Package 'sculpt3d'

January 2, 2012

Type Package

Title A simple toolbar GUI for brushing RGL plots

Version 0.2-2

Date 2009-12-20

Author Justin Donaldson

Maintainer Justin Donaldson <jdonaldson@gmail.com>

Description sculpt3d is a GTK+ toolbar that allows for more interactive control of a dataset inside the RGL plot window. Controls for simple brushing, highlighting, labeling, and mouseMode changes are provided by point-and-click rather than through the R terminal interface.

License GPL

Depends R (>= 2.2.0), rgl (>= 0.66), RGtk2

LazyLoad yes

Repository CRAN

Date/Publication 2009-12-21 07:41:46

R topics documented:

sculpt3d-package *A simple toolbar GUI for brushing RGL plots*

Description

sculpt3d is a GTK+ toolbar that allows for more interactive control of a dataset inside the RGL plot window. Controls for simple brushing, highlighting, labeling, and mouseMode changes are provided by point-and-click rather than through the R terminal interface.

Details

Package:	sculpt3d
Type:	Package
Version:	0.2
Date:	2009-07-15
License:	GPL
LazyLoad:	yes

Author(s)

Justin Donaldson

Maintainer: Justin Donaldson <jdonaldson@gmail.com>

See Also

`sculpt3d` : Create a toolbar to plot and interact with the given data
`sculpt3d.selected` : Retrieve a logical vector indicating currently selected points
`sculpt3d.current` : Retrieve a logical vector indicating currently visible points
`sculpt3d.setCallback` : set a callback function for selection/crop/deletion events

sculpt3d *The sculpt3d GUI for interactive 3d plot brushing and editing*

Description

Provides a simple toolbar GUI for interacting with 3D rgl plots. The data to be plotted is provided to sculpt3d(), which calls relevant rgl plot3d() functions by proxy. A subset of the plot3d() arguments is accommodated. See below for the list of acceptable parameters.

Usage

```
sculpt3d(x, y = NULL, z = NULL, col = 'black', labels = NULL, radius = NULL, type = 'p', alpha = NULL, cal
```

Arguments

x, y, z	coordinates. Any reasonable way of defining the coordinates is acceptable. See the function xyz.coords for details.
col	the color to be used for plotted items.
labels	the labels to be used for plotted items.
radius	the radius to be used if type='s': see Details below.
type	the type of plot to generate. Currently supported types include 'p' for points, and 's' for spheres.
alpha	the alpha transparency to use.
size	the size for plotted points
callback	the callback function to use.

Details

The above arguments are passed on to plot3d for plotting. However, rgl currently does not have a method of determining which data points are currently being represented in a given rgl device. Therefore, it is necessary to first pass this data to sculpt3d to generate both the toolbar and the rgl plot.

Through the course of interaction with the toolbar, the data points become filtered by the select3d function. This type of filtering is not easily applicable for certain rgl plot methods (such as lines, meshes, quads, etc.), so only plots comprised of individual datapoints such as points and spheres are allowed for sculpt3d.

The toolbar provides the following functionality:

1. **Selecting data points** : Clicking the **Select3d** button will activate the select3d mode in rgl. Once clicked, it is possible to select data points in the current rgl plot. These points will be colored by the current color indicated by the **Color** button.
2. **Changing color of selected data points** : Clicking the **Color** button will change the highlight color of the selection mode.
3. **Labeling data points** : This attaches text labels to each datapoint in the plot. By default these are a simple enumeration of the datapoints, but it's possible to override this by passing a label argument.
4. **Cropping data points** : Once data points are selected with **Select3d**, it is possible to crop or delete them with the **Crop** button. The data is not actually deleted, it is just filtered from the current plot view.
5. **Deleting data points** : The **Delete** works similarly to the Crop Data Points button, except it deletes the currently selected points.
6. **Resetting plot** : The **Reset** button will reset the plot back to its original state.
7. **Changing the mouse mode** : The three combo box controls at the bottom can change the mouseMode settings in par3d. See Warning section below.

