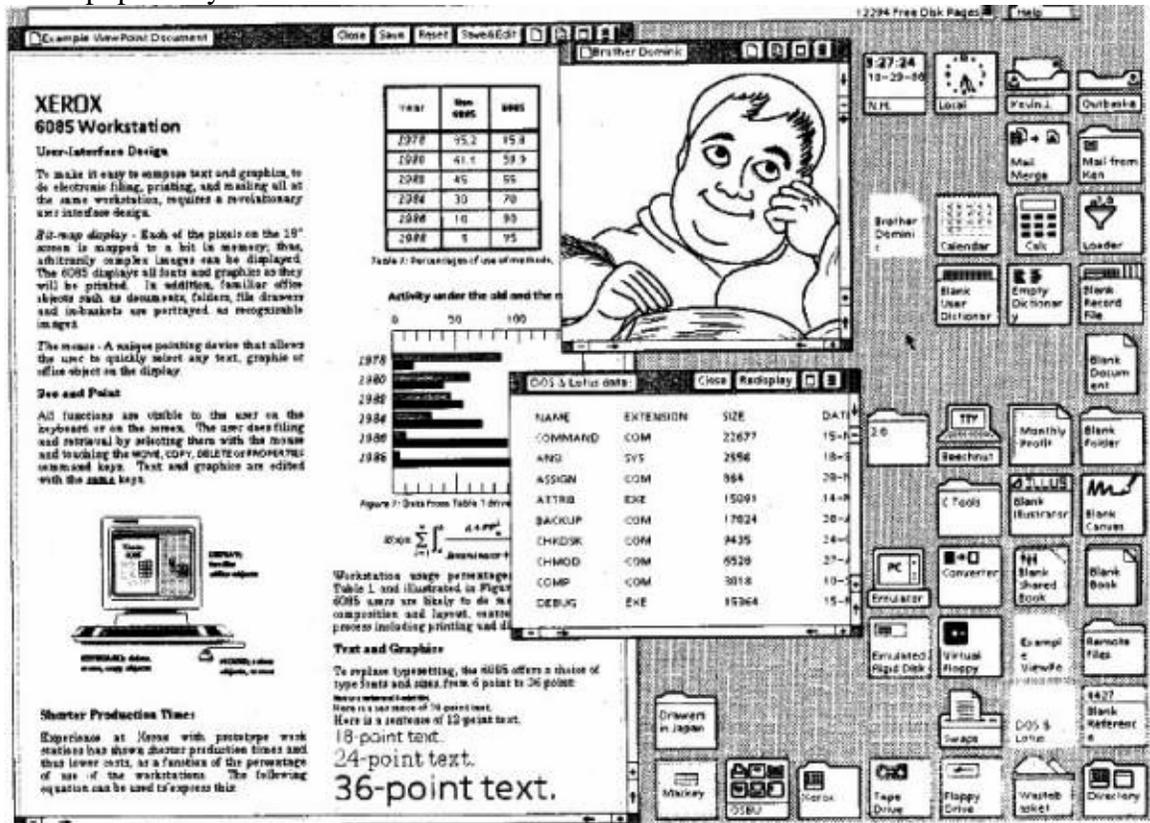


GUI components

Windows

Primary means of presenting applications and displaying information

- Users manage resources pertaining to their tasks; e.g. “working set” of windows
- An application window first appears, then “child” windows are used to manipulate data
- Made popular by Xerox Star



Window

An area on a computer screen usually rectangular and usually delimited by a border that contains a particular view of some data in the computer

