Multiple views map on Nintendo DS

Sorn Jarukasemratana

University of Tampere
Department of Computer Sciences
Computer Science / User Interface
Software Development
Master's thesis
April 2008

University of Tampere
Department of Computer Sciences
Computer Science / User Interface Software Development
Sorn Jarukasemratana: Multiple views map on Nintendo DS
Master’s thesis, 66+3 pages
April 2008

Information societies today are flooded with massive amounts of data. Moreover, with new technology being developed every day, machines can perform faster than they used to at an exponential rate. New processors which are faster and smaller result in a higher capability to produce more data and in a reduction of the size of the device. Even our capabilities to absorb information are still the same. The representation of data or information can still be improved through information visualization, that is to say, the knowledge of how to represent data or information by using interactive 2D or 3D color graphics and animations. By using this knowledge, many new methods for displaying information have been discovered especially for mobile device where screen space is very limited. This thesis contains a study of a multiple-view visual transformation technique on a mobile device, namely, Nintendo DS, by constructing a 2D navigational map program. I first compare several studies about different methods of visual transformation and explain why I chose multiple-view as my preference. To design the prototype map program I used the knowledge from the information visualization field especially information visualization framework. The implementation was done by using homebrew software development kits. The user study on how the technique works showed that the effectiveness and satisfaction score of the program is at medium to good level, while users’ preference for this visualization technique on this machine is very good.

Key words and terms: Information visualization, Nintendo DS, Multiple-view

Contents

1. Introduction

Information societies today are flooded with massive amounts of data. For example, the introduction of blogs has brought us the new world of online writing. Anyone with a minimal knowledge of computers can write about his or her own ideas, comments, or anything else online. In a matter of seconds, that writing can be viewed by everyone on the planet. Moreover, with new technology being developed every day, machines are performing faster than they used to be at an exponential rate. New processors that are faster and smaller result in higher capabilities to produce more data and to reduce the size of devices. However, our capabilities to absorb information have remained the same.

Due to these reasons, we need to develop ways to find the relevant information from the vast amount of data, and then to display that information in the most effective and efficient manner possible. A common problem for displaying information occurs when there is too much data to display on one screen. In particular, for small screens, the matter is more complex than for large screens. For example, a stock market chart can be displayed live on a normal personal computer or television screen without problems. However, if the same chart is to be viewed from a pocket pc or mobile phone, it is almost impossible to receive the same amount of information as that received from a bigger screen.

Thus, information visualization [Card et al., 1999] has become fundamental knowledge for human-computer interaction and displaying information. Information visualization deals with the knowledge of how to represent data or information by using interactive 2D or 3D color graphics and animations. By applying this knowledge, many new methods for displaying information have been discovered. One type of application that has received great benefit from information visualization is the interactive map.

Since the introduction of graphic user interfaces, interactive maps have continuously been developed by various companies and researchers, especially those researchers who focus on visualizing interactive maps. For example, Goal-Directed Zoom by Woodruff et al. [1998] offered an automatic zoom system. Users can choose which object to see and the view will be adjusted automatically. CZWeb by Fisher et al. [1997] offered a map of the WWW that shows where users have already visited and allows users to create their own “interactive internet maps” for themselves.

An interactive map system consists of two parts, the display part and the control part. Each part can have different designs depending on user needs and system limitations. The display part can be on a normal personal computer screen, a regular television screen, a huge plasma screen, or a small device such as a PDA, a mobile phone, a GPS device, or even a wrist watch. Many techniques have been proposed to help users access the information more easily and quickly, including fisheye view [Gutwin and Fedak, 2004a], multiple views [Baldonado et al., 2000], and zoom view [Plumlee and Ware, 2006]. As for the control part, there are many ways to control the interaction, depending on the input device of the system. The input device can be as simple as a mouse and keyboard, touch screen, or touch pen, or as advanced as a 3D pointer, speech recognition device, or gaze-based input device.

A problem that interactive maps face is when users want to display a map that is bigger than the screen size. This issue becomes more troublesome when interactive maps must be displayed on small and limited spaces such as mobile devices [Gutwin and Fedak, 2004b]. Normal computer screens range in size from around 1280x1024 to 1600x1200 pixels. On the other hand, mobile devices sometimes contain only 320x240 pixels or less. Visualization techniques are required to overcome this matter.

In the year 2004, Nintendo [http://www.nintendo.com] released a hand-held game console which may shed some light on the visualization problem with regard to interactive maps. The system is called “Nintendo DS” [http://www.nintendo.com/ds]. The Nintendo DS is a unique mobile machine that has two equally-sized screens, with the bottom screen being a touch screen. The built-in dual screens are perfectly suited for multiple views visualization techniques since one major problem of multiple views is the consumption of screen space [Schafer et al., 2002]. The size of both screens is 256x192 pixels, which is a little small compared to other mobile devices. However, with two screens, the Nintendo DS offers more total space.

It has been almost three years since the release of the Nintendo DS, but there have been few research studies about interactive maps on this device. Therefore, my purpose in this thesis is to demonstrate another possibility for interactive maps on small devices by using the Nintendo DS as hardware. The main idea of this interactive map is to have an overview map on one screen while the other screen displays the zoom map. Users can select which area is shown in the zoom map by simply touching the corresponding area in the overview map. Users can also drag the stylus around on the overview map, and the zoom map will move accordingly to the movement of the stylus.

Another objective of this interactive map is to provide more information on locations or landmarks in the map. For example, when users touch the shop icon, information about that shop will appear onscreen. This information can include such things as place name and opening hours. The idea for this project came from multiple views map programs, which have small overview windows and big zoom windows. I find this kind of map to be very useful and easy to understand because I can simultaneously see my general location within the map along with the details of my location. However, it is impossible to have two views on a normal mobile device (think about mobile phones, palm pilots, and pocket PCs) because the screen size and shape is not suited for supporting two maps. On the contrary, the Nintendo DS has two screens and is perfect for this kind of visualization.

First, in this thesis, I will introduce the main principles of information visualization, including definitions, needs, goals, and reference models. After that, I will consider the definition of interactive map, which is the main program of this thesis. In addition, I will examine the subcategory of interactive maps that consists of navigational maps and explain the elements of navigational maps. Then, I will discuss and compare previous research studies on popular visualization techniques that made use of navigational maps, including panning, fisheye view, zooming, and multiple views.

In the next part of this thesis, I will review the Nintendo DS, the handheld mobile gaming device with two screens. The dual screen system is a special feature that separates this small gadget from other small devices in the market. Following the overview, I will discuss the process of designing the prototype program, which consists of the interactive map on the Nintendo DS. Information about implementation will also be included. Lastly, I will evaluate and discuss the test results on participants, and provide a conclusion.

2. Information visualization

In this chapter, I will introduce the main principles of information visualization, including definitions, needs, goals, and reference models.

2.1. Definitions

Information visualization is a term that has been defined by many people. For example, Card et al. [1999] writes that "information visualization is the use of computer-supported interactive, visual representations of abstract data to amplify cognition." The famous scientist Herbert A. Simon also writes that "information visualization is a process of transforming data and information that are not inherently spatial, into a visual form allowing the user to observe and understand the information." However, no matter the definition, information visualization is meant to help users explore, make sense of or discover the piece of information they need from a vast amount of total information.

Information visualization falls under a branch of computer graphics and user interface design. Active research areas encompass information graphics, computer graphics, human-computer interaction and cognitive science.

2.2. Needs and goals

The need for better visualization comes from the limited cognitive abilities of humans. Information visualization which is count as external cognition is the knowledge that teaches us to use cognitive items to amplify our cognition. The obvious example is to try to multiply many-digit numbers with or without using pencil and paper. The latter is almost impossible. Although the multiplication itself is not hard, the hard part is to remember the partial results.

However, the goals of information visualization are more specific than external cognition. There are three goals of information visualization: discovery, decision making and explanation. For the purpose of discovery, explorative analysis is used. Normally, there is no initial hypothesis about the data. The results of this analysis will provide overall information to better understand the data set. As for the purpose of decision making, confirmative analysis must be conducted. After obtaining a hypothesis about the data, the next step is to look for what users need. The process will focus on determining if the hypothesis is correct and will result in confirmation or rejection of the hypothesis. The last purpose is explanation. For this purpose, the appropriate visualization technique must be used in order to point out certain facts about the data set.

2.3. Reference models

Reference models are important tools for researchers that provide standards to help everyone understand information visualization in the same way. There are several reference models for information visualization, such as the reference model by Spence [2001] and the reference model by Card et al. [1999].

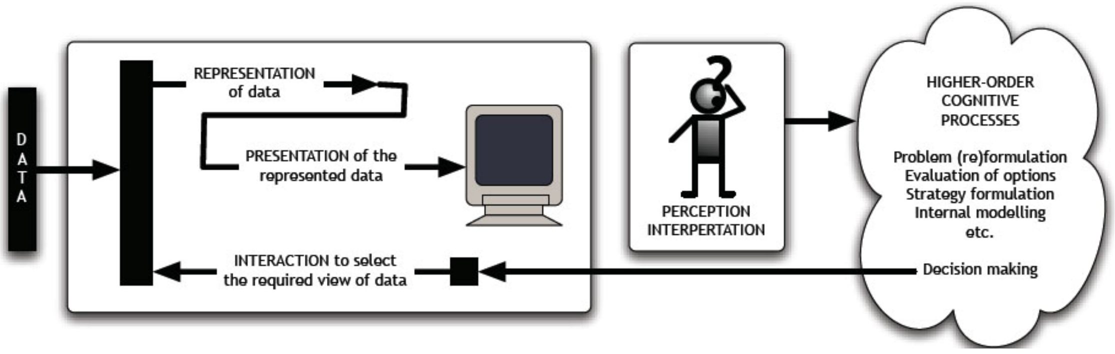

DATA
REPRESENTATION of data
PRESENTATION of the represented data
INTERACTION to select the required view of data
PERCEPTION INTERPERTATION
HIGHER-ORDER COGNITIVE PROCESSES
Problem (re)formulation
Evaluation of options
Strategy formulation
Internal modelling
etc.
Decision making

Figure 1. Reference model by Robert Spence [2001]

In the reference model by Spence, there are four separate processes: selection, encoding, presentation and interaction. Selection refers to the entry of data into the system. In the diagram, encoding refers to the "representation of data" in specific forms, mostly in visual forms. Presentation refers to the method of displaying encoded data in an appropriate way. And lastly, interaction refers to the actions undertaken by users to change or manipulate the view of data.

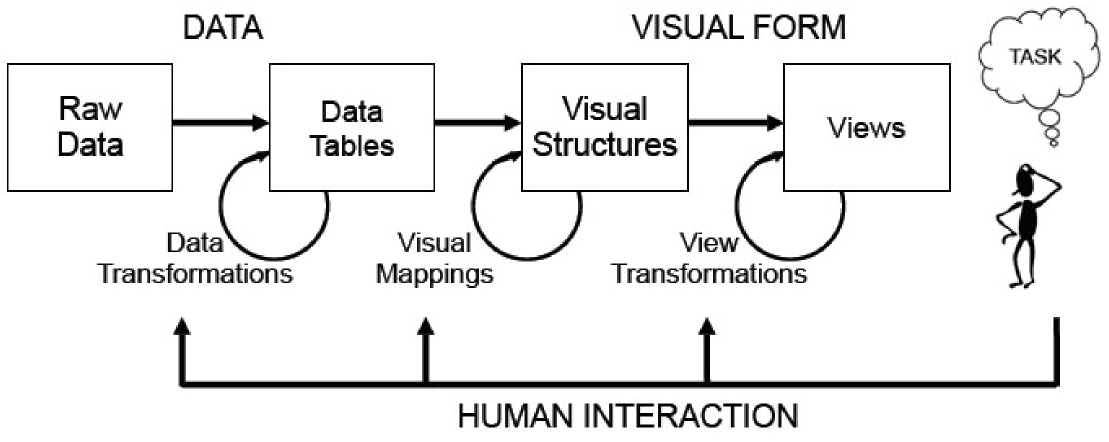


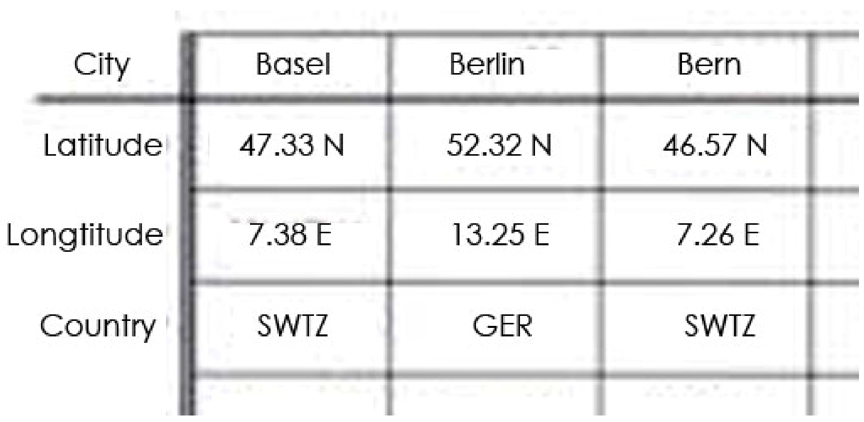
Figure 2. Reference model by Card et al. [1999]

In my opinion, the reference model by Card et al. is easier to understand in comparison to the model by Spence, because in this model, actions and objects are displayed separately. However, the contents of the two models are the same. More specifically, the process of data transformations in the model by Card et al. is equivalent to the process of encoding in the model by Spence. Likewise, visual mappings and view transformations are equivalent to presentation. Finally, interaction is the last process in both models.

2.3.1. Data transformations

The first step of information visualization is data transformation. In this step, raw data will be verified, categorized, ordered, and mapped into a data table. Raw data can be any kind of data. Normally, raw data will be in a form that is hard to understand, which prevents users from obtaining insights or answering questions related to the data. Therefore, the raw data must be transformed into a set of relations and then put into a data table. The data table will then contain organized raw data as well as metadata used for describing the relations between data values.

Raw data can be mapped into a data table in an unlimited number of ways, depending on which variables are the main points of interest. Any time a data table is changed, new information will be gained, while some will be lost. The figure below depicts a sample data table of cities. Latitude, longitude, country and city are metadata.



City	Basel	Berlin	Bern
Latitude	47.33 N	52.32 N	46.57 N
Longtitude	7.38 E	13.25 E	7.26 E
Country	SWTZ	GER	SWTZ

Figure 3. Example of Data Table

2.3.2. Visual mappings

Visual mapping, the second step of information visualization, is the process that transforms data tables into visual structures. Visual structures will be what users see on their screens, so it is very important to make them easy to read and understand. There are many ways to map a data table into a visual structure, depending on the needs and goals of the users. A basic example is a spreadsheet program that allows data tables to be converted into graphs. One table can be converted into various kinds of representations, such as a bar chart or pie chart.

Visual structures have three basic points of concern: spatial substrate, marks, and graphical properties of marks. Spatial substrate refers to the use of space. Space between each data entry can contain information on its own. The most basic form of spatial substrate is a scatter plot, which contains qualitative X and Y axes. There are several techniques related to spatial substrate, including alignment, folding, recursion, and overloading. Each technique uses space to create meaning in some way.

Marks, in my opinion, are the most important part of visual structures since they are the things that users will see in the space. Marks can be points, lines, areas, volumes, pictures or any other objects on the screen. Graphical properties of marks include size, value, orientation, texture, shape, position and color. These properties make each mark meaningful and different from others. However, not every property is appropriate for every kind of data. For example, using shape to represent numerical data is quite pointless because users will not be able to easily determine the exact values of the data or quantitatively compare the differences between the marks. On the other hand, shape works well with nominal data types.

2.3.3. Visual transformations

Visual transformation, the third step of information visualization, is the process that creates the views for visual structures. Visual structures themselves are static pieces of information. However, by viewing static visual structures from different angles, users can receive new information from the same visual structures. The underlying method of visual transformation is to change the graphical properties of marks – by repositioning, rescaling, distorting views, clipping views, and so forth – in order to make a visual structure look different in a better way. The three common view transformations are location probes, viewpoint controls, and distortions.

Location probe is a kind of transformation in which the user's point of interest serves as a guideline to pinpoint which part of the visual structure should be transformed. There have been many research studies that use this transformation technique. For example, iDict [Hyrskykari et al., 2000], a gaze-aware reading aid program, can give extra information on a word that users stare at for a certain period of time.

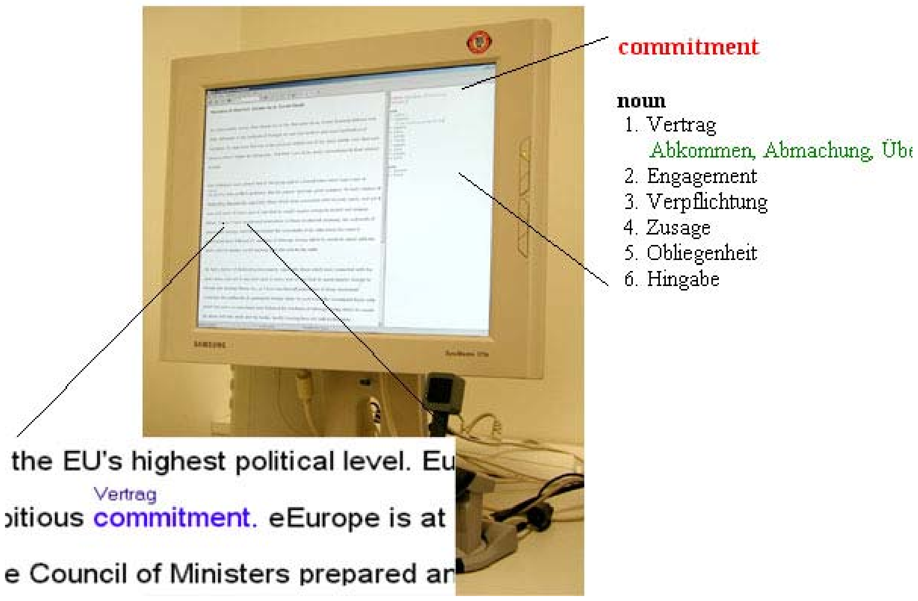


Figure 4. iDict, a Gaze-Aware Reading Aid [Hyrskykari et al., 2000],

Viewpoint control is used to move the viewpoint to create visualization. The common viewpoint controls are panning, zooming, and clipping. Panning is the movement of the viewpoint over the visual structure area, which is normally used when the data area is bigger than the screen size. Zooming is the act of closing in on, or backing out of/adding distance to, a certain area of visual structure. Clipping means removing parts of the display elements that stay outside a given boundary, usually a window or viewport. These three methods are used heavily in almost every type of program. For example, Open Office Writer contains all of these control features.

Distortion provides a partially-distorted image of the visual structure to create a focus area and context area. Distortion has a different objective from the other transformations because it makes visual structures look different than normal. This property makes distortion very complex to use since users may not understand the distorted view. The most common distortion tool is the fisheye view [Gutwin and Fedak, 2004a].

In addition to the three main concepts of visual transformations, there are several other methods that count as transformations, such as multiple views [Plumlee and Ware, 2006, Baldonado et at., 2000, Hornbæk et al., 2002]. Multiple views transformation offers two or more different views of the same object. 3D graphic users are certainly familiar with multiple views systems that show 3D objects from different angles. For map interfaces, zoom and overview windows are normally used.

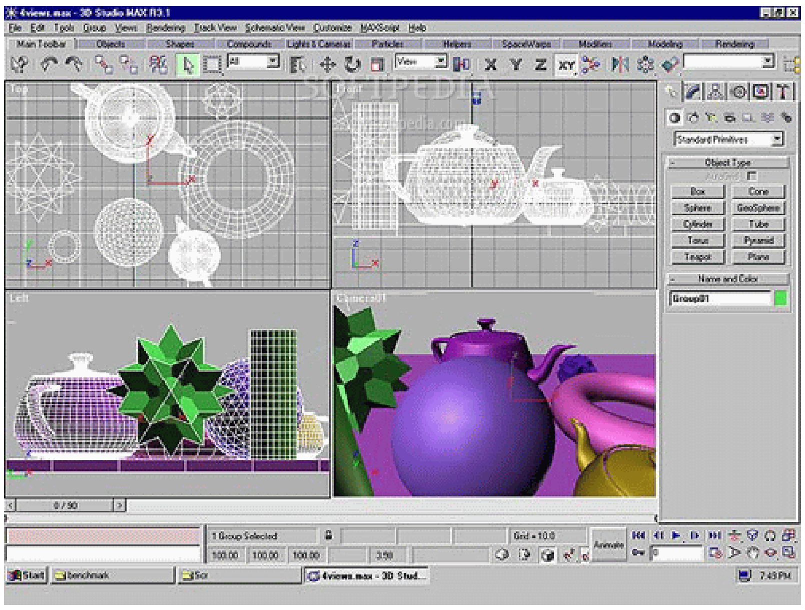


Figure 5. Multiple views in 3D studio max

2.3.4. Human interactions

Human interaction constitutes the last portion of the model. However, the model cannot begin without human interaction as well. Human interaction is actually required throughout the entire process: from picking the raw data, to organizing the data into visual structures, to selecting the views for visual structures, and finally to manipulating the views to gain information about different aspects of the data.

Unlike the other steps in the reference model, human interaction does not fall under the category of computer graphics but mainly falls under the fields of human-computer interaction and cognitive science. In my opinion, this part of the model is mostly neglected in comparison to the other three parts, especially in research fields where there is no need for efficiency and ease-of-use in visualizing the programs. However, this lack of concern creates a big pitfall in a lot of commercial software.

