Package: pfft (via r-universe)

August 26, 2024

Version 0.0.0.9000

Title What the Package Does (one line, title case)

Description What the package does (one paragraph).

License GPL-3

Encoding UTF-8

LazyData true

ByteCompile true

Imports dplyr, rlang, RTriangle, silicate (>= 0.1.5.9001), polyclip, tibble

RoxygenNote 6.1.1

Suggests ggplot2, purrr

Remotes hypertidy/silicate

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

RemoteUrl https://github.com/hypertidy/pfft

RemoteRef HEAD

RemoteSha 061449075960101014be3e9078bcfdaeec2f4809

Contents

edge_RTriangle *Triangles*

Description

Build triangles from PATH

Usage

```
edge_RTriangle(x, ...)
```

Arguments

x	PATH
...	arguments passed to ‘RTriangle::triangulate‘

Details

Input edges in the form of ‘silicate::PATH‘ and return a RTriangle object.

Value

RTriangle triangulation

Examples

```
data("minimal_mesh", package = "silicate")
library(RTriangle)
mm <- silicate::PATH(minimal_mesh)
plot(edge_RTriangle(mm), asp = 1)
plot(edge_RTriangle(mm, D= TRUE), asp = 1)
```

extents *All extents*

Description

(This function probably belongs in spex). Find the extent of all paths within an object.

Usage

```
extents(x)
```

Arguments

x	Object with paths

Details

The ‘path_‘ identifier is included, but won’t be of use without an existing ‘PATH‘ object. The path order is implicit as per the gibble geometry map.

Value

a dataframe of object and extent values (xmin, xmax, ymin, ymax)

Examples

```
data("minimal_mesh", package = "silicate")
extents(minimal_mesh)
```

`path_triangle_map` *Triangles in paths*

Description

Build a map of triangles to paths (polygon ring)

Usage

```
path_triangle_map(x, RTri)
```

Arguments

`x`	PATH object
`RTri`	RTriangle triangulation

Value

data frame mapping triangles to their containing paths

Examples

```
data("minimal_mesh", package = "silicate")
p <- silicate::PATH(minimal_mesh)
tr <- edge_RTriangle(p)
path_triangle_map(p, tr)

library(ggplot2)
library(dplyr)
library(purrr)
ggplot(path_triangle_map(p, tr) %>%
inner_join(reduce(p[c("path", "path_link_vertex", "vertex")], inner_join))) +
geom_polygon(aes(x_, y_, fill = path_)) + facet_wrap(~path_)
```

Index