Interaction Design Guide for Touchscreen Applications

Print version (PDF)

Read First

1. Introduction

2. Design Guidelines

3. Additions

4. Touchscreen Control Zoo


top

Interaction Design Guide for Touchscreen Applications (Experimental)

What's in the Interaction Design Guide? | Status

What's in the Interaction Design Guide?

This **Interaction Design Guide** presents guidelines for **finger-operated touchscreen applications**. It is a collection of guidelines drawn from the literature, from the web, and from own experiences in the design of applications using finger-operated touchscreens.

Note: These guidelines do not cover pen-operated touchscreens as typically used in handheld computers and palmtops.

Status

This **guide** is **not** an official SAP document. It is provided on an "as is" basis, and is based on the personal work of the author.

Currently, these guidelines are in an experimental status. They are offered in order to fill a gap in the SAP guidelines. They also do not adhere to any visual guidelines issued by SAP (currently SAP does not offer visual guidelines for touchscreen devices).

Note: There will probably be no further maintenance for these guidelines.

Version 0.5, December 2000

Gerd Waloszek, Product Design Center
gerd.waloszek@sap.com

This guideline can be found in *Resources* on the SAP Design Guild Website (www.sapdesignguild.org).

 top

Source: Interaction Design Guide for Touchscreen Applications

Introduction

Touchscreens - Interaction of the Future? | Limitations of Touchscreens | The Ultimate Judge is the User

The **Interaction Design Guide for Touchscreen Applications** presents guidelines for designing the user interface for touchscreens. Typically touchscreens are operated at POS stations (point-of-sale), in museums, as city guides, or as kiosks (stand-alone advertising booths with user interaction). Users are often untrained, and interaction has to be simple and fast. These situations require a screen and interaction design, which differs considerably from normal user interface design.

Figure 1: Finger-operated touchscreen (from ELO Touch Systems)

Touchscreens - Interaction of the Future?

Touchscreens are operated with a **finger** or **stylus**. Therefore, they provide a very **direct** interaction - the most direct interaction that is possible on computers today. Touchscreens may be operated very **fast** for certain operations and require **little** or **no training**, if applications are designed adequately.

For these reasons touchscreens have many uses, especially for untrained users. Many people believe that touchscreens will replace keyboard and mouse in the future.

Limitations of Touchscreens

However, touchscreens have **limitations** that - to our opinion - prevent them from becoming the "universal" solution for user interface problems.

Size

Fingers have a certain **size**. So, screen elements have to have a minimum size, to ensure that a touchscreen can be operated with few **errors**. Even with a stylus, which makes possible to use smaller screen elements, there are limiting factors.

Sequential Input

Input on a touchscreen is inherently sequential: One finger is used for clicking. This **slow**s input down compared to keyboard input where several fingers can be used virtually in parallel.

Strain

Keying in many numbers or letters by pointing with the finger is also very straining and tiring. Therefore, touchscreens make no sense in workplaces, where much text or number input is required.

Feedback

On touchscreens, there is no analogue to **mouse-move events**. Mouse users can move the mouse pointer over screen elements, get **feedback** about the selected element (e.g. by highlighting), and may confirm the selection by clicking the mouse button. Touchscreen users directly point on a screen element. If they are lucky, they can withdraw their finger if they touched the wrong screen element. On other touchscreens, the touch immediately initiates an action - there is no opportunity to **cancel** the action.

Drag Operations

Dragging is generally not well suited to finger-operated touchscreens; here pointing is the preferred interaction. However, this is different for stylus-operated touchscreens. Here **gestures** and **handwriting** offer promising possibilities for making interaction with computers easier and more intuitive. But here, too is the limitation of strictly sequential input.

There are also no means to constrain drag operations to, e.g. straight lines, like with mouse-based interfaces.

The Ultimate Judge is the User!

These preliminary design guidelines for touchscreens take the characteristics of touchscreens, their advantages and disadvantages, into account. Of course, these guidelines will be refined, as our experience with touchscreens grows. The ultimate judge, however, is the **user**. As ELO Touch Systems quote in their guidelines:

- Testing a touchscreen application on focus groups will disclose the areas that need improvement.
- If anyone pauses in confusion for even a moment, think how to improve the application.

 top

Source: **Interaction Design Guide for Touchscreen Applications**

Pros and Cons of Touchscreens

Touchscreen Pros | Touchscreen Cons | Summary of Touchscreen Characteristics

The following overview lists advantages and disadvantages of touchscreens and summarizes their characteristics.

Touchscreen Pros

- **Direct**: Direct pointing to objects, direct relationship between hand and cursor movement (**distance**, **speed** and **direction)**, because the hand is moving on the same surface that the cursor is moving, manipulating objects on the screen is similar to manipulating them in the manual world
- **Fast** (but less precise without pen)
- **Finger** is usable, any **pen** is usable (usually no cable needed).
- **No keyboard** necessary for applications that need menu selections only -> saves desk space
- **Suited to**: novices, applications for information retrieval, high-use environments.

Touchscreen Cons

- **Low precision (finger)**: Imprecise positioning, possible problems with eye parallaxis (with pen, too), the finger may be too large for accurate pointing with small objects -> a pen is more accurate.
- **Hand movements (if used with keyboard)**: Requires that users move the hand away from the keyboard; a stylus requires also hand movements to take up the pen.
- **Fatigue**: Straining the arm muscles under heavy use (especially if the screen is placed vertically).
- **Sitting/Standing position**: The user has to sit/stand close to the screen.
- **Dirt**: The screen gets dirty from finger prints.
- **Screen coverage**: The user's hand, the finger or the pen may obscure parts of the screen.
- **Activation**: Usually direct activation of the selected function, when the screen is touched; there is no special "activation" button as with a light pen or a mouse.

Summary of Touchscreen Characteristics

- **Speed**: high
- **Accuracy**: low (finger), high (pen)
- **Speed control**: yes
- **Continuous movement**: yes
- **Directness**: direction, distance, speed
- **Fatigue**: high
- **Footprint**: no
- **Best uses**: point, select

top

Source: **Interaction Design Guide for Touchscreen Applications**

Comparison of Finger-Operated vs. Stylus-Operated Touchscreens

The following table compares finger-operated with stylus-operated touchscreens.

Comparison

	Finger-Operated	Stylus-Operated
Interaction	Click (, drag, duration)	Click, drag, (double click, duration)
Operations	Point, select	Point, select, define path (start and goal, path): drawings, gestures, handwriting
Speed	High	High
Accuracy	Low	High (comparable to mouse)
Size of Controls	Large	Small (as with mouse)
Text Entry	Not recommended (, through selection)	Through handwriting (, through selection -> or only selection of starting letters)
Number Entry	Through selection	Through handwriting or selection
Initiation of Actions	Through point-and-click (pushbuttons)	Through point-and-click (pushbuttons), through selection (e.g. dropdown lists) through gestures
Preferred Interface	Point-and-click interface	"Standard" GUI possible, better: optimized interface for pen (-> gesture entry), point-and-click interface if speed is required
Environment	High speed, low accuracy, "aggressive" environment pen is disturbing (taking up the pen, dirt, loss)	High speed (not mandatory), high accuracy pen usage possible

GUI Elements	Pushbuttons, controls for display and selection of data, graphics	Nearly all standard GUI elements may be used (but not all are optimal); avoid scrollbars, pulldown-menus (at least these should be spring-loaded)


top

Source: **Interaction Design Guide for Touchscreen Applications**

Uses for Touchscreens

Best Suited to Applications Where... | Not Suited to Applications... | Typical Touchscreen Systems

Best Suited to Applications Where...

- Opportunity for training is low
- Frequency of use is low
- Accurate positioning is not required
- Little or no text or numerical input is required
- Desk space is at a premium
- The environment may be chemically or otherwise "aggressive"

Not Suited to Applications...

- Requiring training/trained users
- With high-frequency use
- Requiring accuracy
- Requiring a lot of typing

Typical Touchscreen Systems

- Public information systems: Museums, city guides
- Kiosk systems: Advertising, product information
- Systems requiring pointing and selection only

Examples

For more examples see Examples.

Figure 1: Kiosk System (from Information Kiosk Systems)

Figure 2: Typical flat LCD touchscreen for many uses (from Information Kiosk Systems)

Figure 3: Touch panel for controlling devices (from CUE)

top

Source: Interaction Design Guide for Touchscreen Applications

Examples

Kiosks | Information Systems | POS Applications | Device Control

The following examples for touchscreen devices and applications are taken from various sources on the Internet.

Kiosks

Kiosk systems are used:

- For advertising, product information
- As public information systems, for example, in museums, as city guides

Kiosks are typically embedded in some sort of cabinet, so that they can stand alone. They are built for rugged environments and simple point-and-click interaction.

Figure 1: Kiosk System for use in the public (from Information Kiosk Systems)

Figure 2: A larger "cabinet type" kiosk system used in an information booth, for example on a conference (from Information Kiosk Systems)

Figure 3: A rugged touchscreen device for use in store shelves (from Information Kiosk Systems)

Information Systems

Touchscreen devices can be used for advising customers or clients. This can happen together with a consultant like a physician or a salesman, or the users do this alone.

Often it is hard to separate information systems from kiosk systems. Here, we understand information systems as systems, which are not specifically embedded into cabinets or shelves.

Figure 4: Consultation supported by a flat touchscreen panel (from Information Kiosk Systems)

POs Applications

POs applications (point-of-sale) are touchscreens that are used in combination with a cash register or similar devices.

Figure 5: POs systems are the combination of a cash register with a touchscreen device (left: from DPI Limited; right: from EMAX International Incoporated)

Device Control

Touchscreen panels can be used for controlling devices, for example audio or video equipment. Such panels can be flexibly programmed and configured, and thus avoid the necessity of complex button panels or keyboards.

Figure 6: Touch panel for controlling devices (from CUE)

top

Source: **Interaction Design Guide for Touchscreen Applications**

Golden Rules

Speed

- Make your application run fast. Speedy systems also reduce vandalism.

Intuitiveness

- Try to make the application intuitive.

Choices

- Limit choices.

Guidance

- Guide the user as much as possible.

Testing

- Testing a touchscreen application on focus groups will disclose the areas that need improvement. If anyone pauses in confusion for even a moment, think how to improve the application.

 top

Source: **Interaction Design Guide for Touchscreen Applications**

Touchscreen Usability in Short

Events

- **Contact** (touch down): error cancellation a problem!
- **Loosing contact** (touch release): Provides possibility to cancel errors.
- There is **no** analogue to **mouse-move** events (no comparable feedback possible).
- There is **no** analogue to **mouse button** - pointing and initiating are combined into one step.

Interaction

- **Click**: Initiate actions, make selections, specify positions
- **Double click** (cumbersome): Initiate actions, confirm actions
- **Drag**: Define paths (start, goal), draw simple figures or gestures
- **Duration** (long-lasting Touch; do not know applications): Define durations, enter values

Basic Operations

- **Best**: Point, select
- Position, orient (rotate), define path
- Enter values
- **Worst**: Enter text

Cursors & Pointers, Status Display, Feedback

Mouse pointer vs. active location (in mouse-based interfaces)

- **No mouse pointer** on touchscreens
  - Users do not lose track of their fingers (but they often loose the mouse pointer).
  - Users cannot confuse different types of cursors or activated controls.
  - There is no analogue to mouse pointer as **status** display.
- **Static status display**: Can be provided through highlighting (color, frame, 3D look etc.) or animation.
- **Dynamic status display (Feedback)**: Can be provided **only during** the action (and maybe afterwards): The objects themselves have to indicate whether an action is possible/allowed or not! This can be done similar to static activation.

Experimental Results

- **Finger-operated touchscreen**: Best in speed and worst in accuracy (Albert, 1982).
- **Stylus-operated touchscreen**: Comparable to a mouse on both speed and accuracy measures (Mack & Lang, 1989).

- **Initiation of actions:** Take-off strategy significantly reduced errors (caused less than half of the errors compared to land-on strategy). Target selection with take-off strategy took longer (about 20%) (Potter, Shneiderman & Weldon, 1988).