All in all, the reference model just provides a guideline for how to construct information visualization. In real practice, software design and software engineering may be needed. Nevertheless, in each round of iteration, it is advisable to consider every step of the information visualization reference model.

3. Interactive map

In this chapter, I will first consider the definition of interactive map and then introduce each component of interactive maps as it relates to the reference model of information visualization. Afterward, I will explain the role that information visualization plays in interactive maps, and I will discuss which transformation techniques can be used and will be investigated. In the last part of this chapter, I will compare each technique to determine the appropriate method for use in this thesis.

3.1. Definitions

The term "interactive map" can have many meanings, depending on the field in which it is used. To be specific, the definition of each word in the term will be discussed.

The term "interactive map" consists of two words. I will start by looking at the definition of each word separately. Interactive in the computer science field means "interacting with a human user, often in a conversational way, to obtain data or commands and to give immediate results or updated information" [http://dictionary.reference.com]. The definition of interactive is quite clear and does not need to be discussed further. This definition implies that there must be a human user, a method of communication, a target with which the human user can interact, and an immediate updated result. In this thesis, the target of interaction is the mobile device called Nintendo DS. The methods of communication are through the user interface that the machine provides, which includes stylus, touch screen, and buttons. Immediate results and updated information will be displayed on the device's screens.

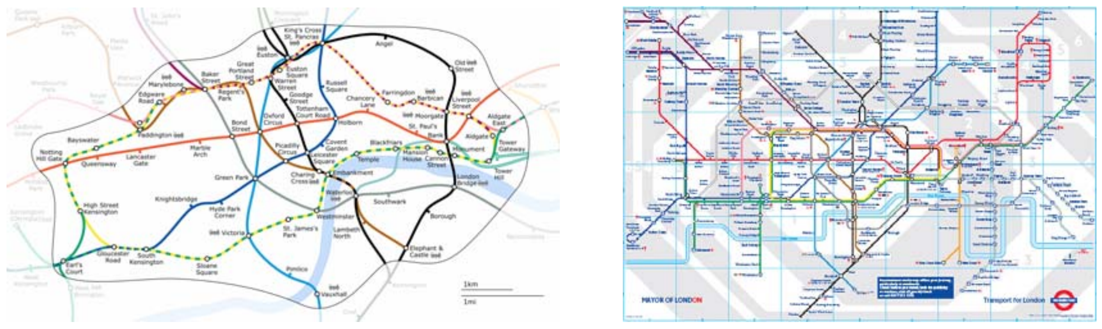
On the other hand, the definition of map can be complex because it can refer to many things. One definition is that "a map is a visual representation of an area – a symbolic depiction highlighting relationships between elements of that space such as objects, regions, or themes" [http://en.wikipedia.org/wiki/Map]. *In this thesis, map will refer to a 2D geographical map, particularly a navigational map.* It is very important to stress this point, because this type of map has certain constraints and needs that can lead to further problematic interactive issues. The most common type of geographical map is a navigational map, such as a road map, nautical chart, hiking trail map, or railroad network map.

3.2. Navigational map

Geographical maps fall under the category of Cartography, the study of map-making in which maps are created to represent the earth as flat areas. Navigational maps comprise a subset of geographical maps that focus heavily on providing traveling directions. There are three main concerns with regard to navigational maps: orientation, scale and accuracy.

Orientation refers to the direction of a map in relation to the 4 axes of a compass. In ancient times, east was the most popular orientation [http://en.wikipedia.org/wiki/Map], since it is the direction of sunrise. Nowadays, the most common but not exclusive orientation is north, thanks to the introduction of compasses. Nevertheless, there are several kinds of maps that have different orientations. For example, a map of a building or small area may use landmarks as the main tools to define orientation. Furthermore, a map of a department store or museum often places the entrance at the bottom of the map. When customers view the map at the entranceway, they easily understand that the path ahead of them leads up in the map. Indeed, the most important point of orientation is to help map users know the direction in which they are heading.

Usually, geographical maps are drawn to scale. For example, a topographic map of the United States by the United States Geological Survey (USGS) [http://www.usgs.gov] is at a scale of 1:24,000. In terms of the reference model of information visualization, a scaled map represents a visual structure that uses spatial substrate as a basis for visual transformation. However, not all maps require scale. One good example is the London underground train map or tube map [http://www.tfl.gov.uk]. The underlying rationale is that people who use the underground train do not care about the distance between each station or the number of turns in each route. They just want to know what the next stations will be and when the trains will arrive. Because user needs are simple, there is no need to have scale.



(a) (b)

Figure 6. (a) Geographical tube map (b). Tube map with out scale [http://www.tfl.gov.uk]

Accuracy is another vital requirement for geographical maps. Maps would be useless if they could not provide accurate information. However, as with scale, not all users need complete accuracy. Sometimes, reducing accuracy can lead to better or faster understanding of a map. For example, in road maps of residential and commercial areas, the roads are sometimes drawn larger than scale. Consequently, roads may look broader on the map than they are in actuality. However, map users prefer this method of depiction because routes are more clearly visible even though accuracy is sacrificed. Furthermore, most goal-specific maps exclude trivial objects from their visual structures in order to make the maps cleaner and easier to understand.

3.3. Elements of interactive maps

Interactive maps consist of two main elements: the map part and the interactive part. The map part, which is made by a cartographer, counts as data, data tables and visual structures in the information visualization reference model. This element will not be discussed in this thesis as it does not fall under the field of computer science.

The other element of interactive maps is the interactive part, which can be further separated into the display part and the control part. In terms of the reference model of information visualization, the display part counts as views and the control part counts as human interactions.

3.3.1. Display part – views and views transformation

Generally, special methods are required to view a map or other large-scale 2D content in its entirety. Many visualization techniques have been applied to view content that cannot fit onto one screen. The most well-known techniques are panning, zooming, fisheye view and multiple views. These interaction techniques are used in many research studies in this field.

3.3.1.1 Panning and scrolling

Panning and scrolling are perhaps the most common methods used to view large content. In panning, the idea is to move the viewport over another part of the content or to move the content under a fixed viewport. Basically, panning allows for movement in both the X and Y coordinates. Scrolling is a special case involving a scroll bar that acts as visual guidance and tools to move the viewport along the bar's direction.

The main problem of panning and scrolling is that the user cannot see objects that do not reside under the current viewport [Smith and Taivalsaari, 1999]. This problem makes panning and scrolling less appealing to users [Gutwin and Fedak, 2004b].

Scrolling has a visual cue advantage over panning due to the scroll bar. The presence of the scrollbar lets users know that it is possible to scroll the view. Furthermore, the position box in the scroll bar indicates the “current position” of the view relative to the entire content area. This visual cue is called affordance [Amant, 1998].

Affordance is a hint to users for how they can use an object. A simple example is a handle. When users see a handle, they instinctively know that they must grab the handle and move it. This affordance of real-world objects can be applied to user interface elements as well.

There have been many research studies about how to implement better scrolling. Smith and Taivalsaari [1999] offered an alternative definition of scrolling in mathematical terms as the "changes of salience of a group of display objects while preserving their relative positions." According to this definition, when users change the value of the scroll bar, not only could the position of the viewport change, but some aspects of user interface could change as well. In other words, users could scroll through some properties besides viewport.

A good example is position scrolling. As seen in Figure 7, when the scroll button is moved to the top of the scroll bar, icons gather at the top half of the screen in a compressed manner, which emphasizes the remaining icons in the bottom half of the screen. In their research, Smith and Taivalsaari demonstrate various other implementations of scrolling based on property changes, such as stationary scrolling using brushing and fisheye techniques.

Figure 7. Position scrolling

Panning can be implemented in many ways. One way is for users to move the viewport over a static object. A second way is for users to move the object under a static viewport. A third way is for users to indicate which direction the viewport should move by touching the screen or moving the mouse toward the edge of the screen. A fourth way is for users to indicate which direction the object should move by touching the screen or moving the mouse toward the edge of the screen. According to Johnson [1995], the ideal method of panning in terms of effectiveness and efficiency of visualization will vary depending on both the input hardware and individual user. With a keyboard or directional pad, users perform best when the camera moves according to the key that is pressed. For example, pressing the "up" key should move the camera up (which is equivalent to moving the map down), allowing users to view areas to the north. However, with a touch screen display, Johnson concluded that the object should move according to the direction of drag. For example, touching the screen and dragging to the left should move the map to the left (which is equivalent to moving the camera to the right), allowing users to view areas to the east.

In panning, affordance is a problem because visual cues will mostly be lacking. This problem was also reported in Johnson's research. At first, users did not know how to manipulate a map on a touch screen display. They had to guess how to use the system by trial and error. Attempts at interacting with the touch screen display were based on the past experiences of individual users. Some users touched the screen and dragged the map, while others touched the edge of the screen or touched a position on the screen in hopes of centering the view on that position.

3.3.1.2 Zooming

When content is very large, panning takes too much time and becomes a boring task. Zooming can provide a quick impression of the whole content when zooming out and can give details when zooming in. The zoom level can be set dynamically by the user, or it can vary automatically depending on what is appropriate for the task at hand. In terms of overall performance rate, zooming is better than panning. [Gutwin and Fedak, 2004a]

The zoom interface was notably introduced by Perlin and Fox in 1993, when they released an early computer interface model called PAD. Although there is no clear definition for a zoom interface, there are two main characteristics of such a user interface. First, the data objects are displayed in space. Second, users interact with the data space via panning and zooming. In zoomable user interfaces, space and scale are used for navigating the information. Generally, the size of data objects that users perceive on the screen is based on scale. The most common zooming technique is geometric zoom, where scale is linearly translated to the size of data objects. A more advanced technique is semantic zoom, where scale can affect various properties perceived by users, such as size, alpha value and the number of objects that are displayed onscreen.

Users interact with the data space via panning and zooming. Panning changes the part of the data space that users see. Zooming changes the scale and thereby changes the amount of data objects that users see. In normal user interfaces, panning and zooming are controlled by a mouse and keyboard. In other user interfaces, alternative input devices like touch screens and gaze-based devices can provide different ways of controlling zoom. The control can be linear or non-linear. With linear control, users can interact with the input device to directly adjust the pan and zoom settings according to their commands. For example, they can press the "up" key on the keyboard to cause the camera to pan up. On the other hand, with non-linear control, users do not directly adjust the pan and zoom settings. They can issue other orders to system, and the system will then automatically adjust the viewport and scale to the appropriate settings. There are several non-linear zooming methods. For example, goal-directed zoom [Woodruff et al., 1998] allows users to click on an object in the visualization, and the viewport will move to that object and zoom to the appropriate level.

There are two ways to implement zooming, jump zoom and animated zoom. In jump zoom, the change of scale occurs instantly. Jump zoom is used in PAD and many commercial programs, including Macromedia Flash and Adobe PhotoShop. The criticism about this method is that users who are not familiar with the data space may get lost in between the different levels of data. This problem may force users to scale out to the overview level. In animated zoom, the transition from the old scale to the new scale is smooth. Two important issues in animated zoom are the duration of transitioning and user control over zoom speed.

Guo et al. [2000] concluded that zoom speed should be about eight factors per second, which means that when users zoom, they will see the map getting bigger or smaller eight times per second. This zoom technique is very effective for maps that provide zooming over a large scale, such as zooming from satellite pictures to specific locations on earth. Users will be able to perceive how much they have zoomed from the amount of time that the zooming process takes. However, Card et al. [1991] proposed that zoom time should be approximately one second in all cases. To determine the effect on users, Bederson and Boltman [1999] conducted a study on whether animated or jump zoom was better for 20 subjects. The subjects were able to remember better with animated zoom, but there was no difference in task completion time or user satisfaction between the two methods.

3.3.1.3 Fisheye view

Fisheye view is a technique that may not be familiar to most people [Schafer et al., 1999], but it is extremely useful and yields probably the best performance in viewing large content [Gutwin and Fedak, 2004b]. Interactive fisheye view allows users to view or edit content via a distortion lens while providing a view of the whole data space on one screen [Gutwin and Fedak, 2004a]. The interface can be set to full-screen or constrained to some part of the screen. However, in a map system, which is essentially a layout of graphical objects, fisheye view can be problematic because distortion will make judging distance, angle and alignment difficult.

Although fisheye view may not be familiar to normal users, there is a wide range of program types that support this technique, including graph visualization tools, document creation software, and website browsers. Fisher et al. [1997] conducted research on the usefulness of fisheye view-based website browsers. They proposed CZWeb, Continuous Zoom-based Web navigation aid, which displays a network of visited web pages through a mouse-hover fisheye interface. In this program, a page node represents a web page while a cluster node represents a cluster or group of web pages. Users can store their browsing history in CZWeb to create their own "interactive internet maps". Feedback from users was very good. They had no problem understanding and using the fisheye interface even though they had never experienced it before.

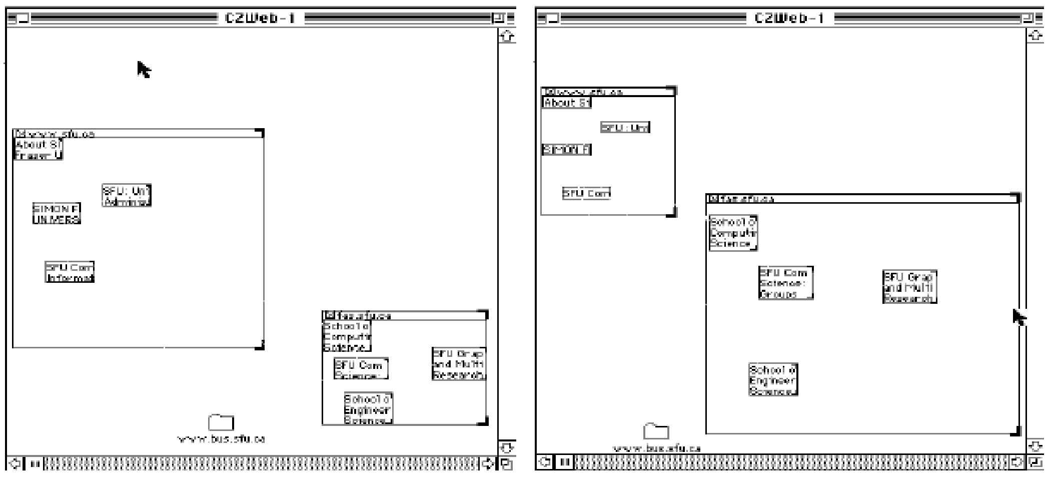


Figure 8. CZWeb's fisheye feature [Fisher et al., 1997]

3.3.1.4 Multiple views

A multiple views system uses two or more windows to view and give information about one or more objects, in order to gain insight from the visual structure [Baldonado et al., 2000]. Each window can focus on the same object from a different angle, as in 3D modeling programs. But not only can different viewports be presented in each window, sometimes different visual transformations can be applied to each window as well. For example, 3D modeling programs can simultaneously display the skeleton, the wire frame, and the fully rendered view of an object in separate windows (Figure 5). In many cases, each view might not even be showing the same object or content. A good example is computer-aided design or CAD. The designer could be looking at the design object in one window, while another window could be displaying a text file explaining the design parameters.

Multiple views systems have been studied by various researchers in many aspects, and the usefulness of such systems have been established in many research papers. For example, North and Shneiderman [1997] observed that multiple views offer better performance and reveal unforeseen relations. Furthermore, Schafer et al. [2002] reported that these systems yield better user satisfaction from testers than pan and zoom systems.

However, multiple views also come with certain drawbacks. One clear disadvantage of this technique is the requirement for space. As the name implies, multiple views require at least two windows for the interface, whereas other techniques require only one. This problem is critical in small display devices such as mobile phones and PDAs, where screen space is limited. Another concern is the potential overload of information on users. However, this problem might not be serious because users might just need more time to digest all the additional information that they see. In an experiment by Hornbæk et al. [2002], maps with an overview feature, which constitute a subset of multiple views systems, were compared to maps without an overview feature. For certain tasks, the researchers found that participants achieved greater accuracy using maps with an overview feature, although the participants also took longer to complete those tasks on average.

3.3.2. Control part – human interaction

In addition to the display, another important element of interactivity is the controller used to navigate the map. Controllers range from simple technologies (such as a mouse and keyboard in normal PCs, a directional pad and number pad in mobile devices, or a touch screen) to advanced technologies (such as a speech-based or gaze-based interface). These advanced interfaces have appeared frequently in research in the navigation visualization field. used

Selecting the proper control for a navigational map depends on the type of visualization being applied and the type of machine being utilized. However, for map navigation purposes, I will divide controllers into two categories, digital input and analog input.

Digital input refers to input that gives the machine a value of zero or one. An example of this kind of device is a keyboard. When users press the "A" key, the value of the function "A is pressed" in the system changes from zero to one, which causes the letter "A" to appear onscreen. Digital input devices include all button-based input devices, such as a television remote control or mobile phone pad. There are several advantages of digital input over analog input. One advantage is that digital input devices are easier to use. For example, many new computer users have a lot of difficulty in moving a mouse pointer to a specific area of the screen, especially to click on a small button. On the other hand, pressing a button on the keyboard is much simpler. Another advantage is that digital input has a lower error rate, although this point may depend heavily on the design of a device. In general, pressing a single button is less prone to error than moving a mouse or a slide bar to an exact position. Speed is also another advantage when comparing digital input to analog input. For digital input, users can only choose a value of zero or one. However, for analog input, users can choose any value between zero and one, inclusive.

Analog input refers to input that gives the machine a value between zero and one, inclusive. Examples of this kind of device include mouse, touch screen, radio tuner, and bathtub water control knob. Analog input devices can give a wide range of values.

Therefore, despite the many advantages of digital input, analog input is needed when users require range-type input. A common need for analog input is in bathtub water control. With simple digital input, users will only be able to turn the hot water on or off, resulting in overall water temperature that is either too hot or too cold. However, with analog input, users can control the level of hot water and overall water temperature more precisely. Some controllers provide both analog and digital input side-by-side, such as Sony's game controller (Figure 9). The controller has several digital input buttons as well as two small analog sticks in the middle.

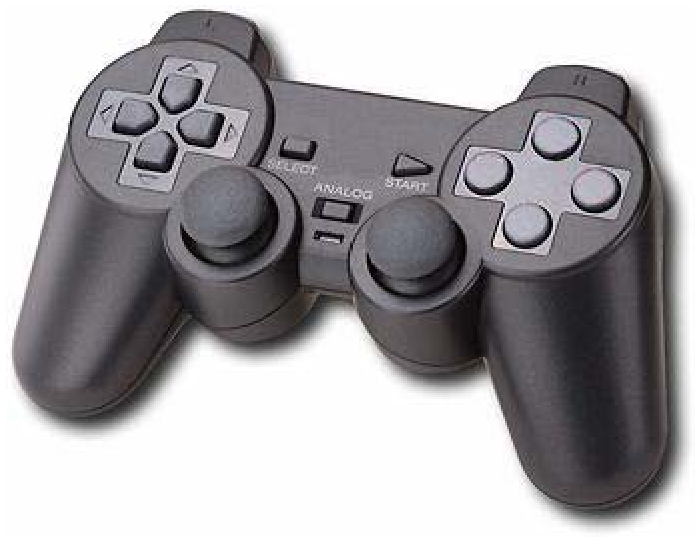


Figure 9. Sony's game controller [http://www.us.playstation.com/PS2/Accessories]

To gain the advantages of both types of input, a new method was invented that used digital input to substitute for analog input. The designer must fist rearrange analog data into digital data, and then apply digital input for the digital data. This new method was adopted for several purposes, such as television volume control. Loudness belongs to the analog data type. In old television sets, the volume control was a slider. With sliding control, users were able to adjust the volume very precisely. However, the slider has been replaced by buttons, with volume range being turned into a set of exact values. Each time users press a button, the volume value changes by one. This new control scheme has certain advantages but some drawbacks as well. From my own experience, I think that it sometimes difficult to make small adjustments. For example, I owned a television set in which volume level thirteen was too quiet and volume level fourteen was too loud.

4. Previous research on panning, zooming, fisheye view and multiple views.

Panning, zooming, fisheye view and multiple views are popular techniques used in 2D visualization. However, some perform better than others under certain situations. This thesis is focused on navigational map programs for small displays. Therefore, I will compare the costs and benefits of each visualization technique in terms of suitability for this function by discussing some past research studies in this field.

4.1. Panning

Panning is the most basic visual transformation technique. However, as mentioned in the previous chapter, panning has a major drawback: users will not be able to see other parts or objects that do not fall under the current viewport. Because of this disadvantage, users need to continue panning until they find the information they need. For text viewing or web surfing, this problem has generally remained unattended since there are alternative methods for users to find the information they want. One alternative is the text search function. This common solution can greatly reduce or eliminate the time that users spend on searching for the desired text. Another alternative is the index function. This technique is often used for lengthy text documents and web pages. Indexing provides jumps to specific points in the document.

Unfortunately, both of these alternative methods are almost useless with regard to map navigation. Most maps come as clusters of 2D flat picture files such that text searching and indexing are inapplicable or ineffective. In order to explore the map or search for specific points, users must keep panning aimlessly over a wide area.

