

Package ‘conn4R’

November 30, 2024

Type Package
Title What the Package Does (Title Case)
Version 0.1.0
Author Who wrote it
Maintainer The package maintainer <yourself@somewhere.net>
Description More about what it does (maybe more than one line)
Use four spaces when indenting paragraphs within the Description.
License What license is it under?
Encoding UTF-8
LazyData true
RoxygenNote 7.3.2

Contents

crear_posicion_aleatoria	1
evaluar_posicion	2
hello	3
iniciar_partida	3
juego_terminado	4
jugadas_disponibles	4
turno_humano	5
visualizar_tablero	6
Index	7

crear_posicion_aleatoria
<i>Crear posición aleatoria</i>

Description

crea un tablero aleatorio de cierta profundidad

Usage

```
crear_posicion_aleatoria(profundidad = 10)
```

Arguments

profundidad número de jugadas que se simulan aleatoriamente

Value

Una lista con los siguientes elementos

- tablero - matriz 6 x 7 que representa el tablero generado
- turnoUltimo - turno del jugador que ha realizado la última jugada en el tablero generado

Examples

```
tablero <- crear_posicion_aleatoria(21)
visualizar_tablero(tablero$tablero)
tablero$turnoUltimo
sum(tablero$tablero == 1)
sum(tablero$tablero == 2)
```

evaluar_posicion

evaluación estática de una posición representada en tablero

Description

evalua una posición mediante criterios estáticos basados en el número de casillas conectadas del jugador en turno

Usage

```
evaluar_posicion(tablero, turno)
```

Arguments

turno qué jugador comienza pa partida: 1 para humano, 2 para IA.
profundidad profundidad de búsqueda del algoritmo minimax.

Examples

```
lo siguiente inicia una partida en la que el jugador humano es el primero en jugar
iniciar_partida(profundidad = 7, turno = 2)
```

```
para que sea la IA quien realice la primera jugada:
iniciar_partida(turno = 2)
```

hello	<i>Hello, World!</i>
-------	----------------------

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

iniciar_partida	<i>Iniciar una partida de conecta 4</i>
-----------------	---

Description

inicia una partida de conecta 4. Se especifica la profundidad de búsqueda y el jugador que inicia la partida

Usage

```
iniciar_partida(profundidad = 5, turno = 1)
```

Arguments

profundidad	profundidad de búsqueda del algoritmo minimax.
turno	qué jugador comienza pa partida: 1 para humano, 2 para IA.

Examples

```
lo siguiente inicia una partida en la que el jugador humano es el primero en jugar
iniciar_partida(profundidad = 7, turno = 2)
```

```
para que sea la IA quien realice la primera jugada:
iniciar_partida(turno = 2)
```

juego_terminado	<i>evaluaa fin del juego y resultado</i>
-----------------	--

Description

evalua una posición mediante criterios estáticos basados en el número de casillas conectadas del jugador en turno

Usage

```
juego_terminado(tablero)
```

Arguments

tablero	una matriz representado el estado del tablero de juego
---------	--

Details

si finalizado == TRUE, resultado contendrá "GANA HUMANO", "GANA IA" o "TABLAS". si finalizado = FALSE, resultado es NA

Value

devuelve una lista con el siguiente contenido finalizado: booleano que representa el estado del juego: TRUE finalizado \item resultado: resultado del juego

Examples

```
tablero <- crear_posicion_aleatoria(21)
juego_terminado(tablero$tablero)
```

jugadas_disponibles	<i>Jugadas disponibles</i>
---------------------	----------------------------

Description

dada una situacvión en el tablero de juego, devuelve las posibles jugadas existentes: columnas no completadas

Usage

```
jugadas_disponibles(tablero)
```

Arguments

tablero	matriz 6 x 7 que representa la situación del tablero de juego.
---------	--

Examples

```
tablero <- crear_posicion_aleatoria(20)
visualizar_tablero(tablero)
jugadas_disponibles(tablero)
```

turno_humano	<i>solicita una jugada al humano.</i>
--------------	---------------------------------------

Description

la función invita al jugador humano a introducir una jugada Si la jugada es legal representeala jugada del humano en el tablero mediante un 1 en la casilla que corresponda. La jugada de la IA será representada en el tablero mediante un 2.

Usage

```
turno_humano(tablero, jugada = NULL)
```

Arguments

tablero	matriz 6 x 7 que representa la situación del tablero de juego.
jugada	un entero del 1 al 7 correspondiente a cada una de las 7 columnas si jugada = NULL (valor por defecto), entonces invita al jugador a introducir su jugada

Details

Si la jugada es legal representa la jugada del humano en el tablero mediante un 1 en la casilla que corresponda a la jugada Si la jugada introducida es ilegal, vuelve a solicitar una jugada legal

Examples

```
tablero <- reiniciar_tablero()
tablero <- turno_humano(tablero, jugada = 4)
visualizar_tablero(tablero)
# tambien se puede hacer lo siguiente (no ejecutar):
# tablero <- reiniciar_tablero()
# tablero <- turno_humano(tablero)
# visualizar_tablero(tablero)
```

visualizar_tablero	<i>visualiza gráficamente la posición de un tablero</i>
--------------------	---

Description

representa gráficamente el estado de un "tablero"

Usage

```
visualizar_tablero(tablero)
```

Arguments

tablero matriz 6x7 que representa la posición de un tablero

Value

devuelve un objeto ggplot listo para representar gráficamente el tablero

Examples

```
tablero <- crear_posicion_aleatoria(21)
p <- visualizar_tablero(tablero$tablero)
print(p)
```

Index

crear_posicion_aleatoria, [1](#)

evaluar_posicion, [2](#)

hello, [3](#)

iniciar_partida, [3](#)

juego_terminado, [4](#)

jugadas_disponibles, [4](#)

turno_humano, [5](#)

visualizar_tablero, [6](#)