Presentation Style

- Tiled; never overlap
- Overlapping
- Mixed

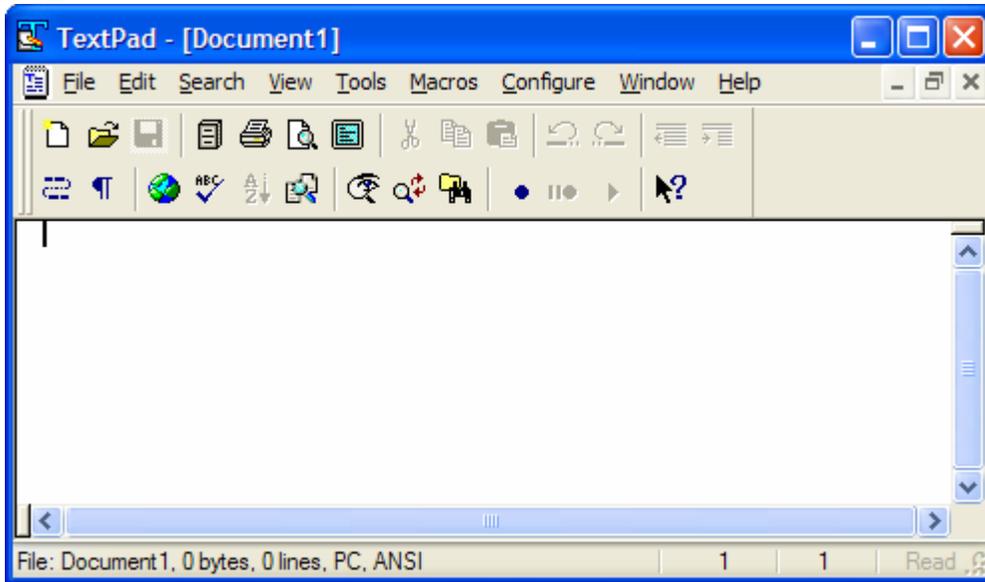
Interaction Styles

- Direct Manipulation - dragging
- Indirect Manipulation - use of commands
- Hybrid - combination of both

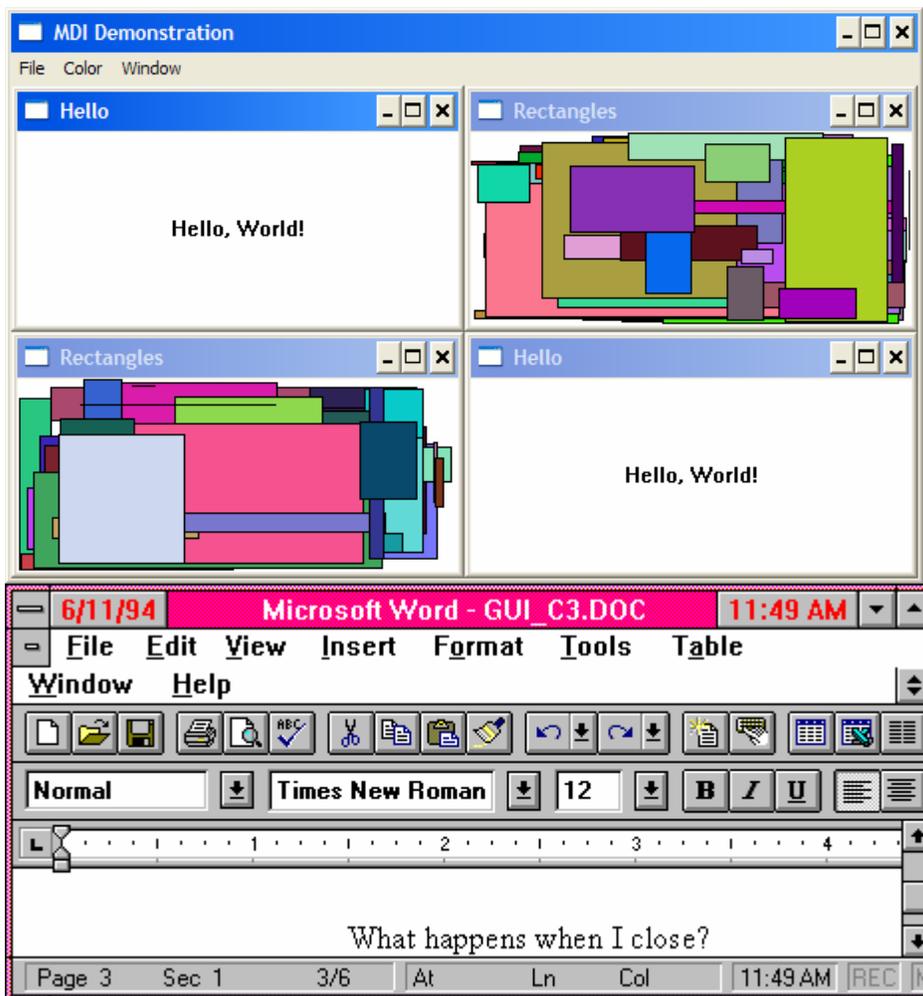
Sets of Operations

- Making window visible
create,
delete,
open,
close,
bring to front,
put in back,
shuffle,
minimize,
maximize,
revert to previous
- Manipulate window & information
move,
resize,
zoom,
rescale,
scroll,
name/rename
- Other functions
end,
system menu,
etc.
- Minimized = Icon
- Maximized = Full screen
- Windowed = Part of the screen

