Package ‘whitechapelR’

October 2, 2018

Title Advanced Policing Techniques for the Board Game ‘Letters from Whitechapel’

Version 0.3.0

Description Provides a set of functions to make tracking the hidden movements of the ‘Jack’ player easier. By tracking every possible path Jack might have traveled from the point of the initial murder including special movement such as through alleyways and via carriages, the police can more accurately narrow the field of their search. Additionally, by tracking all possible hideouts from round to round, rounds 3 and 4 should have a vastly reduced field of search.

Depends R (>= 3.3)
License MIT + file LICENSE
Encoding UTF-8
LazyData true
Imports plyr, igraph
Suggests covr, testthat
RoxygenNote 6.0.1
NeedsCompilation no
Author Mark Ewing [aut, cre]
Maintainer Mark Ewing <b.mark@ewingsonline.com>
Repository CRAN
Date/Publication 2018-10-02 17:40:03 UTC

R topics documented:

alley ................................................................. 2
end_round .......................................................... 2
inspect_space ....................................................... 3
node_locations .................................................... 4
roads ............................................................... 4
show_board ........................................................ 5
start_round ......................................................... 6
end_round

```
take_a_carriage ........................................... 6
take_a_step ............................................... 7
trim_possibilities ......................................... 7
```

Index 9

| alley | Undirected edge pairing of alley connecting nodes |

**Description**

Data used to establish possible connections used by Jack between nodes via alleyways.

**Usage**

```
alley
```

**Format**

A data frame with 452 rows and 2 variables

- x  The smaller integer of the pair of vertices
- y  The larger integer of the pair of vertices

---

end_round

```
Manage list of possible hideouts
```

**Description**

Create or update a list of possible hideouts based on final positions from the list of possible paths traveled.

**Usage**

```
end_round(paths, hideouts = NULL)
```

**Arguments**

- paths  list of all possible paths already traveled
- hideouts  optional vector of possible hideouts from previous rounds. Not used in round 1, only rounds 2 and 3

**Value**

list of all possible hideouts
Example

```r
possibilities = start_round(6T)
possibilities = take_a_step(possibilities, roads)
possibilities = take_a_step(possibilities, roads, blocked=list(c(63L8R), c(63L6U)))
possibilities = inspect_space(possibilities, space = c(29, 30), clue = FALSE)
possibilities = inspect_space(possibilities, space = 49, clue = TRUE)
hideouts = end_round(possibilities, hideouts=NULL)
possibilities = start_round(67)
possibilities = take_a_step(possibilities, roads)
hideouts = end_round(possibilities, hideouts=hideouts)
```

**inspect_space**

*Update paths based on inspections*

**Description**

Updated the list of possible paths based on the results of police investigation

**Usage**

`inspect_space(paths, space, clue)`

**Arguments**

- `paths` list of all possible paths already traveled
- `space` vector of integers of the spaces inspected
- `clue` single logical value indicating if evidence of Jack was found

**Value**

list of all possible paths traveled by Jack

**Examples**

```r
possibilities = start_round(64)
possibilities = take_a_step(possibilities, roads)
possibilities = take_a_step(possibilities, roads, blocked=list(c(63L82), c(63L65)))
possibilities = inspect_space(possibilities, space = c(29, 30), clue = FALSE)
possibilities = inspect_space(possibilities, space = 49, clue = TRUE)
```
node_locations

x, y coordinates of node points from the game board

Description

Data used to place nodes in graphical output according to their relative positions on the game board

Usage

node_locations

Format

A data frame with 195 rows and 4 variables

id  An artifact of the computer vision process used to obtain coordinates
x  The number of pixels from the left edge of the board to the center of the node
y  The number of pixels from the top edge of the board to the center of the node
name  The integer assigned to the node on the game board

roads

Undirected edge pairing of roads connecting nodes

Description

Data used to establish possible connections used by Jack between nodes

Usage

roads

Format

A data frame with 767 rows and 2 variables

x  The smaller integer of the pair of vertices
y  The larger integer of the pair of vertices
**show_board**

*Display game board representation*

**Description**

Show a graph representation of the game board with nodes placed in the appropriate relative spot, colored by the number of paths which include a particular node. Possible hideouts are marked with blue squares.

