Package 'worrrd'

October 25, 2022

Type Package

Title Generate Wordsearch and Crossword Puzzles

Version 0.1.0

Description Generate wordsearch and crossword puzzles using custom lists of words (and clues). Make them easy or hard, and print them to solve offline with paper and pencil!

License MIT + file LICENSE

URL https://www.stochastic-squirrel.com/worrrd/, https://github.com/anthonypileggi/worrrd

BugReports https://github.com/anthonypileggi/worrrd/issues

Encoding UTF-8

RoxygenNote 7.1.2

Imports ggplot2, dplyr, tibble, magrittr, stringr, purrr, yaml, glue, ggtext, ggfittext, cowplot

Suggests magick, emoji, rvest, english, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Anthony Pileggi [aut, cre, cph], Shannon Pileggi [aut] (<https://orcid.org/0000-0002-7732-4164>)

Maintainer Anthony Pileggi <apileggi20@gmail.com>

Repository CRAN

Date/Publication 2022-10-25 17:30:02 UTC

R topics documented:

add_word 2
as_crossword 3
as_wordsearch 3
book 4

crossword 4
image_matrix 5
is_crossword 6
is_wordsearch 6
make_logo 7
max_word_size 7
plot.crossword 8
plot.wordsearch 8
prepare_words 9
print.crossword 10
print.wordsearch 10
printable 11
wordsearch 11
word_intersections 12
word_overlap 13

Index **14**

`add_word` *Add a word to a word matrix*

Description

Add a word to a word matrix

Usage

```
add_word(x, word = "finding", must_intersect = FALSE, shape_matrix = NULL)
```

Arguments

`x`	word matrix
`word`	the word to add (character/scalar)
`must_intersect`	force the added word to intersect with >1 word (logical/scalar)
`shape_matrix`	a binary matrix generated from a call to `image_matrix`

Value

word matrix with word added (if possible)

as_crossword *Assign an object to the 'crossword' class*

Description

Assign an object to the 'crossword' class

Usage

```
as_crossword(x)
```

Arguments

x	an object containing crossword data

Value

crossword object: a matrix reprepresentation of the crossword, with attributes: positions: tibble representation of crossword clues: tibble representation of clue start (i.e., clue number locations)

as_wordsearch *Assign an object to the 'wordsearch' class*

Description

Assign an object to the 'wordsearch' class

Usage

```
as_wordsearch(x)
```

Arguments

x	an object containing wordsearch data

Value

wordsearch object: a list with the following elements:

search: a matrix representation of the wordsearch with 'positions' attribute a tibble representation of the solution words: (character/vector) clues: (character/vector) solution: a matrix representation of the wordsearch solution with 'positions' attribute a tibble representation of the solution image: image for shaping wordsearch (NULL if not provided) shape_matrix: binary matrix representation of shape (NULL if no image)

book *Create a puzzle book*

Description

Create a puzzle book

Usage

```
book(
  input_file = system.file("book.yml", package = "worrrd"),
  output_file = "book.pdf",
  solutions = TRUE
)
```

Arguments

input_file	yaml file containing book details/contents
output_file	full path to output file (with .pdf extension)
solutions	include solutions (logical/scalar)

Value

full path to the created puzzle book

Examples

```
# Create demo book included with package
book(output_file = "demo.pdf")
unlink("demo.pdf")
```

crossword *Create a crossword puzzle*

Description

Create a crossword puzzle

Usage

```
crossword(words, clues, r = 50, c = 50, method = c("optimal", "random"))
```

Arguments

words	a vector of words (character/vector)
clues	a vector a clues (character/vector)
r	number of rows (numeric/scalar)
c	number of columns (numeric/scalar)
method	generate puzzle using 'optimal' or 'random' word order, where the optimal order will place words with the most overlap first

Value

crossword object

Examples

```
# Example 1 ----
words <- c("apple", "pear", "banana")
clues <- c("red fruit", "bartlett", "green then yellow")
x <- crossword(words, clues)
plot(x, solution = TRUE)

# Example 2 ---
dat <-
dplyr::tribble(
  ~word,   ~clue,
  "dog",   "Bark. Bark. Bark.",
  "cat",   "Purrr",
  "horse", "Neighhhhh",
  "frog",  "Ribbit Ribbit",
  "cow",   "Mooooooo",
  "fox",   "Nee Nee Nee (What does the ____ say?)",
  "sheep", "Bleat",
  "snake", "Hissss",
  "duck",  "Quack",
  "bird",  "Chirp"
)
ex2 <- crossword(words = dat$word, clues = dat$clue, r = 40, c = 40)
plot(ex2, solution = TRUE, clues = TRUE)
```

---

image_matrix *Convert an image to a 0/1 matrix*

Description

Convert an image to a 0/1 matrix

Usage

```
image_matrix(
 img = "https://upload.wikimedia.org/wikipedia/commons/9/96/Tux_Paint_banana.svg",
  rows = 10,
  columns = 10
)
```

Arguments

img	full path to image (character/scalar)
rows	number of rows (numeric/scalar)
columns	number of columns (numeric/scalar)

