Package: pixeltrix (via r-universe)

September 2, 2024

Title A Simple Interactive Pixel Art Editor

Version 0.2.2

Description A very simple 'pixel art' tool that lets you click squares interactively in a plot window and returns your final image as a matrix. Provide multiple frames to create an animated gif.

License MIT + file LICENSE

URL https://github.com/matt-dray/pixeltrix, https://matt-dray.github.io/pixeltrix

BugReports https://github.com/matt-dray/pixeltrix/issues

Depends R (>= 2.10)

Suggests gifski, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Repository https://matt-dray.r-universe.dev

RemoteUrl https://github.com/matt-dray/pixeltrix

RemoteRef HEAD

RemoteSha 9ffb4fda351bca9145533a67103773f9b5c1bf99

Contents

`as_pixeltrix` *Coerce to a 'pixeltrix' Object*

Description

Functions to check if an object is of 'pixeltrix' class, or coerce to it if possible.

Usage

```
as_pixeltrix(m)

is_pixeltrix(m)
```

Arguments

`m`	A matrix of integers to coerce.

Details

To be successfully coerced, `m` must be a matrix composed only of integers.

Value

`as_pixeltrix` returns an object of class 'pixeltrix' if possible. `is_pixeltrix` returns `TRUE` if the object has class 'pixeltrix', otherwise `FALSE`. If coerced, a named character-vector attribute, 'colours', is also added, which maps the matrix values to a gradated palette of greys in hex form.

Examples

```
m <- matrix(c(0L, 1L, 1L, 0L), 2, 2)
is_pixeltrix(m)
m

m <- as_pixeltrix(m)
is_pixeltrix(m)
m
```

`click_pixels` *Click 'Pixels' in an Interactive Plot*

Description

Opens an interactive plot with a grid of squares ('pixels') that you can click to cycle through. Returns a matrix 'blueprint' of your image.

Usage

```
click_pixels(
  n_rows = 8L,
  n_cols = 8L,
  n_states = 2L,
  colours = NULL,
  grid = TRUE
)
```

Arguments

`n_rows`	Integer. The number of pixels high that the plot should be. Numeric values are coerced to integer.
`n_cols`	Integer. The number of pixels wide that the plot should be. Numeric values are coerced to integer.
`n_states`	Integer. The number of states that a pixel can be cycled through with successive clicks. Numeric values are coerced to integer. See details.
`colours`	Character vector. As many named/hex colours as 'n_state'. Each click in the interactive plot will cycle a pixel through these colours. Defaults to NULL, which generates a gradation from white to dark grey. See details.
`grid`	Logical. Should a black boundary line be placed around the pixels to help differentiate between them? Defaults to TRUE.

Details

Click repeatedly the pixels in the interactive plotting window to cycle through the provided number of 'states'. The initial state value is 0 and successive clicks increase it by 1, wrapping back to 0 once the maximum number of states is exceeded. Press the ESCAPE key to exit the interactive mode.

If your editor opens a separate graphics window (i.e. not RStudio), each click may result in a brief flash as the image refreshes, while a resized window may return to its original dimensions. You may also hear a bell sound on click, which you can disable by setting `options(locatorBell = FALSE)`.

Value

A 'pixeltrix'-class matrix. The zero-indexed values correspond to the state of each pixel, which is determined by the number of clicks a user gave each pixel. Has a named character-vector attribute, 'colours', which maps the matrix values to hex colours provided by the user (or a gradated set of greys provided by default when the 'colours' argument is NULL).

Examples

```
## Not run:
# Create a 16 x 16 pixel matrix with 3 possible pixel states
my_matrix <- click_pixels(
  n_rows   = 16L,
  n_cols   = 16L,
  n_states = 3L,
  colours  = c("blue", "#FF0000", "yellow")
)

## End(Not run)
```

`draw_pixels` *Draw 'Pixel' Matrix in a Plot Window*

Description

Opens a plot window and draws a provided matrix. Designed for use with the output from `click_pixels` (or `edit_pixels`).

