Package: nara (via r-universe)

September 18, 2024

Type Package

Title Native Raster Image Tools

Version 0.1.1.9024

Maintainer Mike Cheng <mikefc@coolbutuseless.com>

Description Tools for 'nativeRaster' images.

URL https://github.com/coolbutuseless/nara,
https://coolbutuseless.github.io/package/nara/index.html

BugReports https://github.com/coolbutuseless/nara/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 2.10)

Suggests testthat (>= 3.0.0), jpeg, png, purrr, stringr, knitr, rmarkdown, magick

Config/testthat/edition 3

Copyright The included 'deer' sprites are by Calciumtrice and licensed under CC BY 3.0. See 'inst/LICENSE-deer.txt' for more details. The included bitmap font 'spleen' was created by Frederic Cambus (c) 2018-2024. See 'inst/LICENSE-spleen.txt' for full LICENSE

VignetteBuilder knitr

Repository https://coolbutuseless.r-universe.dev

RemoteUrl https://github.com/coolbutuseless/nara

RemoteRef HEAD

RemoteSha 462c3228733749d24d2ff242fded342aa0df79b6

Contents

`deer_sprites` *List of deer native rasters*

Description

The 'deer' sprites are by Calciumtrice and licensed under CC BY 3.0. See 'inst/LICENSE-deer.txt' for more details. To view license information:

Usage

```
deer_sprites
```

Format

An object of class `list` of length 15.

Details

```
cat(readLines(system.file('LICENSE-deer.txt', package = 'nara')), sep = "\n")
```

---

is_nativeraster *Check if object is* nativeRaster

Description

Check if object is `nativeRaster`

Usage

```
is_nativeraster(x)
```

Arguments

`x`	object to check

Value

logical. TRUE if object is a `nativeRaster`

Examples

```
is_nativeraster(mtcars)
```

---

magick_to_nr *Convert a 'magick' image to* nativeRaster

Description

Convert a 'magick' image to `nativeRaster`

Usage

```
magick_to_nr(im, dst = NULL)

nr_to_magick(nr)
```

Arguments

im	image from the magick package
dst	destination nativeRaster object. If NULL (the default) a new nativeRaster will be created If a nativeRaster is supplied here, it must have the exact dimensions to match the matrix
nr	nativeRaster object

Value

nativeRaster

Examples

```
if (requireNamespace('magick', quietly = TRUE)) {
  im <- magick::image_read(system.file("img/Rlogo.png", package = "png"))
  nr <- magick_to_nr(im)
  plot(nr)
}
```

---

matrix_to_nr *Matrix to* nativeRaster

Description

Matrix to nativeRaster

Usage

```
matrix_to_nr(mat, palette, fill = "transparent", min = 0, max = 1, dst = NULL)
```

Arguments

mat	integer matrix
palette	vector of colors. For an integer matrix, this palette must contain at least as many colors as the maximum integer value in mat. For a numeric matrix, any length palette is allowed.
fill	Color to be used for values < 1 when input is an integer matrix. Default: 'transparent'.
min, max	assumed range for the numeric data. values from the palette will be interpolated using this range as the extents. An error will occur if a value lies outside this range. Default: (0, 1)
dst	destination nativeRaster object. If NULL (the default) a new nativeRaster will be created If a nativeRaster is supplied here, it must have the exact dimensions to match the matrix

Value

nativeRaster

Examples

```
m <- matrix(1:12, 3, 4)
palette <- str_cols_to_packed_cols(rainbow(12))
nr <- matrix_to_nr(m, palette)
plot(nr)
```

---

nrs_to_gif *Magick conversion - possibly won't be in final version of 'nara' pkg*

Description

Magick conversion - possibly won't be in final version of 'nara' pkg

Usage

```
nrs_to_gif(nr_list, gif_name, verbose = FALSE, framerate = 30, ...)
```

Arguments

nr_list	list of nativeRasters
gif_name	name of mp4 file to save
verbose	logical. default FALSE
framerate	frames per second
...	other arguments passed to magick::image_write_gif()

---

nrs_to_mp4 *Magick conversion - possible won't be in final version of 'nara' pkg*

Description

Magick conversion - possible won't be in final version of 'nara' pkg

Usage

```
nrs_to_mp4(nr_list, mp4_name, verbose = FALSE, ...)
```

Arguments

nr_list	list of nativeRasters
mp4_name	name of mp4 file to save
verbose	logical. default FALSE
...	other arguments passed to magick::image_write_video()

nr_blit *Copy one* nativeRaster *image into another at an arbitrary location.*

Description

This is useful as a way of positioning sprites or icons in an image.