---

is_crossword *Check if an object is of the 'crossword' class*

Description

Check if an object is of the 'crossword' class

Usage

```
is_crossword(x)
```

Arguments

x	an R object to check

Value

logical/scalar

---

is_wordsearch *Check if an object is of the 'wordsearch' class*

Description

Check if an object is of the 'wordsearch' class

Usage

```
is_wordsearch(x)
```

Arguments

x	an R object to check

Value

logical/scalar

`make_logo` *Make the 'worrrd' logo*

Description

Make the 'worrrd' logo

Usage

```
make_logo()
```

`max_word_size` *Compute maximum word size, based on the current word matrix*

Description

Compute maximum word size, based on the current word matrix

Usage

```
max_word_size(x, shape_matrix = NULL)
```

Arguments

`x`	word_search matrix
`shape_matrix`	shape matrix (logical) of identical size to 'x'

plot.crossword *Plot a crossword puzzle*

Description

Plot a crossword puzzle

Usage

```
## S3 method for class 'crossword'
plot(
  x,
  solution = FALSE,
  clues = FALSE,
  title = "Crossword Puzzle",
  legend_size = 4,
  ...
)
```

Arguments

x	a crossword object (see crossword)
solution	show solution? (logical/scalar)
clues	show clues? (logical/scalar)
title	puzzle title (character/scalar)
legend_size	letter size of word list; set to NULL to auto-size (numeric/scalar)
...	additional printing args

Value

ggplot2 object

plot.wordsearch *Draw a wordsearch puzzle*

Description

Draw a wordsearch puzzle

Usage

```
## S3 method for class 'wordsearch'
plot(
  x,
  solution = FALSE,
  clues = TRUE,
  title = "",
  puzzle_size = NULL,
  legend_size = NULL,
  ...
)
```

Arguments

x	wordsearch object (class: wordsearch)
solution	show solution? (logical/scalar)
clues	show clues? (logical/scalar)
title	puzzle title (character/scalar)
puzzle_size	letter size of puzzle; ignore to auto-size (numeric/scalar)
legend_size	letter size of word list; set to NULL to auto-size (numeric/scalar)
...	additional plotting args

Value

ggplot object

---

prepare_words *Prepare a word(s)*

Description

Prepare a word(s)

Usage

```
prepare_words(x)
```

Arguments

x	word list (character/vector)

print.crossword *Print a crossword puzzle*

Description

Print a crossword puzzle

Usage

```
## S3 method for class 'crossword'
print(x, ...)
```

Arguments

x	a crossword object (see crossword)
...	additional printing args

Value

crossword object

print.wordsearch *Print details for a wordsearch puzzle*

Description

Print details for a wordsearch puzzle

Usage

```
## S3 method for class 'wordsearch'
print(x, ...)
```

Arguments

x	wordsearch object (class: wordsearch)
...	additional printing args

Value

wordsearch object

printable *Prepare a worrrd object for printing*

Description

Prepare a worrrd object for printing

Usage

```
printable(x, filename = "plot.pdf")
```

Arguments

x	ggplot object
filename	name of file

Value

filename of pdf puzzle

Examples

```
words <- c("dog", "cat", "horse", "frog", "cow", "fox")
ex1 <- wordsearch(words, r = 10, c = 10)
my_puzzle <- plot(ex1, solution = FALSE)
printable(my_puzzle, "my_wordsearch.pdf")
unlink("my_wordsearch.pdf")
```

wordsearch *Create a wordsearch puzzle*

Description

Create a wordsearch puzzle

Usage

```
wordsearch(
  words = c("finding", "needles", "inside", "haystacks"),
  clues = words,
  r = 10,
  c = 10,
  image = NULL
)
```

Arguments

words	a vector of hidden words (character/vector)
clues	a vector of word clues (optional; character/vector)
r	number of rows
c	number of columns
image	path to an image that the resulting grid should look like.NULL for no shape

Value

wordsearch object

Examples

```
# Example 1 ----
words <- c("dog", "cat", "horse", "frog", "cow", "fox")
ex1 <- wordsearch(words, r = 10, c = 10)
plot(ex1, solution = TRUE)

# Example 2 ----
clues <- c("Bark", "Meow", "Neigh", "Ribbit", "Moo", "Nee Nee Nee")
ex2 <- wordsearch(words = words, clues = clues)
plot(ex2, solution = TRUE, title = "Animal Sounds", legend_size = 4)

# Example 3 ----
math <- dplyr::tribble(
  ~problem,  ~solution,
  "2 + 2",   "four",
  "5 + 3",   "eight",
  "9 - 4",   "five",
  "1 + 0",   "one",
  "2 + 1",   "three",
  "5 + 5",   "ten",
  "6 - 6",   "zero"
 )
 ex3 <- wordsearch(words = math$solution, clues = math$problem)
 plot(ex3, solution = TRUE, title = "Math is Fun")
```

---

word_intersections *Get possible intersection points based on the current board and a provided word*

Description

Get possible intersection points based on the current board and a provided word

Usage

```
word_intersections(x, word = "needles")
```

Arguments

x	word matrix
word	the word to add (character/scalar)

Value

for each direction, a matrix of crossing-point counts

---

word_overlap *Compute overlap score for a vector of words*

Description

Compute overlap score for a vector of words

Usage

```
word_overlap(words)
```

Arguments

words	vector of words (character/vector)

Index

add_word, 2
as_crossword, 3
as_wordsearch, 3

book, 4

crossword, 4, *8*, *10*

image_matrix, *2*, 5
is_crossword, 6
is_wordsearch, 6

make_logo, 7
max_word_size, 7

plot.crossword, 8
plot.wordsearch, 8
prepare_words, 9
print.crossword, 10
print.wordsearch, 10
printable, 11

word_intersections, 12
word_overlap, 13
wordsearch, 11