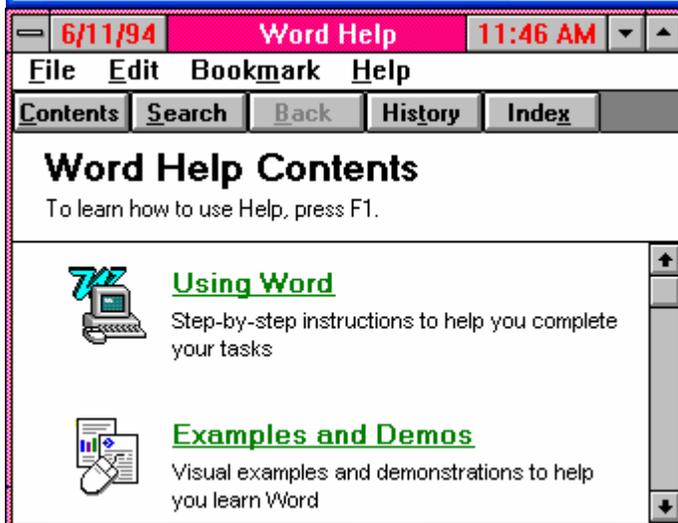
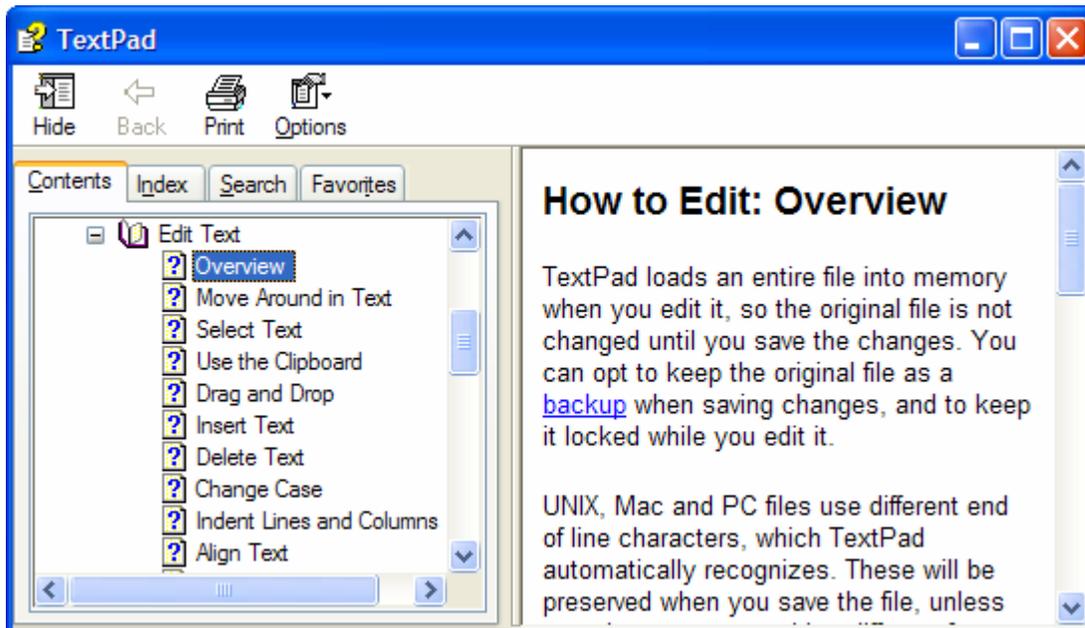
Application Windows



