User Interface Design
Some Interesting Delphi Components

tCustom...

tCustom... components are not instantiated themselves

tCustom... components are used as superclasses

act as repositories for generic fields and behaviours

Fields and behaviours are available for inheritance by multiple descendants

Inherit fields and behaviour from their ancestors

Have their own fields and behaviours – often protected

Generally have multiple descendants

Pass both types of fields and behaviour on to their descendants

Descendants can publish protected members
Some Interesting Delphi Components

TEdit

Wrapper for Windows Edit component
displays text and captures input
simplistic – just a text string, no concept of separate lines

The time has come, the walrus said, to speak of many things
Of shoes and ships and sealing wax, of cabbages and kings
Of why the sea is boiling hot, and whether pigs have wings

Can select text, and alter selected text
  selStart := 0;
  selLength := 8;
  Seltext := ‘The time’;
changes ‘The hour’ to ‘The time’

tEdit has font, style and size properties – can be set for the whole edit box
Some Interesting Delphi Components

**TRichEdit**

Wrapper for Windows RichEdit component

Allows program to alter font characteristics of some text

Text property from `tCustomEdit` is still available

But a new property - `lines: tStrings`

allows the program to treat the memo as an array of strings

Interesting properties of `tStrings`

- **Count** – no of string in the list
- **Strings** – the actual array of strings

Interesting methods of `tStrings`

- **Add** – adds a new string onto the list
- **Delete(index)** – deletes a string from the list
- **Get(index)** – gets the string at a particular location

Var `thisRichEdit : tRichEdit;` declares a new `tRichEdit`

thisRichEdit.lines.strings[5] := 'the sixth line'

thisRichEdit.lines[5] := 'the sixth line'; also works
**Some Interesting Delphi Components**

**tRichEdit**

Wrapper for Windows RichEdit component

Allows program to alter font characteristics of *some* text

**Interesting properties of tCustomRichEdit**

- `SelAttributes` returns or sets font characteristics of currently selected text

```pascal
richEdit1.SelAttributes.Color := clRed;
richEdit1.SelAttributes.Height := Height + 5;
richEdit1.selAttributes.Style := [fsBold]
```

When inserting new text, the font characteristics of the new text will match `SelAttributes`.

**Interesting methods of tCustomRichEdit**

- `getSelStart` & `getSelLength` return selected text parameters
- `print`
**Some Interesting Delphi Components**

**TSPLITTER (on the Additional tab)**

TSplitter divides the client area of a form into resizable panes.

First, create a left-aligned control. Then add a `tSplitter`, aligned identically. Finally, add a client-aligned control. Dragging the splitter automatically resizes both controls. SPLITTERS work both horizontally and vertically. A form can have multiple splitters.
Cut, Copy, & Paste

EVOLUTION OF APPLICATIONS

Single-purpose applications

↓

Multi-purpose applications

↓

Single-purpose applications + cut, copy, paste, with OS support

Multiple storage formats for multiple uses
Source app is responsible for format conversion
Multi-format means large amounts of information in clipboard

↓

Single-purpose applications + publish and subscribe

↓

Single-purpose applications + embedded editing (OLE)
CLIPBOARD MULTIFORMATS

e.g. Excel spreadsheet can be:
   - pasted into another Excel spreadsheet (with formulae)
   - Pasted into a text editor (c.s.v. format)
   - Pasted into a diagram (picture format)

Macintosh has 4-character format names
   TEXT  PICT  EXCL  Etc.
   \{ Required for all applications

Other systems (X Window, MS WINDOWS) allow longer strings, and generate a unique id no for each format

Apple maintains a registry of these names

MS Word
   copies scraps in up to 15 formats \Rightarrow heavy memory usage
   pastes using WORD format 1\text{st}, then Excel
**XWINDOW**

**Demand-based pasting**

Graphics system on UNIX
allows *client* software to run on a remote machine and
display output using graphics *server* software on a local host

Client$_A$ generates window$_A$; Client$_B$ generates window$_B$

Programs have no shared memory
But windows have *properties* (named data spaces on the server)
Program$_A$ can paste into the property of window$_B$

