# Time and gesture based interfaces a test case: Shinjuku Guide

Andrea Caloini (1)(2), Paolo Tosolini (3)

*(1) Dipartimento di Elettronica e Informatica Politecnico di Milano*
*p.za Leonardo da Vinci 32, 20133 Milano, Italy*
*tel. +39-2-2399.3666 - fax. +39-2-2399.3411*
*EMail: caloini@elet.polimi.it*
*(2) NEC central research laboratories*
*4-1-1 Miyazaki dai, Miyamae-ku*
*Kawasaki-shi, Kanagawa-ken, Japan*
*EMail: caloini@joke.cl.nec.co.jp*
*(3) University of Trieste*
*Via Bembo, 5 - 34015 Muggia (TS) - Italy*
*tel. +39-40-275030 - fax. +39-40-311850*
*EMail: tosolini@psicosun.univ.trieste.it*

## Abstract

The main point of this paper is to show how a gesture-based interface may become key interaction factor in human computer interface design when interaction depends on time. We show our point using a prototype tourist guide we developed as a case study. We also take this chance to present other sophisticated interaction techniques such as pie menus and direct object manipulation, and show how they can be realised with ToolBook.

## 1. Introduction

This paper shows how timing and gestures may become key interaction factors in human computer interface design; as a test case we present a prototype of a multimedia tourist guide (called Shinjuku Guide[1]) featuring a *gesture based interface, pie menu* and *direct object manipulation*. In this paper we also discuss the most relevant features of the interface of Shinjuku Guide.

The rest of the paper is organised as follows: in section 2 we discuss Shinjuku Guide navigation modes, in section 3 we describe the ToolBook implementation of Shinjuku Guide and in the final section 4 we draw some conclusions.

[1]Shinjuku is the name of an important Tokyo shopping area.

## 2. The Shinjuku Guide interface

Shinjuku Guide has three interaction modes (apart from traditional push buttons):

- ***jump*** **selects the place to visit from an aerial map**
- ***gesture*****, e.g. turn right, go forward)**
- ***manipulation*****, e.g. play a CD;**

different interaction modes are available depending on which part of Shinjuku Guide the user is viewing.

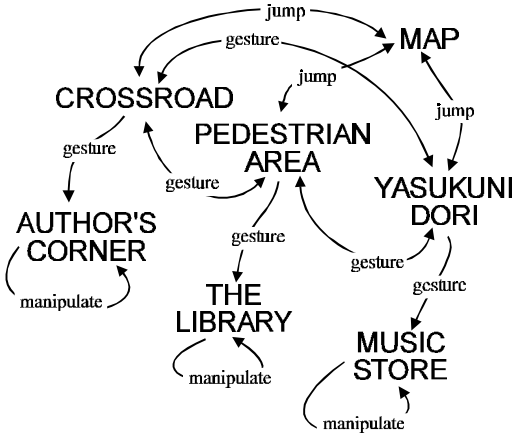


Figure 1. A (partial) representation of the structure of Shinjuku Guide showing how different navigation modes are used to move among the Map, the zones and the shops. (This figure does not show navigation operations activated by a traditional push-button interface.)

**Jump mode**

The user starts navigating in Shinjuku Guide from the Map, where some traditional clickable icons are placed; these icons let the user access (*jump to*) different zones in Shinjuku (e.g. "Walking area" or "Yasukuni avenue").

**Directional Navigation**

When the user jumps into a zone, a video of that zone is shown; for example, if the user clicks on the north ending of Yasukuni avenue, the application starts a video clip shot from a car running southward along the road. While a video of a zone is running, users can sometimes change motion direction (i.e. turn into a side street) or move towards some object displayed on the screen (e.g. walk into a shop); in both cases a new video will start. We also alert users about the possibility of changing direction showing pop up arrows with a descriptive name of the potential destination.