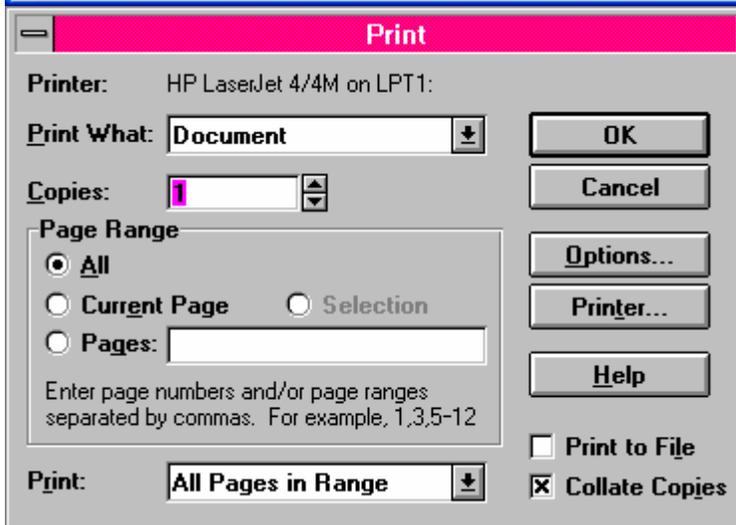
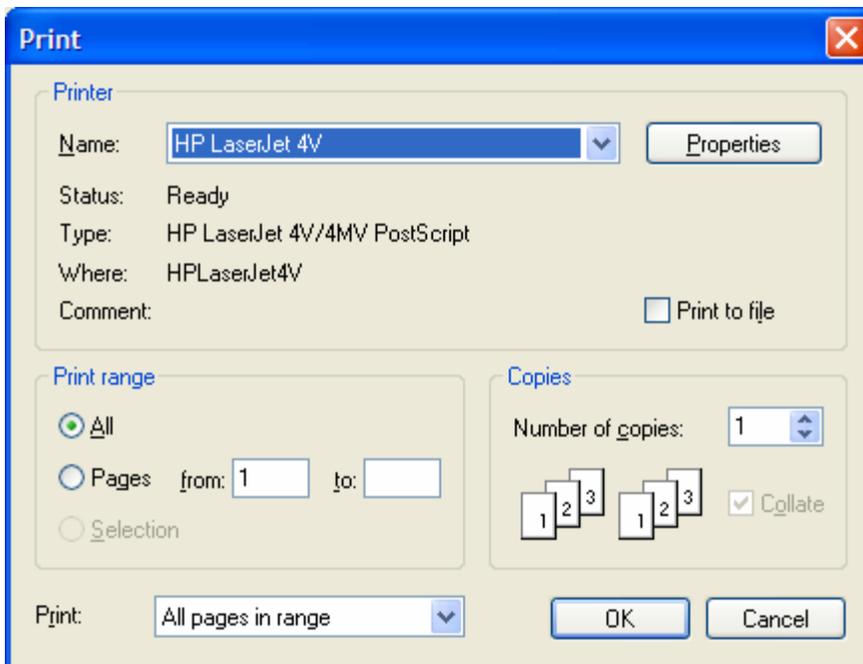
Multiple Document Interface Windows



Help Windows



Dialog Boxes



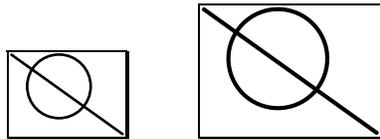
Icons, Bitmaps and Drawings

We are all trained from grades 1-12 to write
Art classes are an afterthought

- presentation is enhanced by good graphics
- presentation is hurt by bad graphics

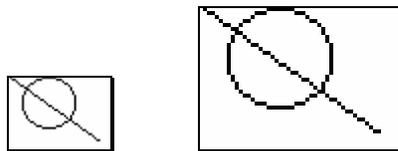
A drawing is a a vector or object diagram

- WMF
- VSD
- CGM
- EPS



A BMP is a raster or pixel only object.

- TIFF
- PNG
- BMP
- GIF



Menus

- Will be covered in another class

Input

- Pointing
- Selecting
- Data entry

MS Windows

Keyboard 104+ keys

Left, Right and Middle mouse buttons

Clues to drag and drop

One of two times you should change a cursor inside an application window
What is the other?

