Interactive Devices and Techniques

Designing a GUI

User Model is specified

Describes the GUI functionality

Can simulate the GUI before writing the application

Window-based GUI

Allowing for multiple skill levels

- ease of use for novice
  - simple version that hides complexity

- speed and ease of use for expert
  - fewer prompts
  - more keyboard input versus mouse input

- multiple level help facilities

Consistency

- predictable operations across different tasks
  - eg single and double clicks for selection and action

Back-up and error handling

- undo keys
- ability to replay part of sequence of command

Feedback

- highlight buttons when selected
- echo keys that are typed
Feedback

Feedback to let know processor is busy

Want quick feedback

invert pixel intensities
highlighting
blinking
color changes

Input of Graphical Data

Various Hardware devices

Logical classes of input devices

Locator - Specifies coordinate position (x,y)

Stroke - Specifies series of coordinate positions

String - Specifies text input

Valuator - Specifies scalar values

Choice - Selects menu options

Pick - Selects picture components

Any logical device can be implemented with any hardware device
Some more convenient than others
Locator Devices

Mouse, joystick, trackball, etc.

How can keyboard be used as a locator?
  cursor-keys
  coordinate type-in

Stroke Devices

Stylus, mouse, etc.

How use the mouse - usually use a button as a switch
  hold down for stroke input
  click for locator input

String Devices

Keyboard

How simulate with a stroke device?

Valuator Devices

input rotation, color intensities, etc.

control dials, etc.

  physical rotary potentiometers

  turning dial varies voltage, which could then be converted
to integer value using an ADC
Valuators, cont.

Simulate valuator?

![Graphical slide potentiometer, using physical locator device (slider)]

Use joystick directly

Use keyboard

Choice Devices

enters a selection from a menu

Touch panel, button boxes, etc.

Simulate with a locator and pop-up menus, etc.

Simulate with a keyboard
Pick Devices

Selects part of scene (picture)

Say want to rotate the ellipse

Must select it first

Typically use a physical locator

How does user know which object was selected?

Feedback - highlight, etc.

How does system know which object user meant?

Can check to see cursor location is within the range of the bounding box of object

Bounding box is the smallest upright rectangle that encloses the object

What problem can arise?

Solutions:

Check cursor location is within bounding box of component of object (i.e. a line of a polygon)
Find closest object
User defines pick window that encloses only selected object or component of selected object

Keyboard simulation: type in object names
Input Modes

Applications program is set up to specify which input mode is used when

Request Mode - program initiates interaction by requesting an action

Initiate request - let user know of the request
Put appropriate device into wait state until input received

program and input devices operate alternately

example:
Program requests users identity for validation purposes
will do nothing until gets it

Sample Mode - program samples values of input devices

program and input device operating simultaneously

example:
Program samples a stroke device as user draws a curve
Program also echoes locator location with "pen" on screen at the same time
(Timing is a real issue here - especially if done without a dedicated processor)

Event Mode - input device sends input to event queue

program and input device operate simultaneously

example:
program checks event queue repeatedly for indication user has resized or closed a window
Interactive Techniques

Basic positioning methods

How make lines line up end to end exactly?

Can have coordinate values displayed for feedback

Or use some kind of constraints

Horizontal or vertical constraints

Only allow vertical and horizontal lines

First end point specifies one end of line

Second endpoint determines whether H or V and location

Grid constraints

Fine grid covers drawing area (may not be displayed)

All points input are converted to the closest grid intersection
Gravity Field Constraints

Any locator location within the gravity field of a line is converted to the closest point on the line

The big fields at the end cause lines to join end to end

The lesser field in the middle allows lines to abut each other

What happens if the fields are too large?

too small?

Rubber banding

E.g. selection box in drawing package

Dragging

Select object, then move along with the cursor

What does the program have to do?

Undraw the object at it's current location
Transform to new location
Draw object

For complex object should you redraw at every sample?

perhaps give feedback with just bounding box, etc.