Master Recherche Informatique - Université Paris-Sud

Fundamentals of Human-Computer Interaction

Exam - 27 November 2013 - 2h

Authorized document: a single A4 handwritten sheet.
Please read the questions carefully. Answer clearly, precisely and concisely.

A. Modeling interaction: Magnetic guidelines (8 points)

Consider a vector drawing application on a desktop computer that uses direct manipulation to create and edit basic shapes such as rectangles and circles.

The input device is a mouse. It sends *Move* events when moved, *Down* and *Up* events when the button is pressed, and *Enter* and *Leave* events when the cursor goes over / outside of an object.

1. Show the state machine for selecting an object by clicking it and moving it by drag-and-drop.

2. Modify the above state machine so that objects snap to a grid when moving them. The grid is defined by a gridsize, *G*. When dragging an object, if the cursor gets closer than a distance *S* from a gridpoint, the dragged object is snapped to that point. (But snapping is cancelled if the cursor gets further away from the grid than the distance *S* before releasing the cursor).

You can use the following functions:

- *CloseToGrid(obj, G, S)* returns *True* if the object is at a distance smaller than *S* from a grid of size *G*.
- *GridPoint(p, G)* returns the coordinates of the point of the grid *G* closest to position *p*.
- *GetPos(obj)* and *SetPos(obj, p)* gets/sets an object position.

3. Another way to implement snapping is to use invisible objects in the drawing that represent each gridpoint by a circle of size *S*: Detecting that the cursor is approaching a grid point can then be done by using the *Enter* and *Leave* events sent when the cursor crosses the boundary of any of these objects (Figure 1).

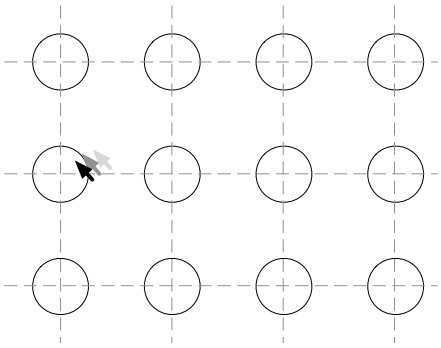


Figure 1:
Cursor enters an invisible circle at a grid intersection and generates an *Enter* event.
A *Leave* event is sent when the cursor moves out of the circle.

Show the state machine that implements snapping with this technique and compare the result with that of question 2.

4. A magnetic guideline is a line that objects snap to when they are moved close to it. Show how the technique from question 3 can be used to create Magnetic Guidelines:

4a. Describe the shape and position of the invisible object associated with the guideline.

4b. Modify the state machine from question 3 to support magnetic guidelines.

4c. Explain what should happen when the user moves the guideline. (Do ***not*** design the state machine!)

Note: For simplicity, you may consider only horizontal or vertical guidelines.

5. Magnetic guidelines are an alternative to the alignment command found in traditional drawing tools. Use the GOMS KLM model, with the following operators, to predict their respective performance:

- K – clicking or hitting a key (0.1s)
- P – pointing (1s)
- D – dragging (1.2s)
- M – Mental processing (1.3s)

Consider the following two methods to align 3 objects:

Alignment command:

A1 – Select 3 objects in sequence (a shift-click adds to the selection).
A2 – Select the vertical alignment command in the Align menu.

Magnetic guideline:

G1 – Create a guideline: Click in a palette to select the guideline tool, then click in the drawing area to create the guideline at the desired position.
G2 – Drag-and-drop each of the three objects onto the guideline.

5a. Compare the times predicted by the GOMS-KLM model for these two sequences, using consistent rules for the placement of the M operator.

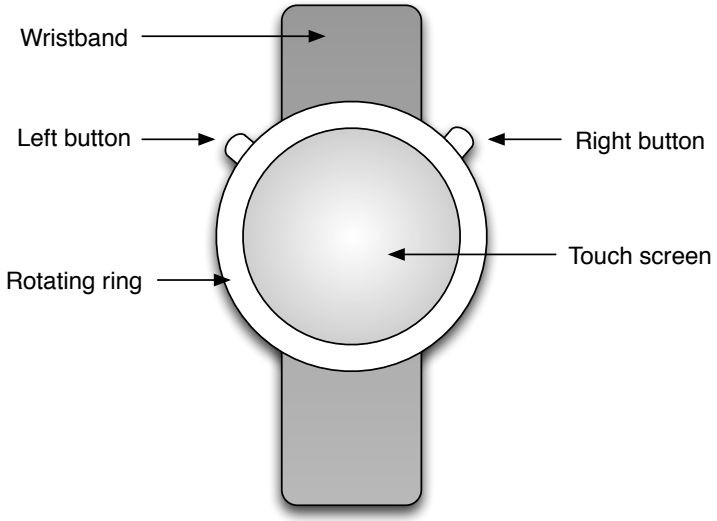
5b. Consider the task of moving the aligned objects. Describe the methods to perform this task with the Alignment command and with the Magnetic guideline. Compare the times predicted by the GOMS-KLM model.