**Usage**

```r
show_board(paths = NULL, hideouts = NULL, roads, alley, node_locations)
```

**Arguments**

- **paths**: optional list of all possible paths already traveled
- **hideouts**: optional vector of possible hideouts from previous rounds.
- **roads**: data.frame of non-directional edge pairs for the road graph
- **alley**: data.frame of non-directional edge pairs for the alley graph
- **node_locations**: data.frame of where nodes should be placed in the graph

**Details**

roads, alley and node_locations are all bundled with the package (e.g. data(roads)). Solid lines in the graph represent road connections between nodes. Dashed lines represent alley way connections.

**Value**

plotted igraph object

**Examples**

```r
possibilities = start_round(64)
possibilities = take_a_step(possibilities, roads)
possibilities = take_a_step(possibilities, roads, blocked=list(c(63,82), c(63,65)))
possibilities = take_a_step(possibilities, alley)
show_board(paths=possibilities, hideouts=NULL, roads, alley, node_locations)
```
### start_round

**Start a new round**

**Description**

Generate the initial list for a new round

**Usage**

```python
start_round(initial_murder)
```

**Arguments**

- `initial_murder`  integer  Space of the initial murder(s)

**Value**

list with the initial murder location(s) as the starting point(s)

**Examples**

```python
possibilities = start_round(64)
possibilities = start_round(128)
```

### take_a_carriage

**Track carriage movement**

**Description**

Track two steps of unknown movement by Jack, on roads

**Usage**

```python
take_a_carriage(paths)
```

**Arguments**

- `paths`  list of all possible paths already traveled

**Value**

list of all possible paths traveled by Jack

**Examples**

```python
possibilities = start_round(64)
possibilities = take_a_carriage(possibilities)
```
**take_a_step**

*Track one movement*

**Description**

Track one step of unknown movement by Jack, either on roads or through alleyways.

**Usage**

```r
take_a_step(paths, roads, blocked = NULL)
```

**Arguments**

- `paths`: list of all possible paths already traveled
- `roads`: data.frame of non-directional edge pairs for either the road graph or the alley graph
- `blocked`: list of node pairs which cannot be traversed because a police officer blocks it (should not be used for special movement)

**Details**

The non-directional edge pairs are available via `data(roads)` or `data(alley)`. This function does not account for the rule that Jack cannot travel through a road occupied by a police officer.

**Value**

list of all possible paths traveled by Jack

**Examples**

```r
possibilities = start_round(64)
possibilities = take_a_step(possibilities, roads)
possibilities = take_a_step(possibilities, roads, blocked=list(c(63L82), c(63L65)))
possibilities = take_a_step(possibilities, alley)
```

**trim_possibilities**

*Trim possible paths*

**Description**

Remove known impossible end points for Jack, typically as a result of having found, but not arrested Jack.

**Usage**

```r
trim_possibilities(paths, node)
```
trim_possibilities

Arguments

paths list of all possible paths already traveled
node vector of length 1 or 2 which specifies blocked nodes due to the presence of a policeman

Value

list of trimmed possible paths traveled by Jack

Examples

possibilities = start_round(6T)
possibilities = take_a_carriage(possibilities)
possibilities = trim_possibilities(possibilitiesL8R)
possibilities = trim_possibilities(possibilitiesLc(66L67))
Index

*Topic datasets
  alley, 2
  node_locations, 4
  roads, 4

alley, 2
end_round, 2
inspect_space, 3
node_locations, 4
roads, 4
show_board, 5
start_round, 6

take_a_carriage, 6
take_a_step, 7
trim_possibilities, 7