

Package ‘gamesNws’

January 2, 2012

Type Package

Title Playing games using a NWS Server

Version 0.5

Date 2009-10-05

Author Markus Schmidberger, Fabian Grandke

Maintainer Markus Schmidberger <schmidb@ibe.med.uni-muenchen.de>

Depends nws

Description This is a package with different card games (e.g. uno,poker, ...) and using a NWS Server as card table. You can play the games with your friends in the whole world. Just install a NWS Server at one machine, send the login data to your friends and start the game.

License GPL

LazyLoad yes

Repository CRAN

Repository/R-Forge/Project gamesnws

Repository/R-Forge/Revision 45

Date/Publication 2009-10-06 10:41:51

R topics documented:

computerPlayerUNO	2
createUnoGame	3
playUno	4
startUnoGame	5
watchUnoGame	6

Index	8
--------------	----------

computerPlayerUNO	<i>Computer Player for UNO</i>
-------------------	--------------------------------

Description

Very stupid - random - computer player for UNO

Usage

```
computerPlayerUNO(ws,  
                  hand,  
                  card_played)
```

Arguments

ws	workspace object, that is used in the whole program
hand	A vector of cards, which the player holds in his hand.
card_played	A string, which indicates the latest played card.

Details

This function will be called in the playUno() function. Please feel free to write your own computer player and assign the new function to playUno() with the parameter computerPlayerFunction='yourComputerPlayer'! The computerplayer does not have to say "UNO". The value can just be changed.

Value

A list of two named strings,

selected_card	A string for the card, the player selected from his hand and has to be removed from his hand.
played_card	A string for the card, the player plays. Especially for color-selection cards (rybg), a color-rybg string has to be returned. If no card is played "NO" is returned. Usual cards are in format "color-value".

Author(s)

Markus Schmidberger <schmidb@ibe.med.uni-muenchen.de>, Fabian Grandke

References

<http://gamesnws.r-forge.r-project.org/>

createUnoGame	<i>Creates new UNO-Game</i>
---------------	-----------------------------

Description

New UNO-Game into the given workspace

Usage

```
.createUnoGame( wsName,  
                ...)
```

Arguments

wsName	Name of the workspace
...	Usual netWorkSpace()-attributes

Details

Checks if workspace already exists. Declares workspacevariables.

Value

ws	the created workspace with all necessary variables for the UNO-Game
----	---

Author(s)

Markus Schmidberger <schmidb@ibe.med.uni-muenchen.de>, Fabian Grandke

Examples

```
## Not run:  
ws<-.createUnoGame("test", "138.245.80.17")  
startUnoGame(ws)  
  
## End(Not run)
```

 playUno

The players play-function

Description

The player calls the function and inputs all necessary values. The function rules the whole course of the UNO-game.

Usage

```
playUno( name,
         user = Sys.info()["user"],
         computerPlayer = FALSE,
         computerPlayerFunction = computerPlayerUNO, ...)
```

Arguments

name	Name of the workspace
user	playername
computerPlayer	a logical, if TRUE the player is controlled by computerplayer, else, human controlled
computerPlayerFunction	which computerplayer is used
...	Usual parameters to initialize a netWorkspace(). See http://nws-r.sourceforge.net/doc/nwsR-1.5.0.pdf (Page 25) for further information

Details

This function rules the whole game course. It asks the player for decisions and shows the relevant information. The function shows the last played card, what cards are on your hand, which players are involved and how many cards each player has. To control the game options, there are specific commands the player has to use:

say-uno: before playing the second-last card, player has to announce "UNO" get-info: shows some game relevant information color*-value**: the card that should be played (one of the player's hand-cards) NO: if the player isn't able or doesn't want to play color*: after playing a rybg-card the player has to name a color, that has to be played next

*one of the "colors"(red,yellow,blue,green,rybg) **one of the "card values"=(0:9,2+,4+,BREAK,BACK)

Author(s)

Markus Schmidberger <schmidb@ibe.med.uni-muenchen.de>, Fabian Grandke

References

<http://gamesnws.r-forge.r-project.org/>

Examples

```

## Not run:
playUno("test",user="tom",serverHost="localhost")

## End(Not run)
## Not run:
playUno("test2",user="paul",computerPlayer=TRUE)

## End(Not run)

```

startUnoGame

Creates workspace and starts the UNO-Game - For Admin

Description

The Admin creates a workspace, starts an UNO-Game and decides what game-conditions apply.

Usage

```

startUnoGame( wsName,
               cardsStart = 7,
               minPlayers = 2,
               maxPlayers = 10,
               log = 0,
               logfile = NULL,
               debug = FALSE,
               config = NULL,
               graphics = TRUE,
               ...)

```

Arguments

wsName	workspacename
cardsStart	the number of cards, each player gets at the beginning
minPlayers	the number of players, that are necessary to start a game
maxPlayers	the number of players, that maximal can join a game
log	number of log mode : 0 = no logging, !0=logging
logfile	the file, where the log should be saved to(not yet implemented)
debug	a logical, if TRUE, there are several debuginformation shown during the game
config	vector of four logicals, if not set, the program asks for the rules later
graphics	a logical, if TRUE, a graphic device is shown to the players
...	Usual parameters to initialize a netWorkspace(). See http://nws-r.sourceforge.net/doc/nwsR-1.5.0.pdf (Page 25) for further information

Details

After starting the game, with this function, the players could log in with `playUno()`. This function only creates and starts the game. It does not play.

Author(s)

Markus Schmidberger <schmidb@ibe.med.uni-muenchen.de>, Fabian Grandke

References

<http://gamesnws.r-forge.r-project.org/>

Examples

```
## Not run:
  startUnoGame("test", serverHost="localhost", config=c(T,T,T,T))

## End(Not run)
## Not run:
  startUnoGame("test", serverHost="localhost", cardsStart=13, minPlayers=4)

## End(Not run)
```

watchUnoGame	<i>logs the game</i>
--------------	----------------------

Description

writes a logfile of the game and returns the winner

Usage

```
.watchUnoGame( ws,
                ...,
                logfile = NULL)
```

Arguments

ws	Name of the workspace
...	no function yet, but in future some logging-specific attributes are available
logfile	the directory, where the log should be saved to(not yet implemented)

Details

This function will be called in a `playUnoMaster()`-function. The logfile-argument is not yet implemented, so actually the logfile is written in your workingdirectory. Different logging-modes will be possible.

Value

winner player that has won the UNO-Game

Author(s)

Markus Schmidberger <schmidb@ibe.med.uni-muenchen.de>, Fabian Grandke

Examples

```
## Not run:  
  if(log!=0){  
winner <- .watchUnoGame(ws, logfile=logfile)  
  }  
  
## End(Not run)
```

Index

`.createUnoGame (createUnoGame)`, 3

`.watchUnoGame (watchUnoGame)`, 6

`computerPlayerUNO`, 2

`createUnoGame`, 3

`playUno`, 4

`startUnoGame`, 5

`watchUnoGame`, 6