Touchscreen Pros

- **Direct**: Direct pointing to objects, direct relationship between hand and cursor movement (**distance**, **speed** and **direction)**, because the hand is moving on the same surface that the cursor is moving, manipulating objects on the screen is similar to manipulating them in the manual world
- **Fast** (but less precise without pen)
- **Finger** is usable, any **pen** is usable (usually no cable needed).
- **No keyboard** necessary for applications that need menu selections only -> saves desk space
- **Suited to**: novices, applications for information retrieval, high-use environments.

Touchscreen Cons

- **Low precision (finger)**: Imprecise positioning, possible problems with eye parallaxis (with pen, too), the finger may be too large for accurate pointing with small objects -> a pen is more accurate.
- **Hand movements (if used with keyboard)**: Requires that users move the hand away from the keyboard; a stylus requires also hand movements to take up the pen.
- **Fatigue**: Straining the arm muscles under heavy use (especially if the screen is placed vertically).
- **Sitting/Standing position**: The user has to sit/stand close to the screen.
- **Dirt**: The screen gets dirty from finger prints.
- **Screen coverage**: The user's hand, the finger or the pen may obscure parts of the screen.
- **Activation**: Usually direct activation of the selected function, when the screen is touched; there is no special "activation" button as with a light pen or a mouse.

Summary of Touchscreen Characteristics

- **Speed**: high
- **Accuracy**: low (finger), high (pen)
- **Speed control**: yes
- **Continuous movement**: yes
- **Directness**: direction, distance, speed
- **Fatigue**: high
- **Footprint**: no
- **Best uses**: point, select

top

Source: **Interaction Design Guide for Touchscreen Applications**

Screen Layout

Screen

Full screen

Applications should be run full screen.

Backgrounds

- Bright background colors (no black!) hide fingerprints and reduce glare
- Dithering or other patterned backgrounds help the eye focus on the screen image instead of reflections, even in areas where there are no icons or menu choices

Handedness

As many users are left-handed, the screen layout should be switchable between a version for right-handed users and a version for left-handed users.

Screen Reversal

The screen reversal should affect mainly the large building blocks of the screen layout. Whether the controls themselves should also be mirrored, depends on how much interaction is adversely affected by the standard layout of the control. For instance, it depends on, whether a standard control is obscured, if operated with the left hand.

Screen Space and Screen Layout

Touchscreens usually have a fixed size of 640*480 pixels or 800*600 pixels. As buttons have to have a minimum size and distance (see "Buttons") for secure finger operation, screen space has to be managed in a different way than on usual screens.

We propose the following procedure for the screen layout:

1. Use a **grid** that divides the screen into a fixed number of cells that have a constant size (see below).
2. In a second step identify **functional groups** of elements that are placed close together and organize them into **functional areas**. Such areas can serve for
    - data display
    - data input or selection
    - grouping of function keys (toolbars)
    - numerical or text keypad
    - status display

Global vs. Local Structure

Finger-Operated

- Global structure is most important.
- Use a simple local structure (not too detailed).

Stylus-Operated

- Global structure is important.
- Local structure may be more detailed, if there are no high demands on speed and accuracy.

Placement of Screen Elements

When you divide the screen in functional areas and place elements on the screen, observe the principle of **location constancy**:

- Keep functional areas stable, e.g. use fixed areas for data display, input, buttons (especially keep the navigation buttons constant), and status display.
- Try to follow this principle not only for functional areas, but also for the actual GUI elements.
- However, as screen space usually is limited, you may exchange GUI elements within the fixed functional areas.

Grouping of Screen Elements

Use the following attributes for **grouping** screen elements:

- Arrangement (flight lines, Gestalt laws (proximity, similarity, ...), white space)
- Frames
- Shape
- Foreground color, background color/texture
- Text size, text attributes

Use attributes **redundantly** for easier identification of groups.

Observe the **Gestalt laws** (proximity, similarity, ...) when you arrange and group screen elements.

Do not base grouping on **color** alone. Use at least enough brightness contrast, so that color blind users can distinguish the colors, too.

top

Source: **Interaction Design Guide for Touchscreen Applications**

Maintaining Screen Space

Tabstrip | "Switchable Areas" | Screen Changes | Do Not...

As touchscreens usually are of fixed and relatively small size (640*480 or 800*600 pixels) and screen element have to be large enough to be operable by fingers, screen space is at premium with touchscreens. Here are some proposals for overcoming the space limitations of touchscreens.

Tabstrip

A **tabstrip** can be used to virtually enlarge a screen by providing a device that enables the users to switch between different views.

A tabstrip consists of a "tab area" which contains buttons for switching the view, and of a "presentation area" which is usually of fixed size (it may be scrollable) and displays the different views.

The buttons allow to access to views in random order.

"Switchable Areas"

A similar technique consists in assigning certain screens areas to fixed functions like data display, function key area, keypad area etc. Pushbuttons can be used to exchange to elements in such a fixed area, for instance to switch between

- different list views
- different versions of a keypad
- different button sets or groups

Screen Changes

Users may navigate between screens which, for instance, provide access to different components of a compound data object. However, this technique may provide problems for nonprofessional users - they may loose orientation and may have problems processing their tasks. Therefore,

- give the screens a consistent layout and look, so that users perceive them as a unity
- make clear which the navigational options are, e.g. how users can proceed and backtrack
- limit he number of screen changes to a small number

Do Not...

- Do not make the screen scrollable like long dynpros in R/3 or HTML pages!!!

- Do not use a multiple-window interface as Windows or Mac OS have!!!

 top

Source: Interaction Design Guide for Touchscreen Applications

Data Entry

Numbers and Letters | Values

Touchscreens are not well suited for data entry. In addition, there are few touchscreens which provide an additional external numeric or alphanumeric keypad.

Numbers and Letters

Data entry is typically provided through virtual keypads or keyboards, which can be placed on the screen on demand.

For applications where numbers or letters are frequently entered, the virtual keypad should stay permanently on the screen.

Examples (permanent keypad): Front office, bar counter

Figure 1: Numeric virtual keyboards in keypad (left) and telephone layout (right)

Figure 2: A virtual alpha keyboard (from Information Kiosk Systems)

For more information on keypads, see Interaction.

Values

Entering values can become very tedious on touchscreens. Value entry should be restricted to few values. The basic interaction modes are:

- Clicking on predefined values (options) = selection
- Clicking repeatedly on certain controls (e.g. to increment or decrement values)
- Dragging sliders or similar controls
- Entering digits or numbers through a keypad or keyboard

These modes should be used in priority from top to bottom; that is, first a solution should be tries that does with direct clicking; only if space restrictions or value ranges require other solutions, these should be tried.

Selection

For selecting values, there are the following interface options:

Multiple Selection

- Checkboxes

Single Selection

- Radiobuttons
- Buttons
- Text lists (top-to-bottom, left-to-right, or arbitrary arrangement)
- Pick lists (Popup menus)

Pick lists need less screen space, but require more interactions steps (they are less "direct"). In addition, only the current selection is visible, not the whole set of choices.

Multiple Clicking

Possible interface options are:

- Spin Buttons
- Scroll buttons

These options need less space than discrete options, but require users to click several times or even cycle through the possible values or options.

Dragging

For mouse-operated or pen-based systems, dragging is a valid and often efficient option. For finger-operated systems, however, these options should be avoided, if possible. Drag operations are imprecise, relatively slow, and smear the screen surface.

The following controls are available for setting discrete or continuous values.

- Sliders
- Scroll bars

These options are better suited for large and/or continuous value ranges.

Entering Digits or Letters

This option often requires the most effort on the users' side. On the other hand, well-trained users are often very fast using this method. Therefore, while this entry modes is not recommended for public systems, it may the be best option for POS or counter systems.

For more information on keypads, see Numbers and Letters and Interaction.

 top

Source: **Interaction Design Guide for Touchscreen Applications**

Buttons and Menus

Buttons, Targets | Menus | Comparison

Buttons, Targets

As touchscreens are mainly used for point-and-click applications, most touchscreens are filled with **pushbuttons** and areas for data display and entry. Therefore, buttons play an important factor in the design of touchscreens.

Button Size

- minimum of 3/4 by 3/4 inch (2*2 cm) and at least 1/8 inch (3 mm) apart for finger-operated touchscreens.

Targets

- Use large and discrete targets that are spread out on the screen (finger-operated touchscreens).

Button Icons

Use three different icons for indicating the state of buttons:

- **Default icon**
- **"Will be activated" icon**: Signals that the users touches a button as long as the button is being touched (for touchscreens only that initiate an action when the finger is released).
  Highlight the whole button, not just the text, number or graphics on it, because the button is obscured by the touching finger.
- **"Has been activated" icon**: Signals for a short period of time that a touch has been registered and that the corresponding action is initiated.
  Highlight the whole button for better recognition of the feedback.

Uses

Buttons can be used for:

- Offering **choices**, like options, alternatives, etc.
- Initiating **functions**

Function buttons should - apart from standard functions - preferably have text labels (or text labels with graphics).

Menus

Menus consists of a number of elements from which the user has to selects one.

Button Groups

Often menus are realized as groups of buttons. The buttons may look like buttons or may simply be graphical elements (often

supplemented with text labels).

Buttons within groups should be separated clearly so that no erroneous activation can occur.

Group buttons by the methods described for grouping screen elements (e.g. shape, color, background color etc.) to let button groups or menus appear as a unity.

For usage recommendations see Comparison.

Examples

Figure 1: A menu with buttons in different groups; the groups are differentiated by their shapes and styles (from Information Kiosk Systems)

Figure 2: A menu with graphical buttons which are not visualized as buttons (from Information Kiosk Systems)

Lists

Lists or link lists provide a text -based approach to menus.

The text links should be typically arranged from top to bottom. For few links, an arrangement from left to right is acceptable (preferably at the top or bottom of the screen), although it is harder to scan visually. Other arrangements should only be used, if a graphic-like look is wanted, but graphics are not available.

Note: Valid for western script systems only - other scripts system, may lead to different recommendations.

For further usage recommendations see Comparison.

Comparison

	Texts (Lists, Links)	Text-based Buttons	Graphical Buttons

Use it...	• If text labels better identify the objects or actions, • e.g. for abstract objects	• For action buttons (actions are often better described through text labels) • If text is better suited for identifying the objects (e.g. complex alternatives)	• To offer options (e.g. concrete objects); Add text labels for a better understanding of the options • To beautify the screen • To attract users, customers
Use it if...	• Screen space is scarce • The screen is character-based	• Screen space is relatively scarce • The screen is character-based (semi-graphics) or graphical	• Screen space is available • The screen has colors and is graphical
Do not use it...	• If better presentations are available (and sufficient space)		• To offer complex choices, abstract objects, difficult to understand concepts etc.
Recommen-dations	• Arrange from top to bottom • Few options: left-to-right (at top or bottom of screen)	• Arrange function buttons in place or in groupings that are clearly recognized as such	• Arrange function buttons in place or in groupings that are clearly recognized as such

top

Source: **Interaction Design Guide for Touchscreen Applications**

Complex Controls

Lists, Tables | Hierarchies, Stacks, etc.

Lists, Tables

With touchscreens, it is easier to interact with items in lists or tables by **pointing** than by dragging. Therefore, we propose:

- Do not use conventional scrollbars for scrolling in lists or tables, even if they are enlarged. Especially, do not use a "thumb" for scrolling the list by dragging.
  Use pushbuttons for scrolling, instead.
  Provide pushbuttons for the important actions only, not the "whole set".
- Do not use drag & drop for moving items between lists or tables. Use a special "transfer" buttons instead. However, drag & drop may be provided as an alternative interaction method.

Provide a means for selecting items in a list or table, such as

- Enlarged rows or columns for easier selection by pointing (-> see drum)
- A fixed row or column that marks the selected item
  Additional items may be selected by pointing, if multiple selection is used (otherwise the selection changes, and the selected item is moved into the fixed position)

Hierarchies, Stacks, etc.

