

Package ‘gridGeometry’

October 13, 2022

Type Package

Title Polygon Geometry in 'grid'

Version 0.3-0

Author Paul Murrell

Maintainer Paul Murrell <paul@stat.auckland.ac.nz>

Description Functions for performing polygon geometry with 'grid' grobs.
This allows complex shapes to be defined by combining simpler shapes.

URL <https://github.com/pmur002/gridgeometry>,
<https://stattech.wordpress.fos.auckland.ac.nz/2019/03/04/2019-01-a-geometry-engine-interface-for-grid/>

Depends R (>= 3.6.0), grid

Imports grDevices, polyclip (>= 1.10-0)

Suggests graphics, lattice

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-03-20 22:10:02 UTC

R topics documented:

grid.polyclip	2
grid.trim	3
polyclip	4
trim	6
xyListFromGrob	7
xyListPath	8

Index	10
--------------	-----------

grid.polyclip

Perform Geometric Operations on Grobs

Description

These functions allow two or more grobs to be combined using one of the following operations: intersection, union, minus, and xor.

Usage

```
polyclipGrob(A, B, op="intersection",
             openFn=xyListToLine, closedFn=xyListToPath,
             name=NULL, gp=gpar(), ...)
grid.polyclip(A, B, ...)
```

Arguments

A	A grob, gList, or gTree, or a gPath or a character value identifying a grob that has already been drawn. This is known as the <i>subject</i> grob.
B	A grob, gList, or gTree, or a gPath or a character value identifying a grob that has already been drawn. This is known as the <i>clip</i> grob.
op	A character value describing the operation. One of "intersection", "minus", "union", or "xor".
openFn	The function used to create grobs from the open shapes in the result.
closedFn	The function used to create grobs from the closed shapes in the result.
name	A name for the resulting grob.
gp	Graphical parameter settings for the resulting grob.
...	For polyclipGrob, arguments passed on to polyclip. For grid.polyclip, arguments used by methods.

Details

The subject grob is combined with the clip grob using the op operation.

The grobs are converted to coordinates by calling `grid::grobCoords` and then the operation is performed by calling `polyclip`.

The result is a new grob. In the case of `grid.polyclip`, this new grob will be drawn on the current device. In the special case that A is a gPath, by default, the new grob will *replace* the old grob (the original grob identified by A) in the current scene (and the new grob will use the same gp settings as the old grob).

The subject grob can be any combination of open or closed shapes (e.g., a combination of lines and polygons), but the clip grob must only consist of closed shapes.

Value

polyclipGrob returns a gTree with two children, one representing the open shapes within the result and one representing the closed shapes within the result.

grid.polyclip is only used for its side-effect of drawing on the current graphics device.

Author(s)

Paul Murrell

See Also

[polyclip](#)

Examples

```
r <- rectGrob(x=1/3, y=1/3, width=.4, height=.4)
c <- circleGrob(x=2/3, y=2/3, r=.2)
grid.draw(r)
grid.draw(c)
grid.polyclip(r, c, gp=gpar(fill="grey"))
```

grid.trim

Subset a Line

Description

These functions generate one or more lines by subsetting a line.

Usage

```
trimGrob(x, from, to, rep=FALSE, name=NULL, gp=gpar(), ...)
grid.trim(x, ...)
```

Arguments

x	A grob, or a gPath (or a character value) identifying a grob that has already been drawn.
from	A numeric vector or a unit object describing the start point of each subset.
to	A numeric vector or a unit object describing the end point of each subset.
rep	A logical value indicating whether the from and to values should be repeated to consume the entire line.
name	A name for the resulting grob.
gp	Graphical parameter settings for the resulting grob.
...	For trimGrob, arguments passed on to trim. For grid.trim, arguments used by methods.

Details

Both `from` and `to` should be either numeric values between 0 and 1, expressing a proportion of the line length, or unit objects. In the latter case, the unit is converted to a proportion of the line length, with "npc" units treated as proportions of the line length.

