Package 'TurtleGraphics'

July 22, 2025

Version 1.0-8

Date 2018-02-13

Title Turtle Graphics

Suggests knitr, digest

VignetteBuilder knitr

Depends R (>= 3.0), grid

Description An implementation of turtle graphics
<http://en.wikipedia.org/wiki/Turtle_graphics>.
Turtle graphics comes from Papert's language Logo and has
been used to teach concepts of computer programming.

License GPL (>= 3)

URL http://www.gagolewski.com/software/TurtleGraphics/

BugReports https://github.com/gagolews/TurtleGraphics/issues

RoxygenNote 5.0.1

NeedsCompilation no

Author Anna Cena [aut],
Marek Gagolewski [aut],
Barbara Zogala-Siudem [aut, cre],
Marcin Kosinski [aut],
Natalia Potocka [aut]

Maintainer Barbara Zogala-Siudem <zogala@rexamine.com>

Repository CRAN

Date/Publication 2018-02-14 16:38:44 UTC

Contents

`TurtleGraphics-package`

Turtle Graphics in R

Description

Learn computer programming while having a jolly time

Details

Move the Turtle with commands that are relative to its own position, e.g., "move forward 100 pixels" or "turn right 30 degrees". From these building blocks you can build more complex shapes like circles, fractals, etc. Combined with R control flow, functions, and recursion, the idea of Turtle graphics allows students to get familiar with computer programming in a very accessible and pleasant way.

Author(s)

Anna Cena [aut]
Marek Gagolewski [aut]
Marcin Kosinski [aut]
Natalia Potocka [aut] Barbara Zogala-Siudem [aut, cre]

See Also

Other TurtleGraphics: `turtle_do`, `turtle_getpos`, `turtle_goto`, `turtle_init`, `turtle_move`, `turtle_param`, `turtle_reset`, `turtle_show`, `turtle_status`, `turtle_turn`, `turtle_up`

turtle_do *Evaluate a Larger Portion of Turtle Drawing Code*

Description

turtle_do evaluates an R expression with the Turtle temporarily hidden (for performance reasons).

Usage

```
turtle_do(expr)
```

Arguments

expr	expression to evaluate

Details

The terrarium must be initialized prior to using these functions, see turtle_init.

In order to decrease the evaluation time of expr, it is evaluated with Turtle temporarily hidden. Basically it means that if a Turtle image is visible (see turtle_show and turtle_hide) turtle_do removes it, evaluates expr and redraws it on the function exit.

See Also

Other TurtleGraphics: TurtleGraphics-package, turtle_getpos, turtle_goto, turtle_init, turtle_move, turtle_param, turtle_reset, turtle_show, turtle_status, turtle_turn, turtle_up

Examples

```
turtle_init()
turtle_do({
   for (i in 1:4) {
      turtle_forward(50)
      turtle_right(90)
   }
})
```

turtle_getpos *Get the Turtle's Current Position and Direction*

Description

`turtle_getpos` returns the Turtle's current position on the plane.

`turtle_getangle` returns the Turtle's current direction, in degrees. An angle of 0 represents a north-facing Turtle.

Usage

```
turtle_getpos()

turtle_getangle()
```

Details

The terrarium must be initialized prior to using these functions, see `turtle_init`.

Value

Both functions return a (named) numeric vector. `turtle_getpos` returns a vector of length two which specifies the `x` and `y` coordinates. The `turtle_getangle` returns the angle.

See Also

Other TurtleGraphics: `TurtleGraphics-package`, `turtle_do`, `turtle_goto`, `turtle_init`, `turtle_move`, `turtle_param`, `turtle_reset`, `turtle_show`, `turtle_status`, `turtle_turn`, `turtle_up`

Examples

```
turtle_init()
turtle_getpos()["x"] # x coordinate
turtle_getpos()["y"] # y coordinate
```

turtle_goto *Set the Turtle's Position and Direction*

Description

`turtle_goto` and `turtle_setpos` move the Turtle to a given location without changing its direction.

