

Package ‘rMouse’

June 22, 2017

Title Automate Mouse Clicks and Send Keyboard Input

Version 0.1

Description Provides wrapper functions to the Java Robot class to automate user input, like mouse movements, clicks and keyboard input.

Depends R (>= 3.4.0)

License CC0

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports rJava (>= 0.9-8)

Suggests knitr, roxygen2, rmarkdown

VignetteBuilder knitr

SystemRequirements Java >= 7

NeedsCompilation no

Author Dejan Prvulovic [aut, cre]

Maintainer Dejan Prvulovic <dejan.prv@gmail.com>

Repository CRAN

Date/Publication 2017-06-22 05:42:09 UTC

R topics documented:

coord	2
delay	2
left	3
move	4
pos	4
record	5
right	6
rMouse	7
setAutoDelay	8
specialKey	8
type	9

Index**11**

coord	<i>Get Coordinates of Mouse Cursor</i>
-------	--

Description

Get the coordinates of the current location of the mouse cursor.

Usage

```
coord()
```

Value

Returns a list with the coordinates.

See Also

[pos](#), [record](#)

Examples

```
## Not run:  
coord()  
coord()$x  
coord()$y  
  
## End(Not run)
```

delay	<i>Time Delay</i>
-------	-------------------

Description

Set the time delay in milliseconds manually.

Usage

```
delay(ms = 500, failSafe = TRUE)
```

Arguments

ms	(milliseconds). Default is 500.
failSafe	Logical flag to allow for emergency stops. Default is TRUE.

Note

If failSafe is TRUE, the function checks whether the mouse cursor is at the upper left position of the screen (pixel: x = 0, y = 0) . If so, the function fails and throws an error, allowing for emergency stops during code execution.

See Also

[setAutoDelay](#)

Examples

```
## Not run:  
delay(2000) # wait 2 seconds  
move(0,0)  
  
## End(Not run)
```

left	<i>Left Click</i>
------	-------------------

Description

Sends a left click, when called.

Usage

```
left(failSafe = TRUE)
```

Arguments

failSafe Logical flag to allow for emergency stops. Default is TRUE.

Note

If failSafe is TRUE, the function checks whether the mouse cursor is at the upper left position of the screen (pixel: x = 0, y = 0) . If so, the function fails and throws an error, allowing for emergency stops during code execution.

See Also

[delay](#), [move](#), [right](#), [pos](#), [record](#)

Examples

```
## Not run:  
move(50, 50)  
delay(1000)  
left()  
  
## End(Not run)
```

move *Move Mouse Cursor to Pixel(x,y)*

Description

Moves the mouse cursor to the specified pixel. Top left corner of screen is pixel(0,0).

Usage

```
move(x, y, failSafe = TRUE)
```

Arguments

x	horizontal coordinate as integer.
y	vertical coordinate as integer.
failSafe	Logical flag to allow for emergency stops. Default is TRUE.

Note

If failSafe is TRUE, the function checks whether the mouse cursor is at the upper left position of the screen (pixel: x = 0, y = 0) . If so, the function fails and throws an error, allowing for emergency stops during code execution.

See Also

[delay](#), [left](#), [right](#), [pos](#), [record](#)

Examples

```
## Not run:  
move(50, 50)  
delay(1000)  
move(364, 200)  
  
## End(Not run)
```

pos *Current position of mouse cursor*

Description

Print the position of the current location of the mouse cursor.

Usage

```
pos()
```

Value

Prints "move(x,y)" to the console.

Note

The console output can be pasted to a script, allowing to write mouse macros faster.

See Also

[coord](#), [record](#)

Examples

```
## Not run:  
pos()  
  
## End(Not run)
```

record	<i>Record mouse movements</i>
--------	-------------------------------

Description

Record subsequent mouse movements and print to console.

Usage

```
record(n = 10, timeInterval = 5, failSafe = TRUE)
```

Arguments

n	number of mouse moves to record. Default is 10.
timeInterval	(seconds). Record after how much seconds. Default is 5.
failSafe	Logical flag to allow for emergency stops. Default is TRUE.

Value

Prints "delay(2000); move(x,y); left()" to the console.

Note

If failSafe is TRUE, the function checks whether the mouse cursor is at the upper left position of the screen (pixel: x = 0, y = 0) . If so, the function fails and throws an error, allowing for emergency stops during code execution.

The record function calls the coord() function after the specified time interval passed and repeats n times. The console output can be pasted to a script to rapidly build a mouse macro. Unfortunately, this function is a work-around since it is not possible to record mouse events with the Java Robot Class. The console output is basically a template and can be adjusted to the needs of the user.

See Also

[coord](#), [pos](#)

Examples

```
## Not run:  
record()  
  
## End(Not run)
```

right

Right Click

Description

Sends a right click, when called.

Usage

```
right(failSafe = TRUE)
```

Arguments

`failSafe` Logical flag to allow for emergency stops. Default is TRUE.

Note

If `failSafe` is TRUE, the function checks whether the mouse cursor is at the upper left position of the screen (pixel: $x = 0$, $y = 0$). If so, the function fails and throws an error, allowing for emergency stops during code execution.

