

Package ‘SetupGameR’

March 23, 2018

Title SetupGameR

Version 0.1.2

Description A Library for Running Games with Students. See my web-site <<https://sites.google.com/a/g.clemson.edu/ja-resources>>. or github <<https://github.com/Jadamso>>.

Depends R (>= 3.4.1)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Published 2018-03-23

Date 2018-03-23

URL <https://sites.google.com/a/g.clemson.edu/ja-resources>

RoxygenNote 6.0.1

Imports MiscUtils

Suggests shinyjs,
formattable

R topics documented:

class_cleaner	2
class_sample	3
create_class_fake	3
create_playergroups	4
ctimeLeft	4
etime_fun	5
exp_design	5
Game	6
GameInit	6
GameLength	7
GameStop	7
game_setup	8
IDSheet	8
idsheet_default	9
passwd_maker	9
player_maker	10

post_global	10
print_exists	11
random.payouts	11
savegame	12
shutdown	12
sign_formatter2	13
sspasswd	13
sspasswd_cmd	14
StartSurveyUI	15
stime_function	15
timeLeft	16
user_urldata_fun	16
utimeLeft	17
ViewPaymentUI	17

Index	18
--------------	-----------

class_cleaner	<i>Clean the Class</i>
---------------	------------------------

Description

Clean the Class

Usage

```
class_cleaner(hdir = "/home/shiny/DoubleAuction/Students/",
  classname = "ClassList_orig.csv", name_id = "Username",
  ID_id = "Student.ID", omits = NA)
```

Arguments

hdir	directory holding master list of class
classname	which class to include
name_id	class names
ID_id	class ids

Value

nothing

class_sample	<i>Format Class into Player Profiles</i>
--------------	--

Description

Format Class into Player Profiles

Usage

```
class_sample(seed = 33, ngame = 5, nstuds = 12,  
  hdir = "/home/shiny/DoubleAuction/Students/", classname = "ClassList.csv")
```

Arguments

hdir	directory holding master list of class
classname	which class to include #param cid which subset of players to include

Value

nothing

create_class_fake	<i>Create Fake Class</i>
-------------------	--------------------------

Description

Create Fake Class

Usage

```
create_class_fake(n = 20, len = 8, patt = "[a-z]", ID = 1:n,  
  gdir = NA)
```

Arguments

n	number of students
len	length of password
patt	password pattern
ID	student ids
gdir	directory to save student list

Value

Class or name of saved file

create_playergroups	<i>Create Fake Class</i>
---------------------	--------------------------

Description

Create Fake Class

Usage

```
create_playergroups(seed = 0, game_dir = NA, groupsize = 2, n = 20,
  len = 8, patt = "[a-z]", ID = 1:n)
```

Arguments

n	number of students
len	length of password
patt	password pattern
ID	student ids
gdir	directory to save student list

Value

Class or name of saved file

ctimeLeft	<i>Gate Keeper</i>
-----------	--------------------

Description

Gate Keeper

Gate Keeper

Usage

```
ctimeLeft(startTime = Sys.time(), etime, tfreq = 1000,
  ctime_title = "Minutes Remaining: ", TradePeriod = Sys.time(),
  savepath = paste0(getwd(), "/"), GameName = "Game_")
```

```
ctimeLeft0(startTime = Sys.time(), etime, tfreq = 1000,
  ctime_title = "Minutes Remaining: ")
```

Arguments

startTime	a time object of what time this game started
etime	a time object EndTime
tfreq	how often to update in milliseconds
ctime_title	what to print
TradePeriod	which trading period is this?
savepath	file to save to, normally tempdir() or getwd()
GameName	name of file

Value

Message of time left in trading session or Save and Exit

Message of time left in trading session or Message

 etime_fun

Game Length Function Helper

Description

Game Length Function Helper

Usage

```
etime_fun(H, M, S)
```

Arguments

H, M, S integers for hours,minutes,seconds

Value

Class or name of saved file

 exp_design

Make Experiment

Description

Make Experiment

Usage

```
exp_design(buyers = c(1, 2, 12), sellers = 12, classtime = 75,
  sessiontime = 10)
```