`turtle_setangle` rotates the Turtle to a given (absolute) angle, where 0 denotes a north-facing Turtle.

Usage

```
turtle_goto(x, y)

turtle_setpos(x, y)

turtle_setangle(angle)
```

Arguments

x, y	numeric; coordinates specifying new Turtle's location.
angle	numeric; rotation angle in degrees.

Details

The terrarium must be initialized prior to using these functions, see turtle_init.

If the given location (x, y) lies outside the terrarium, the behavior of these functions depends on the mode argument in turtle_init.

turtle_goto may draw the path between the current Turtle's position and the new location. Its behavior depends on the current plot settings, see turtle_up, turtle_down. In case of turtle_setpos, however, the path drawing is always disabled.

See Also

Other TurtleGraphics: TurtleGraphics-package, turtle_do, turtle_getpos, turtle_init, turtle_move, turtle_param, turtle_reset, turtle_show, turtle_status, turtle_turn, turtle_up

---

turtle_init *Set Up a New, Shiny Terrarium*

Description

This function creates a new empty plot with the Turtle centered on the board and facing to the north.

Usage

```
turtle_init(width = 100, height = 100, mode = c("error", "clip", "cycle"))
```

Arguments

width	numeric; plot width.
height	numeric; plot height.
mode	character string; one of "error", "clip", or "cycle".

Details

The mode argument determines what happens if the Turtle tries to move outside the terrarium. `clip` allows it to do that, but the drawing will be clipped to the predefined plot region. `error` throws an error. `cycle` makes the Turtle appear on the other side of the board.

After the `turtle_init()` function has been called you can e.g. move the Turtle with the `turtle_forward` function, turn its direction with `turtle_right` or set display parameters of the Turtle's trace, see `turtle_param`.

See Also

Other TurtleGraphics: `TurtleGraphics-package`, `turtle_do`, `turtle_getpos`, `turtle_goto`, `turtle_move`, `turtle_param`, `turtle_reset`, `turtle_show`, `turtle_status`, `turtle_turn`, `turtle_up`

---

turtle_move *Move the Turtle Forward or Backward*

Description

`turtle_forward` moves the Turtle in forward direction and `turtle_backward` moves the Turtle back.

Usage

```
turtle_move(distance, direction = c("forward", "backward"))

turtle_forward(distance)

turtle_backward(distance)
```

Arguments

`distance`	single numeric value; specifies the distance to make. Negative distance results in moving in the opposite direction.
`direction`	character string; moving direction. One of `"forward"` or `"backward"`.

Details

The Turtle must be initialized prior to using these functions, see `turtle_init`.

These functions make use of the Turtle's display options specified by the `turtle_param` function (or if not, use the default options set by `turtle_init`).

Note that if `turtle_up` or `turtle_down` was called, the Turtle's trace will be or not be drawn, respectively.

If you are willing to call these functions in an R loop, you may want to hide the Turtle temporarily (see `turtle_hide` and `turtle_do`) before making actual moves. This will increase the drawing performance significantly.

See Also

Other TurtleGraphics: TurtleGraphics-package, turtle_do, turtle_getpos, turtle_goto, turtle_init, turtle_param, turtle_reset, turtle_show, turtle_status, turtle_turn, turtle_up

Examples

```
turtle_init()
turtle_left(30)
turtle_forward(2)
turtle_up()
turtle_forward(1)
turtle_down()
turtle_right(60)
turtle_forward(9)
```

---

turtle_param *Set Display Options*

Description

Sets the display options for the Turtle's trace. It is possible to change its color, line type and line width.

Usage

```
turtle_param(col = NULL, lwd = NULL, lty = NULL)

turtle_col(col)

turtle_lwd(lwd)

turtle_lty(lty)
```

Arguments

col	numeric or character; trace color, see e.g. colors and gpar.
lwd	numeric; trace line width, see gpar.
lty	numeric; trace line type, see gpar.

Details

The Turtle must be initialized prior to using this function, see turtle_init.