Do **not** use conventional trees (e.g. as in the Windows Explorer) for presenting hierarchies, even if the trees are enlarged, because tey:

- Are cumbersome to use with fingers
- Consume much space if they are enlarged for use with fingers
- Show too much useless information, e.g. to few data, if they are enlarged for use with fingers.

Choose the presentation according to the task at hand and with considering ease of use and speed, especially with pointing for finger-operated touchscreens. Here are some hints for choosing a presentation.

Tabstrip

A tabstrip may be used to display the branches of a **two-level hierarchy**. The top level nodes (categories) serve as tabs or buttons for changing the view.

Criteria

Contrary to the stack the tabstrip provides easy access to all nodes of the same hierarchy level which makes **comparison** of items, though sequential, easier than in a stack.

Presentation

The view may be a list, drum, set of buttons, etc.

Stack

A stack can display part of a **hierarchy of arbitrary depth**, though only a handful of levels is recommended for touchscreens. A stack displays the path through a hierarchy and the nodes at the lowest level. Clicking a node adds this node to the path and displays the nodes of the next lower level. The path is represented as a set of buttons. Clicking a path button moves the user to the respective level and removes lower path buttons.

Presentation

Comparing items of the same hierarchy level is cumbersome with a stack. A stack is best suited to **searching** for items in a hierarchy.

View

The display of the nodes may be a list, drum, set of buttons, etc.

 top

Source: Interaction Design Guide for Touchscreen Applications

Interaction

General Interaction Issues

- **Use**: Use a simple **point-and-click** interface with large buttons.
- **Do not use**: dragging, double-clicks, scroll bars, dropdown menus, multiple windows, or other elements that can confuse the common user and interfere with usability and efficiency.

Cursor: Turn the cursor off - user will focus on the entire screen instead of the arrow and their thinking and action will be direct instead of indirect.

Textual and Numerical Data Entry with Button Keys

Key arrangement for text entry

Alphabetic keyboards on the screen may have a different order that usual typewriter keyboards. Make the keyboard layout switchable or even customizable. Reason:

- Users may not be proficient typist and are not familiar to the typewriter keyboard layout. A different ordering, e.g. alphabetic, may be better suited for them. Use the standard layout only when people are used to it and find the keys faster with this layout.

Minimize text entry

The conventional keyboard layout is not optimal for 10-finger typing. Alternative layouts have been proposed, but are not widely used. Both, the alternative and the standard layout are used for multiple-finger entry. This is not the case with touchscreens. Here people use just one finger, which slows down text entry considerably. Therefore, keep text entry to a minimum or better even avoid it totally on touchscreens.

Key arrangement for number entry

Conventionally, a 3*3 layout is used for number entry. However, there are two layouts used:

- the telephone layout (1 upper left)
- the keypad layout (1 lower left)

Use the telephone layout for untrained users, because they find the numbers easier with this sorting order, and they may be used to telephones.

Use the keypad layout only for people who are used to it, who prefer it, and who might get into conflicts with the telephone layout. Note that the keypad layout on keyboards is for multiple-finger entry. On touchscreens, however, numbers are entered with one finger only and sequentially.

Minimize number entry

Number entry is very fast on numerical keypads where several fingers of one hand may be used. On touchscreens only one finger may be used which slows down number entry. Therefore do not use touchscreens for heavy use with number entry.

Selections

Selection is an important form of interaction on touchscreens. Users may select:

- Actions (via pushbuttons or menus)
- Options
- Data from a fixed or variable set

Selection on touchscreens is usually done with fingers. Therefore, for selection controls the same size requirements exists as do exist for pushbuttons.

As textual or numerical data input through a virtual keyboard is not well suited to touchscreens, data entry by selection is one important input method for touchscreens.

Selections may also be done indirectly:

- The selection can be moved through the items via pushbuttons, e.g. directional pushbuttons
- The selection may stay fixed; the items are moved to the selection area through pushbuttons, e.g. directional pushbuttons

Scrolling

With touchscreens, it is easier to scroll by **pointing** than by dragging. Therefore, we propose:

- Do not use conventional scroll bars, even if they are enlarged, but to use **simple pushbuttons** instead
- In addition, the scroll buttons may may a **repeat** function. The repeat function should start after a predetermined delay (not too long), and then repeat the scrolling while the screen is touched
- Scroll buttons should follow the usual guidelines for buttons. Better make them larger than too small

Do **not** use scrolling for the **screen** itself, but only for data display, i.e. for fixed areas on the screen. At best, do not use any scrolling at all.

In some cases you can replace scrolling through controls that display only a part of the items or functionality on the screen and that allow for easy switching between the views by pointing to buttons. A tabstrip is such a screen element.

Gestures

Simple gestures, that are easy to remember, can be used on stylus-operated touchscreens for often-used functions. Gestures are simple "drawings" like letters or symbols. Here are a few examples:

- Deleting items by striking them through or crossing them out (the Apple Newton used a "W" for this operation.
- Marking items by adding a cross ("X") or "Ã".
- Identifying a user by his or her signature (handwriting).

Gestures are not well suited to finger-operated touchscreens, as the use drag operations with fingers are not recommended in the literature.

Feedback

Buttons should give some sort of feedback as to their state (on/off, active/inactive, activated (transient), etc.).

- **Immediate feedback**: Critical to reassure the user that their touch registered.
- **Visual feedback** (highlighting, 3 D-effect): In response to button presses.
  Auditory feedback (click, ...): Is sometimes also appropriate.
- **Screen changes**: Clear the display clears immediately and displays an hourglass while loading the next screen.
- **Digitized speech**: Can walk users through your application.

Note: Nowadays, feedback is often given, while the mouse pointer is over the pushbutton. However, this behavior cannot be implemented on touchscreens!

Error Robustness

Confirmation after selection: Use it when the consequences are destructive or difficult to undo, minimize inadvertent button selections.

top

Source: **Interaction Design Guide for Touchscreen Applications**

General Recommendations

Metaphors

Metaphors

Metaphors provide users with a familiar environment on the screen. This allows users to transfer prior real-world knowlegde to an application: they learn new applications faster and relearn rarely used applications easier.

So, metaphors might just be THE solution for touchscreens where users are often untrained. However, there are also some caveats:

- Metaphors make break and thus confuse users, because they require interaction that does not make sense to the users, or they make the application behave unexpectedly.
- Often applications are abstract, and there is no suitable real-world metaphor available.
- Metaphors may waste precious screen space, because the graphical elements that maintain the metaphor (e.g. a room) need room to be realistic enough.
- The graphics for the metaphor may be too detailed to allow fast orientation and operation.


top

Source: **Interaction Design Guide for Touchscreen Applications**

Special Touchscreen Applications

Kiosk Applications | Museum Guides, Public Information Systems

Kiosk Applications

- Use a simple point-and-click interface
- Animation and large fonts help attract users to kiosk applications. The kiosk design (cabinet) should also be attractive

Example

Figure 1: Kiosk System (from Information Kiosk Systems)

Museum Guides, Public Information Systems

- Use a simple point-and-click interface
- Images and photos can be used for illustration (esp. in museums)

top

Selected Links

This page presents selected links to touch screen manufacturers and sites about touchscreens. Mostly companies and sites, which provide more than just technical information, are listed!

- ADMetro (www.admetro.com)
  (Canadian distributor)
- ARM Incorporated (www.armnet.com)
- CUE (www.cue.cz)
  (Touch panel)
- Data Asia Technology Ltd. (www.touchscreen.com)
- ELO TouchSystems (www.elotouch.com)
- InfoGenesis (www.infogenesis.com)
- Information Kiosk Systems (www.infokiosks.com)
  (presents screen samples using graphics)
- John Posey Corporation (pos-ey.com)
- Mass Multimedia Company (www.touchscreens.com)
  (contains useful information)
- Micro Touch (www.microtouch.com)
  (contains lots of information)
- NCR (www.ncr.com)
  (includes pages about usability)
- POSnet (pos-net.com/)
  (Website for point of sale resources maintained by the John Posey Corporation; links to systems, software and hardware for POS)
- r/Power (rpower.com)
- Touch Screen Systems (www.link.net.mt/touch.html)
  (not up-to-date)

top

Source: **Interaction Design Guide for Touchscreen Applications**

References

- Balzert, H. et al. (Eds.), (1988). Einführung in die Software-Ergonomie. Berlin: de Gruyter.
- Brown, C. M. (1988). Human-computer interface design guidelines. Norwood, NJ: Able (from Mayhew, 1992).
- ELO Touchscreen Systems (from the web site): 10 Tips for Touchscreen Applications.
- Mayhew, D. J. (1992). Principles and guidelines in software user interface design. Englewood Cliffs: Prentice Hall.
- Shneiderman, B. (1987). Designing the user interface: Strategies for effective human-computer interaction. Reading, MA: Addison-Wesley.
- Wandmacher, J. (1993). Software-Ergonomie. Berlin: de Gruyter.


top

Source: **Interaction Design Guide for Touchscreen Applications**

Introduction

The following collection of controls for finger-operated touchscreens lists existing controls as well as proposals for controls.

These examples present a certain functionality, they are not meant as proposals for the graphical design of such controls. The graphical design of touchscreen controls depends heavily on the context of use, branding strategy, screen technology, etc.

Note: Some of the controls may be used for different purposes. Therefore this list contains some redundancy.

These pages include guidelines for designing and using touchscreen controls from this guide where available.


top

Source: Interaction Design Guide for Touchscreen Applications

Pushbuttons

Guidelines | Button Examples

Generally, there are myriads of buttons with text, with graphics or with both of them. You find good examples in multimedia applications, in some "fancy" applications (like those from Kai Krause...), and on the Web.Therefore, I just pick some typical examples. Most of them are more or less usable on touchscreens, provided they are large enough (for fingers).

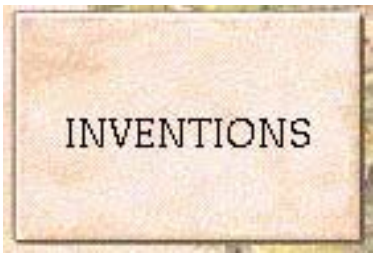
Guidelines

- **Size**: At least 3/4 (ca. 2 cm) inch wide and high and 1/8 inch (ca. 3 mm) apart.
- **Grid**: We propose to divide the touchscreen, which usually has a fixed and relatively small size, into a grid that defines the basic "cell size" for buttons and other screen elements.
  Example: For a 640*480 screen, a grid cell of 64*64 pixels seems adequate. This provides a 10*7 grid with room for a status line at the bottom.
- **Grouping**: Group pushbuttons by using frames, form, color, background color or texture; use attributes redundantly for easier identification of groups.
- **Feedback**: Buttons should give some sort of feedback as to their state (on/off, active/inactive, activated (transient), etc.).
  Note: Nowadays, feedback is often given, while the mouse pointer is over the pushbutton. However, this behavior cannot be implemented on touchscreens!
- **Button Icons**: Use three different icons for indicating the state of buttons
  - **Default icon**
  - **"Will be activated" icon**: Signals that the users touches a button as long as the button is being touched (for touchscreens only that initiate an action when the finger is released).
    Highlight the whole button, not just the text, number or graphics on it, because the button is obscured by the touching finger.
  - **"Has been activated" icon**: Signals for a short period of time that a touch has been registered and that the corresponding action is initiated.
    Highlight the whole button for better recognition of the feedback.

Button Examples

Toggle button, on-state (from Shneiderman, but also found elsewhere). Text labels, graphics, as well as colors identify the on-state.

Toggle button, off-state (from Shneiderman, but also found elsewhere). Text labels, graphics, as well as colors identify the off-state.



A more "stylish" button with text only (from the CD "Leonardo the Inventor").

A button with graphics and an accompanying text below the graphics (from the CD "Leonardo the Inventor").

A button with a photo as graphics on it (from Biolek's CD "Meine Rezepte"). Look also for the shading as additional optical effect!