What do you conclude about the relative performance?

Next exercise on pages 3-4

B. Conceptual modeling: Interactive watch (6 points)

Design the interface for an interactive watch that communicates with a smartphone. The watch has two physical buttons on the side (called Left and Right), a ring that can turn, and a touch screen:



The screen is round and has a resolution of 300x300 pixels. The touch screen can detect only one contact point (because of the small size of the screen) and only the following gestures:

- Tap, Double tap, Long tap,
- Flick in any direction.

The watch has three apps with basic functions:

- Clock Current time, set time zone, set alarm.
- Calendar Browse events, set an alarm for an event.
- Messaging Browse messages, answer message, send message.

When paired with a smartphone, the watch automatically launches the messaging app when a phone call or an SMS is received. This app does not let the user type messages as on a smartphone, but instead lets the user select a message within a predefined list and send it. Some of these messages include a variable part that can be set by the user, for example:

- "I will call you back in <nn> minutes"
- "I am in a meeting"
- "I will be home at <nn>PM"
- …

The calendar app launches automatically to announce a forthcoming event. If the event has an associated contact person, the user can send that person a predefined message that they can complete, such as "I will be <minutes> late. Sorry."

1. Identify the objects and operations of the watch's conceptual model.

2. Design a consistent interface to navigate between apps, to trigger commands and to specify their parameters.

   Use drawings to illustrate your descriptions.

3. Fill out the objects and operations tables to describe the complete conceptual model of the interface. To help you get started, the tables below include one object and one operation (you can change these in your design).

   Objects table:

Objects	Representations	Properties	Operations
Time …	Analog clock	Time of day Timezone	Change timezone

   Operations table:

Operations	Commands	Feedback	Responses
Change timezone …	Flick left or right to get to the timezone screen; Use the ring to adjust time zone; Tap to validate.	Successive settings appear Time changes	 New time flashes

Next exercise on pages 5-6

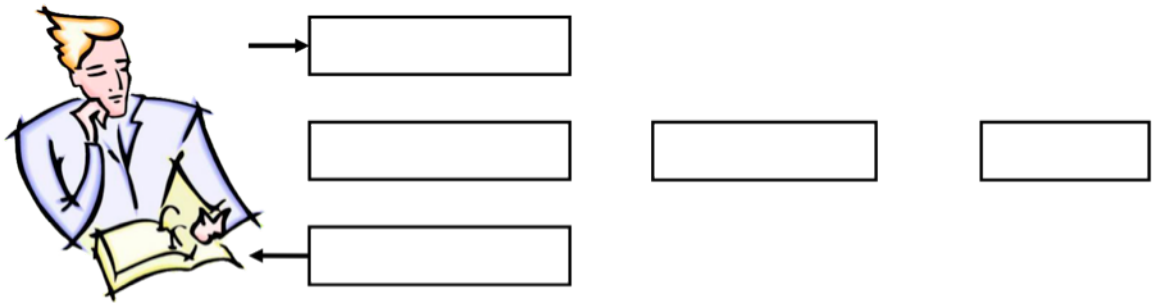
Answer *part C* questions on both sides of this page.

C. Course questions (6 points)

1. Check all that apply to Doug Englebart's NLS/Augment:

[ ] used a mouse
[ ] used a lightpen
[ ] used bimanual interaction
[ ] had videoconferencing
[ ] had windows
[ ] solved geometric constraints

2. Add the missing arrows and labels to this diagram of the *conceptual model*:



3. Cite 4 of the 6 basic interaction tasks, with an example for each of them:

1. _____

_____

2. _____

_____

3. _____

_____

4. _____

_____

4. What are the three senses usually grouped under the sense of touch?

1. _____
2. _____
3. _____

turn over

5. What is a mode? What is a quasi-mode?

6. List and explain two limitations of touch interfaces

1.

2.

7. The notion of affordance was introduced by

[ ] Doug Engelbart [ ] James Gibson [ ] Don Norman

An affordance is

[ ] a property of an object

[ ] the capability of an object for an action

[ ] a relationship between the subject (the user) and the object

[ ] the perceived capability of an object for action

Give an example of an affordance (not necessarily related to computers):