Package 'R2SWF'

March 2, 2024

Version 0.9-9

Title Convert R Graphics to Flash Animations

Date 2024-03-02

Author Yixuan Qiu, Yihui Xie, Cameron Bracken and authors of included software. See file AUTHORS for details.

Maintainer Yixuan Qiu <yixuan.qiu@cos.name>

Imports sysfonts, grDevices, utils

Suggests XML, Cairo

Description Using the 'Ming' library <https://github.com/libming/libming> to create Flash animations. Users can either use the 'SWF' device swf() to generate 'SWF' file directly through plotting functions like plot() and lines(), or convert images of other formats ('SVG', 'PNG', 'JPEG') into 'SWF'.

Copyright see file COPYRIGHTS

License GPL-2

SystemRequirements zlib, libpng, FreeType

URL https://github.com/yixuan/R2SWF

BugReports https://github.com/yixuan/R2SWF/issues

RoxygenNote 7.3.1

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-03-02 06:02:36 UTC

R topics documented:

dev2swf 2
file2swf 3
image2swf 4

svg2swf 5
swf 6
swf2html 7

Index **9**

`dev2swf` *Convert R graphics to SWF using different graphics devices*

Description

Given an R expression that can produce a sequence of images, this function will record the images with the device provided (e.g. `png()` or `jpeg()`) and convert them to a Flash file.

Usage

```
dev2swf(
  expr,
  output = "movie.swf",
  bgColor = "white",
  interval = 1,
  dev = "png",
  file.ext = "png",
  img.name = "Rplot",
  ...
)
```

Arguments

`expr`	an expression to generate a sequence of images
`output`	the name of the output swf file
`bgColor`	background color of the output SWF file
`interval`	the time interval between animation frames
`dev`	the name of the graphics device to use (e.g. `'png'` or `'jpeg'`)
`file.ext`	the file extension for the images
`img.name`	the file name of the images without the extension
`...`	other arguments to be passed to the graphics device

Details

You can also use devices which are not in the **grDevices** package by setting the `dev` argument to the name of the function that opens a device, e.g. `CairoPNG()` in the **Cairo** package. Note that the `file.ext` argument should be set accordingly.

Value

The name of the generated swf file if succeeded.

Author(s)

Yihui Xie <https://yihui.org>

Examples

```
olddir = setwd(tempdir())
output1 = dev2swf({
  for(i in 1:10) plot(runif(20), ylim = c(0, 1))
}, dev='png', file.ext='png', output='movie-png.swf')
swf2html(output1)

if(capabilities("cairo")) {
    output2 = dev2swf({
        for(i in 1:10) plot(runif(20), ylim = c(0, 1))
    }, dev='svg', file.ext='svg', output='movie-svg.swf')
    swf2html(output2)
}

setwd(olddir)
```

---

file2swf *Convert image files to SWF*

Description

This function converts a sequence of PNG/JPEG/SVG image files to SWF. Based on the image format, it calls `image2swf` or `svg2swf`.

Usage

```
file2swf(files, output, bgColor = "white", interval = 1)
```

Arguments

`files`	a character vector of input filenames
`output`	the name of the output swf file
`bgColor`	background color of the output SWF file
`interval`	the time interval between animation frames

Value

The name of the SWF file.

Author(s)

Yihui Xie <https://yihui.org>

image2swf *Convert bitmap images to SWF*

Description

Given the file names of a sequence of images, this function can convert them into a Flash file (.swf). Supported input formats are jpg/jpeg and png. The two formats are allowed to appear in the same sequence.

Usage

```
image2swf(input, output = "movie.swf", bgColor = "white", interval = 1)
```

Arguments

input	the file names of the images to be converted
output	the name of the output SWF file
bgColor	background color of the output SWF file
interval	the time interval (in seconds) between animation frames

Details

This function uses the Ming library (https://github.com/libming/libming) to implement the conversion. If you want to create a Flash file consisting of vector graphics, use svg2swf() instead.

Value

The name of the generated swf file if successful.

Author(s)

Yixuan Qiu <https://statr.me>

Examples

```
if(capabilities("png")) {
  olddir = setwd(tempdir())
  png("Rplot%03d.png")
  for(i in 1:9) plot(runif(20), ylim = c(0, 1))
  dev.off()
  output = image2swf(sprintf("Rplot%03d.png", 1:9))
  swf2html(output)
  setwd(olddir)
}
```

svg2swf *Convert a sequence of SVG files to SWF file*

Description

Given the file names of a sequence of SVG files, this function could convert them into a Flash file (.swf).