Buttons with status display (middle button: selected, bottom button: not selected) (from Soap 1.0)

Simple Buttons for navigation functions (64*64 grid = 54*54 button with 5 pixel border)

Simple buttons for numerical keypads (64*64 grid = 54*54 button with 5 pixel border)

top

Source: **Interaction Design Guide for Touchscreen Applications**

Scrolling and Navigation

Guidelines | Scrollbar | Scroll Buttons

Guidelines

With touchscreens, it is easier to scroll by **pointing** than by dragging.

- Therefore, we propose, not to use conventional scrollbars, even if they are enlarged, but to use **simple pushbuttons** instead.
- In addition, the scroll buttons may may a **repeat** function. The repeat function should start after a predetermined delay (not too long), and then repeat the scrolling while the screen is touched.
- Scroll buttons should follow the usual guidelines for buttons. Better make them larger than too small.

Scrollbar

Scrollbar from the "Leonardo the Inventor" CD.

Requires drag operation, therefore not optimal for finger-operated touchscreens!

...

Scroll Buttons

Simple Buttons for Vertical Scrolling/Movement

Scroll button: Scroll/move to first item

Scroll button: Scroll/move back faster / more lines / a page

Scroll button: Scroll/move back slower / one line

Scroll button: Scroll/move forward slower / one line

Scroll button: Scroll/move forward faster / more lines / a page

Scroll button: Scroll/move to last item

top

Source: **Interaction Design Guide for Touchscreen Applications**

Data Entry I: Clicking

Guidelines

- **Key arrangement for text entry:** Alphabetic keyboards on the screen may have a different order that usual typewriter keyboards. Make the keyboard layout switchable or even customizable. Reason:
  - Users may not be proficient typist and are not familiar to the typewriter keyboard layout. A different ordering, e.g. alphabetic, may be better suited for them. Use the standard layout only when people are used to it and find the keys faster with this layout.
- **Minimize text entry**: The conventional keyboard layout is not optimal for 10-finger typing. Alternative layouts have been proposed, but are not widely used. Both, the alternative and the standard layout are used for multiple-finger entry. This is not the case with touchscreens. Here people use just one finger, which slows down text entry considerably. Therefore, keep text entry to a minimum or better even avoid it totally on touchscreens.
- **Key arrangement for number entry**: Conventionally, a 3*3 layout is used for number entry. However, there are two layouts used, the:
  - Telephone layout (digit 1 located upper left)
  - Keypad layout (digit 1 located lower left)

  Use the telephone layout for untrained users, because they find the numbers easier with this sorting order, and they may be used to telephones.
  Use the keypad layout only for people who are used to it, who prefer it, and who might get into conflicts with the telephone layout. Note that the keypad layout on keyboards is for multiple-finger entry. On touchscreens, however, numbers are entered with one finger only and sequentially.
- **Minimize number entry**: Number entry is very fast on numerical keypads where several fingers of one hand may be used. On touchscreens only one finger may be used which slows down number entry. Therefore do not use touchscreens for heavy use with number entry.

Numerical Keypads

Telephone Layout

A simple numerical keypad with telephone layout.

Grouping is provided by:

- Shape
- Foreground color
- Background color

Function keys are separated by shape and background color. The central number is highlighted by background color to act as target of the finger movement.

Keypad Layout

The same keypad with keypad layout. This layout is better suited to users that are comfortable with keyboards (typewriters, computers).

Ideally, an application should allow users to choose their keypad preferred layout.

Date Entry

A simple control for date entry. It uses a variant of **spin buttons** for incrementing or decrementing values. It should be supplemented by a text field which displays the date (for feedback).

Through simple clicking the day, week, month, and year can be incremented or decremented.

Categories that are not needed (year, ...) can be omitted.

Alpha(numerical) Keypads

Linear Layout (Three Rows)

This keypad for letter entry uses an alphabetical ordering instead of the typical typewriter.

A linear layout is better suited to novices and people who are not used to type-writing.

Tight Layout (6 Rows)

This keypad for letter inout also uses an alphabetical ordering. However, the keys form the shape of nearly a circle.

Scanning for letters may be harder with this layout.

Selecting Single Values

A control for setting discrete values (from Kai Krause's "Convolver 1.0"). The curve maybe well suited to do the adjustment with a finger.

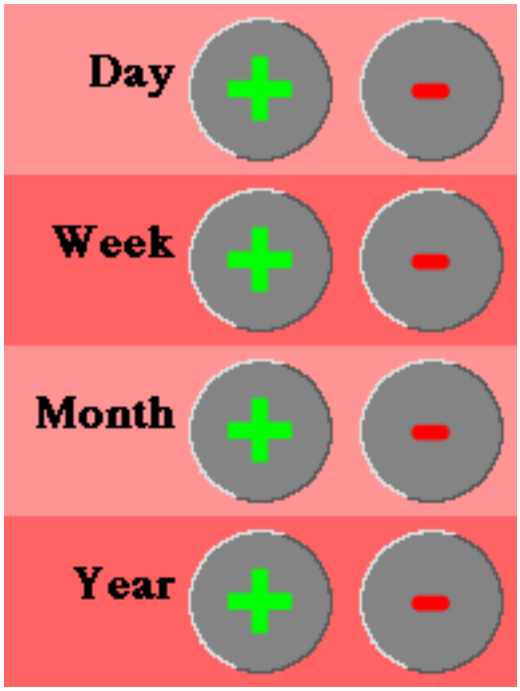
Uses a lot of screen space!

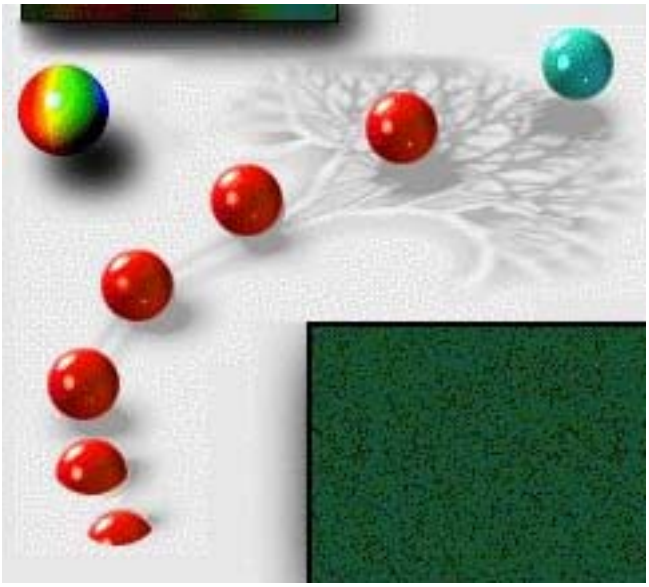
top

Source: **Interaction Design Guide for Touchscreen Applications**


1
2
3
4
5
6
7
8
9
C
0


7
8
9
4
5
6
1
2
3
C
0


Day
Week
Month
Year



Data Entry II: Dragging

The following examples of controls for setting - preferrably continuous or quasi-continuous - values are taken from applications, which do not use touchscreens. But these controls could be used there, if certain restrictions are observed. Note, however, that the controls use drag operations, which are not well suited to finger-operated touchscreens (speed, accuracy).

Examples

Scrollbar from the "Leonardo the Inventor" CD, may be used for setting continuous values.

Too fancy for "everyday use"!

A control for setting continuous values (from Kai Krause's "GOO 1.0").

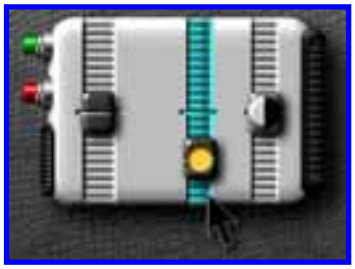
Uses a lot of screen space.

Slider for setting the size in pixels of an image (proportional change only) (from Kai Krause's "Soap 1.0")

Slider for setting the enlargement of an image (proportional change only) (from Kai Krause's "Soap 1.0").

The enlarged image is displayed in a frame left to the control. The image at the top is a preview of the whole image. The empty spaces can be used for storing views of the image.

A button with sliders (in a drawer) for setting text color attributes (from Kai Krause's "Soap 1.0").



Control for adjusting lightness and contrast of an image (from Kai Krause's "Soap 1.0").

The leftmost slider sets the strength of the effects. The bulbs on the upper left act as buttons for undoing an effect or making it permanent. There are drawers for additional functions (see below).

The same control with open drawers (from Kai Krause's "Soap 1.0").

Left drawer: Here you can store settings. Right drawer: Advanced lightness controls.

A similar control for setting color parameters of an image (from Kai Krause's "Soap 1.0").

A control for setting several continuous-valued attibutes of a brush; the marbles in the drawer to the left store different settings (from Kai Krause's Soap 1.0).

top

Source: **Interaction Design Guide for Touchscreen Applications**


RAUS


768 x 512


Farbe


Größe
Weichheit
Deckkraft
Dichte

Selection, Menus

Guidelines | Static Menus | Dynamic Menus

Guidelines

Selection on touchscreens is usually done with fingers. Therefore, for selection controls the same size requirements exists as do exist for pushbuttons.

Selections may also be done indirectly. The Selection:

- Can be moved through the items via pushbuttons, e.g. directional pushbuttons
- May stay fixed; the items are moved to the selection area through pushbuttons, e.g. directional pushbuttons

Static Menus

Keyboard menu: An "alphabet menu" for selecting the beginning letter of a word or even for entering text (Lexikon der Musik).

A little too fancy for "everyday use" (and too small for fingers).

Single selection only!

Static menu (button bar): A button bar for selecting "rooms" where different processing is applied to images (Soap 1.0).

The button bar appears, when the cursor is moved over the button "Plan".

Static button menu for a menu page using buttons which are easily recognizable as buttons. The primary button menu offers choices for the users.

There is a second button menu providing functions; not usable functions are hidden (or grayed-out).

Static button menu for a menu page using **graphical buttons**.

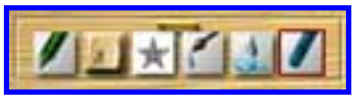
The graphical buttons are not recognizable as buttons. In addition text labels are added to make the options clearer.

Static button menu for a menu page using **text-based buttons** (graphic text). The text menu is optically enhanced by a flower background graphic.

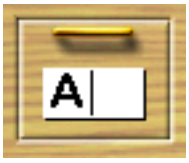
Dynamic Menus

Popup menu: A "remote control" that acts as a sort of "menu bar" (Soap 1.0)

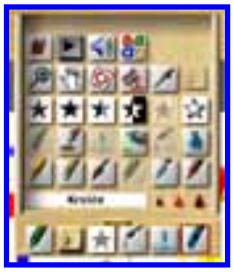
Drawer (1): The drawer acts like a pulldown or popup menu. It provides a number of choices that are displayed on user request. In this case, the drawer does not give any hint on its items. However, the drawer may have a textual or graphical label classifying the items, or it may display the selected item.



Drawer (2): The drawer principle may be extended by displaying a small set of items (the favorites) on the drawer. The selected item(s) is/are highlighted.



Drawer (3): The drawer may also be used as an entry field that allows to enter data that are not provided within the drawer. This corresponds to a dropdown combination box.

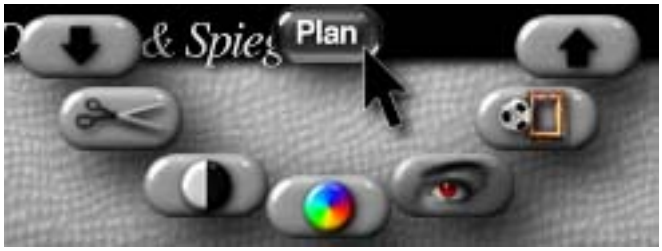


Drawer (2a): The drawer opened. It provides a set of tools or options where the user may select from. The drawer is closed, after the user clicks the handle again.

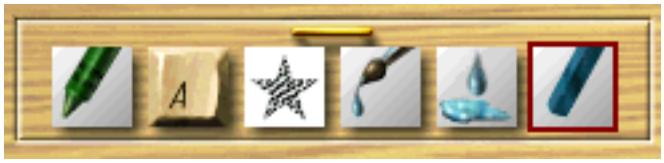
Drawer (2b): The same drawer, after the user clicked the sketch block in the above example. Now tools for handling images and files are displayed.

