Package ‘Snake’

January 20, 2025

Title Game of Snake

Version 1.0

Date 2020-10-08

Description Implements snake in R as a programming example, see <https://en.wikipedia.org/wiki/Snake_(video_game_genre)>.

Depends R (>= 3.5.0)

Imports tcltk, grid

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation no

Author Carsten Croonenbroeck [aut, cre]

Maintainer Carsten Croonenbroeck <carsten.croonenbroeck@uni-rostock.de>

Repository CRAN

Date/Publication 2020-10-24 09:50:08 UTC

Contents

PlaySnake *Starts a game of snake.*

Description

Opens a plot window device and starts a game of snake in it. Steer the snake with the arrow key, press Pause to pause and Esc to end the game.

Usage

```
PlaySnake(FIELDSIZE = 20, Speed = 0.8, Image = "Snake",
Interpolate = FALSE)
```

Arguments

FIELDSIZE	specify the field size here. Values must be greater or equal to 12 and are not limited. However, large fields are hardly fun to play.
Speed	enter values between 0 and 1. 1 is very fast, 0 is very slow movement of the snake.
Image	specifies which background image to use. Options are "Snake" (the default), "Sunset" for a beautify sunset image or anything else such as "White" for a white image (no background).
Interpolate	give TRUE for a bilinear interpolation of the graphics. Makes the background somewhat smoother, but also generates unsharpness, esp. of the snake itself.

Details

This serves as a programming example. R does not provide the R programmer with a system message loop. This is due to different operating systems (Windows, Linux, MacOS, each of them handles system messages very differently) and to the fact that for usual purposes as a statistical software, R programmers hardly require information from the system messaging loop. However, for snake, this is essential, as to asynchronously handle key presses. As R has no interface to the sytem message system, the trick is to open a window using package 'tcltk' and use its message loop interface. The tcltk subsystem pumps keypress events to the R workspace using a callback function, which itself is then used to write the desired values into a variable embedded into a distinct environment. The game can access that and use it to control the snake.

This system works well in plain R in windows (RGui and terminal). It also works under Linux (tested with Ubuntu). Snake is untested on MacOS. Snake will run in itegrated environments including RStudio, R Commander, and alike. Hint: You may want to call X11 before running snake for an extra window that provides more speed and accuracy. On some machines it is not possible to hide the tcltk window (mostly Linux, depending on the window manager used). However, this is only a visual impairment. Just make sure that the window titled "Snake control" is focused so that it receives key presses.

Value

Invisibly returns 0 if the game is terminated properly.

Author(s)

Carsten Croonenbroeck

Examples

```
if (interactive()) PlaySnake()
```

SnakeBG *Snake background image (snake).*

Description

Contains pixel data for a background image.

Usage

```
SnakeBG
```

Format

A matrix of 200 x 200 pixels, containing data as returned by `link{rgb}`.

Author(s)

Carsten Croonenbroeck

Source

Various web sources (lorem pixum).

`SunsetBG` *Snake background image (sunset).*

Description

Contains pixel data for a background image.

Usage

```
SnakeBG
```

Format

A matrix of 200 x 200 pixels, containing data as returned by `link{rgb}`.

Author(s)

Carsten Croonenbroeck

Source

Various web sources (lorem pixum).

Index