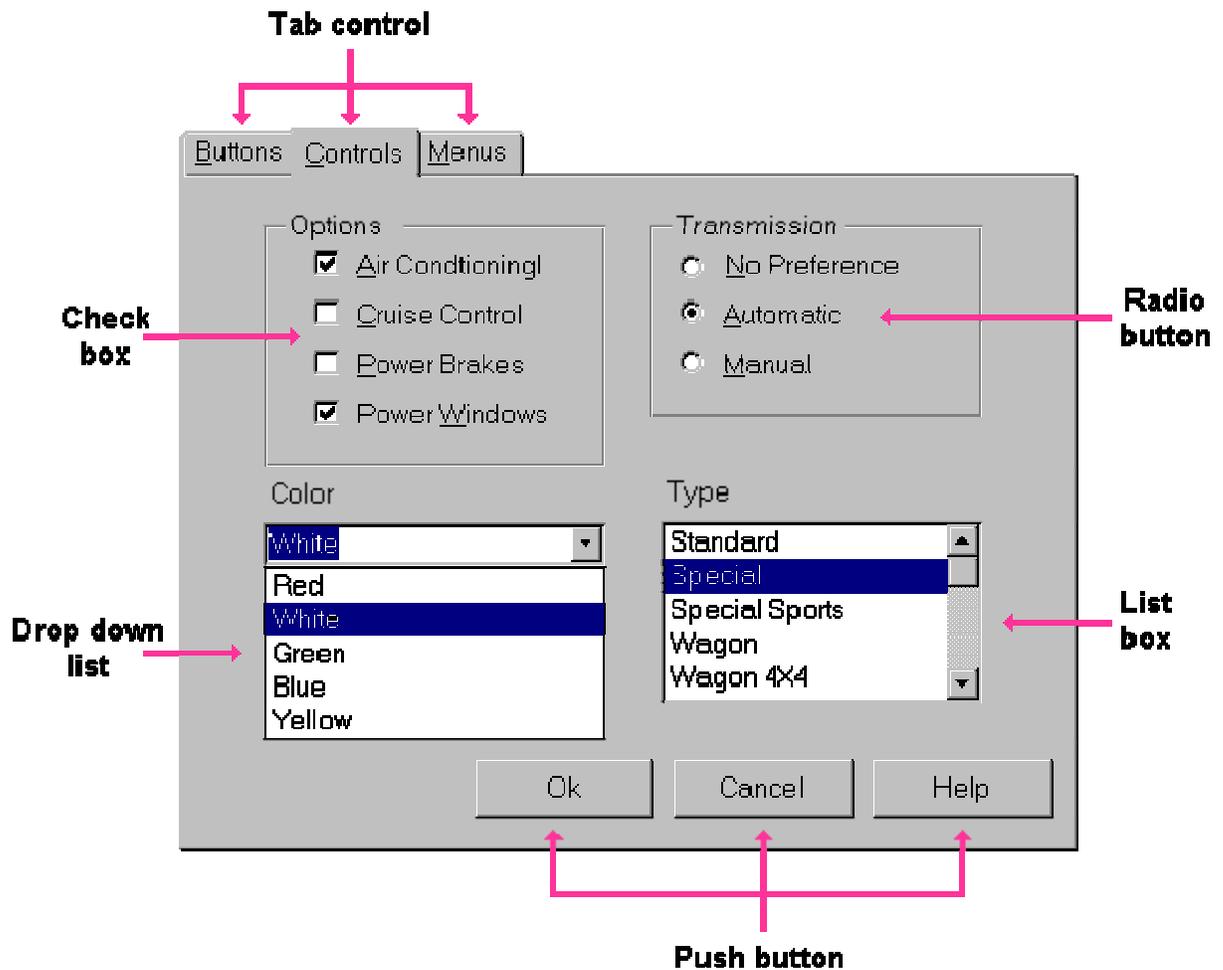
Shortcut and Access keys

What if the mouse breaks and its due in an hour?

Controls and Control Panels

Control

A visual user-interface component that allows a user to interact with data. Controls are usually identified by text; for example, headings, labels in push buttons, field prompts, and titles in windows.



When To Use

- Provide a control in a window to allow a user to interact with information.

Guidelines

- Identify each control or field of controls with a field prompt, a column heading, or a window title, whichever is most appropriate.
- Use a label that clearly indicates the control's function.
- Use the controls provided by the operating environment rather than creating new ones.
- Avoid changing the function, interaction technique, or appearance of controls provided by the operating environment.
- When a window is sized, adjust the size of the controls. For example, make the entry fields longer or shorter as the window is sized larger or smaller. At some minimal useful size, clip the controls instead of adjusting the size.

Use of real-world objects to indicate to a user what to control and how

- buttons,
- sliders,
- knobs,
- etc.

Five essential classes of information:

1. Exclusive Settings
2. Nonexclusive settings
3. Proportional Slider Controls
4. Text Fields and Lists
5. Command Buttons

Exclusive Settings

- Select exactly one value from a group of possible selections

Radio buttons

Value Sets

Nonexclusive Settings

- Lists that can be independently set and cleared by the user
- Any number of nonexclusive settings can be indicated simultaneously
- For example, check lists

Check Boxes

Toggles

Proportional Slider Controls

- Used for values that may be modified within a known range
- Good when there are too many values for exclusive settings

Scroll Bars

Sliders

UpDown

Text Fields and Lists

- Text fields allow input of alphanumeric information
- Lists are usually read-only
- Scrolling lists are used when there are too many items for the allotted space

TextBox

Lists

ComboBox

Labels?