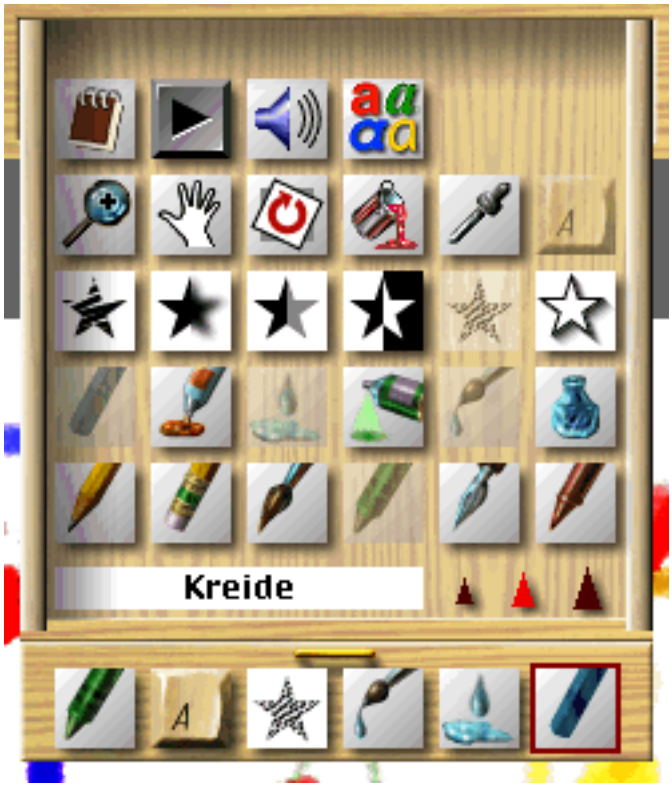
top

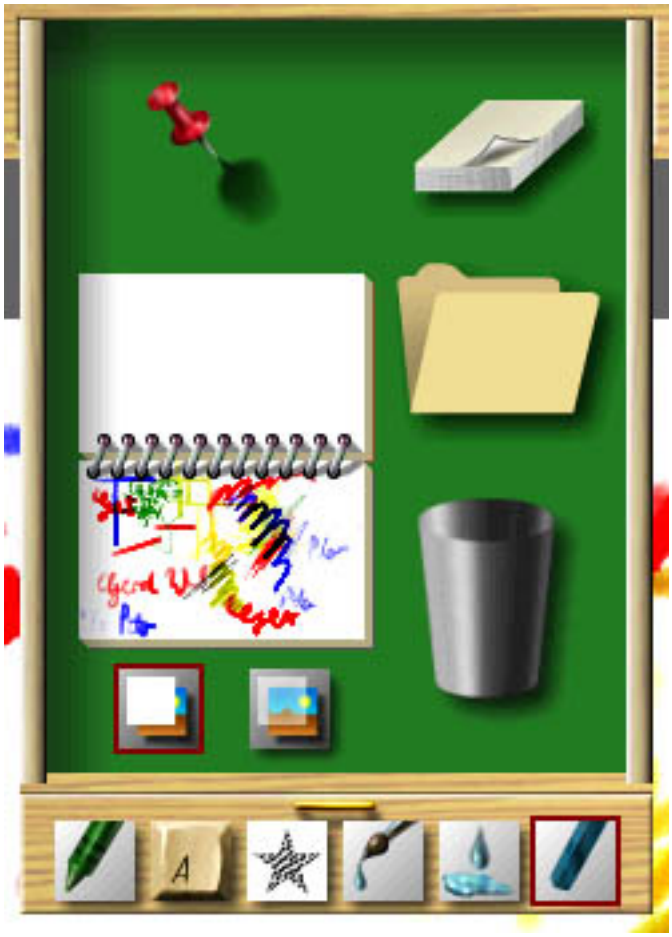
Source: **Interaction Design Guide for Touchscreen Applications**


Plan


Hilfe
Datei
Drucken...
Speichern...
Letzte Version
Ansicht
Cursor
Werkzeuge
Optionen




Kreide



Lists and Tables

Guidelines | Controls for List and Tables

Guidelines

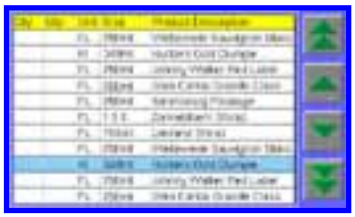
With touchscreens, it is easier to interact with items in lists by **pointing** than by dragging. Therefore, we propose:

- Do not use conventional scrollbars for scrolling in lists or tables, even if they are enlarged. Especially, do not use a "thumb" for scrolling the list by dragging.
  Use pushbuttons for scrolling, instead.
  Provide pushbuttons for the important actions only, not the "whole set".
- Do not use drag & drop for moving items between lists or tables. Use a special "transfer" buttons instead. However, drag & drop may be provided as an alternative interaction method.

Provide a means for selecting items in a list or table, such as

- enlarged rows or columns for easier selection by pointing (-> see drum)
- a fixed row or column that marks the selected item. Additional items may be selected by pointing, if multiple selection is used (otherwise the selection changes, and the selected item is moved into the fixed position)

Controls for List and Tables

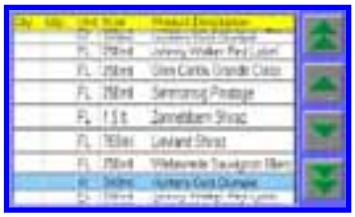


List

Conventional list or table with pushbuttons for scrolling and the following features:

- Four basic pushbuttons are used for the scroll functions:
  scroll one page backwards, scroll one line backwards, scroll one line forward, scroll one page forward.
- The scrolling should be overlapping (different from R/3).
- If a scroll button is pressed for a longer period of time, the button should repeat for easier scrolling.
- One row or column may have a "permanent" selection.
- Use lists for **data display** without manual selection of items through pointing.

Drum

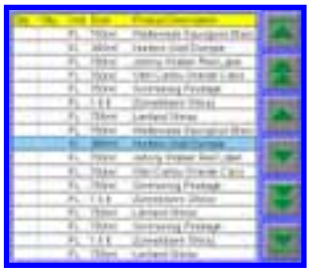


Variant of the list. Drums are better suited to selection of items through fingers:

- The list is **distorted** to enlarge the central row (or column) for easier selection of items by finger.
- The distortion also allows to display the context of the central list element.
- The items in a drum are arranged in a **cyclic** fashion, like in a real drum.
- One row or column may have a **permanent selection**. Users may scroll items "into" this selection row or column in order to select them.

Note: The selected item should not be the central item, because it needs not to be pointed at.

Long list



Features additional buttons for accessing the first and/or last list element.

Long drum

Features additional buttons for accessing the first and/or last list element.

Double list

Consists of two connected lists and allows to move items between the two lists (similar to "mover" dialogues).

- Moved items are deleted in the source list.
- The two lists have one common set of scroll buttons. One of the lists is active; the scroll buttons act on this list.
- A "transfer" buttons between the two lists is used to move selected items from one list to the other (dragging may be provided as an additional interaction method).

Double drum

Consists of two connected drums and allows to move items between the drums (similar to "mover" dialogues).

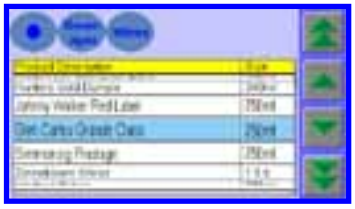
Otherwise similar to double list, but better suited to manual selection of items through fingers.

Stack List

A stack list is one way to display a part of a hierarchy:

- The list displays the items at the current bottom node.
- The buttons at the top display the path in the hierarchy.

Buttons are added to the top, as the users moves deeper into the hierarchy by pointing at items in the list. Buttons are also used to quickly step up the hierarchy, which removes the buttons.



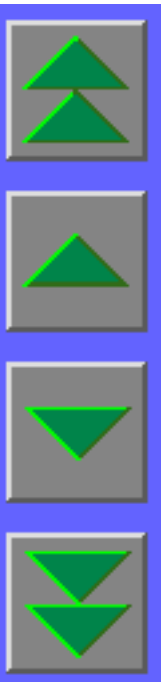
Stack Drum

Like stack list, but with drum. Better suited to manual selection of items through fingers.