See Also

[delay](#), [move](#), [left](#), [pos](#), [record](#)

Examples

```
## Not run:  
move(50, 50)  
delay(1000)  
right()  
  
## End(Not run)
```

Description

rMouse is inspired by packages like `pyautogui` in Python. The goal of the package is to let users create simple mouse and keyboard macros to automate user input. The rMouse package utilizes the Java Robot Class and depends on the rJava package. Make sure the R version corresponds to the Java version, e.g. 64-bit R with 64-bit Java. Troubleshooting rJava can be done via its documentation or online.

Details

rMouse creates a Java Robot Object (`jRobot`) and an Mouse Info Object (`jMouseInformation`) upon loading. The functions in the package are wrapper functions to call the methods of the Java objects. The functions can be divided in three categories: delays, mouse input and keyboard input. Delay functions are `setAutoDelay(ms)` and `delay(ms)`, and work similar to Base R's `Sys.sleep(time)`. The delay functions specify the time interval between commands where the system waits. `move(x,y)`, `left()` and `right()` are mouse input functions to move the mouse cursor and send a left or a right click. With `coord()` and `pos()` it is possible to get the current position of the mouse cursor. `record()` is a work-around solution to record `n` mouse moves after a specified time interval has passed (e.g. every 5 seconds), as the Java Robot Class does not provide any methods to record mouse clicks. Moreover, it prints out an R code template to the console which can be used to rapidly build mouse macros. Finally, there are two functions to send keyboard input: `type(string)` allows to generate key presses for [0-9], [A-Z] and [a-z]. Spaces, periods and commas are allowed, however some special characters like "!" and "?" throw an error as the virtual key mapping in Java deviates from the generated raw bytes in R. `specialKey(key)` allows to send "ESC", "ENTER" or "TAB" key presses.

Each function has a build-in emergency stop procedure. If the mouse cursor is in the top left corner at pixel(`x = 0`, `y = 0`), then the execution is suspended. By setting the `failSafe` parameter of the function to `FALSE`, the emergency stop procedure is turned off.

Examples

```
## Not run:
setAutoDelay(ms) # automatic delay in milliseconds
delay(1000)      # wait 1000 miliseconds

move(0,0) # move to top left corner; pixel(0,0)
left()    # left click
right()   # right click
coord()   # return cursor coordinates as a list
pos()     # print cursor coordinates to console
record()  # record mouse moves and print template

type(string) # type a string; no special characters
specialKey(key) # send "ESC", "ENTER" or "TAB"
```

```
## End(Not run)
```

setAutoDelay	<i>Set automatic delay between commands</i>
--------------	---

Description

Set the time delay in milliseconds between commands.

Usage

```
setAutoDelay(ms = 100)
```

Arguments

ms (milliseconds). Default is 100.

See Also

[delay](#)

Examples

```
## Not run:
setAutoDelay()
setAutoDelay(1000) # 1 Second between commands
move(100, 100)
move(50, 50)

## End(Not run)
```

specialKey	<i>Send special key input</i>
------------	-------------------------------

Description

Send special keys like "ESC", "ENTER" or "TAB"

Usage

```
specialKey(key = "ESC", failSafe = TRUE)
```

Arguments

key "ESC", "ENTER" or "TAB" as string.
 failSafe Logical flag to allow for emergency stops. Default is TRUE.

Note

If failSafe is TRUE, the function checks whether the mouse cursor is at the upper left position of the screen (pixel: $x = 0$, $y = 0$) . If so, the function fails and throws an error, allowing for emergency stops during code execution.

See Also

[type](#), [delay](#)

Examples

```
## Not run:
specialKey("ENTER") # send ENTER
specialKey("ESC")   # send ESC

## End(Not run)
```

type	<i>Type keys</i>
------	------------------

Description

Type keyboard input by sending key press events.

Usage

```
type(string, failSafe = TRUE)
```

Arguments

string	String to be typed.
failSafe	Logical flag to allow for emergency stops. Default is TRUE.

Note

If failSafe is TRUE, the function checks whether the mouse cursor is at the upper left position of the screen (pixel: $x = 0$, $y = 0$) . If so, the function fails and throws an error, allowing for emergency stops during code execution.

Special characters like "?", "!", ";", "<", ">", "+", "*" are not allowed, since the mapping of the virtual keys in Java deviates from the R generated raw bytes.

See Also

[specialKey](#), [delay](#)

Examples

```
## Not run:  
type("This is a sentence. Hello.") # types to R console  
type("Hello World!")              # throws an error due to special character  
  
## End(Not run)
```

Index

- *Topic **coord**,
 - coord, 2
- *Topic **coordinates**
 - coord, 2
 - pos, 4
- *Topic **delay**
 - delay, 2
- *Topic **left**
 - left, 3
 - right, 6
- *Topic **move**
 - move, 4
- *Topic **position**,
 - pos, 4
- *Topic **record**,
 - record, 5
- *Topic **recording**
 - record, 5
- *Topic **setAutoDelay**
 - setAutoDelay, 8
- *Topic **type**
 - specialKey, 8
 - type, 9

coord, 2, 5, 6

delay, 2, 3, 4, 6, 8, 9

left, 3, 4, 6

move, 3, 4, 6

pos, 2-4, 4, 6

record, 2-5, 5, 6

right, 3, 4, 6

rMouse, 7

rMouse-package (rMouse), 7

setAutoDelay, 3, 8

specialKey, 8, 9

type, 9, 9