Package: gifski (via r-universe)

October 13, 2024

Type Package

Title Highest Quality GIF Encoder

Version 1.32.0-1

Description Multi-threaded GIF encoder written in Rust: <https://gif.ski/>. Converts images to GIF animations using pngquant's efficient cross-frame palettes and temporal dithering with thousands of colors per frame.

License MIT + file LICENSE

URL https://r-rust.r-universe.dev/gifski

BugReports https://github.com/r-rust/gifski/issues

SystemRequirements Cargo (Rust's package manager), rustc

Encoding UTF-8

RoxygenNote 7.1.1

Suggests ggplot2, gapminder

Language en-US

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

RemoteUrl https://github.com/r-rust/gifski

RemoteRef HEAD

RemoteSha 51a03ddb37c66fd8dee44da185b2a3459ab9cc52

Contents

gifski *Gifski*

Description

Gifski converts image frames to high quality GIF animations. Either provide input png files, or automatically render animated graphics from the R graphics device.

Usage

```
gifski(
  png_files,
  gif_file = "animation.gif",
  width = 800,
  height = 600,
  delay = 1,
  loop = TRUE,
  progress = TRUE
)

save_gif(
  expr,
  gif_file = "animation.gif",
  width = 800,
  height = 600,
  delay = 1,
  loop = TRUE,
  progress = TRUE,
  ...
)
```

Arguments

png_files	vector of png files
gif_file	output gif file
width	gif width in pixels
height	gif height in pixel
delay	time to show each image in seconds
loop	if the gif should be repeated. Set to FALSE to only play once, or a number to indicate how many times to repeat after the first.
progress	print some verbose status output
expr	an R expression that creates graphics
...	other graphical parameters passed to png

Examples

```
# Manually convert png files to gif
png_path <- file.path(tempdir(), "frame%03d.png")
png(png_path)
par(ask = FALSE)
for(i in 1:10)
  plot(rnorm(i * 10), main = i)
dev.off()
png_files <- sprintf(png_path, 1:10)
gif_file <- tempfile(fileext = ".gif")
gifski(png_files, gif_file)
unlink(png_files)
utils::browseURL(gif_file)


# Example borrowed from gganimate
library(gapminder)
library(ggplot2)
makeplot <- function(){
  datalist <- split(gapminder, gapminder$year)
  lapply(datalist, function(data){
    p <- ggplot(data, aes(gdpPercap, lifeExp, size = pop, color = continent)) +
    scale_size("population", limits = range(gapminder$pop)) + geom_point() + ylim(20, 90) +
    scale_x_log10(limits = range(gapminder$gdpPercap)) + ggtitle(data$year) + theme_classic()
    print(p)
  })
}

# High Definition images:
gif_file <- file.path(tempdir(), 'gapminder.gif')
save_gif(makeplot(), gif_file, 1280, 720, res = 144)
utils::browseURL(gif_file)
```

Index