++ program) and `crunBMLGrid2` (completely rewritten in C++).

Details

| | |
|----------|------------|
| Package: | BMLGrid |
| Type: | Package |
| Version: | 1.1 |
| Date: | 2015-05-08 |
| License: | GPL (>=3) |

Author(s)

Wenhao Wu <wnhhu@ucdavis.edu>

Maintainer: Wenhao Wu <wnhhu@ucdavis.edu>

References

http://eeyore.ucdavis.edu/stat242/Homeworks/BML_C.html

| | |
|------------|--|
| cidx_right | <i>Function to get the vector index of the grid right to the current grid.</i> |
|------------|--|

Description

c++ implementation of the `idx_right()` function

Usage

```
cidx_right(idx, r, c)
```

Arguments

| | |
|-----|--|
| idx | Current locations (vector index in the grid) of cars of a certain color. |
| r | numbers of rows |
| c | number of columns |

| | |
|---------|---|
| cidx_up | <i>Function to get the vector index of the grid above the current grid.</i> |
|---------|---|

Description

c++ implementation of the idx_up() fuction

Usage

```
cidx_up(idx, r)
```

Arguments

| | |
|-----|--|
| idx | Current locations (vector index in the grid) of cars of a certain color. |
| r | numbers of rows |

| | |
|---------------|---|
| createBMLGrid | <i>Constructor for S3 class BMLGrid</i> |
|---------------|---|

Description

Constructor for S3 class BMLGrid

Usage

```
createBMLGrid(r, c, ncars)
```

Arguments

| | |
|-------|--|
| r | A non-negative integer, the number of rows of the grid. |
| c | A non-negative integer, the number of columns of the grid. |
| ncars | A named vector of 2 non-negative integers where ncars['red'], ncars['blue'] represent the number of red/blue cars in the grid, respectively. |

Value

A BMLGrid class object which is essentially a matrix.

Examples

```
library(BMLGrid)
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100, blue = 100))
```

| | |
|--------------|---|
| crunBMLGrid1 | <i>Simulator for Biham-Middleton-Levine Traffic Model written in c++.</i> |
|--------------|---|

Description

The function that actually runs the Biham-Middleton-Levine Traffic Model from an initial state by a given number of steps.

Usage

```
crunBMLGrid1(g, numSteps, warningGridLock = FALSE)
```

Arguments

| | |
|------------------------------|--|
| <code>g</code> | A BMLGrid class object representing the initial state of the grid. |
| <code>numSteps</code> | Number of moves/periods. |
| <code>warningGridLock</code> | bool value indicating whether to prompt to Rcout when grid lock is detected. Default value is false. |

Examples

```
library(BMLGrid)
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100, blue = 100))
g.out = crunBMLGrid1(g, 10000)
plot(g.out)
```

| | |
|--------------|--|
| crunBMLGrid2 | <i>Simulator for Biham-Middleton-Levine Traffic Model, with key operations written in C++.</i> |
|--------------|--|

Description

The function that actually runs the Biham-Middleton-Levine Traffic Model from an initial state by a given number of steps.

Usage

```
crunBMLGrid2(g, numSteps)
```

Arguments

| | |
|-----------------------|--|
| <code>g</code> | A BMLGrid class object representing the initial state of the grid. |
| <code>numSteps</code> | Number of moves/periods. |

Value

a BMLGrid object representing the final state of the simulation.

Examples

```
library(BMLGrid)
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100, blue = 100))
g.out = crunBMLGrid2(g, numSteps = 10000)
plot(g.out)
```

plot.BMLGrid

plot method for BMLGrid class object

Description

Plot the cars on the grid as red/blue squares over a white background.

Usage

```
## S3 method for class 'BMLGrid'
plot(x, ...)
```

Arguments

x A BMLGrid class object.
... Other input arguments are simply ignored.

Examples

```
library(BMLGrid)
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100, blue = 100))
plot(g)
```

runBMLGrid

Simulator for Biham-Middleton-Levine Traffic Model.

Description

The function that actually runs the Biham-Middleton-Levine Traffic Model from an initial state by a given number of steps.

Usage

```
runBMLGrid(g, numSteps, movieName = NULL, recordSpeed = FALSE)
```

Arguments

| | |
|--------------------------|--|
| <code>g</code> | A BMLGrid class object representing the initial state of the grid. |
| <code>numSteps</code> | Number of moves/periods. |
| <code>movieName</code> | If specified as a non-NULL string, functions from package 'animation' will be used to record the BML process as a movie. |
| <code>recordSpeed</code> | The flag value indicating whether to record and return the average speed of the red and blue cars at each step. |

Value

If `recordSpeed` is unspecified or specified as `FALSE`, returns a BMLGrid object representing the final state of the simulation; otherwise return a list where the first element is the final-state grid object and the 2nd and 3rd elements record the average speed of red cars and blue cars, respectively.

Examples

```
library(BMLGrid)
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100, blue = 100))
g.out = runBMLGrid(g, numSteps = 10000)
plot(g.out)
g.out = runBMLGrid(g, numSteps = 50, movieName = 'movieBMLGrid', recordSpeed = TRUE)
plot(g.out$g)
summary(g.out$v.blue)
summary(g.out$v.red)
```

| | |
|-----------------|--|
| summary.BMLGrid | <i>summary method for BMLGrid class object</i> |
|-----------------|--|

Description

The summary includes information on the grid size and the number of red and blue cars in the grid.

Usage

```
## S3 method for class 'BMLGrid'
summary(object, ...)
```

Arguments

| | |
|---------------------|---|
| <code>object</code> | A BMLGrid class object. |
| <code>...</code> | Other input arguments are simply ignored. |

Examples

```
library(BMLGrid)
g = createBMLGrid(r = 100, c = 99, ncars = c(red = 100, blue = 100))
summary(g)
```

Index

*Topic **package**

BMLGrid-package, [2](#)

BMLGrid (BMLGrid-package), [2](#)

BMLGrid-package, [2](#)

cidx_right, [2](#)

cidx_up, [3](#)

createBMLGrid, [2, 3](#)

crunBMLGrid1, [2, 4](#)

crunBMLGrid2, [2, 4](#)

plot.BMLGrid, [2, 5](#)

runBMLGrid, [2, 5](#)

summary.BMLGrid, [2, 6](#)