Both `from` and `to` can be vectors (and they are recycled) so that multiple subsets can be obtained in a single call.

Both `from` and `to` can be negative, in which case the value is treated as distance backwards along the line from its end.

The result is a new grob. In the case of `grid.trim`, this new grob will be drawn on the current device. In the special case that `x` is a `gPath`, by default, the new grob will *replace* the old grob (the original grob identified by `x`) in the current scene (and the new grob will use the same `gp` settings as the old grob).

Value

`trimGrob` returns a polyline grob.

`grid.trim` is only used for its side-effect of drawing on the current graphics device.

Author(s)

Paul Murrell

See Also

[trim](#)

Examples

```
g <- xsplineGrob(c(.2, .5, .8), c(.2, .8, .2))
grid.draw(g)
grid.trim(g, from=.1, to=.2, gp=gpar(lwd=5))
grid.trim(g, from=.1, to=.2, rep=TRUE, gp=gpar(lwd=3))
```

Description

This function combines two sets of coordinates using one of the following operations: intersection, union, minus, and xor.

Usage

```

polyclip(A, B, ...)
## S3 method for class 'grob'
polyclip(A, B, op="intersection", closed=TRUE,
         reduceA = if (closed) "union" else "flatten",
         reduceB = "union",
         fillA = NULL, fillB = NULL,
         ...)
## S3 method for class 'gList'
polyclip(A, B, op="intersection", closed=TRUE,
         reduceA = if (closed) "union" else "flatten",
         reduceB = "union",
         fillA = NULL, fillB = NULL,
         ...)
## S3 method for class 'gPath'
polyclip(A, B, op="intersection", closed=TRUE,
         strict=FALSE, grep=FALSE, global=FALSE,
         reduceA = if (closed) "union" else "flatten",
         reduceB = "union",
         fillA = NULL, fillB = NULL,
         ...)
## S3 method for class 'character'
polyclip(A, B, op="intersection", closed=TRUE,
         strict=FALSE, grep=FALSE, global=FALSE,
         reduceA = if (closed) "union" else "flatten",
         reduceB = "union",
         fillA = NULL, fillB = NULL,
         ...)

```

Arguments

- | | |
|------------------|---|
| A | A set of coordinates describing the <i>subject</i> shape. Or a grob, gList, or a gPath (or a character value) identifying a grob that has already been drawn from which coordinates are generated. |
| B | A set of coordinates describing the <i>clip</i> shape. Or a grob, gList, or a gPath (or a character value) identifying a grob that has already been drawn from which coordinates are generated. |
| op | A character value describing the operation. One of "intersection", "minus", "union", or "xor". |
| closed | A logical value indicating whether the A coordinates describe a closed shape or an open shape. |
| reduceA, reduceB | A character value describing the operation to be used if either A or B need to be reduced to a single set of coordinates. One of "intersection", "minus", "union", or "xor", in which case polyclip is used to reduce multiple shapes, or "flatten", in which case coordinates for all shapes are returned. |

fillA, fillB A character value describing the fill rule. Possible values are "winding" or "evenodd" or NULL. In the latter case (the default), the fill rule will be taken from the coordinates if the coordinates have a "rule" attribute. Otherwise, the default is "winding".

strict, grep, global Arguments controlling the interpretation of the gPath (passed to grid.get).

... Arguments used by methods.

Details

The subject coordinates are combined with the clip coordinates using the op operation.

Value

The result is a new set of coordinates.

Author(s)

Paul Murrell

See Also

[polyclip](#)

Examples

```
r <- rectGrob(x=1/3, y=1/3, width=.4, height=.4)
c <- circleGrob(x=2/3, y=2/3, r=.2)
polyclip(r, c)
```

trim

Generate Subsets of Coordinates

Description

This functions generates a new set of coordinates by subsetting a set of coordinates.