Usage

```
draw_pixels(m, colours = NULL)
```

Arguments

`m`	A matrix of integers. The maximum value is assumed to be the number of pixel states desired. Override by supplying a 'n_states' value larger than the maximum in the matrix.
`colours`	Character vector. As many named/hex colours as unique state values in 'm'. Each click in the interactive plot will cycle a pixel through these colours. Defaults to NULL, which results in an attempt to find and use the 'colours' attribute (a named character vector) of the input matrix, 'm', which is added by default to matrices created with `click_pixels` (recommended).

Value

Nothing.

Examples

```
## Not run:
my_matrix <- click_pixels(n_states = 3L)
draw_pixels(my_matrix, c("black", "#0000FF", "green"))  # a colour per state
## End(Not run)
```

edit_pixels *Edit 'Pixels' in an Interactive Plot*

Description

Opens an interactive plotting canvas with a grid of clickable squares ('pixels') that represent the cells of a matrix provided by the user, ideally the output from `click_pixels`.

Usage

```
edit_pixels(m, n_states = NULL, colours = NULL, grid = TRUE)
```

Arguments

`m`	A matrix of integers. The maximum value is assumed to be the number of pixel states desired. Override by supplying a 'n_states' value larger than the maximum in the matrix.
`n_states`	Integer. The number of states that a pixel can be cycled through with successive clicks. Numeric values are coerced to integer. Defaults to NULL, which results in an attempt to find and use the 'n_states' attribute (integer) of the input matrix, 'm'. The attribute is added by default to matrices created with `click_pixels` (recommended).
`colours`	Character vector. As many named/hex colours as 'n_state'. Each click in the interactive plot will cycle a pixel through these colours. Defaults to NULL, which results in an attempt to find and use the 'colours' attribute (a named character vector) of the input matrix, 'm'. This attribute is added by default to matrices created with `click_pixels` (recommended).
`grid`	Logical. Should a black boundary line be placed around the pixels to help differentiate between them? Defaults to TRUE.

Details

Click repeatedly the pixels in the interactive plotting window to cycle through the provided number of 'states'. The initial state value is 0 and successive clicks increase it by 1, wrapping back to 0 once the maximum number of states is exceeded. Press the ESCAPE key to exit the interactive mode.

If your editor opens a separate graphics window (i.e. not RStudio), each click may result in a brief flash as the image refreshes, while a resized window may return to its original dimensions. You may also hear a bell sound on click, which you can disable by setting `options(locatorBell = FALSE)`.

Value

A 'pixeltrix'-class matrix. The zero-indexed values correspond to the state of each pixel, which is determined by the number of clicks a user gave each pixel. Has a named character-vector attribute, 'colours', which maps the matrix values to hex colours provided by the user (or a gradated set of greys provided by default when the 'colours' argument is NULL).

Examples

```
## Not run:
# Create a 3 x 4 pixel matrix with 3 possible states to cycle through
my_matrix <- click_pixels(
  n_rows   = 3L,
  n_cols   = 4L,
  n_states = 3L,
  colours  = c("white", "red", "#0000FF")
)

.# Update the original matrix
my_matrix_edited <- edit_pixels(m = my_matrix)

# Update the original matrix with additional state, different colours
my_matrix_augmented <- edit_pixels(
  m        = my_matrix,
  n_states = 4L,  # one more than in the original
  colours  = c("bisque3", "orchid", "chartreuse", "olivedrab")
)
## End(Not run)
```

`frame_pixels` *Create Frames of a Pixel Animation*

Description

Opens a new interactive plotting canvas with a grid of clickable squares ('pixels'). When finished, the user is prompted to provide another.

Usage

```
frame_pixels(
  n_rows = 8L,
  n_cols = 8L,
  n_states = 2L,
  colours = NULL,
  grid = TRUE
)
```

Arguments

`n_rows`	Integer. The number of pixels high that the plot should be. Numeric values are coerced to integer.
`n_cols`	Integer. The number of pixels wide that the plot should be. Numeric values are coerced to integer.
`n_states`	Integer. The number of states that a pixel can be. Click a pixel to cycle through the states. Numeric values are coerced to integer. See details.

colours	Character vector. As many named/hex colours as n_state. Each click in the interactive plot will cycle a pixel through these colours. Defaults to NULL, which results in an attempt to find and use the 'colours' attribute (named character vector) of the input matrix, 'm'. This attribute is added by default to matrices created with click_pixels (recommended).
grid	Logical. Should a boundary line be placed around the pixels to make them easier to differentiate between them? Defaults to TRUE.