Arguments

buyers, sellers number of students

classtime how long is the class

sessiontime how long will the games take

Value

nothing

Game	<i>Initiate and Stop Game</i>
------	-------------------------------

Description

Initiate and Stop Game

Usage

```
Game(gdir)
```

Arguments

gdir	directory of game files
------	-------------------------

Value

nothing

GameInit	<i>Initiate Game</i>
----------	----------------------

Description

Initiate Game

Usage

```
GameInit(TradePeriod = Sys.time(), startTime = Sys.time(),
  SessionName = "Trial", etime = SetupGameR::etime_fun("00", "10", "00"),
  gdir = "/srv/shiny-server/DoubleAuction/Game/", param_file = paste0(gdir,
    "time_file.RData"), u_timer = 60000, efreq = 1000, nperiods = 20,
  npractice = 2, ...)
```

Arguments

TradePeriod	starting what trading period is the game?
startTime	when to start game
gdir	directory of game
u_timer, efreq	timing parameters
nperiods, npractice	lengths of periods
game_file	name of game to initiate

Value

string, Name of Game File

GameLength	<i>GameLength</i>
------------	-------------------

Description

GameLength

Usage

```
GameLength(etime = etime_fun("00", "10", "00"), u_timer = 60000,
  gdir = "/srv/shiny-server/DoubleAuction/Game/", param_file = paste0(gdir,
    "time_file.RData"), ...)
```

Arguments

etime	how long does the game endure?
gdir	directory of game
param_file	name of file with game parameters

Value

nothing

GameStop	<i>Stop A Game with a Pause</i>
----------	---------------------------------

Description

Stop A Game with a Pause

Usage

```
GameStop(game_file = paste0(gdir, "game_file.RData"), EXCEPT = c("gdir",
  "GameInit", "Game"), savepath = paste0(tempdir(), "/"),
  GameName = "Game_", RM = FALSE, SAVE = TRUE, period_rest = 5)
```

Arguments

game_file	which game file to load
EXCEPT	exceptions not to be removed
savepath	file to save to
GameName	name of file
RM	remove objects in session?
SAVE	save data
period_rest	seconds to pause after shutdown

Value

nothing

game_setup	<i>Make Market Structure</i>
------------	------------------------------

Description

Make Market Structure

Usage

```
game_setup(hdir = "~/DoubleAuction/Markets/", i, ngame = 5, ni = 12,
           gpass = FALSE)
```

Arguments

hdir	home directory
i	number of sellers
ngame	number of games
ni	number of buyers
gpass	create participants passwords in gamefile

Value

nothing

IDsheet	<i>Create Profile List from Classlist</i>
---------	---

Description

Create Profile List from Classlist

Usage

```
IDsheet(classlist, idsheetfun = idsheet_default, id = "ID", name = "Name",
        SIMPLIFY = FALSE, USE.NAMES = FALSE, ...)
```

Arguments

classlist	matrix or data.frame indicating participants
idsheetfun	which function to use
id	column name with unique identifies
name	column name with unique identifies
SIMPLIFY, USE.NAMES	passed to mapply
...	arguments passed to idsheetfun

Value

list of participants

idsheet_default	<i>Format ID and Name into Empty Profile</i>
-----------------	--

Description

Format ID and Name into Empty Profile

Usage

```
idsheet_default(id_col, name_col)
```

Arguments

id_col	Person ID
name_col	Person Name

Value

a list, Empty userProfile

passwd_maker	<i>Create Passwords for List of Players</i>
--------------	---

Description

Create Passwords for List of Players

Create Passwords for List of Players

Usage

```
passwd_maker(Participants = NA, passwd = "TrialAuction",
             gdir = "/etc/shiny-server/", pfile = "passwd")

passwd_maker0(n = 20, user_append = "JA", passwd = "TrialAuction",
             create = FALSE, sys = FALSE, gdir = "/etc/shiny-server/",
             pfile = "passwd", add_admin = TRUE)
```

