
DYNAPAD: A BRIEF INTRODUCTION

JAMES D. HOLLAN

DISTRIBUTED COGNITION AND HCI LABORATORY
DEPARTMENT OF COGNITIVE SCIENCE
UNIVERSITY OF CALIFORNIA, SAN DIEGO

LAB WEBSITE: HCI.UCS.D.EDU
EMAIL: HOLLAN@COGSCI.UCS.D.EDU

DYNAPAD WEBSITE: HCI.UCS.D.EDU:DYNAPAD
DYNAPAD EMAIL: DYNAPAD@HCI.UCS.D.EDU

Introduction

*Dynapad*¹ is the third generation of our multiscale interface and visualization software. It makes scale a first-class parameter of objects, supports navigation in multiscale workspaces, and provides special mechanisms to maintain interactivity while rendering large numbers of graphical items. Dynapad employs Scheme to provide a high-level programming interface to the multiscale graphical and interaction facilities in the C++ rendering substrate.

Dynapad implements multiscale graphical objects (e.g., rectangles, lines, text, and images) that are interactive (e.g., they can be scaled or moved via user interaction) and dynamic (e.g., they can have behaviors that result from running of attached code). Behaviors can be associated with an object, a set of objects, or a region of the workspace and are triggered by user actions, the behavior of other objects, various events, or timer interrupts. Because Dynapad is based on a class system, objects in Dynapad can inherit characteristics and behaviors from other objects.

To create a Dynapad workspace simply type (*load "workspace.ss"*) in Scheme. This assumes, of course, that both the version of Scheme we are using, PLT-Scheme, and Dynapad have been properly installed. The default workspace, *dynapad* is an instance of *dynapad%* class. You can send messages to this instance to see or change various characteristics of the dynapad. For example,

¹The name *Dynapad* was chosen to reflect the software's heritage from our earlier Pad++ and STkPad software as well as ideas from Dynabook and Sketchpad.

the following code sets the background color to be *red*, the default *font* to Haeberli, and the default *pencolor* to *black* with *width* set to 3.

```
(define (my-defaults)
  (send dynapad background "red")
  (send dynapad defaultfont "Haeberli")
  (send dynapad defaultpen "black")
  (send dynapad defaultpenwidth 3))
```

Also any of these messages without the value will return the current setting. For example, *(send defaultfont)* will return the current setting of defaultfont.

There are numerous messages used to position and scale the view onto the workspace. To move the view of the workspace to a particular location one can use the *moveto* method. *(send dynapad moveto '(0 0 1))* moves the view to the default home location (x location 0, y location 0, and zoom of 1). To find the location of the view one sends the *view* message to a workspace: *(send dynapad view)*. *(send dynapad bbox)* will return the bounding box of the view. The first two numbers are the x and y coordinates of the top left corner and the last two numbers are the x and y coordinates of the bottom right corner.

You can send a dynapad a *zoom* message with a *zoomfactor*. This will zoom by the given zoomfactor centered on the current view. Thus, *(send dynapad zoom 2)* will zoom in by a factor of 2 and *(send dynapad zoom .5)* would then zoom back to the previous view. One can add to the zoom message an optional animation time. For example, *(send dynapad zoom 2 2000)* will take 2 seconds to zoom in a factor of two. One can also specify a location to zoom around. *(send dynapad zoom 2 3000 10 10)* will also zoom by a factor of two around 10 10 but take 3 seconds for the zoom.

One can center the view on an object or list of objects. If *objects* is a list of dynapad objects, *(send dynapad center objects)* will center the view so that all these objects are viewable. One can also control the time to animate to the centered location. For example *(send dynapad center objects 2000)* will take 2 seconds to animate to a location that centers the set of objects specified in the *objects* list. One can also specify that the centering is done in two steps (if the beginning and end are not too close together). In this case, the view zooms out so that both the start point and end point of the animation are visible and then zooms to the endpoint.

Other useful messages to a dynapad instance are *objects* and *selected*. The former will return a list of all the objects in the workspace and the latter will return a list of the currently selected objects.

Homework Problem

Write a procedure that finds all the objects on a worksurface and then centers each of the objects in the view taking three seconds to animate between objects.