

# Control Menus: Execution and Control in a Single Interactor

Stuart Pook<sup>1,2,4</sup>

Eric Lecolinet<sup>1</sup>

Guy Vaysseix<sup>2,3</sup>

Emmanuel Barillot<sup>2,3</sup>

<sup>1</sup> ENST/CNRS URA 820, 46 rue Barrault, 75634 Paris cedex 13, France

<sup>2</sup> Infobiogen, 7 rue Guy Môquet – BP 8, 94801 Villejuif cedex, France

<sup>3</sup> Généthon, 1 bis rue de l'Internationale, 91000 Évry, France

<sup>4</sup> stuart@acm.org

## ABSTRACT

We propose a new type of contextual pop-up menu called a *control menu*. These menus combine the selection of an operation and the control of this operation. They integrate up to two scroll bars or spin-boxes and thus allow users to keep their attention focused on the menu during the operation. Control menus can have sub-menus, and also retain the novice and expert modes of use found in marking menus. We describe control menus and how they are useful in different types of user interfaces. A program incorporating our control menu can be tested at <http://www.infobiogen.fr/services/zomit/>.

## Keywords

interaction, marking menus, menu access, interactors, gestures, user interface design, zoomable user interfaces

## INTRODUCTION

Standard menus such as pull-down, pop-up and marking menus [3] allow users to choose operations. Pop-up menus and marking menus are contextual: they are activated at a particular position in the user interface. The interface can thus adapt the menu's contents to that position and associate the action selected in the menu with the position.

The problem with pop-up and marking menus is that they do not allow the chosen operation to be controlled interactively. They also do not allow any parameters to be supplied. For instance, an operation such as a font size change often requires a dialog box to supply the new font size. Users have to use the menu and then switch their attention to another interactor. Once the new size has been entered the dialog box is dismissed and the users have to refocus their attention on the workspace.

Pan operations require either a dedicated mouse button (or a mouse button and a modifier) so that users can drag the image, or two scroll bars. Panning can not be done with standard menus except with impossible to use commands such as 'move a little to the right.' Zooming is another operation that is difficult to perform with menus. Users want to zoom until the correct scale has been obtained. Standard menus only allow them to zoom in fixed steps and then only via repeated uses of the menu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2000, The Hague, The Netherlands

Copyright 2000 the authors

## CONTROL MENU

We propose a new type of pop-up menu called a control menu (figure 1) that incorporates up to two scroll bars or

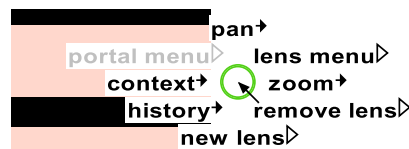


Figure 1: the control menu in our ZUI

spin-boxes. This menu not only allows operations to be selected, it also allows those operations to be controlled via the same interactor in a fluid and continuous way.

Control menu function somewhat like marking menus. The novice user presses (and holds) the mouse button, waits 0.3 second [3] until the menu is displayed centred on the current cursor position, and then simply moves the cursor in the direction of the desired operation. The menu disappears and the operation starts as soon as the cursor has been moved the *activation* distance from the centre of the menu. We found that an activation distance of five times the radius of the circle in the centre of menu works well. The operation is complete when the user releases the mouse button. Expert users make the same gesture as the novice but without the pause to have the menu shown. Users that do not need the menu are not distracted by it.

These menus are especially useful in Zoomable User Interfaces (ZUIs) [1] because of the large number of operations controlled with the mouse in these systems. We thus created a ZUI in which all operations are controlled by our menus.

## ONE SCROLL BAR

A control menu can be used to change the level of zoom in a ZUI. This operation illustrates the integration of a menu and a scroll bar into a single interactor. Figure 2 shows the user's mouse movements during the use of a control menu to select and control a zoom or dezoom operation. The user

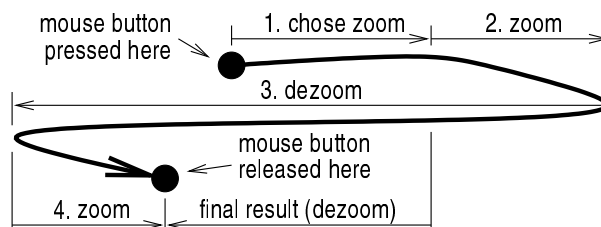


Figure 2: zooming with a control menu

presses the mouse button and moves the cursor the activation distance (movement 1 in figure 2) to the right (the zoom operation is on the right of the menu in figure 1). This starts the zoom and the on-screen cursor changes. From this moment on, mouse movements to the right (movements 2 and 4 in figure 2) zoom the ZUI and mouse movements to the left (movement 3 in figure 2) dezoom. The feedback is immediate: the view's scale changes as the user moves the mouse. The user releases the mouse button when the required scale change, a dezoom in figure 2, has been obtained.