Various research studies have demonstrated the lack of usability of panning-only systems. Gutwin and Fedak [2004b] conducted research comparing panning, zooming, and fisheye view. There were three tasks that users needed to perform. The first task was an editing task that required participants to use Microsoft PowerPoint office tool to edit a document. Participants had to follow a set of instructions, which included creating a file, changing the layout of a slide, adding a rectangle and a circle, selecting and changing the color of objects and saving the updated presentation. The second task was a navigation task that required participants to move from link to link as fast as possible using a standard web browser. In total, participants had to visit eighteen web pages. Each page was cached in the system, so there would be no delay due to the internet connection. The last task was a monitoring task that required participants to monitor the control panel. Whenever a problem occurred, a specific part of the control panel would change color, and participants had to move the mouse to click on a button in that panel. The results from the experiments are shown in Figure 10.

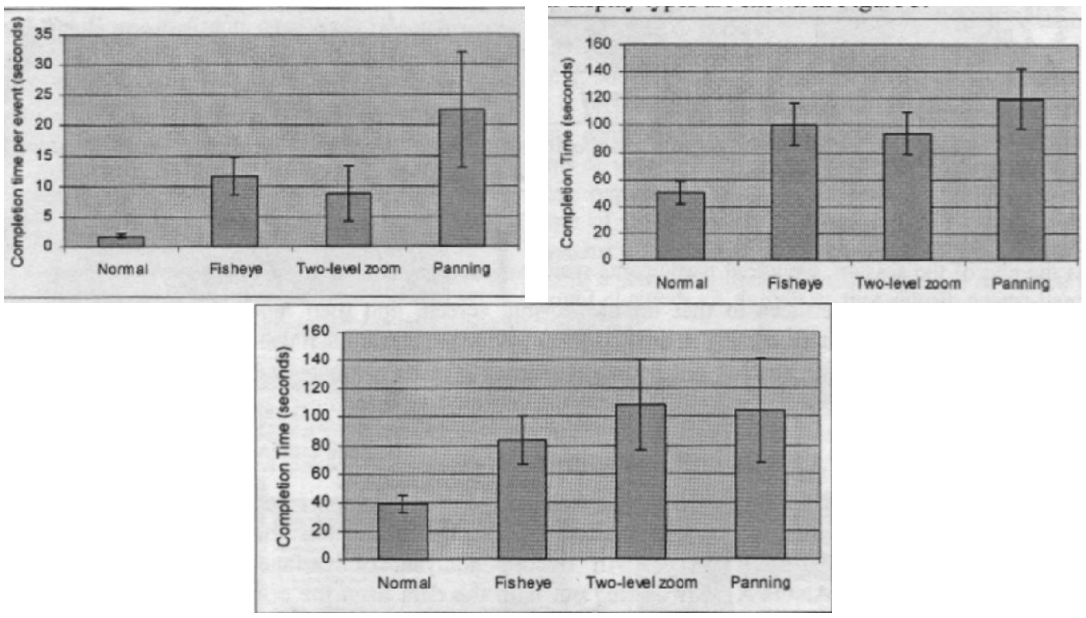


Figure 10. Comparison of completion times for three tasks using panning, zooming and fisheye view [Gutwin and Fedak, 2004b]

The panning method was slower than the other two visualization methods in the editing and monitoring tasks. Only in the web navigation task did panning score slightly better than two-level zoom. Furthermore, the standard deviation for panning was the highest in all three tasks.

Gutwin and Fedak mentioned four factors that contributed to the longer task completion times for panning in comparison to the other visualization methods. The first factor was the time required to move the viewport. The distances over which users had to move the viewport were substantial and required significant time, even with an analog controller such as a mouse. This factor was especially significant for the monitoring task, in which users were required to scan the whole document. The second factor was the arbitrary directions of movement required to find the targets of the tasks. Users had no idea where these targets were located in the documents. For example, in the fast task, users had to find specific parts of a document to edit. The third factor was the problem of missing events that occurred outside the viewport. For example, in the third task, users had to scroll back and forth to detect errors, which partly explains why panning took so much time. The last factor was the difficulty of scrolling through the documents, which was caused by unfamiliarity with the user interface.

In this research, Gutwin and Fedak performed the experiments on a relatively small screen (26cm x 19cm, 1024x768 resolution). They were unable to test a multiple views system because the screen size was too small.

In another research study, Hornbæk and Frøkjær [2003] performed a controlled experiment in which participants were required to read electronic documents via different visual transformation techniques. The participants had to read online documents supported by a linear interface (panning), a fisheye interface, and an overview plus detail interface (multiple views). They were then asked to answer a set of questions and write an essay about each document. Results were obtained from grading the essays and answers of participants, assessing the satisfaction of participants, and monitoring the performance of participants during the tasks.

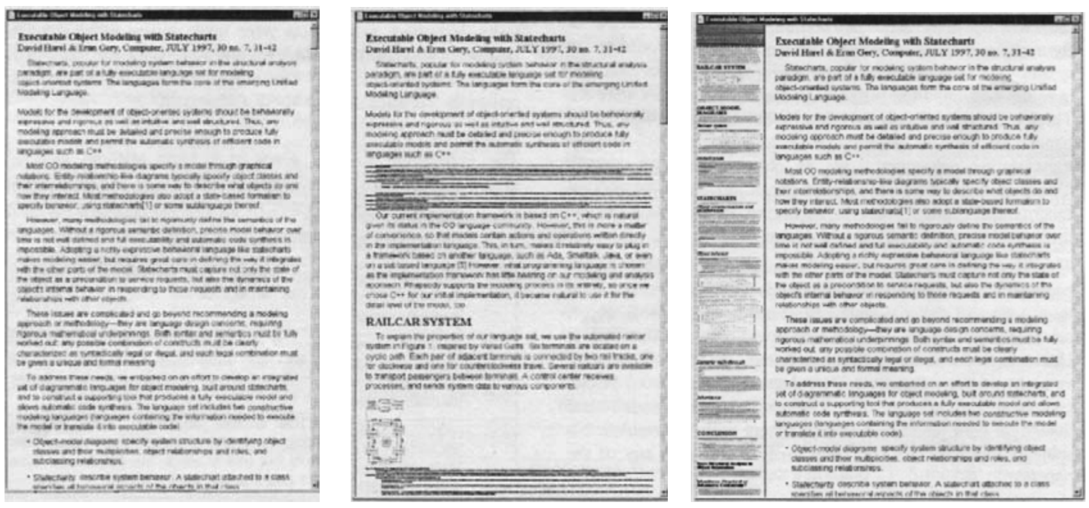
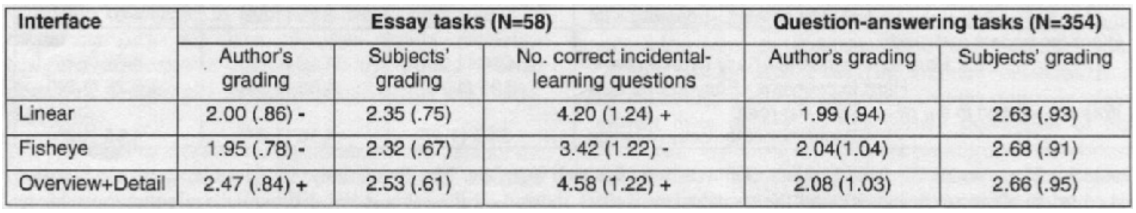

Executable Object Modeling with Statecharts
David Harel & Eran Gery, Computer, JULY 1997, 30 no. 7, 31-42
RAILCAR SYSTEM

Figure 11. Three visualization techniques for reading documents [Hornbæk and Frøkjær, 2003]

In Figure 11, the linear interface, which uses the panning method, is shown on the left. Participants can use the scroll bar to browse through the document. This method is by far the most common method of reading documents. Next, the fisheye interface is shown in the middle. For this method, each paragraph of text may be distorted to a small, unreadable size. Participants can use the mouse to click on various parts of the text to make them readable again. Lastly, the overview plus detail interface, which is one type of multiple views system, is shown on the right. The right frame of this interface is the detail frame, which functions in exactly the same way as the linear interface. The left frame is the overview frame, which depicts the entire document. The entire document is zoomed to fit the size of the overview frame, making it mostly unreadable. However, there is a highlighted zone in the overview frame (grey zone in the figure above) that indicates which part of the document is being displayed in the detail frame. Participants can drag and move the highlighted zone to change the content displayed in detail.

Hornbæk and Frøkjær divided their results into three parts: effectiveness, satisfaction, and efficiency. The scores from the participants' essays and answers were used to measure the effectiveness of each of the three visualization methods. The results, which range from 0 (worst) to 4 (best), are shown in the figure below.

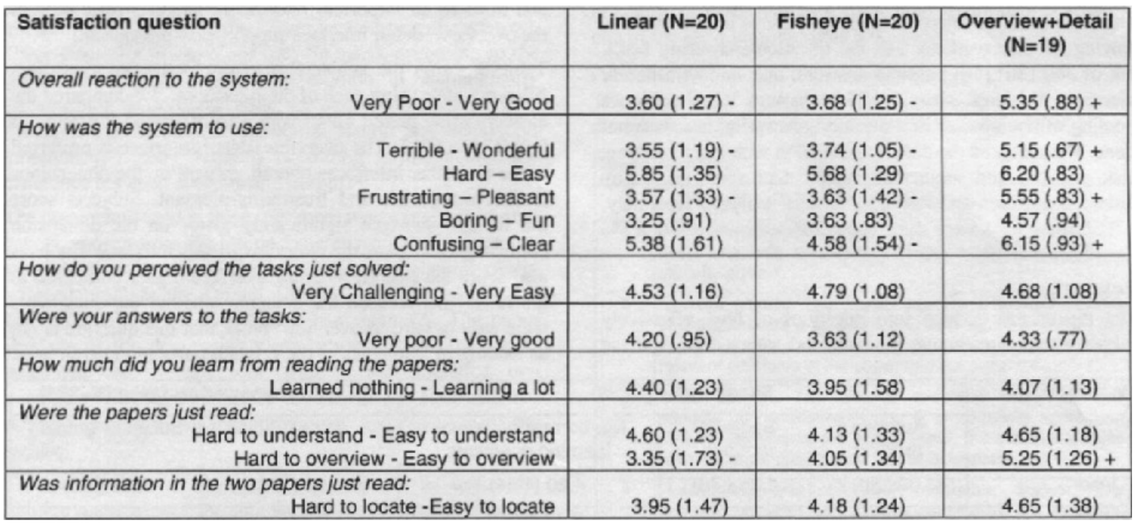


Interface	Essay tasks (N=58)			Question-answering tasks (N=354)	
	Author's grading	Subjects' grading	No. correct incidental-learning questions	Author's grading	Subjects' grading
Linear	2.00 (.86) -	2.35 (.75)	4.20 (1.24) +	1.99 (.94)	2.63 (.93)
Fisheye	1.95 (.78) -	2.32 (.67)	3.42 (1.22) -	2.04(1.04)	2.68 (.91)
Overview+Detail	2.47 (.84) +	2.53 (.61)	4.58 (1.22) +	2.08 (1.03)	2.66 (.95)

Figure 12. Effectiveness results [Hornbæk and Frøkjær, 2003]

For essay tasks, panning and fisheye performed poorly in comparison to multiple views, which gave approximately twenty-percent better scores. On the other hand, there was no difference in effectiveness between the three methods for question-answering tasks.

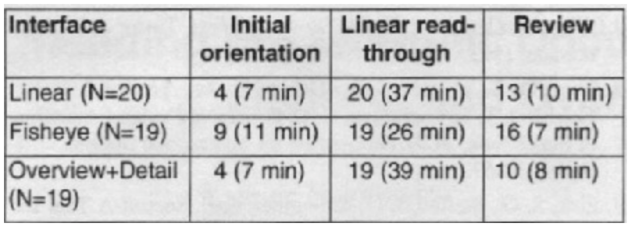
Satisfaction was another concern in this experiment. Immediately after using each interface, participants were asked to answer twelve questions about that interface. Then after having used all three interfaces, participants were asked to respond to another set of questions. Lastly, participants were asked to give comments about the interfaces and describe the reasoning behind the satisfaction scores that they gave.



Satisfaction question	Linear (N=20)	Fisheye (N=20)	Overview+Detail (N=19)
Overall reaction to the system:			
Very Poor - Very Good	3.60 (1.27) -	3.68 (1.25) -	5.35 (.88) +
How was the system to use:			
Terrible - Wonderful	3.55 (1.19) -	3.74 (1.05) -	5.15 (.67) +
Hard – Easy	5.85 (1.35)	5.68 (1.29)	6.20 (.83)
Frustrating – Pleasant	3.57 (1.33) -	3.63 (1.42) -	5.55 (.83) +
Boring – Fun	3.25 (.91)	3.63 (.83)	4.57 (.94)
Confusing – Clear	5.38 (1.61)	4.58 (1.54) -	6.15 (.93) +
How do you perceived the tasks just solved:			
Very Challenging - Very Easy	4.53 (1.16)	4.79 (1.08)	4.68 (1.08)
Were your answers to the tasks:			
Very poor - Very good	4.20 (.95)	3.63 (1.12)	4.33 (.77)
How much did you learn from reading the papers:			
Learned nothing - Learning a lot	4.40 (1.23)	3.95 (1.58)	4.07 (1.13)
Were the papers just read:			
Hard to understand - Easy to understand	4.60 (1.23)	4.13 (1.33)	4.65 (1.18)
Hard to overview - Easy to overview	3.35 (1.73) -	4.05 (1.34)	5.25 (1.26) +
Was information in the two papers just read:			
Hard to locate -Easy to locate	3.95 (1.47)	4.18 (1.24)	4.65 (1.38)

Figure 13. Satisfaction scores [Hornbæk and Frøkjær, 2003]

Despite the fact that the linear interface (panning) received the lowest scores in almost every aspect, this method is easy to understand and has been found to be the best way to learn from reading.



Interface	Initial orientation	Linear read-through	Review
Linear (N=20)	4 (7 min)	20 (37 min)	13 (10 min)
Fisheye (N=19)	9 (11 min)	19 (26 min)	16 (7 min)
Overview+Detail (N=19)	4 (7 min)	19 (39 min)	10 (8 min)

Figure 14. Task completion times [Hornbæk and Frøkjær, 2003]

Figure 14 shows the completion times for each task with each method of visualization. For essay tasks, the fisheye interface gave the fastest results, while the linear and overview plus detail interfaces gave about the same times. I think that the panning method could perform as well as the multiple views system because there is no need to use excessive visualization aids. However, the mechanics for bringing new content to the screen is slower for panning and multiple views than for fisheye. Therefore, based on these results, the researchers concluded that fisheye interfaces should be used for time critical tasks. However, for question-answering tasks, the linear interface performed the fastest. Hornbæk and Frøkjær did not give any reason for this unexpected result. The total number of question-answering tasks was 354, which is large enough to preclude luck or chance. From my point of view, the reason for this result could be that participants were able to find the answers by skimming over the whole content. In this case, the linear interface with scroll bar should outperform the unfamiliar fisheye interface.

In general, the panning method is considerably inferior to the other visualization techniques. Usually, panning requires more time to complete given tasks. The satisfaction level is average – better than fisheye but worse than multiple views. Nevertheless, panning is still favored by users in some cases, such as reading through an entire document.

4.2. Fisheye view

As mentioned earlier, fisheye view allows users to view or edit content via a distortion lens while providing a view of the whole data space on one screen. The part under distortion, in most cases, is the main point of interest and the part that users will perceive. The distortion lens may cover the whole screen or act as moveable magnifying glass over some parts of the screen. Many distortion techniques have been presented.

The previously discussed research studies by Gutwin and Fedak [2004b] and Hornbæk and Frøkjær [2003] were concerned with fisheye view as well. In Gutwin and Fedak’s research, the fisheye interface gave the fastest completion time for the navigation task, and the second fastest completion times for the monitoring and editing tasks (Figure 10). Furthermore, based on user preferences, fisheye was considered to be best suited for the monitoring task. In my opinion, the fisheye interface would have been the fastest overall method if the participants in this experiment were more familiar with it. Gutwin and Fedak reported that participants faced an accuracy problem in the monitoring task. The participants needed to accurately position the mouse cursor over on-screen buttons. Fisheye view caused difficulty in this aspect because the center of the fisheye lens was always positioned over the mouse cursor. Therefore, when participants moved the cursor toward a button that needed to be clicked, the button came under the influence of the distortion lens before the cursor reached it. Since the position of the button became distorted, many participants failed to move cursor over the button in one attempt. Gutwin and Fedak referred to this problem as "overshooting", and they observed that many participants had this difficulty. However, I think that this problem is not necessarily inherent to the fisheye technique, but can be attributed to an error in design and the inexperience of users. From my own experience, the accuracy problem should vanish once users have enough practice with the system. It is also possible to implement a snap function to eliminate this problem – when users move the mouse cursor near a clickable object, the cursor can be designed to jump automatically to that object.

In addition to "overshooting", the researchers reported another problem with fisheye view in this experiment. The problem was that participants were sometimes not sure if a certain button was the right button to click. Figure 15 shows what the fisheye view looks like for this task.

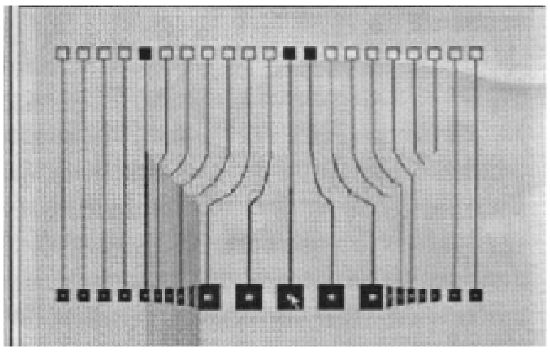
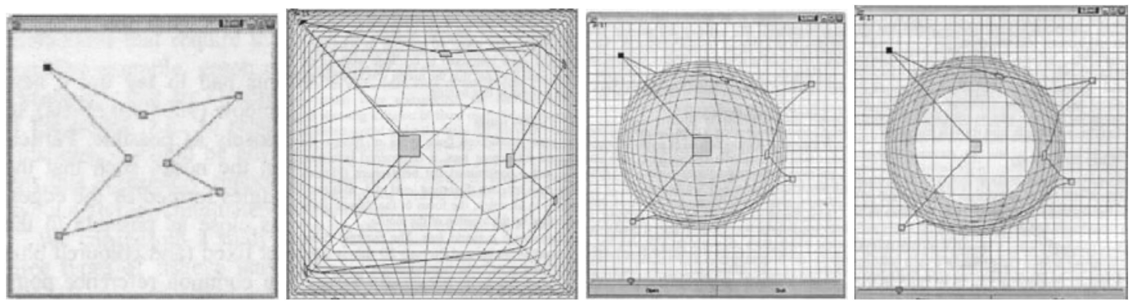


Figure 15. Fisheye view on monitoring task [Gutwin and Fedak, 2004b]

The wires connecting the monitoring devices (boxes) at the top of the screen and the buttons at the bottom of the screen are distorted from the effect of fisheye, especially at the edges of the lens. This distortion caused some participants to lose track and become unsure of which button they should click. In my opinion, this problem represents the real disadvantage of the fisheye technique. When 2D information is viewed under fisheye, spatial information – information about position, orientation, alignment, and relative distance – can be lost or misinterpreted. A research study that clearly shows this weakness of fisheye view will be discussed later in this chapter

In Hornbæk and Frøkjær's experiment, the second interface (Figure 11, in the middle) was claimed to be a fisheye interface. However, in my opinion, it is not a true fisheye interface. As explained by Gutwin and Fedak, interactive fisheye view should allow users to view or edit content via a distortion lens while giving users a view of the whole data space on one screen. In this case, the key phrase is "the whole data space". On the other hand, Hornbæk and Frøkjær describe their second interface as always keeping important parts of the document readable, while distorting less important parts to a small, unreadable size. These distorted parts can be expanded with a click of the mouse and reduced back with another click. The aim is to decrease the time that users take to scroll down the text while supporting an overview oriented reading style – first focus only on the main idea of the document and then come back to read the full details later. Based on this description and as shown in Figure 11, it is clear that users must still scroll down the text in this fisheye interface. This limitation adds to time usage on question-answering tasks, which could partly explain why the fisheye interface was second to the linear interface in this regard. To find the answer to a question, participants needed to move from one distorted part of the text to another while scanning for the answer. If participants did not find the answer on the first page, they needed to scroll down to the next page and repeat the process. Therefore, participants had to move the mouse cursor from the scroll bar to each block of distorted text and then back to the scroll bar again. In contrast, for the linear method, participants could just hover the cursor over the scroll bar all the time while keeping their eyes fixed on the text to scan for the answer. I think that this disparity accounts for the failure of fisheye view to achieve its real performance advantages in this experiment.

Another research study by Gutwin and Fedak [2004a] presented several types of fisheye transformations. The researchers understood that spatial layout – arranging graphical objects in 2D space to satisfy constraints on position, orientation, alignment, and relative distance – is the main weak point of fisheye view because the distortion effect makes it very difficult for users of the system to determine the spatial relations between objects. Consequently, the researchers chose to use 2D graphics as test objects in this experiment.



(a) (b) (c) (d)

Figure 16. Various fisheye transformations [Gutwin and Fedak, 2004a]

Figure 16a shows an undistorted object. Figure 16b represents a full-screen pyramid lens, Figure 16c represents a constrained hemispherical lens, and Figure 16d represents a constrained flat-hemisphere lens.

