Package ‘Rpolyhedra’

March 26, 2019

Type Package

Title Polyhedra Database

Version 0.4.2

Language en-US

Maintainer Alejandro Baranek <abaranek@dc.uba.ar>

Description
A polyhedra database scraped from various sources as R6 objects and 'rgl' visualizing capabilities.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, covr

VignetteBuilder knitr

Depends R (>= 3.4)

Imports R6, geometry, rgl, stringr, dplyr, XML, digest, testthat,
futile.logger, git2r,

Collate 'Rpolyhedra-package.R' 'polyhedra-lib.R' 'ledger-lib.R'
'db-lib.R' 'env-lib.R' 'package-lib.R' 'serialization-lib.R'
'public-lib.R' 'test-lib.R' 'zzz.R'

BugReports https://github.com/ropensci/Rpolyhedra/issues

URL https://github.com/ropensci/Rpolyhedra

NeedsCompilation no

Author Alejandro Baranek [aut, com, cre, cph],
Leonardo Belen [aut, com, cph],
Barret Schloerke [rev],
Lijia Yu [rev]

Repository CRAN

Date/Publication 2019-03-26 17:13:23 UTC

R topics documented:

`Rpolyhedra-package` *Rpolyhedra: Polyhedra Database*

Description

A polyhedra database scraped from various sources as R6 objects and 'rgl' visualizing capabilities.

Details

A polyhedra database scraped from:

- http://paulbourke.net/dataformats/phd/: PHD files as R6 objects and 'rgl' visualizing capabilities. The PHD format was created to describe the geometric polyhedra definitions derived mathematically <http://www.netlib.org/polyhedra/> by Andrew Hume and by the Kaleido program of Zvi Har'El.
- http://dmccooey.com/Polyhedra: Polyhedra text datafiles.

Author(s)

Maintainer: Alejandro Baranek <`abaranek@dc.uba.ar`> [compiler, copyright holder]

Authors:

- Leonardo Belen <`leobelen@gmail.com`> [compiler, copyright holder]

Other contributors:

- Barret Schloerke <`schloerke@gmail.com`> [reviewer]
- Lijia Yu <`yu@lijiayu.net`> [reviewer]

See Also

Useful links:

- `https://github.com/ropensci/Rpolyhedra`
- Report bugs at `https://github.com/ropensci/Rpolyhedra/issues`

`getAvailablePolyhedra` *Get available polyhedra*

Description

Gets the list of names of available polyhedra and its status in the polyhedra database, which can be later called with getPolyhedron

Usage

```
getAvailablePolyhedra(sources, search.string)
```

Arguments

`sources`	A string vector containing the source, which can be obtained from getAvailableSources().
`search.string`	A search string

Value

polyhedra names vector

See Also

getAvailableSources

Examples

```
#gets all polyhedra in the database
available.polyhedra <- getAvailablePolyhedra()

#returns all polyhedra from a given source, in this case, netlib
available.netlib.polyhedra <- getAvailablePolyhedra(sources="netlib")

#search within the polyhedron names

cube <- getAvailablePolyhedra(sources="netlib",search.string="cube")
cube
```

`getAvailableSources` *Get available sources*

Description

Gets the list of names of available sources in database to be used later as references to the package.

Usage

```
getAvailableSources()
```

Value

sources string vector, which can be obtained from getAvailableSources()

See Also

getAvailablePolyhedra, getPolyhedron

Examples

```
#gets all sources in the database
available.sources <- getAvailableSources()

#returns all polyhedra from all sources
available.polyhedra <- getAvailablePolyhedra(sources=available.sources)

#search within the polyhedron names from all sources
cubes <- getAvailablePolyhedra(sources=available.sources,
        search.string="cube")
cubes
```

`getPolyhedraObject` *Get a polyhedra object*

Description

Return the polyhedra database handler.

Usage

```
getPolyhedraObject()
```

Value

.polyhedra

See Also

PolyhedraDatabase.class

`getPolyhedron` *Get polyhedron*

Description

Gets a polyhedron from the database. It returns an R6 Class with all its characteristics and functions. The object returned, of type Polyhedron.class, allows to the user to get access to all the functionality provided.