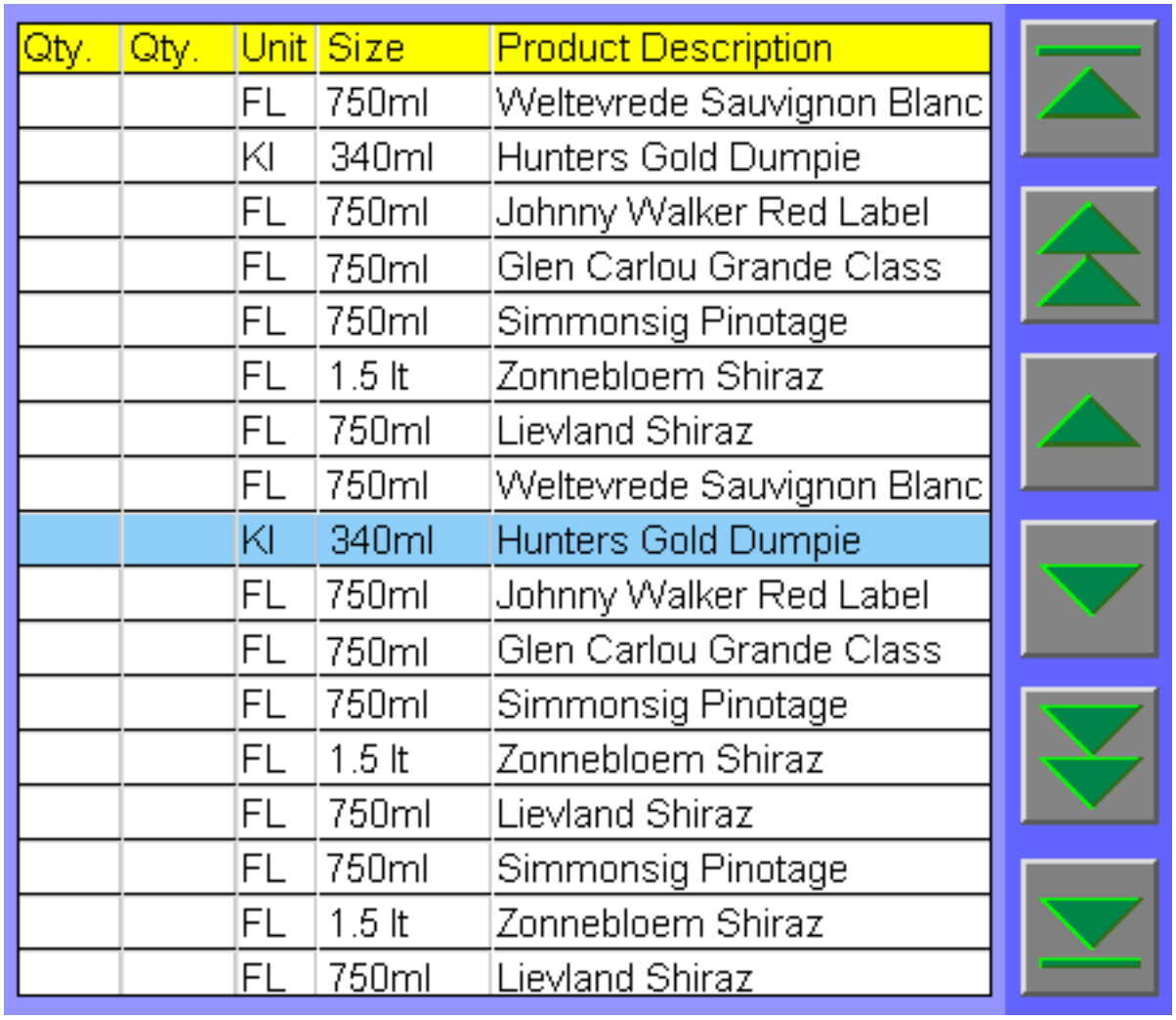
top

Source: **Interaction Design Guide for Touchscreen Applications**

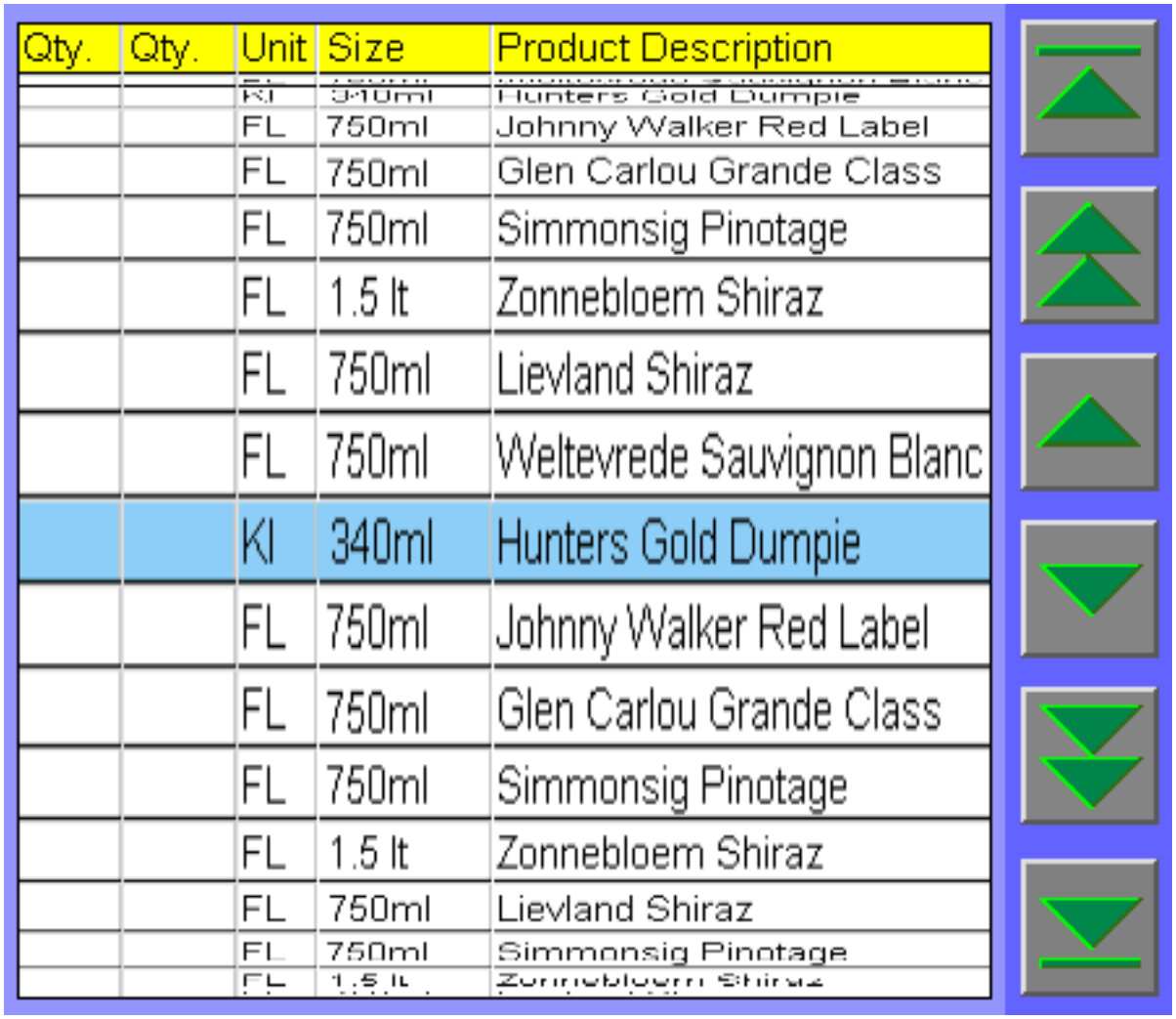
Qty.	Qty.	Unit	Size	Product Description
		FL	750ml	Weltevrede Sauvignon Blanc
		Kl	340ml	Hunters Gold Dumpie
		FL	750ml	Johnny Walker Red Label
		FL	750ml	Glen Carlou Grande Class
		FL	750ml	Simmonsig Pinotage
		FL	1.5 lt	Zonnebloem Shiraz
		FL	750ml	Lievland Shiraz
		FL	750ml	Weltevrede Sauvignon Blanc
		Kl	340ml	Hunters Gold Dumpie
		FL	750ml	Johnny Walker Red Label
		FL	750ml	Glen Carlou Grande Class

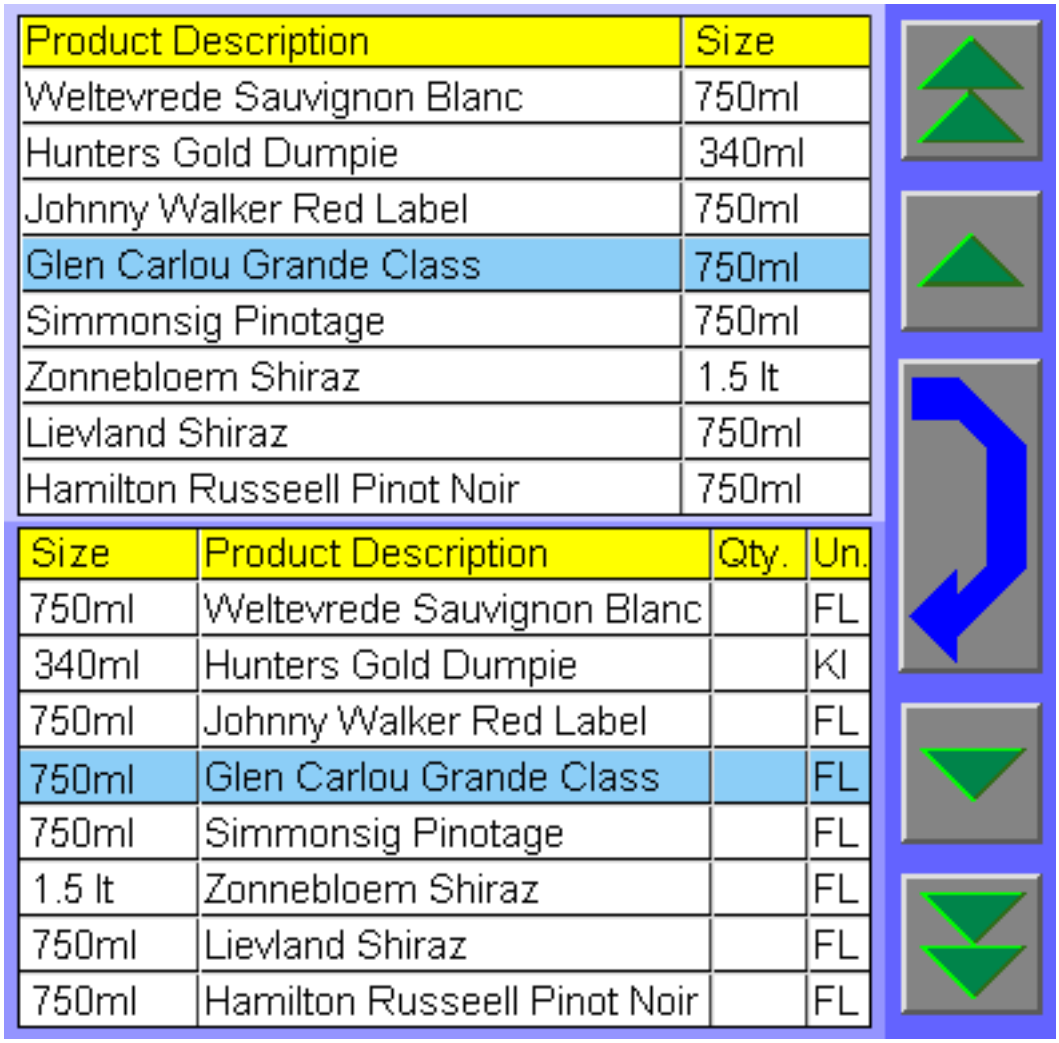



Qty.
Qty.
Unit
Size
Product Description
KI
340ml
Hunters Gold Dumpie
FL
750ml
Johnny Walker Red Label
FL
750ml
Glen Carlou Grande Class
FL
750ml
Simmonsig Pinotage
FL
1.5 lt
Zonnebloem Shiraz
FL
750ml
Lievland Shiraz
FL
750ml
Weltevrede Sauvignon Blanc
KI
340ml
Hunters Gold Dumpie
FL
750ml
Johnny Walker Red Label



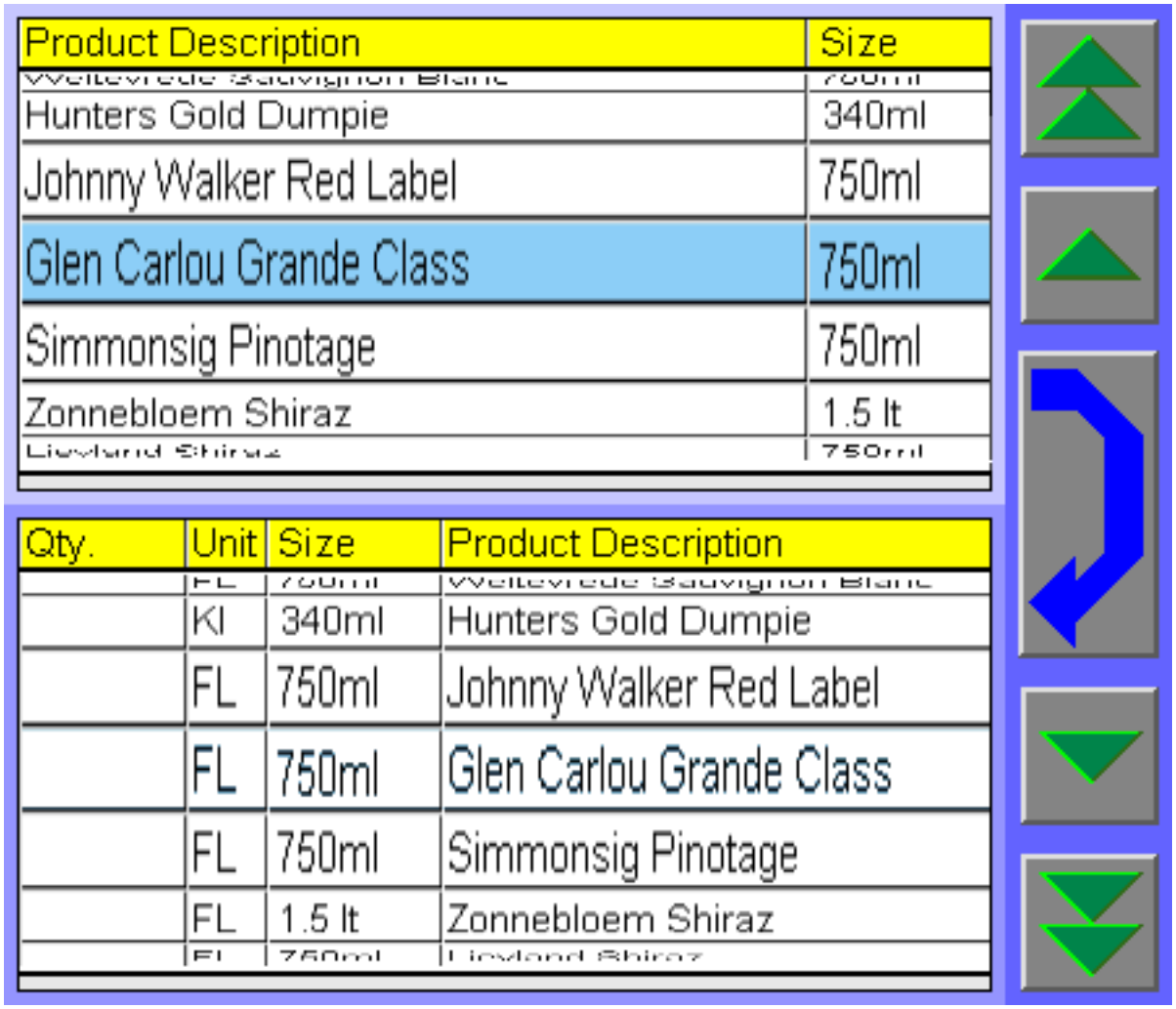
Qty.	Qty.	Unit	Size	Product Description
		FL	750ml	Weltevrede Sauvignon Blanc
		Kl	340ml	Hunters Gold Dumpie
		FL	750ml	Johnny Walker Red Label
		FL	750ml	Glen Carlou Grande Class
		FL	750ml	Simmonsig Pinotage
		FL	1.5 lt	Zonnebloem Shiraz
		FL	750ml	Lievland Shiraz
		FL	750ml	Weltevrede Sauvignon Blanc
		Kl	340ml	Hunters Gold Dumpie
		FL	750ml	Johnny Walker Red Label
		FL	750ml	Glen Carlou Grande Class
		FL	750ml	Simmonsig Pinotage
		FL	1.5 lt	Zonnebloem Shiraz
		FL	750ml	Lievland Shiraz
		FL	750ml	Simmonsig Pinotage
		FL	1.5 lt	Zonnebloem Shiraz
		FL	750ml	Lievland Shiraz