In this research, participants were asked to create copies of target figures consisting of nodes and lines. The original figures were shown on a separate small screen while participants had to create the copies on a big fisheye screen. Four types of visual transformations were tested: no distortion, full-screen pyramid fisheye lens, hemispherical fisheye lens and flat-hemisphere fisheye lens, just as in Figure 16. Each task had a one minute time limit. The results are shown in Figure 17.

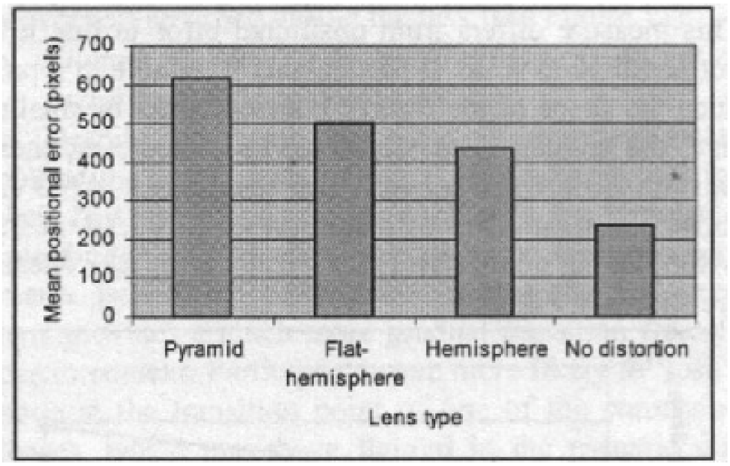


Figure 17. Mean positional error results [Gutwin and Fedak, 2004a]

The results of this research were as Gutwin and Fedak expected. The mean positional error rates for all fisheye transformations were significantly higher than for the method without distortion (Figure 17a). The difference in error rates between the experimental and control methods was almost twice, even with partial distortion lenses that could be moved away from the point of interest. These results imply that participants were able to see how and where they needed to move the nodes but were not able to do so accurately. The researchers also obtained participant opinions on which lens provided the most accuracy (Figure 17b). The results were also as predicted, with no distortion being the favorite method among users.

Based on the various research studies that I examined and discussed in this section, I conclude that the fisheye visual transformation technique is appropriate for tasks that are:

- Time critical – because fisheye view is faster than other techniques for many types of tasks.
- Not concerned with spatial information – because the area under distortion will temporarily lose its spatial properties.

As mentioned earlier, this thesis is about navigational maps, where spatial information is crucial and time is not critical. Thus, *I will not consider fisheye as a visualization option for this thesis.*

4.3. Zooming with and without overview

As mentioned earlier, zooming can provide a quick impression of the whole content when zooming out and can give details when zooming in. The zoom level can be set dynamically by the user, or it can vary automatically depending on what is appropriate for the task at hand. In the previously discussed research study by Gutwin and Fedak [2004b], a two-level zoom method was used for three tasks. Compared with the fisheye and panning techniques, the zoom method gave the fastest completion times for the editing and monitoring tasks, but gave the slowest completion time for the navigation task. In this navigation task, participants were required to visit several links in a limited amount of time. This task exposed a weakness of the zooming method.

Gutwin and Fedak found that while using two-level zoom, users needed to switch between the detail level and the overview level quite often. Every time a switch was initiated, the time spent on the changing process produced some costs, which were not similarly incurred by the panning and fisheye techniques. Moreover, if the link text was too small in the overview level, users needed to switch to the detail level to read the text. This limitation caused participants to switch views more than once per web page. The researchers categorized the costs in this experiment into three types: adjusting to new context, remembering to switch, and performing the switch. The cost of adjusting to new context was incurred when users performed the zoom action. Users needed to spend time to determine if the current viewport was the one they desired, and to figure out where they were in the data space, where the cursor was positioned, and where the targets they wanted to read were located. This cost can be reduced by animated zooming, where the transition from the old to the new scale is smooth [Guo et al., 2000]. There is debate among researchers about how much time animated zoom should consume and what the maximum zoom factor per second should be. Guo et al. argued that zoom speed should be about eight factors per second, while Card et al. [1991] proposed that zoom time should be approximately one second in all cases. Regardless, using animated zoom will reduce the cost of adjusting to new context for users. However, it will instead raise another type of cost, the cost of zooming.

The next cost was the cost of remembering to switch, which was loaded in the visual memory of users. Gutwin and Fedak noticed that users often spent time in the wrong zoom level. For example, after selecting a link, users should zoom out to the overview level to look for the next desired link. However, users often forgot to zoom out and continued onward in the detail level to look for the next target. Actually, I think that some users intended to stay in the detail level since there were costs for repeatedly zooming in and zooming out. If it is possible to find the target links without zooming out to the overview level, users should perform the task more quickly. The last cost was the cost of performing the switch. In this experiment, switching was performed by pressing a key combination, a mechanism that requires both mental and physical effort. It was observed that after pressing the zoom key combination on the keyboard, users generally moved their hands over to the mouse. So when it became necessary to initiate a switch again, users had to look at the keyboard to position their hands correctly for pressing the zoom key combination. In my opinion, this problem is not necessarily inherent to the zoom method, but can be attributed to a design flaw. The problem could be resolved if software makers set the zoom function at the mouse wheel.

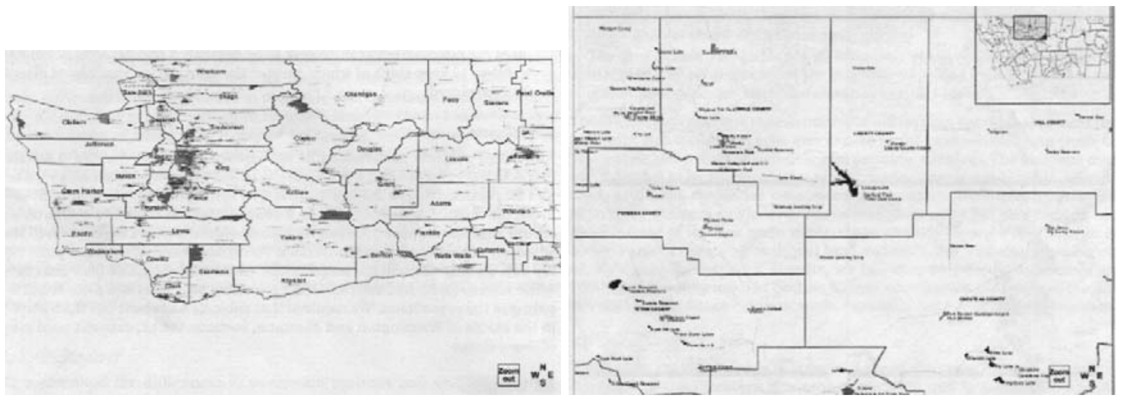
Another research study by Hornbæk et al. [2002] gave valuable information about the usability of the zoom interface. Their research aimed to clarify three assumptions:

- How the presence or absence of an overview affects usability;
- How an overview influences the way users navigate information spaces; and
- How different organizations of information spaces may influence navigation patterns and usability

In conjunction with these three research aims, the researchers also set three hypotheses:

- Recall of objects on the map would be better in the no-overview interface.
- Participants would prefer the overview interface because of the information contained in the overview window and the additional navigation features.
- The overview interface would be faster for tasks that require comparing information objects and scanning large areas.

The researchers created two prototype maps: one with only the zoom feature, and one with the zoom feature plus small overview window. The maps could be viewed from scale 1, which displays the entire area of the map, to scale 20, which displays the most detailed level. The zoom factor was fixed at 8 per second, which means that users must zoom for 2.5 seconds to go from scale 1 to scale 20. Users could zoom in by holding down the left mouse button – zooming would start after 0.4 seconds. Similarly, users could zoom out by holding down the right mouse button. To pan, users had to press the left mouse button and drag the mouse in the desired direction. The map view would follow the mouse while it was being dragged. The overview interface added extra control schemes to the zoom interface. When users clicked on a certain position in the overview map, the main view (zoom map) would instantly move to center on that position while maintaining the current zoom level. Alternatively, users could drag the field-of-view box, a small box appearing in the overview window, to correspondingly change the area displayed in the zoom window. Users could also resize the field-of-view box to correspondingly change the zoom level of the zoom map. Lastly, users could even draw a new field-of-view box to set the displayed area and zoom level for the zoom map. Screenshots of both maps can be seen below.



(a) (b)

Figure 18. (a) map without overview (b) map with overview [Hornbæk et al., 2002]

For each type of interface, the researchers used two state maps: one of Washington and the other of Montana. The Washington map had multi level detail. At the outermost level, only important labels like state names and city names were shown. Then at zoom level 5, details about the cities were shown, and at zoom level 10, all landmarks were shown. On the other hand, the Montana map had only one level of detail, meaning that users could see all landmarks from zoom level 1 (overview level).

Participants were required to perform three types of tasks: navigation, browsing and recall. There were three sub-types of navigation tasks. First, participants had to locate specific objects within the map. Enough details were provided, including the county in which each object was located, so that participants did not need to scan the whole map at zoom level. Second, participants had to compare two objects located in different cities or counties, and last, participants had to pan the camera following described paths. For the browsing tasks, participants were asked to scan the whole map to find specific objects. Next, they were asked to find the county that contained the most cities or the largest cities, and finally, they were asked to locate the nearest objects of various types in nearby counties. There were 10 tasks in total for each map – five navigation tasks and five browsing tasks. Afterward, participants were asked to perform three recall tasks to test their memories of the contents and outlines of the maps. The first task asked participants to list all the objects located within specific areas. The second task asked participants to put as many objects as they could remember onto blank areas of the maps. The last task asked participants to circle the names of cities that were located within certain counties and cross out the names that did not belong.

The results of this research were categorized into five aspects: accuracy of answers, ability to recall, task completion time, user preference and satisfaction, and user action pattern. The accuracy results are displayed in the figure below.

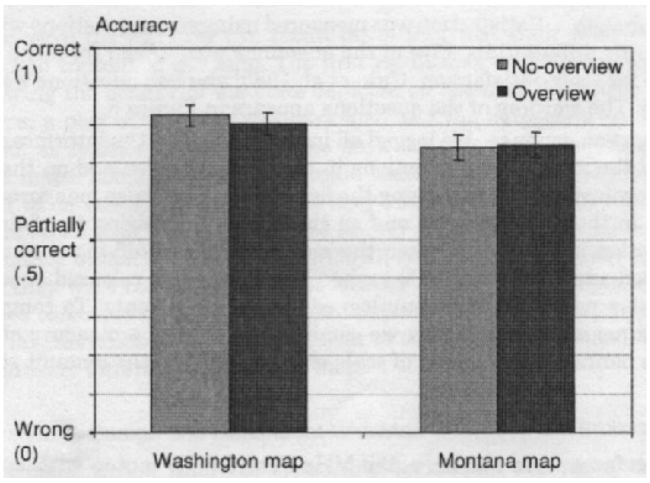


Figure 19. Accuracy results [Hornbæk et al., 2002]

Statistically, there was no difference between the two interfaces, although the Washington map gave slightly better accuracy results than the Montana map. This finding can lead to the conclusion that there is no need to implement an overview map if the zoom feature of the map is well constructed. Two participants did not even use the overview map on every task. The difference in accuracy between the Montana and Washington maps could be a consequence of how the maps were laid out. In terms of detail, the Montana map was a single level map while the Washington map was a multi level map. The other characteristics of the maps were almost identical, including the number of objects (1,591 / 1,540), mean distance between objects (7.1 / 7.8), and screen area occupied by the state at the outermost zoom level (50% / 57%). Hornbæk et al. suggested that there should be a separate experiment to clarify this issue.

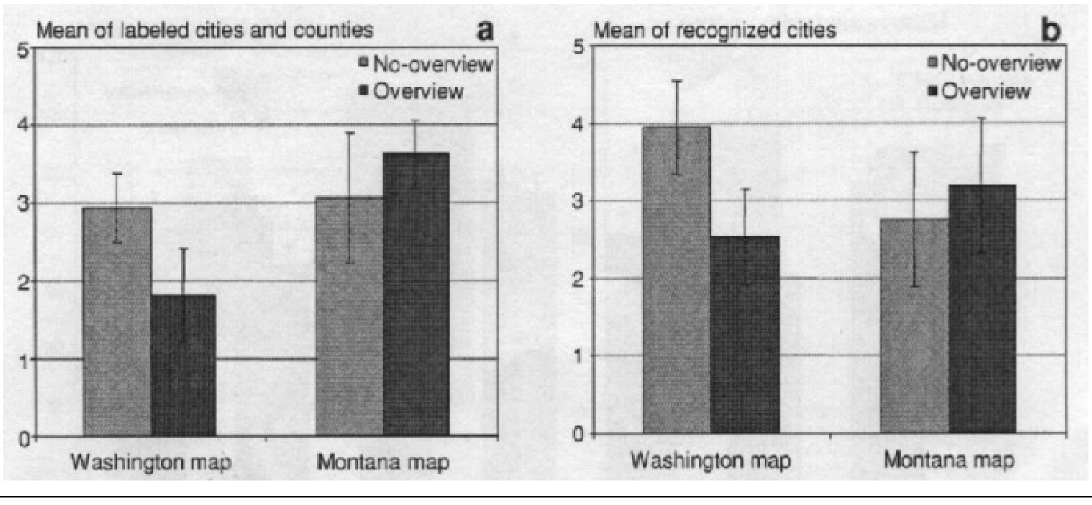


Figure 20. Recollection results [Hornbæk et al., 2002]

There was very high deviation in the ability to recall the map, as shown in Figure 20. Furthermore, as with accuracy, there was no statistical difference in recall ability between the two interfaces. Generally however, the no-overview interface gave better results for the Washington map, while the overview interface gave better results for the Montana map. This finding surprised the researchers and contradicted their hypothesis that recall of objects on the map would be better in the no-overview interface. In my opinion, I think this unexpected finding was due to the organization of details within the maps (single layer / multi layer). For the multi layer Washington map, participants needed to zoom in to see more objects. Thus, zooming is what helps participants to remember better. In contrast, for the single layer Montana map, all objects were displayed at every level of zoom. There was no need to zoom in to see and recognize objects. So in this case, panning with an overview window can help participants to remember more objects. Overall, these recall tasks seemed dependent on individual strengths. One tester scored 19 out of 20, while another tester scored only 1 out of 20.

Task completion time was another important aspect. Based on discussions in the previous chapter, the no-overview interface would likely require less time, which was partly true in this experiment. Mean task completion times are displayed below.

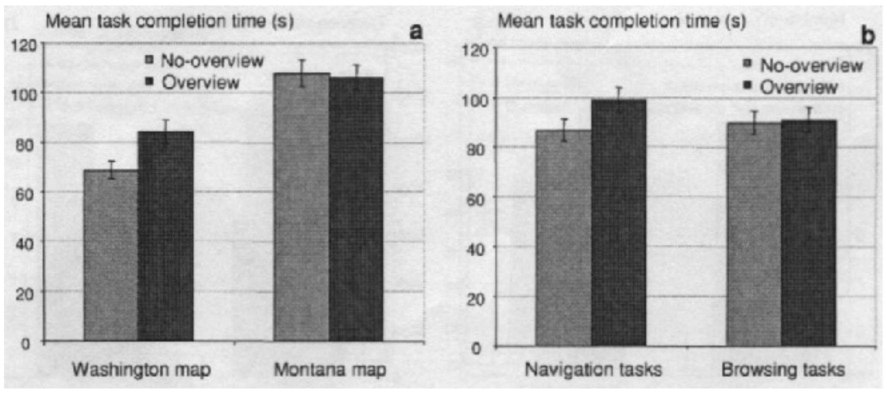


Figure 21. Mean task completion times [Hornbæk et al., 2002]

The Washington map was significantly faster than the Montana map in every regard. For the Washington map, as expected, the no-overview interface performed 22% faster than the overview interface. However, there was no difference between the two interfaces for the Montana map. Again, in my opinion, this finding reflected the different ways in which the maps were laid out. With regard to navigation tasks, the no-overview interface performed faster than the overview interface. However, there was no difference with regard to browsing tasks. This finding also contradicted the hypothesis that the overview interface would be faster for tasks that require comparing information objects and scanning large areas, such as browsing tasks.

In terms of user preference, 26 participants preferred the overview interface while 6 participants preferred the no-overview one. This outcome confirmed the hypothesis that participants would prefer the overview interface because of the information contained in the overview window and the additional navigation features. Several comments were obtained from many participants. Among those who preferred the overview interface, 9 participants stated that “It is easier to keep track of where I am”. Furthermore, 7 participants agreed that the overview window helped support their navigation, while 4 participants said that the overview panning mechanism was useful for scanning a large area. Among those who preferred the no-overview interface, 2 participants mentioned that using only the zoom feature made it possible to finish their tasks more quickly, while 4 participants thought that the overview window at the top right of the screen blocked their view.

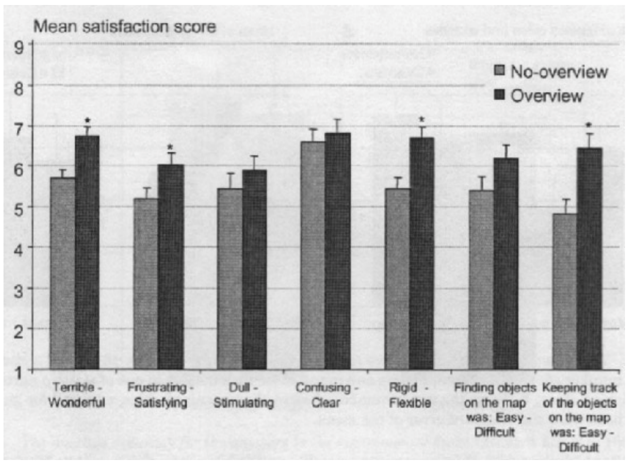


Figure 22. Mean satisfaction scores [Hornbæk et al., 2002]

Satisfaction scores were in line with user preferences. The overview interface received better satisfaction scores on all questions. Statistically, the overview interface scored significantly higher on four questions: Terrible-Wonderful, Frustrating-Satisfying, Rigid-Flexible, and Keeping track of object was: Difficult-Easy.

The last aspect of the results dealt with navigational pattern, which I found to be quite spectacular. Research studies in this field have often ignored this concern and concentrated more on completion time or accuracy rate. This omission may be due to the difficulty in observing participants carefully during the course of the experiments.

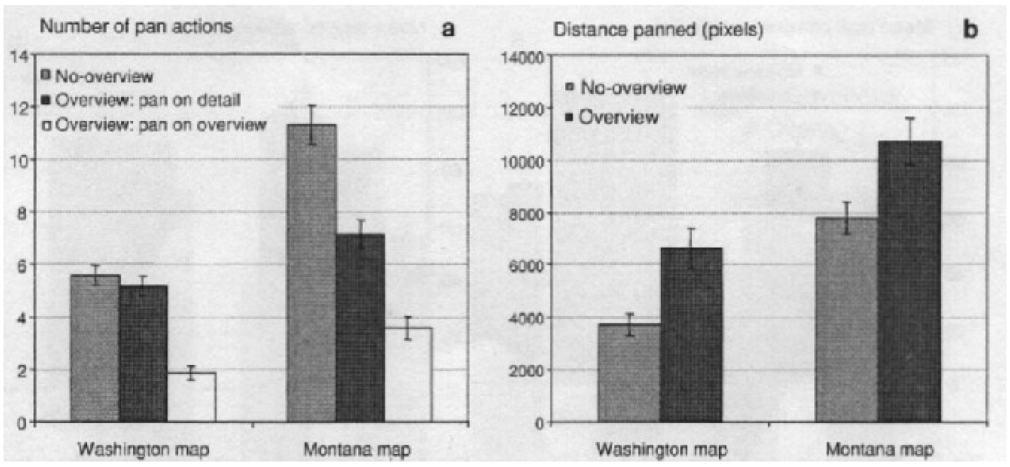


Figure 23. Number of pan actions [Hornbæk et al., 2002]

Some participants preferred to pan by moving the field-of-view box in the overview window, as shown in Figure 23. Moreover, it was noticed that the overview interface yielded greater distances panned for both state maps. In my opinion, this finding represented one of the major advantages of the overview interface. There was also a significant difference in the number of pan actions between the Washington and Montana maps. For the Washington map, the number of "pan on detail" actions in the overview interface was almost equal to the number of pan actions in the no-overview interface. If the number of "pan on overview" actions was also included, there were more pan actions in the overview interface in total. This was not the case for the Montana map. Given how these maps were laid out, it is possible to conclude that for maps with single level detail, the overview interface is on par with the simple zoom interface in terms of user actions required. However, for maps with multi level detail, the overview interface will require more user actions.

The number of zoom actions was also recorded in this experiment (Figure 24).