Usage

```
nr_blit(
  nr,
  x,
  y,
  src,
  x0 = 0L,
  y0 = 0L,
  w = -1L,
  h = -1L,
  hjust = 0,
  vjust = 0,
  respect_alpha = TRUE
)
```

Arguments

nr	native raster to copy into
x, y	Where in nr to place the sprite. These values must be vectors of the same length. If the length is greater than 1, then the sprite will be pasted into nr at multiple locations. Note that the origin of nativeraster images is the top-left where the coordinates are (0, 0).
src	native raster to copy from
x0, y0	start coordiates within src
w, h	size within src. If size is negative, then the actual width/height of the src is used
hjust, vjust	specify horizontal and vertical justification of the src image. e.g. hjust = vjust = 0 the blitting starts at the top-left of the image. Use hjust = vjust = 0.5 to treat the centre of the src_ as the blitting origin. Default (0, 0)
respect_alpha	Should the alpha channel be respected when blitting? Default: TRUE. If FALSE, then contents will be blindly overwritten which can be much much faster. If the src has an any transparent pixels, respect_alpha = TRUE is probably the correct setting.

Value

nativeRaster

Examples

```
nr <- nr_new(50, 50, 'grey80')
nr_blit(nr, x = 0, y = 0, src = deer_sprites[[1]])
plot(nr)
```

`nr_blit_grid` *Blit4*

Description

Blit4

Usage

```
nr_blit_grid(
  nr,
  x,
  y,
  src_list,
  idx_mat,
  width,
  height,
  hjust = 0,
  vjust = 0,
  respect_alpha = TRUE
)
```

Arguments

`nr`	native raster to copy into
`x, y`	Where in `nr` to place the `sprite`. These values must be vectors of the same length. If the length is greater than 1, then the `sprite` will be pasted into `nr` at multiple locations. Note that the origin of `nativeraster` images is the top-left where the coordinates are (0, 0).
`src_list`	list of native rasters
`idx_mat`	integer matrix of indices into `src_list`
`width, height`	tile width/height (constant across all tiles)
`hjust, vjust`	specify horizontal and vertical justification of the `src` image. e.g. `hjust = vjust = 0` the blitting starts at the top-left of the image. Use `hjust = vjust = 0.5` to treat the centre of the `src_` as the blitting origin. Default (0, 0)
`respect_alpha`	Should the alpha channel be respected when blitting? Default: TRUE. If FALSE, then contents will be blindly overwritten which can be much much faster. If the `src` has an any transparent pixels, `respect_alpha = TRUE` is probably the correct setting.

Value

Original `nativeRaster` modified in-place

Examples

```
nr <- nr_new(100, 100, 'grey80')
idx_mat <- matrix(c(
  1, 2, 3,
  4, 5, 6,
  7, 8, 9
), 3, 3, byrow = TRUE)
nr_blit_grid(nr, 0, 0, src_list = deer_sprites, idx_mat = idx_mat, width = 32, height = 32)
```

---

nr_blit_list *Blit from a list of native rasters*

Description

Blit from a list of native rasters

Usage

```
nr_blit_list(
  nr,
  x,
  y,
  src_list,
  src_idx,
  hjust = 0,
  vjust = 0,
  respect_alpha = TRUE
)
```

Arguments

`nr`	native raster to copy into
`x, y`	Where in `nr` to place the `sprite`. These values must be vectors of the same length. If the length is greater than 1, then the `sprite` will be pasted into `nr` at multiple locations. Note that the origin of `nativeraster` images is the top-left where the coordinates are (0, 0).
`src_list`	list of native rasters
`src_idx`	indices into the list of the native raster
`hjust, vjust`	specify horizontal and vertical justification of the `src` image. e.g. `hjust = vjust = 0` the blitting starts at the top-left of the image. Use `hjust = vjust = 0.5` to treat the centre of the `src_` as the blitting origin. Default (0, 0)

respect_alpha	Should the alpha channel be respected when blitting? Default: TRUE. If FALSE, then contents will be blindly overwritten which can be much much faster. If the src has an any transparent pixels, respect_alpha = TRUE is probably the correct setting.