When program$_A$ “copies” information,
it really only registers itself with the server as owner of the “selection”

When program$_B$ “pastes” information
it tells the selection owner the format required, and name of destination property
Prog$_A$ writes info in requested format, to specified property **OR**
refuses, and prog$_B$ specifies an alternative format till gets info, or no more formats

Xwindow equivalent of the clipboard

Demand-based pasting
DEMAND-BASED PASTING in MS WINDOWS

Combines Mac and X Window approaches

Single display, processor and OS, but async processing complicates things
   Apps brackets clipboard operations with openClipboard and closeClipboard calls

When an application has information to copy, it calls
   emptyClipboard
   setClipboardData
   once for each information format (highest priority first)
   Either passes pointer to a copy of the information immediately or a null pointer
   If null pointer is used, information in that format will only be generated on demand

When an application wishes to paste information, it negotiates the format
   isClipboardFormatAvailable if it handles only 1 format
   getPriorityClipboardFormat to get info in the source program’s highest priority format
   enumClipboardFormats to choose the most suitable format for itself
DEMAND-BASED PASTING in MS WINDOWS

After negotiating format, pasting program calls `getClipboardData`
   If information is in the clipboard, paste occurs
   If not, system sends `WM_RENDERFORMAT` message to source program
   Source program generates data, calls `setClipboardData`, passing pointer to the info

Information is returned to pasting program by `getClipboardData`

Problem:
   if source program pastes null info, terminates, and then paste occurs
   Windows sends `WM_RENDERALLFORMATS` message to program when terminates
   Forces prog to fill the clipboard with real information
PUBLISH AND SUBSCRIBE

Clipboard model supports static transfer of information

Dynamic transfer of information is often desirable
  e.g., a Word document that incorporates a spreadsheet that is regularly updated

Should update occur
  whenever source document changes? large overheads
  whenever source document is saved?
  whenever user requests them? tedious for user
PUBLISH AND SUBSCRIBE

Clipboard model supports static transfer of information

Dynamic transfer of information is often desirable
  e.g., a Word document that incorporates a spreadsheet that is regularly updated

On the Mac, user selects information, then *publishes* it

Source application
  puts the information into an *edition* file in multiple formats
  updates the edition file at each save – can’t delay updates
  keeps section reference to published information (e.g. cells A3:B5 in Excel)
  must keep track of alterations (e.g. inserting column between A and B)

In subscribing application
  User specifies destination location of information, and selects the edition file
  App creates a section reference so that subsequent updates occur in the right place
  App scans through representations of information for appropriate format and retrieves it
  Update checks occur when file is opened, and when publisher creates an update event
**EMBEDDED EDITING**

OLE (Microsoft) and OpenDoc (Apple)

Makes “foreign” data editable in place; no format negotiation required
  
  e.g. equations in Microsoft Word; Word is the *container* app; MathType is the editing app

Container may incorporate the data (equation) or a link (e.g. spreadsheet)

Embedded data can be stored as a file within a file, or as a byte stream
  
  Source application has Serialize method to generate byte stream

Container has to present embedded information
  
  Container gets size of presentation area from the source application
  Container then requests source application to render the information
  Source app generates a set of drawing commands to render the information (a *metafile*)

**Editing**

*Edit aside* – occurs in a separate window
  
  data is converts to a byte stream using Serialize and returned to the container app

*Edit in place* – source app is given a window inside the container’s window
  
  Menus and tool palettes must be integrated with container app
An important aspect of interface usability
Nearly everything should be undo-able
User should be warned before non-undo-able actions (e.g. save to file)

A single operation may comprise several user actions
Mousedown
Drag
mouseup

A drag operation: should be treated as atomic

Simple (but slow) approach:
Keep a history of all commands since session started, and a record of the initial state
When undo occurs,
delete last command from the history list
Restore initial state
Re-execute commands in the history list

More commonly
Store a restricted no of commands (often 1)
Store inverse function with each command
OMITTING COMMANDS FROM THE HISTORY