During interaction with the toolbar and plot, it is possible to retrieve a logic vector of the selected or visible items by calling `sculpt3d.selected` or `sculpt3d.current` resepectively. Other rgl commands are still available, such as the `rgl.snapshot` for saving plots as images.

If the rgl plot window is closed, the toolbar will throw an error and close the next time a button is clicked.

Value

`sculpt3d` is called for the side effect of drawing the RGL plot and generating the GUI. No value is returned.

Warning

It is strongly recommended to not call additional rgl plot functions on sculpt3d's current rgl device. This can cause confuse the plotting, selection, and editing routines, and can crash the toolbar.

The toolbar itself is generated from a glade xml file, and is created as window in GTK+'s toolbar window-hint mode. This allows the toolbar to float on top of the OpenGL window on some platforms, and remain immediately active without the need to switch between applications in the window manager. However, this has the unfortunate side effect of obscuring the combo box drop down list on some platforms. Currently, the workaround is to select the combo box, and change the values with the up/down arrows until it shows the correct value.

In addition, the AQUA OpenGL target in OSX is handled differently than the other platforms. The `rgl.select3d` function can only be called directly from the R Aqua console if that console is hosting the interactive session. Therefore, it is necessary to use AppleScript to send the relevant command to the R console directly to prevent a hung session. This has the side effect of including this private method call into the console history, as well as introducing a small delay when clicking the `Select3d` button on the toolbar. Other platforms are not affected in this manner.

Author(s)

Justin Donaldson

See Also

`select3d`, `plot3d`

Examples

```
## Not run:
#run this directly with demo(sculpt3d)
x <- sort(rnorm(1000))
y <- rnorm(1000)
z <- rnorm(1000) + atan2(x,y)
sculpt3d(x, y, z, labels=1:1000, col=rainbow(1000), type='s', radius=runif(1000)/5, alpha = .5)

## End(Not run)
```

sculpt3d.current *Retrieve currently visible points from RGL plot*

Description

Returns a logical vector (of TRUE/FALSE) indicating if the given point index is currently displayed in the RGL interface.

Usage

```
sculpt3d.current()
```

Value

A vector of TRUE/FALSE values of equal length to the number of datapoints provided to sculpt3d().

Author(s)

Justin Donaldson

See Also

sculpt3d, sculpt3d.selected

sculpt3d.selected *Retrieve selected points from RGL plot*

Description

Returns a logical vector (of TRUE/FALSE) indicating if the given point index is selected in the RGL interface.

Usage

```
sculpt3d.selected()
```

Value

A vector of TRUE/FALSE values of equal length to the number of datapoints provided to sculpt3d().

Author(s)

Justin Donaldson

See Also

sculpt3d, sculpt3d.current

`sculpt3d.setCallback` *Set a callback function for the GUI*

Description

Sets a callback function to be called after select/crop/delete commands. The callback function could be used to call/update additional plots using selection/current points in the current sculpt3d rgl plot.

To disable the callback, simply pass NULL.

Usage

```
sculpt3d.setCallback(callback = NULL)
```

Arguments

`callback`	Callback function. The function should accept arguments of 'current' and 'selected' as logical vectors, and 'selected_color' as the selection color: `sculpt3d.setCallback(f = (current, selected, selected_color){...})`

Value

Called for the side effect of calling the corresponding callback function passed as an argument.

Author(s)

Justin Donaldson

See Also

`sculpt3d`,

Examples

```
## Not run:
x <- sort(rnorm(1000))
y <- rnorm(1000)
z <- rnorm(1000) + atan2(x,y)
sculpt3d(x, y, z, labels=1:1000, col=rainbow(1000), type='s', radius=runif(1000)/5, alpha = .5)
f = function(current, selected, selected_color) {
plot(x[current],y[current],type='n') # set plot dimensions
points(x[current & !selected],z[current & !selected]) # current points
points(x[selected],z[selected], col=selected_color)   # selected points
}
sculpt3d.setCallback(f)
# now select/crop/delete using the toolbar to see the auxiliary 2d plot update itself

## End(Not run)
```

Index