Examples

```
nr <- nr_new(50, 50, 'grey80')
nr_blit_list(nr, x = c(0, 25), y = c(0, 25), src_list = deer_sprites, src_idx = c(1, 2))
plot(nr)
```

---

nr_circle *Draw circles on a* nativeRaster *image*

Description

Draw circles on a nativeRaster image

Usage

```
nr_circle(nr, x, y, r, fill = "black", color = NA)
```

Arguments

nr	nativeRaster
x, y	coordinates of centre of circle. [vector]
r	radius [vector]
fill	interior fill color [vector]
color	outline color. Default: NA. [vector]

Value

Original nativeRaster modified in-place

Examples

```
N <- 21
nr <- nr_new(N, N, 'grey80')
nr_circle(nr, x = N/2, y = N/2, r = c(N/3, N/4), fill = c('darkred', 'black'))
plot(nr)
```

nr_copy_into *Copy the contents of one* nativeRaster *into another.*

Description

The source and destination nativeRaster images must have the same dimensions.

Usage

```
nr_copy_into(dst, src)
```

Arguments

src, dst	Source and destination nativeRaster images

Details

If the nativeRaster images are of different sizes, use the nr_blit() function.

Value

The 'dst' nativeRaster

Examples

```
nr1 <- nr_new(200, 100, 'hotpink')
nr2 <- nr_new(200, 100, 'green')
nr_copy_into(nr1, nr2)
plot(nr1)
```

nr_crop *Crop a section out of a nativeRaster into a new nativeRaster*

Description

Crop a section out of a nativeRaster into a new nativeRaster

Usage

```
nr_crop(nr, x, y, w, h)

nr_crop2(nr, loc)
```

Arguments

nr	nativeRaster
x, y, w, h	dimensions of cropped section
loc	dimensions of cropped section. A vector of 4 values i.e. c(x, y, w, h)

Value

New nativeRaster

Examples

```
nr <- nr_new(400, 400, 'hotpink')
nr2 <- nr_crop(nr, 0, 0, 10, 10)
dim(nr2)
plot(nr2)
```

---

nr_duplicate *Create a new* nativeRaster *image and copy the dimensions and contents from an existing image*

Description

Create a new nativeRaster image and copy the dimensions and contents from an existing image

Usage

```
nr_duplicate(nr)
```

Arguments

nr	nativeRaster

Value

New nativeRaster

Examples

```
nr1 <- nr_new(200, 200, 'hotpink')
nr2 <- nr_duplicate(nr1)
plot(nr2)
```

nr_fill *Fill a* nativeRaster *image with the given color*

Description

Fill a nativeRaster image with the given color

Usage

```
nr_fill(nr, color)
```

Arguments

nr	nativeRaster
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb

Value

The original nativeRaster modified in-place.

Examples

```
nr <- nr_new(400, 300, 'hotpink')
nr_fill(nr, 'blue')
plot(nr)
```

nr_fliph *Flip a* nativeRaster *horizontally*

Description

Flip a nativeRaster horizontally

Usage

```
nr_fliph(nr)
```

Arguments

nr	nativeRaster

Value

Original nativeRaster modified in-place

Examples

```
nr <- nr_new(400, 200, 'white')
nr_rect(nr, 0, 0, 30, 15)
plot(nr)
nr_fliph(nr)
plot(nr)
```

`nr_flipv` *Flip a* `nativeRaster` *vertically*

Description

Flip a `nativeRaster` vertically

Usage

```
nr_flipv(nr)
```

Arguments

`nr`	`nativeRaster`

Value

Original `nativeRaster` modified in-place

Examples

```
nr <- nr_new(400, 200, 'white')
nr_rect(nr, 0, 0, 30, 15)
plot(nr)
nr_flipv(nr)
plot(nr)
```

`nr_line` *Draw lines on a* `nativeRaster` *image*

Description

Uses Bresenham's algorithm to draw lines. No antialiasing.

Usage

```
nr_line(nr, x0, y0, x1, y1, color = "black")
```

Arguments

nr	nativeRaster
x0, y0, x1, y1	Vectors of coordinates of endpoints of line
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb

Value

Original nativeRaster modified in-place

Examples

```
N <- 20
nr <- nr_new(N, N, 'grey80')
nr_line(nr, x0 = c(0, N-1), y0 = c(0, 0), x1 = c(N-1, 0), y1 = c(N-1, N-1),
        color = c('red', 'black'))
plot(nr)
```

---

nr_new *Create a* nativeRaster *image*

Description

A nativeRaster in R looks like an integer matrix, but is interpreted differently by graphics devices:

Usage

```
nr_new(width, height, fill = "white")
```

Arguments

width, height	Image dimensions in pixels
fill	Background fill color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb

Details

- The data should be treated as RGBA pixels in row-major ordering
- Each 32-bit integer should be interpreted as 4-bytes - one for each of the R, G, B and A color channels

Value

nativeRaster

Examples

```
nr <- nr_new(400, 300, 'hotpink')
plot(nr)
```

---

nr_point *Draw points on a* nativeRaster *image*

Description

Draw points on a nativeRaster image

Usage

```
nr_point(nr, x, y, color = "black")
```

Arguments

nr	nativeRaster
x, y	Vectors of point coordinates
color	Vector of colors

Value

Original nativeRaster modified in-place

Examples

```
N <- 20
nr <- nr_new(N, N, 'grey80')
nr_point(nr, x = seq(N), y = seq(N), color = rainbow(N))
plot(nr)
```

---

nr_polygon *Draw polygon on a* nativeRaster *image*

Description

Draw polygon on a nativeRaster image

Usage

```
nr_polygon(nr, x, y, fill = "black", color = NA)
```

Arguments

nr	nativeRaster
x, y	Vectors of point coordinates
fill	fill color [scalar]
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb

Value

Original nativeRaster modified in-place

Examples

```
N <- 20
nr <- nr_new(N, N, 'grey80')
nr_polygon(nr, x = c(0, N-1, 0), y = c(0, 0, N-1), fill = 'blue', color = 'red')
plot(nr)
```

---

nr_polygons *Draw multiple polygon on a* nativeRaster *image*

Description

Draw multiple polygon on a nativeRaster image

Usage

```
nr_polygons(nr, x, y, id = NULL, fill = "black", color = NA)
```

Arguments

nr	nativeRaster
x, y	Vectors of point coordinates
id	integer vector used to separate coordinates into multiple polygons. Consecutive runs of the same id value belong to the same polygon. If NULL (the default) then all coordinates are assumed to be vertices of a single polygon.
fill	fill color
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb

Value

Original nativeRaster modified in-place

Examples

```
N <- 20
nr <- nr_new(N, N, 'grey80')
nr_polygon(nr, x = c(0, N-1, 0), y = c(0, 0, N-1), fill = 'blue', color = 'red')
plot(nr)
```

---

nr_polyline *Draw polyline on a* nativeRaster *image*

Description

Draw polyline on a nativeRaster image

Usage

```
nr_polyline(nr, x, y, color = "black", close = FALSE)
```

Arguments

nr	nativeRaster
x, y	Vectors of point coordinates
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb
close	Should the polyline be closed? I.e. should a line be drawn between the last point and the first point? Default: FALSE

Value

Original nativeRaster modified in-place

Examples

```
N <- 20
nr <- nr_new(N, N, 'grey80')
nr_polyline(nr, x = c(0, N-1, 0), y = c(0, 0, N-1), color = 'red')
plot(nr)
```

nr_rect *Draw rectangles on a* nativeRaster *image*

Description

Draw rectangles on a nativeRaster image

Usage

```
nr_rect(nr, x, y, w, h, fill = "black", color = NA, hjust = 0, vjust = 0)
```

Arguments

nr	nativeRaster
x, y	coordinates of lower left corner of rectangle. [vector]
w, h	width and height of rectangle. [vector]
fill	interior fill color [vector]
color	outline color. Default: NA. [vector]
hjust, vjust	specify horizontal and vertical justification of the src image. e.g. hjust = vjust = 0 the blitting starts at the top-left of the image. Use hjust = vjust = 0.5 to treat the centre of the src_ as the blitting origin. Default (0, 0)

Value

Original nativeRaster modified in-place

Examples

```
N <- 20
nr <- nr_new(N, N, 'grey80')
nr_rect(nr, x = c(0, N/2 - 1), y = c(0, N/2 - 1), w = N/2, h = N/4,
        fill = 'blue', color = c('red', 'green'))
plot(nr)
```

nr_replace *Replace colours in a native raster*

Description

Replace colours in a native raster

Usage

```
nr_replace(nr, old, new)
```

Arguments

nr	nativeRaster
old	Vector of old colours
new	Vector of replacement colours

Value

Original nativeRaster modified in-place

Examples

```
nr <- nr_new(10, 10, 'hotpink')
nr_replace(nr, 'hotpink', 'grey80')
plot(nr)
```

---

nr_resize *Scale a nativeRaster*

Description

Scale a nativeRaster

Usage

```
nr_resize(nr, width, height, algo = "nn")
```

Arguments

nr	native raster
width, height	new dimensions
algo	'nn' for nearest neighbour (the default), or 'bilinear' for bilinear interpolation.