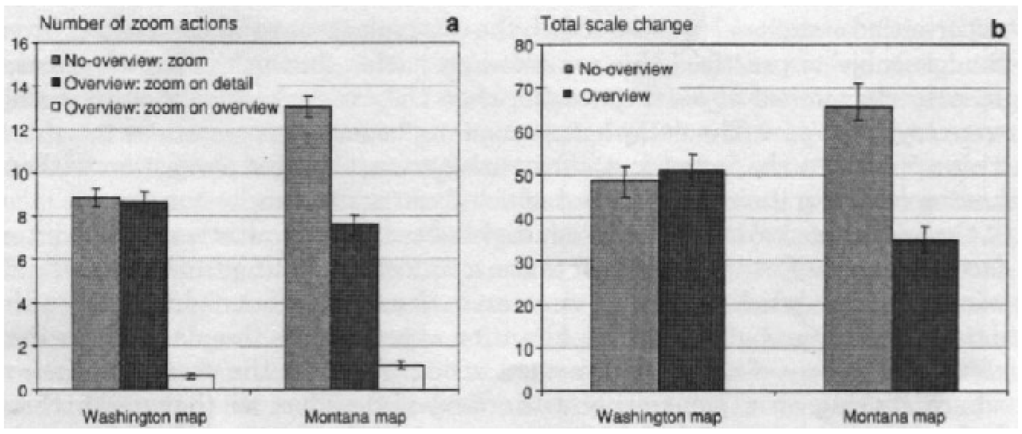


Figure 24. Number of zoom actions and scale changes [Hornbæk et al., 2002]

In tasks involving the overview interface, zooming by resizing the field-of-view box was done less than one-third of the time. This result probably reflected the relative ease of zooming via the detail window and via the overview window. In the zoom interface, participants simply had to click and hold the left or right mouse button to zoom in or out. In comparison, resizing the field-of-view box was more complicated. Some participants complained that the overview window was quite small, and that it was difficult to resize a tiny box in such a small window. Hornbæk et al. noted that the size of the overview window was 256 x 192 pixels, and at a zoom factor of 20, the field-of-view box was only 13 x 10 pixels. I think that if the overview window was made larger, there would be less discrepancy between zooming via the detail window and via the overview window.

The researchers observed another problem called desert fog, which I found very interesting. Desert fog occurred when users zoomed or panned into an empty area, where there were no landmarks or guides. Four participants got caught in desert fog with the simple zoom interface, while only half that number was observed with the overview interface.

In conclusion, this research by Hornbæk et al. yielded valuable insights about zoom interfaces with and without overview. Many results from this research will greatly affect the design part of this thesis. For instance, the findings based on comparisons between the Washington and Montana maps (multi-level versus single level detail) will be particularly useful. With single level detail, there should be no difference in accuracy or task completion time between the two zoom interfaces, although the overview interface should receive significantly higher scores for user preference and satisfaction. Navigational patterns may also be affected by the type of map. There is a slight uncertainty about this research by Hornbæk et al. regarding the small size of the overview window, which made the field-of-view box tiny and difficult to use. The researchers stated clearly that their design of the overview interface was arbitrary and lacked design guidelines for the size of the overview window. In the next section, I will present research with the opposite design for the relative sizes of the windows (large overview window and small, pop-up detail window).

4.4. Multiple views and its cognitive cost

Multiple views interface is another visualization technique to help users make sense of large information spaces such as navigational maps. The principle of multiple views is very simple – there must be more than one view that can be used to explore the data space. By receiving information from more than one view at once, users acquire an information load in their memory, called visual working memory. Plumlee and Ware [2006] carried out critical research about visual working memory used in visual comparison tasks. Based on their previous publications, visual working memory can be separated into many parts, such as the memory for visual and verbal cues or the memory for cognitive instruction sequences. This visual working memory is very limited in both capacity and duration in terms of information retention. Many other research studies have been conducted to better understand this working memory. One test involved showing participants several simple shape objects, followed by a blank screen, and then followed by another set of shape objects. Results from these kinds of experiments are crucial for understanding the capacity of visual working memory.

There are three notable characteristics of visual working memory. First, visual working memory can hold only three objects. So if there is a new object, an old object will be removed from memory. Second, visual working memory can store multiple attributes of an object, such as color, size or shape, as long as there are not more than three objects. Third, when the properties of the objects are too numerous or too complex, visual working memory will hold less than three objects. When it comes to visual comparison, this memory is the main engine for comparing things. People will look at the first picture or view, and then store it in their visual working memory. Afterward, they will look at the second picture or view to make a comparison. If the two things are visible at the same time, it will be quite easy to compare them. However, if it is not possible to see both views simultaneously, the time spent while changing from the first view to the second view should be less than a few seconds, because that is the limit of visual working memory.

Based on these characteristics of visual working memory, Plumlee and Ware proposed a general model for human performance in navigation-intensive tasks:

$$T = S + \sum_{i=1}^{V} (B_i + D_i)$$

where

T is the expected time to complete the task,

S is the expected overhead time for constant-time events such as setup and user-orientation,

V is the expected number of visits to be made to different focus locations during the course of the task,

B_i is the expected time to transit between the prior location and the location corresponding to visit *i* and

D_i is the expected amount of time that a user will spend at the focus location during visit *i*.

In this model, there are four points at which total time usage can be decreased – at S by making set up faster, at B by reducing the time spent on moving from object to object, at D by facilitating the inspection of objects, and at V by reducing the number of visits.

To be more specific for the tasks in their research study, which involved comparing sets of objects, the researchers modified the formula to:

$$T = S + f_v (P, V_p) . (B + D)$$

where

P is the expected number of non-matching comparison sets that will be visited before the task is completed,

V_p is the expected number of visits made for each comparison set,

f_v is a function that calculates the total number of expected visits given P and V_p,

B is the expected time to make a transit between sets on any given visit, and

D is the expected time for the user to make a match determination during a visit.

In order to compare two sets of objects, users must first set up their interface – represented by S in the formula. Then, users must look at the group objects for a sufficient amount of time to remember the details – represented by D. After that, users must move to the next group of objects – represented by B. Users will have to look at the new group of objects and spend time to determine if this group is the correct match – also represented by D. This loop will go on repetitively until the task is done. The total number of times that users need to visit the various sets of objects is given by function f_v. This function is based on two variables – how many times users need to visit one particular group, and how many groups users need to visit. In this case, Plumlee and Care defined function f_v as:

$$f_v (P, V_p) = \frac{(2 + P) . (M + n)}{2M} \quad | n = kM, k \in N$$

where

n is number of objects in each set, and

M is maximum number of objects that can be held in visual working memory.

The next step in creating a model was to modify the formula according to the types of interfaces used in this study. For the zoom interface, the variables affected by this visualization method are S and B. In terms of set up (S), zooming has an advantage over multiple views. On the other hand, transit time between sets (B) will be high – and even higher if users are required to zoom several times. After substituting for function f_v (P, V_p), the final formula for the zoom interface was:

$$T_{zoom} = S_{zoom} + \frac{(2 + P) . (M + n)}{2M} . (B_{zoom} + D) \quad | n = kM, k \in N$$

On the other hand, the formula has to be applied twice for the multiple views interface, because there are two types of B. One type is B_{multi} that occurs when users set up multiple views, and the other type is B_{eye} that occurs when users look back and forth between the windows. Thus, the final formula for the multiple views interface was:

$$T_{multi} = S_{multi} + P. B_{multi} + \frac{(2 + P) . (M + n)}{2M} . (B_{eye} + D) \quad | n = kM, k \in N$$

Both formulas fall under the category of linear formulas, where S_{zoom} and S_{multi} + P. B_{multi} act as the intercept line, while B_{zoom} + D and B_{eye} + D act as the slope. It is notable that S_{zoom} is considered significantly smaller than S_{multi} + P. B_{multi} while B_{zoom} is considered significantly larger than B_{eye}. The relationship between the zoom interface and the multiple views interface for object comparison tasks should look like the graph displayed in Figure 25. However, it is also important to consider that each individual has a different capability for remembering things, which creates an area over each line. From previous Plumlee and Ware researched, the worst case scenario is that users can remember only two objects at a time, while the best case scenario is four objects.

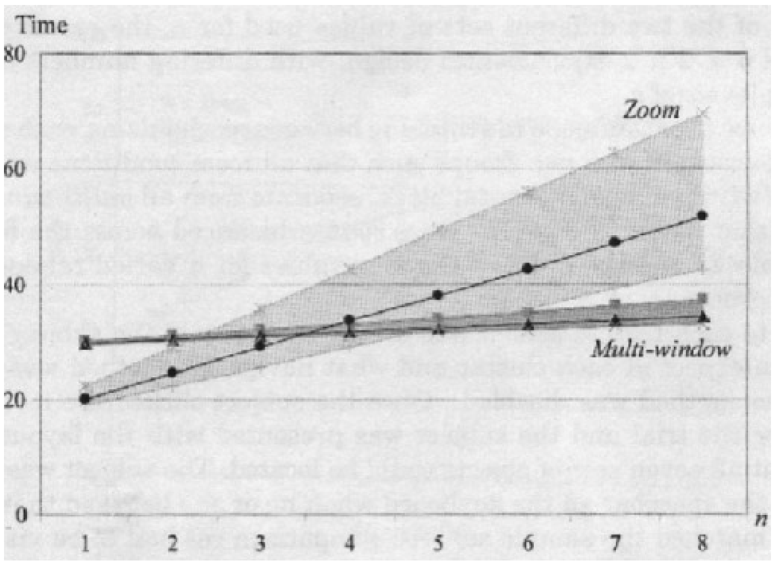


Figure 25. Estimate of task completion time [Plumlee and Ware, 2006]

As n increases, the time required for users to finish the task also increases. This relationship graph shows that there is a certain value of n above which the multiple views interface will be faster than the zoom interface. This outcome arises because users need to spend more time visiting each object group in the zoom interface than in the multiple views interface. However, when n is low, the zoom interface will generally be faster because the multiple views interface requires more set up time from users, both physically and mentally.

To verify their theory, Plumlee and Care set up a controlled experiment that required participants to compare sets of objects. Seven sets of objects were scattered randomly throughout the view window. One was the sample set, indicated by a yellow border, while the rest were comparison sets. Only one comparison set contained exactly the same objects as the sample set. In each of the other comparison sets, there was one object that was different. Participants needed to look at the sample set, remember it, and then compare the other sets to it. The task ended when participants identified the matching set. The number of objects in the sets was random, but every set in each round contained the same number of objects. A total of five shapes were used for objects, as shown in Figure 26.

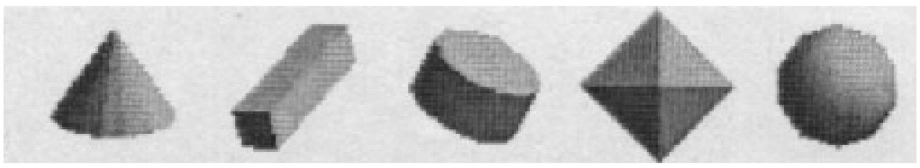


Figure 26. Five objects used in the experiment [Plumlee and Ware, 2006]

Two navigation mechanisms were used in this experiment, zooming and multiple views. Participants could initiate zooming by clicking on any spot on the screen using the middle mouse button. With very fast camera panning, the view became centered on that spot. This panning animation only took about one second. Participants could then zoom in at the rate of seven times per second or zoom out at the rate of eight times per second by moving the mouse forward or backward. They could zoom in and out as many times as needed until the task was finished. For multiple views, it was harder for participants to set up the display. First, participants needed to press Z to open a detail window and resize it. After that, they had to assign an area of the map to be displayed in this window. The detail window came at the optimal zoom level, so there was no need for participants to adjust the view. This was done to totally separate zooming and multiple views. The detail window was linked to the target area by semi-transparent lines. Participants could drag the target area to change the content being displayed in the window. A maximum of two detail windows could be kept open during the task. Participants were expected to use one window to display the sample set, and another window to display the comparison sets. This multiple views interface is shown below.

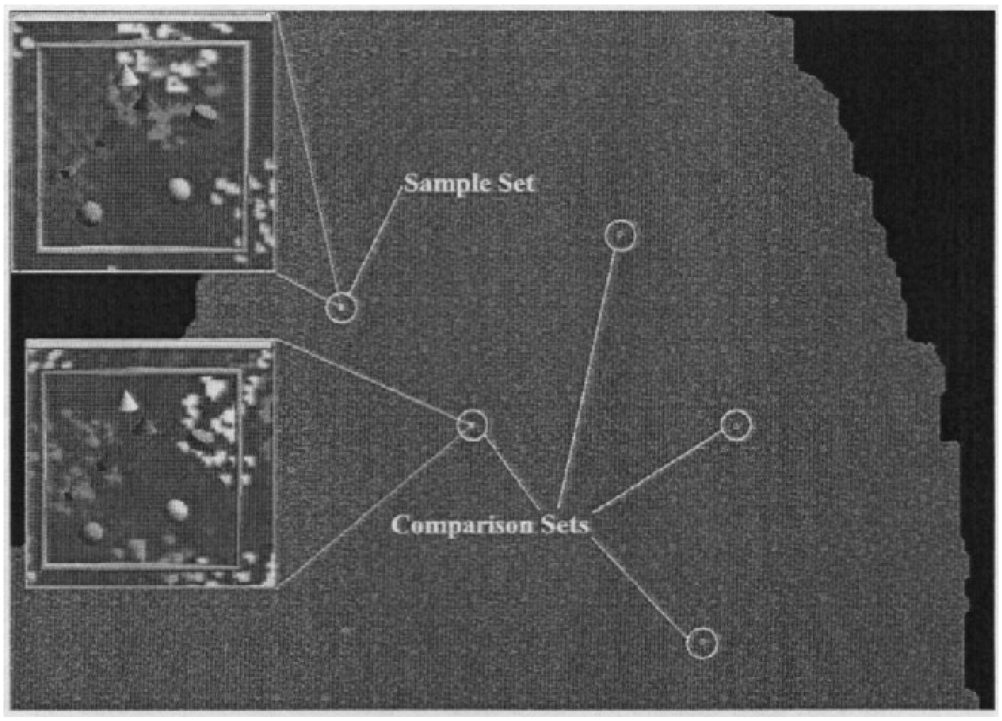


Figure 27. Multiple views interface [Plumlee and Ware, 2006]

The experiment was divided into two rounds. The first round consisted of eight participants and a series of [1, 2, 3, 4] objects per set. The second round consisted of twelve participants while the number of objects per set in the series was increased to [1, 3, 5, 7]. Participants were taught about the control mechanism and were briefed about what they were expected to do prior to the test. A total of 1451 trials were conducted. The results for completion times are shown below.

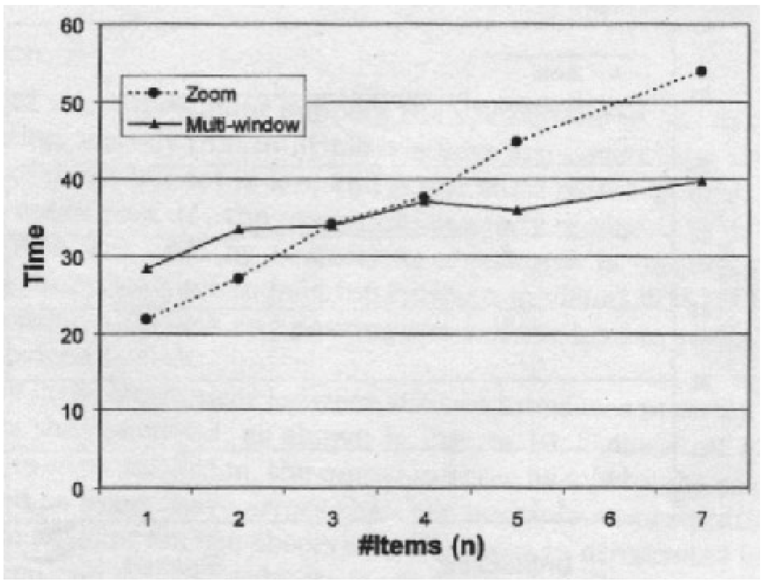


Figure 28. Comparison between zooming and multiple views [Plumlee and Ware, 2006]

The results were as predicted by the model. The slope of the line for zooming was greater than that for multiple views, but the starting point of zooming was lower. The point of intersection at which the two interfaces performed comparably was between n = 3 and n = 4. However, the error rates were very different for the two interfaces, as shown below.

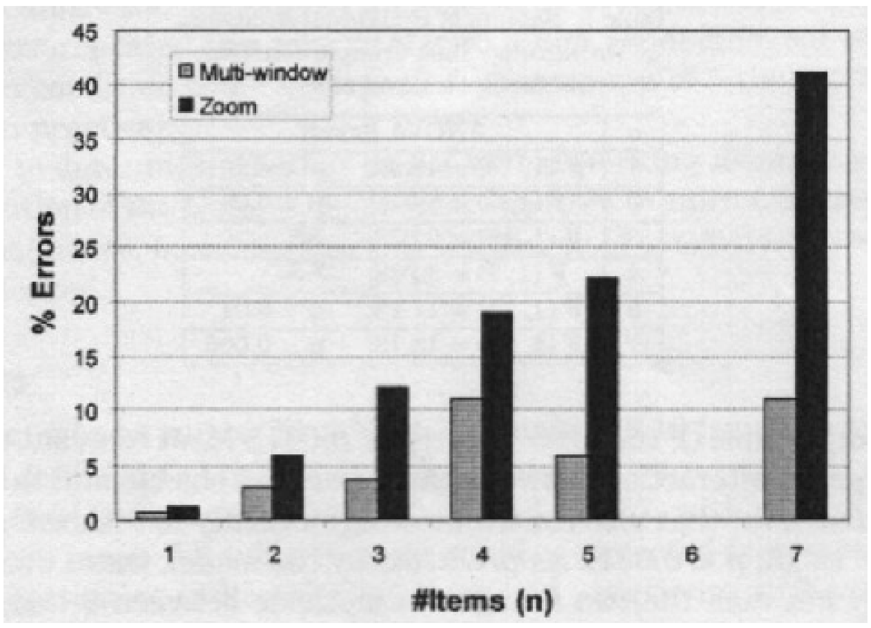


Figure 29. Error rates [Plumlee and Ware, 2006]

Apart from the case of n = 1 (only one object to compare in each group), the error rates for the zoom interface were higher than those for the multiple views interface by 100% or more. Plumlee and Ware conducted another experiment to determine the cause of these differences in error rates. The results are shown in Figure 30.

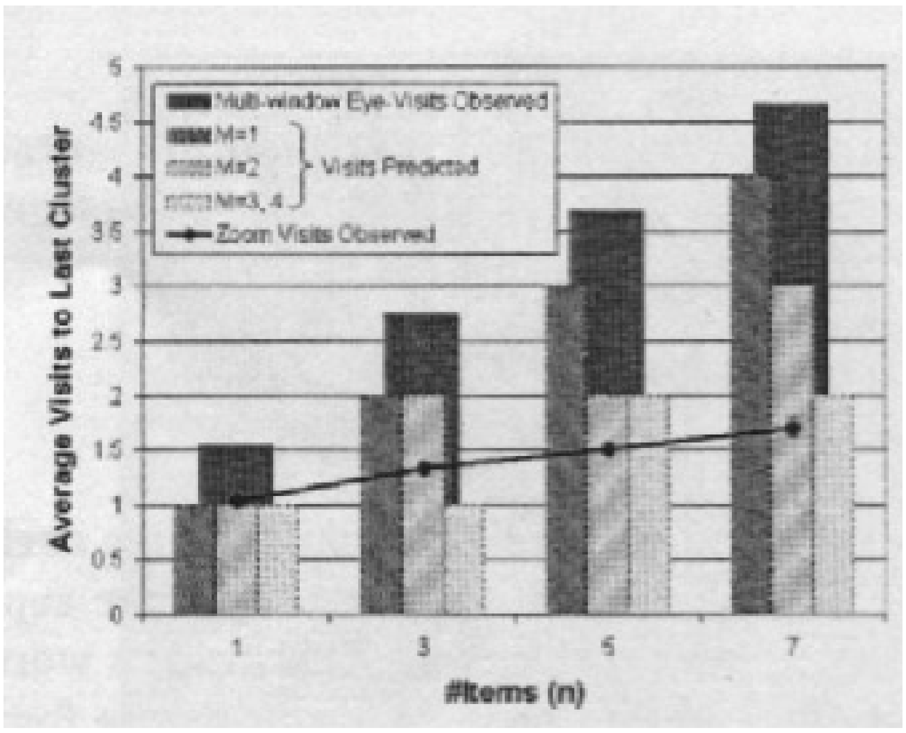


Figure 30. Expected versus actual visits [Plumlee and Ware, 2006]

The results reveal that when using the zoom interface, participants did not visit the object sets enough times as expected by the model. Conversely, participants overly observed the object sets when using the multiple views interface. This finding can be explained by the work of Herbert A. Simon, a Nobel prize-winning political scientist and professor at Carnegie Mellon University. Simon said that when a task is cheap in time and cognitive work, participants will use their visual working memory effectively. In this case, looking between two detail windows can be considered an easy task. On the other hand, for a task that requires a lot of time or heavy cognitive effort, participants will overload their visual working memory, resulting in many errors. Such was the case in zooming. Therefore, to balance the error rates while minimizing completion times, participants would need to spend more time on the zooming interface and less time on the multiple views interface. From this analysis, I think that the multiple views interface presents a better option for visualization than the zoom interface.

This model for comparing multiple views and zooming is valid not only for this experiment. In the research by Hornbæk et al. [2002] discussed in the previous section, comparisons were made between zoom interfaces with and without overview. The Montana map used in that research had single level detail and characteristics similar to the map used in this experiment. The tasks designed by Hornbæk et al. (such as looking for a town with a specific attribute, or searching for the city with the greatest population) can be considered as cases of $n = 1$. Since a small overview window was already open and ready for use, no extra setup for multiple views was required from users. Without extra setup time and at $n = 1$, the completion times for the two interfaces should be comparable. This prediction is supported by the results for the Montana map in Figure 21.