Irreversible commands
- e.g. print command – genuinely irreversible
- save command – too expensive to reverse
- get the user to confirm the action
- does not place the action on the history list

Transient commands
- Do not affect the data model
- e.g., scrolling within a window, bringing a window to the front
SELECTIVE UNDO

Consider a sequence of drawing actions

1. Create A, a white rectangle
2. Change colour of A
3. Draw B, a curve
4. Draw C, a line

User decides to restore rectangle’s original colour

With selective undo, could reverse action 2 without losing interim actions

Consider another sequence

1. Create A, a white rectangle
2. Change colour of A
3. Draw B, a line
4. Duplicate A as C
5. Move B

What should happen if action 3 is removed?

Duplicated rectangle would now be object B
Move must not operate on duplicated rectangle
HIERARCHICAL UNDO

Dialog boxes are a single command, but often change multiple properties

e.g. Format paragraph

alignment
space
indentation

before
after
left
right

Should undo reverse all the settings, or just the last one?
Actions need to be kept and displayed as a hierarchy, so user can choose
OBJECT-BASED UNDO

(Borges and Quiñones
HCI International 2003)

Associate actions with objects
   Select an object
   Specify undo; Most recent action on that object is reversed

Would work well in CAD tools, drawing tools
   Individual objects are the basis of the underlying model
   Problems?
   Must have some way of undoing
      a delete action,
      actions on groups of objects
   Possible also actions on the window (e.g. scrolling, shift in focus)

Not so suitable in a text editor
   When you type in some text, which object has been modified?
Macros

Allow users to add their own commands to an interface
Group a sequence of commands
Give them a name
Associate them with a keyboard shortcut

Command-line interfaces (e.g. UNIX)
Just string together a list of commands
Needs a facility for supplying parameters
Conceptually simple and straightforward to implement

Not so straightforward in GUIs
Scripting languages differ significantly from interface
Instead, record command sequences
PARAMETERISING THE COMMANDS

Just repeating mouse events does not provide a very powerful macro facility
Events would always recur in the same position
How do we replay the macro in different locations?

Simplify by recording complete commands, not individual events
Could make later commands relative to location of first
When replaying macro,
    user selects start point & commands are replayed relative to start point

What if sizes of Xs should relate to sizes of enclosing boxes?
Could select the box first; then record macro commands relative to selected object
Then user would select box before replaying macro
Or user could rehearse commands twice, & system could infer macro (PBD)
ASYNCHRONOUS GROUP WORK

Supports multiple users working on the same data simultaneously or in sequence

In asynchronous model, users are not aware of each other’s actions
   Each modifies a separate copy of the data
   e.g. complex architectural design has a number of interacting components
       Floorplan, air conditioning, networking, plumbing
   Separate work phases must be followed by integration phase
   Integration may force re-working of some aspects

Each set of changes to the data is a patch
   not necessarily equivalent to a single session

Resolving conflicts between patches is important
   One user moves a door 1500mm right, another moves it 1500mm left: who’s right?
   Some systems number objects sequentially or put them in a list in creation order
       One user deletes a door; subsequent object references differ for the two users
Conflicts can be prevented by
  locking modules when someone is working on them
  Providing an integration mechanism for resolving conflicts

Necessary for users to
  Understand what they have changed
  Understand what others have changed
  Understand how to resolve conflicts between sets of changes
SYNCHRONOUS GROUP WORK

Users working simultaneously are aware of each other’s alterations
Cooperative working

Multiple workstations must be updated over a network
Each user command goes into a command history (cf. undo)
Command history is propagated over the network

Note that commands may come from any user
As in asynchronous groupware, conflicts must be resolved
  e.g. drawing actions can’t be in terms of screen coordinates
  Though different users may be viewing same data, they may have
    different viewports
    different zoom ratios

Commands must be represented at a more abstract level than screen coordinates
COMMAND OBJECTS

We usually give a computer commands to operate on data.
Sometimes we want to operate on the commands themselves.
Can be useful to have objects that represent commands.

Undo, Groupware and Macros all rely on a system-level record of commands.