Qty.
Qty.
Unit
Size
Product Description
KI 340ml Hunters Gold Dumpie
FL 750ml Johnny Walker Red Label
FL 750ml Glen Carlou Grande Class
FL 750ml Simmonsig Pinotage
FL 1.5 lt Zonnebloem Shiraz
FL 750ml Lievland Shiraz
FL 750ml Weltevrede Sauvignon Blanc
KI 340ml Hunters Gold Dumpie
FL 750ml Johnny Walker Red Label
FL 750ml Glen Carlou Grande Class
FL 750ml Simmonsig Pinotage
FL 1.5 lt Zonnebloem Shiraz
FL 750ml Lievland Shiraz
FL 750ml Simmonsig Pinotage
FL 1.5 lt Zonnebloem Shiraz

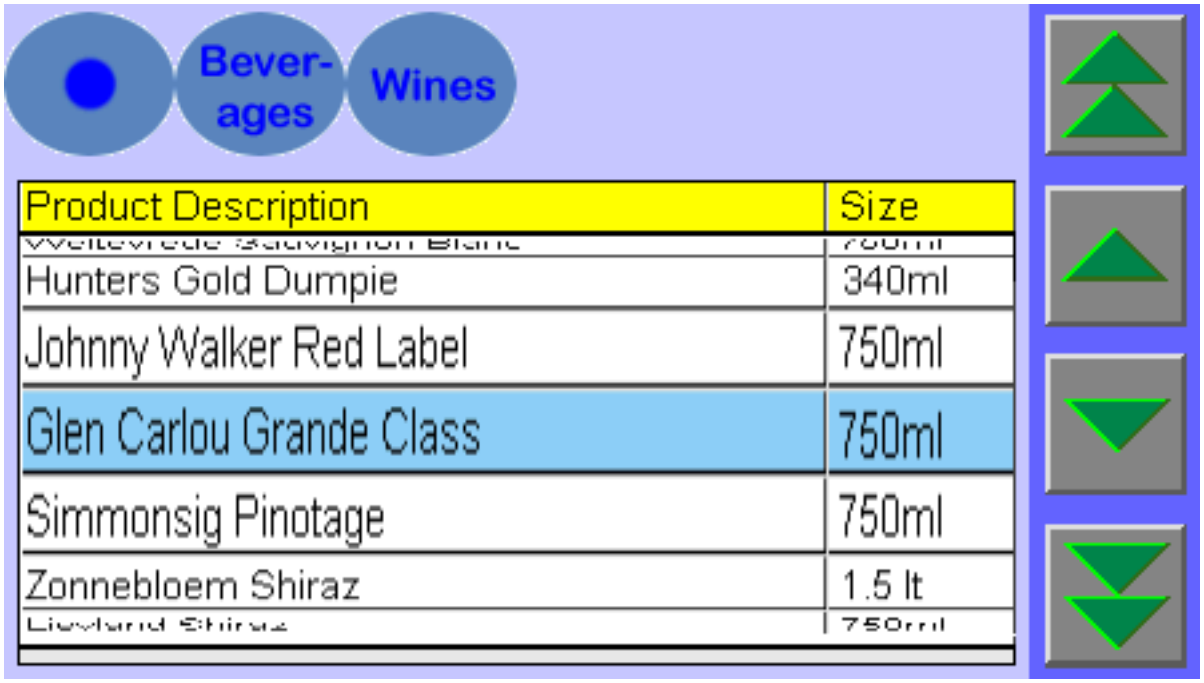


Product Description	Size
Weltevrede Sauvignon Blanc	750ml
Hunters Gold Dumpie	340ml
Johnny Walker Red Label	750ml
Glen Carlou Grande Class	750ml
Simmonsig Pinotage	750ml
Zonnebloem Shiraz	1.5 lt
Lievland Shiraz	750ml
Hamilton Russeell Pinot Noir	750ml

Size	Product Description	Qty.	Un.
750ml	Weltevrede Sauvignon Blanc		FL
340ml	Hunters Gold Dumpie		Kl
750ml	Johnny Walker Red Label		FL
750ml	Glen Carlou Grande Class		FL
750ml	Simmonsig Pinotage		FL
1.5 lt	Zonnebloem Shiraz		FL
750ml	Lievland Shiraz		FL
750ml	Hamilton Russeell Pinot Noir		FL


Product Description
Size
Hunters Gold Dumpie
340ml
Johnny Walker Red Label
750ml
Glen Carlou Grande Class
750ml
Simmonsig Pinotage
750ml
Zonnebloem Shiraz
1.5 lt
Qty.
Unit
Size
Product Description
KI
340ml
Hunters Gold Dumpie
FL
750ml
Johnny Walker Red Label
FL
750ml
Glen Carlou Grande Class
FL
750ml
Simmonsig Pinotage
FL
1.5 lt
Zonnebloem Shiraz


Bever-
ages
Wines
Qty.
Qty.
Unit
Size
Product Description
FL
750ml
Weltevrede Sauvignon Blanc
Kl
340ml
Hunters Gold Dumpie
FL
750ml
Johnny Walker Red Label
FL
750ml
Glen Carlou Grande Class
FL
750ml
Simmonsig Pinotage
FL
1.5 lt
Zonnebloem Shiraz
FL
750ml
Lievland Shiraz


Bever-
ages
Wines
Product Description
Size
Hunters Gold Dumpie
340ml
Johnny Walker Red Label
750ml
Glen Carlou Grande Class
750ml
Simmonsig Pinotage
750ml
Zonnebloem Shiraz
1.5 lt

Hierarchies

Guidelines | Tabstrip and Stack

Guidelines

Do **not** use conventional trees (e.g. as in the Windows Explorer) for presenting hierarchies, even if the trees are enlarged, because

- they are cumbersome to use with fingers
- they consume much space if they are enlarged for use with fingers
- they show too much useless information, e.g. to few data, if they are enlarged for use with fingers.

Choose the presentation according to the task at hand and with considering ease of use and speed, especially with pointing for finger-operated touchscreens. Here are some hints for choosing a presentation:

- **Tabstrip**: A tabstrip may be used to display the branches of a **two-level hierarchy**. The top level nodes (categories) serve as tabs or buttons for changing the view.
  Criteria: Contrary to the stack the tabstrip provides easy access to all nodes of the same hierarchy level which makes **comparison** of items, though sequential, easier than in a stack.
  The view may be a list, drum, set of buttons, etc.
- **Stack**: A stack can display part of a **hierarchy of arbitrary depth**, though only a handful of levels is recommended for touchscreens. A stack displays the path through a hierarchy and the nodes at the lowest level. Clicking a node adds this node to the path and displays the nodes of the next lower level. The path is represented as a set of buttons. Clicking a path button moves the user to the respective level and removes lower path buttons.
  Criteria: Comparing items of the same hierarchy level is cumbersome with a stack. A stack is best suited to **searching** for items in a hierarchy.
  The display of the nodes may be a list, drum, set of buttons, etc.

Tabstrip and Stack

Tabstrip with List

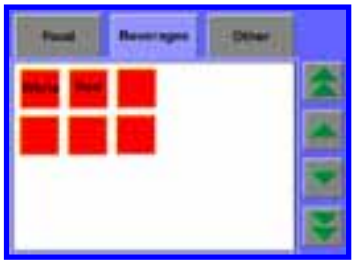
A tabstrip serves to display a two-level hierarchy:

- The tabs at the top (or elsewhere) display the categories.
- The list displays the items of the current category.

Tabs at the top display the categories. The user clicks the tabs to change the view (category).

Tabstrip with Drum

Like tabstrip with list, but with drum. Better suited to manual selection of items with finger.



Tabstrip with Button Group

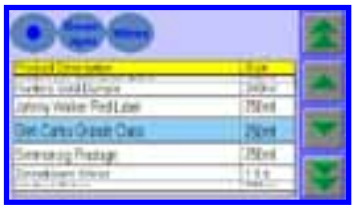
Like tabstrip with list, but with drum. Better suited to manual selection of items with finger.

Stack with List

A stack list is one way to display a part of a hierarchy of arbitrary depth:

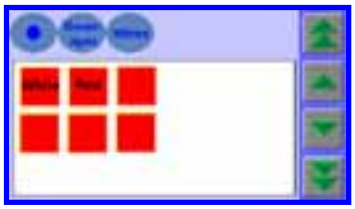
- The list displays the items at the current bottom node.
- The buttons at the top display the path in the hierarchy.

Buttons are added to the top, as the users moves deeper into the hierarchy by pointing at items in the list. Buttons are also used to quickly step up the hierarchy, which removes the buttons.



Stack with Drum

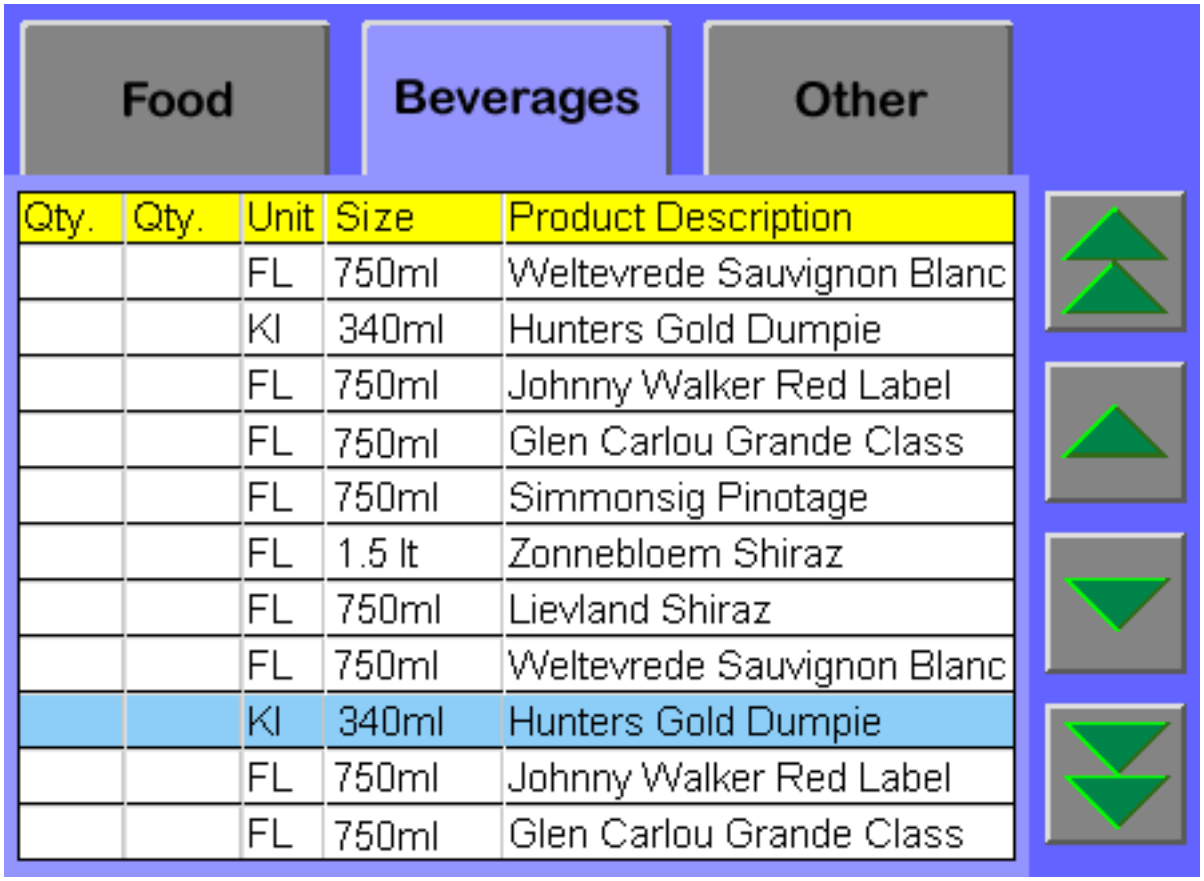
Like stack list, but with drum. Better suited to manual selection of items with finger.