Command Buttons

- Produce an immediate change in the state of the system
- Like menus, they represent commands
- **Action buttons** - OK, Cancel, Close
- **Jump buttons** - Close the current and goto a new dialog box
- **Jump and Return** - Like the Printer button on the Print Dialog
- **Unfold** - Provides technical options that aren't usually necessary

Command

Control Panels

- Can provide consistent access to frequently needed functions
- Usually there are preselected default settings that a user may confirm with a simple action
- Group Boxes - to group radio or check boxes usually

Forms

Frames

Panels

Dialogue Boxes

- Modeless—limited in scope, do not restrict user's operations
- Modal—require user to respond before any other action can be taken
- ☺ nonmodal - just lingers, waiting for input = toolbox
- application modal - application is waiting = print
- ☹ system modal - system is waiting = General Protection Fault
- application semimodal - can be ignored = spreadsheet selec
- Movable, sizeable, or fixed
- Message or dialog

- Whether they unfold

Query Boxes

- Allow system to request a specific piece of information from the user
- Often modal; may be modeless
Logon & Password

Message Boxes

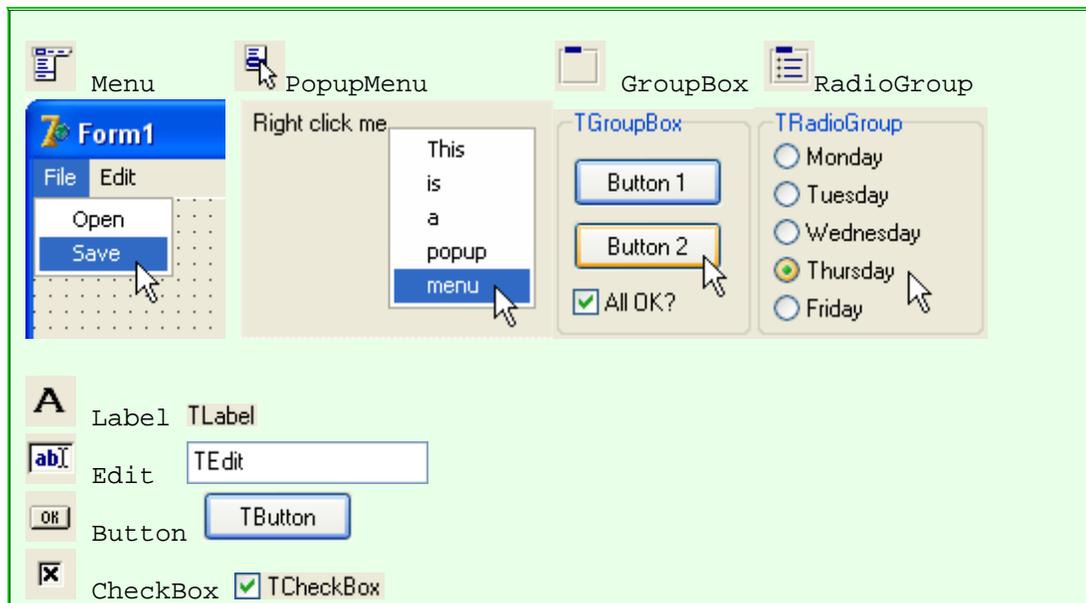
- Provide notification that a certain state has been entered
e.g. “illegal” state;
- user may or may not have a choice to make