Commands must be abstract.
Screen coordinates and event lists are too concrete.

Meyer (Object Oriented Software Construction) introduced Command Objects.
Refined in Apple’s MacApp.

Abstract class called Command, with 3 methods:
- doIt
- undoIt
- redoIt
COMMAND OBJECTS

Consider a CAD program
User creates & links electronic components
There’s a library of circuit design functions
Each new component is added to a list

- addChip(chipType, centrePoint)
- addWire(chip1, term1, chip2, term2)
- selectChip(chipNo)
- moveChip(chipNo, newCentrePoint)
- changeChipName(chipNo, cewName)
- deleteChip(chipNo)
- selectWire(wireNo)
- deleteWire(wireNo)

\[ \text{AND} \quad \text{OR} \quad \text{XOR} \]
**COMMAND OBJECTS**

Could simply call a function when a sequence of user input events occurs.

e.g., select chip to add, then click at chip centrepoint → `addChip(centrepoint)`

For undo, groupware, & macros, need a way to record commands long-term.

```plaintext
Command Objects:
- addChipCommand
- addWireCommand
- selectChipCommand
- moveChipCommand
- changeChipNameCommand
- deleteChipCommand
- selectWireCommand
- deleteWireCommand
```

```
addChip(chipType, centrePoint)
addWire(chip₁, term₁, chip₂, term₂)
selectChip(chipNo)
moveChip(chipNo, newCentrePoint)
changeChipName(chipNo, cewName)
deleteChip(chipNo)
selectWire(wireNo)
deleteWire(wireNo)
```

Diagram:

```
  AND
 /   \
|     |
\   /
  OR
```

```
1 AND
2 OR
3 XOR
```
**A Command-Monitoring Architecture Using Command Objects**

**UNDO**

Could simply call a function when a sequence of user input events occurs
e.g., select chip to add, then click at chip centrepoint → addChip(centrepoint)
For undo, groupware, & macros, need a way to record commands longterm

```
addChipCommand
addWireCommand
selectChipCommand
moveChipCommand
changeChipNameCommand
deleteChipCommand
selectWireCommand
deleteWireCommand
```

Command Objects

Each Command Object has
- a **doIt** method (the functions shown above right)
- an **undoIt** method
- a **redoIt** method
- properties for publicly accessible data (e.g. centrePoint, for an addChipCommand)
A Command-Monitoring Architecture Using Command Objects

UNDO

Could simply call a function when a sequence of user input events occurs
e.g., select chip to add, then click at chip centrepoint → addChip(centrepoint)
For undo, groupware, & macros, need a way to record commands longterm

When user generates the add-a-chip event sequence, the application...
  creates addChipCommand object
  puts centre coordinates into the object’s public property centrePoint
  invokes command object’s doIt method
    creates the chip and stores an ID no for it
  adds addChipCommand object to history list

When user selects undo menu item (or ctrl-X)
  undo menu item’s method invokes historyList.topCommand.undoIt
    Undo method is completely decoupled from undoIt method
  addChipCommand’s undoIt method runs deleteChip method on the chip
  Command object is then added to redo list
    redoIt usually invokes doIt
Consider deleting a chip

User selects chip; tells the interface the chipNo.
User selects chip delete command,

Interface creates deleteChipCommand command object
assigns no. of doomed chip to the object’s chipNo property
Interface calls command object’s doIt method
and the doIt method calls deleteChip method
deleteChip method
  • records, in the command object
    name of deleted chip
    its centre point
  • deletes the chip

Interface adds deleteChipCommand command object to history list
UNDO

There can be subtle problems with a history of command objects.

What if the user had performed an earlier operation on the chip...

If user deletes a chip, (3, say)
    then undoes the delete,
    the recreated chip
    gets a new number (9, say)

    changeChipName(3, “subtract-enable”)
    deleteChip(3)
    undo
    undo

    → Chip 9

...and then tries to undo it?