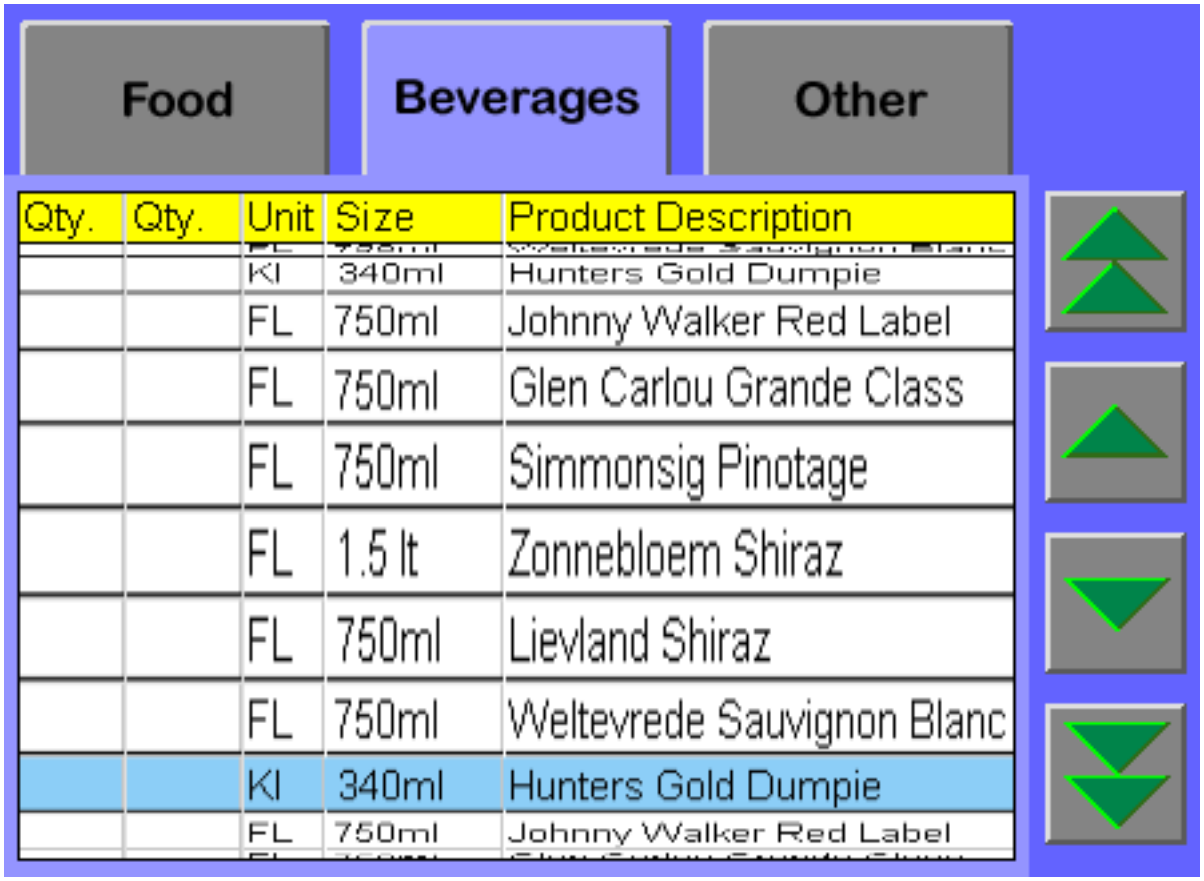


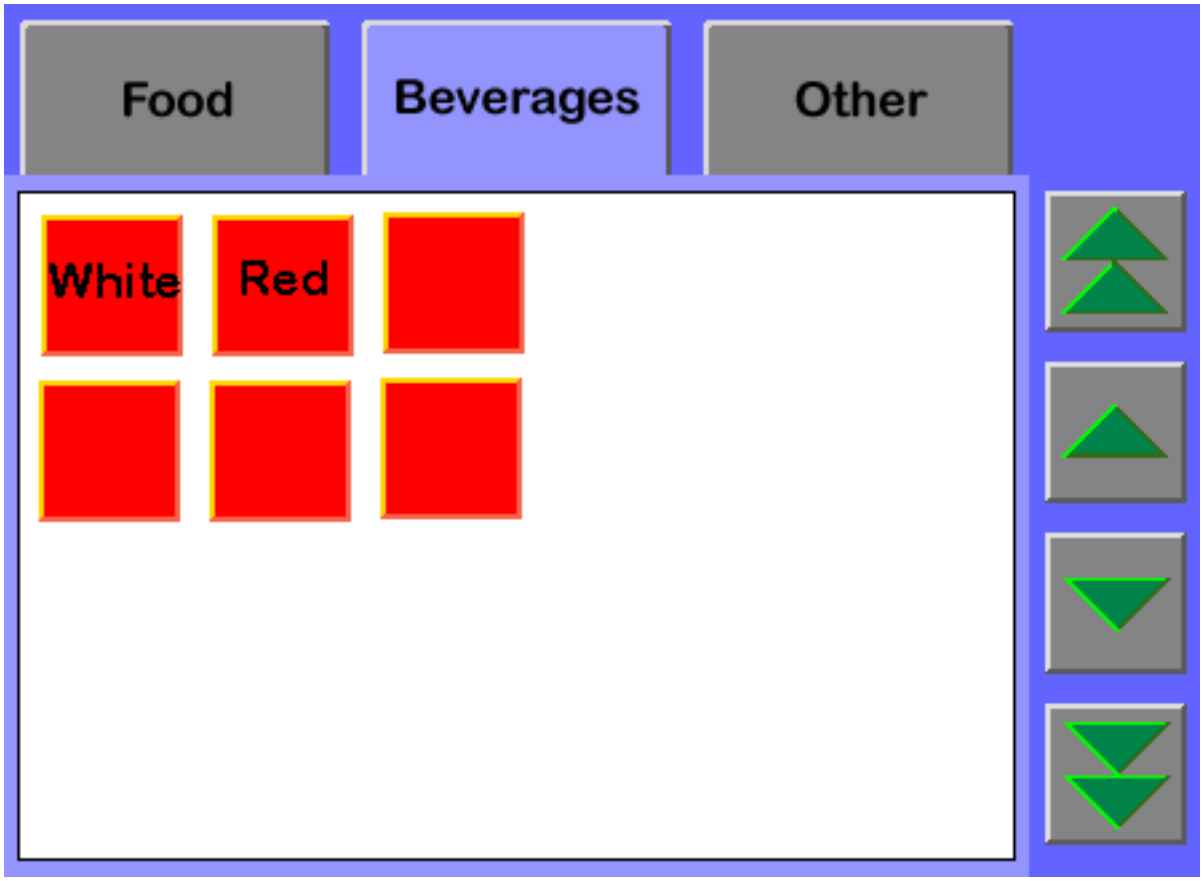
Stack with Button Group

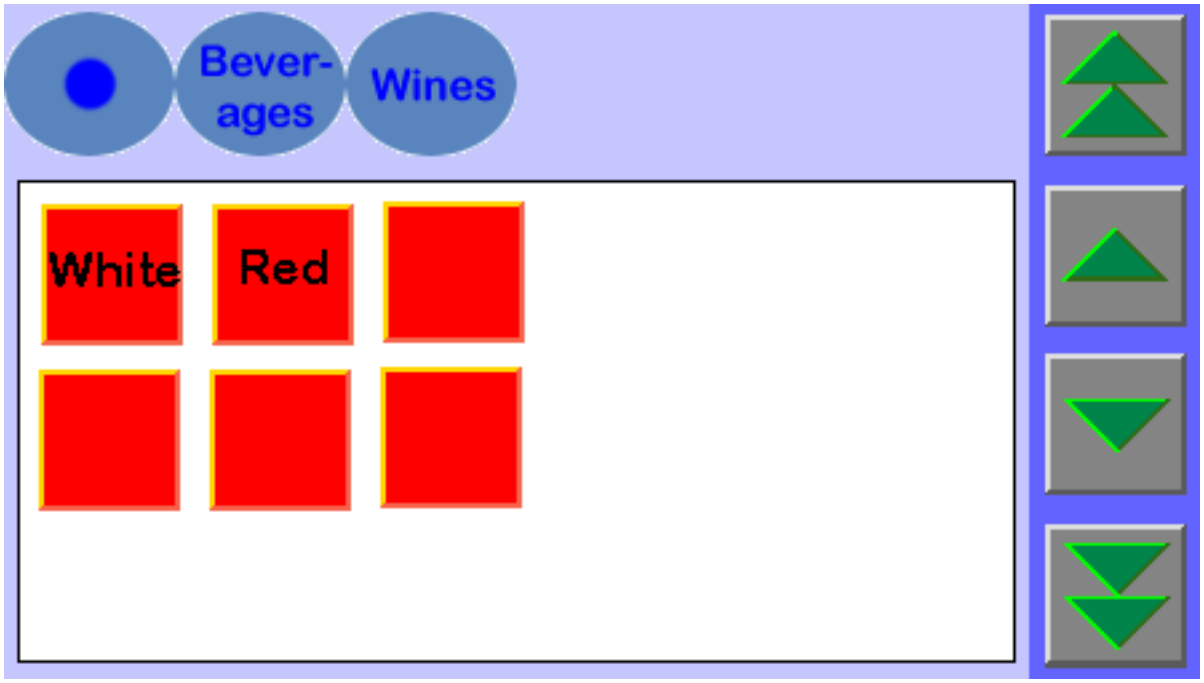
Like stack list, but with button group. Better suited to manual selection of items with finger.

top


Food
Beverages
Other
Qty.
Qty.
Unit
Size
Product Description
FL 750ml Weltevrede Sauvignon Blanc
Kl 340ml Hunters Gold Dumpie
FL 750ml Johnny Walker Red Label
FL 750ml Glen Carlou Grande Class
FL 750ml Simmonsig Pinotage
FL 1.5 lt Zonnebloem Shiraz
FL 750ml Lievland Shiraz
FL 750ml Weltevrede Sauvignon Blanc
Kl 340ml Hunters Gold Dumpie
FL 750ml Johnny Walker Red Label
FL 750ml Glen Carlou Grande Class


Food
Beverages
Other
Qty.
Qty.
Unit
Size
Product Description
KI 340ml Hunters Gold Dumpie
FL 750ml Johnny Walker Red Label
FL 750ml Glen Carlou Grande Class
FL 750ml Simmonsig Pinotage
FL 1.5 lt Zonnebloem Shiraz
FL 750ml Lievland Shiraz
FL 750ml Weltevrede Sauvignon Blanc
KI 340ml Hunters Gold Dumpie
FL 750ml Johnny Walker Red Label


Food
Beverages
Other
White
Red


Bever-
ages
Wines
White
Red