Individual prototype

Christophe Pincemaille

November 13, 2008

Group: Desktop environment

University: Cork Institute of Technology
Department: Computing science
Module: Human Computer Interactions
Lecturer: Paul Rothwell

Contents

1 Introduction

For the HCI module, we have to make the prototype of a user interface of an application, after having studied it. Here, the application is a desktop manager (that is to say, part of an OS[1] GUI[2]). So this is the first drafts of the prototype.

2 Desktop environment

2.1 General presentation

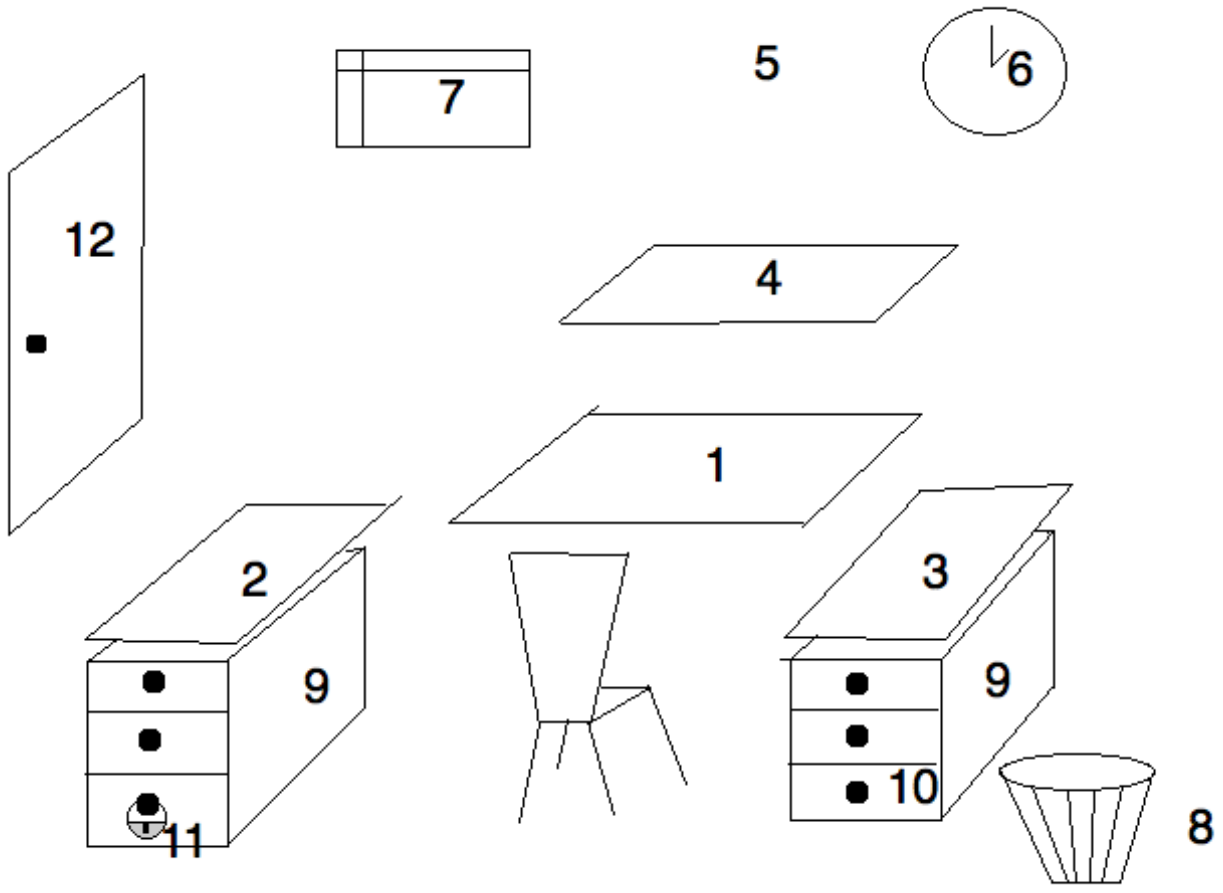

Figure 1: The view of the desktops

In the figure 1, we can see the view of all the virtual desktops. They are placed in a

[1]Operating System
[2]Graphical User Interface

3D environment. The goal is to make a huge metaphor, and to create in the user mind a clear mental model: the user should feel that this environment is just like the real world, just like his own desk with its own data. The user knows that it is virtual, but will be able to know where things are and how to use it without learn or recall. Here, recognition is more used than recall.

In that picture, we can see several elements:

1. First virtual desktop;
2. second virtual desktop;
3. third virtual desktop;
4. fourth virtual desktop;
5. widget space;
6. the clock widget;
7. the calendar widget;
8. the trash (empty, here);
9. drawer pieces of furniture;
10. one drawer, with the handle;
11. padlock;
12. corridor door.

2.2 Some important points to note

You can note that in our model, everything is useful, and is here in a certain goal. It is important not to have a graphical element here for nothing, just to make a beautiful, fancy and useless object. The desktops, trash, door, drawers... are obviously useful, and even the chair, which is not a working element, is useful to reinforce the mental model (make the user think that this is an actual office, with an actual desk, and that he works on it).