5. Nintendo DS, the mobile gaming device

This chapter deals with Nintendo DS [http://www.nintendo.com/ds], the device that will be used for this thesis. First, I will give general details about the Nintendo DS. Then, I will discuss the main advantage of this device, namely screen size, and provide comparisons to other mobile devices on the market. In the rest of the chapter, I will explain how to program on the Nintendo DS. Since Nintendo refused to release its development tools to the academic field, extra tools are needed to write and run successful applications on this machine.

5.1. General information

In 2004, Nintendo launched its new generation of handheld mobile game machine, called "Nintendo DS". The machine has a clamshell design with two LCD screens, one each on the top and bottom lids. At the time, it offered a novel idea for the mobile gaming world by providing a touch screen system (bottom screen). The console also comes with a built-in microphone and supports the wireless IEEE 802.11 or WiFi standard. This feature can be used to connect with up to 16 other Nintendo DS machines (within 10 to 30 meters) at the same time or to access the Nintendo WiFi Connection service through a wireless router. For controlling, apart from the touch screen system, the machine has four-way directional pad, six single face buttons (A, B, X, Y, Select, and Start), two shoulder buttons (L and R), power button, volume control slide bar, and lid closing-opening control. There are two hardware connection ports, one for the Nintendo DS game card situated on the top side of the case, and the other for the Nintendo Gameboy Advance game card on the bottom side. It is believed that "DS" in the Nintendo DS name stands for Dual Screen, the most distinctive feature of this machine.

In March 2006, Nintendo released the Nintendo DS Lite, which is a redesign of the Nintendo DS. The Lite version is lighter and smaller, and has a brighter screen. The old model is not in production anymore and has been replaced by the Nintendo DS Lite worldwide. Notably, despite the fact that the Lite version is much smaller, the screen size has remained the same as the old model. Another benefit of the Lite version is the new positioning of the buttons, stylus pen and microphone, which makes using the machine easier.

Figure 31. Nintendo DS compared with Nintendo DS Lite

The Nintendo DS is powered by two CPUs, one for each screen. Both CPUs are ARM processors: ARM946E-S main CPU and ARM7TDMI co-processor at clock speeds of 67 MHz and 33 MHz, respectively. There are 4 megabytes of mobile RAM for processing help and 256 kilobytes of Serial Flash Memory for storage. In terms of video display capability, the Nintendo DS is theoretically capable of rendering about 120,000 triangles per second at 60 frames per second, and supports many 3D rendering techniques such as anti-aliasing, cell shading, z-buffering and alpha bending. However, there is only one 3D rendering unit, which means that if the machine simultaneously displays 3D graphics on both screens, the frame rate drops significantly. On the other hand, there are two 2D rendering engines, one per screen, which makes rendering 2D graphics on both screens at the same time an easy task. Texture memory per screen is 512 kilobytes, and the maximum texture size is 1024 x 1024 pixels. For navigational maps, this presents quite a limitation – 1024 x 1024 pixels is not big enough for the whole map, so a map partitioning technique has to be used.

5.2. Display compared with other devices

For small display gadgets, screen size is a major concern. The Nintendo DS has two separate 3-inch TFT LCD screens, with resolution of 256 x 192 pixels, dimensions of 62 x 46 mm with 77 mm diagonal, and dot pitch of 0.24 mm. The screen sizes for other current mobile devices on the market are compared in the figure below.

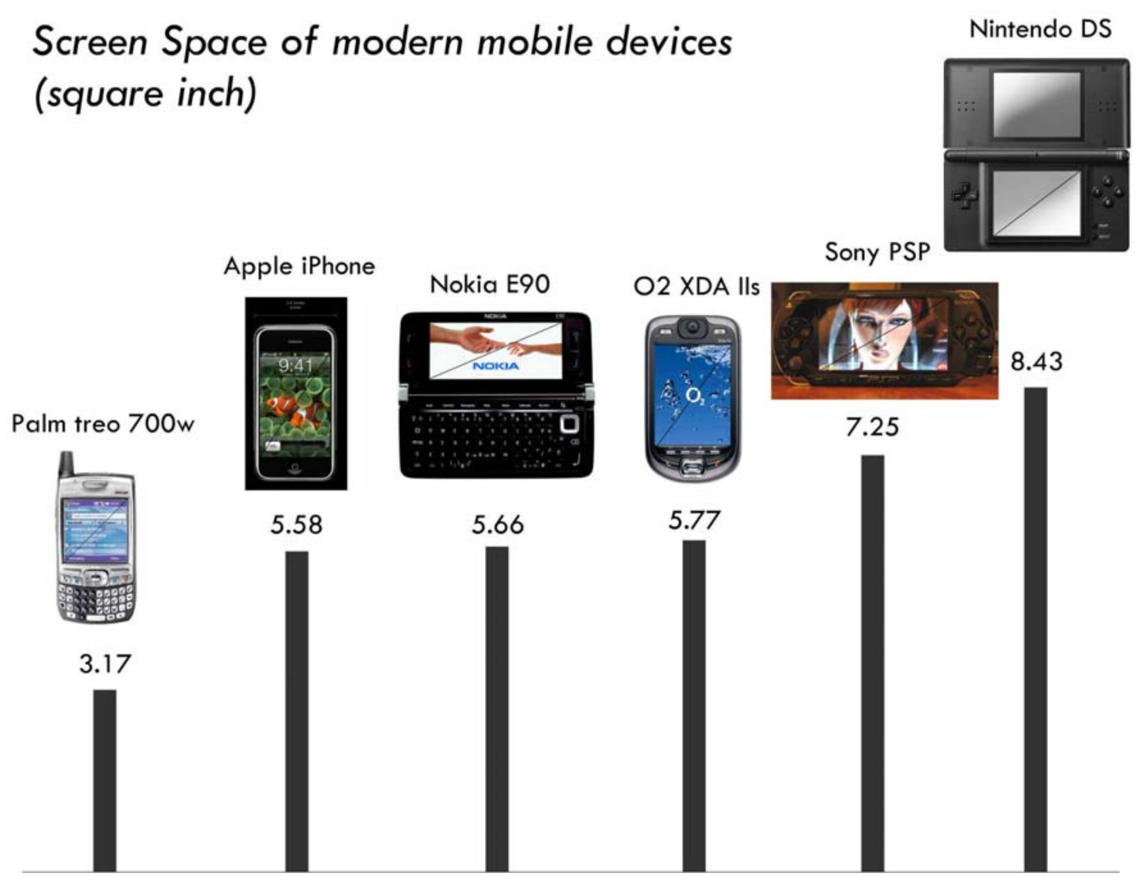


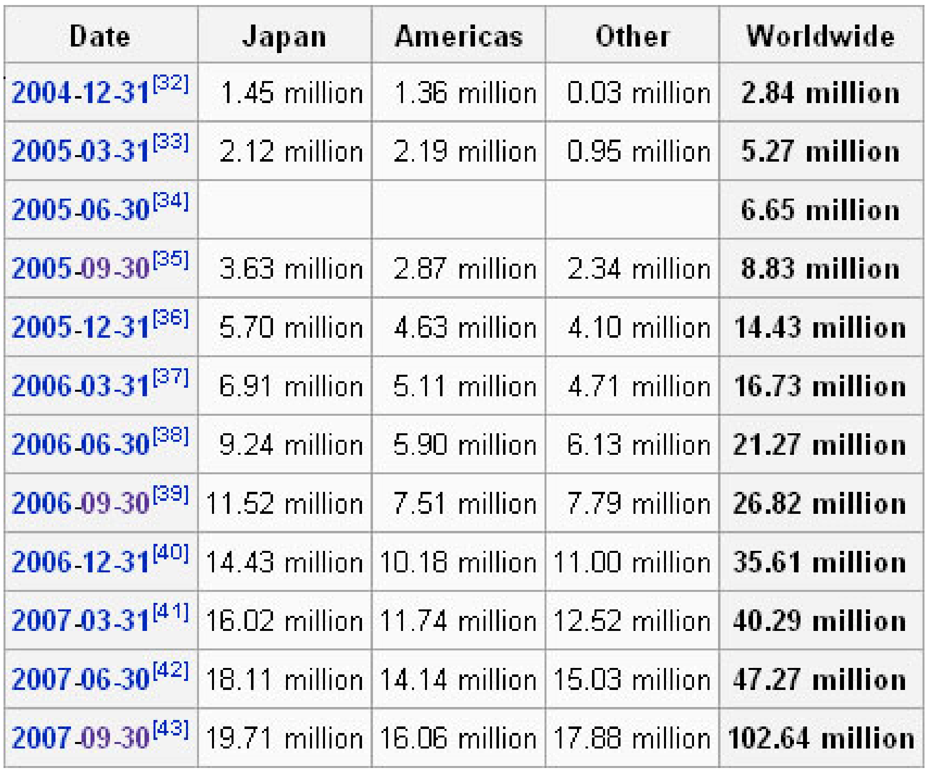
Figure 32. Screen space of modern mobile devices.

The 3-inch screens on the Nintendo DS are considered slightly small by the standards of the latest mobile devices. However, given that it has two screens, the Nintendo DS certainly has the edge in this competition.

On the other hand, the Nintendo DS can display only 260,000 colors, while some other mobile devices can display millions of colors. Also, pixel size in the Nintendo DS is much larger than in comparable devices. As a result, the Nintendo DS is not able to display high definition media or high quality pictures, such as pictures from digital cameras. However, for navigational map purposes, 0.24 mm dot pitch with 260,000 colors is more than sufficient.

5.3. Sales and applications

The Nintendo DS was first launched on November 21, 2004 in Los Angeles, California. The price at that time was set at US $149.99. On December 2, 2004, the console was released in Japan. Australian and European release dates followed in February and March, respectively. Unlike many other media devices such as DVD players, the Nintendo DS is “region-free”. Consequently, there is widespread import and export of units from Japan and the USA (for example, through eBay.com). The Nintendo DS Lite was released in March 2006, further boosting sales. Counting both DS and DS Lite versions, this console is considered the fastest-selling handheld game console of all time, with over 100 million units sold worldwide, and more than 20 million units sold in Japan alone.

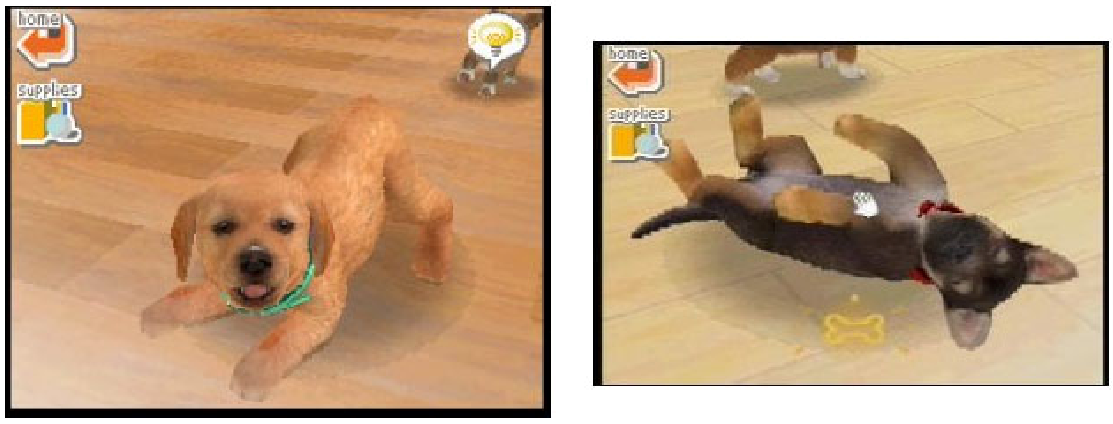


Date	Japan	Americas	Other	Worldwide
2004-12-31[32]	1.45 million	1.36 million	0.03 million	**2.84 million**
2005-03-31[33]	2.12 million	2.19 million	0.95 million	**5.27 million**
2005-06-30[34]				**6.65 million**
2005-09-30[35]	3.63 million	2.87 million	2.34 million	**8.83 million**
2005-12-31[36]	5.70 million	4.63 million	4.10 million	**14.43 million**
2006-03-31[37]	6.91 million	5.11 million	4.71 million	**16.73 million**
2006-06-30[38]	9.24 million	5.90 million	6.13 million	**21.27 million**
2006-09-30[39]	11.52 million	7.51 million	7.79 million	**26.82 million**
2006-12-31[40]	14.43 million	10.18 million	11.00 million	**35.61 million**
2007-03-31[41]	16.02 million	11.74 million	12.52 million	**40.29 million**
2007-06-30[42]	18.11 million	14.14 million	15.03 million	**47.27 million**
2007-09-30[43]	19.71 million	16.06 million	17.88 million	**102.64 million**

Figure 33. Nintendo DS and Nintendo DS Lite sales worldwide [http://en.wikipedia.org/wiki/NintendoDS]

Another factor that greatly contributed to hardware sales was the applications available. The Nintendo DS, as a handheld game console, provides a variety of games to play. However, unconventional and non-gaming software are what give the Nintendo DS a major advantage over rivals such as the Sony Playstation Portable (Sony PSP). First, I will introduce a pet simulation program called Nintendogs, which sold over 15 million copies worldwide. Then, I will introduce a brain training program called Brain Age and last, I will discuss other applications, including Nintendo DS Browser and Nintendo TV.

Nintendogs is a pet simulation video game with virtual dogs. It was originally released by Nintendo in July 2005. Several versions of the game followed through May 2007. In total, over 15 millions copies have been sold, making this game the best selling Nintendo DS game worldwide. The main game-play involves taking care of one or more virtual dogs, just like having real dogs as pets. Basic tasks include feeding, walking, training, and playing with the dogs, but there is much more to do as well. Furthermore, the game-play is based on a real-time clock that is built into the machine, so players must attend to their artificial life pets at least once every one or two days.

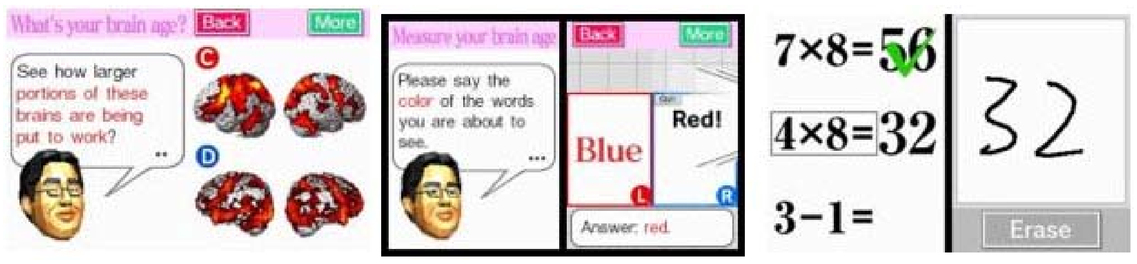


(a) (b)

Figure 34. (a) Give order to dog via microphone, (b) Pet puppy via touch screen [http://www.nintendogs.com]

The touch screen serves as the main control method throughout the game, but the microphone is also heavily used. For instance, players can name their dog and call it through the microphone or teach it actual basic verbal commands like sit or stand. Another main feature is the Bark Mode. In this mode, owners can interact with other players via Nintendo WiFi. This mode is activated when two Nintendo DS machines in hibernate mode (closing the lid without turning power off will put the DS into hibernate mode) come in range of each other. In my opinion, this function was aimed at Japan, where people have plenty of opportunities to come into contact with each other through public transport – especially on trains, the main method of transportation for the Japanese urban population. There are 5 versions of this game in all, with each version offering different dog breeds. Many breeds are specific to each region of the world – for example, Shiba Inu for Japan, Shih Tzu for China and Dalmatian for Europe. Nintendogs has received a lot of awards and recognitions, including “Best Handheld Game of 2005” from the E3 Expo (the biggest electronic entertainment event) and "Best Game of 2005" from the Associated Press (an American news agency and the world's largest such organization).

Apart from pet simulation programs, the Nintendo DS also offers other kinds of unconventional programs to attract users, such as brain training programs. One such program is called “Brain Age: Train Your Brain in Minutes a Day!” or “Brain Age” for short. Brain Age is a puzzle game specially designed to help keep certain parts of the brain active. This program is based on the work of prominent Japanese neuroscientist Dr. Ryuta Kawashima. Dr. Kawashima performed experiments to determine the benefits of certain reading and mathematical exercises toward brain stimulation, which can be replicated through the various input mechanisms of the Nintendo DS, such as touch screen and microphone. As the slogan exclaims, “Exercise Your Brain”, the main concept is to encourage people to play the game at least once a day to regularly stimulate all parts of the brain. The benefits of this game were endorsed by Dr. Elizabeth Zelinski, dean and executive director of University of Southern California's Leonard Davis School of Gerontology, "Americans can do a great deal to maintain and even improve their mental abilities. Aging is about taking on new challenges for our minds. Nintendo's Brain Age is a great way to do that."

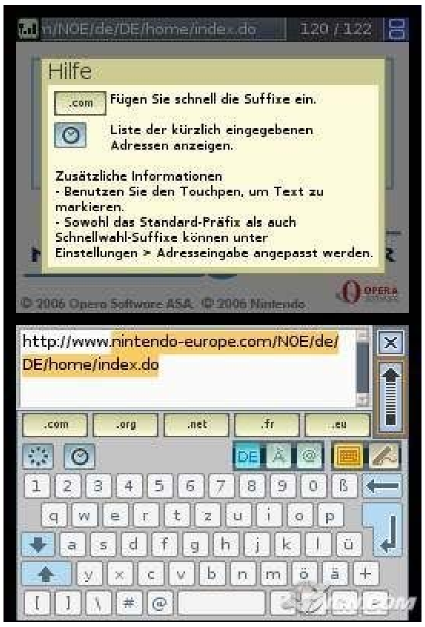

What's your brain age?
Back
More
See how larger portions of these brains are being put to work?
C
D
Measure your brain age
Please say the color of the words you are about to see.
Blue
Red!
Answer: red
7×8=56
4×8=32
3−1=
32
Erase

(a) (b) (c)

Figure 35. (a) Explanation (b) Use of microphone (c) Use of Stylus [http://www.brainage.com]

Research done by Derek Robertson – a teacher who set up Consolarium, the Scottish Centre for Games and Learning – also supported the benefits of Brain Age. Robertson believes that brain training games have positive impacts on both behavior and learning when used in schools. He set up an experiment with three groups of children, two of which were given access to brain training games. Over a one month period, the children who played Brain Age every day scored higher and completed tasks faster than the other group of children. This finding presents a good direction for new interactive learning techniques for children. I personally like the comment by user chakan2 on the gaming website GameSpot [www.gamespot.com] regarding this experiment: "Wow, you get kids to do math for an extra 20-40 minutes a day, and they get better at it...Revolutionary..."

The Nintendo DS also supports various non-gaming applications, such as web browsing. The Nintendo DS Browser is a version of the Opera web browser. Input can be registered with the stylus via an on-screen keyboard or through handwriting recognition. The Nintendo DS browser has two modes, a small screen mode and an overview mode. In the small screen mode, the content of the webpage is rearranged to fit the width of the screen. In the overview mode, a scaled-down version of the webpage is displayed on one screen while a full zoom view is displayed on the other screen. This Opera web browser connects to the internet via the built-in WiFi support, so there must be a WiFi hotspot nearby.


n/NOE/de/DE/home/index.do 120 / 122

Hilfe

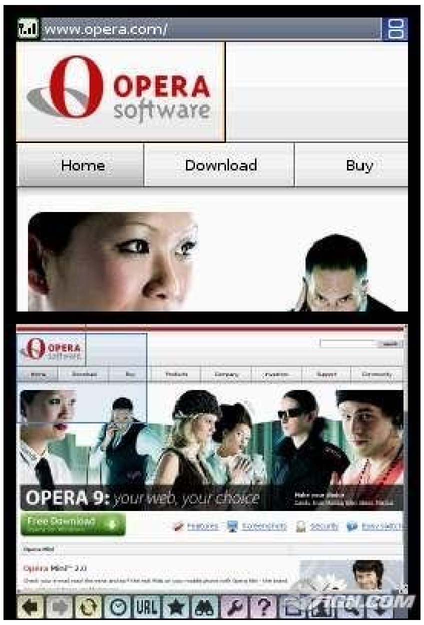
.com Fügen Sie schnell die Suffixe ein.

Liste der kürzlich eingegebenen Adressen anzeigen.

Zusätzliche Informationen
- Benutzen Sie den Touchpen, um Text zu markieren.
- Sowohl das Standard-Präfix als auch Schnellwahl-Suffixe können unter Einstellungen > Adresseingabe angepasst werden.

© 2006 Opera Software ASA. © 2006 Nintendo

http://www.nintendo-europe.com/NOE/de/DE/home/index.do



(a) (b)

Figure 36. (a) On-screen keyboard. (b) Overview mode [http://www.nintendo.com/consumer/systems/dslite/browser.jsp]

Another non-gaming application is TV viewing. A digital TV receiver has been developed for the Nintendo DS, known as the 1seg Receiver Adapter DS Television. It uses a service called 1seg, which is a digital TV provider in Japan. Many new mobile phones and smart phones with G3 capabilities can also use this service to view digital TV channels. The receiver plugs into either game card slot, with the rotatable antenna protruding above the lid. The top screen of the Nintendo DS is used to display video footage, while the bottom screen is used to change channels, adjust volume, and perform other related functions. Currently, this receiver is available only in Japan because the 1seg service only operates in this country.