During the zoom operation, the user can undo the zoom by moving the mouse up or down a large distance. The user can confirm the undo by releasing the mouse button or can undo the undo by moving the mouse back towards the centre of the display. In this case the user still has the mouse button pressed and can continue with the zoom. Not all operations can have a undo operation; only where one of the possible mouse movements does not already have a meaning is it possible to provide an undo in this way.

### TWO SCROLL BARS

A control menu can also be used to pan (or scroll) in two dimensions thus replacing two scroll bars. The pan operation is selected by pressing the button and moving the mouse up (figure 1 shows the pan entry at the top of the menu). During the operation the view follows the cursor.

Control menus work best when controlling two parameters that are integral. Integral parameters [2] are those whose attributes combine to form a single composite attribute in the user's mind. The x and y coordinates of an object are integral parameters because users combine them into the position of the object. A diagonal mouse movement has a simple meaning in this situation: move the object diagonally.

### SELECTING TEXT

Control menus can be used in more traditional text processing programs where one of the most common operations is selecting text. In this case, the select operation is placed at the right of the control menu (i.e. replacing the zoom operation in figure 1). In figure 3a the user has indicated the

- 
- (a) His tender heir might bear his memory  
But thou contracted to thine own bright eyes,  
Feed'st thy light's flame with self-substantial fuel,
  - (b) His tender heir might bear his memory:  
But thou contracted to thine own bright eyes,  
Feed'st thy light's flame with self-substantial fuel,
  - (c) His tender heir might bear his memory:  
But thou contracted to thine own bright eyes,  
Feed'st thy light's flame with self-substantial fuel,

Figure 3: selecting text with a control menu

start of the text to be selected by pressing the mouse button between the words 'might' and 'bear.' The user then moves the cursor the activation distance to the right. At this point the text between where the mouse button was first pressed and the current position of the cursor is selected (figure 3b). The user can then move the cursor down to select more text (figure 3c), left to select less text or up to select text from the new position to the end of the word 'might.'

Operations with parameters, such as font size changes, are decomposed into two steps. The text to be changed is selected via the menu and then the 'change font size' operation is selected via the menu. An operation such as 'cut' which does not require any parameters can be provided in the same way as a font size change (two menu accesses) or via a specialized menu command which allows the user to select text and then removes it at the end of the operation. Simple commands can thus be performed with a minimum of menu use.

### BUTTONS

Control menus can contain simple commands which do not require any parameters. Those simple commands that can be undone are performed as soon as the mouse has been moved the activation distance from where the button was pressed. As the user is still pressing the mouse button, a movement of the mouse in the opposite direction to that used to select the operation is interpreted as an undo. The undo can itself be undone by moving the mouse back. The operation only becomes definitive when the mouse button is released.

### SUB-MENUS

Control menus can have sub-menus. The user presses the mouse button and moves the mouse in the direction of the sub-menu. An operation in the sub-menu is then selected by changing the direction of the mouse movement towards the position of the operation in the sub-menu.

### CONTROL MENU VERSUS MARKING MENU

With marking menus the distance moved by the cursor is not important. Only the form of the gesture is important and is analyzed once the user releases the mouse button or stops moving the cursor. With a control menu the distance moved by the cursor is important. The position of the cursor is constantly analyzed and as soon as the cursor has been moved the activation distance from where the mouse button was pressed, the selected operation starts.

### CONCLUSIONS

We have presented a new type of menu that integrates the selection of operations from a menu and the control of their execution. They allow users to execute complex operations with a minimum of interactors. We think that control menus can usefully replace pop-up and marking menus in some types of user interfaces. Future work will integrate into control menus the selection of items from lists and the entry of simple text strings. These new menus will also be evaluated.

### REFERENCES

1. Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., and Furnas, G. Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. *J. Vis. Lang. Comput.* 7 (Mar. 1996), 3-32.
2. Jacob, R. J. K., and Sibert, L. E. The perceptual structure of multidimensional input device selection. In *CHI '92* (Monterey CA, USA, May 1992), ACM Press, 211-218.
3. Kurtenbach, G., and Buxton, W. User learning and performance with marking menus. In *CHI '94* (Boston MA, USA, Apr. 1994), ACM Press, 258-264.