You can also note that nothing is written anywhere: no instruction, no warning, no information poster... Everything must be obvious and straightforward, directly understood (recognition rather than recall). The only written element that could be added is an "Exit" on or above the door (element 12 in figure 1). But first, an appropriate icon should to the same effect, and second, once used once or twice, the user may have understood, also because it is close to real world working.

2.3 Overview of constitutive elements

You can see on figure 1 that the environment is composed by several elements. The virtual desktops are included on the virtual desk, and are the tables of that desk. The widgets are pinned on the back wall, and the trash is near the desk, like in the real world. The trash will show some papers in it when it is not empty, and an empty bin when it is empty (this is the case in 1, the bin is shown as empty). In each side, there are drawers, in which are widget repository, applications repository, mount points of personal data... Note that there can be a padlock, to make the user figure out which data is personal (so protected), and which is not.

2.4 Access to further actions

To access to further actions, there has to be the possibility to make a right-click on element, that displays a menu with possible actions. A normal click in another location makes the menu disappear. Here, we take the example of the bin, in figure 2 (inspired from Mac OS X desktop environment), but it can be generalized to every elements that composes our desktop environment.

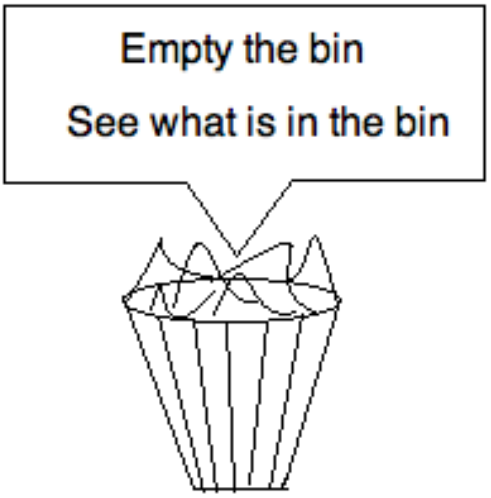

Figure 2: Access to further actions: right click

3 The desktops

3.1 Virtual desktops

As we said before, the virtual desktops are table of a 3D virtual desk. The number of desktops can be changed: the first desktop is like 1 on the figure 1, the second and third come on the sides like 2 and 3 in the figure 1, and the following ones are put in the same way, a level above (like 4 in figure 1).

3.2 Desktop switching

Actually, figure 1 represents the *dashboard*, called through a mouse movement, a key, or an icon. When doing it, the environment moves from the working desktop to this. Then, the user has just to click on the desktop where he wants to go. Then, the desktop is displayed and the dashboard disappears, like in figure 3.

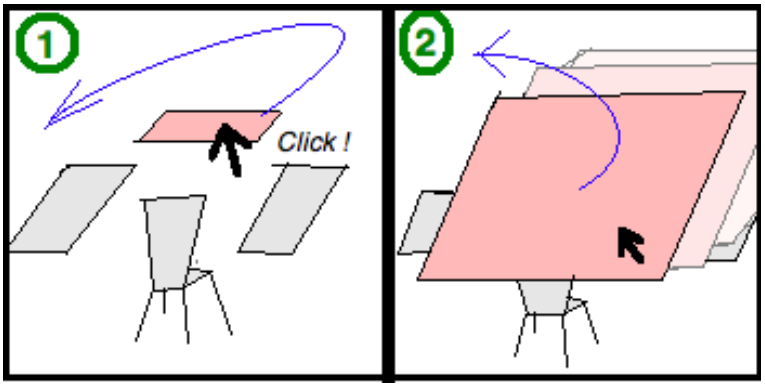

Figure 3: The desktop comes

4 Applications and widgets

4.1 Widgets: placement and moving

The widgets are on the back wall. Moving a widget is very straightforward (like in nearly every system with widgets): click on the widget, moving it by moving the mouse and letting the button be pressed, and when it is in the right place, release the mouse button (actually, drag-and-drop).

4.2 Installation

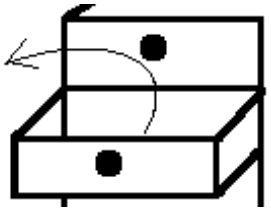

Figure 4: Install from a repository

To install an application, it is necessary to open the repository, and drag-and-drop it from the repository to the application folder. For a widget, it is pretty much the same,

except that the drop is done onto the back wall. The repositories are here represented as drawers, from which something can be picked, as in figure 4. The metaphor is still important here, and integration in 3D environment done.

4.3 Uninstallation

To uninstall an application or a widget, it is very straightforward: the user just has to make it move from its place to the trash, by using drag-and-drop, like in figure 5 (or maybe a keyboard shortcut, but this is for more advanced user, to go faster, because it integrates less in the metaphor).