Arguments

Participants	student participants
passwd	what password
gdir	location of host Game folder
pfile	name of file to hold the passwords
n	number of student participants
create	overwrite and create new passwd_file [CURRENTLY done with new admin]
sys	if TRUE execute the bash script
add_admin	also reset admin password [NOT YET WORKING]
passwd	what password
gdir	location of host Game folder
pfile	name of file to hold the passwords

Value

a string to be executed by system()

a string to be executed by system()

player_maker	<i>Format Class into Player Profiles</i>
--------------	--

Description

Format Class into Player Profiles

Usage

```
player_maker(gdir = NA, Class, IDsheetfun = IDsheet,
             idsheetfun = idsheet_default, id = "ID", name = "Name",
             appendw = "JA", SAVE = TRUE, ...)
```

Arguments

gdir	directory to save player_file: Market/Game
Class	which class to include
IDsheetfun	which function to use
idsheetfun	which function to use
id	column name with unique identifies
name	column name with unique identifies
appendw	what to append student ID's with
SAVE	save the participants

Value

Participants

post_global	<i>Update Global Environment</i>
-------------	----------------------------------

Description

Update Global Environment

Usage

```
post_global(...)
```

Arguments

...	objects to post to the global environment
-----	---

Value

Nothing

print_exists	<i>Prints Objects to Main Screen</i>
--------------	--------------------------------------

Description

Prints Objects to Main Screen

Usage

```
print_exists(objname, fill = NA)

print_message(session)

print_proposals()

print_accepts()
```

Arguments

session session from server.R

Value

Nothing

random.payouts	<i>Payouts</i>
----------------	----------------

Description

Payouts

Usage

```
random.payouts(Participants, base = 0, in0 = 2,
               out0 = length(Participants[[1]]$History), randseed = NULL)
```

Arguments

base a base rate to pay them (showup fee)
in0 first period to sample from (ignore early histories)
out0 last period to sample from
participants List of Participants
pass initial seed

Value

payouts for each person

savegame

Save Game

Description

Save Game

Usage

```
savegame(startTime = format(Sys.time(), "%Y-%m-%d_%H-%M-%S"),
  TradePeriod = format(Sys.time(), "%Y-%m-%d_%H-%M-%S"),
  savepath = paste0(tempdir(), "/"), GameName = "Game_",
  PeriodName = "_Period_")
```

Arguments

startTime	what time did this game start at?
TradePeriod	which trading period is this?
savepath	file to save to
GameName, PeriodName	name of file

Details

see <https://shiny.rstudio.com/articles/persistent-data-storage.html> formerly used as `as.character(Sys.time())`, now `format(Sys.time(), "`

Value

string, Name of Game File

shutdown

Stop Game and Manage Output

Description

Stop Game and Manage Output

Usage

```
shutdown(SAVE = TRUE, RM = FALSE, EXCEPT = NULL, startTime = Sys.time(),
  TradePeriod = Sys.time(), savepath = paste0(tempdir(), "/"),
  GameName = "Game_")
```

Arguments

SAVE	save data
RM	remove objects in session?
EXCEPT	exceptions not to be removed
startTime	what time did this game start at?
TradePeriod	which trading period is this?
savepath	file to save to
GameName	name of file

Value

string, Name of Game File

sign_formatter2	<i>HTML formatting finance style</i>
-----------------	--------------------------------------

Description

HTML formatting finance style

Usage

```
sign_formatter2(x, data = NULL)
```

```
sign_formatter(x, data = NULL)
```

See Also

sign_formatter

sign_formatter2

sspasswd	<i>Create Password for Individual Player</i>
----------	--

Description

Create Password for Individual Player

Usage

```
sspasswd(user, passwd, create = FALSE, sys = TRUE,
  passfile = "/srv/shiny-server/passwd")
```

