A Guide Though the work

```
library(ModelSectionView)
```

Introduction

this document is created in order to help understanding the functions in modelsectionview testpackage.

general understanding: this testpackage store models as control vertex and surface list. in other word, a model with N vertex and S surface will be stored as N * 3 matrix for vertex's cords, and S * 3 matrix that which 3 vertices form a triangular surface. Notice that even there is a square, it will be stored as 2 right triangle in same plane.

Functions

there are varies functions presented in this package. while sectionview function will draw a graph, all other functions will return modified model.

dummymodel

this function stored some premake model, calling the id will return corresponding sample model

```
newmodel = dummymodel(id = 0)
newmodel
#> $name
#> [1] "cube"
#>
#> $points
#>      [,1] [,2] [,3]
#> [1,]    1    1    1
#> [2,]    1   -1    1
#> [3,]   -1    1    1
#> [4,]   -1   -1    1
#> [5,]    1    1   -1
#> [6,]    1   -1   -1
#> [7,]   -1    1   -1
#> [8,]   -1   -1   -1
#>
#> $surface
#>       [,1] [,2] [,3]
#>  [1,]    1    2    3
#>  [2,]    2    3    4
#>  [3,]    5    6    7
#>  [4,]    6    7    8
#>  [5,]    1    2    5
#>  [6,]    2    5    6
```

```
#>  [7,]    3    4    7
#>  [8,]    4    7    8
#>  [9,]    6    2    8
#> [10,]    2    8    4
#> [11,]    5    1    7
#> [12,]    1    7    3
```

```
boxinbox = dummymodel(id = 1)
```

baseshapenew

the most fundemental function, this function has two param that are

- @param preset a scaler that will pull preset model. 1 = square , 2 = triangle
- @param controlpoints the 3 * 3 matrix that each row present 3 points location.

this function either receive a preset number, or 3* matrix as 3 control points to generate a triangle shape in 3d. returning the control points and surface as list. the dummy model function will call this function many times to create a small model.

spin

spin function make the model spin as origin alone 3 axis. receving Vertex as inputs, one of 3 direction and spin angle, output the spined point's location

```
boxinbox$points = spin(boxinbox$points, 0, pi/4)
```

shift

shift function make the entire model shift to certain location, or shift by fixed location.

```
boxinbox$points = shift(boxinbox$points, c(2,0,0))
```

join

this function will take two models, that is the control points and surface from each model, and combining the two together. the function is simple, simply stack corresponding matrix using rbind. and return united model as points and surface.

```
lists = join(newmodel$points, newmodel$surface, boxinbox$points, boxinbox$surface)
points = lists$points
surface = lists$surface
```

scalingmodel

this function will twists the points in desired way. functions receive vertices and vector with length 3 as scalar as well as scaling center vertex.

for each point, the scalingmodel do the following

$ newpoint = (point - center) * scalar + center $

simplifymodel

this function are called when there is redundence in original model vertex. for example, there is two vertex in matrix, both showing (0,0,0) location the function will erease one of the vertex and join related surface to the remain one

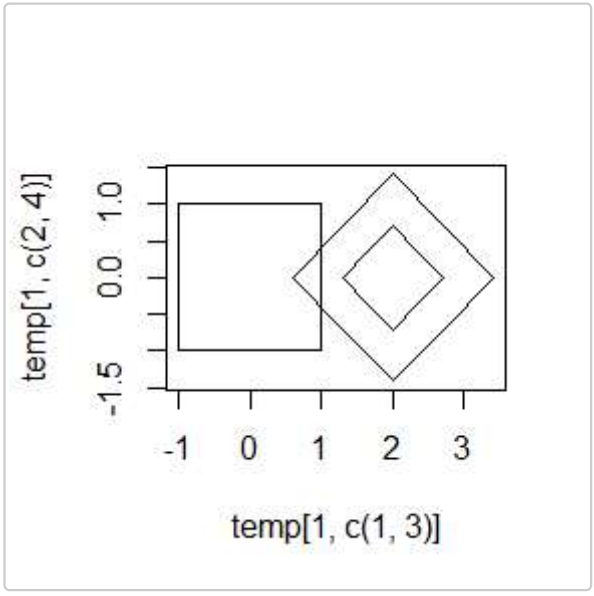
```
samplepoint = matrix(0,nrow = 3, ncol = 3)
samplesurface = matrix(c(1,2,3),ncol = 3)
simplifymodel(samplepoint, samplesurface)
#> $points
#> [1] 0 0 0
#>
#> $surface
#>      [,1] [,2] [,3]
#> [1,]    1    1    1
```

this function often called in dummymodel but are welcomed to use while constructing own model.

sectionview

this function will plot 2d graph when called. if there is no model intersect with plane, function exit with no graph printed

```
a = sectionview(points,surface)
```



with the graph above, we can tell there is a box, and another box inside a box. this model is created along the show