

**tools**

<b>COLLABORATORS</b>
----------------------

	TITLE :  tools		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 25, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>tools</b>	<b>1</b>
1.1	tools.guide . . . . .	1
1.2	modifycogs . . . . .	1
1.3	modifyacs . . . . .	2
1.4	modifynl . . . . .	2
1.5	fontloader . . . . .	2
1.6	meshtosph . . . . .	3
1.7	keyframer . . . . .	3

# Chapter 1

## tools

### 1.1 tools.guide

#### SPECIAL TOOLS

These tools can be accessed by loading the environment SpecialTools from the drawer Environments with the function 'Environment/Insert'.

The window consists of a set of custom gadgets bound to RPL files found in the Macros drawer. When any of the gadgets is activated, the corresponding RPL program is executed.

All these tools are completely implemented using RPL, and are provided also for demonstration purposes. Please feel free to make any improvements and changes so they suit your needs.

The Special Tools window consists of the following gadgets:

- ModifyCOGs
- ModifyACs
- ModifyNL
  
- FontLoader
  
- MeshToSph
  
- KeyFramer

### 1.2 modifycogs

#### MODIFY/COGs

##### DESCRIPTION

This is a file-bind gadget which means that whenever it is hit, the corresponding RPL program 'ModifyCOGs' is executed from the 'Macros' drawer. The RPL program simply defines a set of RPL words and loads a new Tool window whose gadgets are bound to

---

defined words.

The gadgets in the loaded Tool window correspond to the functions in the menu Modify/COGs.

## 1.3 modifyacs

MODIFY/ABOUT COGs

### DESCRIPTION

Loads a new Tool window from the drawer Environments and defines a set of RPL words which correspond to the functions in the menu Modify/About COGs. The gadgets in the window are bound to those words.

## 1.4 modifynl

MODIFY/NONLINEAR/MOVE

### DESCRIPTION

This function opens a Tool window which allows you to execute Modify/Nonlinear/Move functions. The gadgets in the window define the type of the control curve to be used.

## 1.5 fontloader

FONT LOADER

### DESCRIPTION

This function loads a new Tool window 'FontLoader' from the Environment drawer and defines a set of RPL words referred by gadgets.

FontDrawer

This gadget allows you to define the drawer from where the objects are loaded.

FontLoader

This gadget opens a string requester asking the user to type string. The function loops through entered characters and attempts to load objects of that name.

If the character in question is not capital, the prefix '\_' is added to it. Hence the string 'Hello' attempts to load the following objects:

H \_e \_l \_l \_o

Objects are positioned according their COGs and the distance between subsequent objects are determined by object sizes.

#### USAGE

1. Click the gadget 'FontLoader'
2. Click the gadget FontDrawer and select the AmigaDOS drawer from which you would like to 'multi' load your objects.
3. Click the gadget FontLoader and enter the string to be loaded.

## 1.6 meshtosph

#### MESH TO SPHERES

##### DESCRIPTION

This tool allows you to build up a skeletal object where a large number of spheres are controlled by freeform skeleton.

The function takes one or more freeform meshes and creates a given number of spheres which are evenly distributed over the mesh. Also, the hierarchy required for skeletal control is automatically built up so that when you modify the mesh, all spheres are modified as well.

##### EXAMPLE

1. Create one freeform mesh (like a human head).
2. Click the button MeshToSph and define the size and the number of spheres to be created.
3. Render the scene.

Modify the mesh (stretch, bend, etc), refresh the animation system by selecting the menu Animate/Refresh and render the scene again. Spheres are automatically pulled on to the stretched mesh.

##### NOTE

The spheres are created as 'wire frame invisible' in order to save memory. The freeform mesh is made RT-invisible so that when the scene is rendered, only the spheres can be seen.

## 1.7 keyframer

---

## KEY FRAMER

## DESCRIPTION

This example provides you with a simple icon based key framing interface.

The function loads a small Tool window, from the Environments drawer, which contains the following gadgets:

KF\_Start      This gadgets starts the key framing sequence. The current situation is considered as the first key frame.

KF\_Rec        This creates a new key frame object.

KF\_End        This stops the key frame sequence.

KF\_Help       Activates this help.

>>            Play forwards

>|            Go to end

<<            Play backwards

|<            Go to beginning

This tool window allows you to modify the target object of the MORPHING method and copies it to be the new key frame object every time you click the button KF\_Rec. The first key frame object is created when you start the keyframing sequence by clicking the button KF\_Start.

## EXAMPLE

1. Click the gadget 'KeyFramer'. This loads the key framing window and defines all required words referred to by the gadgets.
  2. Select objects to be animated and click KF\_Start.
  3. Select some modify functions and modify objects.
  4. Click the gadget KF\_Rec.
  5. Modify, click KF\_Rec, modify, click etc. until all keyframe objects have been created.
  6. Click KF\_End
  7. Play the animation by clicking >>.
-