Value

New nativeRaster

Examples

```
stretched <- nr_resize(deer_sprites[[1]], 100, 40, algo = 'nn')
plot(stretched)
```

nr_scale *Scale the size of a* nativeRaster *using Nearest Neighbour resizinng*

Description

Scale the size of a nativeRaster using Nearest Neighbour resizinng

Usage

```
nr_scale(nr, scale, algo = "nn")
```

Arguments

nr	nativeRaster
scale	scale factor
algo	'nn' for nearest neighbour (the default), or 'bilinear' for bilinear interpolation.

Value

New nativeRaster

Examples

```
big <- nr_scale(deer_sprites[[1]], 2)
plot(big)
```

nr_text_basic *Draw text on a* nativeRaster *image using the built-in* spleen *bitmapped font.*

Description

The only font currently available is 'spleen' - a monospace bitmap font from: https://github.com/fcambus/spleen

Usage

```
nr_text_basic(nr, x, y, str, color = "black", fontsize = 8L)
```

Arguments

nr	nativeRaster
x, y	coordinates of lower-left corner of text
str	character string
color	Color as a character string. Either a standard R color (e.g. 'blue', 'white') or a hex color of the form #rrggbbaa, #rrggbb, #rgba or #rgb
fontsize	height of font in pizels. Only valid values are 8, 12 and 16. Default: 8.

Details

The 'spleen' font is licensed under BSD and the license is included in this package as "LICENSE-spleen.txt". To view LICENSE:

```
cat(readLines(system.file('LICENSE-spleen.txt', package = 'nara')), sep = "\n")
```

Value

Original `nativeRaster` modified in-place

Examples

```
N <- 20
nr <- nr_new(N, N, 'grey80')
nr_text_basic(nr, x = 0, y = N/2, str = "Hi!")
plot(nr)
```

`nr_to_raster` *Convert* `nativeRaster` *images to/from other R objects*

Description

Convert `nativeRaster` images to/from other R objects

Usage

```
nr_to_raster(nr)

raster_to_nr(ras, dst = NULL)

nr_to_array(nr)

array_to_nr(arr, dst = NULL)
```

Arguments

`nr`	`nativeRaster` object
`ras`	standard R raster i.e. a character matrix of hex color values
`dst`	destination `nativeRaster` If NULL (the default) a new `nativeRaster` will be created.
`arr`	3d numeric array representing R,G,B,A values with dimensions [nrow, ncol, 4] or [nrow, ncol, 3]. Each value is in range [0,1].

Value

raster, array or `nativeRaster`

Examples

```
nr <- nr_new(12, 8, 'hotpink')
nr_to_raster(nr)
```

`packed_cols_to_hex_cols` *Convert packed colors (integer values containing RGBA bytes) to hex colors*

Description

Convert packed colors (integer values containing RGBA bytes) to hex colors

Usage

```
packed_cols_to_hex_cols(packed_cols)
```

Arguments

`packed_cols`	integer values each containing packed RGBA color information

Value

character vector of hex colors

Examples

```
packed_cols_to_hex_cols(c(-16776961L, -1L, -65536L, 16777215L, 16777215L))
```

`plot.nativeRaster` *Plot a* `nativeRaster` *as an image*

Description

Plot a `nativeRaster` as an image

Usage

```
## S3 method for class 'nativeRaster'
plot(x, y, ...)
```

Arguments

x	nativeRaster
y	any argument here will cause grid::grid.newpage() to be called prior to drawing the nativeRaster
...	other arguments passed to grid::grid.raster()

Value

None.

Examples

```
nr <- nr_new(200, 100, 'hotpink')
plot(nr)
```

---

str_cols_to_packed_cols

Convert colors (R colors and hex colors) into packed colors (integer values containing RGBA bytes)

Description

Convert colors (R colors and hex colors) into packed colors (integer values containing RGBA bytes)

Usage

```
str_cols_to_packed_cols(colors)
```

Arguments

colors	character vector of R color names and hex colors e.g. c('red', 'white', NA, 'transparent', '#12345678')

Value

Integer vector. Each integer value contains a packed color i.e. RGBA bytes.

Examples

```
str_cols_to_packed_cols(c('red', 'white', 'blue', NA, 'transparent'))
```

Index