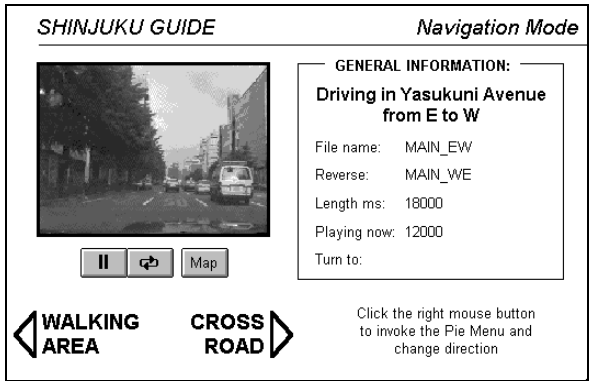


Figure 2. Shinjuku Guide's screen while driving in Yasukuni avenue

This kind of interface makes it quite intuitive to understand *when* it is possible to change direction and *what* the destination of the direction change is. However, timing can become a critical aspect of the human-computer interaction because users can only change direction while the "target" of the direction change is displayed on the screen (i.e. they might not be fast enough to change direction before the target disappears).

This kind of timing problems can be relieved by reducing the time the user needs to issue a direction change command: indeed in Shinjuku Guide it is possible to change direction *with a single gesture*.

Direction change commands are mapped onto the following gestures:

- **Go forward**[2]**:** right click, drag UP, release
- **Go backward:** right click, drag DOWN, release
- **Turn left:** right click, drag LEFT, release
- **Turn right:** right click, drag RIGHT, release

Users unfamiliar with gesture based interfaces are supported by a *pie menu*. A pie menu [1][2][3][4][5][9] is a special type of menu whose items are placed on the sectors of a circle rather than piled as rows in a rectangle (see figure 3). When a pop-up pie menu is invoked, it appears on the screen with its centre placed below the mouse cursor (figure 3a) and the user selects one of its items moving the mouse on the proper sector (figure 3b).

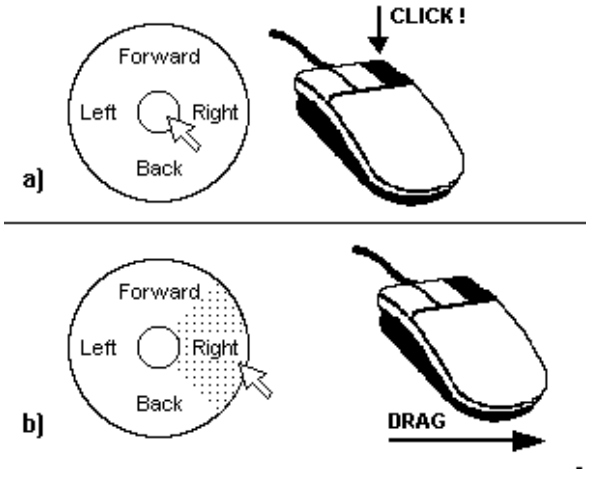


Figure 3. Bringing up a pie menu and selecting one of its items

Direction change commands are mapped onto the sectors of their pie menu as follows:

- **Go forward** TOP SECTOR
- **Go backward** BOTTOM SECTOR
- **Turn left** LEFT SECTOR
- **Turn right** RIGHT SECTOR

A key point is that the mouse operations needed to bring up the pie menu and select one of its sectors are the same mouse operations required by the corresponding gesture based command: indeed a fast selection from a pie menu corresponds to the gesture needed to issue the same change direction command.

In fact, as users become skilled at using the pie menu, their selection becomes faster and faster up to the point when the pie menu is not even displayed – i.e. they naturally get to the point of just doing a gesture!

[2] i.e. resume video after pause

**Direct manipulation**

When users enter into a shop they can use direct object manipulation to get some information on the shop itself; for example, when in a music store, users can listen to several tunes selecting a CD icon and dragging it over a CD-player. Of course, we have been careful to always give visual feedback in response to direct manipulation commands – for example opening the CD player lid when the disc icon is dragged close to it [6].

## 3. The ToolBook implementation

In this section we describe some of the problems we had to solve to implement the interface of Shinjuku Guide.

Jump navigation is implemented trivially because it just consists of active graphic objects placed on an image representing a map.

We could easily implement gesture-based commands as well; in this case we only need to detect where the right mouse button is pressed, where it is released and then compute the direction of the gesture assuming the mouse was moved along a straight trajectory. Computations are performed by a *rightButtonUp* handler, which is also in charge of executing the command associated to the direction selected by the user.