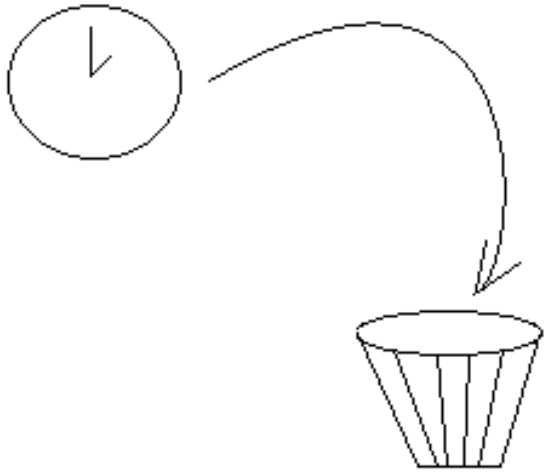

Figure 5: Uninstall (example of the clock widget)

4.4 Access to applications

You can note that in the 3D virtual environment, there is no application, no access to applications. Indeed, applications are not something real, they are just programs to make binaries / computer / electrical information work and display. So, we just consider here that, when a file is opened, the relevant application is automatically called, so that it would be transparent for the user.

In order not to limit advanced users, there must be an access to applications. First, applications are located in the "Applications" folder, so users can go to this folder in order to open a particular application, without opening a file in the same time. Besides, there would be the possibility for users to make links (hard and symbolic) and shortcuts, like in Linux system for example, in order to be able to open quickly applications. In that case, these links can appear in the desktops, but not in the 3D environment.

5 User switching

To switch of user, there is just to click on the side door (cf. 12 on figure 1). That logs of the user's session, and make him goes into the corridor, in which are doors which represent the offices of all the computer users (cf. figure 6). A new user has just to click on the good door to log on the new session.

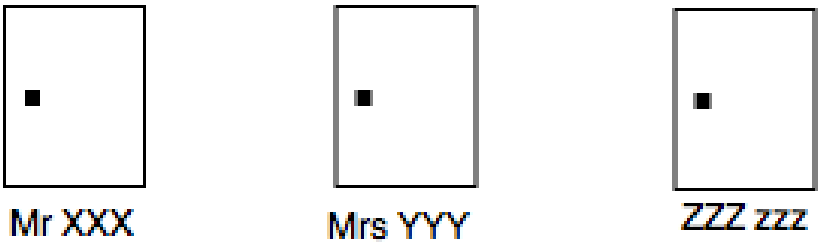

Figure 6: The corridor

6 File management

6.1 The mount points

At the start-up, hard drive are mounted in a "drawer". The metaphor is here important, because it is like for the real world, in which files and folders are stored in actually drawers.

Drawers can not be high level folders, because there are two major issues which prevent that. First, which folder level should be matched with drawers? Second, that will make the number of drawers very variable, and even variable during usage, which could be hard to represent.

Consequently, there are disks, which have to be mounted in drawers. This can cause problem with USB keys, or portable "plug-and-play" devices like that. So we can figure out one solution: having drawers "closed", or "inactivated" (the first solution is preferable as it includes itself better in the metaphor).

6.2 Access to the files : copy, move...

We can think of two accesses: one through a classic file browser, in the actual desktop, and one directly in the *dashboard*.

The first way to access the files is rather well known, so we will not spend a lot of explanations on it. The access itself, as the copy, move, ... functions are the same as in other systems like Windows, Linux, Mac OS... The file manager should look like the Mac OS X one, with some information hidden from the user (typically: absolute path of the current folder), but could be seeable, like in Linux, for advanced users.

The second way is based rather on the same idea as the *stacks* in Mac OS X (cf. figure 8 on appendix page I). Indeed, there is a space problem, in particular if there are a lot of files. This way to present the document can solve the problem: while opening the drawer, the files are shown like that. To continue on this idea, when there are too many files, they can be shown overlapped. The user only sees the top of the document, like if it was a tab. Putting the mouse cursor above the file will make it more visible (cf. figure 7 on page 8).

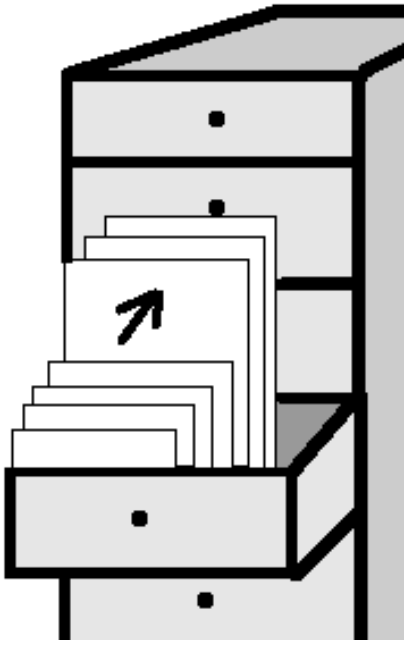

Figure 7: Access to the files

A Stacks in Mac OS X

Figure 8: Mac OS X: screenshot showing the stacks on the Dock