Details

Click the pixels in the plotting window repeatedly to cycle through a number of 'states'. Successive clicks increase the state value by 1 (wrapping back to 0, the default when the canvas is first plotted) and make the pixel darker grey in colour. Press the ESCAPE key to exit the mode. You'll be prompted interactively in the console to add a new frame or not. If you refuse a new frame, you'll be returned a list of matrices, each of which contain the state values of each pixel for each frame of your animation.

If your editor opens a separate graphics window (i.e. not RStudio), each click may result in a brief flash as the image refreshes, while a resized window may return to its original dimensions. You may also hear a bell sound on click, which you can disable by setting `options(locatorBell = FALSE)`.

Value

A list of matrices.

Examples

```
## Not run:
# Begin interactive sequence to create animation frames
my_frames <- frame_pixels(
  n_rows   = 16L,
  n_cols   = 16L,
  n_states = 3L,
  colours  = c("grey25", "green", "#0000FF")
)
## End(Not run)
```

---

gif_pixels *Write Frames of a Pixel Animation to GIF*

Description

Plots (using draw_pixels) each matrix from a list (ideally created using frame_pixels) and writes it to an animated GIF. Requires the 'gifski' package.

Usage

```
gif_pixels(frames, colours = NULL, file, ...)
```

Arguments

frames	A list of matrices. preferably produced by frame_pixels. Each matrix in the list is a frame of the final animation.
colours	A character vector of named colours. One for each unique value in the matrices of the 'frames' list. The order you provide the colours matches the sorted order of the values in the matrix. Defaults to NULL, which means colours will be extracted from the 'colours' attribute of the matrices (which is added to the object when using the recommended frame_pixels function), otherwise a gradated palette of greys will be selected.
file	A filepath expressed as a character vector, ending with extension '.gif'. This is where the output animation will be written.
...	Parameters to pass to save_gif (you must have 'gifski' installed).

Value

Nothing.

Examples

```
## Not run:
# Begin interactive sequence to create animation frames
my_frames <- frame_pixels(
  n_rows   = 16L,
  n_cols   = 16L,
  n_states = 3L,
  colours  = c("grey25", "green", "#0000FF")
)

# Write list of matrices to gif (requires 'gifski' installation)
gif_pixels(
  frames = my_frames,
  file = "example.gif",  # location to write GIF
  delay = 0.1            # passed to gifski::save_gif()
)
## End(Not run)
```

---

logo *Example Sprite Matrix: 'pixeltrix' Package Logo Text*

Description

A matrix output from click_pixels that represents the text of the pixeltrix package logo, which is the word 'pixeltrix' written in a font with letters that are 3 by 3 pixels.

Usage

```
logo
```

Format

A 'pixeltrix'-class matrix with 10 rows and 21 columns, taking integers from 0 to 2.

Examples

```
## Not run:
# Plot the matrix as an image
draw_pixels(m = pixeltrix::logo)
## End(Not run)
```

`mario_frames` *Example Sprite Matrix: Mario*

Description

A list of matrices output from `frame_pixels` that represent the pixels of the Mario sprite from Super Mario Bros. Each matrix is a step in the walk cycle.

Usage

```
mario_frames
```

Format

A list of 'pixeltrix'-class matrices, each with 16 rows and 16 columns, taking integers from 0 to 2.

Source

Super Mario Bros (1983), Nintendo.

Examples

```
## Not run:
# Write the list of matrices to a gif
gif_pixels(
  frames = pixeltrix::mario_frames,
  file = "mario.gif",
  delay = 0.15  # passed to gifski::save_gif()
)
## End(Not run)
```

pkmn_sprite *Example Sprite Matrix: Pokémon*

Description

A matrix output from `click_pixels` that represents the pixels of a player sprite from Pokémon.

Usage

```
pkmn_sprite
```

Format

A 'pixeltrix'-class matrix with 16 rows and 14 columns, taking integers from 0 to 2.

Source

Hand-copied from Pokemon (1996), The Pokemon Company.

Examples

```
## Not run:
# Plot the matrix as an image
draw_pixels(
  m = pixeltrix::pkmn_sprite,
  colours = c("#9bbc0f", "#8bac0f", "#306230")
)
## End(Not run)
```

Index