Usage

```
trim(x, from, to, ...)
## S3 method for class 'grob'
trim(x, from, to, rep=FALSE, ...)
## S3 method for class 'gPath'
trim(x, from, to, rep=FALSE,
      strict=FALSE, grep=FALSE, global=FALSE, ...)
## S3 method for class 'character'
trim(x, from, to, rep=FALSE,
      strict=FALSE, grep=FALSE, global=FALSE, ...)
```

Arguments

x	A set of coordinates. Or a grob, or a gPath (or a character value) identifying a grob that has already been drawn from which coordinates are generated.
from	A numeric vector or a unit object describing the start point of each subset.
to	A numeric vector or a unit object describing the end point of each subset.
rep	A logical value indicating whether the from and to values should be repeated to consume the entire line.
strict, grep, global	Arguments controlling the interpretation of the gPath (passed to grid.get).
...	Arguments used by methods.

Value

A new set of coordinates.

Author(s)

Paul Murrell

See Also

[trim](#)

Examples

```
g <- segmentsGrob(0, .5, 1, .5)
trim(g, from=.1, to=.2)
trim(g, from=.1, to=.2, rep=TRUE)
```

xyListFromGrob

Generate Coordinates from Grobs

Description

This function generates a set of coordinates (a list of (x,y) lists) from a grob.

Usage

```
xyListFromGrob(x, op = "union", closed = TRUE, ...)
```

Arguments

x	A grob.
op	A character value describing the operation to be used if x needs to be collapsed to a single shape. One of "intersection", "minus", "union", or "xor".
closed	A logical value indicating whether we coordinates describing a closed shape or an open shape from.
...	Arguments passed on to polyclip.

Value

The result is a list of lists, each with components x and y.

Author(s)

Paul Murrell

See Also

[polyclip](#)

Examples

```
r <- rectGrob(x=1/3, y=1/3, width=.4, height=.4)
xyListFromGrob(r)
```

xyListPath

Generate Grobs from Coordinates

Description

This function generates a grob from a set of coordinates (a list of (x,y) lists).

Usage

```
xyListToPath(x, rule="winding", name=NULL, gp=gpar())
xyListToPolygon(x, name=NULL, gp=gpar())
xyListToLine(x, name=NULL, gp=gpar())

xyListPath(x, rule="winding", name=NULL, gp=gpar())
xyListPolygon(x, name=NULL, gp=gpar())
xyListLine(x, name=NULL, gp=gpar())
```

Arguments

x	A set of coordinates (e.g., from <code>polyclip</code> or <code>trim</code>).
rule	A fill rule: "winding" or "evenodd".
name	A name for the resulting grob.
gp	Graphical parameter settings for the resulting grob.

Details

Following the addition of the `xyListFromGrob` function, the `*To*()` forms are preferred.

Value

The result is a grob.

Author(s)

Paul Murrell

See Also

[polyclip](#)

Examples

```
r <- rectGrob(x=1/3, y=1/3, width=.4, height=.4)
c <- circleGrob(x=2/3, y=2/3, r=.2)
coords <- polyclip(r, c)
xyListPath(coords)
xyListLine(coords)
```

Index

* **aplot**

- grid.polyclip, 2
- grid.trim, 3
- polyclip, 4
- trim, 6
- xyListFromGrob, 7
- xyListPath, 8

* **dplot**

- grid.polyclip, 2
- grid.trim, 3
- polyclip, 4
- trim, 6
- xyListFromGrob, 7
- xyListPath, 8

grid.polyclip, 2
grid.trim, 3

polyclip, 3, 4, 6, 8, 9
polyclipGrob (grid.polyclip), 2

trim, 4, 6, 7
trimGrob (grid.trim), 3

xyListFromGrob, 7, 8
xyListLine (xyListPath), 8
xyListPath, 8
xyListPolygon (xyListPath), 8
xyListToLine (xyListPath), 8
xyListToPath (xyListPath), 8
xyListToPolygon (xyListPath), 8