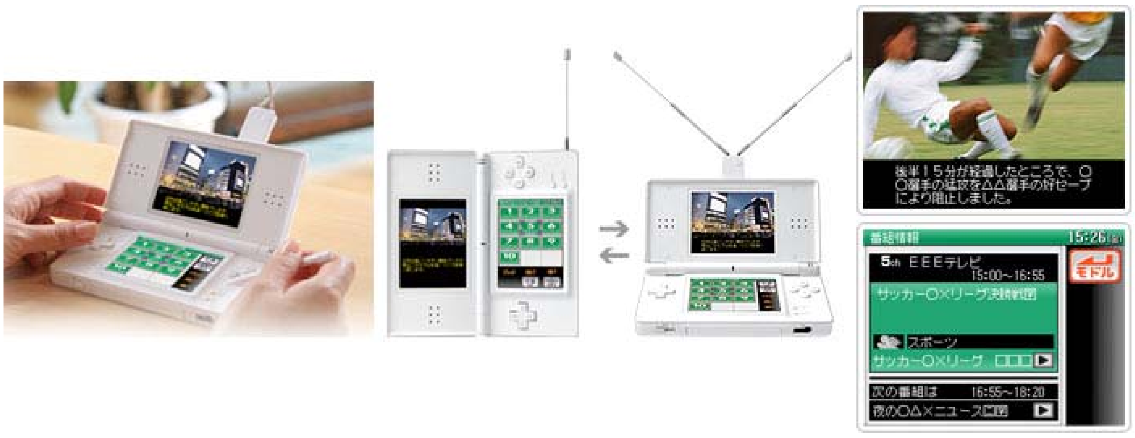

後半15分が経過したところで、○○選手の猛攻を△△選手の好セーブにより阻止しました。

Figure 37. 1seg Receiver Adapter DS Television – Screenshots [http://www.nintendo.co.jp/ds/unsj/index.html]

In this section, I have established that the Nintendo DS is comparable to other mobile devices, such as PDAs and smart phones. Sales of the console are impressive, and the range of available applications is not only limited to gaming. I think that map programs, which constitute the research focus of this thesis, can also be a good addition to the range of application programs for the Nintendo DS.

5.4. Homebrew and Nintendo DS software development kit

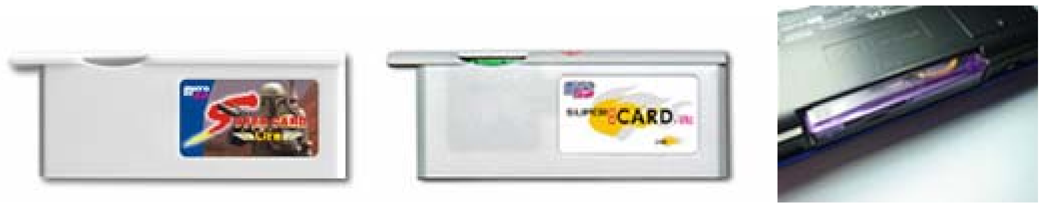
Writing a program on the Nintendo DS is more challenging than on other devices, because Nintendo does not support the academic field. In the more open environment of mobile application development, it is normal for companies to distribute their software development kits for academic purposes. For example, Nokia – one of the leaders in the mobile industry – provides download links to development kits for every series of its mobile phones. Moreover, the Nokia website has a forum that is dedicated to supporting mobile software development. Likewise, Microsoft provides free download of an official software development kit for its game console "Xbox", although this free kit cannot exploit all system features. I myself sent an e-mail to Nintendo's software development department, requesting a Nintendo DS software development kit for use in this thesis. The request was denied in the following reply:

"Hi Sorn - thanks for the email. Unfortunately, because of
confidentiality issues with our technology, we do not distribute
development kits or specifications to students or learning institutions.
Thanks for your interest in Nintendo DS though and good luck with your
project.
Best regards,
Sandy Hatcher
Licensing Manager
Nintendo of America Inc."

However, it is still possible to develop Nintendo DS applications without official development tools through a homebrew approach. Homebrew refers to developing software on proprietary hardware – hardware that is not designed to be programmable by home users, such as video game machines in particular. The first step in a homebrew approach requires hacking into the hardware's firmware – the operating system of the proprietary device. This modification can be made through flash bios or by installing additional hardware, called mod chips, to bypass the system security. Usually, this effort will void the warranty of the product. Software developed by homebrew will generally be positioned as freeware to avoid licensing problems. In Japan, this type of software is called Dojin Soft, which means software by fans.

In order to make a homebrew Nintendo DS program, new firmware must first be installed and the system must then be rebooted. The Nintendo DS, unlike many other devices, does not have protection against firmware overwriting. With new modified firmware, the Nintendo DS will be ready to run homebrew programs. However, the next problem is how to import code into the Nintendo DS machine. This can be accomplished in two ways: through internal Nintendo DS WiFi support or through external hardware that allows the Nintendo DS to read small storage devices, such as Mini SD and Micro SD cards. The WiFi method has the advantage of not needing additional hardware, but imposes a limit of 16 megabytes for software size. This limit is due to the low built-in memory of the Nintendo DS. As a result, the WiFi method is unpopular and only used for emergency procedures, such as reinstalling firmware or performing updates. On the other hand, external devices allow for software size up to 128 megabytes, which is sufficient for any kind of homebrew program.

External devices can be divided into two groups, Slot-1 type and Slot-2 type. The older Slot-2 refers to the Gameboy Advance game card slot situated on the bottom side of the Nintendo DS. Slot-2 devices have certain disadvantages. Firstly, they cannot initiate system hibernation, since a Nintendo DS game card is required to enter hibernate mode. Secondly, they cannot install homebrew firmware by themselves, so another method or boot tool is required for setup. Furthermore, the bigger Slot-2 is normally already used for connecting external accessories, such as external ram, MP3 player or TV Receiver. These problems can be overcome by using the newer and slightly more expensive Slot-1 devices. Since Slot-1 devices are not able to run many homebrew programs written in the Slot-2 format, many coders are now working to convert their homebrew programs to Slot-1 compatible versions. There are several providers of these external devices, including R4 [http://R4DS.com], M3 [http://www.m3adapter.com], and Supercard [http://eng.supercard.cn]. These products are functionally similar but may differ in price, driver, and firmware quality. For this research, I used SupercardLite (Micro), which can extend Nintendo DS storage capacity with a Micro SD card.



(a) (b)

Figure 38. (a) SupercardLite(Micro) (b) Supercard(SD) while inserted in Nintendo DS

After hardware environment setup, next comes software development. DevkitPro [http://www.devkitpro.org] is perhaps the most popular development kit for Nintendo DS homebrew. The entire package consists of development toolchains for Nintendo DS (devkitARM), Nintendo Gamecube (devkitPPC), Sony PSP (devkitPSP), and GamePark GP2X, an open source based handheld game device. The main component of DevkitARM is libnds. Libnds is an open source alternative to Nintendo's commercial SDK for DS. It allows homebrew programmers to create Nintendo DS software. Libnds supports nearly all features of the Nintendo DS, including touch screen, microphone, 3D hardware, 2D hardware, and 802.11b WiFi. In addition, PAlib [http://www.palib.info] is an open source API based on libnds. This library is intended to facilitate homebrew development on the Nintendo DS. At present, more than 40 non-gaming applications have been written with PAlib API – for example, Japanese language learning program, book reading application, NDS media player, and DS-GPS (which requires additional GPS hardware since the Nintendo DS does not have a built-in GPS feature). Nevertheless, PAlib and libnds are still not official SDK. There are several drawbacks in terms of functionality. The clearest drawback is the low quality of emulator. Many commands will not work on the emulator while running tests on a PC, but will work in the real machine. This hampers developers who need to use those functions, because they need to port the code into the Nintendo DS every time they compile to check the outcome. For this thesis, I used PAlib along with devkitPRO to create a prototype navigational map program on the Nintendo DS.

6. Design and implementation

In this chapter, I will explain the design and implementation process behind the navigational map program on the Nintendo DS that was developed for this thesis. As the basis for design, I used the reference model of information visualization by Card et al. (Figure 2) that was discussed in chapter two. This model consists of four main elements (data transformations, visual mappings, visual transformations, and human interactions), which I will discuss separately. In terms of the implementation process that was discussed in the previous chapter involving Nintendo DS homebrew and PAlib API, I will clarify how libnds and PAlib library were used in this project.

6.1. General information

The map that I chose for this project was the map of University of Tampere [http://www.uta.fi], where I am currently studying. This map is a small directional map, so users must orient themselves when looking at this map just as they would do when looking at maps of museums or department stores. In order to look in the right direction, users must make a visual comparison between their surroundings and landmarks in the map. The map of University of Tampere contains some clearly visible landmarks, such as the tube walkway. However, the most important visual guides in this map are the buildings with very unique shapes. These guides facilitate orientation with the map, even for users who have never been to University of Tampere before.

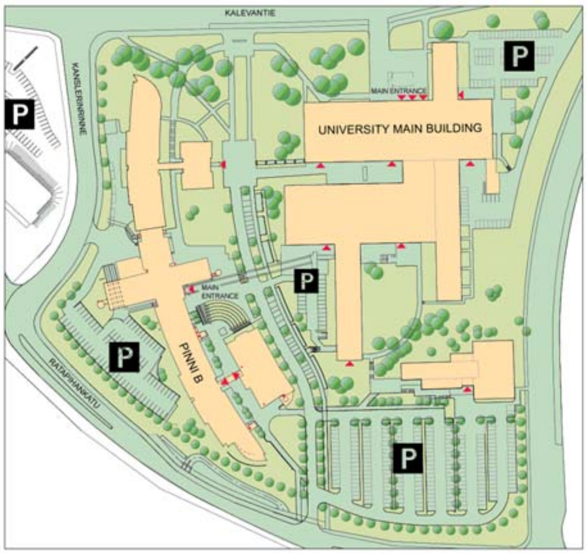
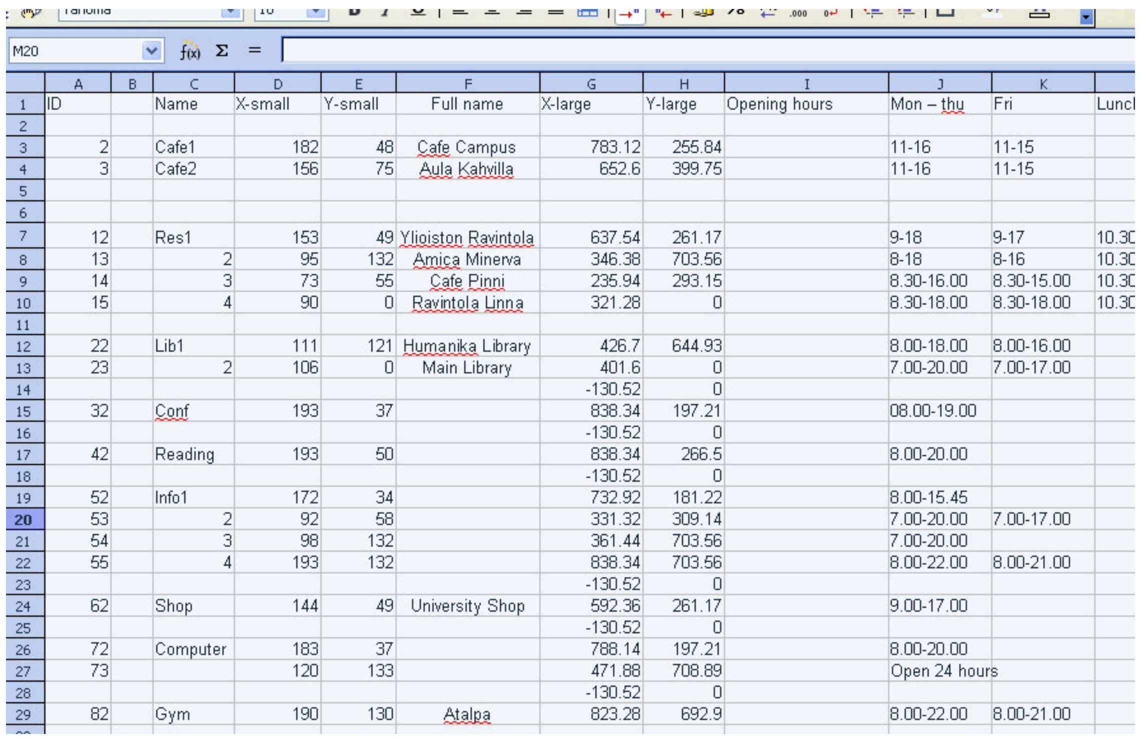


Figure 39. Map of University of Tampere

6.2. Data transformations

The first step of information visualization is data transformation. In this step, raw data will be verified, categorized, ordered, and mapped into a data table. Raw data can be any kind of data. The first piece of raw data, the map of University of Tampere, came in PNG picture format at a size of 929 x 989 pixels. Mobile systems, like the Nintendo DS, are normally hard to program due to various restrictions and limited hardware power. In order to port the picture file for display on the Nintendo DS system, two rules must be observed. The size dimensions must be converted to multiples of two, and the picture must be converted to C code. Additionally, the picture has to be ported into two sizes – one size for the overview which must fit on screen, and another size for the zoom view which must be large enough for users to make visual comparisons with the surrounding area. The size dimensions can be easily converted with Adobe Photoshop by placing a fixed-size white color background under the actual map. The final size of the overview map was 256 x 192 pixels, which is equal to the Nintendo DS screen. The final size of the zoom map was 1024 x 1024 pixels, so the zoom ratio between the two maps was around 6. Picture format was transformed by PAGfx – one of the tools in the libnds toolchains – from .png into several .c files, make the pictures ready for porting into the Nintendo DS machine.

Other data that I chose to fit into this map were the locations of cafés, restaurants, libraries, conference room, reading room, information centers, shop, computer rooms, and gym. In raw data form, this information is given as directions, which are time consuming for locating destinations on a map. Thus, data transformation was needed to transform this raw data into a data table of X-Y coordinates that fit the map. The data for opening hours required no adjustment.



M20

	A	B	C	D	E	F	G	H	I	J	K	
1	ID		Name	X-small	Y-small	Full name	X-large	Y-large	Opening hours	Mon – thu	Fri	Lunc
2												
3	2		Cafe1	182	48	Cafe Campus	783.12	255.84		11-16	11-15	
4	3		Cafe2	156	75	Aula Kahvilla	652.6	399.75		11-16	11-15	
5												
6												
7	12		Res1	153	49	Ylioiston Ravintola	637.54	261.17		9-18	9-17	10.3
8	13		2	95	132	Amica Minerva	346.38	703.56		8-18	8-16	10.3
9	14		3	73	55	Cafe Pinni	235.94	293.15		8.30-16.00	8.30-15.00	10.3
10	15		4	90	0	Ravintola Linna	321.28	0		8.30-18.00	8.30-18.00	10.3
11												
12	22		Lib1	111	121	Humanika Library	426.7	644.93		8.00-18.00	8.00-16.00	
13	23		2	106	0	Main Library	401.6	0		7.00-20.00	7.00-17.00	
14							-130.52	0				
15	32		Conf	193	37		838.34	197.21		08.00-19.00		
16							-130.52	0				
17	42		Reading	193	50		838.34	266.5		8.00-20.00		
18							-130.52	0				
19	52		Info1	172	34		732.92	181.22		8.00-15.45		
20	53		2	92	58		331.32	309.14		7.00-20.00	7.00-17.00	
21	54		3	98	132		361.44	703.56		7.00-20.00		
22	55		4	193	132		838.34	703.56		8.00-22.00	8.00-21.00	
23							-130.52	0				
24	62		Shop	144	49	University Shop	592.36	261.17		9.00-17.00		
25							-130.52	0				
26	72		Computer	183	37		788.14	197.21		8.00-20.00		
27	73			120	133		471.88	708.89		Open 24 hours		
28							-130.52	0				
29	82		Gym	190	130	Atalpa	823.28	692.9		8.00-22.00	8.00-21.00	

Figure 40. Final data table.

6.3. Visual mappings

The second step of information visualization is visual mapping, the process that transforms data tables into the visual structures that users see on their screens. Visual structures have three basic points of concern: spatial substrate, marks, and graphical properties of marks. Spatial substrate refers to the use of space to convey information. For map programs, this is perhaps the most important point of concern because each pixel in the display represents a certain distance that users need to travel in the real world. Unfortunately, I could not find the scale of the map, so I was unable to calculate the distances between the landmarks. These marks constitute the next point of concern. Landmarks are essentially places on the map for displaying the entities in the data table. They can be represented with graphic symbols. The symbols that I chose for each landmark are shown below.

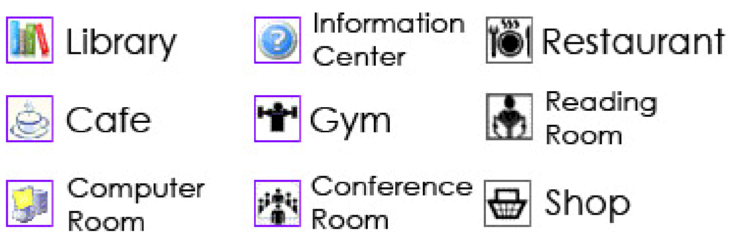


Figure 41. Symbols for landmarks in the map

These icons will appear in both the overview and zoom maps at the same relative locations. When the overview map is on the bottom screen, users can touch the icons on the edge of the screen to get their descriptions. Moreover, when the zoom map is on the bottom screen, users can touch any icon to bring up the opening hours of that location. In total, there are two cafés, four restaurants, two libraries, one conference room, one reading room, four information centers, one shop, two computer rooms, and one gym.

6.4. Visual transformations

The third step of information visualization is visual transformation, the process that creates the views for visual structures. Visual structures themselves are static pieces of information. However, by viewing static visual structures from different angles, users can receive new information from the same visual structures. The four main methods of visual transformation are panning, zooming, fisheye view, and multiple views. The aim of this thesis is to study the possibility of multiple views on mobile machines, which is a rare or non-existent case. The Nintendo DS provides great support for a multiple views interface due to its dual screen system. However, even when discounting the advantages provided by two screens, I think that multiple views transformation is still the best method for viewing navigational maps compared with other techniques.

Panning is the most basic technique, but generally performs the worst in terms of both accuracy rate and completion time. Furthermore, panning often receives the worst scores for user preference. Even though this technique might be inferior, it is still an important component in both zoom and multiple views methods. In the zoom method, panning is employed when users want to navigate the map at a high zoom level without having to repeatedly zoom out and zoom back in at other locations. In the multiple views method, panning is employed when users want to navigate within the zoom window without having to use the controls in the overview window.

On the other hand, fisheye view is an advanced technique, but has proven to be the worst visual transformation technique for map navigation. It receives low scores for user preference, and gives low accuracy rates despite fast completion times. Since the fisheye view destroys the spatial substrate by distorting the values of important visual structures, this particular method was not given consideration.

It is debatable as to whether zooming or multiple views provides the best interface. In my opinion, however, the multiple views interface is superior in this matter. Two research studies found that multiple views had the advantage with regard to user preference. More than 70% of participants preferred multiple views. Furthermore, the comparison model by Plumlee and Ware [2006] predicts that if multiple views and zooming are set up similarly in advance, performance at n = 1 will be comparable. This prediction was supported by Hornbæk et al. [2002] in their Montana map scenario. For this prototype map navigational program, there is no need for users to set up multiple views at all, since the Nintendo DS system already has two separate screens. Moreover, the chosen map for this thesis bears similarity to the Montana map in that both have single level detail – all marks can be seen from the overview level. Under these conditions, multiple views should be on par with zooming in terms of performance, but should have the upper hand in terms of user preference. Another consideration is that zooming can make use of only one screen on the Nintendo DS, since only the bottom screen is a touch screen. In contrast, multiple views can utilize both screens. Overall, it is clear that the multiple views method is the appropriate visual transformation for this thesis.

According to the guidelines for using multiple views in information visualization proposed by Baldonado et al. [2000], there must be a coupling function between the views. This map program uses a coupling technique called navigational slaving – movement in one view is automatically propagated to the other view. More specifically, there is a field-of view box in the overview map, and the area delineated by this box is displayed in the zoom map. If users move the field-of-view box, the content in the zoom map will automatically follow the path of this box. Conversely, if users pan in the zoom map, the field-of-view box in the overview map will move accordingly as well.