Usage

```
svg2swf(input, output = "movie.swf", bgColor = "white", interval = 1)
```

Arguments

input	the file names of the SVG files to be converted
output	the name of the output SWF file
bgColor	background color of the output SWF file
interval	the time interval (in seconds) between animation frames

Details

This function uses the XML package in R and a subset of librsvg (https://wiki.gnome.org/action/show/Projects/LibRsvg) to parse the SVG file, and uses the Ming library (https://github.com/libming/libming) to implement the conversion. Currently this function supports SVG files created by svg() in the **grDevices** package, and CairoSVG() in the **Cairo** package.

Value

The name of the generated SWF file if successful.

Author(s)

Yixuan Qiu <https://statr.me>

Examples

```
## Not run:
if(capabilities("cairo")) {
  olddir = setwd(tempdir())
  svg("Rplot%03d.svg", onefile = FALSE)
  set.seed(123)
  x = rnorm(5)
  y = rnorm(5)
  for(i in 1:100) {
      plot(x <- x + 0.1 * rnorm(5), y <- y + 0.1 * rnorm(5),
           xlim = c(-3, 3), ylim = c(-3, 3), col = "steelblue",
           pch = 16, cex = 2, xlab = "x", ylab = "y")
  }
```

```
  dev.off()
  output = svg2swf(sprintf("Rplot%03d.svg", 1:100), interval = 0.1)
  swf2html(output)
  setwd(olddir)
}

## End(Not run)
```

---

swf *SWF graphics device*

Description

This function opens a SWF device that produces Flash animation in SWF format. Every time you call a high level plotting function like `plot()`, the movie will create a new frame and draw following shapes on it.

Usage

```
swf(
  file = "Rplots.swf",
  width = 7,
  height = 7,
  bg = "white",
  fg = "black",
  frameRate = 12
)
```

Arguments

`file`	a character string giving the output SWF file
`width`	the width of the device in inches
`height`	the height of the device in inches
`bg`	the background color of the SWF file
`fg`	initial foreground color
`frameRate`	how many frames to be played in 1 second

Author(s)

Yixuan Qiu <https://statr.me/>

Examples

```
## Not run:
## A demonstration of K-means clustering, using animation package
olddir = setwd(tempdir())
if(require(animation)) {
    swf("kmeans.swf", frameRate = 1)
    kmeans.ani()
    dev.off()
}

## Test built-in fonts in sysfonts package
swf("fonts.swf", 8, 8)
plot(1, type = "n")

par(family = "sans", cex = 2)
text(0.7, 1.3, "Sans-R", font = 1)
text(0.7, 1.1, "Sans-B", font = 2)
text(0.7, 0.9, "Sans-I", font = 3)
text(0.7, 0.7, "Sans-BI", font = 4)

par(family = "serif")
text(1.0, 1.3, "Serif-R", font = 1)
text(1.0, 1.1, "Serif-B", font = 2)
text(1.0, 0.9, "Serif-I", font = 3)
text(1.0, 0.7, "Serif-BI", font = 4)

par(family = "mono")
text(1.3, 1.3, "Mono-R", font = 1)
text(1.3, 1.1, "Mono-B", font = 2)
text(1.3, 0.9, "Mono-I", font = 3)
text(1.3, 0.7, "Mono-BI", font = 4)

dev.off()
setwd(olddir)

## End(Not run)
```

`swf2html` *Embed the SWF file into an HTML page*

Description

This function will generate an HTML file to display the Flash animation.

Usage

```
swf2html(swf.file, output, width = 480, height = 480, fragment = FALSE)
```

Arguments

swf.file	the path of the SWF file
output	the output path of the HTML file; by default 'foo.swf' produces foo.html if not specified (set FALSE so that no file will be written)
width	width of the Flash
height	height of the Flash
fragment	whether to produce an HTML fragment only

Value

The HTML code as a character string.

Author(s)

Yihui Xie <https://yihui.org>

Examples

```
olddir = setwd(tempdir())
output = dev2swf({
  for (i in 1:10) plot(runif(20), ylim = c(0, 1))
}, output = 'test.swf')
swf2html(output)
setwd(olddir)
```

Index

CairoPNG, *2*
CairoSVG, *5*

dev2swf, 2

file2swf, 3

image2swf, *3*, 4

jpeg, *2*

plot, *6*
png, *2*

svg, *5*
svg2swf, *3*, *4*, 5
swf, 6
swf2html, 7