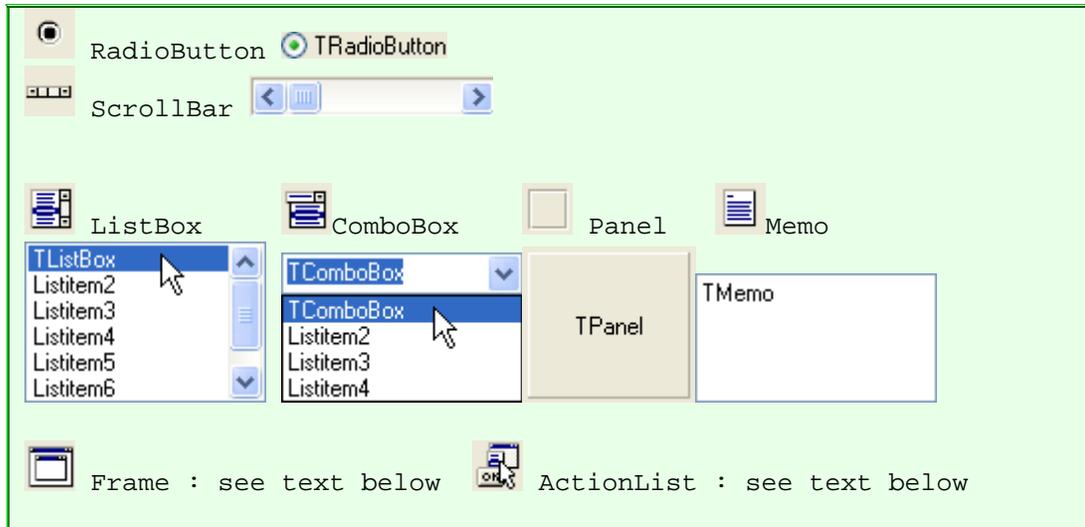
GUI components via Delphi

GUI stands for Graphical User Interface. It refers to the windows, buttons, dialogs, menus and everything visual in a modern application. A GUI component is one of these graphical building blocks.



Each of the components is itemized below with a picture of a typical GUI object they can create:





Menus

A list of choices (menu items) logically grouped and displayed by an application so that a user need not memorize all available commands or options.

Popup menus

A popup menu appears in many applications when you right click on something. For example, when you right click the Windows desktop. You create a popup menu by adding the popup menu component to your form and double clicking it. You then simply type in your menu item list.

A Labels

Labels are the simplest component. They are used to literally label things on a form, but the text, colours and so

on can be changed by your code. For example, you can change the label colour when the mouse hovers over it, and can run code when the user clicks it. This makes the label like a web page link. Normally, they are just kept as plain, unchanging text.



Edit boxes

An edit box allows the user to type in a single line of text. For example, the name of the user. You set up the initial value with the **Text** property either at design time or when your code runs.



Memo boxes

A memo box displays a single string as a multi line wrapped text display. You cannot apply any formatting. The displayed lines are set using the **Lines** property. This may be set at design time as well as at run time.



Buttons

A button is the simplest active item. When clicked by a user, it performs some action. You can change the button label by setting the **Caption** property. Double clicking the button when designing adds code to your form to run when the button is clicked at run time.



Check boxes

Check boxes are used to give a user a **yes/no** choice. For example, whether to wrap text or not. The label is set using the **Caption** property. You can preset the check box to ticked by setting the **Checked** property to **true**.



Radio buttons

Radio buttons are used to give a user **multiple** choices. For example, whether to left, centre or right align text. The

label is set using the **Caption** property. You can preset a radio button to selected by setting the **Checked** property to **true**.

You would normally use radio buttons in groups of two or more. The **RadioGroup** component allows you to do this in a neat and dynamic way.



List boxes

List boxes provide selectable items. For example, a collection of fish names. If you set the **MultiSelect** property to **true**, you allow the user to select more than one. The items in the list are added using the **Items.Add** method, passing the string of each item as a parameter.



Combo boxes

A combo box is like a list box, and is set up in the same way (see above). It just takes up less space on your form by collapsing to a single line when deselected, showing the chosen list item. It is not recommend to use one for multi line selection.



Scroll bars

Many components have built in scroll bars. For those that don't, you can use this to do your own scrolling. You link the scrollbar to your component.



Group boxes

A group box is like a panel. It differs in that it gives a name to the collection of components that you add to it. Use a group box to help the user see what controls affect one particular aspect of the application.



Radio group panels

Radio buttons are used to give a user a **multiple** choices. For example, whether to left, centre or right align text. Unlike individual radio buttons, a group is only set up by your code.:

Empty panels

When building your form, you might want to add many components. These may fall into logical groups. If so, you can add each group to a panel, and use the panel to position the whole group on the form. The panel name can be blanked out by setting the **Caption** property.

Figure 3 Recommended Usage of Text-Entry Controls

Control	Number of Selectable Choices	Length of Entry Field (characters)	Types of Selectable Choices	Shown As	Relative Space Used	Selection Type Supported
Entry field (single line)	None	60 or fewer	None	Alphanumeric	Low	Text
Entry field (multiple-line)	None	Any	None	Alphanumeric	Medium - high	Text
Combination box	Any number	60 or fewer	Variable settings choices or objects	Alphanumeric	High	Single choice in list and text in entry field
Drop-down combination box	Any number	60 or fewer	Variable settings choices or objects	Alphanumeric	Low	Single choice in list and text in entry field