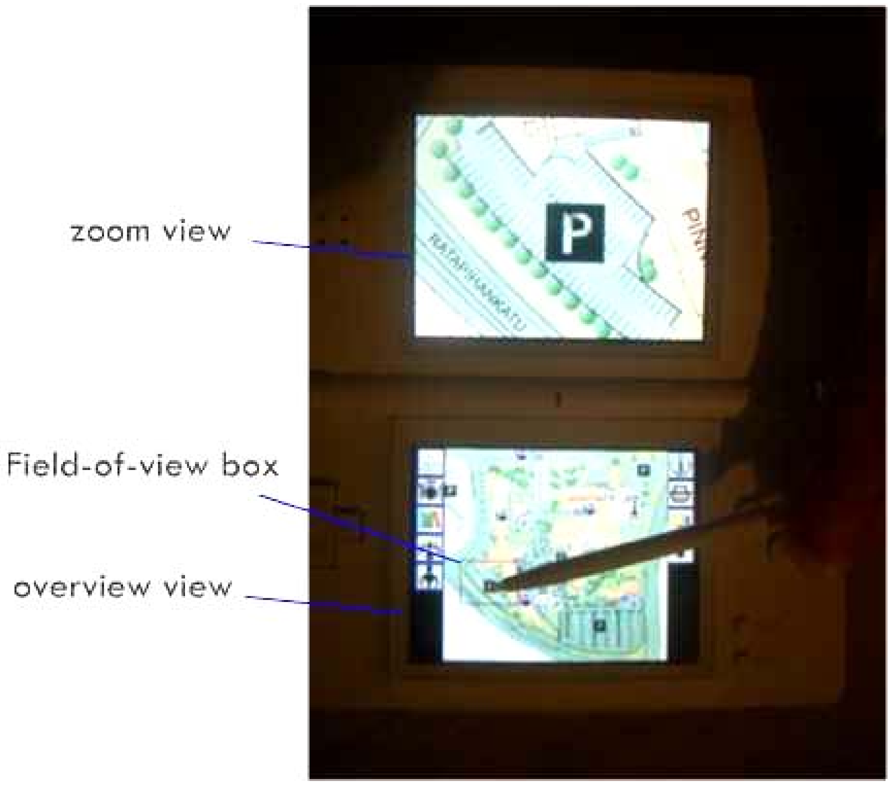


Figure 42. Screenshot of program

According to Woodruff et al [1998], this type of multiple views falls under the rule of diversity, which refers the use of multiple views to observe the same object through different attributes, models, genres, or levels of abstraction. For this map program, different levels of abstraction correspond to different levels of zoom – the overview map has a zoom level of zero (no zoom), while the zoom map has a zoom level of six. The rule of diversity has certain benefits and drawbacks. The main benefit to users is on memory. Displaying extra information on another screen helps reduce the load on the visual memory of users. In this case, it allows users to focus on keeping track of their location. Another benefit is the extra control method for the map – panning in the zoom window or moving the field-of-view box in the overview window. On the other hand, drawbacks include cognitive overhead, use of screen space, and learning curve for new users. Cognitive overhead is unavoidable and always accompanies multiple views. However, Woodruff et al. insightfully pointed out that with multiple views, there is a trade-off between the increased cognitive overhead for using the system and the decreased cognitive tax for no longer having to remember data without extra window support. Meanwhile, screen space is not actually a big concern because the Nintendo DS was designed especially for multiple views. DS refers to the dual screen system, which naturally supports the multiple views technique. Learning curve is not a major obstacle to using the Nintendo DS either, unlike with many other systems. Learning curve is critically tied to affordances – visual cues as to what users can do with system

– and the Nintendo DS presents relatively few affordances to users – namely, directional pad, four face buttons, two shoulder buttons, touch screen and stylus. Therefore, even new users to the system can test out all the buttons within a minute, making the learning curve extremely low.

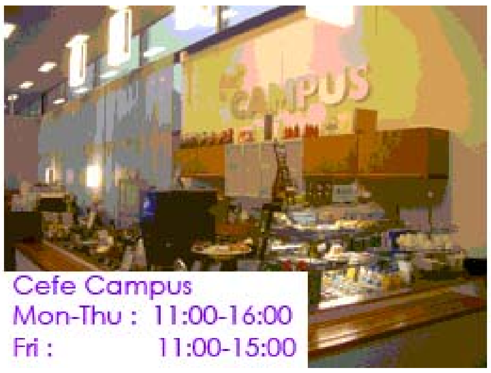
6.5. Human interactions

As with learning curve in the previous section, human interactions are critically tied to affordance. There are two modes of interaction with this navigational map program. The overview mode occurs when the overview window is on the bottom screen, which has touch screen capability. This is the default mode when users open the program. Conversely, the zoom mode occurs when the zoom window is on the bottom screen. Users can switch modes by pressing the L button, which is situated at tip of their left hand index finger when holding the device. This button is also called a trigger button, because it is positioned like a gun trigger.

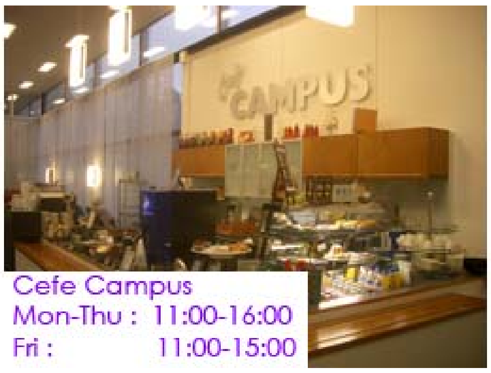
The overview mode is intended to be used to scan the whole map or to quickly move between different areas of the map. The main interaction for this mode is moving the field-of-view box with the stylus. Users can drag the box around, or simply tap an area on the map to reposition the box at that location. It is also possible to move the field-of-view box with the directional pad. The box will move at a static rate in the direction that was pushed. Another interaction for this mode is getting icon descriptions. Users can tap icons at the edge of the screen to get descriptions of those icons. Users can also switch to the zoom mode by pressing the L trigger button. The windows will switch places – overview window to the top screen and zoom window to the bottom screen. In this mode, users can pan around with the directional pad and tap icons to get information about opening hours. Pressing L again will cause the program to revert back to the overview mode.

6.6. Implementations

Both devkitPro and PAlib API are libraries that work in C language. The editor supported by devkitPro is VisualHAM, which is under GNU license. I used this editor along with PAlib library to create a prototype navigational map program. After the code was compiled and tested with emulator on a PC, the executable code was transferred to a Micro SD card and ported into the Nintendo DS via an external storage device. There were some problems while developing the software. The first problem occurred while trying to input photo-quality pictures into the Nintendo DS. The machine itself supports up to 260,000 colors, which should be sufficient. However, the PAGfx graphic converter does not support large pictures with maximum color quality, and places a limit at 256 colors. Moreover, large images consume a lot of video memory, which is quite lacking in the Nintendo DS. So, I decided to drop this idea. The comparative results from this attempt are shown below.



(a)



(b)

Figure 43. (a) 256 colors (b) 16-bit colors

7. Evaluation of results and discussion

This chapter will report the results from the evaluation of the navigational map program that was presented in chapter six. There were a total of eight participants for this evaluation, six males and two females, ranging in age from 19 to 25. All of them were familiar with navigational maps and electronic handheld devices, especially mobile phones and MP3 music players. However, four participants had never seen the Nintendo DS before, while two participants knew about it but had never previously used the system. The other two participants had previous experience with the system but did not own the device.

The test started with an explanation about the purpose of this research, followed by explanations of how to operate the machine and how to use the program. After that, I handed the machine to the participants and allowed them to try using the program. After several minutes, I started asking questions to obtain feedback about this program and this visualization technique. The test was conducted on a one-on-one basis with each participant. Each session took no more than fifteen minutes. User feedback can be divided into three fields of interest: first, effectiveness and satisfaction with the map; second, preference for this visualization technique on the Nintendo DS; and last, comments for further development.

7.1. Effectiveness and user satisfaction with interactive navigational map

To measure effectiveness, I asked participants to use the map to find the opening and closing times of various places on specific days – for example, what time the gym closes on Friday or what time the university shop opens on Saturday. The questions used in this experiment were designed to be difficult enough so that participants would not already know the answers beforehand. Therefore, participants needed to use the map program to find the answers to the questions. After several questions, I asked participants to rate the effectiveness of this map program as well as their satisfaction level with this program. Both responses were on a scale of one to ten, one being worst and ten being best.

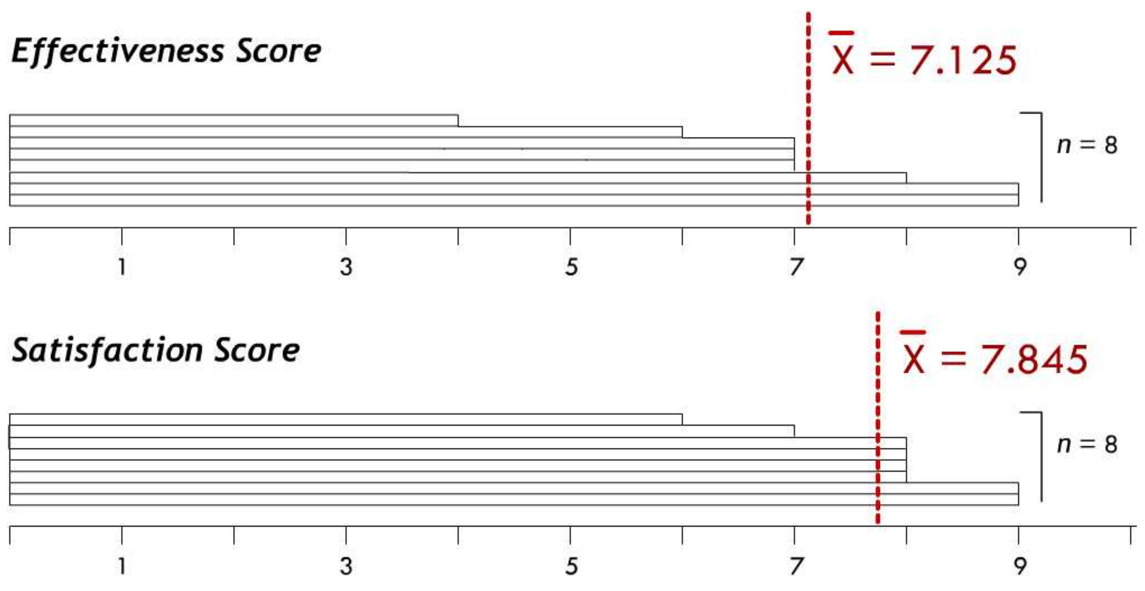


Figure 44. Effectiveness and satisfaction scores.

One participant gave a score of four for effectiveness and six for satisfaction, which were the lowest responses obtained in both categories. On the other hand, two participants gave scores of nine for both effectiveness and satisfaction.

7.2. User preference for this visualization technique on Nintendo DS

Participants generally gave positive feedback on this dual screen multiple views technique. Six participants agreed that the multiple views technique is good even on handheld devices, and is appropriate for navigational map tasks. One participant even commented that "Of course it is appropriate, this is what Nintendo designed for the Nintendo DS, and you don't need to perform research to prove it". With regard to other aspects of the machine, three participants thought that the L trigger button was hard to notice and hard to use – two of these were female. In my opinion, I think this sentiment was due to lack of past experience with gaming machines. Game machine control pads normally have L and R trigger buttons on the shoulder. One participant stated that it was hard for him to simultaneously press the L trigger button while pressing the directional pad with his left thumb – the machine felt unstable in his hands and seemed prone to being dropped.

7.3. User comments for further development

Most of the participants said that they wanted more functionality from this program. Below is a list of important comments.

- The zoom level in the zoom map should be adjustable. Even if it is not possible to zoom out to the same level as the overview map, there should be some flexibility.

- The icons were not clear enough for users to determine the exact boundaries of the locations. One participant cleverly suggested that using color-coded areas would be better. For example, the area for the main building's restaurant would cover the whole left wing of the building – touching anywhere on that area would give the same result as touching the icon. This solution would resolve the problem of overlapping icons as well.
- More information, such as place names, should be given on the map. The participant who gave the lowest scores commented that there should be place names above the icons on the overview map. In his view, it was tedious to find place names because he must first move the field-of-view box over the location, then switch to zoom mode and touch the icon. If it was not the place he was looking for, he needed to switch back to overview mode and repeat the process over again.

The rest of the comments were useful but were not critically concerned with design. For example, one participant suggested that I should add the arrival times for buses at the bus stop in front of the university. Another suggested that this program should be able to show the lunch menus for each restaurant, so that users can decide where to eat. The research was conducted in a friendly manner, which may have resulted in casual conversation with participants, yielding few insightful comments or suggestions.

7.4. Discussion

User preference was as good as expected since the Nintendo DS is a handheld device that was designed especially to provide multiple views visualization. However, the effectiveness and satisfaction scores for the navigational map program was only average. I think this was due to the lack of good, user-friendly design and lack of functionality stemming from the incomplete software development kit. If more functionality had been added to the program, effectiveness and satisfaction scores should have been higher.

However, the negative feedback from participants about the difficulty in using the L trigger button might be difficult to overcome, in my opinion, because the Nintendo DS system has limited affordances. If users hold the stylus in their right hand and the machine in their left hand, then their left thumb will be positioned over the directional pad and their index finger will be positioned at the L trigger button. Users would not be able to access the rest of the buttons without having to put down the stylus. The only available solution that I see so far is to put a switch mode icon on screen so that users can operate it with the stylus. However, this icon will consume precious screen space, especially in zoom mode when the whole of the bottom screen is needed to display content.

Three comments by participants were considered to be particularly important. The first comment was about blending zoom visualization technique into the multiple views technique. This might be a sign that for navigational maps, zooming is always considered valuable. However, this matter still needs to be addressed in further research. The second comment was about using color-coded areas along with icons in the overview map. This idea is very interesting, but it may cause problems in terms of coupling instead. An important criterion for multiple views systems is the connection between views, especially under the rule of diversity. If this concept is to be implemented, care must be taken to ensure sufficient coupling between each view. The last comment was about adding place names on the map. This responsibility falls on me, the programmer. It was my oversight not to put place names on the map because I knew all the places well and forgot to consider that not everyone has been to University of Tampere.

8. Conclusion

This thesis presents research about multiple views visual transformation technique on a mobile device, namely, Nintendo DS, based on constructing a 2D navigational map program. Users were asked to test the program and then give feedback. The first step was to design the prototype map program. I used knowledge from the information visualization field, especially information visualization framework, to design the program. I also compared several research studies about different methods of visual transformation and explained why I chose multiple views as my preference.

The implementation was done by using homebrew software development kits, since Nintendo does not distribute their development kits to the academic field. The language of programming used was C and the developing kits used were PAlib API and devkitPRO package. An external storage hardware, Supercard, was also needed to access this proprietary device. The test was performed with eight participants involved. The effectiveness and satisfaction scores of the program were at medium to good levels, while user preference for this visualization technique on this machine was very high.

During development, I discovered several disadvantages about the Nintendo DS device, such as low graphics memory, low screen resolution, incomplete development kits, and lack of good emulator. However, the Nintendo DS still offers a very unique feature in its dual screen system. In the future, when better homebrew software development kits become available or perhaps when Nintendo releases its official kit to the public, the range of software for this device will be greatly extended. Even now, official Nintendo software, such as Nintendo DS Browser, already makes use of this multiple views technique employing field-of-view box slaving. I am certain that it is just a matter of time before we see official map programs released by Nintendo.

References

[Amant, 1998] Robert St. Amant, Planning and user interface affordances, *Proceedings of the 4th international conference on intelligent user interfaces*, Los Angeles, California, United States, Pages: 135 – 142, 1998

[Baldonado et al., 2000] Michelle Q. Wang Baldonado, Allison Woodruff, Allan Kuchinsky, Guidelines for using multiple views in information visualization, *Proceedings of the working conference on Advanced visual interfaces*, Palermo, Italy, Pages: 110 - 119, 2000

[Bedersen and Boltman, 1999] Bederson, B.B. Boltman, A., Does animation help users build mental maps of spatial information*? Information Visualization, 1999. (Info Vis '99) Proceedings. 1999 IEEE Symposium*, Pages: 28 – 35, October 1999

[Card et al., 1991] Stuart K. Card, George G. Robertson, Jock D. Mackinlay, The information visualizer, an information workspace*, Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, New Orleans, Louisiana, United States, Pages: 181 – 186, 1991

[Card et al., 1999] Stuart K. Card, Jock Mackinlay, Ben Shneiderman, *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann Publishers, 1999.

[Fisher et al., 1997] Brian Fisher, Makrina Agelidis, John Dill, Paul Tana, Gerald Collaudb, Chris Jonesc, CZWeb: Fish-eye Views for Visualizing the World-Wide Web, In *Proceedings of the 7th International Conference on Human-Computer Interaction (HCI International'97)*, pages 719--722, 1997.

[Guo et al., 2000] Guo, Hou, Zhang, Wei Wei, and Wu Jing, The effect of zooming speed in a zoomable user interface, *Report from student HCI Online Research Experiments (SHORE)*, 2000

[Gutwin and Fedak, 2004a] Carl Gutwin, Chris Fedak, A comparison of fisheye lenses for interactive layout tasks, *Proceedings of Graphics Interface 2004 GI '04*, Canadian Human-Computer Communications Society, May 2004.

[Gutwin and Fedak, 2004b] Carl Gutwin, Chris Fedak, Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques, *ACM International Conference Proceeding Series*; Vol. **62** archive, Proceedings of Graphics Interface 2004, London, Ontario, Canada, Pages: 145 – 152, 2004

[Hornbæk and Frøkjær, 2003] Kasper Hornbæk and Erik Frøkjær, Reading patterns and usability in visualizations of electronic documents*, ACM Transactions on Computer-Human Interaction (TOCHI),* Volume **10**, Issue 2, Pages: 119 – 149, June 2003

[Hornbæk et al., 2002] Kasper Hornbæk, Benjamin B. Bederson, Catherine Plaisant, Navigation patterns and usability of zoomable user interfaces with and without an overview, *ACM Transactions on Computer-Human Interaction (TOCHI),* Volume **9**, Issue 4, Pages: 362 – 389, December 2002

[http://dictionary.reference.com] *a multi-source dictionary search service*, Available from http://dictionary.reference.com, [Accessed 12 December 2007]

[http://en.wikipedia.org/wiki/Map] *Wikipedia, the free encyclopedia*, Available from http://en.wikipedia.org/wiki/Map, [Accessed 12 December 2007]

[http://eng.supercard.cn] *Supercard official website,* Available from http://eng.supercard.cn, [Accessed 16 December 2007]

[http://R4DS.com] *R4DS cartridge official website*, Available from http://R4DS.com, [Accessed 16 December 2007]

[http://www.brainage.com] *Brain Age official website*, Available from http://www.brainage.com, [Accessed 12 December 2007]

[http://www.devkitpro.org] *Devkitpro homebrew website*, Available from http://www.devkitpro.org, [Accessed 16 December 2007]

[http://www.gamespot.com] *Video games website,* chakan2, Available from http://www.gamespot.com, [Accessed 12 December 2007]

[http://www.m3adapter.com]. *M3 official website,* Available from http://www.3adaptor.com, [Accessed 16 December 2007]

[http://www.nintendo.co.jp/ds/unsj/index.html] *1seg television news on Nintendo Japan official website*, Available from http://www.nintendo.co.jp/ds/unsj/index.html, [Accessed 12 December 2007]

[http://www.nintendo.com/consumer/systems/dslite/browser.jsp] *Nintendo Browser official customer support website,* Available from http://www.nintendo.com/consumer/systems/dslite/browser.jsp, [Accessed 16 December 2007]

[http://www.nintendo.com/ds] *Nintendo DS home,* Available from http://www.nintendo.com/ds, [Accessed 12 December 2007]

[http://www.nintendo.com] *Welcome to Nintendo homepage,* Available from http://www.nintendo.com, [Accessed 12 December 2007]

[http://www.nintendogs.com] *Nintendogs for Nintendo DS official website,* Available from http://www.nintendogs.com, [Accessed 12 December 2007]

[http://www.palib.info] *Palib library for Nintendo DS homebrew website*, Available from http://www.palib.info, [Accessed 16 December 2007]

[http://www.tfl.gov.uk] *Transport for London Homepage,* Available from http://www.tfl.gov.uk, [Accessed 12 December 2007]

[http://www.us.playstation.com/PS2/Accessories] *Accessories for Sony Playstation,* Available from http://www.us.playstation.com/PS2/Accessories, [Accessed 12 December 2007] [http://www.usgs.gov] *U.S. Geological Survey Homepage,* Available from http://www.usgs.gov, [Accessed 12 December 2007]

[http://www.uta.fi]. *University of Tampere website*, Available from http://www.uta.fi, [Accessed 16 December 2007]

[Hyrskykari et al., 2000] Aulikki Hyrskykari, Päivi Majaranta, Antti Aaltonen, Kari-Jouko Räihä, Design issues of iDICT: a gaze-assisted translation aid, *Proceedings of the 2000 symposium on Eye tracking research & applications*, Palm Beach Gardens, Florida, United States, Pages: 9 – 14, 2000

[Johnson, 1995] Jeff A. Johnson, A comparison of user interfaces for panning on a touch-controlled display, *Proceedings of the SIGCHI conference on Human factors in computing systems*, Denver, Colorado, United States, Pages: 218 – 225, 1995

[North and Shneiderman, 1997] Chris North and Ben Shneiderman, A Taxonomy of Multiple Window Coordinations, *University of Maryland Computer Science Dept. Technical Report*, number: CS-TR-3854, 1997, Also available as *http://citeseer.ist.psu.edu/189031.html*

[Plumlee and Ware, 2006] Matthew D. Plumlee, Colin Ware, Zooming versus multiple window interfaces: Cognitive costs of visual comparisons, *ACM Transactions on Computer-Human Interaction (TOCHI) archive*, Volume **13**, Issue 2, Pages: 179 – 209, June 2006

[Schafer et al., 2002] Wendy A. Schafer, Doug A. Bowman, John M. Carroll, Map-based navigation in a graphical MOO, *Crossroads*, Volume **9** , Issue 1, Pages: 8-15, Fall 2002

[Smith and Taivalsaari, 1999] Randall B. Smith, Antero Taivalsaari, Generalized and stationary scrolling, *Proceedings of the 12th annual ACM symposium on User interface software and technology*, Asheville, North Carolina, United States, Pages: 1 - 9, 1999

[Spence, 2001] Robert Spence, ACM Press, *Information Visualization*, 2001

[Woodruff et at., 1998] Allison Woodruff, James Landay, Michael Stonebraker, Goal-Directed Zoom, *SIGCHI'98 Summary*, pages 305-6, April 1998.