    Undo references a chip that no longer exists
A Command-Monitoring Architecture Using Command Objects

UNDO

There can be subtle problems with a history of command objects…

What if the user had performed an earlier operation on the chip…

```
changeChipName(3, "subtract-enable")
deleteChip(3)
undo
undo
```

→ Chip 9

Problem is caused by using addChip to reverse a delete
allocates a new chip number

Could fix this by using special restoreChip method to undo a delete
would need to keep the chip no in the command object for use by restoreChip

Special restoreChip method could also call addwire to restore wires
(if wires attached to the chip, & deleted with the chip, were stored in the command object)
Each user’s changes to a project are saved as a patch

What’s in a patch?

**First approach**: leave the original file unchanged
Patch file consists of the `doIt` information from the command objects
Apply the `doIt` functions to the original file next time it is opened

**Second approach**: replace the original with the modified file
Patch file consists of list of `undoIt` functions and parameters

Command objects need to be able to save their `doIt` or `undoIt` information

Add 3 new methods:

- `commandType`
- `writeDoItInfo`
- `readDoItInfo`

This returns an identifier unique to the command type

These output data to or input data from a stream
A Command-Monitoring Architecture Using Command Objects

ASYNCHRONOUS MULTI-USER SYSTEMS

Writing the patch file:

```plaintext
while C = commandList.next <> nil do
  outputStream << c.commandType
  c.writeDoItInfo(outputStream)
```

These output data to or input data from a stream

(In Java, it’s easier ‘cos serialisation is built in)

Reading the patch file is the converse of this operation

Create a new command object for each command ID in the input stream
Run `readDoItInfo` and reconstitute the command
Add the command to the history list
Run commands in the history list

This returns an identifier unique to the command type
SYNCHRONOUS GROUPWARE

When many users work on a project simultaneously
  Each user’s commands need to be propagated to all the others
  Same architecture supports this requirement
  When commands are executed, they are serialised and sent to other users
SIMPLE MACROS

Command objects + serialisation could also support macro recording

User turns on macro recording…

Application:
- initialises a new history list
- records command sequence till macro recording is turned off
- serialises commands in the history list, saves them to a file

User runs the macro…

Application:
- inputs serialised commands
- reconstitutes them
- runs their `doIt` methods

Problem: objects may be in existence at record-time but not at replay-time

Make object references general – e.g. make them references to currently selected object
MAKING COMMAND OBJECTS TRANSPARENT

Undo, groupware and macro systems are unaware of data model, so far

- OK for undo, but can cause problems with groupware and macros
- Sometimes operations can cause conflicts
- In synchronous groupware, one user sets circle colour to red, another to blue
  Each receives a command from the other that has an unexpected effect
  Furthermore, users’ models are now inconsistent

Groupware system can’t detect inconsistency without knowing about the model

- Can solve by establishing master/slave relationship between workstations
- When master detects conflicting commands from a pair of slaves
  - Chooses one of the conflicting commands (won’t say how)
  - Sends undo command to all workstations for that command
MAKING COMMAND OBJECTS TRANSPARENT

Undo, groupware and macro systems are unaware of data model, so far

OK for undo, but can cause problems with groupware and macros
Sometimes operations can cause conflicts
In synchronous groupware, one user sets circle colour to red, another to blue
Each receives a command from the other that has an unexpected effect
Furthermore, users’ models are now inconsistent

Macro system needs to be able to apply the macro to new objects
Must replace references to record-time object with references to replay-time object
Add a `SpecifyObject(ID)` method to command class
Before executing the `doIt` methods in the macro’s command sequence,
macro system runs `specifyObject(currentSelection)` on the macro
make the macro operate on the current selection
The PowerPoint Drawing Interface
(what can be done?)
Case Study – The PowerPoint Drawing Interface

IS THERE A PROBLEM?