Usage

```
getPolyhedron(source = "netlib", polyhedron.name)
```

Arguments

`source`	string vector, which can be obtained from getAvailableSources()
`polyhedron.name`	a valid name of a polyhedron in the database. Current names can be found with getAvailablePolyhedra()

Value

polyhedron R6 object

See Also

getAvailablePolyhedra, getAvailableSources

Examples

```
tetrahedron <- getPolyhedron(source = 'netlib',
       polyhedron.name = 'tetrahedron')

# returns name of polyhedra
tetrahedron$getName()

# polyhedron state
tetrahedron.state <- tetrahedron$getState()

# Johnson symbol and Schlafli symbol
tetrahedron.state$getSymbol()

# vertex data.frame
tetrahedron.state$getVertices()
```

```
# List of faces of solid representation (3D)
tetrahedron.state$getSolid()

# List of faces of net representation (2D)
tetrahedron.state$getNet()
```

PolyhedraDatabase.class *Polyhedra database*

Description

Scrapes all polyhedra in data folder to save a representation which is accessible by the final users upon call to `getPolyhedron()`.

Usage

```
PolyhedraDatabase.class
```

Format

`R6Class` object.

Fields

`polyhedra.rds.file` path of rds database file

`sources.config` Sources configuration for scraping different sources

`ledger` rr ledger of scraping process

`data` Polyhedra data from different sources

Methods

`initialize()` Initializes the object

`existsSource(source)` Determines if the source exists on the database

`getPolyhedraSourceDir(source)` Retrieves polyhedra dir of a source

`addSource(source)` Adds a new source to the database

`configPolyhedraRDSPath()` config path for rds database file

`existsPolyhedron(source,polyhedron.name)` Determines if the polyhedron exists on the database

`getPolyhedron(source, polyhedron.name, strict)` Retrieves a polyhedron by source and name

`addPolyhedron(source, polyhedron, overwrite, save.on.change = FALSE)` Adds a polyhedron by source and name, if overwrite is TRUE, it will update any existing one by that source and name

`configPolyhedraSource(source.config, source.filenames, max.quant)` Scrapes all polyhedra in the given directory for adding to db or testing `schedulePolyhedraSources(sources.config, source.filenames, max.quant, test)` Scrapes files applying parameter sources.config

`cover(sources, covering.code, polyhedra.names = NULL, max.quant = 0, save.on.change = FALSE, seed =` Cover all polyhedron with specified code

`scrape(mode = "scrape.queued", sources = names(self$sources.config), max.quant = 0, time2scrape.sou` Scrape file with specified parameters

`saveRDS = function(save.on.change = TRUE)` Save state in file when specified

`getAvailablePolyhedra(sources,search.string)` Retrieves all polyhedron within the source those names match with search.string

`polyhedronToXML` *Polyhedron to XML*

Description

Gets an XML representation out of the polyhedron object

Usage

```
polyhedronToXML(polyhedron.state.defined, is.transformed.vertices = TRUE)
```

Arguments

`polyhedron.state.defined`
the polyhedron to get a representation from

`is.transformed.vertices`
flag which states if vertices are in original position or transformationMatrix applied

Value

an XML document, ready to be converted to String with XML::saveXML()

Examples

```
#get the representation of a cube (netlib library)
XML::saveXML(polyhedronToXML(getPolyhedron("netlib", "cube")$state))
```

scrapePolyhedra *Scrape polyhedra objects*

Description

Gets polyhedra objects from text files of different sources, scheduling and scraping using predefined configurations.

Usage

```
scrapePolyhedra(scrape.config, source.filenames = NULL,
  sources.config = getPackageEnvir(".available.sources"))
```

Arguments

`scrape.config`	predefined configuration for scraping
`source.filenames`	if not null specify which source filenames to scrape
`sources.config`	the sources that will be used by the function

Value

polyhedra db object

scrapePolyhedraSources *Scrape polyhedra sources*

Description

Scrapes polyhedra objects from text files of different sources, in order to make them available to the package.

Usage

```
scrapePolyhedraSources(sources.config =
        getPackageEnvir(".available.sources"),
    max.quant.config.schedule = 0,
    max.quant.scrape = 0, time2scrape.source = 30,
    source.filenames = NULL, retry.scrape = FALSE)
```

Arguments

`sources.config` the sources that will be used by the function

`max.quant.config.schedule` number of files to schedule

`max.quant.scrape` number of files scrape

`time2scrape.source` time applied to scrape source

`source.filenames` if not null specify which source filenames to scrape

`retry.scrape` should it retry scrape?

Value

polyhedra db object

`switchToFullDatabase` *Switch to full database*

Description

Prompts user for changing database to fulldb in user filespace. Also, allows the user to switch back to the package database, which is a minimal one for testing purposes.

Usage

```
switchToFullDatabase(env=NA)
```

Arguments

`env` The environment to run on, can be PACKAGE, HOME or NA. If NA, it asks the user for a an Environment.

Value

.data.env

Index