Arguments

user	who to create a password for
passwd	what password
create	overwrite and create new passwd_file
sys	if TRUE execute the bash script
passfile	what file holds the passwords

Value

a string to be executed by system()

sspasswd_cmd	<i>Create Passwords for List of Players</i>
--------------	---

Description

Create Passwords for List of Players

Usage

```
sspasswd_cmd(Participants, passwd, create = FALSE, sys = FALSE,  
             passfile = "/srv/shiny-server/passwd")
```

Arguments

Participants	list of participants
passwd	what password
create	overwrite and create new passwd_file
sys	if TRUE execute the bash script
passfile	what file holds the passwords

Value

a string to be executed by system()

StartSurveyUI	<i>Enter Survey Information</i>
---------------	---------------------------------

Description

Enter Survey Information

Usage

```
StartSurveyUI(id)
```

```
StartSurvey(input, output, session, GlobSurvey, SaveSurveyFile = getwd(),  
            userPID)
```

Arguments

...

stime_function	<i>Check if current time is in a list of times</i>
----------------	--

Description

Check if current time is in a list of times

Usage

```
stime_function(stimes, e = 0)
```

Arguments

stimes	list of times to check
e	degree of rounding

Value

TRUE/FALSE

timeLeft	<i>How much time left in trading session</i>
----------	--

Description

How much time left in trading session

Usage

```
timeLeft(startTime, etime, units = "mins")
```

Arguments

startTime	a time object
etime	a time object EndTime
units	what units to return

Value

numeric object showing time in minutes

user_urldata_fun	<i>Url Information About User</i>
------------------	-----------------------------------

Description

Url Information About User

Usage

```
user_urldata_fun(user_data)
```

Arguments

user_data	data from user
-----------	----------------

Value

string

utimeLeft	<i>How much time left in trading session</i>
-----------	--

Description

How much time left in trading session

Usage

```
utimeLeft(period_timer, tfreq = 1000, units = "secs")
```

Arguments

period_timer	a time object
tfreq	how often to update in milliseconds

Value

numeric object showing time in seconds

ViewPaymentUI	<i>View Payment Information</i>
---------------	---------------------------------

Description

View Payment Information

Usage

```
ViewPaymentUI(id)
```

```
ViewPayment(input, output, session, GlobSurvey, userPID)
```

Arguments

...

Index

[class_cleaner](#), [2](#)
[class_sample](#), [3](#)
[create_class_fake](#), [3](#)
[create_playergroups](#), [4](#)
[ctimeLeft](#), [4](#)
[ctimeLeft0 \(ctimeLeft\)](#), [4](#)

[etime_fun](#), [5](#)
[exp_design](#), [5](#)

[Game](#), [6](#)
[game_setup](#), [8](#)
[GameInit](#), [6](#)
[GameLength](#), [7](#)
[GameStop](#), [7](#)

[IDsheet](#), [8](#)
[idsheet_default](#), [9](#)

[passwd_maker](#), [9](#)
[passwd_maker0 \(passwd_maker\)](#), [9](#)
[player_maker](#), [10](#)
[post_global](#), [10](#)
[print_accepts \(print_exists\)](#), [11](#)
[print_exists](#), [11](#)
[print_message \(print_exists\)](#), [11](#)
[print_proposals \(print_exists\)](#), [11](#)

[random.payouts](#), [11](#)

[savegame](#), [12](#)
[shutdown](#), [12](#)
[sign_formatter \(sign_formatter2\)](#), [13](#)
[sign_formatter2](#), [13](#)
[sspasswd](#), [13](#)
[sspasswd_cmd](#), [14](#)
[StartSurvey \(StartSurveyUI\)](#), [15](#)
[StartSurveyUI](#), [15](#)
[stime_function](#), [15](#)

[timeLeft](#), [16](#)

[user_urldata_fun](#), [16](#)
[utimeLeft](#), [17](#)

[ViewPayment \(ViewPaymentUI\)](#), [17](#)
[ViewPaymentUI](#), [17](#)