Figure 4 Recommended Usage of Non-Text Entry Controls

Control	Number of Selectable Choices	Types of Choices	Shown As	Relative Space Used	Selection Type Supported
Push button	1 for each push button, 6 or fewer per field	Fixed action or routing choices	Alphanumeric, Graphic	Low	Single
Radio button	1 for each radio button, 6 or fewer per field	Fixed setting choices	Alphanumeric	Medium	Single
Value set	20 or fewer	Fixed settings choices	Short alphabetic,	Medium	Single

			Numeric, Graphic		
List box	Any number	Variable settings choices or objects	Alphanumeric, Graphic	Medium - High	Single, Multiple
Drop-down list	Any number	Variable settings choices or objects	Alphanumeric, Graphic	Low	Single
Check box	1 for each check box, 6 or fewer per field	Fixed settings choices	Alphanumeric, Graphic	Medium	Multiple
Menu bar	10 or fewer	Fixed routing choices	Alphabetic, Graphic	Low	Single
Pull-down menu	15 or fewer	Fixed action or routing choices	Alphanumeric, Graphic	Low	Single
Cascaded menu	15 or fewer	Fixed action or routing choices	Alphanumeric, Graphic	Low	Single
Pop-up menu	15 or fewer	Fixed action or routing choices	Alphanumeric, Graphic	Low	Single
Container	Any number	Objects	Alphanumeric, Graphic	Low	Single

Figure 5 Recommended Usage of Tab Control

Control	Number of Objects and Controls	Types of Objects and Controls	Shown As	Relative Space Used	Selection Type Supported
Tab	Any number	Any (except another Tab)	Alphanumeric, Graphic	Medium - high	As appropriate for each object or control

Guidelines

- Design the initial size and layout of an action window so that a user will not need to scroll the window.
- Within a column of controls, left-align the edges of the controls to the right of the longest field prompt.
- Single Line Text should be defined as Select on Entry
- Arrange controls in a logical order, using rows and columns. Determine the users preferred order
- Align fields and controls vertically and horizontally, so the cursor moves in a straight- line pattern.
- Group related items together. Identify groups with group headings. Separate by using blank lines or spaces. Use a group box only when a group heading or blank space does not visually distinguish groups of fields or controls.

- Use pre-selected choices and pre-filled entry fields whenever possible.
- Combine components to be contextually sensitive to the users need.
- Make windows modeless whenever possible, and determine the most usable default location and size.
- Make sure that the initial cursor placement and the tabbing sequence make sense to the user.
- Place push buttons that affect the entire window horizontally at the bottom of the window and justified from the left edge.
- Avoid placing push buttons in more than two rows.
- If the action indicated by the label on a push button adjusts or is associated with a component within the same window as the push button, place the push button near that component. For example, if the function of a push button is to restore the initial value in an entry field, place that push button beside the entry field that it affects.
- Avoid using a push button to change the size of a window; instead, allow a user to change the size of the window using the size borders and the maximize push button or Maximize choice. For example, do not provide a push button labeled More>> to allow a user to enlarge a window.
- Use a Tab to organize groups of controls when they will not fit in a single window. Avoid placing related controls in separate windows connected by routing choices on push buttons.
- Place a field prompt above or to the left of the field it is associated with.
- Left-aligned entry fields so that a narrow vertical column of space is left between the field prompts and their associated entry fields to allow a user to quickly scan the choices in the window.
- Allow a user to adjust the size of each column in a window, where appropriate. For example, provide column borders that a user can directly manipulate to change the size of columns.
- If a user cannot adjust the width of a column or scroll the column, make the column at least as wide as the widest item in the column or as wide as the column's heading, whichever is wider.
- If a user can adjust the width of a column, make the initial width of the column wide enough to display choices of average width.
- If the order of columns can be changed, allow a user to directly manipulate each column. For example, allow a user to drag column headings to change the order of columns.
- If you provide an information area, place it at the bottom of a window.
- If you provide an information area and there is a horizontal scroll bar in the window, place the information area below the scroll bar and above the window border.

- If you provide a status area, place it below the title bar and below the menu bar, if a menu bar is provided, and above the rest of the window. For example, place a status area below the menu bar and above column headings in a window.
- Use the standard
 - File Open
 - File Save
 - Font
 - Color
 - Print Dialogs
- Make an “About...” Dialog box

For your presentation

- Try explaining it to anyone
- Have two floppy disks, thumb drives, CDs, download site
- Make sure it opens on a machine it was not designed on
- Title the application
- Complete something