PowerPoint is a mature technology
Standard features are already very powerful
Customisation allows designers to alter the interface to suit their work habits
Classic example of a system that does one thing and does it well

Recent alterations have improved the basic system
Task pane has made custom animation easier to use
Animation order and Z-order are no longer the same

And yet…
And yet, designing with PowerPoint often induces a feeling of tedium, of extra work done
Tension between adding more features and adding more bureaucracy
Creeping featurism is *not* invariably a bad thing
It’s true, no-one uses all the features of an application like PowerPoint
but we all use more than we realise
and we don’t all use the same feature-set – any 10 people use the lot
As features get added, interface designers needs to get more creative
can’t just add more dialog boxes to capture user input (cf. custom animation)
Case Study – The PowerPoint Drawing Interface

TOOLBAR CUSTOMISATION ALLOWS FOR PERSONAL STYLE

PowerPoint vocabulary of drawing tools is extensive so tools often get buried inside multi-level menus but drawing often involves selecting the same tool repeatedly e.g., animating a clockhand This can be both tedious and painful

Keyboard shortcuts can really improve the usability of the interface but some commands lack keyboard shortcuts (e.g. align commands) and some tablet PCs lack keyboards

Customise feature allows user to add any command’s icon to a toolbar Docking allows toolbars (& thus commands) to be mnemonically placed Note

- locations of vertical positioning tools and horizontal positioning tools
- relative positions of the groups
- relative positions of tools within the groups

My style of presentation has only become feasible through customisation but it’s not enough…
KEY BINDINGS

in MS Word, any command or macro can be bound to a key combination
   Built-in key bindings can be overridden
   Macros can be executed by a single key combination
   Interface designers’ oversights can be trivialised with key bindings
   Users’ specialised requirements can be accommodated without major redesign

In MS PowerPoint, no such facility exists
   Drawing operations can be repeated dozens of times
   Anything that can reduce the sheer quantity of interaction makes the system easier to use

Don’t the development teams talk to each other?
   Doesn’t Object-Oriented Programming support reuse?
   Perhaps key bindings are more difficult to incorporate into PowerPoint than it might appear…
AND ON A RELATED NOTE: EDIT-TIME MACROS

PowerPoint supports macros

User can
  give a sequence of operations a name
  make them a command
  associate the command with an action button or some other object on the slide
  run the macro during a presentation by clicking or moving the mouse over on the object

Only presentation-time operations can be macro-ised

Editing operations can’t be glued together and given a name, an icon, or a key-binding
  Presentation creator just has to repeat the operations over and over and over again

Repetitive drawing operations are very conducive to RSI
Case Study – The PowerPoint Drawing Interface

POWERPOINT 3’s BUILD FACILITY – THE PRECURSOR TO SOMETHING GREAT

Single style for building a whole slide
Controlled text build only
Could distinguish between current point and previous points
But no recognition of hierarchy in the text
A clock face
Case Study – The PowerPoint Drawing Interface

Check boxes show which objects are animated (any sort of object can now be animated)
Left pane shows all the objects on the screen (no distinction between text objects)
Currently selected object is highlighted
Right pane shows thumbnail of the whole screen (object being animated is selected to distinguish objects with the same name
The scale’s really too small)

Entry effect
Associated property (speed)
Selected from separate drop-down lists
Shorter list of effects
Extra selection required

I repeatedly try to reorder the events in this pane
Yes, I know there’s an Order & Timing tab, but “perception trumps cognition”
I instantly “see” how to reorder the objects when the Check to animate... pane appears
Text animation is more sophisticated
Hierarchical model
Not completely general yet
Case Study – The PowerPoint Drawing Interface

Order and timing tab

- Left pane controls order
  - User must use arrows to alter the order
  - Objects can’t be dragged
  - Mouse clicks can’t be queued!

- Right pane controls timing
  - *After* alternative offers very fine control which is rarely required
  - Simultaneous animations not supported requires objects to be grouped

Grouping and animation are unrelated
Timing and effects are often edited together
They shouldn’t be on separate tabs

And you can’t see much of the screen!

Underneath the improvements, this version is still based on a “build” metaphor
There’s only one animation per object
Animation only controls arrival of objects onto a slide
Case Study – The PowerPoint Drawing Interface

CUSTOM ANIMATION TASK PANE – A GENUINELY GREAT PIECE OF DESIGN

TOGGLE-ABILITY FOR MOTION-PATH SNAP-TO-GRID FEATURE

Eh?