It should be noted that we can only discriminate among very simple gestures (straight mouse movements): the actual recognition of the shape of a mouse trajectory – such as a cross drawn on an object to delete it – would require more sophisticated support from the lower software layers (such as the windowing system or the operative system). However, as we noted earlier, if we can draw a metaphor linking a set of commands to a set of directions, we can use our gesture interface; for example, when we select a paragraph in a word processor, we might interpret the shift right (left) gesture as an align right (left) command and the up (down) gesture as an exchange position with previous (next) paragraph command.

Pie menus require sophisticated feedback to the user. Such feedback must be entirely provided by ToolBook scripts; they have to show and hide the pie menu graphic object and modify its appearance to show which item is currently selected (similarly to what happens with pull down menus). Moreover, pie menus are supposed to be displayed on top of everything, therefore we can not visualise a pie menu by just showing some graphic object stored into a background – because it would appear below any page object. What we currently do is to keep a pie menu master copy in the background and copy it in the foreground each time we enter into a new page (and delete it when we leave the page). The overall management of pie menus is therefore rather heavy but we currently have reasonably good performances even on 386SX-level machines – and blind selection, which is supposed to be the most frequent way of using pie menus, is quite satisfactory.

Direct manipulation, in the case of ToolBook applications, involves dragging (graphic) objects across the screen and dropping them somewhere. Because dragging is controlled by a script that keeps updating the position of the object while it is being dragged, the smoothness of the manipulation depends quite a bit on the speed of the machine being used.

## 4. Conclusions

Shinjuku Guide demonstrates how easy-to-use and natural (yet sophisticated) interaction techniques can be nicely integrated into an application without having to resort to expensive (and hard to use) interface toolkits. We also showed a (common) example where the use of a gesture based interface can greatly improve the usability of a multimedia application.

Our pie menu interface is available as a freeware and we hope this will help to spread its use.

## About the authors

**Andrea Caloini** currently works as researcher at NEC Central Research Laboratories, Kawasaki, Japan. He has got his PhD degree with a thesis on multimedia synchronisation and works in the hypermedia field since 1990.

**Paolo Tosolini**, undergraduate in International Political Sciences of the University of Trieste, Italy, has been visiting student in NEC C&C Labs in Tokyo, Japan researching on hypermedia applications using proprietary authoring systems. He is currently involved in the prototyping of multimedia interfaces within a project supported by the Italian National Research Board (CNR), for which he has been recently awarded by the Sigma Xi, The Scientific Research Society.

## Acknowledgements

The first prototype of Shinjuku Guide has been developed in NEC C&C Labs in Tokyo, Japan using a proprietary time based authoring system denominated Videobook [7][8][10]. For their valuable suggestions and opinions the authors want to thank Ryuichi Ogawa, Komei Harada, Rodrigo Botafogo and Natsuko Yamamoto.

## Bibliography


[1] Callahan J., Hopkins D., Weiser M., Shneiderman B., An empirical comparison of Pie versus Linear Menus, Proceedings of CHI '88

[2] Hopkins D., The Design and Implementation of the Pie Menus, Dr Dobb's Journal, December 1991

[3] Kurtenbach P.G., Buxton W.A.S., Issues in Combining Marking and Direct Manipulation Techniques, Proceedings of UIST '91

[4] Kurtenbach P.G., Sellen A. J., Buxton W.A.S., An Empirical Evaluation of some Articulatory and Cognitive Aspects of Marking Menus, Journal HCI, January 1993

[5] Kurtenbach P.G., Buxton W.A.S., The Limits of Expert Performance Using Hierarchic Marking Menus, Proceedings of CHI '93

[6] Norman A. Donald, The Psychology of everyday things, Basic Books Inc., NY, 1988

[7] Ogawa R., Harada K., Kaneko A., Scenario Based Hypermedia: a Model and a System, Proceedings of the European Conference on Hypertext '90, Cambridge University Press, 1990

[8] Ogawa R., Harada K., Tanaka E., Taguchi D., Design Strategies for Scenario-based Hypermedia: Description of its Structure, Dynamics and Style, Proceedings of the European Conference on Hypertext '92, ACM, 1992

[9] Tosolini P., La gestualità come principio di interazione con l'elaboratore, Nexus, March 1994

[10] Tosolini P., Shinjuku Guide: un prototipo di interfaccia ipermediale interattiva per utenti disabili, Technical Report CNR Progetto URMAD, 1993