See Also

Other TurtleGraphics: TurtleGraphics-package, turtle_do, turtle_getpos, turtle_goto, turtle_init, turtle_move, turtle_reset, turtle_show, turtle_status, turtle_turn, turtle_up

Examples

```
turtle_init()
turtle_forward(5)
turtle_up()
turtle_forward(3)
turtle_down()
turtle_left(90)
turtle_forward(5)
turtle_param(col = "red", lwd = 2, lty = 2)
turtle_forward(5)
```

`turtle_reset` *Reset the Turtle's Position and Direction*

Description

This function resets the Turtle's position, direction, and graphical options.

Usage

```
turtle_reset()
```

Details

The Turtle must be initialized prior to using this function, see `turtle_init`.

After a call to this function, the Turtle will be placed in the terrarium's center and it will be directed to the north.

The drawing remains unchanged.

See Also

Other TurtleGraphics: `TurtleGraphics-package`, `turtle_do`, `turtle_getpos`, `turtle_goto`, `turtle_init`, `turtle_move`, `turtle_param`, `turtle_show`, `turtle_status`, `turtle_turn`, `turtle_up`

Examples

```
turtle_init()
turtle_forward(4)
turtle_param(col="red", lty=2, lwd=3)
turtle_reset()
turtle_left(45)
turtle_forward(3)
```

turtle_show *Show or Hide the Turtle*

Description

These functions enable or disable displaying the Turtle's image on the screen.

Usage

```
turtle_show()

turtle_hide()
```

Details

The Turtle must be initialized prior to using this function, see `turtle_init`.

It is recommended to hide the Turtle when performing multiple Turtle moves, for efficiency reasons, see also `turtle_do`.

See Also

Other TurtleGraphics: `TurtleGraphics-package`, `turtle_do`, `turtle_getpos`, `turtle_goto`, `turtle_init`, `turtle_move`, `turtle_param`, `turtle_reset`, `turtle_status`, `turtle_turn`, `turtle_up`

Examples

```
turtle_init()
turtle_forward(4)
turtle_hide()
turtle_left(30)
turtle_forward(3)
```

turtle_status *Read the Turtle's Status*

Description

This function gives information about the current Turtle's position, direction, and on display options.

Usage

```
turtle_status()
```

Details

The Turtle must be initialized prior to using this function, see turtle_init.

Value

Returns a list with three elements.

See Also

Other TurtleGraphics: TurtleGraphics-package, turtle_do, turtle_getpos, turtle_goto, turtle_init, turtle_move, turtle_param, turtle_reset, turtle_show, turtle_turn, turtle_up

---

turtle_turn *Turn (Rotate) the Turtle*

Description

Turn the Turtle in the given direction by the given angle.

Usage

```
turtle_turn(angle, direction = c("left", "right"))

turtle_left(angle)

turtle_right(angle)
```

Arguments

angle	single numeric value; rotation angle in degrees. A negative value turns the Turtle in the opposite direction than the given one.
direction	character string; direction of the turn. Possible values are "left" and "right".

Details

The Turtle must be initialized prior to using this function, see turtle_init.

See Also

Other TurtleGraphics: TurtleGraphics-package, turtle_do, turtle_getpos, turtle_goto, turtle_init, turtle_move, turtle_param, turtle_reset, turtle_show, turtle_status, turtle_up

Examples

```
turtle_init()
turtle_left(30) # equivalent to turtle_turn(30, "left")
turtle_right(40)
turtle_turn(30, sample(c("left", "right"), 1)) # random turn
```

turtle_up *Turn on or off Turtle Trace Drawing*

Description

When the Turtle moves, it may or may not leave a visible trace. These functions control such a behavior.

Usage

```
turtle_up()

turtle_down()
```

Details

The Turtle must be initialized prior to using this function, see turtle_init.

See Also

Other TurtleGraphics: TurtleGraphics-package, turtle_do, turtle_getpos, turtle_goto, turtle_init, turtle_move, turtle_param, turtle_reset, turtle_show, turtle_status, turtle_turn

Index