Package ‘BoardGames’

July 18, 2016

**Type** Package

**Title** Board Games and Tools for Building Board Games

**Version** 1.0.0

**Description** Tools for constructing board/grid based games, as well as readily available game(s) for your entertainment.

**Depends** R (>= 3.0.2)

**License** GPL (>= 2)

**LazyData** TRUE

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Derek Qiu [aut, cre]

**Maintainer** Derek Qiu <qiu.derek.d@gmail.com>

**Repository** CRAN

**Date/Publication** 2016-07-18 20:37:26

**R topics documented:**

- detect_seq .................................................. 2
- get_cols .................................................... 2
- get_diags ................................................... 3
- get_rows .................................................... 3
- get_surround ............................................... 4
- index2xy .................................................... 4
- is_palindrome .............................................. 5
- UltimateTicTacToe ......................................... 5
- xy2index .................................................... 6

**Index** 7
detect_seq

Detects if a certain sequence is present in a matrix.

Description

This function allows for the detection of a particular sequence in a matrix.

Usage

detect_seq(data, sequence, reps, diag = TRUE)

Arguments

data A matrix.
sequence The desired sequence to search for.
reps Number of repetitions of the sequence.
diag Do you want to search diagonals? Defaults to TRUE.

Examples

M = matrix(sample(c(1,2),25,replace=TRUE),5,5)
detect_seq(data = M, sequence = "2", reps = 5)
#or equivalently
detect_seq(data = M, sequence = "22222", reps = 1)

get_cols

Get all column vectors of a matrix.

Description

This function extracts all column vectors of a matrix and returns the result as a list.

Usage

get_cols(data)

Arguments

data Matrix from which to extract column vectors.

Examples

M = matrix(rnorm(9),3,3)
get_cols(M)
get_diags

Get all diagonals vectors of a matrix.

Description

This function extracts all diagonal vectors of a matrix and returns the result as a list.

Usage

get_diags(data, direction = "right")

Arguments

data Matrix from which to extract diagonal elements
direction Which side to begin on? Takes values of one of "left", "right" or "both". Defaults to "right".

Examples

M = matrix(rnorm(9),3,3)
get_diags(M)

get_rows

Get all row vectors of a matrix.

Description

This function extracts all row vectors of a matrix and returns the result as a list.

Usage

get_rows(data)

Arguments

data Matrix from which to extract row vectors.

Examples

M = matrix(rnorm(9),3,3)
get_rows(M)
**get_surround**

*Get surrounding elements of an element in a matrix.*

**Description**

This function extracts all surrounding elements of a specified element in a matrix and returns the result as a vector.

**Usage**

`get_surround(data, index, type = "all")`

**Arguments**

- **data**
  - Matrix.
- **index**
  - Index position of element. Input as a vector of row then column positions.
- **type**
  - Takes values of "direct" and "all". "direct" returns only the elements directly in contact with the specified element, whereas "all" returns every surrounding element including diagonals. Defaults to "all".

**Examples**

```
M = matrix(1:20, 4, 5)
get_surround(data = M, index = c(2,3))
```

**index2xy**

*Converts a matrix index into a sex of x,y coordinates.*

**Description**

This function converts a matrix index into unit x,y plotting coordinates.

**Usage**

`index2xy(data, index)`

**Arguments**

- **data**
  - Matrix or data frame.
- **index**
  - A vector of index values.

**Examples**

```
M = matrix(1:20, 4, 5)
index2xy(data = M, index = c(3,4))
```
is_palindrome

Description
This function checks if the supplied vector is a palindrome (reads the same forwards and backwards).

Usage
is_palindrome(x, case.sensitive = FALSE)

Arguments
x Numeric or character vector.
case.sensitive Does upper or lower casing matter? Defaults to FALSE.

Examples
test1 = 123
test2 = "12321"
test3 = c("a",1,2,3,2,1,"a")
is_palindrome(test1)
is_palindrome(test2)
is_palindrome(test3)

UltimateTicTacToe

Description
This function allows one to play the Ultimate version of Tic-Tac-Toe. In the Regular version of Tic-Tac-Toe, players take turns placing their marks, with the objective of achieving three marks in a row in any direction. 9x9 Tic-Tac-Toe or more commonly known as Ultimate Tic-Tac-Toe, adds a twist on the regular version of Tic-Tac-Toe that most of us have come to know. Perceive the board as a big Tic-Tac-Toe board, with the goal being to achieve 3 big marks in any direction. Big marks are achieved by winning the corresponding small Tic-Tac-Toe blocks. The player to move first may play anywhere on the board. However, following moves must correspond to the same big Tic-Tac-Toe block of the small Tic-Tac-Toe board where the last move was played.

Usage
UltimateTicTacToe()
**Description**

This function converts a set of unit x,y coordinates into a matrix index.

**Usage**

```r
xy2index(data, x, y)
```

**Arguments**

- `data` : Matrix or data frame.
- `x` : x-coordinate
- `y` : y-coordinate

**Examples**

```r
M = matrix(1:20, 4, 5)
xy2index(data=M, x=3, y=2)
```
Index

+ Topic **case**
  is_palindrome, 5
+ Topic **check**
  is_palindrome, 5
+ Topic **column**
  get_cols, 2
+ Topic **connect**
  UltimateTicTacToe, 5
+ Topic **convert**
  index2xy, 4
  xy2index, 6
+ Topic **coordinate**
  index2xy, 4
  xy2index, 6
+ Topic **dataframe**
  UltimateTicTacToe, 5
+ Topic **detect**
  detect_seq, 2
+ Topic **diagonal**
  get_diags, 3
+ Topic **elements**
  get_surround, 4
+ Topic **element**
  index2xy, 4
  xy2index, 6
+ Topic **extract**
  get_cols, 2
  get_rows, 3
+ Topic **fun**
  UltimateTicTacToe, 5
+ Topic **game**
  UltimateTicTacToe, 5
+ Topic **index**
  index2xy, 4
  xy2index, 6
+ Topic **matrix**
  detect_seq, 2
  get_cols, 2
  get_diags, 3
  get_rows, 3
  get_surround, 4
  index2xy, 4
  UltimateTicTacToe, 5
  xy2index, 6
+ Topic **palindrome**
  is_palindrome, 5
+ Topic **row**
  get_rows, 3
  UltimateTicTacToe, 5
+ Topic **search**
  detect_seq, 2
+ Topic **sensitive**
  is_palindrome, 5
+ Topic **sequence**
  detect_seq, 2
+ Topic **surrounding**
  get_surround, 4
+ Topic **tac**
  UltimateTicTacToe, 5
+ Topic **tictactoe**
  UltimateTicTacToe, 5
+ Topic **tic**
  UltimateTicTacToe, 5
+ Topic **toe**
  UltimateTicTacToe, 5
+ Topic **ultimate**
  UltimateTicTacToe, 5
+ Topic **vectors**
  get_cols, 2
  get_diags, 3
  get_rows, 3
+ Topic **vector**
  detect_seq, 2
  get_surround, 4
  is_palindrome, 5
  detect_seq, 2
  get_cols, 2
get_diags, 3
get_rows, 3
get_surround, 4
index2xy, 4
is_palindrome, 5
UltimateTicTacToe, 5
xy2index, 6