Custom animations include motion paths

Motion path is represented as an dashed arrow at edit time
  Green arrowhead at the start, Red arrowhead and a line at the end
  Start point snaps to centrepoint of selected object – prevents sudden jumps when motion starts (mostly)

Limited editing of path is possible
  User can nudge the complete path left, right, up & down using arrow keys
  User can drag ends of the path
    End points snap to centre of any nearby object, middle of edge of any nearby object
    Snap seems to alternate between start point and end point

Consider adjusting a very short motion path
  Initially, start end is at centrepoint of object
  We adjust the end point
    The end point snaps to the centrepoint of the object
  We drag the start point
    The start point snaps back to the centrepoint of the object

Should the user be able to toggle this unique snap-to-grid feature of motion paths?
CUSTOM ANIMATION TASK PANE – A GENUINELY GREAT PIECE OF DESIGN

Animation ask pane is easier to use than menus
Once the pane’s there, it stays there

Animation order and Z-ordering are decoupled
Previous version of PowerPoint introduced new objects over old ones

Exit animations and motion paths have been added (though still no morphing)
Formerly exit could be achieved by
covering with a background-coloured shape
no good if background is patterned, gradated or has images
transiting to a new slide while retaining the rest of the current slide
still a useful technique when the slide is graphically complex
With multiple animations, can’t just select the object to change its animation
Click on an animation event in the task pane to change it

Task pane shows
actual text of text objects
On-click/with/after relationship between animations
Case Study – The PowerPoint Drawing Interface

CUSTOM ANIMATION TASK PANE – A GENUINELY GREAT PIECE OF DESIGN

Text object animation, whether text object is selected or cursor is inside text
Individual lines of text can have their own animations, but not parts of lines

Double click on an animation event to edit uncommon parameters in a dialog
delay before animation
no of repetitions
trigger event (e.g., click on another object)
a dialog is a suitable interface component for altering the object options screen!
a user who edited these often could make a strong case for keyboard shortcuts!

Default pane is create mode
bottom of the list
it alters pane to change mode
currently selected effect

How can we tell if an animation is an entrance, exit, emphasis or motion path?
animation pane!
CUSTOM ANIMATION TASK PANE – A GENUINELY GREAT PIECE OF DESIGN

Case Study – The PowerPoint Drawing Interface

Custom Animation Task Pane – A Genuinely Great Piece of Design

User Interface Design (159.7/0)
Case Study – The PowerPoint Drawing Interface

CUSTOM ANIMATION TASK PANE – A GENUINELY GREAT PIECE OF DESIGN
ANIMATION STYLES

Speed of a *fade in* can be very slow
or slow
or *medium* standard is *medium*
or fast I generally prefer fast
or very fast

Settings for a whole set of objects can be altered in one operation
But that style of editing doesn’t seem to fit my slides
Objects may have multiple animations – can be difficult to identify desired animation
Slides may have scores of objects – often overlapping, or overlaid
I like to experiment with animations, adding and removing till they look right

Or the user could copy and paste existing objects
They carry their animations, and animation settings with them
But not their font sizes!

User should be able to create non-standard settings - animation styles
with a name and/or a keyboard shortcut
Case Study – The PowerPoint Drawing Interface

OBJECT SELECTION FROM THE CUSTOM ANIMATION TASK PANЕ

Animated diagrams can often be very complex
overlying objects may obscure underlying objects
it can often be difficult to select an object directly on the screen

In previous versions of PowerPoint, the user could
select an object,
go to the animation dialog & edit the animation for that object
select another object in the animation dialog (& maybe edit the animation for that object)
exit the animation dialog
operate on the last object that was handled in the animation dialog

PPT 2002 identifies the object that’s being manipulated by displaying a numeric label beside it
Allows user to delete an animation effect without deleting the object 😊
but reduces the number of ways a user can get access to an object 😞

As objects are created, they’re added to a list
earlier approach allowed the user to select an object from that list
that’s now gone; no other control provides access to that list
Case Study – The PowerPoint Drawing Interface

TOGGLE-ABILITY FOR MOTION-PATH SNAP-TO-GRID FEATURE

Eh?

Custom animations include motion paths

Motion path is represented as a dashed arrow at edit time
  Green arrowhead at the start; Red arrowhead and a line at the end
  Start point snaps to centrepoint of selected object – prevents sudden jumps when motion starts (mostly)

Limited editing of path is possible
  User can nudge the complete path left, right, up & down using arrow keys
  User can drag ends of the path
    End points snap to centre of any nearby object, middle of edge of any nearby object
    Snap seems to alternate between start point and end point

Consider adjusting a very short motion path
  Initially, start end is at centrepoint of object
  We adjust the end point
    The end point snaps to the centrepoint of the object
  We drag the start point
    The start point snaps back to the centrepoint of the object

Should the user be able to toggle this unique snap-to-grid feature of motion paths?
KEYBOARD SHORTCUTS FOR TOGGLING ORDINARY SNAP-TO-GRID

A somewhat less esoteric case…

User can specify a grid size & turn grid on and off
  Objects will snap to the grid when moved
    Top snaps when moving up, bottom when moving down (which can be quite confusing…)

When grid is on,
  arrow keys move objects to the next grid location
  shift-arrow nudges objects 1 pixel at a time (in the current magnification)

When grid is off
  arrow keys move objects 1 pixel at a time
  shift-arrow moves objects 1 pixel at at time

Grid allows rapid movement of objects using arrow keys
  (Forget the actual snapping-to-grid behaviour; it’s much overrated)
  Fine adjustment of screen objects is a frequent need

User should be able to
  invert grid behaviour of arrow keys and drags temporarily using shift key
  toggle grid mode using shift-ctrl-G (ctrl-G is already used for a Grid-control dialog)
Case Study – The PowerPoint Drawing Interface

FULL SELECTABILITY FOR EDIT POINTS ON A FREFORM, CURVE, OR SCRIBBLE

Freeform curves can be edited
   Edit points appear at points along the curve
   Powerful control of shape is possible
      Control points can be smooth, straight, or corner
      Control points can be deleted
   But control points are not fully selectable
      User can’t select a pair of points and move them the same amount
      User can’t select a point and nudge it with the arrow keys
      User can’t select a point and an object and align them

A more useful curve editing facility would allow the user
   To select a point
   To add a point to a selected set of items
   To move multiple points with a single drag
   To nudge or delete a selected point or group of points

Would not allow the user
   To add a selected point to a group
Case Study – The PowerPoint Drawing Interface

A SENSIBLE WAY TO EDIT THE COLOURS IN THE COLOUR SELECTION POPUP

The colour selection popup has
- 8 predefined colours
- room for 8 user-selectable colours
- An option for accessing the Windows colour picker

Any colour selected in the Windows colour picker is always added to the list of 8 user colours
So the user …
- goes to the colour picker & chooses a colour
- exits the colour picker, looks at the newly coloured presentation object
- decides the colour isn’t quite right
- tries three colours more before getting it just right

Finds that there are now four new, virtually indistinguishable, colours in the popup
Finds that four colours that were there before have disappeared

Why the restriction to 8 colours? A Windows colour is just a 32-bit number
Even if only 8 are displayed, a history and a tool for deleting colours from the palette would help…
Topics to cover:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Pages</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant Delphi components</td>
<td>1-6</td>
<td>✓</td>
</tr>
<tr>
<td>Interesting standard components of GUIs; undo, cut, copy, paste</td>
<td>7-19</td>
<td>✓</td>
</tr>
<tr>
<td>Software architecture for components</td>
<td>14-38</td>
<td>✓</td>
</tr>
<tr>
<td>Critical evaluation of PowerPoint</td>
<td>39-47</td>
<td>✓</td>
</tr>
<tr>
<td>Delphi database programming?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internationalisation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Colour in interfaces</td>
<td>1-160</td>
<td>✓</td>
</tr>
<tr>
<td>Innovative interfaces</td>
<td>1-84</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>12)297</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>